

## 情報(2) H8 マイコンを利用した PWM によるモータ速度の制御

### 1. 目的

本実験は PWM と呼ばれる方式を用いて、H8 マイコンにより DC モータの回転速度を変化させる方法について学ぶことを目的とする。

### 2. 予備知識

#### 2.1 PWM とは

PWM は、Pulse Width Modulation (パルス幅変調) の略であり、モータに加える電圧や流す電流を、高速の ON/OFF 信号で可変しようという方法である。ON/OFF を非常に高速に行うことにより、モータ側から見ると、ON/OFF の 1 サイクル (1 周期の時間) に占める ON の時間の割合を全電圧に対する電圧の割合としてモータにかけられる。ON の時間と OFF の時間の比を Duty 比と呼び、この値は 0% (常時 OFF の状態) から 100% (常時 ON の状態) に変化させる (図 1 を参照) ことで、等価的にモータに加える電圧、またはモータを流す電流をゼロとそれぞれの最大値との間で変化させることができる。

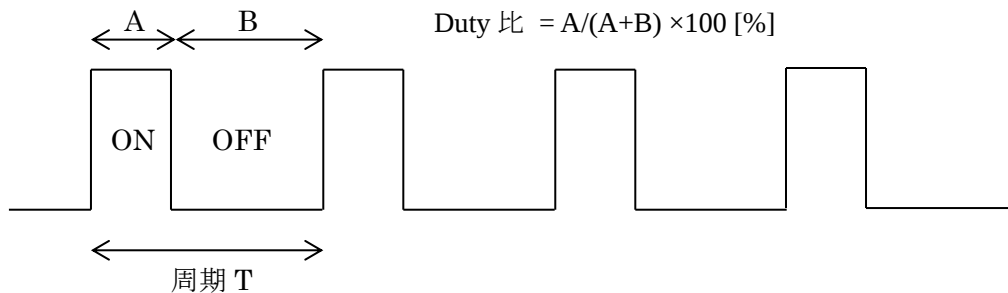


図 1 PWM における Duty 比

本実験において、ON と OFF のそれぞれの時間設定は、C 言語の for ループによるカウンタにより実現している。この方法は、精確な時間設定には向かないが非常に簡便であり直感的である。

#### 2.2 H-Bridge とは

モータへの電流の ON と OFF は、機械的なスイッチによって直接行うことよりも、回路により電子的に行うことが一般的である。そのような回路の 1 つとして H-Bridge 回路が挙げられる。H-Bridge 回路は図 2 に示すように、4 つのトランジスタとモータを、H 字型に配置した回路となっており、トランジスタの ON/OFF を変化させることにより、モータに流す電流を制御し、モータを ON/OFF したり、回転方向を制御したりすることができる。

その原理を図 3 で説明する。バイアスやゲート電圧を印加する等により、例えば図 3 (a) のように、A と D のトランジスタを導通状態にし、B と C は非導通状態としたとき、電流パスは A→モータ→D のようになり、モータに対

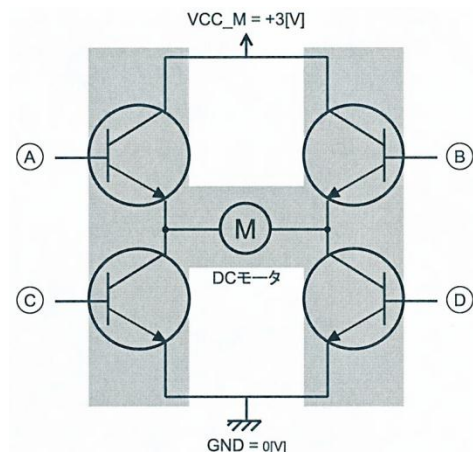


図 2 H-Bridge 回路

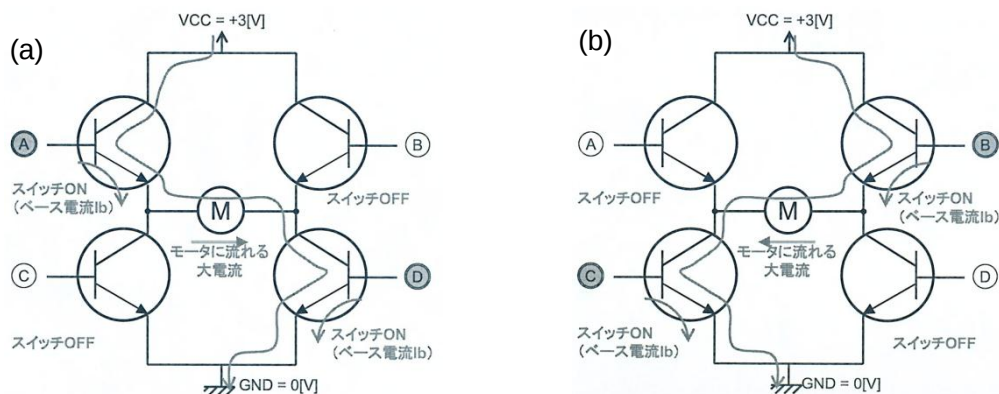


図3 トランジスタの ON/OFF とモータの回転方向

しては、電流が左から右に流れ、それに対応した回転が発生する。今度 A と D を非導通状態に、B と C を導通状態に変えた場合、電流パスは B→モータ→C のように変わり、モータには、電流が右から左に流れ、回転が逆転することになる。また、A、B、C、D のトランジスタの少なくとも 3 つ以上を非導通状態にすれば、モータに流れる電流がゼロになり、モータは回転しなくなる。

本実験のベース基板上にある 4 つの電界効果トランジスタ (FET) は、それぞれ CPU の PORT6 に繋がっており、H-Bridge 回路は、PORT6 に対して出力される信号により制御される (図 4 を参照)。

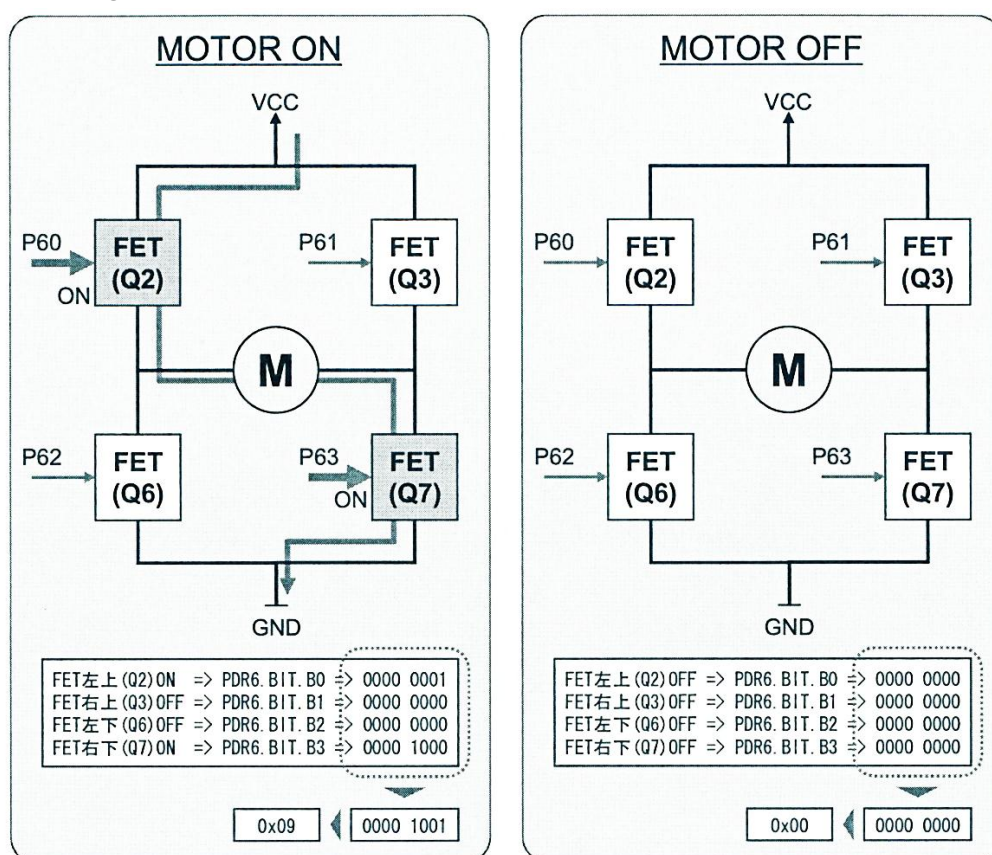


図4 H-Bridge に対する指示方法

モータ (M) を時計回りに回転させるために、図 4 の左のように、Q2 と Q7 を導通に、Q3 と Q6 を非導通にすればいいので、IO.PDR6.BYTE=0x09 とセットする。モータの回転を止めるためにすべての FET

を非導通にすればよく、すなわち、IO.PDR6.BYTE=0x00 と設定すればよい（図 4 の右側を参照）。ちなみに、モータを反時計周りに回転させるためには、Q3 と Q6 を導通状態に、Q2 と Q7 を非導通状態にすればよい。

### 3. 実験内容

この実験では、e-nuvo BASIC のベース基板上にあるポテンショメーター（ボリューム）、LED 及び DC モータを用いて、PWM によりモータ速度の制御について学習する。まず情報（1）で実験したように、ポテンショメーターからの信号を A/D 変換で CPU に取り込み、その結果を 8 ビット信号として、LED に出力する。それと同時に、ポテンショメーターの入力信号に応じて PWM の DUTY 比を計算し、それによりモータの回転速度を変化させる。図 5 に本実験の入出力構成を示す。

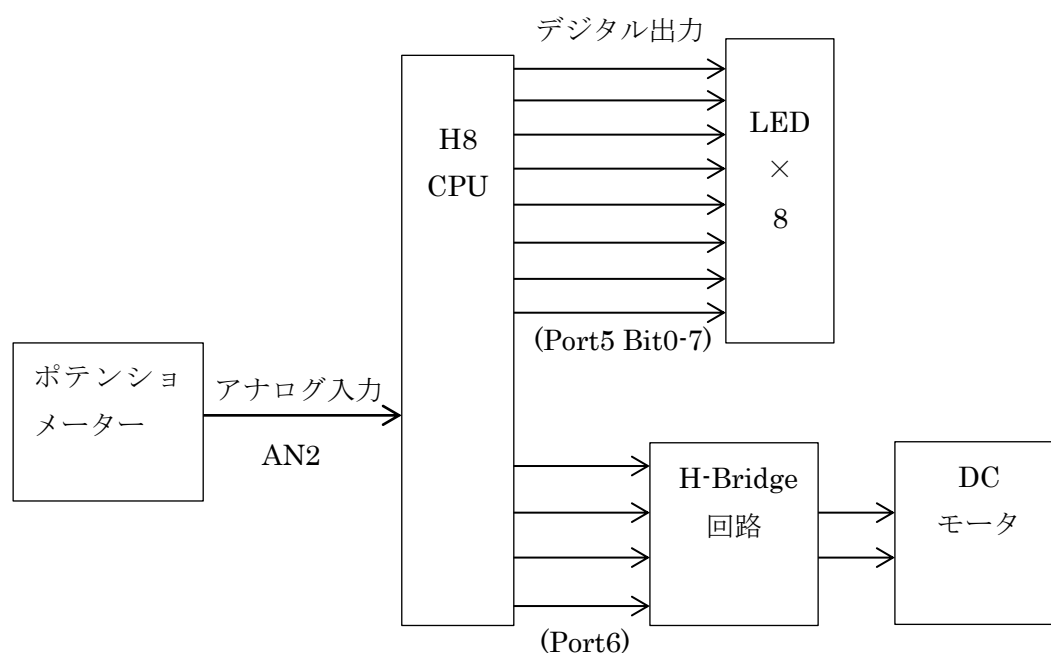


図 5 本実験の入出力構成

ポテンショメーターの電気抵抗値より変換された電圧（アナログ値）を情報（1）の課題 3 のように、電圧値と点灯する LED の数とがだいたい比例するようにする。

なお、本実験の操作は、「電子情報実験 I」で行った「H8 マイコンを用いた電子情報システム実験入門」などで習った実験手順とほぼ同じであり、実験を行う前に、必ず「電磁情報実験 I」の課題 10、課題 11 及び課題 12 の実験マニュアル（以下参照資料という）を復習すること。

### 4. 実験手順

#### 4.1 必要機材

本実験では、「電子情報実験 I」のマイコン関連実験で用いた機材と同じものを使用する。プログラムの編集、コンパイルおよびフラッシュメモリへの書き込み等も同じツールを利用する。主な必要機材は下記に示す。

- e-nuvo BASIC (図 6)
- 各種ケーブル、AC アダプター

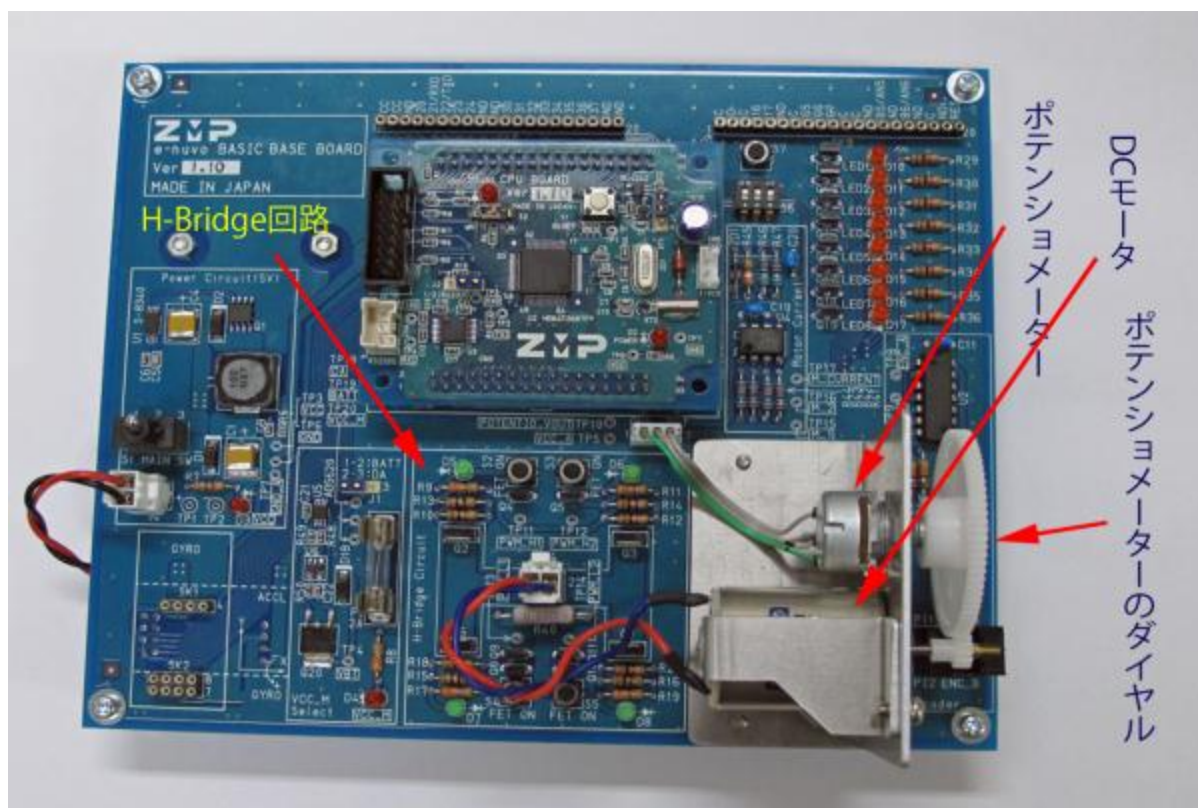


図6 e-nuvo BASIC (BASE BOARD & CPU BOARD)

- パソコン及び開発環境一式

注：e-nuvo BASIC ベースボード上のポテンシオメータのダイヤルと DC モータの回転軸についている小さな歯車とが離れていなければならないので、もし両者が噛み合っている場合（図7の右側を参照）、ベースボードの電源を入れずにすぐに教員に連絡すること。

#### 4.2 実験の手順

実験の流れと手順を以下に示す。

##### A) 新規プロジェクトの作成

H8の統合開発環境である HEW を起動し、新規プロジェクトを作成し、プログラミングの準備をする。各種パラメータを下記のように設定する。

- ・ ワークスペース： PWM\_MOTOR



図7 ポテンシオメータのダイヤルと DC モータの回転軸歯車との関係

- CPU 種別： 300H
- CPU タイプ： 3687
- ディレクトリ： z:\¥PWM\_MOTOR

また、プロジェクトに、ヘッダファイル「3687.h」を追加する。

(不明な場合、関連資料を参照すること。)

## B) プログラムの作成

C 言語で、A/D 変換、LED の点灯及び PWM 生成プログラムを作成する。具体的に、HEW の中で、PWM\_MOTOR.c を下記のように書き換える（網掛けされた箇所は、修正部分である）。網掛けされた部分以外も下記リストと異なる部分もあるかも知れないが、気にしなくて良い。

```
#include "3687.h"
void init(void);
void MOTOR_ON(int duty);
int LED_VAL(int val);
void main(void);
#ifdef __cplusplus
extern "C" {
void abort(void);
}
#endif

//関数：初期設定
void init(void)
{
    // 入出力設定
    IO.PCR5 = 0xFF;           // ポート5の入出力設定 (LED 出力用)
    IO.PCR6 = 0xFF;           // ポート6の入出力設定 (モータ制御用)
                                // (H-Bridge: P60-P63)

    // A/D 変換設定
    AD.ADCSR.BIT.ADST = 0;    // A/D 変換を停止
    AD.ADCSR.BIT.SCAN = 0;    // A/D 変換のモード設定
    AD.ADCSR.BIT.CKS = 1;     // 変換時間を設定
    AD.ADCSR.BIT.CH = 2;     // アナログ入力チャンネルの選択
}

//関数： LED 点灯パターン値の計算
int LED_VAL(int val)
{
    int i, ret=1;
    if (val > 0) {
        for (i=0; i<=(val+1)/32; i++) ret *= 2;
    }
    return ret-1;
}
```

```

// 関数：モータ速度を PWM で制御
void MOTOR_ON(int duty)
{
    int i, j;
    // モータ ON
    IO.PDR6.BYTE = 0x09;          // モータ ON => 0000 1001 => 0x09
                                   // FET ON : Q2 (P60), Q7 (P63)
                                   // FET OFF: Q3 (P61), Q6 (P62)
    for (i=0; i<duty; i++) {      // モータ ON にしている時間
        for (j=0; j<=100; j++);
    }

    // モータ OFF
    IO.PDR6.BYTE = 0x00;
    for (i=duty; i<=256; i++) {   // モータ OFF にしている時間
        for (j=0; j<=100; j++);
    }
}

void main(void)
{
    int result;
    init();          // 初期環境設定

    // メインループ
    while (1)
    {
        // A/D 変換を開始
        AD.ADCSR.BIT.ADST = 1;
        // A/D 変換終了まで待機
        while (AD.ADCSR.BIT.ADF==0) {}
        // A/D 変換結果を 8 ビットの数字に
        result = AD.ADDRC>>8;

        // A/D 変換結果を LED 出力
        IO.PDR5.BYTE = LED_VAL(result);

        // A/D 変換結果をモータ速度として出力
        MOTOR_ON(result);
    }
}

#ifdef __cplusplus
void abort(void)
{
}
#endif

```

C) コンパイル

作成したプログラムをコンパイルし、H8 マイコンが理解できる形式に変換する。

(不明な場合、関連資料を参照すること。)

D) プログラムをマイコンに書き込む

PC と H8 マイコンを接続し、コンパイルしたプログラムをマイコンに書き込む。

(不明な場合、関連資料を参照すること。)

E) 動作確認

電源を入れ、プログラム通り、ポテンシオメータのダイヤルを回して、LED の点灯がそれに従って変化すると同時に、モータの速度もダイヤルの回転に従って変化することを確認する。ポテンシオメータの抵抗値を大きく又は小さくすると、LED の点灯数も多く又は少なくなり、モータの回転が速く又は遅くなるのであれば、実験が成功である。なお、モータ回転の速さについては、目視及び回転音により感覚を掴むこと。

## 5. 課題

### ・課題 1

作成したプログラムのフローチャートを描き、説明せよ。また、モータの回転数を全体的に下げするために、どのようなアプローチがあるか（どうすればよいか）について考察せよ。

### ・課題 2

プログラムを書き換えて、ポテンシオメータダイヤルの真ん中を境目に、時計周りに回すと、モータも時計周りに回転し、ダイヤルを反時計周りに回すと、モータも反時計方向に回転し、ダイヤルが真ん中の位置から離れるほど、回転が速くなるようにする。動作確認をする。PWM\_MOTOR.c の中のその変更当たる修正部分を書き出せ、説明せよ。

### ・課題 3

課題 2 ができたら、LED の点灯動作も課題 2 の変更に合わせて変更するようにプログラムを書き換える。変更のやり方は自由で、直感的で合理的な点灯動作である同時にユニークであれば高く評価される。