

# ETロボについて

## □ 概要

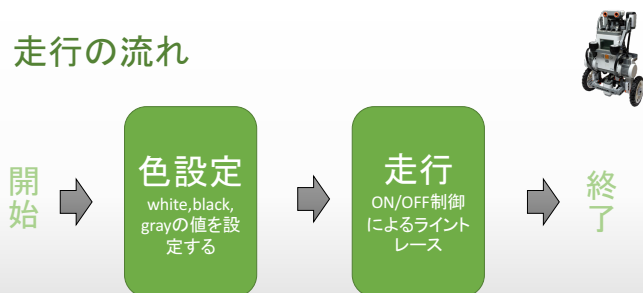
1. ETロボの各部について
2. 走行の流れ
3. 色の設定
4. 色の判断
5. 距離検出について



## ETロボの各部



## 走行の流れ



## 色の設定

※画面イメージ

|       |     |
|-------|-----|
| white | 600 |
| black | 800 |
| gray  | 680 |

- ・ 閾値を決めるにあたってのwhiteとblack, そして、灰色検出のためのgrayの値は重要で、環境が変化することにより、設定しなおさなければならない。
- ・ よって、calibrator()関数を用いて、1回走行させるごとにこれらの値を取得し、画面に表示した。
- ・ また、10msごとに3回抽出し、それらの平均値を設定した。



## 白黒の判断

white = 600  
black = 800 としたとき、その閾値は gray = 680

$(W+B)/2 = 700$ となる。センサーの値が700位上るとき“黒”と判断し、700未満のとき“白”と判断する

しかし、この方法では、灰色を判断できない (680は、“白”と検出する)

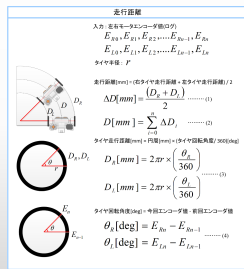


## 灰色の判断

- ・ “黒”でない状態が、20mm続いた場合に、閾値を“灰色-30”とすることで、灰色の検知を可能とした
- ・ 距離の検出は、サーボモータに内蔵されている回転角度検出センサ(ロータリエンコーダ)を利用した



## ロータリエンコーダについて



“とある様”組み込みエンジニアのブログ「ETロボコン状態計測 その1」より引用

- 左図を参考に、10msごとに、10ms前からどれだけ進んだかを計算するコードを作成した。
- 計算された値を走行距離を格納する変数に加算することで、何mm走行したかを計算した。

## 走行距離計算



<グローバル変数>

float r\_kyori, r\_soukyori = 0.0, l\_kyori, l\_soukyori = 0.0, r\_kyori0 = 0.0

```
~~~~~
static int r_angle, l_angle, r_angle0, l_angle0;
```

```
r_angle = r_angle0 - nxt_motor_get_count(NXT_PORT_B);
if(r_angle < 0) r_angle = -r_angle;
r_kyori = 2 * 3.141592 * 75 * ((double)r_angle/360);
r_soukyori += r_kyori;
```

```
l_angle = l_angle0 - nxt_motor_get_count(NXT_PORT_C);
if(l_angle < 0) l_angle = -l_angle;
l_kyori = 2 * 3.141592 * 75 * ((double)l_angle/360);
l_soukyori += l_kyori;
```

```
r_angle0 = nxt_motor_get_count(NXT_PORT_B);
l_angle0 = nxt_motor_get_count(NXT_PORT_C);
~~~~~
```

右に作成したコードを示す

## 気づいたこと



- 最初のスタート時に振動を与えるとブレる(対策:スイッチを押して50ms後にスタートさせるなど)
- コードに誤りがあり、無限ループに入ってしまった場合、電池を取り出す他、停止させる方法がない(めんどくさい)
- コネクタがしっかり刺さっていない状態だと、誤作動を起こしやすい
- とにかく時間をかけて試行錯誤することが大事である。
- 組み込み系はやはり楽しいです。

## 参考文献



- <http://gihyo.jp/dev/serial/01/etrobo/0002>

ETロボの画像

- <http://embedded0engineer.blog.fc2.com/blog-entry-21.html>

移動距離計算について