

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7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](mailto:a-tsuji@is.tokushima-u.ac.jp)

# 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：

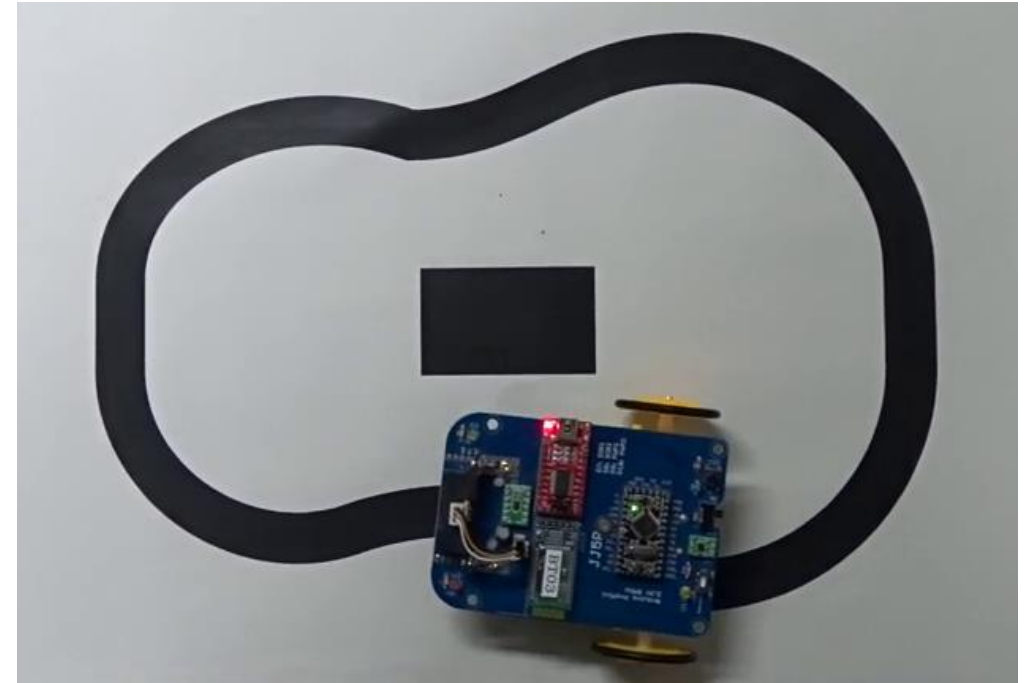
- ① 10/7 概要, ロボットの開発環境
- ② 10/14 ロボットのモーター1 (基本動作)
- ③ 10/21 ロボットのモーター2 (応用動作)
- ④ 10/28 ロボットのセンサー1 (距離センサ, 有線・無線通信)
- ⑤ 11/11 ロボットのセンサー2 (フォトリフレクタ)
- ⑥ 11/18 ロボットの制御1 (モータ・センサの協調動作)
- ⑦ 11/25 ロボットの制御2 (ライントレース)

# 本日の予定

- ▶ ロボット制御
- ▶ ロボット制御（距離センサ）
  - P制御
- ▶ ロボット制御（フォトリフレクタ）
  - センサのON/OFF
  - フォトリフレクタの位置
- ▶ ライントレース(P制御)
- ▶ ライントレース(PD制御)

講座資料(スライド, サンプルスケッチ等)

<https://goo.gl/K44cPc>



# ロボット制御

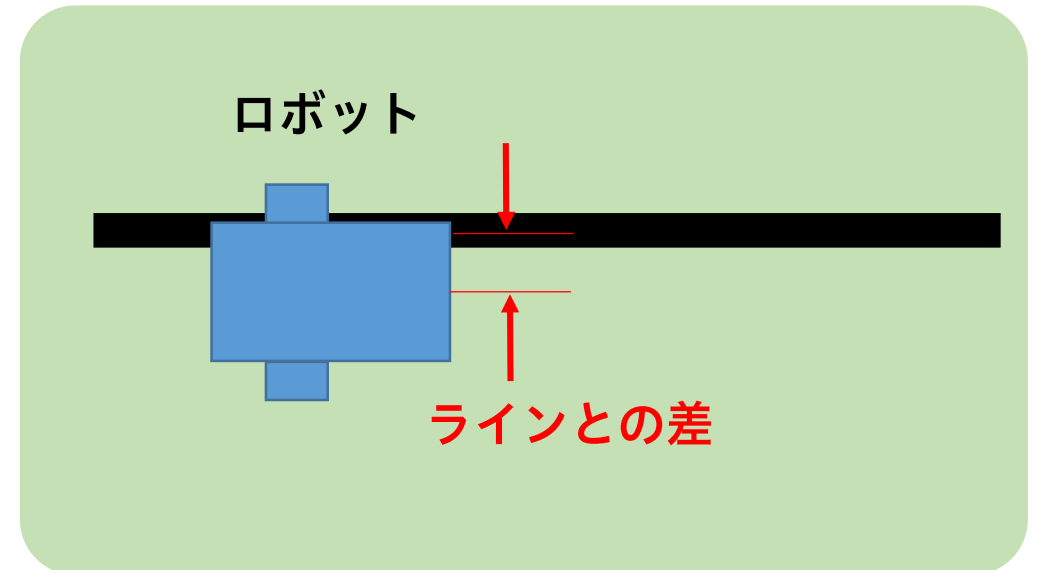
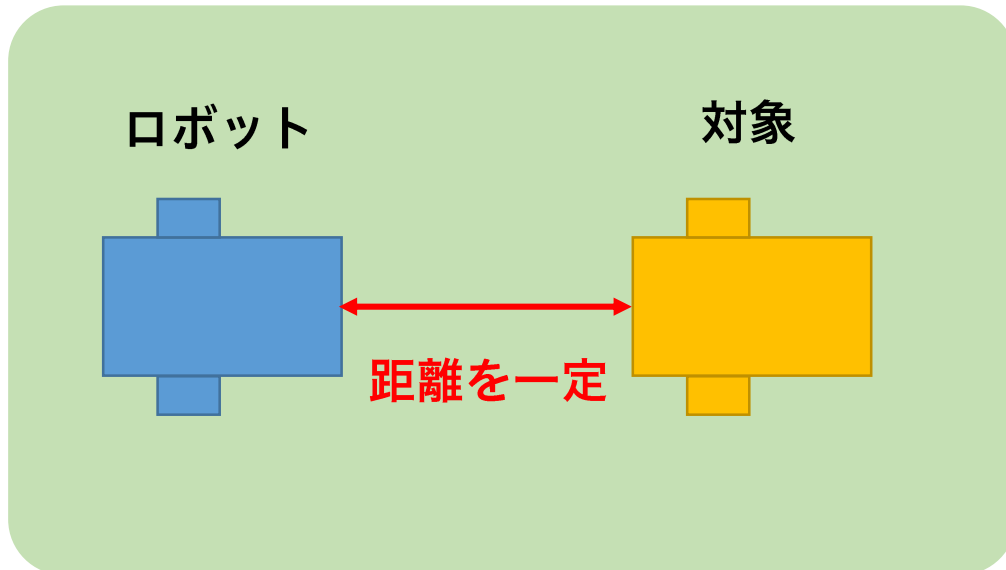
## ▶ ロボット制御

① 目標の設定

② 現在の状態を知る(センサ)

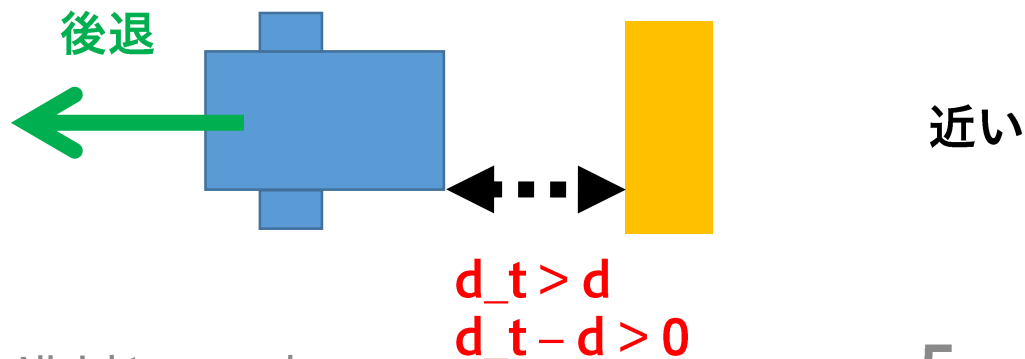
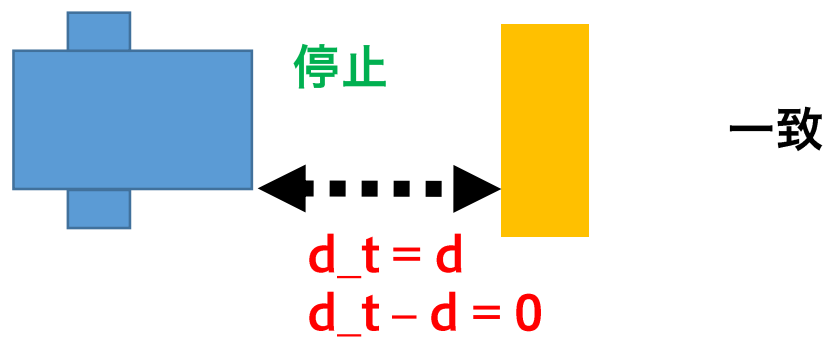
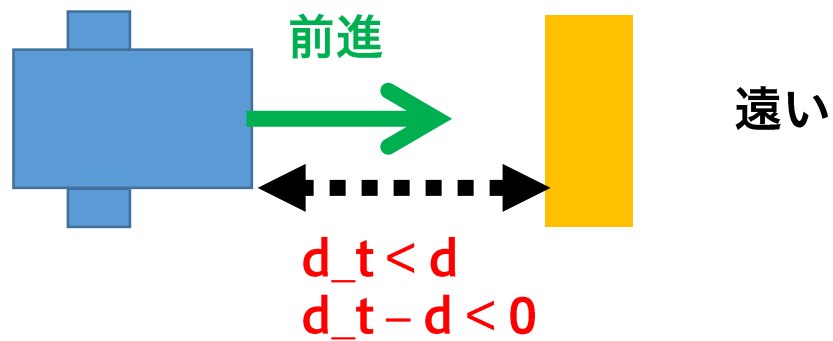
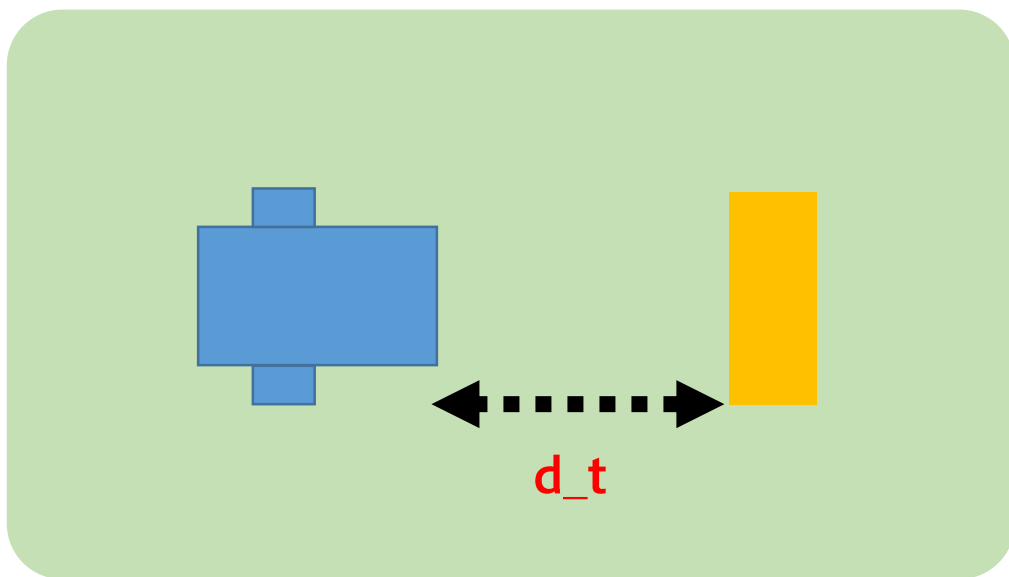
③ 目標と現在の状態との「差」を知る

④ 「差」があれば無くなるよう調整



# ロボットと対象との距離を一定

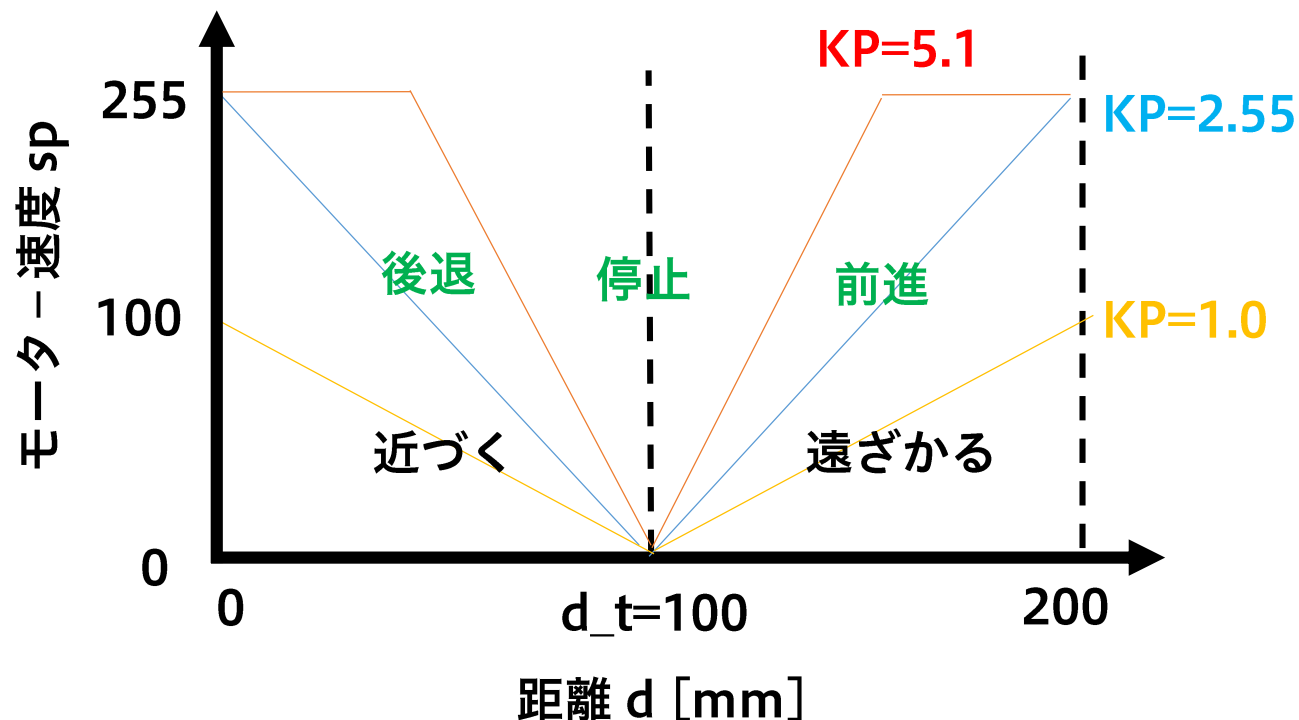
- ▶ ロボットと対象との距離を一定に保つ
  - ・ 目標の距離・・・ $d_t$  [mm]
  - ・ 計測距離・・・ $d$  [mm]
- ▶ 目標の距離と計測距離が0[mm]になるようにモーターを制御
  - ・  $d_t - d \rightarrow 0$  [mm]



# Example0701: ロボットのP制御(距離センサ)

- ▶ 目標距離 :  $d_t = 100$  [mm]
- ▶ 計測距離 :  $d$  (0[mm]~200[mm])
- ▶ モーターの速度 :  $sp$  (0~255)
- ▶ 目標距離 $d_t$ と計測距離 $d$ の差が小さくなるようモーター速度を変化

比例 (Proportional) 制御 :  
目標値と計測値との差に比例して操作量を制御



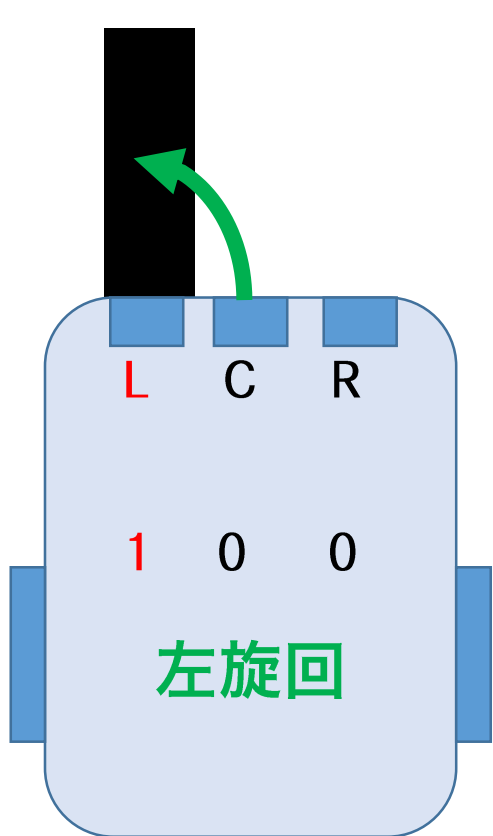
モーターの速度 :  $sp = KP * (d_t - d)$   
 $d_t$  : 目標距離  
 $d$  : 計測距離  
 $KP$  : 比例定数

ロボットの前進と後退 :  
 $sp \geq 0$  : `motor(sp, sp, FWD, FWD);`  
 $sp < 0$  : `motor(sp, sp, BWD, BWD);`

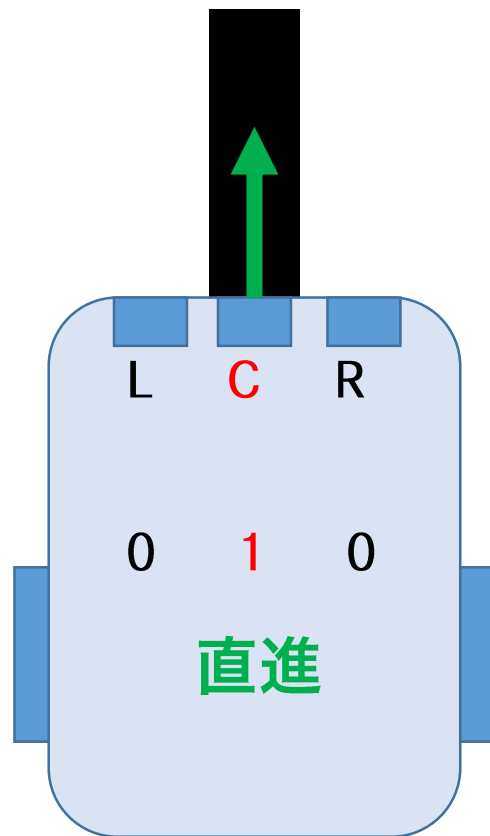
KPを変更してロボットの動作を確認

# ロボットのラインに対する位置から動作を決定

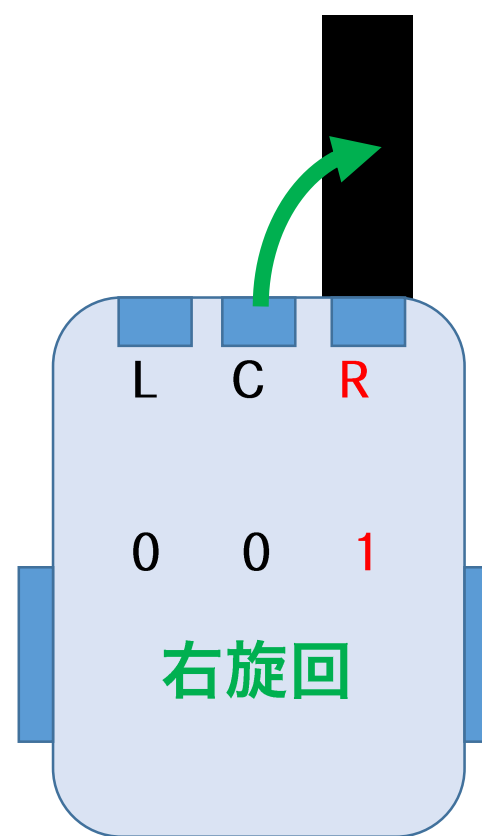
- ▶ ロボットの現在位置をフォトリフレクタで感知



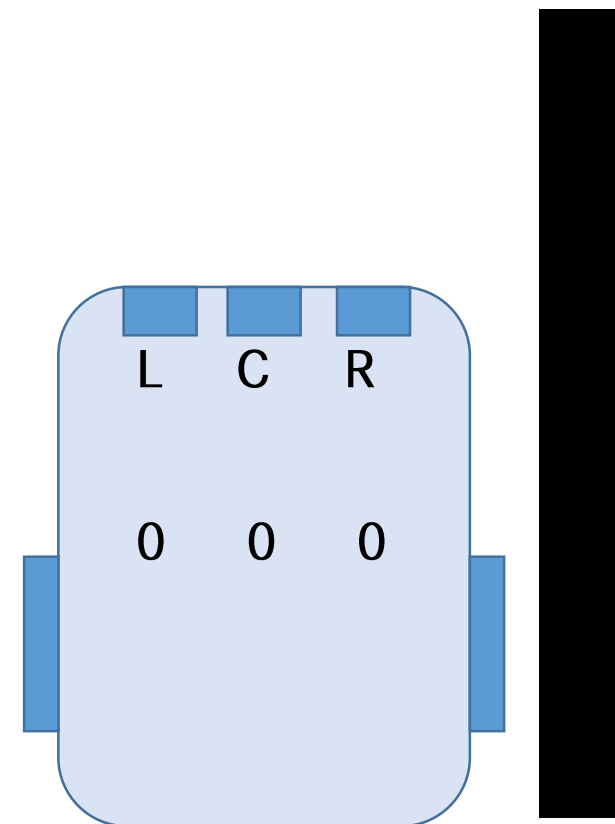
If (pr\_bin[0] == 1 &&  
pr\_bin[1] == 0 &&  
pr\_bin[2] == 0 )



If (pr\_bin[0] == 0 &&  
pr\_bin[1] == 1 &&  
pr\_bin[2] == 0 )



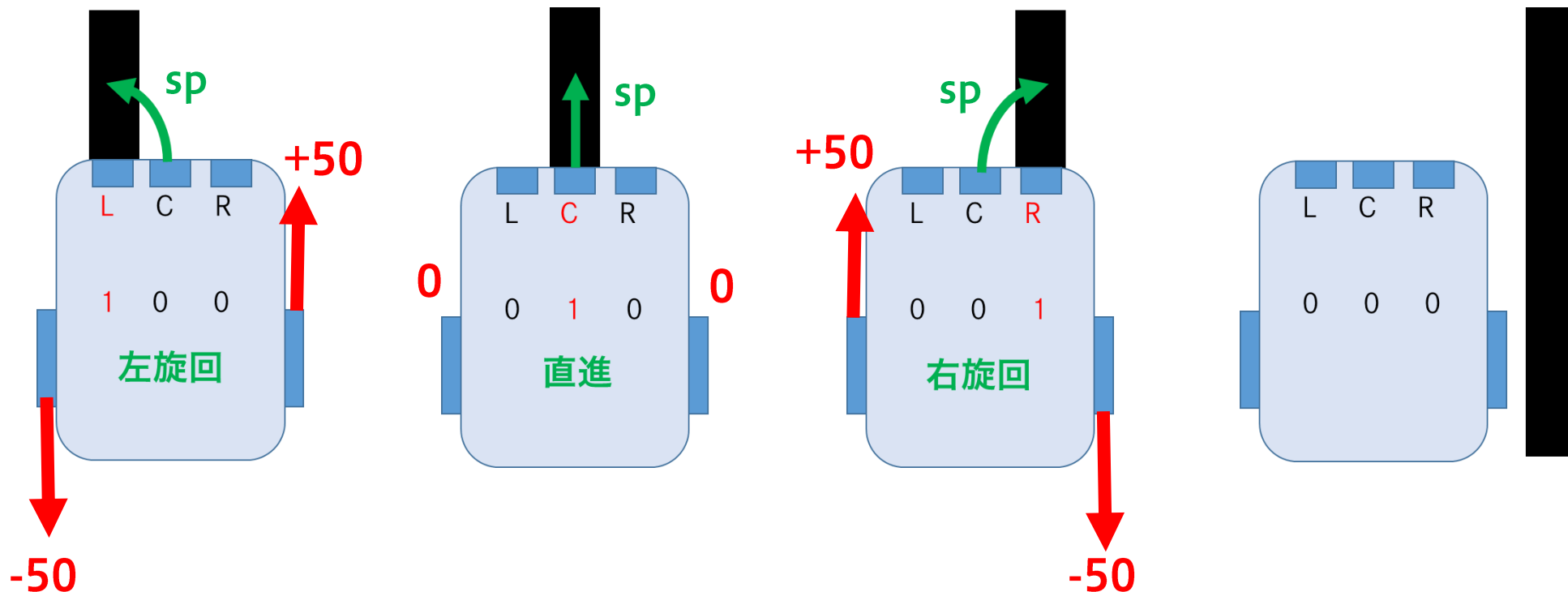
If (pr\_bin[0] == 0 &&  
pr\_bin[1] == 0 &&  
pr\_bin[2] == 1 )



If (pr\_bin[0] == 0 &&  
pr\_bin[1] == 0 &&  
pr\_bin[2] == 0 )

## Example0702 : ライントレース(センサのON/OFF条件)

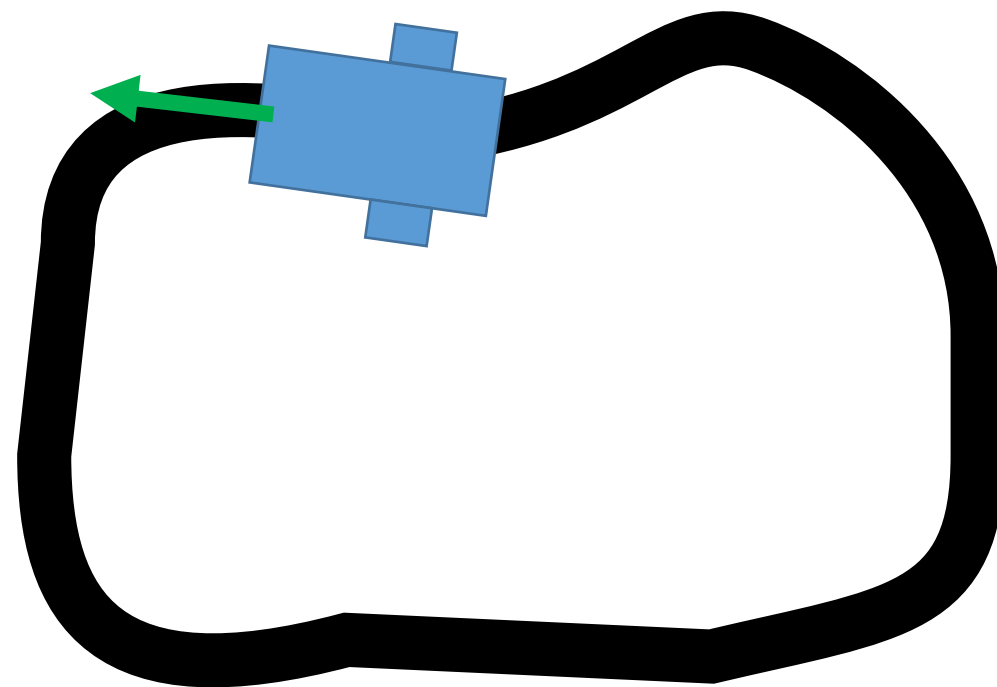
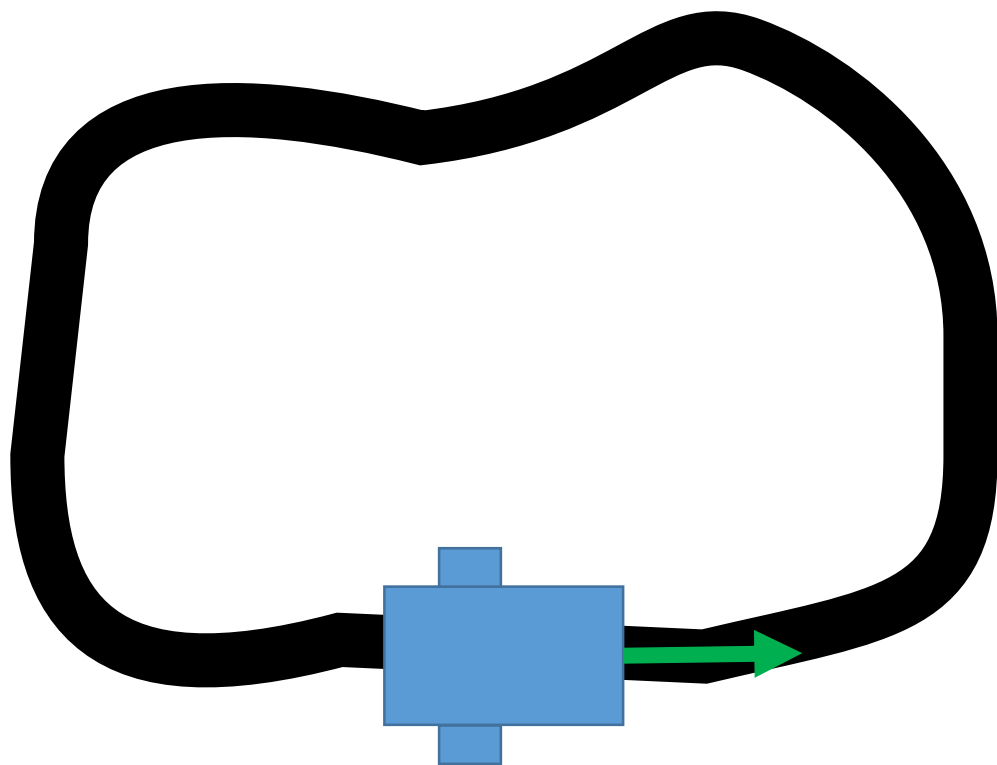
- ▶ モーターの基本速度  $sp$  : 100
- ▶ フォトリフレクタの反応に応じてモーターを直進, 左右旋回( $sp+sp\_diff$ )
- ▶ 例> 左右旋回量  $sp\_diff = 50$



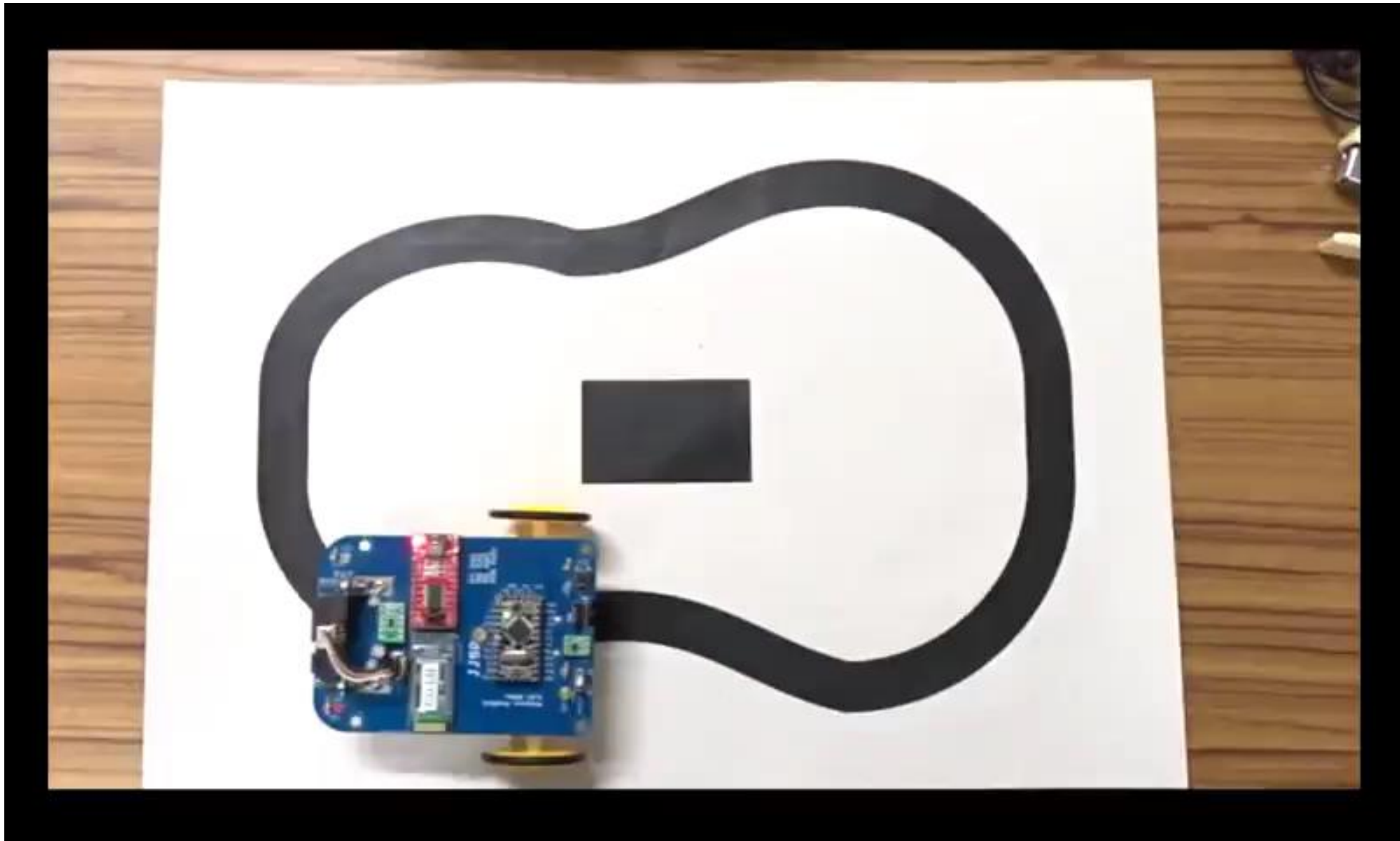


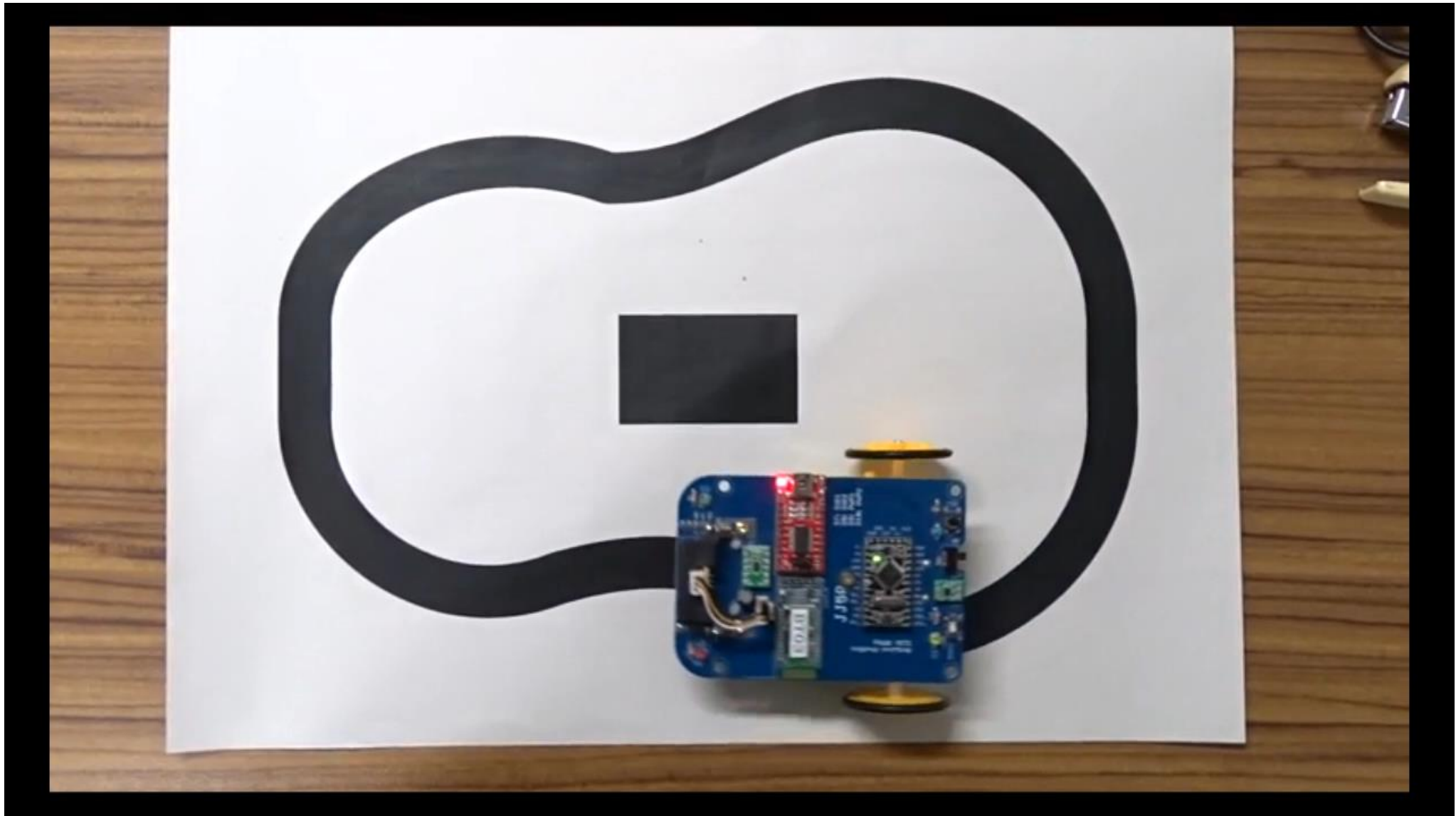
# ライトレースコースの巡回

- ▶ モーターの基本速度(sp), 旋回量(sp\_diff)を変更
- ▶ コースを左回り, 右回りに脱線せずスムーズに走行できるか確認



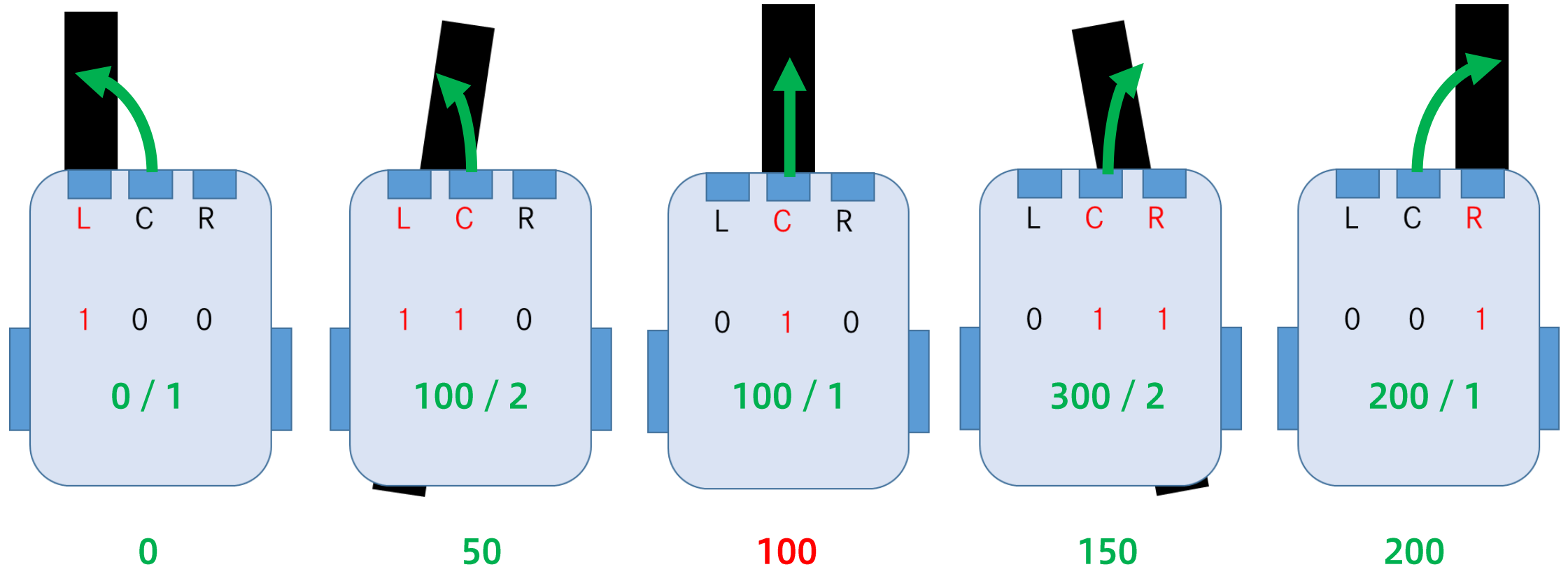
# ON/OFF制御





## Example0703 : フォトリフレクタの位置

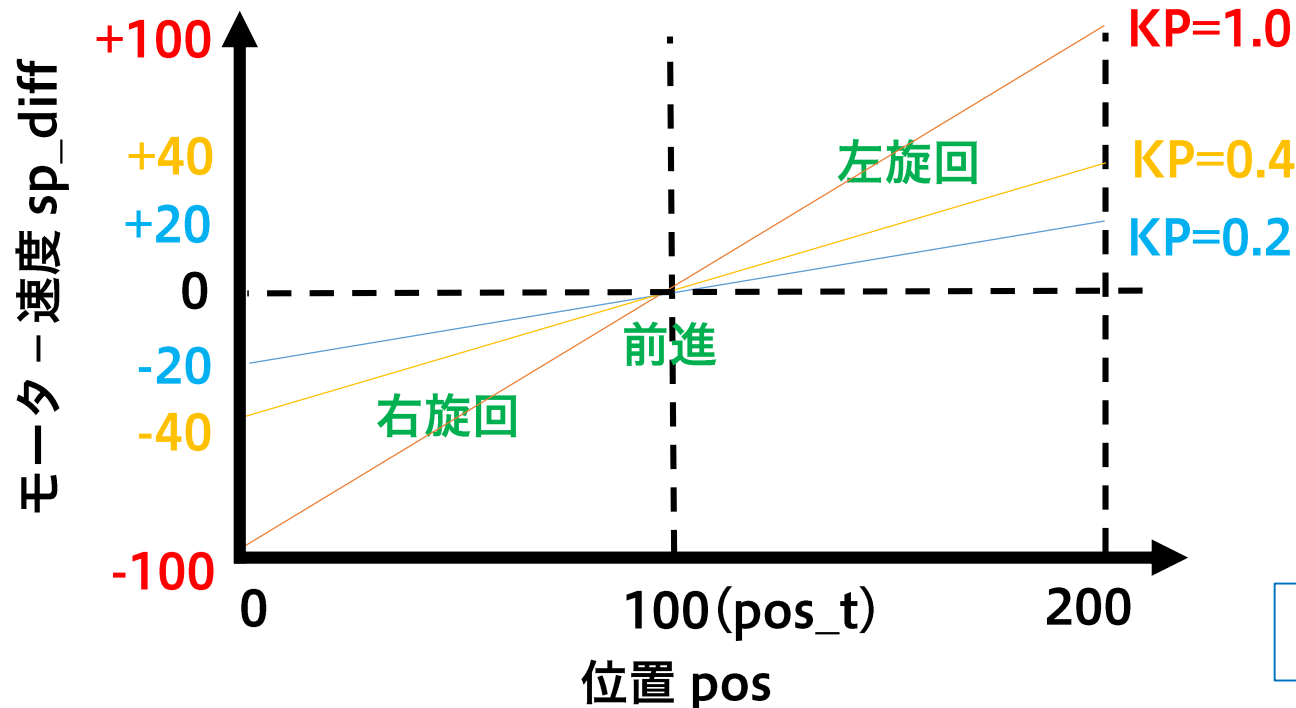
- ▶ フォトリフレクタの位置を数値0～200で表す（センターが100）
  - センサL \* 0 + センサC \* 100 + センサR \* 200  
反応したセンサの数



# Example0704 : ライントレース(P制御)

- ▶ 目標位置 :  $pos\_t = 100$
- ▶ 計測位置 :  $pos$  (0~200)
- ▶ モーターの速度 :  $sp\_l, sp\_r$  (0~255)
- ▶ 目標位置 $pos\_t$ と計測位置 $pos$ の差が小さくなるようモーター速度を変化

比例 (Proportional) 制御 :  
目標値と計測値との差に比例して操作量を制御



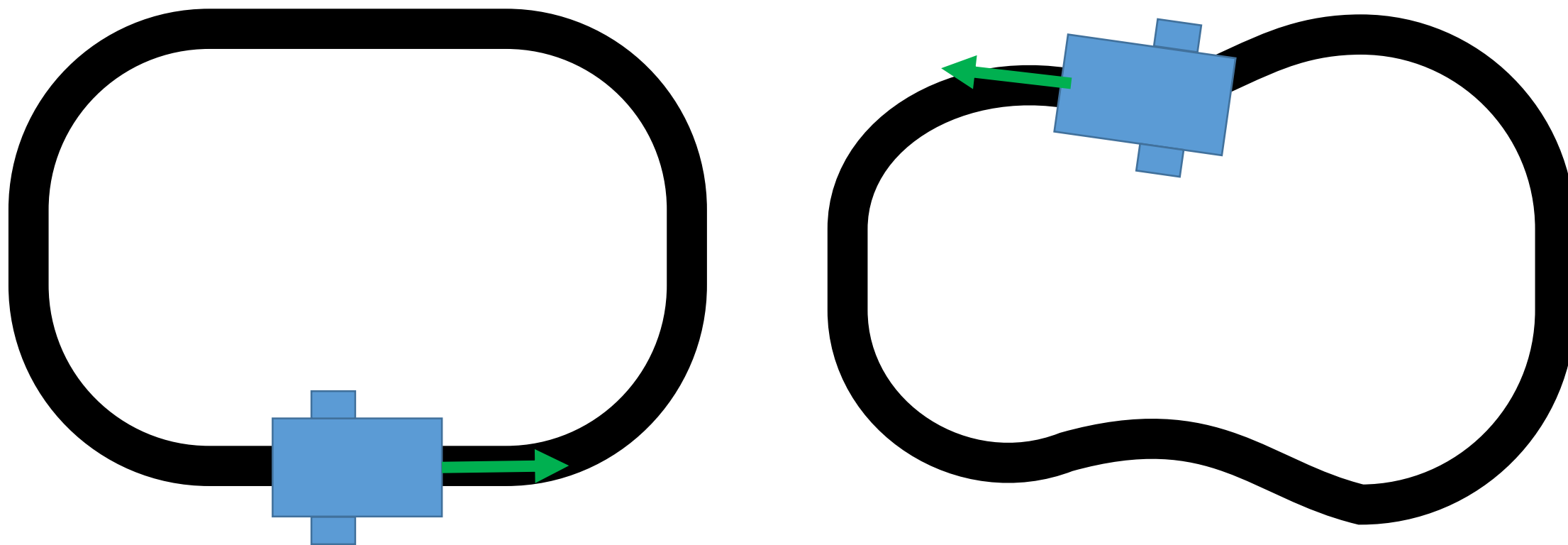
目標との差 :  $sp\_diff = KP * (pos\_t - pos)$   
 $pos\_t$  : 目標位置  
 $pos$  : 計測位置  
 $KP$  : 比例定数

ロボットの前進と後退 :  
左モーター :  $sp\_l = sp - sp\_diff$   
右モーター :  $sp\_r = sp + sp\_diff$   
 $sp$  : モーターの基本速度

KPを変更してロボットの動作を確認

# ライトレースコースの巡回

- ▶ モーターの基本速度(sp), 比例定数(KP)を変更
- ▶ コースを左回り, 右回りに脱線せずスムーズに走行できるか確認

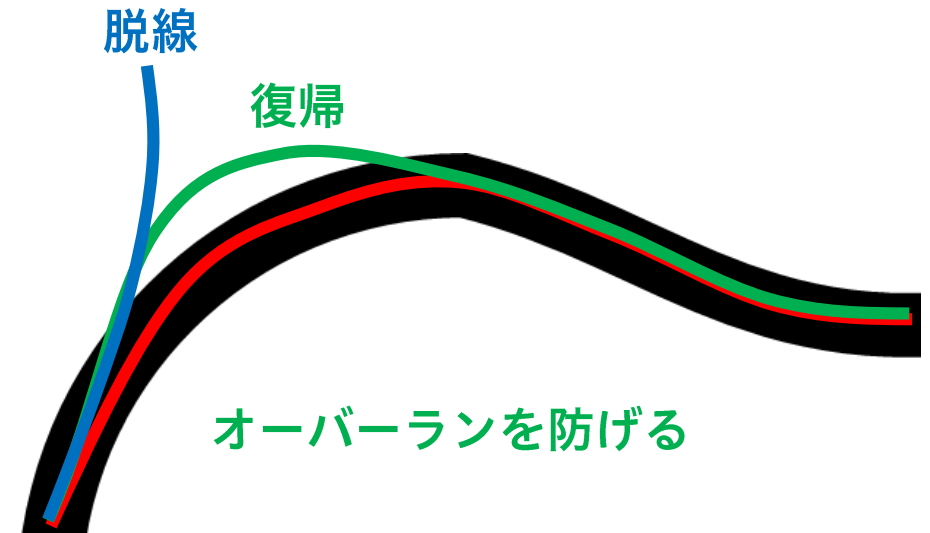
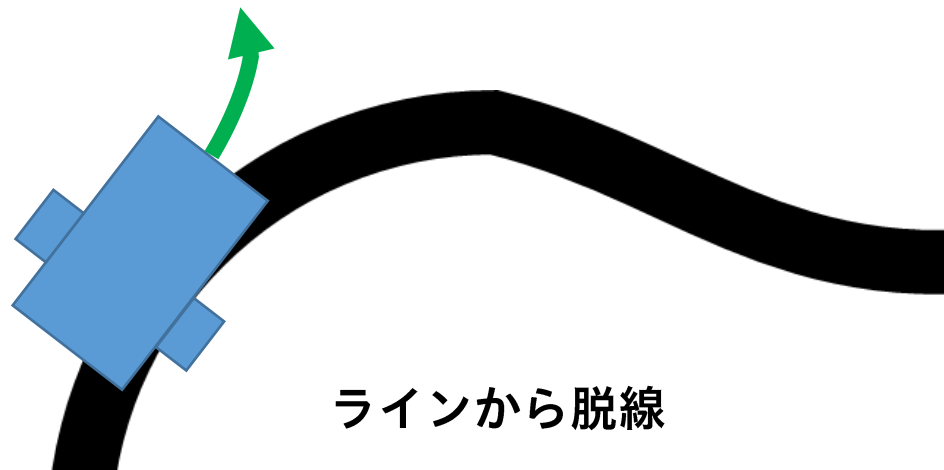


# Example0705 : ライントレース(PD制御)

- ▶ 急なカーブでラインから脱線
- ▶ 目標との差 :  $\text{error} = \text{pos}_t - \text{pos}$
- ▶  $\text{sp\_diff} = \text{KP} * \text{error} + \text{KD} * (\text{error} - \text{last\_error})$
- ▶ 直前の差 :  $\text{last\_error}$  を使用

微分 (Differential) 制御 :

目標値との差の今回と前回の値の変化の大きさに  
比例した操作量で制御



# ライントレースの調整方法

ロボットがライン上を横振れ・脱線せず，早く走行できるよう調整

- ▶ モーターの基本速度(SP)を一定の値に設定
- ▶ KPの値のみを0から徐々に増加させて調整（KDの値は0）
  - ・ ロボットを試走させ，直線・カーブの様子を見る
- ▶ KDの値を調整
  - ・ 急なカーブで脱線するときに追加
  - ・ KDの値が大きすぎると車体が横振れする
- ・ 制御がうまくいかないときは
  - ※ 電池が消耗していないか確認
  - ※ フォトリフレクタの校正が正しく行われているか確認
  - ※ モーター左右の値をシリアルプロッタで表示して確認



# 本講座の情報

## 講座資料のホームページ：

- <http://web.is.tokushima-u.ac.jp/wp/>
- 随時，更新中です。

## ▶ 講座配布資料：

- <https://goo.gl/K44cPc>