**ライントレーサー解析結果**

このプログラムは、5つのラインセンサーを使って自動追跡（ライン追従）を行うロボットのためのものです。各センサーは黒線を検出する能力を持ち、その読み取り結果に応じてロボットの進行方向を決定します。詳細な解説は以下の通りです。

* 最初に、5つの文字配列 **char sensor[5]** を定義します。これは5つのラインセンサーの値を保存するためのものです。
* 次に **auto\_tracking** 関数を定義します。この関数は、ロボットの自動追跡機能を制御します。
* 関数内部で、5つのラインセンサーの読み取り結果をそれぞれの **sensor** 配列のインデックスに保存します。 **digitalRead** 関数は、指定されたピンからデジタル値（HIGHまたはLOW）を読み取ります。 **!** 演算子は、その値を反転させるために使用されます。したがって、 **!digitalRead(LFSensor\_0)** は、LFSensor\_0の値がHIGHであればLOWを、LOWであればHIGHを返します。
* その後、それぞれのセンサーの値に対応する重み（16, 8, 4, 2, 1）を掛け、その結果を **sensorvalue** に加えます。これは、センサーが検出したラインの位置に基づいて値を生成するためのものです。
* 続いて、 **sensorvalue** を2進数の文字列に変換し、その値を **sensorval** という新しい文字列に保存します。これは、センサーの読み取り結果を文字列として扱い、後続の条件分岐で比較するためのものです。
* **Serial.print(sensorval);** は、現在の **sensorval** の値をシリアルモニターに出力します。これはデバッグやモニタリングのために使用されます。
* その後、一連の **if** 文により、 **sensorval** の値に応じてロボットの行動を決定します。各 **if** 文は、特定のセンサーの組み合わせが黒線を検出したときに対応する行動（左折、右折、直進、停止）を定義します。
* それぞれの **if** 文の中で、ロボットは進行方向を決定し、モーターの速度を設定し、ハンドル（ステアリング）を制御し、その行動をシリアルモニターに出力します。
* 最終的に、センサーが全て黒線を検出した場合（**sensorval == "11111"**）、ロボットは停止します。これは、ロボットが黒線の終端（ストップライン）に到達したことを示しています。

以上が、このプログラムの全体的な流れと各部分の役割になります。

**詳細解説**

char sensor[5];

これは、5つのラインセンサーの値を保存するための文字配列を定義しています。

cCopy code

void auto\_tracking() {

この行では、 **auto\_tracking** という名前の関数を定義しています。この関数は、ロボットが自動追跡を行うための機能を制御します。

cCopy code

int sensorvalue = 32;

整数 **sensorvalue** を32で初期化します。後で、この変数に各センサーの読み取り結果を加えます。

次の5つの行では、5つのラインセンサーのデジタル読み取り結果をそれぞれの **sensor** 配列のインデックスに保存しています。それぞれのセンサーが黒線を検出すると、その位置の値は **HIGH**（1）になり、反対に何も検出しない場合は **LOW**（0）になります。

cCopy code

sensor[0] = !digitalRead(LFSensor\_0); sensor[1] = !digitalRead(LFSensor\_1); sensor[2] = !digitalRead(LFSensor\_2); sensor[3] = !digitalRead(LFSensor\_3); sensor[4] = !digitalRead(LFSensor\_4);

cCopy code

sensorvalue += sensor[0] \* 16 + sensor[1] \* 8 + sensor[2] \* 4 + sensor[3] \* 2 + sensor[4];

ここでは、センサーの読み取り結果（0または1）にそれぞれの重み（16, 8, 4, 2, 1）を掛けて合計し、その結果を **sensorvalue** に加えています。これは、5つのセンサーが検出した黒線の位置を2進数で表現するためのものです。

cCopy code

String sensorval = String(sensorvalue, BIN).substring(1, 6);

この行では、 **sensorvalue** を2進数の文字列に変換し、その文字列の部分文字列を新しい文字列 **sensorval** に保存しています。部分文字列は1～6文字目を取得しており、これは5つのセンサーの状態を表現しています。

cCopy code

Serial.print(sensorval);

この行は、現在の **sensorval** の値をシリアルモニターに出力します。これは、センサーの読み取り結果をモニタリングするためのものです。

次の一連の **if** ステートメントは、 **sensorval** の値に応じてロボットの行動を決定します。それぞれの **if** ステートメントは、特定のセンサーの組み合わせが黒線を検出したときのロボットの行動（左折、右折、直進、停止）を定義します。

cCopy code

if (sensorval == "01000" || sensorval == "10000" || sensorval == "11000") { go\_Advance(); analogWrite(ENA, HI\_SPEED); steer(LEFT); Serial.println(" LEFT"); }

この **if** ステートメントは、センサーが黒線を車の左側で検出した場合のロボットの行動を定義します。ロボットは前進し、ハイスピードで左に進むようにステアリングを操作します。その結果をシリアルモニターに " LEFT" と出力します。

以下の **if** ステートメントも同様のパターンで、異なるセンサーの組み合わせに応じてロボットの行動を制御します。

cCopy code

if (sensorval == "11111") { stop\_Stop(); //The car front touch stop line, need stop steer(FRONT); Serial.println("STOP"); }

最後に、すべてのセンサーが黒線を検出した場合（つまり **sensorval** が "11111" になった場合）、ロボットは停止します。これは、黒線の終端（ストップライン）に到達したことを示しています。

cCopy code

}

この行は **auto\_tracking** 関数の終了を表しています。