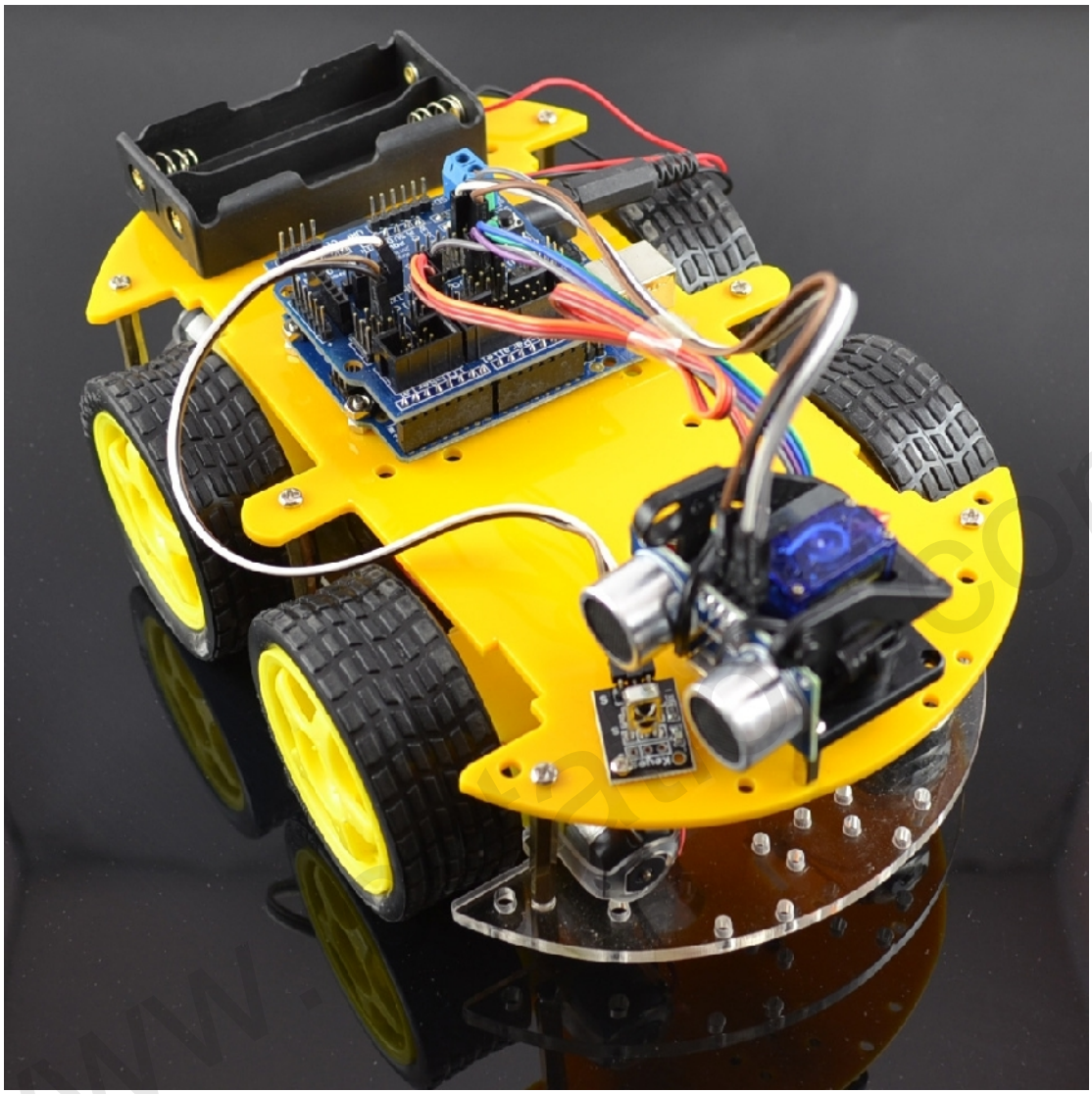
Arduino Bluetooth Multi-Function Instruction Manual



1.Brief instruction:

Arduino Bluetooth Multi_Function Smart Car is a MCU study and application development system base on Atmege328.Complete tracking, obstacle avoidance, infrared remote control and Bluetooth remote control functions. Kits contains a number of interesting programs, and expand external circuit module, thereby increasing car use function.

2.Parameter:

1>.Motor parameter:

Power apply:6V-9V;

Reduction gear ratio:1:48;

2>.Selection L298N module driver motor,segregate form CPU.

3>.Three tracking modules,detection white and black line.Available with fall prevention control.

4>.Infrared remote communication module,composition of remote control system of smart car.

5>.Bluetooth wireless module,can match with mobile phone to control smart car.

6>.Input voltage 7V-12V,can connect to different kinds of sensor.The more sensors ,the more functions.

3.Experimental curriculum:

1>.L298N driver motor;

2>.Tracking smart car;

3>.Ultrasonic obstacle avoidance smart car;

4>.Infrared remote control smart car;

5>.Arduino bluetooth control smart car;

6>.Multi_function smart car(tracking,obstacle avoidance ,infrared remote,bluetooth remote);

4.Listing:

1>.stepper motor x4

2>.wheel x4

3>.motor fixed block x4

4>.100x213x5mm perspex sheet x1

5>.100x213x5mm perspex sheet x1

6>.L298N driver board x1

7>.Arduino uno 328 controller x1

8>.Arduino sensor shield V5 x1

9>.PTZ x1

10>.Servo motor x1

11>.Ultrasonic module x1

12>.Three channel tracking module x1

13>.Infrared receive sensor x1

14>.Transducer x1

15>.18650 battery holder x1

16>.18650 battery x2

17>.18650 battery charger x1

18>.Bluetooth module x1

19>.Dupont wire x30

20>.USB cable 1M x1

22>.Copper cylinder M3*35mm x6

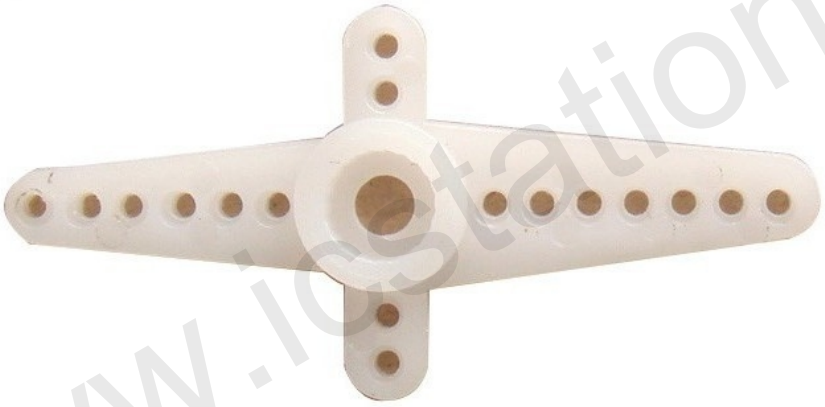
23>.Copper cylinder M3*20mm x3

24>.Copper cylinder M3*6mm x6

25>.M3 3mm screw and nut

5.PTZ and servo motor Installation Instruction

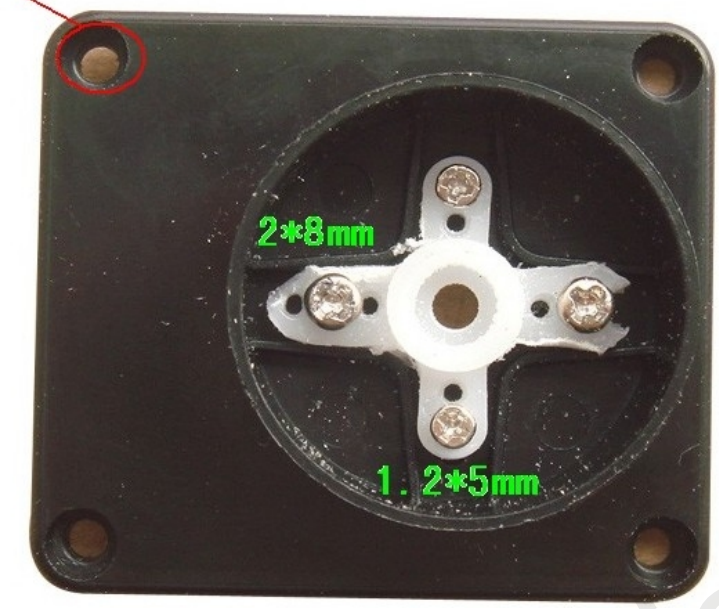
Pick up the cross colloid from accessory of serve

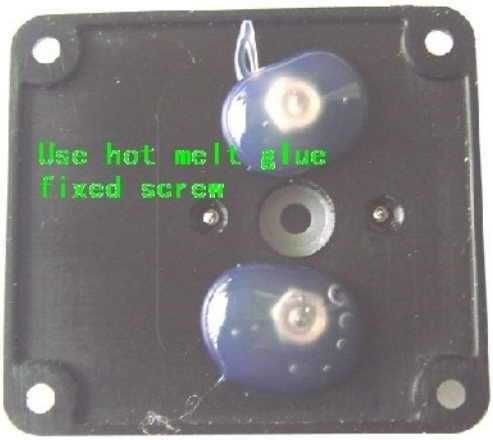


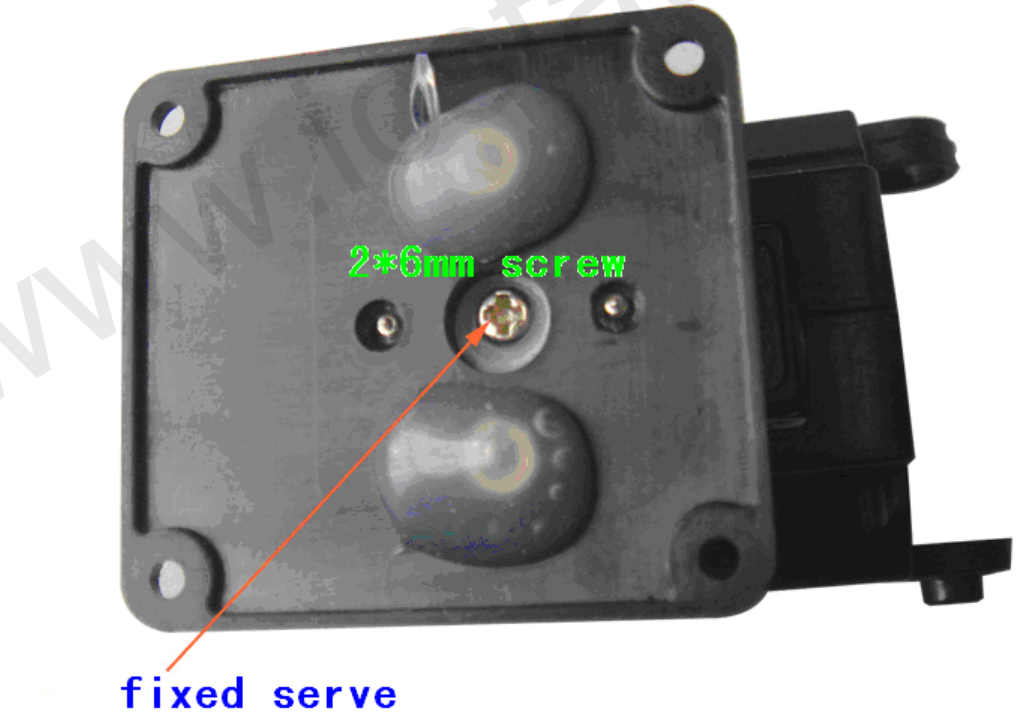
Process like below with Scissors and knives and some others tools

Install as below

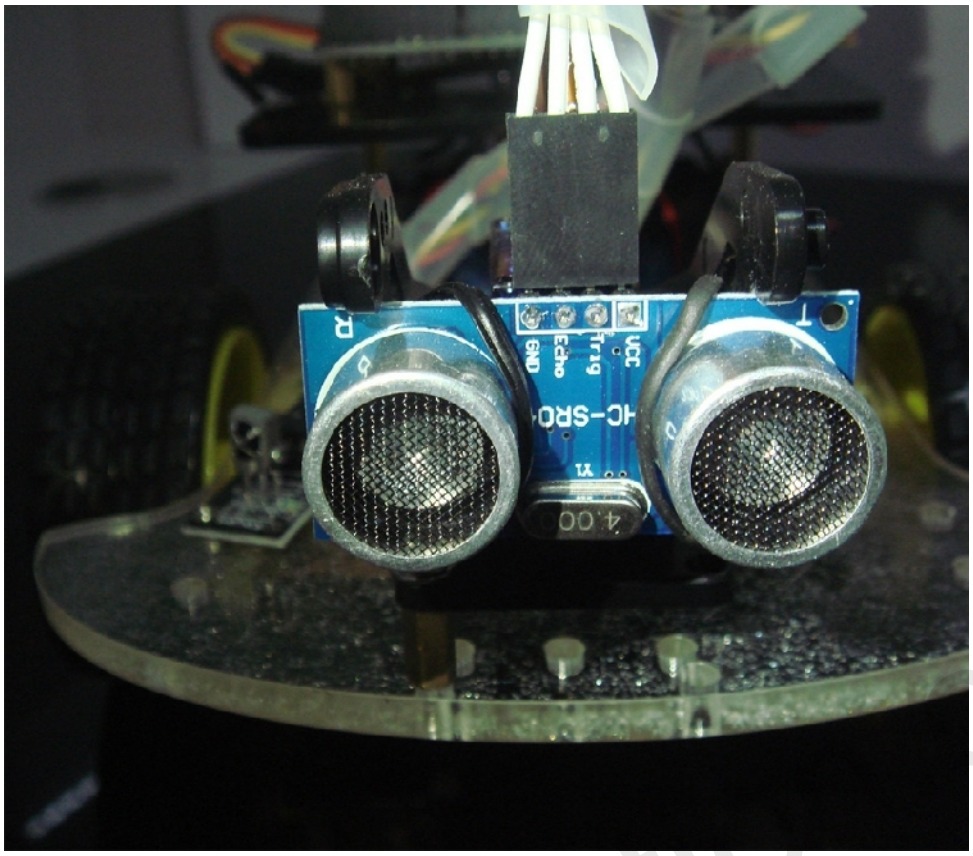
can use scissor or others tools to make the hole bigger if screw can not match with it





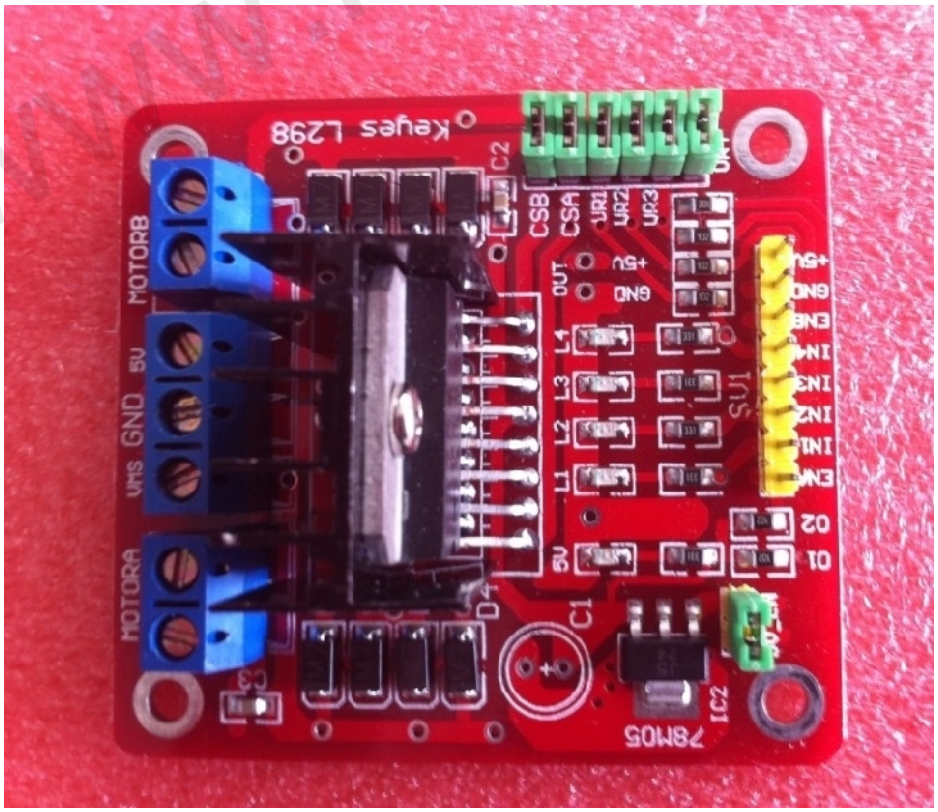



fixed ultrasonic
HC-SR0
Vcc
Trig
Echo



6.Experiment in detail

1>.L298N motor driver module



ENA(yellow in picture):

1(5V/PWM):enable motor A

0(GND/PWM):disenable motor A

IN1 to 5V,IN2 to GND,motor A corotation
IN1 to GND,IN2 to 5V,motor A rollback
ENB(yellow in picture):
1(5V/PWM):enable motor B

0(GND/PWM):disenable motor B

IN3 to 5V,IN4 to GND,motor A corotation

IN3 to GND,IN4 to 5V,motor A rollback

5V_EN(green in picture):

If use the jumper, chip 78 m05 provide power supply for modules

If do not use jumper, need to use 5v-pin & GND-pin provides power supply module for modules

CSA/CSB(green in picture):

Current test pins for motor A/B,can series connection resistance

If do not use jumper, detection of the current

If use the jumper,not detect current

UR1-UR4(green in picture)

Choose whether to use pull-up resistor

For I/O port driver ability of microcontroller, can short circuit, using pull-up resistor

If use the jumper, Do not use the pull-up resistor. If
do not use jumper ,use the pull-up resistor

Test code:

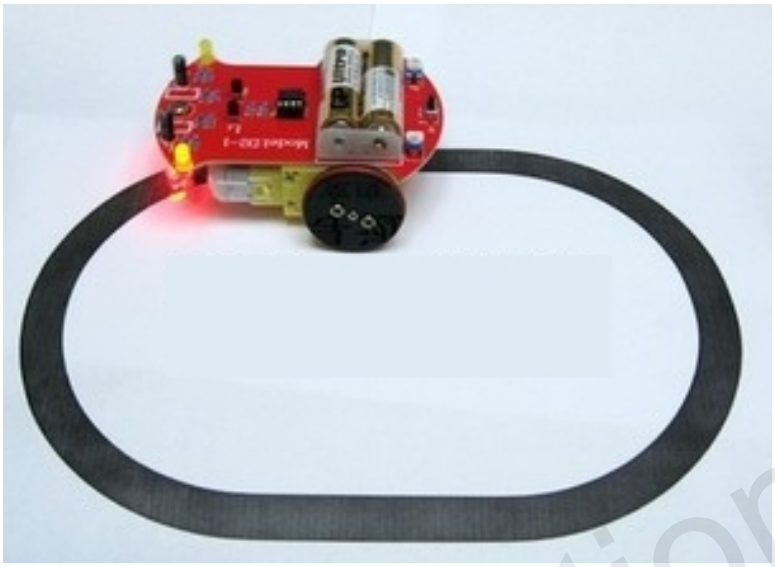
```
int pinI1=8;//define IN1 pin
int pinI2=9;//define IN2 pin
int speedpin=11;//define EA(PWM) pin
int pinI3=6;//define IN3 pin
int pinI4=7;//define IN4 pin
int speedpin1=10;//define EB(PWM) pin
void setup()
{
  pinMode(pinI1,OUTPUT);
  pinMode(pinI2,OUTPUT);
  pinMode(speedpin,OUTPUT);
  pinMode(pinI3,OUTPUT);
  pinMode(pinI4,OUTPUT);
```

```
  pinMode(speedpin1,OUTPUT);
}
void loop()
{
 //go straight
    analogWrite(speedpin,100);//define speed
    analogWrite(speedpin1,100);
    digitalWrite(pinI4,LOW);//right motor move in anticlockwise
    digitalWrite(pinI3,HIGH);
    digitalWrite(pinI1,LOW);//left motor move in clockwise
    digitalWrite(pinI2,HIGH);
    delay(2000);
 //go back
    analogWrite(speedpin,100);//define speed
    analogWrite(speedpin1,100);
    digitalWrite(pinI4,HIGH);//right motor move in clockwise
     digitalWrite(pinI3,LOW);
    digitalWrite(pinI1,HIGH);//left motor move in anticlockwise
    digitalWrite(pinI2,LOW);
    delay(2000);
 //turn left
    analogWrite(speedpin,60);//
    analogWrite(speedpin1,60);
    digitalWrite(pinI4,LOW);//
    digitalWrite(pinI3,HIGH);
    digitalWrite(pinI1,HIGH);//
    digitalWrite(pinI2,LOW);
    delay(2000);
 //turn right
    analogWrite(speedpin,60);//
    analogWrite(speedpin1,60);
    digitalWrite(pinI4,HIGH);//
    digitalWrite(pinI3,LOW);
    digitalWrite(pinI1,LOW);//
    digitalWrite(pinI2,HIGH);
    delay(2000);
 //stop
    digitalWrite(pinI4,HIGH);//
    digitalWrite(pinI3,HIGH);
    digitalWrite(pinI1,HIGH);//
    digitalWrite(pinI2,HIGH);
```

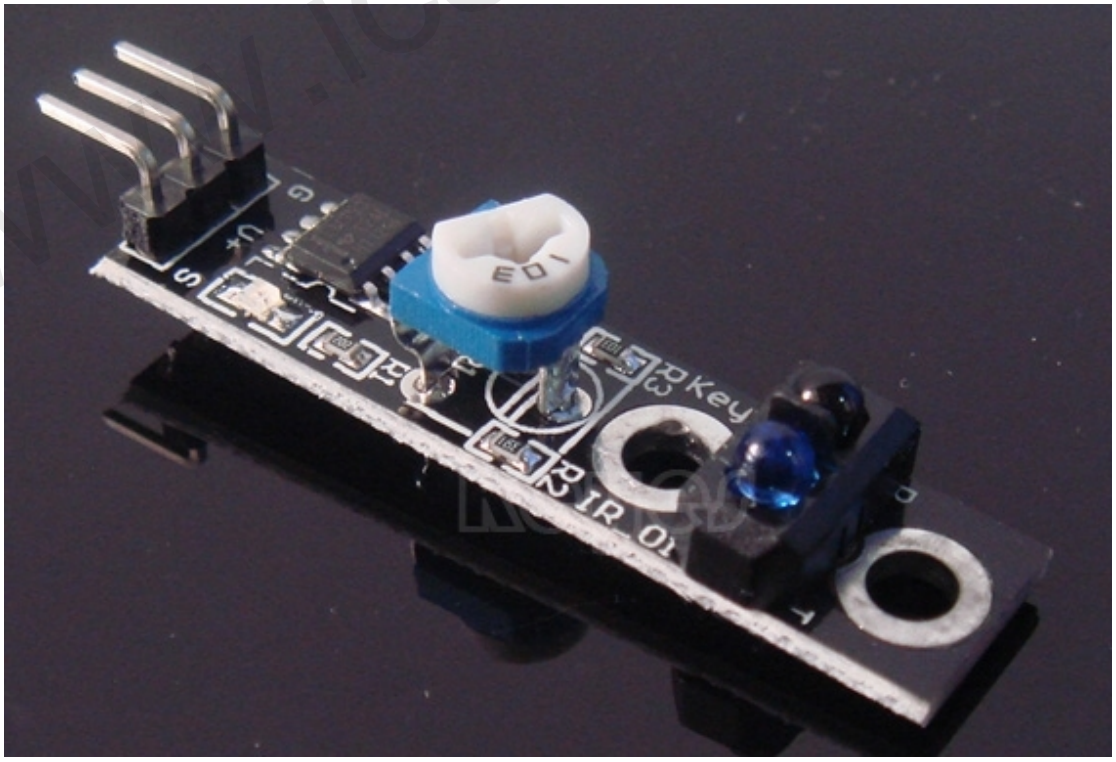
```
    delay(2000);
}
```

NOTE:You can use other code to driver motor.

2>.Tracking smart car



Tracking module principle:TCRT5000 Using infrared reflectivity of color is different, the strength of the reflected signal is converted into electrical signals. Black and white tracing module in high level effectively detect black, white is detected for the low level, effectively detect 0-3 cm in height.



method of application:

1>>.There are 3 row needle sensor interfaces, is GND, VCC, OUT. VCC &gnd for power

supply side, the OUT signal is output.

2>>.An object is detected, the output signal low level; Not detected objects, the output signal of high level.

3>>.Major judgment signal output is 0 or 1, will be able to determine whether an object exists.

performance parameter:

1>>.detect distance:Test white paper is about 2cm..Depending on the color of different distance is different.white is farthest

2>>.supply voltage:2.5V~12V,Not more than 12V(It is best to low voltage power supply, power supply voltage is too high will shorten the life of a sensor.5V power supply is preferred)

3>>.operating current:18-20mA when 5V.By a large number of tests, sensor hardware Settings for 18~20mA best performance when working current, main performance on anti-jamming capability

4>>.An object is detected, the output signal low level; Not detected objects, the output signal of high level.

5>>.Sensor output TTL level, can be directly connected to the 3.3 V or 5 V microcontroller IO port.

Black or white line detection principle:

1>>.Using black to light the reflectivity of the characteristics, when the surface color is not black, infrared sensors to launch out by most reflected. The sensor output low level 0.

2>>.When there is a black line plane, sensors in the black, because black reflection ability is very weak, very few reflected infrared light, short of sensor action level, so the sensor output 1.

3>>.Single chip microcomputer as long as we use to judge the sensor output is 0 or 1, will be able to detect the black line.

4>>.Detection principle of the white line and black line, the principle of the detection of the white line, white line around the color is close to black, then adjust the adjustable resistance of infrared sensor above, will lower sensitivity, has been transferred to the surrounding color just detect, it can detect the white line.

Test code:

```
int pin=7;//
int val;//
void setup()
{
  pinMode(ledPin,OUTPUT);//
  Serial.begin(9600);//
}
void loop()
{
 val=digitalRead(pin);//
 Serial.println(val);//
}
```

Tracking smart test code:

```
int MotorRight1=5;
int MotorRight2=6;
int MotorLeft1=10;
int MotorLeft2=11;
const int SensorLeft = 7;          //left sensor input
const int SensorMiddle= 4 ;        //middle sensor input
const int SensorRight = 3;         //right sensor input
int SL;       //left sensor state
int SM;       //
int SR;       //

void setup()
{
   Serial.begin(9600);
   pinMode(MotorRight1, OUTPUT);   // pin 8 (PWM)
   pinMode(MotorRight2, OUTPUT);   // 9 (PWM)
   pinMode(MotorLeft1,   OUTPUT);   // 10 (PWM)
   pinMode(MotorLeft2,   OUTPUT);   // 11 (PWM)
   pinMode(SensorLeft, INPUT); //
   pinMode(SensorMiddle, INPUT);//
   pinMode(SensorRight, INPUT); //
}

void loop()
  {
  SL = digitalRead(SensorLeft);
  SM = digitalRead(SensorMiddle);
  SR = digitalRead(SensorRight);

        if (SM == HIGH)//middle sensor in black area
        {
            if (SL == LOW & SR == HIGH) // left sensor in black area,right sensor in white area,so
turn left
            {
                digitalWrite(MotorRight1,LOW);
                digitalWrite(MotorRight2,HIGH);
                analogWrite(MotorLeft1,0);
                analogWrite(MotorLeft2,80);
            }
            else if (SR == LOW & SL == HIGH) //
            {
                analogWrite(MotorRight1,0);//
                analogWrite(MotorRight2,80);
                digitalWrite(MotorLeft1,LOW);
                digitalWrite(MotorLeft2,HIGH);
            }
```

```
    else    //
      {
          digitalWrite(MotorRight1,LOW);
          digitalWrite(MotorRight2,HIGH);
          digitalWrite(MotorLeft1,LOW);
          digitalWrite(MotorLeft2,HIGH);
          analogWrite(MotorLeft1,200);
          analogWrite(MotorLeft2,200);
          analogWrite(MotorRight1,200);
          analogWrite(MotorRight2,200);
    }
  }
  else //
 {
    if (SL == LOW & SR == HIGH)//
   {
        digitalWrite(MotorRight1,LOW);
        digitalWrite(MotorRight2,HIGH);
        digitalWrite(MotorLeft1,LOW);
        digitalWrite(MotorLeft2,LOW);
   }
    else if (SR == LOW & SL == HIGH) //
   {
      digitalWrite(MotorRight1,LOW);
      digitalWrite(MotorRight2,LOW);
      digitalWrite(MotorLeft1,LOW);
      digitalWrite(MotorLeft2,HIGH);
   }
    else //
   {
   digitalWrite(MotorRight1,HIGH);
   digitalWrite(MotorRight2,LOW);
   digitalWrite(MotorLeft1,HIGH);
   digitalWrite(MotorLeft2,LOW);;
   }}}
```

3>.Ultrasonic obstacle avoidance intelligent car

Obstacle avoidance intelligent ultrasonic is convenient, simple and easy to do real-time control, and can meet the practical requirements in terms of accuracy of measurement, thus become a commonly used method of obstacle avoidance. Ultrasonic method using reference (Arduino ultrasonic ranging).

Ultrasonic smart wiring diagram;

IN1,IN2--Motor A(right wheel)
IN3,IN4--Motor B(left wheel)

L298N Module	Arduino pin	Define in Code
IN1	11	pinRF
IN2	10	pinRB
IN3	9	pinLF
IN4	6	pinLB

pinRF	pinRB	pinLF	pinLB	Intelligent car
IN1	IN2	IN3	IN4	
1	0	0	1	Advance
0	1	1	0	Back
1	1	1	1	Stop
1	0	1	1	Single wheel turn left
1	1	0	1	Single wheel turn right
1	0	1	0	Double wheel turn left
0	1	0	1	Double wheel turn right

Test code:

```
#include <Servo.h>
int pinLB=6;          // left back
int pinLF=9;          // left front

int pinRB=10;         // right back
int pinRF=11;         // left front

int inputPin = A0;    // ultrasonic echo
 int outputPin =A1;   // ultrasonic trig

int Fspeedd = 0;           // front distance
int Rspeedd = 0;           // right distance
int Lspeedd = 0;           // left distance
int directionn = 0;      // Determine the direction of car turns
Servo myservo;               //    myservo
int delay_time = 250; // Stable steering servo motor

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

```
void setup()
{
  pinMode(pinLB,OUTPUT); // pin 6 (PWM)
  pinMode(pinLF,OUTPUT); // pin 9 (PWM)
  pinMode(pinRB,OUTPUT); // pin 10 (PWM)
  pinMode(pinRF,OUTPUT); // pin 11 (PWM)

  pinMode(inputPin, INPUT);       // Define ultrasound input pin
  pinMode(outputPin, OUTPUT);    // Define ultrasound output pin

  myservo.attach(5);       // Define the servo motor output pin5 (PWM)
}
void advance(int a)         // advance
      {                            //In the mid-point of the two wheels as a reference
       digitalWrite(pinRB,LOW);    //right wheel advance
       digitalWrite(pinRF,HIGH);
       digitalWrite(pinLB,HIGH);    //left wheel advance
       digitalWrite(pinLF,LOW);
       delay(a);
      }
void right(int b)              //turn right (single wheel)
      {
       digitalWrite(pinRB,HIGH);      //right stop
       digitalWrite(pinRF,HIGH);
       digitalWrite(pinLB,HIGH);    //left advance
       digitalWrite(pinLF,LOW);
       delay(b);
      }
void left(int c)               //turn left(single wheel)
      {
       digitalWrite(pinRB,LOW); //righ wheel advance
       digitalWrite(pinRF,HIGH);
       digitalWrite(pinLB,HIGH);      //left stop
       digitalWrite(pinLF,HIGH);
       delay(c);
      }
void turnR(int d)              //turn right(double wheels)
      {
       digitalWrite(pinRB,HIGH);    //right wheel back
       digitalWrite(pinRF,LOW);
```

```
    digitalWrite(pinLB,HIGH); //left wheel advance
    digitalWrite(pinLF,LOW);
    delay(d);
  }
void turnL(int e)          //turn left (double wheels)
  {
    digitalWrite(pinRB,LOW);    //right wheel advance
    digitalWrite(pinRF,HIGH);
    digitalWrite(pinLB,LOW);    //left wheel back
    digitalWrite(pinLF,HIGH);
    delay(e);
  }
void stopp(int f)          //stop
  {
    digitalWrite(pinRB,HIGH);
    digitalWrite(pinRF,HIGH);
    digitalWrite(pinLB,HIGH);
    digitalWrite(pinLF,HIGH);
    delay(f);
  }
void back(int g)           //back
  {
    digitalWrite(pinRB,HIGH);   //right wheel back
    digitalWrite(pinRF,LOW);
    digitalWrite(pinLB,LOW);    //left wheel back
    digitalWrite(pinLF,HIGH);
    delay(g);
  }
void detection()           //Measuring three angles(2.90.178)
  {
    myservo.write(90); //measure distance in the front
    delay(delay_time);          // Waiting for servo motor stable
    ask_pin_F();                // Read the distance of front

    if(Fspeedd < 20)            // If the distance is less than 20cm in front
    {
        stopp(1);                   // clear output ,motor stop
        myservo.write(178);         //measure left distance
        delay(delay_time);
        ask_pin_L();
```

```
            myservo.write(2);          //measure right distance
            delay(delay_time);
            ask_pin_R();

            if(Lspeedd > Rspeedd)      //compare distance of right and left
            {
                directionn = Lgo;           //turn left
            }
            if(Lspeedd <= Rspeedd)     //if the distance is less than or equal to the distance at the right
            {
                directionn = Rgo;           //turn right
            }
        }
        else
        {
           directionn = Fgo;
         }
         myservo.write(90);
         delay(delay_time);
    }
void ask_pin_F()      // Measure the distance in front
    {
        digitalWrite(outputPin, LOW);      //Ultrasonic launch 2us low level
        delayMicroseconds(2);
        digitalWrite(outputPin, HIGH);    // ultrasound transmitting high voltage 10us, there is at least
10us
        delayMicroseconds(11);
        digitalWrite(outputPin, LOW);        // Ultrasonic launch low level
        float Fdistance = pulseIn(inputPin, HIGH);    //measure time
        Fdistance= Fdistance/5.8/10;             // time to distance ⌒cm⌒
        Fspeedd = Fdistance;                        //
    }
 void ask_pin_L()      //
    {
        delay(delay_time);
        digitalWrite(outputPin, LOW);      //
        delayMicroseconds(2);
        digitalWrite(outputPin, HIGH);    //
        delayMicroseconds(11);
        digitalWrite(outputPin, LOW);       //
```

```
        float Ldistance = pulseIn(inputPin, HIGH);    //
        Ldistance= Ldistance/5.8/10;              //
        Lspeedd = Ldistance;                       //
    }
void ask_pin_R()     //
    {
        delay(delay_time);
        digitalWrite(outputPin, LOW);      //
        delayMicroseconds(2);
        digitalWrite(outputPin, HIGH);    //
        delayMicroseconds(11);
        digitalWrite(outputPin, LOW);       //
        float Rdistance = pulseIn(inputPin, HIGH);    //
        Rdistance= Rdistance/5.8/10;              //
        Rspeedd = Rdistance;                       //
    }
void loop()
 {
  detection();              //Measure the Angle and determine which direction to go to
  if(directionn == 2)
  {
    back(600);
  }
  if(directionn == 6)
  {
    turnR(350);
    stopp(1);
  }
  if(directionn == 4)
  {
    turnL(350);
    stopp(1);
  }
  if(directionn == 8)
  {
    advance(10);
    ask_pin_F();
    if(Fspeedd < 20) stopp(1);
  }
 }
```

4>.Infrared remote control of intelligent car

```
#include <IRremote.h>                                    //
const int irReceiverPin = 2;                          //
IRrecv irrecv(irReceiverPin);                          //
decode_results results;                                 //
void setup()
{
  Serial.begin(9600);                                    //
  irrecv.enableIRIn();                                  //
}

//
void showIRProtocol(decode_results *results)
{
  Serial.print("Protocol: ");

  //
  switch(results->decode_type) {
    case NEC:
      Serial.print("NEC");
      break;
    case SONY:
      Serial.print("SONY");
      break;
    case RC5:
      Serial.print("RC5");
      break;
    case RC6:
      Serial.print("RC6");
      break;
    default:
      Serial.print("Unknown encoding");
  }

  //
  Serial.print(", irCode: ");
  Serial.print(results->value, HEX);       //
  Serial.print(",    bits: ");
  Serial.println(results->bits);              //
}

void loop()
{
  if (irrecv.decode(&results)) {               //
    showIRProtocol(&results);                    //
    irrecv.resume();                               //
```

```
  }
}

//******红外控制部分********
long advence = 0x00EF807F;
long back = 0x00EFA05F;
long stop = 0x00EF906F;
long left = 0x00EF00FF;
long right = 0x00EF40BF;
```

```
//******Infrared remote smart car code*******
#include <IRremote.h>
int RECV_PIN = A0;
int pinLB=6;//
int pinLF=9;//
int pinRB=3;//
int pinRF=5;//
//******
long advence = 0x00EF807F;
long back = 0x00EFA05F;
long stop = 0x00EF906F;
long left = 0x00EF00FF;
long right = 0x00EF40BF;

IRrecv irrecv(RECV_PIN);
decode_results results;
void dump(decode_results *results) {
  int count = results->rawlen;
  if (results->decode_type == UNKNOWN)
    {
     Serial.println("Could not decode message");
    }
  else
   {
    if (results->decode_type == NEC)
      {
       Serial.print("Decoded NEC: ");
      }
    else if (results->decode_type == SONY)
      {
       Serial.print("Decoded SONY: ");
      }
    else if (results->decode_type == RC5)
      {
       Serial.print("Decoded RC5: ");
      }
```

```
      else if (results->decode_type == RC6)
        {
          Serial.print("Decoded RC6: ");
        }
        Serial.print(results->value, HEX);
        Serial.print(" (");
        Serial.print(results->bits, DEC);
        Serial.println(" bits)");
    }
        Serial.print("Raw (");
        Serial.print(count, DEC);
        Serial.print("): ");

   for (int i = 0; i < count; i++)
        {
          if ((i % 2) == 1) {
          Serial.print(results->rawbuf[i]*USECPERTICK, DEC);
        }
       else
        {
          Serial.print(-(int)results->rawbuf[i]*USECPERTICK, DEC);
        }
       Serial.print(" ");
        }
          Serial.println("");
        }

void setup()
  {
   pinMode(RECV_PIN, INPUT);
   pinMode(pinLB,OUTPUT);
   pinMode(pinLF,OUTPUT);

   pinMode(pinRB,OUTPUT);
   pinMode(pinRF,OUTPUT);

Serial.begin(9600);
   irrecv.enableIRIn(); // Start the receiver
  }

int on = 0;
unsigned long last = millis();

void loop()
{
   if (irrecv.decode(&results))
```

```
    {
      // If it's been at least 1/4 second since the last
      // IR received, toggle the relay
      if (millis() - last > 250)
         {
          on = !on;
//          digitalWrite(8, on ? HIGH : LOW);
          digitalWrite(13, on ? HIGH : LOW);
          dump(&results);
         }
      if (results.value == advence )
      {digitalWrite(pinRB,LOW);//
      digitalWrite(pinRF,HIGH);
      digitalWrite(pinLB,LOW);//
      digitalWrite(pinLF,HIGH);}

      if (results.value == back )


        {digitalWrite(pinRB,HIGH);//﹀BACK
        digitalWrite(pinRF,LOW);}

      if (results.value == left )
      { digitalWrite(pinRB,LOW);// STOP
        digitalWrite(pinRF,HIGH);
        digitalWrite(pinLB,HIGH);//GO
        digitalWrite(pinLF,LOW);}


      if (results.value == right )
      { digitalWrite(pinRB,HIGH);//﹀GO
        digitalWrite(pinRF,LOW);
        digitalWrite(pinLB,HIGH);//STOP
        digitalWrite(pinLF,HIGH);}

      if (results.value == stop )
         {
        digitalWrite(pinRB,HIGH);//STOP
        digitalWrite(pinRF,HIGH);
        digitalWrite(pinLB,HIGH);//STOP
        digitalWrite(pinLF,HIGH);
    }
      last = millis();
      irrecv.resume(); // Receive the next value
   }
}
```

5>.Mobile phone bluetooth intelligent car

Bluetooth (Bluetooth) technology, is actually a short-range radio technology, using the" Bluetooth "technology, can effectively simplify the handheld computers, laptops and mobile phones for communication between mobile phones and other mobile communication terminal equipment, also can successfully simplify the above communication between the devices and the Internet (Internet), so that the modern communication equipment and data transmission between the Internet become more efficiently, to broaden the way with wireless communications.

Because today is the first time to deal with the bluetooth module, or to a profound, let the Arduino and PC communication success. Wiring, first connect bluetooth motherboard + 5 v VCC, bluetooth motherboard GND connection - GND, motherboard TX connected bluetooth RX, RX connected bluetooth TX. When the success of the bluetooth module when the power is connected to the PC, the bluetooth module power will flashing lights, connect light green light will light up.

Test code:

```
char val;
int ledpin=13;
void setup()
{
  Serial.begin(9600);
  pinMode(ledpin,OUTPUT);
}
void loop()
{
  val=Serial.read();
  if(val=='r')
  {
   digitalWrite(ledpin ’HIGH);
   delay((500);
    digitalWrite(ledpin ’LOW);
    delay(500);
   Serial.println("keyes");
  }
 }
```

Let's learn the Arduino bluetooth remote control programmable intelligent car. Through bluetooth control forward, backward, turn left, turn right, buttons, computers and mobile phones, two kinds of control mode. (mobile phone operating system support Android 2.3.7 above. The computer must bring their own bluetooth)

When used for the first time to mobile phone with bluetooth car matching (pairing for the first time after the later don't have in wireless location), first take a look at the following steps:

1>>.Remember to open mobile phone bluetooth oh, open the software will remind users open the bluetooth

2>>.Then the text as shown in figure tips, connect a bluetooth device, scan matching bluetooth oh, otherwise not be able to connect to the car.

3>>.Match the car, the password is "1234" try it

Test code:

```
int MotorRight1=5;
int MotorRight2=6;
int MotorLeft1=10;
int MotorLeft2=11;
void setup()
{
  Serial.begin(9600);
  pinMode(MotorRight1, OUTPUT);   //   pin 8 (PWM)
  pinMode(MotorRight2, OUTPUT);   //   9 (PWM)
  pinMode(MotorLeft1,   OUTPUT);   //   10 (PWM)
  pinMode(MotorLeft2,   OUTPUT);   //   11 (PWM)
}

void go()//
{
        digitalWrite(MotorRight1,LOW);
        digitalWrite(MotorRight2,HIGH);
        digitalWrite(MotorLeft1,LOW);
        digitalWrite(MotorLeft2,HIGH);

}

void left() //
{
      digitalWrite(MotorRight1,HIGH);
      digitalWrite(MotorRight2,LOW);
      digitalWrite(MotorLeft1,LOW);
      digitalWrite(MotorLeft2,HIGH);

}
void right() //
{
      digitalWrite(MotorRight1,LOW);
      digitalWrite(MotorRight2,HIGH);
      digitalWrite(MotorLeft1,HIGH);
      digitalWrite(MotorLeft2,LOW);

}
void stop() //
{
```

```
        digitalWrite(MotorRight1,LOW);
        digitalWrite(MotorRight2,LOW);
        digitalWrite(MotorLeft1,LOW);
        digitalWrite(MotorLeft2,LOW);

}
void back() //
{
            digitalWrite(MotorRight1,HIGH);
            digitalWrite(MotorRight2,LOW);
            digitalWrite(MotorLeft1,HIGH);
            digitalWrite(MotorLeft2,LOW);;

}
void loop()
{
   char val = Serial.read();
   Serial.write(val);
   if (-1 != val) {
      if ('W' == val)
      go();
      else if ('A' ==val)
      left();
      else if ('D' == val)
      right();
      else if ('S' == val)
      back();
      else if ('Q' == val)
         stop();
      delay(500);
      }
   else
   {
      //stop();
      delay(500);
   }
}
```

6.Multi_Function

```
//******************************
#include <IRremote.h>
#include <Servo.h>
//**********************define motor pin************************
int MotorRight1=5;
int MotorRight2=6;
```

```
int MotorLeft1=10;
int MotorLeft2=11;
int counter=0;
const int irReceiverPin = 2; //Infrared receive sensor

char val;
//************************set detection IRcode************************
long IRfront= 0x00FFA25D;              //go straight
long IRback=0x00FF629D;                //go back
long IRturnright=0x00FFC23D;       //turn right
long IRturnleft= 0x00FF02FD;         //turn left
long IRstop=0x00FFE21D;                //stop
long IRcny70=0x00FFA857;               //CNY70 automatic mode
long IRAutorun=0x00FF906F;           //Ultrasonic automatic mode
long IRturnsmallleft= 0x00FF22DD;
//************************define CNY70 pin*****************************
const int SensorLeft = 7;           //left sensor input
const int SensorMiddle= 4 ;        //middle sensor input
const int SensorRight = 3;          //right sensor input
int SL;       //left sensor state
int SM;       //middle sensor state
int SR;       //right sensor state
IRrecv irrecv(irReceiverPin);    // define IRrecv receive signal
decode_results results;             // decoderesults
//************************define ultrasonic pin****************
int inputPin =13 ; // ultrasonic receive pin
int outputPin =12; //ultrasonic echo pin
int Fspeedd = 0; // front distance
int Rspeedd = 0; // right distance
int Lspeedd = 0; // left distance
int directionn = 0; // front=8; back=2; left=4; right=6
Servo myservo; // define myservo
int delay_time = 250; // servo motor go back state time
int Fgo = 8; // go straight
int Rgo = 6; // turn right
int Lgo = 4; // turn left
int Bgo = 2; // go back
//*********************************(SETUP)
void setup()
{
  Serial.begin(9600);
  pinMode(MotorRight1, OUTPUT);    // pin 8 (PWM)
  pinMode(MotorRight2, OUTPUT);    //pin 9 (PWM)
  pinMode(MotorLeft1,    OUTPUT);    // pin 10 (PWM)
  pinMode(MotorLeft2,    OUTPUT);    // pin 11 (PWM)
  irrecv.enableIRIn();          // start infrared decode
```

```
    pinMode(SensorLeft, INPUT); //
  pinMode(SensorMiddle, INPUT);//
  pinMode(SensorRight, INPUT); //
  digitalWrite(2,HIGH);
  pinMode(inputPin, INPUT); //
  pinMode(outputPin, OUTPUT); //
  myservo.attach(9); //


 }
//*************************************************(Void)
void advance(int a) // go straight
{
          digitalWrite(MotorRight1,LOW);
          digitalWrite(MotorRight2,HIGH);
          digitalWrite(MotorLeft1,LOW);
          digitalWrite(MotorLeft2,HIGH);
          delay(a * 100);
}
void right(int b) //turn right(single wheel)
{
         digitalWrite(MotorLeft1,LOW);
         digitalWrite(MotorLeft2,HIGH);
         digitalWrite(MotorRight1,LOW);
         digitalWrite(MotorRight2,LOW);
         delay(b * 100);
}
void left(int c) //turn left(single wheel)
{
        digitalWrite(MotorRight1,LOW);
        digitalWrite(MotorRight2,HIGH);
        digitalWrite(MotorLeft1,LOW);
        digitalWrite(MotorLeft2,LOW);
        delay(c * 100);
}
void turnR(int d) //turn right(two wheels)
{
        digitalWrite(MotorRight1,HIGH);
        digitalWrite(MotorRight2,LOW);
        digitalWrite(MotorLeft1,LOW);
        digitalWrite(MotorLeft2,HIGH);
        delay(d * 100);
}
void turnL(int e) //turn left(two wheels)
{
        digitalWrite(MotorRight1,LOW);
```

```
        digitalWrite(MotorRight2,HIGH);
        digitalWrite(MotorLeft1,HIGH);
        digitalWrite(MotorLeft2,LOW);
        delay(e * 100);
}
void stopp(int f) //stop
{
      digitalWrite(MotorRight1,LOW);
      digitalWrite(MotorRight2,LOW);
      digitalWrite(MotorLeft1,LOW);
      digitalWrite(MotorLeft2,LOW);
      delay(f * 100);
}
void back(int g) //go back
{
          digitalWrite(MotorRight1,HIGH);
          digitalWrite(MotorRight2,LOW);
          digitalWrite(MotorLeft1,HIGH);
          digitalWrite(MotorLeft2,LOW);;
          delay(g * 100);
}
void detection() //measurement 3 angle(front ,left,right)
{
     int delay_time = 250; //
     ask_pin_F(); // detection distance in front

     if(Fspeedd < 10) // if distance less than 10mm
    {
       stopp(1); // clear output
       back(2); // go back 0.2s
    }
     if(Fspeedd < 25) //    if distance less than 25mm
    {
       stopp(1); // clear output
       ask_pin_L(); // detection distance in left
       delay(delay_time); // Waiting for the servo motor is stable
       ask_pin_R(); // detection distance in right
       delay(delay_time); // waiting for servo motor state

       if(Lspeedd > Rspeedd) //if left distance greater than right
      {
          directionn = Lgo; //go left
      }

       if(Lspeedd <= Rspeedd) //if left distance less than right
        {
```

```
            directionn = Rgo; //go right
          }

          if (Lspeedd < 15 && Rspeedd < 15) //if distance less 10mm both right and left
         {
            directionn = Bgo; //go back
          }
       }
       else //if distance greater than 25mm
     {
         directionn = Fgo; //go straight
     }
}
//*********************************************
void ask_pin_F() // detection distance in front
{
myservo.write(90);
digitalWrite(outputPin, LOW); // ultrasonic echo low level in 2us
delayMicroseconds(2);
digitalWrite(outputPin, HIGH); // ultrasonic echo high level in 10us, At least 10us
delayMicroseconds(10);
digitalWrite(outputPin, LOW); // ultrasonic echo low leve
float Fdistance = pulseIn(inputPin, HIGH); // read time
Fdistance= Fdistance/5.8/10; // turn time to distance
Serial.print("F distance:"); //output distance (mm)
Serial.println(Fdistance); //display distance
Fspeedd = Fdistance; // write distance to Fspeed
}
//*****************************************************
void ask_pin_L() // detection distance in left
{
myservo.write(177);
delay(delay_time);
digitalWrite(outputPin, LOW); // ultrasonic echo low level in 2us
delayMicroseconds(2);
digitalWrite(outputPin, HIGH); // ultrasonic echo high level in 10us, At least 10us
delayMicroseconds(10);
digitalWrite(outputPin, LOW); // ultrasonic echo low level
float Ldistance = pulseIn(inputPin, HIGH); // read time
Ldistance= Ldistance/5.8/10; // turn time to distance
Serial.print("L distance:"); //output distance (mm)
Serial.println(Ldistance); //display distance
Lspeedd = Ldistance; // write distance to Lspeed
}
//*************************************************************************************
void ask_pin_R() // detection distance in right
```

```
{
myservo.write(5);
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; //
}
//*********************************(LOOP)
void loop()
{
        SL = digitalRead(SensorLeft);
        SM = digitalRead(SensorMiddle);
        SR = digitalRead(SensorRight);
        performCommand();
//*****************************normal remote control mode
  if (irrecv.decode(&results))
    {               // Decoding success ,receive infrared signal
/*****************************************/
        if (results.value == IRfront)//go straight
          {
            advance(10);//go straight
          }
/*************************************/
        if (results.value ==    IRback)//go back
          {
            back(10);//go back
          }
/*************************************************************************/
        if (results.value == IRturnright)//turn right
        {
           right(6); // turn right
        }
/*************************************************************************/
       if (results.value == IRturnleft)//turn left
       {
          left(6); // turn left;
       }
/*************************************************************************/
      if (results.value == IRstop)//stop
```

```
    {
      digitalWrite(MotorRight1,LOW);
      digitalWrite(MotorRight2,LOW);
      digitalWrite(MotorLeft1,LOW);
      digitalWrite(MotorLeft2,LOW);
     }
//************************************cny70 automatic mode
     if (results.value == IRcny70)
    {
      while(IRcny70)
      {
        SL = digitalRead(SensorLeft);
        SM = digitalRead(SensorMiddle);
        SR = digitalRead(SensorRight);

        if (SM == HIGH)//middle sensor in black area
        {
            if (SL == LOW & SR == HIGH) // left sensor in black area,right sensor in white area ,so
turn left
            {
                digitalWrite(MotorRight1,LOW);
                digitalWrite(MotorRight2,HIGH);
                analogWrite(MotorLeft1,0);
                analogWrite(MotorLeft2,80);
            }
            else if (SR == LOW & SL == HIGH) //left white,right black ,turn right
            {
                analogWrite(MotorRight1,0);//
                analogWrite(MotorRight2,80);
                digitalWrite(MotorLeft1,LOW);
                digitalWrite(MotorLeft2,HIGH);
            }
          else    // left and right both in white ,go straight
            {
                digitalWrite(MotorRight1,LOW);
                digitalWrite(MotorRight2,HIGH);
                digitalWrite(MotorLeft1,LOW);
                digitalWrite(MotorLeft2,HIGH);
                analogWrite(MotorLeft1,200);
                analogWrite(MotorLeft2,200);
                analogWrite(MotorRight1,200);
                analogWrite(MotorRight2,200);
          }
        }
        else // middle sensor in white area
       {
```

```
            if (SL == LOW & SR == HIGH)// left black,right white,turn left
           {
                digitalWrite(MotorRight1,LOW);
                digitalWrite(MotorRight2,HIGH);
                digitalWrite(MotorLeft1,LOW);
                digitalWrite(MotorLeft2,LOW);
           }
            else if (SR == LOW & SL == HIGH) // left white,right black ,turn right
           {
               digitalWrite(MotorRight1,LOW);
               digitalWrite(MotorRight2,LOW);
               digitalWrite(MotorLeft1,LOW);
               digitalWrite(MotorLeft2,HIGH);
           }
            else // left and right both in white ,stop
           {
           digitalWrite(MotorRight1,HIGH);
           digitalWrite(MotorRight2,LOW);
           digitalWrite(MotorLeft1,HIGH);
           digitalWrite(MotorLeft2,LOW);;
           }
        }
          if (irrecv.decode(&results))
          {
                  irrecv.resume();
                        Serial.println(results.value,HEX);
                  if(results.value ==IRstop)
                  {
                    digitalWrite(MotorRight1,HIGH);
                    digitalWrite(MotorRight2,HIGH);
                    digitalWrite(MotorLeft1,HIGH);
                    digitalWrite(MotorLeft2,HIGH);
                    break;
                  }
          }
       }
        results.value=0;
    }
//*******************************ultracsonic automaitc mode
  if (results.value ==IRAutorun )
         {
               while(IRAutorun)
           {
                myservo.write(90); //make the servo motor reset
                detection(); //
                  if(directionn == 8) //directionn = 8(go straight)
```

```
  {
    if (irrecv.decode(&results))
 {
   irrecv.resume();
   Serial.println(results.value,HEX);
   if(results.value ==IRstop)
   {
     digitalWrite(MotorRight1,LOW);
     digitalWrite(MotorRight2,LOW);
     digitalWrite(MotorLeft1,LOW);
     digitalWrite(MotorLeft2,LOW);
     break;
   }
 }
       results.value=0;
       advance(1); //
       Serial.print(" Advance "); //
       Serial.print(" ");
  }
 if(directionn == 2) //2(go back)
{
  if (irrecv.decode(&results))
 {
   irrecv.resume();
   Serial.println(results.value,HEX);
   if(results.value ==IRstop)
   {
     digitalWrite(MotorRight1,LOW);
     digitalWrite(MotorRight2,LOW);
     digitalWrite(MotorLeft1,LOW);
     digitalWrite(MotorLeft2,LOW);
     break;
   }
 }
    results.value=0;
    back(8); //
    turnL(3); //To prevent the jammed
    Serial.print(" Reverse "); //
}
  if(directionn == 6) // 6(turn right)
{
 if (irrecv.decode(&results))
 {
    irrecv.resume();
    Serial.println(results.value,HEX);
   if(results.value ==IRstop)
```

```
    {
      digitalWrite(MotorRight1,LOW);
      digitalWrite(MotorRight2,LOW);
      digitalWrite(MotorLeft1,LOW);
      digitalWrite(MotorLeft2,LOW);
      break;
    }
  }
    results.value=0;
      back(1);
      turnR(6); //
      Serial.print(" Right "); //
}
  if(directionn == 4) // 4(turn left)
{
    if (irrecv.decode(&results))
  {
    irrecv.resume();
    Serial.println(results.value,HEX);
    if(results.value ==IRstop)
    {
      digitalWrite(MotorRight1,LOW);
      digitalWrite(MotorRight2,LOW);
      digitalWrite(MotorLeft1,LOW);
      digitalWrite(MotorLeft2,LOW);
      break;
    }
  }
       results.value=0;
       back(1);
       turnL(6); //
       Serial.print(" Left "); //
  }

    if (irrecv.decode(&results))
  {
    irrecv.resume();
    Serial.println(results.value,HEX);
    if(results.value ==IRstop)
    {
      digitalWrite(MotorRight1,LOW);
      digitalWrite(MotorRight2,LOW);
      digitalWrite(MotorLeft1,LOW);
      digitalWrite(MotorLeft2,LOW);
      break;
    }
```

```
                }
            }
                    results.value=0;
        }
/*********************************************************************************/
      else
     {
                digitalWrite(MotorRight1,LOW);
                digitalWrite(MotorRight2,LOW);
                digitalWrite(MotorLeft1,LOW);
                digitalWrite(MotorLeft2,LOW);
      }


          irrecv.resume();       //
   }
}

void performCommand() {
  if (Serial.available()) {
    val = Serial.read();
  }
    if (val == 'f') { // Forward
      advance(10);
    } else if (val == 'z') { // Stop Forward
      stopp(10) ;
    } else if (val == 'b') { // Backward
      back(10);
    } else if (val == 'y') { // Stop Backward
       back(10);
    }   else if (val == 'l') { // Right
      turnR(10);
    } else if (val == 'r') { // Left
      turnL(10);
    } else if (val == 'v') { // Stop Turn
      stopp(10) ;
    } else if (val == 's') { // Stop
      stopp(10) ;
    }

}
```