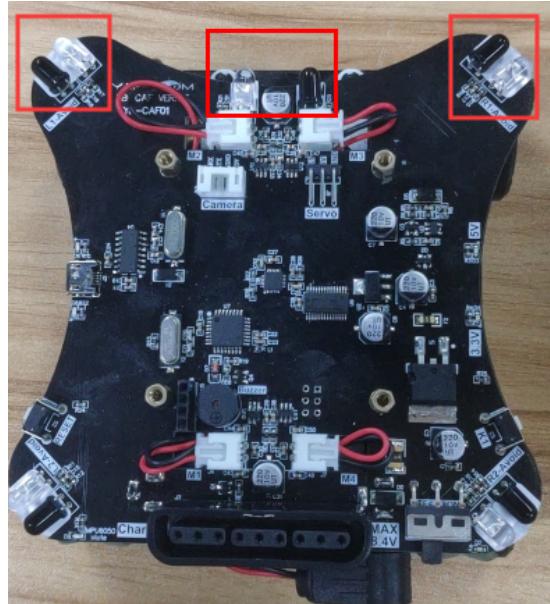


6.7 Car Around mode

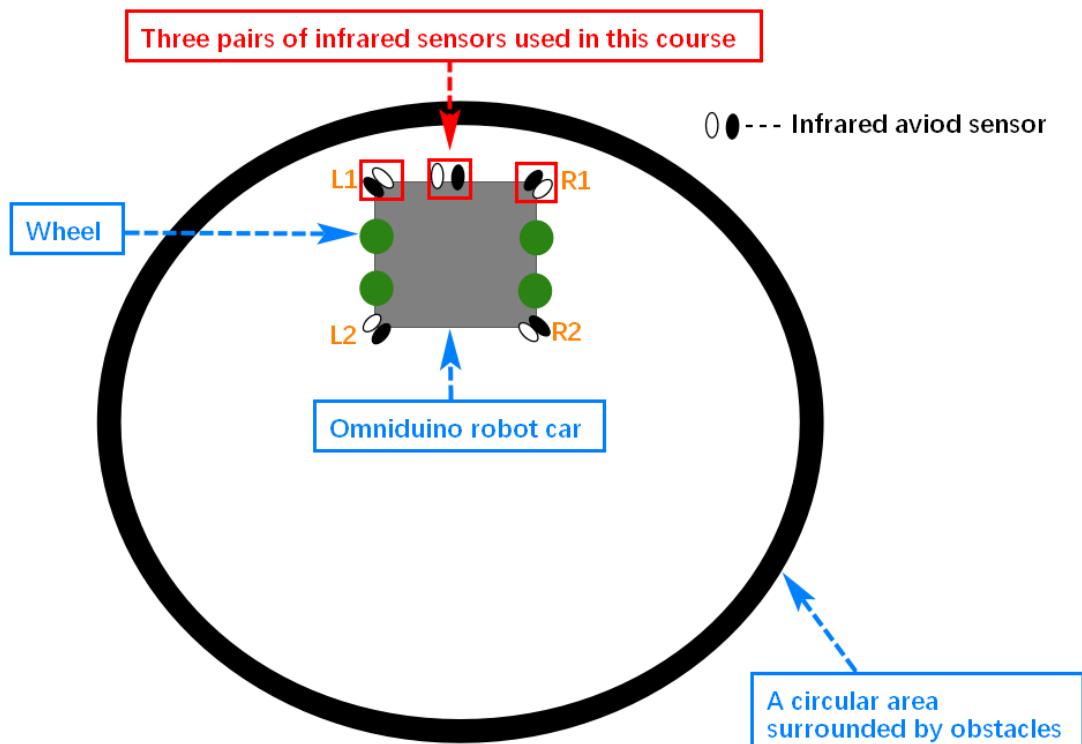
1. Learning goal:

As shown in the figure below, the red wire frame contains the three pairs of infrared sensors used in this course.

According to the distance between the L1, R1, front infrared sensor and the obstacle, adjust the direction of the robot car to make the robot car can walk around the circular inner wall of the obstacle.



As shown below:



2. Experimental phenomena:

Turn on the power switch of the car. When the K1 button is pressed, the buzzer whistle once, then the car starts to go left and do a circular motion along the circular inner wall.

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 The main function of the carRun() is to input four motor speed values to control the movement of the car.

```
void carRun(int speed_L1, int speed_R1, int speed_L2, int speed_R2)
{
    speed_L1 = map(speed_L1, -160, 160, -2560, 2560); //map 160 to 2560
    speed_R1 = map(speed_R1, -160, 160, -2560, 2560);
    speed_L2 = map(speed_L2, -160, 160, -2560, 2560);
    speed_R2 = map(speed_R2, -160, 160, -2560, 2560);

    int speed_wheel[4] = {speed_R1, speed_L1, speed_L2, speed_R2};

    for (int i = 0; i < 4; i++)
    {
        if (speed_wheel[i] >= 0)
        {
            pwm.setPWM(wheel[i][0], 0, speed_wheel[i]);
            pwm.setPWM(wheel[i][1], 0, 0);
        }
        else
        {
            pwm.setPWM(wheel[i][0], 0, 0);
            pwm.setPWM(wheel[i][1], 0, -speed_wheel[i]);
        }
    }
}
```

4.2 The main function of the carSurround () is that the car moves to the left along the circular inner wall, speed is the initial speed, direction is the direction, 0 is to the left, and 1 is to the right.

```

void carSurround(int speed, int direction)
{
    int speed_L1 = speed * 1.5;
    int speed_L2 = speed;
    int speed_R1 = speed * 1.5;
    int speed_R2 = speed;

    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;
    ir_Mid = analogRead(IR_SENSOR_MID) / 10;
    int pid_mid = PIDCal_IR(PID_MID, ir_Mid);

    if (direction == 0)           // left
    {
        if (dig_ir_l1 == LOW && dig_ir_r1 == HIGH)
        {
            speed_L1 = speed_L1 - ir_L1;
            speed_L2 = speed_L2 + ir_L1 - 40;
            speed_R1 = speed_R1 + ir_R1 - 40;
            speed_R2 = speed_R2 - ir_R1 + 40;
        } else if (dig_ir_l1 == HIGH && dig_ir_r1 == LOW)
        {
            speed_L1 = speed_L1 - ir_L1 + 40;
            speed_L2 = speed_L2 - ir_L1 + 40;
            speed_R1 = speed_R1 + ir_R1 - 40;
            speed_R2 = speed_R2 + ir_R1;
        } else if (dig_ir_l1 == LOW && dig_ir_r1 == LOW)
        {
            speed_L1 = speed_L1 - ir_L1 + 43;
            speed_L2 = speed_L2 + ir_L1 - 43;
            speed_R1 = speed_R1 + ir_R1 - 43;
            speed_R2 = speed_R2 - ir_R1 + 43;
        }
        carRun(-speed_L1 + pid_mid, speed_R1 + pid_mid, speed_L2 + pid_mid, -speed_R2 + pid_mid);
    }
    else if (direction == 1)      // right
    {
        if (dig_ir_l1 == LOW && dig_ir_r1 == HIGH)
        {
            speed_L1 = speed_L1 + ir_L1;
            speed_L2 = speed_L2 - ir_L1 + 40;
            speed_R1 = speed_R1 - ir_R1 + 40;
            speed_R2 = speed_R2 + ir_R1 - 40;
        } else if (dig_ir_l1 == HIGH && dig_ir_r1 == LOW)
        {
            speed_L1 = speed_L1 + ir_L1 - 40;
            speed_L2 = speed_L2 + ir_L1 - 40;
            speed_R1 = speed_R1 - ir_R1 + 40;
            speed_R2 = speed_R2 - ir_R1;
        } else if (dig_ir_l1 == LOW && dig_ir_r1 == LOW)
        {
            speed_L1 = speed_L1 + ir_L1 - 43;
            speed_L2 = speed_L2 - ir_L1 + 43;
            speed_R1 = speed_R1 - ir_R1 + 43;
            speed_R2 = speed_R2 + ir_R1 - 43;
        }
        carRun(speed_L1 + pid_mid, -speed_R1 + pid_mid, -speed_L2 + pid_mid, speed_R2 + pid_mid);
    }
}

```

4.3 Increase the PID calculation PID_MID, adjust the distance between the car and the inner wall to keep the car running more stable.

```

typedef struct
{
    float SetPoint;    //Desired value
    float Proportion; // Proportional Const
    float Integral;   // Integral Const
    float Derivative; // Derivative Const
    float LastError;  // Error[-1]
    float PrevError;  // Error[-2]
    float SumError;   // Sums of Errors
} PID;

/*=====
PID Calculation part
=====

PID PID_MID = {60, 0.5, 0, 0, 0, 0, 0};
PID PID_IR = {40, 1.2, 0, 0, 0, 0, 0};
float PIDCal_IR(PID pid, float nowValue);

float PIDCal_IR(PID pid, float nowValue)
{
    float dError, Error;
    Error = pid.SetPoint - nowValue;
    pid.SumError += Error;
    dError = pid.LastError - pid.PrevError;
    pid.PrevError = pid.LastError;
    pid.LastError = Error;

    double val_sensor = pid.Proportion * Error
        + pid.Integral * pid.SumError
        + pid.Derivative * dError;
    return -val_sensor;
}

```

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

```

void loop()
{
    keyscan();
    if (button_press)
    {
        ir_rgb();
        carSurround(CarSpeedControl, 0);
    }
    else
    {
        brake();
        clearRGB();
    }
}

```

5. Compiling and downloading code

5.1 Open the **Surround.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 Copy error messages

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.

```
Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting#upload for suggestions
```

```
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 4 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 5 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 6 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 7 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 8 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 9 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: 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 suggestions
```

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.)