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6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17日(金)

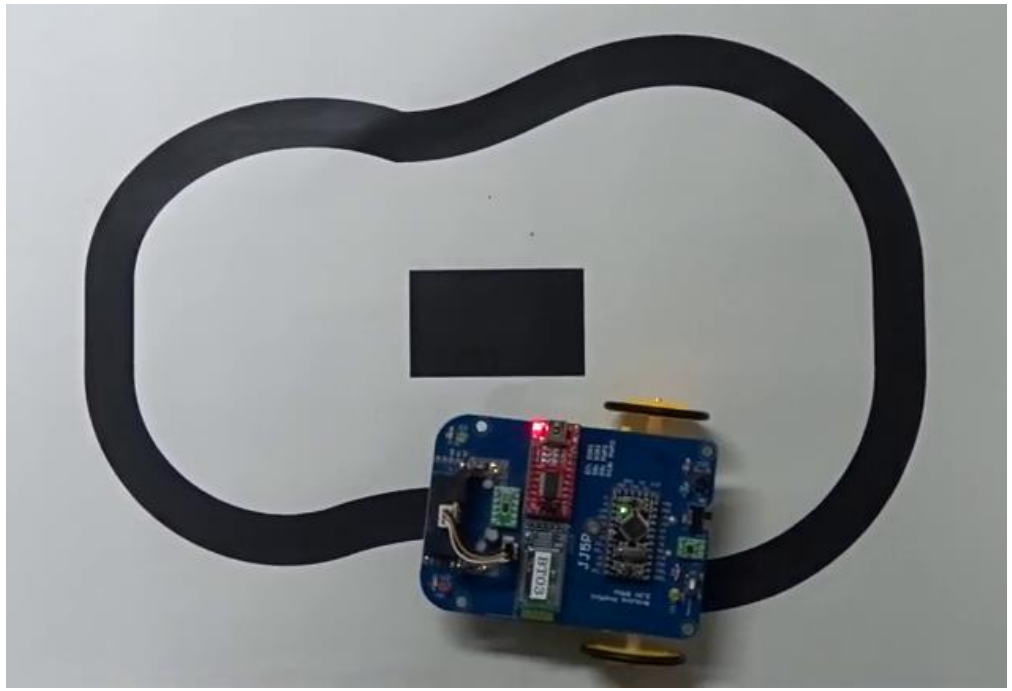
講座日程

- 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

- ロボットの制御
  - ロボットのプログラム
  - 決められた動作と状況に応じた動作
  - ラインを検出して回避
  - わかりやすい例(delayとmillis)
- ライントレースの準備
  - フォトリフレクタの校正
  - 手動校正
  - 自動校正

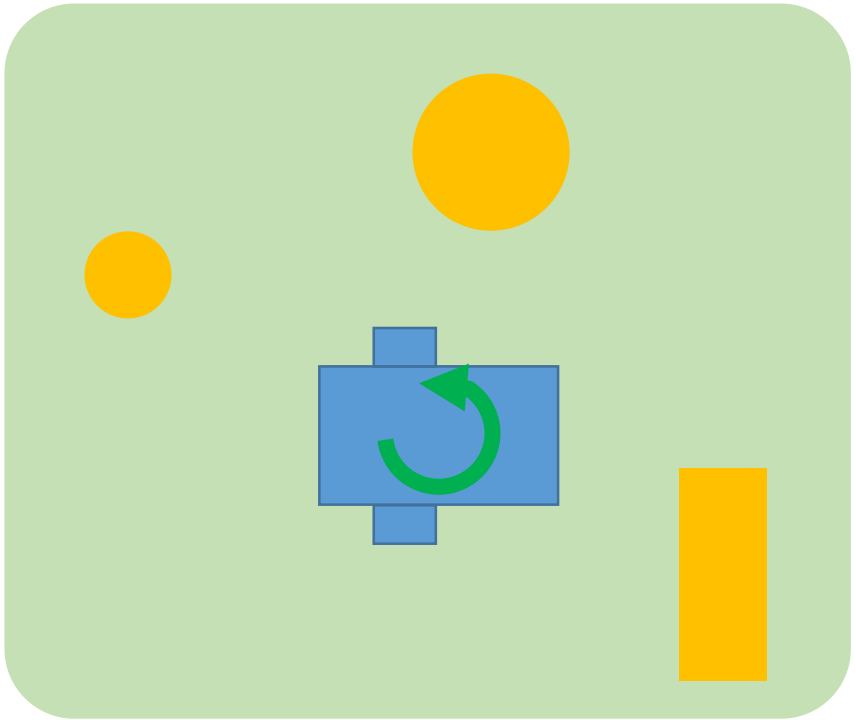
講座資料(スライド，サンプルスケッチ等)
https://goo.gl/K44cPc



ロボットの制御（認知，判断，操作）

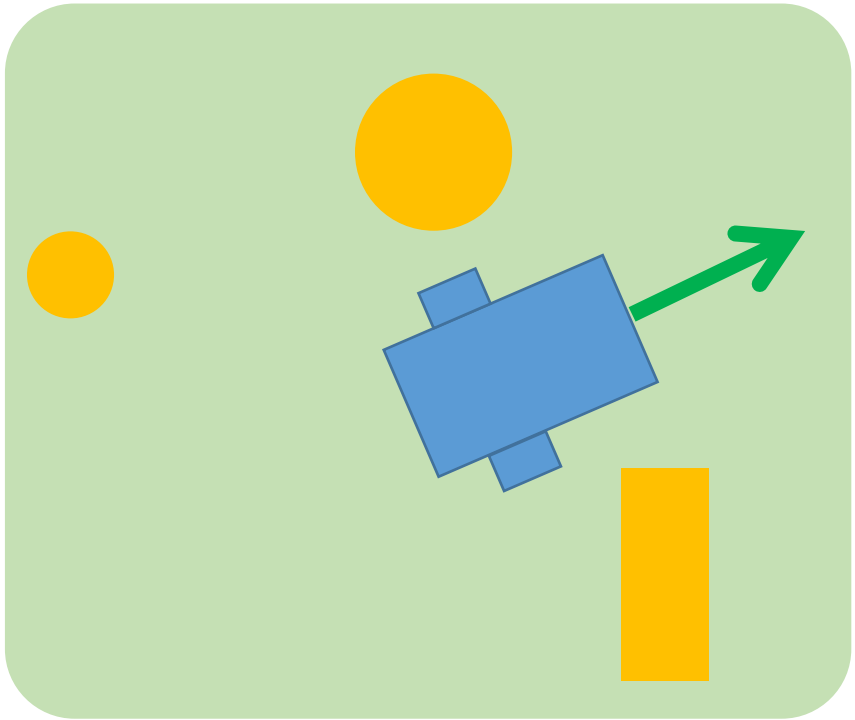
- ロボットの制御
  1. センサにより周囲の状況を認知
  2. プロセッサにより状況の判断
  3. 状況判断に応じたモーターの操作

1, 2, 3をスムーズに滞りなく繰り返す



前方に壁
障害物大と小が1つずつ

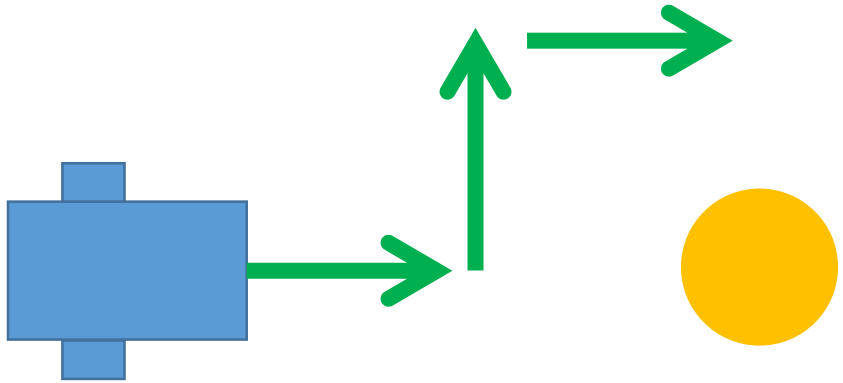
→ 壁と障害物の間を直進しよう



ロボットのプログラム

ロボット＝自分が置かれた状況を把握して行動する

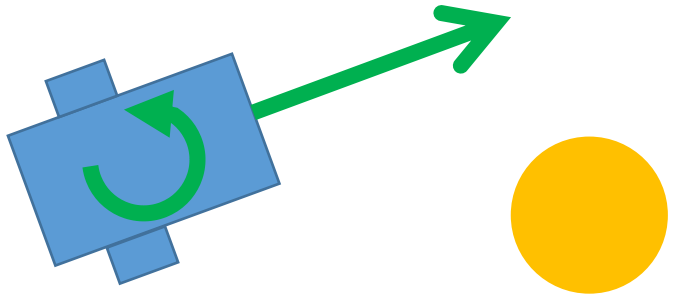
- 決められた動作（第3回）
  - fwd(100); rotL(90); fwd(100); rotR(90); fwd(100);
- 状況に応じた動作
  - センサで周囲の状況を感知して行動する



障害物→決められた動作で回避

あらかじめ周囲の状況がわかっている
15cm 先に障害物がある！

障害物発見！



障害物が見えなくなるまで旋回して直進！

周りの状況に合わせて行動する
センサを使い障害物を回避！

Example0601: ラインを検出して回避

- フォトリフレクタでラインを検出して回避行動

・センサで常時周囲の状況を感知
　フォトリフレクタを50ミリ秒毎に確認

```
if (millis() - pr_time > 50) {
   pr_sensor();
}
```

・センサで感知した状況に応じた動作
　フォトリフレクタ3つがラインに反応

```
if (pr_bin[0] == 1 &&
    pr_bin[1] == 1 &&
    pr_bin[2] == 1)
{
   bwd(2.0); // 少し後退
   rotL(180.0); // 左旋回
} else {
   fwd(1.0); // 直進
}
```

例：2種類のLED点滅プログラム（Example0602, Example0603）

- LEDの点滅
  - delayを使った例
  - millisを使った例

```
// Example0602
const int LED_Y_PIN = 13;

void setup() {
  pinMode(LED_Y_PIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_Y_PIN, HIGH);
  delay(1000);
  digitalWrite(LED_Y_PIN, LOW);
  delay(1000);
}
```

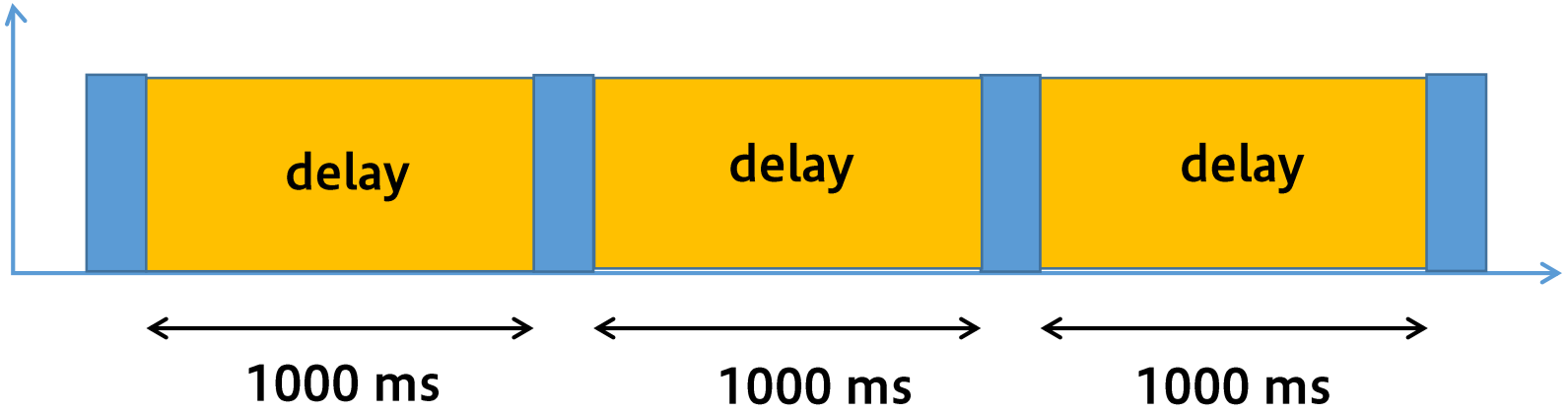
millis() マイコン起動後の経過時間
→ Example0603a参照

```
// Example0603
const int LED_Y_PIN = 13;
unsigned long led_y_time = 0;
bool led_y = HIGH;

void setup() {
  pinMode(LED_Y_PIN, OUTPUT);
}

void loop() {
  if (millis() - led_y_time > 1000) { // 1秒経過したら
    led_y = !led_y;  // HIGH,LOWを反転
    digitalWrite(LED_Y_PIN, led_y);
    led_y_time = millis(); // 時間を記憶
  }
}
```

2種類のプログラムの違い

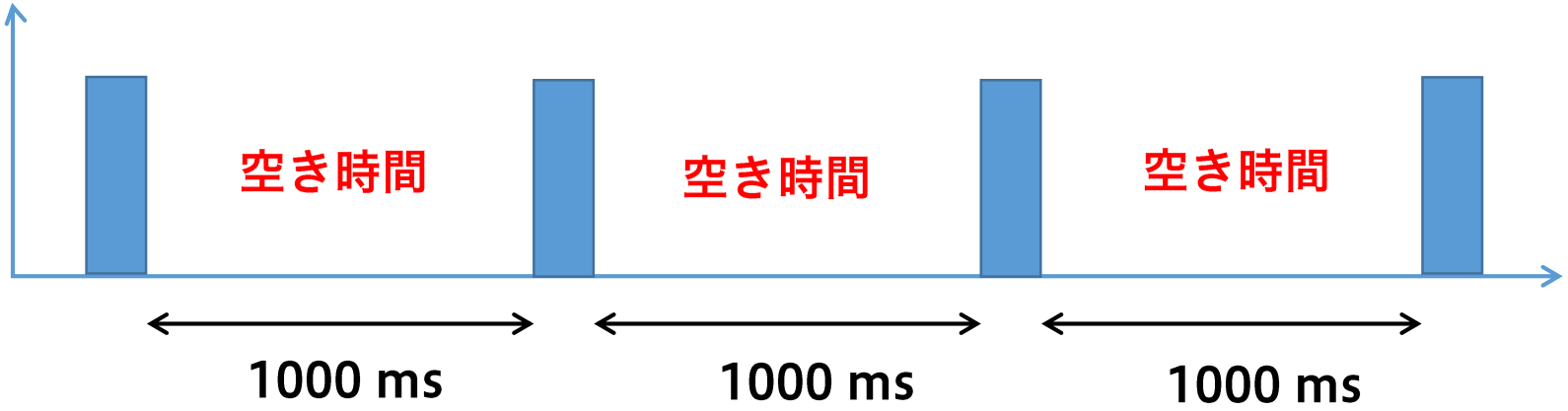
- delay・・・delay関数の実行中，他に何もできない

Example0602



- millis・・・経過時間判定の処理後は，空き時間（他の処理ができる）

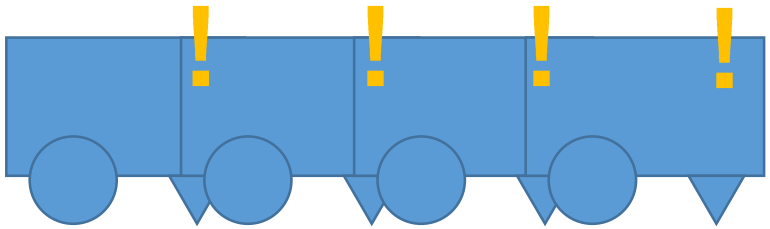
Example0603

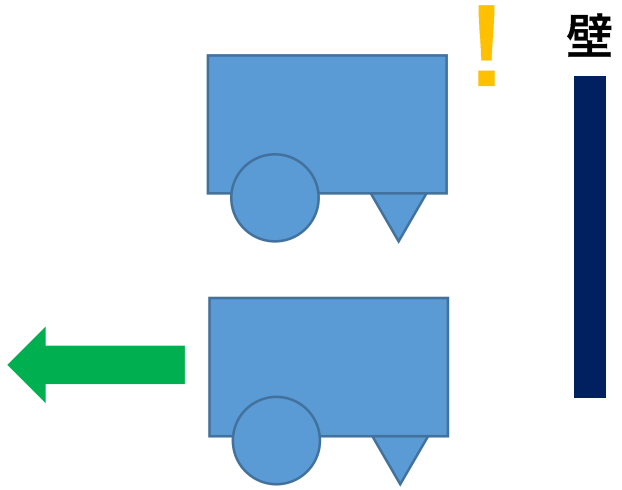


ロボットのプログラムではどうなる？

- ロボット：センサで周囲を確認しながらモーターを動かす
- センサで距離取得，モーターを一定時間delayで動かす

- 常時，センサで情報収集しながらモーターも動かす





Example0604：検出距離に応じてLEDの点滅周期を変える

- 距離センサで距離を計測
- 計測した距離をLEDの点滅周期

```
void loop() {
  if (millis() - dist_time > 50) { // 50ミリ秒毎に距離取得
    d = dist_sensor();
    Serial.println(100);
    dist_time = millis();
  }
  Serial.println(0);

  if (millis() - led_r_time > d) { // 距離に応じて点滅周期を変える
    led_r = !led_r; // HIGH->LOW, LOW-HIGH を繰り返し
    digitalWrite(LED_R_PIN, led_r);
    led_r_time = millis(); // 時間を記憶
  }
   ・・・・
}
```

距離：50ミリ秒毎に常に計測

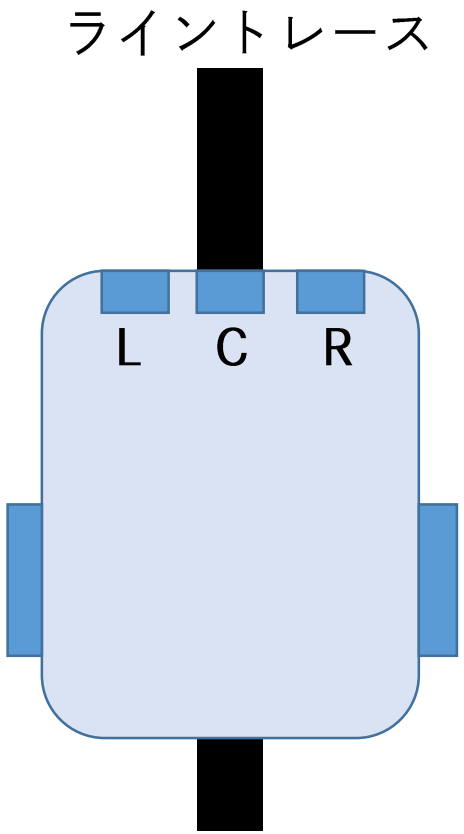
シリアルプロッタで確認

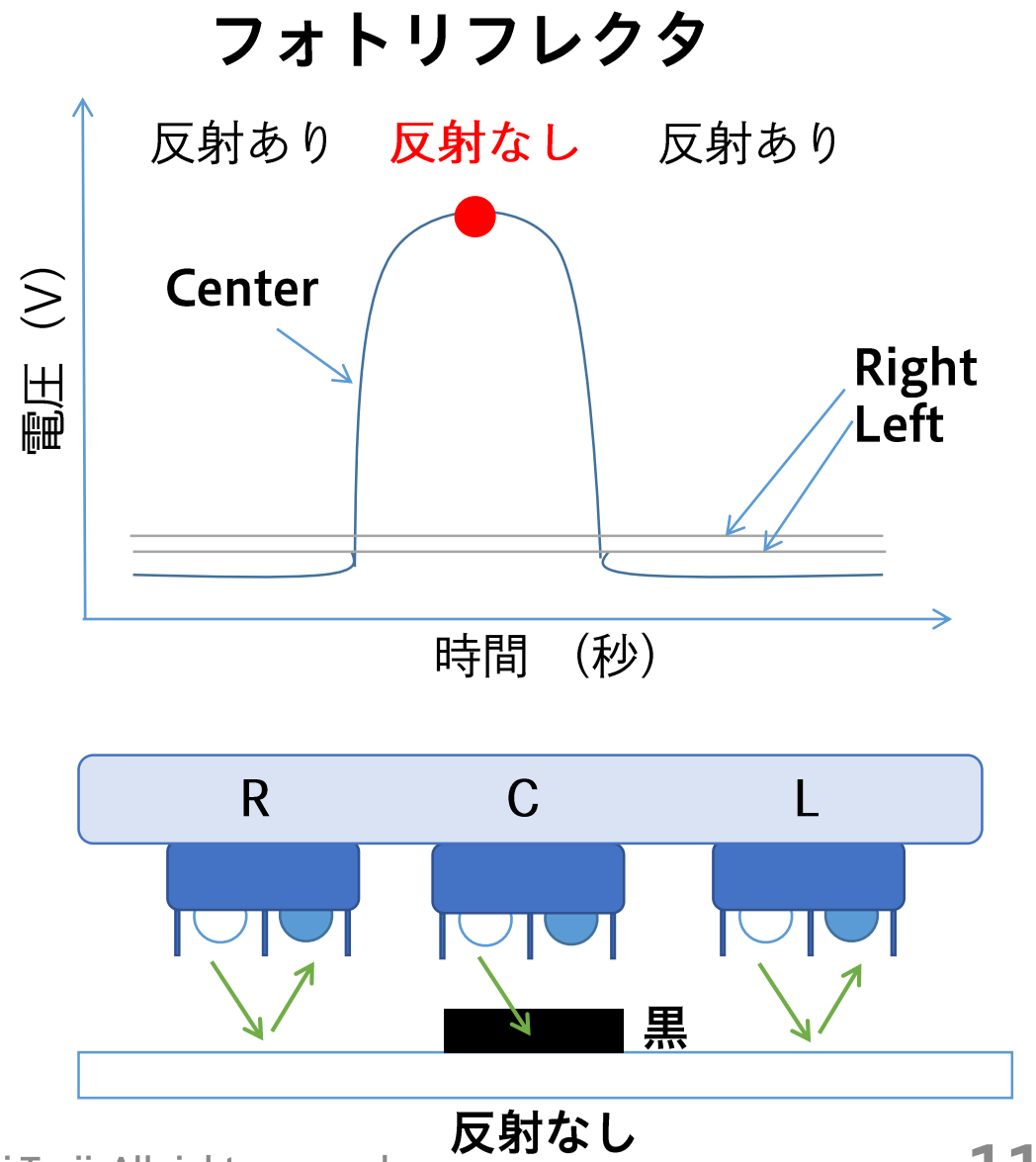
LED：測定した距離に応じて
点滅周期を変更

delayで書けるでしょうか？

ライントレースの準備

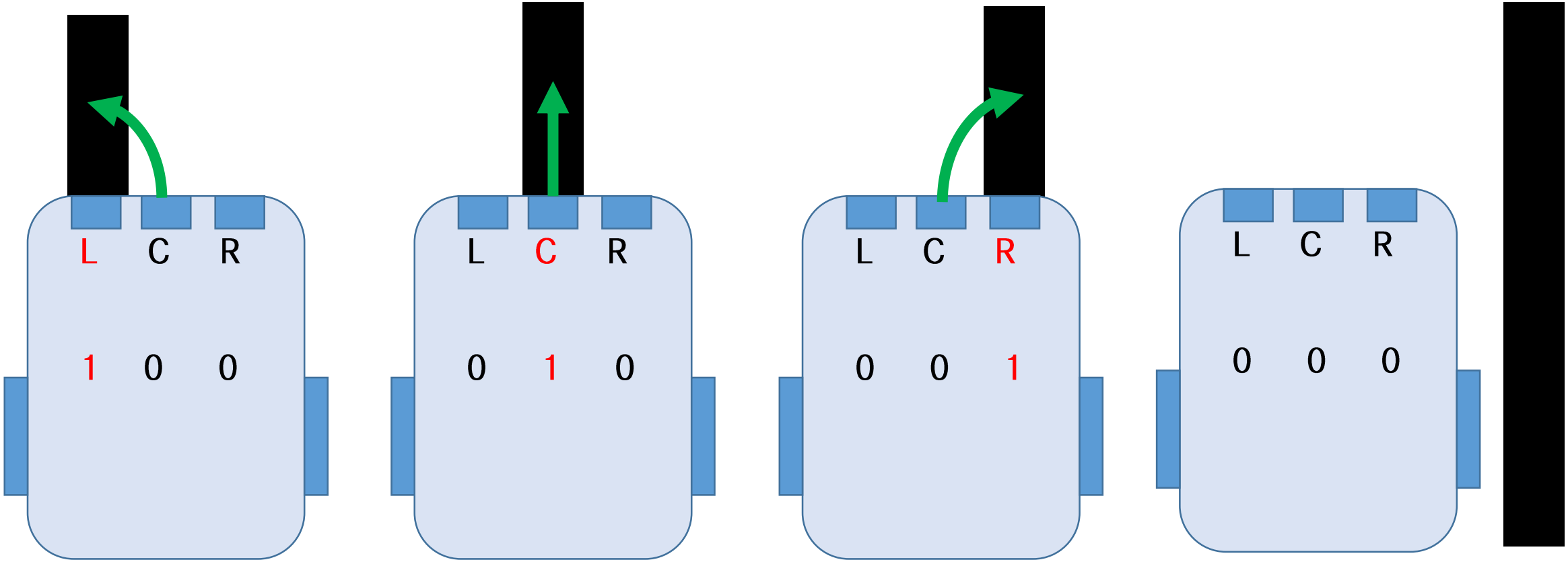
- ロボットのライントレース
  - ライン(黒線)の有無を検出して,
  - ラインの上を走る





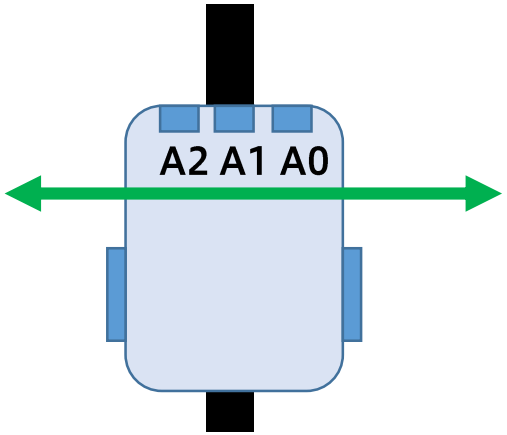
ロボットのラインに対する位置

- ロボットがライン上の何処にいるかによって行動を変更



Example0605: フォトリフレクタの手動校正

- フォトリフレクタの校正
  - 各センサの反応にばらつきがある
    - 前回は，各センサの最小値，最大値から閾値を計算して2値化(0, 1) → Example0503
- 手動で各センサの最小値，最大値を求める
  - ロボットをラインの上に置く
  - シリアルモニタを開く
  - ボタンを押して，ロボットをラインの左右に動かす（5秒間）



手でラインの上，ライン外に動かす

配列の利用・・・3個のセンサをまとめて処理

pr_pins[3]	A2	A1	A0
	pr_pins[0]	pr_pins[1]	pr_pins[2]
pr[3]	センサ値	センサ値	センサ値
	pr[0]	pr[1]	pr[2]

Example0606: フォトリフレクタの自動校正

- フォトリフレクタの自動校正
  - ロボットをラインの上に置く
  - シリアルモニタを開く
  - スイッチを押す
  - ロボットがライン上を動き，各センサの最小値，最大値が得られる
- 自動校正の手順

① 各センサの値xを取得

・各センサの最小値x_{min}，最大値x_{max}を求める
（ロボットをライン上で左右に振る）

② 各センサの最大値を1，最小値を0として正規化x_{nrm}

$$x_{nrm} = (x - x_{min}) / (x_{max} - x_{min})$$

③ x_{nrm}を0%～100%とし，

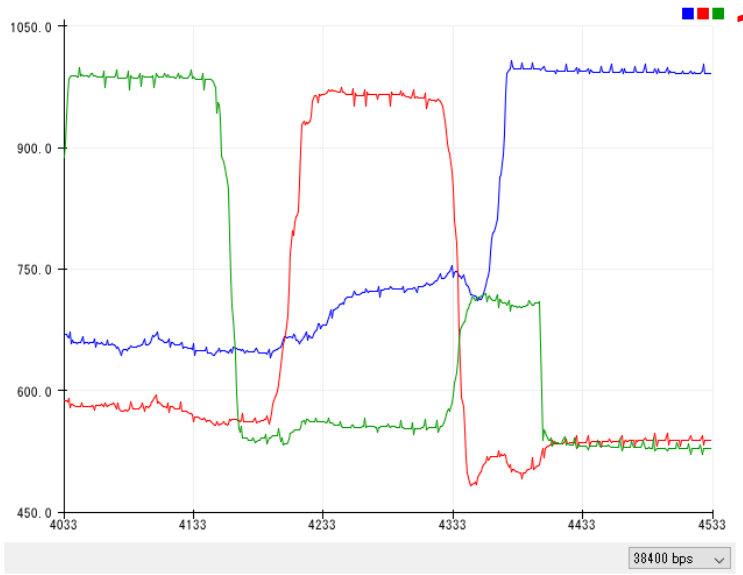
- 50%以上を1，50%未満を0として2値化x_{bin}

シリアルモニタの表示例：
```
int pr_min[PR_NUM] = {379,372,591};
int pr_max[PR_NUM] = {997,990,1008};
```

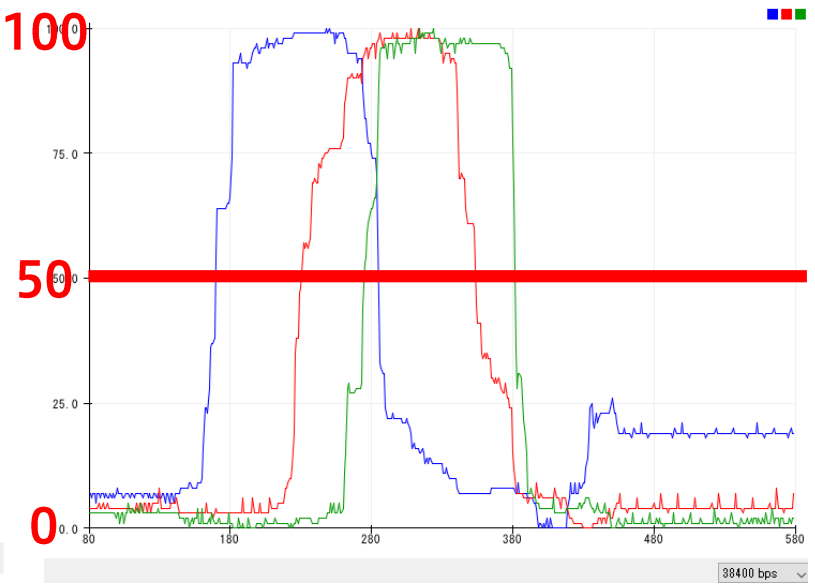
1. 一度，校正関数pr_calibを実行
2. 次回以降，この値を用いる。
3. pr_calibはコメントアウトする。

フォトリフレクタの自動校正(グラフ)

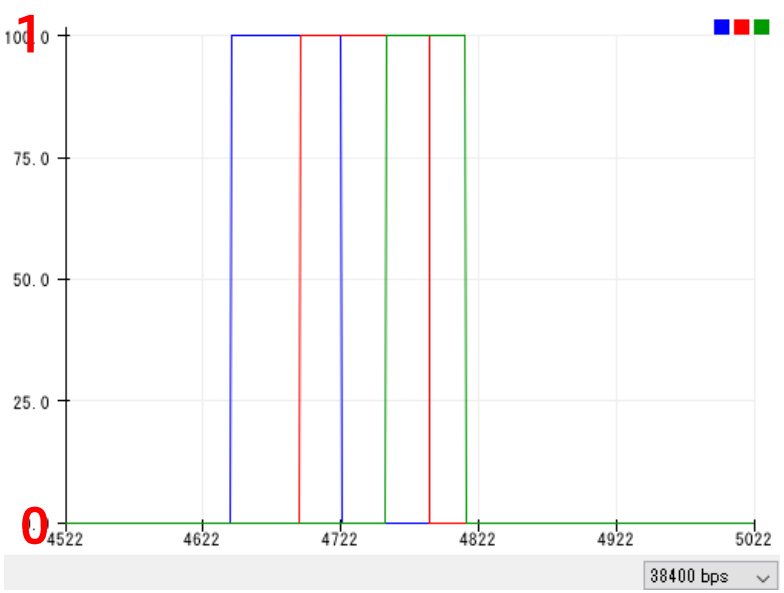
① pr_print(1)：センサの値(x)
② pr_print(2)：正規化後の値(xnrm)
③ pr_print(3)：2値化後の値(xbin)



① xのグラフ



② xnrmのグラフ



③ xbinのグラフ