



実践ロボットプログラミング

LEGO Mindstorms NXT で目指せロボコン!

WEB : <http://robot-programming.jp/>

著者 : 藤吉弘亘, 藤井隆司, 鈴木裕利, 石井成郎

E-mail : support@robot-programming.jp



リフレクションしよう (p.161～)

- ・ リフレクションとは (p.161～)
- ・ 作成中のリフレクション (p.162～)
- ・ 作業記録のポイント (p.163～)
- ・ 作成後のリフレクション (p.164～)



リフレクションとは (p.161~)

- ・ 「自分の活動を振り返って評価する」 活動のこと
- ・ 作成中のリフレクション
 - 作業記録の作成
- ・ 作成後のリフレクション
 - プロセスチャートの作成



作成中のリフレクション (p.162~)

- ・ 定期的に自分のロボット作りのPDSサイクルを記録しましょう
- ・ 記録項目
 - 日時
 - PLAN：アイデアの概要（スケッチを含む）
 - DO：作成したメカニズム・プログラムの概要
 - SEE：成功／失敗とその原因
 - 気づいたこと、わかったこと



作業記録の例

20XX/10/21 13:30~ 実習室

- Plan: ラインを追跡する!



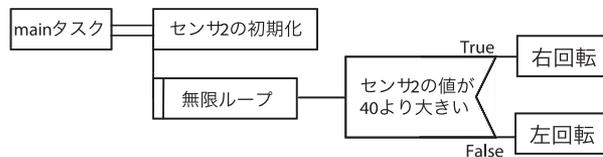
白で左, 黒で右に回転させる

- Do:

- マシン



- プログラム



- See: 失敗

- マシン
スピードが遅い
- プログラム
センサの値が想定より
大きすぎて回転しなかった

15:30 作業終了



作業記録のポイント (p.163~)

- ・ **目標はできるだけ具体的に**

- ロボットの作成
→アームを動かして缶を倒す機構の作成

- ・ **目標とサブ目標を区別しよう**

- 目標：缶を倒す
- サブゴール ① 黒い部分を検知して方向転換する
② 壁を検知して方向転換する
③ アームを動かす など

- ・ **思いついたことはすぐに記録する**

- あとで書こうと思っていると、よく忘れます



作業後のリフレクション (p.164～)

- ・ ロボット作りの過程をプロセスチャートにまとめ、自分の活動全体を振り返って評価しましょう
- ・ プロセスチャートの項目
 - PLAN：作成目標、ロボットの予想図など
 - DO：PAD、ロボットの写真など
 - SEE：作成結果、失敗の原因など

 - 活動の流れを矢印で示しましょう
 - 項目別に色分けしてみましょう

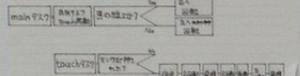


プロセスチャートの例

2回目

とりあえず完走をする

なるべく早く実際の走行に時間をかけたいためまずはスピードよりもとりあえず動くものを重点において、そこから改良を積み重ねて完走をすること一つの目標として目指していきたい。(A)



ジグザク走行でも11111のどにか早く完走を目指しプログラムをしていきたい。()



- 前進、後退、右回転、右回転が異なる動作して完走も目標として作成
- 車輪の数は4輪ではなく前に小さいのが1つ、後ろに普通の同じサイズのものが2つ(イキだけ軽量化するため)
- マシンの改造をしやすいようになるべくデザイン面を重視するよりもシンプルで短時間で簡単に仕様変更が可能なものにした 11/16

- 変更がなるとマシンが稼働した2輪で完走できた 11/30
- しかし...
- カーブ時に摩擦が少いためズバリと動くのが安定しない → 4輪で再スタート!!

新完成予想図



```

void mainLoop() {
  Serial.println("Start");
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
  // ...
}

```

```

void zigzag() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
  // ...
}

```

完走できた!! (46秒81)

- 目標通り完走ができました。
- 一歩引、次は黒→直進、白→ライン探しのプログラムを作成してタイムを短縮したいです11/11。
- 今のプログラムで問題になりそうな所は坂、セラチキ等がある。対策を考えたい。



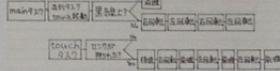
2輪から4輪への変更にもな。てギアの数を替えるためにすることが順番に始めていく。(A)



・ジグザク走行での速度を目標として作成
9.4秒の部分は数字以外とあわせて調整して5秒から値を調整した
前回の値を少しだけ大きく調整
・ライトセンサーの値を43に設定、マシンの設定にマスト
・マシンのライトセンサーの位置に合わせて黒線の読み取りに
・モーターの方から回転させるとよりスムーズに動くことになりました

・タッチセンサの反応を高倍率で認識できるマシンの構造上ベストな状態
前verの採用がで、時間の短縮に、その他の時間で...
・タイヤの数を前輪から前後2輪ずつの4輪にした
↓ 12/7
・ギアの数を左右でそれぞれ3つずつの計6個にした
↓
・車体が軽くなったため坂で引かかた

ジグザク走行、脱出



黒→直進、白→ライン探しといったプログラムを組みたい。(B)

```

void mainLoop() {
  Serial.println("Start");
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
  // ...
}

```

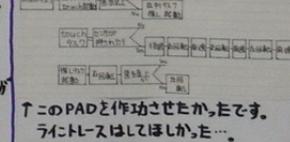
```

void zigzag() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
  // ...
}

```

・白→右回転→左回転→右回転→左回転とし角度を大きくしていくことに大きくして黒→直進のプログラム。
・その他のプログラムは前回のものと同じにした。又電池の値が少なくなってきたために半ヒートプログラムを変更した。並列9タスクは9.4秒以内ではなし。

(B)感想



↑このPADと作功させたから、Eです。ライントレスはしてほしかった...



走行中にコードが邪魔をしないように輪ゴムをとめるという初めは心配な時間があれば、大きい。プログラムの強化、デザイン面の改良などもできれば!(A)

白→右回転→左回転→右回転→左回転とし角度を大きくしていくことに大きくして黒→直進のプログラム。

・その他のプログラムは前回のものと同じにした。又電池の値が少なくなってきたために半ヒートプログラムを変更した。並列9タスクは9.4秒以内ではなし。

・やはりうまくいかず、白のラインを早く動かすには、時間がかかた

Team No. : A-13
Team name : アンカー

(Hardware)
Student ID :
Name(A) :
Name(B) :

(Software)
Student ID :
Name(B) :