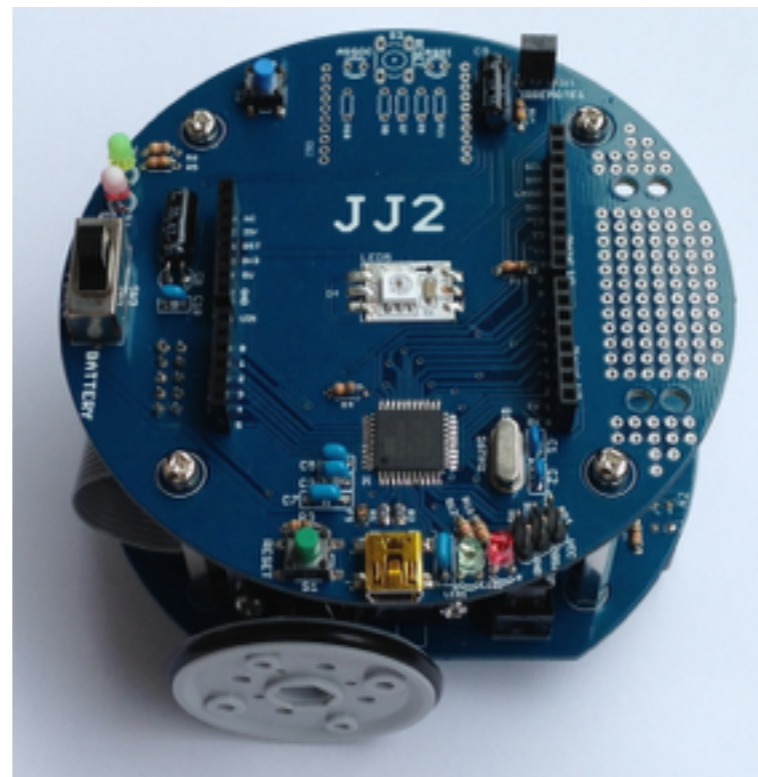


ロボットをつくらう (後編)

第11回 ロボットの走行制御 1



<http://cms.db.tokushima-u.ac.jp/DAV/person/S10723/>

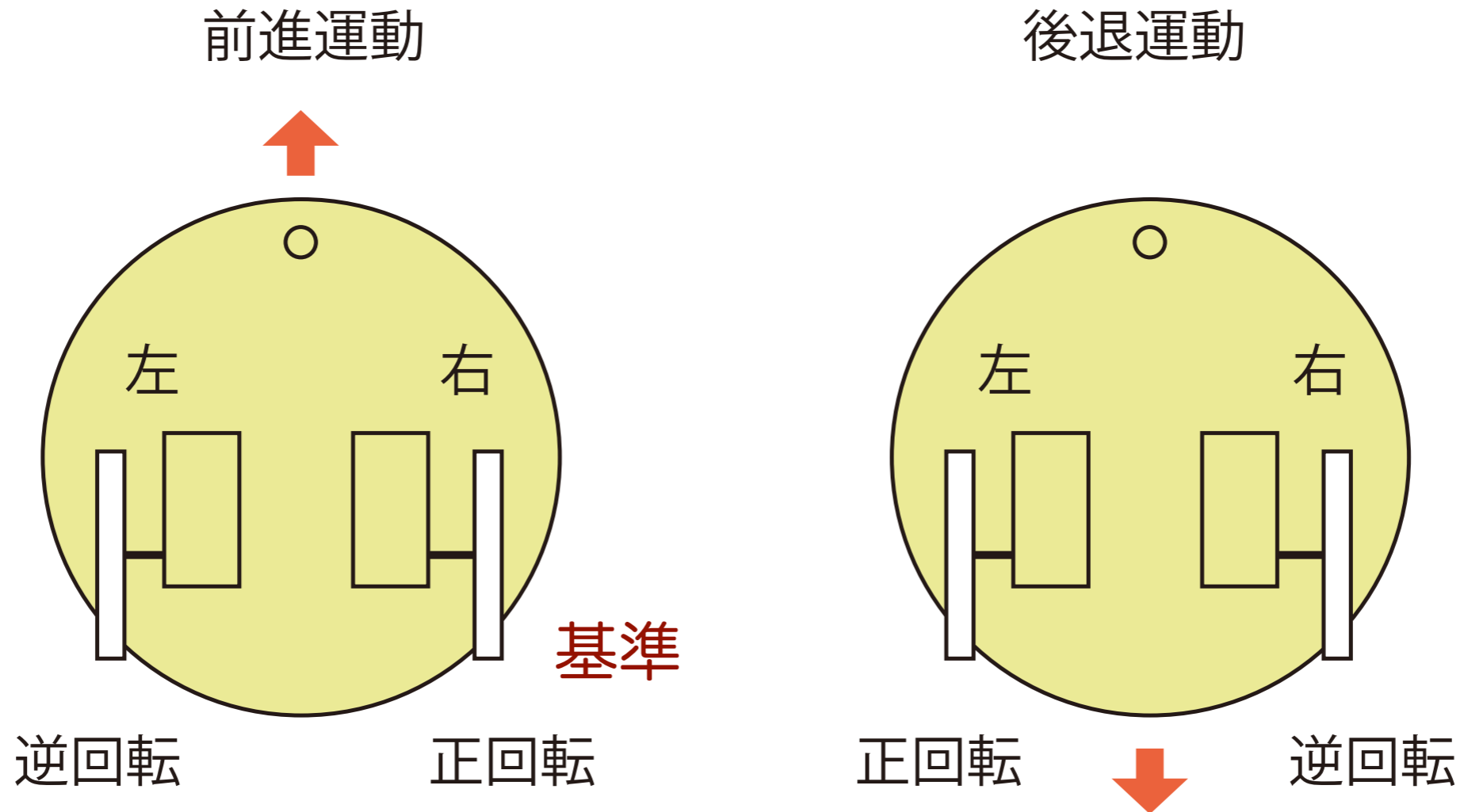
川上 博

2014/11/08

今日のテーマ

- ◎ 2輪ロボットの走行制御：管理者目線か現場目線
- ◎ 黒い曲線に追従する：ON/OFF 制御
- ◎ 状態とその遷移条件：Finite State Machine(FSM)

JJ2の運動：前進，後退

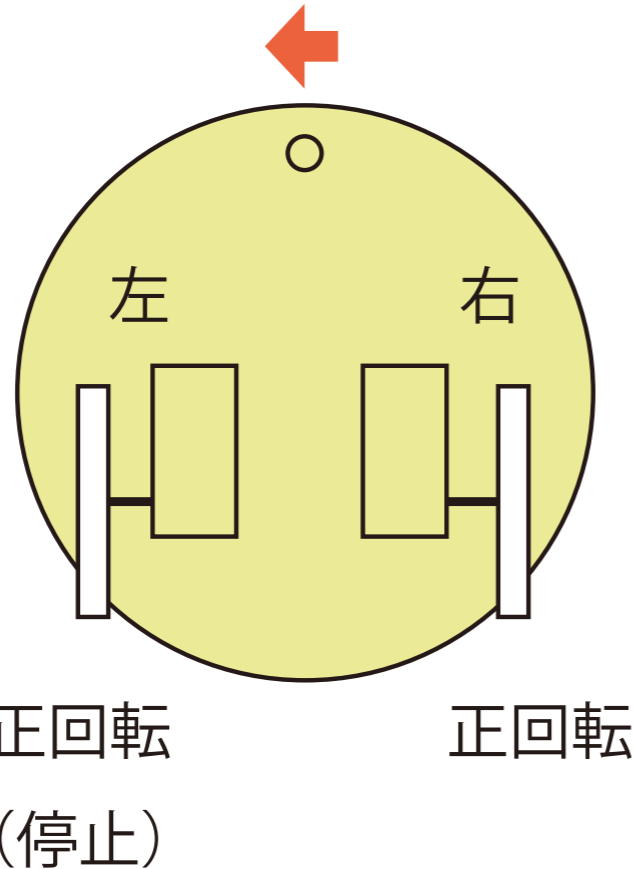


```
int spd=20;
void fwd(){
    servoL.write(90-spd);
    servoR.write(90+spd);
}
```

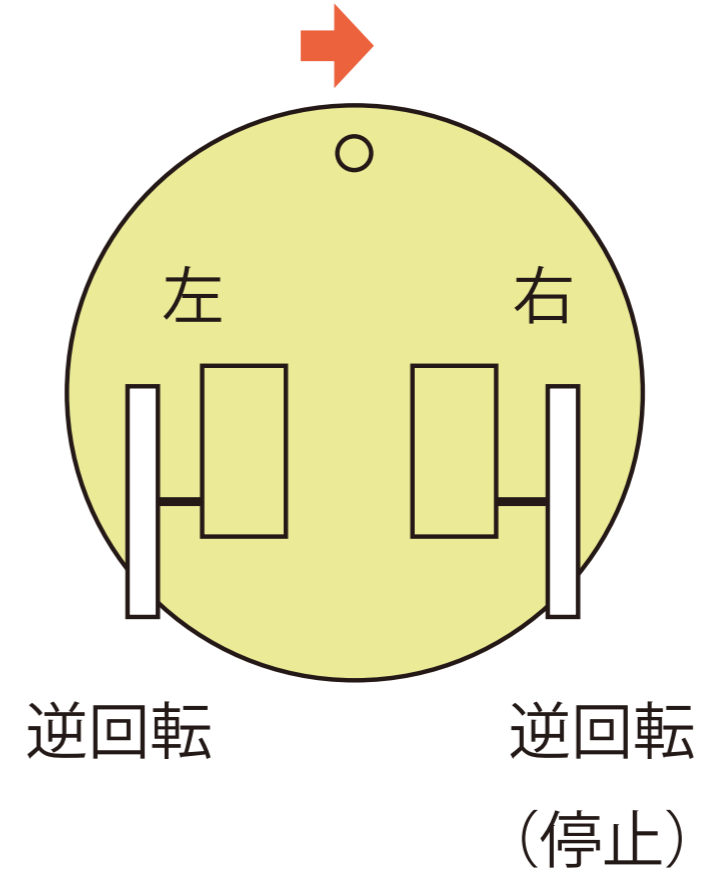
```
void bwd(){
    servoL.write(90+spd);
    servoR.write(90-spd);
}
```

JJ2の運動：左折，右折

左折運動



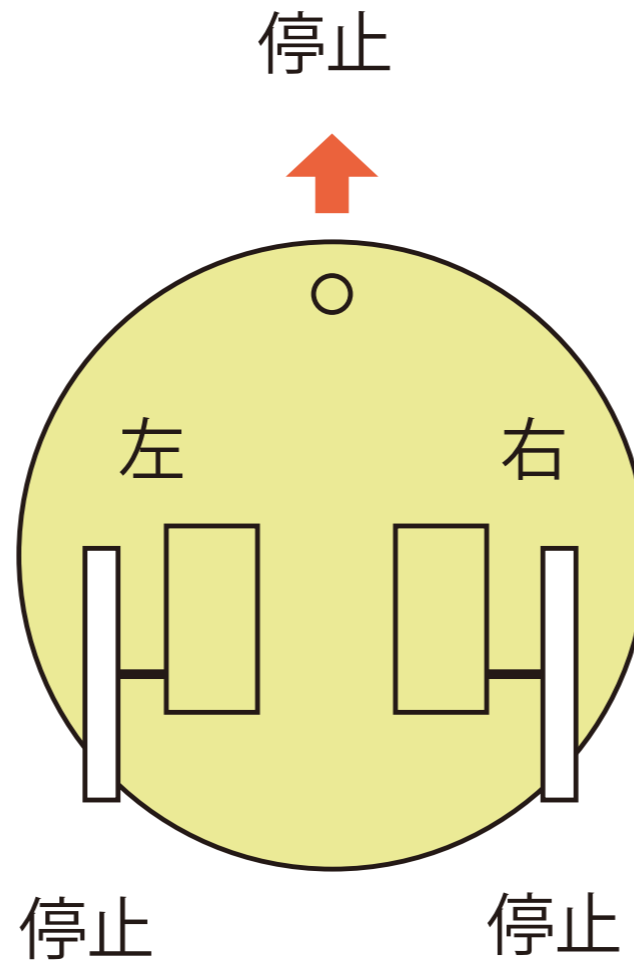
右折運動



```
int spd=20;
void ccw(){
    servoL.write(90);
    servoR.write(90+spd);
}
```

```
void cw(){
    servoL.write(90+spd);
    servoR.write(90);
}
```

JJ2の運動：停止

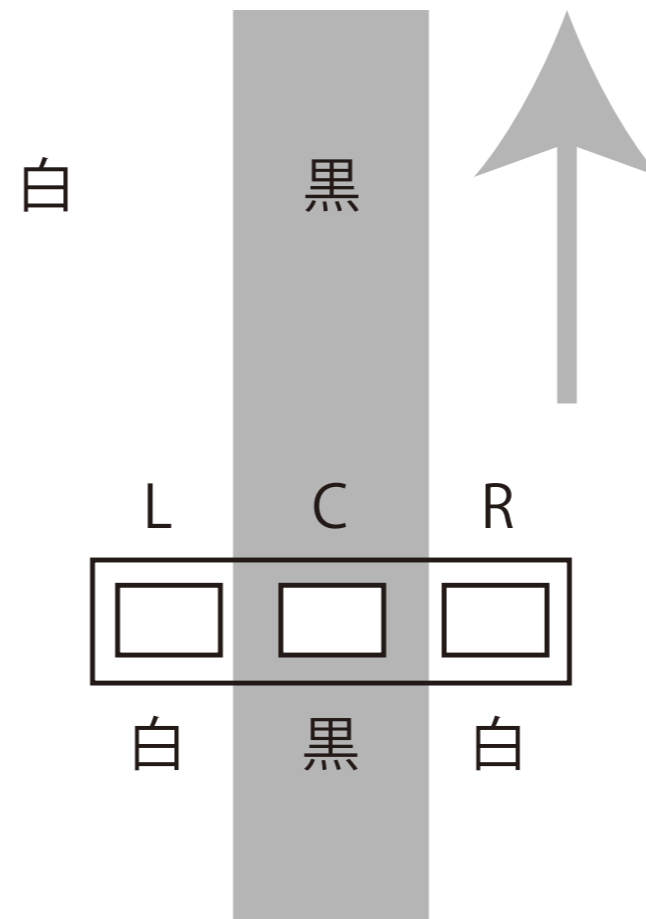


```
void stp(){  
    servoL.write(90);  
    servoR.write(90);  
}
```

制御：4つの要素

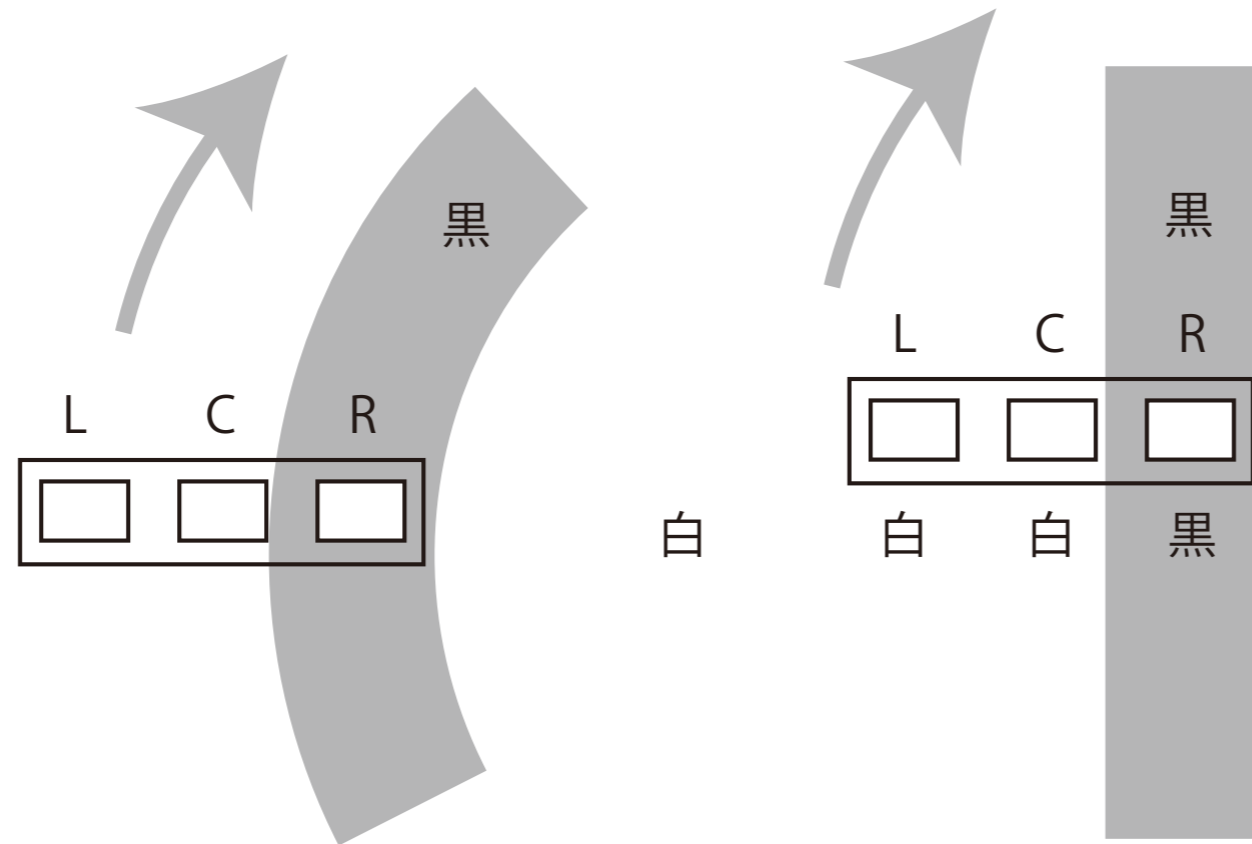
- ① 2輪ロボット：制御対象(fwd, CW, CCW, stop)
- ② 走行環境：軌道曲線
- ③ センサー：エラーの検出：photo reflector
白：値小(30 - 40), 黒：値大(600 - 800)
- ④ 制御器（コントローラ）：
supervisory control vs low level control
ON/OFF vs PID

前進：左右（白），中央（黒）



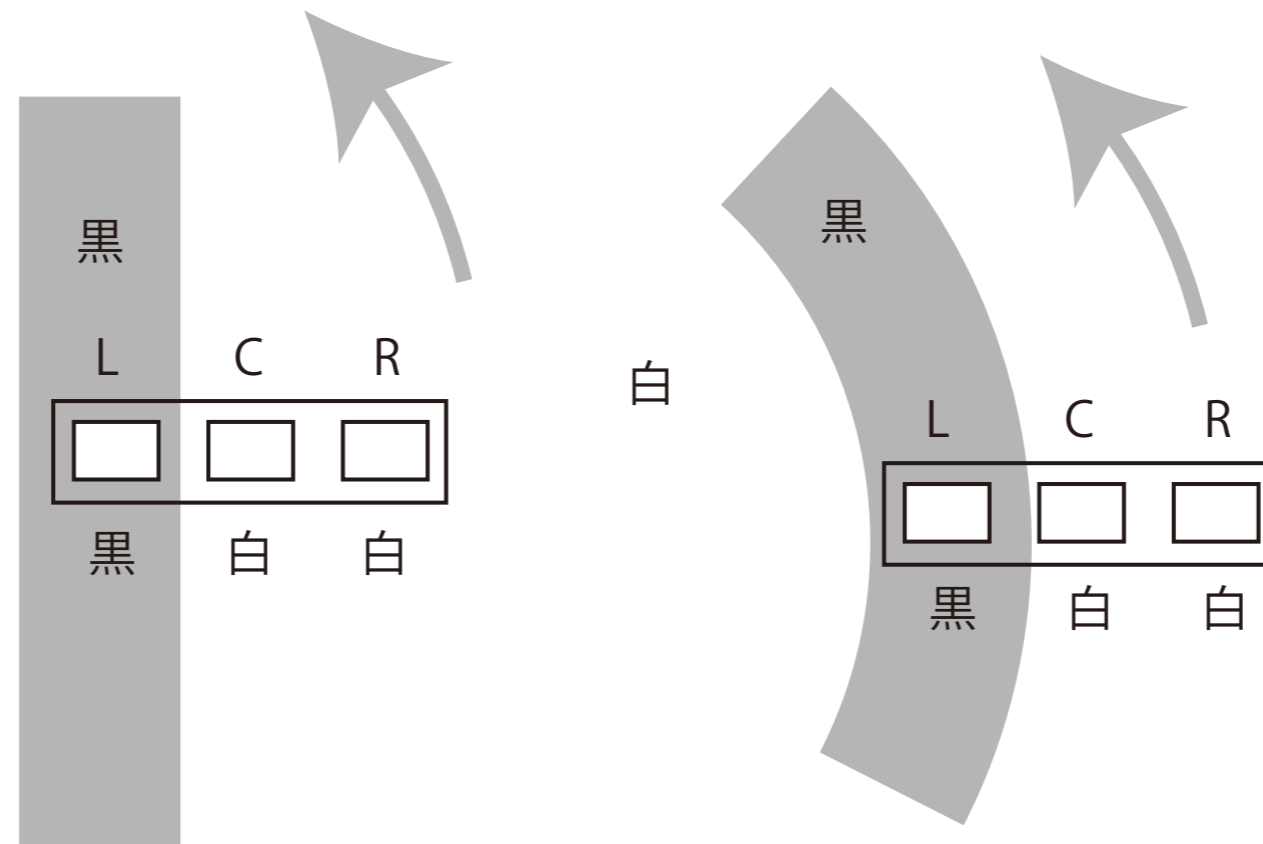
- 条件：
- ① $val_L < th \ \&\& \ val_C > th \ \&\& \ val_R < th$
 - ② $val_C > th$
 - ③ $val_L < th \ \&\& \ val_R < th$

右折：左・中央（白），右（黒）



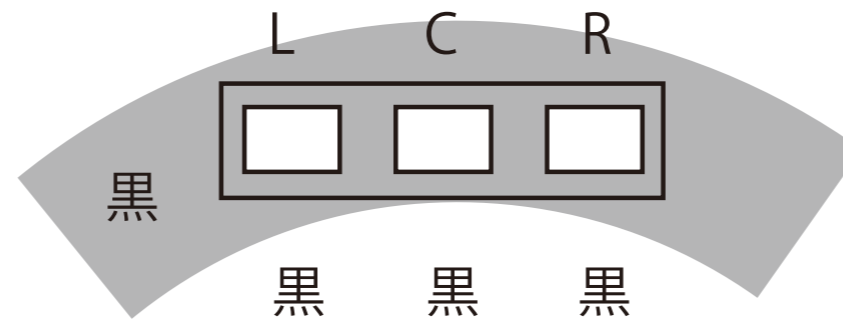
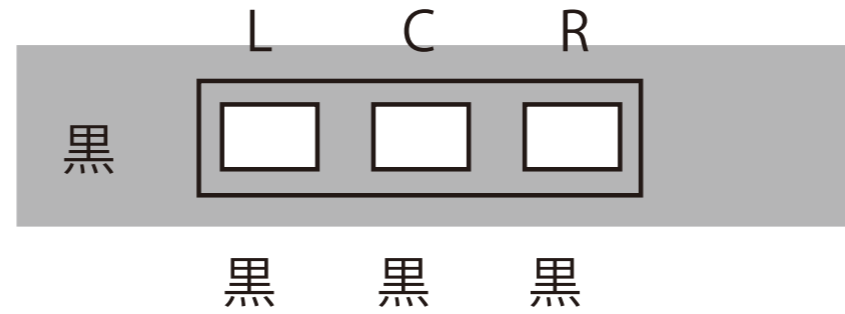
条件： ① $val_L < th \ \&\& \ val_C < th \ \&\& \ val_R > th$
② $val_R > th$

左折：右・中央（白），左（黒）



- 条件：
- ① $val_L > th \ \&\& \ val_C < th \ \&\& \ val_R < th$
 - ② $val_L > th$

停止：右左（黒），中央（黒）



- 条件：
- ① $val_L > th \ \&\& \ val_C > th \ \&\& \ val_R > th$
 - ② $val_L > th \ \&\& \ val_R > th$

Example1101A：とりあえず動かす

3つのセンサーからの入力：

センサーからの入力の表示：

```
// Example1101A

#include <Servo.h>

const int servoL_Pin = 9;
const int servoR_Pin = 5;
Servo servoL;
Servo servoR;

// Left: 1530-1543:1536
//Right: 1490-1503:1496
int centerR=1510, centerL=1541, pw=700;

void setup() {
  delay(1000);
  servoL.attach(servoL_Pin, centerL-pw, centerL+pw);
  servoR.attach(servoR_Pin, centerR-pw, centerR+pw);
  Serial.begin(9600);
}
```

```
int th=200, val_L, val_C, val_R;
void loop() {
  val_L=analogRead(A0);
  val_C=analogRead(A1);
  val_R=analogRead(A2);
  prtData(val_L, val_C, val_R);
  if(val_L>th && val_C>th && val_R>th){
    stp();
  }else if(val_L<th && val_C>th && val_R<th){
    fwd();
  }else if(val_L<th && val_C<th && val_R>th){
    rot_CW();
  }else if(val_L>th && val_C<th && val_R<th){
    rot_CCW();
  }else{
    fwd();
  }
}
```

Example1101Abis : とりあえず動かす

```

// Example1101A

#include <Servo.h>

const int servoL_Pin = 9;
const int servoR_Pin = 5;
Servo servoL;
Servo servoR;

// Left: 1530-1543:1536
//Right: 1490-1503:1496
int centerR=1510, centerL=1541, pw=700;

void setup() {
  delay(1000);
  servoL.attach(servoL_Pin, centerL-pw, centerL+pw);
  servoR.attach(servoR_Pin, centerR-pw, centerR+pw);
  Serial.begin(9600);
}

int th=200, val_L, val_C, val_R;
void loop() {
  3つのセンサーからの入力 : val_L=analogRead(A0);
                             val_C=analogRead(A1);
                             val_R=analogRead(A2);
  センサーからの入力の表示 : prtData(val_L, val_C, val_R);
  if(val_L>th && val_C>th && val_R>th){
    stp();
  }else if(val_C>th){
    fwd();
  }else if(val_R>th){
    rot_CW();
  }else if(val_L>th){
    rot_CCW();
  }else{
    fwd();
  }
}

```

使う関数

前進, 停止, 左折, 右折

```
int spd=20;
void fwd(){
    servoL.write(90-spd);
    servoR.write(90+spd);
}
void stp(){
    servoL.write(90);
    servoR.write(90);
}
void rot_CCW(){
    servoL.write(90);
    servoR.write(90+spd);
}
void rot_CW(){
    servoL.write(90-spd);
    servoR.write(90);
}
```

3つのセンサーからの入力を表示する関数:

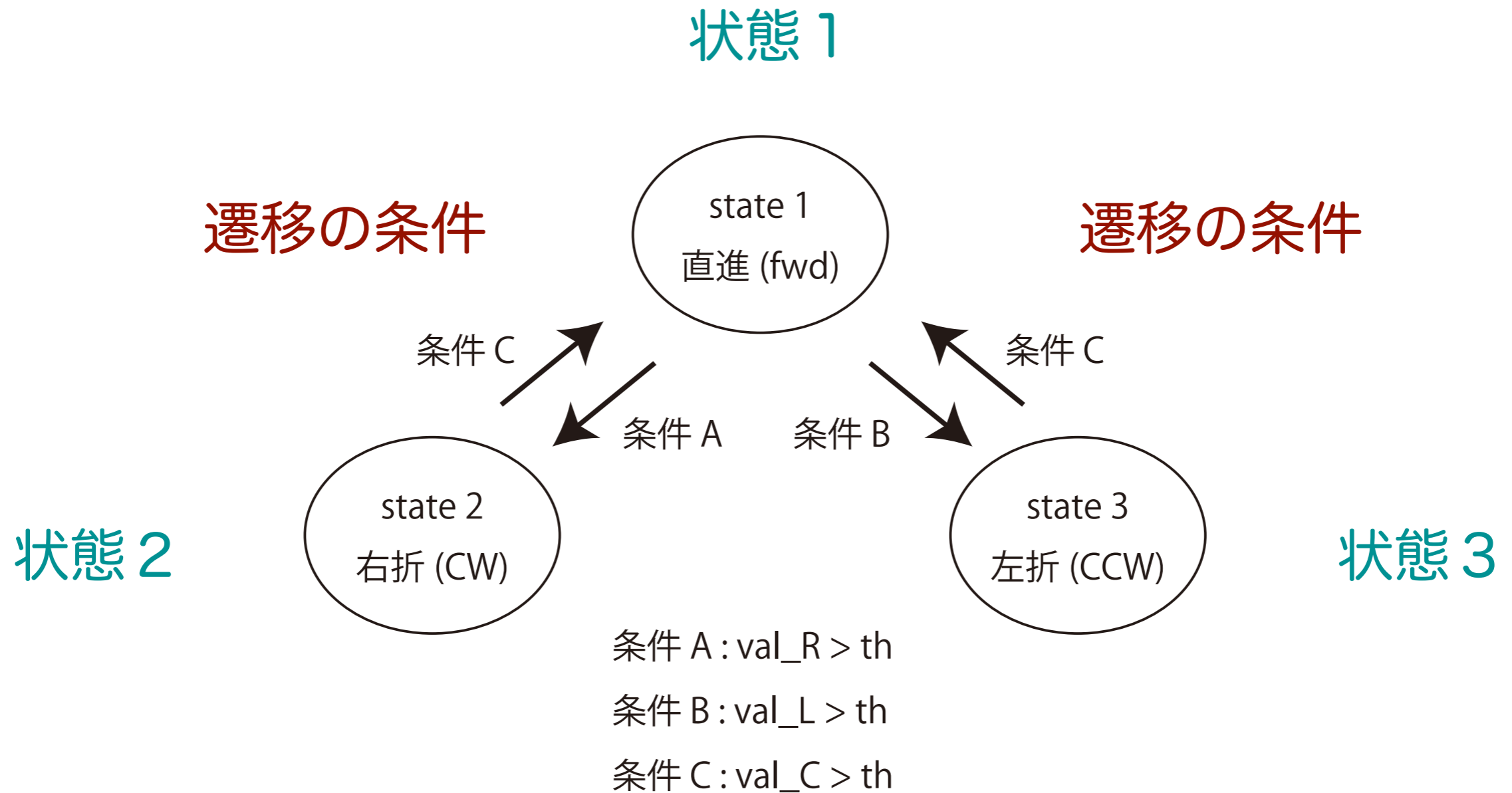
```
void prtData(int x, int y, int z){
    Serial.print("val_L:");
    Serial.print(x,DEC);
    Serial.print("    val_C:");
    Serial.print(y,DEC);
    Serial.print("    val_R:");
    Serial.println(z,DEC);
}
```

Example1102A : LED5も一緒に光らせる

```
int th=200, val_L, val_C, val_R;
int col=127; 光の強さを表す値 : 0~255

void loop() {
  val_L=analogRead(A0); val_C=analogRead(A1); val_R=analogRead(A2);
  prtData(val_L, val_C, val_R);
  if(val_L>th && val_C>th && val_R>th){
    stp(); leds[0]= CRGB(0, 0, 0); LEDS.show();
  }else if(val_L<th && val_C>th && val_R<th){
    fwd(); leds[0]= CRGB(0, col, 0); LEDS.show(); //green
  }else if(val_L<th && val_C<th && val_R>th){
    rot_CW(); leds[0]= CRGB(0, 0, col); LEDS.show(); //blue
  }else if(val_L>th && val_C<th && val_R<th){
    rot_CCW(); leds[0]= CRGB(col, 0, 0); LEDS.show(); //red
  }
}
```

Example1103A : 状態機械 (State Machine)



状態遷移図 (state transition diagram)

Example1103A : 状態機械 (State Machine)

```
int th=200, val_L, val_C, val_R;
int state=1; // initial state

void loop() {
  val_L=analogRead(A0); val_C=analogRead(A1); val_R=analogRead(A2);
  switch(state){
    case 1: fwd();
      if(val_L<th && val_C<th && val_R>th) state=2;
      }else if(val_L>th && val_C<th && val_R<th) state=3;
      break;
    case 2: rot_CW();
      if(val_L<th && val_C>th && val_R<th) state=1;
      break;
    case 3: rot_CCW();
      if(val_L<th && val_C>th && val_R<th) state=1;
      break;
    default:
      break;
  }
  // delay(40);
}
```


センサー情報：3 ビット（8つの場合）を仮定すると

	val_L	val_C	val_R
	白	白	白
右折	白	白	黒
前進	白	黒	白
	白	黒	黒
左折	黒	白	白
	黒	白	黒
	黒	黒	白
停止	黒	黒	黒

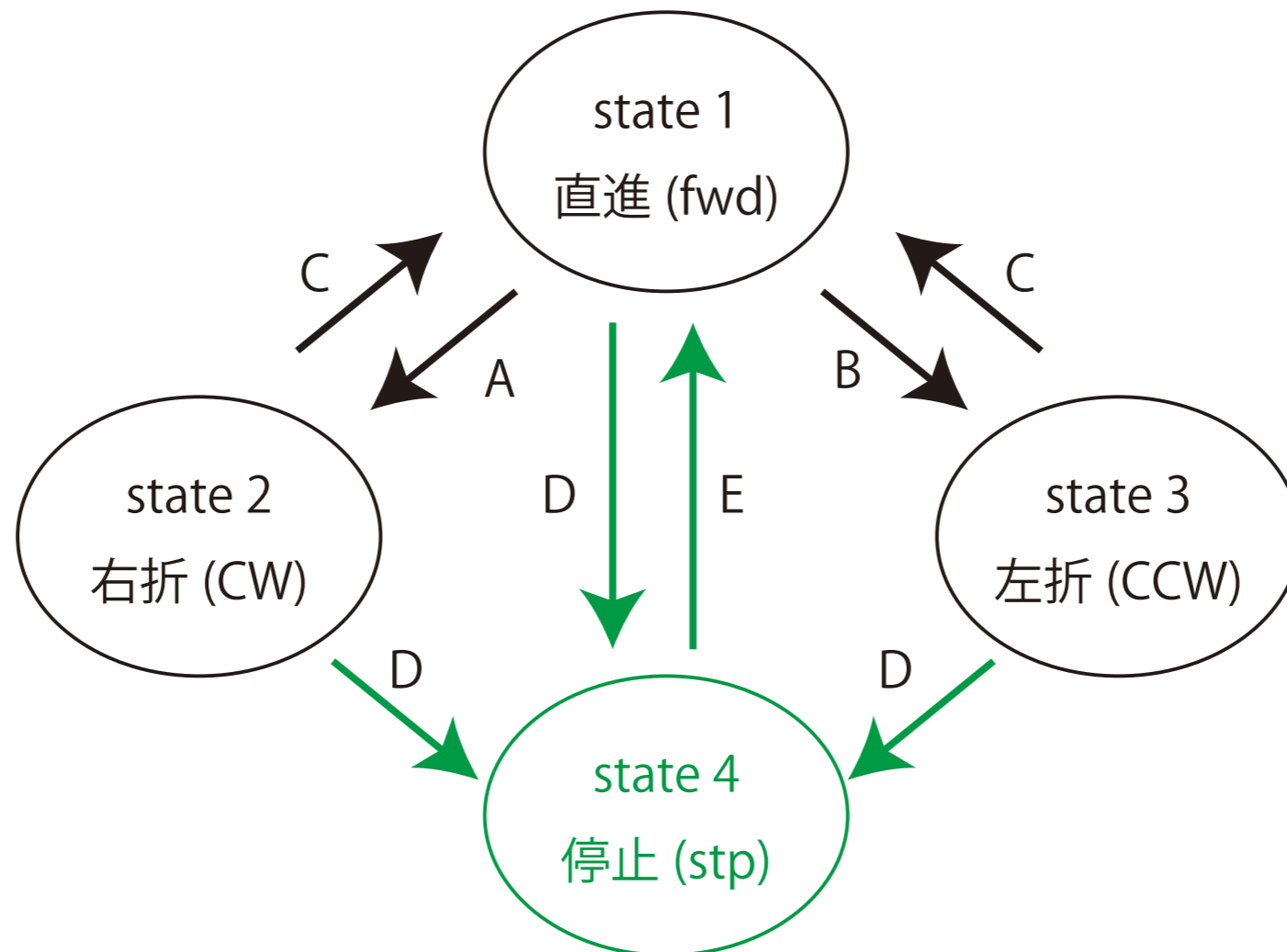
条件式の違いを考えよう

例：「前進」について次の3つの条件式の違いを考えよう

- ① $val_L < th \ \&\& \ val_C > th \ \&\& \ val_R < th$
- ② $val_C > th$
- ③ $val_L < th \ \&\& \ val_R < th$

クイズ：停止状態とその遷移条件を考え、スケッチを書いてみよう

Example1104A : 停止状態を加えた例



A : $val_R > th$, B : $val_L > th$, C : $val_C > th$

D : $val_L > th \ \&\& \ val_C > th \ \&\& \ val_R > th$

E : $!(val_L > th \ \&\& \ val_C > th \ \&\& \ val_R > th)$

Example1104A : 停止状態を加えた例

```
int th=200, val_L, val_C, val_R, state=1;
void loop() {
  val_L=analogRead(A0); val_C=analogRead(A1); val_R=analogRead(A2);
  switch(state){
    case 1: fwd(); //fwd
      if(val_L>th && val_C>th && val_R>th) state=4;
      else if(val_L<th && val_C<th && val_R>th) state=2;
      else if(val_L>th && val_C<th && val_R<th) state=3;
      break;
    case 2: rot_CW(); //CW
      if(val_L>th && val_C>th && val_R>th) state=4;
      else if(val_L<th && val_C>th && val_R<th) state=1;
      break;
    case 3: rot_CCW(); //CCW
      if(val_L>th && val_C>th && val_R>th) state=4;
      else if(val_L<th && val_C>th && val_R<th) state=1;
      break;
    case 4: stp(); //stp
      if(!(val_L>th && val_C>th && val_R>th)) state=1;
      break;
    default:
      break;
  }
  // delay(40);
}
```