10 Ultrasonic obstacle avoidance (no servo)

The purpose of the experiment:

Use the carton to simulate obstacles on the ground, put the smart car that has uploaded the program **avoid.ino** on the ground, press the start button of the tail of the car, the LCD screen of the car shows the distance measured by the ultrasonic sensor and starts to avoid the obstacle ahead.

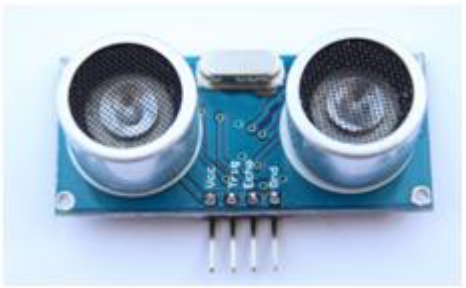
Precautions:

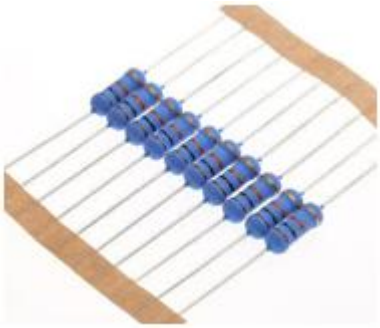
1. If the LCD is not displayed, use a screwdriver to adjust the adjustable resistor.

2. If only the ultrasonic obstacle avoidance function is used, the display distance is not required, and the LCD1602 display and the yellow adjustable resistor are not installed.

List of components required for the experiment:

Arduino Smart Car* 1
USB data cable* 1
DuPont line * n
Breadboard* 1
1602 LCD screen* 1
Adjustable resistance* 1
Active buzzer* 1
Button * 1
Ultrasonic sensor*1
10K resistor * 1





Experimental code analysis:

```
//============================yahboom=========================
//Intelligent car ultrasonic obstacle avoidance(no servo)
//============================================================
//#include <Servo.h>
#include <LiquidCrystal.h> //Declare the function library of 1602 liquid crystals
//Declare the Arduino digital port connected by the pin of the 1602 LCD,
//8 or 4 line data modes, choose one of them.
//LiquidCrystal lcd(12,11,10,9,8,7,6,5,4,3,2);    //8 data port mode connection statement
LiquidCrystal lcd(3,4,7,8,11,12,13); //4 data port mode connection statement
```

```
int Echo = A5;   // Echo(P2.0)
int Trig =A4;   //   Trig(P2.1)
int Distance = 0;
int Left_motor_back=9;      //(IN1)
int Left_motor_go=5;        //(IN2)
int Right_motor_go=6;       //(IN3)
int Right_motor_back=10;    //(IN4)
int key=A0;//Define the key A0 interface
int beep=A1;//Define the buzzer A1 interface
void setup()
{
  Serial.begin(9600);      //Initialize the serial port
  //Initialize the motor drive IO for output mode
  pinMode(Left_motor_go,OUTPUT); // PIN 5 (PWM)
  pinMode(Left_motor_back,OUTPUT); // PIN 9 (PWM)
  pinMode(Right_motor_go,OUTPUT);// PIN 6 (PWM)
  pinMode(Right_motor_back,OUTPUT);// PIN 10 (PWM)
  pinMode(key,INPUT);//Define the key interface for the input interface
  pinMode(beep,OUTPUT);
  //Initialization of ultrasonic pins
  pinMode(Echo, INPUT);     // Define of ultrasonic input pin
  pinMode(Trig, OUTPUT);    // Define of ultrasonic output pin
  lcd.begin(16,2);       //Initialization of 1602 liquid crystal working mode
  //Define the 1602 LCD display range of 2 lines and 16 columns
}
//======================The basic action of
car=========================
//void run(int time)
void run()
{
  digitalWrite(Right_motor_go,HIGH);   //right motor go
  digitalWrite(Right_motor_back,LOW);
  analogWrite(Right_motor_go,100);//PWM ratio 0~255 speed control,
                          //the difference of left and right wheel slightly increase or
decrease
  analogWrite(Right_motor_back,0);
```

```
  digitalWrite(Left_motor_go,HIGH);   // left motor go
  digitalWrite(Left_motor_back,LOW);
  analogWrite(Left_motor_go,100);//PWM ratio 0~255 speed control,
                          //the difference of left and right wheel slightly increase or
decrease
  analogWrite(Left_motor_back,0);
  //delay(time * 100);    //execution time, can be adjusted
}
void brake(int time)
{
  digitalWrite(Right_motor_go,LOW);
  digitalWrite(Right_motor_back,LOW);
  digitalWrite(Left_motor_go,LOW);
  digitalWrite(Left_motor_back,LOW);
  delay(time * 100);//execution time, can be adjusted
}
//void left(int time)
void left()         //turn left(left wheel stop,right wheel go)
{
  digitalWrite(Right_motor_go,HIGH); // right motor go
  digitalWrite(Right_motor_back,LOW);
  analogWrite(Right_motor_go,100);
  analogWrite(Right_motor_back,0);    //PWM ratio 0~255 speed control
  digitalWrite(Left_motor_go,LOW);
  digitalWrite(Left_motor_back,LOW);
  analogWrite(Left_motor_go,0);
  analogWrite(Left_motor_back,0);    //PWM ratio 0~255 speed control
  //delay(time * 100);       //execution time, can be adjusted
}
void spin_left(int time)         //left rotation(left wheel back，right wheel go)
{
  digitalWrite(Right_motor_go,HIGH);   //right motor go
  digitalWrite(Right_motor_back,LOW);
  analogWrite(Right_motor_go,100);
  analogWrite(Right_motor_back,0);     //PWM ratio 0~255 speed control
  digitalWrite(Left_motor_go,LOW);     //left motor back
```

```
  digitalWrite(Left_motor_back,HIGH);
  analogWrite(Left_motor_go,0);
  analogWrite(Left_motor_back,100);   //PWM ratio 0~255 speed control
  delay(time * 100);    //execution time, can be adjusted
}
void right(int time)
//void right()        //turn right (right wheel stop,left wheel go)
{
  digitalWrite(Right_motor_go,LOW);
  digitalWrite(Right_motor_back,LOW);
  analogWrite(Right_motor_go,0);
  analogWrite(Right_motor_back,0); //PWM ratio 0~255 speed control
  digitalWrite(Left_motor_go,HIGH);//left motor go
  digitalWrite(Left_motor_back,LOW);
  analogWrite(Left_motor_go,100);
  analogWrite(Left_motor_back,0);   //PWM ratio 0~255 speed control
  delay(time * 100);    //execution time, can be adjusted
}
void spin_right(int time)        //right rotation(right wheel back,left wheel go)
{
  digitalWrite(Right_motor_go,LOW);
  digitalWrite(Right_motor_back,HIGH); //right motor back
  analogWrite(Right_motor_go,0);
  analogWrite(Right_motor_back,100); //PWM ratio 0~255 speed control
  digitalWrite(Left_motor_go,HIGH);   //left motor go
  digitalWrite(Left_motor_back,LOW);
  analogWrite(Left_motor_go,100);
  analogWrite(Left_motor_back,0);    //PWM ratio 0~255 speed control
  delay(time * 100);    //execution time, can be adjusted
}
void back(int time)
{
  digitalWrite(Right_motor_go,LOW);   //right motor back
  digitalWrite(Right_motor_back,HIGH);
  analogWrite(Right_motor_go,0);
  analogWrite(Right_motor_back,100);   //PWM ratio 0~255 speed control
```

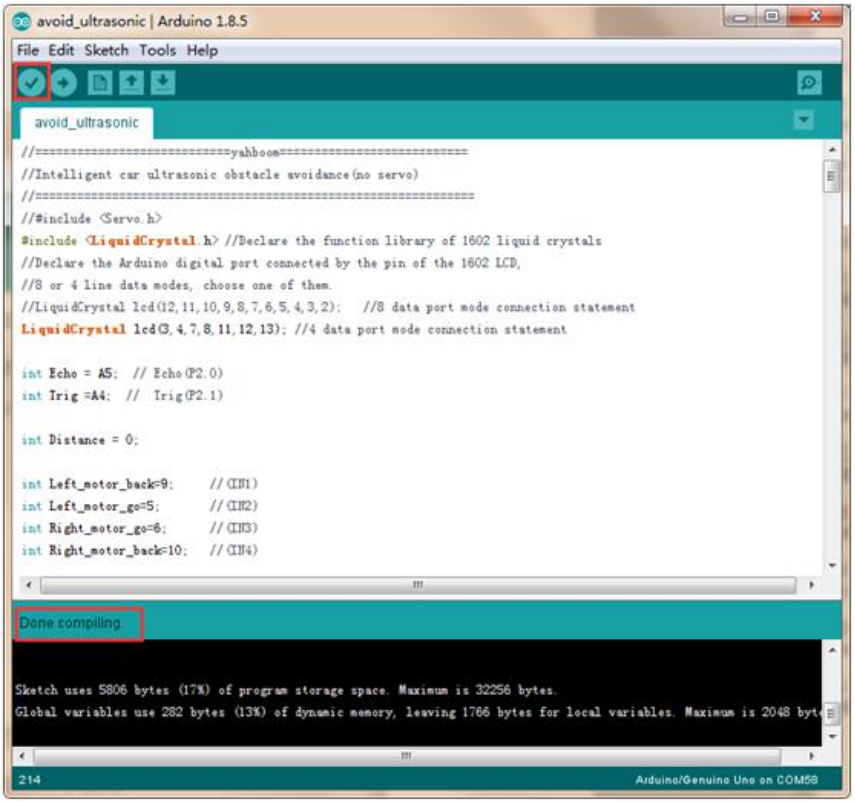
```
  digitalWrite(Left_motor_go,LOW);
  digitalWrite(Left_motor_back,HIGH); //left motor back
  analogWrite(Left_motor_go,0);
  analogWrite(Left_motor_back,150);    //PWM ratio 0~255 speed control
  delay(time * 100);      //execution time, can be adjusted
}
//==========================================================
void keysacn()
{
  int val;
  val=digitalRead(key);//Read the value of the port 7 level to the val
  while(!digitalRead(key))//When the key is not pressed, circulate all the time
  {
    val=digitalRead(key);//This sentence can be omitted and the circulate can
run away
  }
  while(digitalRead(key))//When the key is pressed
  {
    delay(10);
    val=digitalRead(key);//Read the value of the port 7 level to the val
    if(val==HIGH)   //Judge whether the key is pressed again
    {
      digitalWrite(beep,HIGH);         //buzzer sound
      while(!digitalRead(key))            //Judge whether the key isreleased
        digitalWrite(beep,LOW);             //buzzer no sound
    }
    else
      digitalWrite(beep,LOW);          //buzzer no sound
  }
}
void Distance_test()               //Measuring the distance ahead
{
  digitalWrite(Trig, LOW);           // Give the trigger pin low level 2us
  delayMicroseconds(2);
  digitalWrite(Trig, HIGH);          // Give the trigger pin high level 10us，at least
10μs
```

```
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);          //Give the trigger pin low level Continuouly
  float Fdistance = pulseIn(Echo, HIGH);  //Reading high level time(unit：us)
  Fdistance= Fdistance/58;          //  Y meter =（X second *344）/2
  // X second=（ 2*Y meter ）/344 ==》Xsecond =0.0058*Y meter ==》cm = us
/58
  Serial.print("Distance:");      //Output distance（unit：cm）
  Serial.println(Fdistance);      //display distance
  Distance = Fdistance;
}
void Distance_display()
{
   if((2<Distance)&(Distance<400))
  {
    lcd.home();         //Move the cursor back to the upper left corner,
                        //which is the beginning of the output
    lcd.print("    Distance: ");       //display
    lcd.setCursor(6,2);   //Position the cursor in second lines, sixth columns
    lcd.print(Distance);        //display distance
    lcd.print("cm");            //display
  }
  else
  {
    lcd.home();         //Move the cursor back to the upper left corner,
                        //which is the beginning of the output
    lcd.print("!!! Out of range");       //Display beyond distance
  }
  delay(250);
  lcd.clear();
}
void loop()
{
  keysacn();
  while(1)
  {
    Distance_test();      // Measuring the distance ahead
```

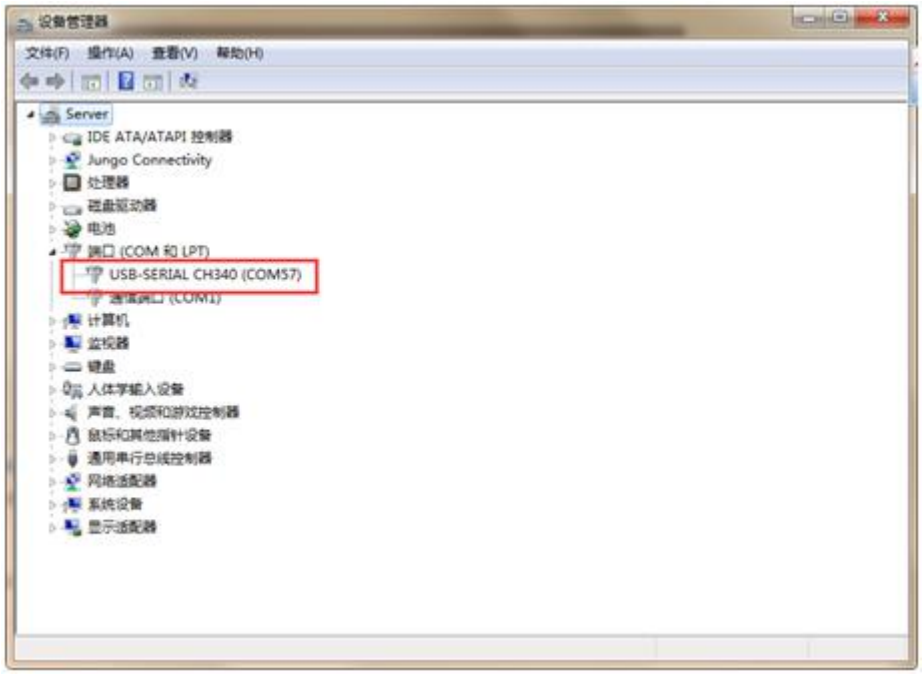
```
    Distance_display();    //LCD screen display distance
    if(Distance < 60)      //The value of the distance to the obstacle can be set
according to the actual situation.
      while(Distance < 60)// Judge whether there is an obstacle, again
                          //if there is a turning direction, continue to judge
      {
        right(1);
        brake(1);
        Distance_test();   // Measuring the distance ahead
        Distance_display();//LCD screen display distance
      }
    else
      run();//No obstacle, run
  }
}
```

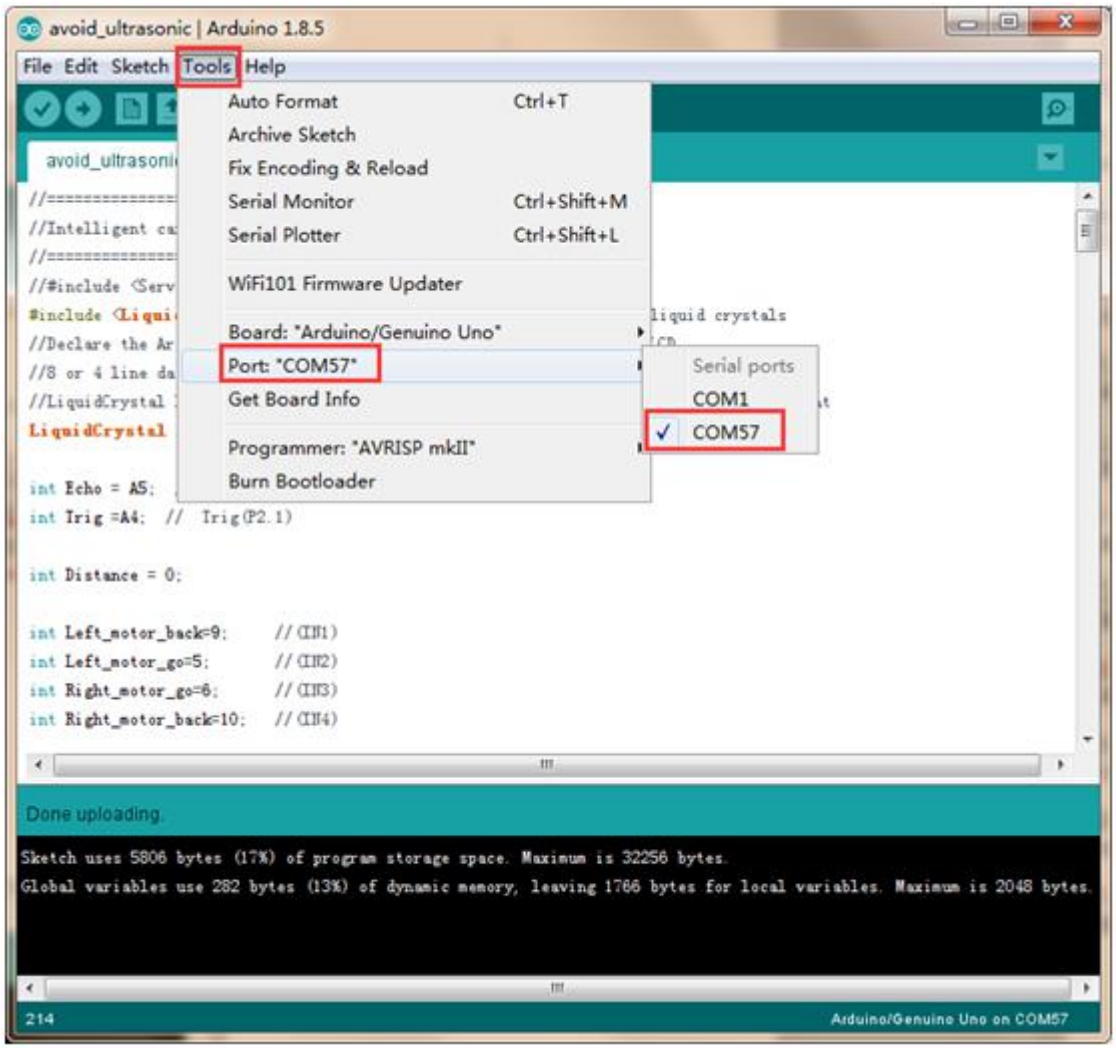
Experimental steps:

1. We need to open the code of this experiment:**avoid_ultrasonic.ino,**click“√” under the menu bar to compile the code, and wait for the word "**Done compiling** " in the lower right corner,as shown in the figure below.

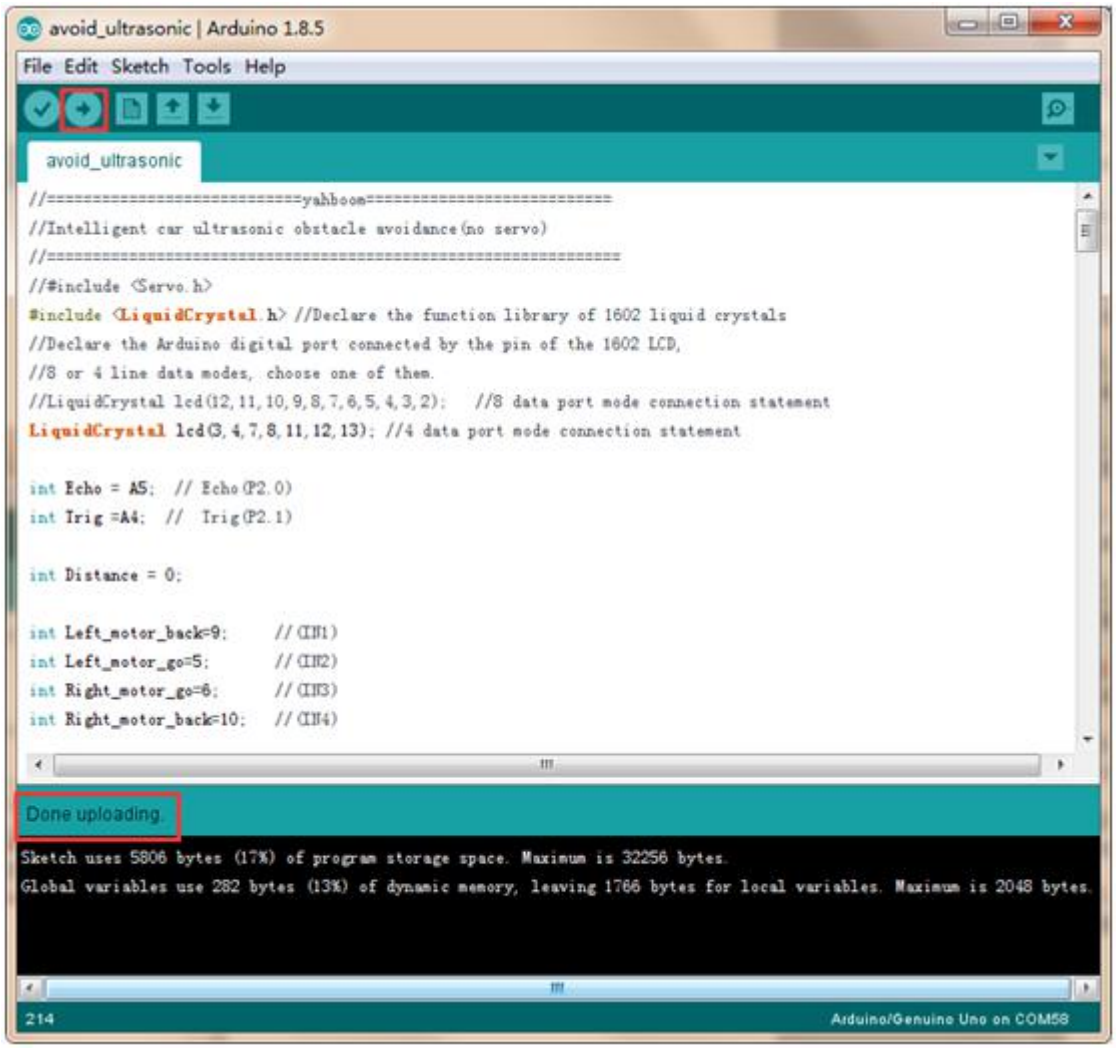


2. In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】---selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.



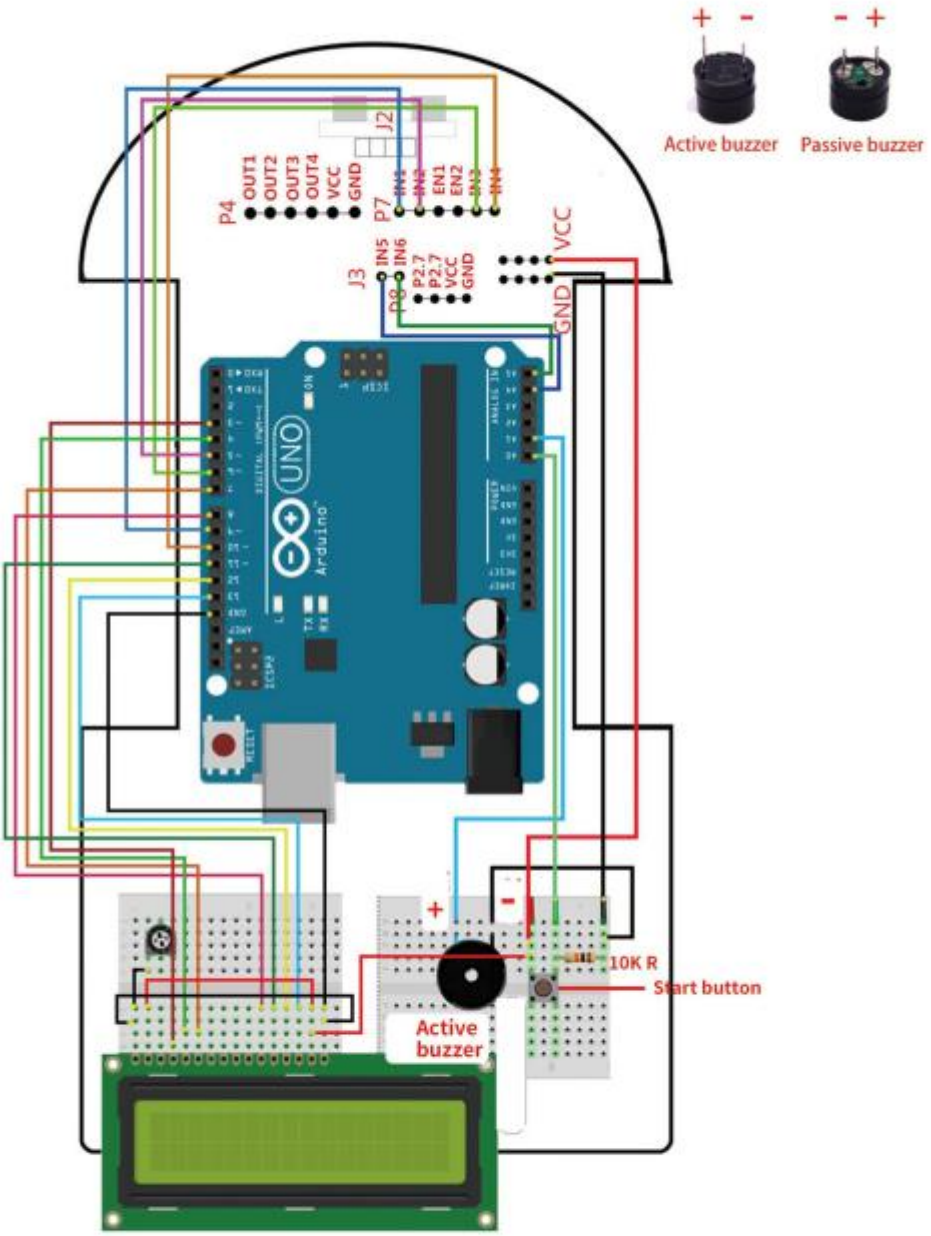


3. After the selection is completed, you need to click “→”under the menu bar to upload the code to the Arduino UNO board. When the word “**Done uploading**” appears in the lower left corner, the code has been successfully uploaded to the Arduino UNO board, as shown in the figure below.



4. Please wire the Smart Car as shown below.

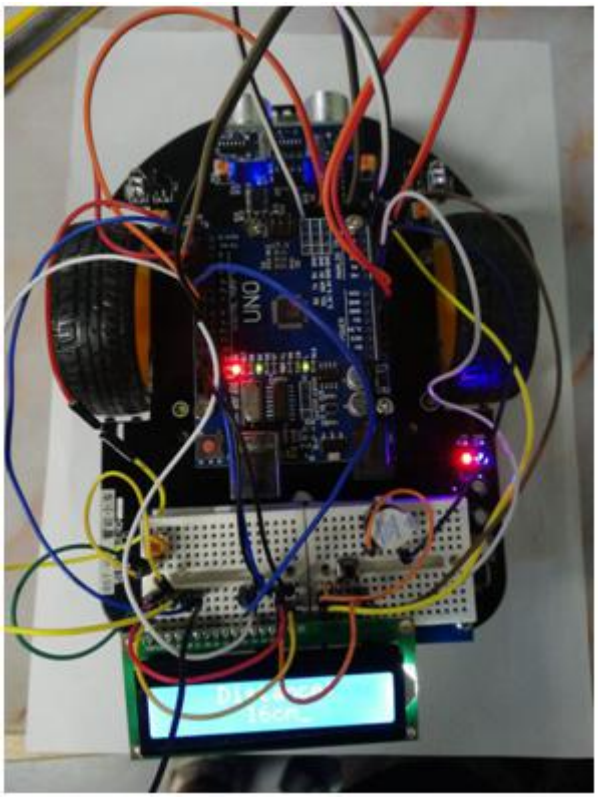
4.4 Ultrasonic obstacle avoidance(no servo) wiring diagram



Note: At the J2 slot, insert the ultrasonic sensor as picture.

If you only use the ultrasonic obstacle avoidance function without displaying the distance, you can not install the 1602 sdisplay and yellow adjustable resistance.

5.Put the smart car that has uploaded the program **avoid_ultrasonic.ino** on the ground, press the start button of the tail of the car, the LCD screen of the car shows the distance measured by the ultrasonic sensor and starts to avoid the obstacle ahead.


UNO