Tutorial 3A

創造設計第二 TA : 三角修一

平成 27 年 10 月 19 日

1. はじめに

今回の Tutorial では，R/C サーボやモータの PWM 制御について, また状態遷移プログラムの書き方を学ぶことを目的とする．今回の Tutorial で理解してもらいたいのは以下の 4 点である．

1. マイコンによる PWM 出力
2. R/C サーボの PWM 制御
3. モータの PWM 制御
4. 状態遷移の概念とそのプログラミング方法

本資料の 2. 章では R/C サーボの説明，3. 章ではモータの説明を行い，4. 章では状態遷移という概念を紹介する．Tutorial での説明も本資料に沿って行われる．
今回の Tutorial の課題は 5. 章にある．2. 章から 3. 章の内容をよく読んだ上で取り組んでほしい．

今回の Tutorial で使用する道具

- マイコンボード (VS-WRC003) (ただし Tutorial1B でハード班が I /O 拡張を行ったもの) 2 個
- 磁気センサ，R/C サーボ，モータ，(外付モータドライバ)
- ブレッドボード，ジャンパ線，わにぐちクリップ一式
- 単 3 電池 4 本
- 単 3 電池が 4 本入る電池ボックス

> **注意** 今回の Tutorial では既製品のセンサを扱うことがあるが，各種センサをマイコンに接続する際には配線が誤っていないかを十分確認せよ．センサの誤接続はセンサ本体に負荷をかけることになり，最悪の場合，センサが使えなくなる．

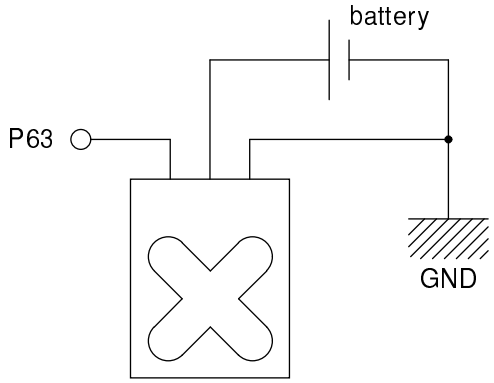
2. R/C サーボ

2.1 R/C サーボ 概要

創造設計第二では，マニピュレーション用にラジコン用サーボ・モータ（R/C サーボ・モータ）が用意されている．R/C サーボは，小型 DC モータとポテンショメータ（可変抵抗器），ギヤ，制御回路を小さなユニットにまとめたもので，入力信号のパルス幅に対応した角度を保つように内部でフィードバック制御を行なっている．

Fig. 2 のように制御パルスの周期は 14-20 [ms] で，パルス幅 (**Fig.** 2 中の t_w) が 1.52 [ms] のときにニュートラル (中立) 位置になり，1.52 ± 0.60 [ms] のときに，$\pm 60^\circ$ となる．なお R/C サーボのハードウェア的な中立位置とパルス幅を 1.52 [ms] 与えたときの位置は異なるので注意せよ．

サーボの内部では駆動軸にポテンショメータが取り付けられており，ポテンショメータの角度に対応した幅のパルスが発生するようになっている．入力されたパルス幅とサーボ内部のパルス幅が異なっている場合，R/C サーボ内部のモータが動作し，二つのパルス幅が等しくなるまで駆動軸が回転する．この様な方式にすることで，サーボの駆動軸角度が外力によって変化した場合でもサーボの駆動力の範囲内で元の角度に戻ろうとする力を発生させることができる．



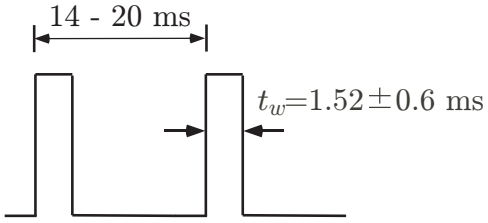


Fig. 2: R/C サーボ・モータの入力信号

Fig. 1: R/C サーボの接続例

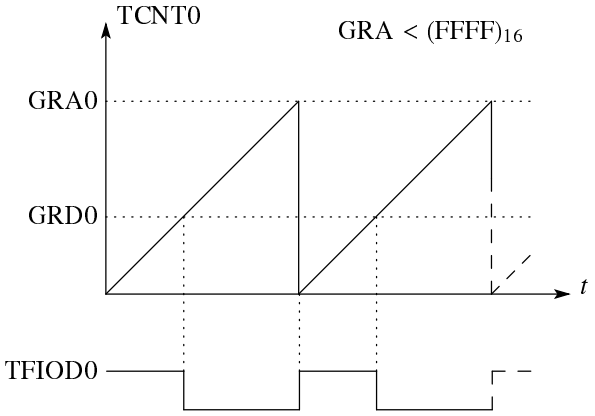


Fig. 3: タイマ Z0 の PWM モード

今回配布する R/C サーボ・モータの仕様は次の通りである．

- 双葉電子工業 (株) 製 S3003
- 寸法：40.4 × 19.8 × 36 [mm]
- 重量：37.2 [g]
- 動作電源：4.8-6.0 [V]
- 動作スピード：0.23 [sec/60°]（4.8 [V] 時），0.19 [sec/60°]（6.0 [V] 時）
- 出力トルク：3.2 [kg·cm]（4.8 [V] 時），4.1 [kg·cm]（6.0 [V] 時）
- 動作角度：± 60 [deg] 程度 [*1]

また，線の割り当ては，1 ピン (白)：制御信号，2 ピン (赤)：電源，3 ピン (黒)：GND となっている．

この R/C サーボは駆動電圧が 4.8[V] 以上であり，本マイコンの電源電圧では足りず，外部電源を用いないと駆動させることができない．R/C サーボ S3003 の接続例を **Fig.** 1 に示す．

2.2 マイコンを用いた R/C サーボの駆動方法

マイコンの PWM 出力モードを用いて R/C サーボを駆動させるには，PWM 信号の周期およびデューティ比をサーボの規定に従い適切に調節せねばならない．ここではタイマ Z0 の PWA モードを用いて PWM 出力を行う例を示す．ハードウェアマニュアルを読むと，タイマ Z0 の入出力端子のひとつである FTIOD0 が P63 につながっているので，P63 を出力端子とする．タイマ Z0 による PWM 出力の概要を **Fig.** 3 に示す．**Fig.** 3 の縦軸はレジスタ TCNT0 の値であり，時間にしたがってインクリメントされる．このレジスタ TCNT0 がインクリメントされる時間間隔はタイマコントロールレジスタ (TCR) の設定によって決定され，数種類の内部クロックから選択することもできるほか，外部クロックを利用することもできる．TCNT0 は GRA0 の値を超えると 0 にリセットされる．FTIOD0 の出力は，TCNT0 の値が GRA0 や GRD0 の値を超えるごとに

[*1] 仕様では ±60° となっているが，実際には約 ±90° 程度まで動作可能である．ただし，R/C サーボの回転角度が構造上の可動範囲を超えた場合，サーボ内部のギアが破損する可能性があるので注意せよ．

反転させられる．TCNT0 がインクリメントされる時間や GRA0，GRD0 の値を適切に設定することで，所望の PWM 波形を出力することができる．

タイマ Z0 を用いて R/C サーボを動作させる際の設定例を以下に示す．

```
set_imask_ccr(1);  // タイマの設定前にはこれを書く

// FTIOD0(P63) から PWM 出力させるためにタイマ Z0 を設定する
IO.PCR6 |= 0x08;       // P63 を出力ポートにする
TZ0.TCR.BIT.TPSC  = 2; // タイマのスケーラは phi/4
TZ0.TCR.BIT.CCLR  = 1; // GRA をカウンタクリア要因にする
TZ.TPMR.BIT.PWMD0 = 1; // FTIOD0 を PWM 出力に設定する
TZ.TOCR.BIT.TOD0  = 1; // FTIOD0 の初期出力を 1 にセット
TZ0.GRA = 0xFFFE;      // 周期は 17.7772...[ms]
TZ.TOER.BIT.ED0   = 0; // FTIOD0 からの出力を許可
TZ.TSTR.BIT.STR0  = 1; // タイマ Z0 を起動させる

set_imask_ccr(0);  // タイマを設定し終わったらこれを書く

// 以下，TZ0.GRD の値を適切に設定することで P63 から PWM 出力される
```

上の例では，タイマクロックとしてシステムクロック ϕ (= 14.7456 [MHz]) の 4 分の 1 を選択している．以下，タイマクロックを $\eta = \phi/4 = 3.6864 \times 10^6$ [Hz] とおく．このとき，GRA0 を $(\mathrm{FFFE})_{16}$ に定めれば，

$$\frac{(\mathrm{FFFE})_{16}}{\eta} \approx 17.77 \times 10^{-3} \text{ [s]}$$

であるから，PWM の周期が 17.77 [ms] となり R/C サーボの入力信号としての要請をみたす．GRD0 も同様の計算を行うことで決定する．

> **注意**
> R/C サーボに規格外の信号を入力すると R/C サーボの寿命が縮まってしまう．最悪の場合，モータが焼けて壊れてしまうので，PWM の設定は十分に確認をし，デバッグにも細心の注意を払うこと．

3. モータの速度制御

モータについても PWM 制御を行うことによりモータの速度制御を行うことができることを確認する．

3.1 モータの回転方向

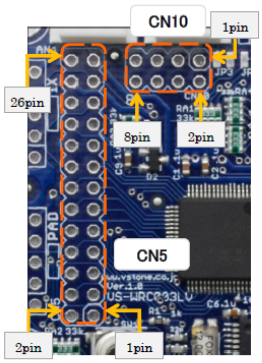
モータの PWM 制御の場合の設定例を見てもらう前に，PWM 出力とモータの回転方向の関係について確認する．

今年度ハード班が作成するモータドライバ回路には 2 つの PWM 波形を入れる仕様になっている．モータの回転方向は，外付モータドライバの端子 (IN1，IN2) に入れられるこの 2 つの PWM 波形の値の組み合わせにより決められる．今回の Tutorial では P66，P67 を用いて PWM 出力をしてモータを駆動する場合を考える．

IN1(P66)，IN2(P67) で PWM 出力をする場合

- 正回転
  IN1 : Low (0) , IN2 : High　⇒ 正転
- 逆回転
  IN1 : High　, IN2 : Low (0) ⇒ 逆転
- ブレーキ
  IN1 : High　, IN2 : High　⇒ ショートブレーキ
- フリー
  IN1 : Low (0) , IN2 : Low (0) ⇒ ストップ

ハード班で説明されている H ブリッジ回路において，ショートブレーキは短絡させてモータを GND のみと繋げている状態 (ブレーキ)，ストップはモータを何にも繋いでいない状態である．



○CN5 ピン配置

1 pin	:	PB6/AN6	14 pin	:	P52/WKP2
2 pin	:	PB5/AN5	15 pin	:	P51/WKP1
3 pin	:	PB4/AN4	16 pin	:	P50/WKP0
4 pin	:	P87	17 pin	:	P37
5 pin	:	P86	18 pin	:	P36
6 pin	:	P85	19 pin	:	P35
7 pin	:	P67/FTIOD1	20 pin	:	P34
8 pin	:	P66/FTIOC1	21 pin	:	RES
9 pin	:	P65/FTIOB1	22 pin	:	NMI
10 pin	:	P63/FTIOD0	23 pin	:	+5V
11 pin	:	P55/WKP5/ADTRG	24 pin	:	+Vbat
12 pin	:	P54/WKP4	25 pin	:	+3.3V
13 pin	:	P53/WKP3	26 pin	:	GND

Fig. 4: I/O ポートのピン配置

3.2 マイコンを用いたモータの駆動方法

マイコンの PWM 出力モードを用いてモータを駆動させるには，R/C サーボのときと同様に PWM 信号の周期およびデューティ比を適切に調節せねばならない．詳しくは R/C サーボの節を参照．
P66，P67 を介してタイマ Z1 を用いてモータを動作させる際の設定例を示す．

```
set_imask_ccr(1);  // タイマの設定前にはこれを書く

// FTIOC1(P66) から PWM 出力させるために IN1
IO.PCR6 |= 0x40;        // P66 を出力ポートにする
TZ.TPMR.BIT.PWMC1 = 1; // FTIOC1 を PWM 出力に設定
TZ1.GRC = 0x0000;       // FTIOC1 のデューティ0
TZ.TOER.BIT.EC1   = 0; // FTIOC1 からの出力を許可
TZ.TSTR.BIT.STR1  = 1; // タイマ Z1 を起動させる

// FTIOD1(P67) から PWM 出力させるために IN2
IO.PCR6 |= 0x80;        // P67 を出力ポートにする
TZ.TPMR.BIT.PWMD1 = 1; // FTIOD1 を PWM 出力に設定
TZ1.GRD = 0x0000;       // FTIOD1 のデューティ0
TZ.TOER.BIT.ED1   = 0; // FTIOD1 からの出力を許可
TZ.TSTR.BIT.STR1  = 1; // タイマ Z1 を起動させる

set_imask_ccr(0);  // タイマを設定し終わったらこれを書く

// 以下，TZ1.GRC,TZ1.GRD の値を適切に設定することで P66,P67 から PWM 出力される
// 2 つの PWM 出力の組み合わせによってモータを正転・逆転・ショートブレーキ・ストップできる
//  例 : 正転
//  TZ1.GRC = 0x0000; // 正転，IN1(P66):0 , IN2(P67):PWM
//  TZ1.GRD = 0x4ABC;
```

4. 状態という概念によるプログラミング

複雑なプログラムを作成するひとつの方法として，プログラムに「状態」を導入するという手法がある．これは，プログラムに状態を与え，内部あるいは外部のイベントによって状態が遷移していくことによりプログラムの挙動を表現しようというものである．その状態の遷移とイベントとの関係を表したものが状態遷移図であり（**Fig.** 5 参照），状態遷移の様子を容易に把握することができる．

今回の Tutorial 程度の単純なプログラムならともかく，本番でロボットを動かすくらいの複雑さになったときには，必ず状態遷移図やプログラム全体のアクティビティ図（流れ図，**Fig.** 7 参照）を作成するべきである．これらの図は自分の理解を深めるだけでなく，プログラムの構造を他者と共有するのにも役立つため，積極的に活用してほしい．アクティビティ図でよく用いられる部品を **Fig.** 6 に掲載してあるので参考にしてもらいたい．

また，常に読みやすいソースコードを書くよう心がけること．例えば，`switch` 文を 2 つに分け，1 つ目でそのモードでの動作を記述，2 つ目で状態遷移条件や遷移する瞬間に一度だけ行いたいことを記述するなどといった工夫をすることで，ソースコードは読みやすくなり，同時に保守性が増すであろう．

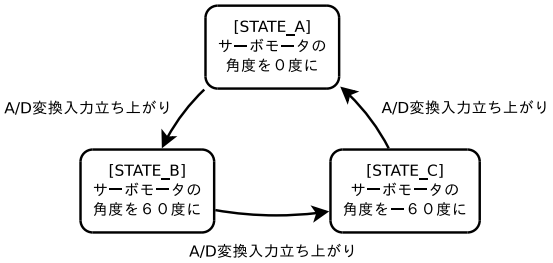


Fig. 5: サンプルプログラムの状態遷移図

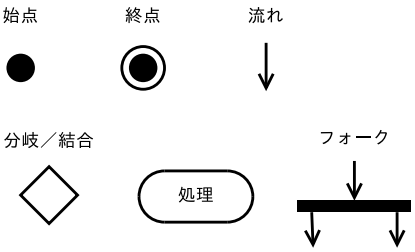


Fig. 6: アクティビティ図で用いられる主な部品

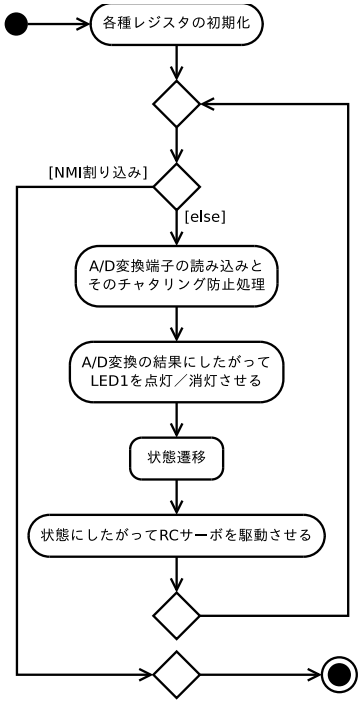


Fig. 7: サンプルプログラムのアクティビティ図

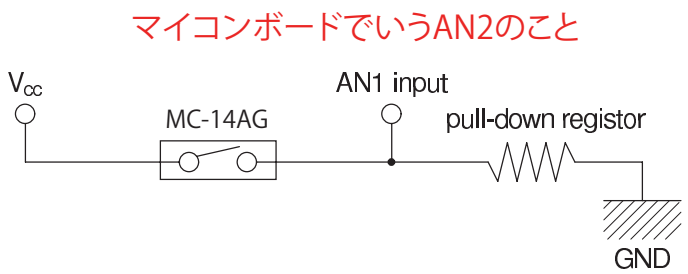


Fig. 8: 磁気センサ接続例

5. 本 Tutorial の課題

講義のホームページから Tutorial3A.zip をダウンロードせよ．展開されたディレクトリ内にある SS2tutorial3Aservo.hws, SS2tutorial3Amotor.hws が本課題で用いるプログラムの HEW ワークスペースである．HEW ワークスペースの開き方および編集の仕方については Tutorial1A の資料を参照してもらいたい．課題をこなすにあたっては，フォルダ内にある H8/36064 ハードウェアマニュアルも適宜参照せよ．

5.1 課題（磁気センサ & R/C サーボ）

課題 1 ： `SS2tutorial3Aservo` は A/D 変換の入力端子に接続させた磁気センサが反応したら（入力電圧がしきい値を超えたら）状態が遷移し，サーボの角度を変化させるプログラムであるが，まだ未完成である．**Fig.** 5 の状態遷移図のように振舞うプログラムになるよう，`SS2tutorial3Aservo/SS2tutorial3Aservo.c` を修正せよ．修正が完了したら TA のチェックを受け，実際に R/C サーボと磁気センサを **Fig.** 1, **Fig.** 8 (図では AN1 と記載されているが，マイコンボードには AN2 と記載されていることに注意）のように接続して動作を確認せよ．磁気センサの繋ぎ方については Tutorial2A の資料を参照してもらいたい．
注意 : R/C サーボの動きが可動範囲を越えてしまったらすぐに電源 (マイコン・R/C サーボ) を切ること．

> `SS2tutorial3Aservo.c` の主な修正点は，78 行目からのポートとタイマ Z0 の設定と，140 行目にある A/D 変換のしきい値設定と，203 行目からの `switch` 文におけるそれぞれの状態の中で `TZ0.GRD` を適切に設定することである．

注意 プログラムを走らせる前に `TZ0.GRD` の値の妥当性を必ず TA に確認すること．

課題 2 ：R/C サーボと磁気センサはつないだままで，今度は STATE B から STATE C の状態遷移を時間によって行うようにプログラムを書き換えよ．STATE B から STATE C へは 3[s] で切り替えることにする．修正が完了したら TA のチェックを受け，実際に動作させる．Debug でコンパイルできないときは Release でコンパイルして ROM 領域に書き込んで動作させる．ROM 領域への書き込み方法については Tutorial1A の資料を参照してもらいたい．

> ヒント : 時間切り替えのために timer という変数を用意した．timer は STATE A，STATE C においてもカウントされることに注意．その他必要な変数があれば適宜定義すること．

5.2 課題 (モータ)

課題 3 ：SS2tutorial3Amotor は 3[s] ごとにモータが動きを変えるプログラムであるが，まだ未完成である．ストップ→正転→ストップ→逆転→ストップ→正転→ ··· のように振舞うプログラムになるよう，SS2tutorial3Amotor/SS2tutorial3Amotor.c を修正せよ．モータの速度については最高速度の 3 割程度の速度を出すように修正すること．修正が完了したら TA のチェックを受け，その後で実際にモータを動かしてみる．Debug でコンパイルできないときは Release でコンパイルして ROM 領域に書き込んで動作させる．モータドライバはハード班が作成したものを使用する．モータドライバの配線等の作業はハード班と協力して行うこと．

SS2tutorial3Amotor.c の主な修正点は，59 行目からのポートとタイマ Z1 の設定と，92 行目からの状態遷移の条件の記述と，117 行目からの switch 文におけるそれぞれの状態の中で TZ1.GRC,TZ1.GRD を適切に設定することである．また 186 行目からの g3sflag を立てる条件を記述．状態遷移が 3[s] で起こるように設定すること．

注意 :Debug ビルドが成功した場合でも，Release ビルドでコンパイルしたプログラムをマイコンの ROM 領域に書き込んで動作を確認すること．