Lesson 5 Line tracking car

Points of the section

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

Learning parts:
- Learn the use of line tracking module
- Understand the principle of line tracking
- Use the program to make line tracking function come true

Before prepared:
- One car (with battery)
- One USB download line
- Three line tracking modules
- Black tape
I. Making runway

Materials: electrical adhesive tape (black tape), A3 paper (or bigger)

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.
The component which is pointed at is potentiometer. It can adjust the sensitivity of the

[Diagram of the board with components labeled]
line tracking module by change its resistance value.

### III. Upload program

After making runway and connecting modules, you just need to upload the program to the UNO controller board.

The code is as follower:

```cpp
int in1 = 6;
int in2 = 7;
int in3 = 8;
int in4 = 9;
int ENA = 5;
int ENB = 10;
int ABS = 120;

void _mForward()
{
    analogWrite(ENA, ABS);
    analogWrite(ENB, ABS);
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    Serial.println("go forward!");
}

void _mBack()
{
    analogWrite(ENA, ABS);
    analogWrite(ENB, ABS);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);"
```cpp
digitalWrite(in4, LOW);
Serial.println("go back!");
}

void _mleft()
{
analogWrite(ENA, ABS);
analogWrite(ENB, ABS);
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
Serial.println("go left!");
}

void _mright()
{
analogWrite(ENA, ABS);
analogWrite(ENB, ABS);
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
Serial.println("go right!");
}
void _mStop()
{
digitalWrite(ENA, LOW);
digitalWrite(ENB, LOW);
Serial.println("Stop!");
}

void setup()
```
```cpp
{  
    Serial.begin(9600);  
}

void loop() {
    int num1, num2, num3;
    num1 = digitalRead(11);
    num2 = digitalRead(4);
    num3 = digitalRead(2);
    if ((num1 == 0) && num2 && num3) {
        _mleft();  // The sensor detected that right car turn left immediately when it meets black line
        delay(2);
        while (1) {
            num2 = digitalRead(2);  // Cycle to judge degree of intermediate sensor,
            if (num2 == 1) {  // If num2 == 1 does not go to the middle position, continue to turn left.
                _mleft();
                delay(2);
            } else
                break;  // Detection of num2 == 0 instructions turned over, out of the loop, detection of three sensors' status and then make appropriate action
        }
    } else if (num2 && num1 && (num3 == 0)) {
        _mright();
        delay(2);
        while (1) {
            num2 = digitalRead(2);
        }
    } else if (num2 && num1 && (num3 == 0)) {
        _mleft();  // The sensor detected that right car turn left immediately when it meets black line
        delay(2);
        while (1) {
            num2 = digitalRead(2);  // Cycle to judge degree of intermediate sensor,
            if (num2 == 1) {  // If num2 == 1 does not go to the middle position, continue to turn left.
                _mleft();
                delay(2);
            } else
                break;  // Detection of num2 == 0 instructions turned over, out of the loop, detection of three sensors' status and then make appropriate action
        }
    } else if (num2 && num1 && (num3 == 0)) {
        _mright();
        delay(2);
        while (1) {
            num2 = digitalRead(2);
        }
    }
```
Open the file Line_Tracking_Car /Line_Tracking_Car.ino and upload the program to the UNO controller board. 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

We have three line tracking modules in one car, and we assemble them side by side on the car chassis.

Line tracking module is made up of an infrared light-emitting diode and an infrared receiving diode. Infrared light-emitting diode is one kind of LED which can launch infrared ray. Receiving diode is one kind of photo resistance that only receive infrared ray and change its resistance value according to the reflectivity.

When the line tracking module is working, the infrared light-emitting diode will launch infrared ray. And the ray will reflect when meeting things. Then the receiving diode can receive the ray. Because the reflectivity of different color is different so that the
receiving diode will change its resistance value when sensing different color. According to the resistance value, we can know the color of the floor under the car.

a → The car moves along the black line. One of the line tracking module 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 leans to right. The module on the left side can detect the black line, then it will send signal to the controller board and the car turn left.
c → The car leans to left. The module on the right side can detect the black line, then it will send signal to the controller board and the car turn right.
start

The car moves

Read the resistance values of line tracking modules

Is the right module sensing the black line?

YES → Turn right

NO

Is the left module sensing the black line?

YES → Turn left

NO

Is the middle module sensing the black line?

YES → Go straight

NO
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）When you making the runway, please make the corner of the runway as smooth as possible. Because the probability of car outgoing will be increased if the corner’s radius is too small.
（2）The runway can not only be black color but also other colors.
（3）If you look up Google, you can see that there are two modes you can play with the game, one is to follow a line, and the other is to stay inside a line.