

レッスン5 ライン追跡車

セクションのポイント

このレッスンでは、滑走路に沿って移動するように車を制御する方法を学びます。

学習パーツ：

- ◆ ライントラッキングモジュールの使用方法を学ぶ
- ◆ ライントラッキングの原則を学ぶ
- ◆ プログラミングでライントラッキングを実装する方法を学ぶ

準備：

- ◆ カー（バッテリー搭載）
- ◆ USB ケーブル
- ◆ 3つのライントラッキングモジュール
- ◆ 黒テープのロール

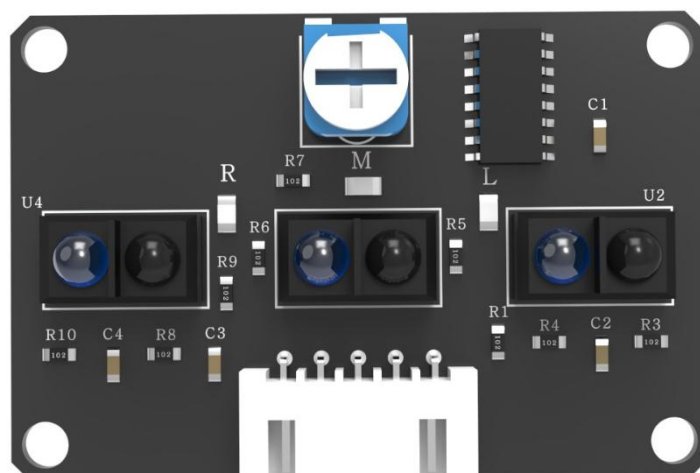
I. 滑走路を作る

材質：電気粘着テープ（黒テープ）

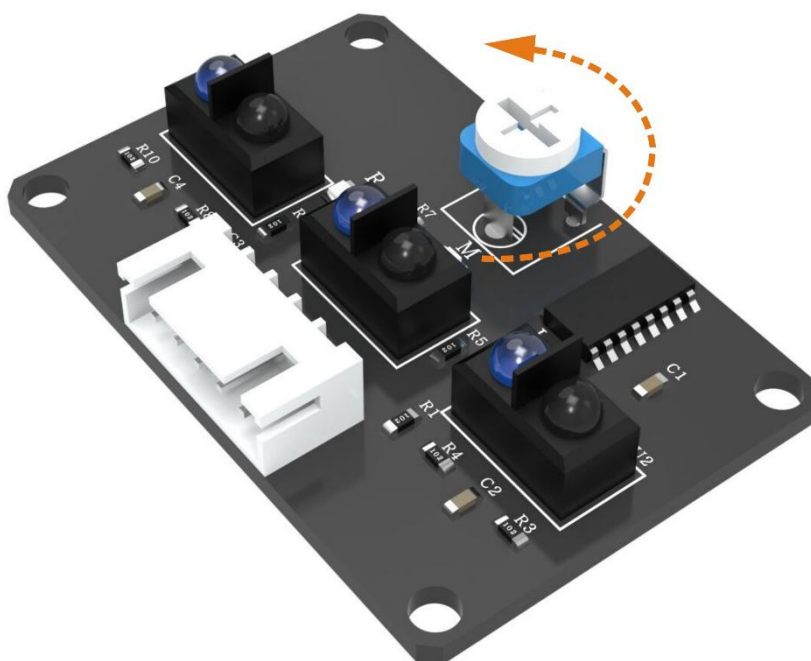
まず、自分で滑走路を作る必要があります。 適当な紙や地面に黒いテープを貼って回路を作ることができます。 貼り付ける前に、滑走路をペンで描いてから、粘着テープで貼り付けます。 コーナーができるだけ滑らかになるように注意してください。 角度が小さすぎるとラインからはみ出しますが、難易度を上げたい場合は小さくすることができます。 滑走路のサイズは一般的に 40 * 60 cm 以上です。



Ⅱ. モジュールを接続してデバッグする

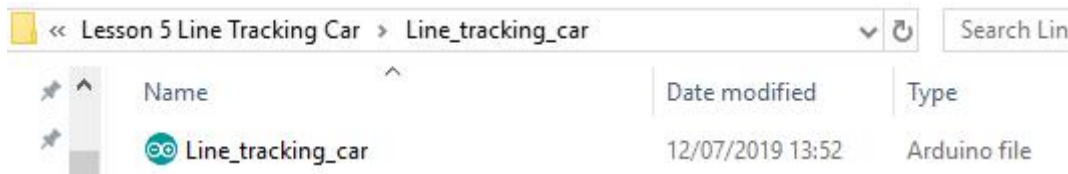


指し示されているコンポーネントはポテンシオメーターです。抵抗値を変更することにより、ライントラッキングモジュールの感度を調整できます。



Ⅲ. プログラムをアップロード

滑走路を作成してモジュールを接続したら、コードファイル “**¥Lesson 5 Line Tracking Car¥Line_Tracking_Car¥Line_Tracking_Car.ino**” を開き、プログラムを UNO コントローラーボードにアップロードするだけです。



コードプレビュー:

```
//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) {
```

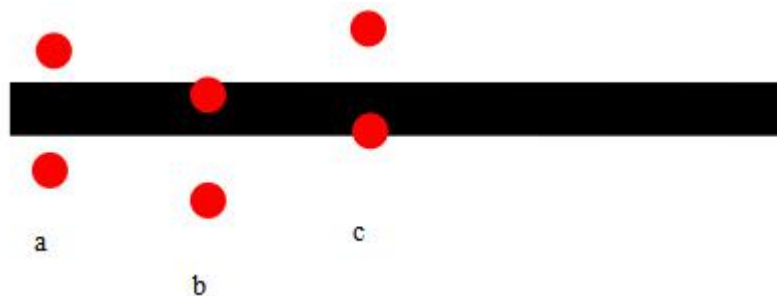
```
right();
while(LT_R);
}
else if(LT_L) {
    left();
    while(LT_L);
}
}
```

車をコンピューターから取り外したら、電源スイッチをオンにして車を滑走路に置くことができます。その後、車は線に従います。期待どおりに動かない場合は、ライントラッキングモジュールのポテンシオメーターを調整してください。

IV. 原理の紹介

ライン追跡モジュール

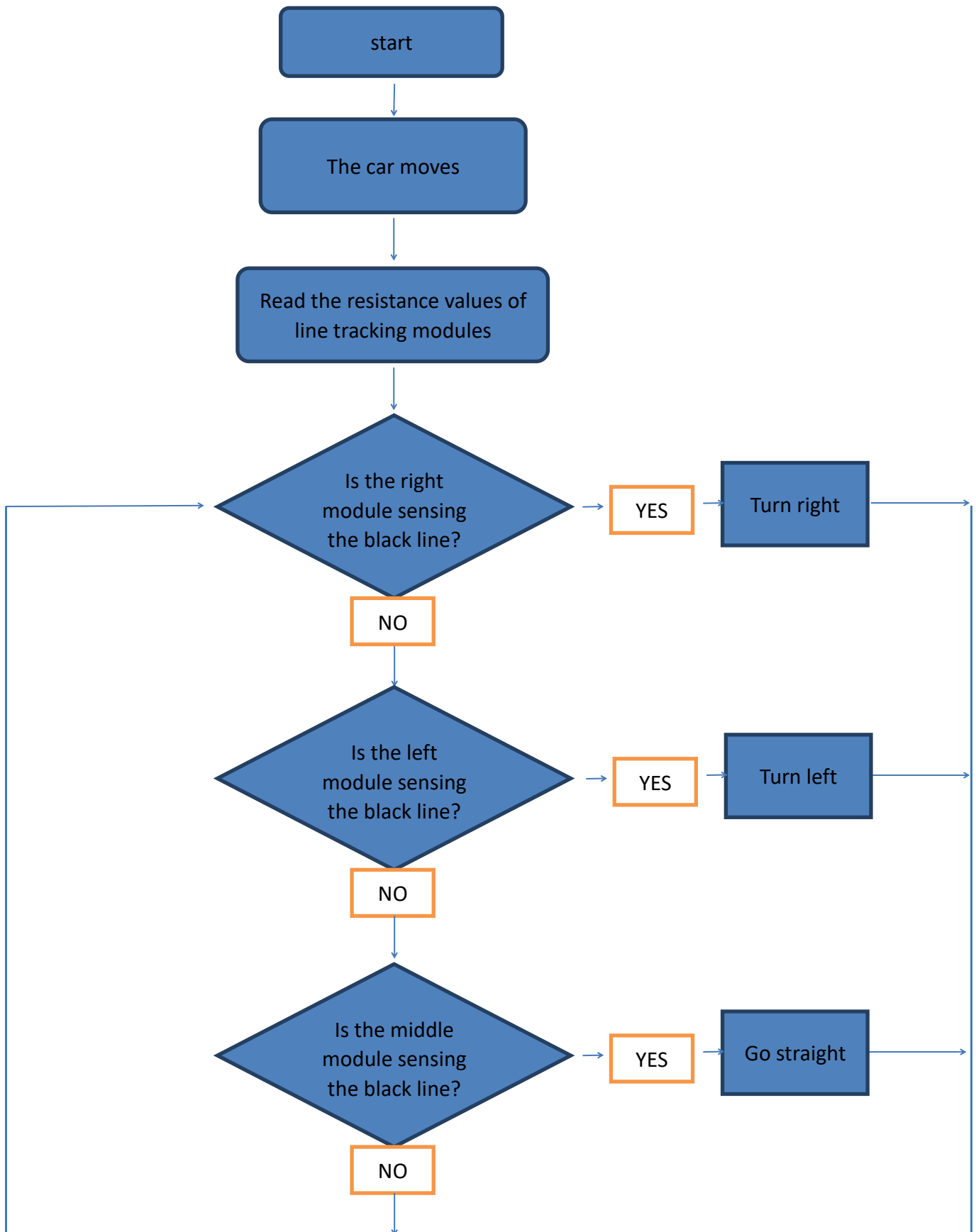
ライントラッキングセンサーは、車の前の下方にある2つのコンポーネントです。ライントラッキングセンサーは、赤外線送信機チューブと赤外線受信機チューブで構成されています。前者は赤外線を送信できるLEDで、後者は赤外線のリフレキターを受信のみです。黒色の表面と白色の表面では光の反射率が異なります。このため、黒い道路で車が受ける反射赤外光の強度は、白い道路とは異なり、抵抗量も変化します。直列抵抗間の電圧分割の原理によれば、センサーの電圧から車の下の道路の色を推測することにより、運動経路を決定できます。



a → 車は黒い線に沿って移動します。ライントラッキングモジュールの1つはラインの左側にあり、もう1つは右側にあります。二つとも黒い線を検出できません。

b → 車は右に移動することを学びます。左側のモジュールが黒い線を検出すると、コントローラーボードに信号が送信され、車が左折します。

c → 車は左に移動することを学びます。右側のモジュールが黒い線を検出すると、コントローラーボードに信号が送信され、車が右折します。



以上の画像から、ライン追跡カーの原理を理解できます。車が起動した後、ライン追跡モジュールは、路面の黒い線を検知し、プログラムに従って対応するアクションを実行するだけです。

これは車のライン追跡プログラムの簡単なアルゴリズムチャートです。PIDなどのより複雑なアルゴリズムは数多くあります。したがって、ライン追跡の機能を実現した後は、自分で車を制御するアルゴリズムをさらに学ぶことができます。

ヒント

- (1) 線の曲げ部分はできるだけ滑らかにしてください。コーナリング半径が小さすぎると、車が滑走路を越えて移動する可能性が高くなります。
- (2) ライントラッキングシーンは、パスと区別される白黒テープまたは任意の色の紙で作成できます。
- (3) ライントラッキングに加えて、ライントラッキングの原則に基づいて、想像力を広げて、モーションに関係なく走路内に車を閉じ込めるような他のプログラムを開発することができます。



<http://www.elegoo.com>

2020.9.24