

Tutorial 3

創造設計第二 TA：青田諒

平成 23 年度 10 月 20 日

1. はじめに

今回の Tutorial では、既製品のセンサーやモジュールの使い方と、状態遷移プログラムの書き方を学ぶことを目的とする。本日使用するものは 200mm 光センサ、磁気センサ、PSD センサ、R/C サーボである。本 Tutorial で理解してもらいたいのは以下の 4 点である。

1. 外部トリガによる割り込み処理
2. A/D 変換によるアナログ信号の計測
3. マイコンによる PWM 出力と R/C サーボの扱い方
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. 章から 5. 章では本 Tutorial で用いるセンサなどの説明を行い、6. 章では状態遷移という概念を紹介する。講義中の説明も本資料に沿って行われる。本 Tutorial の課題は 7 章にある。2 章から 6 章の内容をよく読んで上で取り組んでほしい。

本 Tutorial で使用する道具

- マイコンボード (VS-WRC003)
- 200mm 光センサ、磁気センサ、PSD センサ、R/C サーボ
- ブレッドボード、ジャンパ線、わにぐちクリップ式
- 単 3 電池 4 本
- 単 3 電池が 4 本入る電池ボックス



注意

本 Tutorial では既製品のセンサを扱うことが多いが、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. 反射型距離センサ (200mm 光センサ)

2.1 フォトトランジスタ

フォトトランジスタは光によって駆動するトランジスタであり、近接センサやロータリエンコーダのパターン読み取りなど、光センサとしてさまざまな応用が考えられる。

Fig. 1 のエミッタ出力の回路では入射光と同位相の出力が得られ、Fig. 2 のコレクタ出力では入射光と逆位相の出力が得られる。また、LED と組み合わせることによってフォトカプラを構成することもできる。Fig. 3 の回路では LED の光がフォトトランジスタに入射している場合は 5[V]、入射していない場合は 0[V] の出力となる。これを利用すれば LED とフォトトランジスタの間に障害物が入っていないかを検出する光センサとすることが可能である。また、入力信号により LED を駆動し、フォトトランジスタ側に外部回路を接続することによって、2 つの回路を電氣的に絶縁した状態で駆動することも可能である。

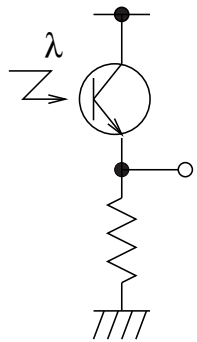


Fig. 1: エミッタ出力

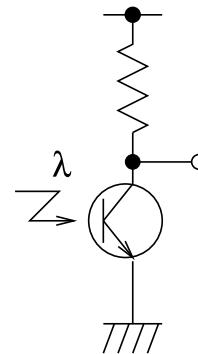


Fig. 2: コレクタ出力

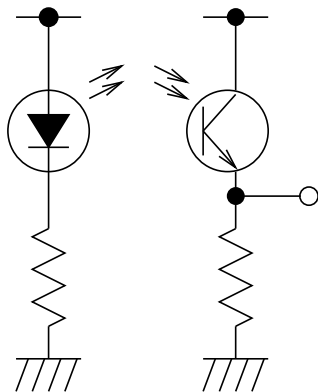


Fig. 3: フォトコプラ回路

2. 基本仕様

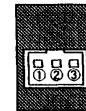
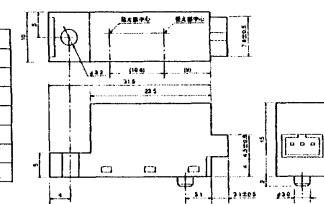
電気的特性

動作温度	-20~+60°C
電源電圧	DC5V±5%
消費電流	11mA(非投光時) 200mA(投光時)
応答速度	1ms
許容照度	外乱光 2000 lx
出力1/F	オープンコレクタ方式
出力信号	物体検知時:ON=L VOL=0.4V以下 物体非検知:OFF=H VOH=4.5V以上

ピン接続

PIN No.	名称	説明
1	VCC	電源DC5V端子
2	VOUT	センサの反応出力端子
3	GND	電源グランド端子

外形寸法



なお、推奨コネクタはIL-S-3S-S2C2-S(日本航空電子製)です。

Fig. 4: 200mm センサ基本仕様 (データシートより)

2.2 200mm 光センサ (ベストテクノロジー - 社製「BTE019」)

創造設計第2では近接センサ(物体の接近を検出する)用に「200mm 光センサ」が用意されている。これは、赤外線LEDとフォトトランジスタがワンパッケージで組み込まれたもので、機能としては、センサから最大200mm離れた白いもの(光を反射するもの)を検出する。コネクタ端子は3つあって、電源とGNDを1,3ピン間につなぎ、出力を2ピンから取る。BTE019の基本仕様をFig.4に示す。

Fig.4を見ればわかるが、BTE019の駆動電圧は約5[V]であり、本マイコンの電源電圧(3.3[V])ではBTE019は駆動しない。したがって、BTE019を駆動するにあたっては外部から電源を供給する必要がある。本Tutorialでは外部電源として充電式ニッケル水素電池4本(1.2[V]×4=4.8[V])を用いる。例えばFig.5,6のように外部電源を用いることで200mm光センサを駆動させることができる。ただしBTE019はオープンコレクタ方式を採用しているため、プルアップ抵抗がないと出力が安定しない^{*1}。また、参考までに本マイコンのIOポートのピン配置をFig.7に示す。



注意

当然のことながら、電源とGNDや信号線を間違えるとセンサは破壊する。これを防ぐ方法は以下の手順を踏み、一つ一つのステップを確実に行うことである。まず、センサをつなぐ前に電源回路や外部回路を構成する。そして、センサをつなぐ前に、コネクタのセンサ側入力の電圧を確認する。こうすることで、センサの破壊はほとんど防ぐことができる。



警告

今回の試作設計で扱うモジュールは、高価なものである。これらの取り扱いには十分注意すること。壊してしまった場合には、ロボット製作において使用できなくなる可能性があるので十分注意しながら作業を行うこと。

*1本Tutorialでは外部回路によりプルアップを施さないで実現しているが、マイコンのポートプルアップコントロールレジスタ(PUCR5)およびRES端子を適当に設定することにより、マイコン側で統一的に各IOポートをプルアップすることができる。詳細はH8/36064ハードウェアマニュアルを参照せよ。

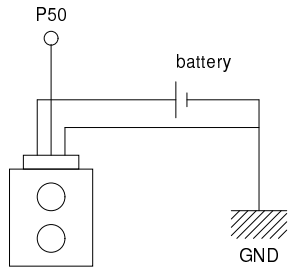


Fig. 5: 200mm センサの接続例 (プルアップなし)

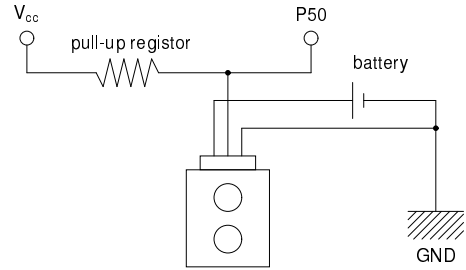
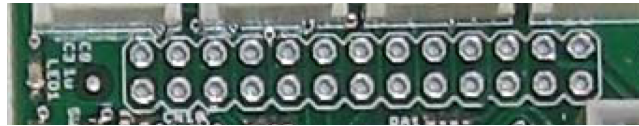


Fig. 6: 200mm センサの接続例 (プルアップあり)

PB6, PB4, P71, P74, P66, P62, P56, P54, P52, P50, NMI, Vbat, GND



PB7, PB5, P70, P72, P67, P63, P57, P55, P53, P51, RES, --, +3.3V

Fig. 7: VS-WRC003 の IO ピン配置

2.3 200mm 光センサを外部トリガとした割り込み処理

ここでは「200mm 光センサが反応したら割り込み処理を引き起こす」という動作をマイコンに組み込んでみる。Fig. 6 のようにマイコンとセンサを接続した場合、マイコンの P50 ポートは、200mm 光センサが反応していないときは 3.3[V]、反応しているときは 0[V] になっている。したがって、P50 を入力ポートに設定し、P50 の電圧が下がったときに割り込み処理が入るようにすればよい。具体的な設定は以下のようになる。

```

/** P50 の外部トリガで割り込みを発生させる */
IENR1.BIT.IENWP = 1; /* WKP 割り込みを利用可能にする */
IO.PMR5.BIT.WKP0 = 1; /* P50 を WKP0 割り込み入力ポートに */
IEGR2.BIT.WPEG0 = 0; /* 立ち下がりを検出 */
IWPR.BIT.IWPF0 = 0; /* 割り込みフラグレジスタ */

.....

/* * * * * *
 * INT_WKP : P50 の立ち下がりを検出して割り込みをかける関数
 * */
__interrupt(vect=18) void INT_WKP(void)
{
    /* 次の割り込みのためにフラグを下げておく */
    IWPR.BIT.IWPF0 = 0;

    /* 以下、割り込み関数の内容を記述する */
}

```

3. PSD センサ

創造設計第二では A 類部品として PSD センサ (GP2D12) が配布されている。PSD センサを利用することで対象との距離を計測することができる。

3.1 PSD センサの概要

PSD はセンサ自体が赤外線を発生し、反射光の戻って来る位置を測定することで距離情報を計測する測距センサである。PSD は反射光の強度ではなく位置を測定に用いるため、反射物の色、反射率の影響を受けにくいという特徴がある。また、物体の接近だけでなく、物体への距離を数値的に計測することができる。GP2D12 のピン配置は、Fig. 9 のようになっている。

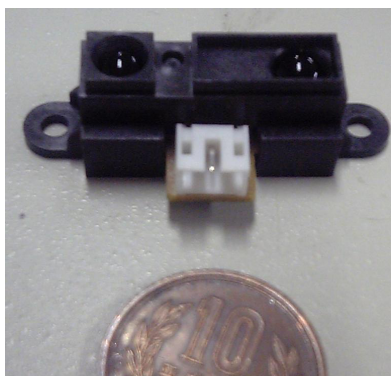
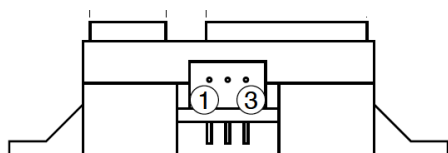


Fig. 8: GP2D12 (PSD センサ)



Terminal connection

- ① V_O (Signal)
- ② GND
- ③ V_{CC} (5V)

Fig. 9: GP2D12(PSD センサ) のコネクタ

3.2 PSD の出力と A/D 変換

PSD センサからの出力はアナログ出力である．今回配布する PSD センサは距離に対応した電圧を出力する．この電圧を A/D 変換してマイコンボードに取り込む．Fig. 10 にデータシートを示す．

■ 出力電圧と距離特性(GP2D12)

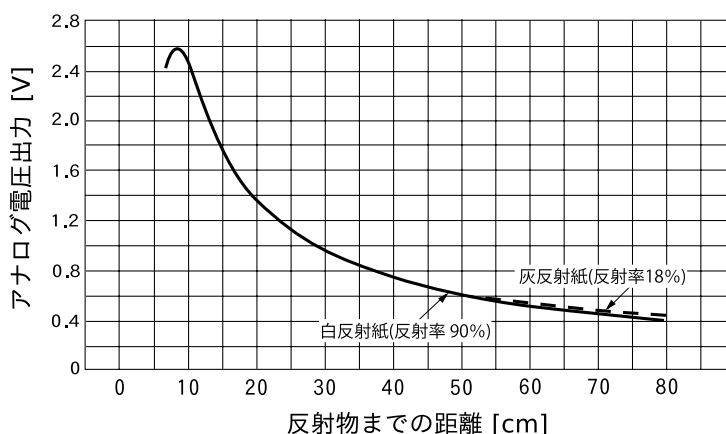


Fig. 10: データシートより

注意

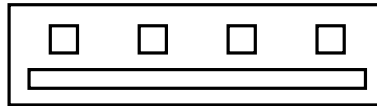
PSD センサは個々で特性線図が異なるため、精密な測定をする場合は実験をして個々の特性線図を作成する必要がある。

3.3 A/D 変換

本マイコンボードには 10 ビット分解能の A/D 変換器が搭載されており、合計 8 チャンネルの入力が可能となっている．A/D 変換の動作モードには単一モードとスキャンモードの 2 種類があり、レジスタを設定することによって動作モードを切り替えることができる．単一モードは特定のレジスタが 1 になったとき A/D 変換を行い待機状態に戻るのに対し、スキャンモードは連続的に A/D 変換を行う．スキャンモードは大変便利であるが、消費電力が大きくなってしまいうので注意が必要である．

以下に、AN1^{*2}を用いてスキャンモードによる A/D 変換の設定例を示す．ただし、マイコンボードの A/D 変換用のピンの配置については Fig. 11 を参照されたい．

^{*2}ハードウェアマニュアルで言うところの AN1 であり、マイコンボードには AN2 と記載されているので注意されたい．



+3.3V, 100Ω-GND, GND, signal (+3.3V pull-up)

Fig. 11: A/D 変換用のピン配置

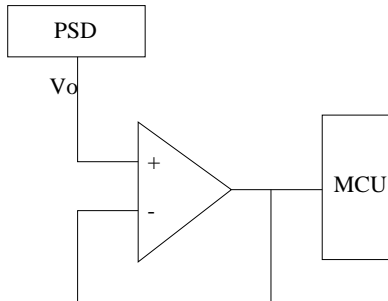


Fig. 12: バッファ

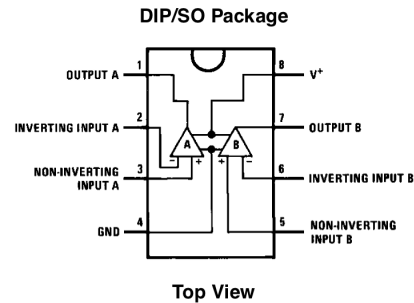


Fig. 13: LM358 ピン配置

```
float adVoltage; /* AD 変換の結果を格納する */

/** AN1 で AD 変換させる **/
AD.ADCSR.BIT.CH = 1; /* チャンネルを 0b001 に設定する */
AD.ADCSR.BIT.ADST = 0; /* クロックセレクト変更のため */
AD.ADCSR.BIT.CKS = 0; /* クロックセレクト変更. この変更は ADST = 0 のもとで行う */
AD.ADCSR.BIT.SCAN = 1; /* SCAN モードにする */
AD.ADCSR.BIT.ADIE = 0; /* 割り込みは入れない */
AD.ADCSR.BIT.ADST = 1; /* AD 変換をスタートさせる */

.....

while (1) {
    adVoltage = 3.3 * (AD.ADDRB >> 6) / 1024.0;
}
```

最後の 1 行について説明を加えておこう．ここで用いた A/D 変換の分解能は 10bit であり，0[V] から 3.3[V] までの値を 10bit に離散化した値が AD.ADDRB の上位 10bit に格納されるようになっている．すなわち，AD 変換の入力に加えられた電圧を $V \in \mathbb{R}$ ，AD.ADDRB の上位 10bit が表す値を $k \in \mathbb{Z}^+ \cup \{0\}$ とすると，

$$k = \left\lfloor \frac{2^{10} \cdot V}{3.3} + \frac{1}{2} \right\rfloor$$

という関係をもつ．ただし 1/2 を加えた後の floor は四捨五入を意味している．上式を書き換えると

$$\frac{3.3}{2^{10}} \left(k - \frac{1}{2} \right) \leq V < \frac{3.3}{2^{10}} \left(k + \frac{1}{2} \right)$$

となるが，今は便宜的に $V \approx 3.3k/2^{10}$ としてしまうことにする．また，16bit 変数である AD.ADDRB の上位 10bit を得るためには AD.ADDRB を 6 回右シフトすればよく，さらに $2^{10} = 1024$ であるから，これを C 言語で記述すると最後の 1 行のようになる．

3.4 回路例

以下に回路の例を示す．必ずしもこの通りにしなければならないわけではない．Fig. 12 に出力と MCU ボードの端子台の間に Op-Amp によるバッファをはさんだ回路の例を示す．必ずしもこの通りにしなければならないわけではないことに注意する．また，Op-Amp は LM358 (ピン配置は Fig. 13) を用いる．

■定格

項目	規格
接点構成	1a接点
定格負荷	DC12V 10mA (抵抗負荷)
接点電圧の最大値	DC100V
接点電流の最大値	DC0.5A
開閉容量の最大値	DC10W
使用温度範囲	0~55℃ (ただし、氷結・結露のないこと)
使用湿度範囲	35~85%RH

■内部接続図

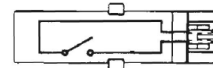


Fig. 14: W2DG-E01 の定格

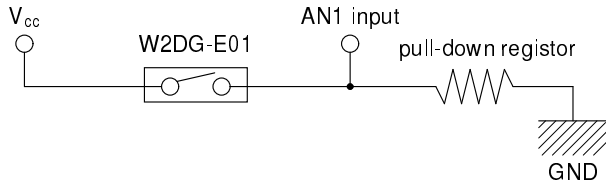


Fig. 15: 磁気センサ接続例

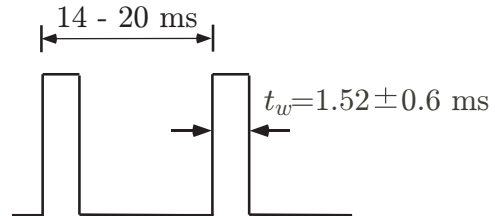


Fig. 16: R/C サーボ・モータの入力信号

4. 磁気センサ

マグネットなどの磁気を検出するために、本 Tutorial では磁気センサ W2DG-E01 を用いる。W2DG-E01 の定格および内部接続図は Fig. 14 の通りである。内部接続図をみると、この磁気センサはただのスイッチのように見えるが、出力をみると磁気の強さによって連続的に出力値が変化しているため、ここでは A/D 変換を用いて磁気センサの出力を監視することにする。A/D 変換については前節を参照されたい。

4.1 接続例

また、ポートの出力を安定化するためにプルアップあるいはプルダウンを必要とする。Fig. 15 はプルダウンを施した場合の接続例である。A/D 変換を利用するためのマイコン側の設定は PSD センサの場合とまったく同様であるので、前節を参照されたい。

5. R/C サーボ

5.1 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 モータとポテンシオメータ (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っている。

Fig. 16 のように制御パルスの周期は 14-20 [ms] で、パルス幅 (Fig. 16 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オメータが取り付けられており、ポテンシオ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。このよう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。

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*3

*3仕様では $\pm 60^\circ$ となっているが、実際には約 $\pm 90^\circ$ 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、サーボ内部のギヤが破損する可能性があるため注意せよ。

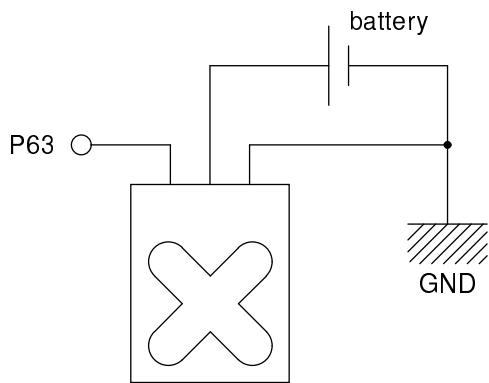


Fig. 17: R/C サーボの接続例

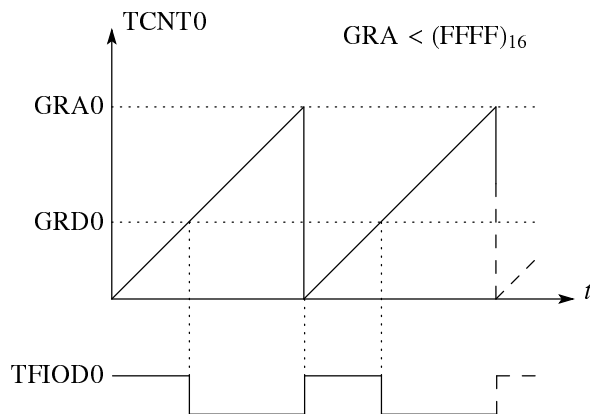


Fig. 18: タイマ Z0 の PWM モード

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 サーボも駆動電圧が 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. 17 に示す。

5.2 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 につながっており、これが一番クリップでつまみやすそうなので、これを出力端子とする。タイマ Z0 による PWM 出力の概要を Fig. 18 に示す。Fig. 18 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。TFIOD0 の出力は、TCNT0 の値が GRA0 や GRD0 の値を超えるごとに反転させられる。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); /* タイマの設定の前にはこれを書く */

/** 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; /* タイマのスケールは phi/4 */
TZ0.TCR.BIT.CCLR = 1; /* GRA をカウンタクリア要因にする */
TZ0.TPMR.BIT.PWMD0 = 1; /* FTIOD0 を PWM 出力に設定する */
TZ0.TOCR.BIT.TOD0 = 1; /* 初期値は High */
TZ0.POCR.BIT.POLD = 0; /* FTIOD0 を Active <--> Low にする */
TZ0.GRA = 0xFFFE; /* 周期は 17.772...[ms] */
TZ0.TOER.BIT.ED0 = 0; /* FTIOD0 からの出力を許可 */
TZ0.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 $(FFFE)_{16}$ に定めれば、

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

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。
---	---

6. 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. 19 参照)、状態遷移の様子を容易に把握することができる。

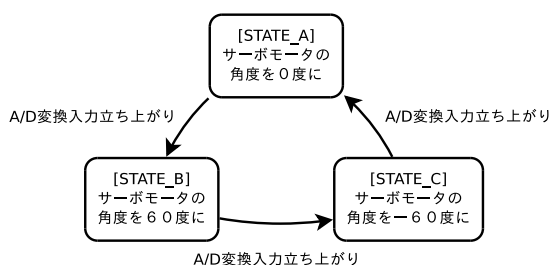


Fig. 19: サンプルプログラムの状態遷移図

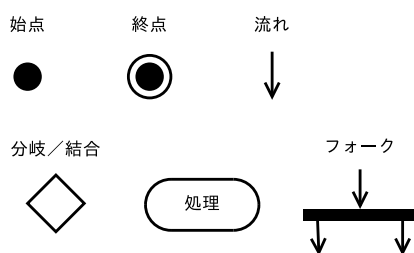


Fig. 20: アクティビティ図で用いられる主な部品

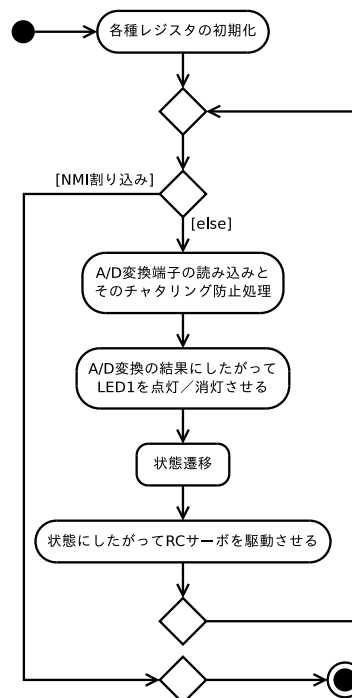


Fig. 21: サンプルプログラムのアクティビティ図

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. 21 参照) を作成すべきである。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. 20 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。

7. 本 Tutorial の課題

講義のホームページから ss2shiasku03.zip をダウンロードし、Z:ドライブに展開してほしい。展開されたディレクトリ内にある SS2Shisaku.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 Tutorial 1 の資料を参照されたい。課題をこなすにあたっては、T:ドライブ内にある H8/36064 ハードウェアマニュアルも適宜参照せよ。

7.1 課題 (200mm 光センサ)

課題 3-1 : Fig. 5 にしたがって回路を構成し、プログラムを実行せよ。プルアップを施していないため、センサ無反応時の出力が乱れるはずである。必要があれば適宜ブレッドボードを活用せよ。

課題 3-2 : 課題 3-1 で作成した回路にプルアップを施し、出力が安定することを確認せよ。

課題 3-3 [OPTIONAL] : 携帯電話に搭載されたカメラなどは赤外線を捕らえることができる。200mm 光センサのどこから赤外線が出ているかを確認し、センサの反応しやすい場所を探せ。

課題 3-4 [OPTIONAL] : 200mm 光センサに反応させる物体の色や角度などによってセンサの感度がどのように変化するかを調べよ。

注意 課題 1 の段階では、R/C サーボとマイコンを接続してはならない。

7.2 課題 (磁気センサ & R/C サーボ)

課題 3-5 以降の課題に取り組む前に、SS2Shisaku02/main.c の 130-133 行目をコメントアウトし、170-173 行目、176-179 行目のコメントを外してもらいたい。さすれば、磁気センサに反応するたびに LDE1 が点滅するようになり、以下の課題でデバッグが容易になる。

課題 3-5 : SS2Shisaku02 は A/D 変換の入力端子に接続させた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. 19 の状態遷移図のように振舞うプログラムになるよう、SS2Shisaku02/main.c を修正せよ。修正が完了したら TA のチェックを受け、実際に磁気センサと R/C サーボを Fig. 15, 17 のように接続し動作を確認せよ。

main.c の主な修正点は、142 行目にある A/D 変換のしきい値設定と、209-213 行目にある 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。A/D 変換のしきい値設定は 2.5[V] 程度になるように設定するのがよいだろう。

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。

7.3 課題 (PSD センサ & R/C サーボ)

課題 3-6 : 課題 3.2 で作成した回路の磁気センサを PSD センサに置き換えた回路を作成し、動作を確認せよ。必要があれば A/D 変換の閾値を設定し直してもよい。

課題 3-7 [OPTIONAL] : AN0 (マイコンボードには AN1 と書いてある) に磁気センサを、AN1 (マイコンボードには AN2 と書いてある) に PSD センサを接続し、Fig. 22 のように振舞うプログラムを作成せよ。状態遷移のトリガとなるセンサが毎回入れ替わるため、チャタリング防止には特に注意を払う必要はないと思われる。

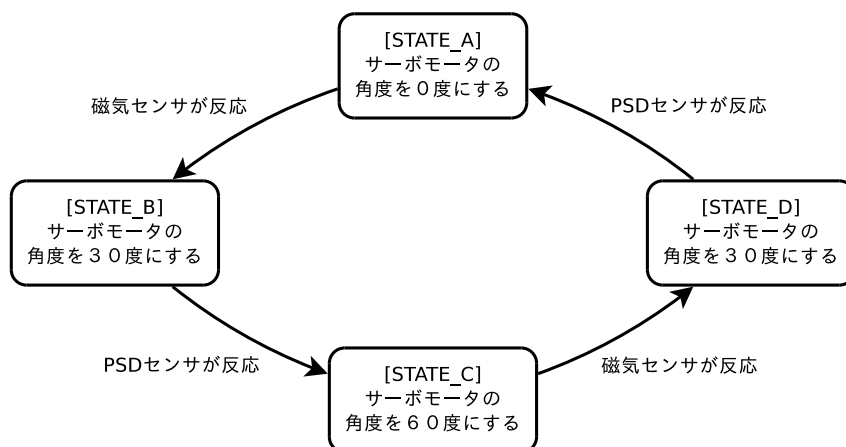


Fig. 22: 課題 3.3.2 の状態遷移図