３章　A／D（アナログ／デジタル）変換とアクチュエータ制御

■■　第３周目　アナログ情報をデジタル情報に変換して扱う　■■

ここでは，A/D 変換と呼ばれる情報処理手法について学習します．A はアナログ，D はデジタルの頭文字を取った技術用語です．マイコン内部には，A/D 変換機（A/D ｺﾝﾊﾞｰﾀ）というユニットが内蔵されています．これを用いることでアナログ信号をデジタル信号に変換することができます．これにより時々刻々，切れ間なくつながりを持って変化するアナログ信号を，コンピュータなどで使われている０，１のデジタル情報に変換できます．

本実験では，A／D 変換の仕組みを学習し，実験を通してアナログ信号とデジタル情報の扱い方，および情報処理能力を養うことを学習の目的とします．

５　ここでは A／D（アナログ／デジタル）変換を利用した実験を主に学習します．

１）ボード上の可変抵抗器の電圧を，テスタで読み取った値と，AD 変換して，モニタに表示される数値の関係をグラフにする．

２）コンデンサマイクを用いて音声アナログ信号をデジタル化しグラフ化する．

５－１．H8 マイコンの AD 変換器

H8 マイコンの AD 変換器のブロック図を図１に示します．これは，アナログ入力端子 P7-0～P7-7 に加えられた０～５V のアナログ電圧を１０ビット（０～１０２３）の数値に対応付けて変換するユニットです．

A/D 変換器の特長

・10 ビットの分解能
・入力チャネル：8 チャネル
・変換時間：1 チャネル当り最小 5.4μs（25MHz 動作時）
・単一モード／スキャンモードの 2 種類の動作モード
　単一モード：1 チャネルの A/D 変換
　スキャンモード：1～4 チャネルの連続 A/D 変換
・4 本の 16 ビットデータレジスタを持つ．A/D 変換された結果は，各チャネルに対応したデータレジスタに転送され，保持されます．
・サンプル&ホールド機能
・外部トリガ信号による，A/D 変換の開始が可能
・A/D 変換終了割込み要求を発生．A/D 変換終了時には，A/D 変換終了割込み（ADI）要求を発生させることができます．

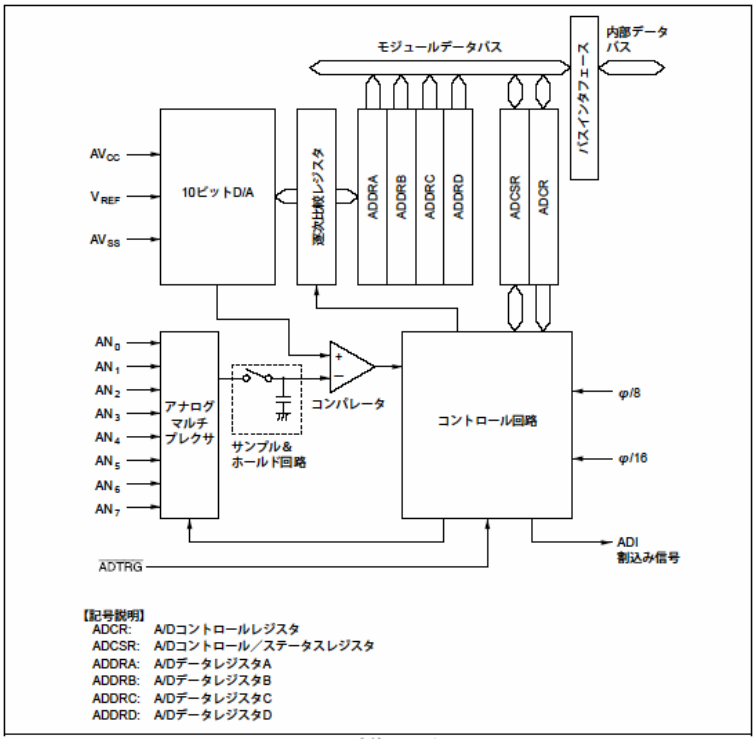


図１　A/D 変換器のブロック図（ハードウエアマニュアル 15-2 より抜粋）

５－２．AD 変換を行う際の流れ

１）　AN0～7 のいずれかのポートに測定したい電圧（０～５V）信号を接続する．

～～～　プログラム側の手順　～～～

２）　信号を接続したポートの番号をプログラムで指定する．

３）　AD をスタートする．

４）　AD 変換が完了を示すフラグ（ビット）が１になるまで待つ．

５）　フラグをリセットして，次の変換を待つ．

６）　AD 変換された値は，A～D の４つのデータレジスタに格納されている．これらは，以下に示すようにレジスタ１つに対して２つのポートが割り当てられている．

A レジスタ(ADDRA)	ch0, ch4	P7-0, P7-4
B レジスタ(ADDRB)	ch1, ch5	P7-1, P7-5
C レジスタ(ADDRC)	ch2, ch6	P7-2, P7-6
D レジスタ(ADDRD)	ch3, ch7	P7-3, P7-7

７）　AD 変換用のレジスタは１６ビットであるが，AD 変換器は１０ビットのため，上位より書き込まれると下位６bit が空いてしまう．このためレジスタの値を６ビット右シフトして通常の変数にしてから，変換したデータをその後に扱う．

５－３．A/D 変換用レジスタ群

・A/D コントロール／ステータスレジスタ（ADCSR）

ADCSR の各ビットの役割を下図に示します．本実験では，単一モード，割込，クロックのセレクトは行いません．基本的には，チャンネルを選択して，AD スタート後，変換終了を待ち，データレジスタから値を取り出す方法で行います．

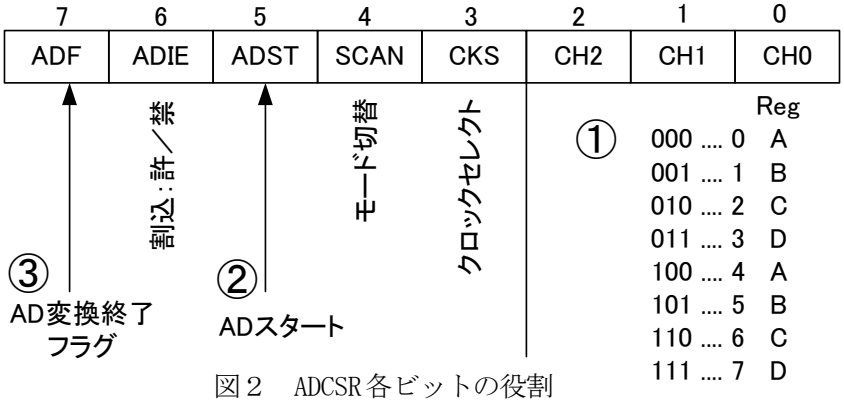
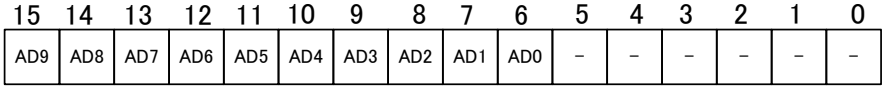


図２　ADCSR 各ビットの役割

・A/D データレジスタ A～D（ADDRA～ADDRD）

AD 変換完了後，チャンネル対応した 16 ビットのデータレジスタに入力された値が，図３のように入ります．これを，変数などに数値として移し替えるには，６ビット右にシフトする必要があります．チャンネルとレジスタの関係を表１に示します．



④　　図３　AD データレジスタのビット配置

表１ チャンネル割当

CH	Register
0	A
1	B
2	C
3	D
4	A
5	B
6	C
7	D

・A/D コントロールレジスタ（ADCR）

外部トリガ用（本テキストでは使用しません）

・可変抵抗器出力電圧の AD 変換によるアナログデータ取り込みプログラム例

```
while (1) {
AD.CSR.BIT.CH=2;              /* ① ch=2 に設定    */
AD.CSR.BIT.ADST=1;            /* ②AD スタート*/
while (AD.CSR.BIT.ADF==0);    /* ③ AD 終了待ち   */
AD.CSR.BIT.ADF=0;             /* ③ ADF  Reset*/
dd=AD.DRC >> 6;               /* ④ 変換データの取り出し後 6bit right shift */
printf("VR=%d  ¥n",dd);     }
```

５－２．可変抵抗を利用したアナログ電圧の A/D 入力

【演習１１】A/D 変換入力実験

実験手順

１）図４はボードの拡大写真で，可変抵抗に５V の電圧を加えおり，中間端子からは，つまみの回転角に比例した電圧を出力します．この出力はポート P7-2 に接続してあります．図４の＋端子 AD2 が P7-2 からの引き出した端子で（直流電圧の１２V レンジにした）テスタのテストリード(黒)を図４の－端子(GND)，赤側を＋端子に接続します．

２）P7-2(ch2)の電圧を AD 変換して，その値をモニタ窓に表示するプログラムを作成し，実行します（VR_servo.c を参照）．

３）０V から５V までをテスタで計測しながら 0.5V 刻みに電圧変化させ，その時の AD 変換した値（変換値）をパソコンのモニタで読み取り，表２のようにまとめます．

４）パソコンのモニタに表示された値を選択し，コピー＆ペーストで Microsoft Excel に貼り付けます．Excel に取り込んだデータをグラフ（散布図）化し，近似曲線の追加機能を使い，近似直線と式を記録します．また，作成するグラフの例を図４に示します．

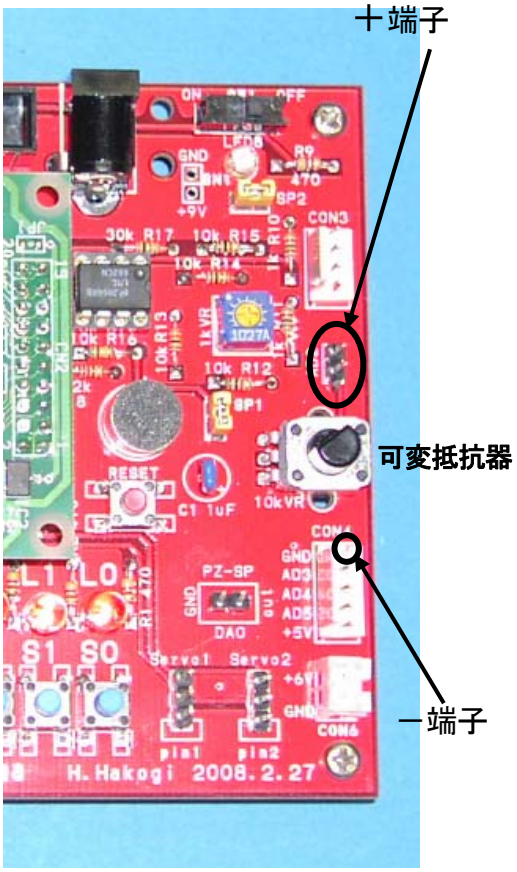


図４　ボード拡大写真

課題１　上記の実験を行い，実験の結果を表２と図４に習ってまとめ，報告しなさい．また，この実験結果について考察しなさい．（VR_servo.c を利用する）

課題２　可変抵抗を絞ったとき，すべての LED を消灯し，一杯に回したとき全ての LED が点灯するプログラムを作成し報告しなさい．（保存名　Prg11_2.c とする）
ヒント：AD 変換された数値の最大・最小を考え，処理を考える．回答は何通りも考えられます．

課題３　課題２のソースプログラムに解説文を加え報告しなさい．

表２　電圧－変換値の実験結果（例）

電圧	変換値
0	0
0.5	110
1	190
1.5	300
2	404
2.5	510
3	600
3.5	707
4	820
4.5	890
5	1023

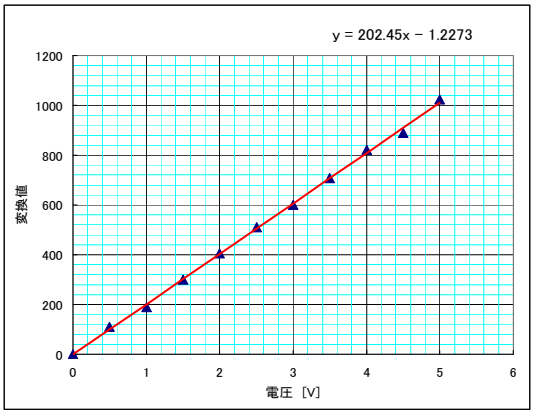


図５　電圧―変換値グラフ（例）

対象プログラム名　VR_servo.c　←　第４週目で使用するプログラムを利用する

```
#include<3052f.h>
#include <stdio.h>

int ad(int ch)
{
      AD.CSR.BIT.CH=ch;
      AD.CSR.BIT.ADST=1;
      while (AD.CSR.BIT.ADF==0);
      AD.CSR.BIT.ADF=0;

    return (AD.DRC >> 6);
}

main()
{
      long int cnt;
      int p1,p2;
      float d0,d1;

      P1.DDR= 0xff;       /* P1-0 ~ P1-7    8 LED */
      P2.DDR= 0x00;       /* P2-0 ~ P2-3    4 switch */
      P6.DDR= 0x0f;       /* P6-0,P6-1      2ch RC servo */
      P1.DR.BYTE =0;

      while (1) {
            p2 = ad(2);
            p1 = p2*2+300;
            printf("VR=%d    ¥n",p2);

            P1.DR.BIT.B3 = !P1.DR.BIT.B3;

            for (cnt=0;cnt<15000;cnt++){
                  if ( cnt<p1 ) P6.DR.BIT.B0 = 1;
                    else P6.DR.BIT.B0 = 0;
            }
      }
}
```

本実験のポイント
となる部分
（AD を使用するための初期設定）

この実験では
本質でない部分

５－３．　MIC を用いた音声信号の AD 変換実験

【演習２】音声の取り込み実験

実験手順

１）マイコン実験ボードの MIC 回路部周辺の拡大写真を図６に示します．

２）ショートピンを SP2 に接続します（電源電圧の約 1/2 にバイアスをかける）．

３）テストプログラム MIC01.c を実行します．モニタ画面に表示される数値がほぼ 512 となるようにオフセット調節用の可変抵抗を，マイクロドライバーで調節する．手法例として，S7 のみが消灯し，他は全て点灯するように調節する方法もほぼ同じ結果が得られます．（プログラムでは AD 変換された値を２bit 右にシフトして表示させています．余裕のある学生はこの理由について考察しなさい）

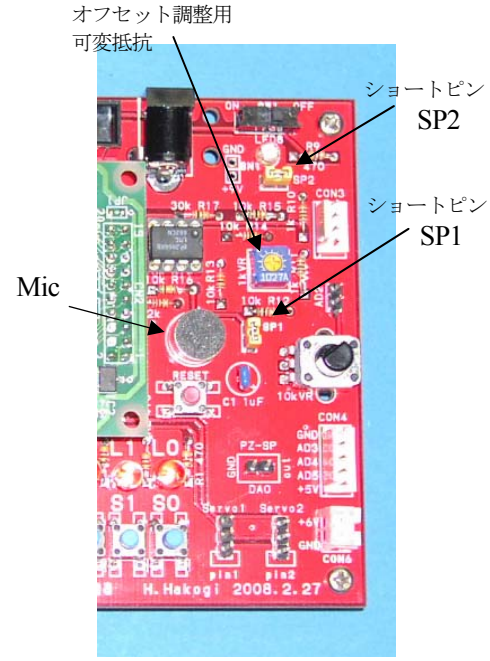


図６　MIC 周辺回路拡大図

４）ショートピンを SP1 へも接続します(図７参照)．これにより MIC 信号が接続されるので，音声がマイコンに取り込まれ，各 LED がパラパラ変化することを確認します．

５）適度に MIC へ近づき，発声します．このときモニタ画面の数値を見て，1023 や 0 を出力しているときは音声が大きすぎて音が割れている状態となっているので，モニタを見ながら適切な値の出る声の大きさを加減します．この状態で音声データを感覚的に 300 個程度取り込んだ直後，リセットスイッチを押しプログラムを止めます．

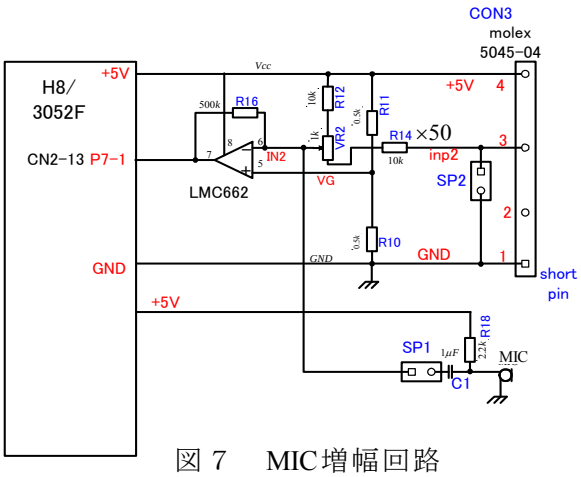


図７　MIC 増幅回路

６）モニタ画面の数値を 100 個から 200 個程度コピーし，Excel を起動し，先頭セルにデータを貼り付けます．

７）グラフウィザード機能を使って波形を表示させます．表示したグラフを確認し，音が割れる，小さすぎるなどの場合は，再度データを取り直します．なお，本実験はこのプロセスの繰り返すことにより，適当なデータ計測のコツがつかめます．

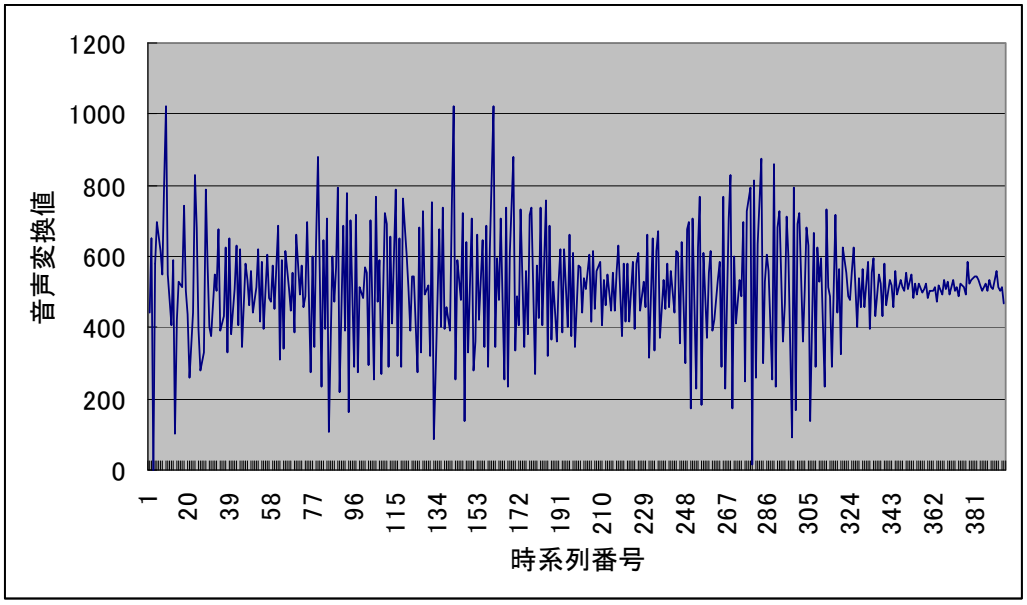


図８　音声データをデジタル化したグラフの例

課題４　Excel を使って音声信号波形を描き，プロパティの書式設定やグラフオプションを利用して，きれいなグラフに編集し，グラフに必要な項目（単位や目盛りなど）を書き込み報告書に添付しなさい．（グラフの書き方をチェックする）．その際，測定方法，手順，条件など必要事項を記載すること．また，声の波形は，このグラフ中で十分大きく，しかも割れていないこととする．また，データ数を１００～２００程度とすること．

課題５　MIC01.c のプログラムを用いて，音声データのサンプリング間隔を長くした場合の音声波形を，課題４同様に行いグラフ化しなさい．
ヒント：遅延時間を入れる．（保存名　Prg12_5.c ）

課題６　課題４と課題５の実験結果を比較検討して，どのような違いがあるか，考察しなさい．（例えば，双方のシステム的な面や，データでの長所と短所やこのような実験対象の場合，異なる条件で比較するためのデータはどのようにサンプリングする必要があるのか，その場合の注意点など，様々な角度で考察できれば良い）

■■　第４周目　マイコンによるアクチュエータ制御 ■■

この実習では，２相ハイブリッド型ステッピングモータと RC サーボモータの基本構造を学び，それぞれのアクチュエータのマイコンによる制御方法，およびそれに必要な周辺回路を学習することを目的とします.

６．　ステッピングモータの原理と実験

本実験では, マイコンから出力する制御信号によって制御されるドライバ IC を介して，２相ハイブリッド型ステッピングモータの各相の励磁を順次切り替え，微少単位でモータのローターが回転する事を確認します.

実践的な内容として，これらの順序をスイッチを押す度に切り替わるようなプログラムを作り，次に，ITU タイマを用いて一定速度で回転させます．さらに，可変抵抗の値を AD 変換した値で，タイマ時間を変化させて可変抵抗の回転角と回転速度を連携させます.

自由な回転制御を可能にした後，モータの回転速度を連続的に変化させ，共振点や限界周波数による脱調を確認します．また，一定ステップ数だけ移動させるプログラムを製作し，１ステップの角度（1.8 度）を確認する実験を行います.

６－１　ステッピングモータの構造と基本原理

ステッピングモータは，歯車形状のロータと歯車形状に対応した凹凸を持つステータからなっています．ハイブリッドステッピングモータのロータは，歯車形状の鉄心２個を軸方向に磁化した永久磁石をはさんだサンドイッチ構造です．ロータの歯に対向して，２組の電磁石が８個配置されています．全体では歯数５０のロータを使用していますが，下図ではその一部を割愛し，各相を励磁する極性によりロータが少しずつ回転する様子を示します.

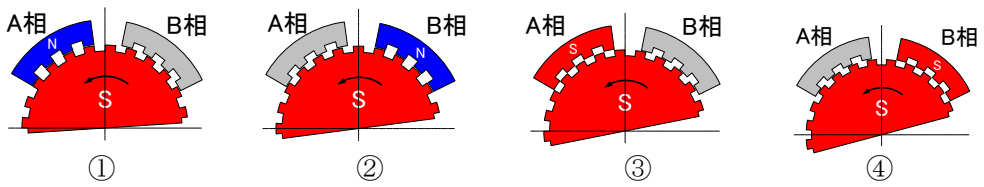


図１　ステッピングモータの励磁順序

６－２　励磁と回転順序

１）A 相のコイルを N 極に励磁します.

2）　A 相の電源を切り, B 相のコイルを N 極に励磁すると, 歯車が 1/4 ピッチ進みます.

3）　次に, B 相の電源を切り, A 相のコイルに逆向き(S 極)に励磁すると, 更に歯車の 1/4 ピッチ進みます.

４）同様に A 相の電源を切り，B 相のコイルに逆向き（S 極）に励磁すると，さらに，歯車は 1/4 ピッチ進みます．

以上の４種類の動作を繰り返すごとに歯車が１ピッチ回転するので．この動作を 200 回繰り返す事によってローターが１回転します．この励磁方法を１相励磁といいます．

A 相，B 相の両方コイルを励磁すると，ローターの回転角は両者の中間点の角度となります．常に２つのコイルが励磁するようにドライバ IC を制御し，順次切り替える方式を２相励磁といいます．２相励磁は，１相励磁に比べて２倍の電力量となり発熱量も大きくなりますが，高トルクで振動も少なくなります．また，１相励磁と２相励磁を交互に励磁すると，1/8 ピッチ進むため，４００ステップで１回転となります．これを１－２相励磁と呼んでいます．

６－３　モータの接続回路

実験で用いるモータドライバ IC のブロック図を図２に，モータの接続回路図を図３に示します．H8 マイコンのポート 3 の 2, 3, 4, 5bit とモータドライバ IC（TA7279AP）が接続されています．このドライバ IC は 2ch の制御が可能なので２相分の励磁が制御できます．この回路では図のように中間タップを接続せず，励磁の向きを 2 方向とするバイポーラ駆動を行っているので，この 2 相分のコイルをドライバ IC に接続することによりモータの制御が可能となります．なお，実験では，モーター用の外部電源として乾電池 4 個(6V)を使用します．

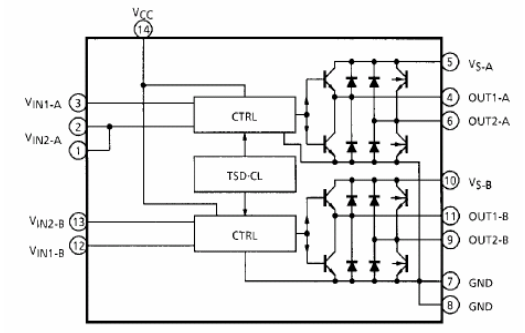


図２　ドライバ IC ブロック図

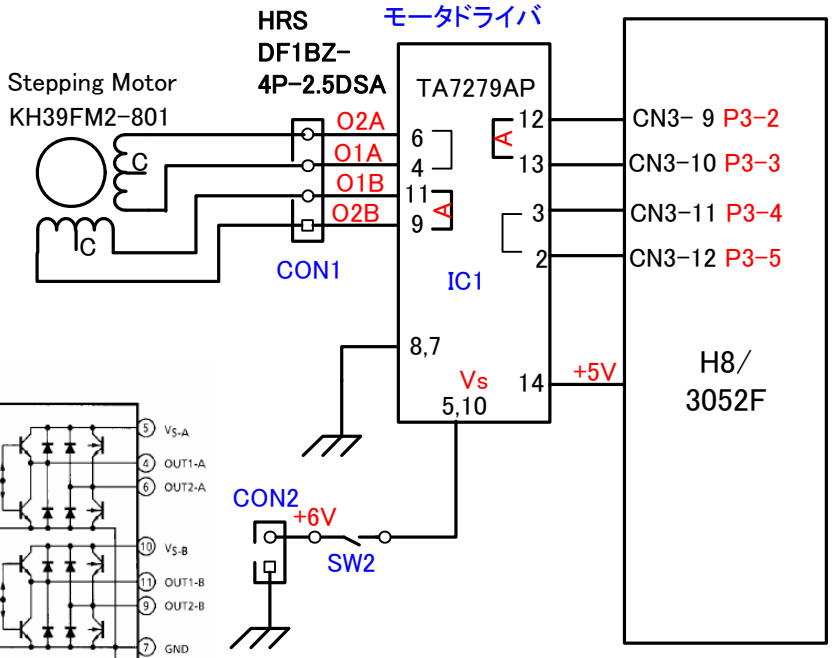


図３　ステッピングモータ接続回路図

６－４．　モータドライバのファンクションテーブルとポート出力

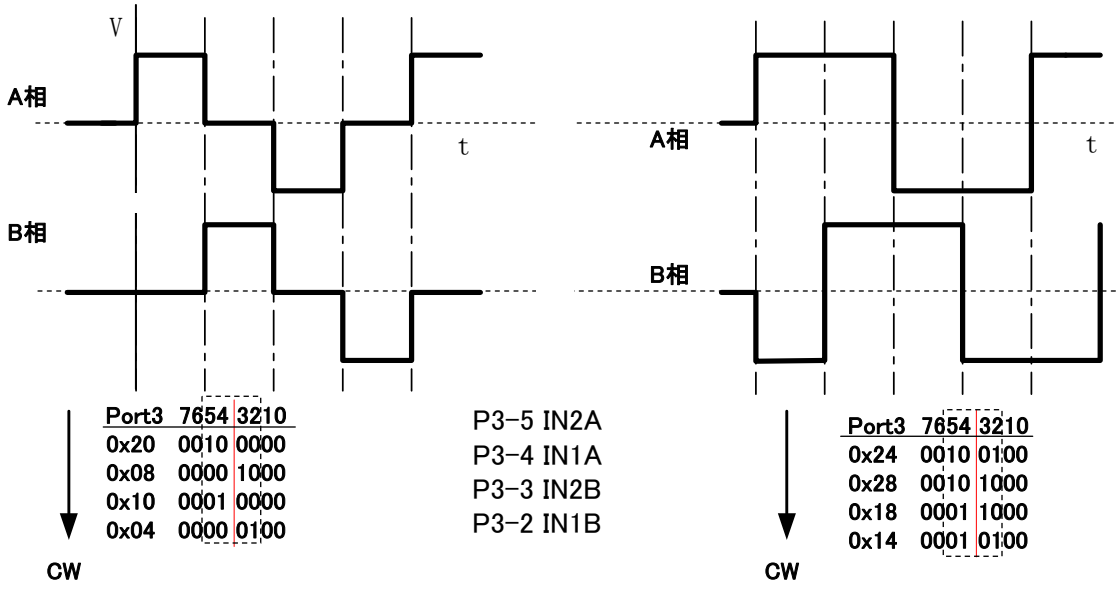
モータドライバ TA7279AP のファンクションテーブルを表３に示します．図２，３のステッピングモータの接続回路が示すように，マイコンの出力ポートはドライバ IC 側の IN1 および IN2 に接続されています. IN1 と IN2 の双方が０の場合はハイインピーダンス状態，つまり，回路とモータを切り離した状態を電気的に作っているので電流が全く流れていない状態と等しくなります．一方，双方が１の場合は，モータコイルの両端をショートさせた状態を電気的に作るため，モータの回転時に発生する誘導起電力による磁界とローターの永久磁石の影響を受け，磁気的な制動作用(ブレーキ)が起こります.

表３　モータドライバのファンクションテーブル

IN1	IN2	OUT1	OUT2	MODE
1	1	L	L	BRAKE
0	1	L	H	CW / CCW
1	0	H	L	CCW / CW
0	0	ハイインピーダンス		STOP

６－５．励磁方式と制御波形

モータドライバ IC と接続されたポート P3 に，励磁方向を切り替えるための制御信号をマイコンから順次出力することにより，モータの回転が制御できます．図４に，１相励磁および２相励磁方式に対するポート３の出力信号と，そのときコイルに印加した電圧の波形を示します．出力順序を逆にする事により逆回転も可能です．また，両者を交互に出力することで, 1 - 2 相励磁制御も可能となります.



a）１相励磁　　b）２相励磁

図４　１相励磁と２相励磁方式の出力信号と印加電圧の波形

【演習１３】ステッピングモータの制御実験

課題１　ステッピングモータの４つの状態をタクトスイッチ４個に割り当て，手動でモータを回転させられる下記のプログラムを作成し実行なさい．また，網掛け部の P3.DR.BYTE の部分の数値の意味を理解し，その説明を簡潔に報告しなさい．

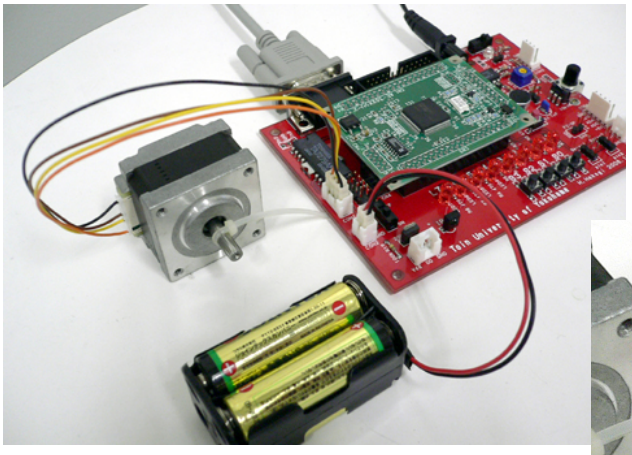
２相励磁制御　　(step_switch4.c)

```
#include<3052f.h>
main()
{
    int i,j;
    P1.DDR= 0xff;    /* P1-0 ~ P1-7  8 LED */
    P2.DDR= 0x00;    /* P2-0 ~ P2-3  4 switch */
    P2.PCR.BYTE=0xff; /* pull up */
    P3.DDR=0xff;

    while (1) {
        for (i=0;i<4;i++) {            /*回転順序*/
            for (j=0;j<30000;j++){ }   /*時間稼ぎ*/
            if ( P2.DR.BIT.B0==0 ) {P3.DR.BYTE=0x24;P1.DR.BYTE=0x01;}
            if ( P2.DR.BIT.B1==0 ) {P3.DR.BYTE=0x28;P1.DR.BYTE=0x02;}
            if ( P2.DR.BIT.B2==0 ) {P3.DR.BYTE=0x18;P1.DR.BYTE=0x04;}
            if ( P2.DR.BIT.B3==0 ) {P3.DR.BYTE=0x14;P1.DR.BYTE=0x08;}
        }
    }
}
```

ステッピングモータの接続方法

H8 マイコンボードへのステッピングモータと電源の接続は，下図ように行います．



拡大図

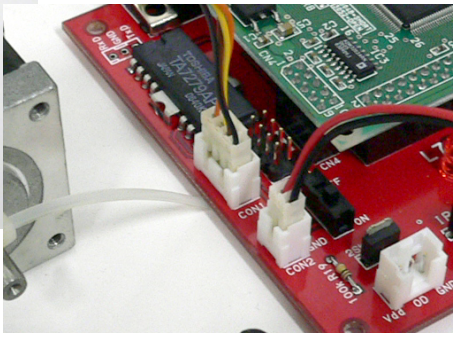


図５　ステッピングモータの接続

課題２　タクトスイッチ１個を用いて，回転と停止を行う制御プログラム（押している間だけ一定速度で回転する）を作成しなさい．また回転速度を変えるためにはどうしたらよいか，実験的に確かめた上で報告書に回答しなさい．

(step_switch1.c)

```
#include<3052f.h>
main()
{
    int i,j;
    P1.DDR= 0xff;       /* P1-0 ~ P1-7  8 LED */
    P2.DDR= 0x00;       /* P2-0 ~ P2-3  4 switch */
    P2.PCR.BYTE=0xff;  /* pull up */
    P3.DDR=0xff;

    while (1) {
        for (i=0;i<4;i++) {             /*回転順序*/
            while (P2.DR.BIT.B0==1){}  /*キー待ち*/
            for (j=0;j<30000;j++){ }    /*時間稼ぎ*/
                if ( i==0 ) {P3.DR.BYTE=0x24;P1.DR.BYTE=0x01;}
                if ( i==1 ) {P3.DR.BYTE=0x28;P1.DR.BYTE=0x02;}
                if ( i==2 ) {P3.DR.BYTE=0x18;P1.DR.BYTE=0x04;}
                if ( i==3 ) {P3.DR.BYTE=0x14;P1.DR.BYTE=0x08;}
        }
    }
}
```

課題３　タクトスイッチを２個使用し，１つのスイッチを押すと正転，もう一つを押すと逆転する正逆転が可能なプログラムを作成しなさい．また，配列(プログラム中の a[4])について調査し，知り得た知見を報告書に述べなさい．(step_f_b.c)

```
#include<3052f.h>
main()
{
    int i,d,a[4]={0x24,0x28,0x18,0x14};

    P2.DDR= 0x00;       /* P2-0 ~ P2-3  4 switch */
    P2.PCR.BYTE=0xff;  /* pull up */
    P3.DDR=0xff;

    while (1) {
         if (P2.DR.BIT.B0==1) d++;
         if (P2.DR.BIT.B1==1) d--;

         if ( d<0 ) d=3;
         if ( d>3 ) d=0;
         P3.DR.BYTE=a[d];
         for (i=0;i<30000;i++){ }    /*時間稼ぎ*/
    }
}
```

課題４　ボード上の可変抵抗の信号(電圧)を AD 変換して，その値に比例した速度制御を行うプログラムを実行し動作を確認しなさい．ただし，待ち時間は ITU を使用すること．作成の方法は，配布するサンプル step01.c を利用し修正します．第 2 週の課題２２を使えば変更は非常に容易です．まず， h8_toin¥cc に保存した，wait.h を以下のようにコピー&ペーストで少々設定を追加します．上書き保存したら，次に，次ページのサンプルプログラムを参考に step01.c を変更し，完成したプログラムのファイル名は step_itu.c で保存します．

(wait.h)

```
init_wait()
{
      ITU0.TCR.BYTE=0x21;   /* GRA compare ,   clock= 1/2 */
      ITU0.GRA=12500;       /* GRA 設定 1ms~25MHz/2/12500 */
}
init_wait01()
{
      ITU0.TCR.BYTE=0x21;
      ITU0.GRA=1000;
}
wait(int t)  /* t=100 →  0.1 秒のウエイト  */
{

      int tim;
                ITU.TSTR.BIT.STR0=1;   /*ITU0 TCNT Start */
                for (tim=0 ; tim<t;tim++){
                while (ITU0.TSR.BIT.IMFA==0){}
                          ITU0.TSR.BIT.IMFA=0;
       }
}
```

上の 5 行をコピーし，
1000 に一桁小さくする

課題５　課題４のプログラムを用いて，パルスの周波数（回転速度）を読み取れるように，AD 変換した値を LED で２進表示させなさい（stepMitu_veloci1.c 参照）．

以降の課題は，余裕のある学生のみ取り組みなさい．

課題６　課題５の回転速度制御プログラムによりステッピングモータの共振周波数（脱調ともいう）を求めよ．なお，低いステップ周波数より上昇させる時，突如制御不能となる点を共振周波数という．

課題７　課題５の回転速度制御プログラムによりステッピングモータの無負荷自起動周波数を求めよ．なお，共振周波数より高い周波数にしてから，モータ電源を接続し，回転可能な最も高い周波数を無負荷自起動周波数という．

課題８　可変抵抗の回転角の中点を境に正逆転し，回転角に比例した速度制御をさせなさい．

課題９　タクトスイッチを１度押すと，ちょうど２回転するプログラムを作成しなさい．

課題 10　パソコンのキーボードより移動パルス数を入力し，その数値に対応して回転（回転角）するプログラムを作成しなさい．

課題 11　可変抵抗の回転角と同一なモータ回転角となるような制御プログラムを作成しなさい．

(step01.c) → step_itu.c に名前を変えて保存

```
#include<3052f.h>
#include<wait.h>

int ad(int ch)
{
 AD.CSR.BIT.CH=ch;
 AD.CSR.BIT.ADST=1;
 while (AD.CSR.BIT.ADF==0);
 AD.CSR.BIT.ADF=0;
 return (AD.DRC >> 6);
}
main()
{
    long int i,j;
    int k,p1,c1,a[4]={0x24,0x28,0x18,0x14};

    P1.DDR= 0xff;       /* P1   8 LED */
    P2.DDR= 0x00;       /* P2   4 switch */
    P2.PCR.BYTE=0xff;  /* pull up */
    P3.DDR=0xff;

    init_wait01();

    while (1) {
        p1 = ad(2)-512;

        if (p1<-20) k--;
        if (p1>20)  k++;

       if (p1<0) {p1=-p1;}
       if ( k<0 ) k=3;
       if ( k>3 ) k=0;

       c1=520-p1;
       P3.DR.BYTE=a[k];

        wait(c1);
   }
}
```

７．　ＲＣサーボモータの原理と制御

ここでは，ラジコンカーや小型ヒューマノイドロボットなどのアクチュエータとして，一般的に用いられているラジコン用サーボモータ（以下，ＲＣサーボモータ）を用いて，これをマイコンによって制御する手法を学習します．ＲＣサーボは，入力するパルス幅に比例して角度制御が行える構造となっているため制御がしやすく，この技術を知っておくことは，上級学年での研究においても非常に有用です．

７－１．　ＲＣサーボモータの構造と動作原理

サーボモータ(Servo motor)とはサーボ機構において，位置，速度等を制御する用途に使用する角度検出器付のモータで，サーボモータ内部でフィードバック制御されているものが一般的です．メカトロニクスの分野では，ロボット用途などにも頻繁に使用されるようになりました．

構造は図６のようになっており，内部には DC モータが内蔵されています．この動力を減速ギヤでトルクを増幅し，ホーンの付いた出力軸を可動させます．　このシャフトは内部でポテンショメーター(可変抵抗器)と連動してあり、これで出力シャフトの回転角を検出し，制御回路に現在位置を伝達します。また, 外部よりある周期(幅)を持ったパルスが入力され，このパルスとポテーションメーターの位置の関係から、制御回路のフィードバック制御によりパルス幅に対して出力軸は一定の角度を保ちます。

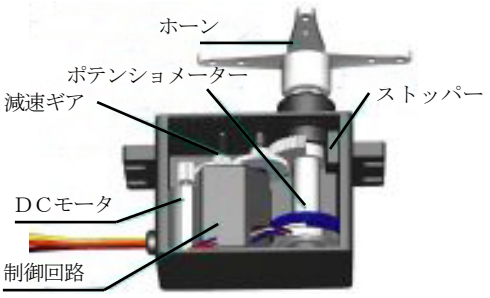


図６，ＲＣサーボ機構

従って，ＲＣサーボモータをコントロールするということは，サーボモータに入力するパルス幅を制御するということになります．市販のＲＣサーボモータを制御するパルス幅は規格化されており，どのメーカもほぼ同じような周期となっています．

次に,ＲＣサーボモータをコントロールしているパルスを図７に示します.ラジコン用の受信機がサーボモータへ出力するパルスの周期は，50Hz 程度の周期です．つまり，20msec に一度，パルスを出力することになります．要点は，実際にサーボモータの回転角を定めるのは，それぞれ 1 パルスのパルス幅ということです．１パルスのパルス幅は，0.5msec～2msec 程度変化し，この変化量が回転角度と比例します．従って，この 0.5msec～2msec 程度幅を持ったパルスをマイコンによって出力信号を制御できれば，マイコンによる RC サーボモータの制御が実現できます．

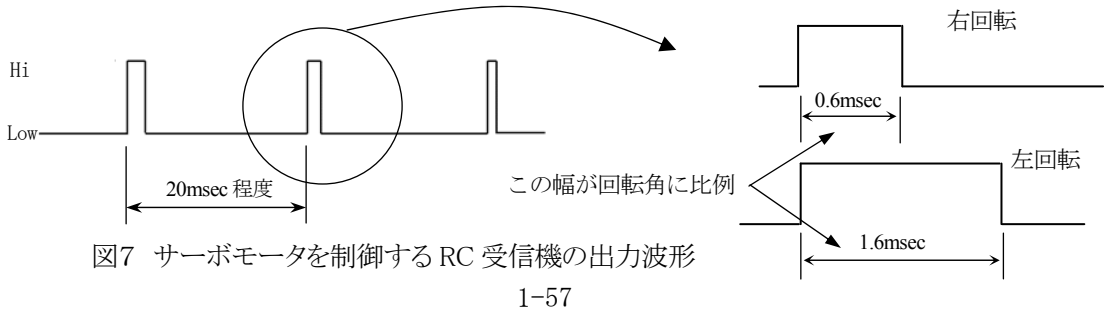


図7　サーボモータを制御する RC 受信機の出力波形

７－２．演習方法

図７の様に，H8 マイコンボードに RC サーボモータを接続します．その際，接続コネクタはどのようにも接続できるので，信号線とグランド線の極性を間違わないように接続してください．また,サーボモータも外部電源を用いて，乾電池 4 本(6V)で供給します．

【演習１】RC サーボモータの基本動作実験

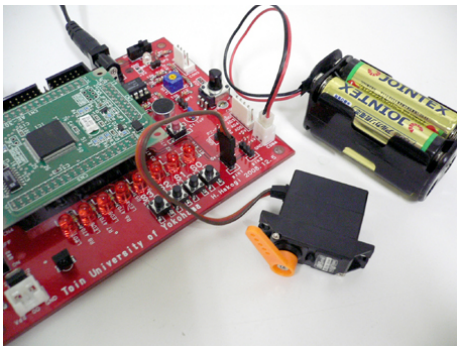
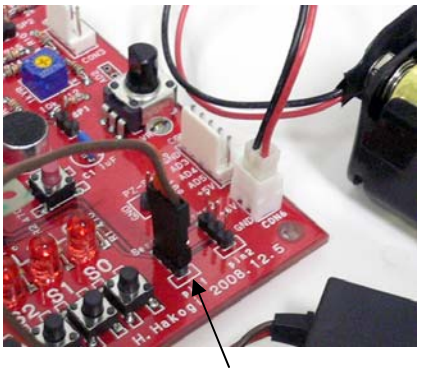


図７　サーボモータの接続

拡大図



サーボによって異なるので注意してください．
コードが白赤黒の場合，**手前が黒**
橙赤茶の場合，**手前が茶**となります．

課題１　RC サーボの制御はパルス幅の制御と同等なので，まず始めに，P35 で学習した for 文で遅延時間を作るプログラムを利用して，Prg8_1.c のプログラムを次のように修正し，実行してください．（これまでの資産を活用すること）

H8 マイコンボードの RC サーボ用接続端子は，ポート６の０と１に接続されていますので，P1 を P6 と書き換えれば良いのです．プログラムが上から流れて来て，１で立ち上がって，ある時間その状態を保って，その後，０出力で立ち下がる，というプロセスのプログラムにします．

（プログラム Prg8_1.c）

```
#include<3052f.h>
main()
{
  long int tim;
  P1.DDR=0xff;

  while (1){
    P1.DR.BIT.B3=0;
    for (tim=0 ; tim<2000000;tim++){}
    P1.DR.BIT.B3=1;
    for (tim=0 ; tim<2000000;tim++){}
  }
}
```

Servo1_1.c（保存ファイル名）

```
#include<3052f.h>
main()
{
     long int tim;
     P6.DDR= 0x0f; /* P6-0,P6-1 2ch servo */

     while (1){
          P6.DR.BIT.B0 = 1;
          for (tim=0 ; tim<5000;tim++){}
          P6.DR.BIT.B0 = 0;
          for (tim=0 ; tim<100000;tim++){}
     }
}
```

課題２　課題１のプログラムで，上側の for 文の 5000 を，9000 に変更して実行してください．同様に 6000，3000 と変更した場合，この３つの値が，どのようにサーボモータの動作に影響するのか，その様子とプログラムの関係を報告しなさい．

課題３　for 文での遅延時間を ITU による遅延時間に変更しなさい．これまで学習した，wait.h と init_wait(), wait(数値) ; の活用を忘れずにすること．保存ファイル名は，Servo1_3.c としてください．まずは，次ページのサンプルプログラムを参考にしながら作成し，完成した者は課題の指示に従って報告しなさい．

(h8_toin¥cc¥wait.h のファイルを変更する)

```
init_wait()
{
      ITU0.TCR.BYTE=0x21;   /* GRA compare ,   clock= 1/2 */
      ITU0.GRA=10000;       /* GRA 設定 1ms~20MHz/2/10000 */
}
init_wait01()
{
      ITU0.TCR.BYTE=0x21;
      ITU0.GRA=1000;
}
init_wait001()
{
      ITU0.TCR.BYTE=0x21;
      ITU0.GRA=100;
}
wait(int t)  /* t=100 →  0.1秒のウエイト  */
{
      int tim;
                ｝
```

ここの変更も忘れない

上の5行をコピーし，
125 に一桁小さくする

次に，課題１のプログラム本体を次のように修正し，実行します．このとき，**wait(128)；の中の数値は，５０～２５５の範囲**で変更しください．

(保存ファイル名は servo_itu01.c)

```
#include<3052f.h>
#include<wait.h>
main()
{
      long int tim;
      P6.DDR= 0x0f; /* P6-0,P6-1 2ch servo */
      init_wait001();
      while (1){
          P6.DR.BIT.B0 = 1;
          wait(128);    /* 50 ～ 255 の範囲*/
          P6.DR.BIT.B0 = 0;
          for (tim=0 ; tim<100000;tim++){}
      }
}
```

動作が確認できた者は，wait の数値(128)を 80 や 240 など，50～255 の範囲で任意の数値に書き換えて実行し，RC サーボモータの制御角の様子を確認し，その様子やプログラムとの関連などを報告書にまとめなさい．

以降の課題は，余裕のある学生のみ取り組みなさい.

課題４　ITU の利用と可変抵抗を使用し，サーボモータが右回転，左回転するように，抵抗の回転角の中点を境に角度制御するプログラムを，配布した VR_servo.c を変更し作成しなさい. 報告書には網掛け部分の説明を述べること.

（保存ファイル名は VR_ITU_servo.c とする）

```
#include<3052f.h>
#include <stdio.h>     /*  printf */
#include<wait.h>

int ad(int ch)
{
    AD.CSR.BIT.CH=ch;
    AD.CSR.BIT.ADST=1;
    while (AD.CSR.BIT.ADF==0);
    AD.CSR.BIT.ADF=0;

   return (AD.DRC >> 6);
}
main()
{
    long int cnt,tim;
    int p1,p2;
    float d0,d1;

    P1.DDR= 0xff;   /* P1-0 ~ P1-7  8 LED */
    P2.DDR= 0x00;   /* P2-0 ~ P2-3  4 switch */
    P6.DDR= 0x0f;   /* P6-0,P6-1   2ch RC servo */
    P1.DR.BYTE =0;
    init_wait001();

    while (1) {                          課題３の可動範囲の値
        p2 = ad(2);                      を考慮した計算式
        p1 = p2/5+50;
        printf("VR=%d  ¥n",p1);
        P1.DR.BIT.B3 = !P1.DR.BIT.B3;

        P6.DR.BIT.B0 = 1;
        wait(p1);
        P6.DR.BIT.B0 = 0;
        for (tim=0 ; tim<100000;tim++){}
    }
}
```

実験報告書について（第３週，第４週分）

・実験報告書は第３週と４週をまとめて，２週間分で提出する．

・実験報告書のタイトル（実験テーマ）は，
　　３・４週目：**「A／D 変換とアクチュエータ制御実験」**とする．

・報告書の内容は，次の項目について記載すること．
　1）実験の目的
　2）実験の原理
　3）実験で使用した機器
　4）実験内容
　5）それぞれの課題の回答や結果報告事項などを記載する
　6）実験結果の考察，または実験全体で得られた知見と考察する
　　（最後にまとめて記載する）
　7）実験の感想
　8）報告書をまとめるにあたって調べた文献や図書があれば，参考図書として書籍名，出版社，著者名等を全て記載する．

・報告書のコピーなどの不正行為は，写した者は無論の事，写された側も同罪として処断する（試験における不正行為と同等の扱い）．

・４週目の授業終了後から１週間以内に S105 実験室前の各担当教員のメールボックスに提出すること．

注意事項

- 実験の共同実験者の記入は不要とします．
- 実験は，全て個人毎（自分のボードのもの）に行うこと．
- マイコンの仕組み，制御方法や制御対象の仕組み等の理解を深めること，プログラムの意味を理解することが主な目的で，プログラム作成が目的ではありません．
  その点を留意して報告書を作成して下してください．