GOVERNMENT POLYTECHNIC, AURANGABAD

(An Autonomous Institute of Government of Maharashtra)

"PURSUIT FOR EXCELLENCE"

PROJECT REPORT

ON

"VOICE CONTROLLED ROBOTIC CAR"

SUBMITTED

BY

Mr. DNYANESHWAR JADHAV (Enrolment No.156008)

Mr. SHAIKH AADIL (Enrolment No.156208)

Ms. MRUNALI SHINDE (Enrolment No. 156209)

Mr. SAGAR WAGHMARE (Enrolment No. 156213)

GUIDED

BY

P.S.HIWALE

(Lecturer in Computer Engineering)

DEPARTMENT OF COMPUTER ENGINEERING

ACADEMIC YEAR 2018-19

GOVERNMENT POLYTECHNIC, AURANGABAD
(An Autonomous Institute of Government of Maharashtra)

CERTIFICATE

This is to certify that

1) Mr. DNYANESHWAR JADHAV
2) Mr. SHAIKH AADIL
3) Ms. MRUNALI SHINDE
4) Mr. SAGAR WAGHMARE

Has successfully completed project work titled **"VOICE CONTROLLED ROBOTIC CAR"** during the academic year 2018-2019, in partial fulfillment of **Diploma in Computer Engineering** of Government Polytechnic, Aurangabad. To the best of my knowledge and belief this seminar work has not been submitted elsewhere.

Date: / /2019

Mr. P.S.HIWALE
Project Guide

Mrs. D. P. SAPKAL
H.O.D CO and IT

Prof. F. A. KHAN
PRINCIPAL

ACKNOWLEDGEMENT

We express our thanks to all those who have provided us valuable guidance towards the completion of this project as part of the curriculum of the diploma course. We express our sincere gratitude towards cooperative department who has provided us with valuable assistance and requirements for the software development.

I hereby take this opportunity to record my sincere thanks and heartily gratitude to **P.S.HIWALE** for her useful guidance, making available to us her knowledge and experience in making "**Voice Controlled Robotic Car**" as a project.

The acknowledgement will be incomplete if we do not record our sense of gratitude to our Principal **Prof.F.A.Khan** Sir, who gave us necessary guidance and encouraged us by providing all the facility to work on this software application.

Mr. DNYANESHWAR JADHAV (Enrolment No.156008)
Mr. SHAIKH AADIL (Enrolment No.156208)
Ms. MRUNALI SHINDE (Enrolment No. 156209)
Mr. SAGAR WAGHMARE (Enrolment No. 156213)

INDEX

CONTENTS	PAGE NO.
List of Figures	II
Table index	III
1. INTRODUCTION	1
1.1 Need	
1.2 Project Scope	
1.3 Objective	2
2. LITERATURE SURVEY	
2.1 Limitation of Existing Systems	3
2.2 Existing System Details & Info.	4
3. SYSTEM DEVELOPMENT	
3.1 Proposed Methodology Technique	5
3.2 System Requirement H/W & S/W	5
3.2.1 PCB Designing Process	29
3.3 System Designing	35
3.3.1 Data Flow Diagram	37
3.4 Sample Code	37
3.5 Project Output	42
4. Performance Analysis	
4.1 Testing	43
4.1.1Test Cases	46
5. CONCLUSION	47
5.1 Applications	
5.2 Advantages	
5.3 Disadvantages	
5.4 Conclusion	
5.5 Future Scope	48
REFERENCES	

FIGURE INDEX

TABLE INDEX

ABSTRACT

Voice Controlled Robot (VCR) is a mobile robot whose motions can be controlled by the user by giving specific voice commands. The speech is received by a microphone and processed by the voice module. When a command for the robot is recognized, then voice module sends a command message to the robot's microcontroller.

This project Voice Controlled Robotic Vehicle helps to control robot through voice commands received via android application. The integration of control unit with Bluetooth device is done to capture and read the voice commands. The robotic vehicle then operates as per the command received via android application. For this Atmel AVR Atmega 328microcontroller is integrated in the system which makes it possible to operate the vehicle via android application.

The controlling device may be any android based Smartphone/tab etc. having an android OS. The android controlling system provides a good interactive GUI that makes it easy for the user to control the vehicle. The transmitter uses an android application required for transmitting the data. The receiver end reads these commands and interprets them into controlling the robotic vehicle.

The android device sends commands to move the vehicle in forward, backward, right and left directions. After receiving the commands, the microcontroller then operates the motors I order to move the vehicle in four directions. The communication between android device and receiver is sent as serial communication data. The microcontroller program is designed to move the motor through a motor driver IC as per the commands sent by android device.

1. INTRODUCTION

The Voice Controlled Robot is controlled through voice commands given by the user who is operating the system. These voice command needs to be given through an android app which is installed on the user's android mobile. Please note that user should have good internet connection in order to have a smooth operation of the android application. Speech recognition is done within the android app and then a respective command is sent to the voice-controlled robot vehicle. Microcontroller fitted on the Robot decodes these commands and gives an appropriate command to the motors connected to the robot.

1.1 Need

In voice operated robot the robotic vehicle movement controlled via voice command. This is Arduino Uno base project. In this project we use Bluetooth device to receive command from user. User use android application to give command to Bluetooth device. Bluetooth device receive command from application and transmit same to Atmel AVR AT mega 328 microcontroller then microcontroller controlling the robotic vehicle as per command. Microcontroller is the main unit of our project. It's a central processing unit (CPU) of the Robot. It receives various commands from BLUETOOTH decoder and gives the respective output motor driver ICs. This robotic vehicle operated on 5 commands like forward, reverse, left, right, and stop. This project we use Atmel AVR AT mega 328 microcontroller after receiving above command microcontroller to move the motors as per command. The communication between android application and Bluetooth is serial communication. The robot is controlled by Android mobile phone that connects to Bluetooth decoder attached to the robot. Thus, we need 1 mobile & 1 Bluetooth decoder. Bluetooth decoder is always connected to Robot and another mobile phone is used to control the movements of the Robot. It consists of Bluetooth decoder. It gives ASCII code output. This receiver enables wireless transmission & reception of serial data. It has 10 meters range. DC motor used for robotic vehicle movement.

The Bluetooth connected on the Arduino Uno board receives text from the Android app as characters and stored them as string to the assigned String. There are words pre-programmed (forward, reverse, right, left and stop) to the Arduino Uno, whenever the received text matches with the pre-programmed words ,the Arduino Uno executes the command that assigned to the words.

1.2PROJECT SCOPE

The main objective of developing this Voice recognition vehicle microcontroller project is to control Vehicles according to human voice command. Project Architecture follows with human input voice and amplifiers, when human sends voice then it automatically converts the voice from Analog to digital signals via converters, here band pass filters are conn ected to fingerprint templates to generate fingerprints, this module works with comparing and controlling digital signals and finally this signal goes to the vehicle.

If the vehicle received correct signal which is sent be user then it can respond as per user project development. Now a day all robotics are working with signals and voice to control their functionality. Micro controllers, Mat lab, Micro semi-conductors are used to develop this electronics system. This application mainly useful for speech enabled vehicle design and development. Advantages of this Voice recognition vehicle follows we control the any electrical or electronic device with voice signals.

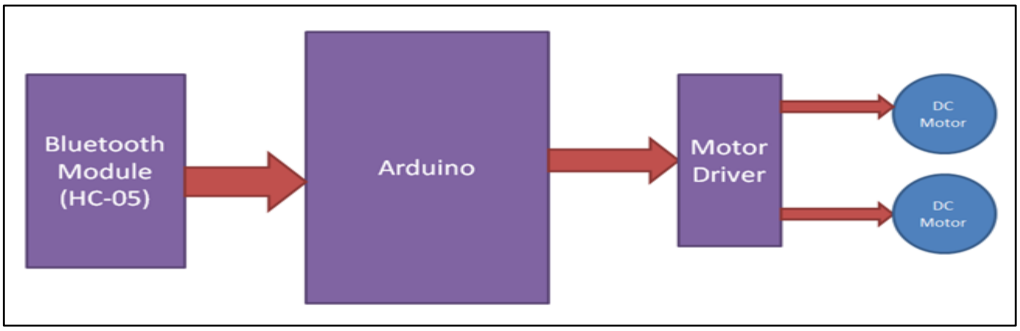

FIG 1.1: PROPOSED SYSTEM

Bluetooth controlled car moves according to button touched in the android Bluetooth mobile app. To run this project first we need to download Bluetooth app form Google play store. We can use any Bluetooth app that supporting or can send data. Here are some apps' name that might work correctly.

- ✓ Bluetooth SPP Protocol
- ✓ Bluetooth controller

1.3 OBJECTIVES

- ✓ The main objective of the project is to control the robotic car in desired position.
- ✓ The project is to control the robot by the voice or push button.
- ✓ The project is designed to control a robotic vehicle by voice and manual control or remote operation.

2. LITERATURE SURVEY

Robot Control Design Using Android Smartphone. The purpose of this project is to provide powerful computational android platforms with simpler robot's hardware architecture. This paper describes how to control a robot using mobile through Bluetooth communication, some features about Bluetooth technology, components of the mobile and robot. It presents a review of robots controlled by mobile phone via moving the robot upward, backward, left and right side by the android application such as Arduino, Bluetooth. Smart Phone Controlled Robot Using ATMEGA328 Microcontroller. In this paper have designed a robot that can be controlled using an application running on an android phone. It sends control command via Bluetooth which has certain features like controlling the speed of the motor, sensing and sharing the information with phone about the direction and distance of the robot from the nearest obstacle.

Android Mobile Phone Controlled Bluetooth Robot Using Atmel AVR AT mega 328 Microcontroller. A robot is usually an electro-mechanical machine that is guided by computer and electronic programming. Many robots have been built for manufacturing purpose and can be found in factories around the world. This paper develops the remote buttons in the android app which control the robot motion with them. And in which Bluetooth communication is use to interface controller and android. Controller is interfaced to the Bluetooth module though UART protocol.

2.1 LIMITAYIONS OF EXISTING SYSTEM

In the traditional cars so far, there are different kinds of internal combustion engines. Diesel engines are one form and gas turbine engines are another. Also HEMI engines, rotary engines and two-stroke engines. Each has its own advantages and disadvantages.

There is such a thing as an external combustion engine. A steam engine in old-fashioned trains and steam boats is the best* example of an external combustion engine. The fuel (coal, wood, oil, whatever) in a steam engine burns outside the engine to create steam, and the steam creates motion inside the engine. Internal combustion is a lot more efficient (takes less fuel per mile) than external combustion, plus an internal combustion engine is a lot smaller than an equivalent external combustion engine. This explains why we don't see any cars from Ford and GM using steam engines. There is a unit called Leaf-spring suspension the leaf spring is the oldest and simplest suspension. Several long, thin steel leaves are bound together in a pack by clamps. One end of the pack is connected to the vehicle's frame Vis a bushing. The other end uses a shackle that can move fore and aft. Combined with the flexing of the leaf pack itself that provides the suspension movement and cushions the ride. Manual transmissions provide a direct connection to the machine—one that makes the whole experience of driving a more rewarding activity. The job of the manual transmission is to transmit the engine's torque from the input shaft, through various gear sets to the output and on to the axle and driven wheels. Those gear sets in the transmission combine with the gears in the axle to multiply the torque of the engine and get the car moving. This basic animation shows how the gears are selected, and what these gears actually do when you move the shifter. A clutch let the driver smoothly couple and de-couple the engine from gearbox so that power can flow from one to the other without stalling the engine or damaging to other components. Every time you push the clutch pedal, the pressure plate releases pressure from the clutch disc, separating it from the engine's flywheel. Releasing the clutch pedal (after a new gear is selected, for example) creates friction between the disc and flywheel, which connects the transmission to the engine again, sending power to the wheels.

2.2 EXISTING SYSTEM DETAILS & INFORMATION

FEASIBILITY REPORT

Preliminary investigation examines project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- ✓ Technical Feasibility
- ✓ Operational Feasibility
- ✓ Economic Feasibility

TECHNICAL FEASIBILITY

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- ✓ Does the necessary technology exist to do what is suggested?
- ✓ Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- ✓ Can the system be upgraded if developed?
- ✓ Are there technical guarantees of accuracy, reliability, ease of access and data security?

OPERATIONAL FEASIBILITY

- ✓ Is there sufficient support for the management from the users?
- ✓ Will the system be used and work properly if it is being developed and implemented?
- ✓ Will there be any resistance from the user that will undermine the possible application benefits?

ECONOMICAL FEASIBILITY

A system can be developed technically and that will be used if used and co-ordinated by the individual user of the vehicle. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible.

3. SYSTEM DEVELOPMENT

3.1 PROPOSED METHODOLOGY TECHNIQUE

The main objective of developing this Voice recognition vehicle microcontroller project is to control Vehicles according to human voice command. Project Architecture follows with human input voice, when human sends voice then it automatically converts the voice from Analog to digital signals via converters, this module works with comparing and controlling digital signals and finally this signal goes to the vehicle. If the vehicle received correct signal which is sent be user then it can respond as per user project development. Now a day all robotics are working with signals and voice to control their functionality. Micro controllers, Mat lab, Micro semi-conductors are used to develop this electronics system. This application mainly useful for speech enabled vehicle design and development. Advantages of this Voice recognition vehicle follows we control the any electrical or electronic device with voice signals.

3.2 SYSTEM REQURIMENTS HARDWARE & SOFTWARE

SOFTWARE REQUIREMENTS

1) Arduino IDE

The Arduino IDE is a cross-platform application written in Java and is derived from the IDE for the Processing programming language and the Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit make files or run programs on a command-line interface. Although building on command-line is possible if required with some third-party tools such an Arduino IDE**.**

The Arduino IDE comes with a C/C++ library called "Wiring" (from the project of the same name), which makes many common input/output operations much easier. Arduino programs are written in C/C++.

ANDROID APPLICATIOM

1) AMR_Voice

Uses android mobiles internal voice recognition to pass voice commands to your robot Pairs with Bluetooth Serial Modules and sends in the recognized voice as a string for example if you say Hello the android phone will return a sting *Hello# to your Bluetooth module *and # indicate the start and stop bits, Can Be used with any micro controller which can handle strings.
Examples Platforms: Arduino, ARM, PICAXE, MSP430, 8051 based and many other processors and controllers.

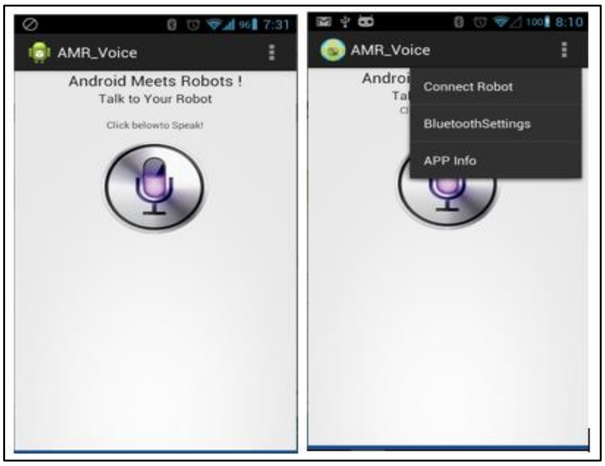

FIG.3.1: AMR VOICEAPPLICATION

2) Bluetooth Terminal HC-05 for Android

One-of-a-kind App that gives you compatibility with all controllers such as Arduino, Raspberry Pi, AVR, PIC, ARM as well as TI controllers. All you need is a HC-05 serial adapter connection with serial ports of the controllers. Control any Micro-controller that uses a Bluetooth Module HC 05 or HC 06 through your smart phone. This app can send and receive commands via Bluetooth so you can debug your hardware problems easily. FEATURES: Separate panels for sending and receiving data. Custom your own buttons for frequent sending of same data. Selection for \r \n at the end of sending data. Monitoring receiving data as HEX or ASCII. Simple copy option in sent data just long press on data. Remove Ads and get uninterrupted access with an Ad - free version of Bluetooth Terminal.

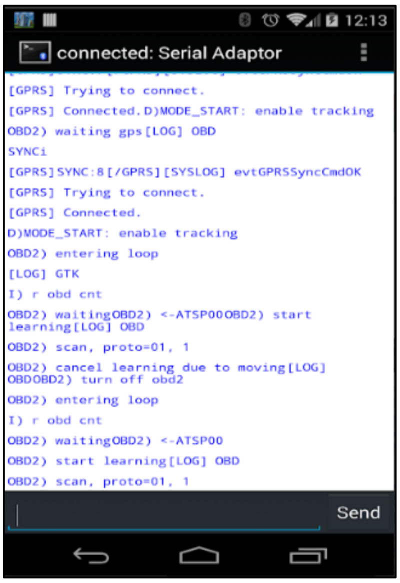

FIG.3.2: BLUETOOTH TERMINAL HC-05 FOR ANDROID

3) Arduino Bluetooth RC Car

The application allows you to control an Arduino based RC car over Bluetooth. This is done using a Bluetooth enabled Android phone. The app lets you control the car with either buttons or the phone's accelerometer. A slider bar allows you to control your car's velocity if the car's control circuit has this feature.

There are also two buttons for front and back lights. A flashing light lets you know when the phone is connected to the car, and arrows light up letting you know the car's driving direction.

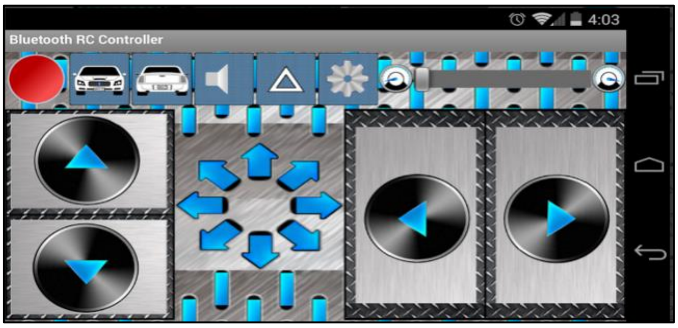

FIG.3.3: ARDUINO BLUETOOTH RC CAR

4) KI CAD

KiCad (pronounced "Key-CAD") is a free software suite for electronic design automation (EDA). It facilitates the design of schematics for electronic circuits and their conversion to PCB designs. KiCad was originally developed by Jean-Pierre Charras. It features an integrated environment for schematic capture and PCB layout design. Tools exist within the package to create a bill of materials, artwork, Gerber files, and 3D views of the PCB and its components.

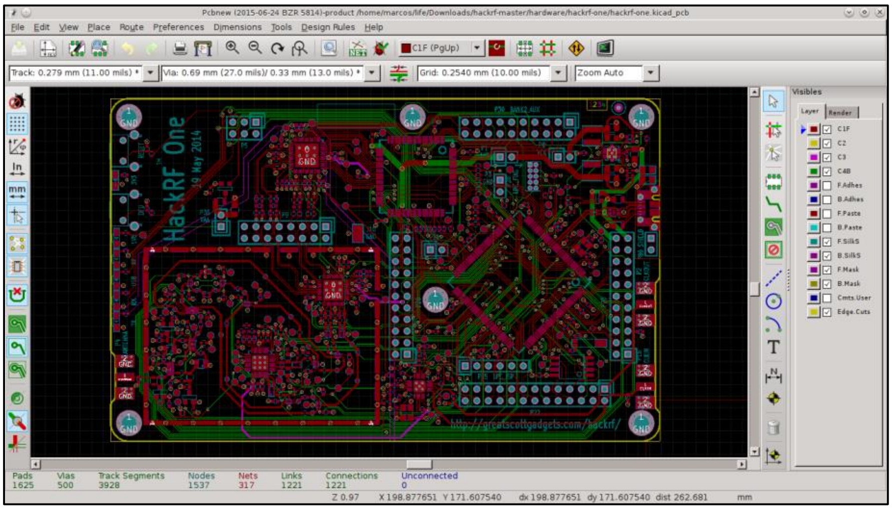

FIG.3.4: KICAD ENVIRONMENT

FEATURES:

KiCad uses an integrated environment for all of the stages of the design process: Schematic capture, PCB layout, Gerber file generation/visualization, and library editing. KiCad is a cross-platform program, written in C++ with widgets to run on FreeBSD, Linux, Microsoft Windows and Mac OS X. Many component libraries are available, and users can add custom components. The custom components can be available on a per-project basis or installed for use in any project. There are also tools to help with importing components from other EDA applications, for instance EAGLE. Configuration files are in well documented plain text, which helps with interfacing version control systems, as well as with automated component generation scripts.

Multiple languages are supported, such as Bulgarian, Catalan, Chinese, Czech, Dutch, English, Finnish, French, German, Greek, Hungarian, Italian, Japanese, Korean, Lithuanian, Polish, Portuguese, Russian, Slovak, Slovene, Spanish, and Swedish.

5) Embedded C

Embedded C is a set of language extensions for the C programming language by the C Standards Committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations. [Citation needed]

In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing. [Citation needed] Embedded C uses most of the syntax and semantics of standard C, e.g., main () function, variable definition, data type declaration, conditional statements (if, switch case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc. [citation needed]

HARDWARE REQUIREMENTS

For designing this hardware many types of devices are used to make it perfectly working. All the devices are purchased from different manufacturers. These components are soldered on a soldering board. The following lists of hardware are required for this system.

- ✓ Arduino Uno
- ✓ Microcontroller ATmega328
- ✓ L293D Motor Driver IC
- ✓ L293D Motor Driver circuit
- ✓ SR-04 Module
- ✓ HC 05 Module
- ✓ 100 Rpm Motors * 2
- ✓ Light Dependent Register
- ✓ LED’s
- ✓ Piezoelectric Buzzer
- ✓ 12 v Power Supply

1) Arduino Uno

Arduino is embedded system. Embedded system is the core of every intelligent device. Everything around you from a simple watch to the space ship is using embedded systems. Your mobile phones, PDAs, Washing Machines, Microwave Ovens, Automobile and all the automated electronics gadgets are having one or another form of embedded systems responsible for their intelligent functioning. Electronic devices are rapidly finding applications in many of the areas, from simple coffee vending machines to space science to environmental friendly products such as big belly etc. Indian embedded market worth $5.9 billion dollars and expects 30 % growth by 202o. This will certainly increase the demand of professionals in the field of embedded systems. Main aim of this Training module is to make the students skill full and efficient for all the hardware and software related concepts for AVR Microcontrollers (At mega 328P).

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

Arduino is some open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments it consists of step down transformer, bridge rectifier, capacitors and voltage regulator ICs. 230V AC is converted to 12V DC using transformer and bridge rectifier. This 12VDC is further reduced to 5V DC using voltage regulator IC.

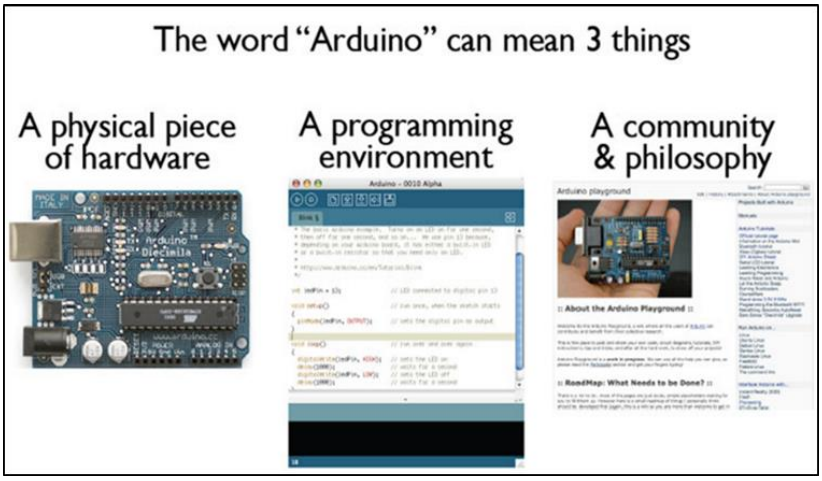

FIG.3.5: ARDUINO UNO MEANS

Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available.

Although the hardware and software designs are freely available under copy left licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in - Arduino.

Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012.The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed *shields*. Multiple and possibly stacked shields may be individually addressable via an I^2Cserial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the Lily Pad, run at 8 MHz and dispense with the on-board voltage regulator due to specific form-factor restrictions.

Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default boot loader of the Arduino UNO is the opt boot loader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Bo Arduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used.

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six Analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board band Board Uno boards may provide male header pins on the underside of the board that can plug into solder less breadboards.

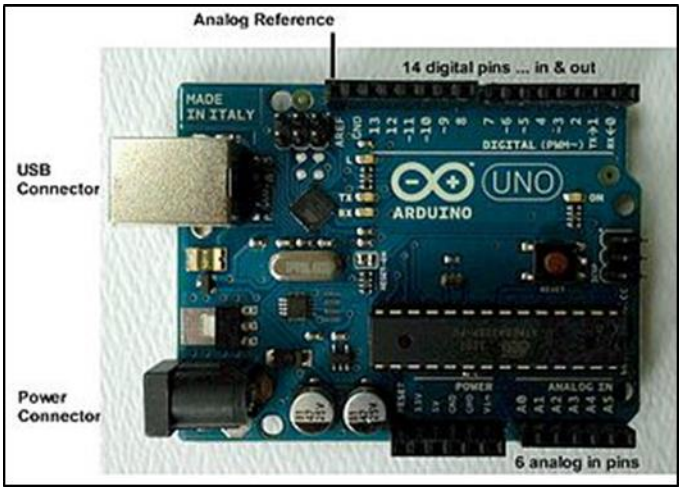

FIG.3.6: ARDUINO UNO

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

A pre-assembled Arduino board includes a microcontroller, which is programmed using Arduino programming language and the Arduino development environment. In essence, this platform provides a way to build and program electronic components. Arduino programming language is a simplified from of C/C++ programming language based on what Arduino calls "sketches," which use basic programming structures, variables and functions. These are then converted into a C++ program.

2) Microcontroller Atmega328

The Atmel AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

As stated before, 20 of the pins function as I/O ports. This means they can function as an input to the circuit or as output. Whether they are input or output is set in the software. 14 of the pins are digital pins, of which 6 can function to give PWM output. 6 of the pins are for analog input/output.

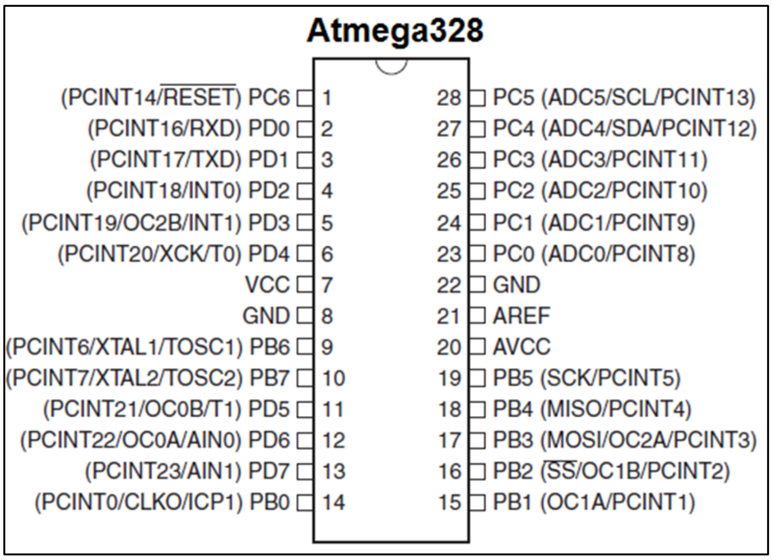

FIG 3.7: PIN DIAGRAM OF ATMEGA328

2 of the pins are for the crystal oscillator. This is to provide a clock pulse for the AT mega chip. A clock pulse is needed for synchronization so that communication can occur in synchrony between the AT mega chip and a device that it is connected to.

The chip needs power so 2 of the pins, Vcc and GND, provide it power so that it can operate. The Atmega328 is a low-power chip, so it only needs between 1.8-5.5V of power to operate.

The Atmega328 chip has an analog-to-digital converter (ADC) inside of it. This must be or else the Atmega328 wouldn't be capable of interpreting analog signals. Because there is an ADC, the chip can interpret analog input, which is why the chip has 6 pins for analog input. The ADC has 3 pins set aside for it to function- AVCC, AREF, and GND. AVCC is the power supply, positive voltage, that for the ADC. The ADC needs its own power supply in order to work. GND is the power supply ground. AREF is the reference voltage that the ADC uses to convert an analog signal to its corresponding digital value. Analog voltages higher than the reference voltage will be assigned to a digital value of 1, while analog voltages below the reference voltage will be assigned the digital value of 0. Since the ADC for the Atmega328 is a 10-bit ADC, meaning it produces a 10-bit digital value, it converts an analog signal to its digital value, with the AREF value being a reference for which digital values are high or low. Thus, a portrait of an analog signal is shown by this digital value; thus, it is its digital correspondent value. The last pin is the RESET pin. This allows a program to be rerun and start over.

Pin Number	Description	Function
1	PC6	Reset
2	PD0	Digital Pin (RX)
3	PD1	Digital Pin (TX)
4	PD2	Digital Pin
5	PD3	Digital Pin (PWM)
6	PD4	Digital Pin
7	Vcc	Positive Voltage (Power)
8	GND	Ground
9	XTAL 1	Crystal Oscillator
10	XTAL 2	Crystal Oscillator
11	PD5	Digital Pin (PWM)
12	PD6	Digital Pin (PWM)
13	PD7	Digital Pin
14	PB0	Digital Pin
15	PB1	Digital Pin (PWM)
16	PB2	Digital Pin (PWM)
17	PB3	Digital Pin (PWM)
18	PB4	Digital Pin
19	PB5	Digital Pin
20	AVCC	Positive voltage for ADC (power)
21	AREF	Reference Voltage
22	GND	Ground
23	PC0	Analog Input
24	PC1	Analog Input
25	PC2	Analog Input
26	PC3	Analog Input
27	PC4	Analog Input
28	PC5	Analog Input

TABLE 3.1.PIN FUNCTION OF AT Mega 328

1. VCC Digital supply voltage.

2. GND Ground.

3. Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit. Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier. If the Internal Calibrated RC Oscillator is used as chip clock source, PB7...6 is used as TOSC2...1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set. The various special features of Port B are elaborated in and “System Clock and Clock Options”.

4) Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5...0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up Resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

5) PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C. If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a Reset.

6) Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

7) AVCC

AVCC is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter. Note that PC6...4 use digital supply voltage, VCC.

8) AREF

AREF is the analog reference pin for the A/D Converter.

9) ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

The Atmega328 provides the following features: 16 Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes of EEPROM, 1 Kbyte of SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two wire Serial Interface, a 6-channel ADC (eight channels in TQFP and QFN/MLF packages) with 10-bit accuracy, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption.

4) L293D Module

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.

FIG.3.8: L293D IC

L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively.

Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.

PIN DIAGRAM:

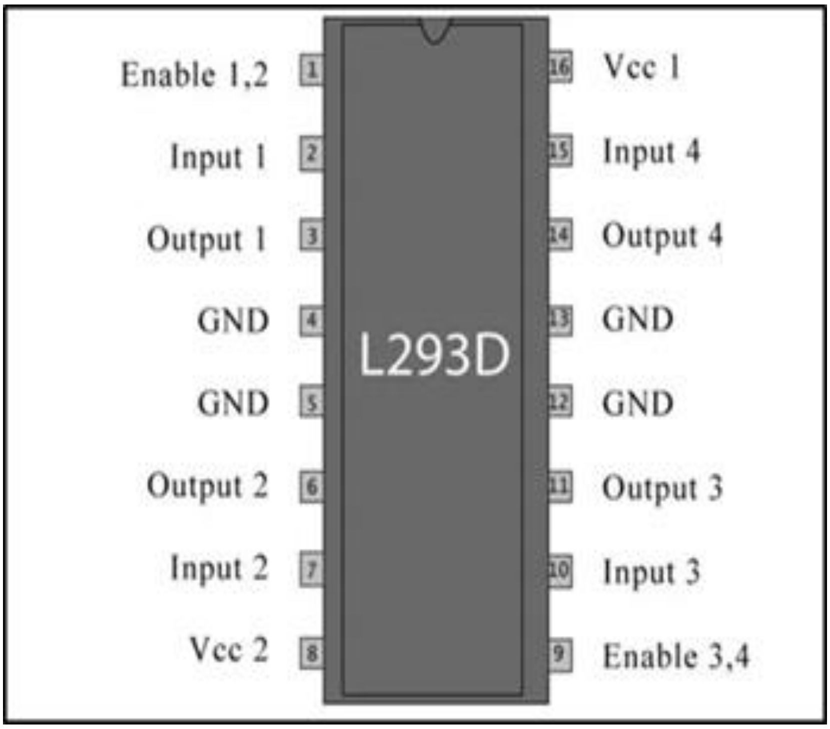

FIG.3.9: PIN DIAGRAM OF L293D

L293D Inputs				Motors	
Input 1	**Input 2**	**Input 3**	**Input 4**	**Motor 1**	**Motor 2**
1	0	1	0	Fwd	Fwd
1	0	1	0	Fwd	Fwd
1	0	0	1	Fwd	Back
1	0	0	0	Fwd	Stop
0	1	1	0	Back	Fwd
0	0	1	0	Stop	Fwd
0	1	0	1	Back	Back
0	0	0	0	Stop	Stop

TABLE.3.2: MOTOR CONTROL TABLE

PIN DESCRIPTION:

Pin No	Function	Name
1	Enable pin for Motor 1; active high	Enable 1,2
2	Input 1 for Motor 1	Input 1
3	Output 1 for Motor 1	Output 1
4	Ground (0V)	Ground
5	Ground (0V)	Ground
6	Output 2 for Motor 1	Output 2
7	Input 2 for Motor 1	Input 2
8	Supply voltage for Motors; 9-12V (up to 36V)	$V_{cc\ 2}$
9	Enable pin for Motor 2; active high	Enable 3,4
10	Input 1 for Motor 1	Input 3
11	Output 1 for Motor 1	Output 3
12	Ground (0V)	Ground
13	Ground (0V)	Ground
14	Output 2 for Motor 1	Output 4
15	Input2 for Motor 1	Input 4
16	Supply voltage; 5V (up to 36V)	$V_{cc\ 1}$

TABLE.3.3: PIN DESCRIPTION OF L293D

4) HC - SR04

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm.

The modules include ultrasonic transmitters, receiver and control circuit. The basic principle of work:

1. Using IO trigger for at least 10us high level signal.
2. The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
3. IF the signal back, through high level, time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time × velocity of sound (340M/S) / 2

Wire connecting direct as following:

1. 5V Supply
2. Trigger Pulse Input
3. Echo Pulse Output
4. 0V Ground

ELECTRIC PARAMETER

Working Voltage	DC 5V
Working Current	15 mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
Measuring Angle	15 degrees
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL Layer signal & the range in propagation
Dimension	45*20*15mm

TABLE.3.4: ELECTRIC PARAMETER OF HC SR-04

ULTRASONIC SENSOR PIN CONFIGURATION

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

TABLE.3.5: PIN CONFIGURATION OF HC SR-04

HC-SR04 Ultrasonic Sensor - Working

As shown above the HC-SR04 Ultrasonic (US) sensor is a 4-pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

$$DISTANCE = SPEED \times TIME$$

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

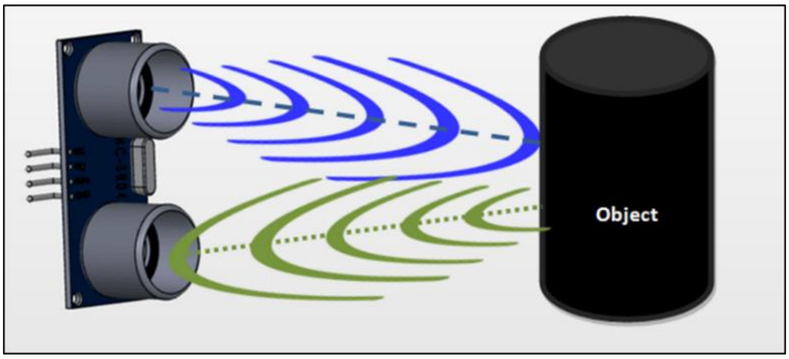

FIG.3.10: HC SR-04 Working Example

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

5) HC-05 MODULE:

HC-05 SensorHC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Blue core 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Bluetooth serial module is used for converting serial port to Bluetooth. These modules have two modes: master and slaver device. The device named after even number is defined to be master or slaver when out of factory and can't be changed to the other mode. The default situation of HC-04 is slave mode. If you need master mode, please state it clearly or Place an order for HC-O4-M directly. The naming rule of HC-06 is same.

SPECIFICATIONS:

Hardware Features

- ✓ Typical -80dBm sensitivity
- ✓ Up to +4dBm RF transmit power
- ✓ Low Power 1.8V Operation, 1.8 to 3.6V I/O
- ✓ PIO control
- ✓ UART interface with programmable baud rate
- ✓ With integrated antenna
- ✓ With edge connector

Software Features

- ✓ Slave default Baud rate: 9600, Data bits: 8, Stop bit: 1, Parity: No parity.
- ✓ Auto-connect to the last device on power as default.
- ✓ Permit pairing device to connect as default.
- ✓ Auto-pairing PINCODE:"1234" as default.

Hardware of HC 05

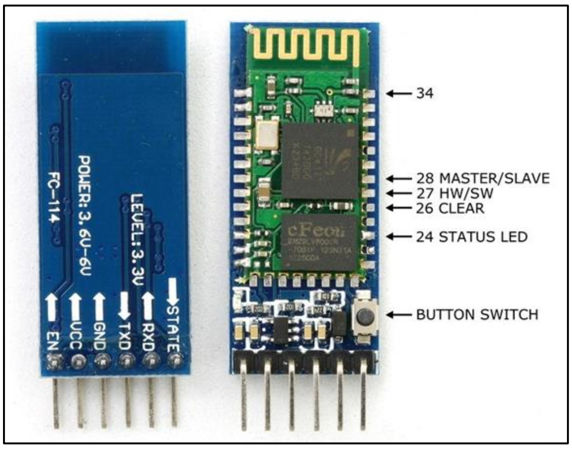

FIG.3.11: HC-05 MODULE

CONFIGURATION OF HC-05

PIN DESCRIPTION

The HC-05 Bluetooth Module has 6pins. They are as follows:

ENABLE: When enable is pulled LOW, the module is disabled which means the module will not turn on and it fails to communicate. When enable is left open or connected to 3.3V, the module is enabled i.e. the module remains on and communication also takes place.

Vcc: Supply Voltage 3.3V to 5V

GND: Ground pin

TXD & RXD: These two pins acts as an UART interface for communication

Pin Name	Pin	Pad type	Description
GND	13 21 22	VCC	Ground pot
3.3 VCC	12	3.3v	Integrated 3.3v(+) supply with on chip regulator output within 3.15-3.3v
A1o0	9	Bi-Directional	Programmable input/output line
P101	10	Bi-Directional	Programmable input/output line
P1o0	23	Bi-Directional RX EN	Programmable input/output line
P101	24	Bi-Directional TX EN	Programmable input/output line

TABLE.3.6: HC-05 PIN CONFIGURATION

STATE: It acts as a status indicator. When the module is not connected to / paired with any other Bluetooth device, signal goes Low. At this low state, the led flashes continuously which denotes that the module is not paired with another device. When this module is connected to/paired with any other Bluetooth device, the signal goes high. At this high state, the led blinks with a constant delay say for example 2s delay which indicates that the module is paired.

BUTTON SWITCH: This is used to switch the module into AT command mode. To enable AT command mode, press the button switch for a second. With the help of AT commands, the user can change the parameters of this module but only when the module is not paired with any other BT device. If the module is connected to any other Bluetooth device, it starts to communicate with that device and fails to work in AT command mode.

6) 100 RPM DC MOTOR

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor.

DC motors were the first form of motor widely used, as they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. The universal motor can operate on direct current but is a lightweight brushed motor used for portable power tools and appliances. Larger DC motors are currently used in propulsion of electric vehicles, elevator and hoists, and in drives for steel rolling mills. The advent of power electronics has made replacement of DC motors with AC motors possible in many applications.

100RPM Centre Shaft Economy Series DC Motor is high quality low cost DC geared motor. It has steel gears and pinions to ensure longer life and better wear and tear properties. The gears are fixed on hardened steel spindles polished to a mirror finish. The output shaft rotates in a plastic bushing. The whole assembly is covered with a plastic ring. Gearbox is sealed and lubricated with lithium grease and require no maintenance. The motor is screwed to the gear box from inside. Although motor gives 100 RPM at 12V but motor runs smoothly from 4V to 12V and gives wide range of RPM, and torque. Tables below gives fairly good idea of the motor’s performance in terms of RPM and no-load current as a function of voltage and stall torque, stall current as a function of voltage.

FIG.3.12: 100 RPM DC MOTOR

ELECTROMAGNETIC MOTORS

A coil of wire with a current running through it generates an electromagnetic field aligned with the centre of the coil. The direction and magnitude of the magnetic field produced by the coil can be changed with the direction and magnitude of the current flowing through it. A simple DC motor has a stationary set of magnets in the stator and an armature with one or more windings of insulated wire wrapped around a soft iron core that concentrates the magnetic field. The windings usually have multiple turns around the core, and in large motors there can be several parallel current paths. The ends of the wire winding are connected to a commentator. The commentator allows each armature coil to be energized in turn and connects the rotating coils with the external power supply through brushes. (Brushless DC motors have electronics that switch the DC current to each coil on and off and have no brushes.)

The total amount of current sent to the coil, the coil's size and what it's wrapped around dictate the strength of the electromagnetic field created. The sequence of turning a particular coil on or off dictates what direction the effective electromagnetic fields are pointed. By turning on and off coils in sequence a rotating magnetic field can be created. These rotating magnetic fields interact with the magnetic fields of the magnets (permanent or electromagnets) in the stationary part of the motor (stator) to create a torque on the armature which causes it to rotate. In some DC motor designs the stator fields use electromagnets to create their magnetic fields which allow greater control over the motor. At high power levels, DC motors are almost always cooled using forced air.

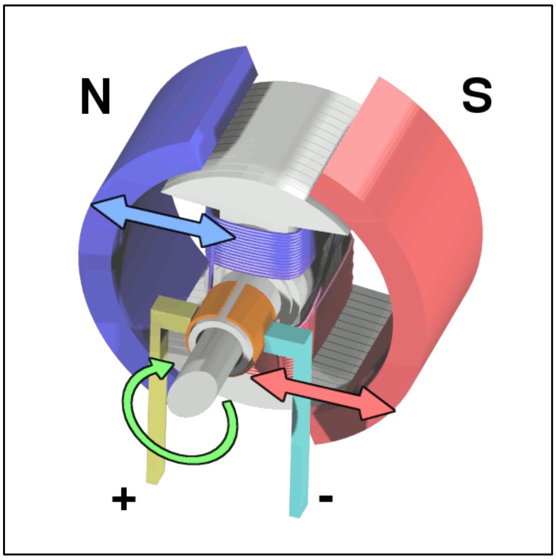

FIG.3.13: DC Motor Working

Different number of stator and armature fields as well as how they are connected provide different inherent speed/torque regulation characteristics. The speed of a DC motor can be controlled by changing the voltage applied to the armature. The introduction of variable resistance in the armature circuit or field circuit allowed speed control. Modern DC motors are often controlled by power electronics systems which adjust the voltage by "chopping" the DC current into on and off cycles which have an effective lower voltage.

Since the series-wound DC motor develops its highest torque at low speed, it is often used in traction applications such as electric locomotives, and trams. The DC motor was the mainstay of electric traction drives on both electric and diesel-electric locomotives, street-cars/trams and diesel electric drilling rigs for many years. The introduction of DC motors and an electrical grid system to run machinery starting in the 1870s started a new second Industrial Revolution. DC motors can operate directly from rechargeable batteries, providing the motive power for the first electric vehicles and today's hybrid cars and electric cars as well as driving a host of cordless tools. Today DC motors are still found in applications as small as toys and disk drives, or in large sizes to operate steel rolling mills and paper machines. Large DC motors with separately excited fields were generally used with winder drives for mine hoists, for high torque as well as smooth speed control using thermistor drives. These are now replaced with large AC motors with variable frequency drives.

If external mechanical power is applied to a DC motor it acts as a DC generator, a dynamo. This feature is used to slow down and recharge batteries on hybrid car and electric cars or to return electricity back to the electric grid used on a street car or electric powered train line when they slow down. This process is called regenerative braking on hybrid and electric cars. In diesel electric locomotives they also use their DC motors as generators to slow down but dissipate the energy in resistor stacks. Newer designs are adding large battery packs to recapture some of this energy.

Brushed:

The brushed DC electric motor generates torque directly from DC power supplied to the motor by using internal commutation, stationary magnets (permanent or electromagnets), and rotating electromagnets.

Advantages of a brushed DC motor include low initial cost, high reliability, and simple control of motor speed. Disadvantages are high maintenance and low life-span for high intensity uses. Maintenance involves regularly replacing the carbon brushes and springs which carry the electric current, as well as cleaning or replacing the commentator. These components are necessary for transferring electrical power from outside the motor to the spinning wire windings of the rotor inside the motor.

Brushes are usually made of graphite or carbon, sometimes with added dispersed copper to improve conductivity. In use, the soft brush material wears to fit the diameter of the commutator and continues to wear. A brush holder has a spring to maintain pressure on the brush as it shortens. For brushes intended to carry more than an ampere or two, a flying lead will be moulded into the brush and connected to the motor terminals. Very small brushes may rely on sliding contact with a metal brush holder to carry current into the brush or may rely on a contact spring pressing on the end of the brush. The brushes in very small, short-lived motors, such as are used in toys, may be made of a folded strip of metal that contacts the commutator.

Permanent magnet stators:

A PM motor does not have a field winding on the stator frame, instead relying on PMs to provide the magnetic field against which the rotor field interacts to produce torque. Compensating windings in series with the armature may be used on large motors to improve commutation under load. Because this field is fixed, it cannot be adjusted for speed control. PM Fields (stators) are convenient in miniature motors to eliminate the power consumption of the field winding. Larger DC motors are of the “dynamo” type, which have stator windings. Historically, PMs could not be made to retain high flux if they were disassembled; field windings were more practical to obtain the needed amount of flux. However, large PMs are costly, as well as dangerous and difficult to assemble; this favour wound fields for large machines.

To minimize overall weight and size, miniature PM motors may use high energy magnets made with neodymium or other strategic elements; most such are neodymium-iron-boron alloy. With their higher flux density, electric machines with high-energy PMs are at least competitive with all optimally designed singly fed synchronous and induction electric machines. Miniature motors resemble the structure in the illustration, except that they have at least three rotor poles (to ensure starting, regardless of rotor position) and their outer housing is a steel tube that magnetically links the exteriors of the curved field magnets.

Wound stators:

There are three types of electrical connections between the stator and rotor possible for DC electric motors: series, shunt/parallel and compound (various blends of series and shunt/parallel) and each has unique speed/torque characteristics appropriate for different loading torque profiles/signatures.

Series connection:

A series DC motor connects the armature and field windings in series with a common D.C. power source. The motor speed varies as a non-linear function of load torque and armature current; current is common to both the stator and rotor yielding current squared (I^2) behaviour. A series motor has very high starting torque and is commonly used for starting high inertia loads, such as trains, elevators or hoists.[2] This speed/torque characteristic is useful in applications such as dragline excavators, where the digging tool moves rapidly when unloaded but slowly when carrying a heavy load.

A series motor should never be started at no load. With no mechanical load on the series motor, the current is low, the counter-Electro motive force produced by the field winding is weak, and so the armature must turn faster to produce sufficient counter-EMF to balance the supply voltage. The motor can be damaged by over speed. This is called a runaway condition.

Series motors called universal motors can be used on alternating current. Since the armature voltage and the field direction reverse at the same time, torque continues to be produced in the same direction. However, they run at a lower speed with lower torque on AC supply when compared to DC due to reactance voltage drop in AC which is not present in DC. Since the speed is not related to the line frequency, universal motors can develop higher-than-synchronous speeds, making them lighter than induction motors of the same rated mechanical output. This is a valuable characteristic for hand-held power tools. Universal motors for commercial utility are usually of small capacity, not more than about 1 kW output. However, much larger universal motors were used for electric locomotives, fed by special low-frequency traction power networks to avoid problems with commutation under heavy and varying loads.

Shunt connection:

A shunt DC motor connects the armature and field windings in parallel or shunt with a common D.C. power source. This type of motor has good speed regulation even as the load varies but does not have the starting torque of a series DC motor. It is typically used for industrial, adjustable speed applications, such as machine tools, winding/unwinding machines and tensioners.

Compound connection

A compound DC motor connects the armature and fields windings in a shunt and a series combination to give it characteristics of both a shunt and a series DC motor. This motor is used when both a high starting torque and good speed regulation is needed. The motor can be connected in two arrangements: cumulatively or differentially. Cumulative compound motors connect the series field to aid the shunt field, which provides higher starting torque but less speed regulation. Differential compound DC motors have good speed regulation and are typically operated at constant speed.

7) PIZOELECTRIC BUZZER

A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke.

Buzzer is an integrated structure of electronic transducers, DC power supply, widely used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers and other electronic products for sound devices. Active buzzer 5V Rated power can be directly connected to a continuous sound, this section dedicated sensor expansion module and the board in combination, can complete a simple circuit design, to "plug and play."

FIG.3.14: PIEZOELECTRIC BUZZER

8) LDR

LDR sensor module is used to detect the intensity of light. It is associated with both Analog output pin and digital output pin labelled as AO and DO respectively on the board. When there is light, the resistance of LDR will become low according to the intensity of light.

Specifications:

- ✓ Operating voltage 3.3V-5V
- ✓ Output Type Digital – Logic High or Low.
- ✓ Power indicator (red) and Output Indicator LEDs.
- ✓ Sensitivity can be adjusted using the on-board potentiometer.

Hardware connections:

The LDR module should be connected to Uno as follows:

- Vcc to 5V
- Gnd to Gnd

D0 to Any Analog pin of Uno

The most common type of LDR has a resistance that falls with an increase in the light intensity falling upon the device (as shown in the image above). The resistance of an LDR may typically have the following resistances:

Daylight = 5000Ω

Dark = 20000000Ω

FIG.3.15: LIGHT DEPENDENT REGISTER

Variation in resistance with changing light intensity

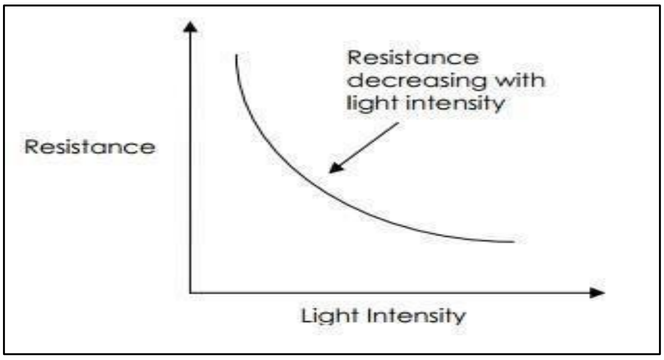

FIG.3.16: TYPICAL LDR RESISTANCE VS. LIGHT INTENSITY GRAPH

You can therefore see that there is a large variation between these figures. If you plotted this variation on a graph you would get something similar to that shown by the graph shown above.

Applications of LDRs:

There are many applications for Light Dependent Resistors. These include:

1) Lighting switch

The most obvious application for an LDR is to automatically turn on a light at a certain light level. An example of this could be a street light or a garden light.

2) Camera shutter control

LDRs can be used to control the shutter speed on a camera. The LDR would be used to measure the light intensity which then adjusts the camera shutter speed to the appropriate level.

Example - LDR controlled Transistor circuit

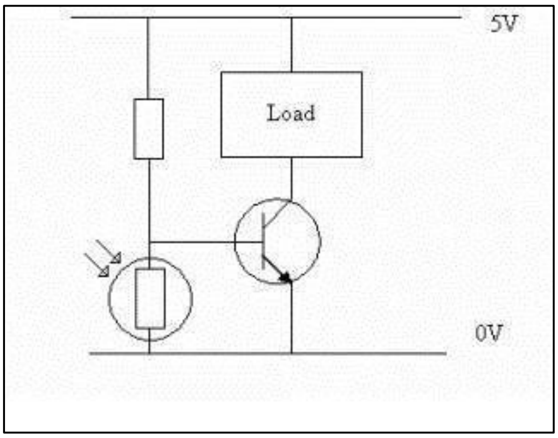

FIG.3.17: LDR CONTROLLED TRANSISTOR CIRCUIT

The circuit shown above shows a simple way of constructing a circuit that turns on when it goes dark. In this circuit the LDR and the other Resistor form a simple 'Potential Divider' circuit, where the centre point of the Potential Divider is fed to the Base of the NPN Transistor.

When the light level decreases, the resistance of the LDR increases. As this resistance increases in relation to the other Resistor, which has a fixed resistance, it causes the voltage dropped across the LDR to also increase. When this voltage is large enough (0.7V for a typical NPN Transistor), it will cause the Transistor to turn on.

The value of the fixed resistor will depend on the LDR used, the transistor used and the supply voltage.

An LED light module, or light emitting diode module, is an LED device that is made to either function alone or that is pluggable into a compatible unit for power. LED modules often have one or more LED bulbs contained in a fixture that powers the LED lights or plugs into a unit that powers the LED module.

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electronic holes, releasing energy in the form of photons. This effect is called electroluminescence. The colour of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.

3.2.1 PRINTED CIRCUIT BOARDS DESIGNING PROCESS

The most vital element in electronic circuits and equipment's is the Printed Circuit Board (PCB). It is also possible to build an electronic circuit with bread boards and zero boards, but the method is a low level and less efficient one wherein the designing circuit is prone to damage and the designing involves a complex process of placing the components of the circuit.

However, the invention of the PCB that supports physical electronic components and their wiring through the surface-mounted copper tracks is really remarkable. We can observe at least one PCB in any electronic gadgets ranging from cell phones to computers.

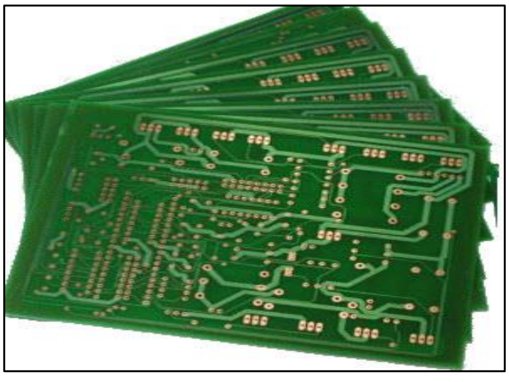

FIG.3.18: PRINTED CIRCUIT BOARD

What is a Printed Circuit Board?

Electronic circuits in engineering and industry are normally manufactured by using printed circuit boards (PCBs). These boards are made up of special materials that do not conduct electricity such as fiber and glass. The circuits are designed on the boards with copper tracks instead of wires for the conduction of electricity between the electronic components.
The electronic components are fixed in their respective positions by drilling holes on the board, placing the components and then soldering them in appropriate positions so that the copper tracks and components together form a circuit. The printed circuit boards used in all electronic products such as automotive, wireless devices, Robotic applications, etc., offer quick functioning, access, control, monitoring and precise and exact results when compared to other wiring methods based devices The below figure shows how the circuit is arranged on a PCB with copper layer.

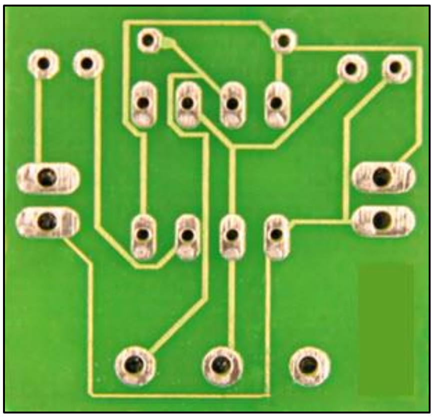

FIG.3.19: PRINTED CIRCUIT BOARD MANUFACTURING

DESIGNING PROCESS OF A PCB

Depend on printed circuit board manufacturer, there are numerous ways available for designing PCBs. This circuit board design can be manufactured as bulk using several machines in PCB fabrication industries including drilling, punching, plating and final fabrication processes that are performed through highly automated machines. Laser drilling with CNC machines, automatic plating machines, strip etching machines, and use of optical inspection equipment’s, flying probe testers for electrical testing of printed circuit board process result in high-quality PCBs (with a greater production yield).

STEP.1: DESIGN THE PCB CIRCUIT WITH A SOFTWARE

Draw the schematic circuit diagram with the PCB layout software such as CAD software, Eagle and Multisim software. This type of PCB design software contains a library of components that can be used to build the circuit. It is also possible to change the circuit design’s position and then to modify according to your convenience and requirement. Here we have selected Eagle software to design the circuit and its procedure is as follows:

- ✓ Open the Eagle circuit board design software.
- ✓ A window with a menu bar appears.
- ✓ Click on the file menu.
- ✓ Select ‘new design’ from the drop-down menu.
- ✓ Click on the library menu.
- ✓ Select ‘pick devices/symbol’ from the drop-down menu.
- ✓ Select a relevant comment by double clicking on it, so that the component appears on the window.
- ✓ Add all the components and draw the circuit with proper connections as shown in the figure.

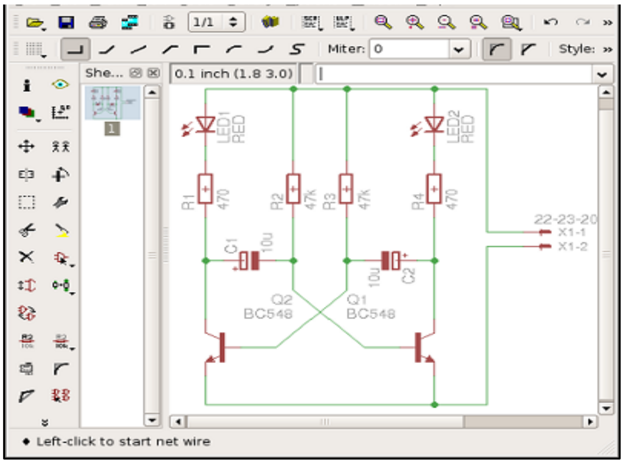

FIG.3.20: PCB DESIGNING STEP 1

- ✓ Enter the rating of each component according to the requirement.
- ✓ Go to Command Toolbar and click Text editor varriages, click on the Varriages, and then close the window.
- ✓ Next, a black screen appears which is of the layout or the film diagram of the circuit as shown in the below figure and save this as an image format.

STEP 2: FILM GENERATION

The film is generated from the finalized circuit board diagram of the PCB layout software which is send to the manufacturing unit where the negative image or mask is printed out on a plastic sheet.

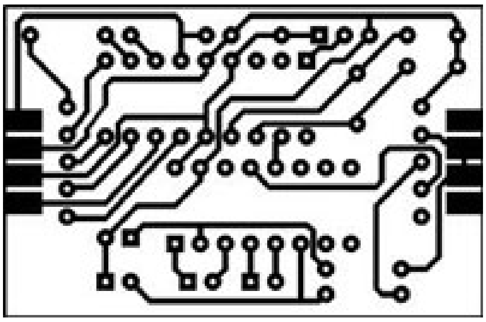

FIG.3.21: PCB FILM GENERATION

STEP 3: SELECT RAW MATERIAL

The bulk of the printed circuit board is made with an unbreakable glass or fiberglass having copper foil bonded unto one or both the sides of the board. Thus, the PCBs made from unbreakable paper phenolic with a bonded copper foil are less expensive and are often used in household electrical devices. Mostly 0.059 industry standard thick, copper clad laminate, either single or double-sided board is required. Panels may be sheared to contain May boards of different sizes.

STEP 4: PREPARING DRILL HOLES

Machines and carbide drills are used to put holes on the printed circuit board. There are two types of machines available to drill the PCBs; they include hand machines and CNC machines. The hand machines require human intervention or effort to drill the holes, whereas CNC machines are computer-based machines that work-based on the machine timetables or programs that run both automatic as well as manually. The drilled pattern is stored in the computer like drill bit sizes, number of holes per panel, drilled stack, drilled time per load, etc. The PCB boards are placed into the CNC machine and the holes are drilled according to the determined pattern to place printed circuit board components.

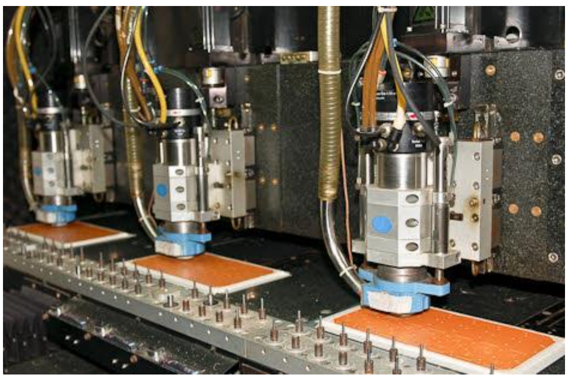

FIG.3.22: PCB PREPARING DRILL HOLES

STEP 5: APPLY IMAGE

The printed circuit layout can be printed in different ways on PCBs like manual pen, dry transfers, pen plotters and printers. The laser printers are a better way to print the layouts on printed circuit boards. The following steps are used to print the PCB layout through a laser printer:

1. Take a clean and neat copper paper and place it on the laser printer.
2. Next, store the designed layout film in the computer.

FIG.3.23: APPLYING IMAGE ON BOARD

3. A laser printer prints the designed circuit layout on a copper paper whenever it gets a print command from the computer.

STEP 6: STRIPPING AND ETCHING

This process involves removing the unwired copper on the PCBs by using different types of chemicals like ferric chloride, ammonium per-sulphate, etc. Make the solvent by mixing 1% of sodium hydroxide and 10 grams of sodium hydroxide pellets to one liter of water and mix it until everything is dissolved. Next, the PCB is put on a chemical bowl and cleaned up with a brush. During this process, if the PCB is still greasy, due to applied sunflower or seed oil, the developing process may take about 1 minute.

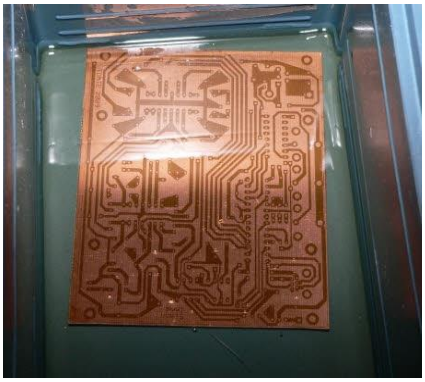

FIG.3.24: STRIPPING AND ETCHING OF PCB

The film is generated from the finalized circuit board diagram of the PCB layout software which is send to the manufacturing unit where the negative image or mask is printed out on a plastic sheet.

After finishing the manufacturing process of the Printed Circuit Board, the Board undergoes a testing process to check whether the PCB is working properly. Nowadays many automatic testing equipment's are available for the high-volume testing of the PCBs.

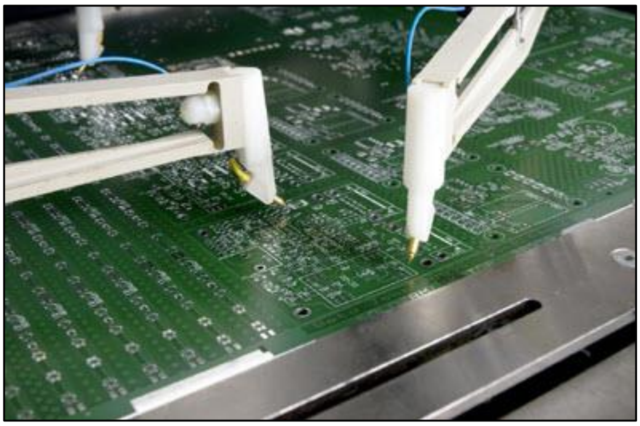

FIG.3.25: TESTING PCB

The two different types of testing equipment available today that test your circuit boards include ATG test machines that are flying probe, fixtureless testers and in addition to a universal grid testing capability as well.

PCB PREPARATION

For PCB layout, the following points ought to be considered carefully.

- ✓ Record size of components used.
- ✓ Overall area covered is normally kept rectangular or square.
- ✓ Vcc and ground lines should be provided at the sides to facilitate external connection.
- ✓ Input and output terminals may be placed giving through to external connection.
- ✓ Make a rough sketch placing components and interconnect components with jumpers.
- ✓ Do not place components pointing in differed direction unless needed. Make them parallel to the either side of the board.

- ✓ Make the neat final scaled sketch on the inch graph sheet.
- ✓ Lines mounted are of uniform width.
- ✓ Invest the layout to confirm that all the components are connected properly and given sufficient place in the layout

SOLDERING:

PCB soldering required proper soldering technique, as explained below:

1. A light duty soldering iron of 25W or 30 W rating should be used to prevent damage to the printed circuit wiring due to excessive heating. The tip of soldering iron should not have an oxide coating. Clean it using sand paper.

2. Do not use excess solder to avoid solder flouring to adjacent conducing paths forming bridges, which cause short circuits.

3. Clean the surface of traces before you start soldering. It is advisable to use flux. Layout of desired circuit diagram and preparation is first and most important operation in any printed circuit board manufacturing process. First of all, layout of component side is to be made in accordance with available components dimensions. The following points are to be observed while forming the layout of P.C.B:

- Between two components, sufficient space should be maintained.
- High wattage/max, dissipated components should be mounted at a sufficient distance from semiconductors and electrolytic capacitors

3.3. SYSTEM DESIGING

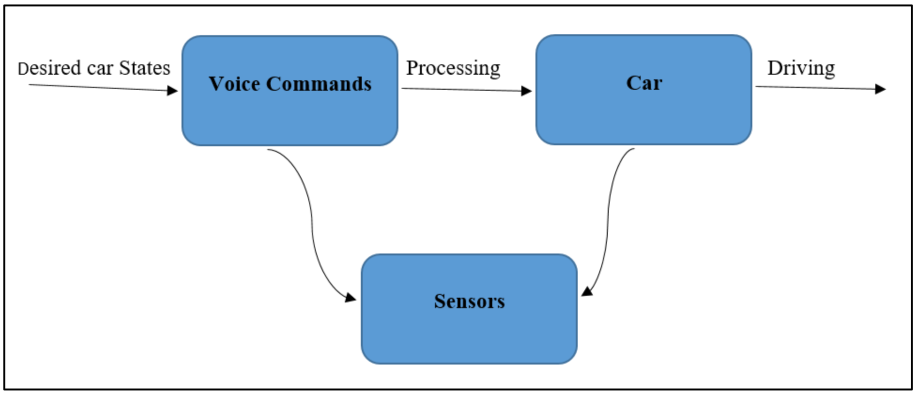

FIG.3.26: BLOCK DIAGRAM OF SYSTEM

The Fig 3.2 is the block diagram voice control robotic car. The car is control by voice command. Human voice is recognizing, the voice command is convert into text command. This text command is compare with code written in Arduino Uno programming, if the command is match the car is moving like forward, back, left, right this process is work through by Bluetooth module. This robotic car has other sensor like L293d motor driving IC, LDR, LED, SR-04 ULTRASONIC SENSOR ETC.

SYSTEM ARCHITECTURE:

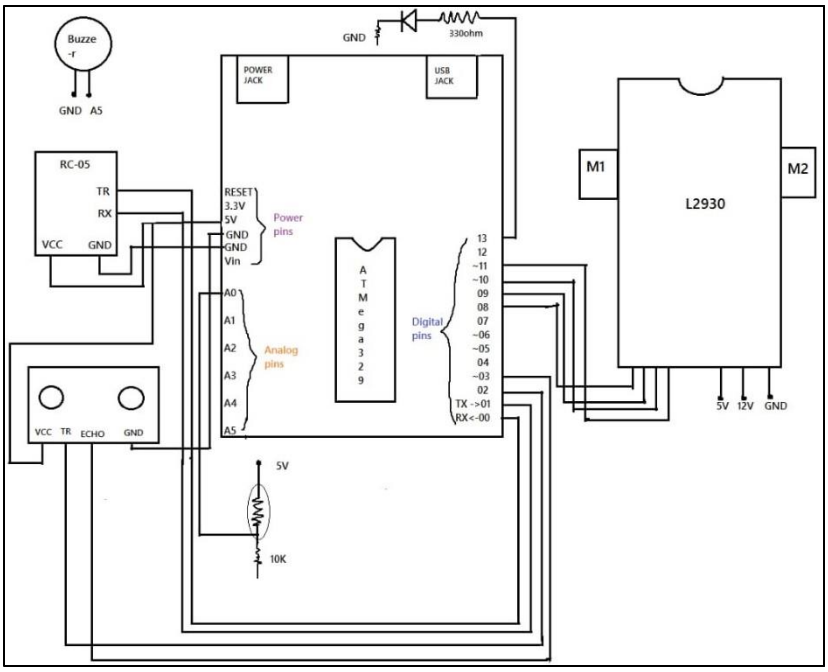

FIG.3.27: SYSTEM ARCHITECTURE

The Fig 3.27 Shown Voice controlling robotics car using Arduino system Architecture. In this different circuit board Connected to the Arduino circuit for making voice controlled. +5 V power supply given to the Arduino circuit to Active Arduino board. 6 Analog pins are used for Analog signal and 13 digital pins are used for receiving and transmitting signals.

HC SR 05 is a Bluetooth module connected to Arduino Uno board for establishing serial communication between mobile and robot. HC SR 05 module have 4 pins VCC, ground, Rxd, Txd.The VCC is connected to+5-volt pin of Arduino board, Gnd pin Hc 05 is connected to ground pin of pin of Arduino board, Rxd pin i.e. receive is connected to TX pin of Arduino board, Txd pin i.e. Transmit Pin (TX) Connected to the Receiver Pin (Rx) of Arduino Circuit. Receiver pin of HR SC 05 Connect to the TX pin of Arduino Circuit.

Ultrasonic sensor HC SR has 4 pins. Vcc pin connect to the Power Supply pin (Vcc) and GND pin connect to the GND of Arduino Circuit. Trigger input signal pin connected to the Pin 2 Digital pin and echo output signal pins connect to the Pin 3 of Arduino Circuit board.

Photoelectric buzzer has Two Pins One pin connected to the Digital of Arduino and one pin connected to the GND pin of Arduino Circuit board. In this system two LED's used. Two pins of Led connected to the

Analog pin of Arduino circuit for Analog signal input and output. Two remaining two pins of Led connect to GND using 10K register.

The L293D module is used to run DC motors. Two DC motors are connected to L293D module. Each motor has two connection this pin is connected to digital pin of Arduino Uno board. One is Vcc pin L293D IC required s+5V power supply to run. This pin another Vcc pin is used to give power supply to run DC motors. Motor required +12V power supply and last is ground in is connect to GND pin of Arduino Uno circuit board.

3.3.1 DATA FLOW DIAGRAM

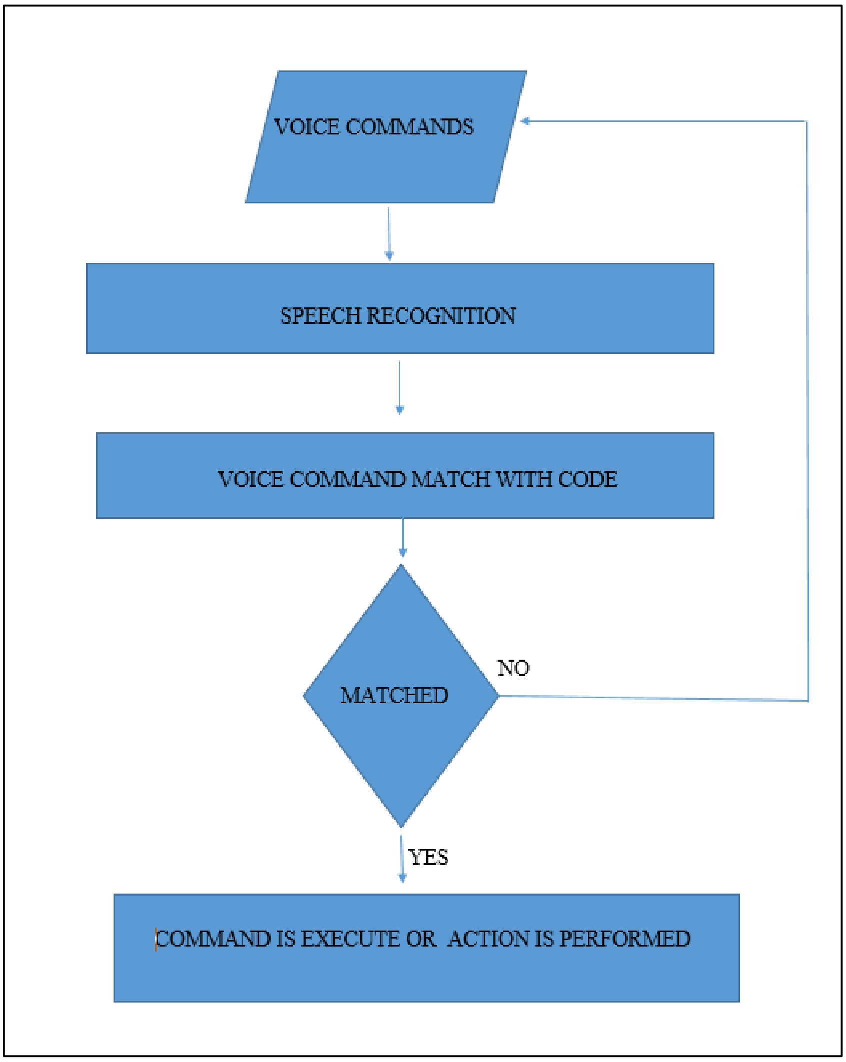

FIG 3.28 DFD

3.4 SAMPLE CODE

Arduino Uno Code:

```
char data;
int m1_c = 8;
```

```
int m1_a = 9;
int m2_c = 10;
int m2_a = 11;
const int trigPin1 = 6;
const int echoPin1 = 7;
int buz = A5;
int led1 = A4;
int led2 = A3;
void setup ()
{

  Serial.begin(9600);
  pinMode (m1_c, OUTPUT);
  pinMode (m1_a, OUTPUT);
  pinMode (m2_c, OUTPUT);
  pinMode (m2_a, OUTPUT);
  pinMode (buz, OUTPUT);
  pinMode (led1, OUTPUT);
  pinMode (led2, OUTPUT);

  digitalWrite (m1_c, HIGH);
  digitalWrite (m1_a, HIGH);
  digitalWrite (m2_c, HIGH);
  digitalWrite (m2_a, HIGH);
  digitalWrite (buz, LOW);
  digitalWrite (led1, LOW);
  digitalWrite (led2, LOW);
  delay (1000);
}

void loop ()
 {
  data = Serial.read();
```

```
long duration1, inches1, cm1;
pinMode (trigPin1, OUTPUT);
digitalWrite (trigPin1, LOW);
delayMicroseconds (2);
digitalWrite (trigPin1, HIGH);
delayMicroseconds (10);
digitalWrite (trigPin1, LOW);
pinMode (echoPin1, INPUT);
duration1 = pulseIn (echoPin1, HIGH);
inches1 = microsecondsToInches (duration1);
cm1 = microsecondsToCentimeters (duration1);

if (analogRead(A0) <= 500)
{
  digitalWrite (led1, HIGH);
  digitalWrite (led2, HIGH);
}
else
{
  digitalWrite (led1, LOW);
  digitalWrite (led2, LOW);
}

if (cm1 >= 20)
{
  digitalWrite (buz, LOW);

  if (data == 'a')
  {

    digitalWrite (m1_c, HIGH);
    digitalWrite (m1_a, LOW);
```

```
  digitalWrite (m2_c, HIGH);
  digitalWrite (m2_a, LOW);
  delay (10);
}

if (data == 'b')
{

  digitalWrite (m1_c, LOW);
  digitalWrite (m1_a, HIGH);
  digitalWrite (m2_c, LOW);
  digitalWrite (m2_a, HIGH);
  delay (10);

}
if (data == 'c')
{

  digitalWrite (m1_c, HIGH);
  digitalWrite (m1_a, LOW);
  digitalWrite (m2_c, HIGH);
  digitalWrite (m2_a, HIGH);
  delay (10);

}
if (data == 'd')
{
  digitalWrite (m1_c, HIGH);
  digitalWrite (m1_a, HIGH);
  digitalWrite (m2_c, HIGH);
  digitalWrite (m2_a, LOW);
  delay (10);
}
```

```
    if (data == 'e')
    {
      digitalWrite (m1_c, HIGH);
      digitalWrite (m1_a, HIGH);
      digitalWrite (m2_c, HIGH);
      digitalWrite (m2_a, HIGH);
      delay (10);
    }
  }

  else

  {

    digitalWrite (buz, HIGH);
    digitalWrite (m1_c, HIGH);
    digitalWrite (m1_a, HIGH);
    digitalWrite (m2_c, HIGH);
    digitalWrite (m2_a, HIGH);

 }

}

long microsecondsToInches (long microseconds)
{
  return microseconds / 74 / 2;
}
long microsecondsToCentimeters (long microseconds)
{
  return microseconds / 29 / 2;
}
```

3.5. OUTPUT OF PROJECT

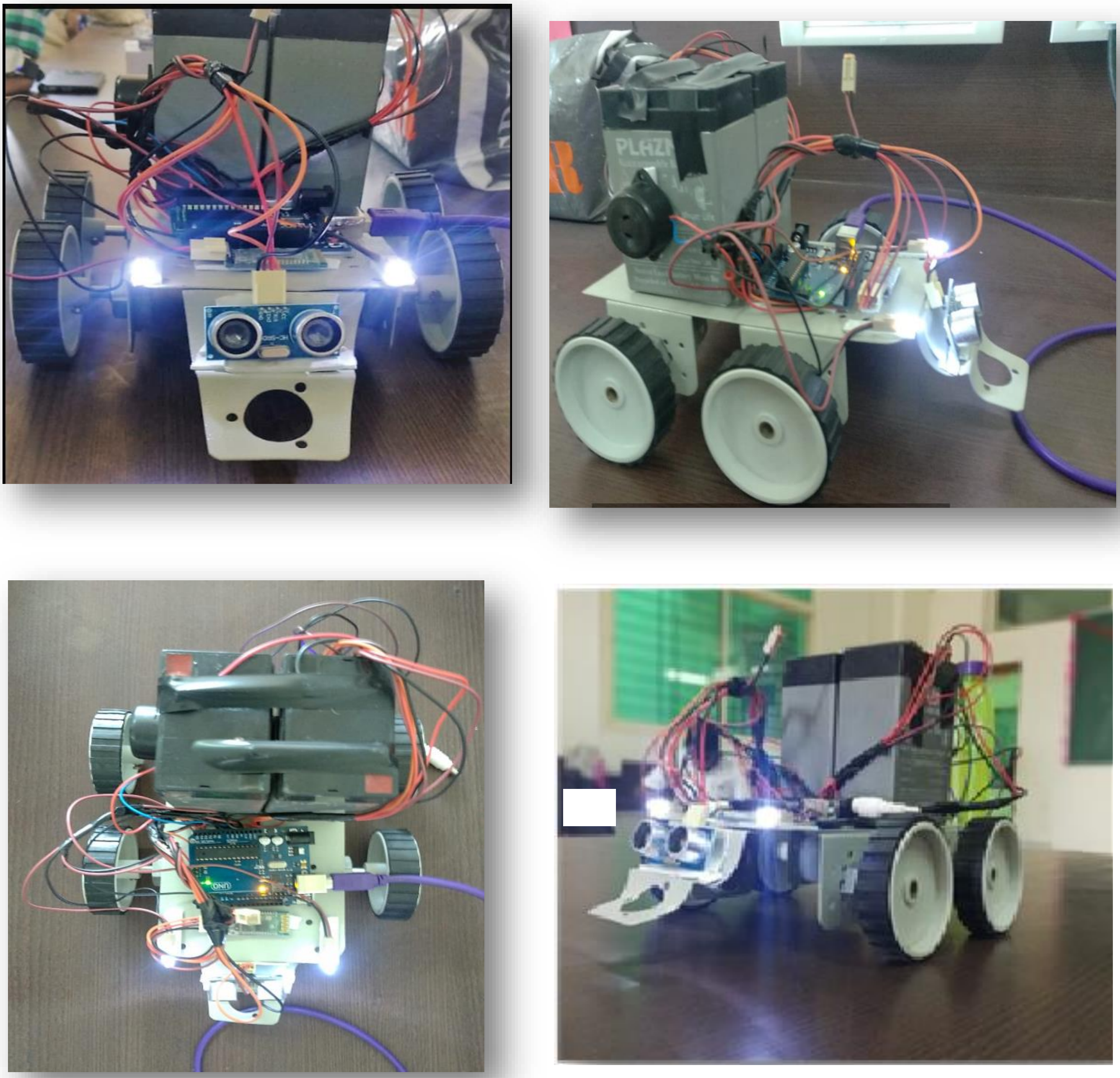

FIG 4.1. PROJECT SNAPSHOTS

4. PERFORMANCE ANALYSIS

4.1 TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behaviour, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software, we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally, we arrive at system testing, where the software and other system elements are tested as a whole.

TYPES OF TESTING:

WHITE BOX TESTING

It is testing of a function with knowing internal structure of the program. It is also known as glass box, structural, clear box and opens box testing. A software testing techniques whereby explicit knowledge of the internal working of the items is being tested is used to select the test data. Unlike black box testing, white box testing uses specific knowledge of programming code to examine outputs. The test is accurate only if the tester knows what the program is supposed to do.

He or she can then see if the program diverges from its intended goal. White box testing does not account for errors caused by omission, and all visible code must also be readable.

BLACK BOX TESTING

A black box testing of integration builds includes functional, interface, error recovery, and stress and out of bound input testing. All black box software tests are traced to control requirements. In addition to static requirements a black box of a fully integrated system against scenario sequence of events is designed to model fields operation. Performance testing in integrated as an internal part of the black box testing. Synonyms for black box include: behavioural, functional, opaque-box and closed box. White box test design allows one to peek inside the “box”, and it focuses specifically on using internal knowledge of the software to guide the selection of the test data.

UNIT TESTING

In computer programming, a unit test is a method of testing the correctness of a particular module of source code. The idea is to write test cases for every non-trivial functions or method in the module so that each test case is separate from the others if possible. This type of testing is mostly done by the developers. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. It provides a written contract that the piece must satisfy. This isolated testing provides four main benefits. Unit testing allows the programmer to refractor code at a later date, and make sure the module still works correctly (regression testing). This provides the benefit of encouraging programmers to make changes to the code since it is easy for the programmer to check if the piece is still working properly.

INTEGRATION TESTING

Integration testing was carried out to check whether each module works properly with the integrated module. Different inputs were given to the secondary module from the primary to check whether it works properly for each input which comes from the higher level. Both Bottom Up and Top-Down tests were carried out. The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. These “design items”, i.e. assemblages (or group of units), are exercised through their interfaces using Black box testing, success and errors cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter process communication is tested; individual subsystems are exercised through their input interface. All test cases are constructed to test that all components within assemblages interact correctly, for example, across procedure calls and process activations

PERFORMANCE TESTING

In software engineering, performance testing is testing that is performed to determine how fast some aspect of a system performs under a particular workload. Performance testing can serve different purposes. It can demonstrate that the system meets performance criteria. It can compare two systems to find which performs better. Or it can measure what parts of the system or workload cause the system to perform badly. In the diagnostic case, software engineers use tool such as profilers to measure what parts of a device or software contribute most to the poor performance or to establish throughput levels for maintained acceptable response time.

STRESS TESTING

Stress testing is a form of testing that is used to determine the stability of a given system or entity. It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results. For example, a web server may be stress tested using scripts, bots, and various denials of service tools to observe the performance of a web site during peak loads

ALPHA TESTING

In software development, testing is usually required before release to the general public. Inhouse developers often tests the software in what is known as ‘alpha’ testing which is often performed under a debugger or with hardware- assisted debugging to catch bugs quickly. It can then be handed over to testing staff for additional inspection in an environment similar to how it was intended to be used. This technique is known as black box testing. This is often known as the second stage of alpha testing

BETA TESTING

In software development, testing is usually required before release to the general public. Inhouse developers often tests the software in what is known as ‘Beta’ testing which is often performed under a debugger or with hardware- assisted debugging to catch bugs quickly. It can then be handed over to testing staff for additional inspection in an environment similar to how it was intended to be used. This technique is known as black box testing. This is often known as the second stage of beta testing.

TEST CASES, SUITS AND SCRIPTS

Black box testers usually write test cases for the majority of their testing activities. A test case is usually a single step, and its expected result, along with various additional pieces of information. It can occasionally be a series of steps but with one expected result or expected outcome. In a database system, you may also be able to see past test results and who generated the result and the system configuration used to generate those results. These past results would usually be stored in a separate table.

The most common term for a collection of test cases is a test suite. The test suite often also contains more detailed instructions or goals for each collection of test cases. It definitely contains a section where the tester identifies the system configuration used during testing. A group of test cases may also contain prerequisite states or steps, and descriptions of the following tests. Collections of test cases are sometimes incorrectly termed a test plan. They may also be called a test script, or even a test scenario. Most white box testers write and use test scripts in unit, system, and regression testing. Test scripts should be written for modules with the highest risk of failure and the highest impact if the risk becomes an issue. Most companies that use automated testing will call the code that is used their test scripts.

Verification is one aspect of testing a product's fitness for purpose. Validation is the complementary aspect. Often one refers to the overall checking process as Verification & Validation (V & V): "Are we building the right product?" i.e. does the product do what the user really requires. Verification: "Are we building the product, right?" i.e. does the product conform to the specifications. The verification process consists of static and dynamic parts. E.g. for a software product one can inspect the source code (static) and run against specific test cases (dynamic). Validation usually can only be done dynamically, i.e. the product is tested by putting it through typical usages and atypical usages ("can we break it?").

4.1.1 TEST CASES

- DC motors is rotating properly
  Test to pass:
  SR_04 Not detecting Objects from the path accurately and controlling the car.
  Test to fail:
  If object is not detected by SR-04 module

- Object is being not detected

  Test to pass:

  The car will respond that something is going wrong with it.

  Test to fail:

  Car won't run according to detected objects.

- Wireless Connection password
  Test to pass:
  Authenticated password will allow a user to access the car.
  Test to fail:
  Device will give the pairing problem.

- Obstacle Detection
  Test to pass:
  If there is an obstacle then car will stop.
  Test to fail:
  If there is an obstacle then also car doesn't stop.

5. CONCLUSION

5.1. APPLICATION

- ✓ In Spying Operations: This robot can help in spying operations. The object recognition and android control makes it Hi-Fi.
- ✓ For Handicapped People: This project can help the handicapped people especially those who had lost their feet unfortunately.
- ✓ Robot Races: The tilt control of robots can be used in robot races which will be revolutionary.
- ✓ The Robot is useful in places where humans find difficult to reach but Human voice reaches. E.g.: a) in highly toxic areas b) in fire situations c) in pipeline
- ✓ It is the one of the important stage of Humanoid robots.

5.2. ADVANTAGE

- ✓ The robot is small in size.
- ✓ We can access the robot form the distance of meters as we are using Bluetooth
- ✓ It reduces the human efforts

5.3. DISADVANTAGE

- ✓ Even the best speech reorganization systems sometimes make errors, if there is noise or some other sound in the room the number of errors will increase.
- ✓ Speech reorganization works best if the microphone is close to the used
- ✓ In Speech reorganization system, there is a possibility of unauthorized usage, since this doesn’t depend upon which person is speaking

5.4. CONCLUSION

- ✓ The integration of voice reorganization system into robotics vehicle which help for disable people.
- ✓ The speech control system, though quite simple, show the ability to apply speech reorganization technique to control the application.
- ✓ The method its provide real time operation, in this system android application is used to recognize human voice and is converted to text, the text is further processed and use to control robotics movements.

5.6 FUTURE SCOPE

The knowledge is ever expanding and so are the problems which the mankind strive to solve. In this spirit, it is hoped that the current activity will lead to further enhancements. For example; work on future for military purpose by the robot.

Thus a solution that can be derived from this is to introduce a voice control interface that would enable the users to interact with the controller with just the help of voice commands. The voice control interface helps in attaining a hands-free control that can be implemented on any kind of wheel chair and can also be used by those who are unable to monitor using the joysticks. This kind of system can be implemented in several other systems too. The motorized wheelchair can move at a fair speed it is important that it be able to avoid obstacles automatically in real time. The Voice controlled robot, a prototype to demonstrate the voice control mechanism that can be implemented for a wheelchair, is controlled with the help of voice activated - Arduino Uno microcontroller board. This system utilizes an Arduino Uno, a Bluetooth module- HC05 and a smartphone to control the motors driving the robot. The commands for the robot are sent via Android speech recognition app created using the MIT Inventor software. These voice commands are sent through the app via Bluetooth as a string of data to the Arduino. The microcontroller then processes these data strings and correspondingly controls the motors of the robot. The command is also displayed on the app as well as the com port connecting the Arduino module in the IDE.

The automatic voice control robot utilises a simple voice app that recognises voice commands. The app could be further developed using background noise eliminating tools in order to capture only the command and hence making the system more efficient. There is also a vast research undergoing in the stream of sensors used in smartphones that eliminate noise and detect only the required speech signals. Using such advanced signal processing techniques, the wheelchair mechanism can be efficiently developed and implemented.

REFERANCES

[1] Aniket R. Yeole, Sapana M. Bramhankar, Monali D. Wani, "Smart Phone Controlled Robot Using ATMEGA328 Microcontroller", ISO 3297: 2007 Pg:352-356

[2] R. M. Narayana, Harsha Chapala, "Voice Control Robot using Android Application “, Volume :4, ISSN: 2277 5668 Pg:332-337

[3] Ritika Pahuja, Narender Kumar, "Android Mobile Phone Controlled Bluetooth

Robot Using 8051 Microcontroller”, ISSN (Online): Pg:2347 – 3878

[4] K. Kannan, Dr. J. Selvakumar, “Arduino Based Voice Controlled Robot “, Volume :02 issue :01, Mar-2015 ISSN: 2395-0072

[5] Mrumal.K. Pathak, Javed Khan, “ROBOT CONTROL DESIGN USING ANDROID

SMARTPHONE ",2 Feb 2015, ISSN :2347-5471

[6] Pratik Chopra, Harshad Dange, "VOICE CONTROLLED ROBOT" Pg:31- 33

[7] Kishan Raj KC, "CONTROLLING A ROBOT USING ANDROID INTERFACE

AND VOICE" Pg:1-36

[9] R. M. Narayana, Harsha Chapala, "VOICE CONTROL ROBOT BY ANDROID"

APP Pg:49-54

[8] Harshad Dange, “Smart Phones Android Operated Robot" Pg:551-567