ACHARYYA PRAFULLA CHANDRA ROY POLYTECHNIC

JADAVPUR,KOL-32

PROJECT

VOICE CONTROL VEHICLE

PROJECT MEMBER:

1) RAMPRASAD MANDAL(81608)
2) MOJAFFAR ANSARI(81606)
3) SK NOORALAM RAHAMAN(81609)
4) AZAHARUDDIN SEIKH(81607)

CONTENTS

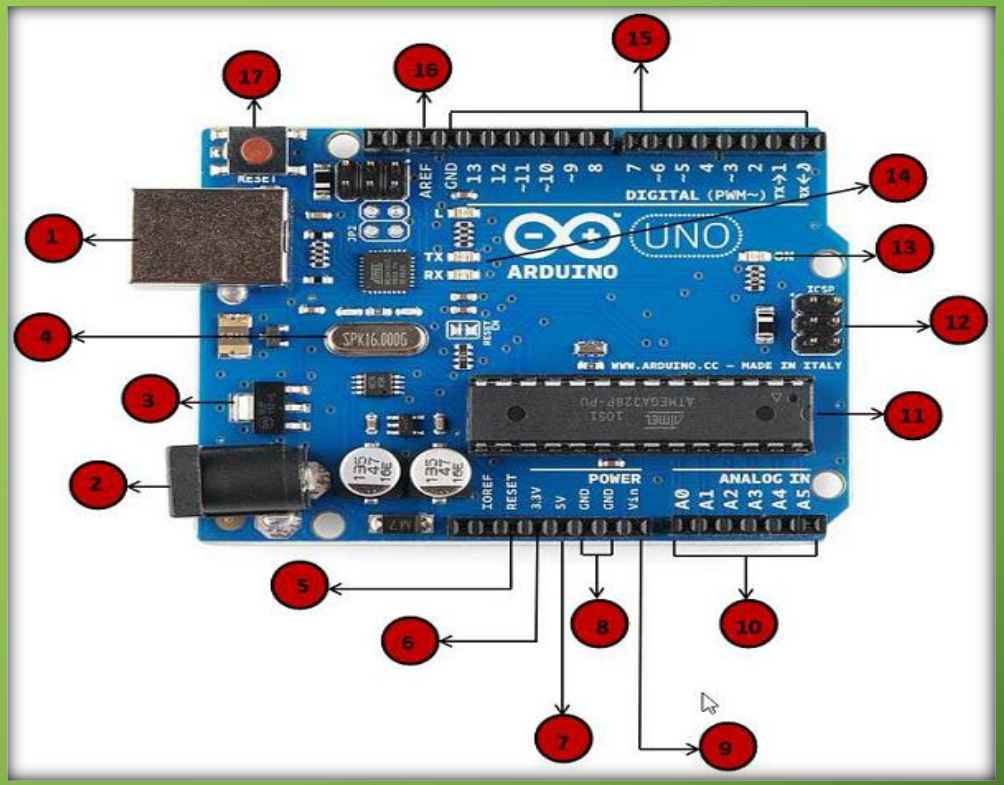
INTRODUCTION

We are making a Robotic Car that we can operate using our Android mobile phones by connecting Bluetooth with Bluetooth module of the car. It will work through our voice commands. We will use the Application “BT-Voice Controller for Arduino” and it’s available on Play Store. When the Bluetooth device is connected with mobile through Bluetooth, we will open the Application “BT-Voice Controller for Arduino” and we will say the command (For example- Forward, Backward, Left, Right Etc.). And by these commands the vehicle will move according to the commands that is, if we want to move the car Forward, we have to command “forward” and if we want to move backward, we have to command “backward” and so on. When we say any command, the AMR voice controller catch the command and send it to the Bluetooth module, then Bluetooth module passes the command into Arduino board. The microcontroller of the Arduino board makes decision and send the decision into Motor driver L298N or Motor shield via 3,4,5,6 no pin respectively. The Arduino Board controls the 4 pins i.e. which pin is High and which is Low. Then the L298N Motor driver operate the four motors.

REQUIREMENT ANASYSIS

- For this project we need some components. And these components are-
  - 1) Arduino Uno board
  - 2) L298N Motor driver
  - 3) HC-05 Bluetooth module/Driver
  - 4) 4 pics. 12-volt D.C Motor with wheel (200 rpm.)
  - 5) Jumper Cable (M to M, M to F, F to F)
  - 6) One 12-volt battery
  - 7) Vehicle Chassis
- And we will use the following software for this project……
  - Arduino 1.8.6 Hourly Build
  - Proteus 8 professional
  - Mobile Application “BT-Voice Control for Arduino”

1) ARDUINO UNO BOARD



1) **POWER USB:** Arduino board can be powered by using the USB from our Computer. All we need to do is connect the USB cable to the USB connection.

2) **POWER (Barrel Jack):** Arduino board can be powered directly from the A.C main power supply by connecting It to Barrel Jack.

3) **VOLTAGE REGULATOR:** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

4) **CRYSTAL OSCILLATOR:** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz

5) ARDUINO RESET: You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET.

6,7,8,9) PINS (3.3, 5, GND, VIN):

- 3.3V (6) – Supply 3.3 output volt
- 5V (7) – Supply 5 output volt
- Most of the components used with Arduino board works fine with 3.3 volt and 5 volts.
- GND (8) (Ground) – There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- Vin (9) – This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

10. **ANALOG PINS:** The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

11) **Main microcontroller:** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

12) ICSP pin: Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

13) **Power LED indicator:** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

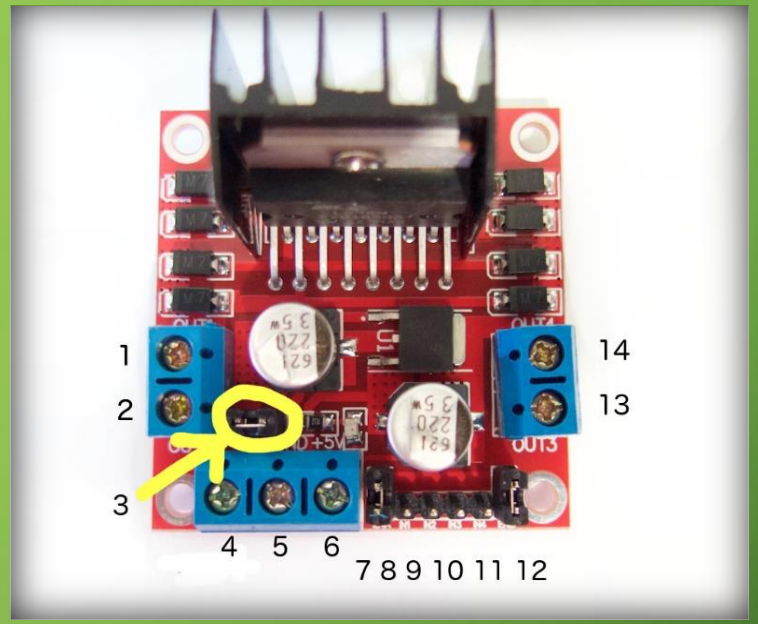
14) **TX and RX LEDs:** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

15) Digital I/O: The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.

16) AREF: AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins

17) RESET: In this bottom are used in to reset the microcontroller program. If we upload the new any program at first reset the first program then upload the new program.

2) L298N MOTOR DRIVER:



This motor controller from Triaxles Australia is based on the L298N heavy-duty dual H-bridge controller, which can be used to drive two DC motors at up to 2A each, with a voltage between 5 and 35V DC - or one stepper motor with ease. The controller has fast short-circuited protection diodes, and a nice heatsink to keep the L298N happy.

There is also an onboard 5V regulator - so if you're using between 7 and 12V DC to driver the motors, the module can also supply your Arduino (etc.) with 5V DC. If you are using more than 12V DC, please remove the 12V jumper (see below). With a 12-month warranty we think this is the best value L298N dual motor controller driver module in Australia.

1) DC motor 1 "+" or stepper motor A+

2) DC motor 1 "-" or stepper motor A-

3) 12V jumper - remove this if using a supply voltage greater than 12V DC. This enables power to the onboard 5V regulator

4) Connect your motor supply voltage here, maximum of 35V DC. Remove 12V jumper if >12V DC

5) GND

6) 5V output if 12V jumper in place, ideal for powering your Arduino (etc.)

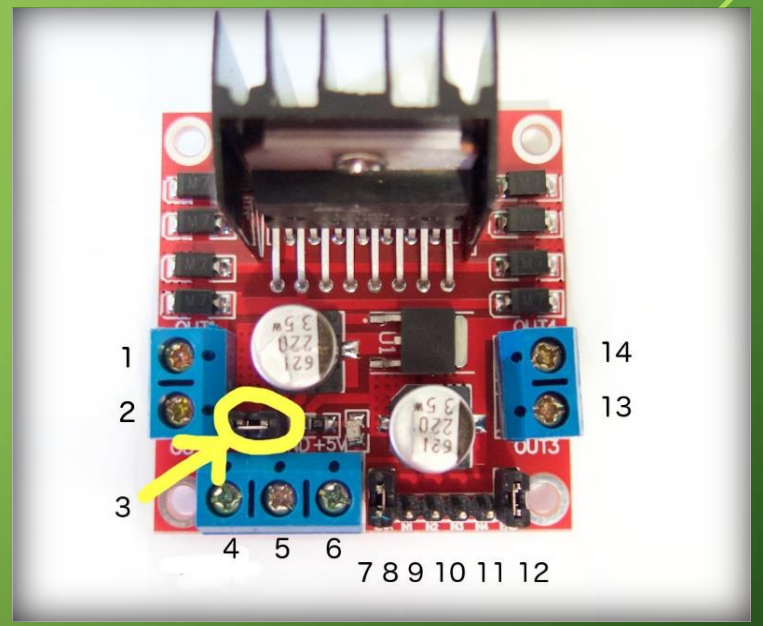
7) DC motor 1 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.

8) IN1

9) IN2

10) IN3

11) IN4



12) DC motor 2 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.

13)DC motor 2 "+" or stepper motor B+

14) DC motor 2 "-" or stepper motor B-

- **SPECEFICATION OF L298N MOTOR DRIVER:**

1) logic voltage - 5V
2) Motor drive voltage - 5~35V DC
3) maximum pow

3) HC-05 BLUETOOTH MODULE:

- HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection set up. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate)3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Blue core 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Freud The Bluetooth module HC-05 is a MASTER/SLAVE module. By default, the factory setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. envy Hopping Feature).

- **Hardware Features:**
  - Typical -80dBm sensitivity.
  - Up to +4dBm RF transmit power.
  - 3.3 to 5 V I/O.
  - PIO (Programmable Input/Output) control.
  - UART interface with programmable baud rate.
  - With integrated antenna.
  - With edge connector.
- **Software Features:**
  - Slave default Baud rate: 9600, Data bits:8, Stop bit:1,Parity:No parity.
  - Auto-connect to the last device on power as default.
  - Permit pairing device to connect as default.
  - Auto-pairing PINCODE:"1234" as default.
- **Pin Description:**
  - The HC-05 Bluetooth Module has 6pins. They are as follows:

1) **ENABLE:**

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

2) **Vcc:** Supply Voltage 3.3V to 5V

3) **GND:** Ground pin

4) **TXD & RXD:**

- These two pins acts as an UART interface for communication

5) **STATE:**

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

6) **BUTTON SWITCH:**

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

- The HC-05 Bluetooth the default pairing code is 1234.
- It has range up to <100m with depend upon transmitter and receiver, atmosphere, geographic & urban conditions.

4) 12-volt D.C Motor:

- We are using four 12-volt D.C motors. 200RPM Centre Shaft Economy Series DC Motor is high quality low cost DC geared motor. It has steel gears and pinions to unpropertied. The gears are fixed on hardened steel spindles polished to a mirror finish. The output shaft rotates in a plastic bushing. The whole assembly is covered with a plastic ring. Gearbox is sealed and lubricated with lithium grease and require no maintenance. The motor is screwed to the gear box from inside. Although motor gives 200 RPM at 12V but motor runs smoothly from 4V to 12V and gives wide range of RPM, and torque. Tables below gives fairly good idea of the motor's performance in terms of RPM and no-load current as a function of voltage and stall torque, stall current as a function of voltage sure longer life and better wear and tear.

- SPECIFICATION:
  - DC Supply: 4 to 12-volt
  - RPM: 200 at 12 volts
  - Gear Assembly: Spur
  - Output shaft: Centered.

- 5) **Jumper Cable:** In this project are also used Male to Male, Male to Female, and Female to Female. A **jump cable** (also known as jumper wire, jumper, DuPont wire, or DuPont cable – named for one manufacturer of them) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering. Jump wires at the end of a multi-colored ribbon cable are used to connect the pin header at the left side of a blue USB2Serial board to a white breadboard below. Another jumper cable ending in a USB micro male connector mates to the right side of the USB2Serial board. Red and black tinned jump wires can be seen on the breadboard. Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

6) One 12-volt battery

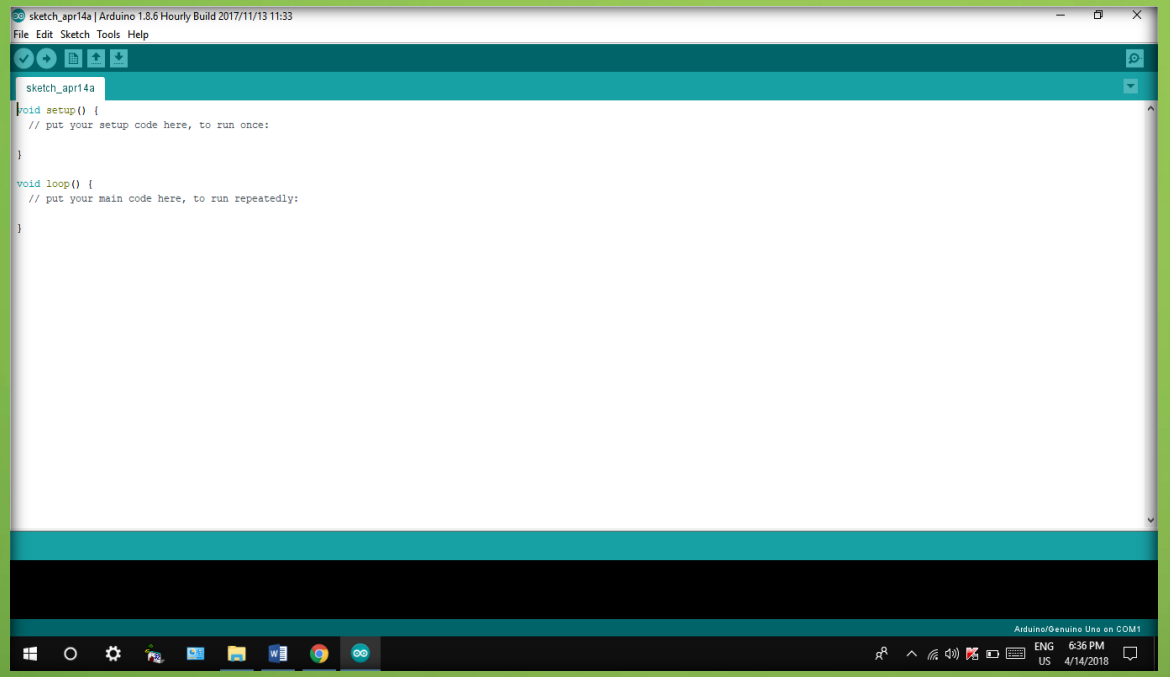
- Our project we will used 12-volt de battery. But We will use the 4-volt three Battery in Series connection. And the are 2 amperes.

Due to low price cost we are using this plain.

7) Vehicle Chassis:

We are using a rectangular type chassis for carrying all the equipment and it has four holes both left and right side for support four motor. The weight of the chassis around 250g.

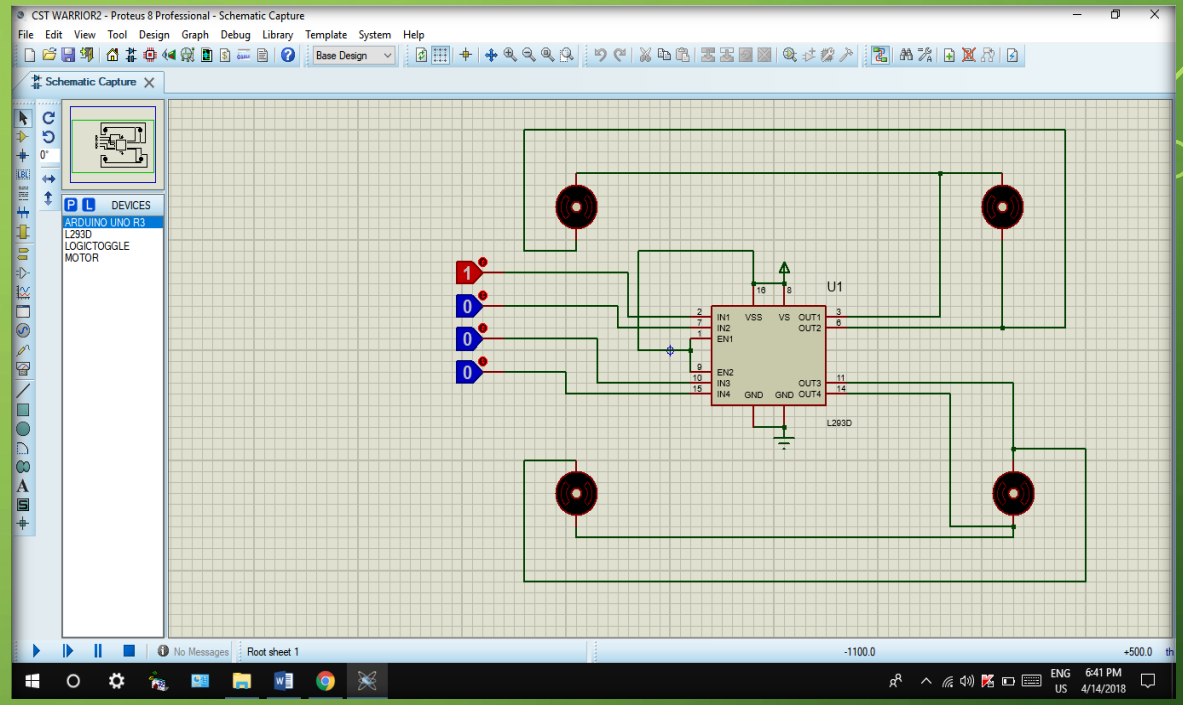
1) ARDUINO 1.8.6 HOURLY BUILD:



- It is mainly an IDE where we can write our code (embedded C), can compile the program and debug as well.
- In this software there are many readymade programs are internally stored. That can help us to understand the coding concept. After compile the program and conform that the program is error free, we will upload the compiled program into Arduino Uno board (microcontroller) via USB cable.
- We are using the latest Arduino 1.8.6 version application software. We can download the software from the following link… https://www.arduino.cc/en/Main/Software

We can also install the software on android mobile phones, though it is better on PC but it works same on both on pc and mobile.

2. PROTEUS 8.6 PROFESSIONAL:



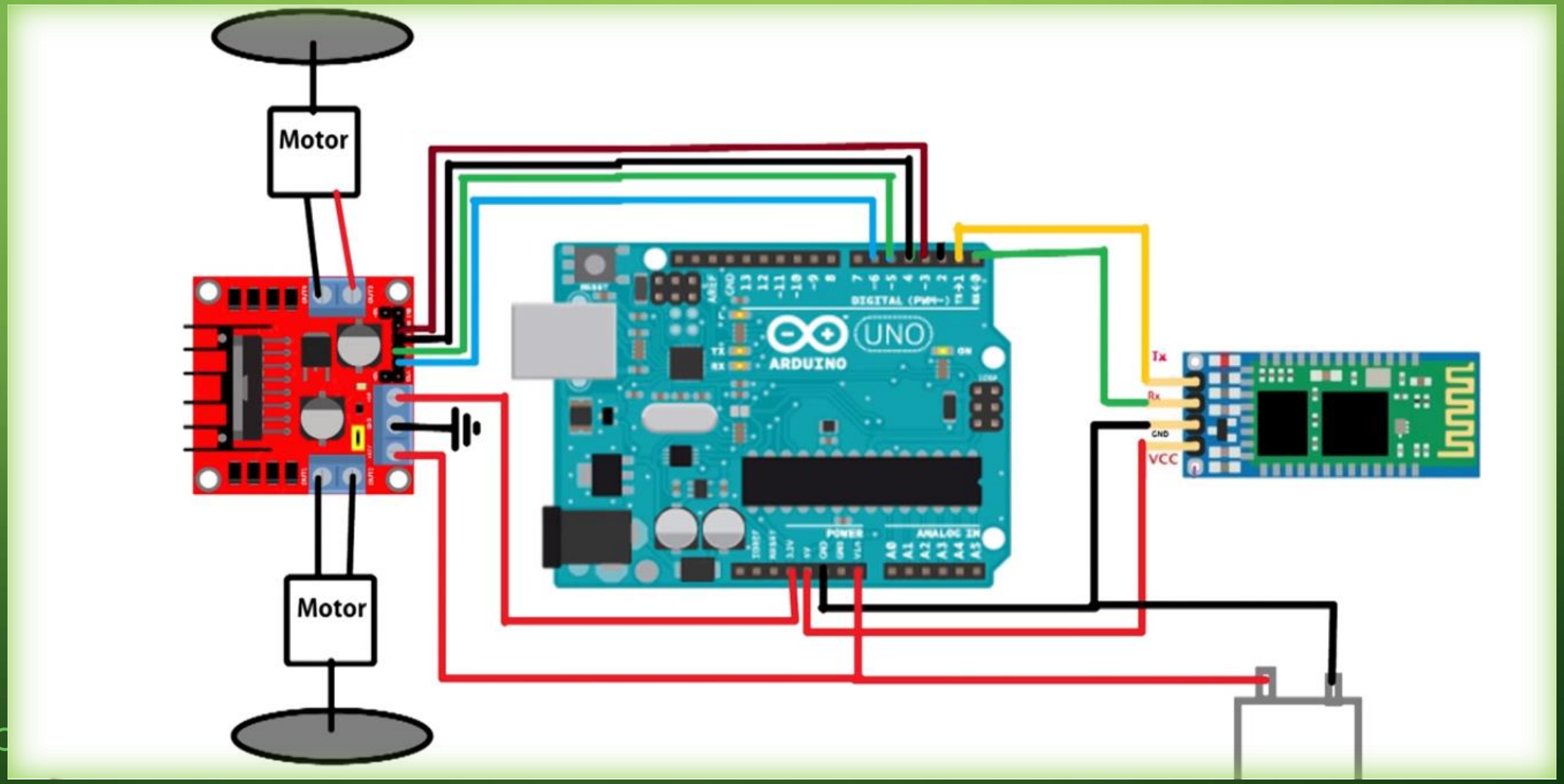
- At-first Logical representation of the project are design the following software. And we will use the latest version of software 8.6. It's basically electronic design Automation.
- The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.
- It was developed in Yorkshire, England by Lab center Electronics Ltd and is available in English, French, Spanish and Chinese languages.

3) AMR VOICE CONTROL FOR ARDUINO:

- AMR Voice Control for Arduino is a mobile Application. It's used to send command to the HC-05 Bluetooth module. It is very light weight Application And it's available on Play store. The Adaptive Multi-Rate audio codec is an audio compression format optimized for speech coding. AMR speech codec consist of Multi-rate narrowband speech codec that encodes narrowband (200-3400 Hz) signals at variable bit rate ranging from 4.75 to 12.2 Kbit/s with toll quality speech starting at 7.4 Kbit/s.
- Sampling frequency 8 MHz/13-bit (160 samples for 20 MS frame), filtered to 200-3400 Hz.
- The AMR codec uses 8 source codes with bit rate of 12.2, 10.2, 7.95, 7.40, 6.70, 5.90, 5.15 and 4.75 Kbit/s.
- AMR is a hybrid speech coder, and as such transmits both speech parameters and waveform signal.

DESIGN:

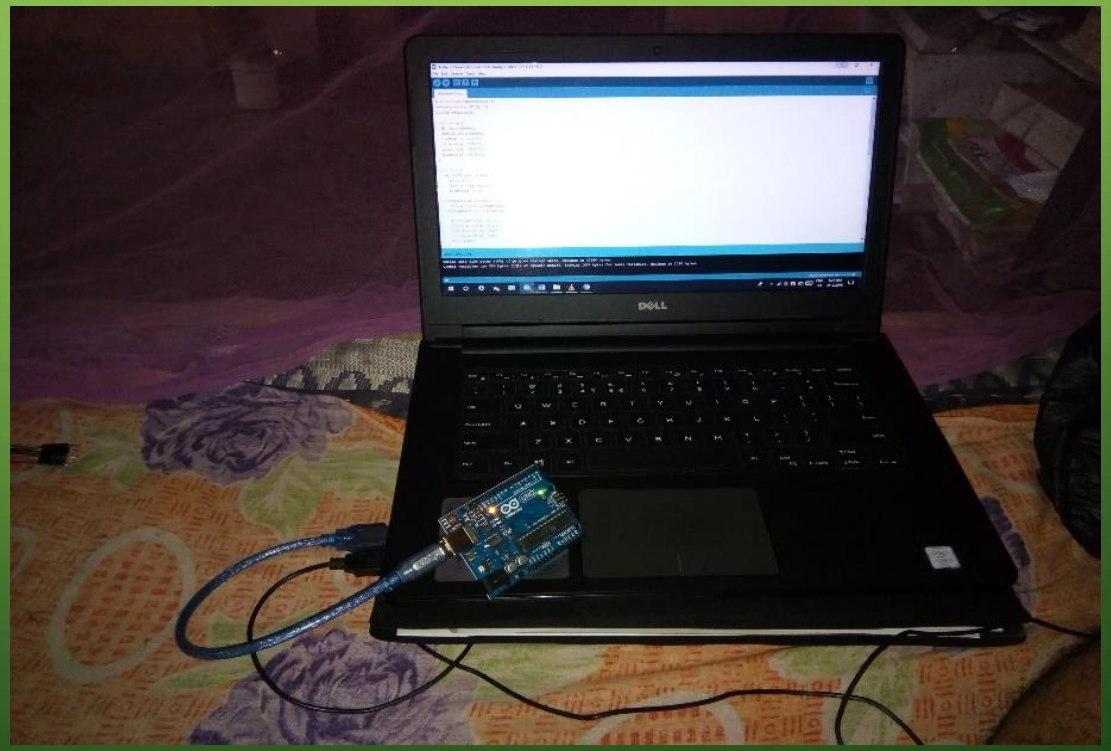
TOTAL CIRCUIT IN OUR PROJECT



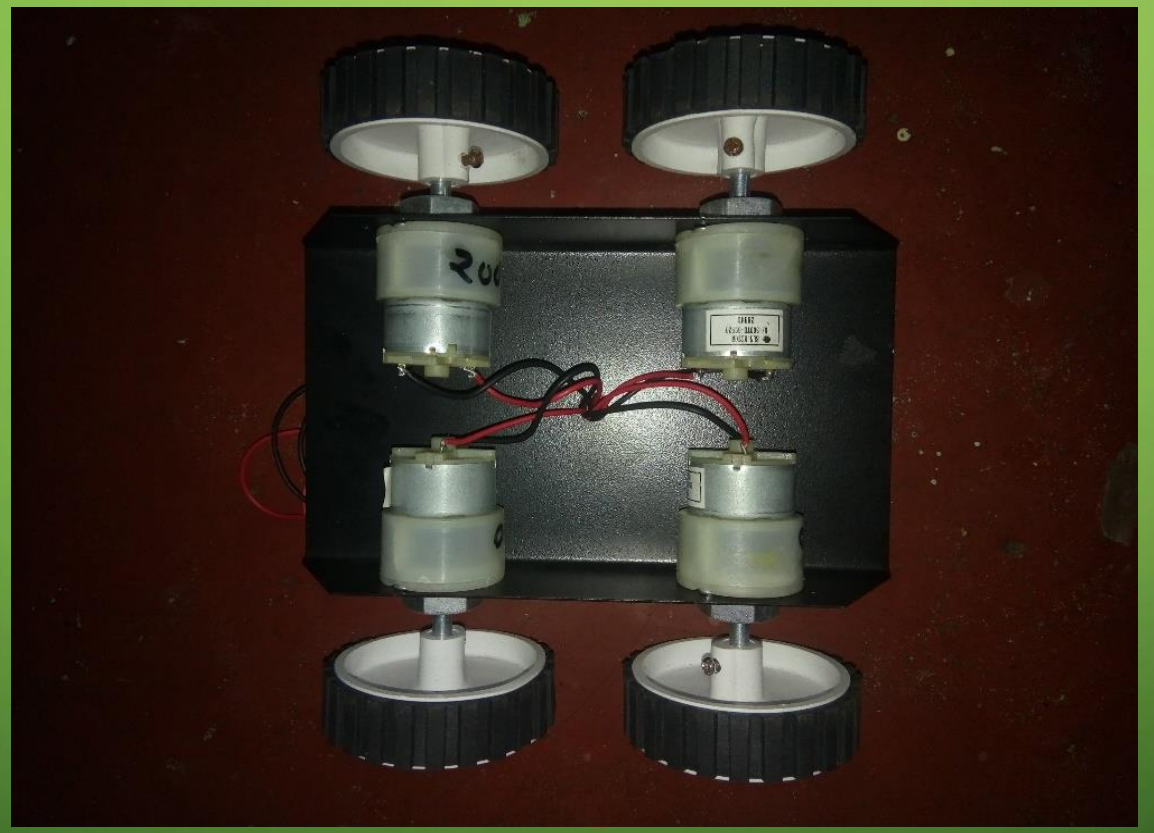
IMPLEMENTATION

- According to the circuit diagram we have assembled every hardware component and devices by using Jumper wares step by step.

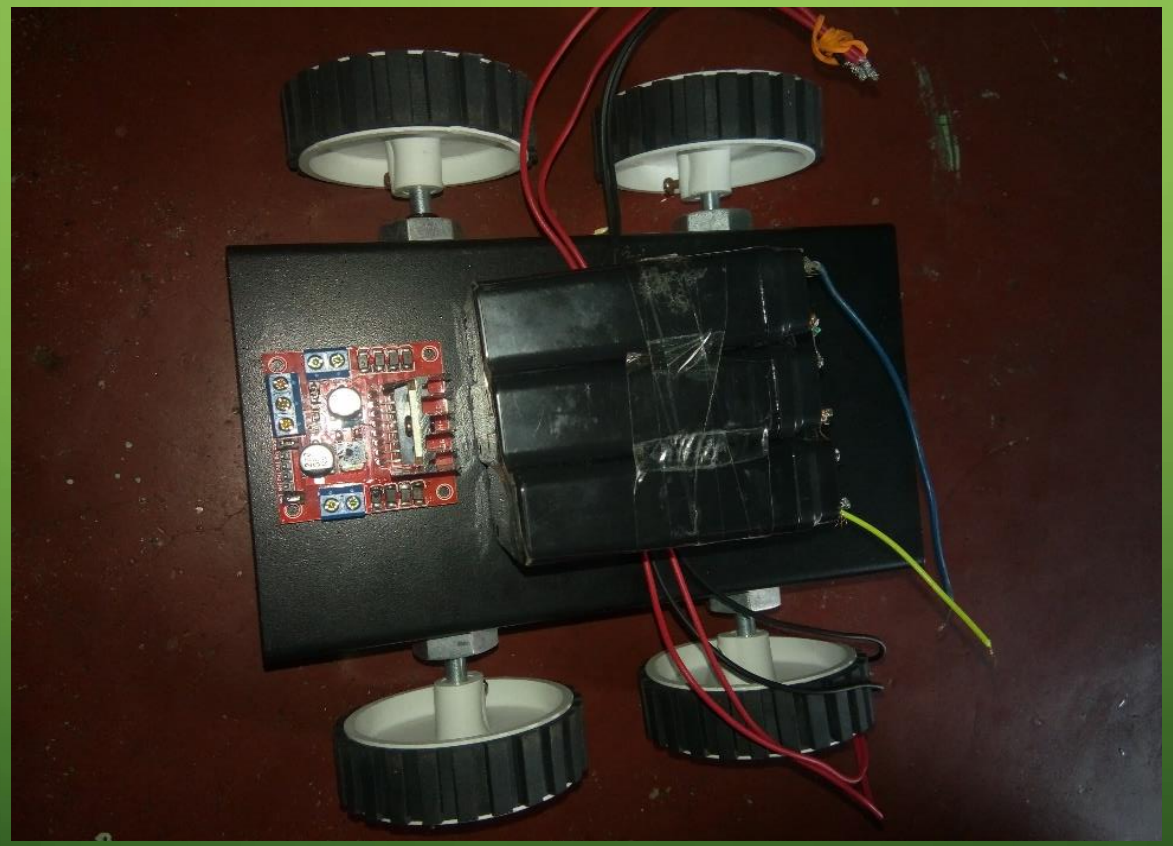
A) At first, we have compiled the program that we written, after that we have sure the program is error free then we have uploaded the program into “Arduino UNO R3” board by USB cable.



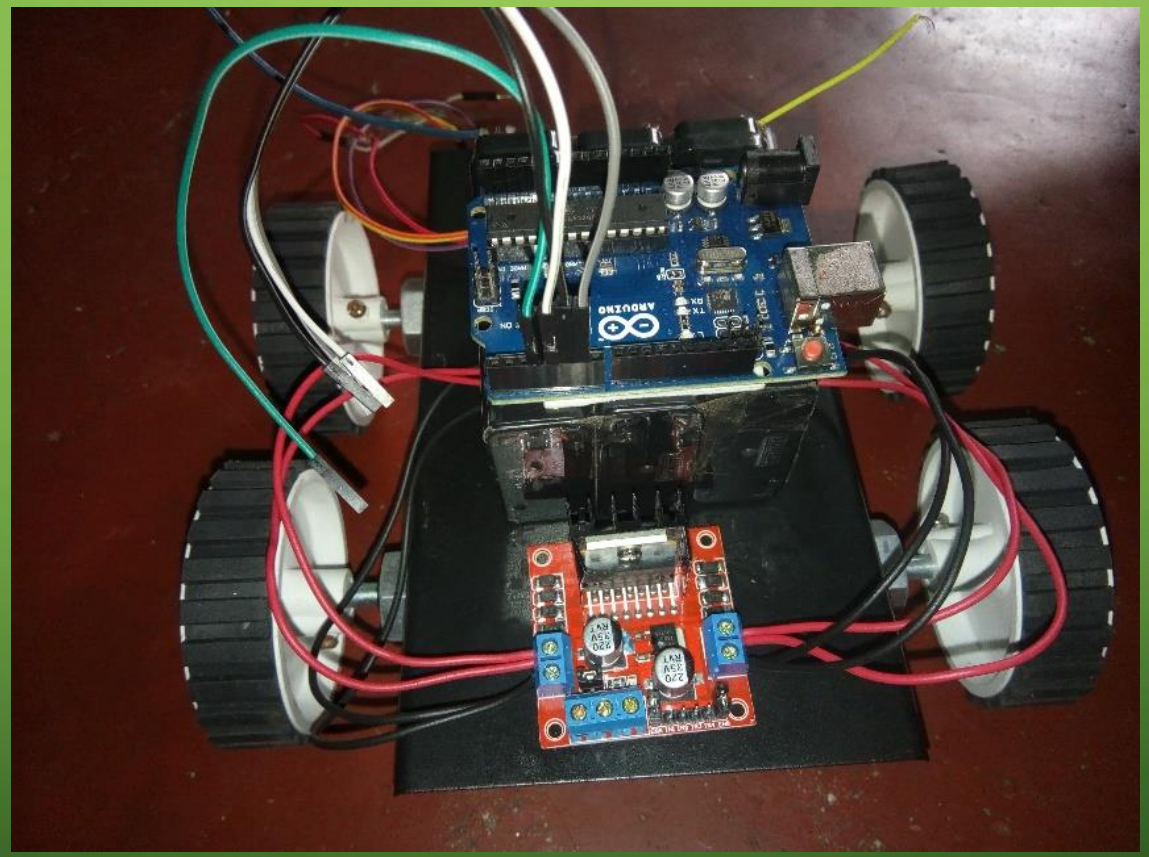
B) Now, we have assembled the 4 motors and wheels to the car chassis using nuts and bolts. Like the following picture…….



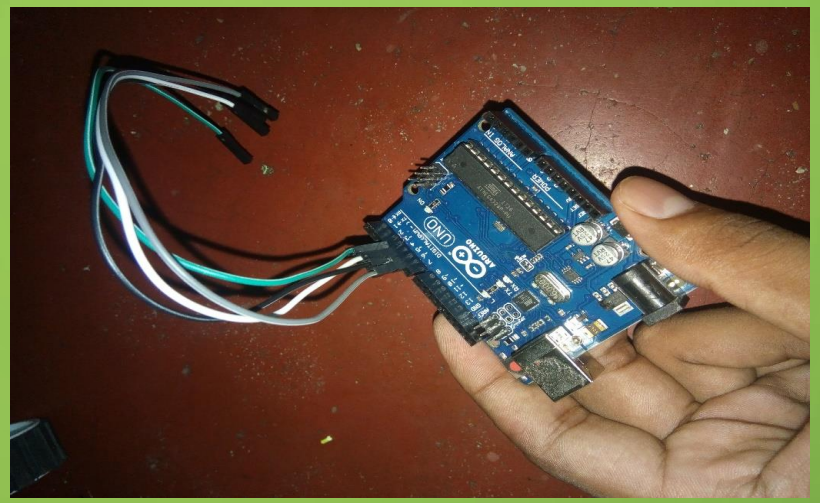
C) The Battery and the L298N Motor driver are fitted on the chassis' surface using two-sided tape. We have fitted the battery to the center of the chassis surface. Because it will work efficiently due to center of the gravity.



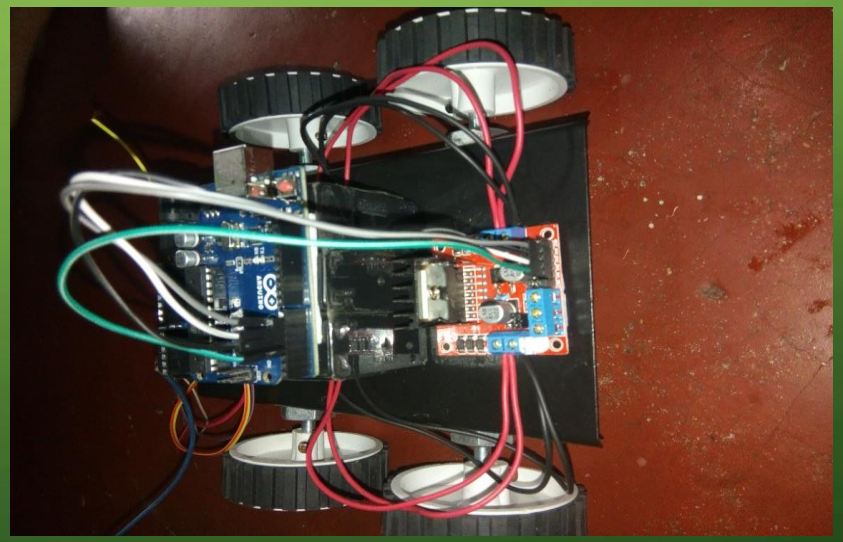
D) We have connected the right side two motor parallelly and also left side. The right side motor's wires are connected to the L298N Motor driver's output pin 1, and left motor's wires are also connected to Motor driver's output pin 2.



E) We have fitted the jumper wires to pin no 3,4,5,6 digital output pins of the Arduino UNO board.



After that we have fitted the above pins to the pin no 1,2,3,4 of the Motor driver, i.e. pin 3 to pin 1, pin 4 to 3, pin 5 to 2, pin 6 to 4 respectively.



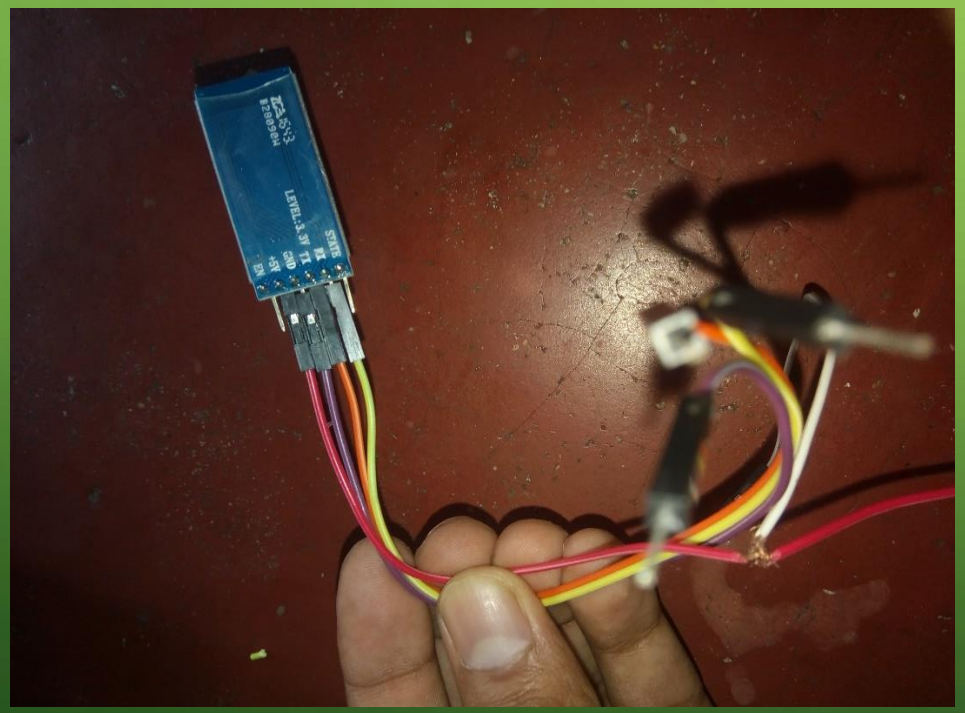
F) The Bluetooth device has four pins TX, RX, GROUND, VCC. We have fitted the jumper wires to each pin and these are connected to the pin TX, RX, 5-volt vcc, Ground of the Arduino Board.

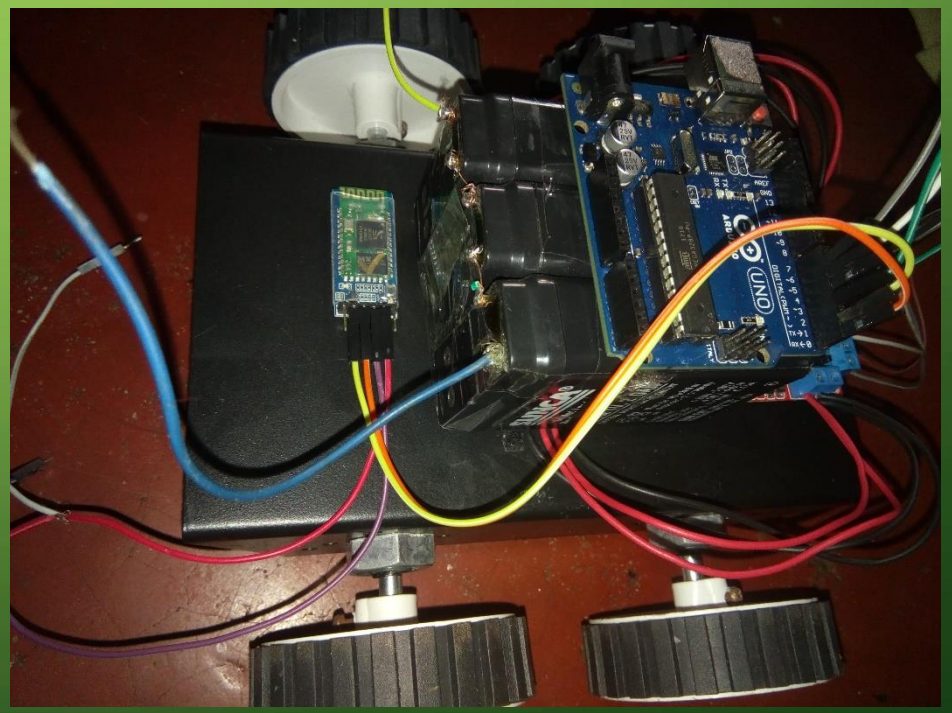
i.e. TX --- RX

RX --- TX

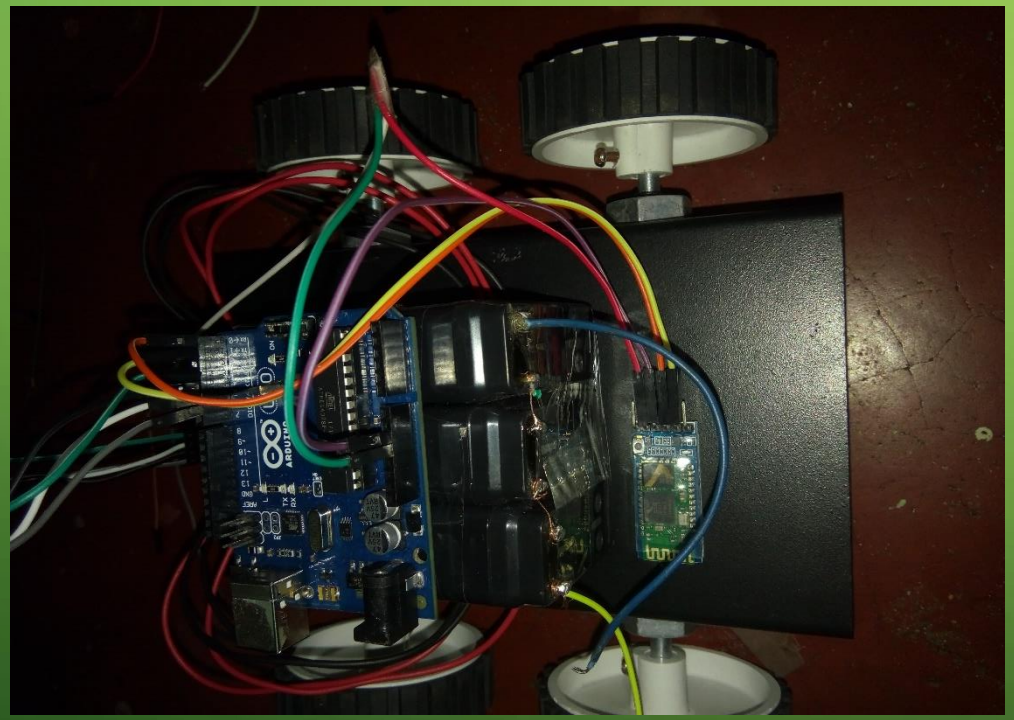
GND --- GND

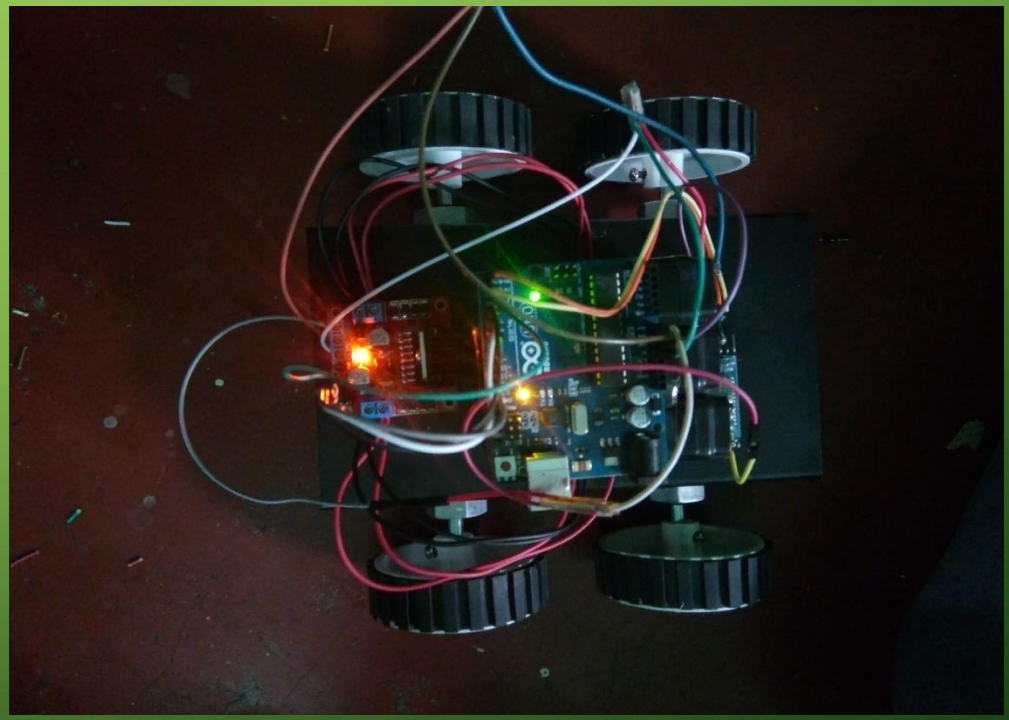
5 volts --- 5volt





G) Finally, we have reached in the end of the assembling, so we will give power to Arduino Board and L298N Motor driver from 12 volts Battery. We have to conform that every Board or Device are taking correct power. That is the LED light of the board are kindling.





- H) The assembling process has done and every device getting correct power so, now we will open AMR Voice Control Application on Mobile and will connect to the HC-05 Bluetooth Module to Mobile. And now we will give command by clicking the voice button on the Mobile Application to run the Car according to the program.

  i.e. (go, back, left, right)

Coding

WE HAVE TO UPLOAD THE FOLLOWING PROGRAM INTO ARDUINO UNO BOARD.

```
#include<SoftwareSerial.h>
SoftwareSerial BT(0 , 1);  //TX, RX Respectively
Srting readvoice;
Void setup()
{
        BT.begin(9600);
        Serial.begin(9600);
        pinMode(3,OUTPUT);
        pinMode(4,OUTPUT);
        pinMode(5,OUTPUT);
        pinMode(6,OUTPUT);
}
Void loop()
{
        While(BT.avalable())   //check available byte to read

```

```
{
        Delay(10);                                      //delay added to make things stable
        Char c=BT.read();//conduct a serial read
        readvoice += c;         //build the string- left,right,forword,backword
        }
        If(readvoice.length()>0){
                Serial.println(readvoice);
        if(readvoice="*forward#")
        {
                digitalwrite(3,HIGH);
                digitalwrite(4,HIGH);
                digitalwrite(5,LOW);
                digitalwrite(6,LOW);
        }
        else if(readvoice="*back#")
        {
                digitalwrite(3,LOW);
                digitalwrite(4,LOW);
                digitalwrite(5,HIGH);
                digitalwrite(6,HIGH);
        }
```

```
else if(readvoice="*left#")
                {
                        digitalwrite(3,LOW);
                        digitalwrite(4,HIGH);
                        digitalwrite(5,LOW);
                        digitalwrite(6,LOW);
                }
                else if(readvoice="*right#")
                {
                        digitalwrite(3,HIGH);
                        digitalwrite(4,LOW);
                        digitalwrite(5,LOW);
                        digitalwrite(6,LOW);
                }
                else if(readvoice="*STOP#")
                {
                        digitalwrite(3,LOW);
                        digitalwrite(4,LOW);
                        digitalwrite(5,LOW);
                        digitalwrite(6,LOW);
                }
readvoice=" ";  //reset the variable
}
}
```

6) TESTING

After implementation of the car and given power to the boards, we confirm that the circuit is working correctly i.e. all the lights are kindling. After conformation we have opened the AMR voice control application and connected to the Bluetooth module of the car. Then we sent the voice commands (“go” to move forward, “back” to move backward, “left” to move left and “right” to move right respectively) to move the car according to the program. Finally, we confirmed, all the commands are working correctly and the car is moving correctly.

7) EVOLUTION AND RISK ANALYSIS:

- **A. ADVANTAGE:**

When we say voice command, the first term to be consider is speech Recognition i.e. making the system to understand human voice. Speech recognition is a technology where the system understands the word (not its meaning) given through speech.

Speech is ideal method for robotic control and communication. The speech recognition circuit we will online, function independently from the robot's main indolence [CPU]. This is a good thing because it doesn't take any of the Robot's main CPU processing power for word recognition. The CPU must merely poll the speech circuit's recognition lines occasionally to check if a command has been issued to the Robot.

We can even improve upon this by connecting the recognition line to one of the robot's CPU interrupts lines. By doing this, a recognized word would cause an interrupt, letting the CPU know a recognized word has been spoken. The advantage of using an interrupt is that polling the circuit's recognition line occasionally would no longer be necessary. Further reducing any CPU overhead.

Another advantage to this stand-alone speech-recognition circuit (SRC) is its programmability. You can program and train the SRC to recognize the unique words you want recognized. The SRC can be easily interfaced to the robot's CPU.

To control and command an appliance (computer, VCR, TV security system, etc.) by speaking to it will increasing the efficiency and effectiveness of working with that device. At its most basic level speech recognized allows the user to perform parallel tasks, (i.e. hand and eyes are busy elsewhere) while continuing to work with the computer or appliance.

Robotics is an evolving technology. There are many approaches to building Robots, and no one can be sure which method or technology will be used 100 years from now. Like biologically system, robotics is evolving following the Darwinian model of the fittest. Suppose you have to control a menu driven system. What is the most striking properly that you can do of?

Well the first though that come to our mind is that the range of input is a menu driven system is limited. In fact, by using a menu all we are doing is limiting the input domain space. Now, this is one character which can be very useful in implementing the menu in standalone system.

B) DISADVANTAGE:

"It makes man even more lazier" .....

- **APPLICATION THIS PROJECT:**
  - Military Application
  - Industrial Application
  - Household Application
  - Shopping Mall
  - Most important thing it is a gift for physically Handicapped.

- **FUTURE SCOPE:**
  - We can also extend our project to advantaged application by extending its range, by using Zibee technology or by using GPS.
  - We can also process to construct a full-fledged robot to carry out more complex tasks.
  - We can program the robot to shoot a target.

6) CONCLUTION:

The main aim of this project is to give a small gift to the unfortunate people who are physically handicapped. By giving simple voice command, their daily chores can be very easily accomplished. The main challenge that we faced during this project's implantation include, the range of the Bluetooth, the daily between the transmissions and reception of the voice command. This project proves to be a significant asset to the field of Robotics.