徳島大学 大学開放実践センター 公開講座

無線で動くロボットを作ろう　第5回

徳島大学技術支援部
徳島大学社会産業理工学研究部総合技術センター
辻　明典
E-mail: a-tsuji@is.tokushima-u.ac.jp

2017年11月7日（火）

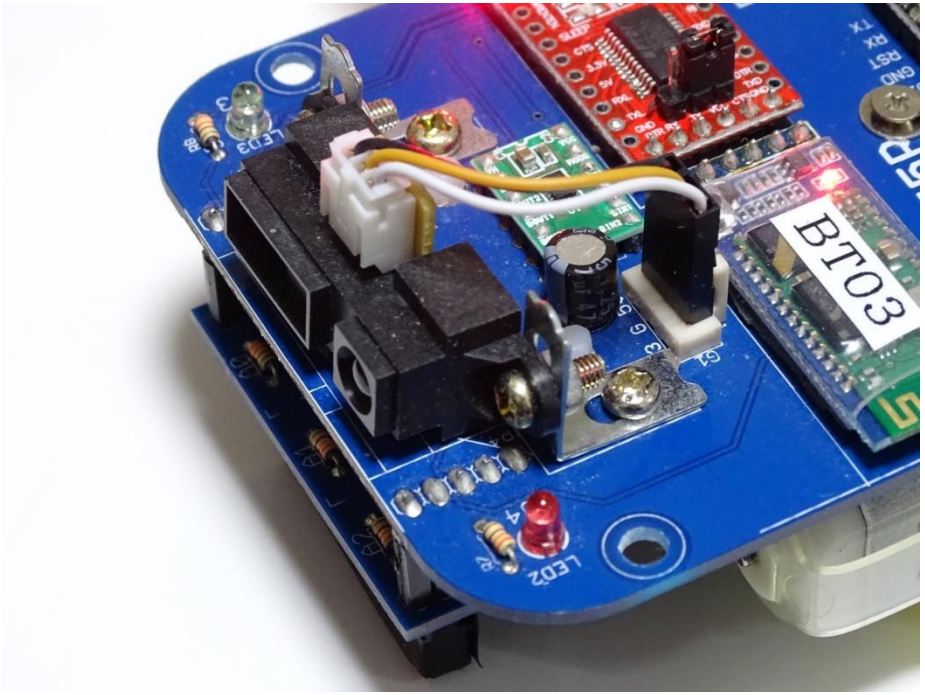
講座日程

- 無線で動くロボットを作ろう
- 講師：辻 明典(徳島大学技術支援部)
  桑折 範彦(徳島大学名誉教授)
  川上　博(徳島大学名誉教授)
- 曜日・時間：土曜日　10時00分～11時30分
- スケジュール：
  ①　10／7　　概要，ロボットの開発環境
  ②　10／14　ロボットのモーター１(基本動作)
  ③　10／21　ロボットのモーター２(応用動作)
  ④　10／28　ロボットのセンサー１(距離センサ，有線・無線通信)
  ⑤　11／11　ロボットのセンサー２(フォトリフレクタ)
  ⑥　11／18　ロボットの制御１(モータ・センサの協調動作)
  ⑦　11／25　ロボットの制御２(ライントレース)

本日の予定

- 前回の復習
  - 距離センサとLED
  - 距離センサとモーター
  - スタートボタン
- フォトリフレクタ
  - ロボットへの取り付け
  - 動作原理
  - 動作確認
- フォトリフレクタの応用
  - ラインの検出（アナログ）
  - ラインの検出（デジタル）

講座資料（スライド，サンプルスケッチ等）
https://goo.gl/K44cPc



Example0404: 距離に応じてLED点灯

- ロボットがある範囲の距離に入ったときにLEDを点灯

- if文を使う

```
if (d >= 0 && d < 100) {          // 0mm以上 100mm未満
    ・・・（LED赤点灯）
} else if (d >= 100 && d < 200) { // 100mm以上 200mm未満
    ・・・（LED緑点灯）
} else if (d >= 200) {            // 200mm以上
    ・・・（LED黄点灯）
} else {                          // それ以外
    ・・・（LED消灯）
}
```

（例）LED赤だけ点灯

```
digitalWrite(LED_R_PIN, HIGH);
digitalWrite(LED_G_PIN, LOW);
digitalWrite(LED_Y_PIN, LOW);
```

コードが冗長

Example0405: led_on関数

- ロボットがある範囲の距離に入ったときにLEDを点灯

- led_on関数をつくる

```
void led_on(bool r, bool g, bool y) {
  digitalWrite(LED_R_PIN, r);
  digitalWrite(LED_G_PIN, g);
  digitalWrite(LED_Y_PIN, y);
}
```

```
 if (d >= 0 && d < 100) {            // 0mm以上 100mm未満
   led_on(0, 0, 1)      // 赤点灯
} else if (d >= 100 && d < 200) {  // 100mm以上 200mm未満
   led_on(0, 1, 0);     // 緑点灯
} else if (d >= 200) {              // 200mm以上
   led_on(1, 0, 0);     // 黄点灯
} else {                             // それ以外
   led_on(0, 0, 0);      // 消灯
}
```

Example0406: モーター関数を追加（Example0304より）

- ロボットがある範囲の距離に入ったときにLEDを点灯

- 距離に応じてモーターを前進，停止，後退

```
if (d >= 0 && d < 100) {            // 0mm以上 100mm未満
   bwd(1);    // 後退
} else if (d >= 100 && d < 200) {  // 100mm以上 200mm未満
   stp(1);     // 停止
} else if (d >= 200) {              // 200mm以上
   fwd(1);     // 前進
} else {                            // それ以外
    stp(1);     // 停止
}
```

停止の距離範囲を狭くしていくとどうなるか？

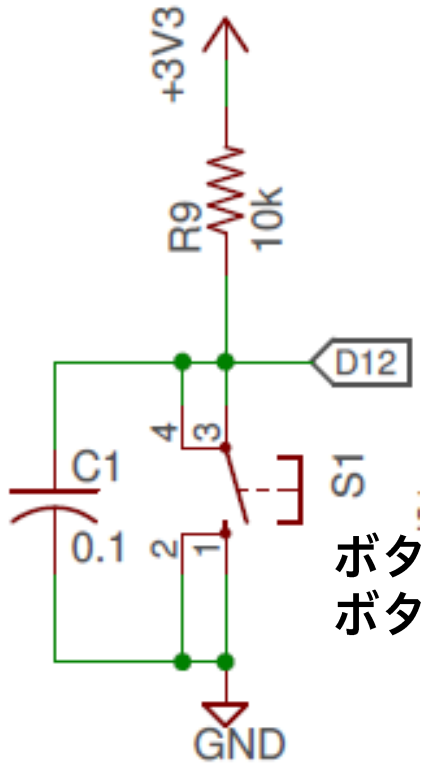
Example0407: スタートボタン

- スイッチ(D12)を押したらロボット動作開始
  - while文を使う
- スイッチが押されるまで待機
  - while (digitalRead(SW_PIN) == HIGH);

```
while (条件) {
    コード;
}
```

while文：条件が成立している間,
{ } のコードを繰り返す

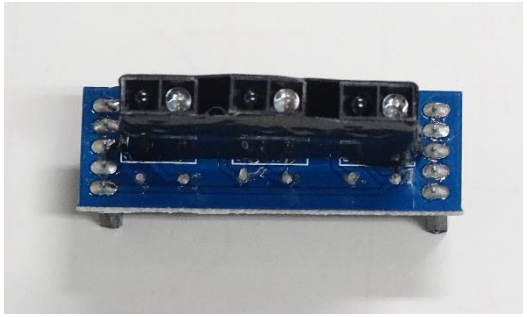
スイッチ(D12)の回路



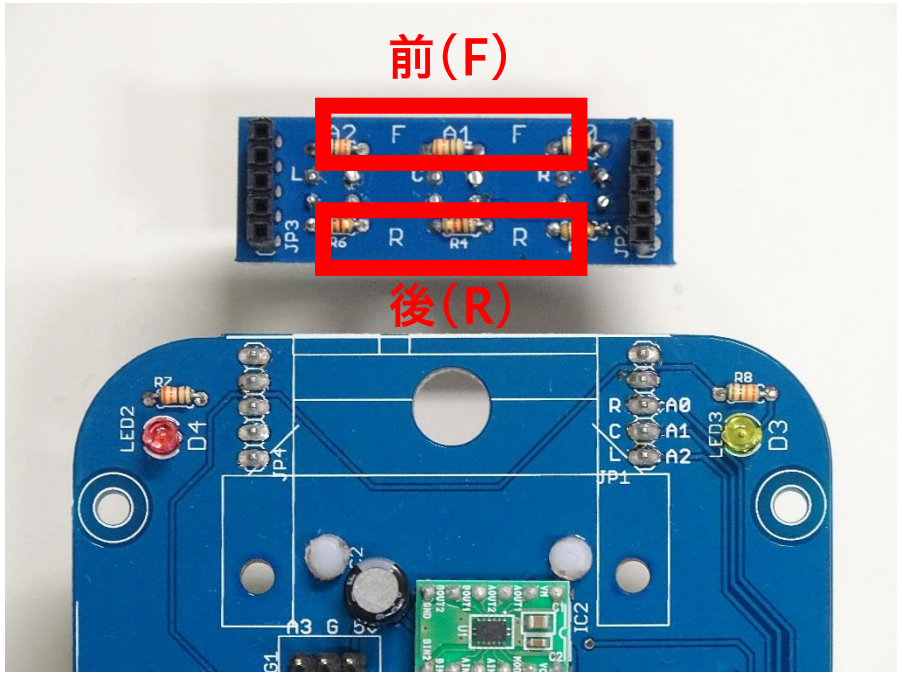
ボタンが離れた：HIGH
ボタンが押された：LOW

フォトリフレクタの取り付け

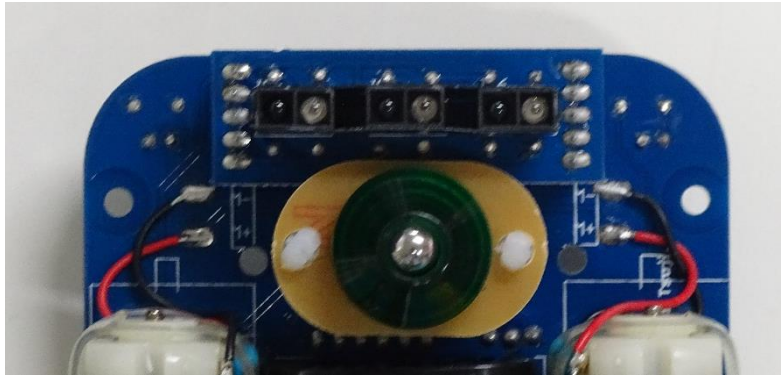
- ロボットにフォトリフレクタを取り付け
  ※ 取り付け向き注意（基板のF，Rを確認）



フォトリフレクタ



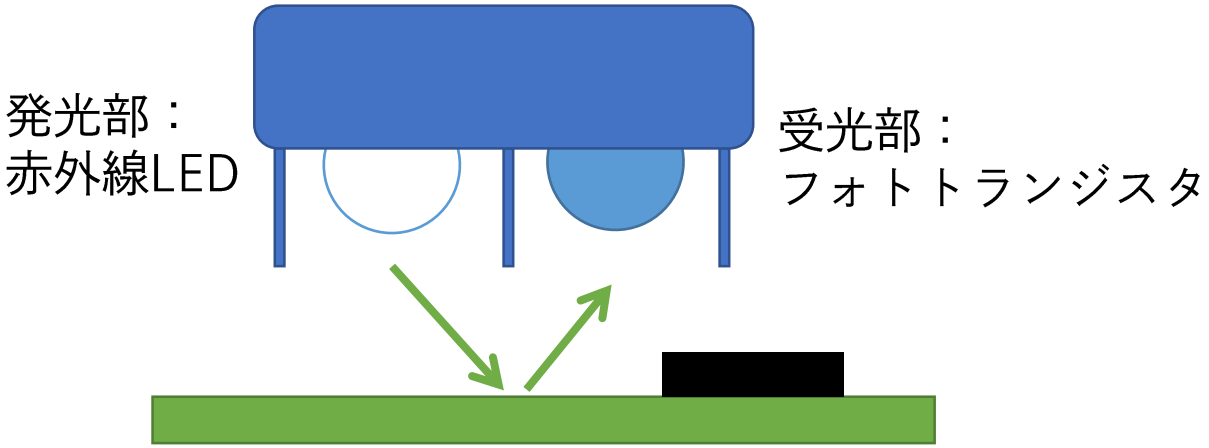
取り付け向き



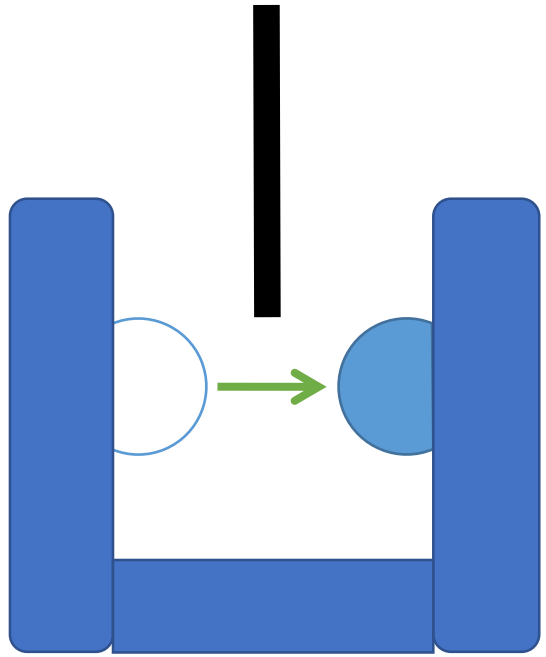
取り付け向き(ロボット裏)

フォトリフレクタの動作原理

- 赤外線の光を使い、物体の光の反射を用いて計測
  - 反射型と透過型の2種類
  - 検出距離：1mm～10mm程度
  - プリンタ，モーター，スマートフォンなど



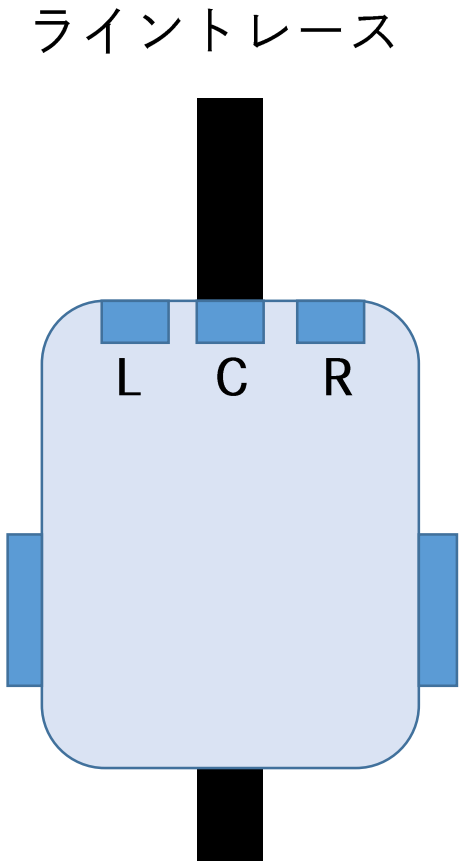
反射型
（フォトリフレクタ）

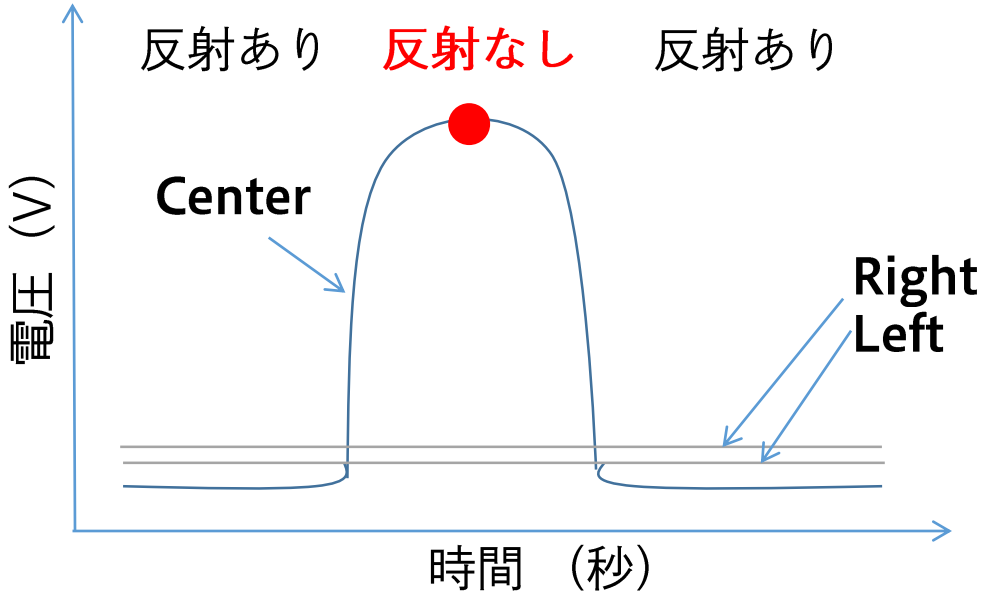


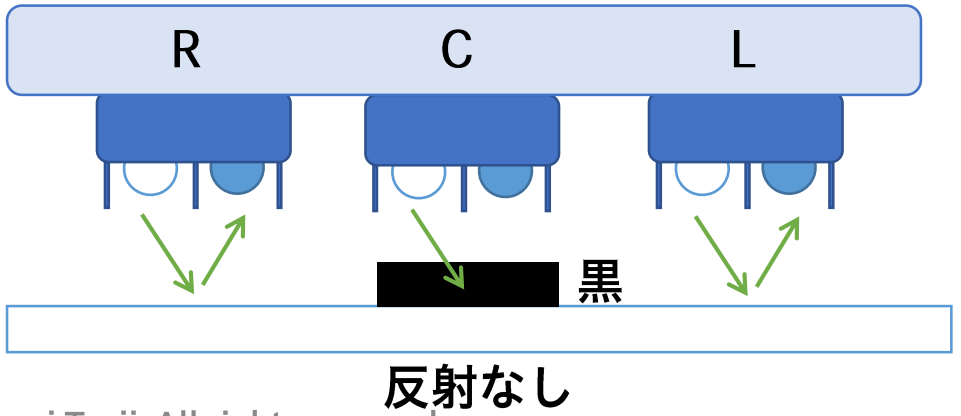
透過型
（フォトインタラプタ）

フォトリフレクタの応用

- ロボットのライントレース
  - 線(ライン)の有無の検出





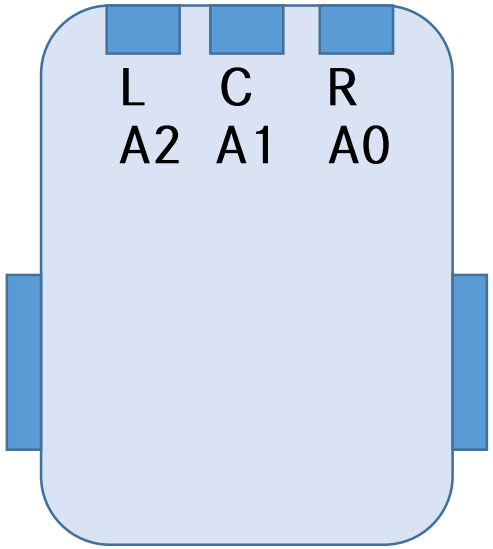


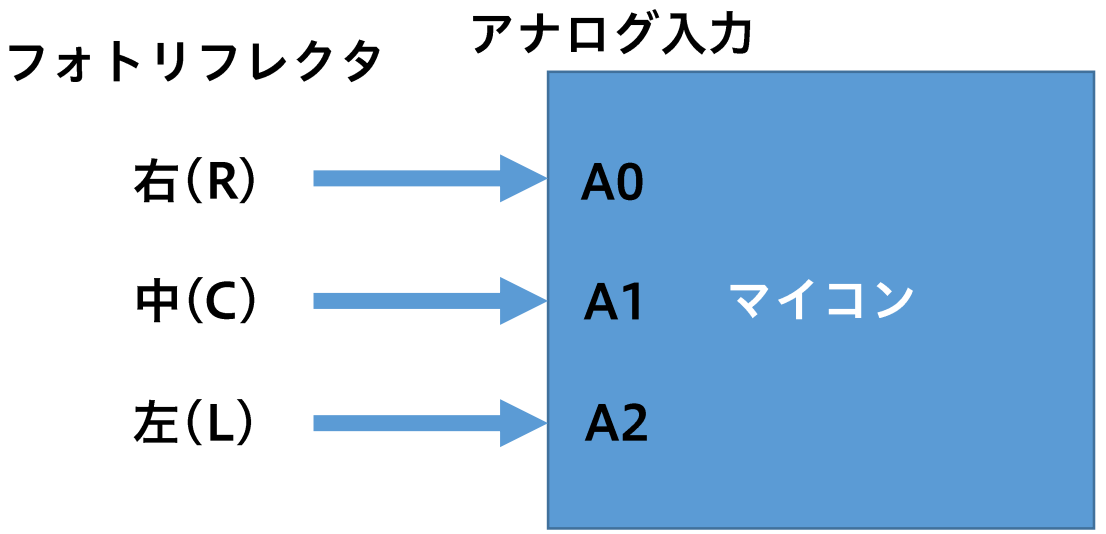
フォトリフレクタの使用方法

- ロボットのフォトリフレクタは3つ
  - 右(R)，中(C)， 左(L)
- フォトリフレクタとマイコンの接続
  - 右： A0 (アナログ入力0)
  - 中： A1 (アナログ入力1)
  - 左： A2 (アナログ入力2)

アナログ入力
int pr_r = analogRead(A0);

A0: 0 – 1023
A1: 0 – 1023
A2: 0 – 1023



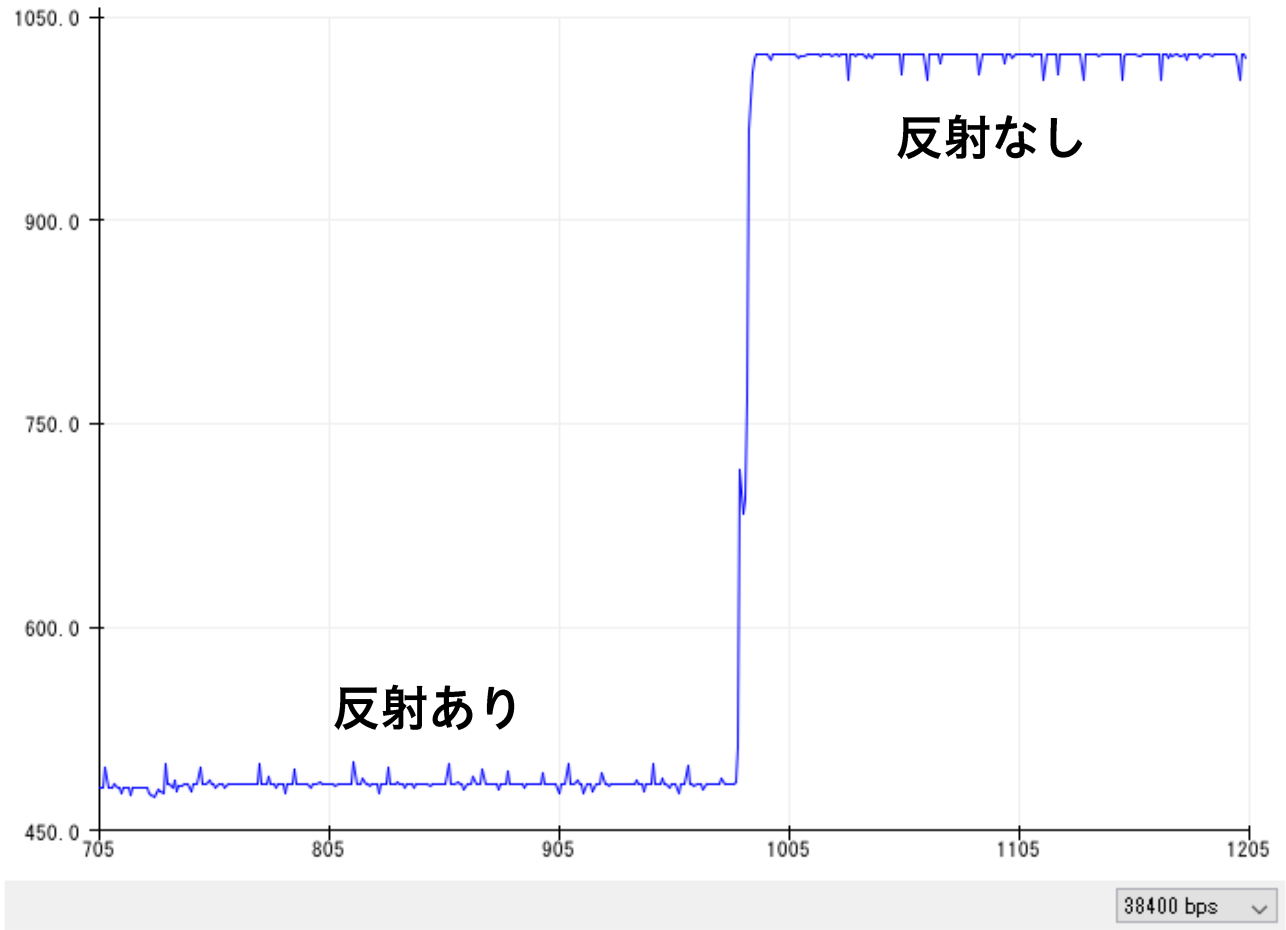


Example0501: フォトリフレクタの動作確認

- フォトリフレクタの動作確認
- シリアルプロッタで確認
  - 通信速度を38,400bpsに設定
- 右(A0)，中(A1)，左(A2)を順に確認

```
void setup() {
  Serial.begin(38400);
  delay(1000);
}
void loop() {
  int pr_r = analogRead(A0);
  Serial.println(pr_r);
  delay(50);
}
```

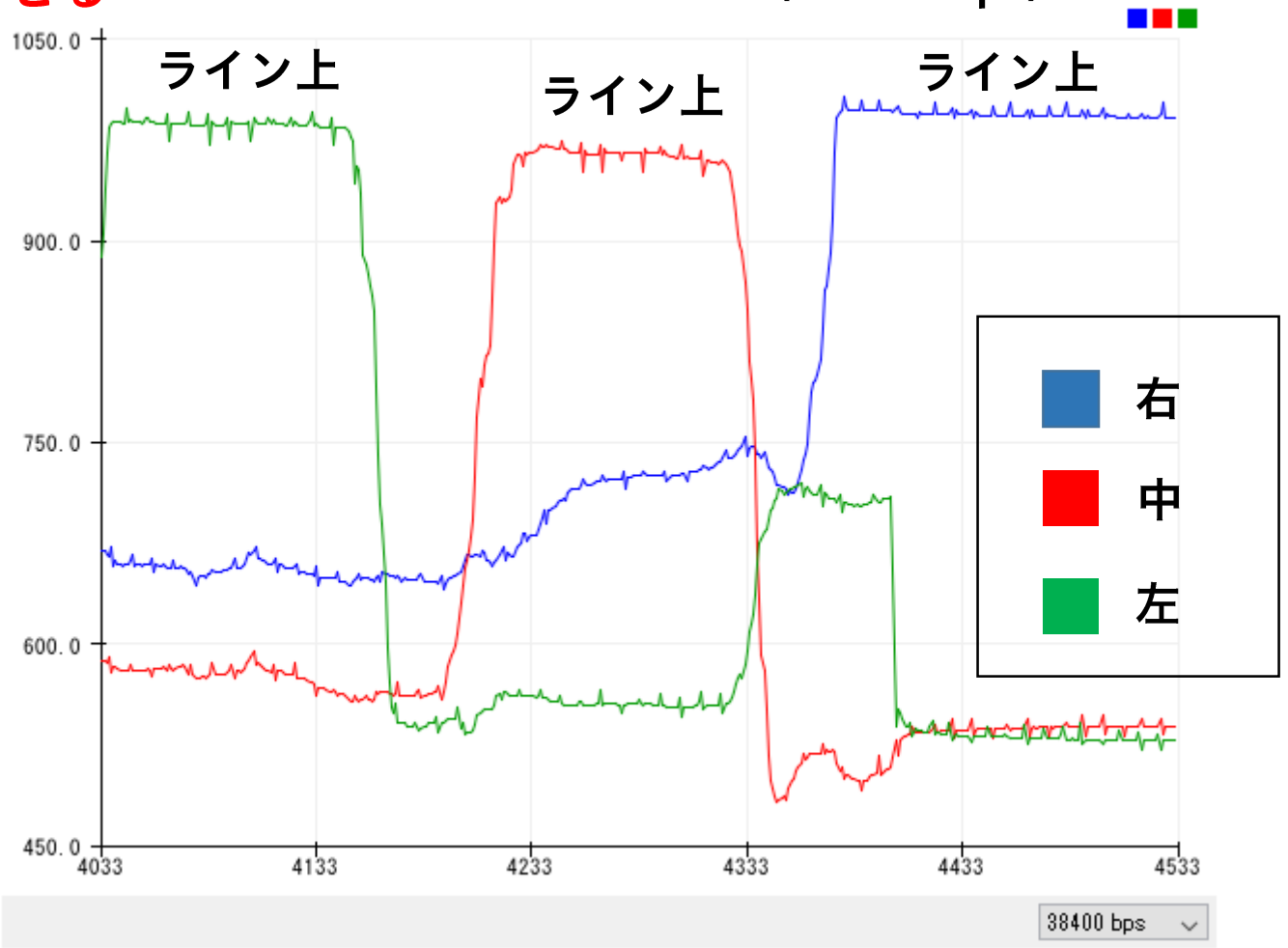
シリアルプロッタ（38400 bps）



Example0502: ラインの検出(アナログ)

- フォトリフレクタの動作確認
  - ライン上を左右にゆっくり移動させる

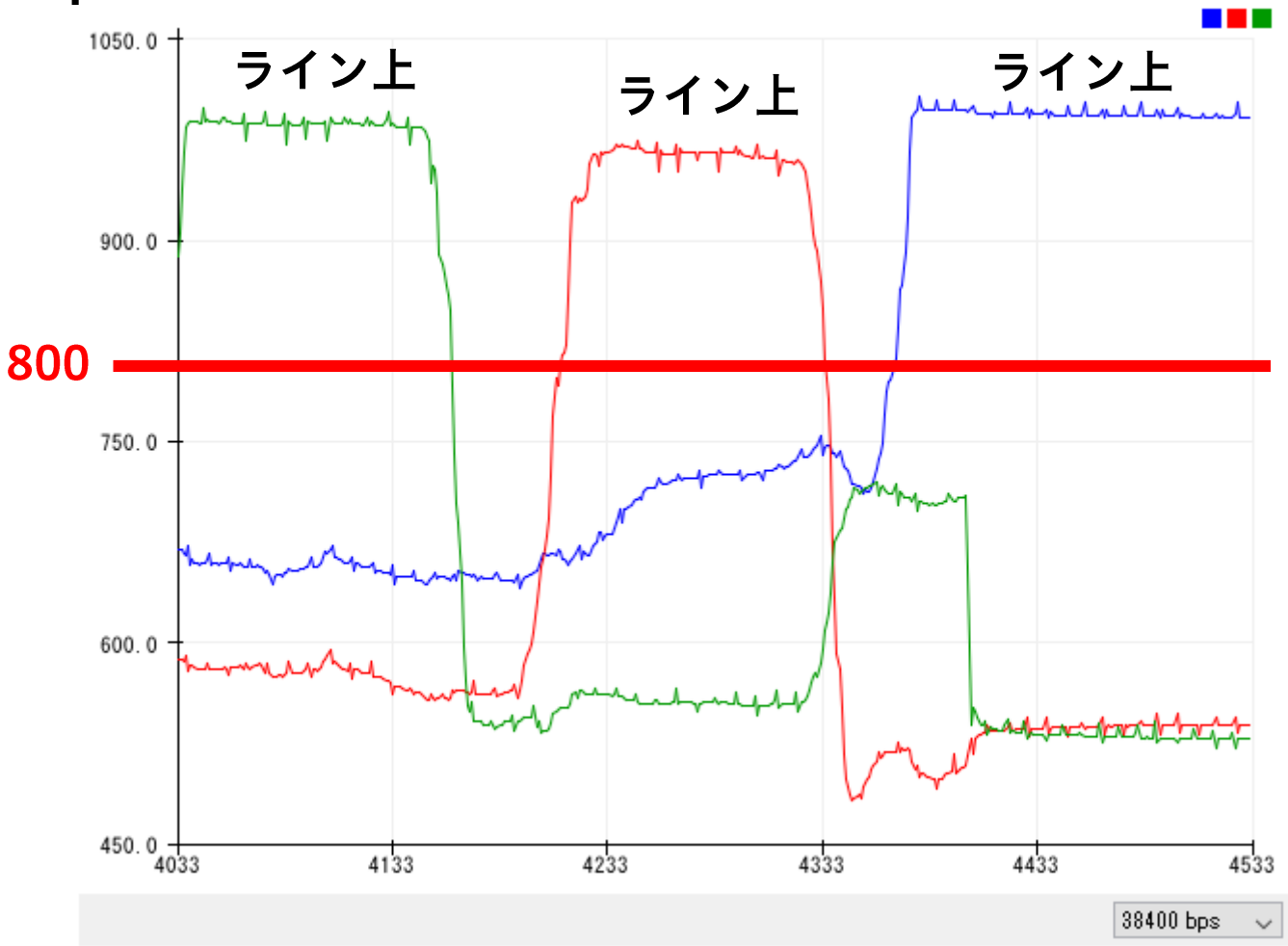
```
void setup() {
  Serial.begin(38400);
  delay(1000);
}
void loop() {
  int pr_r = analogRead(A0);
  int pr_c = analogRead(A1);
  int pr_l = analogRead(A2);
  Serial.print(pr_r); Serial.print(",");
  Serial.print(pr_c); Serial.print(",");
  Serial.println(pr_l);
  delay(50);
}
```



センサのライン検出

- センサがラインの上にあるとき1，それ以外は0
  - ある値(閾値)より大きければ → 1
  - それ以下 → 0
- 0か1の状態にする(2値化)

```
if (pr_l > 800) {
    pr_l = 1;
} else {
    pr_l = 0;
}
```

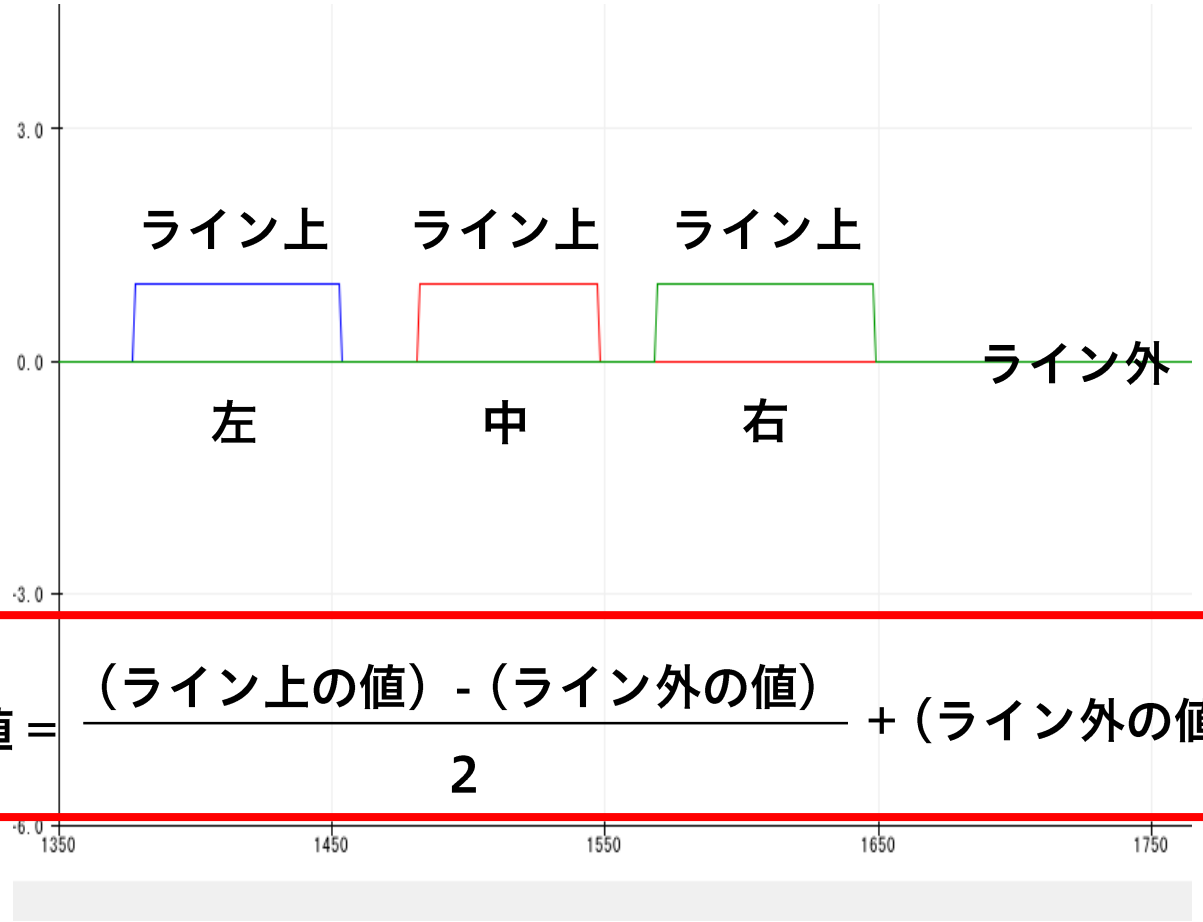


Example0503: ラインの検出(デジタル)

- センサがラインを検出すると1，それ以外は0

Example0501を書き込み，
ライン上，ライン外の値を
シリアルモニタで確認
→ 閾値を求める

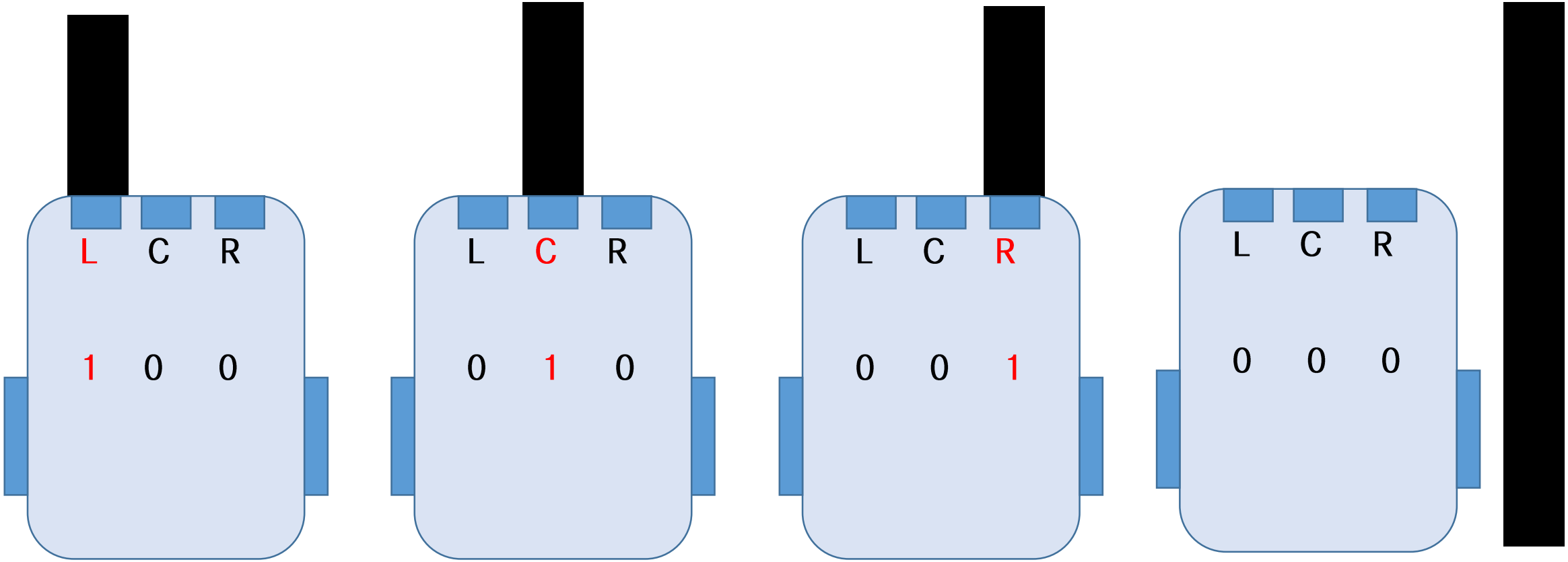
	ライン上	ライン外	閾値
右 (A0)			
中 (A1)			
左 (A2)			



$$閾値 = \frac{(ライン上の値) - (ライン外の値)}{2} + (ライン外の値)$$

ロボットのラインに対する位置

- ロボットがライン（直線）上の何処にいるか？



ロボットのラインレース

- ロボットをライン(黒)に沿って走らせる
- 障害物を避ける