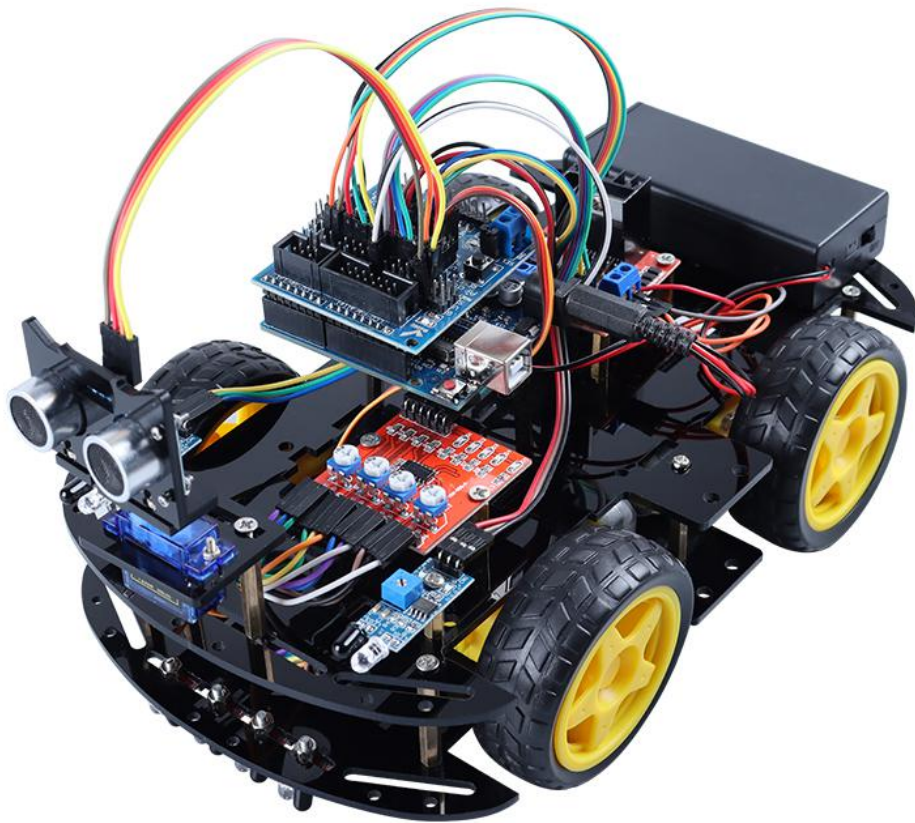
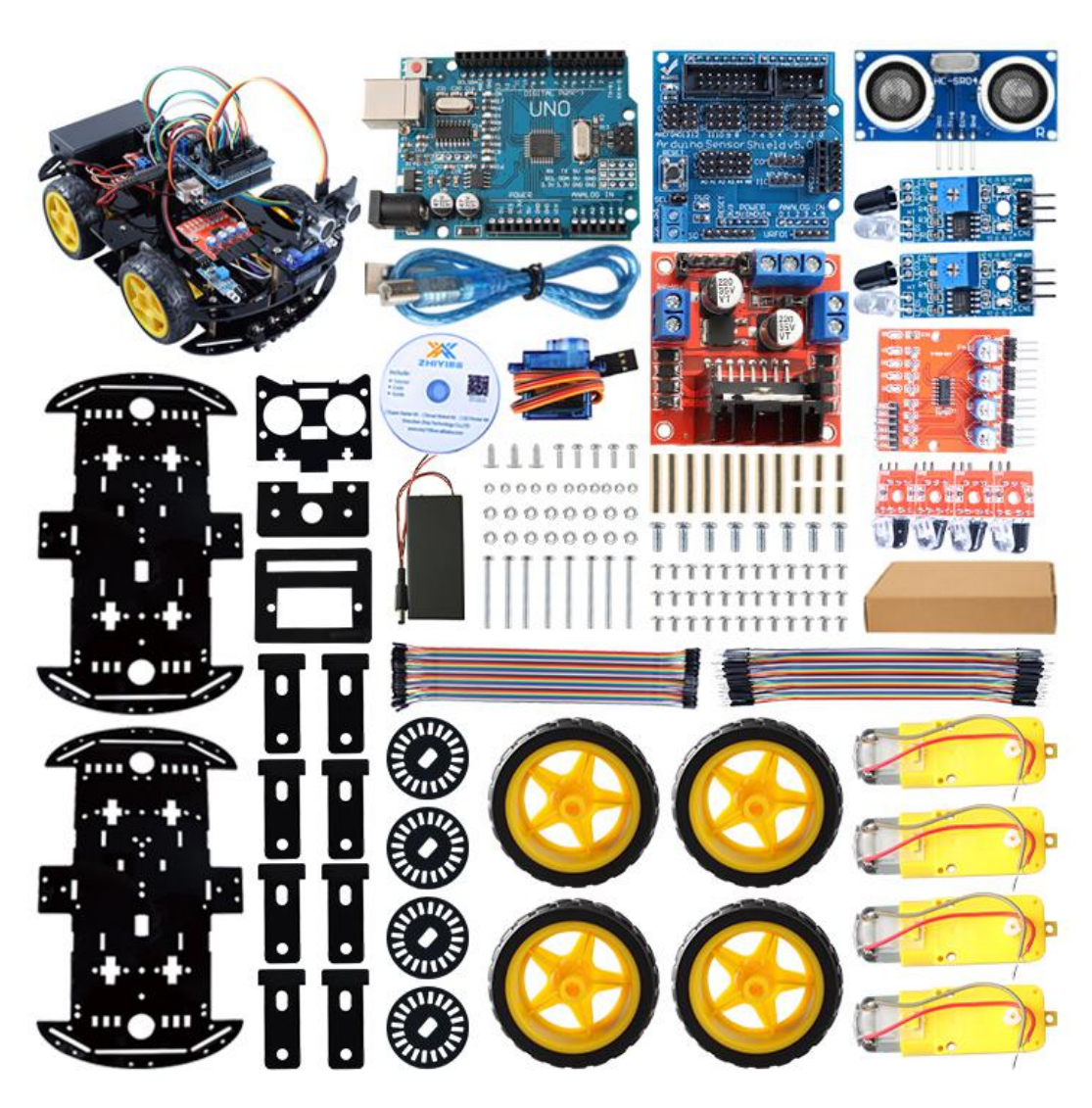

Arduino -based four way tracking car kit tutorial V2.0



V2.0.20.12.9



Preface

Welcome to use the DIY four way tracking and Ultrasonic obstacle avoidance car kit tutorial launched by ZHIYI! Whether you are an experienced maker or a novice with a small amount of professional skills, this tutorial will help you make something very cool: DIY four way tracking and Ultrasonic obstacle avoidance car kit tutorial mode.

The kit is based on the popular open source electronic platform Arduino. You can share and exchange your experiences and design ideas with fans all over the world. All components in the kit are individually packaged, including all the electronic components, modules and mechanical components required for these items. At the same time, there are detailed assembly and debugging instructions in this tutorial. To ensure that your smart car can be assembled and run smoothly, if you encounter difficult technical problems, you can contact us at any time, and we will provide you with high-quality technical support services for free as soon as possible.

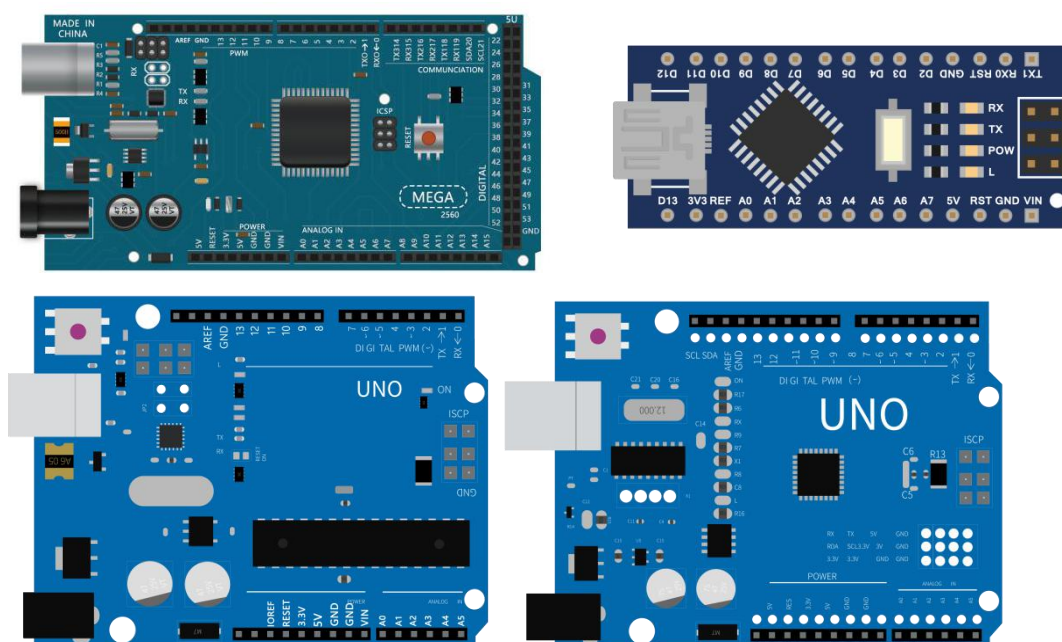
The content of this tutorial can ensure that novices without any professional skills can accomplish the goal. If you are interested in Arduino and want to learn how to program and build circuits, please visit our shop or contact us to purchase an Arduino learning starter kit specially prepared for beginners.

Arduino

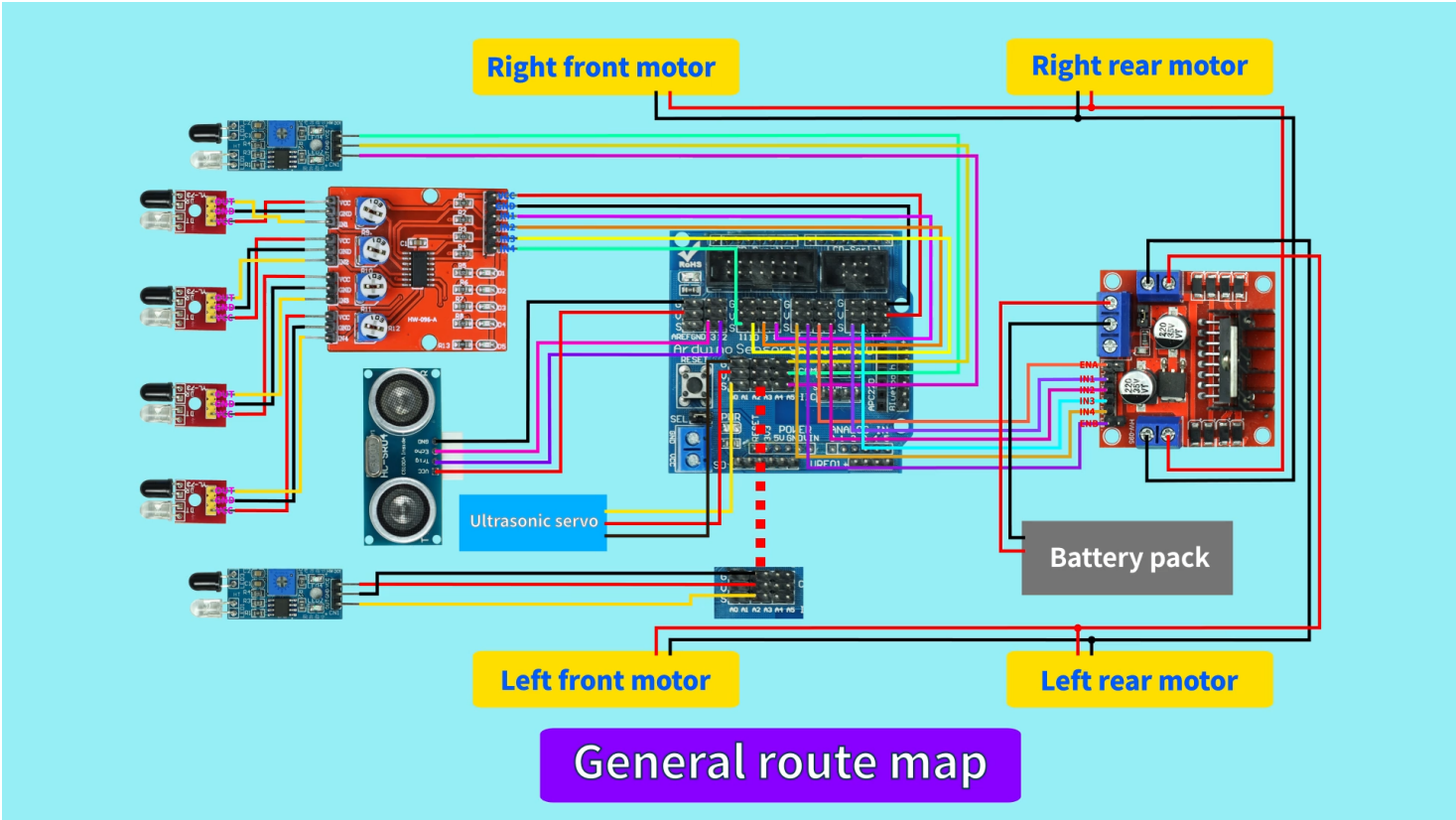
Arduino is an open source electronic platform based on easy-to-use hardware and software. Suitable for anyone working on interactive projects. Usually, an Arduino project consists of circuits and codes.

Arduino board

The Arduino board is a circuit board that integrates a microcontroller, input and output interfaces, etc. The Arduino board can use sensors to sense the environment and receive user operations to control LEDs, motor rotation, and so on. We just need to assemble the circuit and write the code. Currently, there are several models of Arduino development boards, and the codes between different types of development boards are common (due to different hardware, some development boards may not be fully compatible). Popular main control boards include



Circuit diagram of the product



Component description:

L298N:

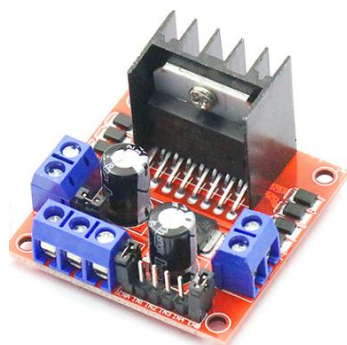
1. Input voltage of logic part: 6 ~ 7V
2. input voltage of drive part vs: 4.8 ~ 46v
3. working current of logic part is: $\leq 36\text{ma}$
4. working current of driving part IO: $\leq 2\text{A}$
5. maximum dissipation power: 25W ($t = 75\text{ }^{\circ}\text{C}$)
6. Control signal input level: high level: $2.3\text{V} \leq V_{\text{IN}} \leq v_{\text{s}}$
low level: $-0.3\text{V} \leq V_{\text{IN}} \leq 1.5\text{V}$
7. working temperature: $-25\text{ }^{\circ}\text{C} \sim +130\text{ }^{\circ}\text{C}$
- 8.driving mode: double channel high power H-bridge drive

How does L298N module work?

- 1.ENA, ENB control enable
- 2.Out1,out2 and out3,out4 controls the motor,
- 3.IN1, IN2, IN3, IN4 connect control level, control positive and negative rotation.

State	ENA	ENB	IN1	IN2	IN3	IN4
Stop	0	0	X	X	X	X
Brake	1	1	0	0	0	0
Forward	1	1	1	0	1	0
Back	1	1	0	1	0	1
Turn Left	1	1	0	0	1	0
Turn Right	1	1	1	0	0	0

Figure:



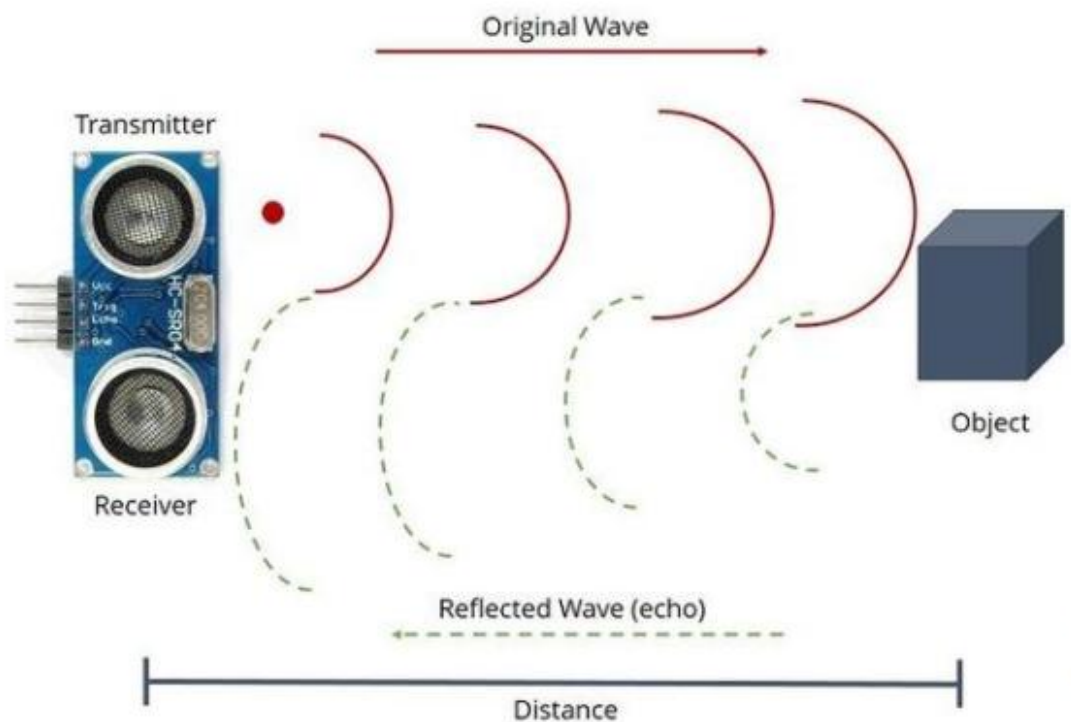
HC-SR04 module:

Electrical parameters	HC-SR04 Ultrasonic module
Working voltage	DC-5V
Working current	15mA
Working frequency	40KHz
Maximum range	4m
Minimum range	2cm
Measuring angle	15°
Input trigger signal	10 US TTL pulse
Output echo signal	Output TTL level signal, proportional to the range
Size	45*20*15

How does ultrasonic obstacle avoidance work?

1. Transmitter (trigger pin) sending signal: high-frequency sound.
2. When a signal finds an object, it is reflected.
3. Receiver (echo pin) : Receives the signal reflected back from it

Figure:



The HC-SR04 ultrasonic sensor detects how far ahead an obstacle has appeared. If there is an obstacle in your specified detection range, the car can make corresponding actions. For example, forward, back, turn left or turn right.

SG90 module:

The control signal enters the signal modulation chip from the channel of the receiver to obtain the DC bias voltage. It has an internal reference circuit, which generates a reference signal with a cycle of 20ms and a width of 1.5ms. The dc bias voltage obtained is compared with the voltage of the potentiometer to obtain the voltage difference output. Finally, the positive and negative output of the voltage difference to the motor driver chip determines the positive and negative rotation

of the motor. When the motor speed is fixed, the potentiometer is driven to rotate by the cascade reduction gear, so that the voltage difference is 0 and the motor stops rotating.

How does SG90 module work?

Control of steering gear:

The control of steering gear generally requires a time base pulse of about 20ms, and the high level part of this pulse is generally the Angle control pulse part within the range of 0.5ms~2.5ms. Taking the servo at an Angle of 180 degrees as an example, the corresponding control relationship is as follows:

0.5 ms ----- 0°

1.0 ms ----- 45°

1.5 ms ----- 90°

2.0 ms ----- 135°

Figure:



Four channel tracking module:

Electrical parameters	Four channel tracking module
Working voltage	DC 3.3V-5V
Working current	1A
working temperature	-10℃ to 50℃
Mounting aperture	M3
detect distance	1mm to 60cm
Size	42*38mm
Output signal	TTL level

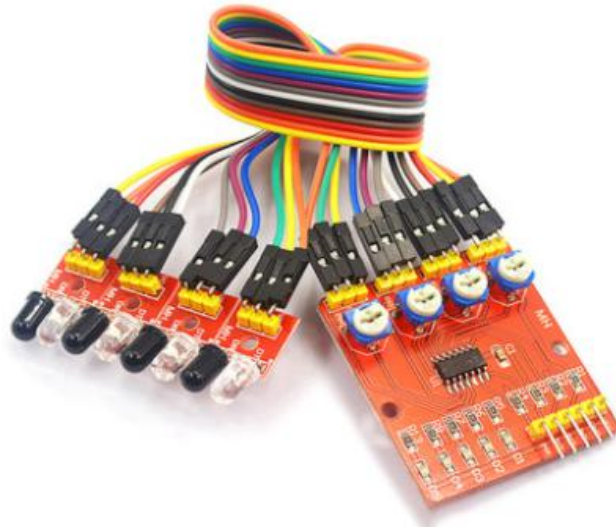
How does Four channel tracking module work?

The sensor belongs to infrared reflection detection, so the reflectivity and shape of the target is the key to the detection range. Among them, the black detection distance is the smallest and the white detection distance is the largest. Small area object distance is small, large area distance is big.

Infrared detection method, that is, the use of infrared light in different colors of the object surface has different reflection intensity characteristics, in the process of the car constantly emitting infrared light to the ground, when the infrared light encounter white paper floor diffuse reflection, reflected light is installed in the car received by the

receiving tube; If a black line is encountered, the infrared light is absorbed and the receiver tube on the car does not receive the infrared light. MCU on whether to receive the reflected infrared light as the basis to determine the location of the black line and car travel route.

Figure:



LM393 module:

The sensor module light is adaptable to the environment, it has a pair of infrared transmitting and receiving tube, tubes, the frequency of ir when detecting direction meet with obstacles (reflecting surface), reflected infrared receiving tube, after the comparator circuit processing, green indicator will light up, at the same time signal output interface and output digital signal (a low level signal), can be adjusted through the potentiometer knob detection distance, effective distance range 2 ~ 30 cm, the working voltage of 3.3 V to 5 V. The detection distance of the sensor can be adjusted by potentiometer, which has the characteristics of small interference, easy assembly and easy use.

How does Infrared detection module work?

1. When the module detects the obstacle signal in front, the green indicator light on the circuit board lights up the level, and the OUT port continuously outputs the low-level signal. The detection distance of the module is 2 ~ 30cm, and the detection Angle is 35 ° .Counterclockwise potentiometer to reduce detection distance.

2. Active infrared reflection detection of the sensor, so the reflectivity and shape of the target are the key to the detection range.Among them, the black detection distance is small, while the white is large.Small area object distance is small, large area distance is big.

3. The output port OUT of the sensor module can be directly connected with IO port of single-chip microcomputer, or it can drive a 5V relay directly;Connection mode:

VCC-VCC; GND-GND; OUT-IO

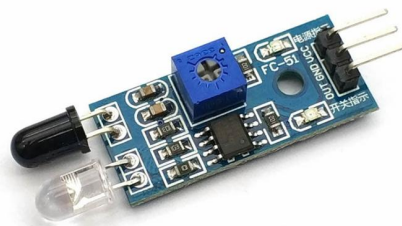
4. The comparator adopts LM393 and works stably;

5. 3-5V DC power supply can be used to supply power to the module.When the power is switched on, the red power indicator lights up.

6. Screw hole of 3mm for easy fixing and installation;

7. Circuit board size: 3.2cm * 1.4cm

Figure:



Code description:

Part 1: Ultrasonic obstacle avoidance program

```
/*
*****
*   TIME:2020.12.9
*   Development Team: Zhiyi Technology Co., Ltd.
*
*****
/
*****
/
*****
*   TIME:2021.12.31
*   Development Team: Zhiyi Technology Co., Ltd.
*
*   *****
/
#include <Servo.h>    //servo library
Servo myservo;       // create servo object to control servo

int Echo = 13;//Pin to D13
int Trig = 12;//Pin to D12

#define ENA 5
#define ENB 6
#define IN1 4
#define IN2 2
#define IN3 3
#define IN4 7

#define carSpeed 130//Set the carSpeed to 160
int Sensor1 = A5;//pin A5
int Sensor2 = A2;//pin A2

int SensorLeft;
int SensorRight;
int rightDistance = 0, leftDistance = 0, middleDistance = 0;

void forward(){//forward function
    analogWrite(ENA, carSpeed);//Set the speed of ENA
    analogWrite(ENB, carSpeed);//Set the speed of ENB
    //The right forward
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    //The left forward
    digitalWrite(IN3, HIGH);
```

```

    digitalWrite(IN4, LOW);

    Serial.println("Forward");
}

void back() { //back function
    analogWrite(ENA, carSpeed); //Set the speed of ENA
    analogWrite(ENB, carSpeed); //Set the speed of ENB
    //The rihgt back
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    //The left back
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    Serial.println("Back");
}

void left() { //left function
    analogWrite(ENA, 200); //Set the speed of ENA
    analogWrite(ENB, 200); //Set the speed of ENB
    //The right forward
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    //The left back
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);

    Serial.println("Left");
}

void right() { //right function
    analogWrite(ENA, 200); //Set the speed of ENA
    analogWrite(ENB, 200); //Set the speed of ENB
    //The rihgt back
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    //The left forward
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);

    Serial.println("Right");
}

void stop() { //stop function

```



```

digitalWrite(ENA, LOW);//Set the speed of ENA to low
digitalWrite(ENB, LOW);//Set the speed of ENB to low
Serial.println("Stop!");
}

float GetDistance()
{
    float distance;
    // Send a low short pulse to Trig to trigger the ranging
    digitalWrite(Trig, LOW); //Send a low level to Trig
    delayMicroseconds(2);
    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);

    distance = pulseIn(Echo, HIGH) / 58.00;

    Serial.print("Distance = ");
    Serial.println(distance);//The serial output distance is converted into cm

    return distance;
}

void setup() {
    myservo.attach(A0,700,2400); // attach servo on pin 3 to servo object
    Serial.begin(9600);
    pinMode(Echo, INPUT);
    pinMode(Trig, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
    pinMode(ENA, OUTPUT);
    pinMode(ENB, OUTPUT);

    stop();
    myservo.write(100); //setservo position according to scaled value
    delay(100);
}

void loop()
{

```

```
SensorLeft = digitalRead(A5);//The sensor on the left  
SensorRight = digitalRead(A2);//The sensor on the Right
```

```
middleDistance = GetDistance();//getDistance();
```

```
if(middleDistance <= 18)
```

```
{
```

```
    stop();
```

```
    delay(500);
```

```
    myservo.write(10);
```

```
    delay(500);
```

```
    rightDistance = GetDistance();//getDistance();
```

```
    delay(500);
```

```
    myservo.write(100);
```

```
    delay(500);
```

```
    myservo.write(180);
```

```
    delay(500);
```

```
    leftDistance = GetDistance();//getDistance();
```

```
    delay(500);
```

```
    myservo.write(100);
```

```
    delay(500);
```

```
    if(rightDistance > leftDistance)
```

```
    {
```

```
        back();
```

```
        delay(300);
```

```
        right();
```

```
        delay(300);
```

```
    }
```

```
    else if(rightDistance < leftDistance)
```

```
    {
```

```
        back();
```

```
        delay(300);
```

```
        left();
```

```
        delay(300);
```

```
    }
```

```
    else
```

```
    {
```

```
        forward();
```

```
    }
```

```
}
```

```
    else
    {
        forward();
    }
    if(!SensorLeft)
    {
        right();
        delay(300);
    }
    else if(!SensorRight)
    {
        left();
        delay(300);
    }
    else
    {
        forward();
    }
}
```

explanation for the program:

When the front side detects the obstacle, the trolley will stop, and the steering engine will rotate left and right to detect the distance between the two sides of the obstacle, judge the distance between the two sides is further, and the trolley will rotate to which side.

Part 2: Four road tracking program

```
/******  
*      TIME:2021.12.31  
*      Development Team: Zhiyi Technology Co., Ltd.  
*  
*      *****/  
#define ENA 5  
#define ENB 6  
#define IN1 4  
#define IN2 2  
#define IN3 3  
#define IN4 7  
  
int Sensor1;  
int Sensor2;  
int Sensor3;  
int Sensor4;  
  
void setup() {  
  
    pinMode(ENA, OUTPUT);  
    pinMode(IN1, OUTPUT);  
    pinMode(IN2, OUTPUT);  
  
    pinMode(ENB, OUTPUT);  
    pinMode(IN3, OUTPUT);  
    pinMode(IN4, OUTPUT);  
  
    pinMode(8, INPUT);  
    pinMode(9, INPUT);  
    pinMode(10, INPUT);  
    pinMode(11, INPUT);  
    Serial.begin(9600);  
}  
  
void loop()  
{  
  
    Sensor1 = digitalRead(8);  
    Sensor2 = digitalRead(9);  
    Sensor3 = digitalRead(10);  
    Sensor4 = digitalRead(11);
```

```

    if ((Sensor4 == LOW || Sensor3 == LOW) && (Sensor2 == HIGH || Sensor1 ==
HIGH))
    {

        analogWrite(ENA, 150);
        analogWrite(ENB, 150);

        digitalWrite(IN1, LOW);
        digitalWrite(IN2, HIGH);

        digitalWrite(IN3,HIGH);
        digitalWrite(IN4, LOW);
        Serial.println("left");
    }

    else if((Sensor4 == HIGH || Sensor3 == HIGH) && (Sensor2 == LOW || Sensor1
== LOW))
    {
        analogWrite(ENA, 150);
        analogWrite(ENB, 150);

        digitalWrite(IN1, HIGH);
        digitalWrite(IN2, LOW);

        digitalWrite(IN3,LOW );
        digitalWrite(IN4, HIGH);
        Serial.println("right");
    }

    else    if(Sensor4 == LOW && Sensor3 == LOW && Sensor2 == LOW && Sensor1
== LOW)
    {
        analogWrite(ENA, 180);
        analogWrite(ENB, 180);
        //FORWARD
        digitalWrite(IN1, HIGH);
        digitalWrite(IN2,LOW);
        digitalWrite(IN3, HIGH);
        digitalWrite(IN4, LOW);
        Serial.println("run");
    }

    else    if((Sensor3 == HIGH || Sensor2 == HIGH ) && Sensor4 == LOW &&
Sensor1 == LOW)

```

```

{
    analogWrite(ENA, 180);
    analogWrite(ENB, 180);
    //FORWARD
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    Serial.println("run");
}
else
{
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
}
}

```

explanation for the program:

When Sensor1 and Sensor2 infrared sensors detect the black line, and Sensor3 and Sensor4 do not detect the black line, then MCU controls the L298N module to make the car turn right.

When Sensor3 and Sensor4 infrared sensors detect the black line, and Sensor1 and Sensor2 do not detect the black line, then MCU controls the L298N module to make the car turn left.

When the black line is detected by Sensor3 and Sensor2 infrared sensors and no black line is detected by Sensor1 and Sensor4, the MCU controls the L298N module to make the car move forward.

When the black line is not detected by Sensor1, Sensor2, Sensor3 and Sensor4 infrared sensors, the MCU controls the L298N module to make the car move back.

Part 3: the Car follow program

```
/******  
*      TIME:2021.12.31  
*      Development Team: Zhiyi Technology Co., Ltd.  
*  
*      *****/  
  
#include <Servo.h>  
  
#define Trig 12 //Pin Trig connects to D12  
#define Echo 13 //Pin Echo connects to D13  
float cm; //Distance variable  
float temp; //  
  
int ENA = 5;  
int IN1 = 4;  
int IN2 = 2;  
  
int ENB = 6;  
int IN3 = 3;  
int IN4 = 7;  
  
int Sensor1 = A5;//pin A5  
int Sensor2 = A2;//pin A2  
  
int SensorLeft;  
int SensorRight;  
  
Servo myservo;  // create servo object to control a servo  
// twelve servo objects can be created on most boards  
  
void setup()  
{  
    myservo.attach(A0);  // attaches the servo on pin 9 to the servo object  
    //Set the pin mode  
    pinMode(ENA, OUTPUT);  
    pinMode(IN1, OUTPUT);  
    pinMode(IN2, OUTPUT);  
  
    pinMode(ENB, OUTPUT);  
    pinMode(IN3, OUTPUT);  
    pinMode(IN4, OUTPUT);  
  
    pinMode(Sensor1, INPUT);
```

```

    pinMode(Sensor2, INPUT);
    pinMode(Trig, OUTPUT);
    pinMode(Echo, INPUT);
    Serial.begin(9600);
    myservo.write(100);
}

void loop()
{
    cm = GetDistance();

    SensorLeft = digitalRead(A5); //The sensor on the left
    SensorRight = digitalRead(A2); //The sensor on the Right

    //If the sensor on the right detects the signal and the sensor on the left does
    not
    //and the distance of the obstacle detected by the ultrasonic is less than 10cm
    and greater than 5cm
    //turn right
    if (SensorLeft == HIGH && SensorRight == LOW && ( cm > 5 && cm < 10))
    {
        right();
    }
    //If the sensor on the left detects the signal and the sensor on the right does
    not
    //and the distance of the obstacle detected by the ultrasonic is less than 10cm
    and greater than 5cm
    //turn left
    else if(SensorLeft == LOW && SensorRight == HIGH&& ( cm >5 && cm <10))
    {
        left();
    }
    else if(SensorLeft == HIGH && SensorRight == HIGH&&( cm >11 && cm <20))
    {
        forward();
    }
    else if(SensorLeft == HIGH && SensorRight == HIGH && ( cm >20 || cm <3))
    {
        stop();
    }
    else if((SensorLeft == LOW && SensorRight == LOW) || ( cm <5))
    {
        back();
    }
}

```

```

    if (SensorLeft == LOW)
    {
        myservo.write(160);
    }
    else if (SensorRight == LOW)
    {
        myservo.write(40);
    }
    else
    {
        myservo.write(100);
    }
}

/*
Function: obtain ultrasonic sensor ranging data
Parameters: Trig, Echo
Parameter description: sensor connected to the motherboard pin port A4,A5
Trig -----> pin D12
Echo -----> pin D13
*/
float GetDistance()
{
    float distance;
    // Send a low short pulse to Trig to trigger the ranging
    digitalWrite(Trig, LOW); //Send a low level to Trig
    delayMicroseconds(2);
    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);

    distance = pulseIn(Echo, HIGH) / 58.00;

    Serial.print("Distance = ");
    Serial.println(distance); //The serial output distance is converted into cm

    return distance;
}

//turn left
void left(){
    //Set the speed of the motor

```

```

        analogWrite(ENA, 60);
        analogWrite(ENB, 160);
        //The right forward
        digitalWrite(IN1, LOW);
        digitalWrite(IN2, LOW);
        //The left STOP
        digitalWrite(IN3, HIGH);
        digitalWrite(IN4, LOW);
        Serial.println("left");
    }
//turn right
void right(){
    //Set the speed of the motor
    analogWrite(ENA, 160);
    analogWrite(ENB, 60);
    //The right stop
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    //The left forward
    digitalWrite(IN3,LOW);
    digitalWrite(IN4,LOW);
    Serial.println("right");
}
//turn forward
void forward(){
    //FORWARD
    //Set the speed of the motor
    analogWrite(ENA, 130);
    analogWrite(ENB, 130);
    //The right forward
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2,LOW);
    //The left forward
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    Serial.println("forward");
}

//stop
void stop(){
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
    //set IN1,IN2,IN3,IN4 to LOW
    digitalWrite(IN1,LOW);

```

```

    digitalWrite(IN2,LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    Serial.println("stop");

}
//back up
void back(){
    //Set the speed of the motor
    analogWrite(ENA, 110);
    analogWrite(ENB, 110);
    //The right back
    digitalWrite(IN1, LOW);
    digitalWrite(IN2,HIGH);
    //The left back
    digitalWrite(IN3,LOW);
    digitalWrite(IN4, HIGH);
    Serial.println("back");

}

```

explanation for the program:

The realization of the function of car following module requires the combination of two infrared detection modules and ultrasonic module. When the sensor on the left detects the signal, while the sensor on the right does not, and the ultrasonic detects the distance of the obstacle in front is less than 10cm and greater than 5cm, the car turns left.

When the sensor on the right detects the signal while the sensor on the left does not, and the ultrasonic detects the distance of the obstacle in front is less than 10cm and greater than 5cm, the car turns right.

When the ultrasonic detects the distance of the obstacle in front is less than 25cm and greater than 15cm, the trolley will move forward. When the signal is detected by both sensors and the distance of the obstacle in front is less than 3cm detected by the ultrasonic wave, the car backs away.

When no signal is detected by sensors on both sides and the distance of the obstacle in front is greater than 30cm or less than 3cm, the car stops.