

RT トレーニングトレーサーを使った研修会(六日目)

目標 復習とカーブマーカースを見よう

前回までの講習会で、「出るレベル」に到達したはずですが。

次の目標は「第2 走行をする」、そこで新型コロナ感染拡大で全て停止となったと記憶しています。まずはプログラム(Exercise19.ino)を読み直し、資料と一緒に復習して下さい。

今回はここから再スタートしますが、コース記憶について変更します。

理由はいつもRT トレーサで使っているような高価なエンコーダを購入することはできません。特に自作する場合マイコンカー部品を流用するためです。

前回の「.コースを記憶する」

ロボットレース競技のコース長は最大 60m と設定されています。

カーブの最小局半径 10cm ということもあり、10cm を 1 区画としコース全長を覚えることとする。

(1) 区画数

$60\text{m} \div 10\text{cm} = 600$ 区画

(2) 確保

`int wmap[600], wmap2[600] wmap3[600]` 確保

(3) `wmap[]` の内容

初期値 0、ラインレースの制御量(Line_Control)を記録

(4) `wmap2[]` の内容

`wmap[]` から作成。初期化 0、

その区画が直線ならば 1、カーブならば 2 を記録。



変更

今回の「.コースを記憶する」提案

カーブマーカースを読んで区画を確定、区画の状況を記録する。

本日は復習優先です。

課題は

探査走行時、カーブマーカースが見つかったらブザーを鳴らしながら走行しなさい。

RT トレーニングトレーサーを使った研修会(六日目)

```
/*
 * Arduino_Exercises/Exercise19/Exercise19.ino
 *
 * Copyright(C) 2020 RT Corporation <support@rt-net.jp>
 * All rights reserved.
 *
 * License: Apache License, Version 2.0
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 */
//yamagata robot club
//ピンの設定
int DIR_R_Pin = D12; //右モータ方向
int DIR_L_Pin = D0; //左モータ方向
int PWM_R_Pin = D11; //右モータPWM端子
int PWM_L_Pin = D10; //左モータPWM端子
int SW1_Pin = D7; //左SW
int SW2_Pin = D8; //右SW
int LINE_L2_Pin = A5; //左外センサ
int LINE_L1_Pin = A4; //左内センサ
int LINE_R1_Pin = A3; //右内センサ
int LINE_R2_Pin = A2; //右外センサ
int LED_Pin = D13; //LED
int BUZZER_Pin = D2; //ブザー
// 変数宣言
long Line_Controller;
//int inside_offset = -34; //org
//int outside_offset = -146; //org
int inside_offset = -23;
int outside_offset = 5;
int CW_R = LOW; //右モータ前進
int CCW_R = HIGH; //右モータ後進
int CW_L = HIGH; //左モータ前進
int CCW_L = LOW; //左モータ後進
int PWM_R_Value = 0; //右モータPWM=0
int PWM_L_Value = 0; //左モータPWM=0
int cross_line_flag = 0; //クロスラインフラグ
//ラインが白の場合:1、ラインが黒の場合-1
//int Line_signed = 1; //ライン白(本番)
int Line_signed = -1; //ライン黒(自宅練習)
```

RT トレーニングトレーサーを使った研修会(六日目)

```

int counter = 0;

void LED_DRIVER(char n, int on_ms, int off_ms){
for( char i = 0; i < n ;i++){
digitalWrite(LED_Pin, HIGH); // LED点灯
delay(on_ms);
digitalWrite(LED_Pin, LOW); // LED消灯
delay(off_ms);
}
}

void BUZZER_DRIVER(char n, int on_ms, int off_ms){
for( char i = 0; i < n ;i++){
digitalWrite(BUZZER_Pin, HIGH); //ブザー-on
delay(on_ms);
digitalWrite(BUZZER_Pin, LOW); //ブザー-off
delay(off_ms);
}
}

void MOTOR_STOP(void)
{
digitalWrite(LED_Pin, LOW);
analogWrite(PWM_L_Pin, 0);
analogWrite(PWM_R_Pin, 0);
digitalWrite(BUZZER_Pin, HIGH);
delay(100);
digitalWrite(BUZZER_Pin, LOW);
while(1) {
if (digitalRead(SW1_Pin) == LOW) {
digitalWrite(LED_Pin, HIGH);
break;
}
}
}

void TRACE(void)
{
//ラインセンサの値から制御量を算出する
Line_Control = (analogRead(LINE_L1_Pin) - analogRead(LINE_R1_Pin) -
inside_offset)
+ 2 * (analogRead(LINE_L2_Pin) - analogRead(LINE_R2_Pin) -
outside_offset);
PWM_L_Value = 188 - Line_signed * Line_Control / 10; //80(org)
120(自宅コースOK) 200,190(自宅コース×) 160,180 185(自宅コースOK)

```

RT トレーニングトレーサーを使った研修会(六日目)

```

PWM_R_Value = 188 + Line_signed * Line_Control / 10; //80(org)
120(自宅コースOK)
//左モーターPWM出力
if( PWM_L_Value < 0 ) digitalWrite(DIR_L_Pin, CCW_L); //モーター後進設定
else digitalWrite(DIR_L_Pin, CW_L); //モーター前進設定
PWM_L_Value = abs(PWM_L_Value);
if( PWM_L_Value > 255 ) PWM_L_Value = 255;
//モーター制御値上下ガード処理(上限PWM100%)
if( PWM_L_Value <= 0 ) PWM_L_Value = 0; //モーター制御値上下ガード処理(下限PWM
0%)
analogWrite(PWM_L_Pin, PWM_L_Value); //左モータPWM出力
//右モーターPWM出力
if( PWM_R_Value < 0 ) digitalWrite(DIR_R_Pin, CCW_R); //モーター後進設定
else digitalWrite(DIR_R_Pin, CW_R); //モーター前進設定
PWM_R_Value = abs(PWM_R_Value);
if( PWM_R_Value > 255 ) PWM_R_Value = 255;
//モーター制御値上下ガード処理(上限PWM100%)
if( PWM_R_Value <= 0 ) PWM_R_Value = 0; //モーター制御値上下ガード処理(下限PWM
0%)
analogWrite(PWM_R_Pin, PWM_R_Value); //右モータPWM出力
delay(1);
//PWM、コントロールの値確認用
//Serial.printf(“%n¥r PWM_R=%d PWM_L=%d CONT=%d”,PWM_R_Value,PWM_L_Value,
Line_Control);
//delay(100);
}
void setup() {
// put your setup code here, to run once:
// IOポート設定
pinMode(BUZZER_Pin, OUTPUT); //ブザーピンは出力
pinMode(LED_Pin, OUTPUT); //LEDピンは出力
pinMode(SW1_Pin, INPUT_PULLUP); //SW1端子は入力 プルアップ
pinMode(SW2_Pin, INPUT_PULLUP); //Sw2端子は入力 プルアップ
pinMode(DIR_R_Pin, OUTPUT); //右モータ方向端子は出力
pinMode(DIR_L_Pin, OUTPUT); //左モータ方向端子は出力
LED_DRIVER(4,100,100);
Serial.begin(115200); //シリアル初期化
while (1) {
if( digitalRead(SW2_Pin)==LOW ) { //SW2(右SW)押されたら
//digitalWrite(BUZZER_Pin, HIGH); // ブザー鳴らす
//delay(100); // 指定時間待ち

```

RT トレーニングトレーサーを使った研修会(六日目)

```
//digitalWrite(BUZZER_Pin, LOW); // ブザー止める
//delay(100); // 指定時間待ち
BUZZER_DRIVER( 1,100,100);
while(1) {
Serial.printf("¥n¥r LL2=%d LL1=%d LR1=%d LR2=%d
inside_offset=%d outside_offset=%d ML=%d MR=%d",
analogRead(LINE_L2_Pin),
analogRead(LINE_L1_Pin), analogRead(LINE_R1_Pin), analogRead(LINE_R2_Pin),
(analogRead(LINE_L1_Pin) -
analogRead(LINE_R1_Pin)), (analogRead(LINE_L2_Pin) - analogRead(LINE_R2_Pin)),
adc_read_value(PB_1, 10),
adc_read_value(PB_0, 10));
delay(100);
if( digitalRead(SW2_Pin)==LOW ) {
//再度SW2(右SW)押されたら
delay(200);
break; //ループ(while(1))から抜ける
}
} // end of while(1)
}
//左のスイッチを押したら、走行開始
if( digitalRead(SW1_Pin)==LOW ) {
digitalWrite(LED_Pin, HIGH); // LED点灯
digitalWrite(BUZZER_Pin, HIGH); // ブザー鳴らす
delay(100); // 指定時間待ち
digitalWrite(BUZZER_Pin, LOW); // ブザー止める
delay(100); // 指定時間待ち
digitalWrite(BUZZER_Pin, HIGH); // ブザー鳴らす
delay(100); // 指定時間待ち
digitalWrite(BUZZER_Pin, LOW); // ブザー止める
delay(100); // 指定時間待ち
digitalWrite(BUZZER_Pin, HIGH); // ブザー鳴らす
delay(200); // 指定時間待ち
digitalWrite(BUZZER_Pin, LOW); // ブザー止める
delay(100); // 指定時間待ち
//BUZZER_DRIVER(2,70, 70);
break;
}
} //end of while(1)
}
void loop() {
```

RT トレーニングトレーサーを使った研修会(六日目)

```
// put your main code here, to run repeatedly:

int i;
//マーカを検出?
if( Line_signed == 1 ) {
//中央ライン白
//クロスラインチェック
if( cross_line_flag==0 ) {
if( ( analogRead(LINE_L1_Pin) + analogRead(LINE_R1_Pin) +
analogRead(LINE_L2_Pin) + analogRead(LINE_R2_Pin) ) > 2400 ) {
//while( analogRead(LINE_L1_Pin) + analogRead(LINE_R1_Pin) +
analogRead(LINE_L2_Pin) + analogRead(LINE_R2_Pin) ) > 2400 ); //クロスラインパスする
cross_line_flag = 1;
}
}
//マーカを検出?
if( adc_read_value(PB_0, 10) > 80 ) { //PBO:スタート・ゴールセンサ
//スタート・ゴールマーカ見つかった
while( adc_read_value(PB_0, 10) > 80 ); //スタート・ゴールマーカパスする
if( cross_line_flag==1 ) {
//見つかったマーカはクロスラインと判断
cross_line_flag = 0;
} else {
counter++;
cross_line_flag = 0;
}
if( counter > 1 ){
i = 0;
while( i<500 ) { // ロボットを完全にゴール区画に入れる
TRACE();
i++;
}
MOTOR_STOP(); // ロボットを停止
counter = 0;
}
} else {
//中央ライン黒ならば
//クロスラインチェック
if( cross_line_flag==0 ) {
if( ( analogRead(LINE_L1_Pin) + analogRead(LINE_R1_Pin) +
analogRead(LINE_L2_Pin) + analogRead(LINE_R2_Pin) ) < 500 ) {
```

RT トレーニングトレーサーを使った研修会(六日目)

```
//while( analogRead(LINE_L1_Pin) + analogRead(LINE_R1_Pin) +
analogRead(LINE_L2_Pin) + analogRead(LINE_R2_Pin) ) < 2400 ); //クロスラインパスする
cross_line_flag = 1;
}
}
if( adc_read_value(PB_0, 10) < 30 ) { //PB0:スタート・ゴールセンサ org:30
113のはず
//スタート・ゴールマーカ-見つかった
while( adc_read_value(PB_0, 10) < 30 ): //スタート・ゴールマーカ-パスする
org:30 113のはず
if( cross_line_flag==1 ) {
//見つかったマーカ-はクロスラインと判断
cross_line_flag = 0;
} else {
counter++;
cross_line_flag = 0;
}
if( counter > 1 ){
//見つかったマーカ-, 1本目はスタートマーカ-, 2本目がゴールマーカ-
i = 0;
while( i<500 ) { // ロボットを完全にゴール区画に入れる
//走行距離等改善の余地あり
TRACE();
i++;
}
MOTOR_STOP(); //ロボット停止
counter = 0;
}
}
}
//ラインセンサの値から制御量を算出する
TRACE();
}
```