

---

# User manual

For

## Arduino Ultrasonic Car Kit



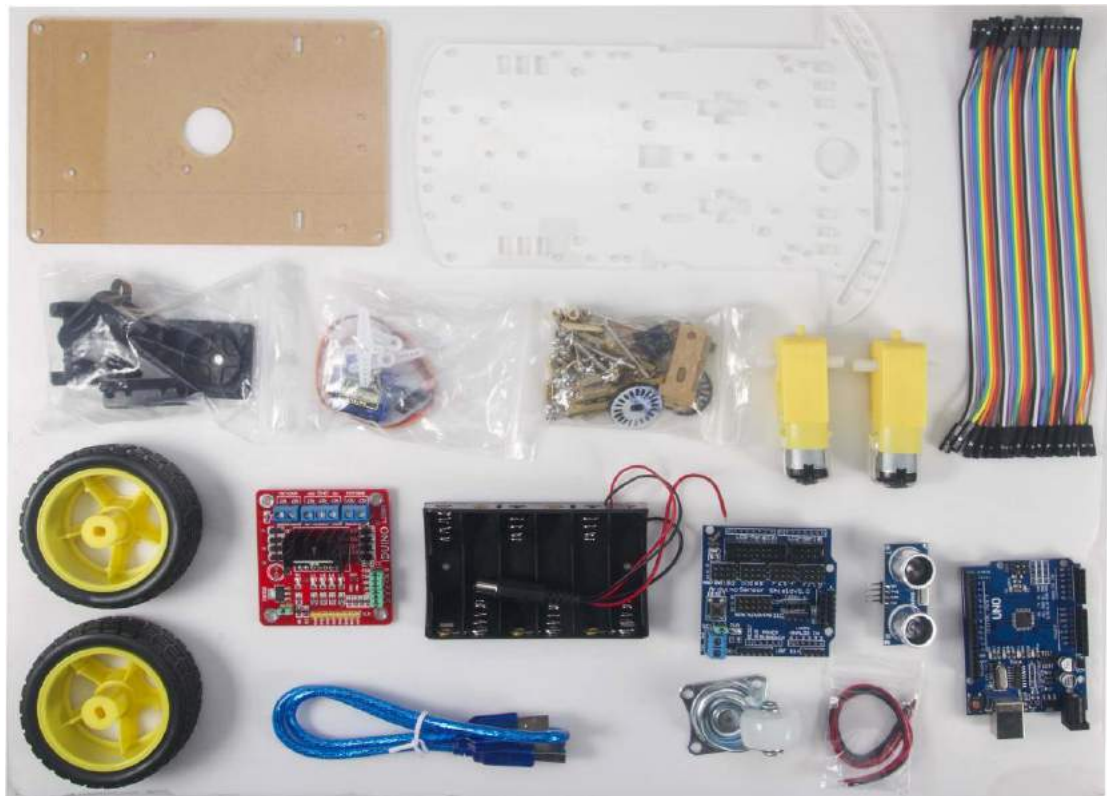
---

## Introduction:

This ultrasonic DIY car is based on Arduino development board(open source electronic platform). In this manual, we mainly use four Arduino parts: UNO board(Compatible with arduino); Ultrasonic module; L298N motor driver board; Arduino sensor extension board. All of this parts are very common used in Arduino project.

## Component list

- Arduino compatible UNO R3 \*1
- L298N motor driver board\*1
- Arduino sensor extension board \*1
- Ultrasonic module \*1
- SG90 servo\*1
- Car chassis \*2
- Car Wheels 2
- DC Gear Motor(1:48) \*2
- 10 Jumper Cables
- Fasteners \*4
- Universal wheel\*1
- 6AA battery box \*1
- Several screw nut
- FPV holder(for Servo)\*1



This kit contains some unused parts, such as

- One black plastic part from the FPV holder bag
- Some plastic parts and screw supplied in the servo bag

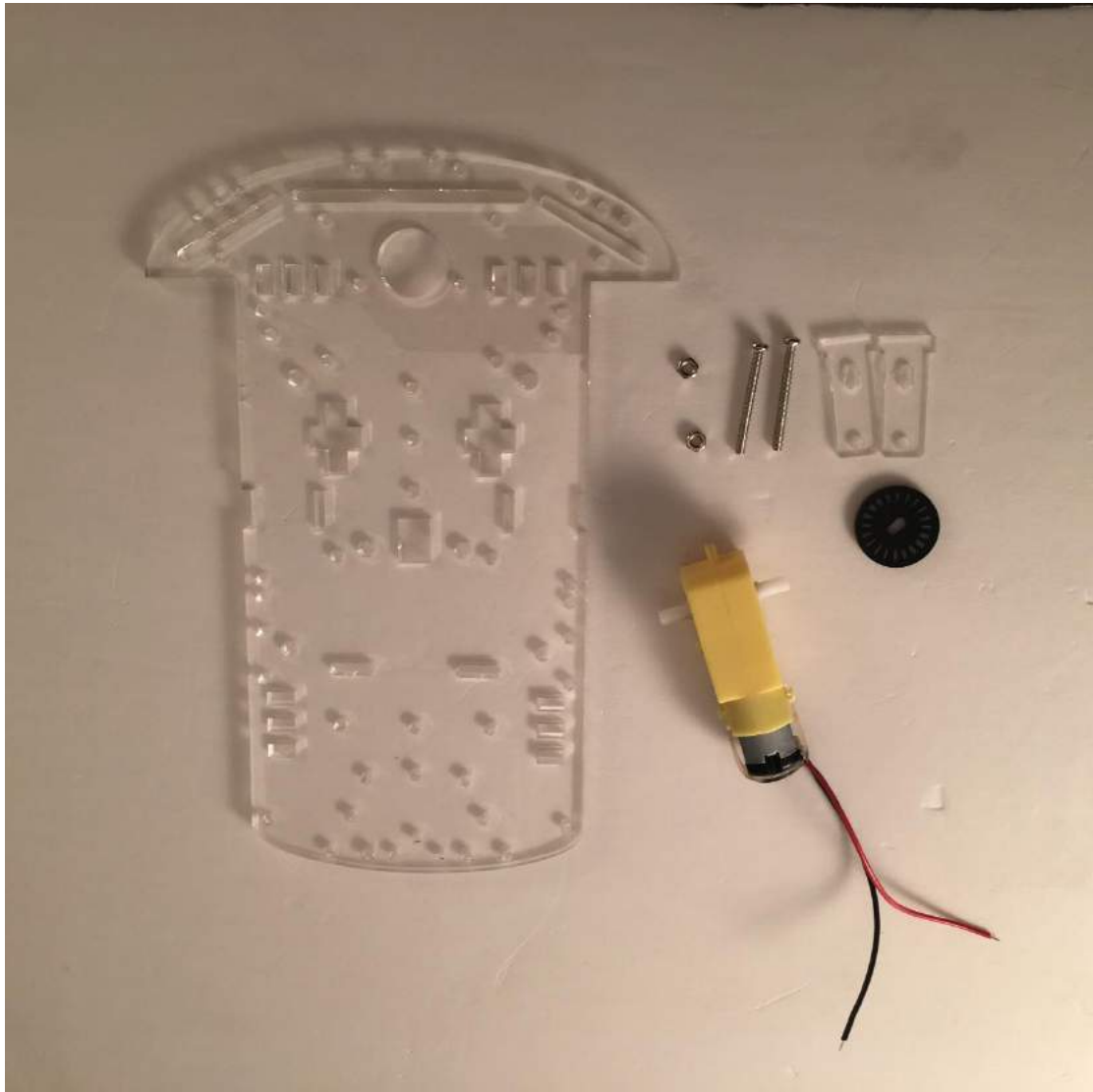
The FPV and Servo supplied are generic parts that may require some modification, as mentioned through out this guide.

---

## Installation steps

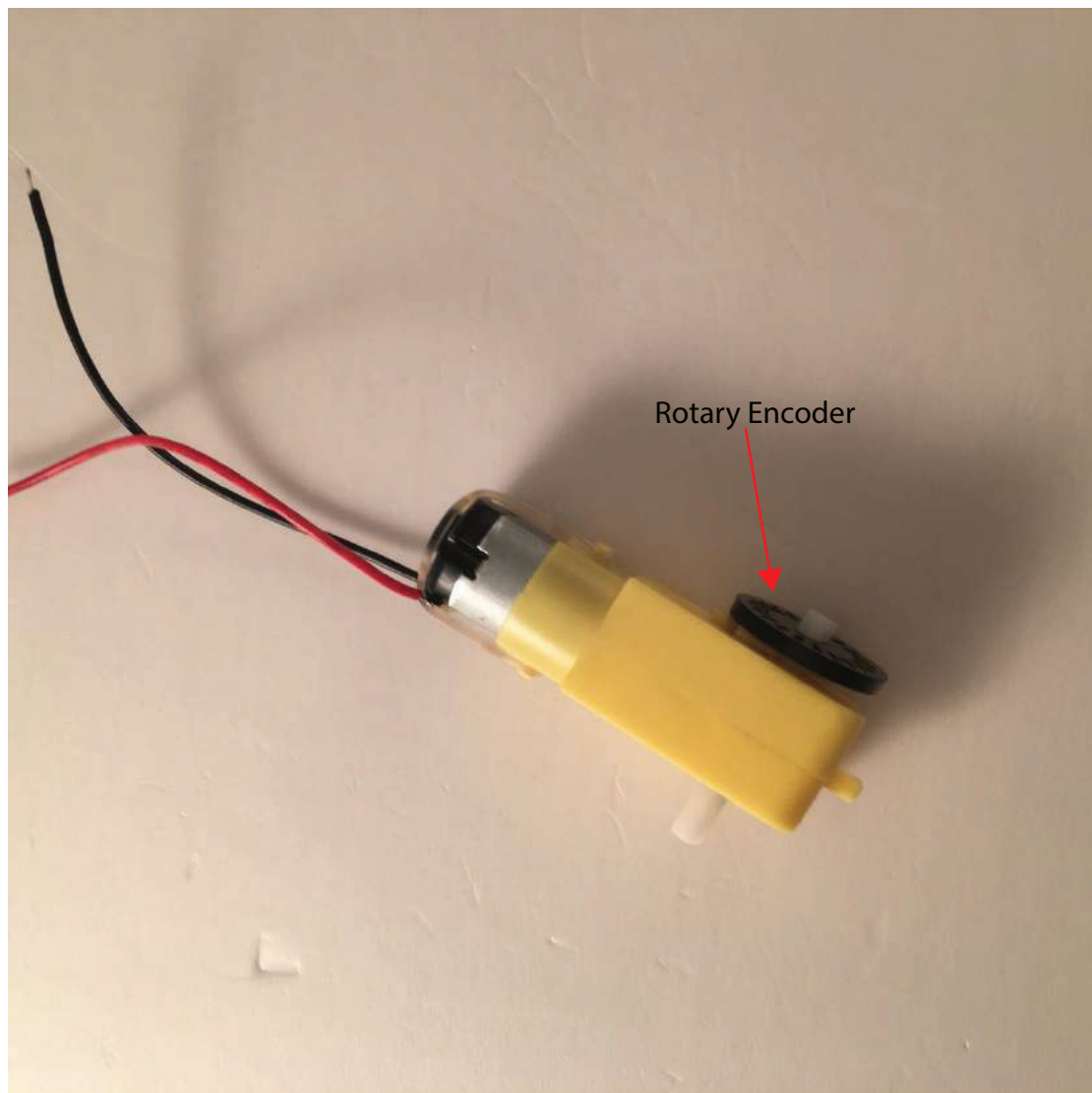
Step 1: Create the car chassis

Parts for Step 1



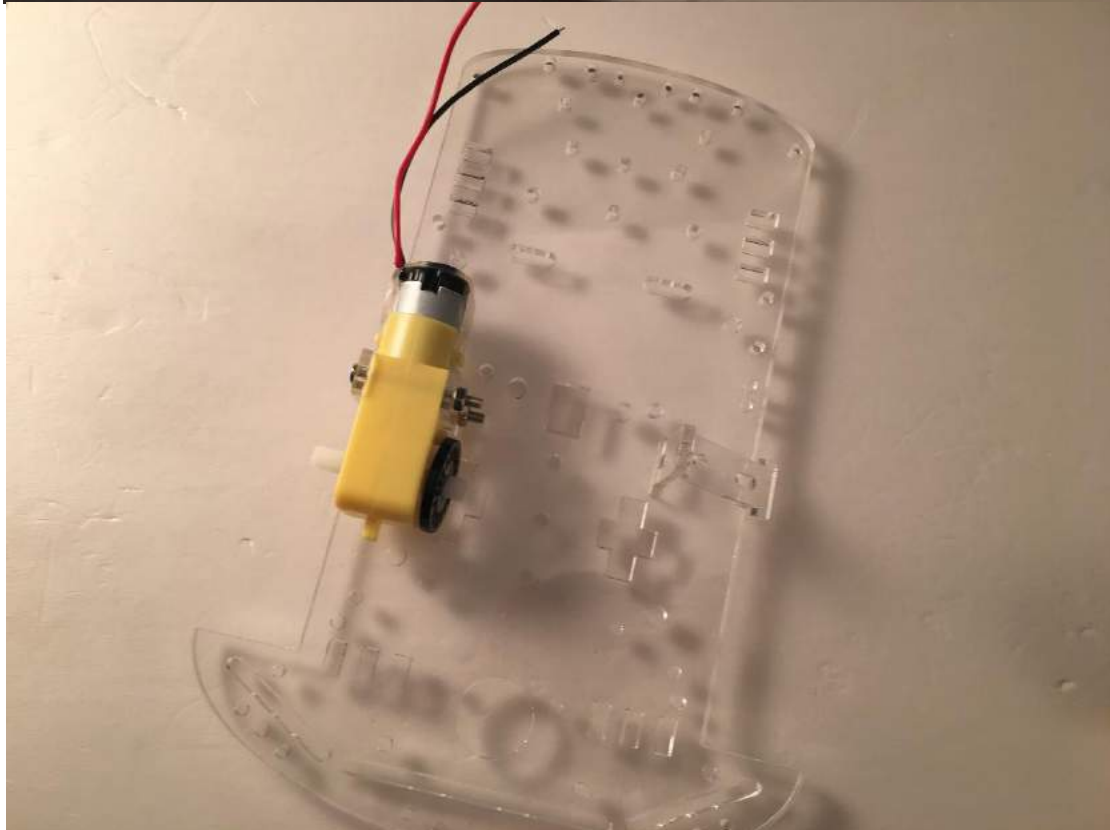
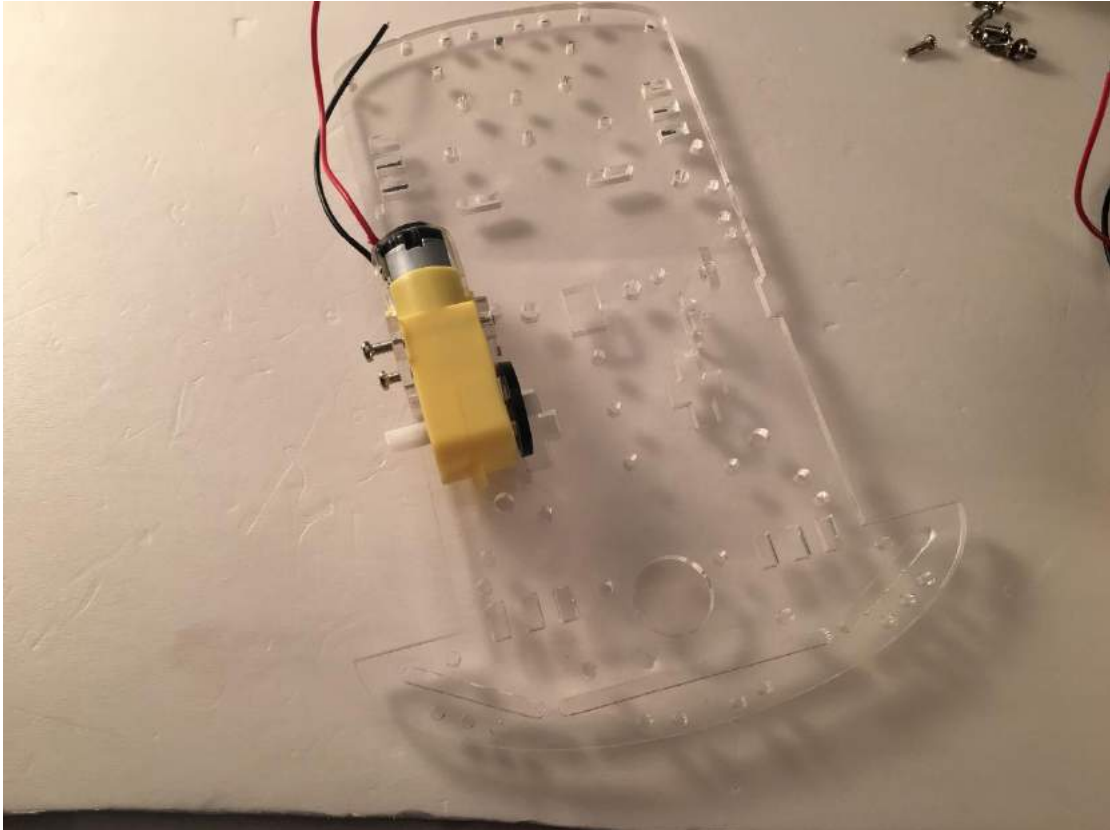
---

First, plug the rotary encoder in to the motor.



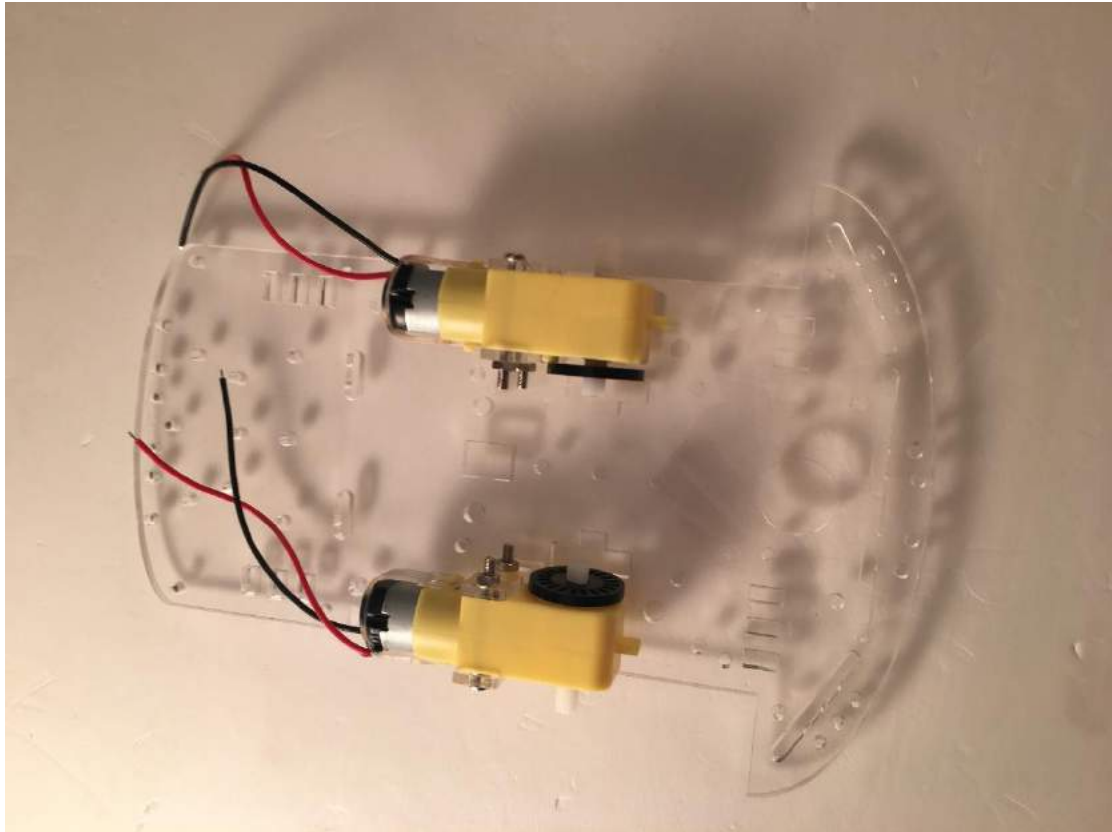
---

Second, use 2 fasteners to fix one motor on the bottom board. Please note which hole should be used and the motor's direction should be forward.

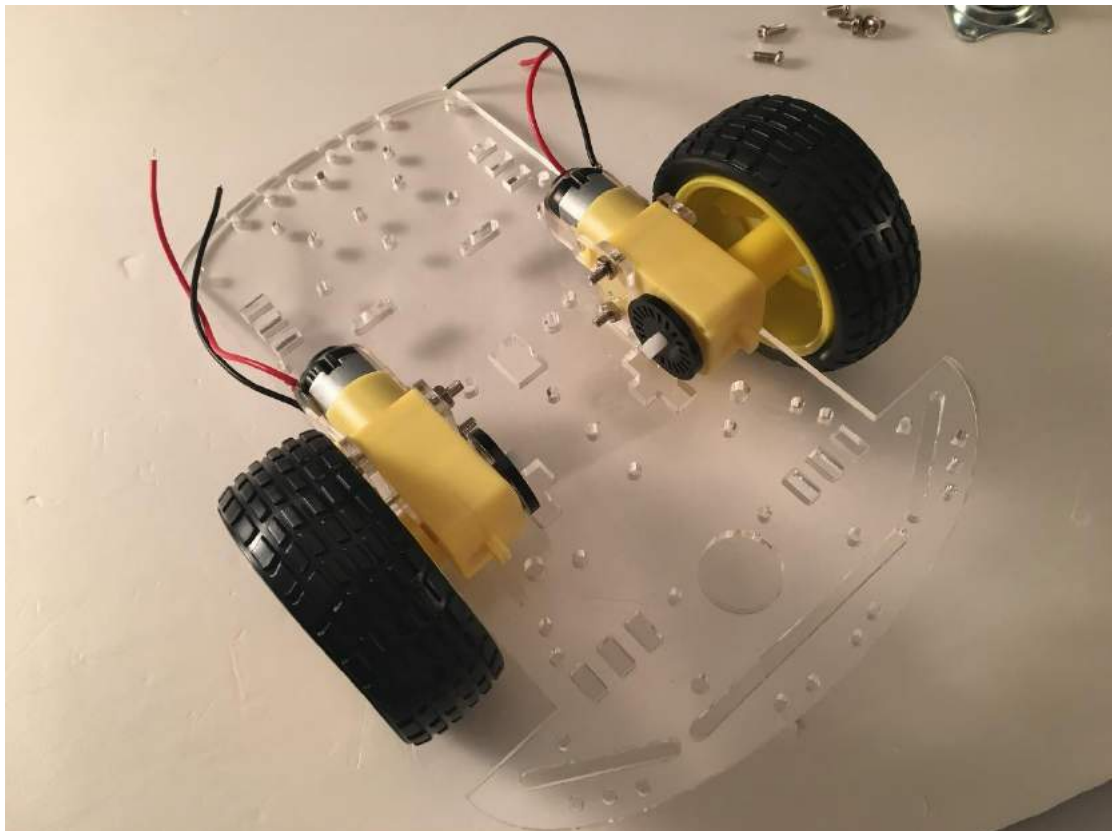


---

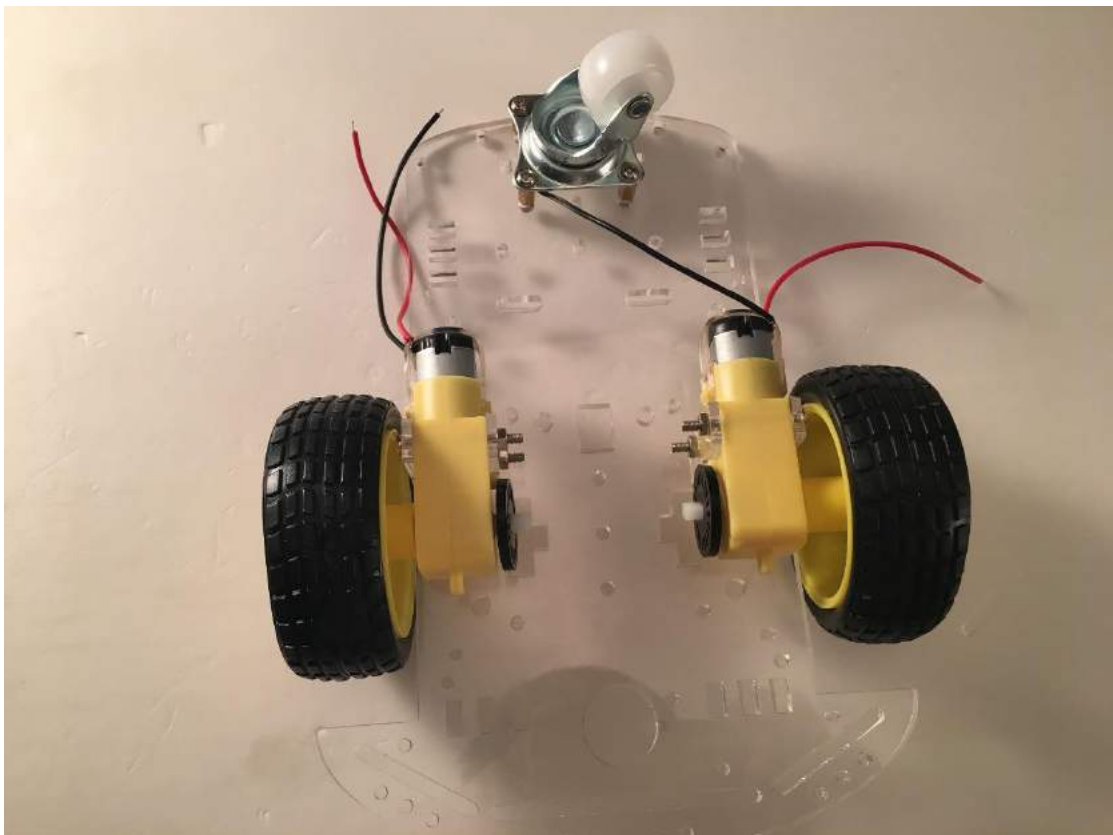
Then proceed to install the second motor using the same steps as the first motor



Plug the rubber wheel in the motor, please note the angle of the socket on the wheel.



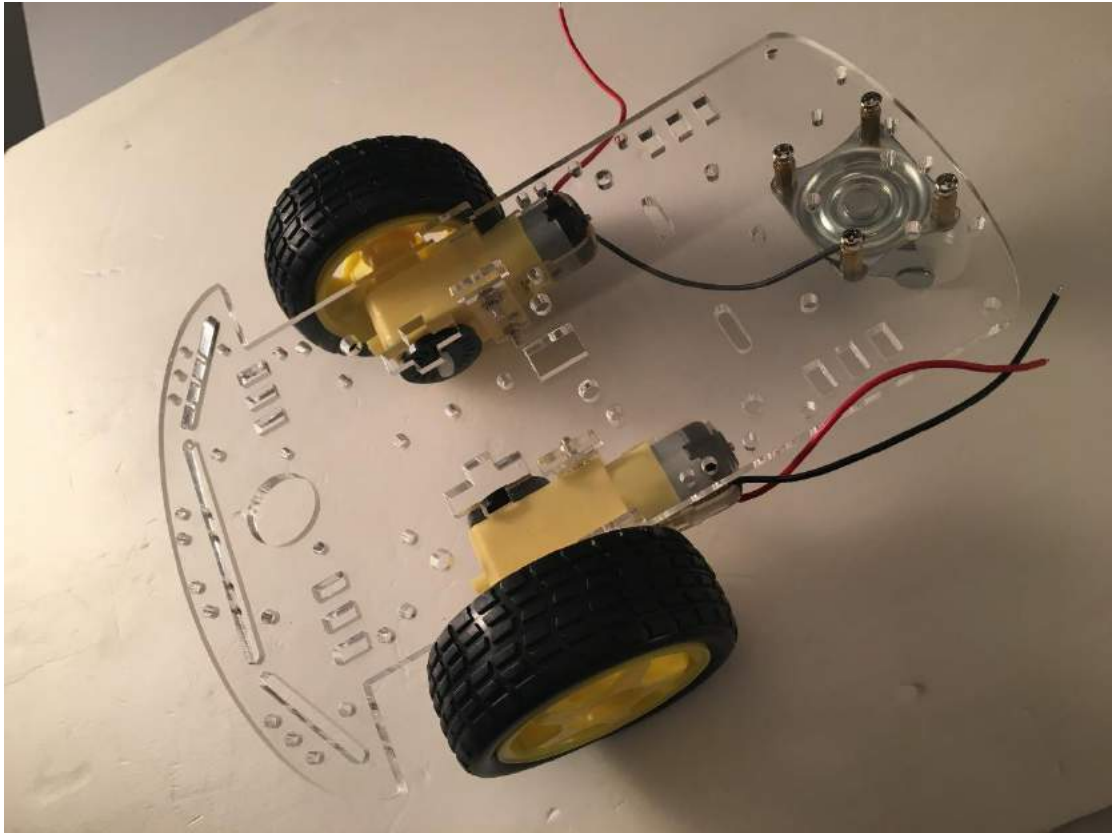
Then, install the universal wheel, please note which hole should be used on the board.





---

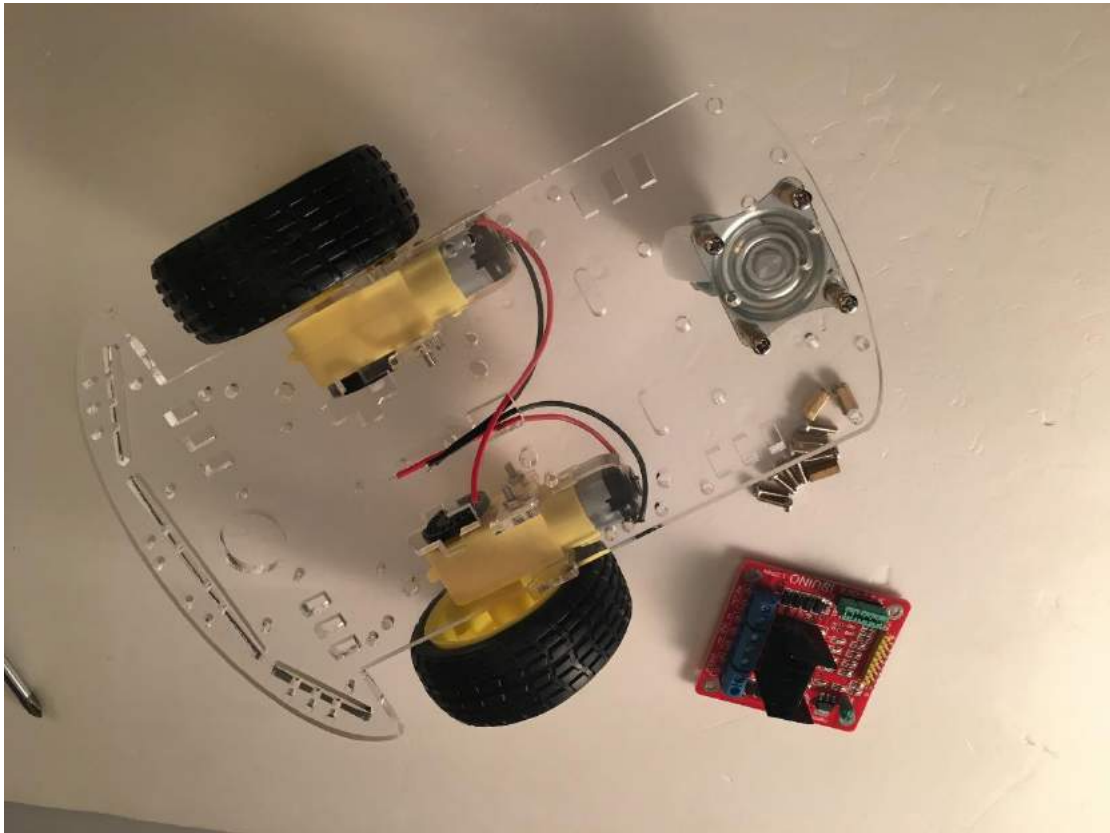
The first step should be finished and should look like the following picture:



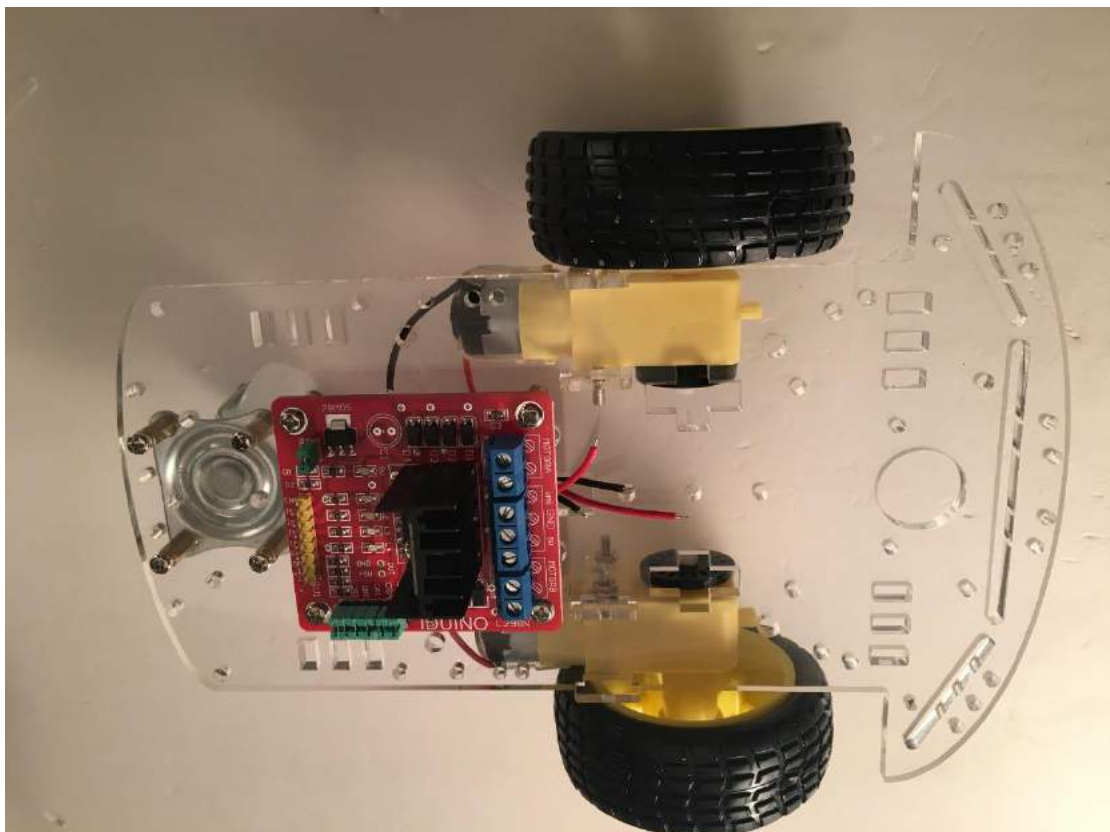
---

## Step2: Fix the main components on the car chassis

First, Fix the L298N driver board.

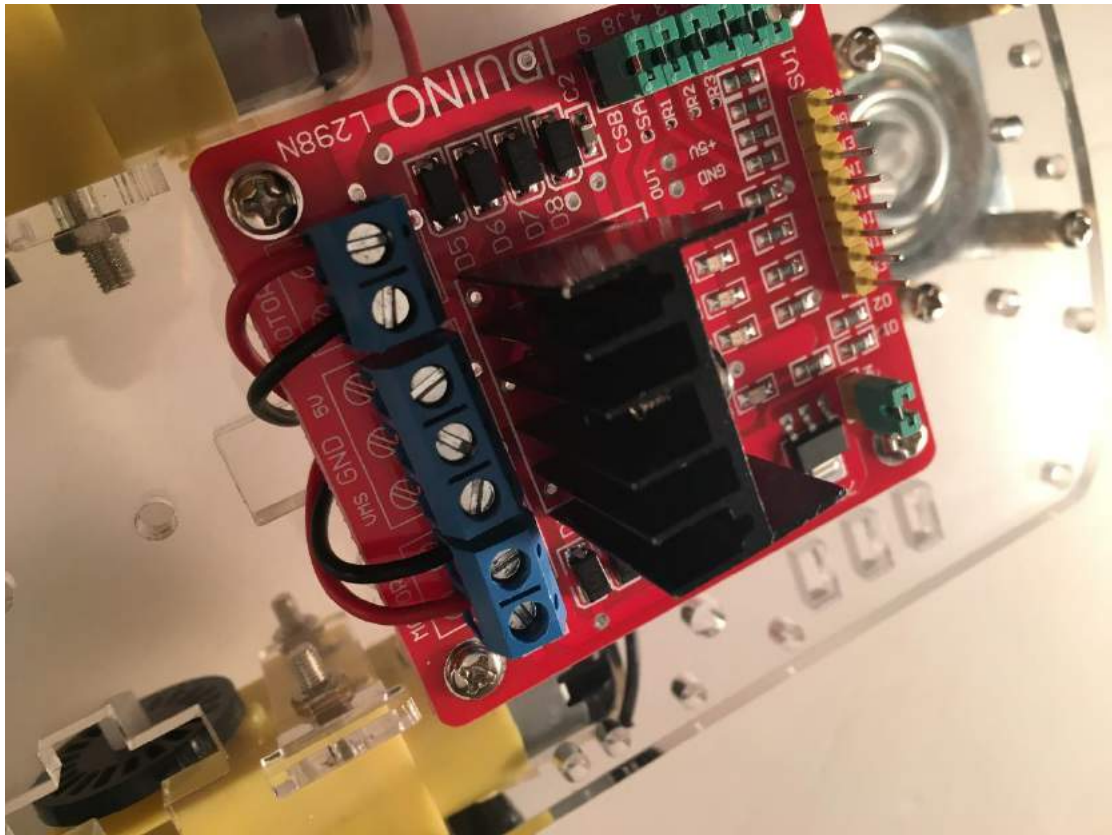


Before you fix the screw, passing the wire on the motor through the board helps your next step.



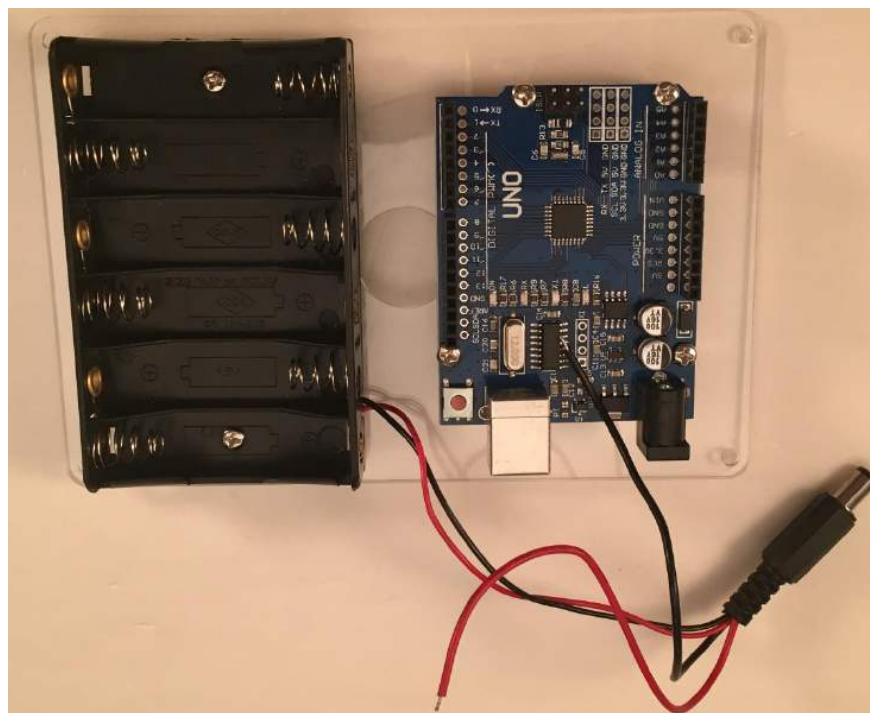
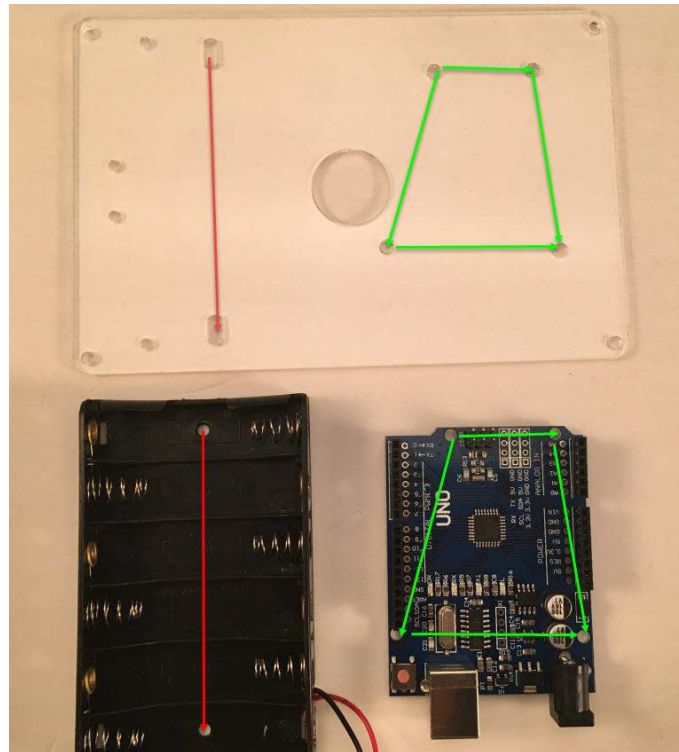
---

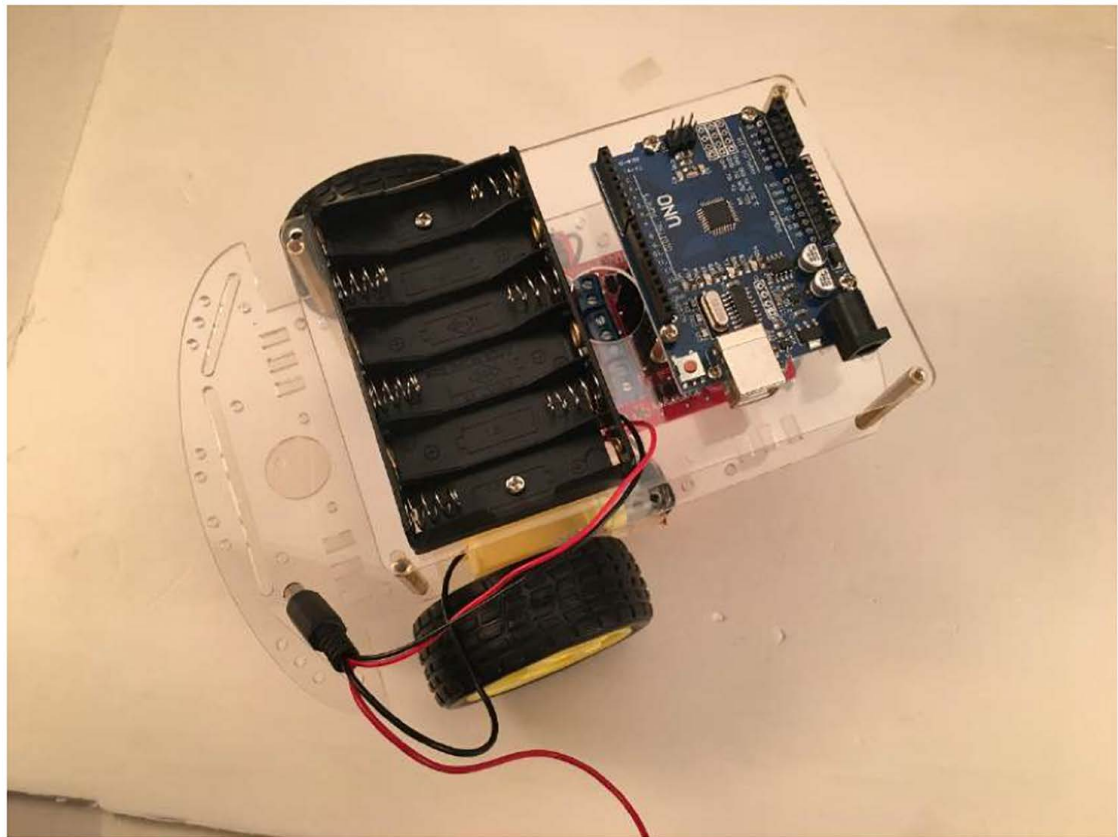
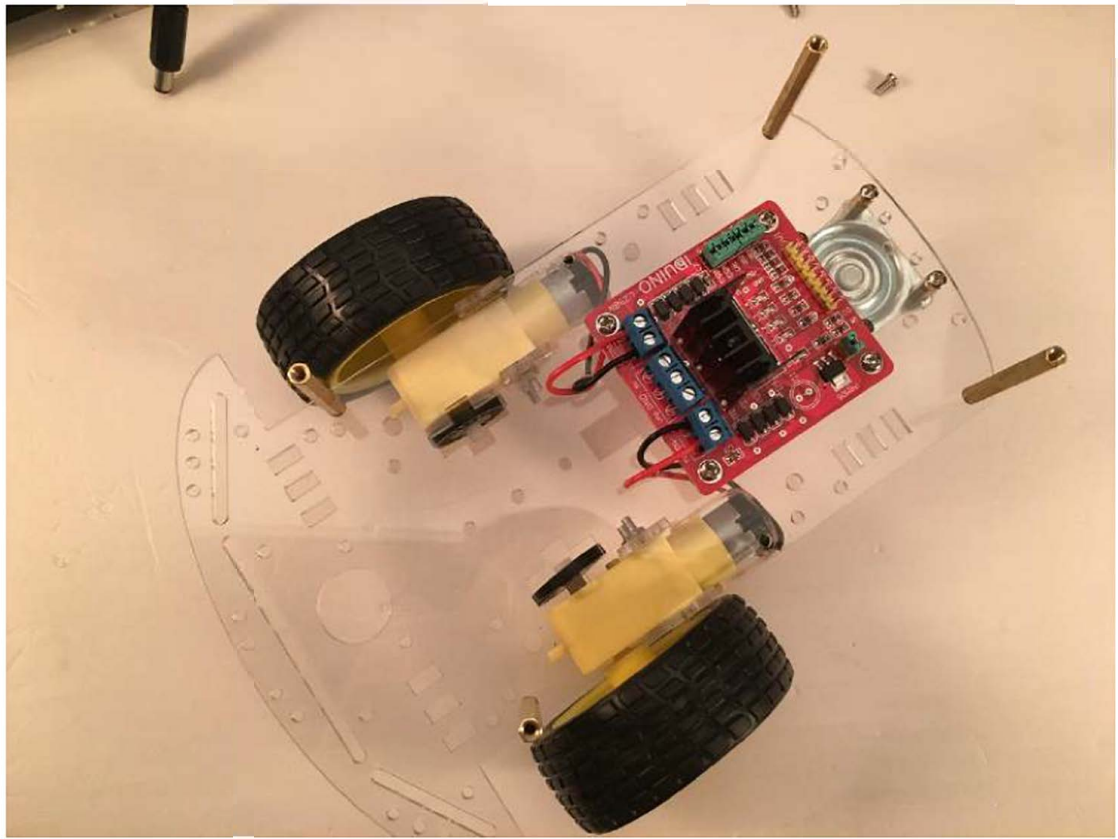
Then connect the L298N module with the motor, please note which wire should be fixed on which screw, the red wire is VCC and the black wire is GND.



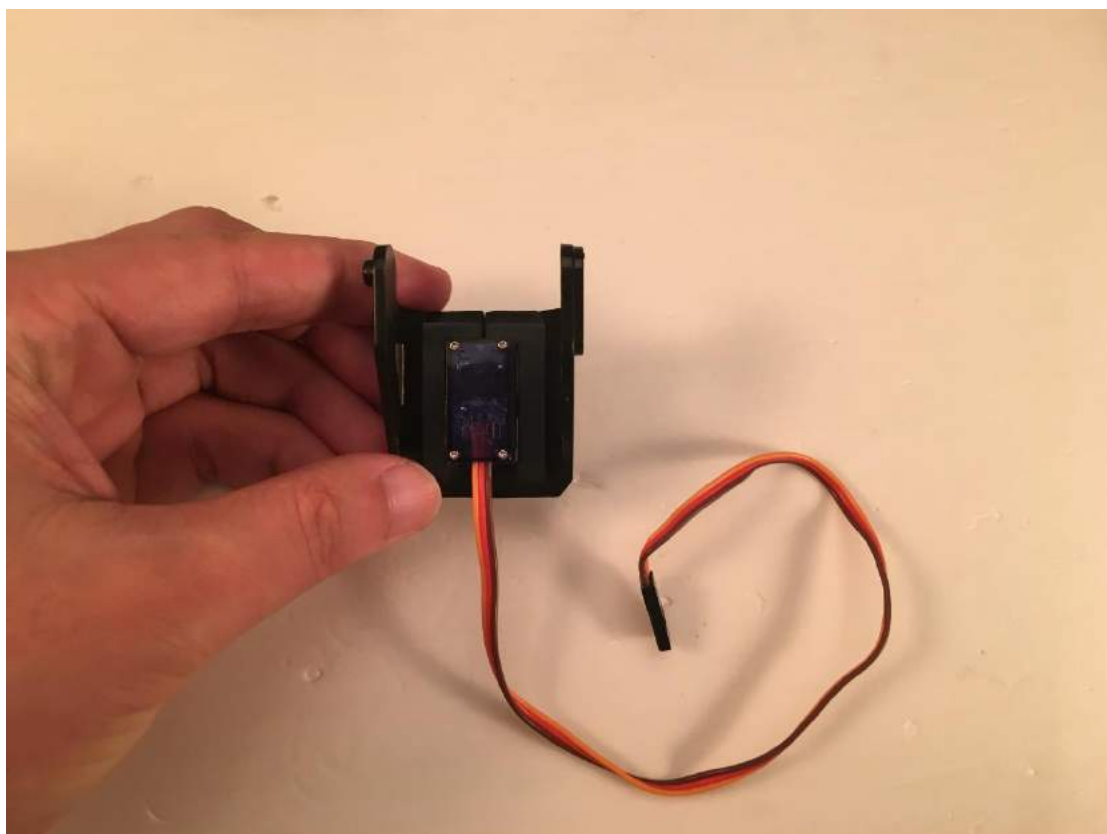
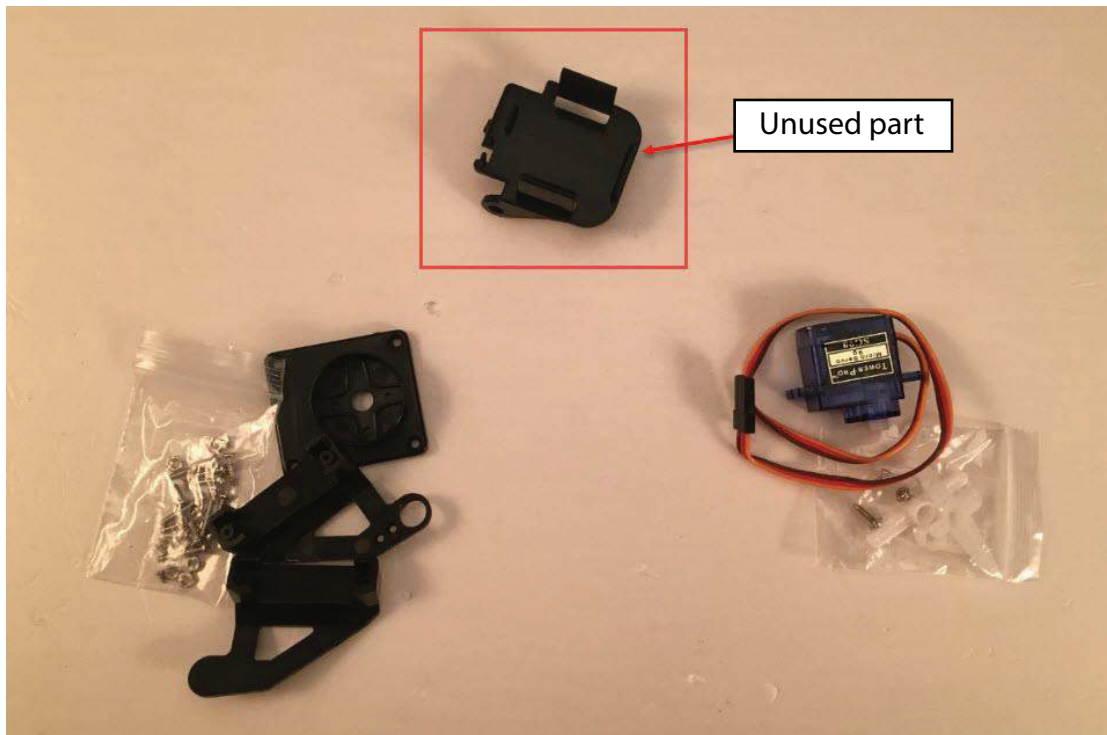
We have other three parts need to be fixed, the UNO board, sensor extension board and ultrasonic module. If we put these part also on the one acrylic board used on the above step, the space may be crowded, which makes the appearance of the ultrasonic car not beautiful. So in the next step, we can use another acrylic board. We can fix the UNO and battery holder on the acrylic board first, then fix the new board on the bottom board.

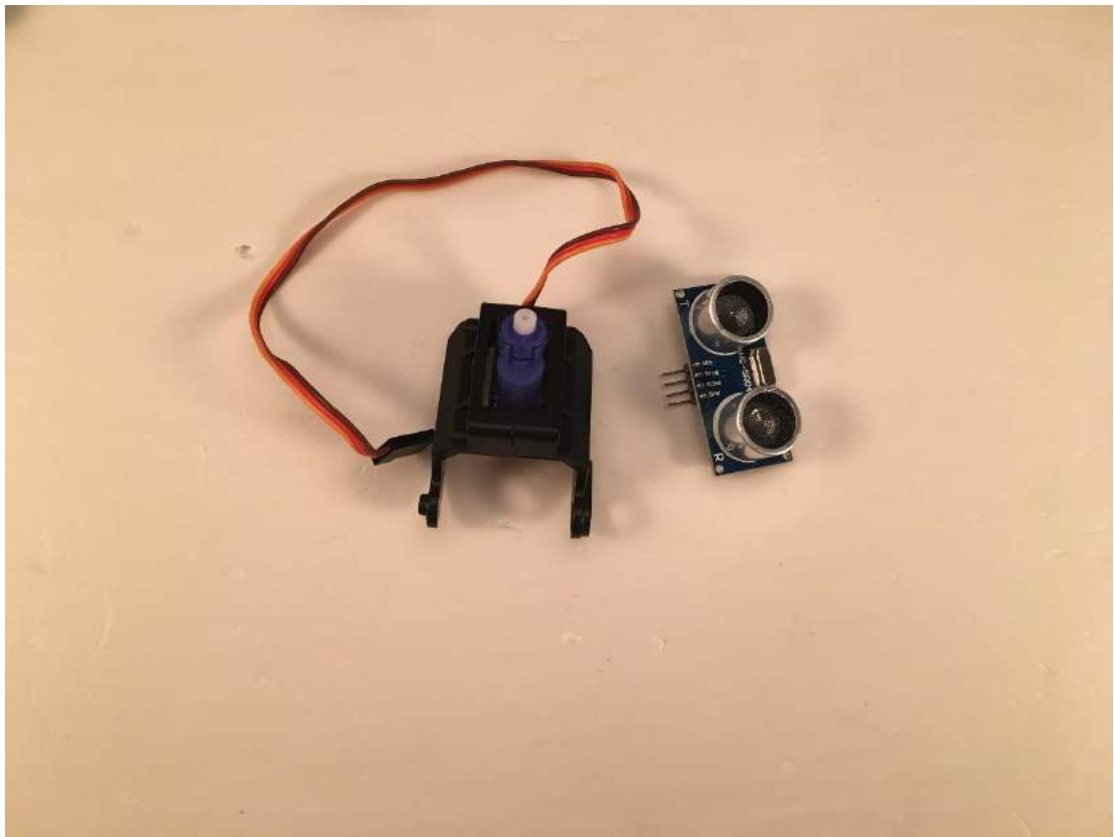
Please note the needed fix hole.





Then fix the ultrasonic module. In this step, we need a middleware to connect the car with the ultrasonic module. The middleware actually a FPV holder, like the following picture:





You need use two ribbon to fix this module.



---

In order to fit the holder, we need cut the rubber holder cross as the same length on four direction.

Cut this (use a shear or other tools)



To this



This step need a sharp knife, you must be careful. Then put this cross in the holder like following:





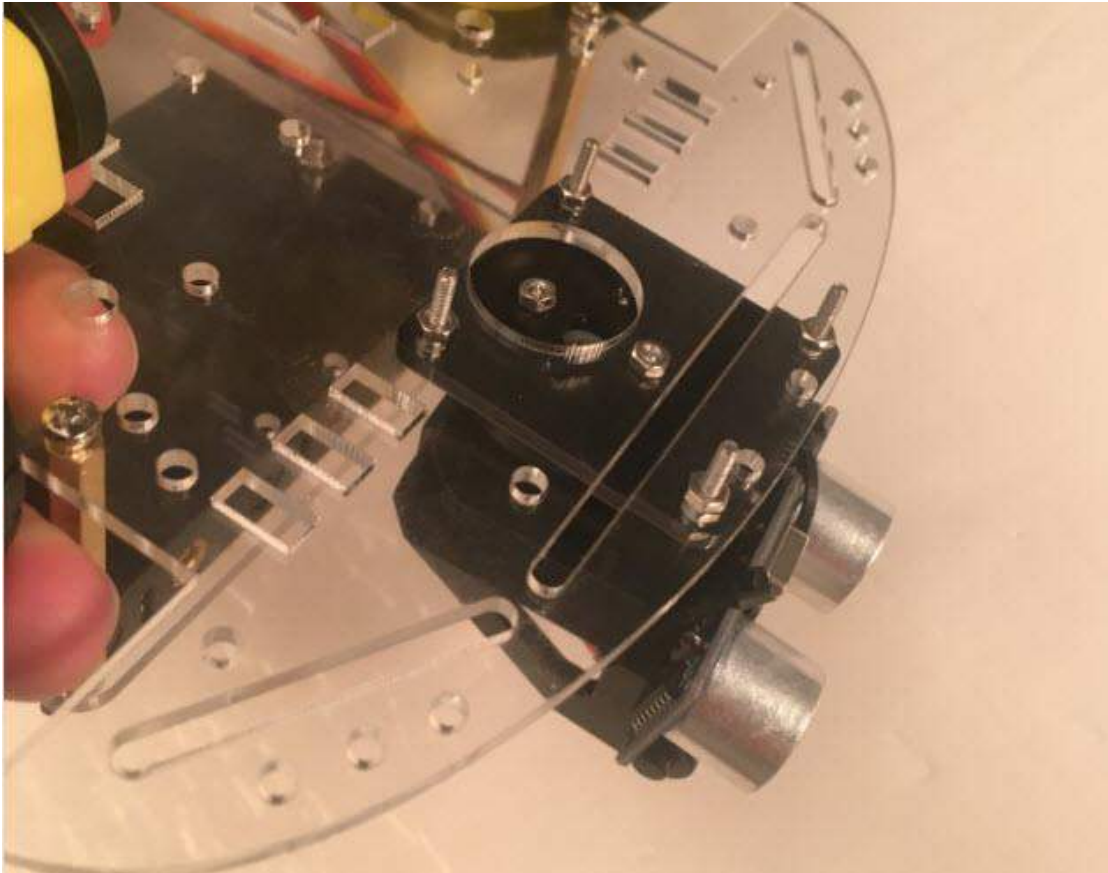


**Note:** when you fix servo holder on it's base(black plastic of the right on above the picture), you need read the actual code to adjust your servo on the most correct angle.

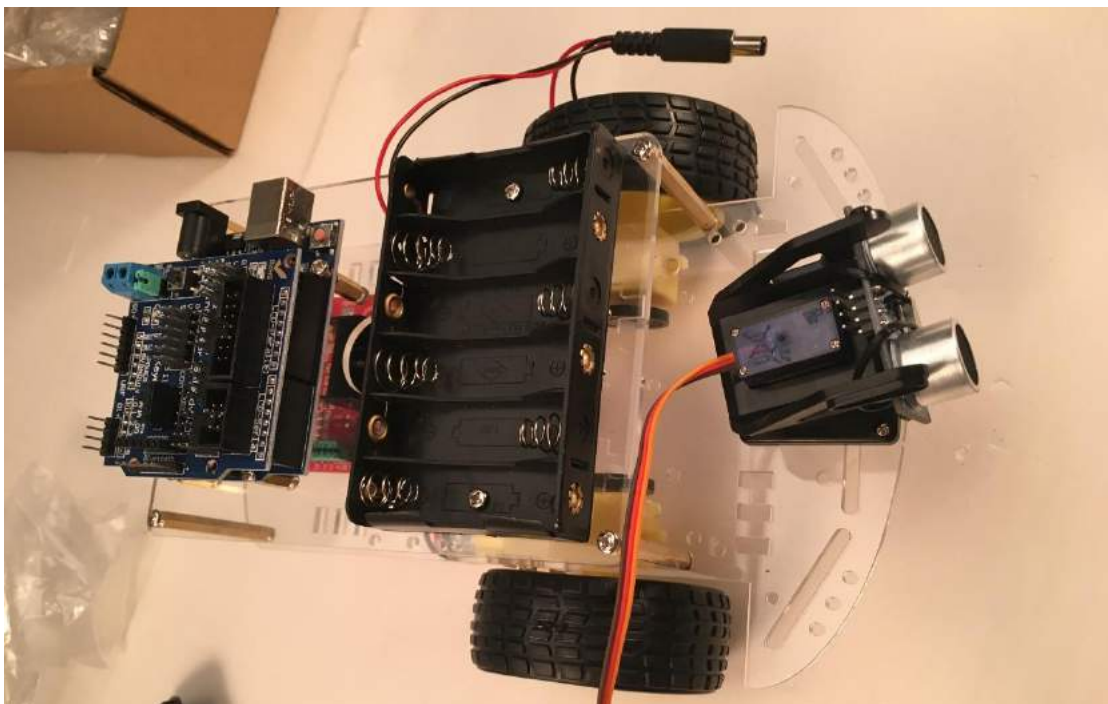


---

And fix the sevro holder on the bottom board:

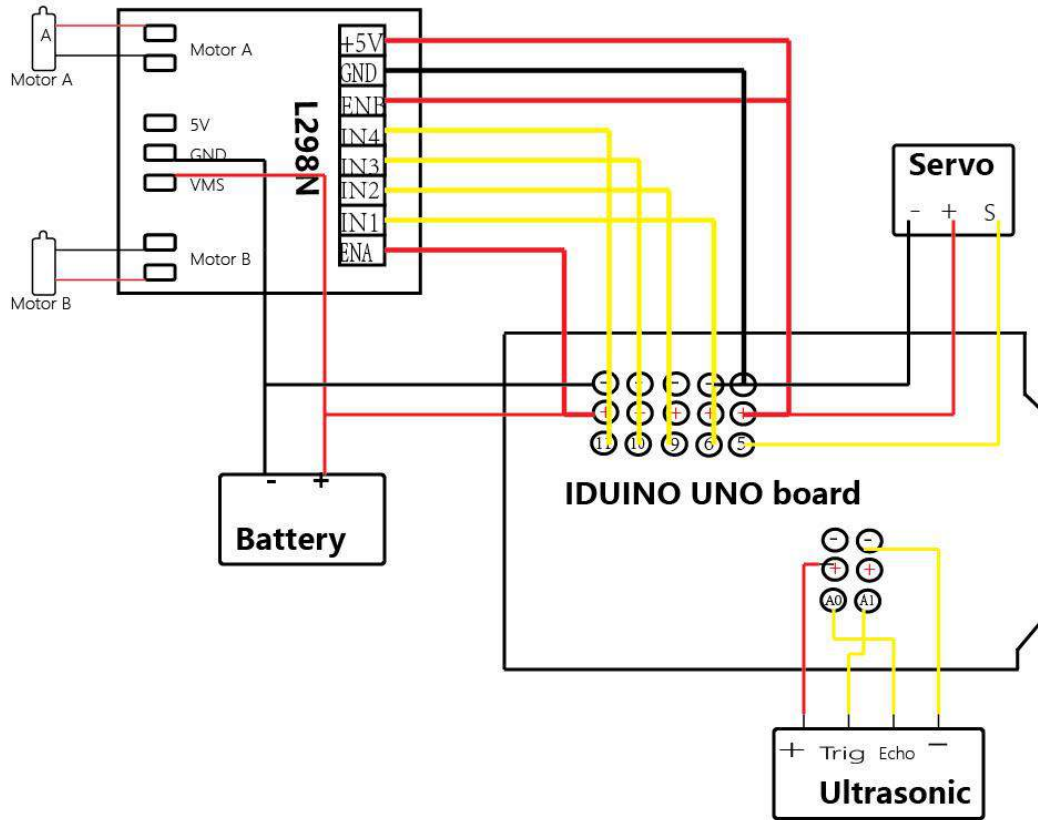


When finish that, the assemble for this DIY ultrasonic car is finished. Next step we need wire connection and write sketch.

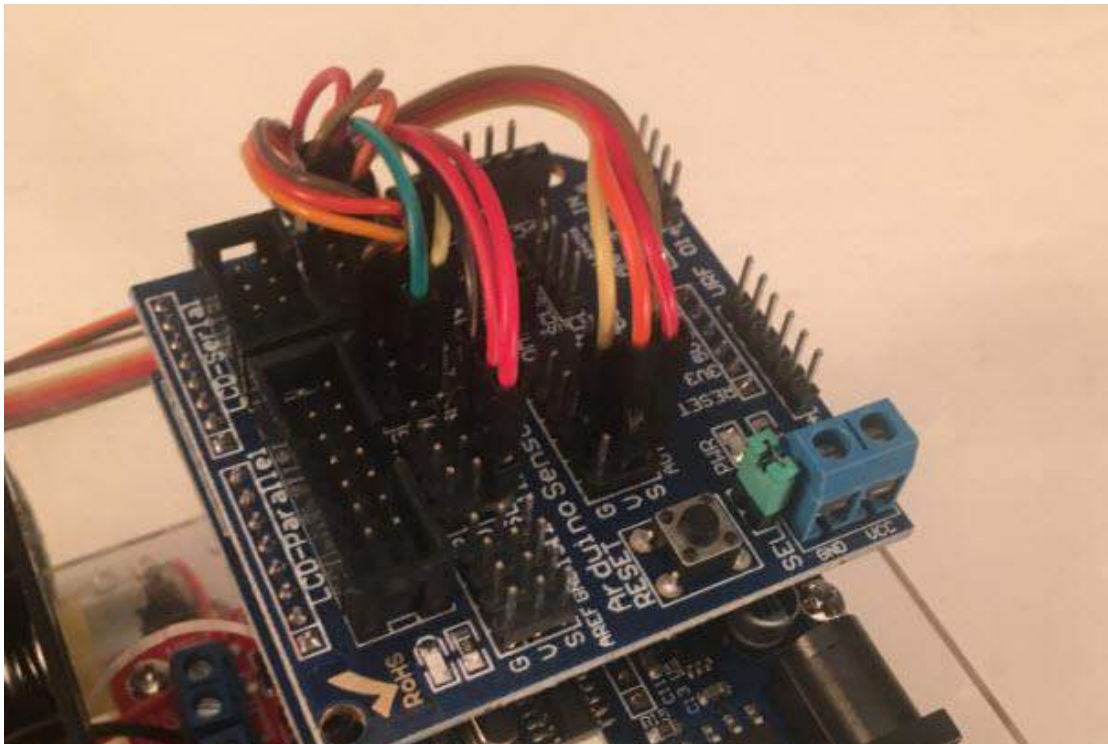


### Step 3: Wire connection and upload code.

In this step, we need connect wire as the following picture.



After the connection, upload the following code to your UNO board. Then power on the uno board and L298N module, the ultrasonic car starts moving.



Then you need adjust the servo's connection with its base to find the perfect angle.

---

\*\*\*\*\*Code begin\*\*\*\*\*

```
#include <Servo.h>

int pinLB=6;    // define pin6 as left back connect with IN1
int pinLF=9;    // define pin9 as left forward connect with IN2

int pinRB=10;   // define pin10 as right back connect with IN3
int pinRF=11;   // define pin11 as right back connect with IN4

int inputPin = A0; // define ultrasonic receive pin (Echo)
int outputPin =A1; // define ultrasonic send pin(Trig)

int Fspeedd = 0;    // forward distance
int Rspeedd = 0;    // right distance
int Lspeedd = 0;    // left distance
int directionn = 0; //
Servo myservo;     // new myservo
int delay_time = 250; // set stable time

int Fgo = 8;       // forward
int Rgo = 6;       // turn right
int Lgo = 4;       // turn left
int Bgo = 2;       // back

void setup()
{
  Serial.begin(9600);
  pinMode(pinLB,OUTPUT);
  pinMode(pinLF,OUTPUT);
  pinMode(pinRB,OUTPUT);
  pinMode(pinRF,OUTPUT);

  pinMode(inputPin, INPUT);
  pinMode(outputPin, OUTPUT);
}
```

---

```
myservo.attach(5);    // define the servo pin(PWM)
}
void advance(int a)    // forward
{
    digitalWrite(pinRB,LOW);
    digitalWrite(pinRF,HIGH);
    digitalWrite(pinLB,LOW);
    digitalWrite(pinLF,HIGH);
    delay(a * 40);
}

void turnR(int d)      //turn right
{
    digitalWrite(pinRB,LOW);
    digitalWrite(pinRF,HIGH);
    digitalWrite(pinLB,HIGH);
    digitalWrite(pinLF,LOW);
    delay(d * 50);
}
void turnL(int e)     //turn left
{
    digitalWrite(pinRB,HIGH);
    digitalWrite(pinRF,LOW);
    digitalWrite(pinLB,LOW);
    digitalWrite(pinLF,HIGH);
    delay(e * 50);
}
void stopp(int f)     //stop
{
    digitalWrite(pinRB,HIGH);
    digitalWrite(pinRF,HIGH);
    digitalWrite(pinLB,HIGH);
    digitalWrite(pinLF,HIGH);
    delay(f * 100);
}
void back(int g)      //back
{
    digitalWrite(pinRB,HIGH);
    digitalWrite(pinRF,LOW);
    digitalWrite(pinLB,HIGH);
    digitalWrite(pinLF,LOW);
    delay(g * 300);
}
```

---

```

void detection()          //test the distance of different direction
{
    int delay_time = 250; //
    ask_pin_F();         // read forward distance

    if(Fspeedd < 10)     // if distance less than 10
    {
        stopp(1);
        back(2);
    }

    if(Fspeedd < 25)     // if distance less than 10
    {
        stopp(1);
        ask_pin_L();
        delay(delay_time);
        ask_pin_R();
        delay(delay_time);

        if(Lspeedd > Rspeedd) //if left distance more than right distance
        {
            directionn = Rgo;
        }
        if(Lspeedd <= Rspeedd)//if left distance not more than right
//distance
        {
            directionn = Lgo;
        }

        if (Lspeedd < 10 && Rspeedd < 10) //if left distance and right
//distance both less than 10
        {
            directionn = Bgo;
        }
    }
    else
    {
        directionn = Fgo;         // forward go
    }
}

```

---

```
    }
void ask_pin_F() // test forward distance
{
    myservo.write(90);
    digitalWrite(outputPin, LOW);
    delayMicroseconds(2);
    digitalWrite(outputPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(outputPin, LOW);
    float Fdistance = pulseIn(inputPin, HIGH);
    Fdistance= Fdistance/5.8/10;
    Serial.print("F distance:");
    Serial.println(Fdistance);
    Fspeedd = Fdistance;
}
void ask_pin_L() // test left distance
{
    myservo.write(5);
    delay(delay_time);
    digitalWrite(outputPin, LOW);
    delayMicroseconds(2);
    digitalWrite(outputPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(outputPin, LOW);
    float Ldistance = pulseIn(inputPin, HIGH);
    Ldistance= Ldistance/5.8/10;
    Serial.print("L distance:");
    Serial.println(Ldistance);
    Lspeedd = Ldistance;
}
void ask_pin_R() // test right distance
{
    myservo.write(177);
    delay(delay_time);
    digitalWrite(outputPin, LOW);
    delayMicroseconds(2);
    digitalWrite(outputPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(outputPin, LOW);
    float Rdistance = pulseIn(inputPin, HIGH);
    Rdistance= Rdistance/5.8/10;
    Serial.print("R distance:");
    Serial.println(Rdistance);
    Rspeedd = Rdistance;
}
```

---

```
void loop()
{
  myservo.write(90);
  detection();

  if(directionn == 2)
  {
    back(8);
    turnL(2);
    Serial.print(" Reverse ");
  }
  if(directionn == 6)
  {
    back(1);
    turnR(6);
    Serial.print(" Right ");
  }
  if(directionn == 4)
  {
    back(1);
    turnL(6);
    Serial.print(" Left ");
  }
  if(directionn == 8)
  {
    advance(1);
    Serial.print(" Advance ");
    Serial.print(" ");
  }
}
}
*****Code End*****
```

Once coded, the DIY ultrasonic Arduino car finish. This car kit offer you a learning platform. Based on this car, you can adapt other modules (Bluetooth, infrared remoter and wifi video module), to make a smarter car.