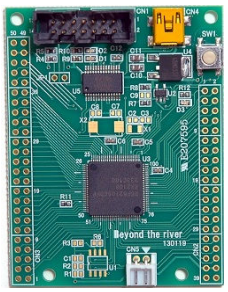
ＢＣＲＸ２１０　マイコン開発セット　マニュアル

第１版２０１３．３．１７　　**第１版**

【　製品概要　】

本マニュアルはＢＣＲＸ２１０　ＣＰＵボードのソフトウエア開発を行うために必要なソフトウエアインストゥール手順、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。本ＣＰＵボード開発にはルネサスエレクトロニクス社製Ｅ１が必要です。



１．開発環境、事前準備

１－１．　開発環境

ａ：開発セット　同梱物

ｂ：ＢＣＲＸ２１０　ＣＰＵボードの特徴

ｃ：デバッカＥ１

ｄ：無償のＨＥＷ、ＲＸ用Ｃコンパイラのダウンロード

ｅ：ＣＤコピー、デバイスドライバＤ２ＸＸのインストゥール

１－２　　動作、デバック

ａ：ＨＥＷ起動、コンパイル、書き込み、動作

ｂ：新しいプログラムを作る

２．サンプルプログラム

２－１．　ｓａｍｐｌｅ１　出力ポートのＯＮ，ＯＦＦ

２－２．　ｓａｍｐｌｅ２　ＳＩＯ（ＵＳＢ）でパソコンとやりとり

２－３．　ｓａｍｐｌｅ３　Ａ／Ｄ変換をＵＳＢ出力、Ｄ／Ａ出力

２－４．　ｓａｍｐｌｅ４　割り込み

２－５．　ｓａｍｐｌｅ５　新規ワークスペース製作見本

２－６．　ｓａｍｐｌｅ６　三角、対数、平方根関数を使う

１−１．開発環境

ａ：開発セット同梱物

ＲＸ２１０　ＣＰＵボード
ＣＤ（サンプルプログラム、デバイスドライバ、ドキュメント）
マニュアル（本誌）
電源ケーブル

※開発に必要なルネサスエレクトロニクス社製デバッカＥ１は同封されておりません。別途必要です。

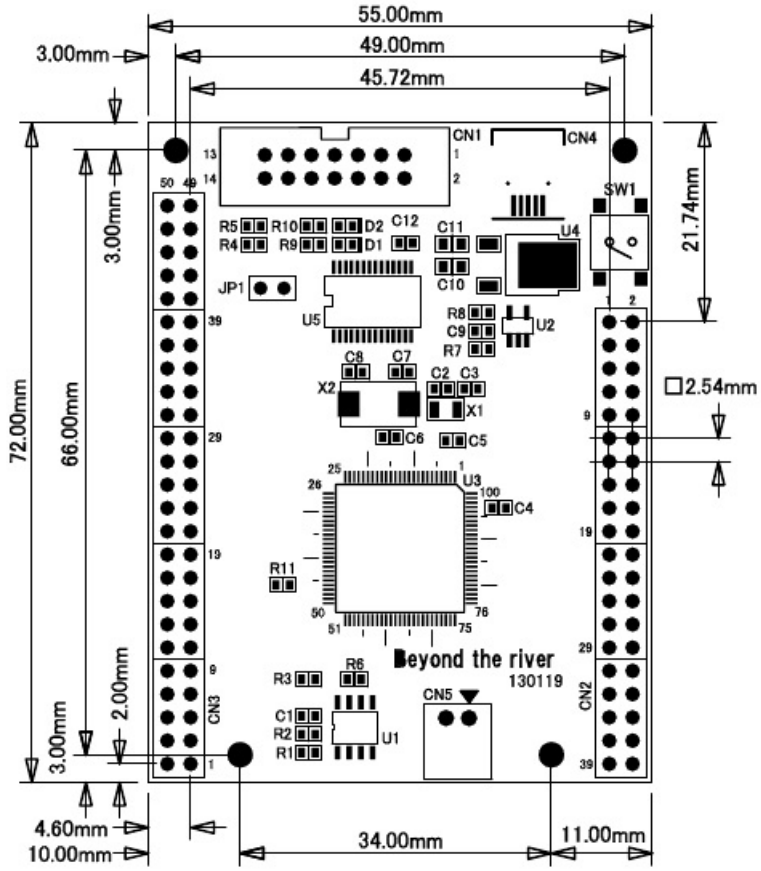
ｂ：ＢＣＲＸ２１０　ＣＰＵボードの特徴

●ＲＸアーキテクチャコア（３２ビットシングルチップＣＩＳＣ　最大７８ＤＭＩＰＳ　５０ＭＨｚ動作、３２ビット乗算器内蔵）ＲＸ２１０８　ＣＰＵを搭載。

ＲＸはルネサスの従来のマイコンに比べ、処理性能２倍、コード効率３０%向上、消費電力１／３、４倍の高集積を目標に設計されています。例えば１．６２Ｖで２０ＭＨｚ動作、２．７Ｖ以上で５０ＭＨｚという高速動作を実現。動作時わずか０．２ｍＡ／ＭＨｚ。デバイスでエコを実現しています。※１
データ処理でＨ８Ｓの３．６倍、コードサイズで３８%の向上を達成。
※１　３．３Ｖ、５Ｖどちらでもフルスペックで動きますが、消費電力、発熱を考えると３．３Ｖ動作がお勧めです。

●コンパクト５５×７２㎜、ＵＳＢインターフェイス。
●動作電圧３．３Ｖ～５．５Ｖ、３０ｍＡ　ＴＹＰＥ(３．３Ｖ、５０ＭＨｚ動作時）、ＵＳＢ接続時、５００ｍＡ迄であれば外部電源不要。但し、パソコンにより、ＵＳＢポートから５００ｍＡ供給できない場合もあります。
●大容量メモリ内蔵　Flash　512KB、RAM　64KB　R5F52108CDFM
●豊富な周辺機能
A/D 変換器：12 ビット、16ch、D/A 変換器：10 ビット 2ch、リアルタイムクロック内蔵、シリアル(SCI)：７ch、強力なタイマ：MTU2(16bit×6ch)、ウオッチドグタイマ、コンペアマッチタイマ(16bit×2ch)、温度センサ、コンパレータ内蔵
●USB ドライバ IC　FTDI 社の FT232RL 搭載。
●デバッカＥ１によるデバック用コネクタ搭載。
●オプションで外部クリスタル、シリアルフラッシュ ROM、シリアル EEPROM 搭載パッドあります。お問い合わせ下さい。

基板大きさ



ｃ：Ｅ１デバッカ

概要

E1 エミュレータは、ルネサス主要マイコンに対応したオンチップデバッギングエミュレータです。基本的なデバッグ機能を有した低価格の購入しやすい開発ツールで、フラッシュプログラマとしても使用可能です。

C 言語ソースデバックが可能で、1 行実行、ブレークポイント設定、変数、レジスタ、メモリ参照等々、従来であれば高価な ICE しか出来なかった機能が、安価に実現されています。また、使い方もＨＥＷ（統合開発環境）の E8a と同じで、経験があれば半日で、無くても 1 日で必要な操作を会得することが出来ると思います。

マイコンとの通信として、シリアル接続方式と JTAG 接続方式の２種類に対応しています。使用可能なデバッグインタフェースは、ご使用になるマイコンにより異なります。
また、基本デバッグ機能に加え、ホットプラグイン機能（動作中のユーザシステムに後から E1 エミュレータを接続して、プログラムの動作確認を行うことが可能）を搭載しているため、プログラムのデバッグ・性能評価に大きく貢献できます。

対応MPU

- V850 ファミリ
- RX ファミリ
- RL78 ファミリ
- R8C ファミリ
- 78K ファミリ

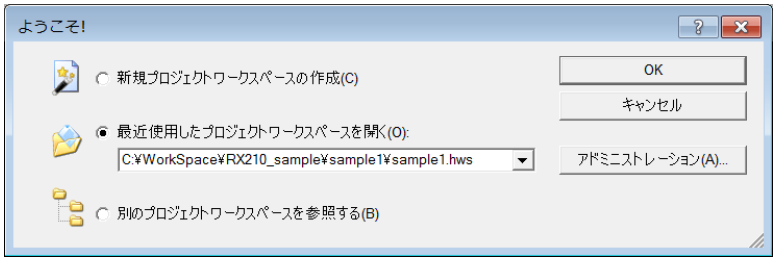
Ｅ１を購入するとＣＤが添付されていて、ドライバーのインストールとセルフチェックを行った後に、ネットから開発環境ＨＥＷとＣコンパイラのダウンロードを行います。

省略

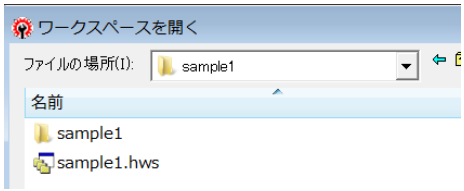
１－２　　動作、デバック

ａ：ＨＥＷ起動、コンパイル、書き込み、動作

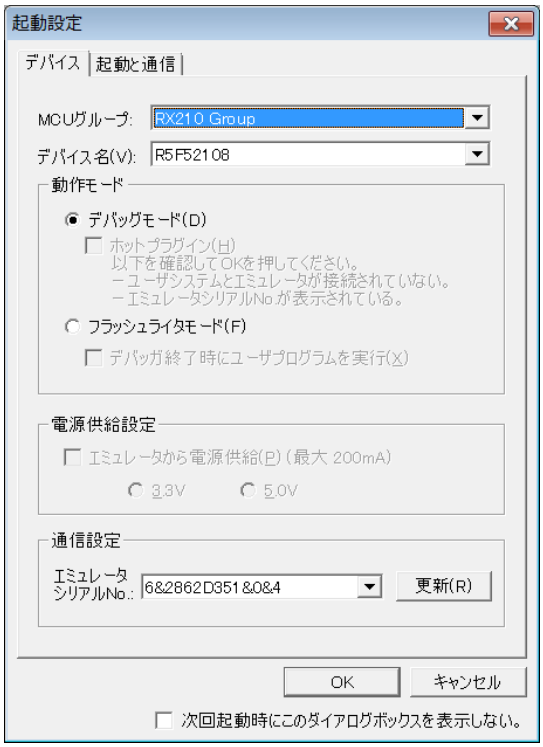
ＨＥＷを起動します。ここでは例としてＲＸ２１０￥ｓａｍｐｌｅ１を動作させます。初めてのときは「別のプロジェクトワークスペースを参照する」を選択し



拡張子ｈｗｓファイルをダブルクリック。以降、同じｓａｍｐｌｅ１でしたら「最近使用した、、」でＯＫです。

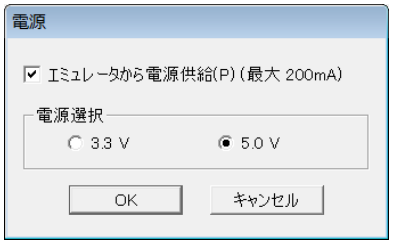


起動設定が表示されます。

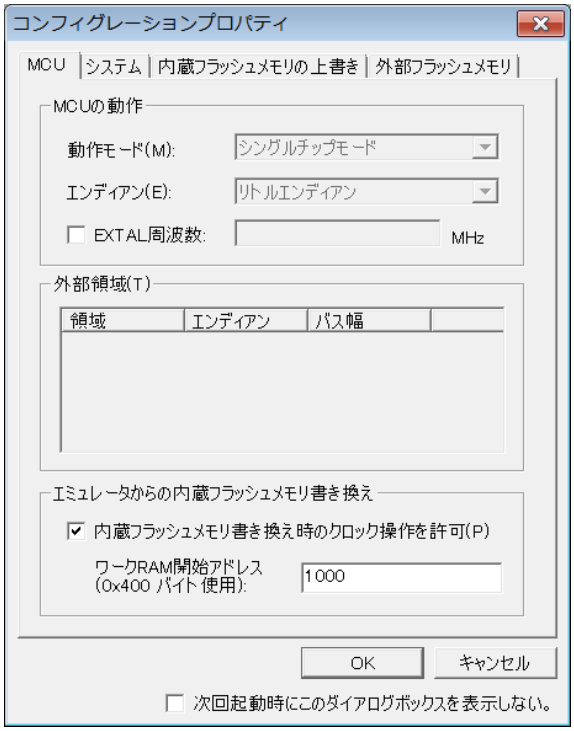


外部電源から電源を供給する場合はその電源をＯＮにします。ＵＳＢケーブルを挿入すると自動的にＵＳＢケーブルから電源３．３Ｖが供給されます。ＯＫをクリック。

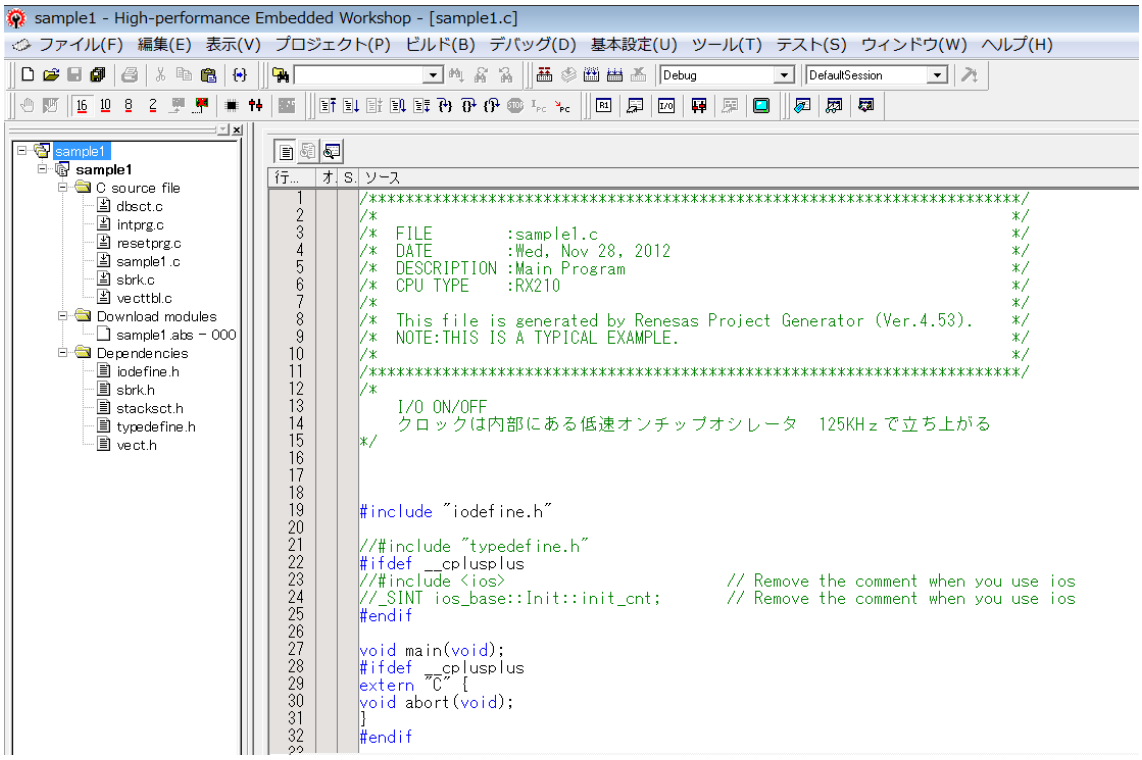
例ではＥ１から電源を供給してみます。本ボードは電源電圧３．３Ｖ、５Ｖいずれでも動作しますが、例では５Ｖを選択します。（多くの用途に消費電力、発熱にメリットのある３．３Ｖを推奨します）



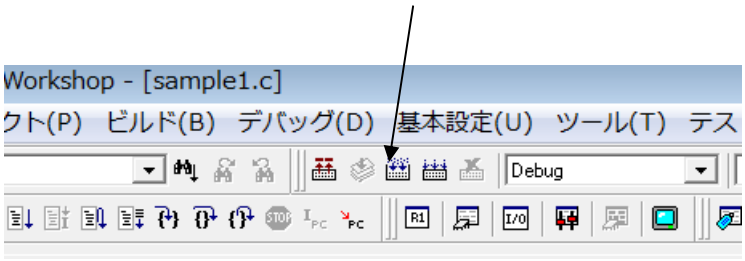
コンフィグレーションプロパティが表示されます。「内蔵フラッシュメモリ、、」にチェックが入っている必要があります。確認後、ＯＫ。



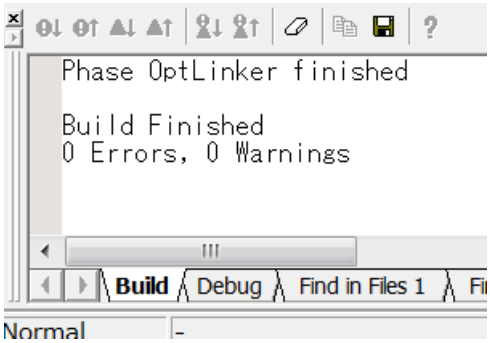
ｓａｍｐｌｅ１のエディタ画面等が表示されます。



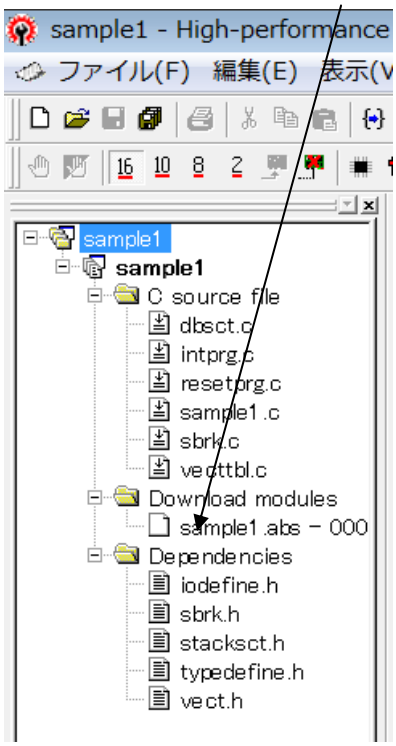
プログラムのコンパイルは、以下の「ビルド」ボタンで行います。ボタンはマウスを乗せると意味が表示されます。



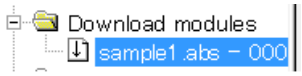
ｓａｍｐｌｅ１は出荷時にすでにコンパイラされていますので、０ Errors、０ Warnings と表示されます。新たにプログラムを製作してコンパイルしたときに Errors が０で無い場合、プログラムに文法上の問題がありますので、エディタでソースファイルを修正し、Errors ０にする必要があります。Errors が０で無い場合、書き込み用ファイル xxx．ｍｏｔが新たに作成されません。前のファイルは残りますので、勘違いして昔のファイルを書き込んでしまわないように注意が必要です。



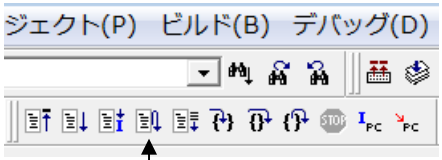
ｓａｎｐｌｅ１．ａｂｓ　　をダブルクリックします。



正常にダウンロードされると空白だった部分に↓マークが入ります。

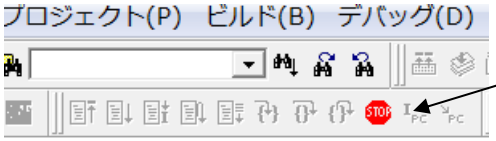


実行させてみます。



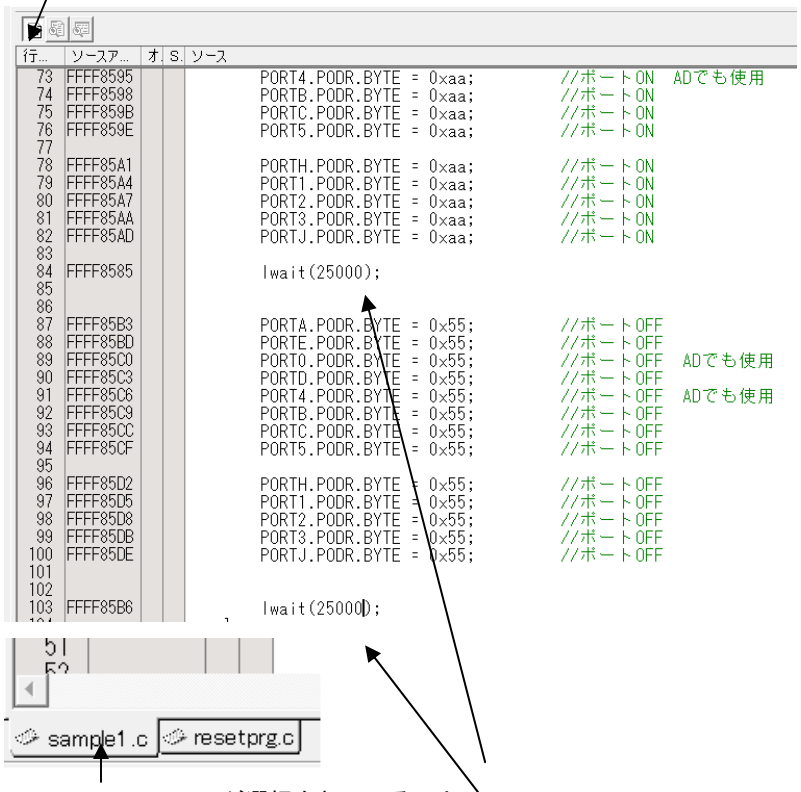
リセット後実行をクリック

ＣＰＵボードのＬＥＤが点滅しているのが見えると思います。停止はＳＴＯＰをクリックします。



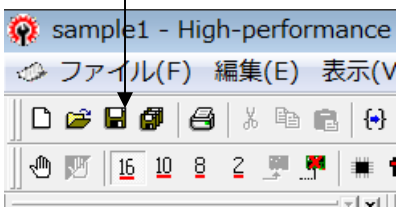
以上がプログラム開発に必要な「コンパイル」「書き込み」「実行」です。
例としてｓａｍｐｌｅ１を書き換えて、コンパイル、書き込み、動作の変化の確認、を行います。

Ｃソースファイルが選択されていること

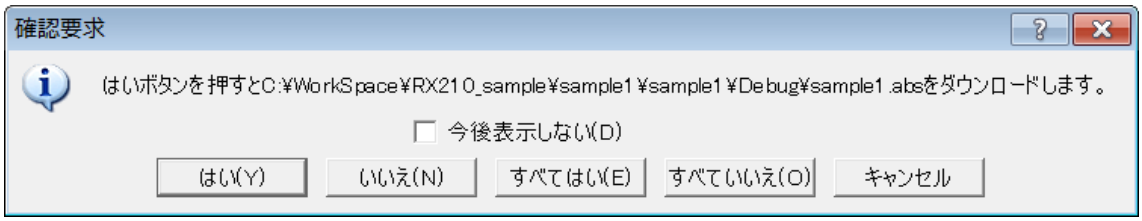


ｓａｍｐｌｅ１．ｃが選択されていること。
ｌｗａｉｔ(2500)の中の数値を１桁増やして 25000 にしてみます（２箇所）。プログラムをセーブします。

ファイルの保存はここをクリックします。エディタで書き込み、保存すると色が変わります（未保存の確認が出来ます）。



次ぎに、「コンパイル」し、エラーが無い場合、以下が表示されます。はい（Ｙ）をクリックでプログラムがダウンロードされます。



「リセット後、実行」をクリックします。ＬＥＤの点滅周期が遅くなったのがおわかりいただけると思います。

以上のように、プログラム開発は「エディタ（プログラム作成）」→「セーブ」→「コンパイル」→「エラーが無いことを確認」→「書き込み」→結果によって頭の「エディタ」に戻る繰り返しになります。

エディタは使い慣れたものでも使用可能で、その場合、ＨＥＷのエディタは使えなくなります。

省略

２．サンプルプログラム

２－１　ｓａｍｐｌｅ１　出力ポートのＯＮ，ＯＦＦ

```
/***********************************************************************/
/*                                                                     */
/*  FILE        :sample1.c                                             */
/*  DATE        :Wed, Nov 28, 2012                                     */
/*  DESCRIPTION :Main Program                                          */
/*  CPU TYPE    :RX210                                                 */
/*                                                                     */
/*  This file is generated by Renesas Project Generator (Ver.4.53).    */
/*  NOTE:THIS IS A TYPICAL EXAMPLE.                                    */
/*                                                                     */
/***********************************************************************/
/*
        I/O ON/OFF
        クロックは内部にある低速オンチップオシレータ　125KHｚで立ち上がる
*/

①#include "iodefine.h"

//#include "typedefine.h"
#ifdef __cplusplus
//#include <ios>                        // Remove the comment when you use ios
//_SINT ios_base::Init::init_cnt;       // Remove the comment when you use ios
#endif

void main(void);
#ifdef __cplusplus
extern "C" {
void abort(void);
}
#endif

②void lwait(long wait_time)
{
        while(wait_time != 0)
        {
                wait_time--;
        }
}



③void main(void)
```

```
{

④       PORTA.PDR.BYTE = 0xff;                  //全部出力　ポート検査用
        PORTB.PDR.BYTE = 0xff;                  //全部出力　ポート検査用
        PORTC.PDR.BYTE = 0xff;                  //全部出力　ポート検査用
        PORTD.PDR.BYTE = 0xff;                  //全部出力　ポート検査用
        PORTE.PDR.BYTE = 0xff;                  //全部出力　ポート検査用

        PORTH.PDR.BYTE = 0xff;                  //全部出力　ポート検査用
        PORTJ.PDR.BYTE = 0xff;                  //全部出力　ポート検査用


        PORT0.PDR.BYTE = 0xff;                  //全部出力　ポート検査用

        PORT1.PDR.BYTE = 0xff;                  //全部出力　ポート検査用
        PORT2.PDR.BYTE = 0xff;                  //全部出力　ポート検査用
        PORT3.PDR.BYTE = 0xff;                  //全部出力　ポート検査用

        PORT4.PDR.BYTE = 0xff;                  //全部出力　ポート検査用

        PORT5.PDR.BYTE = 0xff;                  //全部出力　ポート検査用

⑤       while(1)
        {
⑥               PORTA.PODR.BYTE = 0xaa;         //ポート出力
                PORTE.PODR.BYTE = 0xaa;         //ポート出力
                PORT0.PODR.BYTE = 0xaa;         //ポート出力        AD でも使用
                PORTD.PODR.BYTE = 0xaa;         //ポート出力
                PORT4.PODR.BYTE = 0xaa;         //ポート出力        AD でも使用
                PORTB.PODR.BYTE = 0xaa;         //ポート出力
                PORTC.PODR.BYTE = 0xaa;         //ポート出力
                PORT5.PODR.BYTE = 0xaa;         //ポート出力

                PORTH.PODR.BYTE = 0xaa;         //ポート出力
                PORT1.PODR.BYTE = 0xaa;         //ポート出力
                PORT2.PODR.BYTE = 0xaa;         //ポート出力
                PORT3.PODR.BYTE = 0xaa;         //ポート出力
                PORTJ.PODR.BYTE = 0xaa;         //ポート出力

⑦               lwait(2500);


⑧               PORTA.PODR.BYTE = 0x55;         //ポート出力
                PORTE.PODR.BYTE = 0x55;         //ポート出力
                PORT0.PODR.BYTE = 0x55;         //ポート出力　AD でも使用
```

```
                PORTD.PODR.BYTE = 0x55;                 //ポート出力
                PORT4.PODR.BYTE = 0x55;                 //ポート出力　AD でも使用
                PORTB.PODR.BYTE = 0x55;                 //ポート出力
                PORTC.PODR.BYTE = 0x55;                 //ポート出力
                PORT5.PODR.BYTE = 0x55;                 //ポート出力

                PORTH.PODR.BYTE = 0x55;                 //ポート出力
                PORT1.PODR.BYTE = 0x55;                 //ポート出力
                PORT2.PODR.BYTE = 0x55;                 //ポート出力
                PORT3.PODR.BYTE = 0x55;                 //ポート出力
                PORTJ.PODR.BYTE = 0x55;                 //ポート出力


                lwait(2500);
        }
}
void abort(void)
{

}
#endif

/***********************************************************************************
Option-Setting Memory
***********************************************************************************/
⑨#pragma address OFS1_REG = 0xFFFFFF88   /* OFS1 register */
const unsigned long OFS1_REG = 0xFFFFFFFF;

#pragma address OFS0_REG = 0xFFFFFF8C   /* OFS0 register */
const unsigned long OFS0_REG = 0xFFFFFFFF;
```

【　解説　】

本プログラムは、出荷検査のポート検査用に作られたもので、全てのポートをＯＮ，ＯＦＦさせています。ＲＸ２１０は内部にクロック発振子を内蔵していて、電源立ち上がり時には低速オンチップオシレータ１２５ＫＨｚで動作しています。動作速度と発振精度をそれほど要求しない動作には消費電力も少なく、良いモードです。

①#include "iodefine.h"
各レジスタの定義が入っているヘッダです。

```
②void lwait(long wait_time)
{
        while(wait_time != 0)
        {
```

```
                wait_time--;
        }
}
```
メイン関数で使用するＬＥＤのＯＮ，ＯＦＦ時間を設定するウエイトです。

```
③void main(void)
{
```
メインルーチンです。

```
④       PORTA.PDR.BYTE = 0xff;                          //全部出力　ポート検査用
```
ビットに１を立てて、全て出力にしています。０を設定すると入力になります。

```
⑤       while(1)
        {
```
ここから無限ループです。

```
⑥               PORTA.PODR.BYTE = 0xaa;                 //ポート出力
```
０ｘａａを出力しています。

```
⑦               lwait(2500);
```
ウエイトです。２５００が０になるまで戻ってきません。

```
⑧               PORTA.PODR.BYTE = 0x55;                 //ポート出力
```
０ｘ５５を出しています。２進数で書くと０ｂ０１０１０１０１です。前が０ｘａａでしたが２進数表記では０ｂ１０１０１０１０となります。お互いを反転させた値をポートに書き込んでいます。

０ｘａａ　＝　～０ｘ５５；

以下省略

２－６　三角、対数、平方根関数を使う

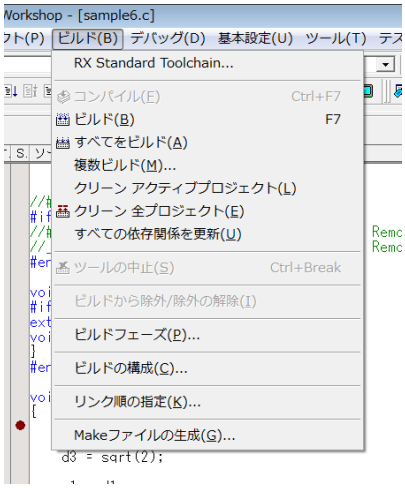
```
/************************************************************************/
/*                                                                      */
/*  FILE        :sample6.c                                              */
/*  DATE        :Tue, Apr 02, 2013                                      */
/*  DESCRIPTION :Main Program                                           */
/*  CPU TYPE    :RX210                                                  */
/*                                                                      */
/*  This file is generated by Renesas Project Generator (Ver.4.53).     */
/*  NOTE:THIS IS A TYPICAL EXAMPLE.                                     */
/*                                                                      */
/************************************************************************/

①#include <math.h>

②double d1,d2,d3;
short s1,s2,s3;

③#define PI 3.14159265


//#include "typedefine.h"
#ifdef __cplusplus
//#include <ios>                        // Remove the comment when you use ios
//_SINT ios_base::Init::init_cnt;       // Remove the comment when you use ios
#endif

void main(void);
#ifdef __cplusplus
extern "C" {
void abort(void);
}
#endif

void main(void)
{
④       d1 = log10(10000);
        d2 = sin((PI/180)*45);
        d3 = sqrt(2);

⑤       s1 = d1;
        s2 = d2;
        s3 = d3;

        while(1)
```

```
                    ;
}

#ifdef __cplusplus
void abort(void)
{

}
#endif
```

【　解説　】

①#include <math.h>

三角関数や、対数、平方根を使うためにはｍａｔｈ．ｈをインクルードする必要があります。加えて、ビルド（Ｂ）→　ＲＸ　Ｓｔａｎｄａｒｄ　Ｒｏｏｌｃｈａｉｎ、、をクリック。



以下省略

②double d1,d2,d3;
short s1,s2,s3;

結果を入れるｄｏｕｂｌｅ（３２ビット）３つ、キャストするｓｈｏｒｔ（１６ビット）３つをここで定義しています。

③#define PI 3.14159265

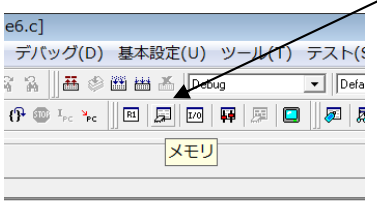
三角関数計算で角度を入力して数値を出すために使います。

```
④       d1 = log10(10000);
        d2 = sin((PI/180)*45);
        d3 = sqrt(2);
```

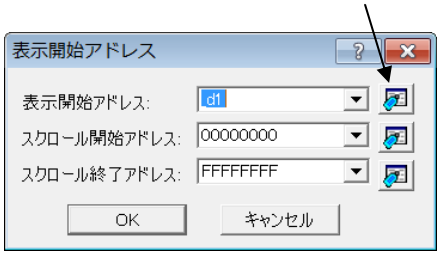
頭から

ｄ１＝ｌｏｇ１０（１００００）、答えは４になるはずです。

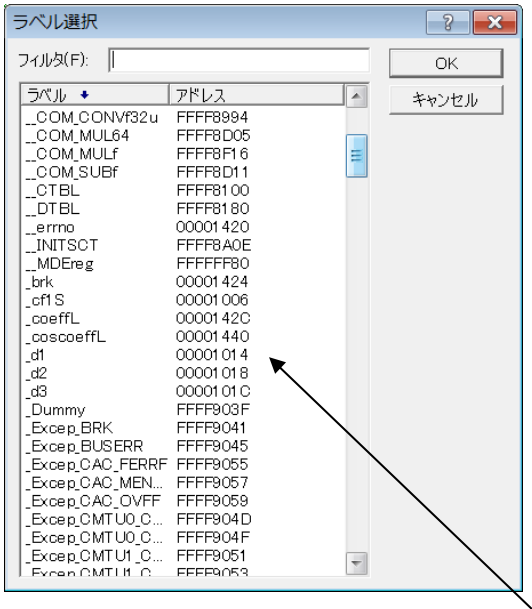
ｄ２＝ｓｉｎ（（ＰＩ／１８０＊４５）→ｓｉｎ（４５°）という意味です。答えは０．７０７１０６、、になるはずです。
ｄ３＝ｓｑｒｔ（２）→平方根の２です。答えは１．４１４２１３５６になるはずです。
ｄ１、ｄ２、ｄ３の結果を見るために、メモリをクリック



変数名ｄ１を探します。ここをクリック。



ラベル名一覧が表示されます。

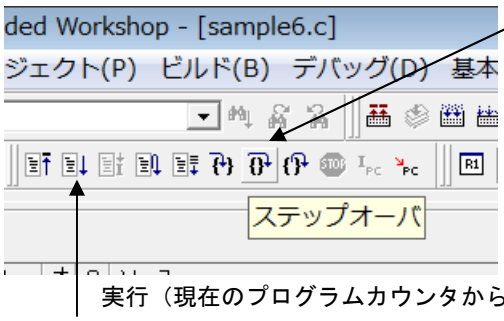


ｄ１，ｄ２，ｄ３共連続した４バイトとびにあるのが確認出来ます。ｄ１をクリック。
同様の方法でｓ１、ｓ２、ｓ３も見れるように二つの窓を開いておいて下さい。

例えば下記２箇所にブレークポイントを設定します。「リセット後、実行」をクリックします。

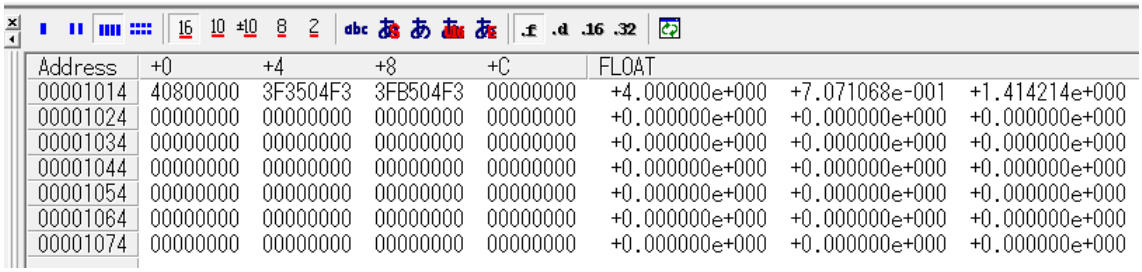
```
36  FFFF861C    void main(void)
37              {
38  FFFF861C        d1 = log10(10000);
39  FFFF862E        d2 = sin((PI/180)*45);
40  FFFF8640        d3 = sqrt(2);
41
42  FFFF8650        s1 = d1;
43  FFFF8666        s2 = d2;
44  FFFF867C        s3 = d3;
45
46  FFFF868A        while(1)
47                      ;
48              }
```

はじめは３８行で止まるはずです。それをステップオーバーまたは実行で４６行まで実行させます。



実行（現在のプログラムカウンタから実行を継続する）

ｄ１、ｄ２、ｄ３はどうなっているでしょうか？



Address	+0	+4	+8	+C	FLOAT		
00001014	40800000	3F3504F3	3FB504F3	00000000	+4.000000e+000	+7.071068e-001	+1.414214e+000
00001024	00000000	00000000	00000000	00000000	+0.000000e+000	+0.000000e+000	+0.000000e+000
00001034	00000000	00000000	00000000	00000000	+0.000000e+000	+0.000000e+000	+0.000000e+000
00001044	00000000	00000000	00000000	00000000	+0.000000e+000	+0.000000e+000	+0.000000e+000
00001054	00000000	00000000	00000000	00000000	+0.000000e+000	+0.000000e+000	+0.000000e+000
00001064	00000000	00000000	00000000	00000000	+0.000000e+000	+0.000000e+000	+0.000000e+000
00001074	00000000	00000000	00000000	00000000	+0.000000e+000	+0.000000e+000	+0.000000e+000

左からｄ１、ｄ２、ｄ３の４バイトＨＥＸデータです。右が浮動小数点付きデータ表示です。答えは合っていますね。

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　