Tug of War

—

SRB DEMO FILES
RoboVR - The Olympics of Robots
WWW.ROBOVR.WORLD

Introduction

The Tug of war bot is a four-wheel drive car with the body attached to it and it has a c sectional grove to tie the rope at the body and is powered by minimal number of electrical components, featuring instantaneous response to driver's input. The bot is compact and light, enabling it to traverse through any terrain. The driver can modify various features according to his will, the main objective here is to pull the opposition.

Requirements

A list of requirements and goals was developed to help guide me in the development of this project. The requirements are as follows.

- The bot shall be controlled by a Bluetooth controller.
- The bot shall be battery powered.

Materials

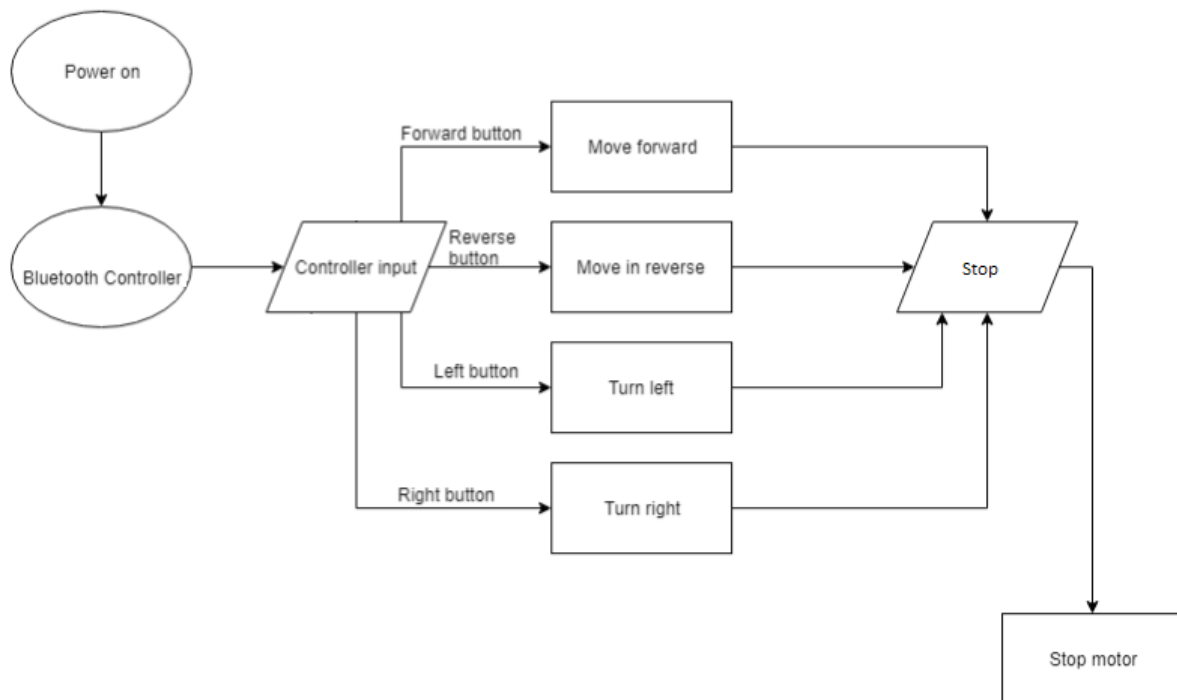| Arduino Uno R3 |
| --- |
| HC-05 Bluetooth Module |
| 12V Lithium-ion Battery |
| L298N Motor Driver Module |
| DC Motors (300 RPM) |
| Car Chassis |
| Servo Motors |

In addition to the hardware used on the arms and the car itself, there was some other hardware necessary to build and the arms of the bot and to run the car. These additional tools included a soldering iron and some solder to assemble the arms of the bot and to assemble RC car and solder wires to the pins connected to the motor batteries. In additional an Android phone is required along with the app "Arduino Bluetooth Control" which can be found in the Google Play Store.

The app has prebuilt symbols, the user just needs to assign various characters(A-Z or a-z) to these symbols and press enter after assigning characters to each symbols.

Once all the required characters are assigned the user should select "switch to controller mode" for connecting the bluetooth with the arduino microcontroller.

If asked for a password, the password may be 0000 or 1234 and the name of the bluetooth module will be HC-05.

Working

The working of this tug of war bot is based on Arduino UNO microcontroller. The controller is used for communication between various electrical components. Motor driver is connected to the controller and is used to provide sufficient power to the motors connected to it and acts as a medium for communication between Arduino and the motors. The Bluetooth module helps in communicating the desired motion via a mobile application. The predefined code helps to achieve forward, backward, right turn and left turn motion.

NOTE:

When uploading the program to the Arduino the connection to the GPIO pin D0 and D1 had to be temporarily removed until uploading. This became somewhat of a hassle after many times having to upload new code. This must be done since the D0 and D1 pin is used for serial communication between the Arduino and computer.
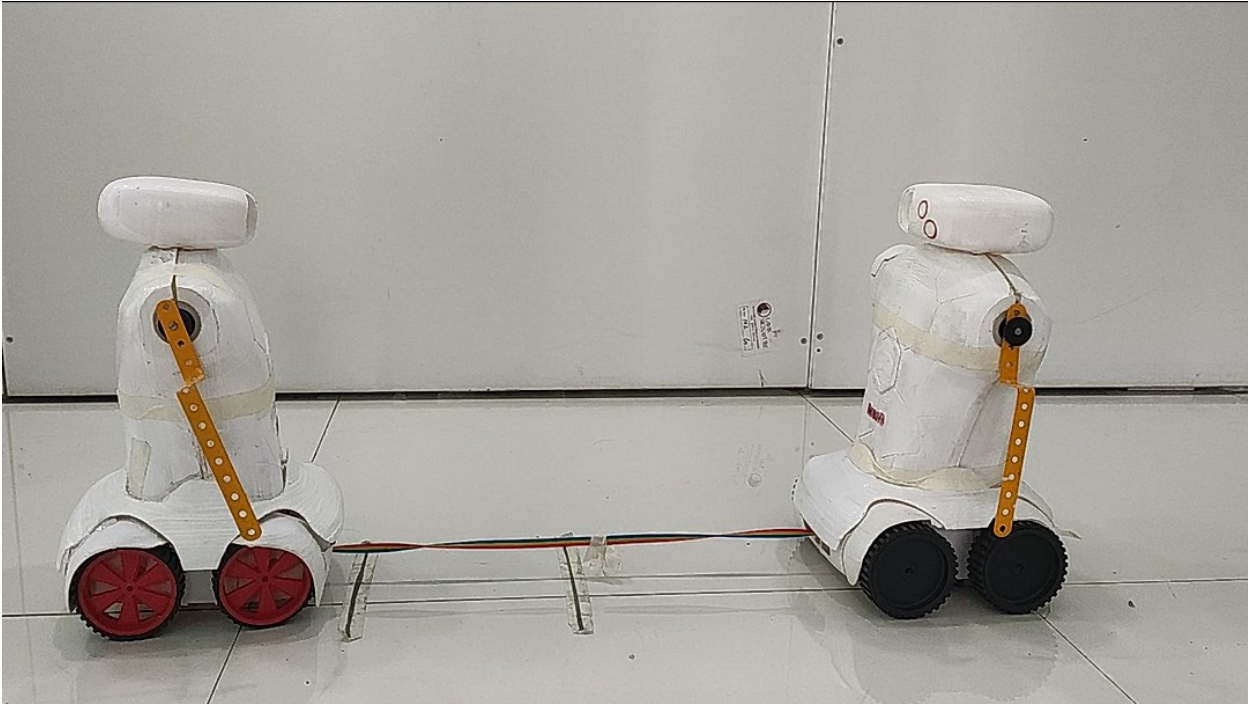
Circuits and Connections
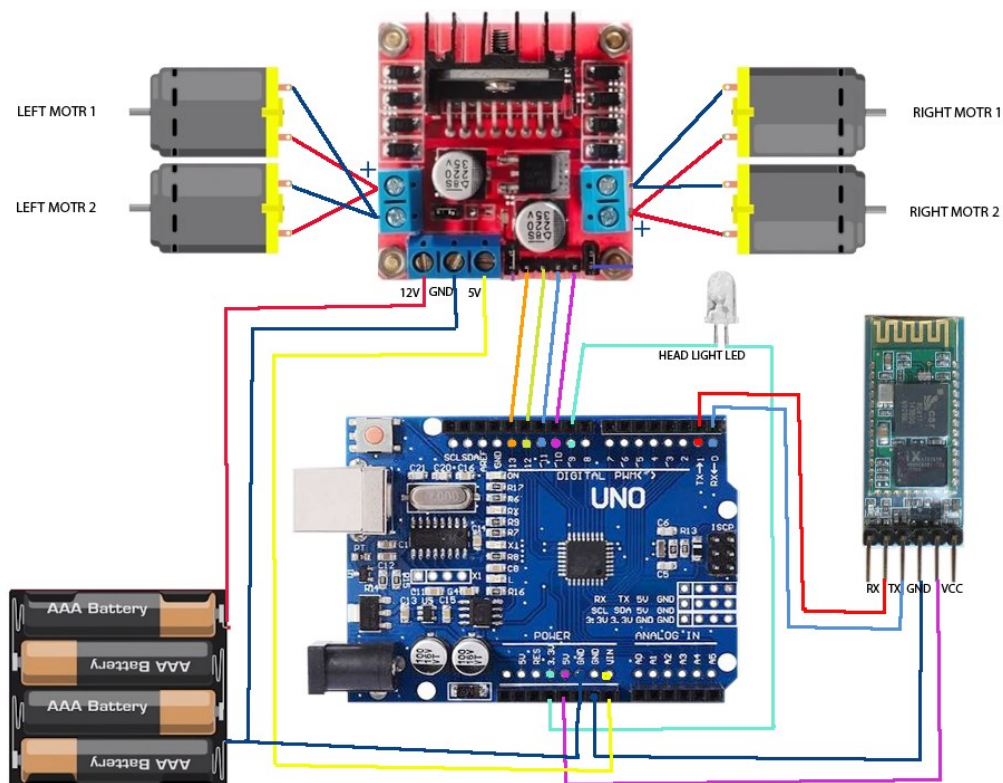
The circuit is quite connections are quite simple, there is one servo motors connected to a pin of the Arduino module that will be used as control inputs for performing arm movements.
The Vcc and Ground pins of the servo motors should be connected to the Vcc and ground pins of the Arduino.

The four motors that are used to move the bot are connected to the L298N motor driver module as shown in the figure.

RoboVR Design:

RIGHT SERVO MOTOR — LEFT SERVO MOTOR

RIGHT REAR WHEEL — LEFT REAR WHEEL

BOT'S BODY MOUNTED ON A 4 WHEEL DRIVE CAR

RIGHT FRONT WHEEL — LEFT FRONT WHEEL

Code:

```
char ch='S'; //initially STOP condition
const int IN1=9;// Define RIGHT MOTOR
const int IN2=10;
const int IN3=11;// Define LEFT MOTOR
const int IN4=12;

void setup()
{
 pinMode(IN1,OUTPUT);
 pinMode(IN2,OUTPUT);
 pinMode(IN3,OUTPUT);
 pinMode(IN4,OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if(Serial.available())   // check if data is availabe or not
  {
   ch=Serial.read();        //  read the incoming byte
  }
 Serial.println(ch);
    if (ch=='R')
    {
      RIGHT();
    }
    if (ch=='L')
    {
      LEFT();
    }
    if (ch=='F')
    {
     FORWARD();
    }
    if (ch=='B')
    {
      BACKWARD();
```

```arduino
    }
    if (ch=='S')
    {
      STOP();
    }
}
void FORWARD(){
  Serial.println("forward");
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);
}
//-------------------------------------------
void BACKWARD(){
  Serial.println("backward");
 digitalWrite(IN1,LOW);
 digitalWrite(IN2,HIGH);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
}


//-------------------------------------------
void LEFT(){
  Serial.println("LEFT");
 digitalWrite(IN1,LOW);//MOVING RIGHT MOTOR IN CLOCKWISE DIRECTION
  digitalWrite(IN2,HIGH);
  digitalWrite(IN3,LOW);//MOVING LEFT MOTOR IN ANTICLOCKWISE DIRECTION
  digitalWrite(IN4,HIGH);
}
//-------------------------------------------
void RIGHT(){
  Serial.println("RIGHT");
  digitalWrite(IN1,HIGH);//MOVING RIGHT MOTOR IN ANTICLOCKWISE DIRECTION
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,HIGH);//MOVING LEFT MOTOR IN CLOCKWISE DIRECTION
  digitalWrite(IN4,LOW);

}
//-------------------------------------------
void STOP(){
  Serial.println("stop");
```

```
digitalWrite(IN1,LOW);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,LOW);
digitalWrite(IN4,LOW);
  delay(100);
}
```