ACKNOWLEDGEMENT

We extent our sincere gratitude to **Dr. Sabeena M V,** Principal of AWH Engineering college for giving us the arena and providing us with all the amenities to do the project.

We are equally indebted to **DR. Namrutha. U**, Asso. Professor, Head Of Department of ECE and project coordinator for their timely suggestions that have helped us in the venture. We express our sincere gratitude to **Ms. Punitha V,** Asst. Prof. ECE Department**,** project guide for their valuable advice and help completing the project. We also express our heartfelt gratitude to all our teaching and non-teaching staff for their guidance for completion of our project.

In course of completion of the project, we were fortunate to receive the assistance of many friends and relatives who were extremely generous with their valuable suggestions, time and energy. We would like to thank all of them and recognize the fact that without them this project would have been inconceivable.

Above all we thank God Almighty for proving us with right atmosphere and mental strength to work and for helping us to make our project a success.

ABSTRACT

The inspiration driving android and voice automated smart wheelchair venture is to construct a smart wheelchair that helps physically impaired people to locomote from one spot then onto the next. To overcome this disability, a smart voice-controlled fully automated wheelchair is designed for physically disabled, patients, or pregnant women. This smart wheelchair will help them move from one place to another without any problem. Numerous wheelchairs are accessible with various running advancements, yet the expense is high and it isn't much successful. For the most part, designing voice and android control wheelchairs is to conquer a few burdens of the current frameworks. The customer needs to interact with the wheelchair with the help of the application. This framework enables the client to vigorously communicate with the wheelchair at various dimensions of the control (turn left, turn right, proceed, return and stop). This task utilizes a microcontroller circuit and motor drivers to make the development of the wheelchair.

Table of Contents

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Since past 20 years, electric-powered wheelchairs have been frequently deployed and developed because it has a pivotal role to improve the quality of life for the elderly and disabled people. The wheelchair is the most ubiquitous equipment used by people with lower limb disability. It enables them some degree of freedom in mobility and independence as opposed to those with both upper and lower limb disabilities. Most of the wheelchairs available in the market are manual in nature with some available with motorized option. Anything beyond that is custom made which is costly and not within the reach of most people. People with severe lower and upper disabilities have to resort to costly electronic controlled wheelchairs or be totally dependent on another person to move them around in their manual wheelchairs. Motorized wheelchairs controlled through joystick, softball, finger, tablet, chin and head are readily available at a high cost but most of them do not cater for those with upper limb disability. The advances in speech recognition technology have made it possible to control any electronics-based device using voice command. This technology is capitalized for voice-controlled wheelchair to assist those with both upper and lower limb disabilities. A variety of voice-controlled wheelchairs have also been developed by other researchers. Recent developments in smart wheelchair have heightened the need for smart control system such as hand-propelled wheelchairs, electrically controlled wheelchairs and automated-guided wheelchairs. This project is developed as an assisted technology in wheelchair development that provides a new method to control the movement of wheelchair using main voice control. There are four movement conditions designed in this controller system; going forward, going backward, going to the right and going to the left. Bluetooth protocol used to communicate between the Android device and the control system. This project is aim to ease the burden for wheelchair user especially for elderly people to move around. The event of an emergency, people may send a message to their guardian through voice command via the application.

CHAPTER 2

LITERATURE SURVEY

Many researchers have focused their work on an accident monitoring system to control road accidents and save lives. Some research is focusing on developing smart phone applications to detect accidents. Some research is in progress in assessing the impact of accidents. Some research focuses on alerting the relevant personnel. When accidents are detected, through some media say SMS or call corresponding people will be intimated. This section of the paper presents some of the work done in accident detection and alerting field.

SL.NO	REPORT NAME & AUTHOR	CONTENTS
1.	ACCIDENT DETECTION AND REPORTING SYSTEM USING GPS, GPRS AND GSM TECHNOLOGY by Md. Syedul Amin, Jubayer Jalil and M. B. I. Reaz Published on IEEE/OSA/IAPR International Conference on Informatics, Electronics & Vision	In this paper the author has predetermined that when the speed is below a specified speed the accident detection mechanism will trigger. But this will cause ineffective triggers when the speed is too low in some conditions.
2.	METHOD OF FREEWAY INCIDENT DETECTION USING WIRELESS POSITIONING by Li Chuan-zhi and Hu Ru-fu and YE Hong-wu Published on Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China September 2008	Its working depends on the principal of Air bags and it is only applicable in Four-wheeler.

3.	ARDUINO BASED ACCIDENT PREVENTION AND AUTO INTIMATION SYSTEM by R. Palanisamy, PLS Sai Kumar, Mekala Paavan Kiran, Ashutosh Mahto, Md. Irfan, Maharishi Bhowmick Published on Indonesian Journal of Electrical Engineering and Computer Science	This paper presents an accident prevention system.it is a collision detection system. It can only detect collision at the hairpin bend on a Hilly track, Ghats, or other Zero visibility turns.
4.	GPS AND MAP MATCHING BASED VEHICLE ACCIDENT DETECTION SYSTEM Md. Syedul Amin, Mohammad Arif Sobhan Bhuiyan and Mamun Bin Ibne Reaz, Salwa Sheikh Nasir 2013 IEEE Student Conference on Research and Development (SCOReD), 16 -17 December 2013, Putrajaya, Malaysia	This system uses map matching algorithm for accident detection

CHAPTER 3

PROPOSED SYSTEM

At present, an electrical wheelchair cannot meet the needs of patients. This project presents the control system of a voice-operated wheelchair. The main objective of this work is to process the voice signal and is implemented to control a wheelchair by the voice signal which is processed earlier.

.

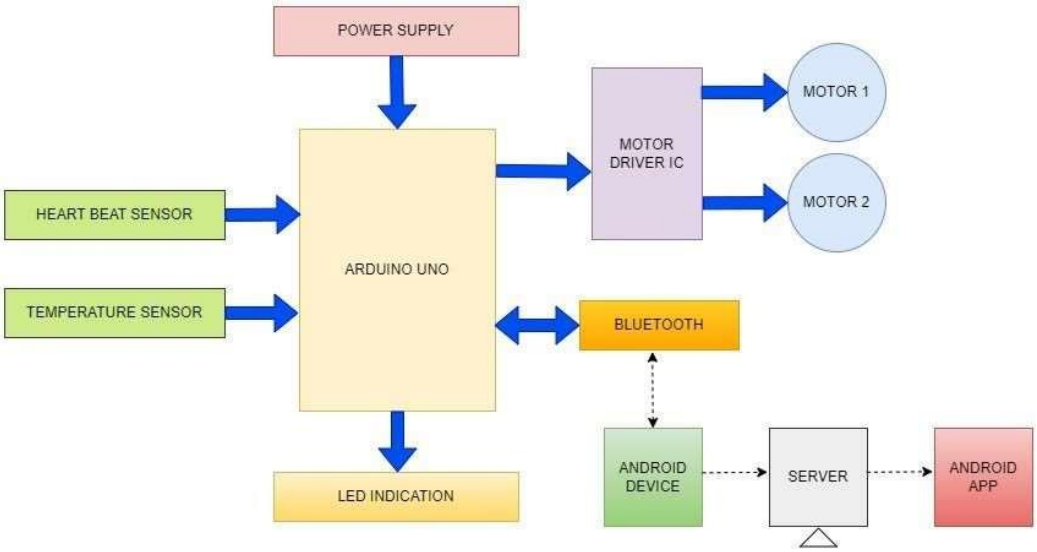


Fig. 1 Block Diagram

Once voice command is recognized, the signal is transferred to the controller where stepper motor module is connected. Then motor driver IC to move the wheels in desired direction. Also measures temperature and heartbeat of the wheelchair user and send through the telegram bot. So that required person will get the reading whenever they want. The concept developed in this project are proven convenience innovation and provides better benefits to the targeted market and public interest.

3.1 Arduino uno:

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53- R0), 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. You can tinker with your Uno without worrying too much about doing something wrong, worst-case scenario you can replace the chip for a few dollars and start over again. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Power Jack: Arduino can be power either from the pc through a USB or through external source like adaptor or a battery. It can operate on a external supply of 7 to 12V. Power can be applied externally through the pin Vin or by giving voltage reference through the IORef pin.

Digital Inputs: It consists of 14 digital inputs/output pins, each of which provide or take up 40mA current. Some of them have special functions like pins 0 and 1, which act as Rx and Tx respectively, for serial communication, pins 2 and 3-which are external interrupts, pins 3,5,6,9,11 which provides pwm output and pin 13 where LED is connected.

Analog inputs: It has 6 analog input/output pins, each providing a resolution of 10 bits.

ARef: It provides reference to the analog inputs.

Reset: It resets the microcontroller when low.

Programs written in Arduino are known as sketches. A basic sketch consists of 3 parts

1. Declaration of Variables
2. Initialization: It is written in the setup () function.
3. Control code: It is written in the loop () function.

- The sketch is saved with. no extension. Any operations like verifying, opening a sketch, saving a sketch can be done using the buttons on the toolbar or using the tool menu.
- The sketch should be stored in the sketchbook directory.
- Chose the proper board from the tool's menu and the serial port numbers.
- Click on the upload button or chose upload from the tool's menu. Thus, the code is uploaded by the bootloader onto the microcontroller.

Fig. 2 Arduino uno

Arduino uno internal architecture

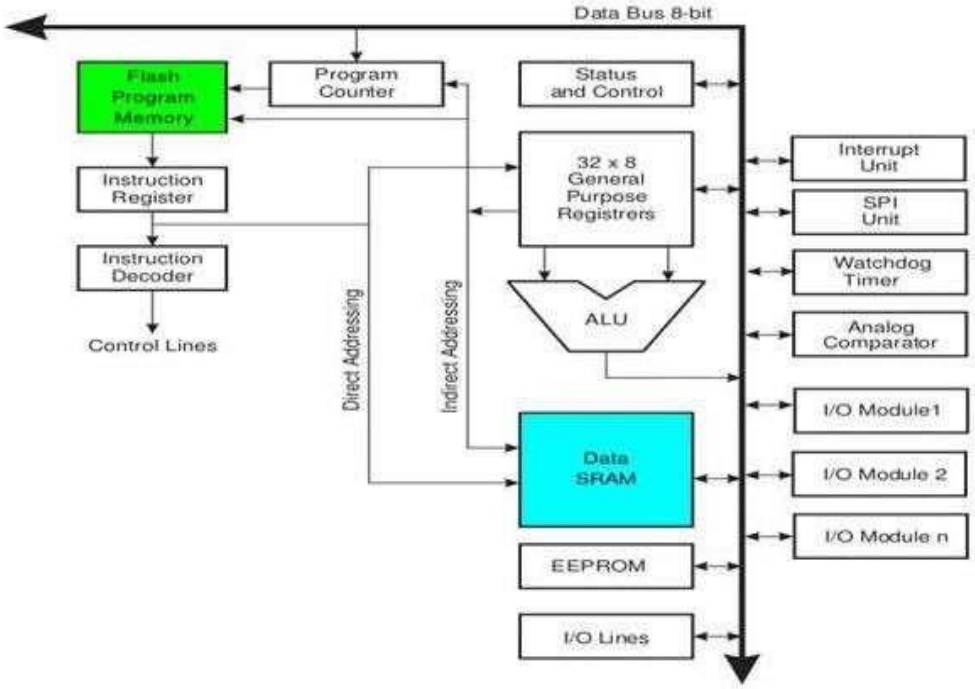


Fig. 3 Uno internal architecture

Arduino's processor basically uses the Harvard architecture where the program code and program data have separate memory. It consists of two memories- Program memory and the data memory. The code is stored in the flash program memory, whereas the data is stored in the data memory. The Atmega328 has 32 KB of flash memory for storing code (of which 0.5 KB is used for the bootloader), 2 KB of SRAM and 1 KB of EEPROM and operates with a clock speed of 16MHz.

3.2 Temperature sensor (DALLAS):

The DS18B20 temperature sensor is a one-wire digital temperature sensor. This means that it just requires one data line (and GND) to communicate with the Arduino. It can be powered by an external power supply or it can derive power from the data line (called "parasite mode"), which eliminates the need for an external power supply.

DS18B20	Arduino
GND	GND
DQ	Any digital pin (with 4.7k Ohm pull-up resistor)
VDD	5V (normal mode) or GND (parasite mode)

- Communicates over one-wire bus communication
- Power supply range: 3.0V to 5.5V
- Operating temperature range: -55ºC to +125ºC
- Accuracy +/-0.5 ºC (between the range -10ºC to 85ºC)

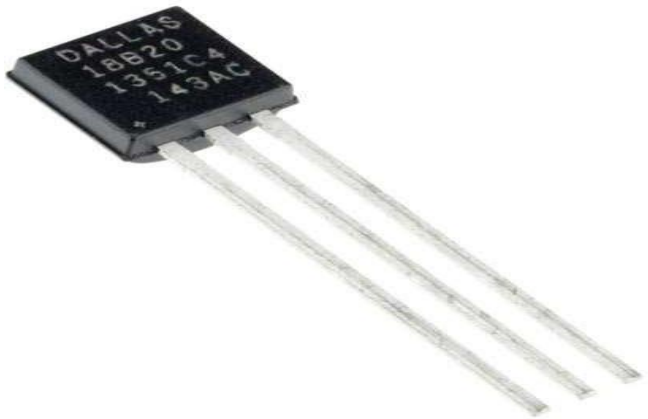


Fig. 4 DS18B20 Temperature sensor

Pin configuration

Each DS18B20 temperature sensor has a unique 64-bit serial code. This allows you to wire multiple sensors to the same data wire. So, you can get temperature from multiple sensors using just one Arduino digital pin

3.3 Heartbeat Sensor / Pulse Sensor:

A pulse wave is the change in the volume of a blood vessel that occurs when the heart pumps blood, and a detector that monitors this volume change is called a pulse sensor. Pulse sensor on the fingertip that measures the changes in light absorption and reflection onto the skin to measure blood flow. This sensor clips on an earlobe or a fingertip.

Fig. 5 Heartbeat Sensor

3.4 Driver IC:

The Raspberry Pi processor output is not sufficient to drive the DC motors, a high Voltage and high current drivers are required. The L293D is a quadruple high current half H-driver designed to provide bidirectional drive currents of up to 600Ma at voltages from 4.5V to 36V. It will become easier to drive DC motors with such drivers. Drivers are Enabled in pairs with drivers1and2 enabled by 1, 2EN and drivers 3 and 4 are enabled by 3, 4EN. When an enable input is high, the associated drivers are enabled, their outputs are Active and in-phase with their input

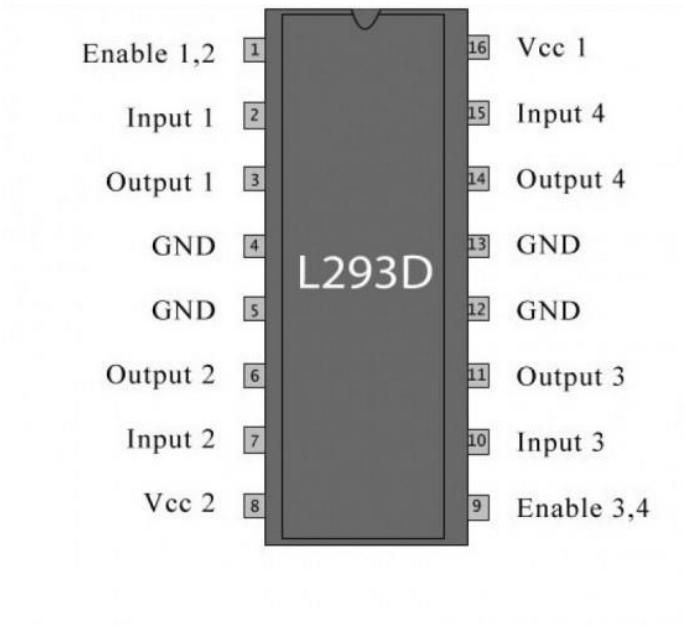


Fig. 6 Driver IC

3.5 Motor:

Gear motors are used in applications that require lower shaft speed and higher torque output. This describes a wide range of applications and scenarios, including many of the machines and equipment we interact with daily. From ATV wipers to hospital beds, servo mechanisms to packaging equipment, paint mixers to juice

dispensers, gear motors are used to power a significant number of machines and applications. Simply put, a gear motor is any electric motor coupled with a gear train.

Gear Motors use either AC (Alternating Current) or DC (Direct Current) power. In most cases, the addition of a gear box is intended to limit the speed of the motor's shaft and increase

The motor's ability to output torque. Gears transform shaft speed into torque at specific ratios, with minimum efficiency losses, making it possible to create the ideal Torque output and speed by adding the appropriately sized and configured gear box.

Fig. 7 Motor

3.6 Bluetooth Module (HC05):

HC05 module is a Bluetooth module using serial communication, which is an affordable communication method in PAN network, with a maximum data rate of 1Mb/S, working in a nominal range of 100 meters using 2.4 G frequency is a common way of wireless communicating. HC05 module has an internal 3.3v regulator but In order to prevent the module from damages and for working properly, you should use a resistance division circuit (5v to 3.3v) between Raspberry pi TX pin and module RX pin. When master and slave are connected, blue and red LEDs on the board blink every 2seconds. If they aren't connected, only blue one blinks every 2 seconds.

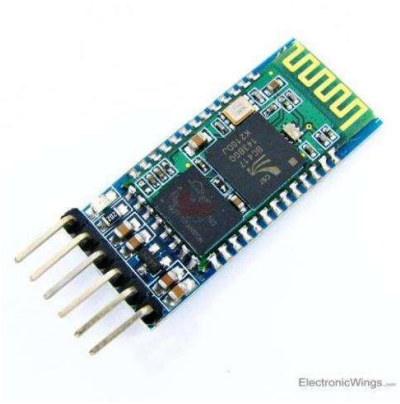


Fig. 8 Bluetooth Module (HC05)

3.7 Power supply

The input to the circuit is applied from the regulated power supply. The ac. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating dc voltage. So, to get a pure

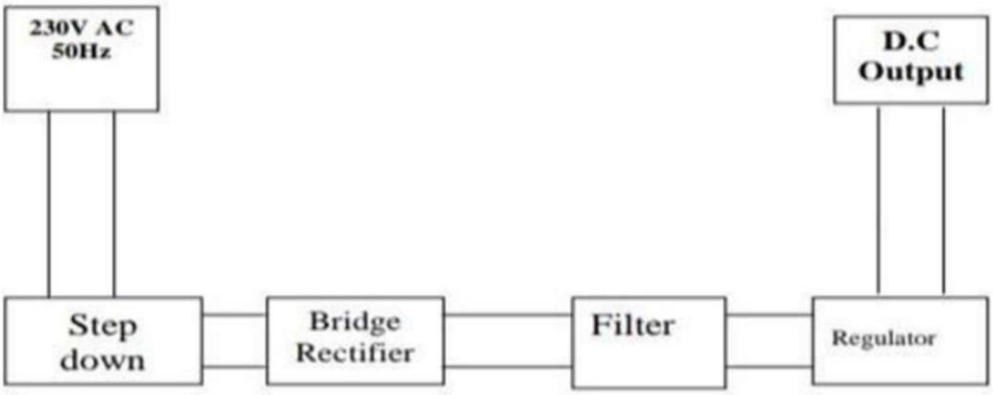


Fig 9. Power supply

dc voltage, the output voltage from the rectifier is fed to a filter to remove any ac components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage. Power supply is a device or system that supplies electrical or other types of energy to an output load or group of loads

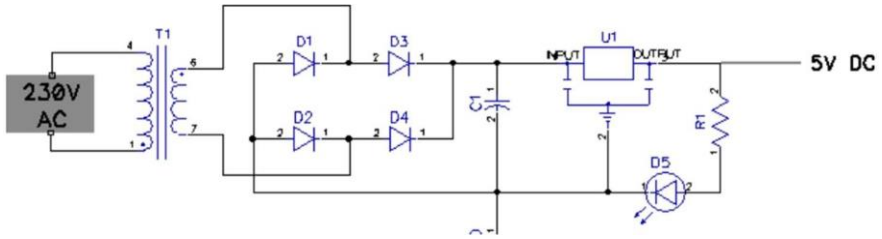


Fig 10. Circuit diagram of power supply

A simple AC powered linear power supply usually uses a transformer to convert the voltage from the wall outlet (mains) to a different, usually a lower voltage. If it is used to produce DC a rectifier circuit is employed either as a single chip, an array of diodes sometimes called a diode bridge or Bridge Rectifier, both for full wave rectification or a single diode yielding a half wave (pulsating) output. More elaborate configurations rectify the AC voltage at first to pulsating DC. Then a capacitor smooths out part of the pulses giving a type of DC voltage. The smaller pulses remaining are known as ripple. Because of a full wave rectification, they occur at twice the mains frequency. Finally depending on the requirements of the load, a linear regulator may be used to reduce the ripple sometimes also allowing for adjustment of the output to the desired but lower voltage.

3.8 Rectifier

The output from the transformer is fed to the rectifier. It converts A.C. into pulsating DC. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used because of its merits like good stability and full wave rectification.

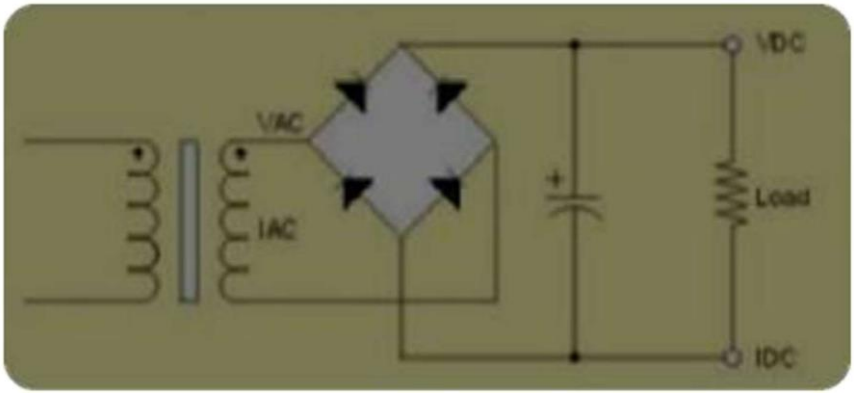


Fig 11. Circuit diagram of Bridge rectifier

The Bridge rectifier is a circuit, which converts an ac voltage to dc voltage using both half cycles of the input ac voltage. The Bridge rectifier circuit is shown in the figure. The circuit has four diodes connected to form a bridge. The ac input voltage is applied to the diagonally opposite ends of the bridge. The load resistance is connected between the other two ends of the bridge. For the positive half cycle of the input ac voltage, diodes D1 and D3 conduct, whereas diodes D2 and D4 remain in the OFF state. The conducting diodes will be in series with the load resistance RL and hence the load current flows through RL.

For the negative half cycle of the input ac voltage, diodes D2 and D4 conduct whereas, D1 and D3 remain OFF. The conducting diodes D2 and D4 will be in series with the load resistance RL and hence the current flows through RL in the same direction as in the previous half cycle. Thus, a bi-directional wave is converted into a unidirectional wave

3.9 Filter

Capacitive filter is used in this project. It removes the ripples from the output of rectifier and smoothens the D.C. Output received from this filter is constant until the mains voltage and load is maintained constant. However, if either of the two is varied, D.C. voltage received at this point changes. Therefore, a regulator is applied at the output stage.

3.10 Voltage Regulator

As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this project, power supply of 5V and 12V are required. In order to obtain these voltage levels, 7805 and 7812 voltage regulators are to be used.

The voltage regulator is a device, which maintains the output voltage constant irrespective of the change in supply variations, load variations and temperature variations. Regulator IC units contain the circuitry for reference source, comparator, amplifier, control device and overload protection, all in a single IC. Other features are, Output Current up to 1A, Output Voltages of 5v, Thermal Overload Protection, Short Circuit Protection, Output Transistor Safe Operating Area Protection.

The 78xx (sometimes LM78xx) is a family of self-contained fixed linear voltage regulator integrated circuits. The 78xx family is commonly used in electronic circuits requiring a regulated power supply due to their ease-of-use and low cost. For ICs within the family, the first number 78 represents positive supply and the xx is replaced with two digits, indicating the output voltage. 78xx ICs have three terminals and are commonly found in the TO220 form factor, although smaller surface-mount and larger TO3 packages are available. These devices support an input voltage anywhere from a couple of volts over the intended output voltage, up to a maximum of 35 to 40 volts depending on the make, and typically provide 1 or 1.5 amperes of

current. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1 A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltage and currents. These are monolithic integrated circuits designed as fixed voltage regulators for a wide variety of applications including local, on card regulation. These regulators employ internal current limiting, thermal solution and safe area compensation. They can also be used with external components to obtain adjustable voltages and current. Its features are,

• Output Current up to 1A

• Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24

• Thermal Overload Protection

• Short Circuit Protection

• Output Transistor Safe Operating Area Protection

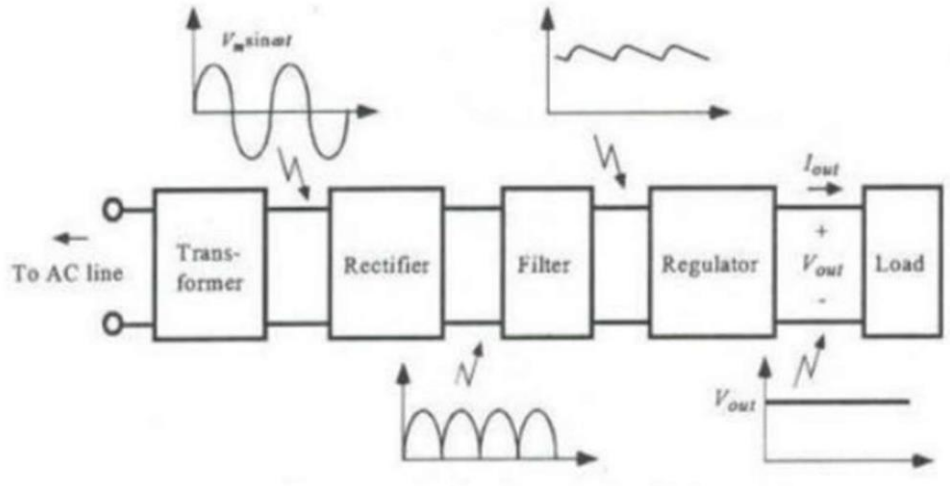


Fig 12. power supply Block diagram

3.11 Resistors

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. Resistors act to reduce current flow, and, at the same time, act to lower voltage levels within circuits. In electronic circuits, resistors

are used to limit current flow, to adjust signal levels, bias active elements, and terminate transmission lines among other uses. High-power resistors that can dissipate many watts of electrical power as heat may be used as part of motor controls, in power distribution systems, or as test loads for generators. Fixed resistors have resistances that only change slightly with temperature, time or operating voltage. Variable resistors can be used to adjust circuit elements (such as a volume control or a lamp dimmer), or as sensing devices for heat, light, humidity, force, or chemical activity.

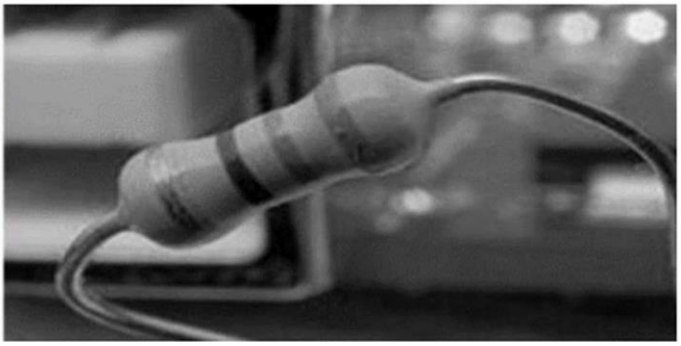


Fig 13. Resistor

3.12 Capacitors

A capacitor (originally known as a condenser) is a passive two-terminal electrical component used to store electrical energy temporarily in an electric field. The forms of practical capacitors vary widely, but all contain at least two electrical conductors (plates) separated by a dielectric (i.e. an insulator that can store energy by becoming polarized). The conductors can be thin films, foils or sintered beads of metal or conductive

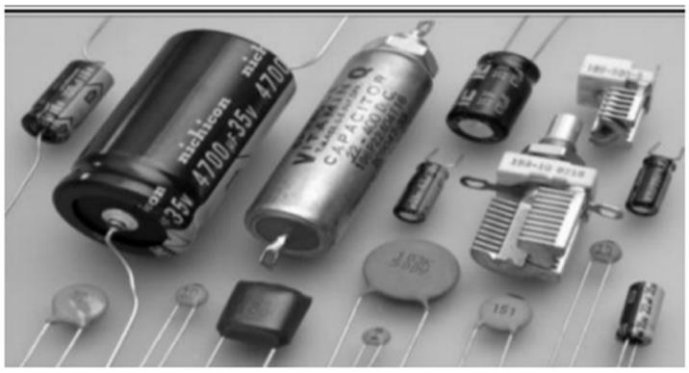


Fig 14. Capacitor

electrolyte, etc. The no conducting dielectric acts to increase the capacitor's charge capacity. A dielectric can be glass, ceramic, plastic film, air, vacuums, paper, mica,

oxide layer etc. Capacitors are widely used as parts of electrical circuits in many common electrical devices. Unlike a resistor, an ideal capacitor does not dissipate energy. Instead, a capacitor stores energy in the form of an electrostatic field between its plates.

3.13 Diode (1N4007)

Fig 15. Diode

1N4007 is a rectifier diode, designed specifically for circuits that need to convert alternating current to direct current. It can pass currents of up to 1 A, and have peak inverse voltage (PIV) rating of 1,000 V.

Features:

- Maximum Recurrent Peak Reverse Voltage 1000V
- Maximum RMS Voltage 700V
- Maximum DC Blocking Voltage 1000V
- Average Forward Current: 1.0A
- Peak Forward Surge Current: 30A
- Maximum Instantaneous Forward Voltage:1.0V
- Maximum DC Reverse Current at Rated DC Blocking Voltage: 5.0μA @ 25°C
- Typical Junction Capacitance: 15pF
- Typical Reverse Recovery Time: 2.0us
- Mounting Type: Through Hole
- Operating Temperature: -55°C ~ 150°C

3.14 LED

A light-emitting diode (LED) is a two-lead semiconductor light source. It is a p–n junction diode, which emits light when activated. When a suitable voltage is applied to the leads, electrons can recombine with electron holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the color of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor.

Fig 16. LED

Voltage specification

Featuring Unit rode L293 and L293D, Wide Supply Voltage Range: 4.5V to 36V. Separate Input-Logic Supply, Internal ESD Protection, Thermal Shut down, High- Noise Immunity Inputs, Functionally Similar to SGS L293 and SSGS L293D Output Current 1A Per Channel (600mAforL293D). Peak Output Current 2 A Per Channel (1.2AforL293D). Output Clamp Diodes for Inductive Transient Suppression (L293D), description/ordering information.

CHAPTER 4

CIRCUIT DIAGRAM

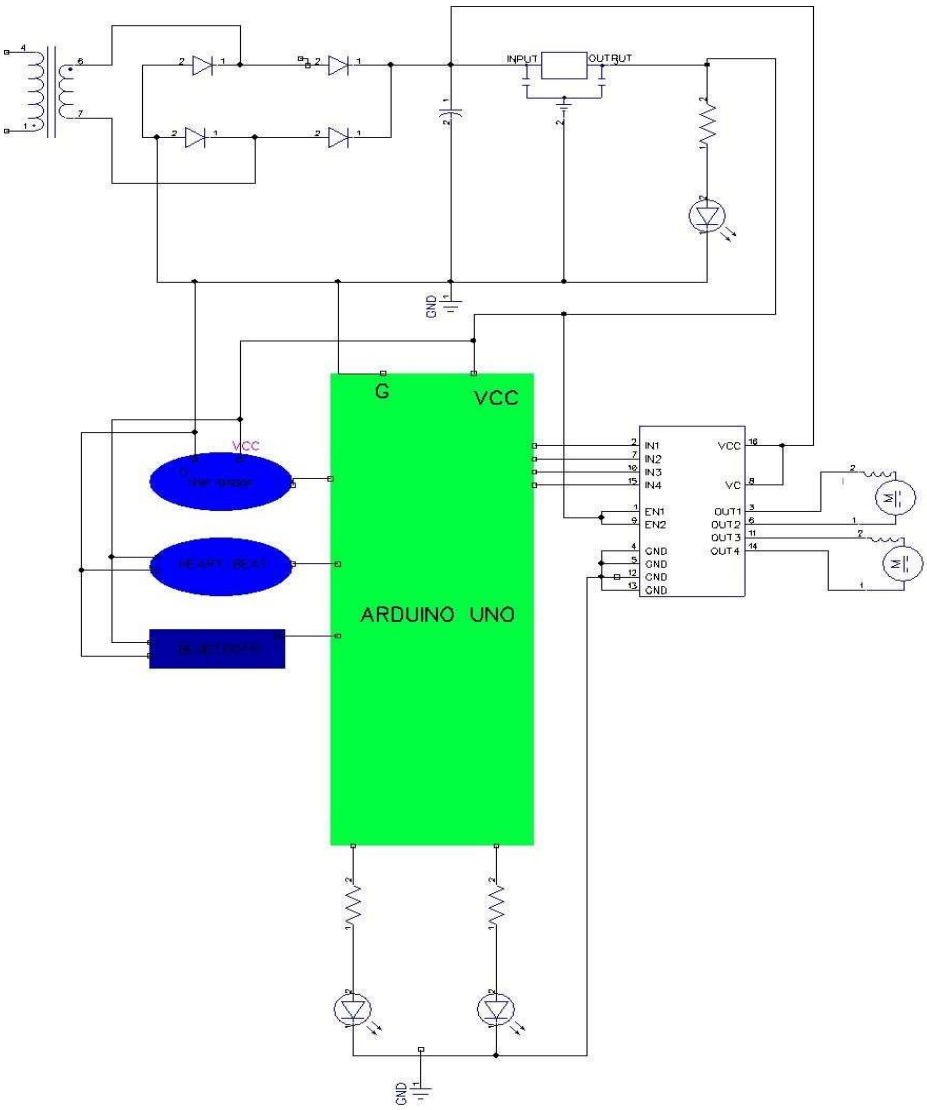


Fig. 17 Circuit Diagram

.

Here we used Arduino Uno as the micro-controller as the main part of the circuit. All the input of the various components are taken from the Arduino Uno. Mainly three sensors have been used. The DS18B20 temperature sensor is a one-wire digital temperature sensor. A pulse wave is the change in the volume of a blood vessel that occurs when the heart pumps blood, and a detector that monitors this volume change is called a pulse sensor. HC05 module is a Bluetooth module using serial communication, which is an affordable communication method in PAN network, with a maximum data rate of 1Mb/S, working in a nominal range of 100 meters using 2.4 G frequency is a common way of wireless communicating. The above part of the circuit defines the power supply, in which step-down transformer is used to lower down the voltage, and a Bridge rectifier comes after it, and the it comes to the voltage regulator, after that it comes to the led. When it reaches the last its voltage will be reduced into 5V. Here the capacitor is used for filtering. whole this power supply is connected to the VCC of the Arduino Uno. The input pin of the Driver IC is connected to the Arduino Uno. The two motors are connected to the Driver IC. The working will start after switch on the Arduino Uno

4.1 Working Principle:

This project is aim to ease the burden for wheelchair user especially for elderly people to move around. It Controls the DC motor, communicates with the Android phone application through a Bluetooth communication. Rapid development tool connected with one main control interface using voice command. Platform based on four conditions; forward, backward, right and left. The speed movement is controlled by Pulse Width Modulation (PWM) signal while the direction movement guided by an electronic compass in the Android device., the concept developed in this project are proven convenience innovation and provides better benefits to the targeted market and public interest. The powered wheel chair depends on motors for locomotion and voice recognition for command. Voice control, also called voice assistance, is a user interface that allows hands-free operation of a digital device. Voice control does not require an internet connection to work. Communication is one way (person to device) and all processing is done locally. The enables a disabled person to move around independently, using a voice recognition application which is interfaced with motors. The prototype of

the wheelchair is built using a micro-controller, chosen for its low cost, in addition to its versatility and performance in mathematical operations and communication with other electronic devices. The system has been designed and implemented in a cost- effective way so that if our project is commercialized the needy users in developing countries will benefit from it. Generally, micro-controller generates codes in its output according to the input. So, using this module we can control the dc motor corresponding to the voice command.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 DIPTRACE

Dip Trace is EDA/CAD software for creating schematic diagrams and printed circuit boards. The developers provide multi-lingual interface and tutorials (currently available in English and 21 other languages). Dip Trace has 4 modules: Schematic Capture Editor, PCB Layout Editor with built-in shape-based auto-router and 3D Preview & Export, Component Editor, and Pattern Editor.

Dip Trace is an advanced PCB design software application that consists of 4 modules PCB Layout with efficient auto-router and auto-placer, schematic capture, component and pattern editors that allow you to design your own component libraries. Dip Trace has a powerful automatic router, superior to many routers included in other PCB layout packages. It can route a single layer and multilayer circuit boards, and there is an option to auto route a single layer board with jumper wires, if required. Dip Trace also provides you with external auto router support. Smart manual routing tools allow users to finalize the design and to get the results they want in a blink of an eye. There are number of verification features, that allows you to control accuracy of your project. Dip Trace modules allow you to exchange schematics, layouts and libraries with other EDA and CAD packages. Output formats are DXF, Gerber, Drill and G-code. Standard libraries contain more than 98,000 components.\

Basic Features

• Simple UI

• Multi-sheet and hierarchical schematics

• High-speed shape-based autoroute

• Smart manual routing tools

• Differential pairs

• Wide import / export capabilities

• Advanced verifications with real-time DRC

• Real-time 3D PCB preview & STEP export

• ODB++ and Gerber manufacturing outputs.

5.2 Schematic Capture

Advanced circuit design tool with support of multi-sheet and multi-level hierarchical schematics that delivers a few features for visual and logical pin connections. Cross-module management ensures that principal circuits can be easily converted to PCB, back annotated, or imported/exported from/to other EDA, CAD formats and net lists. Dip Trace Schematic has ERC Verification and Spice export for external simulation.

5.3 PCB Layout

Engineering tool for board design with smart manual routing, differential pairs, shape-based autoroute, advanced verification, and wide import/export capabilities. Design requirements are defined by net classes, class-to-class rules, and detailed settings by object types for each class or layer. When routing with real-time DRC, the program reports errors on the fly before actually making them. DRC also checks length and phase tolerances for differential pairs. The board can be previewed in 3D and exported to STEP format for mechanical CAD modeling. Design Rule Check with in- depth detailing and Net Connectivity verification procedures are available

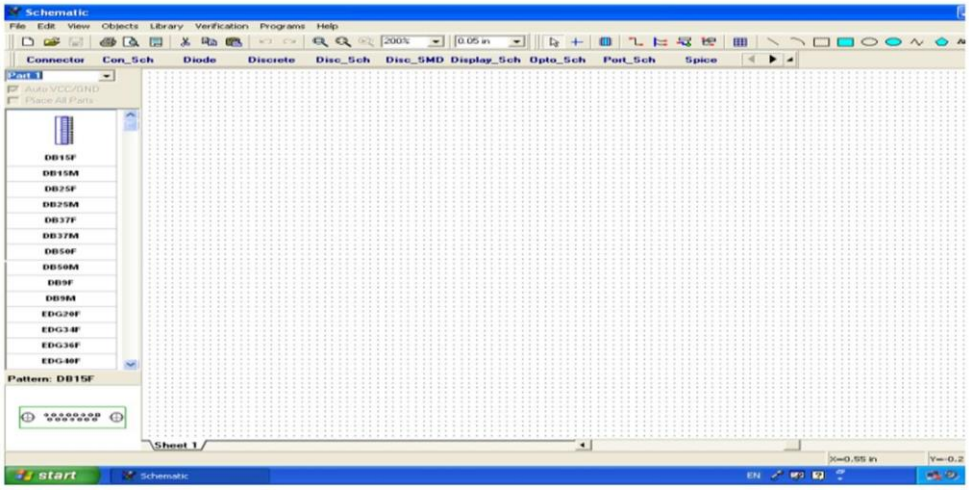
5.4 3D Preview and Export

This module includes real-time 3D preview & export feature. It shows the model of manufactured printed circuit board with all components installed. Rotate board in three axes, zoom in and out in real time, change colors of the board, copper areas, solder mask, silkscreen, and background. 3D preview works on all stages of the design. Board can be exported to STEP or VRML 2.0 formats for mechanical CAD modelling. More than 6500 3D models of PCB packages are supplied for free. Externally designed 3D models in *.wrl,*.step, *.iges, and *.3ds formats can be uploaded and attached to patterns in Pattern Editor or PCB Layout.

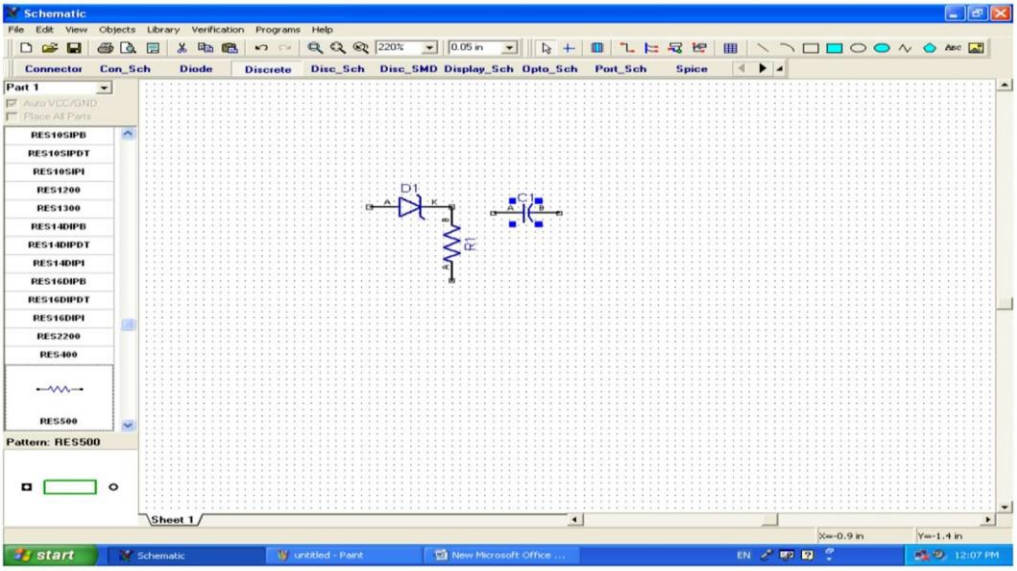
5.5 Pattern Editor

Draw patterns with various types of shapes, pads, holes, and dimensions. Circle, Lines (headers, DIP), Square (QFP), Matrix (BGA), Rectangle (RQFP), and Zig-Zag standard templates. Creation of pattern is basically selecting a template, entering a couple of vital parameters, drawing the silkscreen, and launching automatic pad renumbering. Custom templates can be created for non-standard patterns. DXF import makes creating complex layouts easier.

a) open diptrace->select schematic

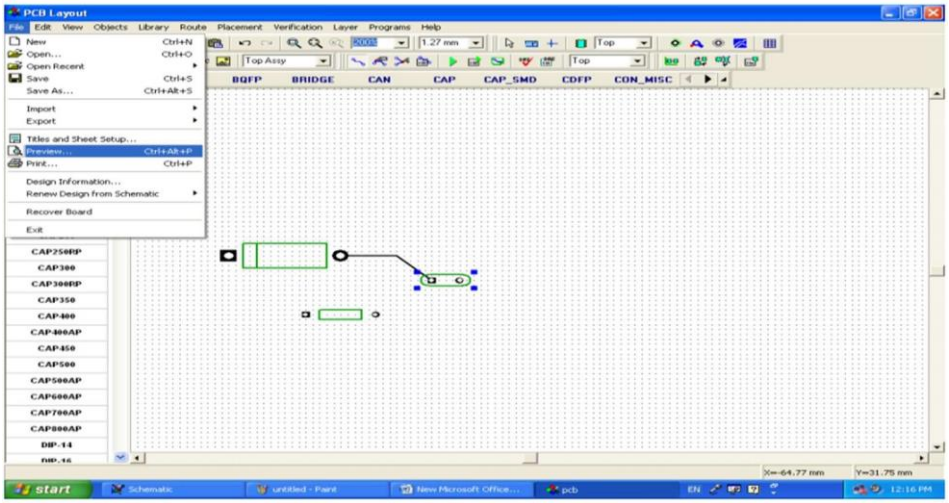


b) pick and place components

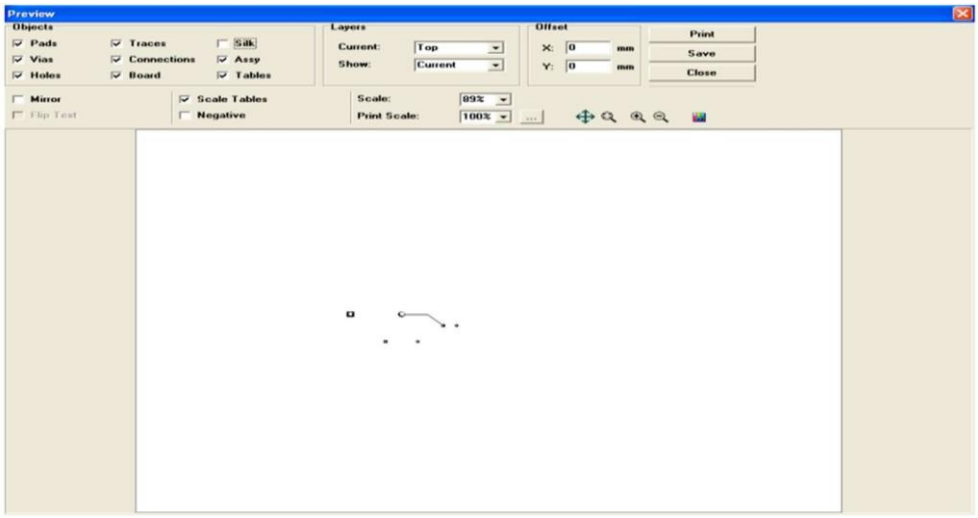


c) file->convert to PCB

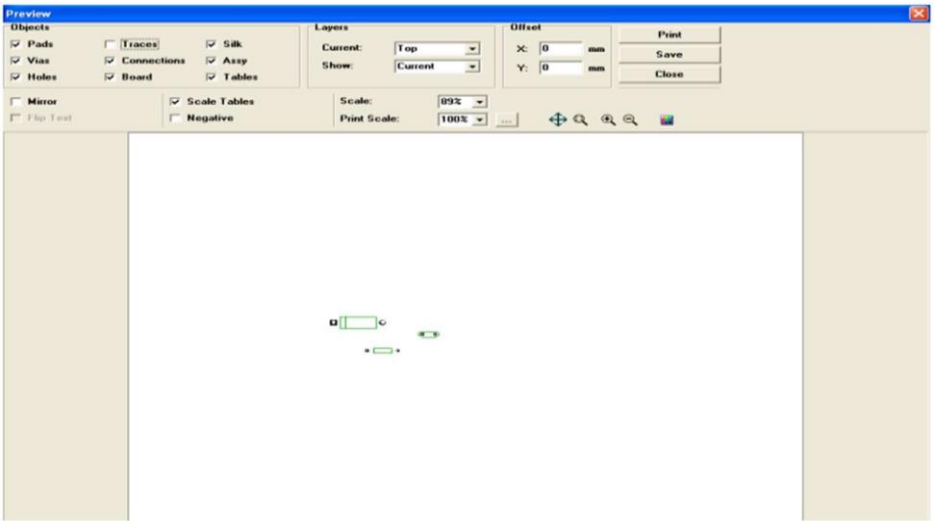
d) route correctly Then file - > preview



e) click silk to show layout



f) click traces to show component



5.6 PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas the other languages use punctuations. It has fewer syntactical constructions than other languages. Python features a dynamic type of system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. Python, the reference implementation of Python, is open-source software and has a community- based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit Python Software Foundation. Python is a multi-paradigm programming language: object-oriented programming and structured programming are fully supported, and many language features support functional programming and aspect-oriented programming (including by metaprogramming and meta objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a mix of reference counting and a cycledetecting garbage collector for memory management. An important feature of Python is dynamic name resolution (late binding), which binds method and variable names during program execution. The design of Python offers some support for functional programming in the Lisp tradition. The language has filter(), map(), and reduce() functions; list comprehensions, dictionaries, and sets; and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

Python Features Python's features include:

• Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax. This allows a student to pick up the language quickly.

• Easy-to-read: Python code is more clearly defined and visible to the eyes.

to- maintain: Python's source code is fairly easy-to-maintain.

• A broad standard library: Python's bulk of the library is very portable and cross platform compatible on UNIX, Windows, and Macintosh.

• Interactive Mode: Python has support for an interactive mode, which allows interactive testing and debugging of snippets of code.

• Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

• Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

• Databases: Python provides interfaces to all major commercial databases.

• GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

• Scalable: Python provides a better structure and support for large programs than shell scripting

Python has five standard data types

• Numbers

• String

• List

• Tuple

• Dictionary

Types of Operators Python language supports the following types of operators

• Arithmetic Operators

• Comparison (Relational) Operators

• Assignment Operators

• Logical Operators

• Bitwise Operators

• Membership Operators

• Identity Operators64

5.7 MySQL

MySQL is the world's most popular open-source database software, with over 100 million copies of its software downloaded or distributed throughout its history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance, and administration for modern, online applications. Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software — including industry leaders such as Yahoo!, Alcatel- Lucent, Google, Nokia, Wikipedia, and Booking.com. The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production tested software, proactive monitoring tools, and premium support services available in an affordable annual subscription.

FEATURES OF MYSQL

• Ease of use—Go from download to complete installation in less than 15 minutes Low TCO—Deploy MySQL for mission-critical applications with significant cost saving over Microsoft SQL Server.

• Scalability and performance Meet the scalability and performance requirements of the most trafficked web sites and the most demanding applications.

• Production support—Oracle Premier Support helps lower the total cost and risk of owning you‘re mySQL solutions

• MySQL is a database management system. Smart Vehicle Safety System Project Report 2021- 2022 Dept. of Electronics 48 St. Thomas College (Autonomous), Thrissur

• MySQL databases are relational.

• MySQL software is Open Source.

• The MySQL Database Server is very fast, reliable, scalable, and easy to use.

5.8 PYCHARM

PyCharm is the most popular IDE used for Python scripting language. It is developed by the Czech company jetBrains.it provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems and supports web development with Django.

PYCHARM FEATURES:

PyCharm offers some of the best features to its users and developers in the following aspects:

• Code completion and inspection Advanced debugging Support for web programming and frameworks such as Django and Flask Features of PyCharm Besides, a developer will find PyCharm comfortable to work with because of the features mentioned below:

• Code Completion PyCharm enables smoother code completion whether it is for built in or for an external package.

• SQLAlchemy as Debugger You can set a breakpoint, pause in the debugger and can see the SQL representation of the user expression for SQL Language code.

• Git Visualization in Editor When coding in Python, queries are normal for a developer. You can check the last commit easily in PyCharm as it has the blue sections that can define the difference between the last commit and the current one.

• Code Coverage in Editor You can run .py files outside PyCharm Editor as well marking it as code coverage details elsewhere in the project tree, in the summary section etc.

CHAPTER 6

PCB DESIGNING

6.1 BASIC STEPS IN PCB MANUFACTURING

Forming a printed circuit board is essential and the most prominent step in the formation of one electronic device. For a device to work properly the components we planned to use should be well placed in a PCB. In this section we are explaining about the formation of our PCB. The design of a PCB can be considered as the last step in electronic circuit designing. In the electronic circuit performance and reliability depends on the productivity of PCB. Assembling and servicing ability also depends on the design. A proper PCB ensures that various components are interconnected as per the circuit diagram. Once they have been placed on the PCB in their proper positions and subsequently soldered PCB design and fabrication techniques have undergone so much of development that it has become a subject. Double sided PCBs, multiplayer PCBs with plated through holes (PTH), flexible PCBs, etc. are only some of the developments. Manufacturing of PCB involves the following steps.

1 Print and etch

2 plate and etch

The single sided PCBs are usually using the print and etch method and the double-sided plates through hole boards are made by print, plate and etch method. The production of multiplayer boards uses both the technique

6.2 Penalization

The schematic or the artwork of this circuit applied by the customer is transformed to working positive or negative films. The circuit is repeated conveniently to accommodate economically as many circuits as possible in a panel, which can be operated in every sequent step in the PCBs process. This is called penalization.

6.3 Drilling

This is the state-of-the-art operation. Very small holes are drilled with a highspeed CMC drilling machine.

6.4 Plating

The heart of the PCB manufacturing process lies in the electrolytic plating process. The holes drilled are treated both mechanically and chemically before depositing the copper by the electrolytic copper plating process

6.5 Etching

Once a multiplayer board is drilled and electro less copper is deposited the image available in the form of a firm is transferred on to the outside by photo printing process. The boards are then with copper and tin. This is called etching.

6.6 Solder mask

Since PCB design may call for very close spacing conductors, a solder mask has to be applied on both sides of the circuit to avoid bridging of conductors. This ink is applied by screening. This is dried, exposed to UV, developed in a mild alkaline solution, and finally treated by both UV and thermal energy.

6.7 Hot air leveling

After the above-mentioned process, the circuit pads are soldered using hot air leveling process while removing the board from solder path, hot air blown on both the sides of the board through air knives in the machine leaving the board soldered and leveled.

Fabrication of demonstration unit

1 The total circuit diagram and list of components are prepared, and procurement of the components is done.

2 The components layout and interconnection track diagram are prepared and hole drilling as per the size of the components is done

3 To remove the unwanted copper other than the track part the board is etched, and it is washed with plenty of water and is dried well

CHAPTER 7

PCB LAYOUT

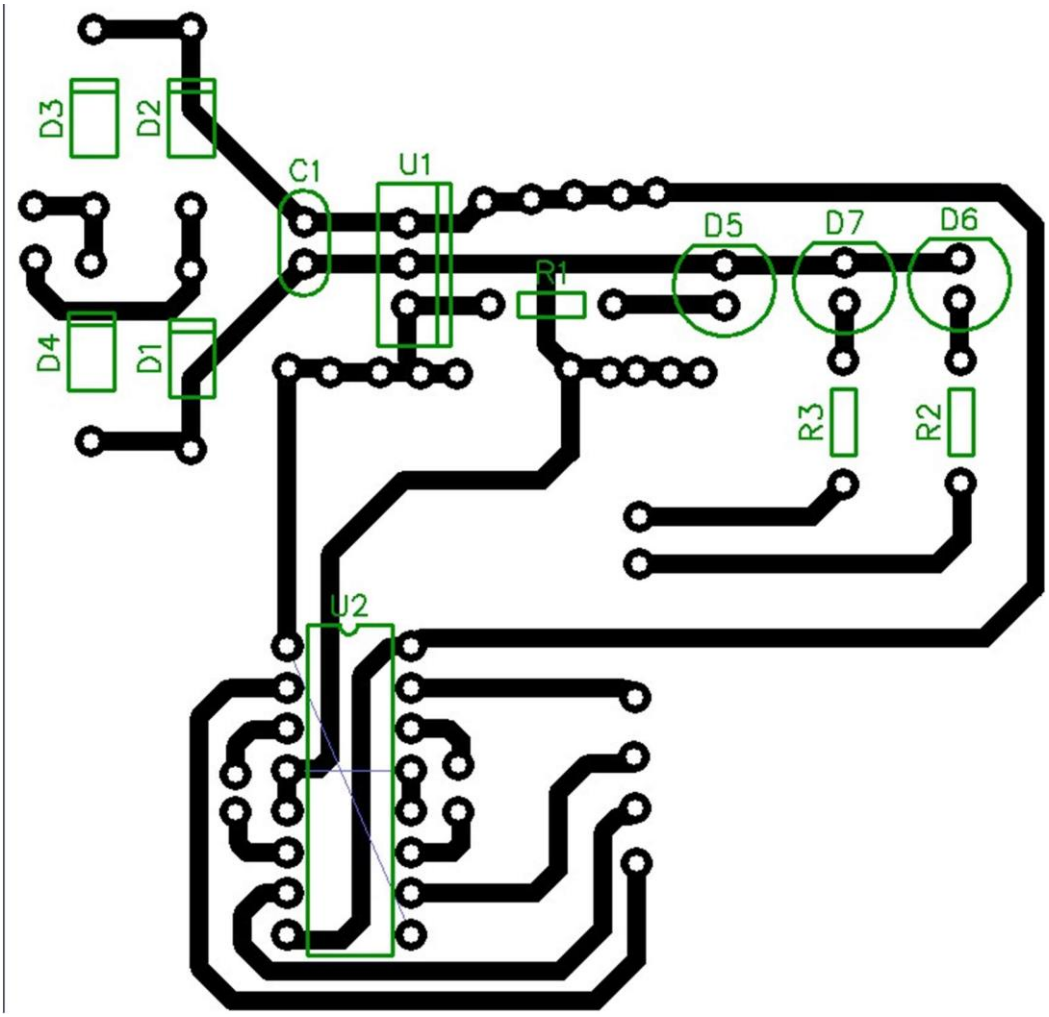


FIG 7 PCB LAYOUT

CHAPTER 8

RESULT

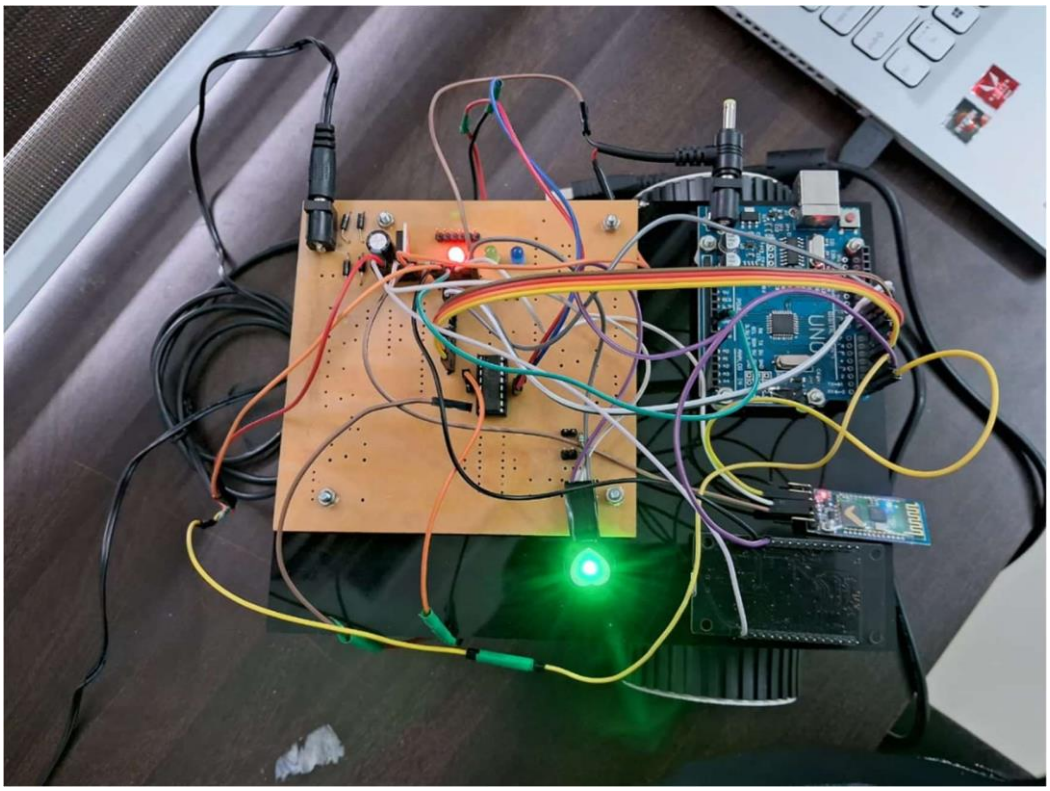


FIGURE 8.1 RESULT

Our device is designed to improve the comfort and convenience of individuals with cerebral palsy by incorporating automation and ergonomics principles. One of the key features of our system is the storage mechanism, which allows us to save and access data for each disabled individual using a microcontroller (Arduino). Let's consider an example with a person named 'A' who has cerebral palsy. When person A sits on the chair for the first time, we manually record their preferred sitting position. The chair is equipped with four movements, three of which are controlled by servo motors, while the fourth movement is handled by a DC motor. The backrest of the chair can be adjusted to a comfortable angle for person A by making changes in the code and programming the servo motor accordingly. Similarly, the side support, also controlled by a servo motor, can be adjusted to the appropriate angle to prevent excessive leaning or falling to the sides. The neck rest position can also be adjusted using the same servo motors to protect the neck from strain. Additionally, the footrest can be adjusted using

the DC motor, which moves it upward until the person's foot comes in contact with an infrared (IR) sensor integrated into the footrest. Once all the manual adjustments are made during the initial setup, the chair can automatically adjust to person A's preferred position by simply pressing a designated button assigned to them. Importantly, all three movements can be adjusted simultaneously when the button is pressed. We currently have the capability to store the data of three individuals (person A, person B, person C) at a time, allowing easy access to the settings of each person. The LCD display on the chair shows the name of the person whose position is currently being used. In case of an emergency or the need for assistance, an emergency button is provided for easy communication. When pressed, the button activates a buzzer, alerting others to the individual's need for help. For instance, if a person requires water, they can press the button on the table, and by hearing the buzzer sound, we can quickly understand their request and make communication easier and more efficient. Overall, our device combines automation, ergonomics, and personalized settings to provide improved comfort and convenience for individuals with cerebral palsy. The storage mechanism allows for easy access to individual preferences, while the emergency button enhances communication and ensures prompt assistance when needed.

8.1 PROGRAMS

NODE MCU PROGRAM

```
#include<ESP8266WiFi.h>

#include<ESP8266HTTPClient.h>

#include<HttpClient.h>

#include<WiFiClient.h>

#include <OneWire.h>

#include <DallasTemperature.h>


#define N 10

#define ONE_WIRE_BUS 5

#define stat_led 16
```

```
WiFiClient wificlient;

HTTPClient http;

OneWire onewire(ONE_WIRE_BUS); //Creates a OneWire instance

DallasTemperature tempSensor(&onewire); //Pass OneWire reference to Dallas Temperature.


volatile int BPM;

volatile int Signal;

volatile int IBI = 600;

volatile boolean Pulse = false;

volatile boolean QS = false;


volatile int Rate[N];

volatile unsigned long CurrBeatTime = 0;

volatile unsigned long LastBeatTime = 0;

volatile int P = 500;

volatile int T = 500;

volatile int Threshold = 512;

volatile int Amplifier = 100;


int PulseSensorPin = 17;

int FadePin = 4;

int FadeRate = 0;


const char* ssid="NEXUS EXTENDER";

const char* password="nxs@1234";
```

```
unsigned long int last_post=0;
int post_interval=1000;
int count=0;

void setupTimer(int m /* msec */) {
  timer0_isr_init();
  timer0_attachInterrupt(timer0_ISR);
  timer0_write(ESP.getCycleCount() + 80000L * m); // 80MHz/1000 == 1msec
}

void setup_wifi()
{
  WiFi.begin(ssid,password);
  while(WiFi.status()!= WL_CONNECTED)
  {

    delay(1000);

    Serial.println("CONNECTING..... ");
  }
  Serial.println("CONNECTED");
}

void send_data(String A)
{
  if(WiFi.status()==WL_CONNECTED&&(millis()-last_post)>=post_interval)
  {
```

```
    http.begin(wificlient,"http://192.168.124.95:5000/test");
    http.addHeader("Content-Type","application/x-www-form-urlencoded");
    String data=A;
    int httpcode=http.POST(data);
    String payload=http.getString();
    Serial.println(httpcode);
    last_post=millis();
    if(httpcode>0)
    {
      String res=http.getString();
      Serial.println(res);
    }
    http.end();
  }
}


void timer0_ISR (void) {
  noInterrupts();
  Signal = system_adc_read();
  CurrBeatTime = getCurrentTime(); // msec
  unsigned long interval = CurrBeatTime - LastBeatTime;


  // hold bottom
  if ((Signal < Threshold) && (interval > (IBI * 3) / 5)) {
    if (Signal < T) {
      T = Signal;
      //Serial.println("T:" + String(T));
```

```
  }
}

// hold peak
if (Signal > Threshold && Signal > P) {
  P = Signal;
  //Serial.println("P:" + String(P));
}

if (interval > 250 /* ms */) {

  // check if Signal is over Threshold
  if ((Signal > Threshold) && !Pulse && (interval > (IBI * 3) / 5)) {
    Pulse = true;
    IBI = interval;

    if (Rate[0] < 0) { // first time
      Rate[0] = 0;
      LastBeatTime = getCurrentTime();
      setupTimer(10);
      noInterrupts();
      return;
    } else if (Rate[0] == 0) {  // second time
      for (int i = 0; i < N; ++i) {
        Rate[i] = IBI;
      }
    }
```

```
    word running_total = 0;
    for (int i = 0; i < N - 1; ++i) {
      Rate[i] = Rate[i + 1];
      running_total += Rate[i];
    }

    Rate[N - 1] = IBI;
    running_total += IBI;
    running_total /= N;
    BPM = 60000 / running_total;
    QS = true;
    LastBeatTime = getCurrentTime();
  }
}

// check if Signal is under Threshold if
((Signal < Threshold) && Pulse) {
Pulse = false;
  Amplifier = P - T;
  Threshold = Amplifier / 2 + T; // revise Threshold
  P = Threshold;
  T = Threshold;
}

// check if no Signal is over 2.5 sec
if (interval > 2500 /* ms */) {
```

```
    Threshold = 512;
    P = 500;
    T = 500;
    LastBeatTime = getCurrentTime();
    for (int i = 0; i < N; ++i) {
      Rate[i] = -1;
    }
  }
  setupTimer(10);
  interrupts();
}
unsigned long getCurrentTime() {
  return ESP.getCycleCount() / 80000L;
}
void setup() {
  setup_wifi();
  tempSensor.begin(); //Start temp sensor
  pinMode(FadePin, OUTPUT);
  analogWriteRange(255);
  Serial.begin(9600);
  pinMode(stat_led,OUTPUT);

  noInterrupts();
  setupTimer(10);

  interrupts();
  LastBeatTime = getCurrentTime(); // msec
```

```
}


void loop() {
  count++;
 // Serial.println(count);
  if(count==10){
  digitalWrite(stat_led,HIGH);
    }
  if(count==20){
    count=0;
    digitalWrite(stat_led,LOW);
    }
  tempSensor.requestTemperatures();
  int myTEMP = tempSensor.getTempCByIndex(0);  // Calls function on our tempSensor object that returns temp as an int.
  if (QS) {
    FadeRate = 255;
    if (BPM > 125) {
    BPM = 130;
    }
    else if (BPM < 60) {
      BPM = 60;
    }
    else {
      BPM = BPM;
    }
```

```
    Serial.print("BPM: ");
    Serial.println(BPM);


    Serial.print("BPM: ");
    Serial.println(BPM);
    Serial.print("Temp: ");
    Serial.println(myTEMP);
    String fval=String(BPM)+","+String(myTEMP)+",";
    send_data(fval);
    QS = false;
  }


  FadeRate -= 15;
  FadeRate = constrain(FadeRate, 0, 255);
  analogWrite(FadePin, FadeRate);
  delay(20);
}
```

ARDUINO PROGRAM

```
#define USE_ARDUINO_INTERRUPTS true
#include  <PulseSensorPlayground.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2
#define ledPin 10
#define data_led 9
const int PulseWire = 0;
int myBPM = 0, Celsius;
```

```
int count = 0;
String val;
int flag1 = 0,flag2 = 0,flag3 = 0,flag4 = 0,flag5 = 0;
PulseSensorPlayground pulseSensor;
OneWire  oneWire(ONE_WIRE_BUS);
DallasTemperature  sensors(&oneWire);
const int motor1F = 4;
const int motor1R = 5;
const int motor2F = 6;
const int motor2R = 7;

void setup()
{
  Serial.begin(9600);//serial communication enabling by 9600 baud rate
  pinMode(ledPin, OUTPUT);
  pinMode(data_led, OUTPUT);
  pinMode(motor1R, OUTPUT);

  pinMode(motor2R, OUTPUT);

  pinMode(motor2F, OUTPUT);

  pinMode(motor2R, OUTPUT);



  digitalWrite(motor1R, LOW);
  digitalWrite(motor1F, LOW);
```

```
  digitalWrite(motor2R, LOW);
  digitalWrite(motor2F, LOW);
  pulseSensor.analogInput(PulseWire);


  pulseSensor.begin();
  if (pulseSensor.begin()) {
    //Serial.println("We created a pulseSensor Object !");  //This prints one time at Arduino power-up,  or on Arduino reset.
  }
  // initialize timer1
  // enable all interrupts
}


void writeString(String stringdata)
{
  for (int i = 0; i < stringdata.length(); i++) {
    Serial.write(stringdata[i]);


  }
}


void loop()
{
  while (Serial.available() > 0) {
    char msg = Serial.read();
    Serial.println(msg);
    if (msg == 'A') {
```

```
    flag1 = 1;
   }
   else  if (msg == 'B') {
    flag2 = 1;
   }
   else  if (msg == 'C') {
    flag3 = 1;
   }
   else  if (msg == 'D') {
    flag4 = 1;
   }
   else  if (msg == 'E') {
    flag5 = 1;
   }
  }
  digitalWrite(ledPin,  HIGH);
 // delay(250);
  digitalWrite(ledPin, LOW);
  //delay(250);
  sensors.requestTemperatures();
  Celsius = sensors.getTempCByIndex(0);
  //Serial.println(Celsius);
  myBPM = pulseSensor.getBeatsPerMinute();  // Calls function on our pulseSensor
object that returns BPM as an "int".
  //Serial.println(myBPM);

  if (myBPM > 120) // Calculates BPM
```

```
{
  if (myBPM > 125 && myBPM < 150)
  {
    myBPM = myBPM - 45;
  }
  else
  {
    myBPM = myBPM - 150;
  }
}

//Serial.println(myBPM);

count++;



if (count == 5) {
  val = String(myBPM) + "," + String(Celsius) + "*";
  count = 0;
  writeString(val);
  digitalWrite(data_led, HIGH);
  delay(250);
  digitalWrite(data_led, LOW);
  delay(250);
}
if (flag1 == 1) {
  flag1=0;
```

```
  digitalWrite(motor1F,  HIGH);
  digitalWrite(motor1R,   LOW);
  digitalWrite(motor2F,  HIGH);
  digitalWrite(motor2R,  LOW);
 }
if (flag2 == 1) {
  flag2=0;
  digitalWrite(motor1R, HIGH);
  digitalWrite(motor1F, LOW);
  digitalWrite(motor2R, HIGH);
  digitalWrite(motor2F, LOW);
 }
if (flag3 == 1) {
  flag3=0;
  digitalWrite(motor1R,  LOW);
  digitalWrite(motor1F,   LOW);
  digitalWrite(motor2R,  LOW);
  digitalWrite(motor2F,  LOW);
 }
if (flag4 == 1) {
  flag4=0;
  digitalWrite(motor1R, LOW);
  digitalWrite(motor1F, HIGH);
  digitalWrite(motor2R, LOW);
  digitalWrite(motor2F, LOW);
  delay(500);
  digitalWrite(motor1R, LOW);
```

```
   digitalWrite(motor1F,  LOW);
   digitalWrite(motor2R,  LOW);
   digitalWrite(motor2F,  LOW);
  }
  if (flag5 == 1) {
   flag5=0;
   digitalWrite(motor1R, LOW);
   digitalWrite(motor1F, LOW);
   digitalWrite(motor2R, LOW);
   digitalWrite(motor2F, HIGH);
   delay(500);
   digitalWrite(motor1R, LOW);
   digitalWrite(motor1F, LOW);
   digitalWrite(motor2R, LOW);
   digitalWrite(motor2F,  LOW);
  }

}
```

CHAPTER 9

CONCLUSION

Technological advancement of today's era has served society to a very large extent. Implementation of smart and voice-controlled devices are also one of these advancements that helped handicapped and paralyzed patients of any kind. A lot of prototypes had been designed to help community in this regard. Some of these prototypes were developed using voice-based models and some of them were designed using android based GUI controls. These models do not incorporate all situations as deaf-mute individuals cannot operate voice based smart prototypes. Keeping in mind this deficiency, a fully automated voice and android controlled smart wheelchair is designed that is capable of working in both scenarios i.e., voice based and android based. Overhead of manual wheelchair is removed with fully automated smart wheelchair that is capable of moving in all directions. This proposed structure adds to the confidence of different inadequate people. This model can further be enhanced with the involvement of vision-based approach. Involvement of high-definition camera and smart sensors will make this prototype more realistic and automated to work in all environmental scenarios.

REFERENCES

1. Md. Syedul Amin, Jubayer Jalil, M. B. I. Reaz; “Accident Detection and Reporting System using GPS, GPRS and GSM Technology”. International Conference on Informatics, Electronics & Vision IEEE 2012
2. L. Chuan-zhi, H. Ru-fu, Y.E. Hong-wu, “Method of Freeway Incident Detection Using Wireless Positioning,” in Proceedings of the IEEE International Conference on Automation and Logistics, 2008, pp. 2801 - 2804.
3. C. Thompson, J. White, B. Dougherty, A. Albright, and D. C. Schmidt, "Using Smart phones to Detect Car Accidents and Provide Situational Awareness to Emergency Responders," in 3rd International ICST Conference on MOBILE Wireless Middle WARE, Operating Systems, and Applications (Mobil ware 2010), 2010.
4. Md. Syedul Amin, Mohammad ArifSobhan Bhuiyan, Mamun Bin IbneReaz, Salwa Sheikh Nasir; “GPS and Map Matching Based Vehicle Accident Detection System”. 2013 IEEE Student Conference on Research and Development (SCOReD).
5. Ancy John, Nishanth P.R; “Real Time Embedded System for Accident Prevention”. International Conference on Electronics, Communication and Aerospace Technology IEEE 2017.
6. Lyla B Das,"Embedded Systems” – an Integrated Approach, First Impression, Pearson, 2013.
7. Parag Parmar, Ashok M. Sapkal; “Real Time Detection and Reporting of Vehicle Collision”. International Conference on Trends in Electronics and Informatics IEEE 2017.
8. Venkata Krishna Kota, Nagendra Kumar Mangali, Thirumal Kumar Kanakurthi, A. Rakesh Kumar, T. Velayutham; “Automated Accident Detection and Rescue System”. IEEE WiSPNET 2017.