

T-SH2A

PC/104バスシリーズCPUボード

開発環境HEWのインストールとサンプルプログラムの説明

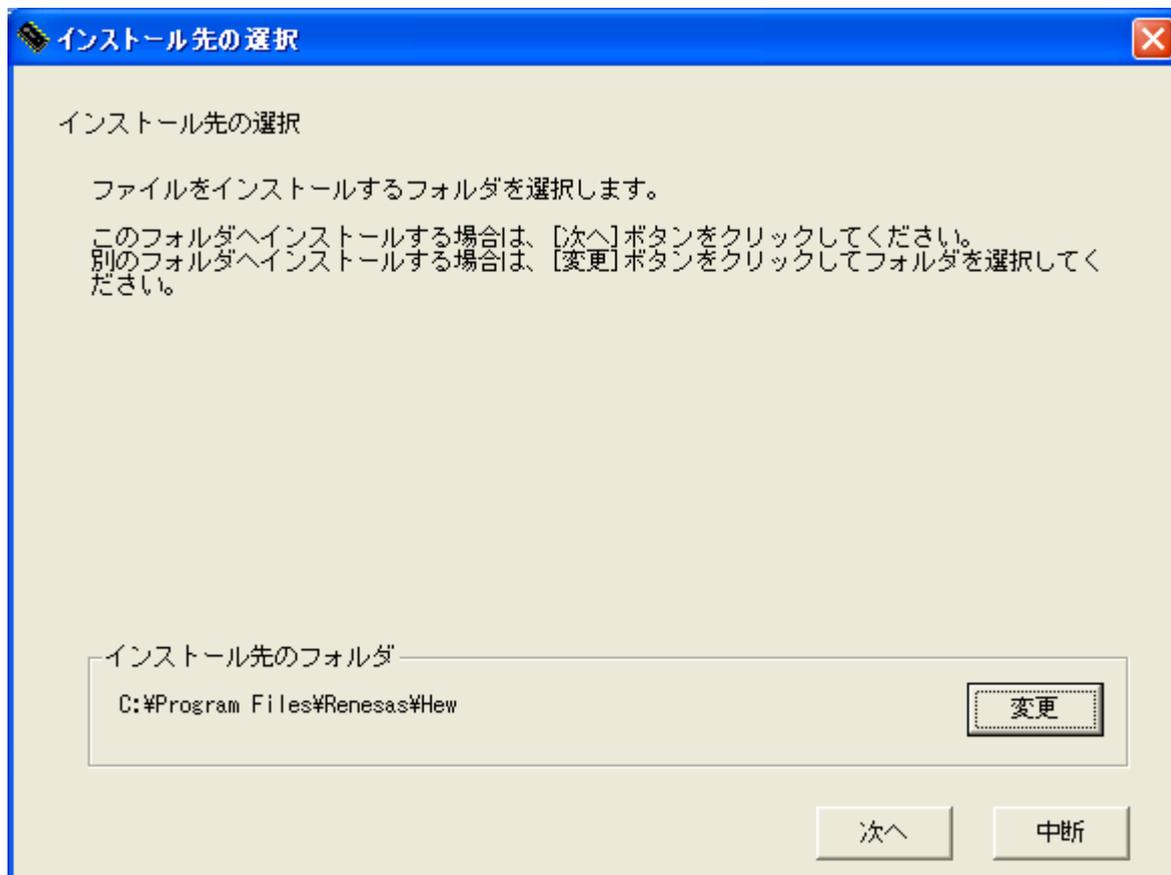


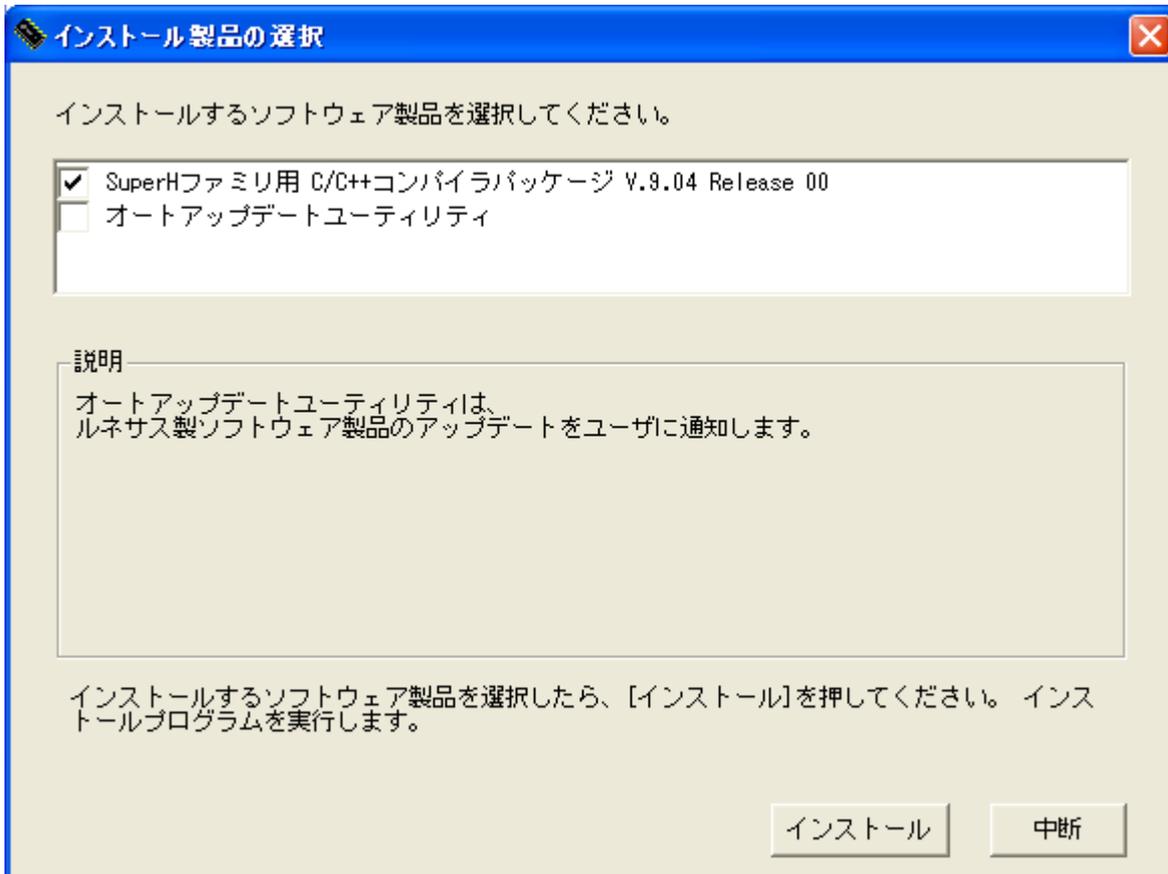
目次

1. 開発環境HEWのインストール
2. シリアル接続 HEW モニタのセットアップ
3. 仮想シリアル Windows 用 USB ドライバのセットアップ
4. シリアル接続 HEW モニタのHEW上での設定
5. I/Oファイルのロード
6. サンプルプログラムの説明

1. 開発環境 HEW のインストール

ルネサスエレクトロニクスのホームページから評価版ソフトウェアツール V.9.04Release00 をダウンロードします。





2. シリアル接続 HEW モニタのセットアップ

CQ 出版社 Cqconnect より下記をダウンロードする。

シリアル接続 HEW モニタ セットアップ説明ファイル (2010年4月24日版)

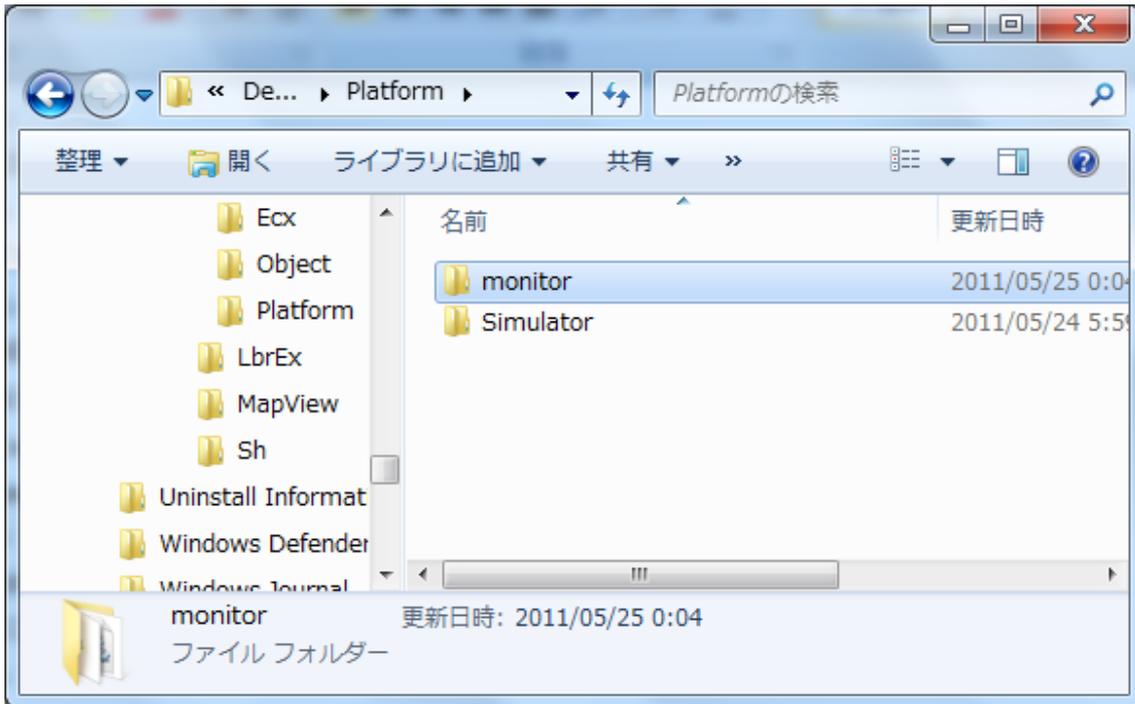
HEWmonitor.zip を展開すると

dll

image

readme.htm

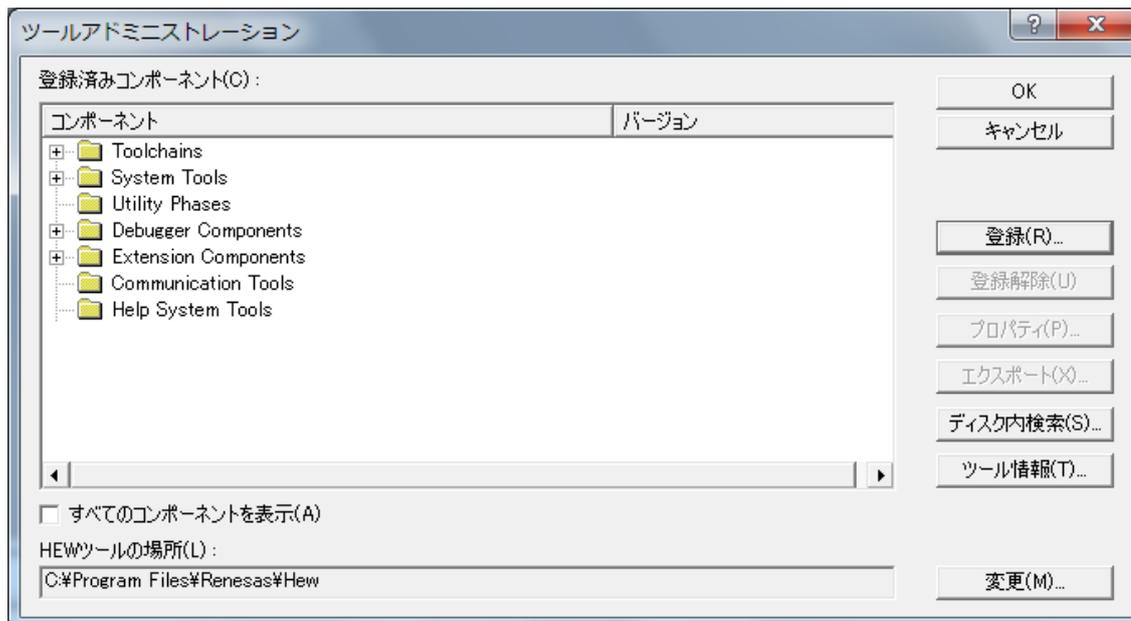
dll¥monitor をHEWをインストールしたフォルダー以下の Tools¥Renesas¥DebugComp¥Platform にコピーします。



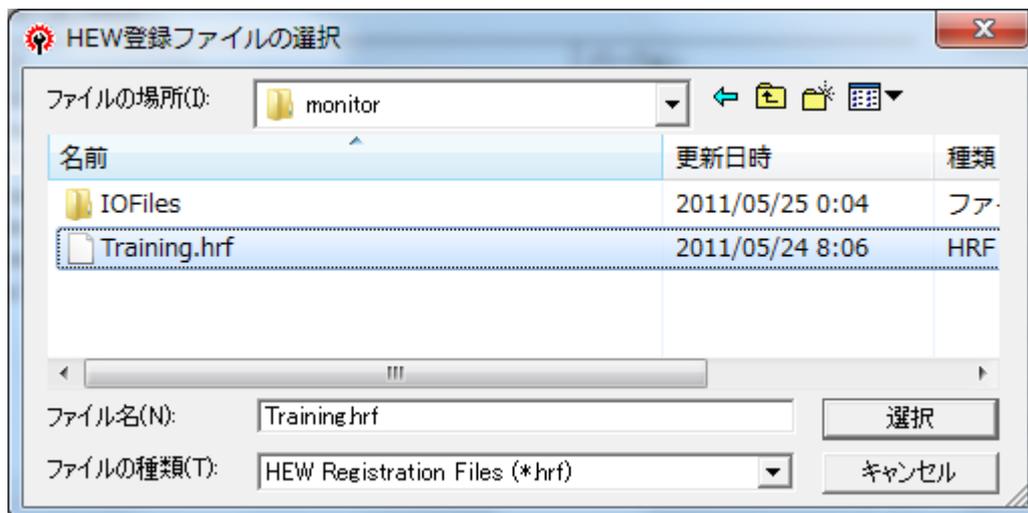
次にHEWを起動して“ようこそ！”ダイアグロの“アドミニストレーション”を選択します。



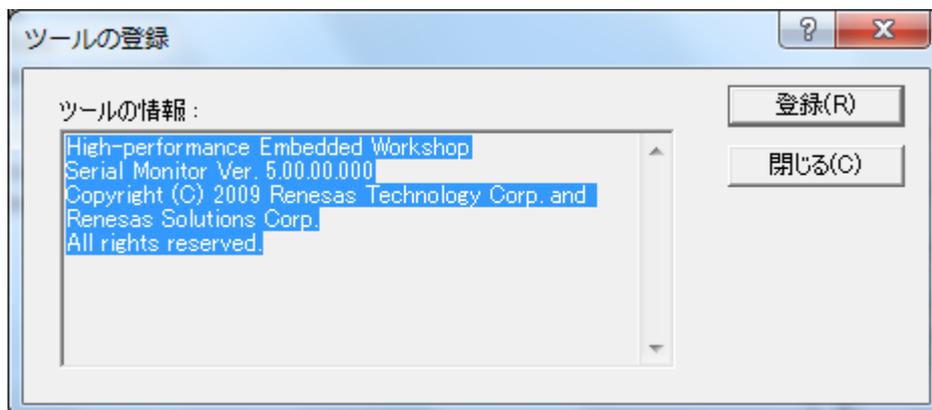
続いて表示される“ツールアドミニストレーション”では“登録”を選択します。



するとファイル選択ダイアログが表示されるのでHEWインストール先にコピーした monitor フォルダー内にある Training.hrf を選択します。



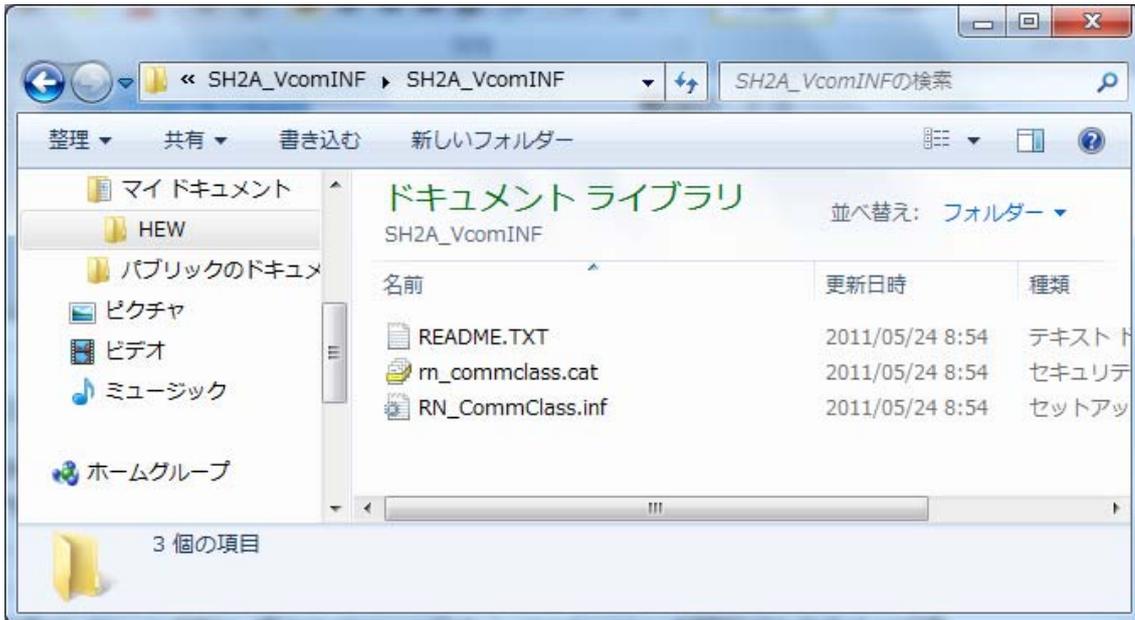
登録完了



3. 仮想シリアル Windows 用 USB ドライバのセットアップ

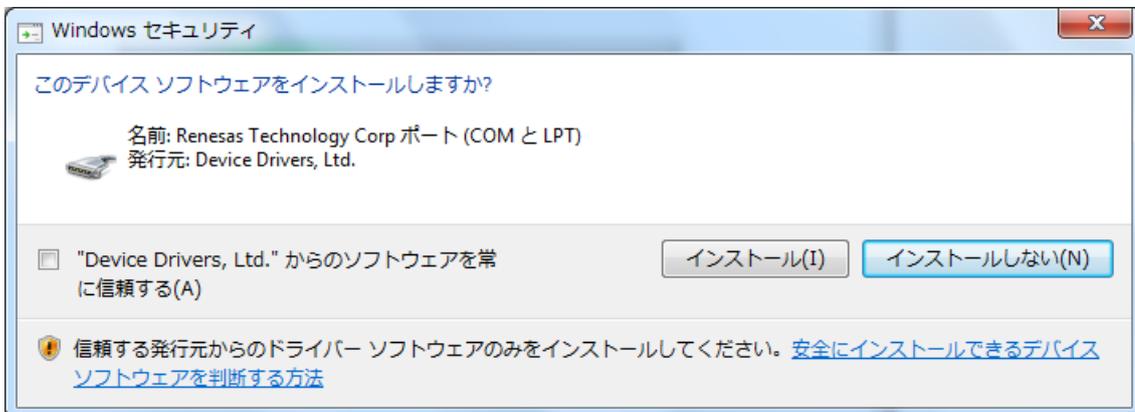
SH2Aマイコン基板対応仮想シリアルWindows 用ドライバ“SH2A_VcomINF.zip

”を CQ 出版社 Cqconnect よりダウンロードしアーカイブを解凍する。

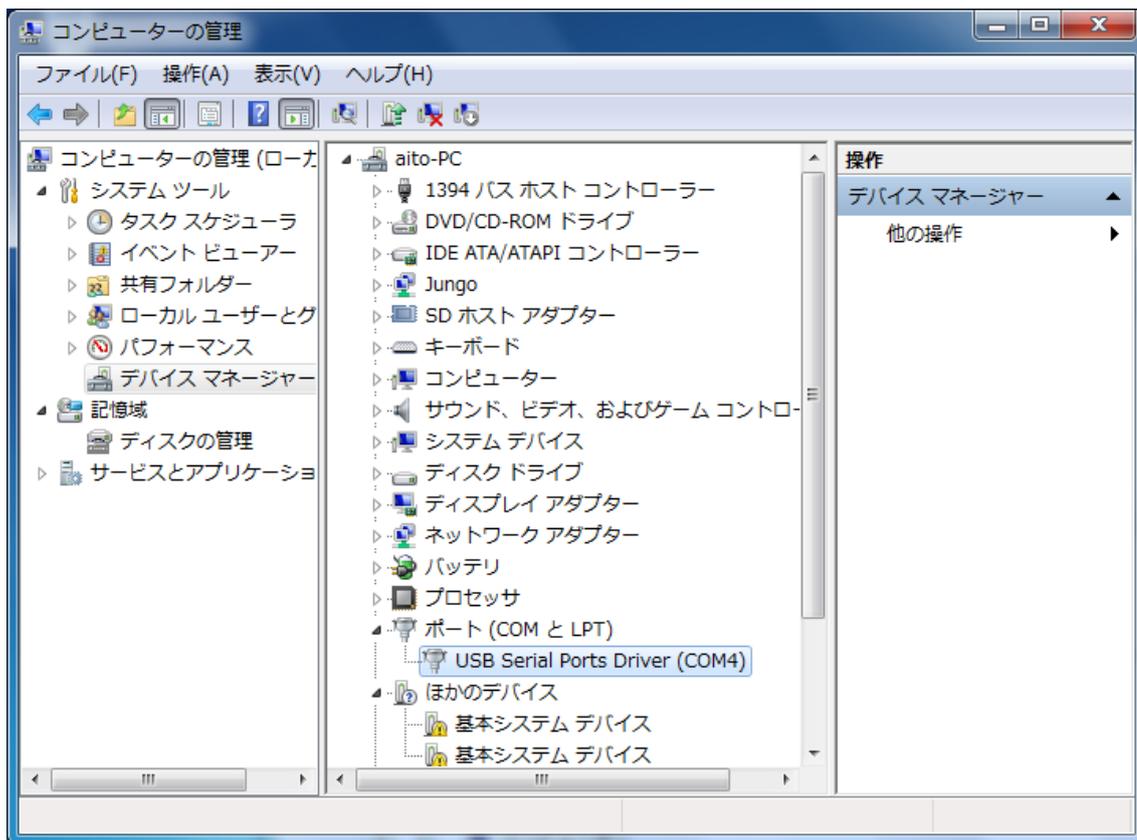


自動インストールできない場合は Windows のデバイスマネージャーを開き“ほかのデバイス”項目で基本システムデバイス??の“ドライバーの更新”でインストールしてください。

ドライバーの更新



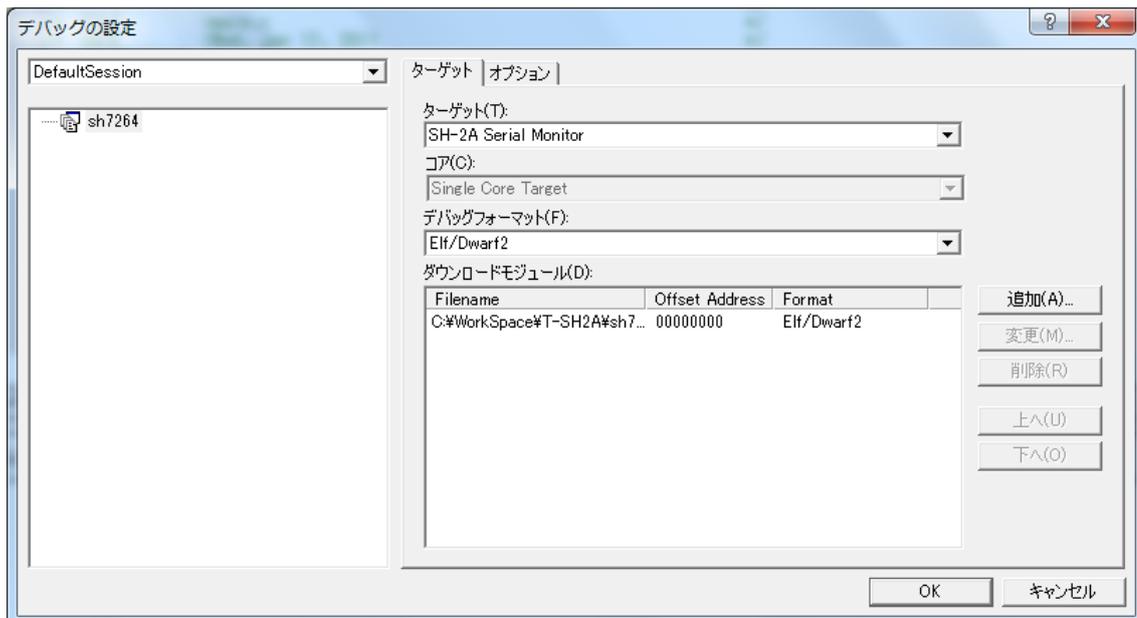
更新完了



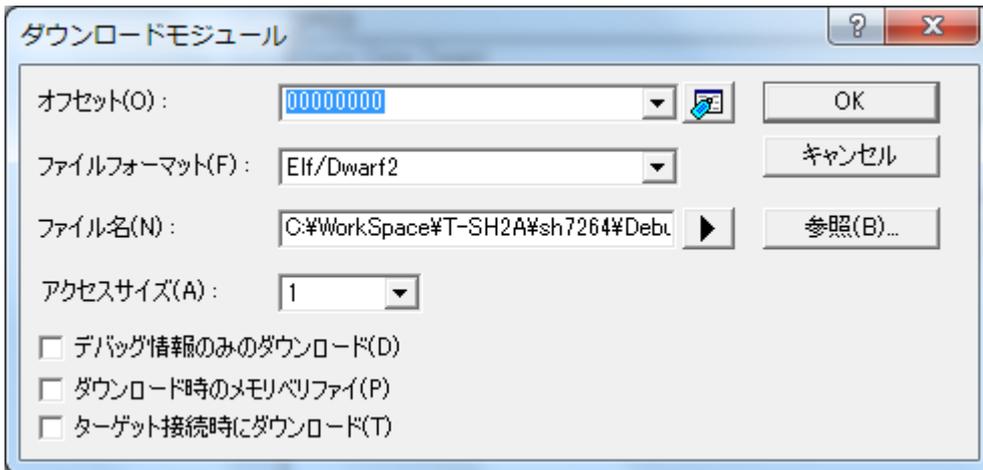
4. シリアル接続 HEW モニタのHEW上での設定

サンプルプログラムでHEWモニタを使用するために、HEWで“WorkSpace¥TSH2A ¥TSH2A.hws”を選択します

HEWのメニューから“デバッグ (D)” → “デバッグの設定 (D)” を選択しダイアログを開きます



ダウンロードモジュールの設定



シリアル接続HEWモニタは割り込み優先レベル15を使用しています、ユーザーの使用を禁止するために自動生成される“resetprg.c”の

```
#define SR_Init 0x000000F0
```

↓

```
#define SR_Init 0x000000E0 に変更します。
```

5. I/Oファイルのロード

シリアル接続のHEWモニタでは以下に示すアイコン、又は表示・メニューの「CPU」にある「I/O」コマンドにより、内蔵周辺機能のレジスタ状態をウィンドウで表示することが可能です。

ただし、初期の接続状態では、I/Oファイルが未ロードのため、I/Oウィンドウは空の状態です。そこでI/Oウィンドウ上で右クリックし、「I/Oファイルのロード」コマンドを実行してください。

そうしたら「参照」コマンドを実行し、表示されたファイル選択のダイアログ上で「IOFiles」フォルダ内にある目的のCPUのI/Oファイル「拡張子：Sh72620.io」を選択してください。



7. サンプルプログラムの説明

boot.c

cmt_func.c

intprg.c

led_func.c

scif.c

w5300.c

main.c

boot.c

void boot(void)

CPU 初期化を行っています、I/Oピンの指定をし、DRAMアクセスの設定をしています。

/* レジスタ・バンク使用許可 */

```
INTC.IBNR.WORD &= ~0xC000;
```

```
INTC.IBNR.WORD |= 0x4000;
```

をしていますので利用する割り込みで vect.h で(resbank)の記述を追加します。

cmt_func.c

void io_init_cmt0(void)

CMT0の初期化で、1 m s の割り込みに設定しています。

void int_cmt_cmi0(void)

1 m s ごとに呼び出される関数で500 m s ごとに緑LEDをON/OFFしています。

scif.c

void scif2_init (void)

COMポートの設定を、57600bps,ビット長8ビット、パリティなし、ストップビット1です。

int scif2_get(BYTE *d)

受信した値があれば受信バッファから取り出します。

void scif2_put (BYTE d)

送信する値を送信バッファに書き込みます。

void int_scif_scif2_rxi2 (void)

割り込み受信した値を受信バッファに書き込みます

void int_scif_scif2_txi2 (void)

割り込みで送信バッファに値があれば送信します。

intprg.c

タイマー割り込み記述

```
// 175 CMT CH0 CMI0
```

```
void INT_CMT_CMI0(void){/* sleep0; */
```

```
    extern void int_cmt_cmi0(void);
```

```
    int_cmt_cmi0();
```

```
}
```

SCIF2の受信割り込みの記述

```
// 242 SCIF SCIF2 RXI2
void INT_SCIF_SCIF2_RXI2(void){/* sleep0; */
    extern void int_scif_scif2_rxi2 (void);
    int_scif_scif2_rxi2();
}
```

SCIF2の送信割り込みの記述

```
// 243 SCIF SCIF2 TXI2
void INT_SCIF_SCIF2_TXI2(void){/* sleep0; */
    extern void int_scif_scif2_txi2 (void);
    int_scif_scif2_txi2();
}
```

led_func.c

```
void led_init(void)
```

LED出力ポートの初期化関数です。

```
void ?led_on()
```

```
void ?led_off()
```

ポートをON/OFFする関数です。

main.c

```
i = scif2_get(&cd);
```

```
if(i > 0)
```

```
{    scif2_put(cd); }
```

SCIF2でエコーバックを行います(送受信割り込み使用)、SCIF2から受信した値をそのままSCIF2へ送信します。

ネットワークの設定は下記です。

```
uint8 mip[4] = {192,168,1,32};          /* 自局 */
uint8 gip[4] = {192,168,1,1};          /* ゲートウェイ */
uint8 sip[4] = {255,255,255,0};        /* サブネット */
uint8 tip[4] = {192,168,1,100};        /* 相局 */
uint8 serverip[4] = {192,168,1,30};     // "TCP SERVER" IP address for loopback_tcp()
int    lpt = 9600;                      /* ローカルポート */
int    rpt = 1030;                      /* リモートポート */
```

```
loopback_tcp(0,50000,data_buf,0);      // TCP SERVER
```

TCP/IPのエコーバックでポート50000で接続します。

```
loopback_udp(7,9600,data_buf,0);      // UDP loopback
```

UDP/IPのエコーバックでポート9600です。