Ｒ８Ｃ／Ｍ１２Ａ　　反射神経ゲーム　取扱説明書

第１版２０１１．１１．２３　**第１版**

【　製品概要　】

反射神経ゲームはルネサスエレクトロニクス社の２０ピンＤＩＰマイコン、Ｒ８Ｃ/Ｍ１２Ａを使用し、ＬＥＤの点灯スキャンとゲーマーが指で押すスイッチのタイミングで「当たり」をゲットするゲームです。スキャン速度を可変する切り替えスイッチが付いています。難易度最高レベルではかなりの反射神経を必要とします。目で確認し、指で動作させます。

特徴は以下の通りです。

１．ゲームとして楽しめます。
　キットではなく完成品です。プログラム書き込み済み、単三電池２本ご用意いただければ、すぐに遊べます。

２．反射神経の向上、回復、リハビリなどに使えます。若い方からご年配の方まで楽しんでいただけます。

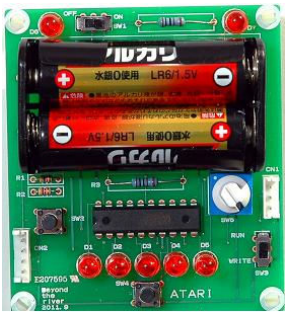
以下はマイコン学習に最適な特徴です。

３．動作プログラムが解説されているので、プログラム作成の方法が理解ができます。（Ｃ言語で記述されています）

４．別売の書き込み用Ｖケーブルを用意すれば、他は無償の開発環境でプログラムを好きなようにカスタマイズできます。

５．Ｅ８ａを使用した、ブレークポイントや変数値を参照できる高度なデバックにも対応できます。

本マニュアルは　反射神経ゲームの遊び方、プログラム動作解説、プログラム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、について解説されています。



１．遊び方

１－１．　各部の機能と名称

１－２．　遊び方

２．プログラム動作解説

３．プログラムを自分で改造してみる

３－１．　開発環境の準備

ａ：無償版ＨＥＷ、Ｒ８Ｃ用Ｃコンパイラのダウンロード
ｂ：書き込みツールＦＤＴのダウンロード
ｃ：プログラム、デバイスドライバのダウンロード、インストゥール

３－２．　ＨＥＷコンパイル、フラッシュＲＯＭ書き込み

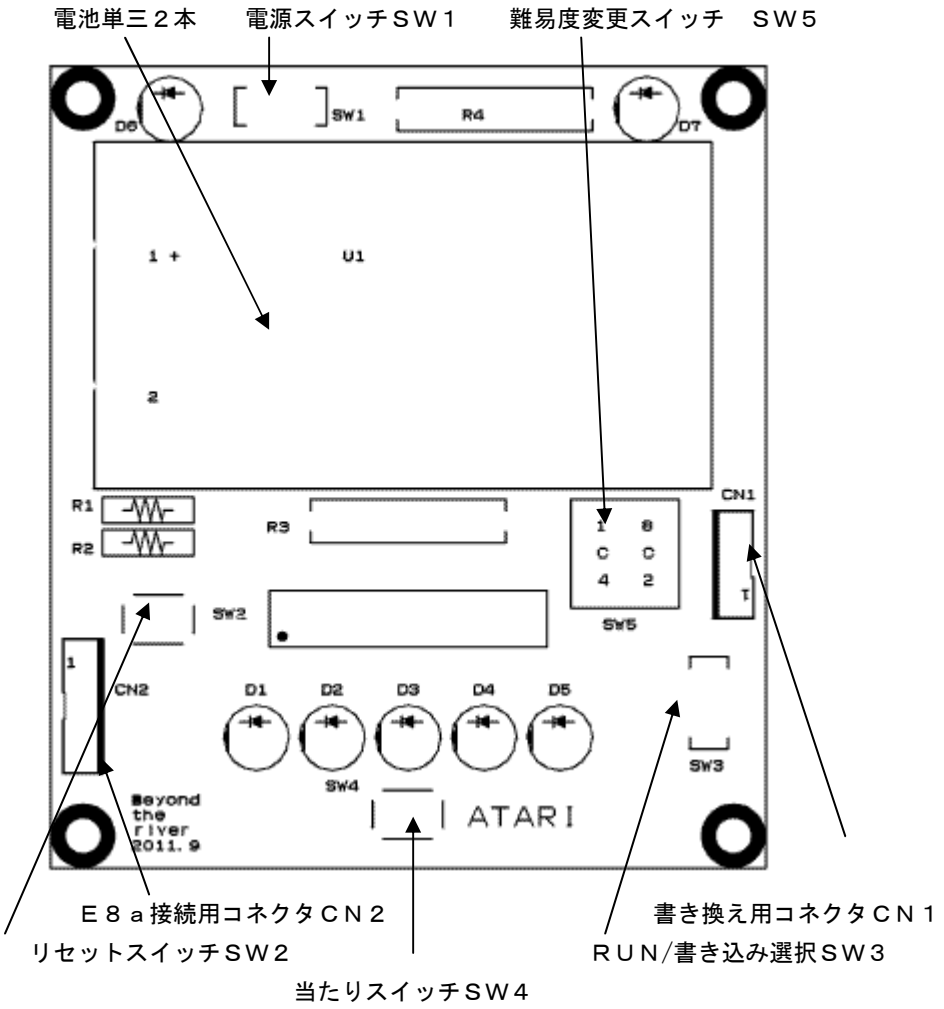
ａ：フラッシュＲＯＭライタ準備
ｂ：ＨＥＷ起動、コンパイル、書き込み、動作
ｃ：ＦＤＴでプログラムを書き込む
ｄ：プログラムを変更してみる
ｅ：新しいプログラムを作る

３－３．　Ｅ８ａでの動作、デバック

ａ：Ｅ８ａについて
ｂ：デバックの概要
ｃ：説明のアイコンが見当たらない場合

１．遊び方

１－１．　各部の機能と名称



１－２．　遊び方

ａ．難易度設定スイッチを０に設定してください。
ｂ．ＲＵＮ/ＷＲＩＴＥスイッチをＲＵＮ側
ｃ．電源スイッチＳＷ１をＯＮ

当たりスイッチＳＷ４を押すとＤ１からＤ５の方向にＬＥＤが点滅します。Ｄ３が点灯したときに、ＳＷ４が押されると「当たり」でＬＥＤ１～７まで５回、０．２秒間隔で点滅します。初めからＳＷ４を押していても「当たり」ません。

難易度設定が４まで、点滅周期が早くなり、難易度が上がります。５からは、Ｄ３とＤ６，Ｄ７の３つのＬＥＤが点灯時のみ、ＳＷ４を受け付けます。６、７、８、９と周期が早くなります。

２．プログラム動作解説

```
/***********************************************************************/
/*                                                                     */
/*  FILE        :R8CM12_game.c                                         */
/*  DATE        :Wed, Nov 23, 2011                                     */
/*  DESCRIPTION :main program file.                                    */
/*  CPU GROUP   :M12A                                                  */
/*                                                                     */
/*  This file is generated by Renesas Project Generator (Ver.4.19).    */
/*  NOTE:THIS IS A TYPICAL EXAMPLE.                                    */
/***********************************************************************/

①#include "sfr_r8m12a.h"


//割り込み処理
②#pragma interrupt _timer_rc(vect=7)

③volatile unsigned char cdata;
volatile unsigned short int_timer,int_timer_d6,int_timer_d7;

④#define D1_ON p1_0 = 0
#define D1_OFF p1_0 = 1
#define D2_ON p1_1 = 0
#define D2_OFF p1_1 = 1
#define D3_ON p1_2 = 0
#define D3_OFF p1_2 = 1
#define D4_ON p1_3 = 0
#define D4_OFF p1_3 = 1
#define D5_ON p3_3 = 0
#define D5_OFF p3_3 = 1
#define D6_ON p3_4 = 0
#define D6_OFF p3_4 = 1
#define D7_ON p3_5 = 0
#define D7_OFF p3_5 = 1

⑤#define t_1byou 100
#define t_2byou 200
#define t_3byou 300
#define t_4byou 300
#define t_500mbyou 50
#define t_200mbyou 20
#define t_100mbyou 10
#define t_50mbyou 5
```

⑥void _timer_rc(void)//約 10msec
{

//　　p1_0 = 1;　　//マーカー

　　imfa_trcsr = 0;　　//IMFA flag clear

　　if(int_timer != 0){int_timer--;}
　　if(int_timer_d6 != 0){int_timer_d6--;}
　　if(int_timer_d7 != 0){int_timer_d7--;}

//　 p1_0 = 0;　　//マーカー
}

⑦void timer_init(void)
{
　// timer on/off
　cts_trcmr = 0;　　// Stop TRC Count

　// timer RC enable
　msttrc = 0;　　//

　// pwm mode setup
　pwmb_trcmr = 0;　　// TRCIOB PWM mode
　pwmc_trcmr = 0;　　// TRCIOC PWM mode
　pwmd_trcmr = 0;　　// TRCIOD PWM mode
　pwm2_trcmr = 1;　　// PWM2 mode selection bit:PWM mode

　bufea_trcmr = 0;　　// TRCGRC:General register
　bufeb_trcmr = 0;　　// TRCGRD:General register

　cclr_trccr1 = 1;　　// The TRC register clear at the compare match with TRCGRA

　// count value setup
　trccnt = 0;　　//
　trcgra = 1250+120;　　// 初めは 125KHｚである　8μsec　10msec として

　　//+125 は補正数

　// interrupt settings
　//割り込みレベル設定
　ilvl35 = 1; //レベル 2

```
    ilvl34 = 1;
    imiea_trcier = 1;                                   // imiea enable interrupts


    cts_trcmr = 1;                              // Start TRC Count


}
```

【　上記までの解説１　】

①#include "sfr_r8m12a.h"
これはポートの絶対アドレス等を記入したヘッダファイルで、必ず、この１行を加える必要があります。

//割り込み処理
②#pragma interrupt _timer_rc(vect=7)

本プログラムではタイマーで定周期割り込みをかけています。その割り込み発生時に実行されるプログラム名です。

③volatile unsigned char cdata;
volatile unsigned short int_timer,int_timer_d6,int_timer_d7;
これは本プログラムで使用している変数の指定です。Volatile が無いと、コンパイラのほうで勝手にコンパイルしない可能性があります。

④#define D1_ON p1_0 = 0
LED の D1 を ON にするためには p1_0 ポートを０にする必要があります。（別紙回路図参照）このように定義するとそれをプログラムでは D1_ON と書けばよくなりますので、間違いが少なくなる上に、覚えるのが簡単です。

⑤#define t_1byou 100
②の割り込み関数は約 10msec に１回実行されます。１０ｍｓｅｃ×１００＝１秒。

以下省略

３－１．　開発環境の用意

ａ：無償版ＨＥＷ、Ｒ８Ｃ用Ｃコンパイラのダウンロード

　プログラムの開発は例えばルネサスエレクトロニクス社の統合開発環境ＨＥＷでＣ言語を用い動作させることができます。ＣＤ添付のサンプルプログラムはこの環境下で作成されています。無償版をダウンロードして使用します。

　ネット検索で→ルネサス　マイコン　Ｒ８Ｃ　と入力し、関連リンク→無償評価版ソフトウエアダウンロードをクリック。

→Ｍ１６Ｃシリーズ、Ｒ８Ｃファミリ用　Ｃ／Ｃ＋＋コンパイラパッケージをダウンロードします。

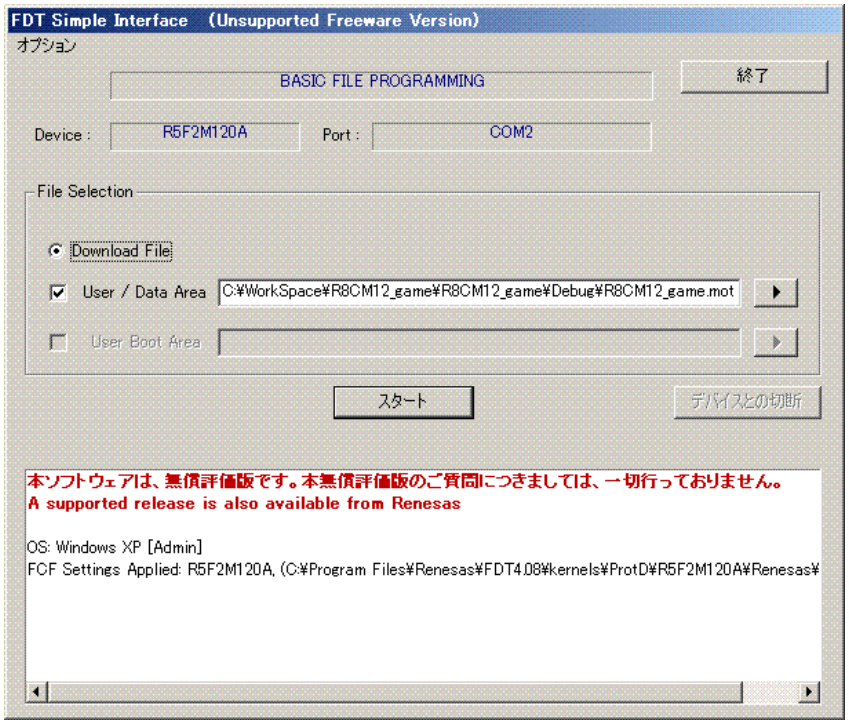
	M16Cシリーズ, R8Cファミリ用 C/C++コンパイラパッケージ (M3T-NC30WA) 製品ページ 評価版ダウンロード 製品版をご購入の際は、R8C, M16Cファミリ用C/C++コンパイラパッケージをご注文ください。詳細は 製品ページを参照ください。	V.6.00 Release 00 • 試用期限内は製品版と同じ。 • 試用期限を過ぎるとリンクサイズが64Kバイト以内に制限されます。 • High-Performance Embedded Workshopおよびシミュレータデバッガを同梱。	60日 初めて評価版ソフトウェアツールをインストールした後、最初にビルドを行った日から60日。 *一度でもビルドを行った場合は評価版ソフトウェアツールをインストールしなおしても試用期限の延長はできません。

　無償版は６０日経過後、リンクサイズが６４ＫＢと制限されますが、ＲＯＭ容量２ＫＢのＲ８Ｃ／Ｍ１２Ａにとっては事実上、有償版と違いがありません。
統合開発環境ＨＥＷとＣコンパイラがインストゥールされます。

ｂ：書き込みツールＦＤＴのダウンロード

Ｒ８Ｃ/Ｍ１２にプログラムを書く方法は
１．ルネサスのサイトからＦＤＴを無償ダウンロードして使用する。
２．Ｅ８ａを使う
の二つがあります。ここではＦＤＴで書く方法を説明します。別売の弊社Ｖケーブルが必要です。

ＦＤＴ　ダウンロード　で検索すると画面が表示されますので、ダウンロードしてください。
２種類のＦＤＴがありますが、ＣＯＭポートを使うＢＡＳＩＣを使用します。
画面を以下のように設定します。変更は「オプション」を操作することで可能です。
ＣＯＭポートの確認、変更は別紙「ＣＯＭ番号を調べる」を参照下さい。下例ではＣＯＭ２で設定しています。



詳細はダウンロード添付のマニュアルrjj10i0142_0806um.pdf「M16C Flash Starterユーザースマニュアル」をご参照下さい。

ｃ：プログラム、デバイスドライバのダウンロード、インストゥール

　購入時、添付されるダウンロードパスワードで、弊社サイトよりプログラムとデバイスドライバのダウンロードを行います。
プログラムＲ８ＣＭ１２＿ｇａｍｅホルダは C:¥WrokSpace¥にコピーしてください。WorkSpace は HEW をインストゥールすると自動形成されます。

次に、プログラムの書き込をＶケーブルで行う場合、デバイスドライバを以下の手順でインストゥールして下さい。

以下省略

３－２　　ＨＥＷコンパイル、フラッシュＲＯＭ書き込み

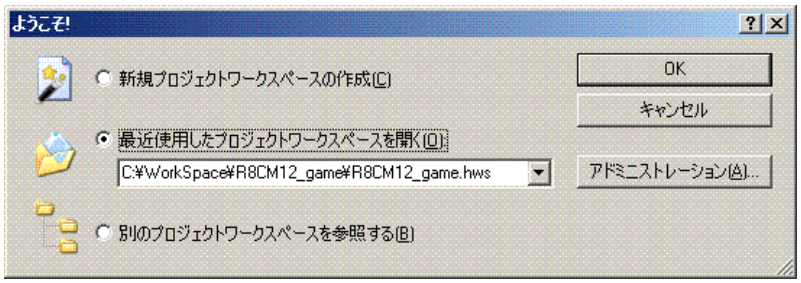
ａ：フラッシュＲＯＭライタ準備

インストゥールしたＦＤＴ　ＢＡＳＩＣのショートカットを表画面に出します。

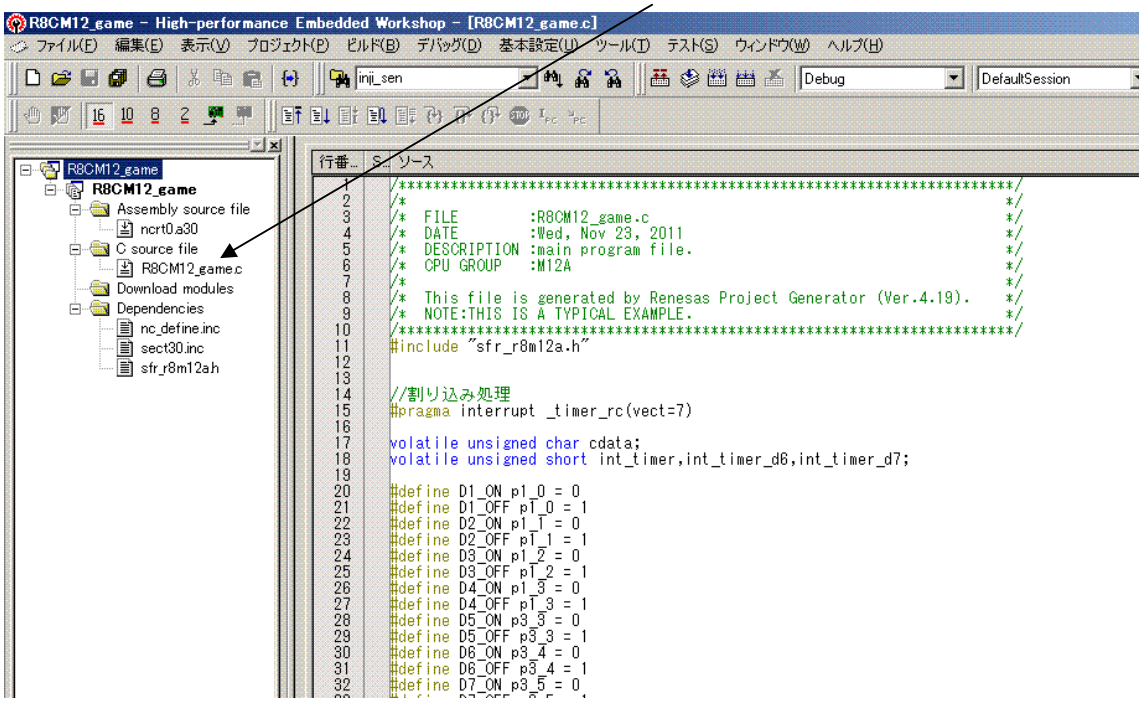
ｂ：ＨＥＷ起動、コンパイル、書き込み、動作

ＨＥＷを起動します。

初めてのときは、「別のプロジェクトを参照する」をクリックし、「ＯＫ」をクリックして C:¥WorkSpace¥にセーブした C:¥WorkSpace¥R8CM12_game¥R8CM12_game.hws を選択します。次回からは「最近使用した、、」を選択し、「ＯＫ」をクリックするだけで使用できます。



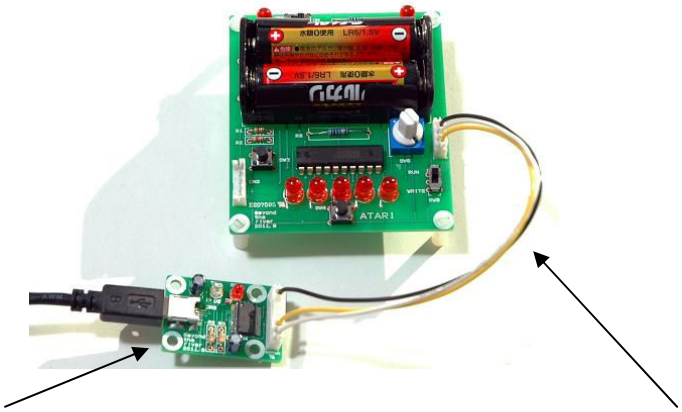
ソースファイルが表示されます。表示されない場合、Ｒ８ＣＭ１２＿ｇａｍｅ.ｃをダブルクリックします。



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

コンパイルするとＲ８ＣＭ１２＿ｇａｍｅ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うことがあるので注意が必要です。

ｃ：ＦＤＴでプログラムを書き込む



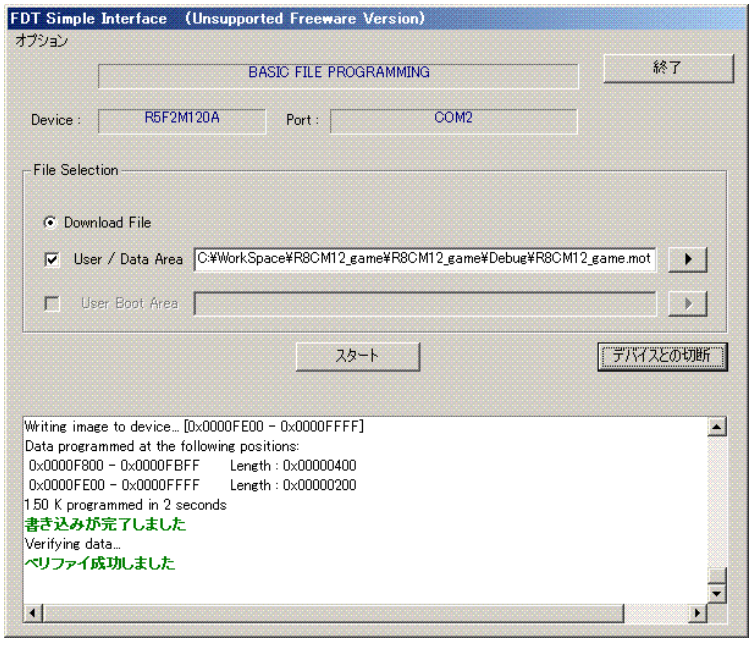
ＶケーブルをパソコンからのＵＳＢケーブルに挿入し、Ｒ８Ｃアダプタをゲーム機に接続します。電源は電池を挿入してください。

１．電源スイッチＳＷ１はＯＮ

２．動作切り替えＳＷ３はＷＲＩＴＥ側

３．ＲＥＳＥＴスイッチＳＷ２を１回押す

これで準備完了です。



書き込むファイルは

C:¥WorkSpace¥R8CM12_game¥R8CM12_game¥Debug¥R8CM12_game.mot です。

「スタート」をクリックし、問題なければ最後の「書き込みが完了しました」「ベリファイに成功しました（選択した場合）」が表示されます。電池が弱っていたりすると失敗することがあります。

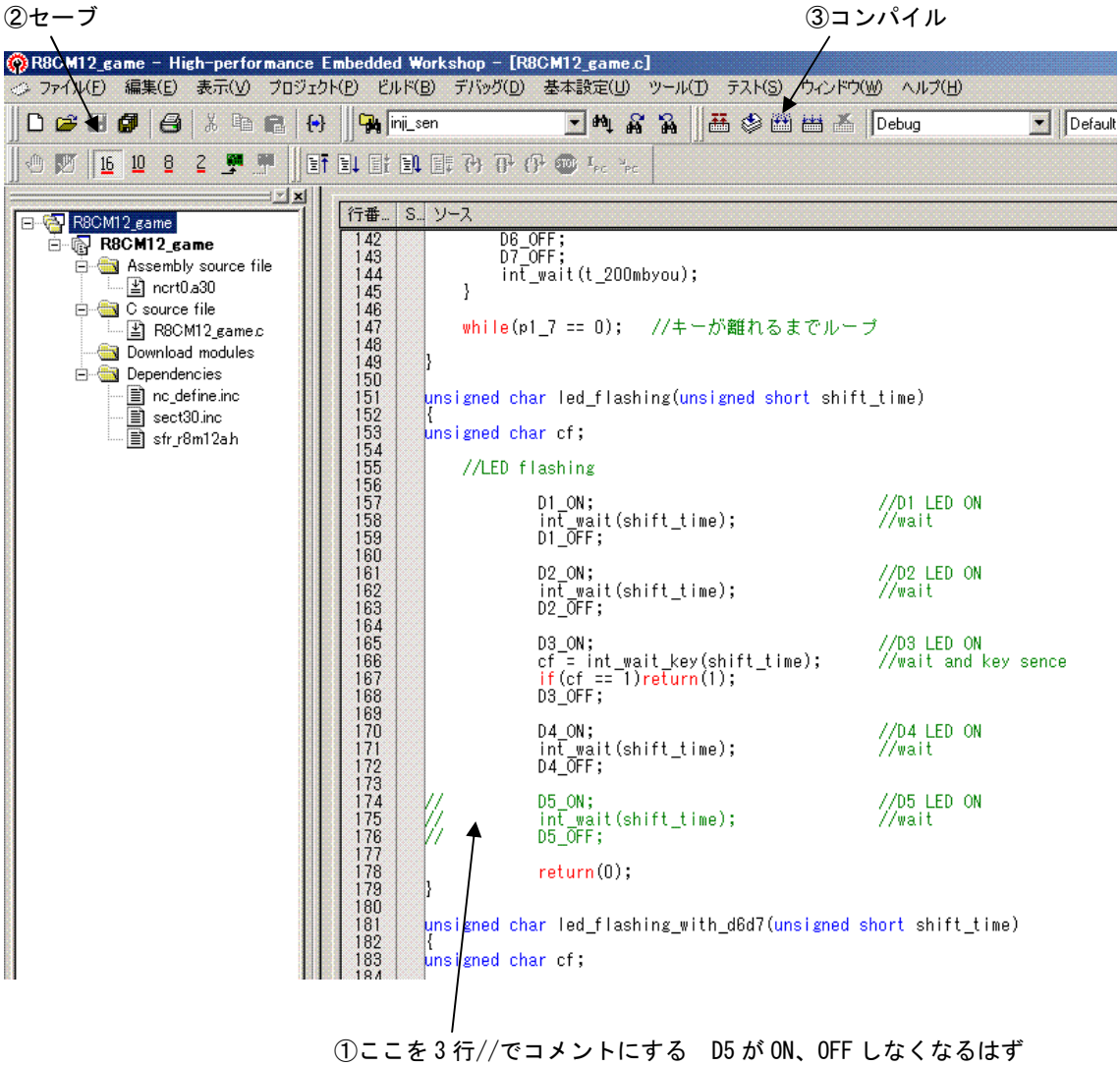
４．ＳＷ３をＲＵＮ側に切り替え

５．ＲＥＳＥＴスイッチＳＷ２を１回押すと書き込まれたプログラムが動作します。

６．「デバイスとの切断」をクリック→　次回書き込みのため一度切断しないとだめなようです。

ｄ：プログラムを変更してみる

①例として 3 行//でコメントにします。コメントは色が緑に変わります。
②セーブします
③コンパイルします
④エラーが無いことを確認
⑤FDT で書き込んで、D5 の点滅が無くなったことを確認します。



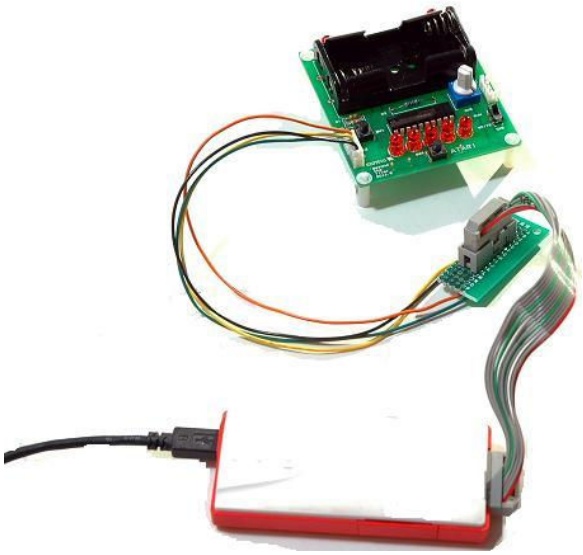
④エラーが無いことを確認
⑤FDT で書き込んで動作を確認

以下省略

３－３．　Ｅ８ａでの動作、デバック

ａ．Ｅ８ａについて

Ｅ８ａ本体



Ｅ８ａとＲ８Ｃ/Ｍ１２＿反射神経ゲーム機の接続例

　Ｅ８ａはＵＳＢ接続で使用する、ルネサスセミコンダクタ社のマイコン書き込み器、ＪＴＡＧデバッカです。書き込み器としてはＳＨマイコンからＴｉｎｙマイコンまで書けます。デバッカとしてはＨ８／Ｔｉｎｙシリーズ、Ｒ８シリーズなどでＣ、アセンブラソースデバックに対応します。ブレークポイント設定やメモリ、Ｉ／Ｏの読み込み、書き込みができます。

さらに、ＵＳＢバスパワーで最大３００ｍＡまでマイコンに電源（３．３Ｖ、5Ｖ）を供給できますので、その範囲のハードウエアであれば特に電源を用意する必要がありません。電源のＯＮ，ＯＦＦはＨＥＷから行います。電源スイッチをオフにするか電池ははずして、Ｅ８ａからの電源でデバックするのが電池の消耗を気にせず良いでしょう。

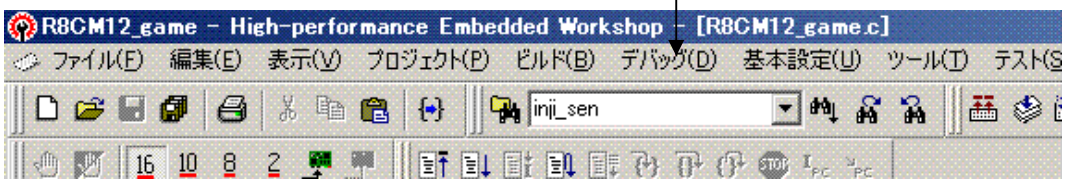
　古いＥ８ａではＲ８Ｃ／Ｍ１２に対応できていない場合がありますが、Ｅ８ａの最新ファームをダウンロードすれば対応できます。

Ｅ８ａを使うためにＨＥＷの設定に加える必要があります。初めてＥ８ａ使う場合、デバイスドライバ等のインストゥールはメーカーの説明書によって事前に行ってください。以下の接続を行います。

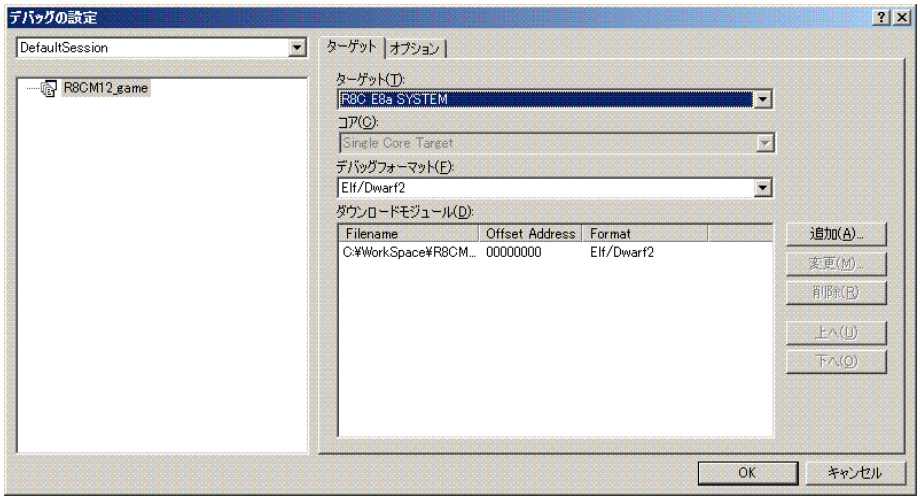
●Ｅ８ａとパソコンはＵＳＢで接続されている

●Ｅ８ａとゲーム器は回路図で示されるように接続されている→このハーネスは弊社では販売しておりませんので、回路図を参考に自作等お願いします。詳細は添付ファイルｒｊｊ10ｊ2020_e8a_s.pdf を参照下さい。

「デバック」をクリックします。



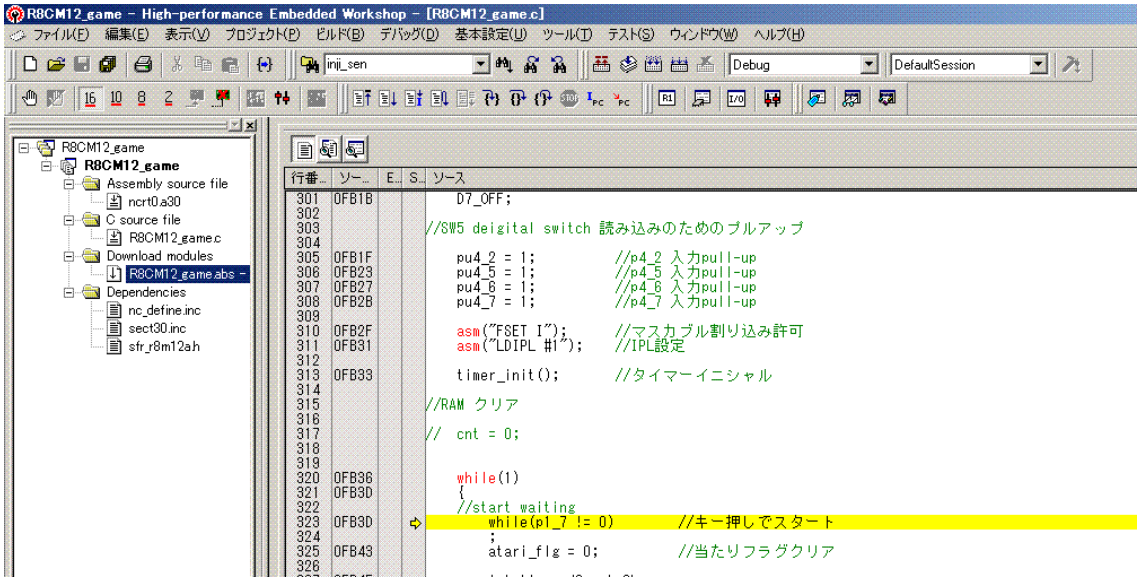
以下のように設定します。



拡張子がａｂｓがデバック情報込みのファイルです。

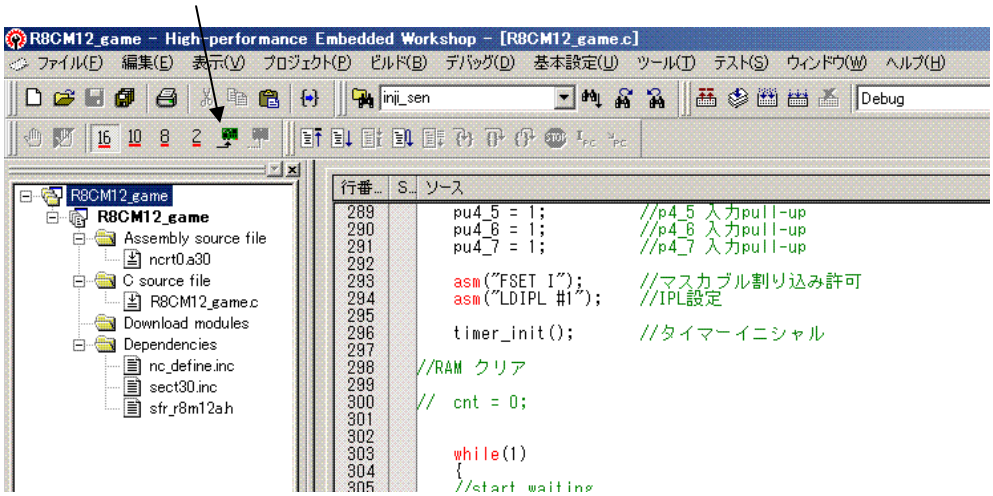
続行すると、以下の表示が示されます。

エミュレータから電源を供給する場合、電源スイッチをＯＦＦにするか、ゲーム機の電池を必ずはずして使用してください。
うまく動作するとＣソースの左にアドレスが表示されます。

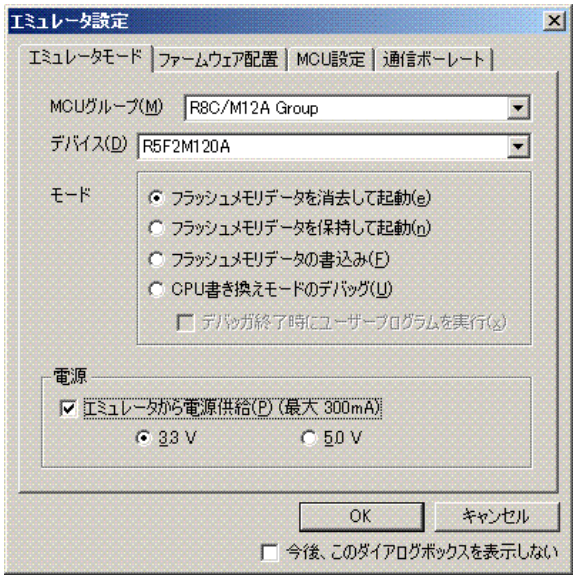


b. デバックの概要

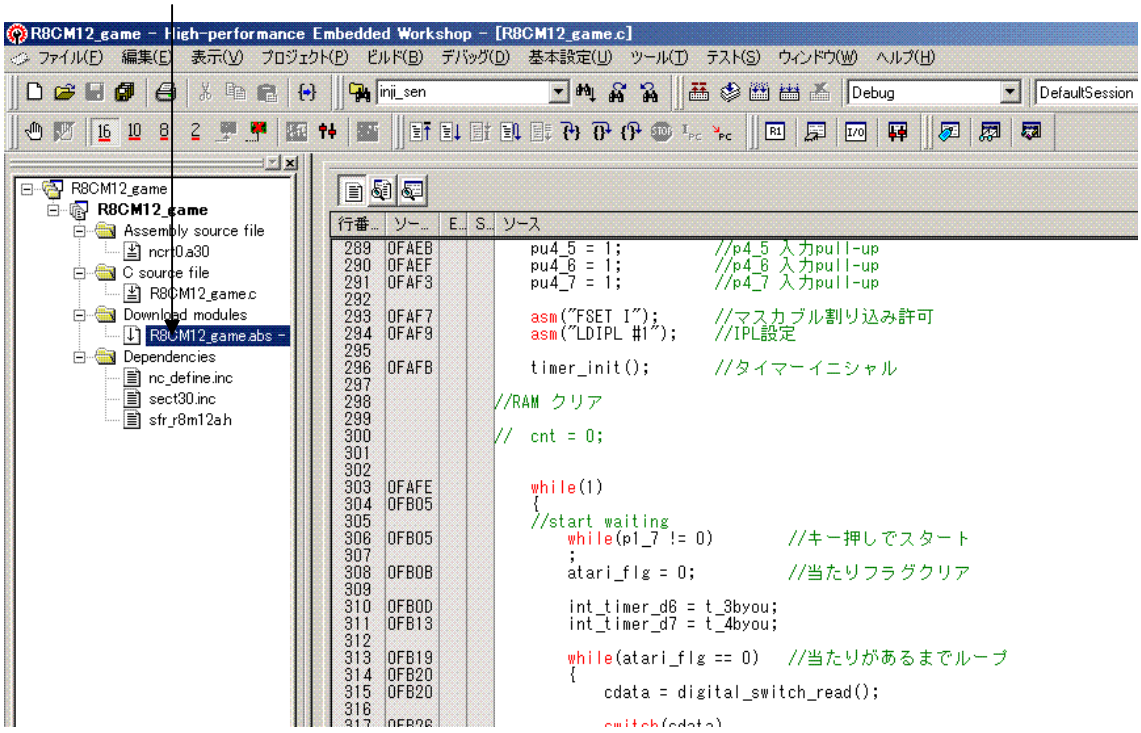
「接続」をクリックします。



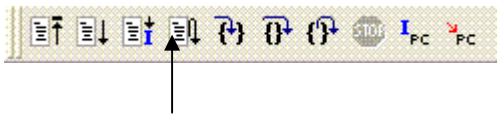
エミューレータの設定が表示されますから、以下のように設定し、最後に、「エミュレータから電源供給」3.3Ｖを選択します。なお、デバック時はモードが下記でＯＫですが、出荷時は「フラッシュメモリデータの書き込み」にしないと、単独で動作しませんのでご注意下さい。



ここをダブルクリックしてデバック用のａｂｓファイルをダウンロードします。



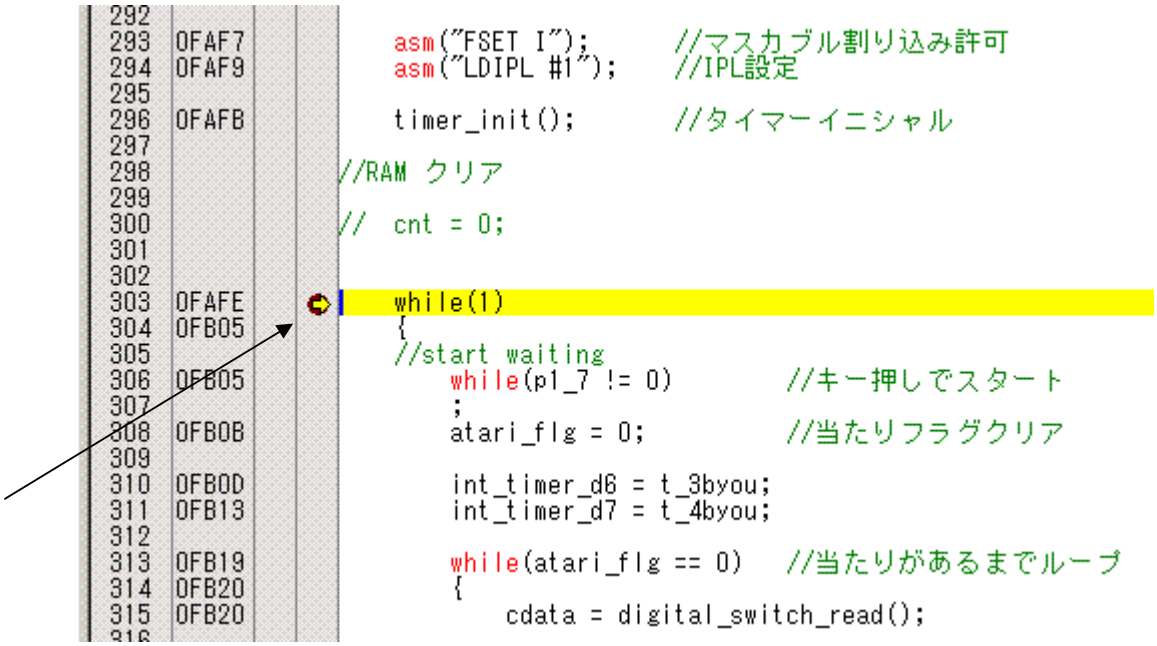
「リセットして実行」をクリックするとプログラムが実行されます。



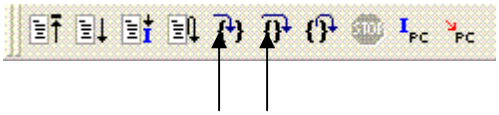
プログラムのリアルタイム動作中は以下のような表示になります。

赤丸は「ＳＴＯＰ」です。ブレークポイントまで動作、ステップ動作を行う場合、「ＳＴＯＰ」をクリックして停止状態にしてから行います。

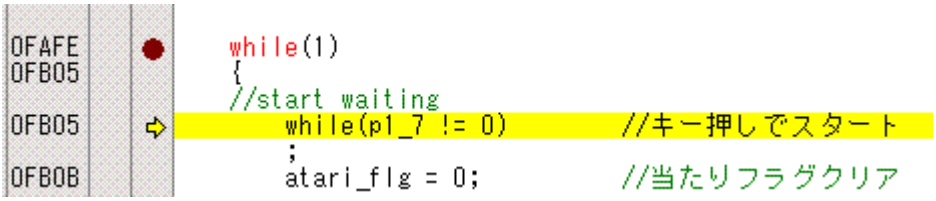
ブレークポイントを設定したい場合、



矢印の部分をマウスでダブルクリックすると茶色い丸が出て設定できます。解除もそこをダブルクリックします。
「リセットして実行」をクリックし、それ以降、１行ずつステップ動作をさせるためにはここをクリックします。

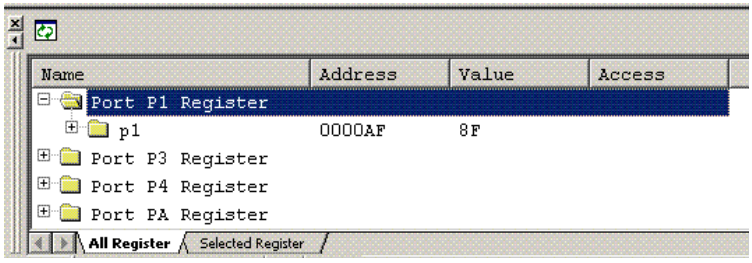


　左がステップイン、右がステップオーバーですが、違いは途中に関数があった場合、関数の中まではいるのがステップイン、入らず次行に行くのがステップオーバーです。用途により使い分けます。



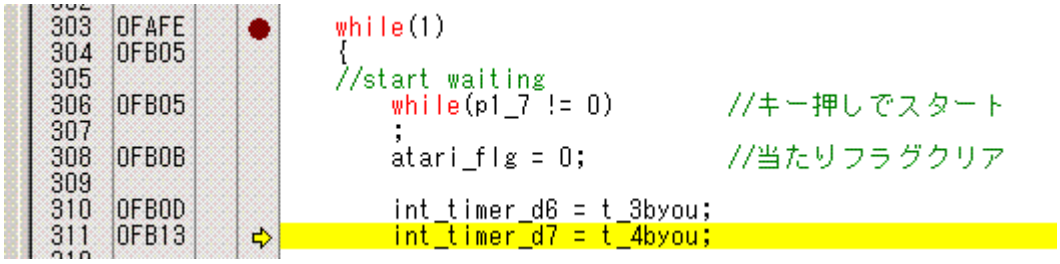
１行実行しました。

このときＰ１（ポート１）の内容はＩ／Ｏの表示ウインドウで確認できます。



ｐ１＿７は’1’ですので、{↓}をクリックしてもこの行から動きません。ＳＷ４スイッチを押しながら{↓}をクリックすると、下の行に移動することが確認できます。

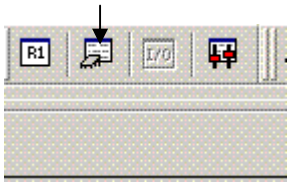
ここまでステップして、int_timer_d6 の中身を見てみます。



ｔ_3byou は頭の方で 300 と設定しています。

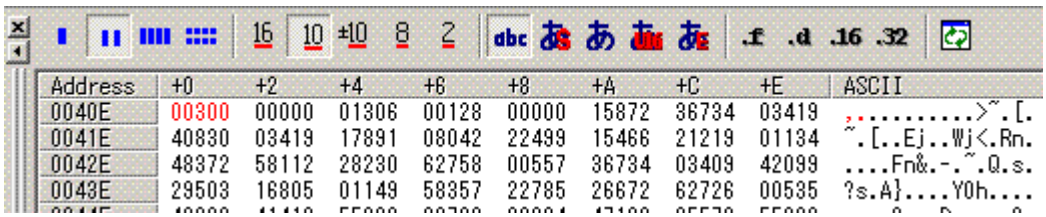
```
#define t_3byou 300
```

メモリをクリック。

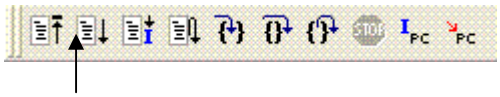


int_timer_d6 を選択。

以下は 10 進数、2byte で表示。確かに 300 が入っています。



この地点からブレークポイントを新たに設定して、そこまで動作させるのは、「実行」をクリックします。



以上がデバックの概要です。ＨＥＷでこのように簡単にＣソースデバックができます。もちろん、アセンブラでもできます。

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　