# Lesson 5-Testing Line Tracking Module

## Table



## 1. Knowledege of Line Tracking Module

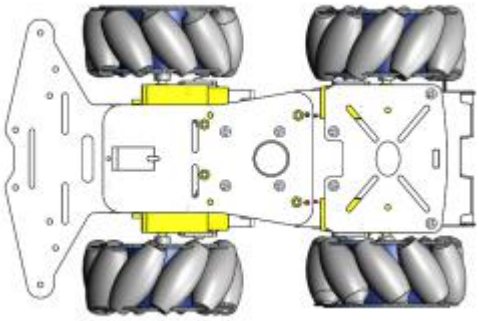
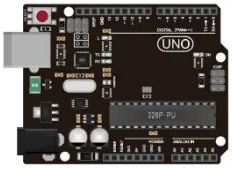
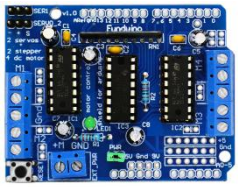
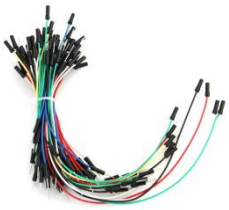
There are three Reflective Optical Sensors on Line Tracking Module. When the infrared light emitted by infrared diode shines on the surface of different objects, the sensor will receive light with different intensities after reflection.
As we know, black objects absorb light better. So when black lines are drawn on the white plane, the sensor can detect the difference. The sensor can also be called Line Tracking Sensor.
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.

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 |


Support： cokoino@outlook.com

## 2. Component/Module List

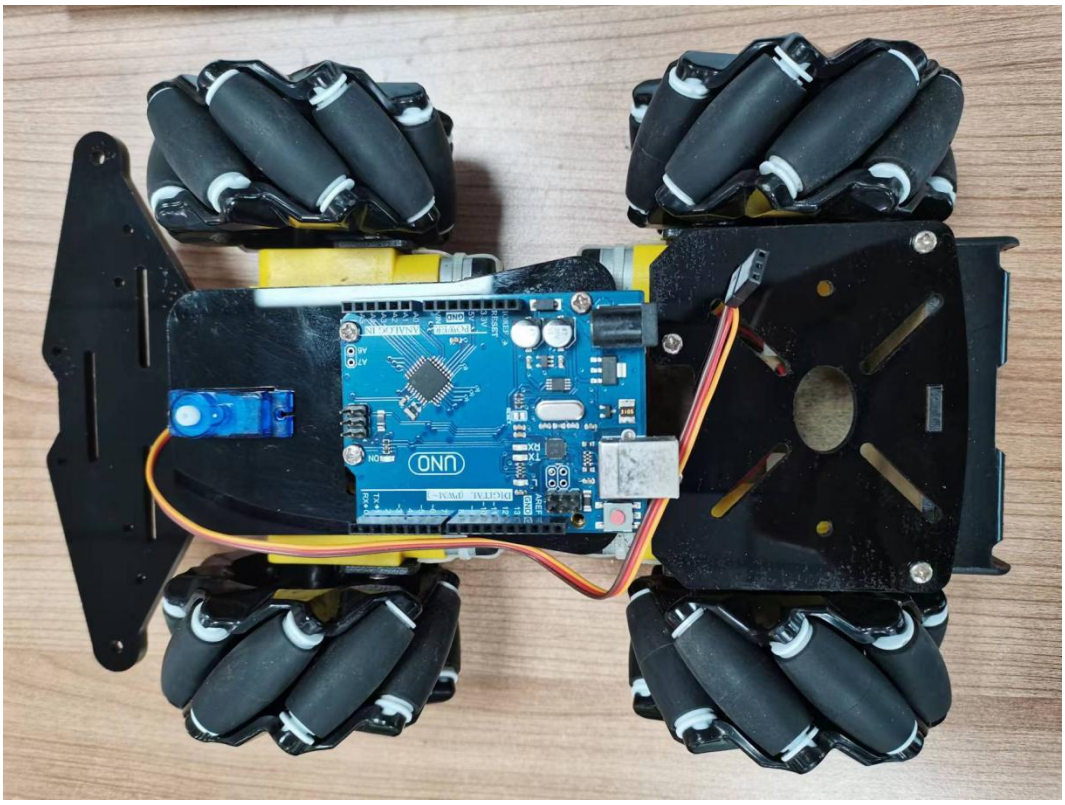
| Component/Module | QTY | Picture | Remark |
|---|---|---|---|
| 4WD Mecanum Wheel Car Chassis | 1 |  | Provided by the 4WD Mecanum Wheel Car Chassis Kit,you need assembled by yourself |
| UNO R3 Board | 1 |  | You need to prepare these by yourself.These just as an example,you can DIY what is you want and prepared |
| L293D Motor control shield | 1 |  | |
| Line Tracking Module | 1 |  | |
| Dupont line | Some |  | |



Support： cokoino@outlook.com

## 3. Build an experiment

### 3.1 Assemble the UNO R3 Board to the 4WD body as shown below



### 3.2 Install the L293D Motor Control Shield on the UNO board as shown below

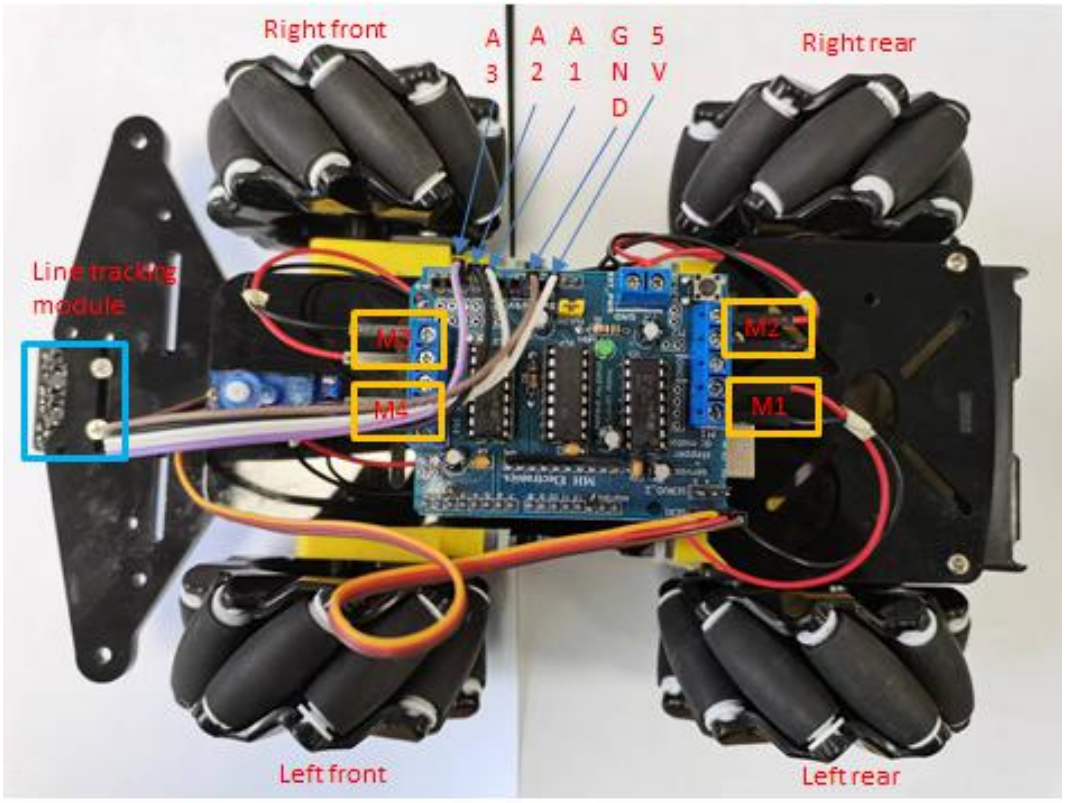


Support： cokoino@outlook.com

## 3.3 circuit connection

Connect the left front wheel motor to the M4 port of the Control Shield, the left rear wheel motor to the M1 port of the Control Shield, the right front wheel motor to the M3 port of the Control Shield, and the left rear wheel motor to the M2 port of the Control Shield;
Connect VCC of Line Tracking Module to 5V of Control Shield, GND to GND of Control Shield, S3(left) to A1 pin of Control Shield, S2(Middle) to A2 pin of Control Shield, S1(right) to the A3 pin of the Control Shield. As shown below

| Line tracking module pin | Motor control shield pin |
|---|---|
| VCC | 5V |
| GND | GND |
| S3 | A1 |
| S2 | A2 |
| S1 | A3 |




## 3.4 Circuit inspection

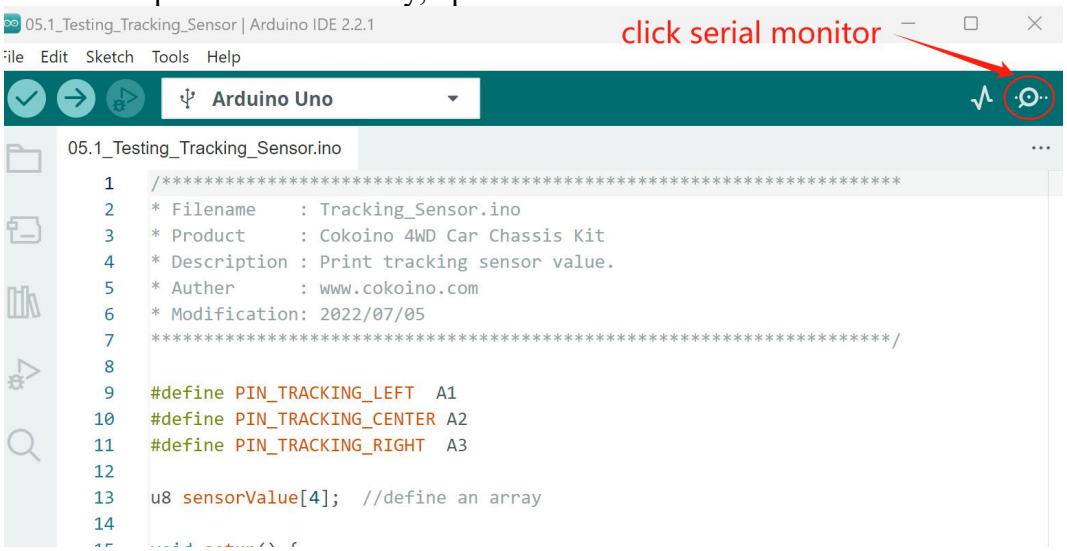
Before powering on, please carefully check whether the connected circuit is open or short-circuited, especially GND and 5V, GND and 3.3V, and must not be short-circuited. A short circuit can cause high current in your circuit, create excessive component heat and cause permanent damage to your hardware!


Support： cokoino@outlook.com

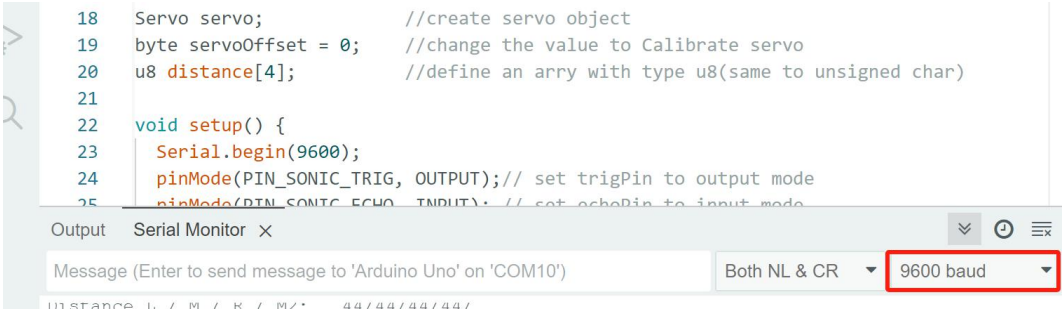
# 4. Program test

## 4.1 Test Line Tracking Module alone

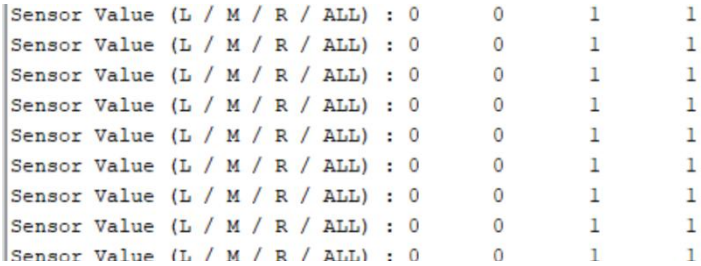
Click "File"---"Open" in the IDE interface, select the code under the path "E:\CKK0015-main\Tutorial\Arduino\Sketches\05.1_Testing_Tracking_Sensor". After the code is compiled successfully, connect the UNO board to the computer through the USB cable, select the corresponding serial port, Upload the program to the UNO board.
After the code is uploaded successfully, open the serial monitor.




Set the baud rate to 9600




The monitor is shown below:



**The code is below:**

```
#define PIN_TRACKING_LEFT A1
#define PIN_TRACKING_CENTER A2
#define PIN_TRACKING_RIGHT  A3

u8 sensorValue[4]; //define a array, u8 = unsigned char, 0~255.

void setup() {
  Serial.begin(9600); //set baud rate
  pinMode(PIN_TRACKING_LEFT, INPUT); //
  pinMode(PIN_TRACKING_RIGHT, INPUT); //
  pinMode(PIN_TRACKING_CENTER, INPUT); //
}

void loop() {
  sensorValue[0] = digitalRead(PIN_TRACKING_LEFT);
  sensorValue[1] = digitalRead(PIN_TRACKING_CENTER);
  sensorValue[2] = digitalRead(PIN_TRACKING_RIGHT);
  sensorValue[3] = sensorValue[0] << 2 | sensorValue[1] << 1 | sensorValue[2];
  Serial.print("Sensor Value (L / M / R / ALL) : ");
  for (int i = 0; i < 4; i++) {
    Serial.print(sensorValue[i]);
    Serial.print('\t'); //means Tab
  }
  Serial.print('\n'); //means new line
  delay(500);
}
```

## Bitwise Operators

There are some Bitwise Operators.

<< (bitshift left)

If sensorValue[0] =1, sensorValue[1]= 1, sensorValue[2]=1 sensorValue[0] << 2 then sensorValue[0]=100,(Binary), namely 4(Decimal) sensorValue[1] << 1 then sensorValue[0]=010,(Binary), namely 2(Decimal) sensorValue[2]=001(Binary)

| (bitwise or)

The code turns to: 100 | 010 | 001 =111(Binary), namely 7(Decimal)

>> (bitshift right) &
(bitwise and)
^ (bitwise xor)
~ (bitwise not)



Support： cokoino@outlook.com

For more information, please refer to: https://www.arduino.cc/reference/



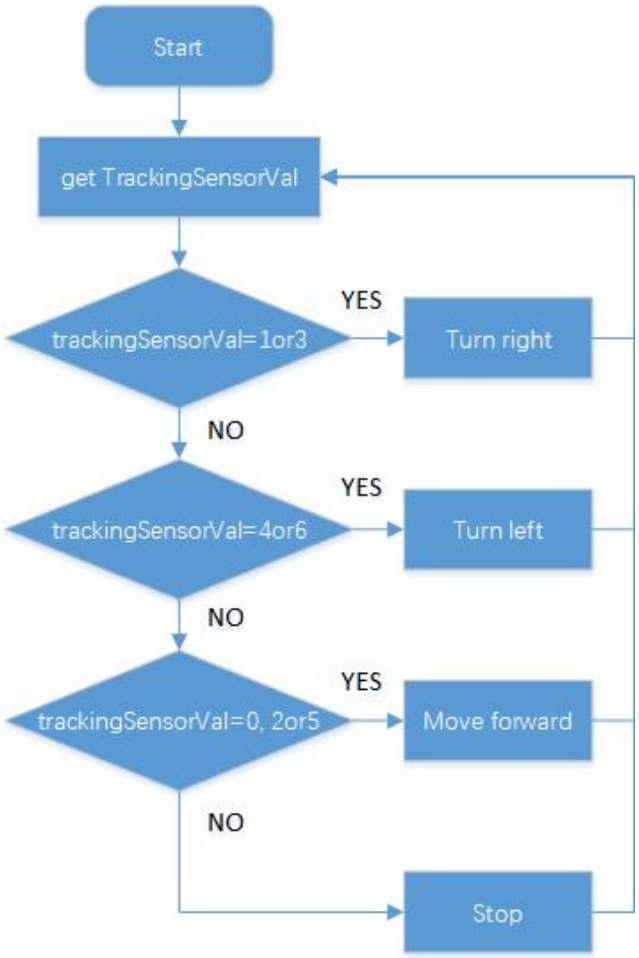
Support： cokoino@outlook.com

## 4.2 Line Tracking 4WD car

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

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

Flow chart of line tracking car is as below:

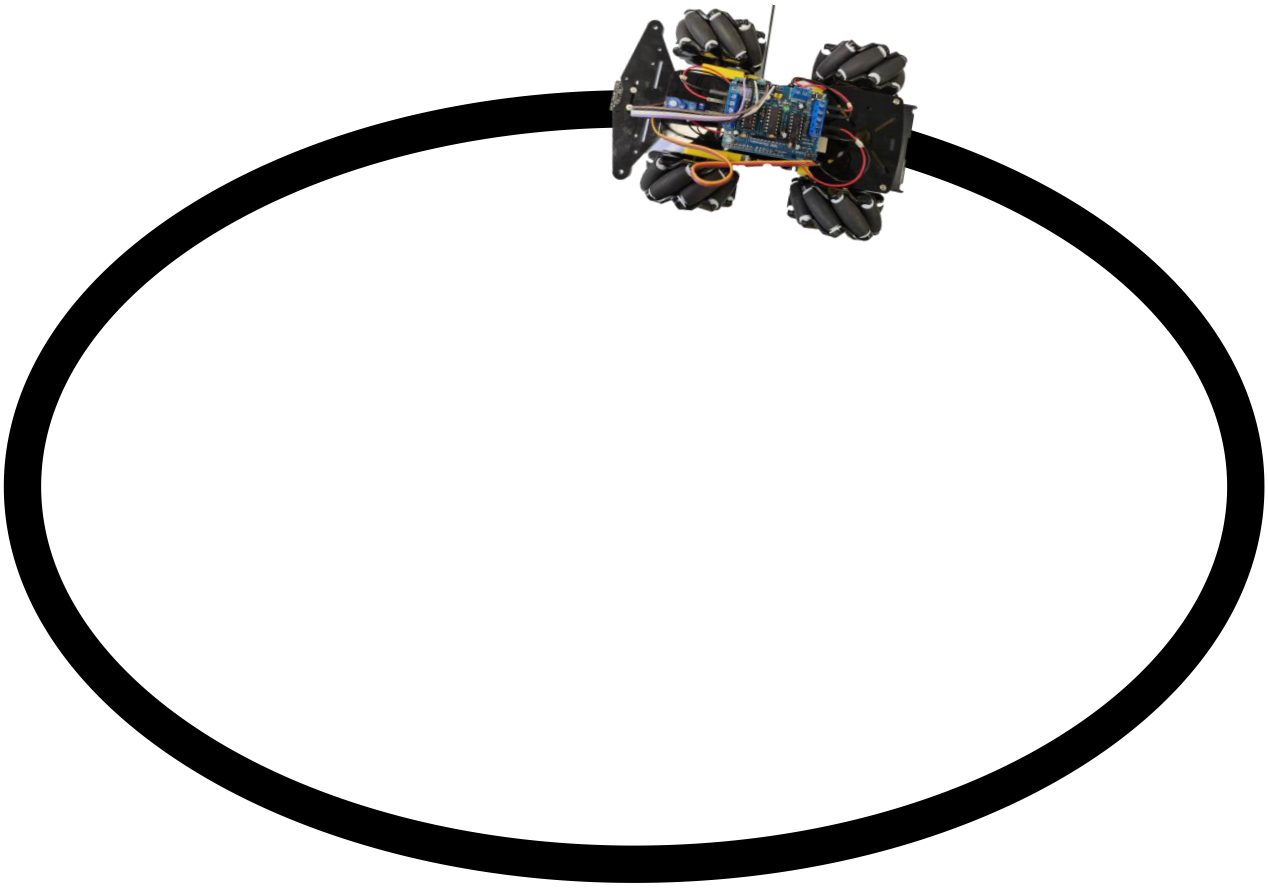





Support： cokoino@outlook.com

Click "File"---"Open" in the IDE interface, select the code under the path "E:\CKK0015-main\Tutorial\Sketches\05.2_Line_Tracking_Car". After the code is compiled successfully, connect the UNO board to the computer through the USB cable, select the corresponding serial port, upload the program to the UNO board.

Turn on the power. Use a black tape to build a line and then put your car on it as below.



You will see that the 4WD car will follow the trajectory

**The code is below：**

```
#include <AFMotor.h>
 AF_DCMotor motor1(1);//define motor1
 AF_DCMotor motor2(2);//define motor2
 AF_DCMotor motor3(3);//define motor3
 AF_DCMotor motor4(4);//define motor4
#define PIN_TRACKING_LEFT A1
#define PIN_TRACKING_CENTER A2
#define PIN_TRACKING_RIGHT A3

```



Support： cokoino@outlook.com

```
10.  void setup() {
11.    pinsSetup(); //set up pins
12.    getTrackingSensorVal();//Calculate Voltage speed Compensation
13.  }
14.
15.  void loop() {
16.    u8 trackingSensorVal = 0;
17.    trackingSensorVal = getTrackingSensorVal(); //get sensor value
18.
19.    switch (trackingSensorVal)
20.    {
21.      case 0:   //000//car stop
22.       motor1.run(RELEASE);// motor1 stop run
23.       motor2.run(RELEASE);
24.       motor3.run(RELEASE);
25.       motor4.run(RELEASE);
26.        break;
27.      case 7:   //111//car stop
28.       motor1.run(RELEASE);// motor1 stop run
29.       motor2.run(RELEASE);
30.       motor3.run(RELEASE);
31.       motor4.run(RELEASE);
32.        break;
33.      case 1:   //001//car turn
34.        motor1.setSpeed(150);//setup the speed of motor2
35.        motor4.setSpeed(150);//setup the speed of motor3
36.        motor1.run(FORWARD);//motor1 run forward
37.        motor4.run(FORWARD);
38.        break;
39.      case 3:   //011//car turn right
40.        motor1.setSpeed(100);//setup the speed of motor2
41.        motor4.setSpeed(100);//setup the speed of motor3
42.        motor1.run(FORWARD);//motor1 run forward
43.        motor4.run(FORWARD);
44.        break;
45.      case 2:   //010
46.      case 5:   //101//car move forward
47.        motor1.setSpeed(200);//setup the speed of motor1
48.        motor2.setSpeed(200);//setup the speed of motor2
49.        motor3.setSpeed(200);//setup the speed of motor3
50.        motor4.setSpeed(200);//setup the speed of motor4
51.        motor1.run(FORWARD);// motor1 run FORWARD
52.        motor2.run(FORWARD);
53.        motor3.run(FORWARD);
54.        motor4.run(FORWARD);
55.        break;
56.      case 6:   //110//car turn left
57.         motor1.run(RELEASE);// motor1 stop run
58.         motor4.run(RELEASE);
59.         motor2.setSpeed(150);//setup the speed of motor2
60.         motor3.setSpeed(150);//setup the speed of motor3
61.         motor2.run(FORWARD);// motor2 run FORWARD
62.         motor3.run(FORWARD);
63.        break;
64.      case 4:   //100 //car turn right
65.         motor1.run(RELEASE);
66.         motor4.run(RELEASE);
67.         motor2.setSpeed(150);//setup the speed of motor2
```

```
68.         motor3.setSpeed(150);//setup the speed of motor3
69.         motor2.run(FORWARD);// motor2 run FORWARD
70.         motor3.run(FORWARD);
71.        break;
72.      default:
73.        break;
74.    }
75.  }
76.
77.
78.  //when black line on one side is detected, the value of the side will be 0, or the value is 1
79.  u8 getTrackingSensorVal() {
80.    u8 trackingSensorVal = 0;
81.    trackingSensorVal = (digitalRead(PIN_TRACKING_LEFT) == 1 ? 1 : 0) << 2 | (digitalRead(PIN_TRACKIN
     G_CENTER) == 1 ? 1 : 0) << 1 | (digitalRead(PIN_TRACKING_RIGHT) == 1 ? 1 : 0) << 0;
82.    return trackingSensorVal;
83.  }
84.
85.  void pinsSetup() {
86.    //define tracking sensor pin
87.    pinMode(PIN_TRACKING_LEFT, INPUT);
88.    pinMode(PIN_TRACKING_RIGHT, INPUT);
89.    pinMode(PIN_TRACKING_CENTER, INPUT);
90.  }
```

**Exp1 ? Exp2 : Exp3;**

If Exp1 is true, the result of this code is Exp2. If Exp1 is false, the result of this code is Exp3. For example

If y=8;

var = (y < 10) ? 30 : 40; then var=30 If y=10

var = (y < 10) ? 30 : 40; then var=40

**switch...case**



Support： cokoino@outlook.com

```
switch (var)
  { case 1:
    //do something when var equals
    1 break;
  case 2:
    //do something when var equals
    2 break;
  default:
    // if nothing else matches, do the default
    // default is
    optional break;
}
```

For more information, please refer to:
https://www.arduino.cc/reference/en/language/structure/control-structure/switchcase/

## 5. Make your suggestion and get support

THANK YOU for participating in this learning experience!

We have reached the end of this Tutorial. If you find errors, omissions or you have suggestions and/or questions about this lesson, please feel free to contact us: cokoino@outlook.com

We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our Amazon Store by search for "LK COKOINO" on Amazon. We will continue to launch fun, cost-effective, innovative and exciting products.

Thank you again for choosing Cokoino products.

**LK COKOINO**