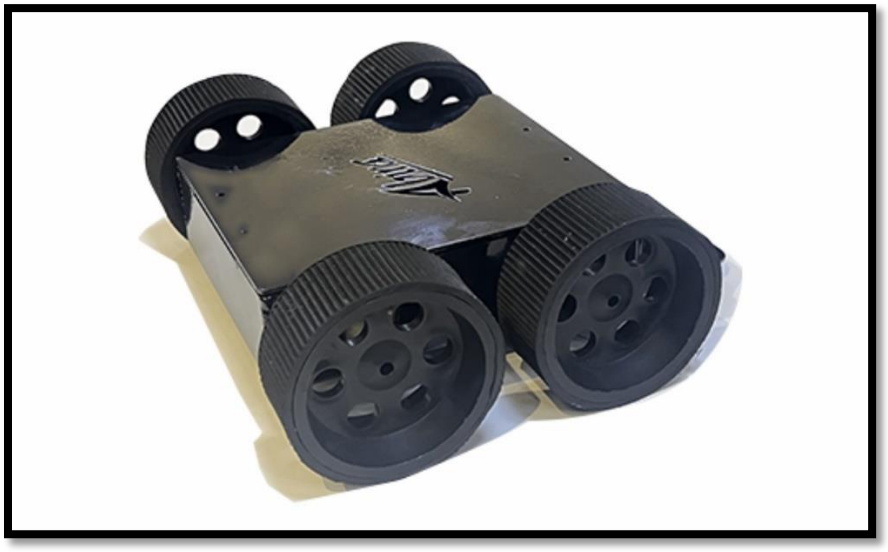
Manual for Remote Controlled Mobile robot Vehicle



A **remote-control vehicle**, is defined as any vehicle that is tele-operated by a means that does not restrict its motion with an origin external to the device. This is often a radio-control device, a cable between the controller and the vehicle, or an infrared controller or controlled using Bluetooth based communication.

As the name suggests, remote-control robot cars are smart vehicles. They are organized via remote control devices from a long distance. The remote control devices may include transmitters or apps on mobile phones. Remote control robot cars are designed for various tasks besides transportation and adventures. They can be used for educational purposes or real-world applications like investigation or assessment.

Working of Remote Control Robot Cars

Remote Control robot cars work through two main components. One is a transmitter, and the other is the receiver. The transmitter is controlled by the user, who sends different commands. These commands are then received through the receiver on the other end.

Remote Control robot cars have wireless communication. So, the transmitter and receiver communicate with each other through wireless technologies. These technologies include radio frequency (RF) or any Bluetooth signals. These signals help in carrying the commands of users to the car.

Regarding the movement of wheels, electric motors are inside the Remote Control robot cars. These motors help in controlling wheel movement. So, the main component is the receiver because it will process the commands and control these motors accordingly. Robot cars work on batteries, especially rechargeable ones. So the motors and the receivers get energy through this power source

Features of Remote Control Robot Cars

Features of Remote control robot cars are as follows:

- Ability to control the car remotely. It allows users to exercise it as per their plea.
- A speed control option is available.
- The option of direction control is also present in RC robot cars.
- Larger battery capacities.

Out of all the different ways of controlling the mobile robot we have used **Bluetooth** as the communication protocol for controlling our RC mobile robot vehicle.

Here we will be working with RC Car which will be controlled using a Smart Phone so the phone becomes our transmitter and the mobile vehicle having the controller itself becomes the receiver unit.

Let us divide the entire work into few parts to make things easy:

- **Parts and Tools Required**

1.Robot Chassis kit

2. Arduino Uno with Usb Cable

3. LM298 H bridge Module

4. Bluetooth Module HC-05

5. 12v Li-po Battery

6. Male-Female Jumper Wires

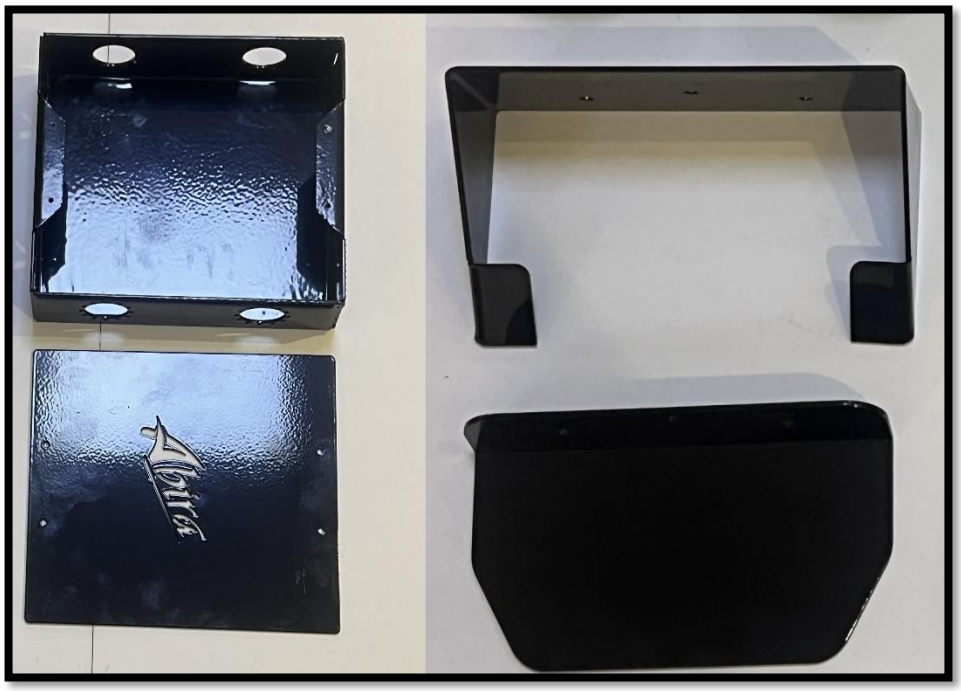
7. Male-Male Jumper Wires

8. Alley keys, ScrewDriver

8. Duct Tape or any other tape

9. Smartphone

- **Structure & Chassis**



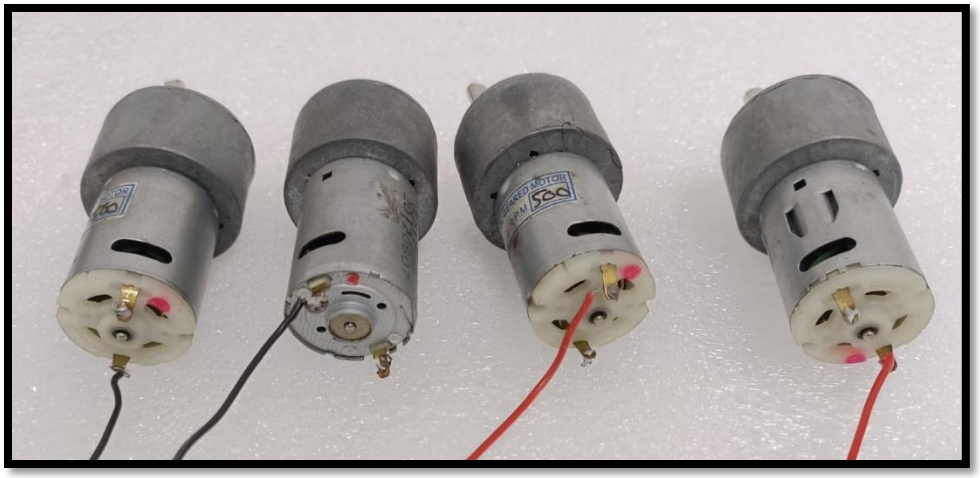
- **Motors/Actuators**

Actuator

A device that converts energy into linear motion, rotation, or bending. Actuators can use a variety of energy sources, including electricity, air, oil, steam, and heat.

Motor

A type of actuator that converts electrical energy into rotary motion. Motors can drive components directly or be part of a larger actuator system. For example, in a toy car, the motor spins a wheel to move the car



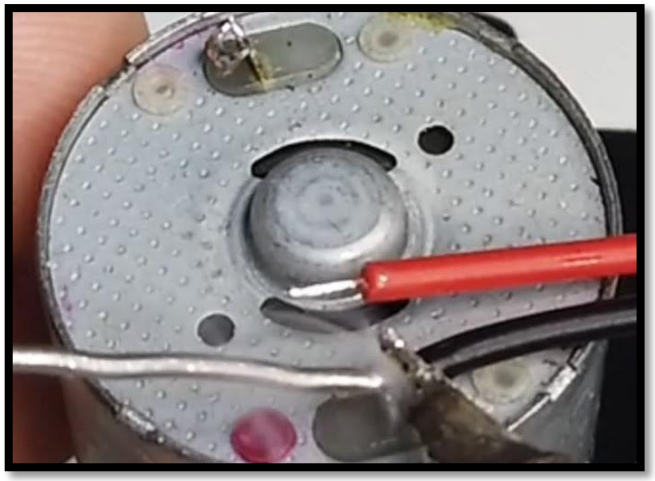
- **Prepare the motor terminal**

  Cut 4 pieces of red and black wires with length approximately 5 to 6 inch.

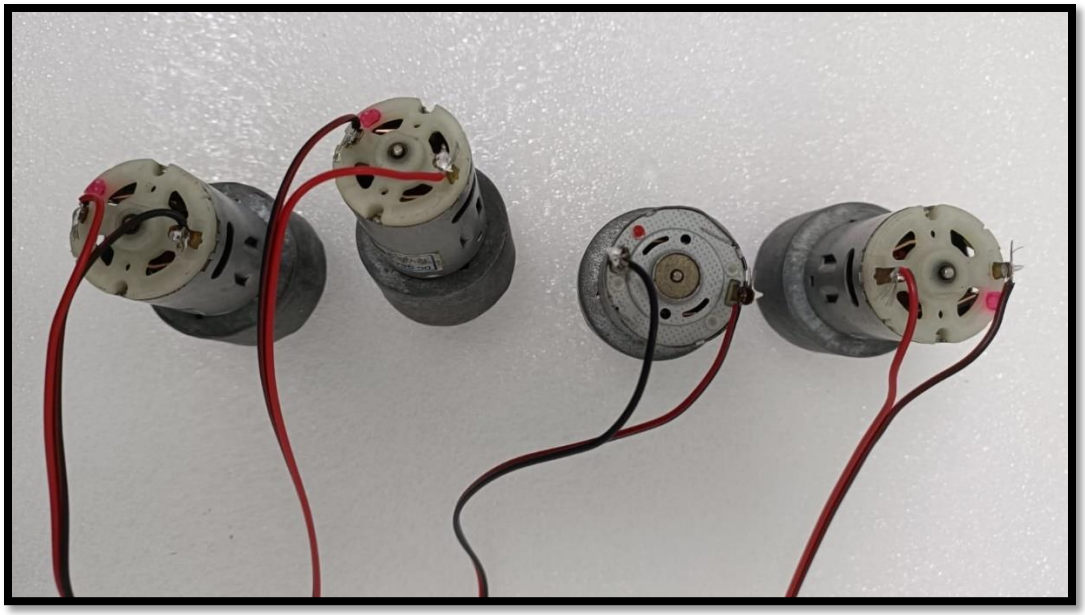
  0.5 sqmm wires can be used .

  Strip out the insulation from the wires at each end Solder the wires to the motor terminal

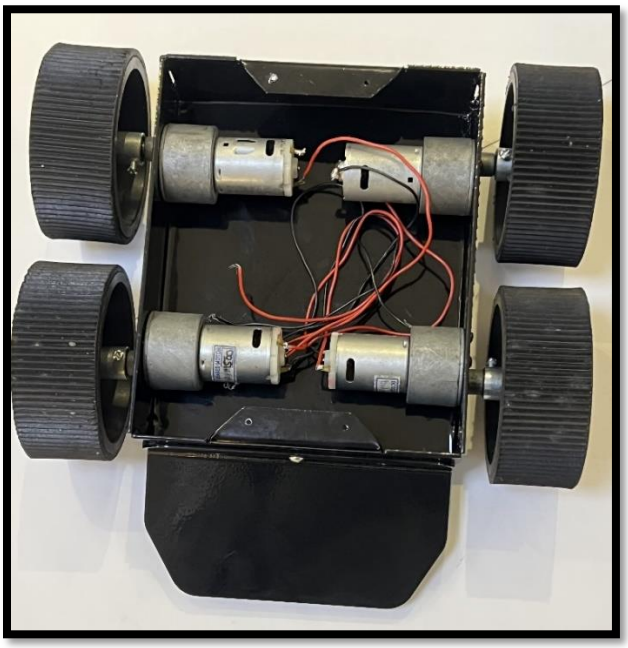
  You can check the motor polarity by connecting it to the battery pack. If it rotates in forward direction (red wire with positive and black wire with negative terminal of the battery) then the connection is correct .

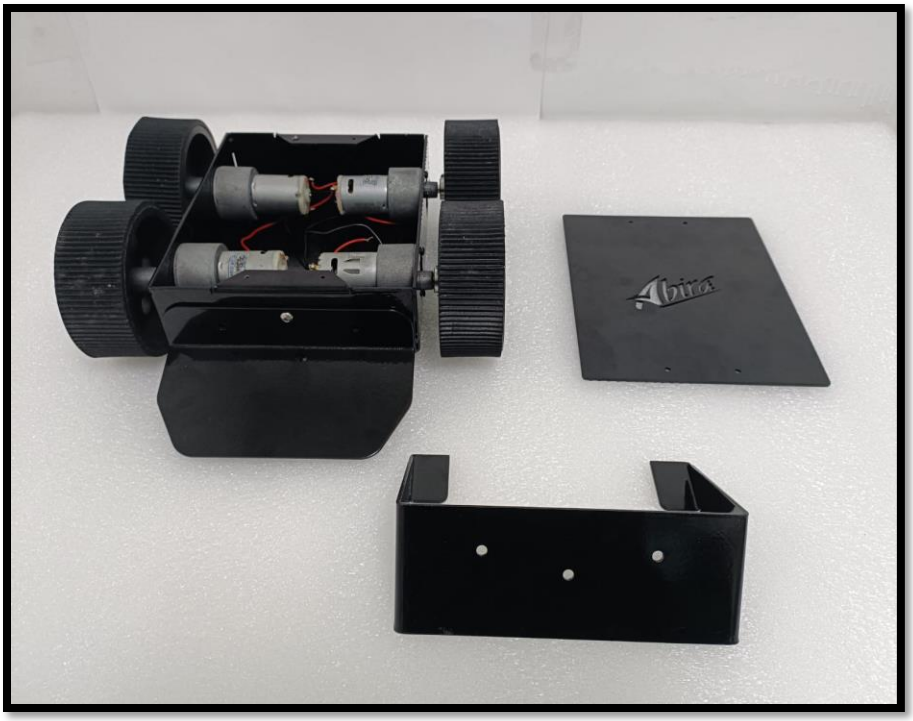


After done with all the wiring for the motors it should look like this

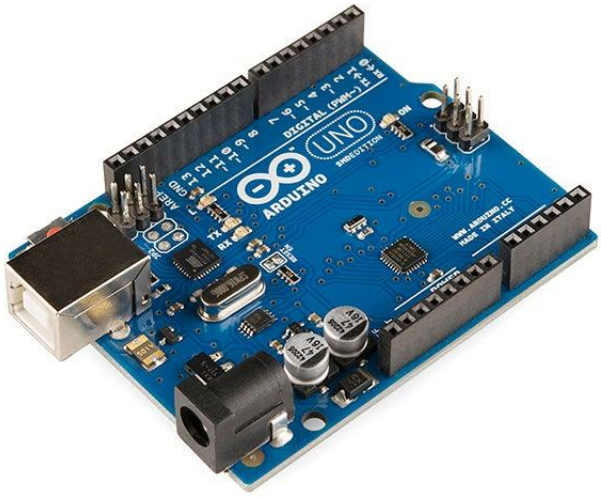


- **Mount the motors and install the chassis**





- **Controller**



The **Arduino UNO** is an open-source microcontroller board based on the Microchip ATmega328P and developed by Arduino.cc

The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.

The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable.

It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo.

The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0.

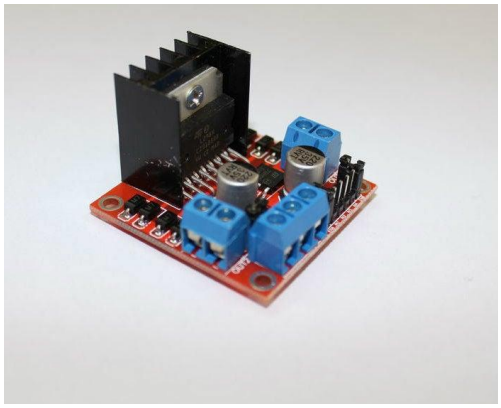
The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform.

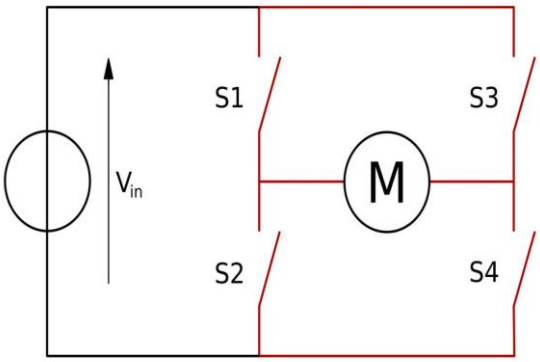
The ATmega328 on the Arduino Uno comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.[3] It communicates using the original STK500 protocol.

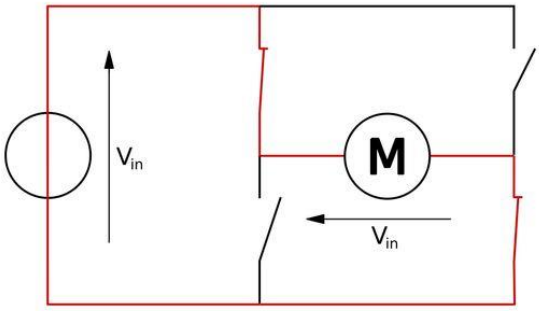
The Uno also differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

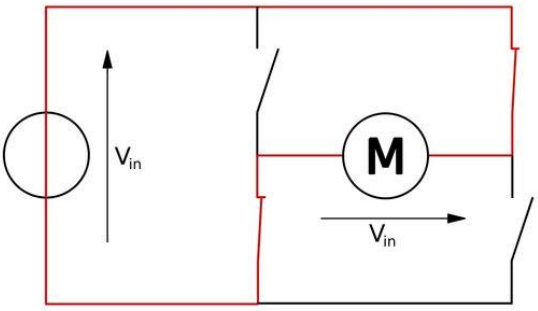
The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

- **H Bridge (L298 Module)**









What is H Bridge?

The term H Bridge is derived from the typical graphical representation of such a circuit. It is a circuit which can drive a DC motor in forward and reverse direction.

Working: See the above picture for understanding the working of the H bridge.

It is consists of 4 electronics switches S1,S2,S3 and S4 (Transistors / MOSFETs/ IGBTS). When the switches S1 and S4 are closed (and S2 and S3 are open) a positive voltage will be applied across the motor.So it rotates in the forward direction. Similarly when S2 and S3 are closed and S1 and S4 are opened a reverse voltage is applied across the motor, so rotates in revers direction.

Note : The switches in the same arm (either S1,S2 or S3,S4) are never closed at a same time, it will make a dead short circuit. H bridges are available as integrated circuits, or you can built your own by using 4transistors or MOSFETs. In our case we are using LM298 H-bridge IC that can allows to control the speed and direction of the motors.

Pin Description :

Out 1: DC motor 1 "+" or stepper motor A+

Out 2: DC motor 1 "-" or stepper motor A-

Out 3: DC motor 2 "+" or stepper motor B+

Out 4: Motor B lead out

12v Pin :12V input but you can use 7 to 35V

GND: Ground

5v Pin: 5V output if 12V jumper in place, ideal for powering your Arduino (etc)

EnA: Enables PWM signal for Motor A (Please see the "Arduino Sketch Considerations" section)

IN1: Enable Motor A

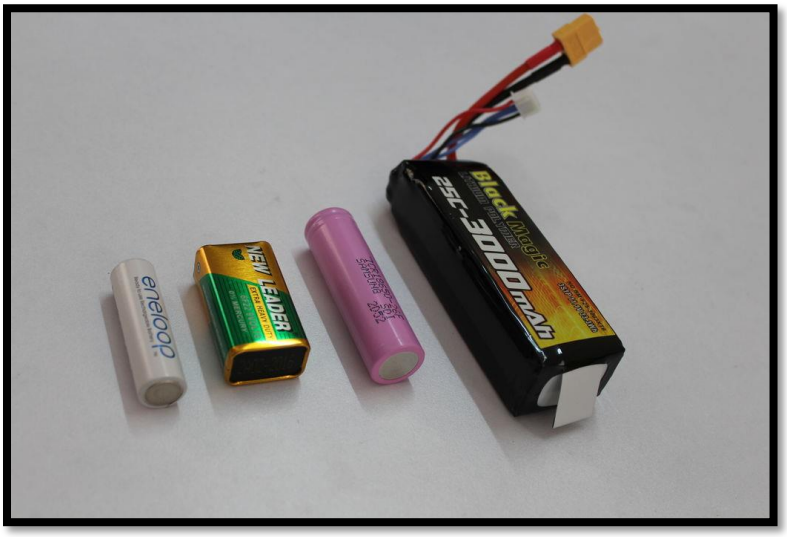
IN2: Enable MotorA

IN3: Enable MotorB

IN4: Enable MotorB

EnB: Enables PWM signal for Motor B

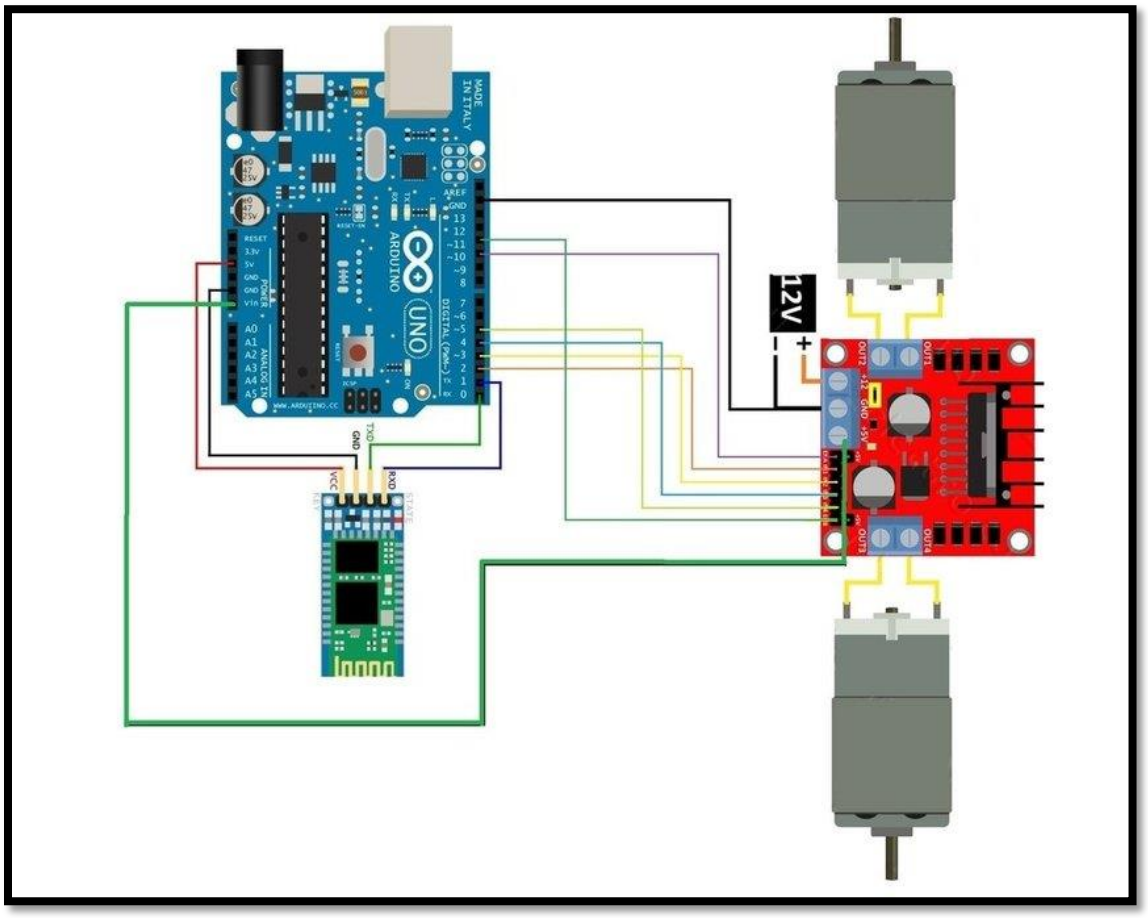
- **Power Source**



Those Battery can be used:

1. AA Alkaline Battery (Non Rechargeable)
2. AA NiMh or NiCd Battery (Rechargeable)
3. Li Ion Battery
4. LiPo Battery

- **Electrical Wiring**



For wiring we need some jumper wires.

Connect the red wires of two motors (on each side) together and black wires together.

So finally you have two terminals in each side. MOTORA is in charge of two right side motors, correspondingly two left side motors are connected to MOTORB Follow the instruction below to connect everything.

Motors Connection:

Out1 -> Left Side Motor Red Wire (+)

Out2 -> Left Side Motor Black Wire (-)

Out3 -> Right Side Motor Red Wire (+)

Out4 -> Right Side Motor Black Wire (-)

L298 - > Arduino

IN1 -> D9

IN2-> D10

IN2 ->D7

IN2-> D8

ENA ->D5

ENB-> D6

Bluetooth Module -> Arduino

Rx-> Tx

Tx ->Rx

GND -> GND

Vcc -> 3.3V

Power :

12V - > Connect Battery Red Wire

GND -> Connect Battery Black wire and Arduino GND pin

5V -> Connect to Arduino 5V pin

- **Control Logic**

Input Pin	Logic Applied	Output Pin	Output Voltage	Motor Action
IN1	0	Out1	0	Stop
IN2	0	Out2	0	
IN1	0	Out1	0	Forward Spin
IN2	1	Out2	Vs	
IN1	1	Out1	Vs	Reverse Spin
IN2	0	Out2	0	
IN1	1	Out1	Vs	Stop
IN2	1	Out2	Vs	
IN3	0	Out3	0	Stop
IN4	0	Out4	0	
IN3	0	Out3	0	Forward Spin
IN4	1	Out4	Vs	
IN3	1	Out3	Vs	Reverse Spin
IN4	0	Out4	V0	
IN3	1	Out3	Vs	Stop
IN4	1	Out4	Vs	

- **Software**

User Interface for the phone application

The software part is very simple, it does not need any library. If you understand the logic table in the earlier steps then you can write you own code. To control the Robot Car we are using our smartphone. The smartphone is connected to the controller via a Bluetooth module (HC -06 /05) Download the App After installing the app, you have to pair it with the Bluetooth module.

Download link:

https://play.google.com/store/apps/details?id=com.SanitTech.RCBluetoothControllerHC05&hl=en_IN

- **Arduino Code**

Attach the program:

Here

```
#define enA 5

#define in1 9

#define in2 10

#define in3 7

#define in4 8

#define enB 6


char Incoming_value = 0;

void setup(){


pinMode(enA, OUTPUT);

pinMode(in1, OUTPUT);
```

```
pinMode(in2, OUTPUT);

pinMode(in3, OUTPUT);

pinMode(in4, OUTPUT);

pinMode(enB, OUTPUT);

//digitalWrite(enA, HIGH);

//digitalWrite(enB, HIGH);

analogWrite(enA,100);// adjust the value to control the rpm
analogWrite(enB,100);

delay(1000);

stop();

}

void  loop() {

 // put your main code here, to run repeatedly:
 if (Serial.available()  > 0)
   {
    Incoming_value = Serial.read();
    //Serial.print(Incoming_value);
    //Serial.print("/n");

    if (Incoming_value == 'F')
     forward();

    else if(Incoming_value == 'B')
     backward();

    else if(Incoming_value == 'L')
     turnLeft();

    else if(Incoming_value == 'R')
     turnRight();

    else if(Incoming_value == 'S')
     stop();

    else stop();

   }
```

```
}

void forward(){
digitalWrite(in1, HIGH); //Right Motor forword Pin

digitalWrite(in2, LOW);  //Right Motor backword Pin

digitalWrite(in3, HIGH); //Left Motor backword Pin

digitalWrite(in4, LOW);  //Left Motor forword Pin
}

void backward(){
  digitalWrite(in1, LOW);  //Right Motor forword Pin

digitalWrite(in2, HIGH); //Right Motor backword Pin

digitalWrite(in3, LOW);  //Left Motor backword Pin

 digitalWrite(in4, HIGH); //Left Motor forword Pin
}

void turnRight(){ //turnRight

digitalWrite(in1, HIGH);  //Right Motor forword Pin

digitalWrite(in2, HIGH); //Right Motor backword Pin

digitalWrite(in3, HIGH);  //Left Motor backword Pin

 digitalWrite(in4, LOW); //Left Motor forword Pin

}

void turnLeft()
{ //turnLeft

digitalWrite(in1, HIGH); //Right Motor forword Pin

digitalWrite(in2, LOW);  //Right Motor backword Pin

digitalWrite(in3, HIGH); //Left Motor backword Pin

digitalWrite(in4, HIGH);  //Left Motor forword Pin

}

void stop()
```

```
{
  digitalWrite(in1, HIGH);  //Right Motor forword Pin

digitalWrite(in2, HIGH); //Right Motor backword Pin

digitalWrite(in3, HIGH);  //Left Motor backword Pin

 digitalWrite(in4, HIGH); //Left Motor forword Pin
}
```

- **Final testing**

  Steps

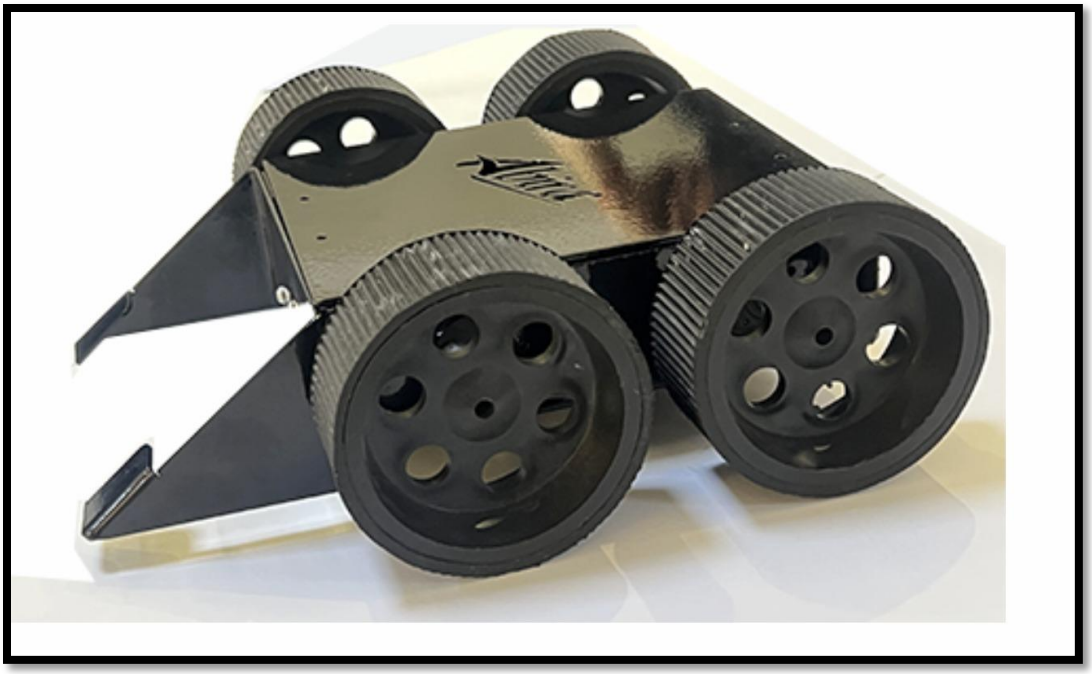
  - Upload the Code to the controller and power the entire unit
  - Check the lights on the HC-05 unit
  - Turn on the Bluetooth in the phone and make it visible to the nearby devices
  - Pair it with HC-05 MAC Address
  - Open the app in the phone and connect to the module
  - Test the bot with the corresponding commands
  - If the wheels the not rotating in same direction check the wirings of the motors.

- **Extension of the basic robot vehicle**

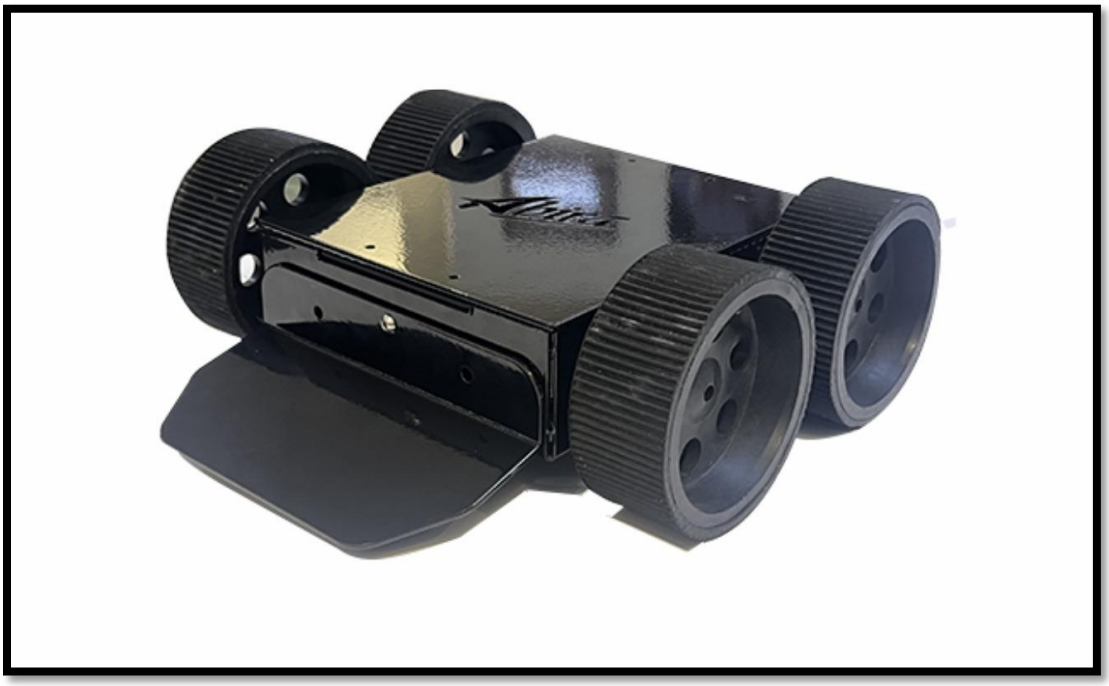
  Once we have assembled the unit and tested the working of our mobile robot vehicle

  We are going to use the same built in making four different type of robots namely

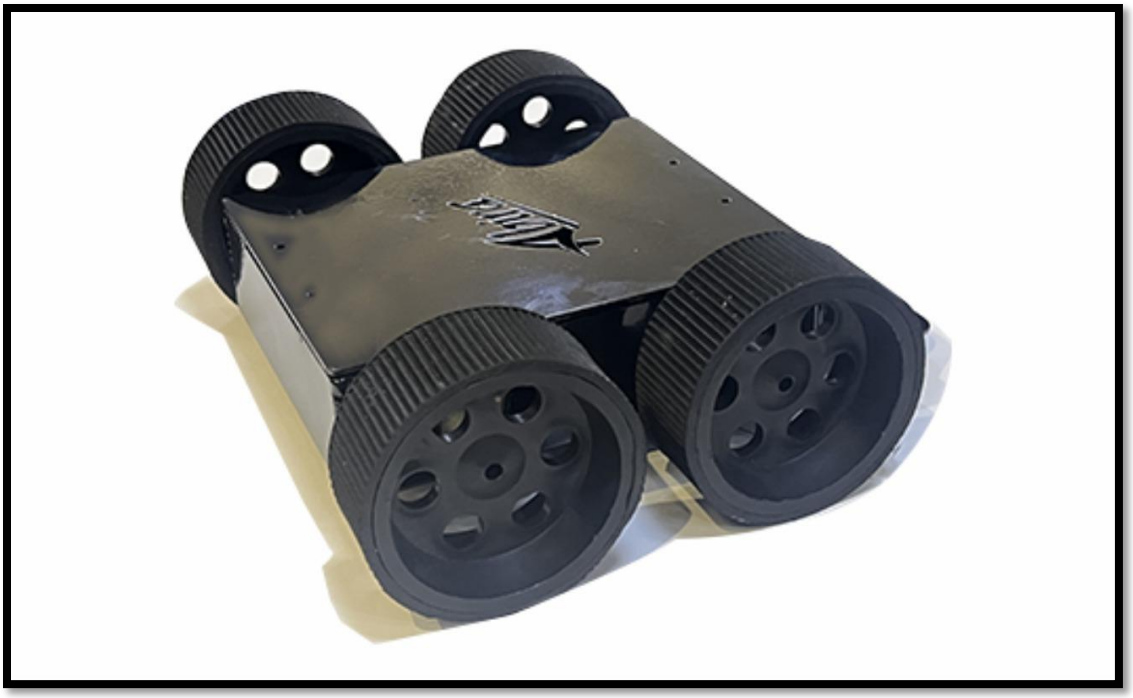
  - Sumo Bot
  - Hockey Bot
  - Racing Bot
  - Soccer Bot



Soccer and Hockey Bot share the same design



Sumo Bot Design



Racing Bot Design

*[**Attach individual bot pics here 4 respective designs**]