A REPORT ON MAJOR PROJECT

MOBILE GESTURE CONTROLLED ROBOT USING ARDUINO

SUBMITTED IN PARTIAL FULFILLMENT FOR AWARD OF DEGREE OF

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

BY

Kapil Mahajan (1702078)

Dixant Dutt (1702075)

Varinder Karlupia (1702092)

UNDER THE GUIDANCE OF

Mr. Parveen Kumar (Assist. Prof, ECE)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BEANT COLLEGE OF ENGINEERING AND TECHNOLOGY, GURDASPUR

JULY-DEC. 2020

CANDIDATE'S DECLARATION CERTICICATE

We hereby certify that the work which is being presented in the Project entitled “Mobile Gesture Controlled Robot using Arduino” by “Kapil Mahajan, Dixant Dutt and Varinder Karlupia” in partial fulfillment of requirements for the award of degree of B.Tech. (ECE) submitted to Department of Electronics and Communication Engineering at Beant College of Engineering and Technology, Gurdaspur. under PUNJAB TECHNICAL UNIVERSITY, JALANDHAR is our own work carried out during a period from July to November, 2011 under the supervision of NAME OF GUIDE. The matter presented in this thesis has not been submitted by us in any other University / Institute for the award of B.Tech. Degree.

Signature of the Students

(Kapil Mahajan - 1702078)

(Dixant Dutt - 1702075)

(Varinder Karlupia - 1702092)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Signature of the Project Guide
Name:- Parveen Kumar
Designation: (Assist. Prof, ECE)

Abstract

In this project we have designed a basic robot that can be easily controlled with a help of accelerometer and gyroscope sensor which is already present in most of Smartphone's. Accelerometers in mobile phones are used to detect the orientation of the phone. The gyroscope, or gyro for short, adds an additional dimension to the information supplied by the accelerometer by tracking rotation or twist.

An accelerometer measures linear acceleration of movement, while a gyro on the other hand measures the angular rotational velocity. Both sensors measure rate of change; they just measure the rate of change for different things.

In practice, that means that an accelerometer will measure the directional movement of a device but will not be able to resolve its lateral orientation or tilt during that movement accurately unless a gyro is there to fill in that info.

It is a 3-axis estimation gadget, so the mobile position is sensed and co ordinates are generated also the necessary condition and the statement prescribed to arduino code is executed and the motor driver works accordingly. So that it can perform the task such as forward moving, backward moving, turning left, turning right and stop. In many cases, the robot devices are some tough and complex while we control it with the help of buttons and switches.
Sometimes the operator may get confuse in the switch control and button control, so a new concept is used to manage the gadget with the motion of the mobile and at the same time it will manage the motion of the device.

ACKNOWLEDGEMENT

The project titled Mobile Gesture controlled Robot using Arduino, which we have taken as our Final year project work, has been a great experience for all us. We have learnt a lot of things while doing this project. We would like to thank all the faculties of ECE department for giving us this opportunity. We would specially like to thank Head of the Department of ECE **Dr. Jaswinder singh** and mentor of our group **Mrs. Neetu sagar**, without whose support implementing this project would not have possible. Also our special thanks to all class fellows and seniors, who helped us in clarification of any issue as well as implementation and in documentation.

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF TABLES

CHAPTER- I
INTRODUCTION

1.1 INTRODUCTION

Robots are used to do the work, that human can't perform. To raise the usage of robotics where restrictions that are not mandatory; for example, fire handling task or protection task. The device receives the input from the user and work out according to the received input.

The robots travel by motion made by the user mobile tilting. The objective of this gesture control device is achieved using Arduino, accelerometer and gyroscope sensors. The Arduino microcontroller receives the analog input values (x axis, y axis) from the accelerometer and converts that analog value to digital value.

The input is received by the Arduino Uno microcontroller after mobile titling. The robot turns left and right when we tilt our mobile to left and right, it moves towards front when the mobile tilted towards forward and the device halts while it is corresponding to the surface.

 # Some Important Terms

- ROBOT. A robot is usually electro-mechanical machine that can perform tasks automatically. Some robots require some degree of guidance, which may be done using a remote control or with a computer interface. Robots can be autonomous, semi-autonomous or remotely controlled .Robots have evolved so much and are capable of mimicking humans that they seem to have a mind of their own.
- HUMAN MACHINE INTERACTION. An important aspect of a successful robotic system is the Human-Machine interaction. In the early years the only way to communicate with a robot was to program which required extensive hard work. With the development in science and robotics, gesture based recognition came into life. Gestures originate from any bodily motion or state but commonly originate from the face or hand. Gesture recognition can be considered as a way for computer to understand human body language. This has minimize the need for the text interfaces and GUI(Graphical User Interface).

GESURE A gesture is an action that has to be seen by someone else and has to convey some piece of information. Gesture is usually considered as a movement of part of the body, esp. a hand or the head, to express an idea or meaning

1.2 Necessity

Nowadays, robotics are becoming one of the most advanced in the field of technology. The applications of robotics mainly involve in automobiles, medical, construction, defense and also used as a fire fighting robot to help the people from the fire accident. But, controlling the robot with a remote or a switch is quite complicated. So, a new project is developed that is, an accelerometer based mobile gesture control robot. The main goal of this project is to control the movement of the robot with hand gesture using accelerometer.

Also, we humans tend to protect ourselves and decrease our work. Therefore we developed this robot which is controlled by gesture which would not only reduces the complexity of design but also reduces the hardware and in turn reduce the cost
Also in this pandemic time we need a device that could aid us.

1.3 Motivation

As we know we are going through tough times due to pandemic era there is more need to look for our health . Also we need to socially distance ourselves from others , for this we built this project to develop a robot that can help us to maintain distance for example we need to provide food supply in a quarantine centre , we can use this robot to send food inside centre.

1.4 Objective

The main objective of this project is what all invention or gadgets based upon, to ease the human work load and to save the human lives from unnecessary danger, therefore objective of this project is to make a robot that could be controlled simple mobile gesture and could be handled pretty easily by anyone with a smartphone. The reason for making it simple was that it could be mass produced easily and there are so many ways it can aid people for example attach camera and use for survailecne, attach a basket and u can use it to maintain social distancing in shops .

CHAPTER- II
SYSTEM MODEL AND LITERATURE SURVEY

2.1. Introduction

Our project "Gesture controlled robot" works on the principle of Bluetooth which records the data and transfers to implement. The robot is integrated with Bluetooth Dongle HC05 that allows capturing and reading the commands. The robot is also integrated with an Arduino Nano that is used to operate the robot as per user commands. These decisions are passed to the Motor Driver IC L293D which triggers the motors in different configurations to make the robot in specific direction. The following block diagram helps to understand the working of the robot

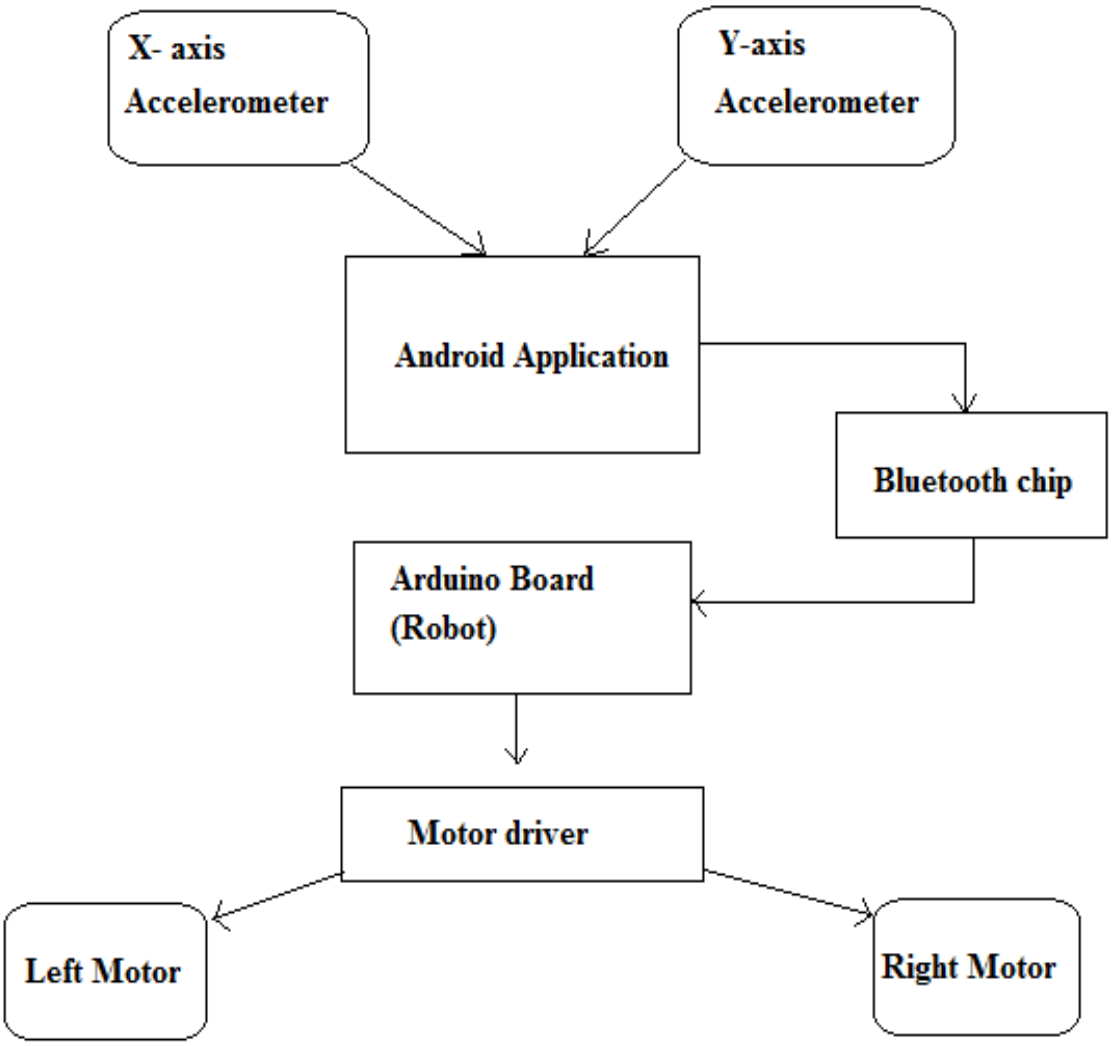

Fig.2.1. Block Diagram

2.2. Block Diagram

In order to understand the working operation of Mobile Gesture Controlled Robot, let us divide the project into three parts.

1. Android Application.

2. Arduino board.

3. Motor Driver.

1. Android Application: Here Android application is used for sending Data like where and how to move "ROBO CAR". In this, the application basically uses gyroscope and accelerometer sensor of mobile by which the car which we are using can move in different direction using this application. This application mainly Bluetooth sharing by which data transmit from one place to another place this application also convert that data into waveform which is received by another Bluetooth chip which is connected with Arduino Nano board.

2. Arduino board: Arduino board basically used here for receiving data from Bluetooth chip and convert that data into signal so that the data can forward to Motor driver.

3. Motor driver: Here motor driver main work is to receive data coming from Arduino board and work according to that data so that the motors which are connected to that motor driver can move in specific direction.

2.2.1 Description of Various Blocks:

1. Android Application: The Android application is the key to control the robot using hand gestures. The application reads the accelerometer state and X, Y, and Z values are obtained in the application. There are two threshold values assigned for each movement: one is the MAX_THRESHOLD, and the other is the MIN_THRESHOLD. If the obtained value lies between these thresholds of a certain movement, then the character assigned to denote that movement, which is called the DET or determinant is sent to the robot via Bluetooth. The application continuously sense this until the application is ON. A graphical user interface has been designed for the comfort of the user. The application abstracts the calculations and accelerometer values, but the user interface shows the direction of movement of the hand so that the user is aware of wrong turns in the bot.

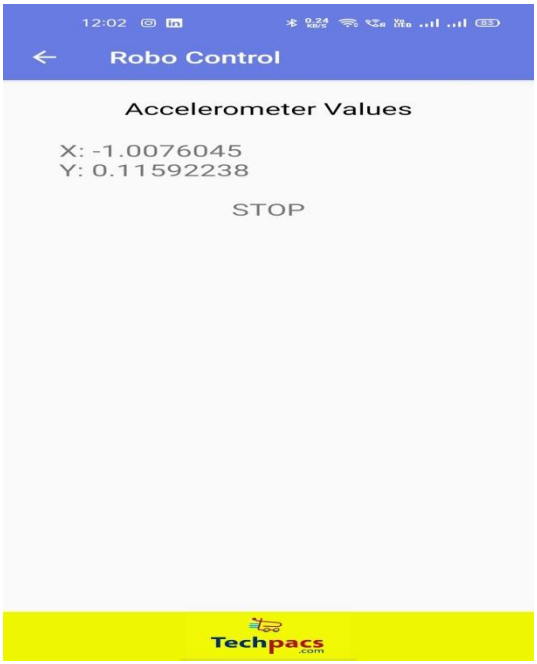

Figure 2.2: Screenshot of the Android application at the default state

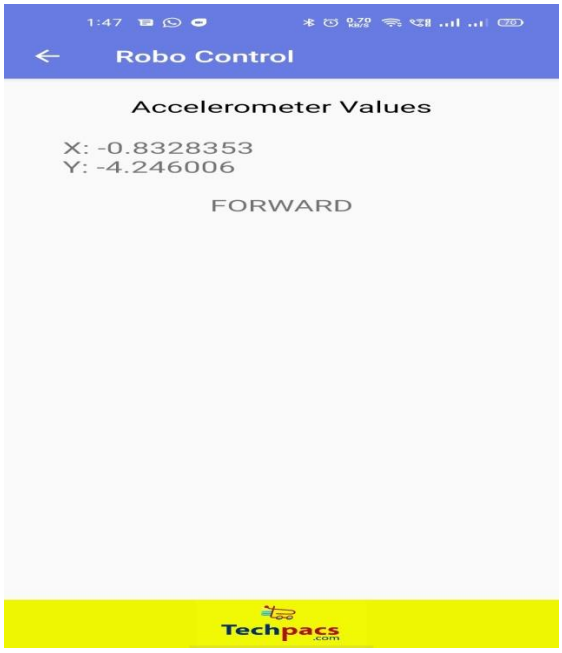

Figure 2.3: Screenshot of the Android application when the user moves his hand forwards

Arduino Nano ATmega328:

Arduino Nano is one type of microcontroller board, and it is designed by Arduino.cc. It can be built with a microcontroller like Atmega328. This microcontroller is also used in Arduino UNO.

It is a small size board and also flexible with a wide variety of applications. This board has many functions and features like an Arduino Duemilanove board. However, this Nano board is different in packaging. It doesn't have any DC jack so that the power supply can be given using a small USB port otherwise straightly connected to the pins like VCC & GND. This board can be supplied with 6 to 20volts using a mini USB port on the board.

Bluetooth Chip:

Wireless communication is swiftly replacing the wired connection when it comes to electronics and communication. Designed to replace cable connections HC-05 uses serial communication to communicate with the electronics. Usually, it is used to connect small devices like mobile phones using a short-range wireless connection to exchange files. It uses the 2.45GHz frequency band. The transfer rate of the data can vary up to 1Mbps and is in range of 10 meters. The HC-05 module can be operated within 4-6V of power supply. It supports baud rate of 9600, 19200, 38400, 57600, etc. Most importantly it can be operated in Master-Slave mode which means it will neither send nor receive data from external sources.

Motor Driver:

L293d IC is known as a motor driver. It is a low voltage operating device like other ICs. The other ICs could have the same functions like L293d but they cannot provide the high voltage to the motor. L293d provides the continuous bidirectional Direct Current to the Motor. The Polarity of current can change at any time without affecting the whole IC or any other device in the circuit. L293d has an internal H-bridge installed for two motors.

DC gear motor:

A gear motor is an all-in-one combination of a motor and gearbox. The addition of a gear head to a motor reduces the speed while increasing the torque output. The most important parameters in regards to gear motors are speed (rpm), torque (lb-in) and efficiency (%). In order to select the most suitable gear motor for your application you must first compute the load, speed and torque requirements for your application. ISL Products offers a variety of **Spur Gear Motors**, **Planetary Gear Motors** and **Worm Gear Motors** to meet all application requirements. Most of our DC motors can be complimented with one of our unique gearheads, providing you with a highly efficient gear motor solution.

2.3. Literature Survey:

Various researches have been made by various researchers but they serve in different applications and implement different hardware and technology.

- Jorge Kazacos Winter developed android controlled automation to transfer information wirelessly.
- M.Selvan used Bluetooth to connect its smartphone to robot.
- Vito M Guardi evolved the method of Bluetooth by developing an app for android phone which drives microcontroller.
- Arduino Bluetooth Controller by Ioannis Tzanellis.

CHAPTER-III

HARDWARE DISCRIPTION

3.1. Introduction:

As the user moves his smart phone, the accelerometer reading changes. It is then retrieved by the application. There are two values: One is maximum value and the other is minimum value. The range is specified using these two values for each function of the robot. If the value retrieved by the application lies between these specified values, then the corresponding determinant is generated. This determinant is sent to the microcontroller, which then receives the determinant value, process it to recognize the corresponding gesture, and sends signals to move the robot accordingly. When the user tilts his smart phone forward, the gesture is recognized as the forward movement, and the robot moves in the forward direction. The angle of the tilt or the difference between the angle of tilt of users smart phone and the threshold value of forward movement gesture determines the speed of the robot. When the user tilts his smart phone on the right direction, the gesture is recognized as the right turn, and the robot moves in the right direction. When the user tilts his smart phone in the left direction, the gesture is recognized as the left turn, and the robot moves in the right direction. The angle of the tilt of users smart phone determines whether the left or right turn is a normal turn or a sharp turn. A sharp turn is one in which a car changes direction without slowing down before turning. When the user tilts his smart phone backwards, the gesture is recognized as the move backward gesture, and the robot moves in the backward direction. If the users smart phone is somewhere between the two gestures, i.e., the accelerometer value is somewhere between the threshold of two directions(forward and left turn, left turn and backwards, backwards and right turn, forward and right turn), then the robot moves in that diagonal direction.

There are two DC motors used in the design of this robot, one motor for each rear wheel. The functions are called from the program burnt in the Arduino microcontroller. The signal is sent to the motor shield that runs the motors. The wheels are connected to the motors. Two DC motors are used one for left rear wheel, and another for right rear wheel. When the signal received in the motor shield is to move forward, both wheels of motors rotate forward, this turns all the three wheels in the forward direction. The robot moves in the forward direction. When the signal received in the motor shield is to turn the robot in the forward left direction, the left diagonal

motors are rotated backwards while the right diagonal motors are made rotated forwards. This makes the robot turn in the forward left direction. When the signal received in the motor shield is to turn the robot in the forward right direction, the right diagonal motors are rotated backward while the left diagonal motors are rotated forwards. This makes the robot turn in the forward right direction. When the signal in the motor shield is to move backward, both the pairs of the motors are rotated backwards resulting the robot to move backwards. When the signal in the motor shield is to stop the robot, all the motors are made stationary resulting the robot to stop. Similarly, to rotate the robot in backward directions, similar methodology is used. To turn the robot in the backward left direction, the left diagonal motors are rotated forwards while the right diagonal motors are rotated backwards. This makes the robot turn in the backward left direction. To turn the robot in the backward right direction, the right diagonal motors are rotated forwards while the left diagonal motors are rotated backwards. This makes the robot turn in the backward right direction.

3.2. CIRCUIT DIAGRAM:

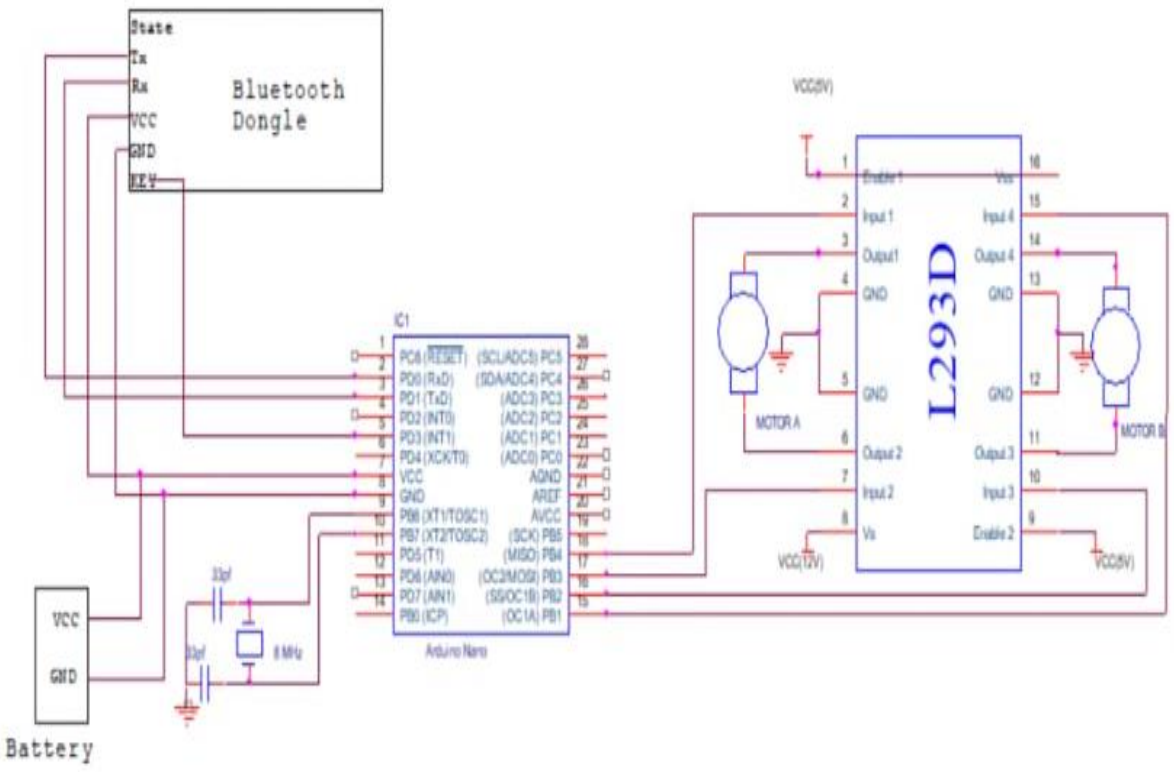

Fig.3.1. Circuit Diagram

Working: The proposed system allows the user to control the robo car from a safe distance using Bluetooth module chip HC05.Its working is very simple that you firstly needs to connect mobile Bluetooth with robo car Bluetooth, once Bluetooth gets paired just open the application by which you are going to control robo car. Once the connection get established successfully you just need to tilt your phone in particular direction; once the signal reached at other end (robo car Bluetooth) the signal transfer to the Arduino Nano chip and start processing that signal and send that particular command to the L293D chip then that chip decide which motor rotates in which direction.

3.2.1 LIST OF COMPONENTS USED:

- Bluetooth chip
- Arduino Nano
- L293D Motor Driver IC
- DC Geared Motors
- Wheels
- Robot Chassis
- Battery's
- Voltage regulator
- Resistors
- Diode's
- LED'S
- Dotted PCB
- Probes or wires

1. BLUETOOTH MODULE: HC-05 Bluetooth Module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Its communication is via serial communication which makes an easy way to interface with

controller or PC. HC-05 Bluetooth module provides switching mode between master and slave mode which means it able to use neither receiving nor transmitting data.

Fig.3.2: HC 05 Bluetooth Module

PIN Configuration:

Pin Number	Pin Name	Description
1	Enable / Key	This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default it is in Data mode
2	Vcc	Powers the module. Connect to +5V Supply voltage
3	Ground	Ground pin of module, connect to system ground.
4	TX – Transmitter	Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data.
5	RX – Receiver	Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth
6	State	The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly.
7	LED	Indicates the status of Module • Blink once in 2 sec: Module has entered Command Mode • Repeated Blinking: Waiting for connection in Data Mode • Blink twice in 1 sec: Connection successful in Data Mode
8	Button	Used to control the Key/Enable pin to toggle between Data and command Mode

Table No. 3.1: Pin configuration of HC-05

HC-05 Default Settings:

Default Bluetooth Name: HC-05

Default Password: 1234 or 0000

Default Communication: Slave

Default Mode: Data Mode

Data Mode Baud Rate: 9600, 8, N, 1

Command Mode Baud Rate: 38400, 8, N, 1

Default firmware: LINVOR

Technical Specifications:

- Serial Bluetooth module for Arduino and other microcontrollers
- Operating Voltage: 4V to 6V (Typically +5V)
- Operating Current: 30mA
- Range: <100m
- Works with Serial communication (USART) and TTL compatible
- Follows IEEE 802.15.1 standardized protocol
- Uses Frequency-Hopping Spread spectrum (FHSS)
- Can operate in Master, Slave or Master/Slave mode
- Can be easily interfaced with Laptop or Mobile phones with Bluetooth
- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800

Where to use HC-05 Bluetooth module:

The **HC-05** is a very cool module which can add two-way (full-duplex) wireless functionality to our projects. We- can use this module to communicate between two microcontrollers like Arduino or communicate with any device with Bluetooth functionality like a Phone or Laptop. There are many android applications that are already available which makes this process a lot easier. The module communicates with the help of USART at 9600 baud rate hence it is easy to

interface with any microcontroller that supports USART. We can also configure the default values of the module by using the command mode. So if you looking for a Wireless module that could transfer data from your computer or mobile phone to microcontroller or vice versa then this module might be the right choice for you.

How to Use the HC-05 Bluetooth module:

The **HC-05** has two operating modes, one is the Data mode in which it can send and receive data from other Bluetooth devices and the other is the AT Command mode where the default device settings can be changed. We can operate the device in either of these two modes by using the key pin as explained in the pin description.

It is very easy to pair the HC-05 module with microcontrollers because it operates using the Serial Port Protocol (SPP). Simply power the module with +5V and connect the Rx pin of the module to the Tx of MCU and Tx pin of module to Rx of MCU as shown in the figure below

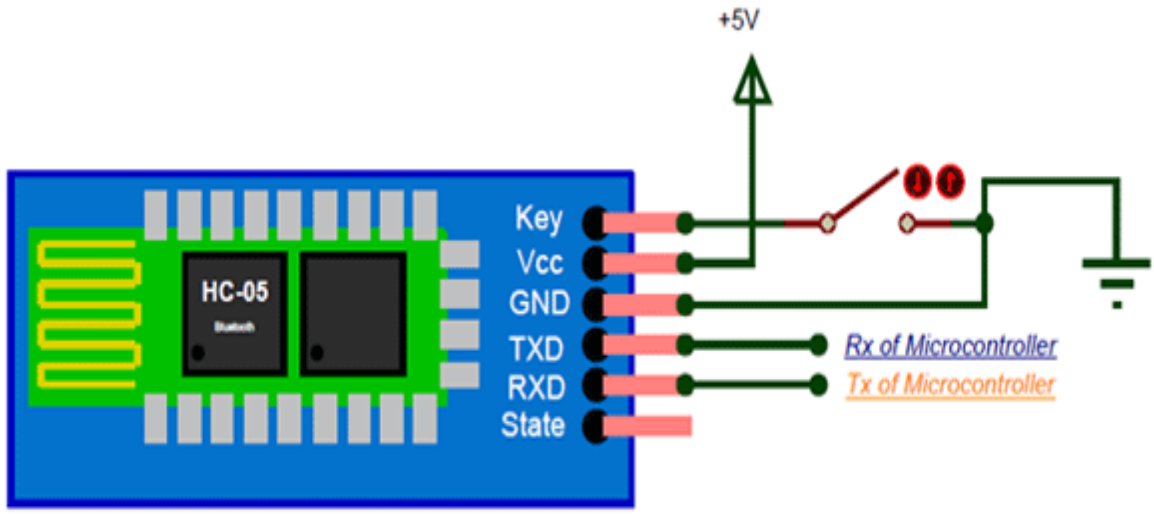

Fig.3.3: PIN Diagram

During power up the key pin can be grounded to enter into Command mode, if left free it will by default enter into the data mode. As soon as the module is powered you should be able to discover the Bluetooth device as “HC-05” then connect with it using the default password 1234 and start communicating with it.

2. ARDUINO NANO ATmega328:

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. Arduino Nano is a surface mount breadboard embedded version with integrated USB. It is a smallest, complete, and breadboard friendly. It has everything that Diecimila / Duemilanove has (electrically) with more analog input pins and onboard +5V AREF jumper. Physically, it is missing power jack. The Nano is automatically sense and switch to the higher potential source of power, there is no need for the power select jumper.

Fig.3.4: Arduino Nano ATmega328

Technical Specifications:

Microcontroller: ATmega328P – 8 bit AVR family microcontroller

Operating Voltage: 5V

Recommended Input Voltage for Vin pin: 7-12V

Analog Input Pins: 6 (A0 – A5)

Digital I/O Pins: 14 (Out of which 6 provide PWM output)

DC Current on I/O Pins: 40 mA

DC Current on 3.3V Pin4: 50 mA

Flash Memory: 32 KB (2 KB is used for Bootloader)

SRAM: 2 KB

EEPROM: 1 KB

Frequency (Clock Speed): 16 MHz

Communication: IIC, SPI, USART

PIN CONFIGURATION:

Pin Category	Pin Name	Details
Power	**Vin, 3.3V, 5V, GND**	**Vin:** Input voltage to Arduino when using an external power source (6-12V). **5V:** Regulated power supply used to power microcontroller and other components on the board. **3.3V:** 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. **GND:** Ground pins.
Reset	**Reset**	Resets the microcontroller.
Analog Pins	**A0 – A7**	Used to measure analog voltage in the range of 0-5V
Input/Output Pins	**Digital Pins D0 - D13**	Can be used as input or output pins. 0V (low) and 5V (high)
Serial	Rx, **Tx**	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	**13**	To turn on the inbuilt LED.
IIC	A4 (SDA), A5 (SCA)	Used for TWI communication.

Table no.3.2: Pin configuration of Arduino Nano

Understanding Arduino Nano:

The Arduino board is designed in such a way that it is very easy for beginners to get started with microcontrollers. This board especially is breadboard friendly is very easy to handle the connections. Let's start with powering the Board.

Powering you Arduino Nano: There are totally three ways by which you can power your Arduino board.

USB Jack: Connect the mini USB jack to a phone charger or computer through a cable and it will draw power required for the board to function

Vin Pin: The Vin pin can be supplied with an unregulated 6-12V to power the board. The on-board voltage regulator regulates it to +5V

+5V Pin: If you have a regulated +5V supply then you can directly provide this o the +5V pin of the Arduino.

Input/output: There are totally 14 digital Pins and 8 Analog pins on your Nano board. The digital pins can be used to interface sensors by using them as input pins or drive loads by using them as output pins. A simple function like pinMode() and digitalWrite() can be used to control their operation. The operating voltage is 0V and 5V for digital pins. The analog pins can measure analog voltage from 0V to 5V using any of the 8 Analog pins using a simple function liken analogRead().

These pins apart from serving their purpose can also be used for special purposes which are discussed below:

- **Serial Pins 0 (Rx) and 1 (Tx):** Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.
- **External Interrupt Pins 2 and 3:** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM Pins 3, 5, 6, 9 and 11:** These pins provide an 8-bit PWM output by using analogWrite() function.
- **SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK):** These pins are used for SPI communication.

- **In-built LED Pin 13:** This pin is connected with a built-in LED, when pin 13 is HIGH – LED is on and when pin13 is LOW, it's off.
- **I2C A4 (SDA) and A5 (SCA):** Used for IIC communication using Wire library.
- **AREF:** Used to provide reference voltage for analog inputs with analogReference() function.
- **Reset Pin:** Making this pin LOW, resets the microcontroller.

These special functions and their respective pins are illustrated in the **Arduino Nano pin diagram** shown below

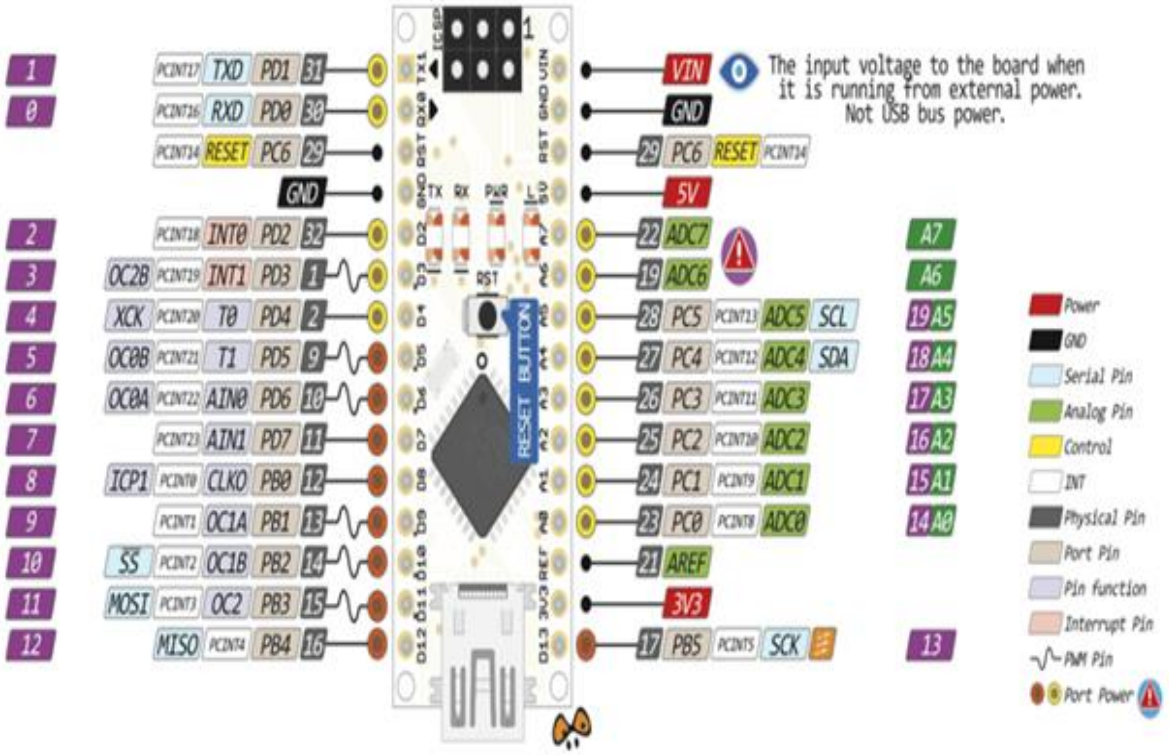

Fig.3.5: Arduino Nano pin diagram

How to use Arduino Nano:

It will hardly take 5-10 minutes to upload your program to Arduino Nano. All you need the Arduino IDE an USB cable and your Nano board itself.

Once Arduino IDE is installed on the computer, connect the board with computer using USB cable. Now open the Arduino IDE and choose the correct board by selecting Tools>Boards>Arduino/Nano, and choose the correct Port by selecting Tools>Port. Arduino

Uno is programmed using Arduino programming language based on Wiring. To get it started with Arduino Nano board and blink the built-in LED, load the code. Once the code is loaded into your IDE, click on the 'upload' button given on the top bar. Once the upload is finished, you should see the Arduino's built-in LED blinking.

3. MOTOR DRIVER IC L293D:

It is also known as H-Bridge or Actuator IC. Actuators are those devices which actually gives the movement to do a task like that of a motor. In the real world there are different types of motors available which work on different voltages. So we need a motor driver for running them through the controller. The output from the microcontroller is a low current signal. The motor driver amplifies that current which can control and drive a motor. In most cases, a transistor can act as a switch and perform this task which drives the motor in a single direction.L293D IC Turning a motor ON and OFF requires only one switch to control a single motor in a single direction. We can reverse the direction of the motor by simply reversing its polarity. This can be achieved by using four switches that are arranged in an intelligent manner such that the circuit not only drives the motor, but also controls its direction.

Fig.3.6: Motor Driver L293D

Technical Specifications:

- Can be used to run Two DC motors with the same IC.
- Speed and Direction control is possible

- Motor voltage Vcc2 (Vs): 4.5V to 36V
- Maximum Peak motor current: 1.2A
- Maximum Continuous Motor Current: 600mA
- Supply Voltage to Vcc1 (vss): 4.5V to 7V
- Transition time: 300ns (at 5Vand 24V)
- Automatic Thermal shutdown is available
- Available in 16-pin DIP, TSSOP, SOIC packages

PIN CONFIGURATION:

Pin Number	Pin Name	Description
1	Enable 1,2	This pin enables the input pin Input 1(2) and Input 2(7)
2	Input 1	Directly controls the Output 1 pin. Controlled by digital circuits
3	Output 1	Connected to one end of Motor 1
4	Ground	Ground pins are connected to ground of circuit (0V)
5	Ground	Ground pins are connected to ground of circuit (0V)
6	Output 2	Connected to another end of Motor 1
7	Input 2	Directly controls the Output 2 pin. Controlled by digital circuits
8	Vcc2 (Vs)	Connected to Voltage pin for running motors (4.5V to 36V)
9	Enable 3,4	This pin enables the input pin Input 3(10) and Input 4(15)
10	Input 3	Directly controls the Output 3 pin. Controlled by digital circuits

Pin Number	Pin Name	Description
11	Output 3	Connected to one end of Motor 2
12	Ground	Ground pins are connected to ground of circuit (0V)
13	Ground	Ground pins are connected to ground of circuit (0V)
14	Output 4	Connected to another end of Motor 2
15	Input 4	Directly controls the Output 4 pin. Controlled by digital circuits
16	Vcc2 (Vss)	Connected to +5V to enable IC function

Table no. 3.3: Pin configuration of L293D

Where to use L293D IC:

The L293D is a popular 16-Pin Motor Driver IC. As the name suggests it is mainly used to drive motors. A single L293D IC is capable of running two DC motors at the same time; also the direction of these two motors can be controlled independently. So if you have motors which has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, digital gates or even Micron rollers like Arduino, PIC, ARM etc.

How to use a L293D Motor Driver IC:

Using this L293D motor driver IC is very simple. The IC works on the principle of Half H-Bridge, let us not go too deep into what H-Bridge means, but for now just know that H bridge is a set-up which is used to run motors both in clock wise and anti-clockwise direction. As said earlier this IC is capable of running two motors at the any direction at the same time, the circuit to achieve the same is shown below.

All the Ground pins should be grounded. There are two power pins for this IC, one is the Vss (Vcc1) which provides the voltage for the IC to work, this must be connected to +5V. The other

is Vs (Vcc2) which provides voltage for the motors to run, based on the specification of your motor you can connect this pin to anywhere between 4.5V to 36V, here I have connected to +12V.

The Enable pins (Enable 1, 2 and Enable 3, 4) are used to Enable Input pins for Motor 1 and Motor 2 respectively. Since in most cases we will be using both the motors both the pins are held high by default by connecting to +5V supply. The input pins Input 1, 2 are used to control the motor 1 and Input pins 3, 4 are used to control the Motor 2. The input pins are connected to the any Digital circuit or microcontroller to control the speed and direction of the motor. You can toggle the input pins based on the following table to control your motor.

Motor1 Input1 = LOW	Motor2 Input1 = LOW	Robot moves in Backward Direction.
Motor Input2 = HIGH	Motor2 Input2 = HIGH	

Motor1 Input1 = HIGH	Motor2 Input1 = HIGH	Robot moves in Forward Direction.
Motor1 Input2 = LOW	Motor2 Input2 = LOW	

Motor1 Input1 = LOW	Motor2 Input1 = HIGH	Robot moves in Left Direction.
Motor1 Input2 = HIGH	Motor2 Input2 = LOW	

Motor1 Input1 = HIGH	Motor2 Input1 = LOW	Robot moves in Right Direction.
Motor1 Input2 = LOW	Motor2 Input2 = HIGH	

Motor1 Input1 = LOW	Motor2 Input1 = LOW	Robot don't moves in Direction. (STOP)
Motor1 Input2 = LOW	Motor2 Input2 = LOW	

Table no.3.4: Motors working system

4. DC GEAR MOTOR:

A geared DC Motor has a gear assembly devoted to the motor. The speed of motor is counted in terms of rotations of the shaft per minute and is termed as RPM .The gear assembly helps in increasing the torque and dropping the speed. Using the correct arrangement of gears in a gear motor, its speed can be reduced to any required figure. This concept of reducing the speed with the help of gears and increasing the torque is known as gear reduction. Reducing the speed put out by the motor while increasing the quantity of applied torque is an important feature of the reduction gear trains found in a gear motor.

Fig.3.7: DC Gear Motor

The decrease in speed is inversely relative to the increase in torque. This association means that, in this sort of device, if the torque were to double, the speed would decrease by one half. Small electric motors, such as the gear motor, are able to move and stand very heavy loads because of these reduction gear trains. While the speed and ability of larger motors is greater, small electric motors are sufficient to bear these loads.

5. Wheels:

Wheel design is simpler than using treads or legs and by using wheels they are easier to design, build, and program for movement in flat, not-so-rugged terrain. They are also better controlled than other types of robots. Disadvantages of wheeled robots are that they cannot navigate well over obstacles, such as rocky terrain, sharp declines, or areas with low friction. Wheeled robots are most popular among the consumer market; their differential steering provides low cost and simplicity. Robots can have any number of wheels, but three wheels are sufficient for static and dynamic balance. Additional wheels can add to balance; however, additional mechanisms will be required to keep all the wheels in the ground, when the terrain is not flat. Large diameter wheels give the robot low torque but high velocity.

Fig.3.8: Standard Wheel

Here in our project, we use 3-Wheeled robot.

3-wheeled robots may be of two types: differentially steered (2 powered wheels with an additional free rotating wheel to keep the body in balance) or 2 wheels powered by a single

source and a powered steering for the third wheel. In the case of differentially steered wheels, the robot direction may be changed by varying the relative rate of rotation of the two separately driven wheels. If both the wheels are driven

in the same direction and speed, the robot will go straight. Otherwise, depending on the speed of rotation and its direction, the center of rotation may fall anywhere in the line joining the two wheels. The center of gravity in this type of robot has to lay inside the triangle formed by the wheels. If too heavy of a mass is mounted to the side of the free rotating wheel, the robot will tip over.

In our project we are using differentially steered. Two are standard wheels and another one is ball wheel.

Fig.3.9: Ball Wheel

6. Chassis:

A chassis is the load-bearing framework of an artificial object, which structurally supports the object in its construction and function. It is analogous to an animal's skeleton. An example of a chassis is the under part of a motor vehicle, consisting of the frame (on which the body is mounted) with the wheels and machinery. Chassis main work is to hold the whole project in still situation by which during work user get safety as well as well finished work.

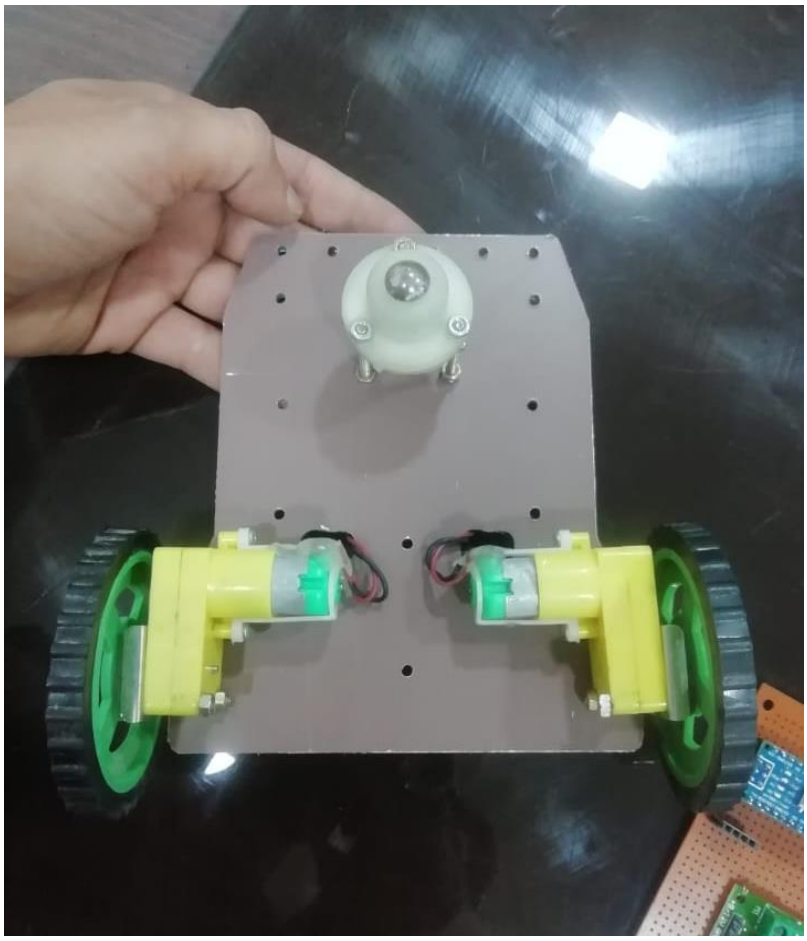

Fig.3.10: Chassis with motors and wheels

7. BATTERY:

A battery is a device consisting of one or more electrochemical cells with external connections for powering electrical devices such as flashlights, mobile phones, and electric cars. When a battery is supplying electric power, its positive terminal is the cathode and its negative terminal is the anode. The terminal marked negative is the source of electrons that will flow through an external electric circuit to the positive terminal. When a battery is connected to an external electric load, a redox reaction converts high-energy reactants to lower-energy products, and the free-energy difference is delivered to the external circuit as electrical energy. In our project we use 2 battery's of 4V each and 1.5Ah, which we use as 8V by connecting them in the series connection and our project requirement is of 5V.

Fig.3.11: Sealed lead acid battery

8. Voltage Regulator:

Voltage sources in a circuit may have fluctuations resulting in not providing fixed voltage outputs. A voltage regulator IC maintains the output voltage at a constant value. 7805 Voltage Regulator, a member of 78xx series of fixed linear voltage regulators used to maintain such fluctuations, is a popular voltage regulator integrated circuit (IC).

The xx in 78xx indicates the output voltage it provides. 7805 IC provides +5 volts regulated power supply with provisions to add a heat sink.

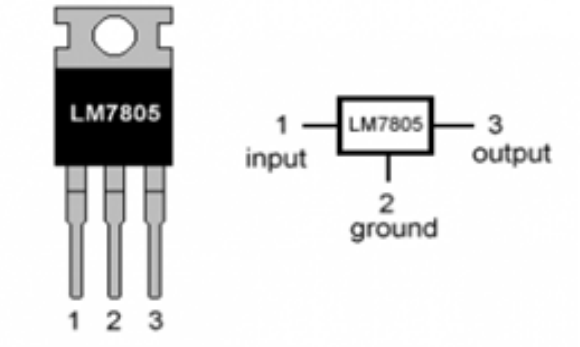

Fig.3.12: 7805 Voltage Regulator

There is a significant difference between the input voltage & the output voltage of the voltage regulator. This difference between the input and output voltage is released as heat. The greater the difference between the input and output voltage, more the heat generated.

If the regulator does not have a heat sink to dissipate this heat, it can get destroyed and malfunction. Hence, it is advisable to limit the voltage to a maximum of 2-3 volts above the output voltage. So, we now have 2 options. Either design your circuit so that the input voltage going into the regulator is limited to 2-3 volts above the output regulated voltage or place an appropriate heatsink that can efficiently dissipate heat.

9. Resistors:

A **resistor** is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, 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.

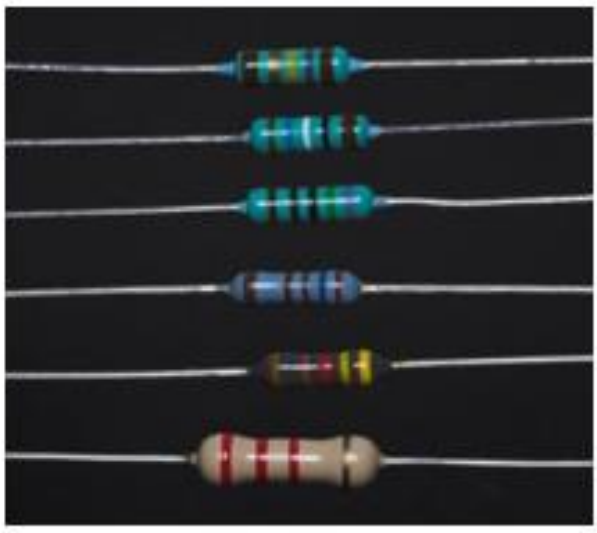

Fig.3.13: Resistors

Resistors are of following types:

1. Wire wound resistors.
2. Carbon resistors.
3. Metal film resistors.
4. Variable resistor.

Wire Wound Resistors:

Wire wound resistors are made from a long (usually Ni-Chromium) wound on a ceramic core. Longer the length of the wire, higher is the resistance. So depending on the value of resistor required in a circuit, the wire is cut and wound on a ceramic core. This entire assembly is coated with a ceramic metal. Such resistors are generally available in power of 2 watts to several hundred watts and resistance values from 1ohm to 100k ohms. Thus wire wound resistors are used for high currents.

Carbon Resistors:

Carbon resistors are divided into three types:

a. Carbon composition resistors are made by mixing carbon grains with binding material (glue) and moduled in the form of rods. Wire leads are inserted at the two ends. After this an insulating material seals the resistor. Resistors are available in power ratings of 1/10, 1/8, 1/4, 1/2, 1.2 watts and values from 1 ohm to 20 ohms.
b. Carbon film resistors are made by deposition carbon film on a ceramic rod. They are cheaper than carbon composition resistors.
c. Cement film resistors are made of thin carbon coating fired onto a solid ceramic substrate. The main purpose is to have more precise resistance values and greater stability with heat. They are made in a small square with leads.

Metal Film Resistors: They are also called thin film resistors. They are made of a thin metal coating deposited on a cylindrical insulating support. The high resistance values are not precise in value; however, such resistors are free of inductance effect that is common in wire wound resistors at high frequency.

Variable Resistors:

Potentiometer is a resistor where values can be set depending on the requirement. Potentiometer is widely used in electronics systems. Examples are volume control, tons control, brightness and contrast control of radio or T.V. sets.

Fig.3.14: Variable Resistor

10. Diode:

A diode is defined as a two-terminal electronic component that only conducts current in one direction (so long as it is operated within a specified voltage level).An ideal diode will have zero resistance in one direction, and infinite resistance in the reverse direction.

Although in the real world, diodes cannot achieve zero or infinite resistance. Instead, a diode will have negligible resistance in one direction (to allow current flow), and very high resistance in the reverse direction (to prevent current flow). A diode is effectively like a valve for an electrical circuit.

Semiconductor diodes are the most common type of diode. These diodes begin conducting electricity only if a certain threshold voltage is present in the forward direction (i.e. the “low resistance” direction). The diode is said to be “forward biased” when conducting current in this direction. When connected within a circuit in the reverse direction (i.e. the “high resistance” direction), the diode is said to be “reverse biased”.

Fig.3.15: Diode

The diode is said to be "forward biased" when conducting current in this direction. When connected within a circuit in the reverse direction (i.e. the "high resistance" direction), the diode is said to be "reverse biased".

A diode only blocks current in the reverse direction (i.e. when it is reverse biased) while the reverse voltage is within a specified range. Above this range, the reverse barrier breaks. The voltage at which this breakdown occurs is called the "reverse breakdown voltage".

When the voltage of the circuit is higher than the reverse breakdown voltage, the diode is able to conduct electricity in the reverse direction (i.e. the "high resistance" direction). This is why in practice we say diodes have a high resistance in the reverse direction – not an infinite resistance.

A PN junction is the simplest form of the semiconductor diode. In ideal conditions, this PN junction behaves as a short circuit when it is forward biased, and as an open circuit when it is in the reverse biased.

11. LED:

A light-emitting diode (LED) is a semiconductor device that emits light when an electric current is passed through it. Light is produced when the particles that carry the current (known as electrons and holes) combine together within the semiconductor material.

Since light is generated within the solid semiconductor material, LEDs are described as solid-state devices. The term solid-state lighting, which also encompasses organic LEDs (OLEDs),

distinguishes this lighting technology from other sources that use heated filaments (incandescent and tungsten halogen lamps) or gas discharge (fluorescent lamps).

Fig.3.16: LED's

Different colours:

Inside the semiconductor material of the LED, the electrons and holes are contained within energy bands. The separation of the bands (i.e. the bandgap) determines the energy of the photons (light particles) that are emitted by the LED.

The photon energy determines the wavelength of the emitted light, and hence its colour. Different semiconductor materials with different bandgaps produce different colours of light. The precise wavelength (colour) can be tuned by altering the composition of the light-emitting, or active, region.

LEDs are comprised of compound semiconductor materials, which are made up of elements from group III and group V of the periodic table (these are known as III-V materials). Examples of III-V materials commonly used to make LEDs are gallium arsenide (GaAs) and gallium phosphide (GaP).

Until the mid-90s LEDs had a limited range of colours, and in particular commercial blue and white LEDs did not exist. The development of LEDs based on the gallium nitride (GaN) material system completed the palette of colours and opened up many new applications.

Main LED materials:

the main semiconductor materials used to manufacture LEDs are:

- **Indium gallium nitride (InGaN):** blue, green and ultraviolet high-brightness LEDs
- **Aluminium gallium indium phosphide (AlGaInP):** yellow, orange and red high-brightness LEDs
- **Aluminium gallium arsenide (AlGaAs):** red and infrared LEDs
- **Gallium phosphide (GaP):** yellow and green LEDs

12. Dotted PCB Board:

This is the board on which basically our whole circuit is designed. It help use in to stable the components of project. It is a thin, rigid sheet with holes pre-drilled at standard intervals across a grid, usually a square grid of 2.54 mm (0.1 in) spacing. These holes are ringed by round or square copper pads. Since each pad is electrically isolated, the builder makes all connections with either wire wrap or miniature point to point wiring techniques. Discrete components are soldered to the prototype board such as resistors, capacitors, and integrated circuits. The 0.1 in grid system accommodates integrated circuits in DIP packages and many other types of through-hole components. Dotted PCB board is not designed for prototyping surface mount devices. Before building a circuit on Dotted PCB board, the locations of the components and connections are typically planned in detail on paper or with software tools.

Fig.3.17: Dotted PCB Board

13. Probes or wires:

A wire is a single, usually cylindrical, flexible strand or rod of metal. Wires are used to bear mechanical loads or electricity and telecommunications signals. Wire is commonly formed

by drawing the metal through a hole in a die or draw plate. Wire gauges come in various standard sizes, as expressed in terms of a gauge number. The term *wire* is also used more loosely to refer to a bundle of such strands, as in "multi stranded wire", which is more correctly termed a wire rope in mechanics, or a cable in electricity. Wire comes in solid core, stranded, or braided forms. Although usually circular in cross-section, wire can be made in square, hexagonal, flattened rectangular or other cross-sections, either for decorative purposes, or for technical purposes such as high-efficiency voice coils in loudspeakers. Edge-wound[] coil springs, such as the Slinky toy, are made of special flattened wire.

Fig. 3.18: Ribbon wires

CHAPTER- IV

IMPLEMENTATION AND DEBUGGING

4.1 INTRODUCTION:

Mobile gesture control robo car using Arduino Nano is very simple technology that even student having normal understanding related python language can make this project. In this project it basically uses 4 main components or things for operating this experiment.

1. Android application.

2. Arduino Nano ATmega328

3. Bluetooth module chip.

4. L293D Motor driver IC.

Its working is very simple that we just need to install software or application from playstore and after installing you to pair the mobile Bluetooth with robo car Bluetooth chip HC-05. Once the pairing is done open the software and wait for connection; once connection gets established you just need to tit your mobile in the particular direction. The Accelerometer and gyroscope sensor are working in background which convert that motion in some specific value and transfer to the robo car through Bluetooth connection. Once the data is received at other end the data is start decoding and processor decide which command is need to forward to L293D Motor driver IC. When data is received by Motor driver IC its start processing what should need to be done by which motor. That's how our project works.

4.2 HARDWARE DEVELOPMENT:

In this section we developed the external hardware. The following process is design to develop.

4.2.1 PCB Designing:

A printed circuit board (PCB) mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate. Components — capacitors, resistors or active devices

— are generally soldered on the PCB. Advanced PCBs may contain components embedded in the substrate.

PCBs can be single sided (one copper layer), double sided (two copper layers) or multi-layer (outer and inner layers). Conductors on different layers are connected with bias. Multi-layer PCBs allow for much higher component density.

Printed circuit boards are used in all but the simplest electronic products. Alternatives to PCBs include wire wrap and point-to-point construction. PCBs require the additional design effort to lay out the circuit, but manufacturing and assembly can be automated. Manufacturing circuits with PCBs is cheaper and faster than with other wiring methods as components are mounted and wired with one single part. Furthermore, operator wiring errors are eliminated.

When the board has no embedded components it is more correctly called a printed wiring board (PWB) or etched wiring board. However, the term printed wiring board has fallen into disuse. A PCB populated with electronic components is called a printed circuit assembly (PCA), printed circuit board assembly or PCB assembly (PCBA). Chemical etching is usually done with ammonium per sulphate or ferric chloride. For PTH (plated-through holes), additional steps of electro less deposition are done after the holes are drilled, then copper is electroplated to build up the thickness, the boards are screened, and plated with tin/lead. The tin/lead becomes the resist leaving the bare copper to be etched away.

The simplest method, used for small-scale production and often by hobbyists, is immersion etching, in which the board is submerged in etching solution such as ferric chloride. Compared with methods used for mass production, the etching time is long. In bubble etching; air is passed through the etchant bath to agitate the solution and speed up etching. Splash etching uses a motor-driven paddle to splash boards with etchant; the process has become commercially obsolete since it is not as fast as spray etching. In spray etching, the etchant solution is distributed over the boards by nozzles, and recirculates by pumps. Adjustment of the nozzle pattern, flow rate, temperature, and etchant composition gives predictable control of etching rates and high production rate.

Holes through a PCB are typically drilled with small-diameter drill bits made of solid coated tungsten carbide. Coated tungsten carbide is recommended since many board materials are very abrasive and drilling must be high RPM and high feed to be cost effective. Drill bits must also remain sharp so as not to mar or tear the traces. Drilling with high-speed-steel is simply not feasible since the drill bits will dull quickly and thus tear the copper and ruin the boards. The drilling is performed by automated drilling machines with placement controlled by a drill tape or drill file.

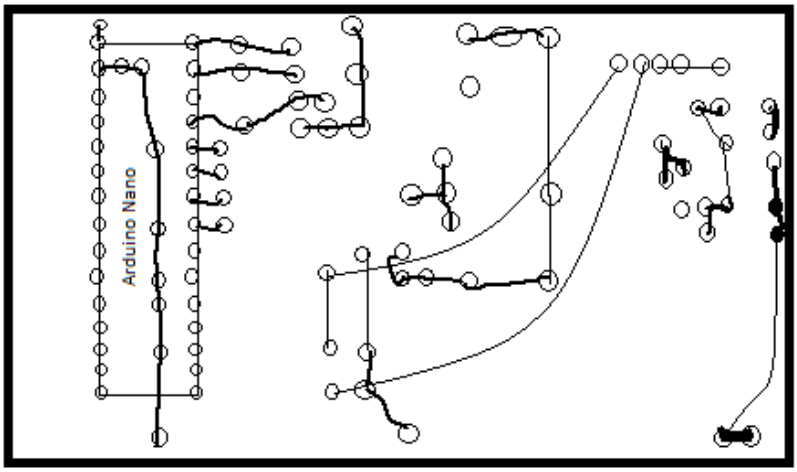

Fig. 4.1: PCB design

4.2.2 Soldering of Components:

Soldering is a process in which two or more items (usually metal) are joined together by melting and putting a filler metal (solder) into the joint, the filler metal having a lower melting point than the adjoining metal. Soldering differs from welding in that soldering does not involve melting the work pieces. In brazing, the filler metal melts at a higher temperature, but the work piece metal does not melt. In the past, nearly all solders contained lead, but environmental and health concerns have increasingly dictated use of lead-free alloys for electronics and plumbing purposes.

Soldering filler materials are available in many different alloys for differing applications. In electronics assembly, the eutectic alloy of 63% tin and 37% lead (or 60/40, which is almost identical in melting point) has been the alloy of choice. Other alloys are used for plumbing,

mechanical assembly, and other applications. Some examples of soft-solder are tin-lead for general purposes, tin-zinc for joining aluminium, lead-silver for strength at higher than room temperature, cadmium-silver for strength at high temperatures, zinc-aluminium for aluminium and corrosion resistance, and tin-silver and tin-bismuth for electronics.

A eutectic formulation has advantages when applied to soldering: the liquids and solidus temperatures are the same, so there is no plastic phase, and it has the lowest possible melting point. Having the lowest possible melting point minimizes heat stress on electronic components during soldering. And, having no plastic phase allows for quicker wetting as the solder heats up, and quicker setup as the solder cools. A non-eutectic formulation must remain still as the temperature drops through the liquids and solidus temperatures. Any movement during the 0plastic phase may result in cracks, resulting in an unreliable joint.

The purpose of flux is to facilitate the soldering process. One of the obstacles to a successful solder joint is an impurity at the site of the joint, for example, dirt, oil or oxidation. The impurities can be removed by mechanical cleaning or by chemical means, but the elevated temperatures required to melt the filler metal (the solder) encourages the work piece (and the solder) to re-oxidize. This effect is accelerated as the soldering temperatures increase and can completely prevent the solder from joining to the work piece. One of the earliest forms of flux was charcoal, which acts as a reducing agent and helps prevent oxidation during the soldering process. Some fluxes go beyond the simple prevention of oxidation and also provide some form of chemical cleaning (corrosion).

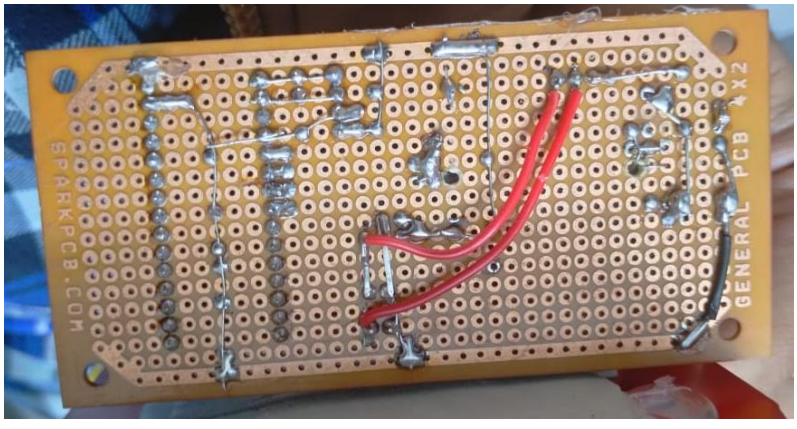

Fig. 4.2: Soldering of component

For many years, the most common type of flux used in electronics (soft soldering) was rosin-based, using the rosin from selected pine trees. It was ideal in that it was non-corrosive and non-conductive at normal temperatures but became mildly reactive (corrosive) at the elevated soldering temperatures. Plumbing and automotive applications, among others, typically use an acid-based (hydrochloric acid) flux which provides cleaning of the joint. These fluxes cannot be used in electronics because they are conductive and because they will eventually dissolve the small diameter wires. Many fluxes also act as a wetting agent in the soldering process, reducing the surface tension of the molten solder and causing it to flow and wet the work pieces more easily.

4.3 INTERFACING OF HARDWARE:

4.3.1 Hardware:

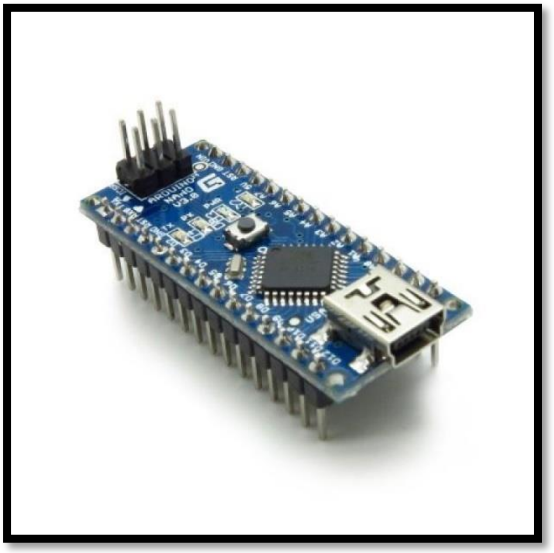

Fig: 4.3: Arduino hardware.

Arduino NANO Version 3 is the open source smallest Embedded Development board launched by Arduino based on Atmega328 SMD Package Microcontroller. It is a Surface mount Breadboard Friendly board integrated with Mini USB Port. DC Power Jack is not available on this Board, so power can be given through Mini USB Cable. It automatically sense and switch to the higher potential source of power, there is no need for the power select jumper.

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Nano's digital pins. The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

Technical specifications:

- Microcontroller: Microchip ATmega328P
- Operating Voltage: 5 Volts
- Input Voltage: 6 to 20 Volts
- Digital I/O Pins: 14 (plus 6 can PWM output pins)
- Analog Input Pins: 8
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader

4.3.2 Software:

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 makefiles or run programs on a command-line interface. Although building on command-line is possible if required with some third-party tools such as Ino. 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++, although users only need define two functions to make a runnable program: * setup() – a function run once at the start of a program that can initialize settings * loop() – a function called repeatedly until the board powers off

The integrated pin 13 LED

A typical first program for a microcontroller simply blinks an LED on and off. In the Arduino environment, the user might write a program like this

It is a feature of most Arduino boards that they have an LED and load resistor connected between pin 13 and ground, a convenient feature for many simple tests.[29] The above code would not be seen by a standard C++ compiler as a valid program, so when the user clicks the "Upload to I/O board" button in the IDE, a copy of the code is written to a temporary file with an extra include header at the top and a very simple main() function at the bottom, to make it a valid C++ program. See Cyclic executive

The Arduino IDE uses the GNU toolchain and AVR Libc to compile programs, and uses avrdude to upload programs to the board.As the Arduino platform uses Atmel microcontrollers Atmel's development environment, AVR Studio or the newer Atmel Studio, may also be used to develop

software for the Arduino. For educational purposes there is third party graphical development environment called Minibloq available under a different open source license.

4.3.3 SOFTWARE DEVELOPMENT AND IMPLEMENTATION:

In this project we will use the Arduino IDE for developing the programs in c-language. The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common

input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program argued to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Programming the Arduino Nano isn't extremely troublesome errand. When the circuit has been made on the breadboard, you'll have to transfer the program (known as a draw) to the Arduino. The draw is an arrangement of guidelines that tells the board what capacities it needs to perform. An Arduino load up can just hold and perform one outline at any given moment. The product used to make Arduino portrays is known as the IDE which represents Integrated Development Environment. For, this particular project we have developed a simple code. The code has been illustrated using screen shots taken from our laptop.

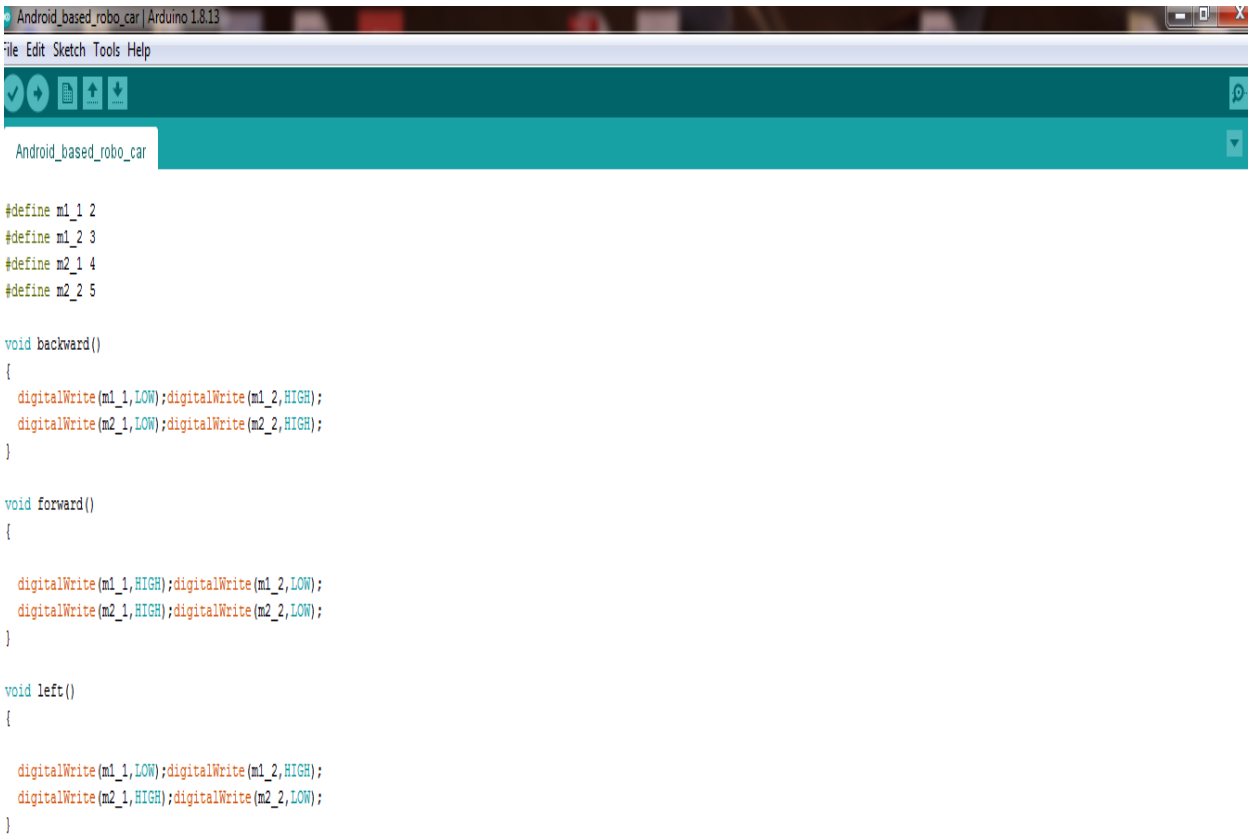

Fig. 4.4 Part 1st of Arduino programming

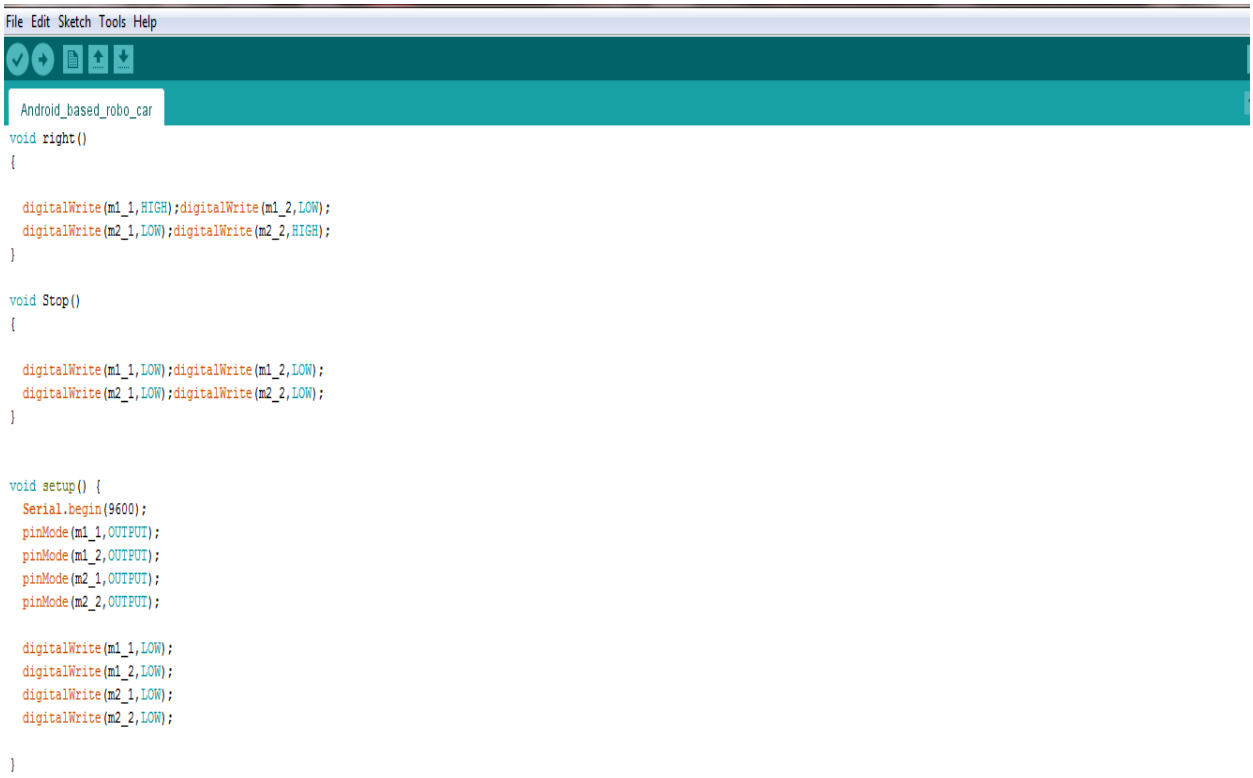

Fig.4.5 Part 2nd of Arduino programming

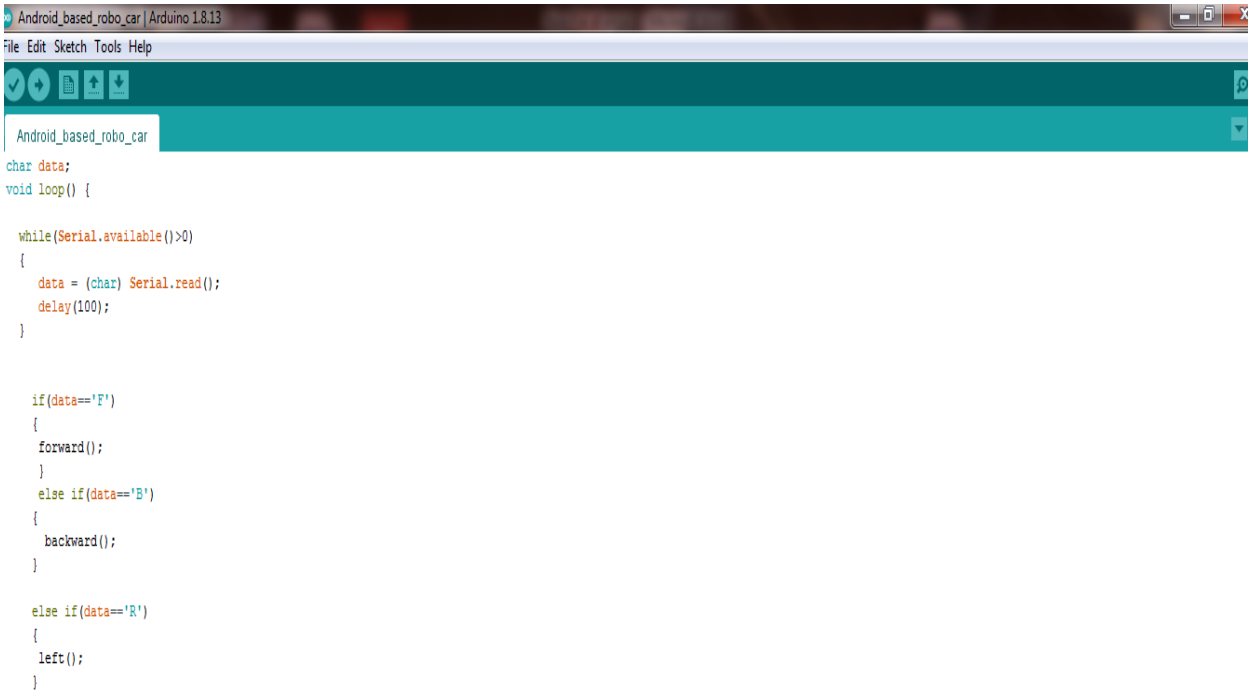

Fig. 4.6 Part 3rd of Arduino programming

```
 else if(data=='L')
{
  right();

}

 else if(data=='S')
{

  Stop();

}

delay(500);


}
```

Fig. 4.7 Part 4th of Arduino programming

The given code is basically used for moving Robo car in Different Directions.

The above written code can easily upload in Arduino Nano using Arduino IDE by following some simple steps.

1. Firstly we need to install Arduino IDE from the internet.

2. Then write the required code in the application and compile the program.

3. Now select the Arduino model and processor model in the tools option.

4. After this select the COM port and just click on the upload button. This is present near the compile button.

5. Once the uploading is done, disconnect it and use that in the circuit board.

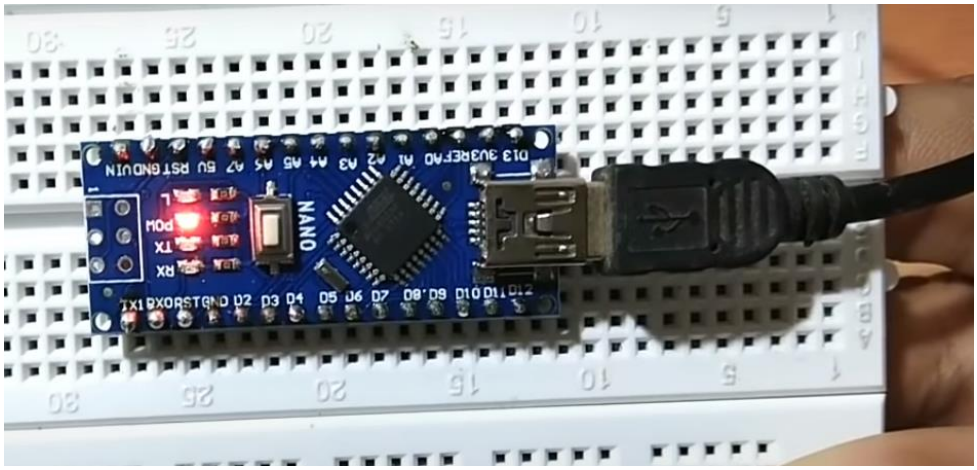

Fig.4.8: Image of uploading data in Arduino Nano

4.3.4 INTERFACING OF BLUETOOTH CHIP WITH ARDUINO NANO

In this project, we will learn about HC-05 Bluetooth Module, how to interface this Bluetooth Module with Arduino and how the HC-05 Bluetooth Module can be used for controlling the Arduino Board over Wireless Communication (Bluetooth).

Bluetooth Communication is a 2.4GHz frequency based RF Communication with a range of approximately 10 meters. It is one of the most popular and most frequently used low range communications for data transfer, audio systems, handsfree, computer peripherals etc. Coming to usage of Bluetooth Communication in DIY projects, HC-05 Bluetooth Module is the go to device.

Pins of HC-05 Bluetooth Module:

The HC-05 Module supports for UART, USB as well as SPI communication and depending on the application, necessary pins can be used. In my case, the board uses the UART communication.

Coming to the pins of the Bluetooth Module, generally, four pins are sufficient for successfully enabling a wireless communication link but the modules produced now-a-days come with six pins namely: VCC, GND, TX, RX, EN and STATE.

Image below shows the pins and other components on a typical HC-05 Bluetooth Module.

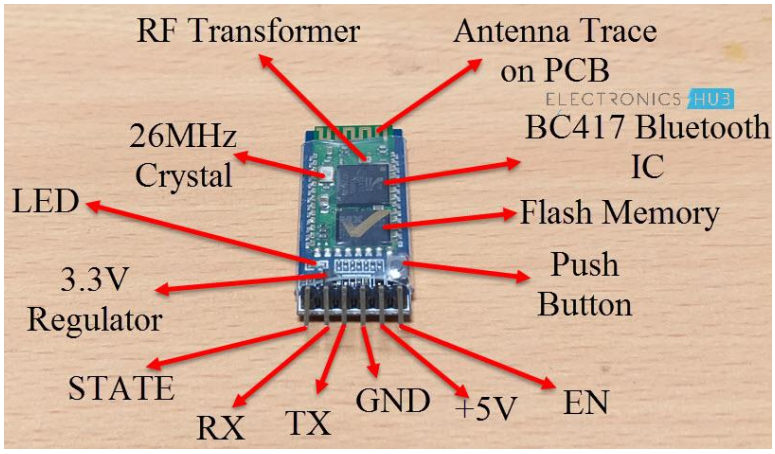

Fig.4.9: Bluetooth chip details

For interfacing with Arduino Nano you to create a connection between both HC-05 and Arduino Nano chip.

NOTE: The on-board LED is used to indicate the status of the connection. When the module is not paired, the LED blinks or flashes repeatedly. Once the module is paired, the LED blinks at a constant delay of 2 seconds.

PIN connections are given below.

Arduino Nano chip	Bluetooth chip (HC-05)
TX1	RXD
RX0	TXD
5V	VCC
GND	GND

Table no.4.1: Interfacing pins of both boards

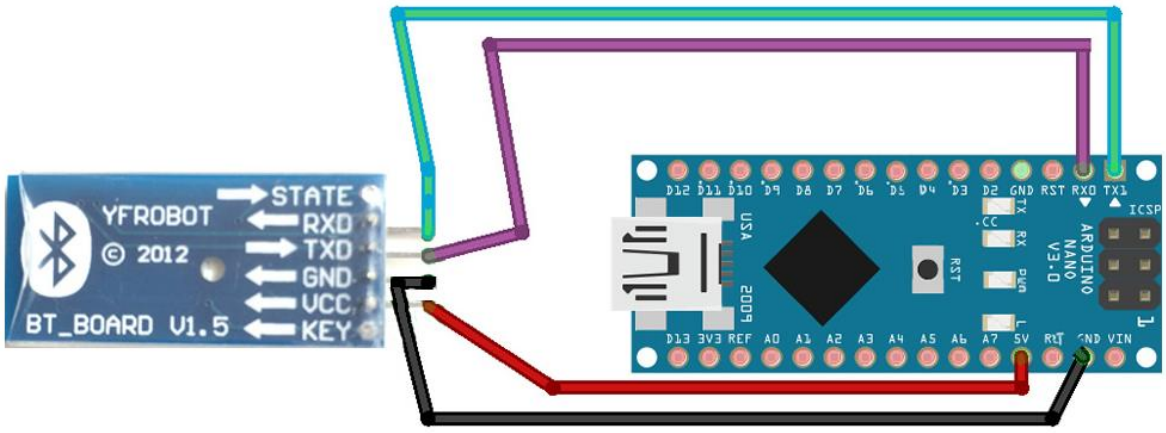

Fig.4.10: Interfacing pin diagram

Feasibility Study:

In this we did analysis on all project's relevant factors into account , therefore we took the following factors

1. **Economic Factor:**

   Due to this factor we had to change our project few times because at first we were controlling the robot with hand gesture but after some research we found out that the equipment required to make the hand controller is already present in smart phones therefore we used it and reduced the cost also we used the equipment that are very easy to found and cost very less.

   By removing that hand controller we also reduced components such as RF transmitter and receiver, Encoder and Decoder, additional Arduino and in the end we replaced the forward two Geared Motor with Ball wheel and it also increase maneuver ability.

2. **Technical Factor:**

   In this factor we take technical aspect of this project such how practical area of the project .So while this is very important and helpful project it certainly has its pros and cons to it

   PROS:

   - It is fairly easy to understand and control.
   - It has lot of applications
   - It is also cost effective

   CONS:

   - Due the connection being Bluetooth it has limitation in the range factor
   - Chasse of the robot is flat and without shockers so it would be quiet difficult to cross obstacles.

3. **Handling Factor:**

   In this we have taken those factors that can see to its handling by any person. So this projects was build all around the fact that it could be used by anyone what u need is just a smart phone and the connection is also fairly simple and also it works on the gesture such as tilting so it could handled with ease.

4.5 FINAL PICTURE OF PROJECT:

Fig.4.11: Picture of Project

CHAPTER-V

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion:

In this robot has been developed which is according to human Mobile gesture. The robot is showing proper responses whenever we tilt our mobile. Different tilt gestures to make the robot in specific directions are left, right, forward, backward. In this project our robot cannot senses any obstacle comes in its path, avoids it and resumes it's running. In the making of this we can upgrade or update it in many for example you can add camera and use it for surveillance Wi-Fi can be used for communication instead of Bluetooth to access it from a greater distance. Edge sensors can be incorporated to it to prevent the robot from falling from any surface. Some camera can be installed which can record and send data to the nearby computer or cell-phone. It can be implemented on a watch, or in any home appliances like Room heater. Modern ARDUINO chips support Intranet as well as Internet connections which can be utilized to a greater extent. This robotic car can be enhanced to work in the military surveillance where it can be sent to some enemy camps and track it's activities via Internet. With a mind full of creation, the possibilities are endless. In this paper, the design and implementation of Gesture Controlled Robot is presented and developed using Arduino microcontroller and Android Smartphone. An algorithm has been provided and its working is detailed thoroughly. Since the updating possibilities are endless, updating the system has been kept as a future scope. The built device is cheap, and is easy to carry from one place to another. The addition of the some additional sensors or camera will make it more productive. The limitation of the hardware being associated with a system has been reduced to a great extent. As an end thought, the system will allow the user to control it in a way that reduces the gap between the physical world and the digital world with an output more intuitive.

5.2 Future Scope:

1. The on-board batteries occupy a lot of space and are also quite heavy. We can either use some alternate power source for the batteries or replace the current DC Motors with ones which require less power.

2. As we are using Bluetooth chip module for wireless transmission, the range is quite limited; nearly 10-15m. This problem can be solved by utilizing a GSM module for wireless transmission. The GSM infrastructure is installed almost all over the world. GSM will not only provide wireless connectivity but also quite a large range.

3. An on-board camera can be installed for monitoring the robot from faraway places.

4. We can install ultrasonic sonic in our robot so that it can stop itself from getting accident.

5.3 Applications and Advantages:

1. Wireless controlled robots are very useful in many applications like remote surveillance, military etc.
2. Mobile gesture controlled robot can be used by doing some physically challenges in wheelchairs.
3. Mobile gesture controlled robot can be used in Hospitals where staff always take risk in daily life during this pandemic.
4. Gesture can be used to control the interactions for entertainment purposes such as gaming to make the game player's experience more interactive and immersive.
5. Mobile gesture controlled robot can be used as toys for entertaining purpose.

Reference

1. https://www.researchgate.net/profile/Asoke_Nath/publication/304624684_Gesture_Controlled_Robot_using_Arduino_and_Android/links/57750adb08ae1b18a7dfa026/Gesture-Controlled-Robot-using-Arduino-and-Android.pdf?origin=publication_detail
2. https://www.geeksforgeeks.org/all-about-hc-05-bluetooth-module-connection-with-android/
3. https://microcontrollerslab.com/l293d-motor-driver-ic-introduction-pinouts-example/
4. https://www.electronicshub.org/hc-05-bluetooth-module/
5. http://www.uugear.com/portfolio/bluetooth-communication-between-raspberry-pi-and-arduino/
6. https://www.google.com/search?q=interfacing+of+bluetooth+chip+with+arduino+nano&rlz=1C1CHBD_enIN916IN916&sxsrf=ALeKk02MoGsEmfVMAkr0k0EGCuNW3wDj9w:1609400177143&source=lnms&tbm=isch&sa=X&ved=2ahUKEwj2o_Du2vftAhVZ8XMBHSbACyYQ_AUoAXoECBQQAw&biw=1517&bih=694#imgrc=1s6uMecvGfCHIM&imgdii=cVhDAsIwuMNJqM
7. https://www.islproducts.com/design-notes/dc-motor-dc-gear-motor-basics/#:~:text=A%20gear%20motor%20is%20an,while%20increasing%20the%20torque%20output.&text=Most%20of%20our%20DC%20motors,highly%20efficient%20gear%20motor%20solution.
8. https://www.gsmarena.com/glossary.php3?term=sensors#:~:text=Accelerometers%20in%20mobile%20phones%20are,by%20tracking%20rotation%20or%20twist.
9. https://www.youtube.com/watch?v=kaTDF-oiUVY
10. https://microcontrollerslab.com/l293d-motor-driver-ic-introduction-pinouts-example/
11. https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html
12. https://www.google.com/search?q=ribbon+wire&tbm=isch&ved=2ahUKEwjx2YLPn_PtAhX2HrcAHbAMDaQQ2-cCegQIABAA&oq=ribon+w&gs_lcp=C

13. **https://www.google.com/search?q=ball+wheel&rlz=1C1CHBD_enIN916IN916&sxsrf=ALeKk01RzjGgvkq-k2NoDlZKVhbBD20VDA:1609246810692&source=lnms&tbm=isch&sa=X&ved=2ahUKEwi06YbEn_PtAhUXgtgFHTMrD-YQ_AUoAXoECBMQAw&biw=1517&bih=694**
14. **https://www.google.com/electronics.stackexchange.com/questions/18447/what-is-back-emfcounterelectromotive-force**

APPENDICS

```
#define m1_1 2
#define m1_2 3
#define m2_1 4
#define m2_2 5

void backward()
{
  digitalWrite(m1_1,LOW);digitalWrite(m1_2,HIGH);
  digitalWrite(m2_1,LOW);digitalWrite(m2_2,HIGH);
}
void forward()
{
  digitalWrite(m1_1,HIGH);digitalWrite(m1_2,LOW);
  digitalWrite(m2_1,HIGH);digitalWrite(m2_2,LOW);
}
void left()
{
  digitalWrite(m1_1,LOW);digitalWrite(m1_2,HIGH);
  digitalWrite(m2_1,HIGH);digitalWrite(m2_2,LOW);
}
void right()
{
```

```
  digitalWrite(m1_1,HIGH);digitalWrite(m1_2,LOW);
  digitalWrite(m2_1,LOW);digitalWrite(m2_2,HIGH);
}
void Stop()
{
  digitalWrite(m1_1,LOW);digitalWrite(m1_2,LOW);
  digitalWrite(m2_1,LOW);digitalWrite(m2_2,LOW);
}
void setup()
 {
  Serial.begin(9600);
  pinMode(m1_1,OUTPUT);
  pinMode(m1_2,OUTPUT);
  pinMode(m2_1,OUTPUT);
  pinMode(m2_2,OUTPUT);

  digitalWrite(m1_1,LOW);
  digitalWrite(m1_2,LOW);
  digitalWrite(m2_1,LOW);
  digitalWrite(m2_2,LOW);
}
char data;
void loop()
```

```
{
 while(Serial.available()>0)
 {
    data = (char) Serial.read();
    delay(100);
 }
   if(data=='F')
   {
    forward();
    }
    else if(data=='B')
   {
     backward();
   }
   else if(data=='R')
   {
    left();
   }
    else if(data=='L')
   {
     right();
   }
```

```
    else if(data=='S')
    {
     Stop();
    }

    delay(500);

}
```