

Lesson 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、車を走路に沿って移動するように制御する方法を学習します。

学習パーツ：

- ◆ ライントラッキングモジュールの使い方を学ぶ
- ◆ ライントラッキングの原則を学ぶ
- ◆ プログラミングによるライントラッキングの実装方法を学ぶ

準備：

- ◆ 車（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
- ◆ 黒いテープのロール

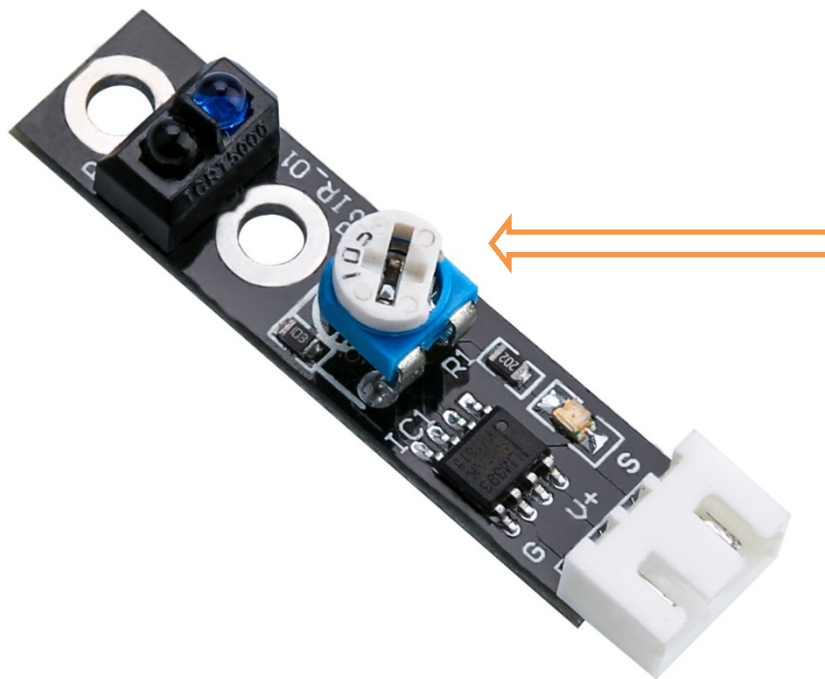
I. 走路を作る

材料：電気接着テープ（黒色テープ）

まず第一に、自分たちで滑走路を作る必要があります。適切な紙や地面に黒いテープを貼り付けることで走路を作ることができます。貼るする前に、滑走路をペンで描き、電気粘着テープで貼り付けることができます。コーナーを可能な限り滑らかにするように注意してください。角度が小さすぎると車が脱線します。もっと難しくしたい場合は小さくすることもできます。走路のサイズは一般に 40×60cm 以上が望ましいです。



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ョメータです。ライントラッキングモジュールの感度を調整するには、抵抗値を変更します。

Ⅲ. プログラムをアップロードする

走路とモジュールを接続したら、UNO コントローラボードにプログラムをアップロードするだけです。

コードは以下です：

```
int in1=6;
int in2=7;
int in3=8;
int in4=9;
int ENA=5;
int ENB=11;
int ABS=115;
void _mForward()
{
  analogWrite(ENA,ABS);
  analogWrite(ENB,ABS);
  digitalWrite(in1,HIGH);//digital output
  digitalWrite(in2,LOW);
  digitalWrite(in3,LOW);
  digitalWrite(in4,HIGH);
  Serial.println("go forward!");
}

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

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

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

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

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

```

}

void loop() {
  int num1,num2,num3;
  num1=digitalRead(10);
  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)
      { _mleft();              //If num2==1 does not go to the middle position, continue
to turn left.
        delay(2);}
      else
        break;                //Detection of num2==0 instructions turned over, out of the
loop, detection of three sensors' statusand then make appropriate action
    }                          //The following and so on
  }

  else if(num2&&num1&&(num3==0))
  {
    _mright();
    delay(2);
    while(1)
    {
      num2=digitalRead(2);
      if(num2==1){
        _mright();

```

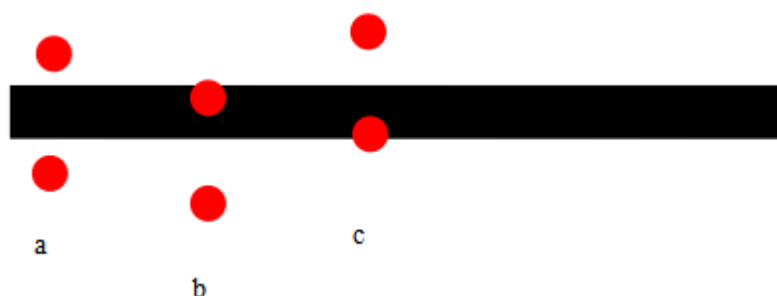
```
    delay(2);}
  else
    break;
}
}
else
{
  _mForward();
  delay(2);
}
}
```

ファイル Line_Tracking_Car /Line_Tracking_Car.ino を開き、プログラムを UNO コントローラボードにアップロードします。車をコンピュータから切断した後、電源スイッチをオンにして滑走路に乗せます。その後、車はラインをたどります。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ョメータを調整してください。

IV. 原理の導入

ライントラッキングモジュール

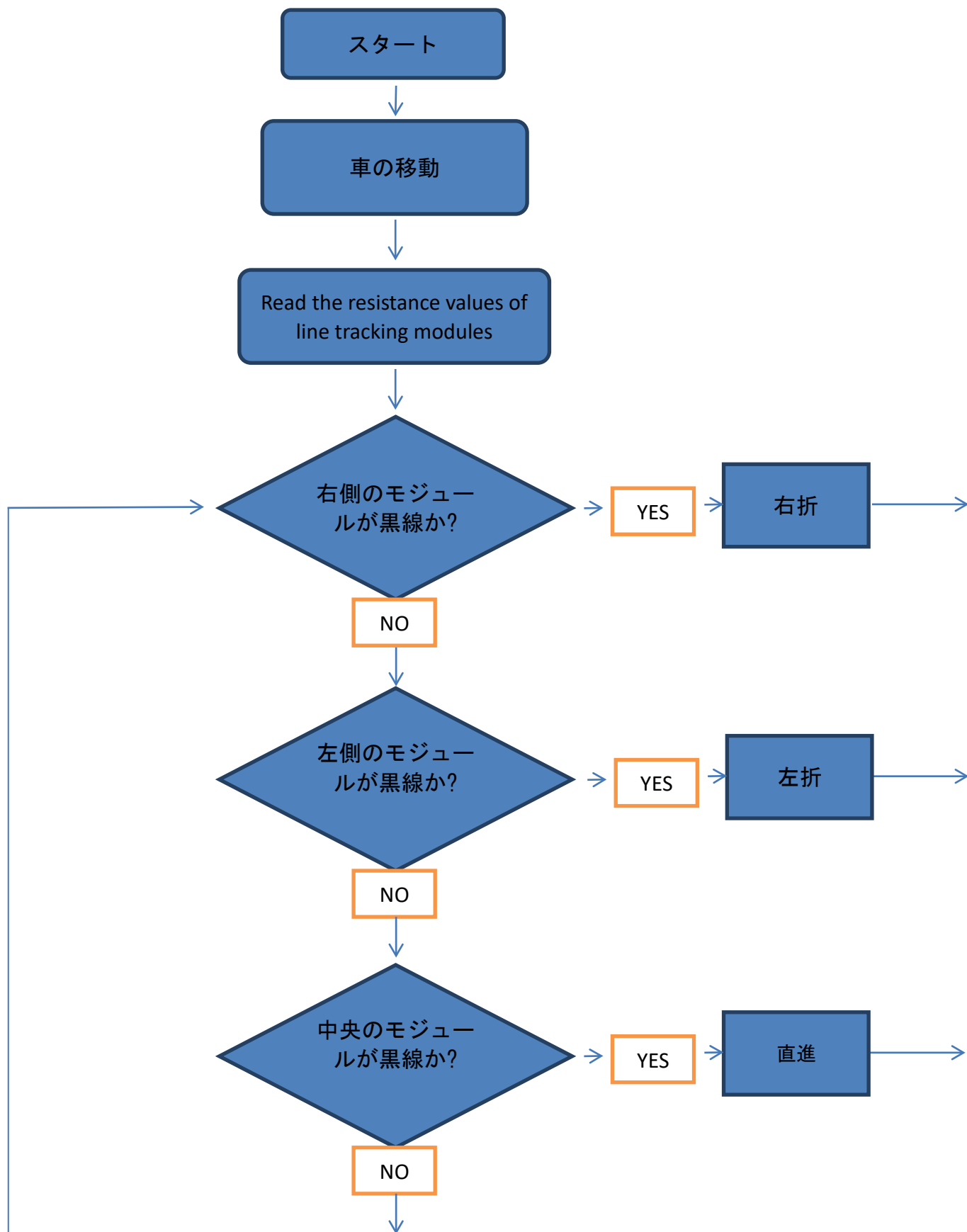
ライントラッキングセンサは、車の前方に位置する 2 つのコンポーネントで構成されています。ライントラッキングセンサは、赤外線送信管と赤外線受信管で構成されています。前者は赤外線を送信する LED であり、後者は赤外線を受光するだけの光抵抗である。黒色表面の光反射率は、白色表面の光反射率とは異なる。このため、黒路での赤外線反射光の反射強度と白路における赤外線反射強度とが異なり、抵抗値も変化する。直列抵抗間の電圧分割の原理を使い、センサの電圧から車の下の道路の色を推定することによって動作経路を決定することができる。



a→車は黒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。彼らは黒い線を検出することはできません。

b→車は右に移動した場合。左側のモジュールは黒線を検出し、コントローラボードに信号を送り、車は左に曲がります。

c→車は左に移動した場合。右側のモジュールは黒線を検出し、コントローラボードに信号を送信し、右折します。



上から、私達はライン追跡車の原理を理解ができます。車が始動した後、ライン追跡モジュールは、路面上の黒線を検知し、プログラムに従って対応する動作を行うだけでよい。

これは、車線追跡プログラムの簡単なアルゴリズムチャートです。PID のようなもっと複雑なアルゴリズムがあります。したがって、ライントラッキングの機能を実現させた後、自分で車を制御するアルゴリズムをもっと学ぶことができます。

小さなヒント

(1) ラインの曲げ部分はできるだけ滑らかでなければならない。コーナリング半径が小さすぎると、車は走路を越えて移動する可能性が非常に高くなります。

(2) ライントラッキングシーンは、パスと区別できる白黒テープまたは任意の色の紙で作ることができる。

(3) ライントラッキングに加えて、モーションに関係なく地域内に車を閉じ込めるなどのライントラッキングの原則に基づいて、想像力を広げて他のプログラムを開発することができます。