

Robo Race Manual

Description:

This robot must run faster in the arena and at the same time it must cross all obstacles in the arena.

Components:

4 Wheel Chassis Kit



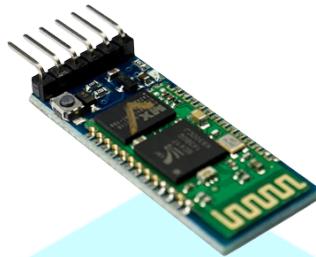
Arduino UNO:



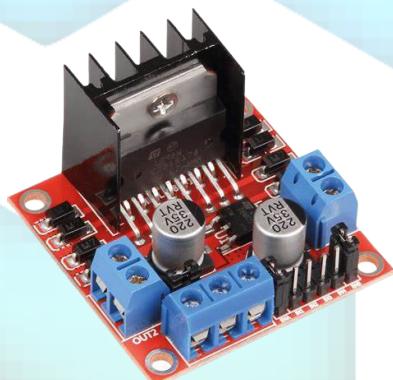
Arduino UNO Cable:



HC-05 Bluetooth Module



L298n Motor Driver:



Male to Female Jumper Wires:



Male to Male Jumper Wires:



18650 2200mAh Lithium-Ion Battery:



OR

Lithium-Polymer Battery(Opt):

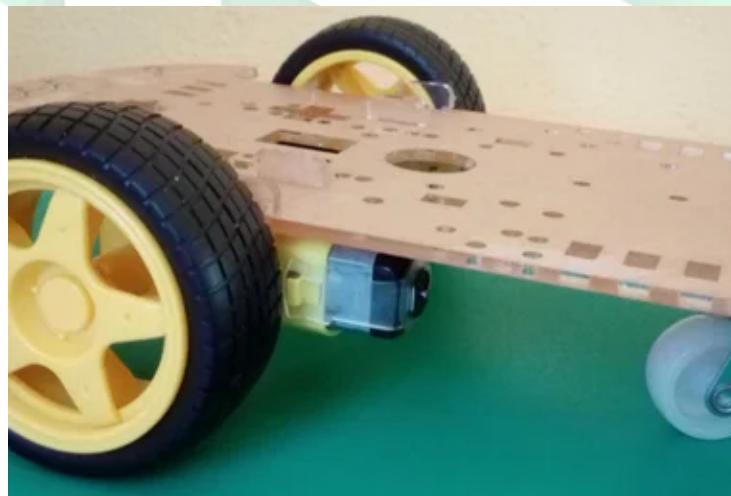
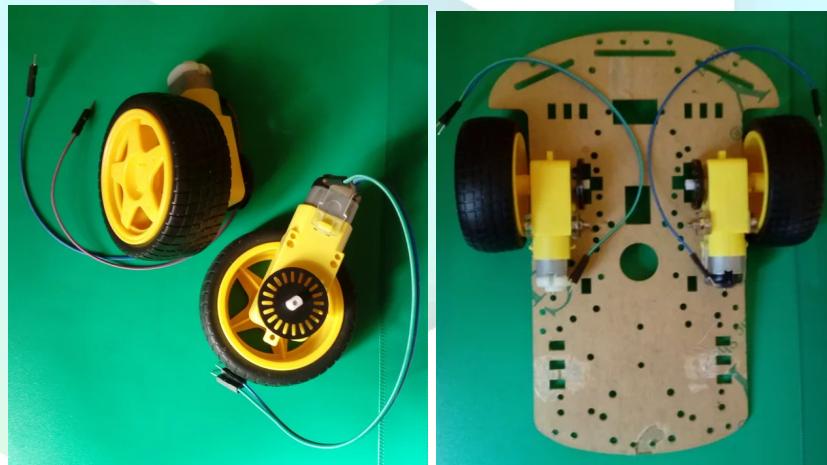


Battery Holder:

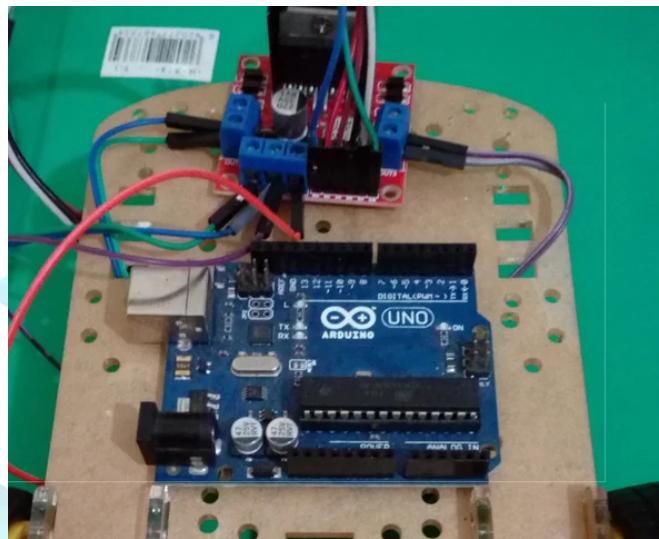


Procedure:

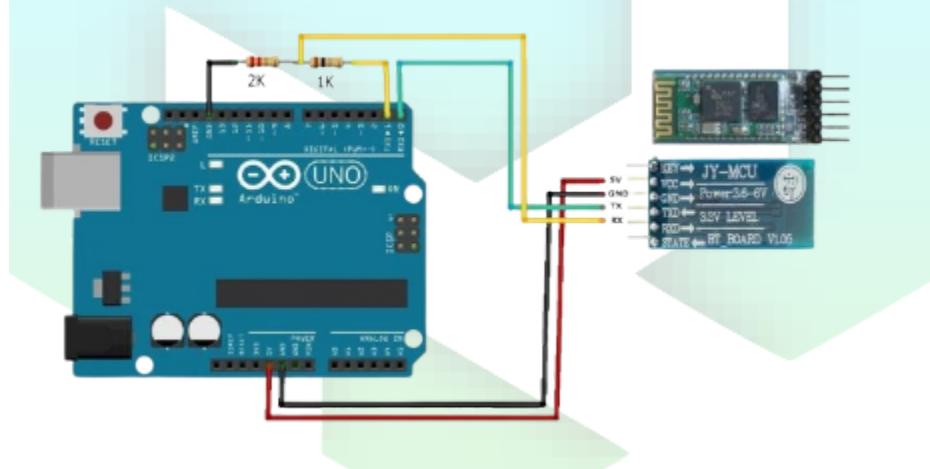
- ❖ Step 1 - Assemble the motor with the chassis kit's base.



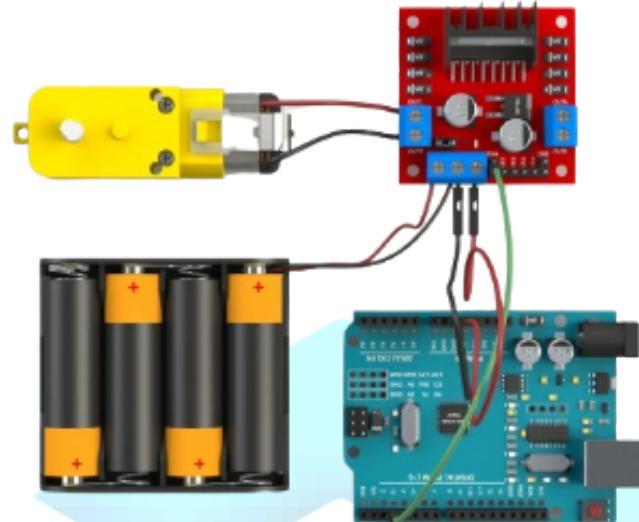
- ❖ **Step 2** - Then place the code uploaded arduino UNO on the base of the chassis.
- ❖ **Step 3** - Place the motor driver and connect the motors with the motor driver and connect the input pin from the motor driver with the arduino board as mentioned pins in the code respectively.



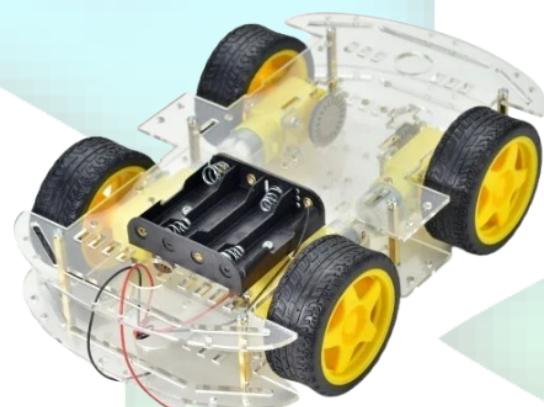
- ❖ **Step 4** - Place the bluetooth module and connect it with the arduino board as mentioned pins in the code respectively.



- ❖ **Step 5** - connect the positive and negative wires of the battery holder to the motor driver then take the output power from the motor driver, connect it to the arduino board.



- ❖ **Step 6** - Then assemble the top of the chassis and place the battery holder on the top. If you are using the lithium polymer battery there is no need for a holder you must place the switch. The advantage of the lithium polymer battery is it is more power efficient then the lithium ion battery.



Code:

```
#include <SoftwareSerial.h>
SoftwareSerial bluetooth(1,2); // RX, TX
char t;

#define MLa 8 //left motor 1st pin
#define MLb 9 //left motor 2nd pin
#define MRa 10 //right motor 1st pin
```

```
#define MRb 11 //right motor 2nd pin
```

In this part of the code, first we have to import a library called **<SoftwareSerial>**, Because we are going to use the bluetooth module our bot acts by the commands we are going to give from the mobile so that our arduino and mobile must connect with each other. This library is helpful for that. Then the motor input pins are defined. I gave the name like mla, mlb this name was our wish. But once you define the name in this place you will not use another name inside the code.

```
void setup()
{
  Serial.begin(9600);
  bluetooth.begin(9600);
  pinMode(MLa,OUTPUT); //left motors forward
  pinMode(MLb,OUTPUT); //left motors reverse
  pinMode(MRa,OUTPUT); //right motors forward
  pinMode(MRb,OUTPUT); //right motors reverse
}
```

In this part of the code, I am defining the out motors as output or input, and telling them to start reading the bluetooth signal.

```
void loop()
{
if(bluetooth.available()>0)
{
  t = bluetooth.read();
  Serial.println(t);
}
```

In this part of the code, I gave the command like if the bluetooth module sends any signal store it in the variable called **t**, this variable name also your wish is the same as the defining process.

```
switch(t)
{
  case 'F':
    { //move forward(all motors rotate in forward direction)
      digitalWrite(MLa,HIGH);
      digitalWrite(MLb,LOW);
```

```

    digitalWrite(MRa,HIGH);
    digitalWrite(MRb,LOW);
}
break;

case 'B':
{ //move reverse (all motors rotate in reverse direction)
  digitalWrite(MLa,LOW);
  digitalWrite(MLb,HIGH);
  digitalWrite(MRa,LOW);
  digitalWrite(MRb,HIGH);
  delay(10);
}
break;

```

In this part of the code, I am creating a switch using the **t** variable then if the bluetooth module received the **F** do the case in that case I gave the motor's forward rotation input HIGH backward rotations as LOW, so the bot will move forward. If the signal is **B** the motors will rotate in backward direction so the bot will move backward.

```

case 'L':
{ //turn right (left side motors rotate in forward direction, right side motors
  doesn't rotate)
  digitalWrite(MLa,HIGH);
  digitalWrite(MLb,LOW);
  digitalWrite(MRa,LOW);
  digitalWrite(MRb,LOW);
}
break;

case 'R':
{ //turn left (right side motors rotate in forward direction, left side motors
  doesn't rotate)
  digitalWrite(MLa,LOW);
  digitalWrite(MLb,LOW);
  digitalWrite(MRa,HIGH);
  digitalWrite(MRb,LOW);
}
break;
case 'S':

```

```
{
    //STOP (all motors stop)
    digitalWrite(MLa,LOW);
    digitalWrite(MLb,LOW);
    digitalWrite(MRa,LOW);
    digitalWrite(MRb,LOW);
}
```

In this part of the code, same as like above if the case is **L** the bot will turn left, else if **R** it will turn right or else if **S** the bot will stop running, the break statement i was used at the each case end because, you press the button then only that action will run otherwise it will stop automatically, if the break was not there, if you press the forward key the bot will run forward until you stop.

Full code:

```
#include <SoftwareSerial.h>
SoftwareSerial bluetooth(3,2); // RX, TX
char t;

#define MLa 8 //left motor 1st pin
#define MLb 9 //left motor 2nd pin
#define MRa 10 //right motor 1st pin
#define MRb 11 //right motor 2nd pin

void setup()
{
    Serial.begin(9600);
    bluetooth.begin(9600);
    pinMode(MLa,OUTPUT); //left motors forward
    pinMode(MLb,OUTPUT); //left motors reverse
    pinMode(MRa,OUTPUT); //right motors forward
    pinMode(MRb,OUTPUT); //right motors reverse
}

void loop()
{
    if(bluetooth.available()>0)
    {
        t = bluetooth.read();
        Serial.println(t);
    }
}
```

```

}

switch(t)
{
  case 'F':
    {      //move forward(all motors rotate in forward direction)
      digitalWrite(MLa,HIGH);
      digitalWrite(MLb,LOW);
      digitalWrite(MRa,HIGH);
      digitalWrite(MRb,LOW);
    }
    break;

  case 'B':
    {    //move reverse (all motors rotate in reverse direction)
      digitalWrite(MLa,LOW);
      digitalWrite(MLb,HIGH);
      digitalWrite(MRa,LOW);
      digitalWrite(MRb,HIGH);
      delay(10);
    }
    break;

  case 'L':
    {    //turn right (left side motors rotate in forward direction, right side motors doesn't
rotate)
      digitalWrite(MLa,HIGH);
      digitalWrite(MLb,LOW);
      digitalWrite(MRa,LOW);
      digitalWrite(MRb,LOW);
    }
    break;

  case 'R':
    {    //turn left (right side motors rotate in forward direction, left side motors doesn't
rotate)
      digitalWrite(MLa,LOW);
      digitalWrite(MLb,LOW);
      digitalWrite(MRa,HIGH);
      digitalWrite(MRb,LOW);
    }
    break;
}

```

```
case 'S':  
{      //STOP (all motors stop)  
  digitalWrite(MLa,LOW);  
  digitalWrite(MLb,LOW);  
  digitalWrite(MRa,LOW);  
  digitalWrite(MRb,LOW);  
}  
}  
}
```

Mobile App:

Link : <https://play.google.com/store/apps/details?id=braulio.calle.bluetoothRCcontroller>

QR Code:

