



Lesson 5 Line tracking car





Points of the section

In this lesson, we will learn how to control car to move along a runway.

Learning parts:

-  *Learn how to use the line tracking module*
-  *Learn the line tracking principles*
-  *Learn how to implement line tracking via programming*

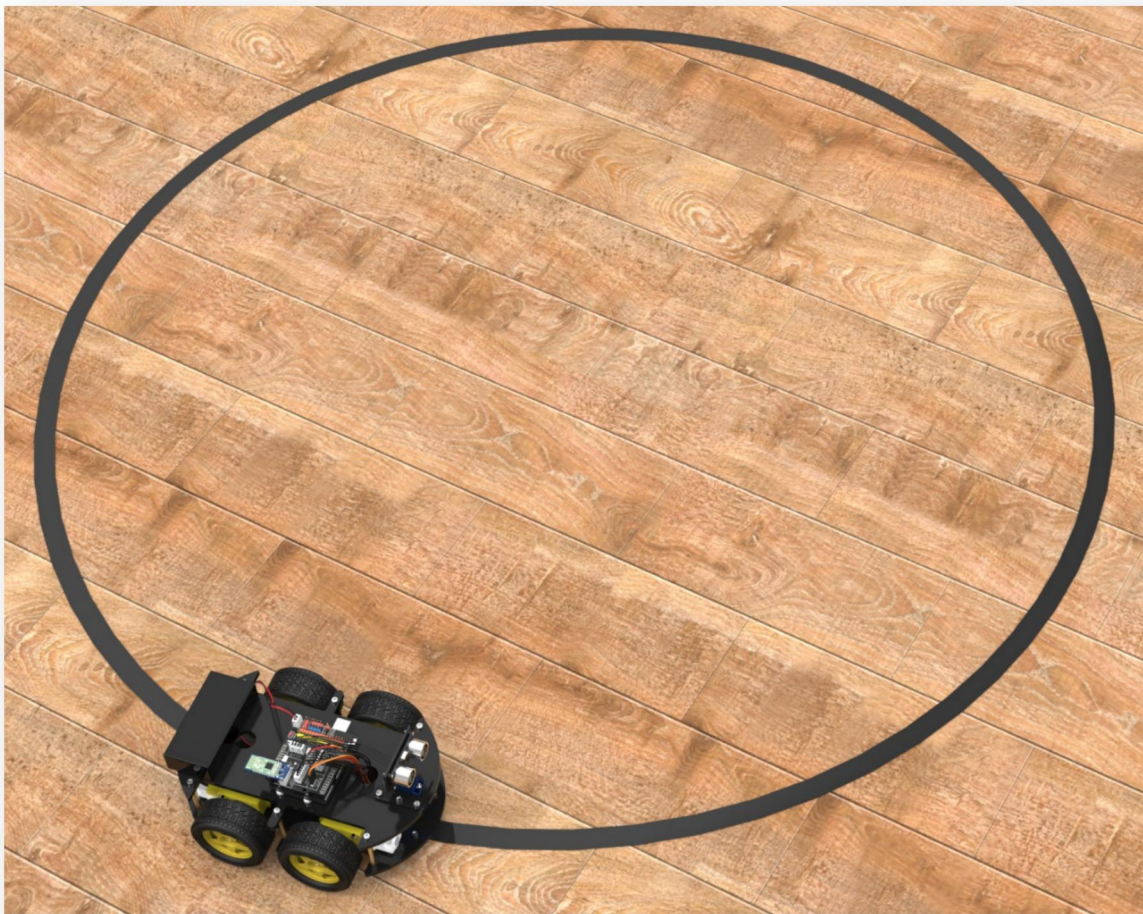
Preparations:

-  *A car (equipped with battery)*
-  *A USB cable*
-  *Three line tracking modules*
-  *A roll of black tape*

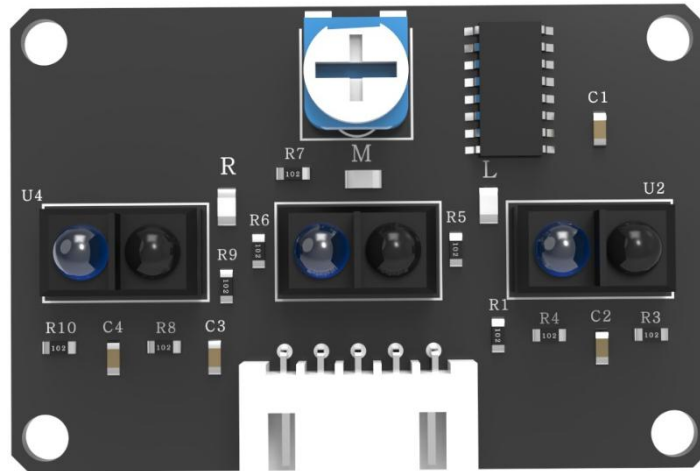
I . Making runway

Materials: electrical adhesive tape (black tape)

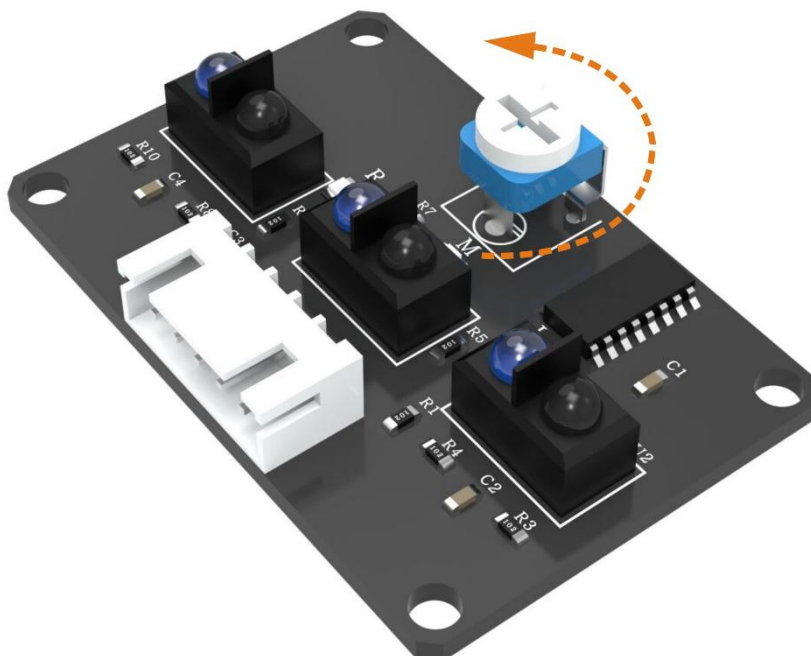
First of all, we need to make a runway on our own. We can make a circuit by pasting black tape on a suitable paper or the ground. Before pasting, you can draw a runway by pen, and then paste with electrical adhesive tape. Pay attention to make the corner as smooth as possible. Because the car will outgo of the line if the angle is too small, but if you want to make it more difficulty, you can make it small. The size of runway is generally not smaller than 40*60 cm.



II. Connect modules and debug

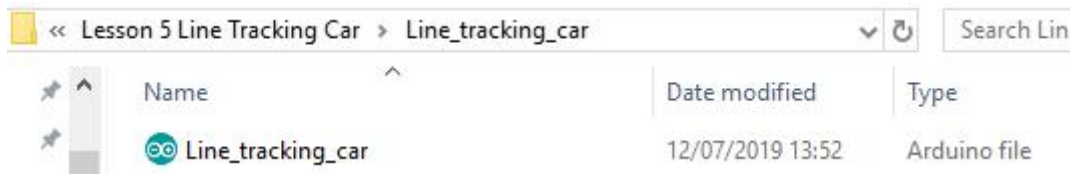


The component which is pointed at is potentiometer. It can adjust the sensitivity of the line tracking module by change its resistance value.



III. Upload program

After making runway and connecting modules, you just need to open the the code file “\Lesson 5 Line Tracking Car\Line_Tracking_Car\Line_Tracking_Car.ino” and upload the program to the UNO controller board.



Code preview:

```
//www.elegoo.com

//Line Tracking IO define
#define LT_R !digitalRead(10)
#define LT_M !digitalRead(4)
#define LT_L !digitalRead(2)

#define ENA 5
#define ENB 6
#define IN1 7
#define IN2 8
#define IN3 9
#define IN4 11

#define carSpeed 250

void forward() {
  analogWrite(ENA, carSpeed);
  analogWrite(ENB, carSpeed);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  Serial.println("go forward!");
}

void back() {
  analogWrite(ENA, carSpeed);
  analogWrite(ENB, carSpeed);
  digitalWrite(IN1, LOW);
```

```
digitalWrite(IN2, HIGH);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
Serial.println("go back!");
}

void left () {
  analogWrite(ENA, carSpeed);
  analogWrite(ENB, carSpeed);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  Serial.println("go left!");
}

void right () {
  analogWrite(ENA, carSpeed);
  analogWrite(ENB, carSpeed);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  Serial.println("go right!");
}

void stop () {
  digitalWrite(ENA, LOW);
  digitalWrite(ENB, LOW);
  Serial.println("Stop!");
}

void setup () {
  Serial.begin(9600);
  pinMode(10, INPUT);
  pinMode(4, INPUT);
  pinMode(2, INPUT);
}

void loop () {
  if(LT_M) {
    forward();
  }
  else if(LT_R) {
```

```

right();
while(LT_R);
}
else if(LT_L) {
left();
while(LT_L);
}
}

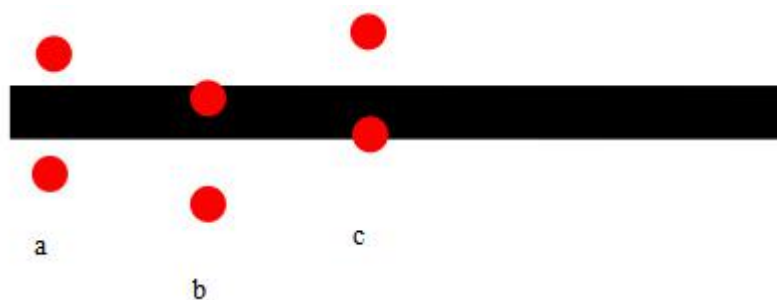
```

After disconnecting the car to the computer, you can turn on the power switch and put the car on the runway. Then the car will follow the lines. If you find that it can't move as you expected, please adjust the potentiometer on the line tracking module.

IV. Introduction of principle

Line tracking module

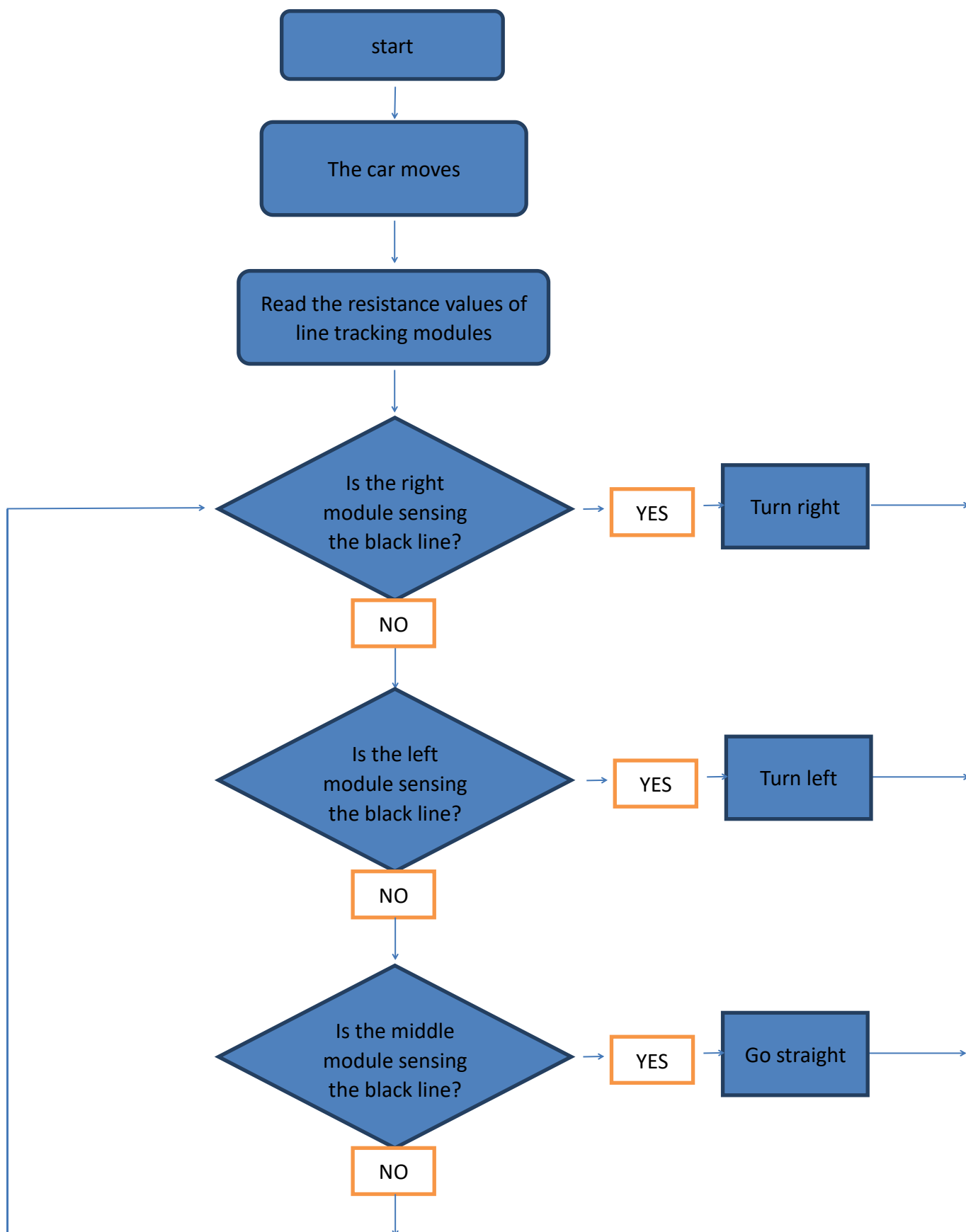
The line tracking sensors are the two components located downward and abreast before the car. The line tracking sensor consists of an infrared transmitter tube and an infrared receiver tube. The former is a LED that can transmit the infrared ray, while the latter is a photoresistor which is only responsible for receiving the infrared light. Light reflectance for the black surface is different from that for the white surface. Hence, the intensity of the reflected infrared light received by the car at the black road differs from that at the white road, and the resistance quantity also changes. According to the principle of voltage division among series resistance, motion path can be determined by inferring the color of road below the car from the voltage of the sensor.



a → The car moves along the black line. One of the line tracking modules is on the left side of the line and the other one is on the right side. They can't detect the black line.

b → The car learns to move right. The module on the left side can detect the black line, then it will send a signal to the controller board and the car turns left.

c → The car learns to move left. The module on the right side can detect the black line, then it will send a signal to the controller board and the car turns right.



From above, we can see the principle of the line tracking car. After the car starting, the line tracking module just need to sense the black line on the road surface, make corresponding action according to the program.

This is a simple algorithm chart of car line tracking program. There are many more complex algorithms such as PID. So after making the function of line tracking come true, you can learn more algorithms of controlling car on your own.

Small tips

- (1) The bending part of the line should be as smooth as possible. If the cornering radius is too small, the car is very likely to move beyond the runway.
- (2) The line tracking scene can be made with the black and white tape or the paper of any color that be distinguished from the path.
- (3) In addition to line tracking, we can stretch imagination to develop other programs based on the line tracking principles, such as those that confine the car within a region regardless of its motion.



<http://www.elegoo.com>

2019.7.15
