

Infrared obstacle avoidance

The purpose of the experiment:

Debug potentiometers SW1 and SW2 according to the debug diagram at the end of the article. Turn on the power switch of the BatCar, press the start button K1, and when you hear the short whistle, the BatCar avoids the obstacle.

List of components required for the experiment:

BatCar*1

USB data cable*1



Experimental code analysis:

```
int Left_motor_back = 9;
int Left_motor_go = 5;
int Right_motor_go = 6;
int Right_motor_back = 10;
int Right_motor_en = 8;
int Left_motor_en = 7;
/*Set Button port*/
int key=4;
/*Set BUZZER port*/
int beep=3;
/*Line Walking*/
const int SensorRight = A3; // Set Right Line Walking Infrared sensor port
const int SensorLeft = A2; // Set Left Line Walking Infrared sensor port
int SL; // State of Left Line Walking Infrared sensor
int SR; // State of Right Line Walking Infrared sensor
/*Infrared obstacle avoidance*/
const int SensorRight_2 = A4; // Right Infrared sensor
const int SensorLeft_2 = A5; // Left Infrared sensor
int SL_2; // State of Left Infrared sensor
int SR_2; // State of Right Infrared sensor
```

```

void setup()
{
  //Initialize motor drive for output mode
  pinMode(Left_motor_go,OUTPUT);
  pinMode(Left_motor_back,OUTPUT);
  pinMode(Right_motor_go,OUTPUT);
  pinMode(Right_motor_back,OUTPUT);
  pinMode(key,INPUT);// Set button as input
  pinMode(beep,OUTPUT);// Set buzzer as output
  pinMode(SensorRight, INPUT); // Set Right Line Walking Infrared sensor as
input
  pinMode(SensorLeft, INPUT); // Set left Line Walking Infrared sensor as input
  pinMode(SensorRight_2, INPUT); //Set Right Infrared sensor as input
  pinMode(SensorLeft_2, INPUT); //Set left Infrared sensor as input
  digitalWrite(key,HIGH);//Initialize button
  digitalWrite(beep,HIGH);// set buzzer mute
}
//=====================================================Motor=====
void run()
{
  digitalWrite(Right_motor_go,HIGH);// right motor go ahead
  digitalWrite(Right_motor_back,LOW);
  analogWrite(Right_motor_go,100);//PWM--Pulse Width Modulation(0~255). It
can be adjusted to control speed.
  analogWrite(Right_motor_back,0);
  digitalWrite(Left_motor_go,HIGH);// set left motor go ahead
  digitalWrite(Left_motor_back,LOW);
  analogWrite(Left_motor_go,100);//PWM--Pulse Width Modulation(0~255). It
can be adjusted to control speed.
  analogWrite(Left_motor_back,0);
}
void brake() //stop
{
  digitalWrite(Right_motor_go,LOW);
  digitalWrite(Right_motor_back,LOW);
  digitalWrite(Left_motor_go,LOW);
}

```

```

    digitalWrite(Left_motor_back,LOW);
}
void left()//turn left
{
    digitalWrite(Right_motor_go,HIGH); // right motor go ahead
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,100);
    analogWrite(Right_motor_back,0);// PWM--Pulse Width Modulation(0~255)
    control speed
    digitalWrite(Left_motor_go,LOW);    // left motor stop
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,0);// PWM--Pulse Width Modulation(0~255)
    control speed
}
void spin_left(int time)    //Left rotation
{
    digitalWrite(Right_motor_go,HIGH); // right motor go ahead
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,100); // PWM--Pulse Width Modulation(0~255)
    control speed
    analogWrite(Right_motor_back,0);
    digitalWrite(Left_motor_go,LOW);    // left motor back off
    digitalWrite(Left_motor_back,HIGH);
    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,100); // PWM--Pulse Width Modulation(0~255)
    control speed
    delay(time * 100);
}
void right() //turn right
{
    digitalWrite(Right_motor_go,LOW);    // right motor stop
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,0);
    analogWrite(Right_motor_back,0);
    digitalWrite(Left_motor_go,HIGH);// left motor go ahead
    digitalWrite(Left_motor_back,LOW);

```

```

    analogWrite(Left_motor_go,100);
    analogWrite(Left_motor_back,0);// PWM--Pulse Width Modulation(0~255)
control speed
}
void spin_right(int time)    //Right rotation
{
    digitalWrite(Right_motor_go,LOW);  // right motor back off
    digitalWrite(Right_motor_back,HIGH);
    analogWrite(Right_motor_go,0);
    analogWrite(Right_motor_back,200);// PWM--Pulse Width Modulation(0~255)
control speed
    digitalWrite(Left_motor_go,HIGH);// left motor go ahead
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,200);
    analogWrite(Left_motor_back,0);// PWM--Pulse Width Modulation(0~255)
control speed
    delay(time * 100);
}
void back(int time) //back off
{
    digitalWrite(Right_motor_go,LOW); //right motor back off
    digitalWrite(Right_motor_back,HIGH);
    analogWrite(Right_motor_go,0);
    analogWrite(Right_motor_back,150);// PWM--Pulse Width Modulation(0~255)
control speed
    digitalWrite(Left_motor_go,LOW); //left motor back off
    digitalWrite(Left_motor_back,HIGH);
    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,150);// PWM--Pulse Width Modulation(0~255)
control speed
    delay(time * 100);
}
//=====================================================
void keysachn()
{
    int val;
    val=digitalRead(key);// Reads the button ,the level value assigns to val

```

```

while(digitalRead(key))// When the button is not pressed
{
    val=digitalRead(key);
}
while(!digitalRead(key))// When the button is pressed
{
    delay(10); //delay 10ms
    val=digitalRead(key);// Reads the button ,the level value assigns to val
    if(val==LOW) //Double check the button is pressed
    {
        digitalWrite(beep,LOW);//The buzzer sounds
        delay(50);//delay 50ms
        while(!digitalRead(key)) //Determine if the button is released or not
        digitalWrite(beep,HIGH);//mute
    }
    else
        digitalWrite(beep,HIGH);//mute
    }
}
/*main loop*/
void loop()
{
    keysacn();//Press the button to start
    while(1)
    {
        /*****
        Infrared signal back means there is something obstructed ,returns low level
        and led lights up.
        Infrared signal gone means there is nothing obstructed ,returns high level and
        led lights off.
        *****/

        SR_2 = digitalRead(SensorRight_2);//Right infrared sensor detects the
        obstacle,then LED[L5] light illuminates and otherwise it goes off.

        SL_2 = digitalRead(SensorLeft_2);//Left infrared sensor detects the
        obstacle,then LED[L4] light illuminates and otherwise it goes off.
    }
}

```

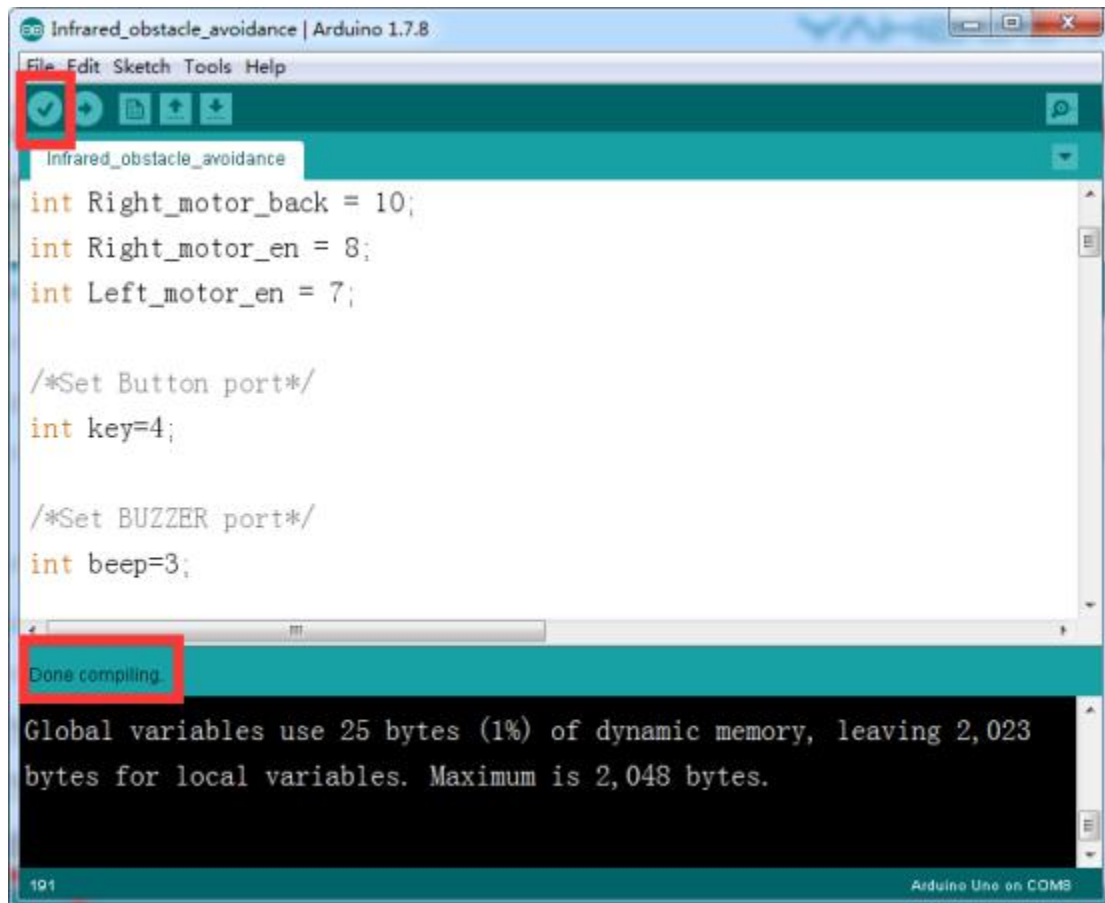
```

    if (SL_2 == HIGH && SR_2 == HIGH) // There is nothing obstructed ,goes
    ahead.
        run();
    else if (SL_2 == HIGH & SR_2 == LOW) // There is something obstructed on
    the right then LED[L4] light illuminates, turns left.
        left();
    else if (SR_2 == HIGH & SL_2 == LOW) // There is something obstructed on
    the left then LED[L4] light illuminates, turns right.
        right();
    else // // There is something obstructed, back off and adjust direction.
    {
        back(6); // back off for 600ms
        spin_right(3); // Right rotation for 300ms to adjust direction
    }
}
}
}

```

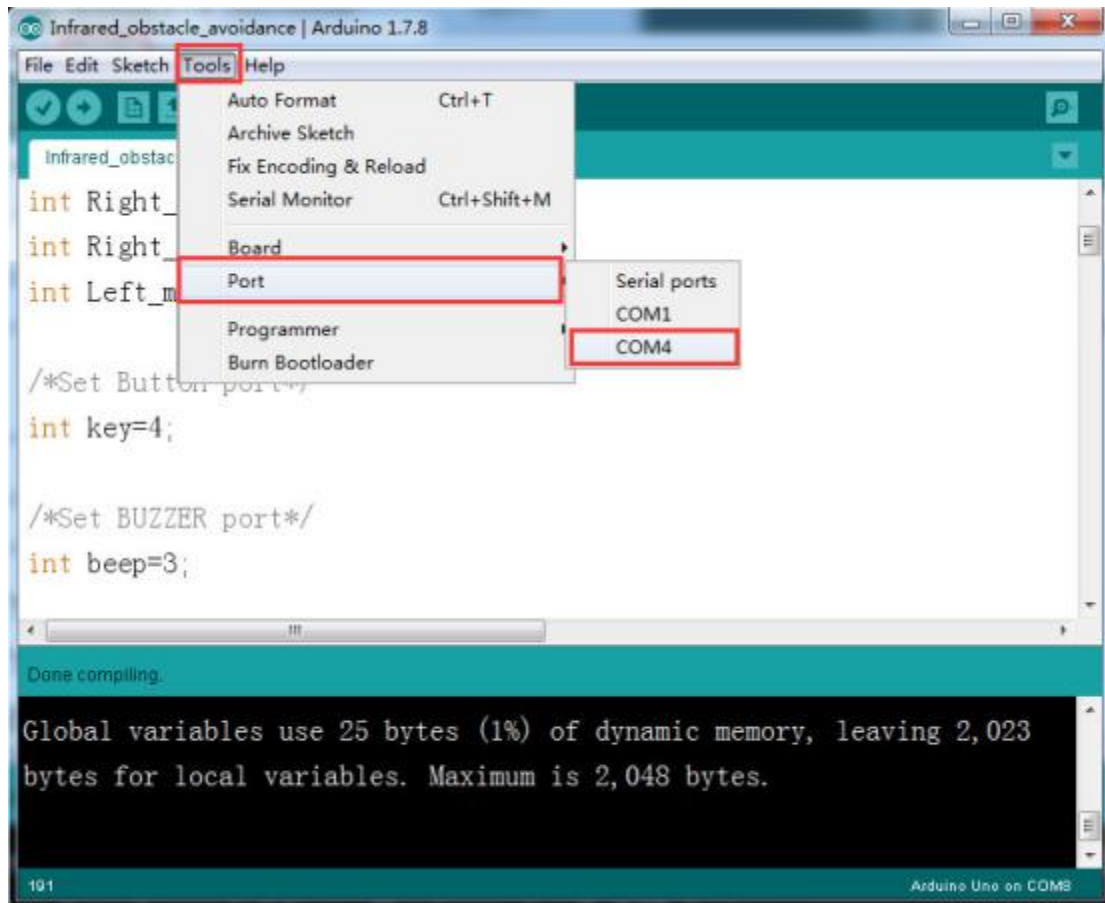
Experimental steps:

1. We need to open the code of this experiment: **Infrared_obstacle_avoidance.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.

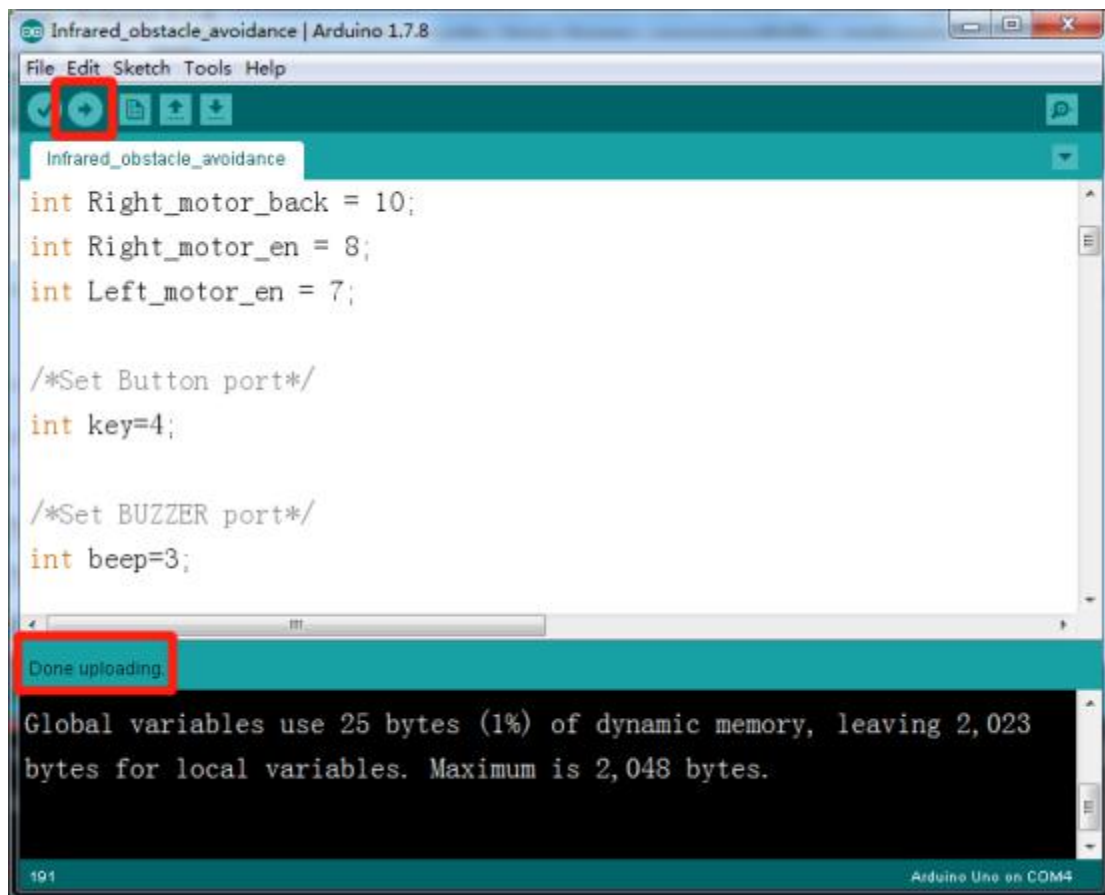


2. In the menu bar of Arduino IDE, we need to select **【Tools】** --- **【Port】** --- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.





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.



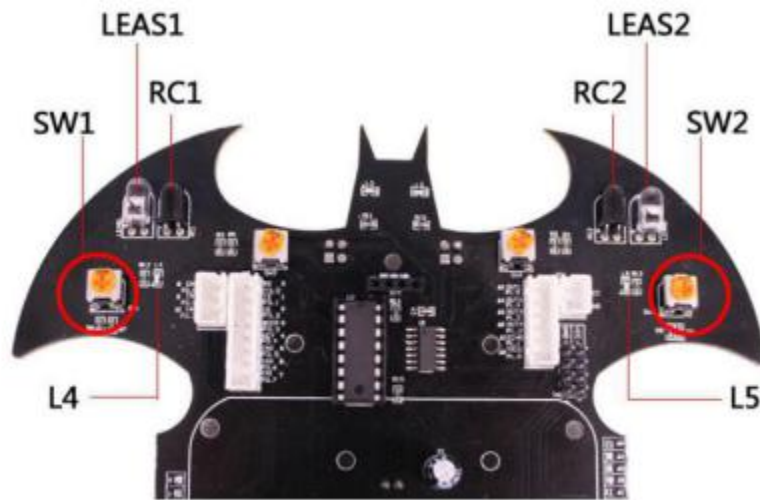
4. After the program is successfully uploaded, debug the BatCar as shown below.

Debugging:

① Adjust potentiometer [SW1] to make the infrared light-emitting diode [LEAS1] and infrared light-receiving diode [RC1] away from obstacle less than 10 cm, then LED light [L4] illuminates, otherwise, it goes off.

② Adjust potentiometer [SW2] to make the infrared light-emitting diode [LEAS2] and infrared light-receiving diode [RC2] away from obstacle less than 10 cm, then LED light [L5] illuminates, otherwise, it goes off.

Caution : Don't excessively rotate potentiometer while adjusting. It should be within 30°.



5. Place the BatCar in an open space and place some cartons as obstacles. Turn on the BatCar's power switch, press the start button K1, and after hearing the short whistle, BatCar begins to avoid obstacles.