2014年6月28日:草稿
2014年11月5日:初稿

ロボットをつくろう(後編)
ー ロボットの走行制御 2 ー

2014年11月15日(土) 10:00—11:30

徳島大学大学院ソシオテクノサイエンス研究部
技術専門職員　　辻　　明典

連絡先：
770-8506 徳島市南常三島町2-1
TEL/FAX： 088-656-7485
E-mail:：a-tsuji@is.tokushima-u.ac.jp

講座の概要

講座名：ロボットをつくろう－後編－

講師：川上博（徳島大学名誉教授）
　　辻明典（徳島大学ソシオテクノサイエンス研究部
　　総合技術センター）

曜日・時間：土曜日　10時00分～11時30分

スケジュール：
① 10／4　2輪移動ロボットの組み立て
② 10／11 ロボットのセンサの機能試験
③ 10／18 ロボットの走行実験１（キャリブレーション）
④ 10／25 ロボットの走行実験２（赤外線リモコン）
⑤ 11／8　ロボットの走行制御１（前進，後退，左折，右折，停止）
⑥ 11／15 ロボットの走行制御２

本日の予定

1　ロボットの制御

2　ライントレース

3　ライントレース応用

4　次期講座内容について

付録

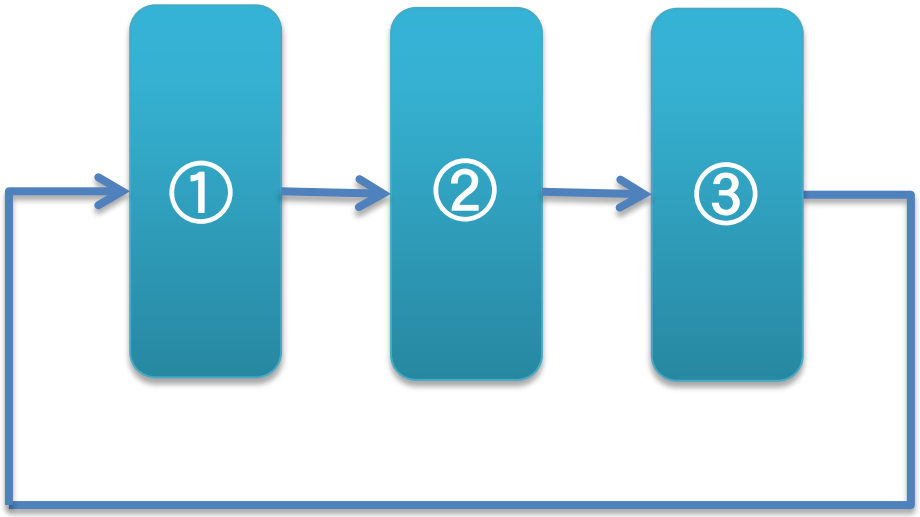
1 ロボットの制御

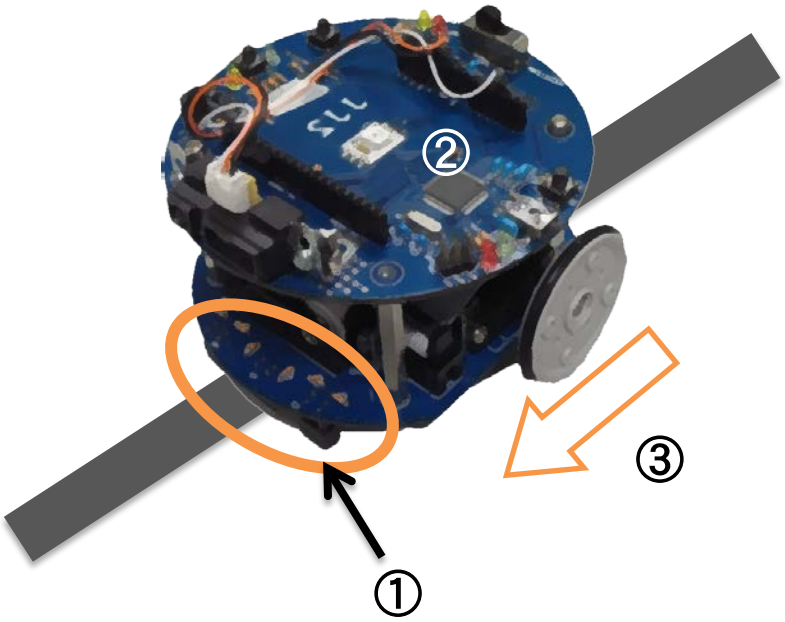
ロボットの制御

① センサにより周囲の状況を理解（認知）

② ロボットの行動を決定（判断）

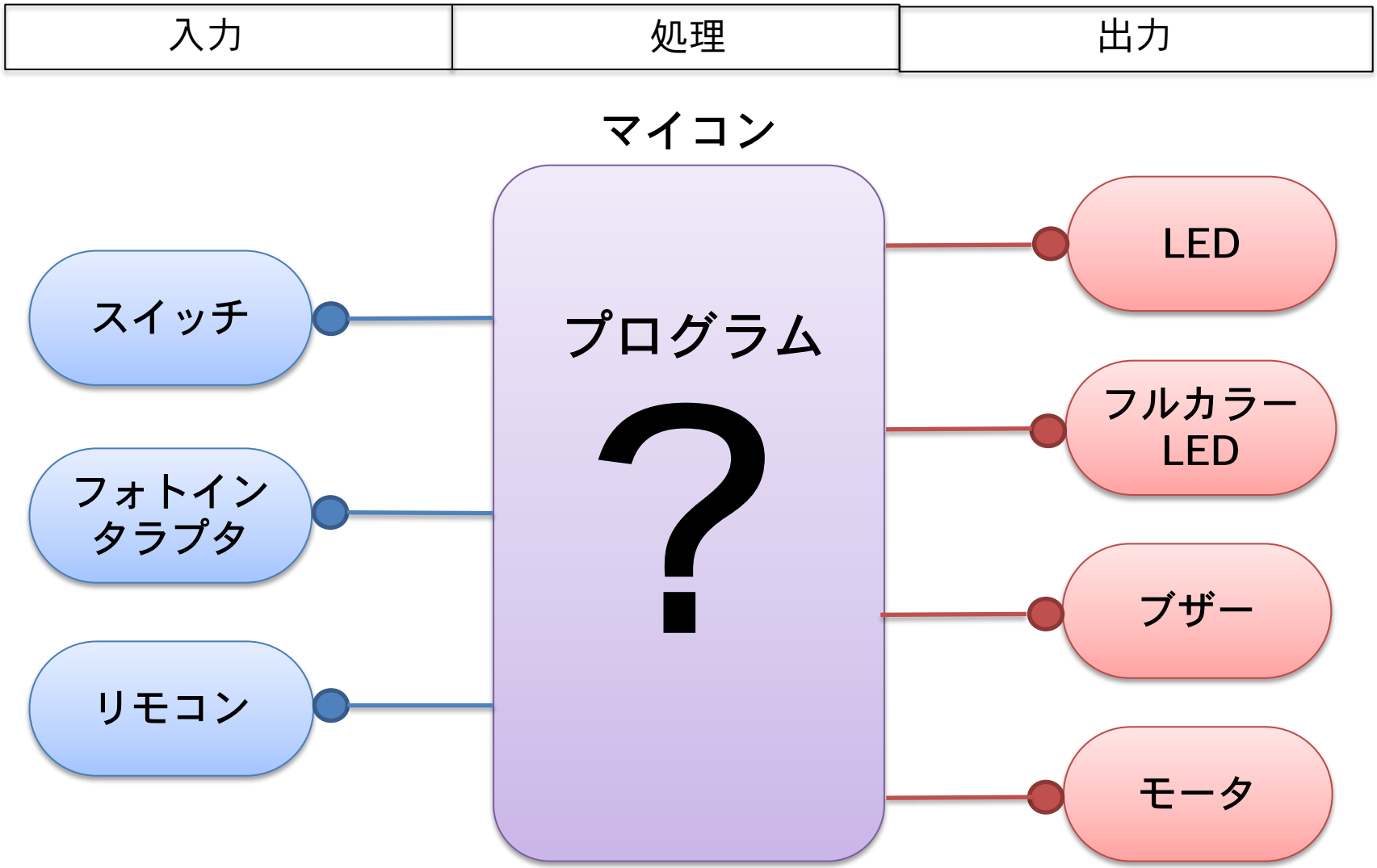
③ ロボットの動作（操作）





（例）ライントレース

制御プログラムを考える

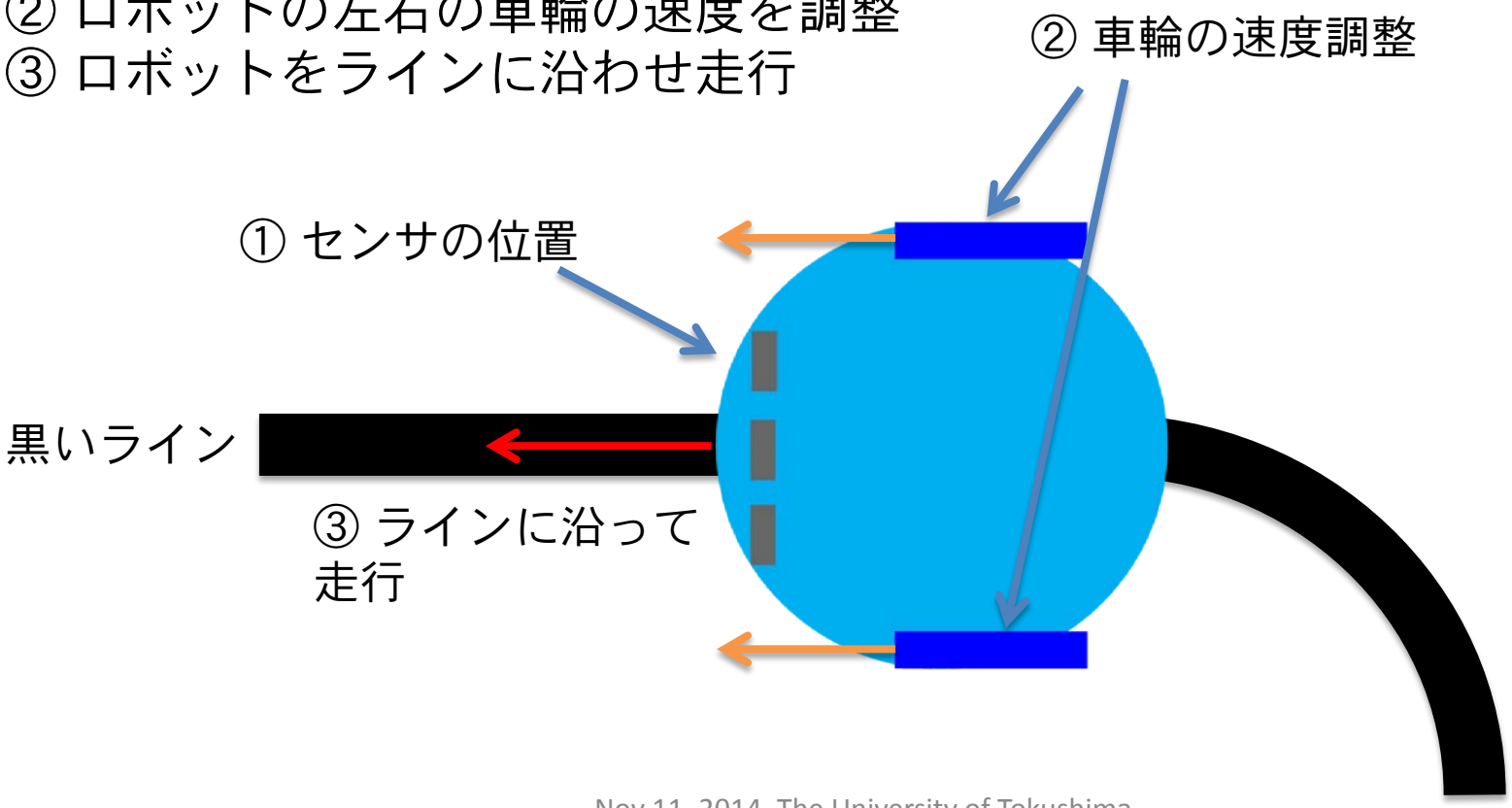


2 ライントレース

ライントレース

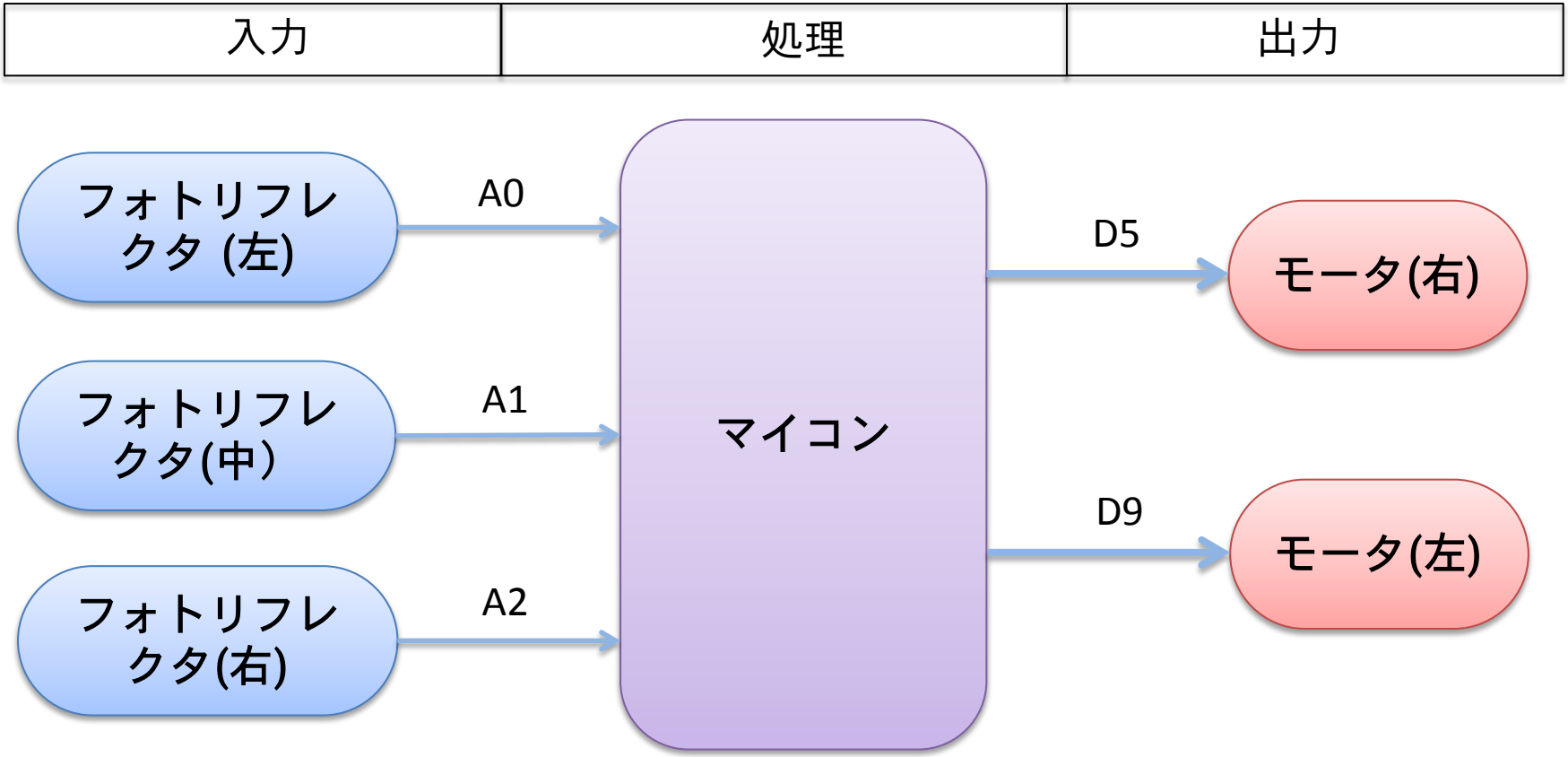
ライントレースロボット（Line Following Robot)
ラインに沿ってロボットを走行させる

① センサによりラインの位置を検出
② ロボットの左右の車輪の速度を調整
③ ロボットをラインに沿わせ走行



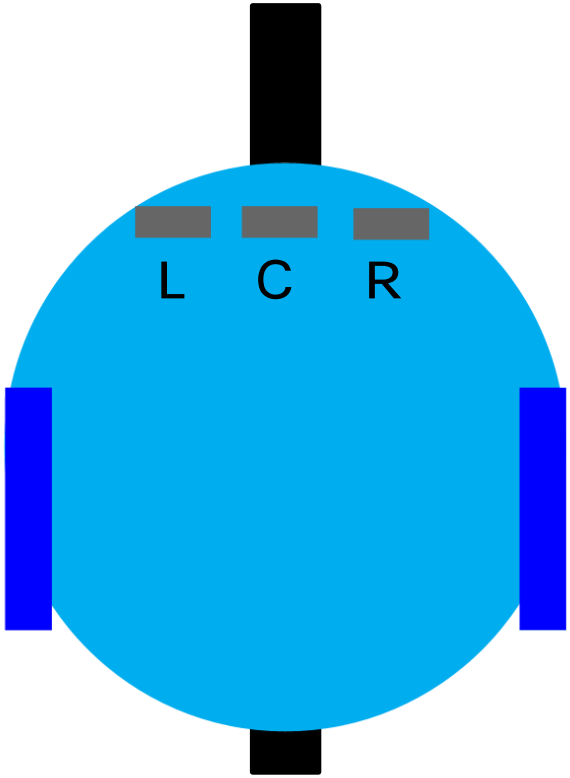
ロボットとセンサの構成

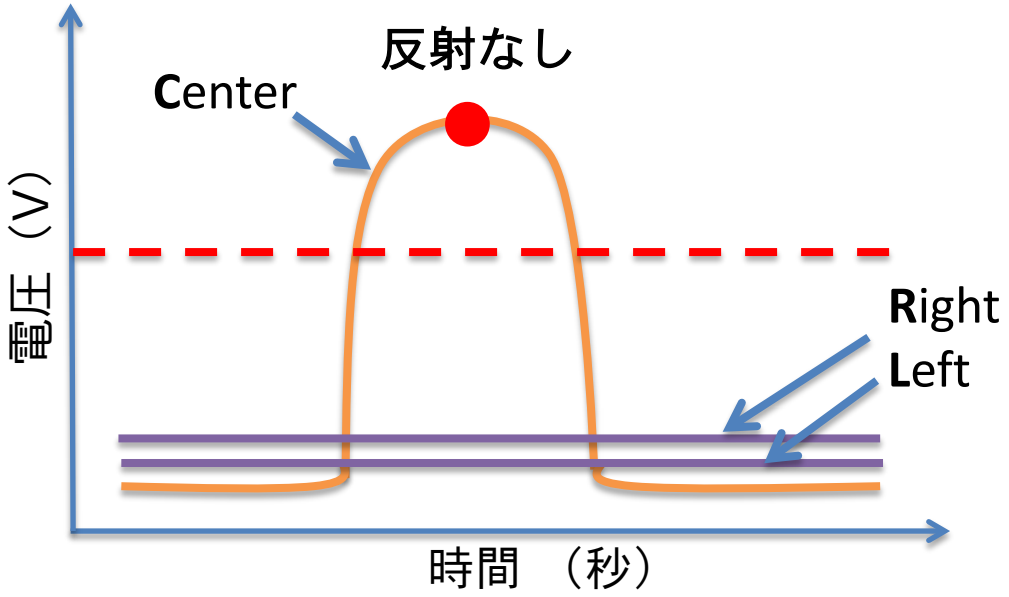
・フォトリフレクタ　3個
・サーボモータ　2個

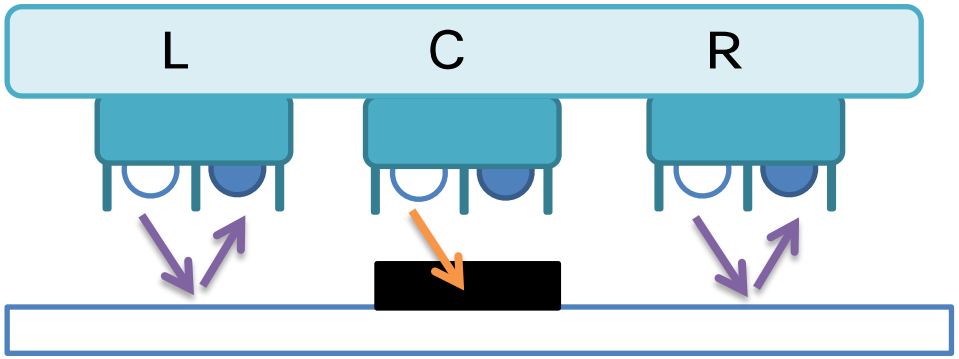


①フォトリフレクタの動作

フォトリフレクタの配置
　左　：L（Left）
　中央：C（Center）
　右　：R（Right)



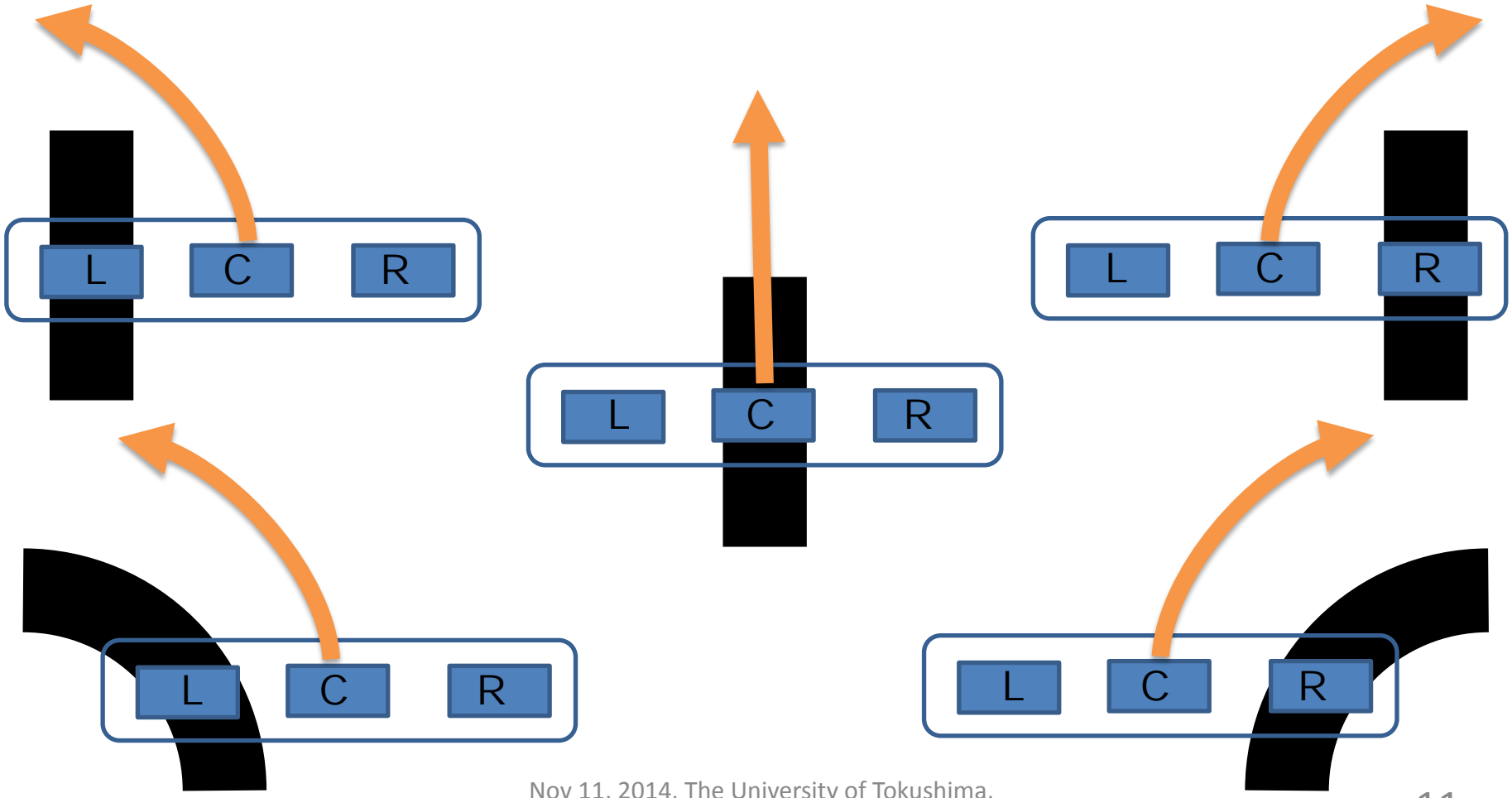




②ラインの位置とロボットの動作

ロボットがラインの中央を走るよう設計

左折	直進	右折



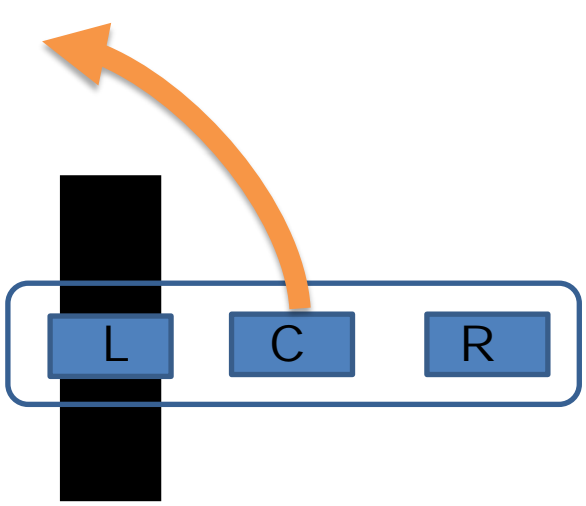
③ センサの状態に合わせロボットの動作変更

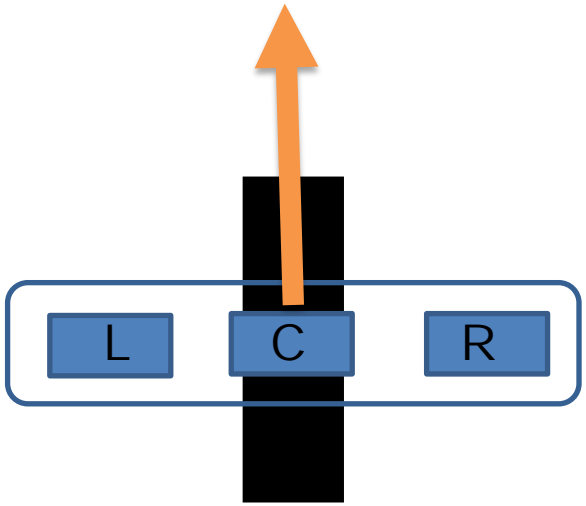
ロボットがラインの中央を走るよう設計（前回）

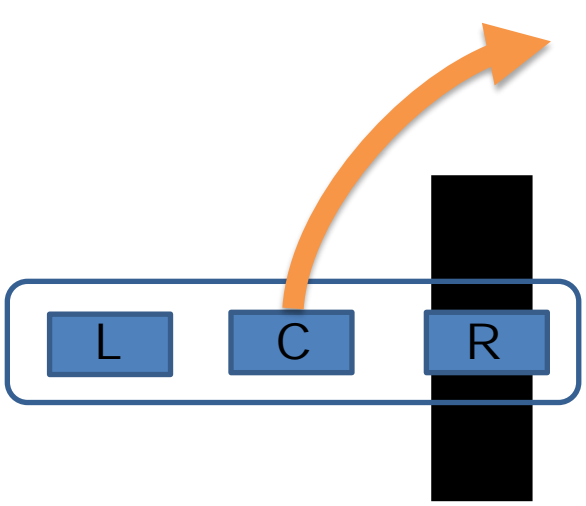
```
if (val_l > TH) {
   rot_ccw();
}
```

```
if (val_c > TH) {
   fwd();
}
```

```
if (val_r > TH) {
   rot_cw();
}
```







Example1201A ロボットのキャリブレーション（確認）

1.ロボットが停止するかを確認 motor(0,0)
2.ロボットが直進するかを確認 motor(20,20)
　centerL, centerR, pwL, pwR を調整

```
#include <Servo.h>
const int servoL_Pin  = 9; // uses timer1
const int servoR_Pin  = 5; // uses timer1
Servo servoL; // left servo
Servo servoR; // right servo
const int swPin = 8;

int centerL=1495, centerR=1472, pwL=680, pwR=720;

void setup() {
  servoL.attach(servoL_Pin, centerL-pwL, centerL+pwR);
  servoR.attach(servoR_Pin, centerR-pwL, centerR+pwR);
  motor(0, 0); // 1.停止  motor(0, 0)   2.直進   motor(20, 20)
  pinMode(swPin, INPUT_PULLUP);
  while(digitalRead(swPin));
}
```

```
void motor(int l, int r) {
  servoL.write(90-l);
  servoR.write(90+r);
}
```

ボタン(D8)を押すまでロボット動作

Example1202A フォトリフレクタの閾値（確認）

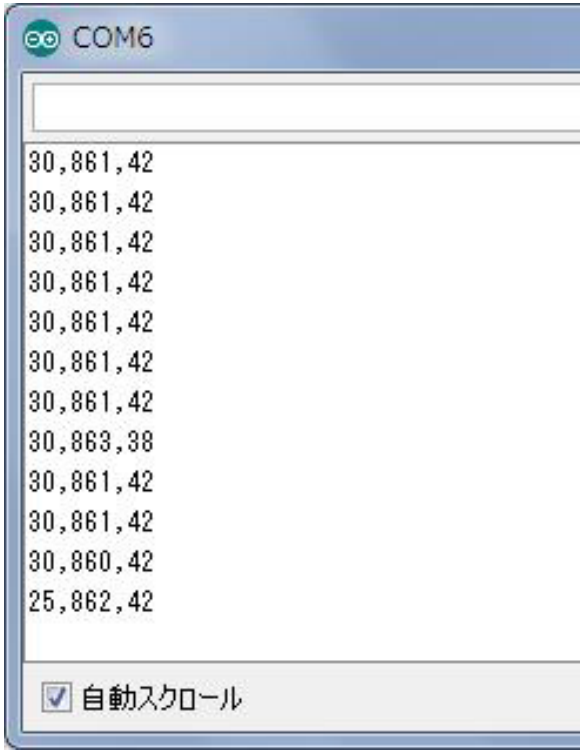
1.フォトリフレクタ（L, C, R）の値を確認
　－ライン上に，フォトリフレクタを配置して
　　L,C,Rの最小，最大値を確認
　→　閾値（TH）を検討

```
int val[3] = {0, 0, 0}; // フォトリフレクタ L, C, R

void loop() {
  val[0] = analogRead(A0); // left
  val[1] = analogRead(A1); // center
  val[2] = analogRead(A2); // right

  Serial.print(val[0]);
  Serial.print(",");
  Serial.print(val[1]);
  Serial.print(",");
  Serial.println(val[2]);
}
```

シリアルモニタにて値を確認



Example1203A ロボットの走行制御

1.フォトリフレクタの反応位置に対して，
モータの速度を調節する．

```
#define TH   750        // 閾値
int std_speed = 20;    // 速度
int val[3] = {0, 0, 0};   // L, C, R
int pos[3] = {-std_speed, 0, std_speed}; // L, C, R

void loop() {
  ・・・・・・・・・・・・・・・・・
  data_count = 0;
  sum = 0;
  for (int i = 0; i < 3; i++) {
    if (val[i] > TH) {
      sum += pos[i];
      data_count++;
    }
  }
```

センサの値が閾値を超えているか確認
センサ位置に応じた速度を設定して総和(sum) を計算する

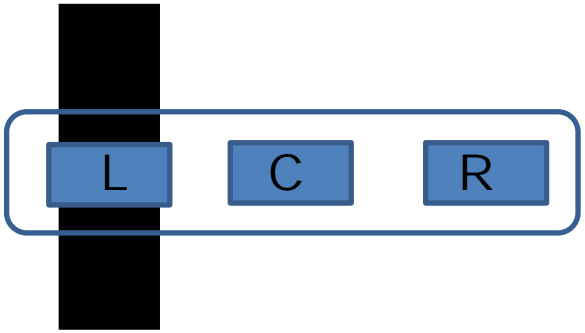
シリアルモニタにて値を確認

```
  if (data_count == 0) {
    e0 = 0;
  } else {
    e0 = sum / data_count;
  }
  Serial.println(e0);
   ・・・・・・・・・・・・・・・
}
```

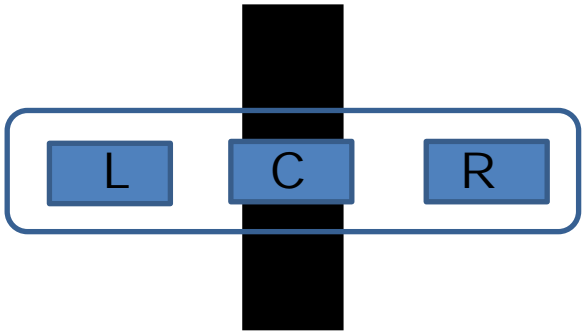
総和をラインに反応したセンサの数で割り
±std_speedの範囲におさめる

Example1203A ロボットの走行制御

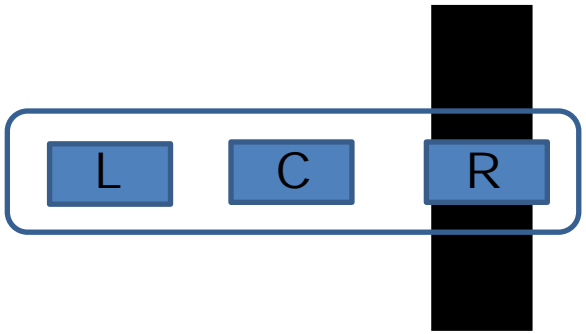
フォトリフレクタの反応位置に対するモータの速度調整量
（モータの電源を切った状態でライン上に配置して値を確認）



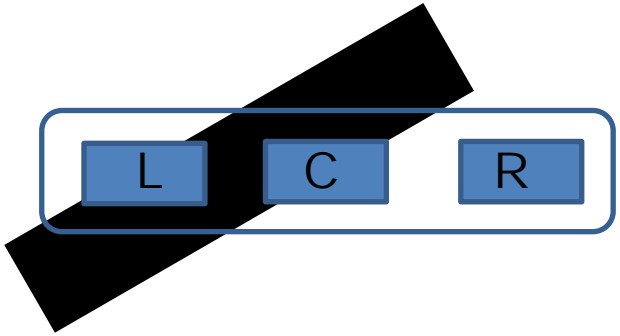
-20



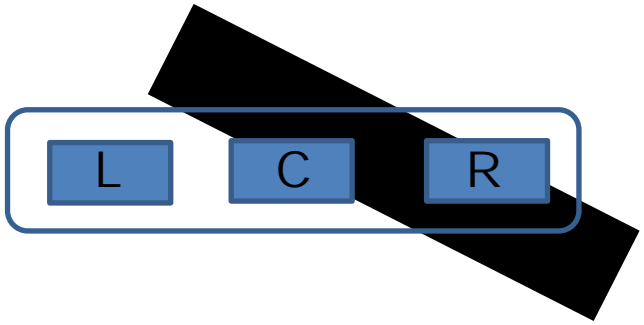
0



20



-10



10

Example1204A ロボットの走行制御（改良）

・ラインから脱線してしまうことがあれば，センサが反応しなかった場合に一つ前の速度調整量を用いる．

```
#define TH   750
int std_speed = 20;
int val[3] = {0, 0, 0}; // L, C, R
int pos[3] = {-std_speed, 0, std_speed}; // L, C, R
float e1 = 0;

void loop() {
  ・・・・・・・・・・・・・・・・・
  data_count = 0;
  sum = 0;
  for (int i = 0; i < 3; i++) {
    if (val[i] > TH) {
      sum += pos[i];
      data_count++;
    }
  }
  if (data_count == 0) {
    e0 = e1;
  } else {
    e0 = sum / data_count;
  }
  e1 = e0;
  Serial.println(e0);
}
```

3 ライントレース応用

ライントレース応用

1．自分でコースを作成してみる．
付録1参照

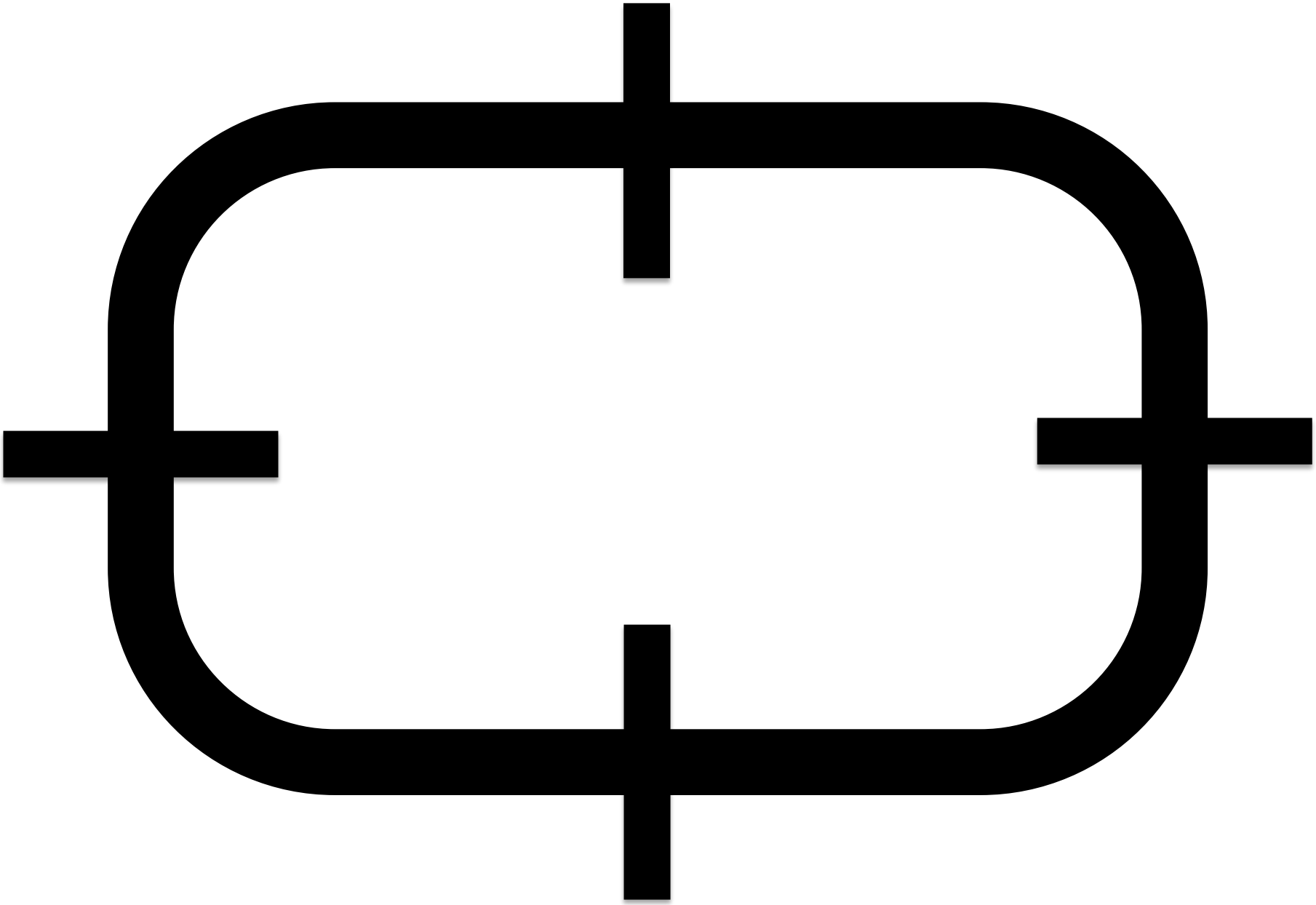
2．ラインの幅を変えてみる．
9mm, 17mmなど

3．ラインの濃さを変えてみる．
ヒント：走行前にセンサのキャリブレーションを行う．
白地，黒地の上でボタン（D8）を押してデータ取得（付録2参照）
閾値＝（白地の値－黒地の値）/ 2

4．ラインが途切れた場合を考える．
5．ラインが横切った場合を考える．
ヒント：センサの過去の値を用いる．

6．Processing，ControlP5を使ってロボットの状態をパソコンでみる．
Processing: http://www.processing.org/
ControlP5: http://www.sojamo.de/libraries/controlP5/

コース作成例

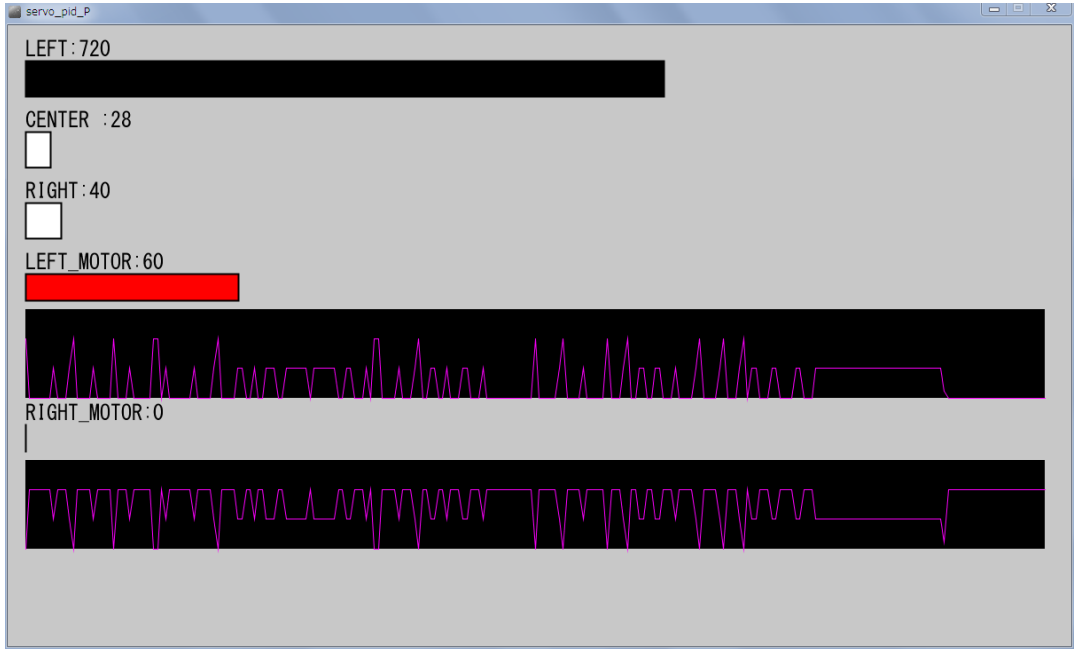


４ 次期講座内容について

次期公開講座について

1．JJ1ロボットの改良
　－ZigBee無線を搭載してパソコンと無線通信
2．今回と同じ講座内容
3．パソコン上のプログラム
　－Processing
4．まったく新しいこと
　－皆様からのご要望・ご意見を反映させた内容

（例）ZigBee無線によりロボットの内部状態をパソコン上で可視化



付録１　コースの作成

コースの作成

(1) 印刷用紙を準備する.

・A4用紙に印刷
・A3用紙に印刷
・A3用紙に印刷（または，A4用紙の141％拡大コピー）

(2) パワーポイント，ワードなどを利用してライン（円，楕円，直線，曲線など）を描く.

・線幅はポイント数(pt)で与えられるので次の換算を行う.
1pt <=> 0.353 mm
1 mm <=> 2.835 pt
1 inch <=> 25.4 mm <=> 72 pt

(例) 1 cm の線幅のラインを引きたい
1 [cm] => 10 [mm] => 10 [mm] * 2.835 [pt/mm] => 28.35 pt (約 28pt)

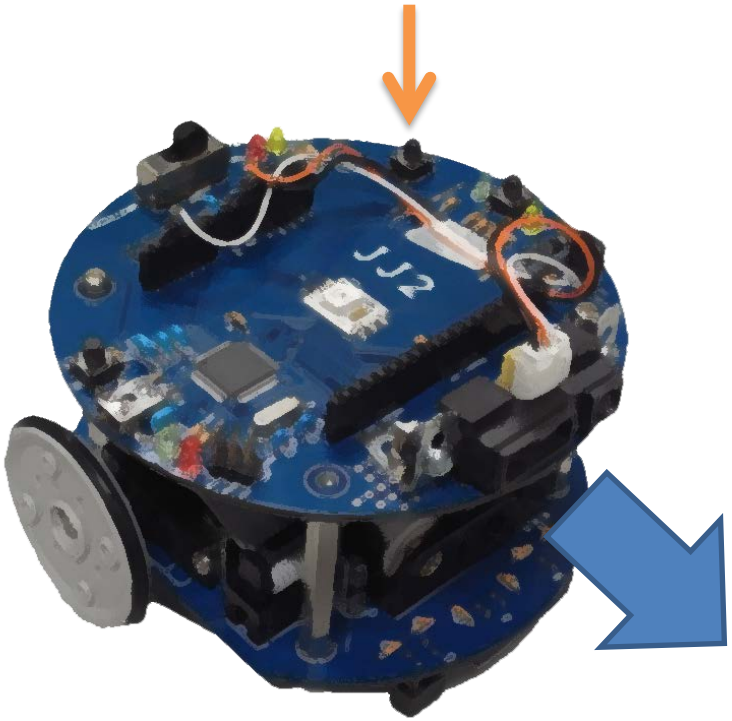
(3) 用紙に印刷する.

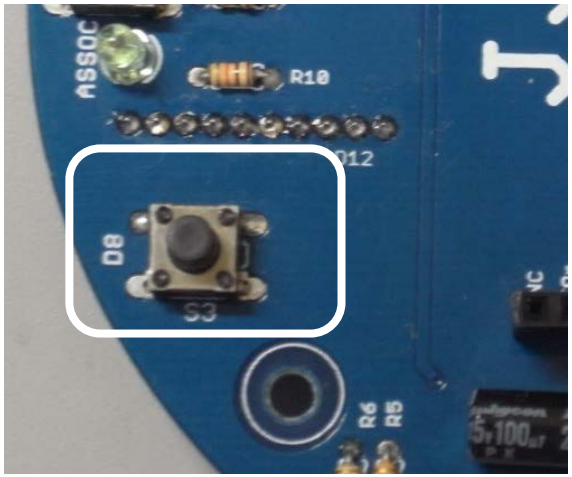
・レーザプリンタで印刷する.
・インクジェットプリンタで印刷したものをコピー機で拡大する.

付録２ ロボットのスイッチ

ロボットのスイッチ

JJのスイッチが押されたら，ロボットの動作開始



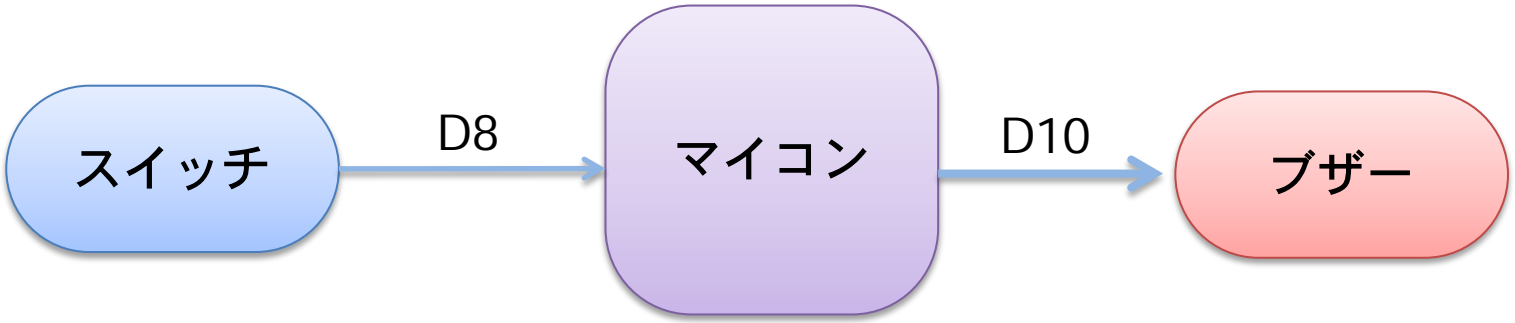


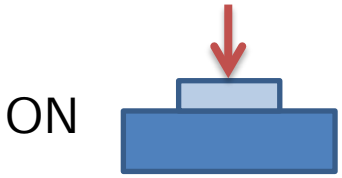
スイッチ
（D8）

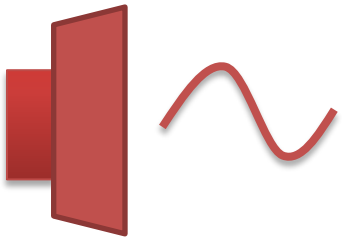
（例）スイッチが押されたらブザーを鳴らす

◆スイッチが押されたら，ブザーを鳴らす．
①　②　③　④

入力	処理	出力



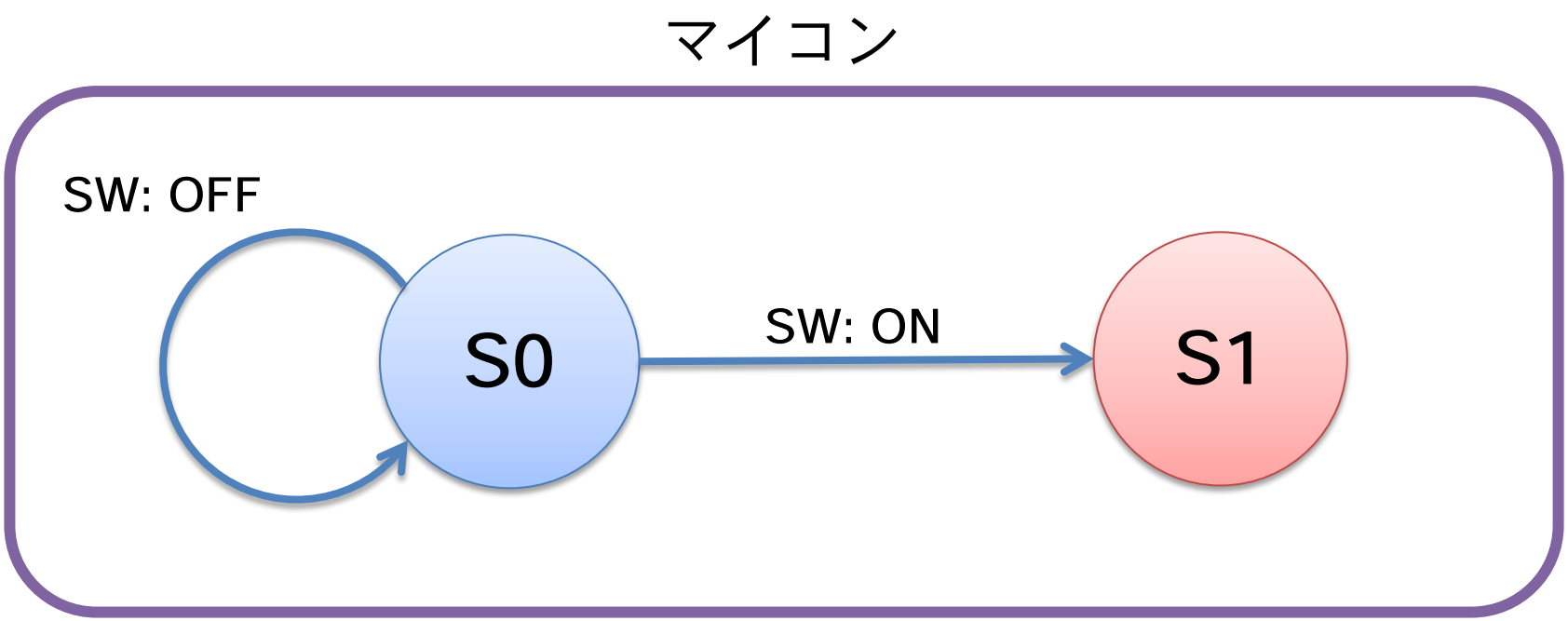




スイッチとブザーの状態遷移

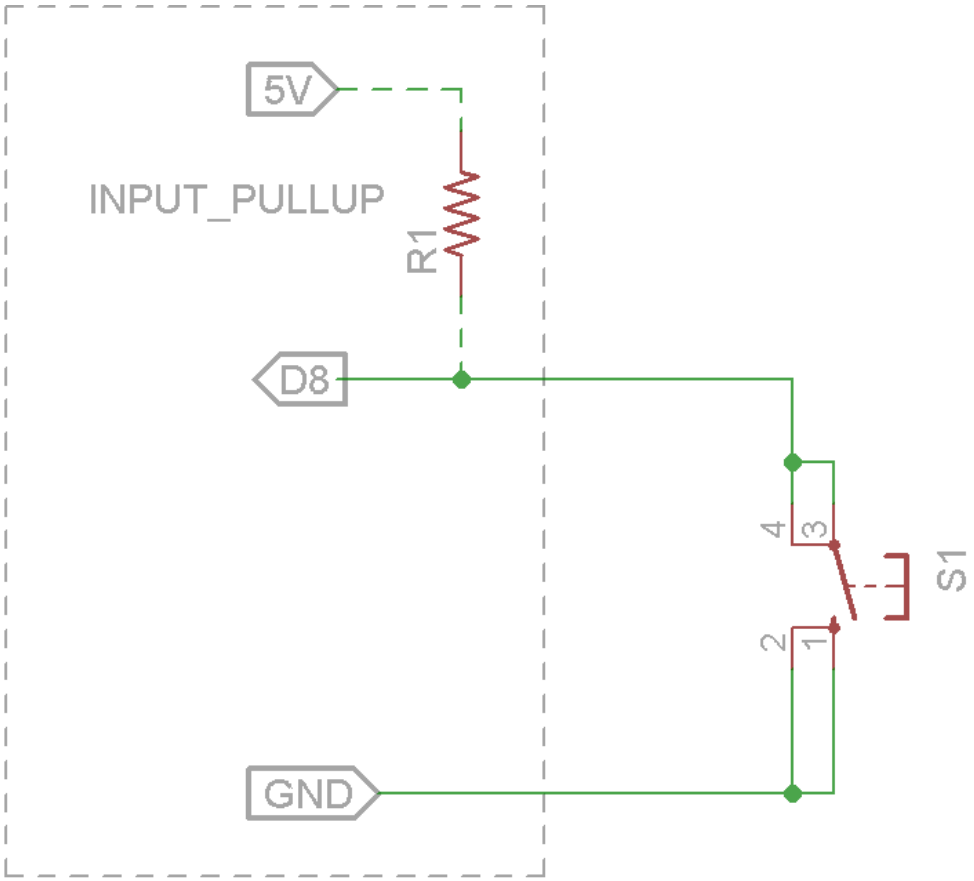
- スイッチが押されたら，ブザーを鳴らす．

スイッチの状態：ONかOFF
S0: スイッチ(SW)が押されるまで待つ
S1: ブザーを鳴らす



スイッチ配線の確認

JJマイコン



スイッチの状態	D8
ON	0 (LOW)
OFF	1 (HIGH)

デジタル：8番ポート
スケッチにて，INPUT_PULLUPを定義

スイッチが押されたらブザーを鳴らすスケッチ

```
① const int swPin = 8;
③ const int buzzerPin = 10;
  void setup() {
②     pinMode(swPin, INPUT_PULLUP);  // 8番を入力(H)に設定
      while (digitalRead(swPin));  // スイッチが押されるまで待つ
④     buzzer_beep(400, 500); // 400Hz, 500ms
      buzzer_beep(800, 500); // 800Hz, 500ms
  }
  void loop() {}
  void buzzer_beep(int f, int d) {  // 周波数, 継続時間
      tone(buzzerPin, f, d);
④     delay(d);
      noTone(buzzerPin);
  }
```