

制御用小型コンピュータによる制御実験

1 目的

制御用小型コンピュータによる制御実験を通じて、コンピュータ周辺機器とのインタフェースや機器制御の基礎を学習する。実際に入出力を行うプログラムを動かしながらハードウェアとソフトウェアの関係を学び、情報系のものづくり技術を体得する。

2 概要

H8/3664 は制御用の小型コンピュータであり、内部に命令用メモリ (32KB)、データ用メモリ (2KB)、CPU、入出力ポート 6 系統等が収められている。パソコンで開発したプログラムを、シリアル通信により H8/3664 に転送し、そのプログラム動作で、入出力ポートを自由に制御することができる。

前期の実験で経験したポートの入出力の知識を活かし、コンピュータ周辺機器のインタフェースや機器制御の基礎的なテーマに取り組んでもらう。

以下に示す複数のテーマの中から 1 つを選び、2 ~ 3 名の共同でプログラムを作成せよ。

1. 電子サイコロ

1 つのスイッチとサイコロの目の形に並べた 6 個の LED を使って電子サイコロを作る。

スイッチを押している間、サイコロが変化すること。

入力が 1 つで簡単すぎる様であれば、出力ポートの先に圧電ブザー等を接続し、ゲームっぽく改良せよ。

2. キーマトリックススキャン

3 × 3 の 9 個の押しボタンスイッチの状態を読み込む。ただし入力に使うポートは、必要最低限 6bit の使用に留めること。

3. ライントレース自走車

2 つの反射型光センサをつけた模型を、床に描いた黒い線の上を走行させる。

各テーマ毎に必要な考え方などの情報は、

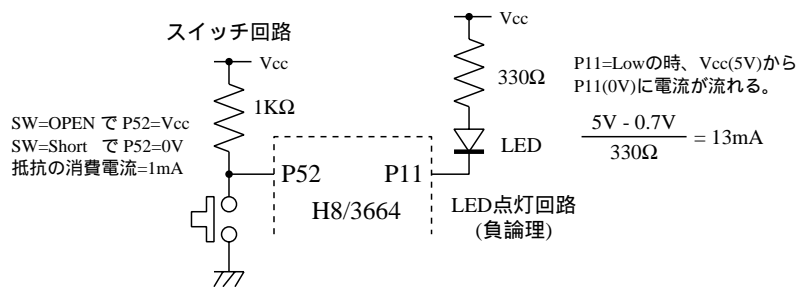
<http://www.ei.fukui-nct.ac.jp/t-saitoh/exp/h8/>

にて公開しているので、参照すること。

3 基本となる回路とプログラムの説明

H8/3664 の動作確認をするための基本プログラムとして、スイッチから入力した High/Low に応じて LED を点滅させるプログラム¹ を補足資料 A に示す。

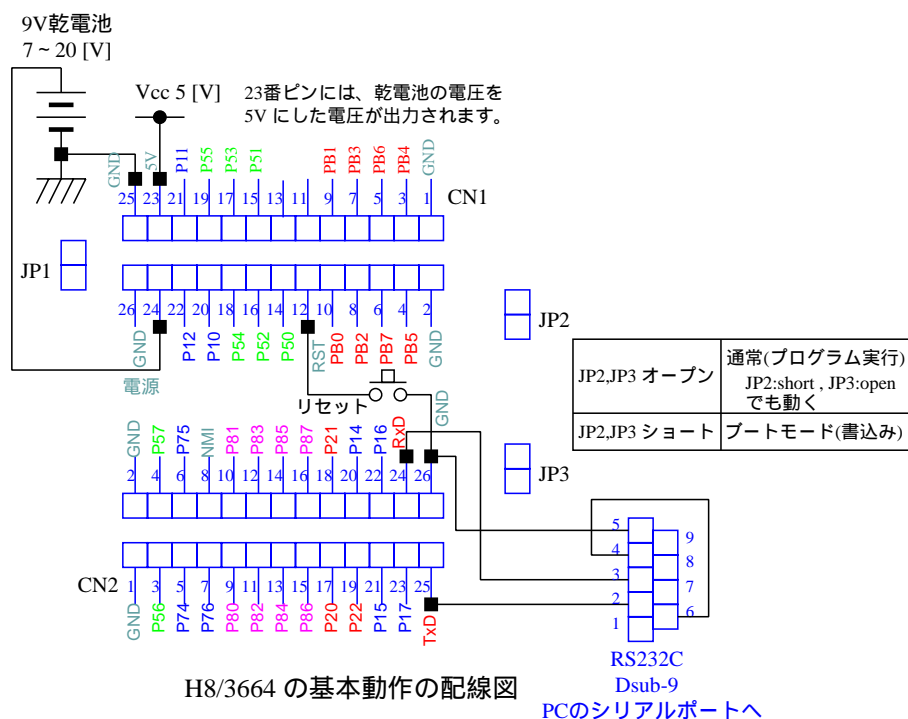
¹ サンプルでは、ビット演算のプログラム例とするため、P11,P52 といった統一性の無い端子を用いている。プログラムを作る際には、判りやすい端子を使用すること。



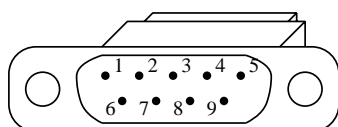
レポート作成での補足

1. レポート作成時には、補足資料は記載しなくても良い。
2. レポートには、H8、スイッチ等の周辺回路の配線を記載せよ。
3. プログラムの主要部分のリストを添付せよ。
4. 考察テーマは、各テーマ毎の補足資料を参照せよ。

4 H8/3664 のピン配置



5 シリアルコネクタのピン配置



配線に注意せよ！ シリアルケーブル
と接続する端子を正面にして、左上か
ら1番ピン。

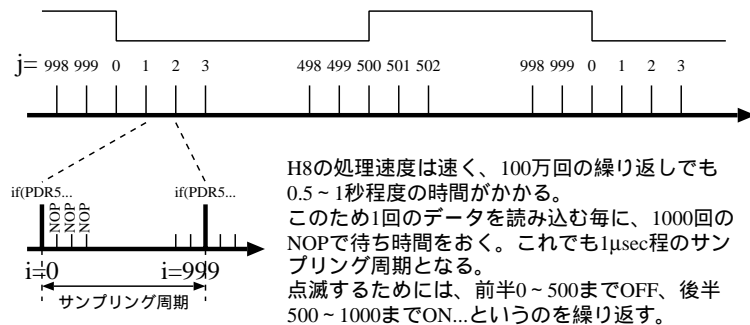
A 基本プログラム

```

#include "3664.h"
/* P11 を LED に接続、 P52 を SW に接続 */
#define P1OUT 0x2 /* P11:0000,0010 */
#define P5IN 0x4 /* P52:0000,0100 */

/* P52 のスイッチが ON の間、 P11 の LED が点滅するプログラム */
int main()
{
    int i , j ; /* モータの PWM 制御にも使えるサンプル */
    /* データシート 9.1 ポート 1 を参照 */
    PMR1 = 0x00 ; /* 汎用入出力ポートに設定 */
    PCR1 = 0xff ; /* ポート 1 を全ビット出力用に使用 */
    PDR1 = 0x00 ; /* ポート 1 をクリア */
    /* データシート 9.3 ポート 5 を参照 */
    PMR5 = 0x00 ; /* 汎用入出力ポートに設定 */
    PCR5 = 0x00 ; /* ポート 5 を全ビット入力用に使用 */
    j = 0 ;
    while( 1 ) {
        if ( (PDR5 & P5IN) == 0 ) { /* &周りの () は必須 */
            /* P5IN が ON の時 */
            if ( j >= 500 )
                PDR1 = P1OUT ; /* j が 500 ~ 999 なら P1OUT を ON */
            else
                PDR1 = 0 ; /* j が 0 ~ 499 なら P1OUT を OFF */
        } else {
            /* P5IN が OFF の時 */
            PDR1 = 0 ;
        }
        /* 1msec ほどの処理 */
        for( i = 0 ; i < 1000 ; i++ )
            NOP();
        /* 0 ~ 999 の値を繰り返す */
        if( ++j > 1000 )
            j = 0 ;
    }
}

```



B 補足資料

以下に、実験の開発環境に依存する情報だけを、補足資料として抜粋した。

B.1 雛型プログラムのコピー

電子情報演習室の環境は Z: ドライブはファイル共有で自分専用のディレクトリとなっている。
Z: ドライブ配下に演習用のディレクトリを作り、雛型となるプログラム (以下のディレクトリ参照) を

```
¥ ¥10.120.11.60 ¥home ¥ei ¥t-saitoh ¥h8 ¥h8sample ¥h8car
```

まとめてその自分のディレクトリにコピーせよ。

編集するプログラムは、h8car.c である。

B.2 H8 プログラムのコンパイル手順

デスクトップの Cygwin というショートカットもしくは、スタートメニューのプログラムの中にある “Cygwin Bash Shell” を起動し、コマンドラインから命令を入力できる状態にする。

```
$ cd Z:/ 演習用ディレクトリ /h8car
$ ls -al          ディレクトリ内容の確認
$ make           必要なコンパイル作業を自動実行
```

B.3 H8 プログラムの転送

1. スタートメニューのプログラムの中にある H8write を起動
2. CPU の種類を H8/3664F を選択
3. CPU ボードの JP2,JP3 を Short にする。
4. CPU ボードのリセットを押す。
5. Z:/ 演習用ディレクトリ /h8car/h8car.mot を、ドラッグ&ドロップで H8write のウィンドウに入れる。
6. “OK” の確認によりプログラムが転送される。
7. CPU ボードの JP3 を抜き、リセットを押すとプログラムが起動する。