ＲＬ７８　マイコン学習セット　マニュアル　実用編

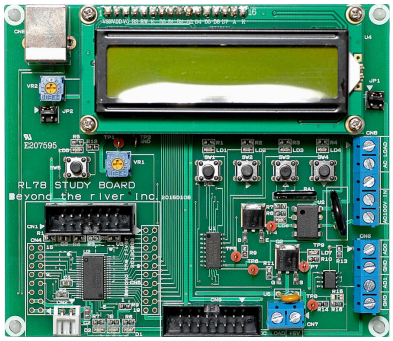
第１版２０１５．４．２３　　**第１版**

【　製品概要　】

本マニュアルはＲＬ７８／Ｉ１Ａ　Ｒ５Ｆ１０７ＤＥ（３８ピン）マイコンを使ったマイコン学習セットの添付ＣＤのサンプルプログラムの動作について解説されています。

実用編では入門編で見につけた知識を元に、実用になる機器のプログラムの書き方、ハードウエアの扱い方をサンプルプログラムから重点的に学習します。

※本学習セット開発にはルネサスエレクトロニクス社製Ｅ１が必要です。



実用

３．サンプルプログラム

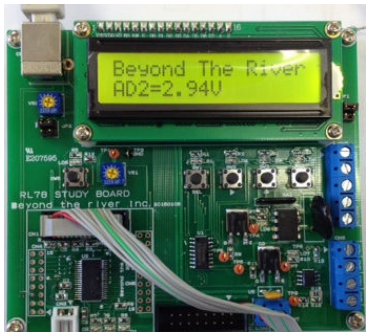
３＿１　　ＬＣＤ表示　電圧計　：ｓａｍｐｌｅ１０
３＿２　　ＰＷＭ　モーター速度制御　キーでＵＰ/ＤＯＷＮ　出力%表示
　　：ｓａｍｐｌｅ１１
３＿３　　ゲーム　スロットマシーン
　　：ｓａｍｐｌｅ１２
３＿４　　サーミスタ温度計測、ヒーター制御
　　：ｓａｍｐｌｅ１３　　ＯＮ/ＯＦＦ制御
　　：ｓａｍｐｌｅ１３＿ａ　　比例（Ｐ）制御

３．サンプルプログラム

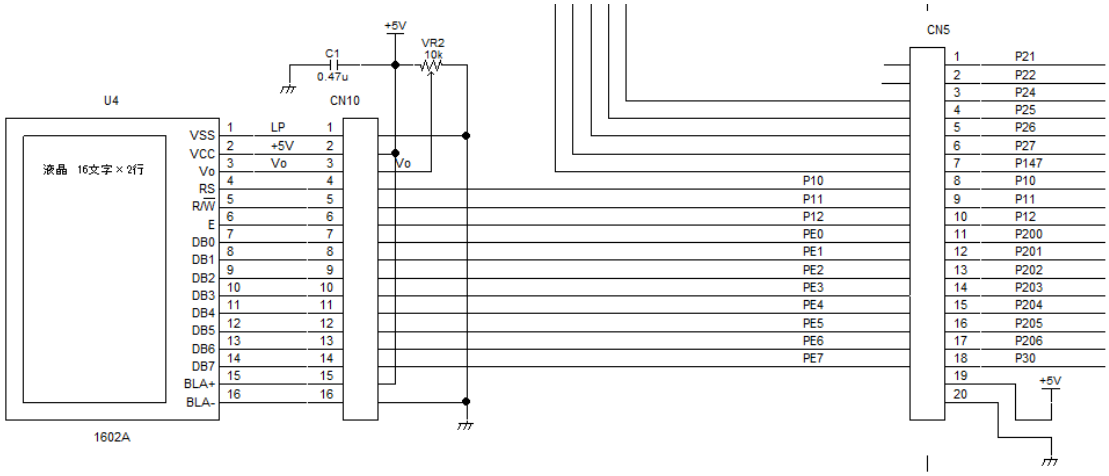
３－１　ｓａｍｐｌｅ１０　ＬＣＤ表示　電圧計

【　概要　】

２桁１６文字の液晶表示機にＢｅｙｏｎｄ　ｔｈｅ　Ｒｉｖｅｒ　と　ＡＩＮ２　の電圧値を表示します。



【　ハードウエア　】



使われているキャラクタタイプ液晶は１６０２Ａです。例えば秋月電子やストロベリーリナックスで販売しているものと使い方は同じです。理由はコントロールＩＣ、ＨＤ４４７８０（または互換品）がこのキャラクタＬＣＤ駆動のデファクトスタンダード（業界標準）だからで、世界中同じです。

８ビットバスモードで使用しています。回路図のＰＥ０～ＰＥ７がデータバス、Ｐ１０～１２がコントロール線です。データバスはリード、ライト時で方向が異なりますので、ポートのイン、アウトをライブラリ中で切り替えています。コントロール線は３本、ＲＳはデータ／コマンドの切替、Ｒ/__Ｗ（リード、ライト）、Ｅ信号は日立がモトローラ６８００系のセカンドソーサー（ライセンスを得て、互換品を作っているメーカー）だった時代の名残で、同期信号です。

ＶＲ２は液晶の輝度調整が出来ます。

【　ライブラリ　】

```
/* Start user code for include. Do not edit comment generated here */
#include "LCD_RL78107.h"
/* End user code. Do not edit comment generated here */
```

この１６０２Ａ用に弊社で用意したライブラリ、ＬＣＤ＿ＲＬ７８１０７．ｈがあります。使い方は本プログラム、解説を参考にして下さい。

【　プログラム　】

```
void main(void)
{
    R_MAIN_UserInit();
    /* Start user code. Do not edit comment generated here */

①      lcd_init();                          //LCD初期化
②      lcd_disp_0();                        //Beyond  the  river  液晶上段表示

    while (1U)
    {

                ad_cnt = 0;
③               R_ADC_Start();
                                                    //AD変換開始
                while(ADCS)
                                                    //変換待ち
                        ;

④               sdata = ad_data2 / 2.046;           //1023-＞500になるよう演算
                                                    //AD2 VR1
⑤               sprintf(lcd_buffer, "AD2=%04d¥n¥r",sdata); //lcd_buffに10進ASCII変換固定長セーブ

⑥               lcd_cursor_2();                     //液晶２段目表示

⑦               lcd_data_write(lcd_buffer[0]);
                lcd_data_write(lcd_buffer[1]);
                lcd_data_write(lcd_buffer[2]);
                lcd_data_write(lcd_buffer[3]);
                lcd_data_write(lcd_buffer[5]);
                lcd_data_write('.');
                lcd_data_write(lcd_buffer[6]);
                lcd_data_write(lcd_buffer[7]);
                lcd_data_write('V');
    }
    /* End user code. Do not edit comment generated here */
}
```

以下はｒ＿ｃｇ＿ａｄｃ＿ｕｓｅｒ．ｃの中の割り込みソフト

```
⑧__interrupt static void r_adc_interrupt(void)
{
    /* Start user code. Do not edit comment generated here */

        switch(ad_cnt)
        {
            case 0:R_ADC_Get_Result(&ad_data0);
        //AD読み込み ad_dataに&（アンパサンド）を付ける
                                    break;
            case 1:R_ADC_Get_Result(&ad_data1);
        //AD読み込み ad_dataに&（アンパサンド）を付ける
                                    break;
            case 2:R_ADC_Get_Result(&ad_data2);
        //AD読み込み ad_dataに&（アンパサンド）を付ける
                                    break;
            case 3:R_ADC_Get_Result(&ad_data3);
        //AD読み込み ad_dataに&（アンパサンド）を付ける
                                    break;

        }
        ad_cnt++;
        if(ad_cnt > 3){ad_cnt = 0;}

    /* End user code. Do not edit comment generated here */
}
```

【　簡単な解説　】

①　　lcd_init();　　//LCD初期化

LCD_RL78107.h　の中にある関数で液晶で使うポートを初期化しています。

②　　lcd_disp_0();　　//Beyond　the　river　液晶上段表示

液晶の上段にＢｅｙｏｎｄ　ｔｈｅ　ｒｉｖｅｒと表示させています。

```
    while (1U)
    {

            ad_cnt = 0;
③           R_ADC_Start();                  //AD変換開始
            while(ADCS)
                                            //変換待ち
```

Ａ／Ｄ変換開始、終了待ちで

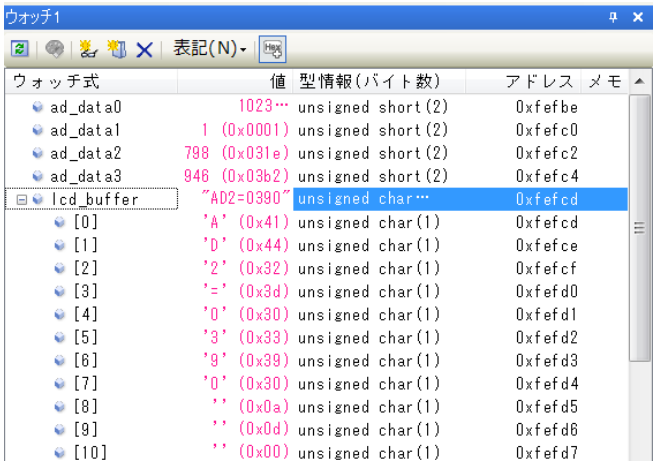
ＡＮＩ２のＡ／Ｄデータは割り込みでａｄ＿＿ｄａｔａ２に入ります。

④　　sdata = ad_data2 / 2.046;　　//1023-＞500 になるよう演算

０－１０２３を０－５．００Ｖ電圧表示に変換するために２．０４６で割っています。

⑤　　sprintf(lcd_buffer, "AD2=%04d¥n¥r",sdata);　//lcd_buffに10進ASCII変換固定長セーブ
まだ１６進数ですので、液晶に表示できるように１０進、８ビットのアスキーコードにｓｐｒｉｎｔｆ文で変換し、ｌｃｄ＿＿ｂｕｆｆｅｒにセーブしています。

ウオッチ１で確認するとａｄ＿＿ｄａｔａ２＝７９８（０ｘ０３ｂ２）です。２．０４６で割ると３９０．０。ｌｃｄ＿＿ｂｕｆｆｅｒにはＡＤ２＝０３９０が入っています。



⑥　　lcd_cursor_2();　　//液晶２段目表示
液晶表示カーソルを２行目にします。

⑦　　lcd_data_write(lcd_buffer[0]);
lcd_data_write(lcd_buffer[1]);
lcd_data_write(lcd_buffer[2]);
lcd_data_write(lcd_buffer[3]);
lcd_data_write(lcd_buffer[5]);
lcd_data_write('.');
lcd_data_write(lcd_buffer[6]);
lcd_data_write(lcd_buffer[7]);
lcd_data_write('V');
液晶に１文字ずつ表示しています。ｌｃｄ＿＿ｂｕｆｆｅｒ［４］の０は表示しません。小数点とＶをキャラクタで追加しています。

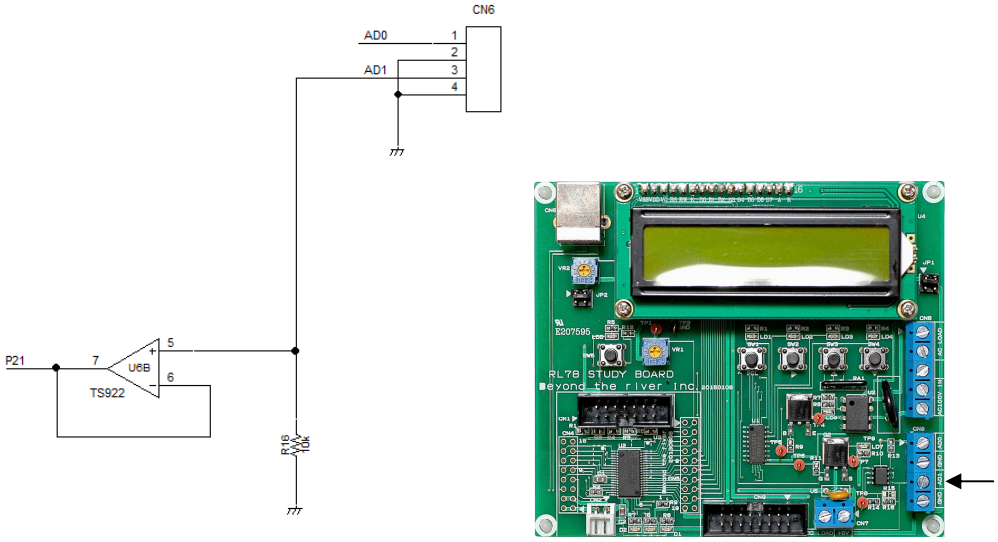
以下はｒ＿＿ｃｇ＿＿ａｄｃ＿＿ｕｓｅｒ．ｃの中の割り込みソフト。
カウンターａｄ＿＿ｃｎｔを０～３まで動かし、割り込みの度に＋１して、数によってＡＩＮ０，１，２，３　４ｃｈのデータをそれぞれａｄ＿＿ｄａｔａ０，１，２，３にセーブしています。

⑧__interrupt static void r_adc_interrupt(void)

```
{
    /* Start user code. Do not edit comment generated here */

            switch(ad_cnt)
            {
                    case 0:R_ADC_Get_Result(&ad_data0);
            //AD読み込み  ad_dataに&（アンバサンド）を付ける
                                            break;
                    case 1:R_ADC_Get_Result(&ad_data1);
            //AD読み込み  ad_dataに&（アンバサンド）を付ける
                                            break;
                    case 2:R_ADC_Get_Result(&ad_data2);
            //AD読み込み  ad_dataに&（アンバサンド）を付ける
                                            break;
                    case 3:R_ADC_Get_Result(&ad_data3);
            //AD読み込み  ad_dataに&（アンバサンド）を付ける
                                            break;

            }
            ad_cnt++;
            if(ad_cnt > 3){ad_cnt = 0;}

    /* End user code. Do not edit comment generated here */
}
```

【　応用　】

ＣＮ６のＡＤ１入力はＡＮＩ１（Ｐ２１）に接続されています。メインプログラムのａｄ＿＿ｄａｔａ２をａｄ＿＿ｄａｔａ１に書き換えてコンパイル、実行すれば外部の電圧を液晶に表示することが出来ます。

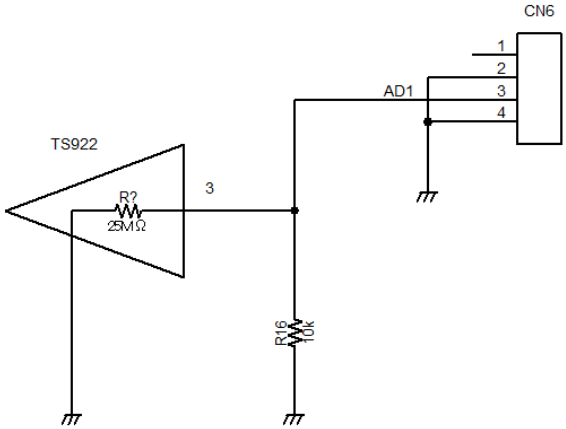


Ｕ６Ｂのオペアンプ ＴＳ９２２は入力に１Ｖが入ったときに出力に１Ｖを出します。電圧増幅が１倍です。電圧増幅が１倍で、何のために入っているか（入れる必要があるのか？、線でつないであるのと同じでは？）というとインピーダンス変換のために入っています。

入っている詳細は
１．このマイコンはアナログ入力源（ＡＮＩｘ）に出力インピーダンスは１ＫΩ以下のものを接続する、という決まりがあります。
２．オペアンプを入れることにより、接続されるもののインピーダンスが無視できるようになります。

１．の理由により、例えば１０ＫΩのボリウムを直接Ａ／Ｄの入力に接続しても正確に測定できない場合があります。

オペアンプＴＳ９２２は入力インピーダンスが非常に高く、出力インピーダンスが低いです。具体的には入力バイアス電流が最大でも１００ｎＡ（Ｖｏｕｔ＝Ｖｃｃ/２）ですので、Ｖｉｎを２．５Ｖ（Ｖｃｃ＝５Ｖ　ゲイン１）とすると２．５Ｖ／１００ｎＡ＝２５ＭΩです。ＣＮ６の３番はＲ１６　１０ＫΩにも接続されていますので、入力インピーダンスは１０ＫΩとなります。



ＣＮ６　３番からみた入力インピーダンスは２５ＭΩと１０ＫΩの並列ですので、２５Ｍ＊１０Ｋ／２５Ｍ＋１０Ｋ　＝　２．５[11]　／　２５０１００００　＝　９９９６．０Ω　≒　１０ＫΩ　＝　Ｒ１６の抵抗値となります。２５ＭΩ　＞＞　１０Ｋ　なので、接続されていないのと等価になります。

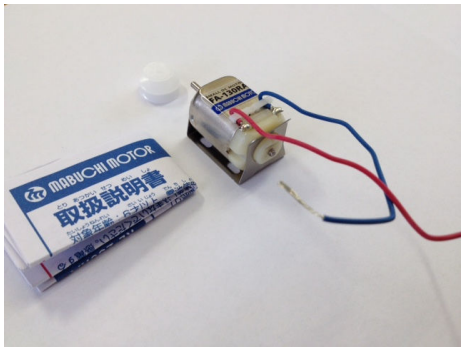
出力インピーダンスは負荷抵抗の変化時に出力電圧の変動の割合で計算しますが、限りなく０Ωになります。（許容出力電流内であれば出力抵抗の変化で、出力電圧は変わりません）
よって、１．の規定である１ＫΩ以下を十分満足します。

接続先の影響を受けないように入力インピーダンスは高く、接続先に影響を与えないように出力インピーダンスは低く、というのが電子回路接続の定石です（インピーダンス整合を除く）。

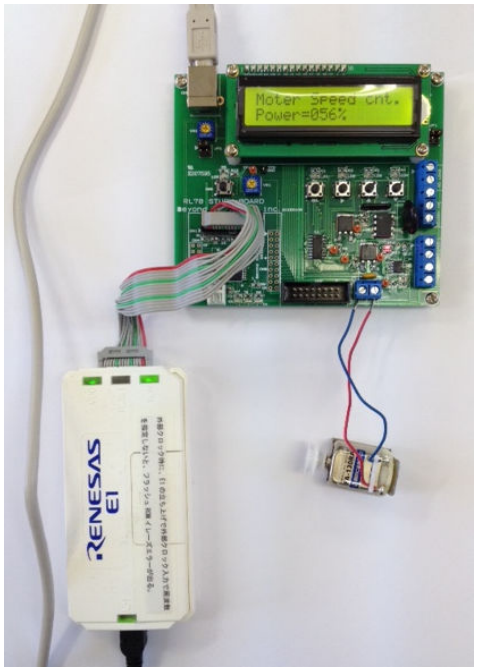
３－２　ｓａｍｐｌｅ１１　モーター速度制御　キーでＵＰ/ＤＯＷＮ　出力%表示

【　概要　】

ＤＣモーターをＰＷＭで速度制御して回してみます。添付のモーターＦＡ－１３０ＲＡを箱から出して、プーリーを取り付け、ＣＮ７の１番に青、２番に赤線を接続して下さい。

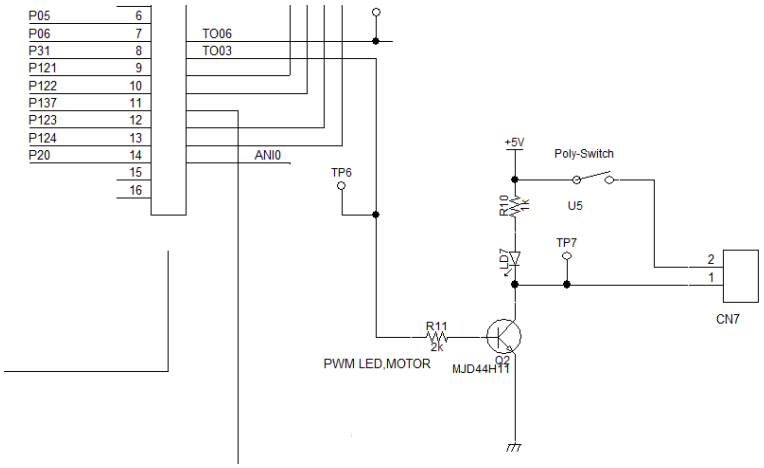


プログラムを動作させると初め０%出力ですので、モーターは回りませんが、ＳＷ１を押すと出力が増加していきます。５０%超えたあたりから回転が確認出来ます。減速はＳＷ２を押します。



【　回路図　】

ＬＥＤのときと違い、モーターを回す電流は＋５ＶからＰｏｌｙ－Ｓｗｉｔｃｈを通してＣＮ７の２番－モーター線赤－モータ線青－ＣＮ７の１番－Ｑ２のＣ（コレクタ）へと流れます。



Ｐｏｌｙ－Ｓｗｉｔｃｈは１種のヒューズですが、電流が減少すると復帰する面白いヒューズです。保守性に優れます。

以下省略

お問い合わせ先：
〒３５０－１２１３　埼玉県日高市高萩１１４１－１
TEL　０４２（９８５）６９８２
FAX　０４２（９８５）６７２０
Homepage：http//beriver.co.jp
e-mail：info@beriver.co.jp
有限会社ビーリバーエレクトロニクス　