

# 人間コミュニケーション学実験

## マイクロコンピュータ応用

平成 21 年 10 月版

使用実験室 西 6 号館 1 階 1 0 9 号室

実験期間 4 週

### I. 目的

実世界とコンピュータ世界の間で相互に情報をやり取りするシステムの構築手法を理解する。前期で学んだマイクロコンピュータの外部入出力機能と、新たに学ぶ PC を用いた通信、描画プログラミングにより、マイクロコンピュータと PC を連携させた双方向的なアプリケーションを開発する。

### II. 実験日と実験内容

1 日目：マイクロコンピュータ H8 の復習

2 日目：PC と H8 マイコンのシリアル通信，多目的ライブラリ DxLib

3 日目：自由課題

4 日目：自由課題

## 目次

I.	目的	1
II.	実験日と実験内容	1
III.	実験についての諸注意	3
IV.	実験内容	4
1.	マイクロコンピュータ H8 の復習	4
1.1.	実験の準備	4
1.2.	LED・ボタン入力処理の復習	4
1.3.	LCD (液晶表示器) 表示とタイマー割り込みの復習	4
1.4.	AD 変換 (加速度センサ) の復習	5
1.5.	外部出力 (RC サーボモータの制御) の復習および振動モジュール	5
1.6.	総合課題	8
2.	PC プログラミング	9
2.1.	H8 から PC への送信	9
2.2.	PC から H8 への送信	10
2.3.	多目的ライブラリ DxLib の準備	10
2.4.	プログラムの起動, 終了, 画面描画	11
2.5.	総合課題	12
3.	自由課題	13
3.1.	準備: シリアル通信プログラムとの融合	13
3.2.	自由課題	13
V.	付録	15
A.	GCC Developer Lite および H8 Write Turbo の使用方法	15
B.	H8 用ソフトウェアの設定	16
C.	RealTerm の使用方法	16
D.	Visual C++ 2008 Express Edition の操作	17
E.	Visual C++ 2008 Express Edition の設定	17
F.	回路図	19
G.	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	20
H.	レポートに関する諸注意	23
I.	入手先等の情報	25
J.	レポート用チェックリスト	26

### III. 実験についての諸注意

- 事前準備に関する注意

実験は前期の実験を履修していることを前提とする。

前期実験のテキストと実験ノートで復習しておくこと。

実験日には前期実験のテキスト，および実験ノートを持参すること。

- 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

✓ 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すること。

✓ 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。

✓ 提出の際はチェックリストにチェックして提出すること。

- 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。

✓ 参考文献を借りた場合，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 の復習

#### 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・ボタン入力処理の復習

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。

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1] サンプルプログラムを参考にして、ボタンを一回押すごとに右から順に LED が点灯するようにせよ。

なおレポートでは以下のポイントに注意し、各ポイントに対するプログラム上の工夫について説明せよ。

- 出力論理 0 のときに LED が点灯する。
- ボタンを押したとき入力論理 0 とする。
- ボタンを押し続けたときに LED の点灯が移動してはならない。「押す」と「押し続けること」の違いをどのように表現するか。

#### 1.3. LCD（液晶表示器）表示とタイマー割り込みの復習

LCD（液晶表示器）に文字を表示する。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、文字列が表示されることを確認する。

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。

[課題2] サンプルプログラムを変更して、ボタン押下の周波数、すなわち一秒に何回ボタンを押したかを表示するようにせよ。

なおレポートでは以下のポイントに対するプログラム上の工夫について説明せよ。

- 一秒ごとに押した回数を表示するのではなく、ボタンを押すたびに表示を変更するにはどうしたらよいか。

#### 1.4. AD 変換（加速度センサ）の復習

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．

利用している加速度センサは次のようなスペックを持つ．

$$\text{出力振幅 (感度)} = V_{dd}/5(V/G)$$

$$0G \text{ 時の出力電圧} = V_{dd}/2(V)$$

$V_{dd}/5(V/G)$ 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  $5/5=1V$  の出力があることを意味する．また  $0G$  時の出力電圧= $V_{dd}/2(V)$ とは，例えば電源電圧が 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．

[課題3] 重力加速度ベクトルの大きさ (ノルム) が常に 1G であることを示すプログラムを作成せよ．

このためには次のようなポイントに注意する必要がある．各ポイントに対する結果を含めてレポートせよ．

- x 軸，y 軸，z 軸がどの方向かを，基板を手にとって傾きを変えることで把握する．
- 各軸での最大出力電圧および最少出力電圧を記録する．この結果から較正し，各軸の出力を重力加速度の単位(G)で表示するための変換式を得る．(ヒント:最大の電圧は 1G，最小の電圧は-1G が加わったときに生じる．その間は線形補間)
- 以上から，x, y, z 軸方向の重力加速度成分が求められるので，これらの二乗和を計算して LCD で表示するプログラムを作成する．

$$a^2 = a_x^2 + a_y^2 + a_z^2$$

- 理想的には LCD には常に 1.0 が表示されるはずであるが，当然実際には結果はぶれる．どの程度ぶれたかレポートせよ．

#### 1.5. 外部出力（RC サーボモータの制御）の復習および振動モジュール

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 2 のように接続せよ．

RC サーボモータの制御信号は図 1 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．

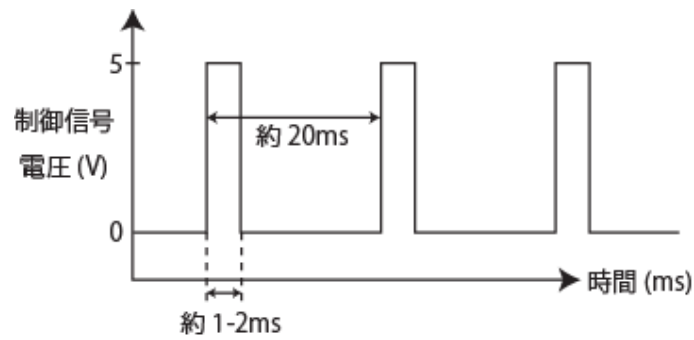


図 1 RC サーボモータの制御信号

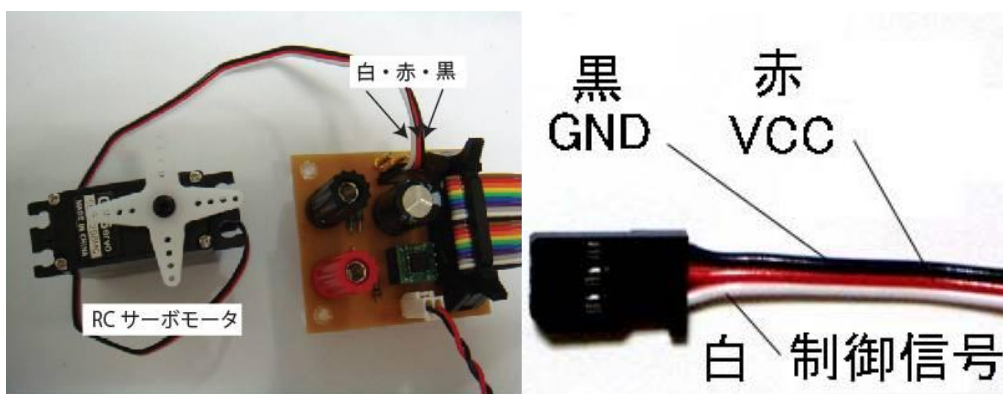


図 2 RC サーボモータの接続 (向きに注意)

ここで RC サーボモータの代わりに振動モジュールを装着する. 使用する振動モジュールは図 3 のように, 振動モータ, 抵抗およびトランジスタで構成されている. RC サーボモータと同様に 3つの端子 (VCC, GND, 制御信号) があり, そのまま接続して使用できる.

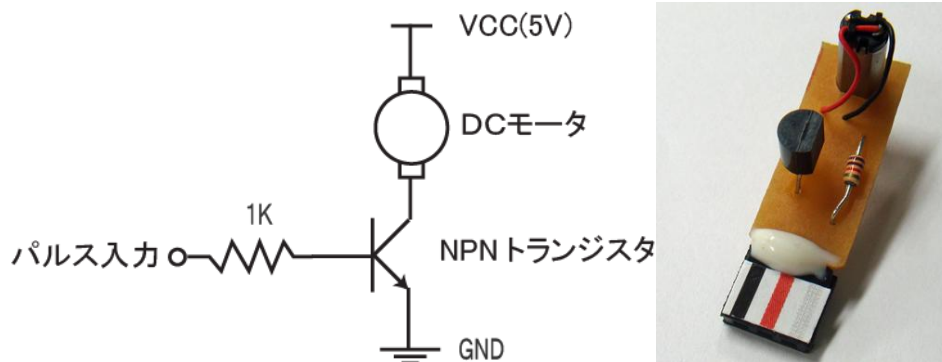


図 3 振動モジュール. (左) 回路図, (右) 写真

マイコンから 5V の信号が入力される(ON)とトランジスタを介して振動モータに電流が流れ, 0V の信号が入力される(OFF)と電流は流れない. 全体の時間に対する ON 時間の比率

(Duty 比)を変えることで、振動の強さを変更することができる。このような変調方法を PWM(Pulse Width Modulation : パルス幅変調)と呼ぶ(図 4)。

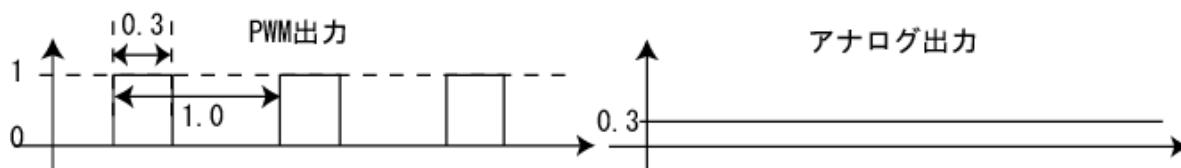


図 4 PWM 方式による駆動. Duty 比 0.3 の場合 (左). これは平均して, 0.3 のアナログ出力を行っているのと等価になる (右).

[課題4] RC サーボモータのコネクタを外し、振動モジュールを装着せよ。この振動モジュールは PWM の Duty 比が大きいほど強く振動する。

サンプルプログラム(rc\_servo.c)を改変し、全体の周期は 20ms のまま、ボタンを押すとパルス幅が 0ms, 2.5ms, 5ms, 7.5ms となるようにせよ。これによって振動の強さが変わることを確認せよ。(ヒント: 前期の実験で同様のプログラムを書いている)

[課題5] [課題 4]で作成したプログラムをサンプルプログラム(accel.c)と組み合わせ、傾きに応じて振動の強さが変化するようにせよ。その際、図 5 のように、水平に保持したときには振動せず、垂直に保持したときに最大の振動になるようにせよ。

なおレポートでは以下のポイントに対するプログラム上の工夫について説明せよ。

- 加速度センサの x,y,z 軸のうちどの軸の出力を選ぶべきか。またそれはなぜか。
- ユーザの操作に対する応答速度という観点では、全体の周期 20ms は必ずしも十分な応答速度ではない。どの程度が望ましいか、実験的に気付いたことがあればレポートせよ。

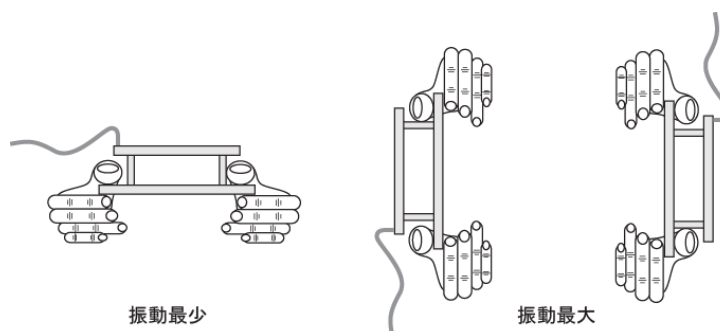


図 5 姿勢と与える振動の強さ

[課題6] PWM について調べ、より詳しく原理および通常のアナログ出力に対する利点を説明せよ。

## 1.6. 総合課題

時間に余裕があれば次の課題を行うこと。レポートではプログラム上の工夫および気づいた点を述べよ。

[課題7] [課題5]では傾きに応じて「常に」振動するプログラムとした。これは非常に不快なインターフェースである。これを変更し、いくつかの指定した傾きに達した瞬間のみ振動させることで、一種の「クリック感」を生成するようにせよ。

[課題8] クリスタルイヤフォンを用いた音出力 (d.a.c) に関して復習し、加速度センサと組み合わせて、傾きに応じて音の高さが変わる楽器を作成せよ。



## 2. PC プログラミング

PC 側でプログラミングを行い，ハイパーターミナルを用いずに直接 H8 マイコンとの通信を行う．さらにライブラリ `DxLib` を用いてインタラクティブシステムを開発するための基礎的手法を学ぶ．

### 2.1. H8 から PC への送信

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．

次に PC 側をハイパーターミナルではなく，自作のソフトウェアに置き換える．付録 D を参考にして `¥PC_Serial¥Serial_Receive¥Serial_Receive.vcproj` をダブルクリックして `Visual Studio` を立ち上げ，コンパイルして動作させよ．H8 マイコンをリセットし，H8 マイコンから数字が送られ続けることを確認せよ．

※シリアル通信のポート番号がプログラム中で固定されている．ハイパーターミナルの場合と同様にこれを変更してコンパイルしないと正常に動作しない場合がある．

※ここでは PC 用のシリアル通信関数群を `Serial.h`, `Serial.cpp` の中で C++ クラスライブラリの形で定義している．関数の定義に興味があればこれらのファイルを参照すること．

[課題9] H8 側のプログラムの次の行の動作をわかりやすく説明せよ．結果だけでなく，なぜそうなるかも説明すること．

```
uc_count=(uc_count+1)%256;
```

[課題10] PC 側プログラムの次の行の動作をわかりやすく説明せよ．結果だけでなく，なぜそうなるかも説明すること．

```
while(!_kb_hit()){...}
```

今回のプログラムでは，通信データは，PC 側でも，H8 側でも”`unsigned char`”（符号なし 8bit データ）として定義している．これは 0 から 255 までの数値を表わすデータである．

[課題11] PC 側のデータを `char`(符号付き 8bit データ)とすると，どのような影響が出るか説明せよ．結果だけでなくその理由もわかりやすく説明すること．

また実際にこれを実験によって確認せよ．次のような改変をすれば良い．

- PC 側の改変：`unsigned char receive_data` ⇒ `char receive_data`
- H8 側の改変：結果がすぐ見えるように，`uc_count=0` ⇒ `uc_count=120` とする．

[課題12] H8 側の加速度センサプログラム(`accel.c`)とシリアル送信プログラム

(serial\_send.c)を改変し, x,y,z 軸の値が順に PC に送信されるようにせよ. PC 側ではこれらの値を順に表示するようにせよ.

なおレポートでは以下のポイントに注意し, プログラム上の工夫について説明せよ.

- Accel.c 中で加速度センサのデータとして得られるデータ(usVoltageX, usVoltageY, usVoltageZ)は unsigned short (符号なし 16bit データ) であり, この中に 10bit(0 ~1023)のデータが格納される. これをシリアル通信で送信するためにはどのようにしたらよいか.  
(例 1) 4 で割って最大値を 255 に抑えて送信する.  
(例 2) 2 回に分けて上位 8bit, 下位 8bit を送信する. (やや高度)
- 実験では実現する必要はないが, 現実的なアプリケーションでは H8 から毎回送られてくるデータが x 軸か, y 軸か, z 軸かを PC 側で判断する必要がある. これはなぜか. またどのようにして判断する方法が考えられるか, 調べてレポートせよ.

## 2.2. PC から H8 への送信

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.

次に 2.1 節と同様に, PC 側をハイパーターミナルではなく, 自作のソフトウェアに置き換える. 付録 D を参考にして¥PC\_Serial¥Serial\_Send¥Serial\_Send.vcproj をダブルクリックして Visual Studio を立ち上げ, コンパイルして動作させよ. H8 マイコンをリセットし, H8 マイコンの LCD にキーボード入力が表示されることを確認せよ.

[課題13] PC 側のプログラムの下記 2 行が行っていることをわかりやすく説明せよ. 結果だけでなく, なぜそうなるかも説明すること.

```
while(!_kbhit());  
uc = _getch();
```

[課題14] H8 のプログラムを改変し, [課題 5]のプログラムを参考にして, PC 上でキーボードの 0 から 9 を押すと, 異なる強さで振動モジュールが振動するプログラムを作成せよ. 単純には送信された文字 10 通りについて場合わけすれば良いが, 数値として扱えばシンプルになる. これを試み, レポートではこの工夫について言及せよ.

## 2.3. 多目的ライブラリ DxLib の準備

DxLib を使ったプログラミングを行う. DxLib は Microsoft 社の多目的ライブラリ DirectX の機能を簡単に使うために開発されたもので, 特に今回は 2 次元画面描画機能を用いる. なお DxLib は今回扱う機能以外に, PC 間ネットワーク通信機能, 音楽・動画再生機能等を

備えており、C言語によってインタラクティブなマルチメディアアプリケーションを簡単に作成する優れたライブラリである。

コピーした Help フォルダ中の index.html を立ち上げ、「DX ライブラリの関数リファレンスマニュアル」をクリックする。ここに DxLib に関する全てのマニュアルがある。この情報を使って以降のプログラミングを行うこと。

## 2.4. プログラムの起動, 終了, 画面描画

次のプロジェクトを立ち上げ、コンパイルし、実行せよ。これは 640×480 の解像度のフルスクリーン画面を設定し、その中央にドットを打つだけのシンプルなプログラムである。

¥DxLib¥SimpleWindow¥ SimpleWindow.vcproj

[課題15] プログラムを改変し、次のようなアプリケーションを作成せよ。

「マウスクリックをするたびに、クリックした場所から円が現れ、自由落下していく」レポートでは「自由落下運動」をシミュレートする方法を詳細に述べよ。

[課題 15]を実現するために参考となる手順を以下に示す。手順を追えば難しくはない。

1. 関数 DrawCircle, GetColor のヘルプを参照しながら、赤色に塗りつぶされた半径 10 の円を描画するプログラムを作成する。
2. 関数 printfDx, ScreenFlip のヘルプを参照しながら、適当な文字列を描画するプログラムを作成する。次のような行を追加すればよい。

```
printfDx("出力したい文字列"); //文字列を出力する
ScreenFlip(); // 出力した文字列を表示する
```

※ScreenFlip 関数を用いないと printfDx 関数の出力結果は画面に反映されない。また printfDx 関数は非常に動作が遅いのでデバッグ用にのみ使用すること。

3. ここまでのプログラムでは、関数 WaitKey によって、キーボードが押されると終了するようになっている。これを改変し、無限ループのプログラムとする。

具体的には関数 CheckHitKey を用い、Z キーが押されるまで動くようにする。

(参考ソース。正しいとは限らない)

```
int n=0;
while(CheckHitKey(KEY_INPUT_Z)==0){
    n++;
    (数値 n を表示)
}
```

4. 関数 `GetMouseInput` および `GetMousePoint` を用い、「マウスがクリックされたときにクリックされた座標を表示する」プログラムを作成する。

(参考ソース. 正しいとは限らない)

```
int MouseX, MouseY;
if( ( GetMouseInput() & MOUSE_INPUT_LEFT ) != 0 ){
    GetMousePoint( &MouseX, &MouseY );
    (数値 MouseX,, MouseY を表示)
}
```

5. ここまでのプログラムを使って、マウスをクリックすると赤色で塗りつぶされた円が表示されるようにする。クリックしながらドラッグすると絵を描けるはずである。

6. 関数 `ClearDrawScreen` を用い、右ボタンを押すと画面が初期化されるようにする。

(参考ソース. 正しいとは限らない)

```
if( ( GetMouseInput() & MOUSE_INPUT_RIGHT ) != 0 ){
    ClearDrawScreen();
}
```

7. マウス入力があると、円が描画され、その後 10 ミリ秒ごとに画面の消去と描画を繰り返す。描画する座標をずらすことによって、円が一定速度で落ちていくようにする。

(参考ソース. 正しいとは限らない)

```
if(MouseY < 480){ //画面の一番下の座標は 480
    ClearDrawScreen() ;//画面全体を消去する.
    DrawCircle(MouseX,MouseY,10,GetColor(255,0,0),1);
    MouseY++; //描画位置を下にずらす.
}
WaitTimer(10);//10 ミリ秒待つ
```

8. これまでに一定速度で落ちていくアニメーションが完成するはずである。あとは自由落下運動 (= 等加速度運動) をシミュレートすればよい。

## 2.5. 総合課題

時間に余裕があれば次の課題を行うこと。

[課題16] [課題 15]では落下する物体は常に一つである。これを複数個対応可能とせよ。つまりマウス左ボタンを連打するとその回数だけ球が落下するようにせよ。さらに余裕があれば関数 `SetDrawScreen` 等を用いてチラつきのないアニメーションを実現せよ。

### 3. 自由課題

#### 3.1. 準備：シリアル通信プログラムとの融合

前回使用していた DxLib のプロジェクトのフォルダに Serial.h および Serial.cpp をコピーし、プロジェクトにこれらのファイルを追加せよ。追加の方法は付録 E の(4)を参考にせよ。Serial\_Receive.cpp のファイル等を参考にしながら、WinMain 関数のある cpp ファイル中でシリアル通信のためのヘッダファイルの記述、変数の宣言、シリアル通信の初期化部分を追加し、DxLib とシリアル通信を同時に行えるようにせよ。

[課題17] [課題 12] をベースとした H8 側のプログラム、および[課題 15]をベースとした PC 側のプログラムを用いて、次のようなプログラムを作成せよ。

「画面に円が描画されており、加速度センサの値によって円が前後左右に移動する」

図 6 参照。なおレポートでは、加速度センサのどの軸を用いているか、および各軸を最大限に動かしたときに円が画面中を最大に動くようにするための工夫を説明せよ。

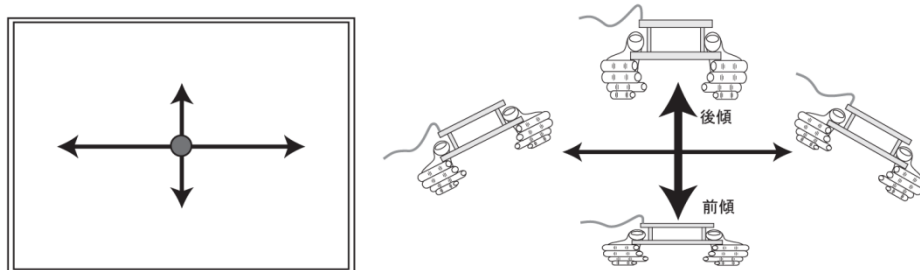


図 6 加速度センサによるコントロール

#### 3.2. 自由課題

自由課題には次の要素が入っていることが必要である。

- H8 による計測，およびその結果の PC へのシリアル通信
- PC による描画

また次の要素が入っているとよい。

- PC から H8 へのシリアル通信，およびその結果による H8 側の動作
- ゲーム的要素，あるいは計測実験的要素
- PC 上での物理シミュレーション．例えばボールが跳ねる等。
- PC 間の通信（ただし複数グループの協力が必要である．DxLib で可能だが，高度）

例えば次のようなアプリケーションが考えられる。

- 加速度センサによって入力するシーソー．画面上にボールとシーソーがあり，シーソーを操作するとボールが動く．

- 加速度センサによって入力するピンポンゲーム。ボールが跳ねるたびに振動モジュールで触覚を提示する。
- PCの画面を使ったオシロスコープ，データーロガー。
  - 温度の時間変化などを画面にリアルタイムで表示する。
  - バッテリーの充放電特性計測を計測する。バッテリーに抵抗を接続し，抵抗間の電圧から電流を計算する。積分することで容量を測定する。
  - LED の電圧－電流特性を自動的に測定する。太陽電池によって LED に流れる電流と発光量の関係を調べる。
  - 太陽電池を用いて太陽の日射量変化を測定する。
- 他にオペアンプ，DC モータ，アンプなどを用意できる。ヒューマンメディア工房の電子部品は自由に使ってよい。

自由課題は3日目，4日目に行う。作成は4日目の4時で打ち切り，残りの時間で各グループの品評会を行う。

全く成果が無い状態を避けるため，プログラムは簡単な部分から順序立てて作成し，異なるバージョンで保存するように心がけること。

[課題18] 3日目の早い段階で作成するアプリケーションについて実験パートナーと話し合うこと。この段階では少なくとも次の項目が話し合わなければならない。

- 自由課題のタイトル
- 自由課題の目的
- PC側のプログラムが行う内容。およびPCからH8に送る情報。
- H8側のプログラムが行う内容。およびH8からPCに送る情報。

話し合いの結果を3日目のレポート課題とする。4日目の実験開始時に提出すること。

[課題19] 4日目までの成果に関するレポートでは，実際に作成したアプリケーションの仕様と結果を報告せよ。レポートには次の項目が含まれていなければならない。

- 自由課題のタイトル
- 自由課題の目的
- PC側のプログラムが行う内容。およびPCからH8に送る情報。
- H8側のプログラムが行う内容。およびH8からPCに送る情報。
- PC側のプログラムの関数リストおよびそれぞれの機能
- H8側のプログラムの関数リストおよびそれぞれの機能
- 工夫した点
- 気づいた点
- 周囲の反応

## 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 7)

- (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) 電源が入っていないか確認する.
- (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 7)

- (ア) 電源が入っていないか確認する.
- (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.

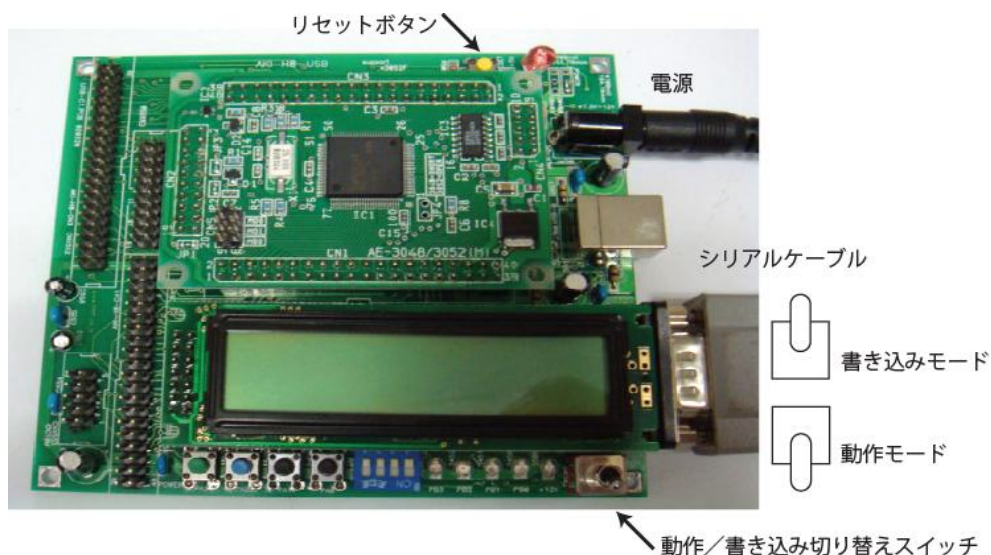


図 7 動作/書き込みモード切り替えとリセットボタン

## B. H8用ソフトウェアの設定

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 内蔵フラッシュ 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. Visual C++ 2008 Express Edition の操作

実験で使用する操作に限定して説明する。実験ではプロジェクトの設定等は済んでおり、ソースファイルを書き換えてコンパイル、動作させればよい。自力でプロジェクトを作成する方法に関しては付録 E を参照すること。

- (1) プロジェクトの立ち上げ。拡張子.vcproj のファイルをダブルクリックする。
- (2) ソリューションエクスプローラ中のソースをダブルクリックして表示
- (3) ソースを書き換える。

**※実験中はバージョンの異なるソースを頻繁に別の名前でバックアップすること※**

- (4) コンパイル。「ビルド」⇒「ソリューションのビルド」を選択してコンパイルする。F7 キーを押してもよい。下のメッセージ画面を見て、エラーが生じていれば修正する。
- (5) 実行。「デバッグ」⇒「デバッグなしで開始」を選択する。Ctrl+F5 を押してもよい。

#### E. Visual C++ 2008 Express Edition の設定

Visual Studio Express Edition は web からダウンロードできるフリーの開発環境である。この機会に自学自習することを強く勧める。以下では Visual C++を用いてコンソールアプリケーションとウインドウアプリケーションを作成するもっとも簡単な手順を述べる。

・コンソールアプリケーションは次の手順で作成する。

- (1) ファイル⇒新規作成⇒プロジェクトを選択し、出てきた画面の中で、「プロジェクトの種類」を win32, 「テンプレート」を win32 コンソールアプリケーションとする。適当なプロジェクト名を入力し、場所を選ぶ。「ソリューションのディレクトリを作成」というチェックボックスを外し、OK ボタンを押す。
- (2) 次の「win32 アプリケーションウィザード」では左の「アプリケーションの設定」を選び、アプリケーションの種類を「コンソールアプリケーション」とし、追加のオプションを「空のプロジェクト」とする。完了ボタンを押す。
- (3) 画面右側にソリューションエクスプローラが表示されている。新規にソースファイルやヘッダファイルを作成していく場合、ソリューションエクスプローラ中の「ソースファイル」「ヘッダファイル」タブを右クリックし、「追加」⇒「新しい項目」でソースファイルやヘッダファイルを作成すればよい。  
※注：ソースファイル名の拡張子は c ではなく cpp とすること。
- (4) 既存のファイルをプロジェクトに組み込みたい場合（例えばシリアル通信用のライブラリとして Serial.cpp や Serial.h を組み込む場合）、事前にプロジェクトのフォルダにフ

ファイルをコピーしておき（これは必ずしも必要ではないがプロジェクトの可搬性を考えると行った方が無難）、(3)と同様にソリューションエクスプローラ中のタブを右クリックし、「追加」⇒「既存項目」でファイルを指定する。

DxLib を用いたウインドウアプリケーションについては、Help フォルダ中の index.html で作成方法について詳細に述べられている。「Dx ライブラリの使い方」⇒「Visual C++ Express Edition」で書かれている手順に従う。

※DxLib 関数の動作はグラフィックボードに依存する部分があり、想定通りに動かない場合、あるいは使用 PC によって動作が異なる場合がある。このような場合、PC のグラフィックアクセラレーション機能を停止させることで解決する場合が多い。この設定は次のように行う。

- (1)PC のデスクトップ画面中でマウス右クリックし、プロパティを選択。
- (2)「設定」タブをクリックし、「詳細設定」ボタンを押す。
- (3)「トラブルシューティング」タブをクリックし、「ハードウェアアクセラレータ」バーを左にドラッグし、アクセラレーション機能をなしにする。
- (4)OK ボタンを押す。

## F. 回路図

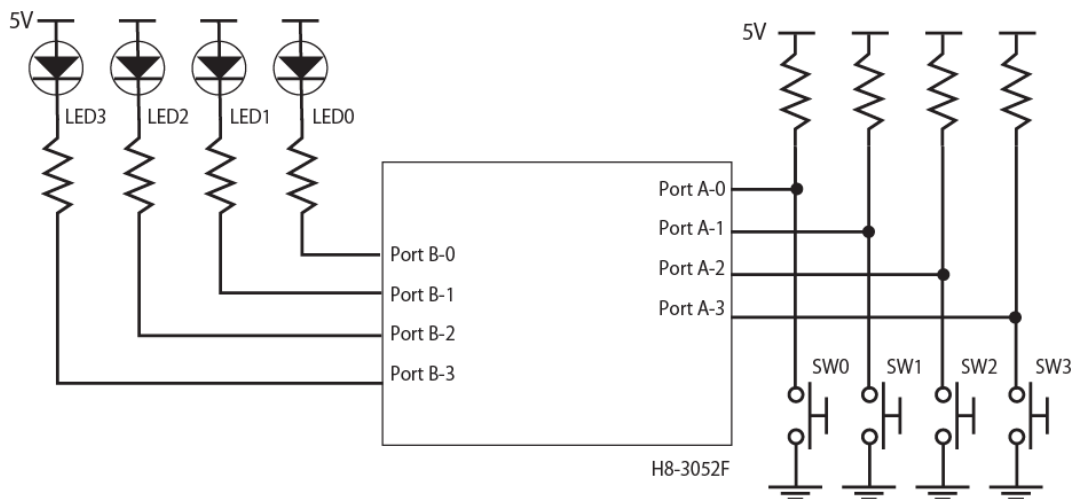


図 8 H8 ボード回路 (LED, スイッチのみ抜粋)

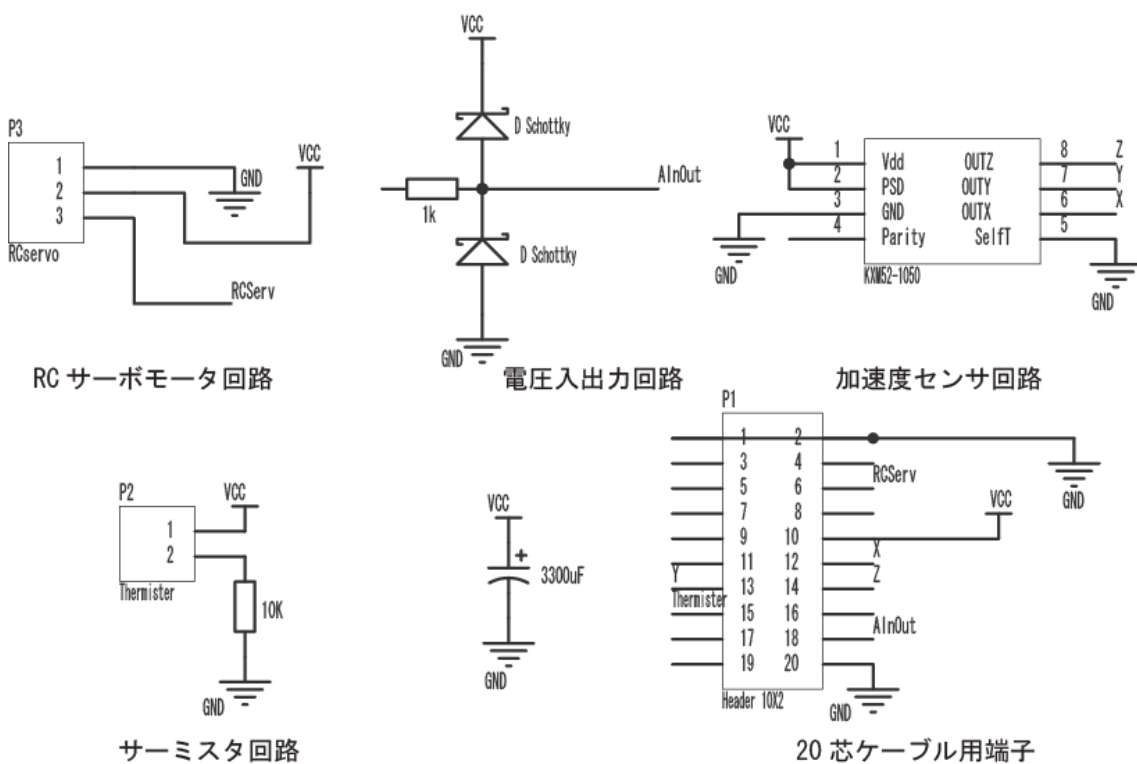


図 9 入出力基板回路

## G. 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		ポート名	外部デバイス	
P1-0	SRAM-A0		P5-0	SRAM-A16	
P1-1	SRAM-A1	USB9604-A0	P5-1	SRAM-A17/CS	
P1-2	SRAM-A2		P5-2	SRAM-A18/NC	
P1-3	SRAM-A3		ポート名	外部デバイス	
P1-4	SRAM-A4		P6-4	SRAM-OE	USB9604-RD
P1-5	SRAM-A5		P6-5	SRAM-WE	USB9604-WR
P1-6	SRAM-A6		ポート名	外部デバイス	
P1-7	SRAM-A7		P8-2		USB9604-CS
ポート名	外部デバイス		P8-3	SRAM-CE	
P2-0	SRAM-A8		ポート名	外部デバイス	
P2-1	SRAM-A9		P9-1	RS232C-CH1	
P2-2	SRAM-A10		P9-3	RS232C-CH1	
P2-3	SRAM-A11		P9-5		USB9604-INTR
P2-4	SRAM-A12		ポート名	外部デバイス	
P2-5	SRAM-A13		PA-0	SW0(PA0) [0=ON]	
P2-6	SRAM-A14		PA-1	SW1(PA1) [0=ON]	
P2-7	SRAM-A15		PA-2	SW2(PA2) [0=ON]	
ポート名	外部デバイス		PA-3	SW3(PA3) [0=ON]	
P3-0	SRAM-D8	USB9604-D0	ポート名	外部デバイス	
P3-1	SRAM-D9	USB9604-D1	PB-0	LED0(PB0) [0=ON]	LCD-DB4
P3-2	SRAM-D10	USB9604-D2	PB-1	LED1(PB1) [0=ON]	LCD-DB5
P3-3	SRAM-D11	USB9604-D3	PB-2	LED2(PB2) [0=ON]	LCD-DB6
P3-4	SRAM-D12	USB9604-D4	PB-3	LED3(PB3) [0=ON]	LCD-DB7
P3-5	SRAM-D13	USB9604-D5	PB-4		LCD-RS
P3-6	SRAM-D14	USB9604-D6	PB-7		LCD-E
P3-7	SRAM-D15	USB9604-D7			

※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	AKI-H8/3052F	名称	拡張コネクタピン
CN1-1	GND	CNX551-1	CN3-1	GND	CNX552-1
CN1-2	GND	CNX551-2	CN3-2	GND	CNX552-2
CN1-3	P8-0	CNX551-3	CN3-3	P4-4	CNX552-3
CN1-4	P8-1	CNX551-4	CN3-4	P4-5	CNX552-4
CN1-5	P8-2	CNX551-5	CN3-5	P4-6	CNX552-5
CN1-6	P8-3	CNX551-6	CN3-6	P4-7	CNX552-6
CN1-7	P8-4	CNX551-7	CN3-7	P3-0	CNX552-7
CN1-8	PA-0	CNX551-8	CN3-8	P3-1	CNX552-8
CN1-9	PA-1	CNX551-9	CN3-9	P3-2	CNX552-9
CN1-10	PA-2	CNX551-10	CN3-10	P3-3	CNX552-10
CN1-11	PA-3	CNX551-11	CN3-11	P3-4	CNX552-11
CN1-12	PA-4	CNX551-12	CN3-12	P3-5	CNX552-12
CN1-13	PA-5	CNX551-13	CN3-13	P3-6	CNX552-13
CN1-14	PA-6	CNX551-14	CN3-14	P3-7	CNX552-14
CN1-15	PA-7	CNX551-15	CN3-15	P1-0	CNX552-15
CN1-16	PB-0	CNX551-16	CN3-16	P1-1	CNX552-16
CN1-17	PB-1	CNX551-17	CN3-17	P1-2	CNX552-17
CN1-18	PB-2	CNX551-18	CN3-18	P1-3	CNX552-18
CN1-19	PB-3	CNX551-19	CN3-19	P1-4	CNX552-19
CN1-20	PB-4	CNX551-20	CN3-20	P1-5	CNX552-20
CN1-21	PB-5	CNX551-21	CN3-21	P1-6	CNX552-21
CN1-22	PB-6	CNX551-22	CN3-22	P1-7	CNX552-22
CN1-23	PB-7	CNX551-23	CN3-23	P2-0	CNX552-23
CN1-24	VPP/RES0	CNX551-24	CN3-24	P2-1	CNX552-24
CN1-25	P9-0	CNX551-25	CN3-25	P2-2	CNX552-25
CN1-26	P9-1	CNX551-26	CN3-26	P2-3	CNX552-26
CN1-27	P9-2	CNX551-27	CN3-27	P2-4	CNX552-27
CN1-28	P9-3	CNX551-28	CN3-28	P2-5	CNX552-28
CN1-29	P9-4	CNX551-29	CN3-29	P2-6	CNX552-29
CN1-30	P9-5	CNX551-30	CN3-30	P2-7	CNX552-30
CN1-31	P4-0	CNX551-31	CN3-31	P5-0	CNX552-31
CN1-32	P4-1	CNX551-32	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	-

## H. 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紛失しないこと。
- (2) ワードプロセッサでも良いが、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）は表の上を書く。
- (3) グラフに描く例を図10に示す。グラフのキャプションはグラフの下を書く。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ェアを用いても良い。
- (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。 小学校の読后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

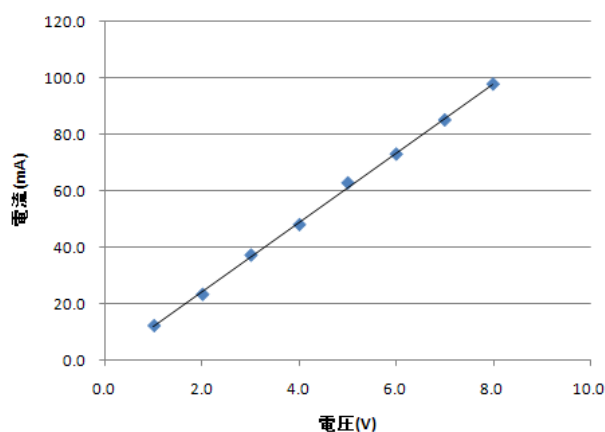


図 10 印加電圧と抵抗を流れる電流の関係



## I. 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
  - 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.
  
- DxLib 入手先：<http://homepage2.nifty.com/natupaji/DxLib/>
  
- 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.
  - [3] 大槻 有一郎：14 歳からはじめる C 言語わくわくゲームプログラミング教室 V—  
Windows XP/Vista 対応, ラトルズ(2008)  
DxLib およびゲームプログラミングに関する入門書.
  
- サンプルプログラム等  
暫定的に下記 web に掲載している. 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>

## J. レポート用チェックリスト(コピーしてレポートに毎回添付すること)

### 一般

- それぞれの実験課題の内容が明確に記述されているか。
- それぞれの実験課題の結果および考察はなされているか。
- それぞれの実験課題で発見した問題点，およびそれに対する自分自身の工夫は明確に記述されているか。

### 1 日目

- [課題 1]入出力の論理に関する工夫を記述しているか。   
ボタンを「押す」と「押し続ける」の違いに対する工夫を記述しているか。
- [課題 2]リアルタイムにボタン押下の周波数を表示する工夫を記述しているか。
- [課題 3]x, y, z 各軸に対する較正の式は求められているか。   
各軸の重力加速度の二乗和のぶれはどの程度か記述しているか。
- [課題 4]プログラムの改変部分に関する説明はなされているか。
- [課題 5]選んだ軸，およびその理由に関する説明はなされているか。   
応答周期に関する記述はなされているか。
- [課題 6]PWM について分かりやすく記述されているか。

### 2 日目

- [課題 9]動作について，理由も含めてわかりやすく記述されているか。
- [課題 10] 動作について，理由も含めてわかりやすく記述されているか。
- [課題 11]どのような結果になったか，理由も含めて記述されているか。
- [課題 12]3 軸の加速度データが連続して送信され，PC 側で受け取られる工夫について説明しているか。   
データの型変換に関する工夫について結果も含めて説明しているか。
- [課題 13] 動作について，理由も含めてわかりやすく記述されているか。
- [課題 14]受信した文字 0~9 を数値として扱う工夫について説明しているか。
- [課題 15]自由落下運動をシミュレートする方法が詳細に述べられているか。

### 3 日目

- [課題 17]加速度センサのどの軸を用いているか述べられているか。
- [課題 18]自由課題のタイトル，目的，PC，H8 それぞれの役割分担が，図も使って詳しく説明されているか。

### 4 日目

- [課題 19]自由課題のタイトル，目的，PC，H8 それぞれの役割分担，工夫した点，気づいた点が，図も使って詳しく説明されているか。