T - S H 2 A

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ジから評価版ソフトウェアツール V.9.04 Release 00 をダウンロードします。

| 💊 High-performance Embedded | Workshopインストールマネージャ | × |
|--|--|---|
| RENESAS | はじめにお読み下さい 標準インストール(推奨) | |
| Multi installation | マルティンス マンテナンス | |
| Active High-performance Embedded Workshop Switch over! | - 説明 High-performance Embedded Workshopインストールマネ ージャは、統合開発環境High-performance Embedded Workshopのインストールをガイドするユーティリティで す。このユーティリティを使用することで、ひとつのPC Iこ複数のHigh-performance Embedded Workshopをインス トールすることが可能になります。 | |
| Non-active | 終了 | |

| ♦ インストール先の選択 | × |
|---|---|
| インストール先の選択 ファイルをインストールするフォルダを選択します。 このフォルダヘインストールする場合は、[次へ]ボタンをクリックしてください。 別のフォルダヘインストールする場合は、[変更]ボタンをクリックしてフォルダを選択してく ださい。 | |
| インストール先のフォルダ C:¥Program Files¥Renesas¥Hew 変更 次へ 中断 | |

| ≫ インストール製品の選択 | × |
|---|---|
| インストールするソフトウェア製品を選択してください。 | |
| ✔ SuperHファミリ用 C/C++コンパイラバッケージ V.9.04 Release 00 ┃ オートアップデートユーティリティ | |
| _ EXAB | |
| オートアップデートユーティリティは、 ルネサス製ソフトウェア製品のアップデートをユーザに通知します。 | |
| | |
| インストールするソフトウェア製品を選択したら、[インストール]を押してください。 インス トールプログラムを実行します。 | |
| インストール 中断 | |

| C/C++ compiler package for SuperH RISC engine family V.9.04 Release 00 | | |
|--|--|--|
| | | |
| | InstallShield Wizard の完了 | |
| | セットアッフペは、コンピュータへC/C++ compiler package for SuperH RISC engine family のインストールを終了し ました。 | |
| RENESAS | | |
| | | |
| High-performance Embedded Workshop ⁴ | | |
| | | |
| InstallShield | < 戻る(B) 完了 キャンセル キャンセル | |

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。

| ようこそ! | | ? × |
|-------|------------------------------|----------------|
| 2 | ● 新規プロジェクトワークスペースの作成(C) | <u> </u> |
| 6 | C 最近使用したプロジェクトワークスペースを開く(0): | アドミニストレーション(A) |
| | ○ 別のプロジェクトワークスペースを参照する(B) | |

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。

| ツールアドミニストレーション | | ? × |
|---|-------|------------|
| 登録済みコンボーネント(C): | | ок |
| コンボーネント | バージョン | キャンセル |
| Toolchains | | |
| Utility Phases | | |
| Debugger Components Extension Components | | 登錄(R) |
| Communication Tools | | 登錄解除(U) |
| Help System Tools | | プロパティ(P) |
| | | エクスポート(ン) |
| | | ディスク内検索(S) |
| • | | ツール情報(T) |
| 「」すべてのコンボーネントを表示(A) | | |
| HEWツールの場所(L): | | |
| C¥Program Files¥Renesas¥Hew | | 変更(M) |

するとファイル選択ダイアグロが表示されるので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。

| 🖗 HEW登録ファイ | (ルの選択 | | × |
|--------------|-------------------------------|-----------------|-----|
| ファイルの場所(I): | 🎉 monitor | | |
| 名前 | * | 更新日時 | 種類 |
|]] IOFiles | | 2011/05/25 0:04 | ファ・ |
| Training.hrf | | 2011/05/24 8:06 | HRF |
| | | | |
| • | III | | Þ |
| ファイル名(N): | Traininghrf | 译 | 訳 |
| ファイルの種類(T): | HEW Registration Files (*hrf) | ▼ +n | ven |

登録完了

| ツールの登録 | | 2 × |
|--|---|-----------------|
| ツールの情報: High-performance Embedded Workshop Serial Monitor Ver. 5.00.00.000 Copyright (C) 2009 Renesas Technology Corp. and Renesas Solutions Corp. All rights reserved. | * | 登録(R) 閉じる(C) |

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 Windowsn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

| → Windows セキュリティ | |
|---|-----------------------------------|
| このデバイス ソフトウェアをインストールしますか? | |
| 名前: Renesas Technology Corp ポート (COM と LPT) | |
| "Device Drivers, Ltd." からのソフトウェアを常 に信頼する(A) | インストール(I) インストールしない(N) |
| ・ 信頼する発行元からのドライバー ソフトウェアのみをインストー ソフトウェアを判断する方法 | ・ルしてください。 <u>安全にインストールできるデバイス</u> |

```
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のに、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)"を選択しダイアグロを開きます

| デバッグの設定 | | ? × |
|----------------|---|---|
| DefaultSession | ターゲット オプション | |
| 🕞 sh7264 | ターゲット(T): SH-2A Serial Monitor コア(C): Single Core Target デバッグフォーマット(F): EH/Dwarf2 マット(F): | I |
| | Filename Offset Address Format C:¥WorkSpace¥T-SH2A¥sh7 00000000 Elf/Dwarf2 | <u>追加(A)</u> 変更(M) 肖J除(R) 上へ(U) 下へ(O) |
| | ОК | ++>UL |

ダウンロードモジュールの設定

| ダウンロードモジュー | IL | ? X |
|-----------------------|---------------------------------|-------|
| オフセット(0): | 00000000 | ОК |
| ファイルフォーマット(F): | Elf/Dwarf2 | キャンセル |
| ファイル名(N): | C:¥WorkSpace¥T-SH2A¥sh7264¥Debt | 参照(B) |
| アクセスサイズ(A): | 1 • | |
| 🔲 デバッグ情報のみの | ダウンロード(D) | |
| □ ダウンロード時のメモリベリファイ(P) | | |
| 🔲 ターゲット接続時にダ | ウンロード(T) | |

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 0x00000E0 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立ち上がります。 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。

| I/0ファイルを指定 | ? X |
|---|----------------|
| I/Oファイル(I): bugComp¥Platform¥monitor¥IOFiles¥Sh72620.io ▶ 参照… ▼ IOファイルのパス名を保存する(S) | OK キャンセル(C) |

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、1msの割り込みに設定しています。 void int_cmt_cmi0(void) 1msことに呼び出される関数で500ms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ーから取り出します。 void scif2_put (BYTE d) 送信する値を送信バッファーに書き込みます。 void int_scif_scif2_rxi2 (void) 割り込み受信した値を受信バッファーに書き込みます

void int_scif_scif2_txi2 (void)

割り込みで送信バッファー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){/* sleep(); */
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){/* sleep(); */

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){/* sleep(); */
```

}

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

```
loopback_tcps(0,50000,data_buf,0); // TCP SERVER
T C P / I P のエコーバックでポート 5 0 0 0 0 で接続します。
loopback_udp(7,9600,data_buf,0); // UDP loopback
UD P / I P のエコーバックでポート 9 6 0 0 です。
```