Arduino UNO Car 1.0Tutorial

V1.0.2019.07.31

Lesson 3 Line tracking car

main points

In this lesson, we will learn how to control the car running along the black track.

First , Learning

1. Learn how to use the line tracking module
2. Learn the principle of line tracking
3. Learn how to realize line tracking by programming

Second, Preparations

Smart car(with battery)

USB cable

2-way tracking module

A roll of black tape

1、track production

Material：Electric adhesive tape（Black tape）First of all, we need to make a track. We can use black tape to make the right paper or floor for the circuit. Before pasting, you can draw the runway with a pen, and then paste

Be careful to smooth the corner as much as possible. The size of the track is generally not less than 40 * 60 cm.

Upload program

After you create the track and connect the module, you just need to open the code file in the path：“\ Lesson 3 Car Tracking Mode\ Line_Tracking_Car \ xunji1-1.ino” And upload the program to the Arduino Nano controller board.

Code preview:

```
#define ENA 5
#define ENB 6
#define IN1 7
#define IN2 8
#define IN3 9
#define IN4 11
//Motor PWM speed regulation
#define carSpeed 150
int left_LED = 16;
int Right_LED = 17;

const int SensorLeft = 4;          //    Left tracking sensor

const int SensorRight = 10;         // Right tracking sensor
int SL;    //
int SR;    //

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

 /****back****/
void back_run()
{
    analogWrite(ENA,carSpeed);
    analogWrite(ENA,carSpeed);
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);
    digitalWrite(IN3,LOW);
```

```
    digitalWrite(IN4,HIGH);
    Serial.println("go back!");

 }

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

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

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


/*************************************************************************
***********
```

```
* Port initialization
**********************************************************************************
**********/
void setup()
{
        Serial.begin(9600);
        pinMode(IN1, OUTPUT);
        pinMode(IN2, OUTPUT);
        pinMode(IN3, OUTPUT);
        pinMode(IN4, OUTPUT);
        pinMode(ENA, OUTPUT);
        pinMode(ENB, OUTPUT);
        pinMode(SensorLeft,INPUT);
        pinMode(SensorRight,INPUT);
        pinMode(left_LED,OUTPUT);
        pinMode(Right_LED,OUTPUT);
        stop();
}


/**********************************************************************************
***********
* Main program
**********************************************************************************
**********/
void loop()
{
        SL = digitalRead(SensorLeft);
        SR = digitalRead(SensorRight);

        if (SL == LOW&&SR==LOW)    //The left and right sensors detect white, the left and right
status lights are on, and the car advances.
        {

            forward_run();//forward

         }
         else
         {
                if (SL == HIGH & SR == LOW)       //The left sensor detects the black track, the
left status light is off, and the car turns left
                {

                        delay(1);
```

```
                left_run();//left


            }
             else if (SR == HIGH & SL == LOW)      //The sensor on the right detects the black track, the state light on the right is off, and the car turns right
            {


                delay(1);
                right_run();


            }
             else //All detected by the left and right sensors are black, the left and right state lights are off, the car stops, and basically does not exist in the track
            {
                delay(1);
                stop();

            }
        }
}
```

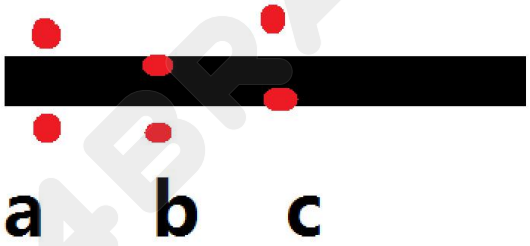
After disconnecting the car from the computer, you can turn on the power switch and put the car on the track of the computer. Then the car will follow this line. If you find it unable to move as expected, adjust the potentiometer on the tracking module.

Secondly, Introduction of principle：

tracking module

The tracking sensor is located in front of the car with two parts.The tracking sensor is composed of infrared emission tube and infrared receiving tube. The former is an LED that can transmit infrared light, and the latter is a photoresistor for receiving infrared rays.The light reflectance of the black surface is different from that of the white surface. Therefore, the intensity of reflected infrared light received by the car on the black road is different from that of the white road, and the resistance force will also change. According to the principle of voltage division between series resistance, the 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 tracking modules is on the left side of the line and the other is on the right. The black line could not be detected.

b→The car learn to move to the right. The module on the left can detect the black line, and then it will detect the sending signal to the controller board, and the car will turn left.。

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

Combining the above information, we can see the principle of tracking cars. After the car starts, the tracking module only needs to sense the black line on the road and make the corresponding action according to the need. There are many more complex algorithms, such as PID. Therefore, after making the tracking function a reality, you can learn more algorithms to control the car yourself.

Tips:

(1) The curved part of the line shall be as smooth as possible. If the turning radius is too small, the car is likely to overtake the track.

（2）Line tracking scenes can be made of black and white tape or paper of any color, for differing the path。

（3）Except for line tracking, We can also develop other program line tracking principles.For example, the principle of limiting cars to areas regardless of their movement.