試作検討 2A

試作検討 2A：モータドライブ回路 (ソフト編)

創造設計第二 TA：須永 智

平成 23 年 10 月 17 日

1. 本日の目的

本日の試作検討では，モータドライバ (別資料ではモータアンプと記されているところもある) 回路の役割を理解し，PWM 制御により DC モータを回転させる．また，Vstone 社製マイコンボード (VS-WRC003) には DC モータドライバ回路 (TB6552) が内蔵されているが，実際の使用においてモータ駆動時の大きな負荷に十分耐えられないことが例年の実習において明らかになっている．そのため，外付けのモータドライバ回路を作成し，マイコンボードと接続して同じく PWM 制御により DC モータを駆動させる方法を紹介する．

2. PWM(Pulse Width Modulation) 制御

マイコンボードからの出力は，0(Low) か 1(High) の 2 値である．2 値の出力により DC モータを制御しようとした場合，回転させる (1) と停止させる (0) の 2 つの状態しか制御することができず，モータの回転数を 5 割で回転させるという制御はできない．そのため，PWM(Pulse Width Modulation：パルス幅変調) を利用する．一般に，PWM では周期を一定にして，パルス幅を変更する．出力 1 が DC モータ回転に，出力 0 が DC モータ停止に対応するとすれば，1 を出力する時間が長いほどモータの出力・回転数が大きくなることが直感的にも理解できるだろう．

3. DC モータドライバ回路 (H ブリッジ回路)

マイコンのポートからの出力電流はモータから力・回転数を得るために流れる電流に比べ一般に非常に小さいため，通常はドライバ回路を用いてマイコンの出力を増幅し，DC モータや各種アクチュエータに接続する．また，単純にマイコンの出力を増幅させただけでは，DC モータを一方向へ回転させることしかできない．そこで，正転と逆転ができるようにすることもドライバ回路の役割となる．

DC モータのドライバ回路は Fig. 1 に示すような H ブリッジと呼ばれる回路がよく用いられる．電気的に切り替えなスイッチの ON/OFF を組み合わせることで，(a) 正転，(b) 逆転，(c) ショートブレーキ，および (d) ストップを実現できる．

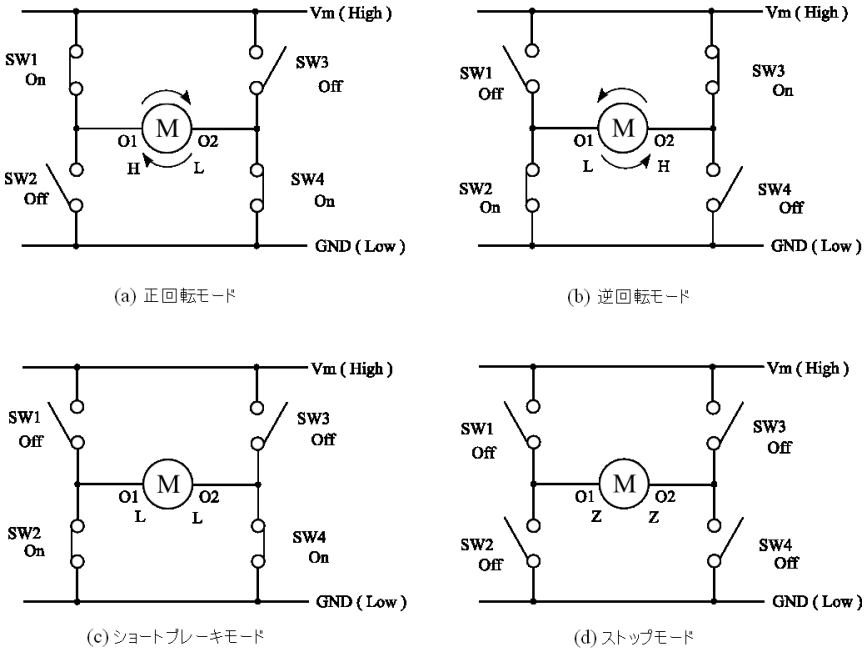

Fig. 1: H ブリッジ回路の基本モード (Z はハイインピーダンスを表す)

4. 外付 DC モータドライバ回路を利用した PWM 制御

先述のように，Vstone 社製のマイコンボード (VS-WRC003) には DC モータドライバ回路 (TB6552) が内蔵されているが，この 2 年間の創造設計第二の実習により，DC モータを回転させたときにこの TB6552 に大きな負荷をかけると，TB6552 が破損し動作しなくなる恐れがあることが分かっている．そこで，今回の試作検討では，この TB6552 の代わりとして VS-WRC003 に外付けできる Vstone 社製のモータドライバ (VS-WRC005) を紹介・作成し，PWM 制御により DC モータの回転を制御する．なお，VS-WRC005 の詳しい製作方法等は試作検討 2B の資料に記載されているので，そちらを参考にしてほしい．

5. サンプルプログラムに関して

本節では，本試作検討でのサンプルプログラムと，外付のモータドライバを用いた DC モータを回転させるためのプログラム内の関数に関して解説する．

まずは，創造設計第二の web ページから試作検討第 2 回分のサンプルプログラム SS2shisaku02 をダウンロードする．そのフォルダの中に SS2shisaku02.hws というサンプルプロジェクトを行う．

サンプルプログラム SS2shisaku02.c では，タイマ B1 を用いて，3 秒ごとに状態 A，B，C を遷移させ，DC モータの回転方向および速さを変化させるように記述されている．サンプルプログラムは一部プログラムが欠けている部分がある．これは，後で述べる課題で行うプログラムラインを補ってもらう行であることを意味している．サンプルプログラムを見てもらうと分かるかもしれないが，まず Switch 文を使い状態だけを遷移している．そして，その後，再び Switch 文によりそれぞれの状態における外付けモータなどのアクチュエータの出力信号を設定している．このように，状態遷移とそれぞれの状態における処理を分けることにより，多状態となった場合のプログラムの煩雑さが回避しやすくなる．これは，実際の試技におけるプログラムの多状態を考えてもらうと分かりやすいだろう．

また，ロボットを動かすときなどに用いるプログラムは while 文ループを永遠に回し続ける処理をとる．このとき，While 文ループ内に，外からの信号を取り込む短い時間を設ければマイコンに任意の入力を入れることができると考えがちである．しかし，実際にやってもらえば理解してもらえると思うが，センサ等の連続信号ならばともかく，人の手によるプッシュスイッチの入力などではスイッチを押したとしても設定した取り込み時間に押せるということはほとんどなく，こちらの思うように動作をしてくれないことが多々である．そこで，通常は「ボタンが押された」,「値が閾値より大きくなった」等の変化をプログラム実行中に「割り込む」という割り込み機能を使う．本試作検討においても，intprg.c の下のほうに，_ _ interrupt 関数として記述されている．本プログラムでは，ステートマシンの状態遷移に 3 秒間のタイマ割り込みを用いている．5ms×600=3s を生成している．

5.1 状態という概念によるプログラミング (ステートマシーン)

複雑なプログラムを作成する 1 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実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. 2 参照)，状態遷移の様子を容易に把握することができる．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．今回の試作検討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. 4 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. 3 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 つめ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．

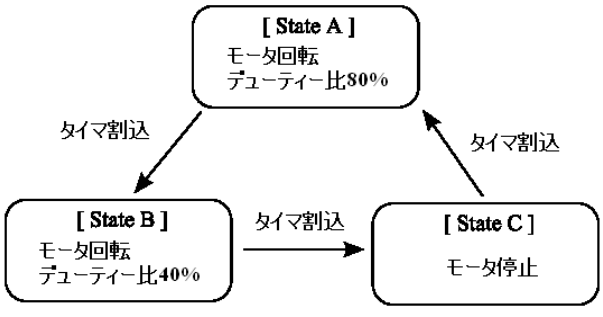

Fig. 2: サンプルプログラムの状態遷移図の例

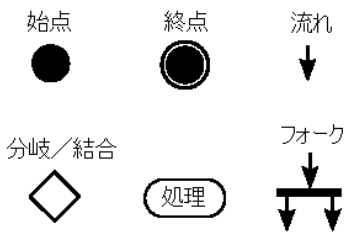

Fig. 3: アクティビティ図で用いられる主な部品の例

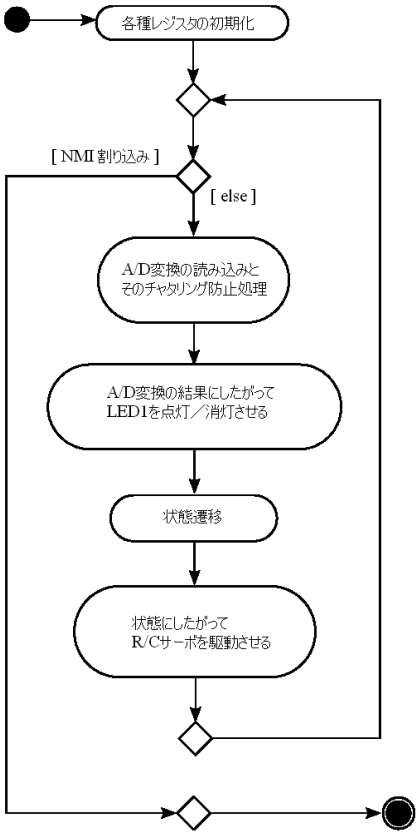

Fig. 4: サンプルプログラムのアクティビティ図の例

5.2 DC モータ関連関数

DC モータに関連する関数を簡単に述べる．プログラム上では，motor.c ファイルにある．以下の関数は，本資料で紹介した Vstone 社製の外付 DC モータドライバ回路に対応している．すなわち，ポートとは，P61(FTIOB0)が外付モータドライバへの出力である．

MotorInit()
機能 : DC モータに関する初期化処理を行います．DC モータを使用する際は，プログラムの先頭で必ず実行してください．
引数 : port: ポートの選択 (0:P61(FTIOB0))
戻り値: なし

Motor()
機能 : DC モータの速度を設定します．
引数 : port: ポートの選択 (0:P61(FTIOB0))
dir: 方向 (P32 が 1:前進，P33 が 1:後退)
duty: 速度 (0-65535(MAX))
戻り値: なし

今回は，マイコン内蔵モータドライバ回路 (TB6552) の出力をそのまま大出力モータに接続できるように信号増幅を行う Vstone 社製モータドライバ (VS-WRC005) を使用する．そのため，本試作検討では，モータ初期化関数 MotorInit() およびモータ駆動速度設定関数 Motor() において port 変数の引数を “0”を指定する．

6. 課題

以下に実際に行ってもらう課題を記す.

6.1 ポート初期化

今回は，マイコンボードの CN2 よりモータドライバ回路への入力信号として，PWM 信号を出力してもらう (Fig. 5 参照). そのためには，その CN2 へ PWM 信号を出力するポートの設定をしなければならない．ここで，CN2 より PWM 信号を出力するのはマイコン内蔵モータドライバ TB6552 を用いる．そのため，この TB6552 への入力として，PWM 信号を入力するポート P61 とモータ回転方向を制御するポート P32，P33 を出力として設定する必要がある．その初期化を行うプログラムが MotorInit() 関数であるが，この部分にあるプログラムの欠け/* ?????????? */の部分を埋めてもらう．具体的には，P61 から PWM 信号を出力するためには，先に述べたように P61 から PWM 信号を出力するようにレジスタのビットの値を変更してもらう.

Table 1: ポートの接続関係

ポートの名前	接続先	備考
P61 / FTIOB0	TB6552FN の BPWM	タイマ Z の出力，TB6552FN を介して CN2 への出力を制御
P32	TB6552FN の BIN1	TB6552FN を介して CN2 への出力を制御
P33	TB6552FN の BIN2	TB6552FN を介して CN2 への出力を制御

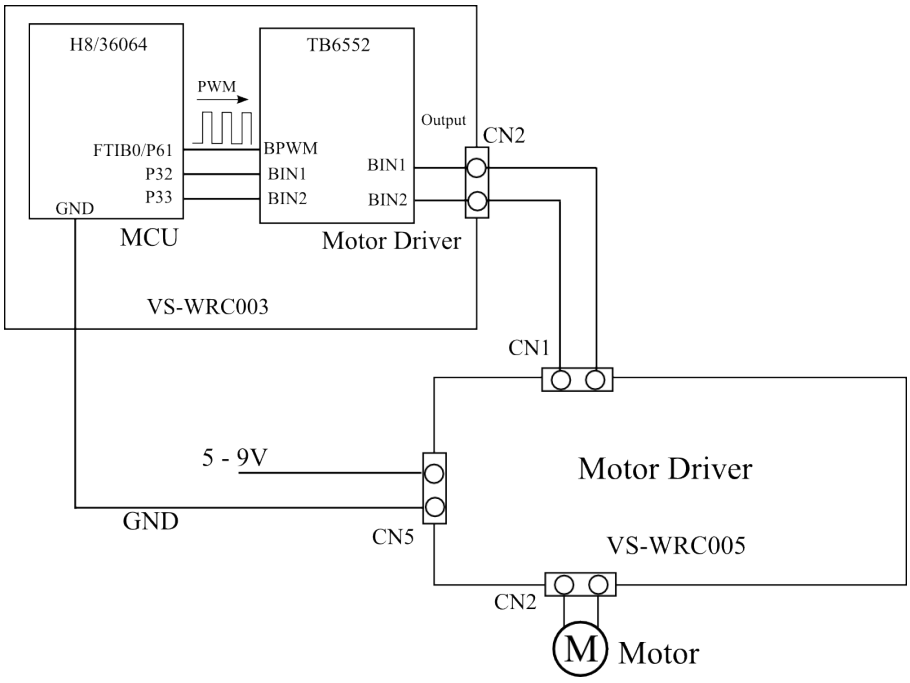

Fig. 5: マイコンと内蔵ドライバ，外付けドライバ，モータの接続関係 (課題に関連する部分のみ)

6.2 課題プログラム SS2shisaku02 の修正

ポート初期化ができたら，実際にモータを回転させるための出力を出してもらう．しかし，ただ出力を一定値 1 方向に出すのは役に立たないので，今回は Fig.6 のようなモータの正・逆回転を含むステートマシンを作成してもらう．状態としては STATE_A，STATE_B，STATE_C，STATE_D を用意したのでこれを使う．実際に自分たちがステートマシンを作成するときの状態についてはこの表記にとらわれる必要はない．また，状態遷移については，今回はタイマ B1 により作成した 3 秒周期の割込みを用いる．この状態遷移についても，今回のものに縛られる必要はない．なお，プログラムの欠けについては，/* ?????????? */の付近にコメントが付してあるので，それとハードウェアマニュアル，試作検討資料などを参考にしてほしい.

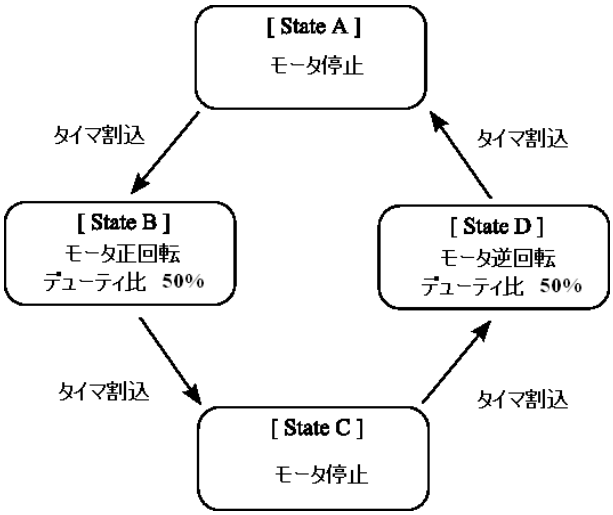

Fig. 6: 課題のステートマシン

7. 付録

本試作検討では，タイマ Z を PWM 信号の出力用に用いた．タイマ Z は 2 つのチャンネルを持ち，2 つのタイマを並列に実行することができる．試作検討では，このうちタイマ Z0 を用いている．このタイマ Z0 を用いて，Fig. 7 のような原理で FTIOB0 信号に PWM 信号を出力している．まず，TCNT0 レジスタを用いて，0xFFFF までカウントアップを行っている．そして，この TCNT0 レジスタの値が GRB0 レジスタの値より小さい場合には FTIOB0 信号の出力を “1”に，多きい場合には FTIOB0 信号の出力を “0”にセットするようコンペアマッチを行っている．GRB0 レジスタの値が試作検討におけるデューティ比を格納する変数 duty の値 (0x0000(0)〜0xFFFF(65535)) に相当する．

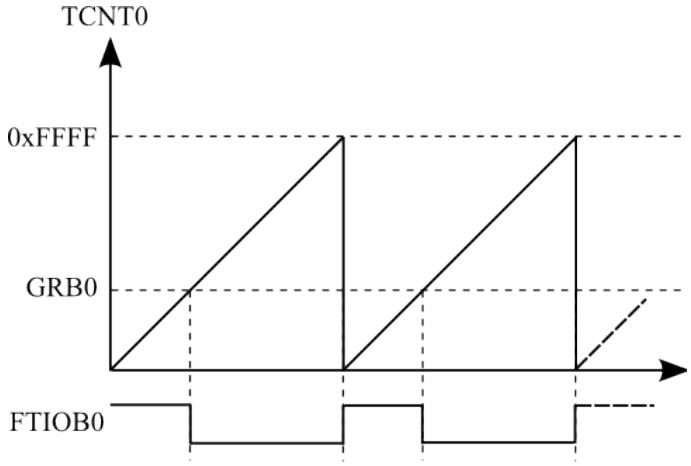

Fig. 7: PWM モードの原理

本試作検討においては，これを実現するためにプログラム Motor.c において，TB6552 へのポートの指定の後に，タイマ Z0 を用いるための初期化を行っている．

```
TZ0.TCR.BIT.TPSC = xxxx;
```

まず，タイマ Z0 の TCR レジスタの TPSC ビット 3 列分の初期化を行う．これにより，TCNT0 のカウントの周波数を決定する．今回はφ/4 周期でカウントさせるので，“2”と設定する．

```
TZ.TPMR.BIT.PWMB0 = xxxx;
```

次に，タイマ Z の PWM 許可をつかさどる TPMR レジスタの設定を行う．PWMB0 を用いるのでこれを “1”にセットする．

```
TZ.TOER.BIT.EB0 = xxxx;
```

そして，P61//FTIOB0 をタイマ出力として使用するために，TOER レジスタの EB0 ビット目を “0”にセットする．

```
TZ.TOCR.BIT.TOB0 = xxxx;
```

また，FTIOB0 の初期出力を “1”にセットするために，TOCR レジスタの TOB0 ビット目を “0”にセットする．

```
TZ0.TIORA.BIT.IOB = xxxx;
TZ0.GRB = xxxx;
```

その後，先に述べた PWM 信号出力を実現するために，コンペアマッチを行った結果，大きい場合において FTIOB0 の出力を “0”にするために，TIORA レジスタの IOB ビット列を “1”にセットする．同時に，GRB レジスタをコンペアマッチの値として用いるため，とりあえず出力を “0”とするために GRB レジスタの値を “0”にセットする．

```
TZ.TSTR.BIT.STR0 = xxxx;
```

最後にタイマ Z0 のカウントを開始させるために，TSTR レジスタの STR0 ビットを “1”にセットする．以上で，PWM 信号の出力設定は終了である．

なお，このタイマ Z の詳しい使い方・解説は第 1 回目試作検討で紹介された H8/36064 ハードウェアマニュアルの 12 章を参照してほしい．