# 人間コミュニケーション学実験

# マイクロコンピュータ基礎

平成 20 年 4 月版

使用実験室　西６号館１階１０９号室
実験期間　３週（２００８年度は　A:火曜，B:土曜）

## I.　目的

実世界で動作するシステムを構築する方法を理解する．具体的にはマイクロコンピュータの外部入出力機能，および PC との通信機能を用い，計測と制御を含めた実験システムを構築する．

## II.　実験項目

１日目：マイクロコンピュータ H8
1.1.　LED の点滅
1.2.　ボタン入力の処理
1.3.　LCD 表示
1.4.　タイマー割り込み
1.5.　課題
２日目：外部からの入力を読み取る
2.1.　ハイパーターミナルによる読み出し
2.2.　AD 変換
2.3.　サーミスタ＋オペアンプを用いた温度表示
2.4.　加速度センサを用いた傾き検出
2.5.　課題
３日目：外部へ出力する
3.1.　ハイパーターミナルによる書き込み
3.2.　RC サーボモータの制御
3.3.　DA 変換
3.4.　自由課題

# III.　実験についての諸注意

- 事前準備に関する注意

実験では C 言語を用いる．C 言語を習得済みであることを前提としているため，予習しておくことが望まれる．

- 安全に関する注意

下記の注意点は生死にかかわることがあるので，日頃から注意して体で覚えておくこと．

✓ 実験機器や器具の配置に注意して，どのように配置すれば実験がやりやすいかを考慮して設定すること．
✓ 配線の接続は必ず電源を切った状態で行うこと．
✓ 電源を入れる場合は，配線に間違いのないことを確認してから行うこと．
✓ 実験終了後は電源を切った状態で配線を取り外すこと．
✓ 機器，器具が破損した場合は直ちに担当者に申し出ること．

- レポートに関する注意

✓ 実験専用ノートを一冊用意すること．
✓ 測定値の記録は表にするだけではなく，必ずグラフ用紙に測定点を記入しながら実験を進めること．実験の本質的誤りや計器の故障，メーターの読み間違いを防ぐだけではなく，再度の実験時間を節約できる．
✓ 実験中に気付いたことがあれば，随時ノートに記録しておくと実験報告書を書く時に役に立つことがある．
✓ おおむね二人一組で班を作るが，レポートは個別に提出すること．
✓ **実験で用いたソースコードを添付するため，実験終了後にプリントしておくこと．**
✓ **レポートの提出は次の実験日に必ず提出すること．**最後の実験の場合，原則として次週の同曜日に提出するが，別途指示がある場合はそれに従うこと．
✓ 指定された表紙を付けて提出すること．
✓ この他のレポートに関する諸注意は付録 G を参照すること．

- その他の注意

✓ プログラミング環境は 2 名に 1 セットであるが，実際のプログラミングが一人に集中するのは望ましくない．意識的に一日のうちで役割分担を交代するようにせよ．
✓ 参考文献を貸し出すが，次の実験グループのために必ず実験最終週の次の週の実験開始直前までに返却すること．

# IV.　実験内容

## 1. マイクロコンピュータ H8

### 1.1. 実験の準備

適当な場所に自分達のグループのフォルダを作成し，サンプルプログラムを所定のフォルダからコピーせよ．以後の実験ではすべてのプログラムの改変は自分のフォルダ内のファイルに対して行うこと．またレポートでは変更したサンプルプログラムを掲載する必要があるため，課題ごとに異なるファイル名で保存すること．

### 1.2. LED の点滅

サンプルプログラム(led.c)をコンパイル，マイコンへ書き込み，LED が点滅することを確認する．コンパイル，書き込み方法の詳細は付録 A を参照すること．

[課題1] サンプルプログラム中の

```
while(1) {...}
t=(t+1)%2;
```

が実現している機能を，なぜそうなるか分るようにそれぞれ説明せよ．

[課題2] サンプルプログラムを変更して，LED4 つのうち一つだけが点滅するようにせよ．なお 0 で点灯することに注意．

[課題3] 回路図（付録 E，図 15）中の LED 回路を参照し，なぜ 0 で点灯するのか考察せよ．

[課題4] サンプルプログラムを変更して，LED 4 つが右から順に点灯するようにせよ．

表 1 の対応表を参照し，LED が H8 マイコンのポート B の 0-3 番に接続されていることを確認すること．

### 1.3. ボタン入力の処理

サンプルプログラム(button.c)をコンパイル，マイコンへ書き込み，ボタン PA0 押下時に対応する LED が点灯することを確認する．

[課題5] 4 つのボタンを押した時に，対応する LED がそれぞれ点灯するようにせよ．なお押下時の論理は 0 となることに注意．

[課題6] 回路図（付録 E，図 15）中のボタン回路を参照し，なぜ押下時の論理が 0 になるのか考察せよ．

[課題7] サンプルプログラム中の入出力レジスタの設定部分を次のように変更せよ．

```
PB.DDR=0x00;
```

動作はどのようにかわったか．またそれはなぜか考察せよ．このような入出力レジスタの設定は多くのマイコンで採用されている．その理由を考察せよ．

（参考）上記入出力レジスタの設定のための変数（たとえば PB.DDR）はヘッダファイル 3052.h 中で構造体として定義されている（C:¥Program Files¥BestTech¥GCC Developer Lite¥TARGET¥3052F）．これを確認しておくこと．またこの構造体は内部に共用体(union)を持ち，ポートへのビットアクセスおよびバイトアクセスを可能としている(図 1)．

✓ ビットアクセス例：PB.DR.BIT.B0=0　　//LED 1 個の点灯

✓ バイトアクセス例：PB.DR.BYTE=0x0F　　//LED 4 個の同時点灯

```
struct st_pb {                                          // struct PB
             unsigned char      DDR;                    // PBDDR
             char               wk;                     //
             union {                                    // PBDR
                   unsigned char BYTE;                  //  Byte Access
                   struct {                             //  Bit  Access
                          unsigned char B7:1;           //    Bit 7
                          unsigned char B6:1;           //    Bit 6
                          unsigned char B5:1;           //    Bit 5
                          unsigned char B4:1;           //    Bit 4
                          unsigned char B3:1;           //    Bit 3
                          unsigned char B2:1;           //    Bit 2
                          unsigned char B1:1;           //    Bit 1
                          unsigned char B0:1;           //    Bit 0
                          }      BIT;                   //
                   }             DR;                    //
};                                                      //
```

図 1　ヘッダファイル 3052.h 中で定義されたデータアクセスのための構造体

### 1.4.　LCD（液晶表示器）表示

LCD（液晶表示器）に文字を表示する．サンプルプログラム(lcd.c)をコンパイル，マイコンへ書き込み，文字列が表示されることを確認する．

[課題8]　適当な C 言語のマニュアルを参照し，サンプルプログラム中の

**sprintf(s, "%d", m);**

が実現している機能を説明せよ．

[課題9]　サンプルプログラムを変更して，ボタンを押している間だけ表示されている数字が増加し続けるようにせよ．

[課題10] ボタンを押す度にこれまで押された回数が累積表示されるようにせよ．押し続けている間カウント回数を増加させないための工夫を考察し，実装せよ．

### 1.5.　タイマー割り込み

サンプルプログラム(timer.c)をコンパイル，マイコンへ書き込み，1 秒ごとに LED が点滅

することを確認する．
このサンプルプログラムでは，タイマー割り込み関数 init_imib0 が 10ms ごとに呼び出される．main 関数中の while ループは何も行っていない．もちろん while ループ内に処理を追加することもできる．

[課題11] サンプルプログラム中の

**PB.DR.BIT.B0 = ~PB.DR.BIT.B0;**

が実現している機能を，なぜそのようになるか分るように説明せよ．

[課題12] サンプルプログラムを変更し，LED 4 つが 0.5 秒おきに右から順に点灯するようにせよ．

[課題13] タイマー割り込み以外によく利用するマイコンの割り込み機能に外部割り込みがある．これがどのようなものか調べよ．

### 1.6. 1 日目総合課題

[課題14] これまでのプログラムを総合して，ボタン入力によるストップウオッチを作成せよ．ひとつのボタンが Start，Stop 機能を，もう一つのボタンが Clear 機能を持ち，100 分の 1 秒刻みで計測し，分の表示もできるようにせよ．
ただし LCD の表示には時間がかかるため，表示自体は 1 秒おきに更新し，Stop 時に 100 分の 1 秒刻みの値を表示するようにせよ．

## 2. 外部からの入力を読み取る

### 2.1. ハイパーターミナルによる読み出し

サンプルプログラム(serial_snd.c)をコンパイル，マイコンへ書き込み，PC でハイパーターミナル RealTerm を起動し，数字が送られ続けることを確認せよ．ハイパーターミナルの使用方法は付録 C を参照し，通信速度は 9600bps(bit per sec：一秒間に送信できるビット数)とせよ．ここで使用しているマイコンーPC 間のデータ通信手法をシリアル通信という．なお狭義のシリアル通信とは PC のシリアルポートを用いた通信のことであり，広義のシリアル通信とはデータの伝送を一本の信号線で行う通信方式全般を指す．

[課題15] 1 秒間に送信される行数を大まかに観察し，記録せよ（例えばリセットボタンを押してから 10 秒後の行番号を観察すればよい)．プログラム中では通信速度が"br9600"という定数で定義されている．この定数はヘッダファイル 3052.h 中で定義されている．これを"br19200"に変更し，さらにハイパーターミナルの通信速度を 19200bps としてみよ． 1 秒間に送信される行数はどのように変わったか．またこの変化を定量的に考察せよ．（ヒント：送信文字列に含まれる文字数（バイト数）をカウントする）

### 2.2. AD 変換

マイコンの AD 変換機能を用いて電圧の計測を行う．AD(Analog-Digital)変換とは，電圧のアナログ量のディジタルデータへの変換である(図 2)．

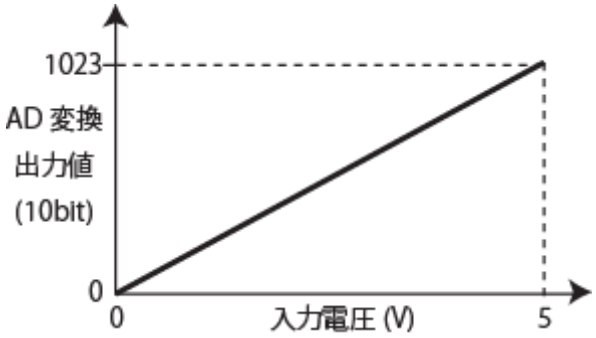


**図 2 AD 変換(10bit)**

サンプルプログラム(voltage.c)をコンパイル，マイコンへ書き込む．図 3 のように，H8 マイコン基板，計測用回路，外部電源を接続せよ．接続の向きに注意すること．
接続を確認した上でマイコンを起動し，電源のボリュームを最小（左回転）にし，電源を起動せよ． 電源のボリュームをゆっくりと上げていった時に，電圧が LCD 上で表示されていることを確認せよ．5V 以上にはしないこと．

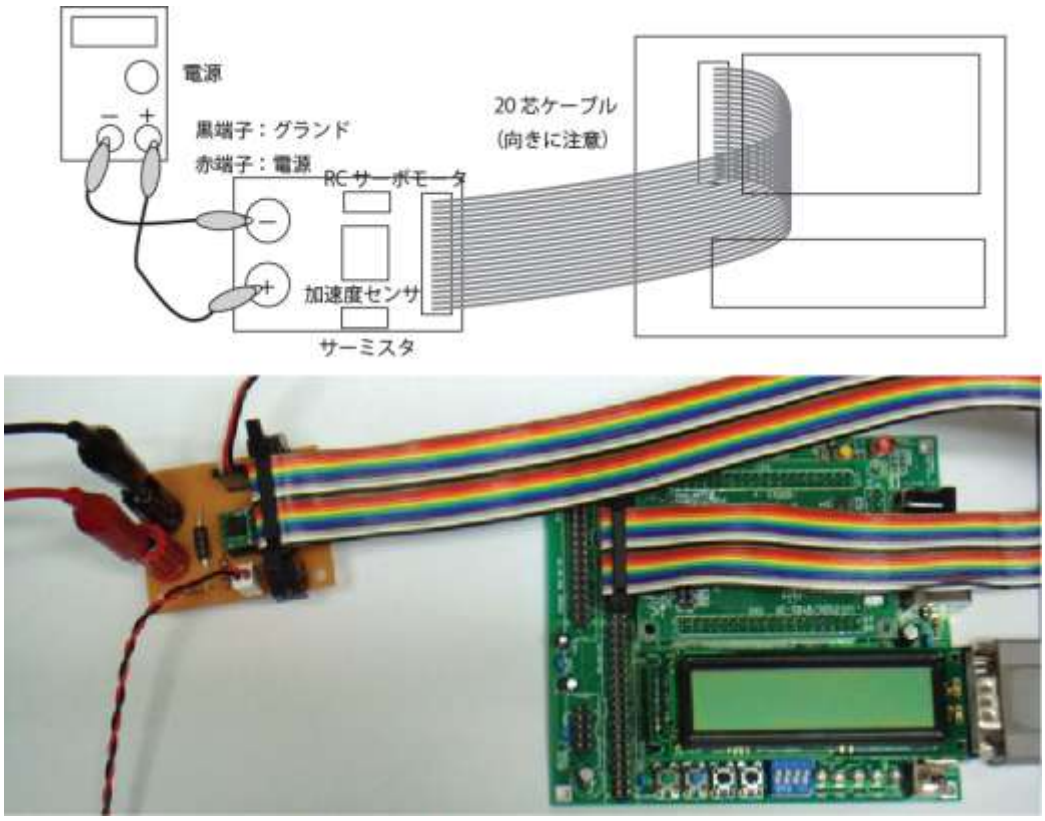


図 3　電圧計測実験接続図：ケーブルの向きに注意

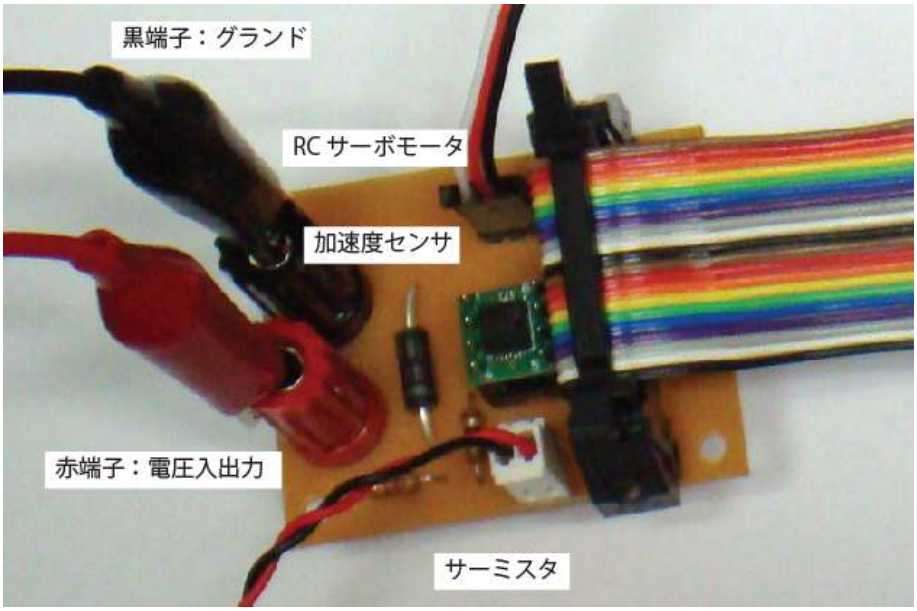


図 4　入出力基板

[課題16] サンプルプログラム中の

```
dVoltage = (double)usVoltage / 1024 * 5.0;
```

が実現している機能を，なぜそのようになるか分るように説明せよ．（ヒント：H8 マ

イコンの AD 変換器は 10 ビットである．すなわち 0～5V を 0～$2^{10}$ の値に変換する）

[課題17] サンプルプログラム voltage.c と serial_snd.c を統合し，ハイパーターミナルで電圧を観察できるようにせよ．

### 2.3. サーミスタを用いた温度表示

サーミスタを用いた温度計測を行う．利用するサーミスタは石塚電子製高精度サーミスタ 103JT である．これは温度に対して次のような抵抗変化を示す．

$$R_1 = R_2 \times \exp(B(1/T_1 - 1/T_2))$$

$T_1$,$T_2$：絶対温度(K)

$R_1$:温度 $T_1$ における抵抗値

$R_2$:基準温度 $T_2$ における抵抗値

B：定数(K)

この抵抗値変化は温度に対して非線形，かつ単調減少（温度が上がるにつれて抵抗値が下がる）であるが，図 5 のような回路で単調増加の電圧変化を得ることができる．

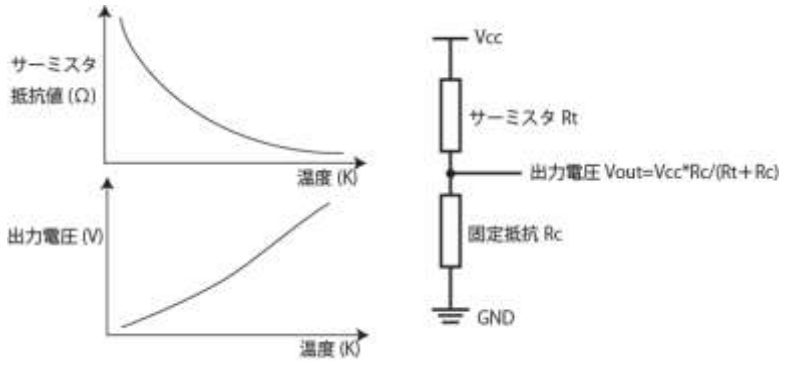


**図 5　温度変化に対するサーミスタの抵抗値変化および変換回路．**

[課題18] 用意された湯，氷水，および室温での出力電圧を記録せよ．計測には電圧計測プログラムを改変して用いよ（電圧入力ポートの番号が 6 番ではなく 3 番となる）．またこの結果をもとに，温度と出力電圧間の関係が直線的であると仮定し，出力電圧から温度を計算，出力するようなプログラムを作成せよ．そのうえでサーミスタを指でつまみ，指先の体温を測定せよ．結果は妥当か？妥当でないとすると理由は何か考察せよ．

なおこのように，測定値と物理量の関数を求める操作を較正（Calibration）と言い，センサを利用する際には必須の操作である．通常このような少数のデータをもとに較正してはならない．また直線でフィッティング出来るとは限らない．

## 2.4. 加速度センサを用いた傾き検出

加速度センサを用いた傾き計測を行う．利用する加速度センサは Kionix 社製 3 軸加速度センサモジュール KXM52-1050 である．これは次のようなスペックを持つ．

出力振幅（感度）＝Vdd/5(V/G)
0G 時の出力電圧＝Vdd/2(V)

出力振幅は，加速度に対する出力電圧を表す．Vdd/5(V/G)とは，例えば電源電圧が 5V の場合，1G(重力加速度)を加えられた際に 5/5=1V 出力があることを意味する．また 0G 時の出力電圧=Vdd/2(V)とは，たとえば電源電圧が 5V の場合，加速度が加わらなければ 2.5V の出力があることを意味する．

ただしこれらの値はあくまで目安であり，加速度センサによっても，同じセンサの出力軸によっても異なるため，較正する必要がある．

図 3 の接続のまま電源との接続を外し，サンプルプログラム(accel.c)をコンパイル，マイコンへ書き込む．x 軸，y 軸，z 軸方向の加速度に相当する出力電圧が LCD 上に表示されていることを確認せよ．

[課題19] 基板を手にとって，表，裏など傾きを変えてみよ．表示される値はどう変化したか．このことは，「加速度」を計測するセンサが，「傾き」を計測出来ていることを意味するが，これはなぜか論ぜよ．

[課題20] 以後 z 軸（鉛直方向）のみ考える．垂直に立てた分度器を用い，図 6 を参考に，センサを 0 度から 360 度まで，10 度刻みで傾け，その際の電圧を記録せよ．横軸が角度，縦軸が出力電圧となるグラフとしてまとめよ．またこの結果を論ぜよ．

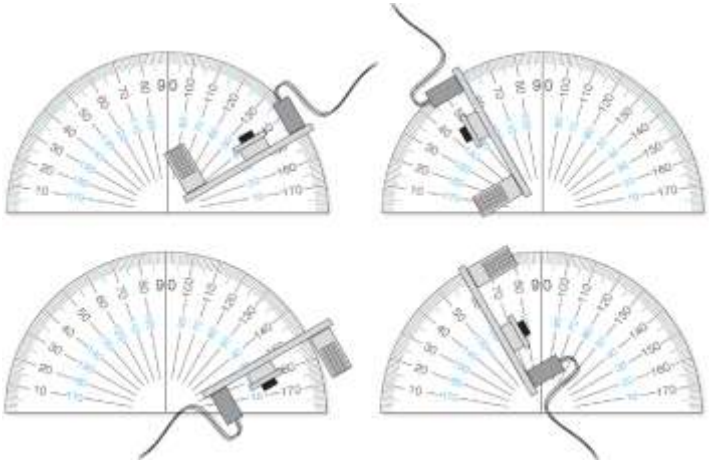

**図 6　分度器による角度指定（左上）30 度（右上）120 度（左下）210 度（右下）300 度**

## 2.5. 2 日目総合課題

[課題21] [課題 20]における 0 度，60 度，90 度，120 度，180 度の場合に着目する．このとき加速度センサに加わる重力加速度は 1G，0.5G，0G，-0.5G，-1G であると考えられる．これはなぜか簡潔に述べよ．このことを利用し，加速度センサに加わる重力加速度と出力電圧の間の関係をプロットした上で，出力電圧から加速度を算出する式を求めよ（これが較正作業である）．またこの結果を用い，現在の表示を変更し，重力加速度の単位で PC のハイパーターミナルに表示するようにせよ．

さらに余裕があれば傾きの角度を表示するようにせよ（ヒント：使用しているコンパイラは#include<math.h>によって数学ライブラリをインクルードすれば acos，asin，atan 等の関数を利用できる）

## 3. 外部へ出力する

### 3.1. ハイパーターミナルによる書き込み

サンプルプログラム(serial_rcv.c)をコンパイル，マイコンへ書き込み，PC でハイパーターミナル RealTerm を起動し，キーボード入力が LCD に表示されることを確認せよ．

[課題22] 受信用サンプルプログラム(serial_snd.c)と組み合わせ，打ち込んだキーが即座にハイパーターミナルの画面に表示されるようにせよ（例えば “A”と打ったら， “you pressed A”と表示されるようにする）．

### 3.2. RC サーボモータの制御

RC（ラジコン）サーボモータとは主にラジコンの操舵のために開発された駆動ユニットである．モータ，ギアボックス，制御回路から成り，比較的小型で強いトルクを出力し，制御が簡単なため，自作ロボットなどにも多く用いられている．

RC サーボモータの制御信号は図 7 のようなパルスである．通常，約 20ms おきに送られるパルスの幅が指定角度となる．パルス幅は通常 1.5ms を中心とし，1～2ms 程度である．

サンプルプログラム(rc_servo.c)をコンパイル，マイコンへ書き込み，RC サーボモータを図 8 のように接続せよ．

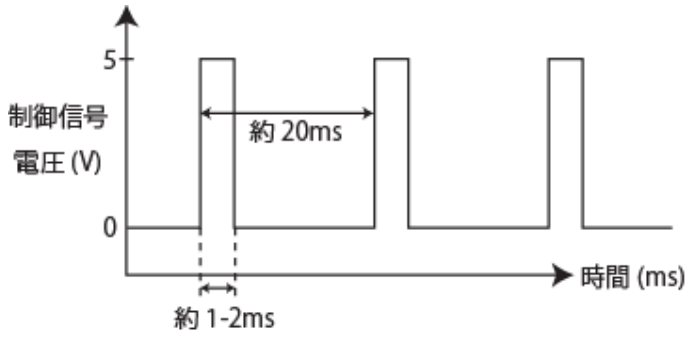


図 7　RC サーボモータの制御信号

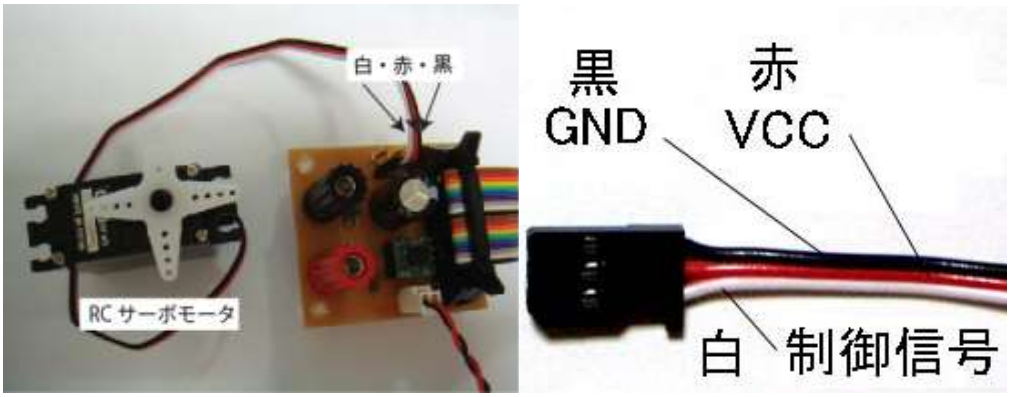


図 8　RC サーボモータの接続（向きに注意）

[課題23] 図 9 のように基板上のテストピンにプローブを接続し，オシロスコープを用いて望ましいパルス波形が出力されていることを確認せよ（RC サーボモータは外さなくて良い）．オシロスコープの画面を印刷し，レポートに添付すること．オシロスコープの使い方は付録 D を参照すること．

電源を切り，サーボモータの出力軸を適当に手でまわし，電源を投入せよ．どのような挙動を示したか．

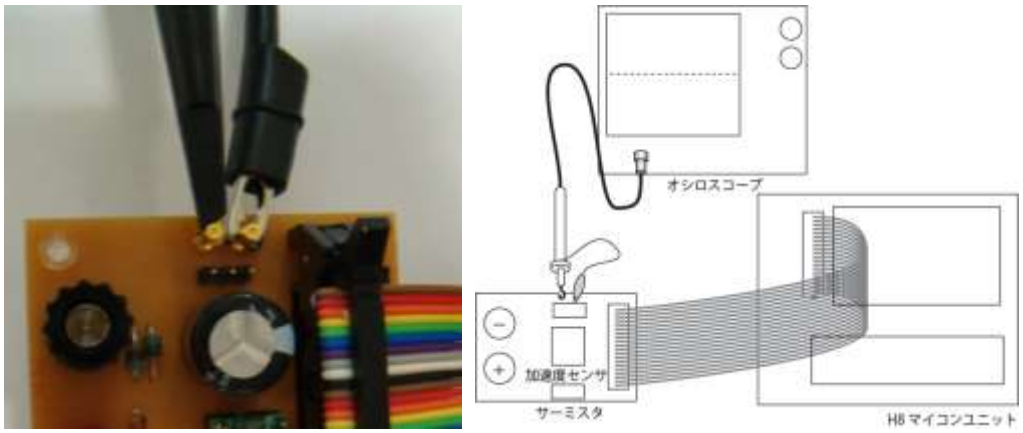


**図 9　RC サーボモータ制御信号観察のためのオシロスコープ用プローブ接続**

[課題24] 上記プログラムを改編し，また[課題 5]のプログラム(button.c)を参考に，ボタンを押すことによって RC サーボモータの回転角度を変更できるようにせよ．（ヒント：main 関数中の while 文中でボタン入力に応じてパルス幅を変更すればよい）オシロスコープで観察し，望ましい波形が出ていることを確認しつつ実験すること．

### 3.3.　DA 変換を用いた音生成

H8 マイコンには DA 変換出力がある．DA(Digigal-Analog)変換とは，数値を指定することによってアナログ電圧を出力する機能である（この機能のないマイコンは多い）．H8 マイコンの場合，AD 変換ポートと DA 変換ポートは共有されており，プログラムの設定によって機能を切り替えることが出来る．DA 変換の精度は 8bit であり，0 から 255 までの数値を設定することで 0 から 5V の電圧を出力できる(図 10).

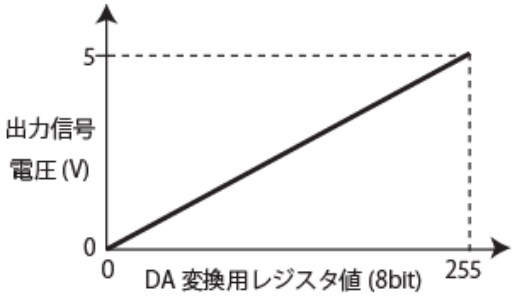


**図 10　　DA 変換(8bit)**

サンプルプログラム(da.c)をコンパイル，マイコンへ書き込み，オシロスコープを図 11 のように接続せよ．正弦波が出力されていることを確認せよ．

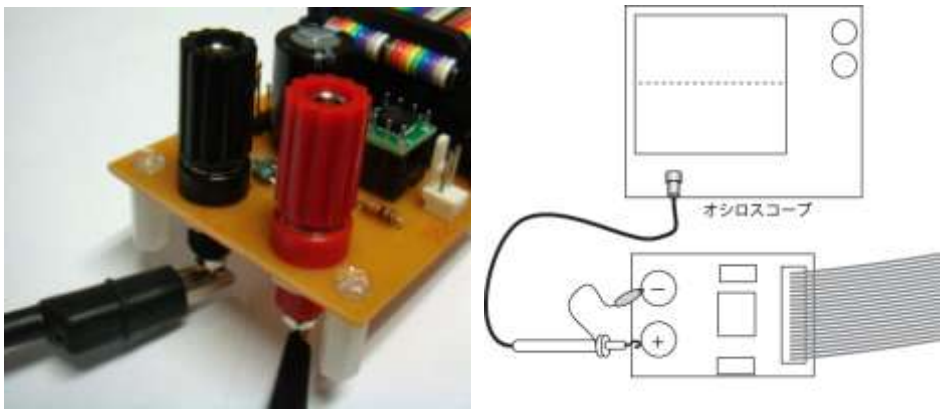


**図 11　DA 出力観察のためのオシロスコープ用プローブの接続**

さらに，出力端子に図 12 のようにクリスタルイヤフォンを接続し，音を聞いてみよ．クリスタルイヤフォンは通常のイヤフォンとは異なり圧電素子が用いられており，少ない電流で音を発生させることができる．


**図 12　クリスタルイヤフォンの接続**

プログラム da.c は正弦波を出力するプログラムである．実験で用いている C コンパイラは通常の sin 関数もサポートしているが，計算に非常に時間がかかるため，ここでは正弦波一周期を 64 分割した配列 sin_data を用意し，0.04ms 周期で生じるタイマー割り込みの度に配列の次のインデックス値を出力している(図 13)．このように事前の計算によって用意した配列を用いる手法をテーブルルックアップ(Table Lookup)方式と呼ぶ．

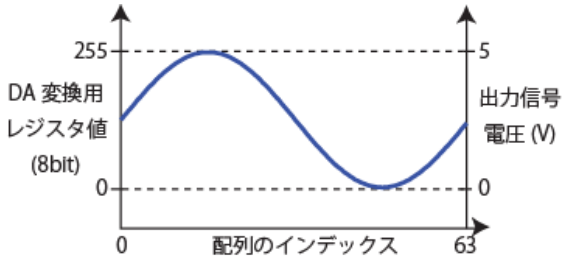


**図 13　正弦波配列 sin_data**

[課題25] オシロスコープで観察される波形を印刷し，振幅(V)，周期(s)を記録せよ．また周期から計算できる周波数はいくらか（周波数=1/周期）．さらに本プログラムにおいて，なぜこの周期となるのかを数値的に説明せよ．（ヒント：タイマー割り込みの周期と配列のサイズに着目）

[課題26] プログラムを改編し，正弦波の周波数を２倍にせよ（ヒント：タイマー割り込み中の iCount の増加量に着目．現在は 1 ずつ増えている）．オシロスコープの波形を印刷し，周期(s)を記録せよ．またクリスタルイヤフォンで音を聞いてみよ．

[課題27] プログラムをさらに改編し，[課題 5]のプログラム(button.c)を参考に，基板上のボタンを押すことで出力される音の高さが変化するようにせよ． クリスタフイヤフォンで音を聞き，望み通りの結果となっていることを確認せよ．

### 3.4. 総合課題

[課題28] [課題 24]のプログラムを改編し，サンプルプログラム serial_rcv.c と統合し，ハイパーターミナルから RC サーボモータの回転角度を指定できるようにせよ.（例えば a，b，c と入力したらそれぞれ対応する角度に回転すればよい．文字の比較は例えば if(c==’a’)のようになる．Switch 文を使っても良い）

[課題29] 何か面白いものを作成せよ．これまでに学んだ RC サーボモータ，スピーカ，加速度センサ，サーミスタ，ボタン，LCD,LED 等をすべて使って良い．

（例１） PC のキーボードを用いたピアノ

（例２） 温度センサと RC サーボモータを用いたアナログ温度計

（例３） 温度センサと LED による「さわると明るくなる」インタフェース

# V.　付録

## A　GCC Developer Lite および H8 Write Turbo の使用方法

(1) GCC Developer Lite による編集

(ア) GCC Developer Lite を立ち上げ，「ファイル」⇒「開く」でソースを開く．

(イ) 編集する．多くの場合「ファイル」⇒「名前を付けて保存」で名前を変えて保存．

(ウ) 「コンパイル」⇒「ビルド」を行う．エラーがある場合下に表示される．

(エ) ソースファイルのあるフォルダに，拡張子「mot」のファイルが出来ていることを確認する．必要な場合「ファイル」⇒「印刷」でソースを印刷する．

(2) H8 マイコンの準備（図 14）

(ア) PC とシリアルケーブルで接続されていることを確認する．

(イ) 電源が入っていなければ入れる．

(ウ) ボード右下のスイッチを上方向に切り替える．

(エ) ボード右上のリセットボタンを押す．これにより「書き込みモード」となる．

(3) H8 Write Turbo による書き込み

(ア) 拡張子「mot」のファイルをダブルクリックする．約 10～30 秒で書き込み終了のメッセージが表示される．書き込めないときは H8 マイコンの設定を見直し，リセットボタンを押して再度試みる．

(4) H8 マイコンの駆動（図 14）

(ア) 電源が入っていなければ入れる．

(イ) ボード右下のスイッチを下方向に切り替える．

(ウ) ボード右上のリセットボタンを押す．これにより「動作モード」となる．

(エ) リセットボタンを押すたびにプログラムは最初から動作する．

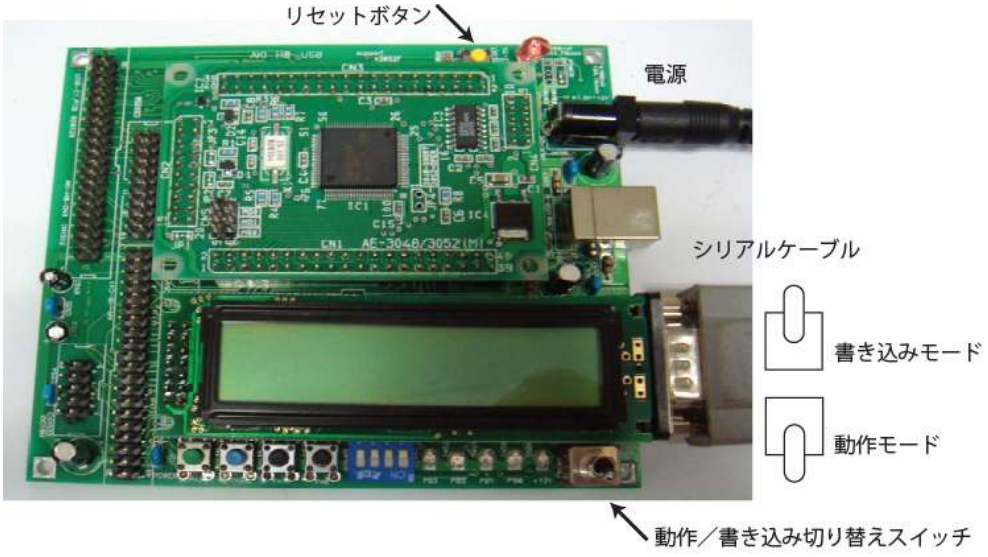


図 14　動作／書き込みモード切り替えとリセットボタン

## B　ソフトウエアの設定

以下は GCC Developer Lite と H8 Write Turbo の設定である．実験中は変更する必要はないが，自分で開発環境を構築する場合にはこれらの情報が必要となる．

- GCC Developer Lite の設定
  - ツール⇒GCC オプション⇒「設定リスト」のプルダウンメニューで，「H8/3052F 内臓フラッシュ ROM」とする．
  - ツール⇒GCC オプション⇒etc の「Objcopy ツール」「出力ファイルタイプ」を MOT とする．

（参考）今回使用するキットは本来 CPU モード 6 で動く（外付 RAM が有効になる）．ただし実験では書き込み作業の簡略化のため内蔵 RAM のみ使用するモード 7 を用いている．

- H8WriteTurbo の設定
  - H8/3052F とする．
  - COM ポートを指定する（ポート番号はデバイスマネージャー等で前もって確認）．
  - 速度は 115200bps に設定．
  - チェックボックスは全てチェック．
  - MOT ファイルへの関連付けを行う（クリックしておく）．

## C　RealTerm の使用方法

(1) RealTerm を立ち上げる．

(2) 通信開始

(ア) 「Port」タブを開く．

(イ) Baud（通信速度）が H8 マイコンのプログラムと同じであることを確認する．例えば 9600 に設定する．

(ウ) これ以外の設定は変える必要はないが, Parity=None, Data Bits=8, Stop Bits=1, Hardware Flow Control=None，Port=1 である．

(エ) 「Open」ボタンを押す.

(3) 文字列の受信

(ア) 何もしなくても受信した文字は画面に表示される．

(イ) 文字化けが起きているときには Baud(通信速度)の設定を見直し，「Change」ボタンを押す.

(4) 文字列の送信

(ア) 「Send」タブを開く．

(イ) 送信したい文字を入力し，「Send ASCII」ボタンを押す．これにより文字が送信される．

注意：RealTerm と書き込みソフト H8 Write Turbo は同じシリアルポートを使用するため，

書き込みを行う際には RealTerm の接続を切断しておく必要がある．「Port」タブを開き，「Open」ボタンを再度押すことで切断される．

## D　オシロスコープの使用方法

オシロスコープの取扱説明書を参照．簡単にまとめたオシロスコープの使い方を１枚用意している．

(1) 電源スイッチ（ＳＷ）のＰＯＷＥＲボタンを押して，電源を入れる．もう一度ボタンを押すと電源が切れる（ＯＦＦ）．

(2) 時間軸：液晶表示の水平軸の１目盛りが何秒に相当するか読み取れようになること．

(3) 電圧軸：垂直軸の１目盛が何ボルトに相当するか読み取れるようになること．

E　回路図

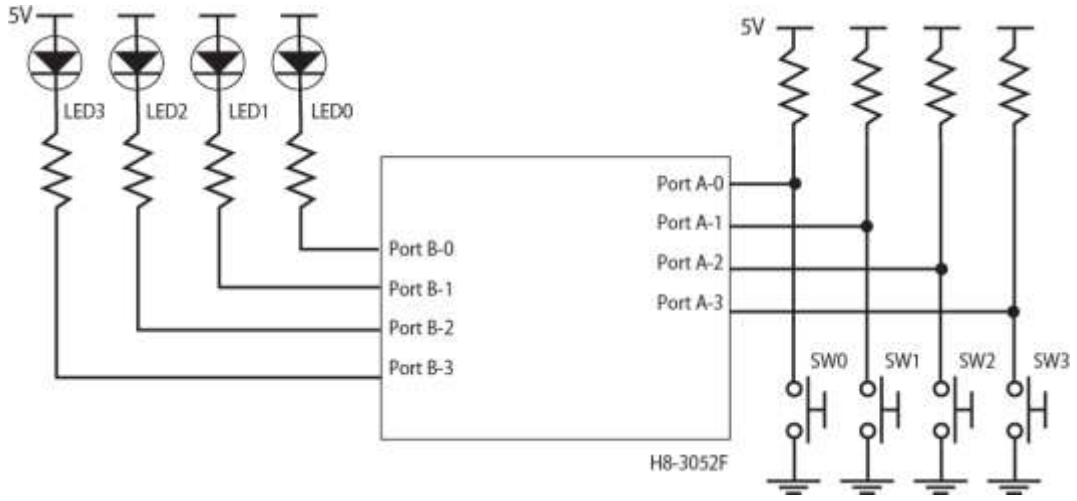


図 15 H8 ボード回路（LED，スイッチのみ抜粋）

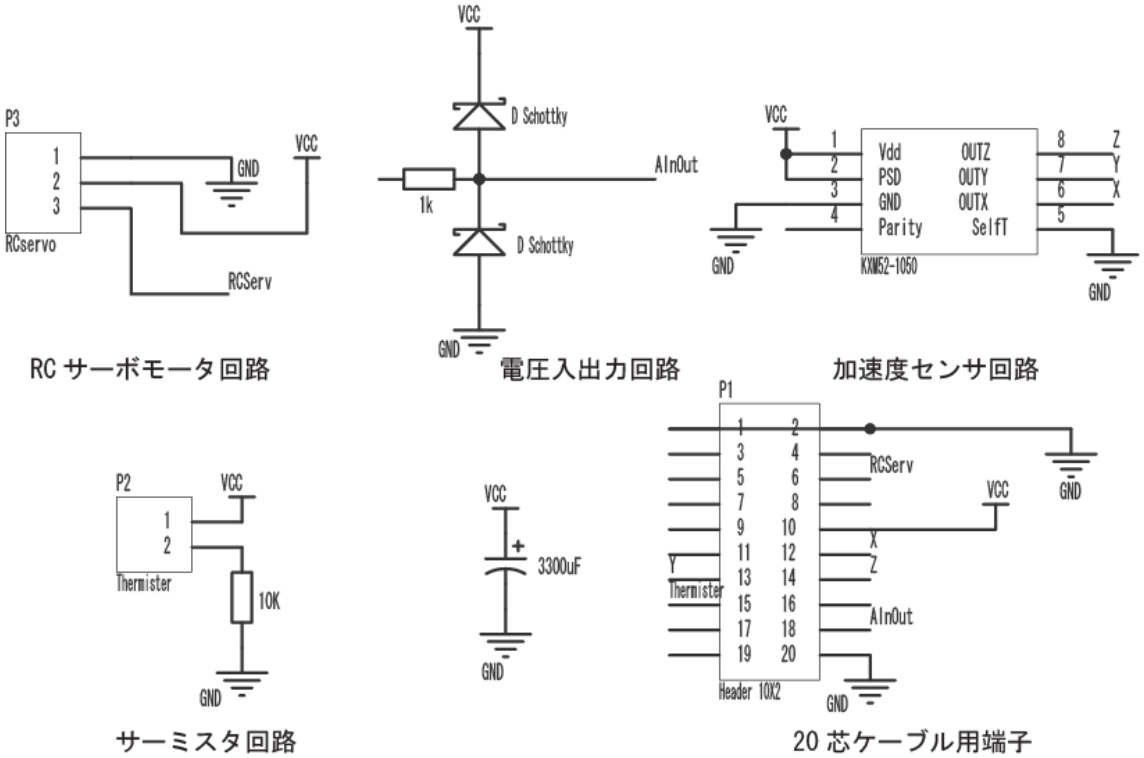


図 16 入出力基板回路

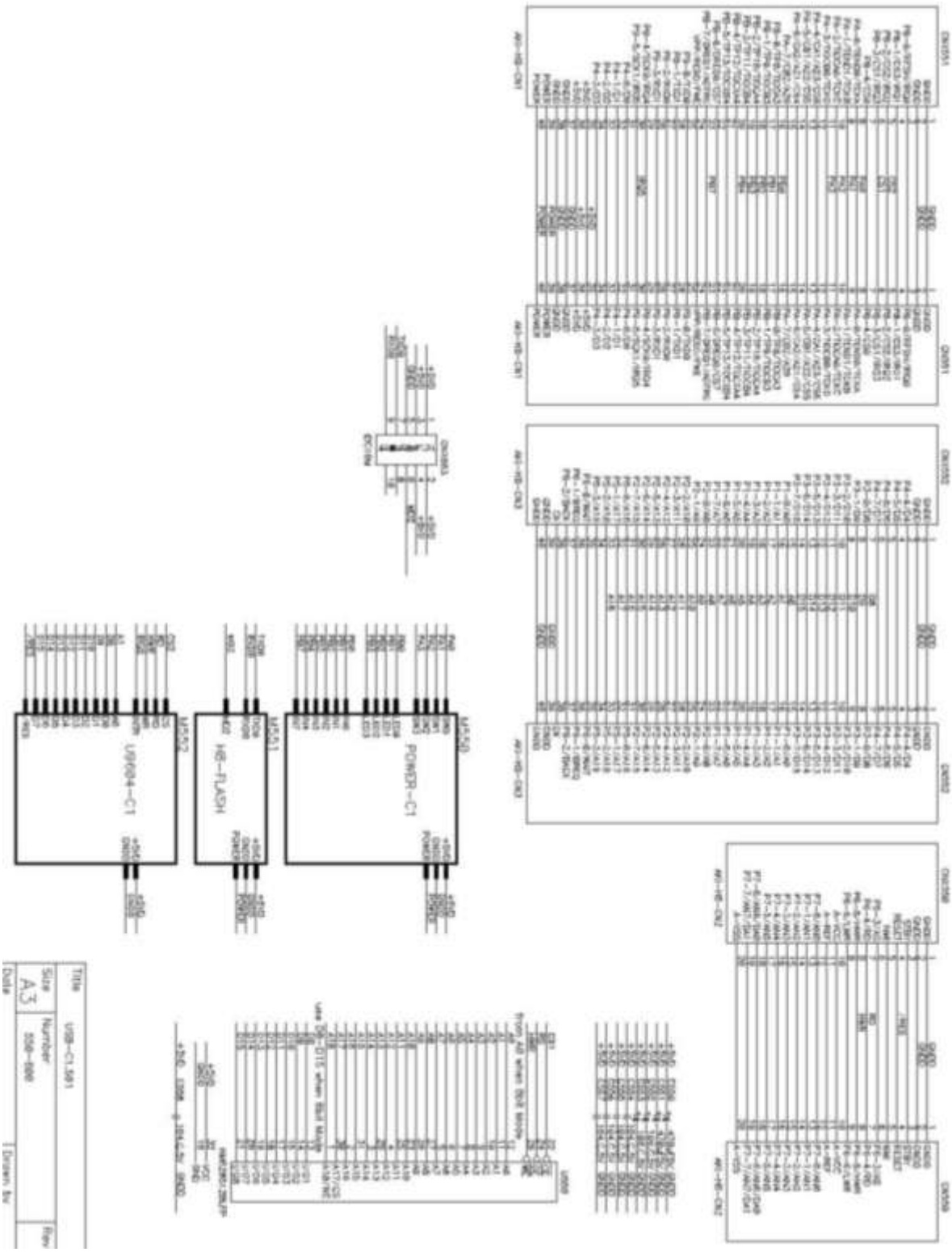

図 17　H8 マイコンマザーボード回路図(1/4)

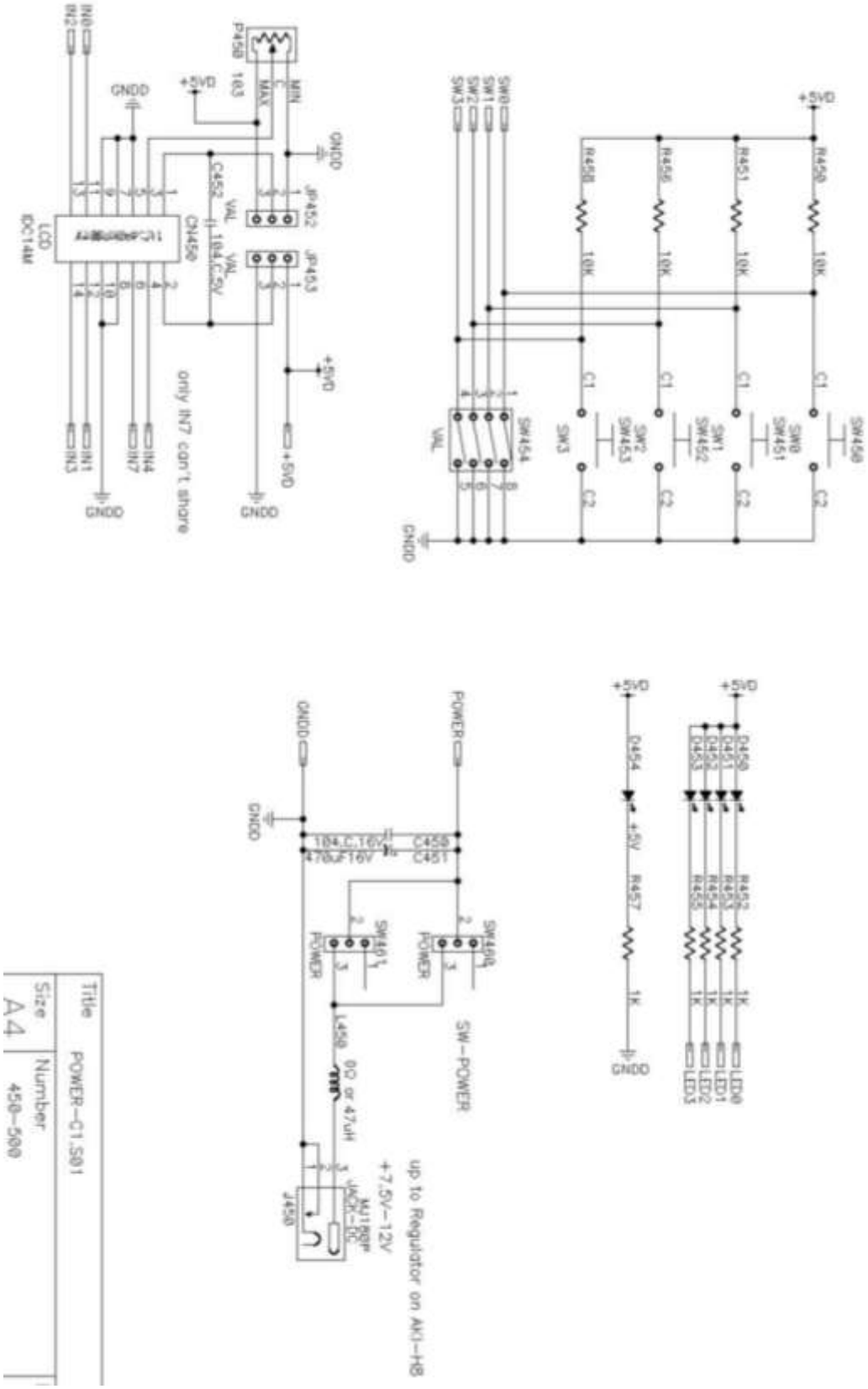


図 18　H8 マイコンマザーボード回路図(2/4)

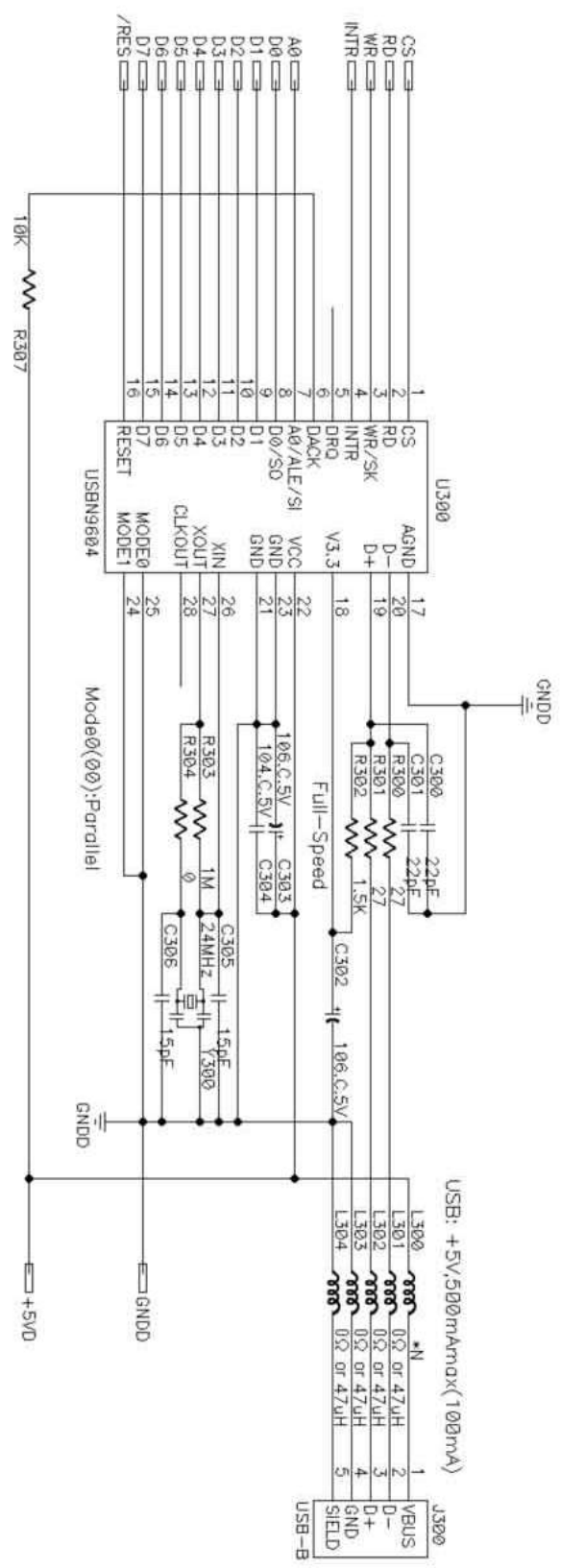


図 19　H8 マイコンマザーボード回路図(3/4)

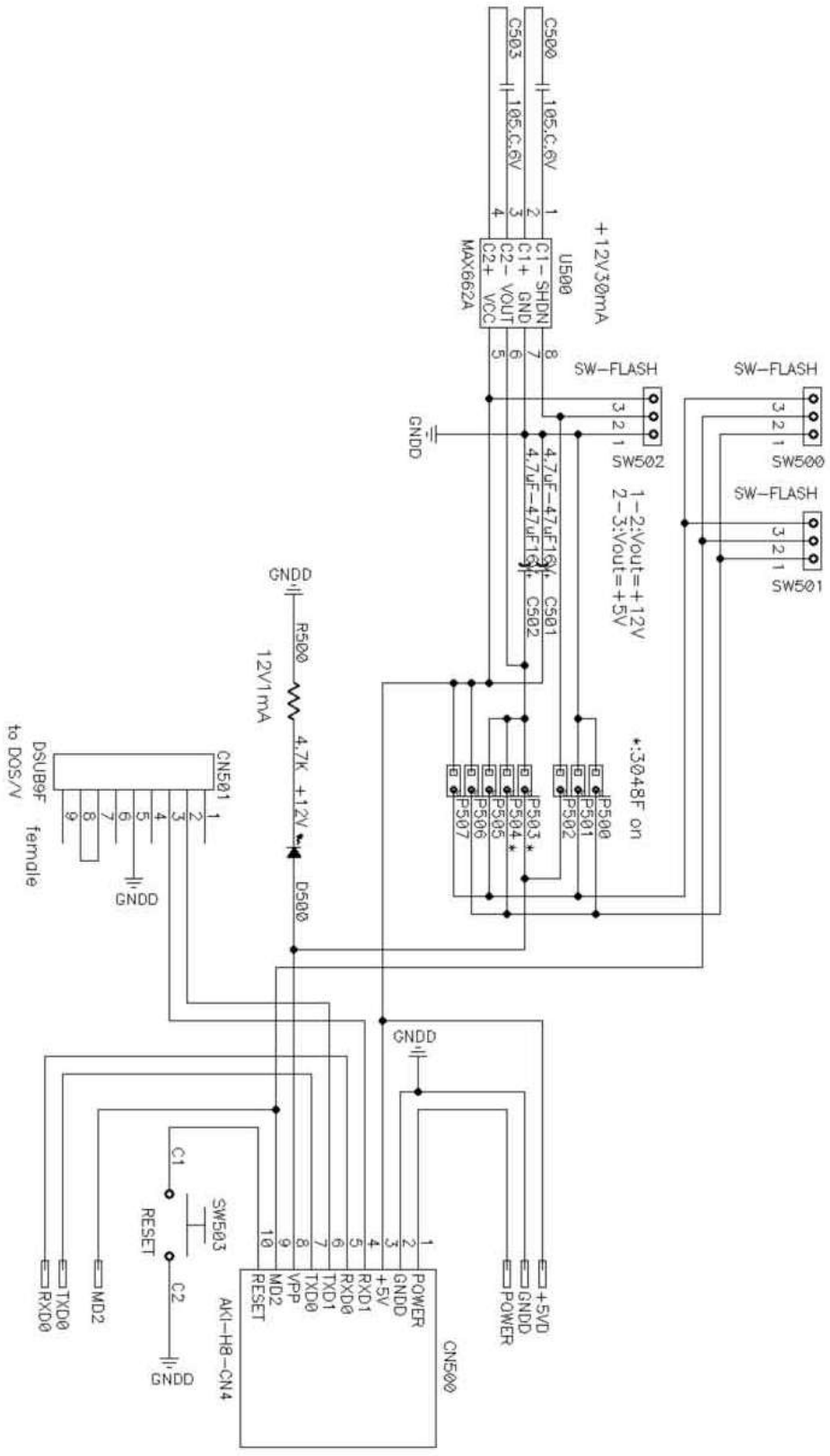


図 20　H8 マイコンマザーボード回路図(4/4)

## F　H8/3052F と外部デバイスの割り付けおよび拡張コネクタ端子表

実験で使用した AKI-H8/3052F マイコンキットの仕様をまとめる.

表 1 は H8-3052F のポートに割り当てられた外部デバイスに関する情報である．例えば LED が H8-3052F の PB-0 から PB-3 に，スイッチが PA-0 から PA-3 に接続されていることがわかる. LED やスイッチを用いたプログラミングはこの情報をもとに行う必要がある.

**表 1　H8-3052F と外部デバイスの割り付け**

| ポート名 | 外部デバイス | |
|---|---|---|
| P1-0 | SRAM-A0 | |
| P1-1 | SRAM-A1 | USBN9604-A0 |
| P1-2 | SRAM-A2 | |
| P1-3 | SRAM-A3 | |
| P1-4 | SRAM-A4 | |
| P1-5 | SRAM-A5 | |
| P1-6 | SRAM-A6 | |
| P1-7 | SRAM-A7 | |
| ポート名 | 外部デバイス | |
| P2-0 | SRAM-A8 | |
| P2-1 | SRAM-A9 | |
| P2-2 | SRAM-A10 | |
| P2-3 | SRAM-A11 | |
| P2-4 | SRAM-A12 | |
| P2-5 | SRAM-A13 | |
| P2-6 | SRAM-A14 | |
| P2-7 | SRAM-A15 | |
| ポート名 | 外部デバイス | |
| P3-0 | SRAM-D8 | USBN9604-D0 |
| P3-1 | SRAM-D9 | USBN9604-D1 |
| P3-2 | SRAM-D10 | USBN9604-D2 |
| P3-3 | SRAM-D11 | USBN9604-D3 |
| P3-4 | SRAM-D12 | USBN9604-D4 |
| P3-5 | SRAM-D13 | USBN9604-D5 |
| P3-6 | SRAM-D14 | USBN9604-D6 |
| P3-7 | SRAM-D15 | USBN9604-D7 |

| ポート名 | 外部デバイス | |
|---|---|---|
| P5-0 | SRAM-A16 | |
| P5-1 | SRAM-A17/CS | |
| P5-2 | SRAM-A18/NC | |
| ポート名 | 外部デバイス | |
| P6-4 | SRAM-OE | USBN9604-RD |
| P6-5 | SRAM-WE | USBN9604-WR |
| ポート名 | 外部デバイス | |
| P8-2 | | USBN9604-CS |
| P8-3 | SRAM-CE | |
| ポート名 | 外部デバイス | |
| P9-1 | RS232C-CH1 | |
| P9-3 | RS232C-CH1 | |
| P9-5 | | USBN9604-INTR |
| ポート名 | 外部デバイス | |
| PA-0 | SW0(PA0) [0=ON] | |
| PA-1 | SW1(PA1) [0=ON] | |
| PA-2 | SW2(PA2) [0=ON] | |
| PA-3 | SW3(PA3) [0=ON] | |
| ポート名 | 外部デバイス | |
| PB-0 | LED0(PB0) [0=ON] | LCD-DB4 |
| PB-1 | LED1(PB1) [0=ON] | LCD-DB5 |
| PB-2 | LED2(PB2) [0=ON] | LCD-DB6 |
| PB-3 | LED3(PB3) [0=ON] | LCD-DB7 |
| PB-4 | | LCD-RS |
| PB-7 | | LCD-E |
| | | |

※LCD と LED を駆動するピンは重複しているので， LCD 表示中に同時に LED も光る.

※プッシュスイッチとディップスイッチは同じ割り当てである.

表 2 は拡張コネクタの端子表である．この情報は実際に外部回路を作成する場合に重要であり，表 1 のポートと外部回路を接続する場合にどのコネクタピンに接続すればよいかの情報を提供する．ただし実験では回路はすでにほぼ作成されているのでこの情報を使うことはない．

**表 2 AKI-H8/3052F 用拡張コネクタ端子表**

| AKI-H8/3052F | 名称 | 拡張コネクタピン |
|---|---|---|
| CN1-1 | GND | CNX551-1 |
| CN1-2 | GND | CNX551-2 |
| CN1-3 | P8-0 | CNX551-3 |
| CN1-4 | P8-1 | CNX551-4 |
| CN1-5 | P8-2 | CNX551-5 |
| CN1-6 | P8-3 | CNX551-6 |
| CN1-7 | P8-4 | CNX551-7 |
| CN1-8 | PA-0 | CNX551-8 |
| CN1-9 | PA-1 | CNX551-9 |
| CN1-10 | PA-2 | CNX551-10 |
| CN1-11 | PA-3 | CNX551-11 |
| CN1-12 | PA-4 | CNX551-12 |
| CN1-13 | PA-5 | CNX551-13 |
| CN1-14 | PA-6 | CNX551-14 |
| CN1-15 | PA-7 | CNX551-15 |
| CN1-16 | PB-0 | CNX551-16 |
| CN1-17 | PB-1 | CNX551-17 |
| CN1-18 | PB-2 | CNX551-18 |
| CN1-19 | PB-3 | CNX551-19 |
| CN1-20 | PB-4 | CNX551-20 |
| CN1-21 | PB-5 | CNX551-21 |
| CN1-22 | PB-6 | CNX551-22 |
| CN1-23 | PB-7 | CNX551-23 |
| CN1-24 | VPP/RES0 | CNX551-24 |
| CN1-25 | P9-0 | CNX551-25 |
| CN1-26 | P9-1 | CNX551-26 |
| CN1-27 | P9-2 | CNX551-27 |
| CN1-28 | P9-3 | CNX551-28 |
| CN1-29 | P9-4 | CNX551-29 |
| CN1-30 | P9-5 | CNX551-30 |
| CN1-31 | P4-0 | CNX551-31 |
| CN1-32 | P4-1 | CNX551-32 |

| AKI-H8/3052F | 名称 | 拡張コネクタピン |
|---|---|---|
| CN3-1 | GND | CNX552-1 |
| CN3-2 | GND | CNX552-2 |
| CN3-3 | P4-4 | CNX552-3 |
| CN3-4 | P4-5 | CNX552-4 |
| CN3-5 | P4-6 | CNX552-5 |
| CN3-6 | P4-7 | CNX552-6 |
| CN3-7 | P3-0 | CNX552-7 |
| CN3-8 | P3-1 | CNX552-8 |
| CN3-9 | P3-2 | CNX552-9 |
| CN3-10 | P3-3 | CNX552-10 |
| CN3-11 | P3-4 | CNX552-11 |
| CN3-12 | P3-5 | CNX552-12 |
| CN3-13 | P3-6 | CNX552-13 |
| CN3-14 | P3-7 | CNX552-14 |
| CN3-15 | P1-0 | CNX552-15 |
| CN3-16 | P1-1 | CNX552-16 |
| CN3-17 | P1-2 | CNX552-17 |
| CN3-18 | P1-3 | CNX552-18 |
| CN3-19 | P1-4 | CNX552-19 |
| CN3-20 | P1-5 | CNX552-20 |
| CN3-21 | P1-6 | CNX552-21 |
| CN3-22 | P1-7 | CNX552-22 |
| CN3-23 | P2-0 | CNX552-23 |
| CN3-24 | P2-1 | CNX552-24 |
| CN3-25 | P2-2 | CNX552-25 |
| CN3-26 | P2-3 | CNX552-26 |
| CN3-27 | P2-4 | CNX552-27 |
| CN3-28 | P2-5 | CNX552-28 |
| CN3-29 | P2-6 | CNX552-29 |
| CN3-30 | P2-7 | CNX552-30 |
| CN3-31 | P5-0 | CNX552-31 |
| CN3-32 | P5-1 | CNX552-32 |

| | | |
|---|---|---|
| CN1-33 | P4-2 | CNX551-33 |
| CN1-34 | P4-3 | CNX551-34 |
| CN1-35 | +5V | CNX551-35 |
| CN1-36 | +5V | CNX551-36 |
| CN1-37 | GND | CNX551-37 |
| CN1-38 | GND | CNX551-38 |
| CN1-39 | POWER | CNX551-39 |
| CN1-40 | POWER | CNX551-40 |

| | | |
|---|---|---|
| CN3-33 | P5-2 | CNX552-33 |
| CN3-34 | P5-3 | CNX552-34 |
| CN3-35 | P6-0 | CNX552-35 |
| CN3-36 | P6-1 | CNX552-36 |
| CN3-37 | P6-2 | CNX552-37 |
| CN3-38 | CK | CNX552-38 |
| CN3-39 | GND | CNX552-39 |
| CN3-40 | GND | CNX552-40 |

| AKI-H8/3052F | 名称 | 拡張コネクタピン |
|---|---|---|
| CN2-1 | GNDD | CNX550-1 |
| CN2-2 | GNDD | CNX550-2 |
| CN2-3 | STBY | CNX550-3 |
| CN2-4 | RESET | CNX550-4 |
| CN2-5 | NMI | CNX550-5 |
| CN2-6 | P6-3 | CNX550-6 |
| CN2-7 | P6-4 | CNX550-7 |
| CN2-8 | P6-5 | CNX550-8 |
| CN2-9 | P6-6 | CNX550-9 |
| CN2-10 | AVCC | CNX550-10 |
| CN2-11 | AREF | CNX550-11 |
| CN2-12 | P7-0 | CNX550-12 |
| CN2-13 | P7-1 | CNX550-13 |
| CN2-14 | P7-2 | CNX550-14 |
| CN2-15 | P7-3 | CNX550-15 |
| CN2-16 | P7-4 | CNX550-16 |
| CN2-17 | P7-5 | CNX550-17 |
| CN2-18 | P7-6 | CNX550-18 |
| CN2-19 | P7-7 | CNX550-19 |
| CN2-20 | AVSS | CNX550-20 |

| AKI-H8/3052F | 名称 | 拡張コネクタピン |
|---|---|---|
| CN4-1 | +5V | CNX553-3 |
| CN4-2 | +5V | CNX553-3 |
| CN4-3 | +5V | CNX553-3 |
| CN4-4 | +5V | CNX553-3 |
| CN4-5 | GND | CNX553-2 |
| CN4-6 | MD2 | CNX553-9 |
| CN4-7 | TXD0 | CNX553-7 |
| CN4-8 | 未接続 | - |
| CN4-9 | RXD0 | CNX553-5 |
| CN4-10 | 未接続 | - |

## G　レポートに関する諸注意

[1] 報告書についての一般的事項

(1) 報告書表紙は定められたものを使用し，必要な事項を記入する．表紙の再発行はしないので粉失しないこと．

(2) ワードプロセッサーでも良いが，決して他人のデータをコピーしてはいけない．

(3) 用紙は報告書表紙と同程度の大きさとする（Ａ４程度）．

(4) 表紙を付けて上部を綴じ紐あるいはステップラーで綴じる．

(5) 図（回路図，グラフ，その他の図）及び表は各々に番号と名称を明確に記入し，全て定規等を用いて丁寧に描くこと．

(6) 報告書の提出は次の実験日に提出すること，最後の実験は原則として次週の同曜日に提出するが，別途指示がある場合はそれに従う．

[2] 報告書の内容について

(1) 目的の記述について

実験の目的は２～３行程度とする．

(2) 理論の記述について

原理的な回路図（等価回路など）を含めて実験の理論を記述する．ここで用いる実験方法で実験の目的が達成されることを証明できれば簡単な記述でよい．ただし，データを式に代入して実験結果を算出する場合は，その式を導出しながら説明をくわえること．単に，式に代入した結果，この数値が得られたでは点数を与えられない．計測器原理等については，計測器の原理や構造図は指定しない限り書（描）かなくてよい．

[3] 実験方法と使用器具の記述について

記述していれば役立つと判断した場合は記述すること．例えば製造会社，製造年，仕様等

[4] 実験結果の記述について

(1) 実際に実験した方法を実験過程に沿って述べながら結果を記述すること．実験内容を理解する上で役に立つ．実験結果は表，グラフで示すことを原則とする．

(2) 表の書き方の例を表 3 に示す．表のキャプション（説明書き）は表の上に書く．

(3) グラフに描く例を図 21 に示す．グラフのキャプションはグラフの下に書く．グラフを描くときはグラフ用紙を用いる．グラフ描画ソフトウエアを用いても良い．

(4) 表，グラフ共に物理単位の記述を必ず行うこと．グラフは縦軸，横軸の名称を記載すること．

[5] 考察

(1) 実験項目の実験結果について検討し，自分が予測した事と異なったり，実験点が線の上に乗らなかったり，自分で気付いた事柄を自分なりに検討してみた内容を記述する．

(2) 記述については，自分の妥当性のある意見を入れるように努め，できるだけ具体的な数値を用いた客観的，定量的表現をする．小学校の読後感想文ではないので，この実験が単に面白かった等のことを書かないこと．

[6] 課題

実験日に指定された課題について必ず図書館等で学習しておくこと

**表 3　抵抗の両端の電圧と電流の関係（実験値）**

| 電圧(V) | 電流(mA) |
|---|---|
| 1.0 | 12.2 |
| 2.0 | 23.3 |
| 3.0 | 37.2 |
| 4.0 | 48.0 |
| 5.0 | 62.8 |
| 6.0 | 73.0 |
| 7.0 | 85.1 |
| 8.0 | 97.8 |

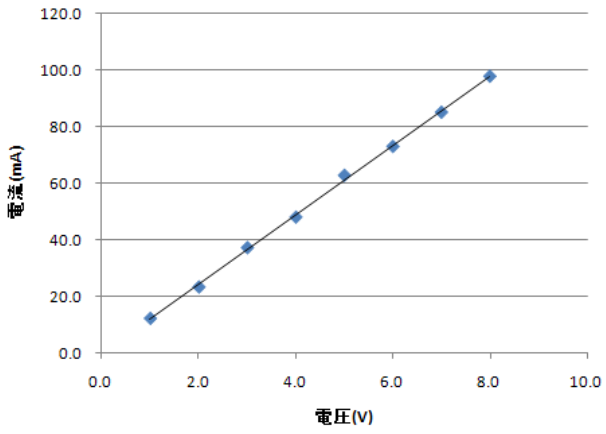


**図 21　印加電圧と抵抗を流れる電流の関係**

## H　入手先等の情報

- AKI-H8 マイコンキット入手先：秋月電子通商　http://akizukidenshi.com/
  - 使用マイコンボード　AKI-H8/3052F マイコンボード
  - 使用マザーボード AKI-H8/3052F USB 開発キット
    ※自作する場合は AKI-H8/3664 および H8 タイニーI/O ボードを用いると小型化が可能．この場合も同じ GCC Developer Lite を用いて開発可能．大きな違いは DA ポートの有無．
  - 書き込みソフト H8TurboWrite が付属する．

- GCC Developer Lite 入手先：Best Technology http://www.besttechnology.co.jp/
  - GCC Developer Lite をインストール後, H8/300H シリーズターゲットファイルを所定のフォルダに解凍．
    ※2008 年 3 月現在公開されている Ver2.0.0.0 では一部関数(sprintf)使用時にコンパイルエラーを生じる．実験では Ver1.9 を用いている．

- ハイパーターミナル RealTerm 入手先：　http://sourceforge.net/projects/realterm
  なお使用可能なハイパーターミナルは RealTerm に限らない．例えば GCC Developer Lite 付属の簡易ターミナルでもよい．他に代表的なものとして Tera Term がある．

- その他電子部品入手先：
  実験で使用した加速度センサ，サーミスタ，RC サーボモータ等はすべて下記の店舗で購入可能である．
  - 秋月電子通商　http://akizukidenshi.com/
  - 千石電商　http://www.sengoku.co.jp/

- 参考書籍
  [1] 横山直隆：C 言語による H8 マイコン プログラミング入門，技術評論社(2003)
  GCC Developer Lite を用いた H8 マイコン開発に関するもの．
  [2] 後閑 哲也：作る，できる/基礎入門電子工作の素，技術評論社(2007)
  電子工作全般の入門．

- サンプルプログラム等
  暫定的に下記 web に掲載している．なお ID とパスワードは授業で使ったものと同じである．
  http://www.kajimoto.hc.uec.ac.jp/3nenjikken/index.html