

追いかけてロボットを作ろう

障害物センサーを使って物体追従を行う

このレッスンで身につける力

- ☐ トラッキングセンサーを正しく取り付けられる
- ☐ ジャンパーワイヤーを正しく接続出来る
- ☐ トラッキングセンサーの感度を調整できる
- ☐ サンプルコードを実行できる
- ☐ 条件式(and or)の書き方を理解してコードを修正できる
- ☐ コースを走破するためにサンプルコードを修正できる

ミッションの準備

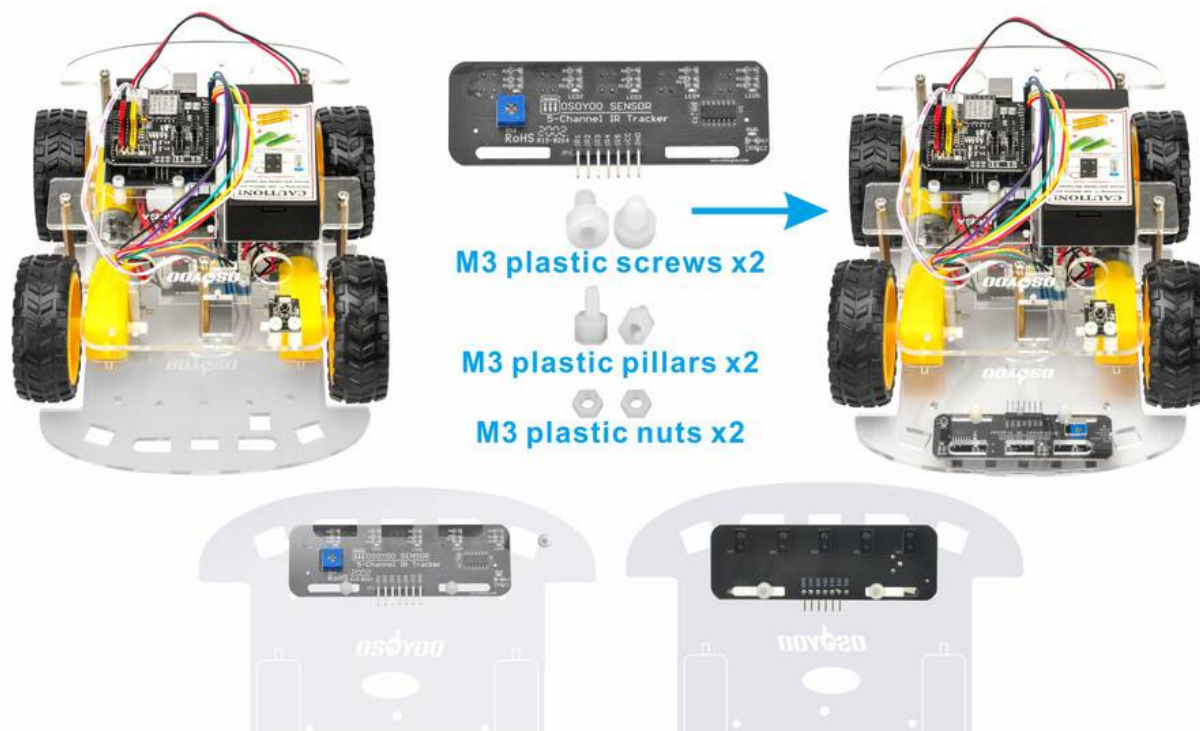
- ☐ Osoyoo UNO Board x 1
- ☐ 赤外線コントローラー
- ☐ トラッキングセンサーモジュール
- ☐ アクティブブザーモジュールx 1
- ☐ F/Mジャンパー
- ☐ USBケーブルx 1
- ☐ パソコン x 1

ミッションチャレンジ

今回のレッスンではこの線上を自動で走破してもらおう！

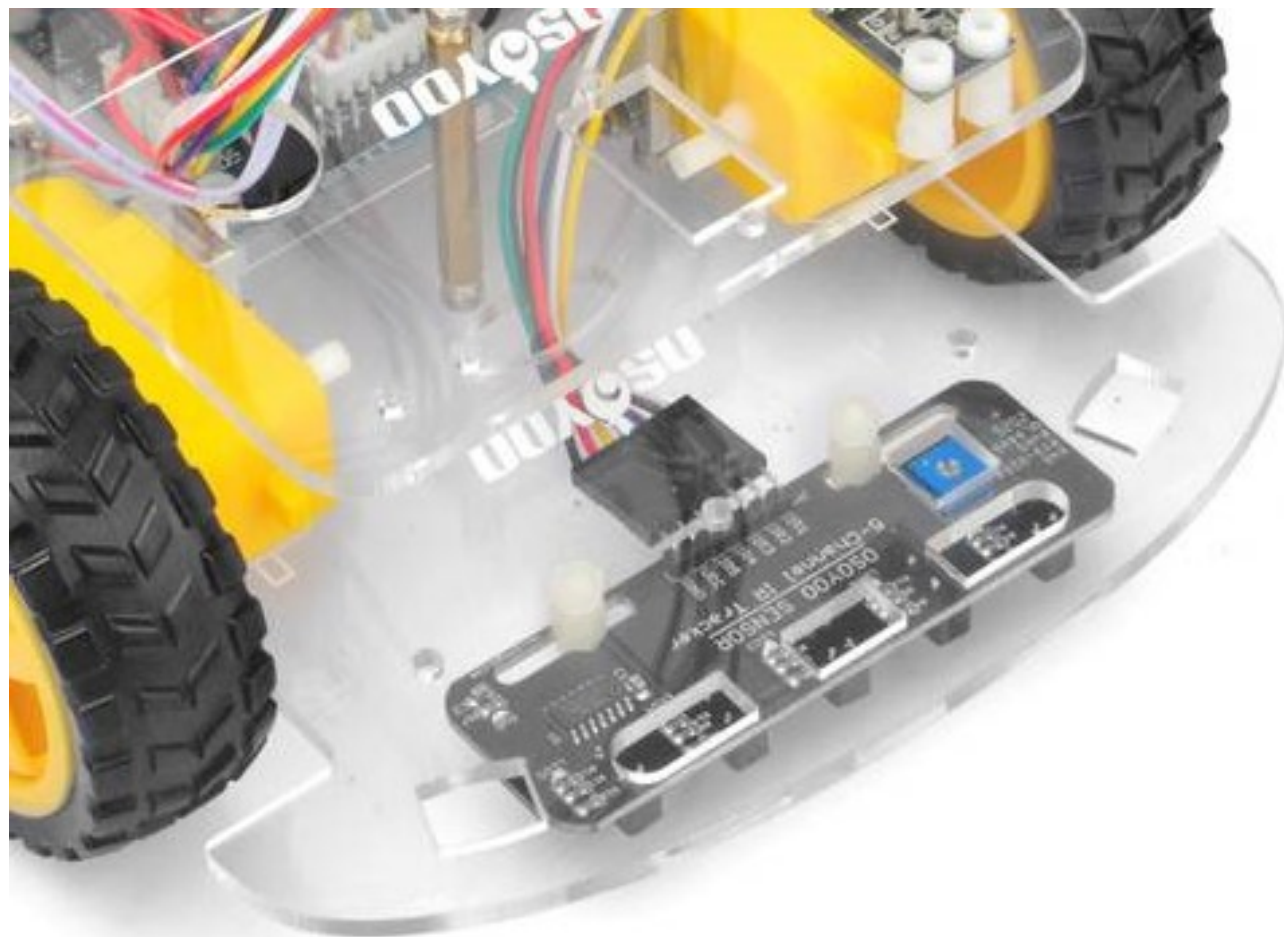


トラッキングセンサーを正しく取り付けられる

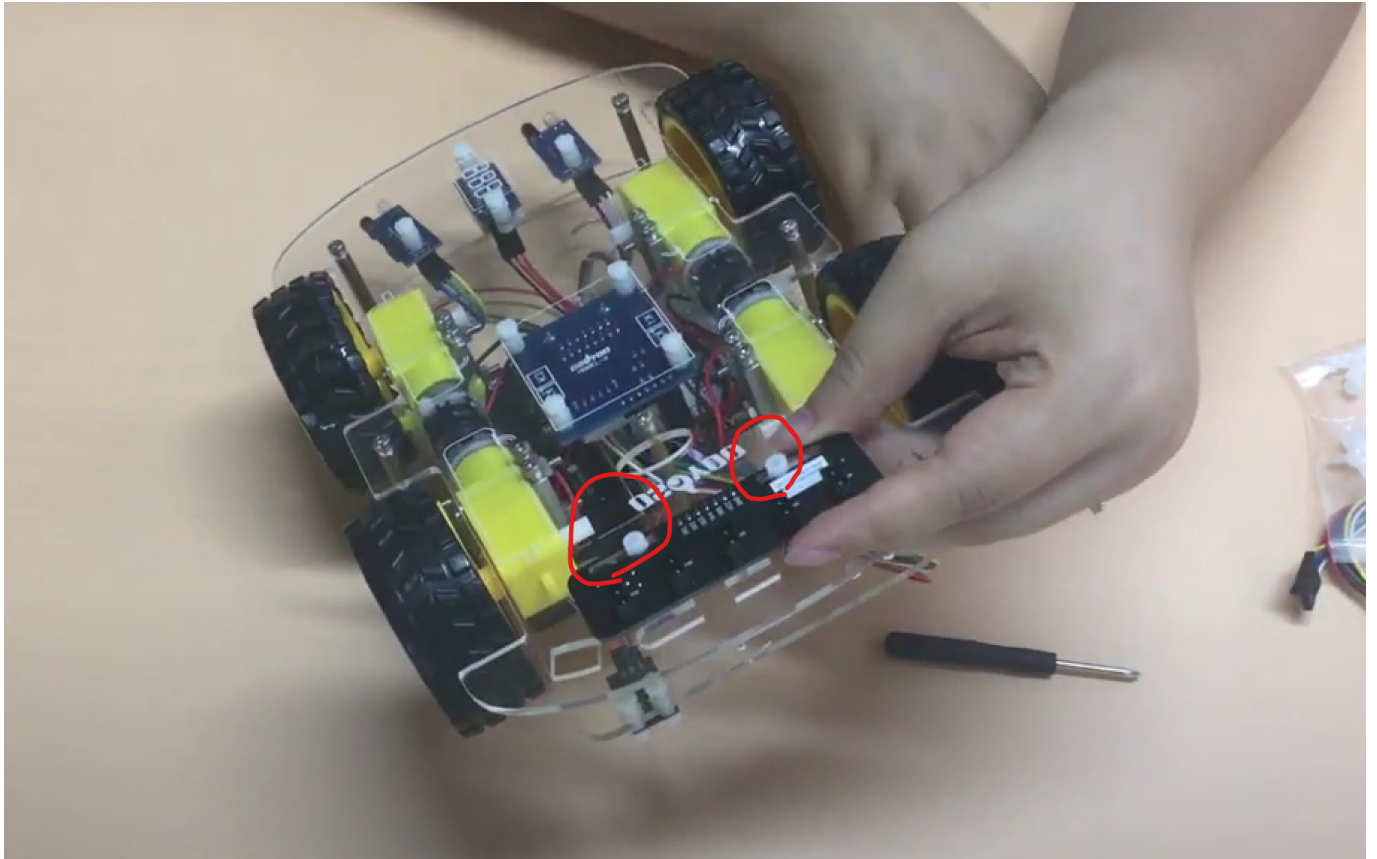


2個のM3プラスチックネジ、プラスチックピラー、およびプラスチックナットを使用して、追跡センサーモジュールを下部シャーシの下面側に取り付けます。

近くで見るとこんな感じだよ。



裏側から見るとこんな感じだよ.

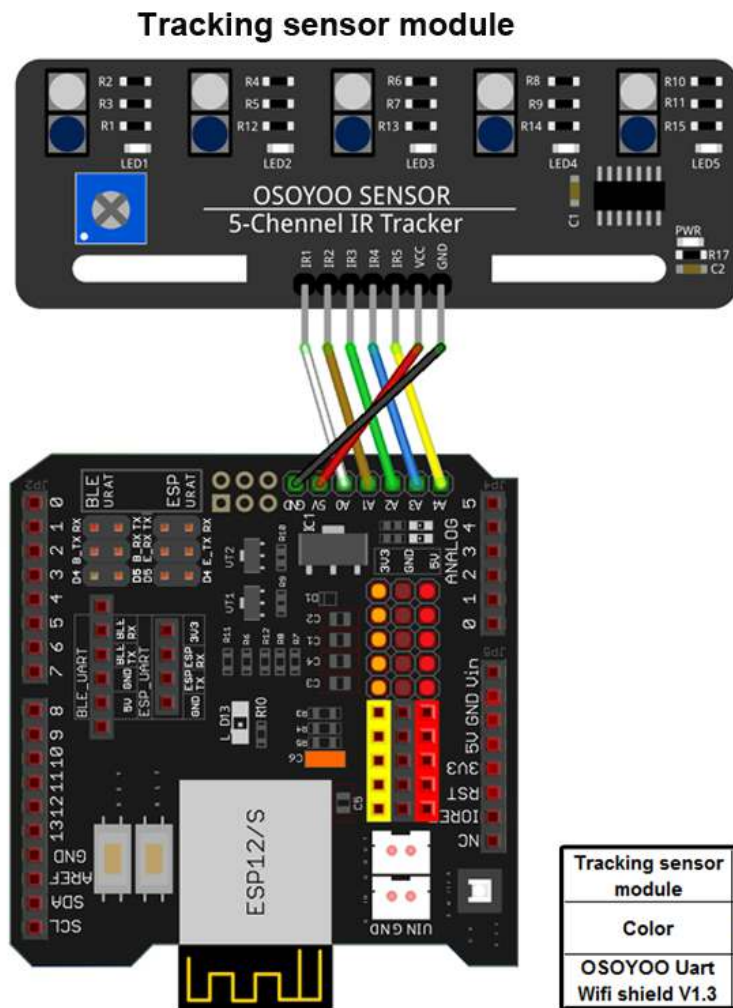


- ☐ トラッキングセンサーを正しく取り付けられる

ジャンパーワイヤーを正しく接続出来る

トラッキングセンサーモジュールのGNDとVCCピンをOSOYOO UartWiFiシールドV1.3のボードGNDと5Vにそれぞれ接続します.

下の図に示すように、IR1、IR2、IR3、IR4、IR5ピンをA0、A1、A2、A3、A4に7ピン25cmメス-メスケーブルで接続します.



7pin 25cm Female to Female Cable

Tracking sensor module	IR1	IR2	IR3	IR4	IR5	VCC	GND
Color	White	Ochre	Blue	Green	Yellow	Red	Black
OSOYOO Uart Wifi shield V1.3	A0	A1	A2	A3	A4	5V	GND

OSOYOO Uart WiFi shield V1.3

- ジャンパーワイヤーを正しく接続出来る

サンプルコードを実行できる

スケッチに以下のコードをコピー&ペーストして、スケッチを実行してみよう。

```
#define m1 7 //右モーター-MA1
#define m2 8 //右モーター-MA2
#define m3 12 //左モーター-MB1
#define m4 11 //左モーター-MB2
#define e1 9 //右モーター使用可能ピンEA
#define e2 6 //左モーター使用可能ピンEB

//*****5チャンネルIRセンサー接続*****//
#define ir1 A0
#define ir2 A1
#define ir3 A2
#define ir4 A3
#define ir5 A4
//*****//
```

```
void setup() {
  pinMode(m1, OUTPUT);
  pinMode(m2, OUTPUT);
  pinMode(m3, OUTPUT);
  pinMode(m4, OUTPUT);
  pinMode(e1, OUTPUT);
  pinMode(e2, OUTPUT);
  pinMode(ir1, INPUT);
  pinMode(ir2, INPUT);
  pinMode(ir3, INPUT);
  pinMode(ir4, INPUT);
  pinMode(ir5, INPUT);
  Serial.begin(9600);
}

void loop() {
  //センサー値の読み取り
  int s1 = digitalRead(ir1); //左端のセンサ
  int s2 = digitalRead(ir2); //左センサー
  int s3 = digitalRead(ir3); //中央センサ
  int s4 = digitalRead(ir4); //右センサー
  int s5 = digitalRead(ir5); //右端のセンサ

  Serial.print(!s1);
  Serial.print(!s2);
  Serial.print(!s3);
  Serial.print(!s4);
  Serial.println(!s5);

  //中央のセンサの黒い線の場合
  if((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 1) && (s5 == 1))
  {
    //全速力で前進する
    analogWrite(e1, 120); //モーターの速度を0~255の範囲で調整できます
    analogWrite(e2, 120); //モーターの速度を0~255の範囲で調整できます
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
  }

  //左センサーの黒線の場合
  if((s1 == 1) && (s2 == 0) && (s3 == 1) && (s4 == 1) && (s5 == 1))
  {
    //左に移動
    analogWrite(e1, 150); //モーターの速度を0~255の範囲で調整できます
    analogWrite(e2, 150); //モーターの速度を0~255の範囲で調整できます
    digitalWrite(m1, LOW);
    digitalWrite(m2, LOW);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
  }
}
```



```
}

//センサの黒い線が一番左にある場合
if((s1 == 0) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 1))
{
    //全速力で左へ
    analogWrite(e1, 180); //モーターの速度を0~255の範囲で調整できます
    analogWrite(e2, 180); //モーターの速度を0~255の範囲で調整できます
    digitalWrite(m1, LOW);
    digitalWrite(m2, HIGH);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);

}

//右側のセンサの黒い線の場合
if((s1 == 1) && (s2 == 1) && (s3 == 1) && (s4 == 0) && (s5 == 1))
{
    //going right
    analogWrite(e1, 150); //モーターの速度を0~255の範囲で調整できます
    analogWrite(e2, 150); //モーターの速度を0~255の範囲で調整できます
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, LOW);
}

//右端のセンサーのみが黒線を検出した場合
if((s1 == 1) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 0))
{
    //going right with full speed
    analogWrite(e1, 180); //モーターの速度を0~255の範囲で調整できます
    analogWrite(e2, 180); //モーターの速度を0~255の範囲で調整できます
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, HIGH);
}

//中央と右のセンサーが黒線を検出した場合
if((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 0) && (s5 == 1))
{
    //going right
    analogWrite(e1, 150); //モーターの速度を0~255の範囲で調整できます
    analogWrite(e2, 150); //モーターの速度を0~255の範囲で調整できます
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, LOW);
}

//中央と左のセンサが黒線を検出した場合
if((s1 == 1) && (s2 == 0) && (s3 == 0) && (s4 == 1) && (s5 == 1))
```

```
{
    //going left
    analogWrite(e1, 150); //モーターの速度を0~255の範囲で調整できます
    analogWrite(e2, 150); //モーターの速度を0~255の範囲で調整できます
    digitalWrite(m1, LOW);
    digitalWrite(m2, LOW);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
}

//中央、左、左のほとんどのセンサが黒い線を検出した場合
if((s1 == 0) && (s2 == 0) && (s3 == 0) && (s4 == 1) && (s5 == 1))
{
    //going left
    analogWrite(e1, 150); //モーターの速度を0~255の範囲で調整できます
    analogWrite(e2, 150); //モーターの速度を0~255の範囲で調整できます
    digitalWrite(m1, LOW);
    digitalWrite(m2, LOW);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
}

//中央、右、一番右のセンサーが黒い線を検出した場合
if((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 0) && (s5 == 0))
{
    //going right
    analogWrite(e1, 150); //モーターの速度を0~255の範囲で調整できます
    analogWrite(e2, 150); //モーターの速度を0~255の範囲で調整できます
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, LOW);
}

//すべてのセンサーが黒い線上にある場合
if((s1 == 0) && (s2 == 0) && (s3 == 0) && (s4 == 0) && (s5 == 0))
{
    //stop
    digitalWrite(m1, LOW);
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, LOW);
}
}
```

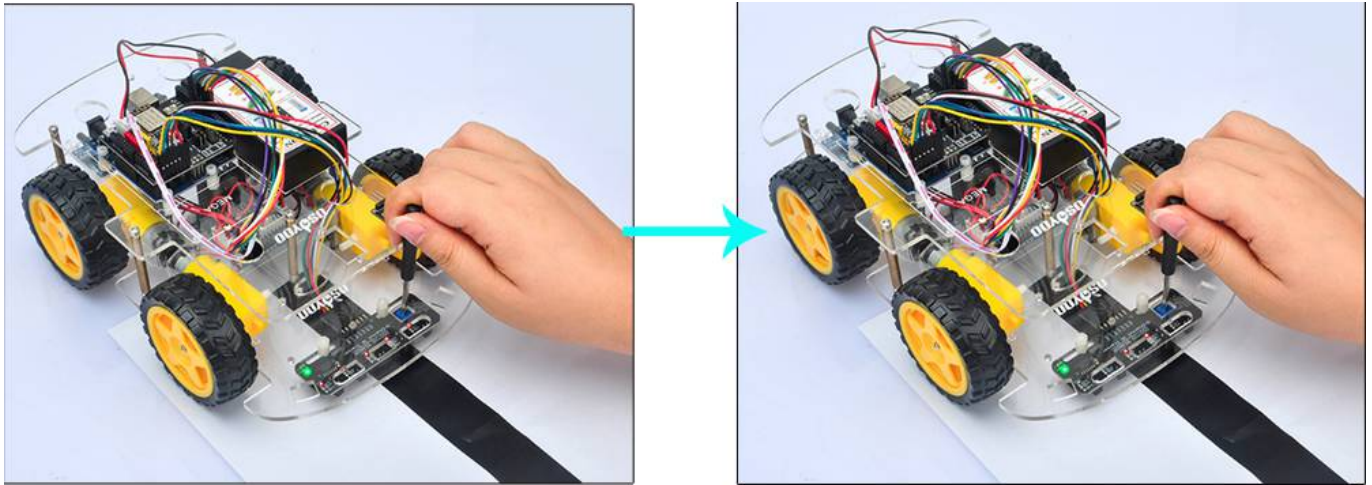
- ☐ サンプルコードを実行できる

トラッキングセンサーの感度を調整できる

トラッキングセンサーモジュールの感度を調整します。

車の電源をオンにして、フィリップスのドライバーでトラッキングセンサーのポテンショメーターを調整して、一番よい感度の状態にします。

信号はトラックの上にある黒いトラックの上にいる時にLEDライトが点灯し、センサーが白いトラックの上にあるときに消灯する様に調整します。



- ☐ トラッキングセンサーの感度を調整できる

条件式(and or)の書き方を理解してコードを修正できる

先ほどサンプルコードをコピー&ペーストしてもらったコードの中身を見てみよう！

```
//中央のセンサの黒い線の場合
if((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 1) && (s5 == 1))
{
    //全速力で前進する
    analogWrite(e1, 120); //モーターの速度を0~255の範囲で調整できます
    analogWrite(e2, 120); //モーターの速度を0~255の範囲で調整できます
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
}
```

このようなコードの中に

&&

がたくさんあるのが確認できるかな？この記号の意味は次の例を見て理解していこう！

5 && 7

この文の意味は「5かつ7」という意味になります。

要は5と7どちらの条件も満たしているという意味になります。

このコードと似たような意味で

```
||
```

という記号もあるよ。

例を示すよ。

```
5 || 7
```

この文の意味は「5または7」という意味になります。

要は5と7どちらかの条件を満たしているという意味になります。

上記のサンプルコードをもう一度確認してみよう！

```
//中央のセンサの黒い線の場合
if((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 1) && (s5 == 1))
{
    //全速力で前進する
    analogWrite(e1, 120); //モーターの速度を0~255の範囲で調整できます
    analogWrite(e2, 120); //モーターの速度を0~255の範囲で調整できます
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
}
```

このコードの意味は

センサ1が黒線の上にいるかつ

センサ2が黒線の上にいるかつ

センサ3が黒線の上にいるかつ

センサ4が黒線の上にいるかつ

センサ5が黒線の上にいるとき

```
//全速力で前進する
analogWrite(e1, 120); //モーターの速度を0~255の範囲で調整できます
analogWrite(e2, 120); //モーターの速度を0~255の範囲で調整できます
```

```
digitalWrite(m1, HIGH);  
digitalWrite(m2, LOW);  
digitalWrite(m3, HIGH);  
digitalWrite(m4, LOW);
```

この処理が実行されるという意味です。

要するに,

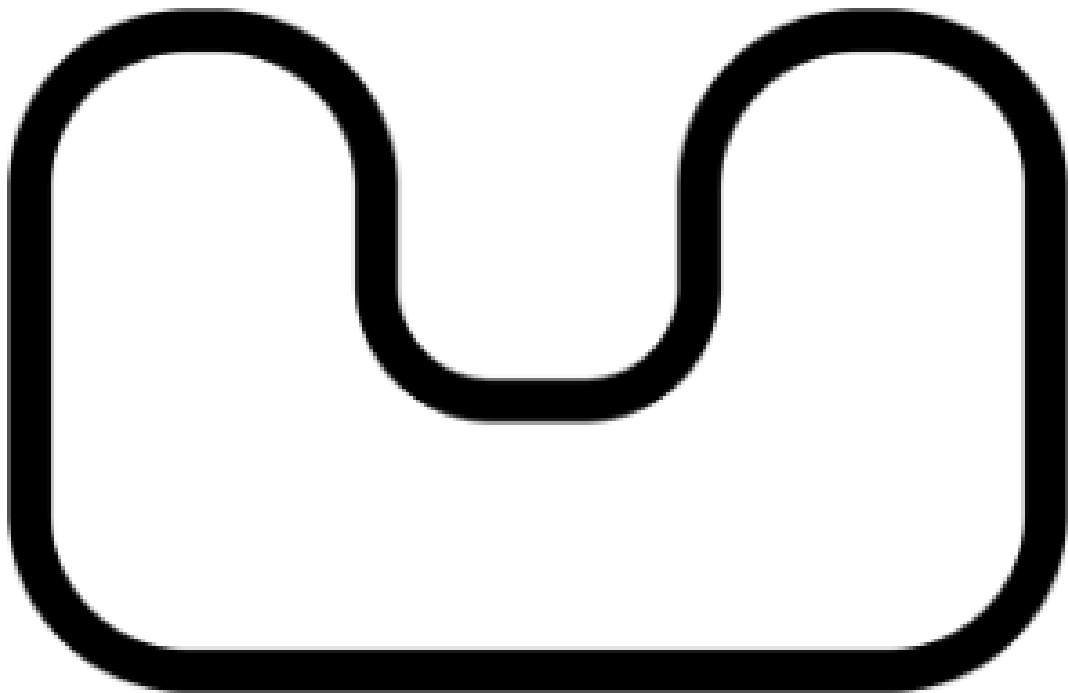
「センサ3だけが黒線の上にいるときにロボットが前進する」

という意味と同じです。

- ☐ 条件式(and or)の書き方を理解してコードを修正できる

コースを走破するためにサンプルコードを修正できる

このコースを走破してみよう。



カーブの部分の走行の速度調整をできるかな？

- ☐ コースを走破するためにサンプルコードを修正できる

まとめ

5 && 7

この文の意味は「5かつ7」という意味になります。

要は5と7どちらの条件も満たしているという意味になります

5 || 7

この文の意味は「5または7」という意味になります。

要は5と7どちらかの条件を満たしているという意味になります。

出来たことをチェックしよう

- ☐ トラッキングセンサーを正しく取り付けられる
- ☐ ジャンパーワイヤーを正しく接続出来る
- ☐ トラッキングセンサーの感度を調整できる
- ☐ サンプルコードを実行できる
- ☐ 条件式(and or)の書き方を理解してコードを修正できる
- ☐ コースを走破するためにサンプルコードを修正できる