# Lesson 14 Line Following Robot Car

**Table**

[1. What do you need to prepare 1](#_Toc31195)

[2. Control logic 2](#_Toc6359)

[3. Upload the code 5](#_Toc2731)

[4. Troubleshooting 6](#_Toc3459)

[5. Code 6](#_Toc28146)

[6. Any questions and suggestions are welcome 12](#_Toc31734)

We have learned the relevant knowledge of Line Tracking module and the control logic based on Arduino.

Now let's write code to control the car to automatically drive along the black trace.

# 1. What do you need to prepare

|  |  |  |  |
| --- | --- | --- | --- |
| **Components** | **Quantity** | **Picture** | **Remark** |
| USB Cable | 1 |  |  |
| PC | 1 |  | Prepared by yourself |
| Black tape | 1 |  | Prepared by yourself |
| Smart Robot Car | 1 |  |  |

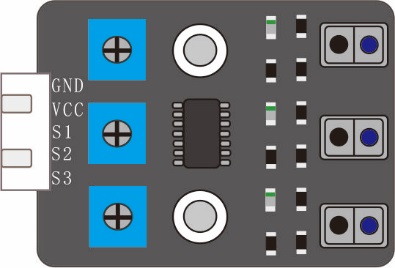
# 2. Control logic

There are three Reflective Optical Sensors on Line Tracking Module. When the infrared light emitted by infrared diode shines on the surface of different objects, the sensor will receive light with different intensities after reflection.

As we know, black objects absorb light better. So when black lines are drawn on the white plane, the sensor can detect the difference. The sensor can also be called Line Tracking Sensor.So we put black tape on the white paper so that the car could follow.

Warning:

Reflective Optical Sensor (including Line Tracking Sensor) should be avoided using in environment with infrared interference, like sunlight. Sunlight contains a lot of invisible light such as infrared and ultraviolet. Under environment with intense sunlight, Reflective Optical Sensor cannot work normally.



Right Line

Middle Line

Left Line

The following table shows the values of all cases when three Tracking Sensors detect objects of different colors. Among them, black objects or no objects were detected to represent 1 and white objects were detected to represent 0.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Left | Middle | Right | Value(binary) | Value(decimal) |
| 0 | 0 | 0 | 000 | 0 |
| 0 | 0 | 1 | 001 | 1 |
| 0 | 1 | 0 | 010 | 2 |
| 0 | 1 | 1 | 011 | 3 |
| 1 | 0 | 0 | 100 | 4 |
| 1 | 0 | 1 | 101 | 5 |
| 1 | 1 | 0 | 110 | 6 |
| 1 | 1 | 1 | 111 | 7 |

The car will make different actions according to the value transmitted by the line-tracking sensor.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Left | Middle | Right | Value(binary) | Value(decimal) | Action |
| 0 | 0 | 0 | 000 | 0 | backward |
| 0 | 0 | 1 | 001 | 1 | turnRight |
| 0 | 1 | 0 | 010 | 2 | forward |
| 0 | 1 | 1 | 011 | 3 | turnRight |
| 1 | 0 | 0 | 100 | 4 | turnLeft |
| 1 | 0 | 1 | 101 | 5 | forward |
| 1 | 1 | 0 | 110 | 6 | turnLeft |
| 1 | 1 | 1 | 111 | 7 | forward |

Flow chart of line tracking car is as below:

START

get TrackingSensorVal

trackingSensorVal=0

backward

YES

trackingSensorVal=1 or 3

turn Right

YES

trackingSensorVal=4 or 6

YES

turn Left

NO

NO

forward

NO

# 3. Upload the code

3.1 The code used in this lesson is placed in this folder:

E:\CKK0002-master\Tutorial\sketches\12\_1\_Automatic\_Tracking\_Line

3.2 Open the Arduino IDE

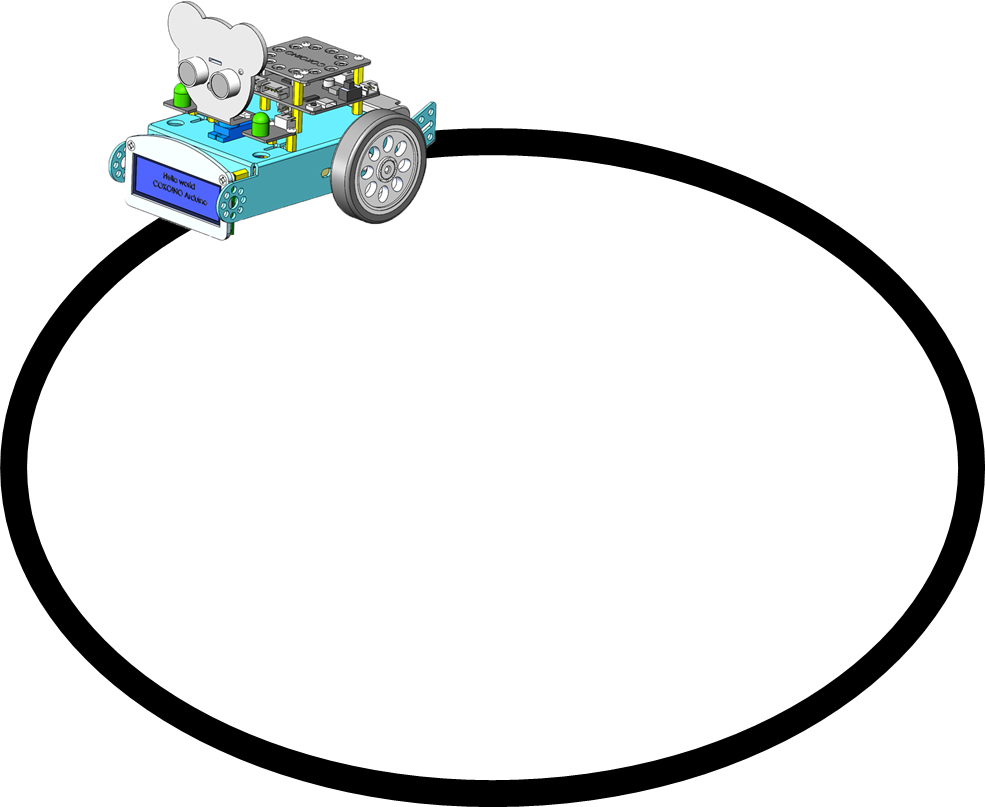
Select the board "Arduino UNO"and Port "COM3"

Before uploading the code, turn the ESP-01 switch to the side away from the "ESP-01" silk screen

Click compile button，successfully compiled the code will display “Done compiling”

Click upload button，successfully uploading the code will display “Done uploading”.

3.3 Use black tape to stick an oval runway with a width of 30mm on the white paper, or use a black watercolor pen to draw an oval runway with a width of 30mm on the white paper, then place the Smart Robot Car on the runway, and align the line tracking module with the black track.



3.4 Turn the power switch on the control board to ON side, and the smart car starts to run along the black track.

# 4. Troubleshooting

4.1 The car cannot follow the line track

You need to adjust the sensitivity of the three sensors on the line tracking module to be the same.

4.2 The car ran out of the line track

Check whether the black line track has a width greater than 30mm.

Check if the motor speed is set too fast.

# 5. Code

12\_1\_Automatic\_Tracking\_Line.ino

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

samrt robot car

            \_\_

-------------|  |-----------------

-------------|  |-----------------

      M1 (------) M4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <SCoop.h> //Import multithread library

#include <Adafruit\_PWMServoDriver.h> //Import PWM library

Adafruit\_PWMServoDriver pwm = Adafruit\_PWMServoDriver();

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27,16,2);

#include <Adafruit\_NeoPixel.h>

#ifdef \_\_AVR\_\_

#include <avr/power.h>

#endif

#define WS2812\_PIN 6   //WS2812 PIN

#define WS2812\_COUNT 12  // How many NeoPixels are attached to the Arduino?

#define BRIGHTNESS 10  // NeoPixel brightness, 0 (min) to 255 (max)

// Declare our NeoPixel strip object:

Adafruit\_NeoPixel strip = Adafruit\_NeoPixel(WS2812\_COUNT, WS2812\_PIN, NEO\_GRB + NEO\_KHZ800);

#define Buzz 11     //buzzer PIN

#define led\_R 9

#define led\_L 5

#define Line\_L A0   //left PIN

#define Line\_M A1   //middle PIN

#define Line\_R A2   //right PIN

defineTask(TaskOne);    // Create subthread 1

defineTask(TaskTwo);    // Create subthread 2

///////////////////////////////////////////////////////////////////////////////////

void TaskOne::setup()   //Thread 1 setup

{

  Serial.begin(9600);

  pwm.begin();

  pwm.setPWMFreq(50);

  pwm.setPWM(2, 0, 0);

  pwm.setPWM(3, 0, 0);

  pwm.setPWM(4, 0, 0);

  pwm.setPWM(5, 0, 0);

  lcd.init();

  lcd.backlight();

  lcd.clear();

  lcd.setCursor(2,0);

  lcd.print("HELLO WORLD!");

  lcd.setCursor(1,1);

  lcd.print("HELLO COKOINO!");

  pinMode(Line\_L, INPUT\_PULLUP);

  pinMode(Line\_M, INPUT\_PULLUP);

  pinMode(Line\_R, INPUT\_PULLUP);

}

void TaskTwo::setup(){  // Thread 2 setup

  pinMode(led\_R, OUTPUT);

  pinMode(led\_L, OUTPUT);

  strip.begin();

  strip.show();

  strip.setBrightness(BRIGHTNESS);

}

void setup(){

  mySCoop.start();//start setup

}

///////////////////////////////////////////////////////////////////////////////////

void TaskOne::loop() // loop subthread 1

{

  u8 trackingSensorVal = 0;

  trackingSensorVal = getTrackingSensorVal(); //get sensor value

  switch (trackingSensorVal)

  {

    case 0:   //000

      backward();//backward

      delay(15);

      break;

    case 7:   //111

      forward(); //car forward

      break;

    case 1:   //001

      turnRight(); //car turn Right

      delay(120);

      break;

    case 3:   //011

      turnRight(); //car turn Right

      delay(120);

      break;

    case 2:   //010

    case 5:   //101

    case 6:   //110

      turnLeft(); //car turn left

      delay(120);

      break;

    case 4:   //100

      turnLeft(); //car turn left

      delay(120);

      break;

    default:

      break;

  }

}

//when black line on one side is detected, the value of the side will be 0, or the value is 1

u8 getTrackingSensorVal() {

  u8 trackingSensorVal = 0;

  trackingSensorVal = (digitalRead(Line\_L) == 1 ? 1 : 0) << 2 | (digitalRead(Line\_M) == 1 ? 1 : 0) << 1 | (digitalRead(Line\_R) == 1 ? 1 : 0) << 0;

  return trackingSensorVal;

}

void forward()

{

 pwm.setPWM(2, 0, -650);

 pwm.setPWM(3, 0, 650);

 pwm.setPWM(4, 0, 650);

 pwm.setPWM(5, 0, -650);

}

void backward()

{

 pwm.setPWM(2, 0, 750);

 pwm.setPWM(3, 0, -750);

 pwm.setPWM(4, 0, -750);

 pwm.setPWM(5, 0, 750);

}

void turnRight()

{

 pwm.setPWM(2, 0, -1200);

 pwm.setPWM(3, 0, 1200);

 pwm.setPWM(4, 0, 650);

 pwm.setPWM(5, 0, -650);

}

void turnLeft()

{

 pwm.setPWM(2, 0, -650);

 pwm.setPWM(3, 0, 650);

 pwm.setPWM(4, 0, 1200);

 pwm.setPWM(5, 0, -1200);

}

void Stopcar()

{

 pwm.setPWM(2, 0, 0);

 pwm.setPWM(3, 0, 0);

 pwm.setPWM(4, 0, 0);

 pwm.setPWM(5, 0, 0);

}

void TaskTwo::loop() // loop subthread 2

{

  digitalWrite(led\_R,HIGH);

  digitalWrite(led\_L,HIGH);

  colorWipe(strip.Color(255, 0, 0), 10); // Red

  delay(800);

  colorWipe(strip.Color(255, 150, 0), 10); // yellow

  delay(800);

  colorWipe(strip.Color(0, 255, 0), 10); // Green

  delay(800);

  colorWipe(strip.Color(0, 255, 255), 10); // CYAN

  delay(800);

  colorWipe(strip.Color(0, 0, 255), 10); // Blue

  delay(800);

  colorWipe(strip.Color(180, 0, 255), 10); // purple

  delay(800);

  colorWipe(strip.Color(127, 127, 127), 10); // White

  delay(800);

  colorWipe(strip.Color(0, 0, 0), 30); // Clear

  Serial.println("OK");

}

void colorWipe(uint32\_t c, uint8\_t wait)

{

  for(uint16\_t i=0; i<strip.numPixels(); i++) {// For each pixel in strip...

    strip.setPixelColor(i, c);                 //  Set pixel's color (in RAM)

    strip.show();                              //  Update strip to match

    delay(wait);

  }

}

//Theatre-style crawling lights.

void theaterChase(uint32\_t c, uint8\_t wait) {

  for (int j=0; j<10; j++) {  //do 10 cycles of chasing

    for (int q=0; q < 3; q++) {

      for (int i=0; i < strip.numPixels(); i=i+3) {

        strip.setPixelColor(i+q, c);    //turn every third pixel on

      }

      strip.show();

      delay(wait);

      for (int i=0; i < strip.numPixels(); i=i+3) {

        strip.setPixelColor(i+q, 0);        //turn every third pixel off

      }

    }

  }

}

void loop(){

  yield();//start multithread task

}

# 6. Any questions and suggestions are welcome

Thank you for reading this document!

If you find any errors and omissions in the tutorial, or if you have any suggestions and questions, please feel free to contact us:

[cokoino@outlook.com](mailto:cokoino@outlook.com)

We will do our best to make changes and publish revisions as soon as possible.

If you want to learn more about smart cars, robots, learning kits and other technology products from us, please bookmark and pay attention to our website:

<http://cokoino.com/>

We will continue to launch interesting, cost-effective, innovative, user-friendly products.

LK COKOINO