

6.6 Avoid mode

1. Learning goal:

Learn about the how to process the data acquired by the infrared sensor to allow the car to avoid obstacles.

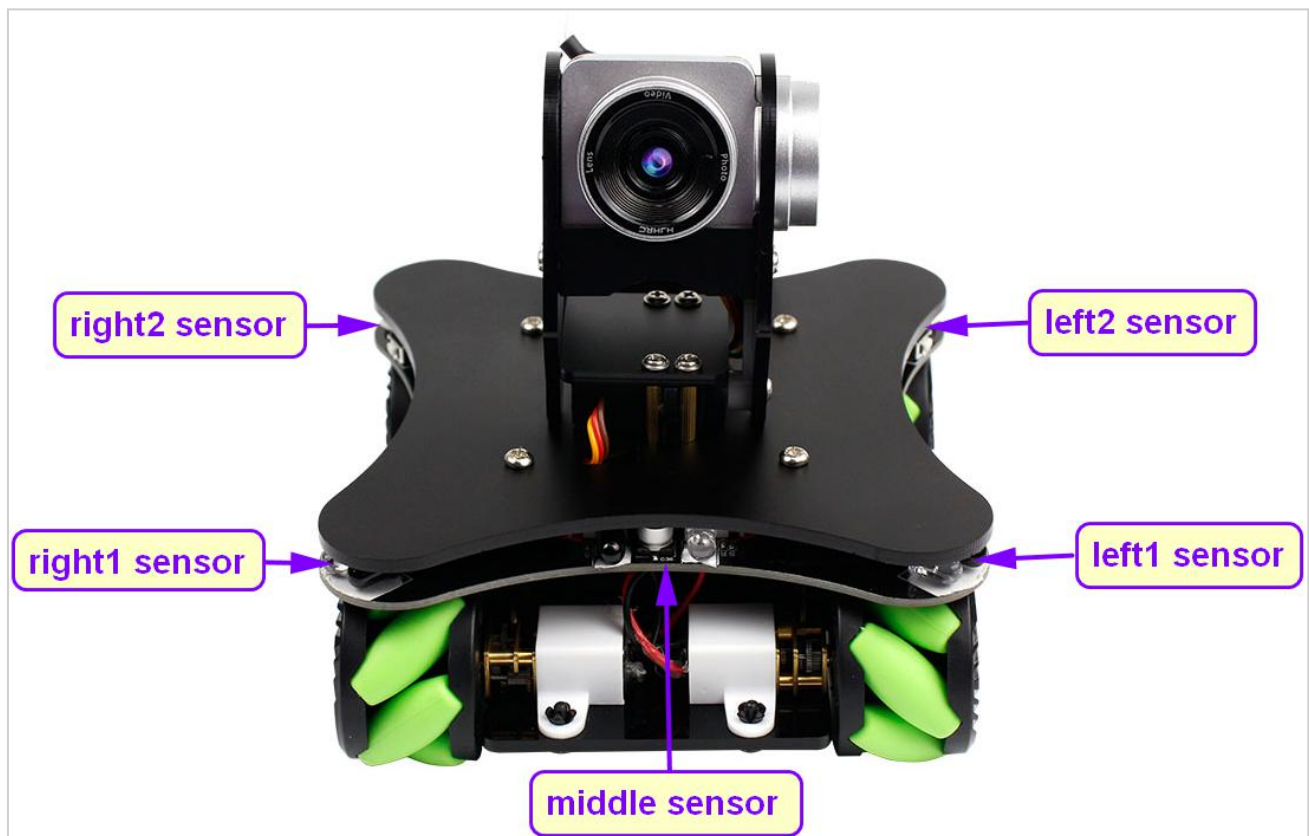
2. Experimental phenomena:

After power is turned on, when we press K1 button, buzzer will whistle. Then, car start avoid mode.

3. About infrared sensor

This course mainly uses infrared obstacle avoidance sensors(infrared pair tubes), white is sent tube, black is receive tube.

When an obstacle is detected, the closer the obstacle is to the robot car, the smaller the analog value returned by the infrared sensor.



4. Code analysis

4.1 `ir_rgb()` function is that when the infrared sensor detects an obstacle, the corresponding RGB light next to it lights up red.

```

void ir_rgb()
{
    //Read IR sensor if detect obstacle
    dig_ir_l1 = digitalRead(IR_SENSOR_L1);
    dig_ir_l2 = digitalRead(IR_SENSOR_L2);
    dig_ir_r1 = digitalRead(IR_SENSOR_R1);
    dig_ir_r2 = digitalRead(IR_SENSOR_R2);

    //If sensor detected the obstacle,the RGB li
    if (dig_ir_l1 == LOW)
        showRGB(3, 255, 0, 0);
    else
        showRGB(3, 0, 0, 0);

    if (dig_ir_l2 == LOW)
        showRGB(0, 255, 0, 0);
    else
        showRGB(0, 0, 0, 0);

    if (dig_ir_r1 == LOW)
        showRGB(2, 255, 0, 0);
    else
        showRGB(2, 0, 0, 0);

    if (dig_ir_r2 == LOW)
        showRGB(1, 255, 0, 0);
    else
        showRGB(1, 0, 0, 0);
}

```

4.2 The main function of the turn_direction() function is to control the car turn direction. The parameter degree=1 is spin right about 90 degrees, degree=2 is about spin right 180 degrees, degree=-1 is spin left about 90 degrees, and degree=-2 is spin left about 180 degrees.

This method is a fixed write value. It is possible that the rotation angles of different environments of different motors are inconsistent, so we can only control the general direction of the rotation, not precisely control the angle.

```

void turn_direction(int degree)
{
    switch (degree)
    {
        case 1:
            spin_right(65);
            delay(200);
            brake();
            delay(1);
            break;
        case 2:
            spin_right(65);
            delay(380);
            brake();
            delay(1);
            break;
        case -1:
            spin_left(65);
            delay(200);
            brake();
            delay(1);
            break;
        case -2:
            spin_left(65);
            delay(380);
            brake();
            delay(1);
            break;

        default:
            break;
    }
}

```

4.3 The main obstacle() function. Read obstacle sensor reads the analog values of the five sensors, and according to the values returned by the sensor, the car is avoided by obstacles according to multiple logic judgments. This kind of logic judges the way to modify the motion state of the car is not unique. The user can modify it according to the actual situation to make the obstacle avoidance effect better.

```

void obstacle()
{
    //Read IR sensor data
    dig_ir_l1 = digitalRead(IR_SENSOR_L1);
    dig_ir_l2 = digitalRead(IR_SENSOR_L2);
    dig_ir_r1 = digitalRead(IR_SENSOR_R1);
    dig_ir_r2 = digitalRead(IR_SENSOR_R2);
    ir_Mid = analogRead(IR_SENSOR_MID) / 10;
    ir_L1 = analogRead(IR_SENSOR_L1) / 10;
    ir_L2 = analogRead(IR_SENSOR_L2) / 10;
    ir_R1 = analogRead(IR_SENSOR_R1) / 10;
    ir_R2 = analogRead(IR_SENSOR_R2) / 10;

    //According to the situation detected by the sensor, the corresponding o
    if (ir_Mid < 80 && ir_L1 < 30 && ir_R1 < 30 && ir_R2 < 30)
    {
        turn_direction(-1);
    }
    else if (ir_Mid < 80 && ir_L1 < 30 && ir_R1 < 30 && ir_L2 < 30)
    {
        turn_direction(1);
    }
    else if (ir_Mid < 80 && ir_L1 < 40 && ir_R1 < 40)
    {
        back(CarSpeedControl);
        delay(100);
        turn_direction(2);
    }
    else if (ir_L1 < 40 && ir_R1 < 40)
    {
        run(CarSpeedControl);
        delay(100);
    }
    else if (ir_L1 < 40 && ir_L2 < 40)
    {
        right(CarSpeedControl);
    }
    else if (ir_Mid < 80 && ir_L1 < 40 && ir_L2 < 40)
    {

```

```

    right(CarSpeedControl);
}
else if (ir_R1 < 40 && ir_R2 < 40)
{
    left(CarSpeedControl);
}
else if (ir_Mid < 80 && ir_R1 < 40 && ir_R2 < 40)
{
    left(CarSpeedControl);
}
else if (ir_Mid < 80 && dig_ir_ll == LOW && dig_ir_rl == HIGH)
{
    spin_right(CarSpeedControl);
}
else if (ir_Mid < 80 && dig_ir_ll == HIGH && dig_ir_rl == LOW)
{
    spin_left(CarSpeedControl);
}
else if (dig_ir_ll == HIGH && dig_ir_rl == LOW)
{
    spin_left(CarSpeedControl);
}
else if (dig_ir_ll == LOW && dig_ir_rl == HIGH)
{
    spin_right(CarSpeedControl);
}
else if (ir_Mid < 90)
{
    back(CarSpeedControl);
    delay(100);
    turn_direction(2);
}
else //In other cases, the car keeps going straight.
{
    run(CarSpeedControl);
}
}

```

4.4 In the loop() function scan button, when the button state is true(press K1 button for the first time),the buzzer whistle once, the car is started to avoid obstacles. When the button state is false(press K1 button for the second time), the buzzer whistle once, the car exit avoid mode.

```

void loop()
{
  keyscan();
  if (button_press)
  {
    ir_rgb();
    obstacle(); // start avoid mode
  }
  else
  {
    brake();
    clearRGB();
  }
}
}

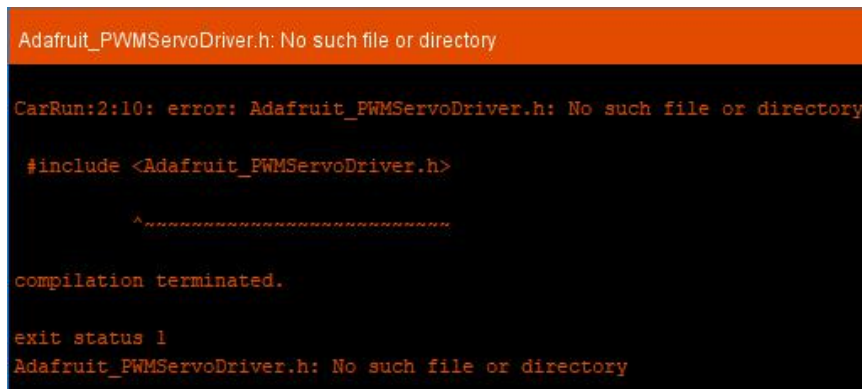
```

5. Compiling and downloading code

5.1 Open the **Obstacle.ino** program , select the serial port and click upload directly (the Omniduino car must first be connected to the computer via the USB data cable).

5.2 If there is an error like the following, it means that the library file is missing. Please copy the library file provided by the omniduino omnibus to the library file directory compiled by arduinoIDE.

please refer to 【3.Development Environment Construction】 ---- 【3.4 Add additional library files】



```

Adafruit_PWMServoDriver.h: No such file or directory

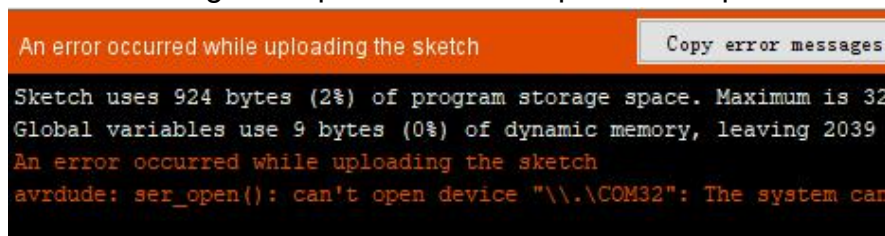
CarRun:2:10: error: Adafruit_PWMServoDriver.h: No such file or directory
#include <Adafruit_PWMServoDriver.h>
          ^~~~~~

compilation terminated.

exit status 1
Adafruit_PWMServoDriver.h: No such file or directory

```

5.3 If the compilation passes normally, but the following error occurs during uploading, the reason may be that the wrong serial port or the serial port is occupied.



```

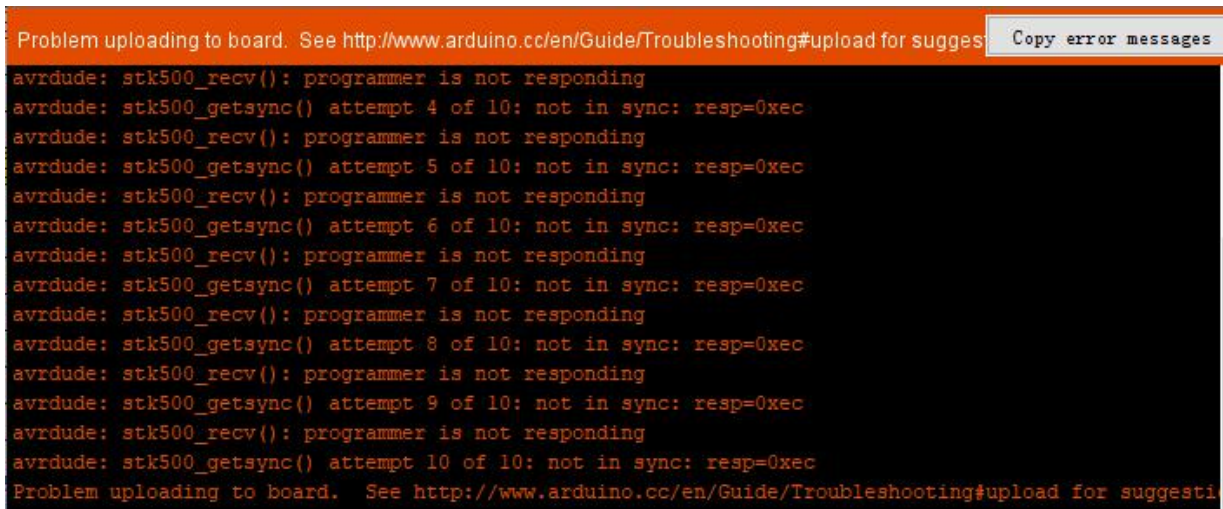
An error occurred while uploading the sketch
Sketch uses 924 bytes (2%) of program storage space. Maximum is 32
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039
An error occurred while uploading the sketch
avrdude: ser_open(): can't open device "\\.\COM32": The system can

```

Solution: Open the device manager to see if there is a serial port with CH340 tag. If not, please restart the Omniduino car, then, re-plug the USB cable or replace a USB cable; If there is a serial port number, we need to close the other serial port or assistant software, avoid serial port occupation, and then re-select the serial port to ArduinoIDE 【Tool】 --> 【Port】.

5.4 After clicking the upload button, the upload is always displayed, but it can't be uploaded

successfully for a long time.

A screenshot of an Arduino IDE terminal window. The window has an orange title bar with the text "Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting#upload for sugges" and a button labeled "Copy error messages". The terminal text is as follows:

```
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 4 of 10: not in sync: resp=0xec
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 5 of 10: not in sync: resp=0xec
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 6 of 10: not in sync: resp=0xec
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 7 of 10: not in sync: resp=0xec
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 8 of 10: not in sync: resp=0xec
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 9 of 10: not in sync: resp=0xec
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 10 of 10: not in sync: resp=0xec
Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting#upload for suggesti
```

Because the uploading program and the WIFI camera communication is realized through the serial port, when the serial port is occupied by the WIFI camera, and the program cannot be uploaded.

Solution:

- ①Unplug the USB cable, turn off the power of the car, wait for the D2 indicator to go out.
- ②Then, plug in the USB data cable. At this time, your mobile phone should not connect the WiFi signal of the car.
- ③You can upload the program to the car according to the normal steps.
- ④After the program is successfully uploaded, unplug the USB data cable, open the power switch of the car. The corresponding experimental phenomenon will appear.

(Tip: If you upload APP control program. After the program is successfully uploaded, unplug the USB data cable, open the power switch of the car. Mobile phone connect the car to the WIFI signal, and then open the APP to control.)