

## AP-SH3D-2A(SH7727 CPU BOARD)

## サンプルプログラム解説

第1版 2008年4月15日

## 目次

|  |    |
|--|----|
| 1. 概要  | 1  |
| 1.1 概要   | 1  |
| 1.2 動作モード                                      | 1  |
| 1.3 開発環境について                                   | 4  |
| 2. サンプルプログラムの構成                                | 5  |
| 2.1 フォルダ構成                                     | 5  |
| 2.2 ファイル構成                                     | 6  |
| 2.2.1 ビッグエンディアン設定時 (AP-SH3D-2A_Big) のファイル構成    | 6  |
| 2.2.2 リトルエンディアン設定時 (AP-SH3D-2A_Little) のファイル構成 | 9  |
| 3. ビルド・デバッグ方法                                  | 11 |
| 3.1 ビッグエンディアン設定時                               | 11 |
| 3.1.1 Renesas 社製 SHC 用サンプルプログラム (Ver6 以降)      | 11 |
| 3.2 リトルエンディアン設定時                               | 13 |
| 4. 動作説明  | 15 |
| 4.1 SHC 用サンプルプログラムの動作説明                        | 15 |
| 4.2 GCC 用サンプルプログラムの動作説明                        | 16 |
| 4.3 RAM 動作時のメモリマップ                             | 17 |
| 4.4 ROM 動作時のメモリマップ                             | 18 |
| 4.5 USB ホスト動作                                  | 19 |
| 4.5.1 USB メモリ接続時の動作内容                          | 19 |
| 4.5.2 USB マウス接続時の動作内容                          | 20 |
| 4.6 USB ファンクション動作                              | 21 |
| 4.6.1 USB 仮想シリアルインストール方法 (WindowsXP の場合)       | 21 |
| 4.6.2 USB 仮想シリアルインストール方法 (Windows2000 の場合)     | 24 |
| 4.6.3 USB 仮想シリアルインストール方法 (WindowsVista の場合)    | 27 |
| 4.6.4 USB シリアルの動作内容                            | 30 |

## 1. 概要

### 1. 1 概要

本サンプルプログラムは、AP-SH3D-2A の動作を確認する簡単なプログラムです。また、弊社製 JTAG デバッガ XrossFinder を使用するためのサンプルとしてご利用いただけます。本サンプルプログラムの概要を以下に示します。

<収録サンプルプログラム>

#### 1. Renesas 社製 SHC 用サンプルプログラム

Renesas 社製 SHC 用サンプルプログラムは、起動処理、バス設定等のほか、周辺機能の中で使用頻度の高いと思われるリアル通信、タイマ割り込み処理を行います。それらに加え、USB ホスト・ファンクション処理や DSP を使用した処理を行います。

#### 2. KPIT 社製 GCC 用サンプルプログラム

KPIT 社製 GCC 用サンプルプログラムは、上記 Renesas 社製 SHC 用サンプルプログラムの処理から、DSP を使用した処理を除いたプログラムです。

### 1. 2 動作モード

本サンプルプログラムは、AP-SH3D-2A で動作します。CPU 動作モード、各メモリ設定は下記のようになっています。

モードの設定方法等につきましては、「AP-SH3D-2A ハードウェアマニュアル」をご覧ください。

なお、下記以外の条件で動作させる場合には、ソースファイルやコンパイラオプションなどを変更する必要があります。

|                     |   |   |
|---------------------|---|---|
| クロックモード             | : | MODE1 (システムクロック=133.333MHz、バスクロック=66.666MHz、<br>周辺クロック=33.333MHz) |
| エリア 0 空間バス幅         | : | 16bit   |
| エンディアン              | : | 使用するサンプルプログラムに合わせて下さい   |
| FLASHROM 設定         | : | FLASHROM を使用する  |
| FLASHROM ライトプロテクト設定 | : | ライトプロテクト解除  |
| SDRAM 設定            | : | SDRAM を使用する   |
| 動作モード               | : | XrossFinder 使用時は ON に、未使用時は OFF にして下さい                            |

CPU ボードの設定を製品出荷時の状態とし、以下の各スイッチの設定を行って下さい。

|  |   |
|--|---|
|  | <p>クロックモード : MODE1<br/>                 エリア 0 空間バス幅 : 16bit<br/>                 エンディアン : ビッグエンディアン</p>                         |
|  | <p>FLASHROM 設定 : FLASHROM 使用<br/>                 FLASHROM ライトプロテクト設定 : ライトプロテクト解除<br/>                 SDRAM 設定 : SDRAM 使用</p> |
|  | <p>動作モード : XrossFinder 使用時</p>  |
|  | <p>動作モード : XrossFinder 未使用時</p>   |

Fig1.2-1 ビッグエンディアン時の動作モード設定

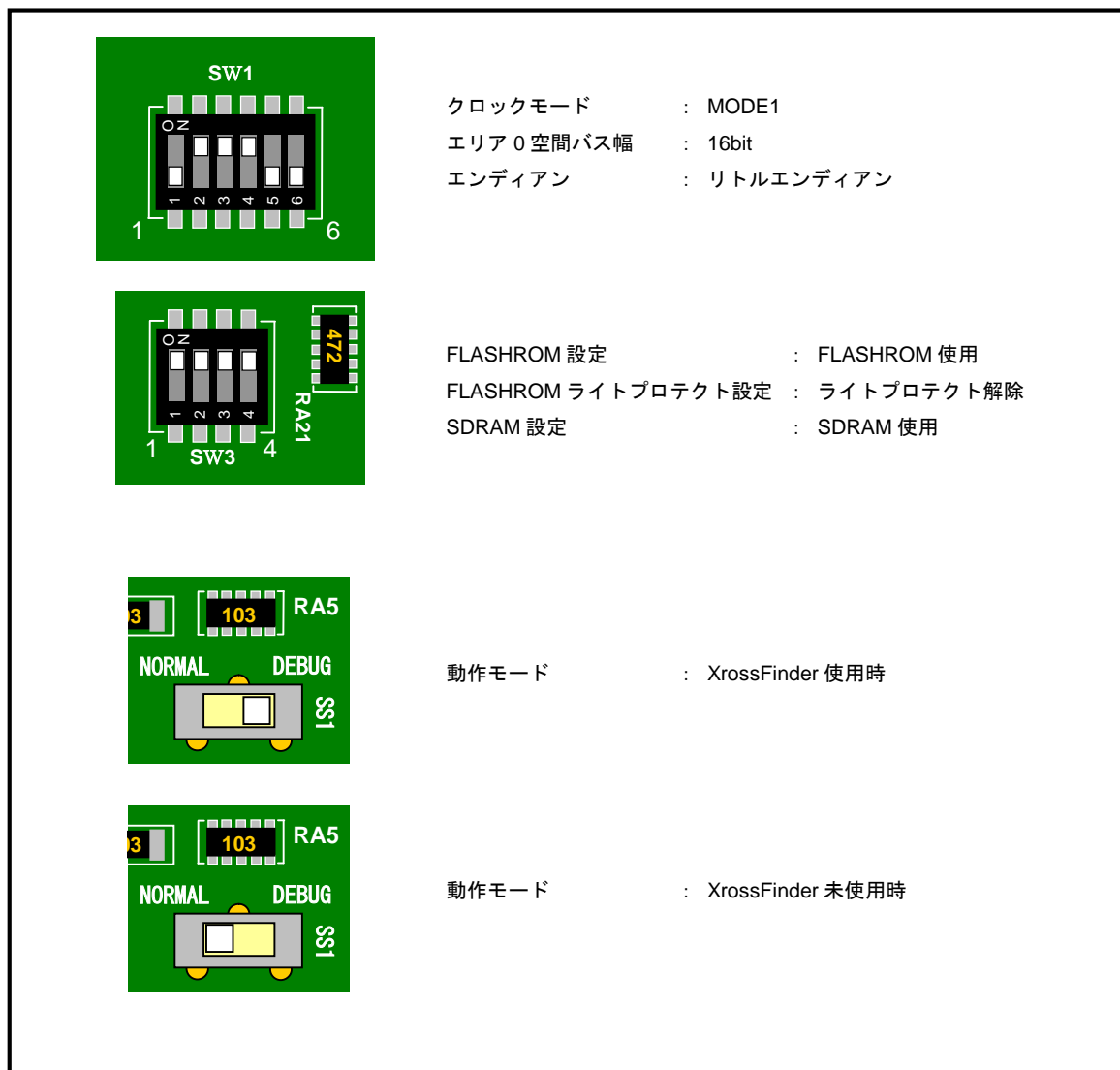


Fig1.2-2 リトルエンディアン時の動作モード設定

### 1. 3 開発環境について

本サンプルプログラムは総合開発環境 HEW を用いて開発されており、使用しているコンパイラにより、「shc」、「gcc」とフォルダ別に収録しています。各フォルダに収録されているサンプルプログラムに対応する開発環境、コンパイラのバージョンは次のようになります。

| フォルダ | 開発環境 | バージョン                   | コンパイラ名               | バージョン        | 備考                            |
|------|------|-------------------------|----------------------|--------------|-------------------------------|
| shc  | HEW  | V 1.2 (release 9)<br>以降 | *1SHC<br>(ルネサステクノロジ) | V6.0.0<br>以降 | SuperHファミリ用C/C++コンパイラパッケージに付属 |
| gcc  | HEW  | V4.02.00.022<br>以降      | **2GNUSH<br>(KPIT)   | V0701<br>以降  | ELF形式、HEW対応                   |

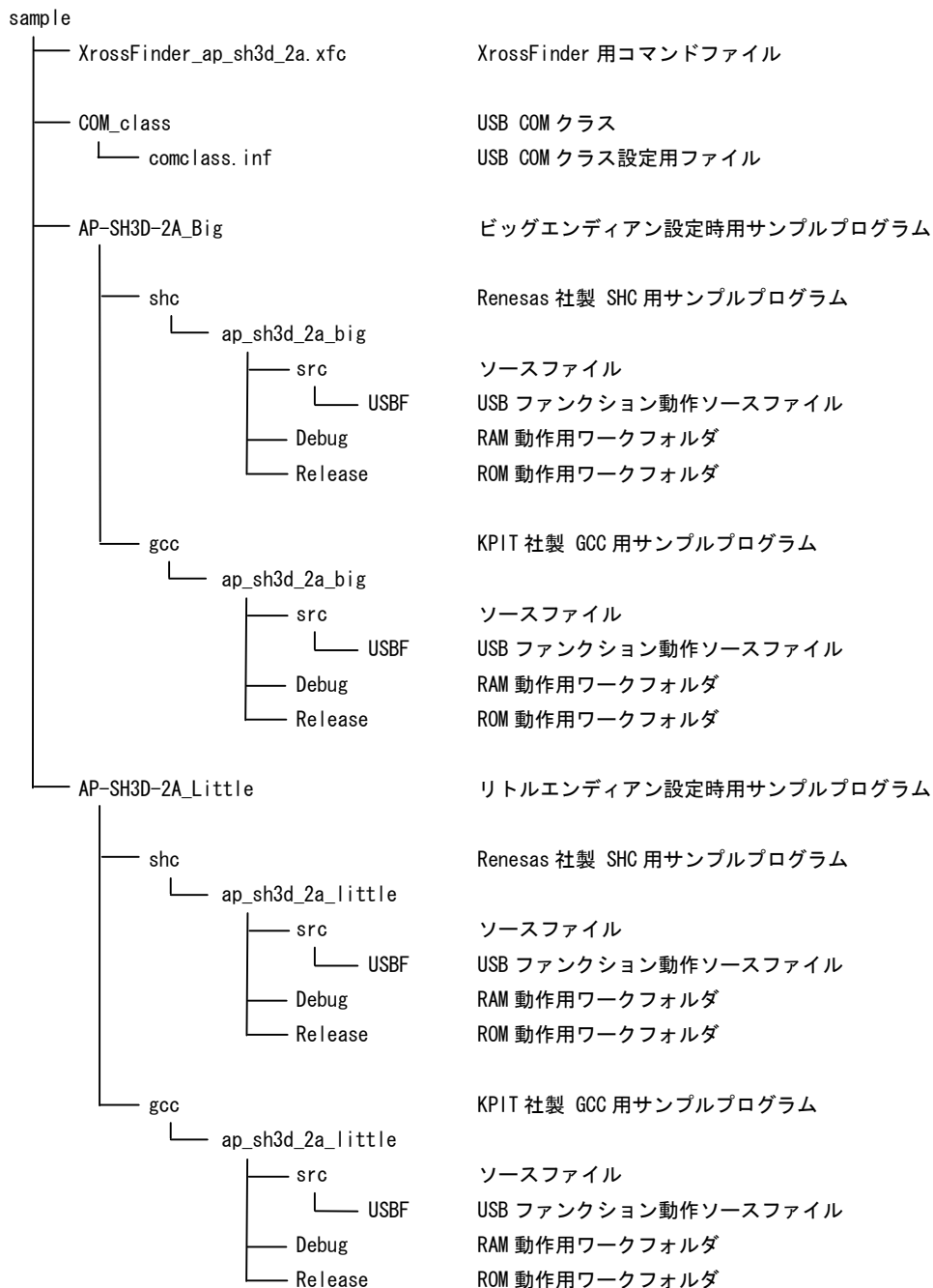
※1：ルネサステクノロジ社製「SuperHファミリ用C/C++コンパイラパッケージ」です。ルネサステクノロジ社のウェブサイトより評価版をダウンロードできます。

※2：KPIT社製「GCC」です。GNUSH及びHEWはKPIT社のウェブサイトよりダウンロードできます。詳しくは弊社ウェブサイトの技術コラムを参照して下さい。

## 2. サンプルプログラムの構成

### 2. 1 フォルダ構成

サンプルプログラムは下記のようなフォルダ構成になっています。



## 2. 2 ファイル構成

### 2. 2. 1 ビッグエンディアン設定時 (AP-SH3D-2A\_Big) のファイル構成

#### (1) SHC 用サンプルプログラム

SHC 用サンプルプログラムは以下のファイルで構成されています。

##### <shc¥フォルダ内>

ap\_sh3d\_2a\_big.hws … HEW 用ワークスペースファイル

##### <shc¥ap\_sh3d\_2a\_big¥フォルダ内>

ap\_sh3d\_2a\_big.hwp … HEW 用プロジェクトファイル

##### <shc¥ap\_sh3d\_2a\_big¥src フォルダ内>

|             |   |                        |
|-------------|---|------------------------|
| main.c      | … | メイン処理                  |
| tmr.c       | … | タイマ処理                  |
| sci.c       | … | シリアル処理                 |
| boot.c      | … | CPU 初期化処理              |
| common.h    | … | 共通ヘッダファイル              |
| 7727.H      | … | SH7727 内部レジスタ定義ヘッダファイル |
| Vector.c    | … | 例外処理                   |
| dsp.src     | … | DSP 処理                 |
| startup.src | … | スタートアップ処理              |
| section.src | … | セクション定義                |
| cpu.h       | … | CPU 定義ファイル             |
| Usbh.c      | … | USB ホスト処理              |
| Usbh.h      | … | USB ホストヘッダファイル         |

##### <shc¥ap\_sh3d\_2a\_big¥src¥USBF フォルダ内>

|                       |   |                              |
|-----------------------|---|------------------------------|
| dobulk.c              | … | バルク転送処理                      |
| docontrol.c           | … | コントロール転送処理                   |
| dorequest.c           | … | 標準コマンドのデコード処理                |
| dorequestcomcommand.c | … | コミュニケーションコマンド実行処理            |
| usbmain.c             | … | USB ファンクションメイン処理             |
| catprototype.h        | … | 関数・変数のプロトタイプヘッダファイル          |
| cattypedef.h          | … | 構造体定義ヘッダファイル                 |
| setmacro.h            | … | マクロ定義ヘッダファイル                 |
| setsystemswitch.h     | … | システム設定ヘッダファイル                |
| setusbinfo.h          | … | USB ファンクション情報設定ヘッダファイル       |
| systemmap.h           | … | USB ファンクション関係メモリマップ設定ヘッダファイル |

## &lt;shc¥ap\_sh3d\_2a\_big¥Debug フォルダ内&gt;

- ap\_sh3d\_2a\_big.abs
  - ap\_sh3d\_2a\_big.mot
  - ap\_sh3d\_2a\_big.map
- … RAM 動作オブジェクトファイル(elf 形式)
  - … RAM 動作モトローラ S フォーマット形式ファイル
  - … RAM 動作マップファイル
- コンパイル後は、.obj,.lib 等のファイルが生成されます

## &lt;shc¥ap\_sh3d\_2a\_big¥Release フォルダ内&gt;

- ap\_sh3d\_2a\_big.abs
  - ap\_sh3d\_2a\_big.mot
  - ap\_sh3d\_2a\_big.map
- … ROM 動作オブジェクトファイル (elf 形式)
  - … ROM 動作モトローラ S フォーマット形式ファイル
  - … ROM 動作マップファイル
- コンパイル後は、.obj,.lib 等のファイルが生成されます



## (2) GCC 用サンプルプログラム

GCC 用サンプルプログラムは以下のファイルで構成されています。

|                                     |  |
|-------------------------------------|--|
| <gcc¥フォルダ内>                         |  |
| ap_sh3d_2a_big.hws                  | … HEW 用ワークスペースファイル                           |
| <gcc¥ap_sh3d_2a_big¥フォルダ内>          |  |
| ap_sh3d_2a_big.hwp                  | … HEW 用プロジェクトファイル                            |
| DefaultSession.hsf                  | … セッション情報ファイル                                |
| <gcc¥ap_sh3d_2a_big¥src フォルダ内>      |  |
| main.c                              | … メイン処理                                      |
| tmr.c                               | … タイマ処理                                      |
| sci.c                               | … シリアル処理                                     |
| boot.c                              | … CPU 初期化処理                                  |
| common.h                            | … 共通ヘッダファイル                                  |
| 7727.H                              | … SH7727 内部レジスタ定義ヘッダファイル                     |
| Vector.c                            | … 例外処理                                       |
| crt0.s                              | … スタートアップ処理                                  |
| cpu.h                               | … CPU 定義ファイル                                 |
| Usbh.c                              | … USB ホスト処理                                  |
| Usbh.h                              | … USB ホストヘッダファイル                             |
| <gcc¥ap_sh3d_2a_big¥src¥USBF フォルダ内> |  |
| dobulk.c                            | … バルク転送処理                                    |
| docontrol.c                         | … コントロール転送処理                                 |
| dorequest.c                         | … 標準コマンドのデコード処理                              |
| dorequestcomcommand.c               | … コミュニケーションコマンド実行処理                          |
| usbmain.c                           | … USB ファンクションメイン処理                           |
| catprototype.h                      | … 関数・変数のプロトタイプヘッダファイル                        |
| cattypedef.h                        | … 構造体定義ヘッダファイル                               |
| setmacro.h                          | … マクロ定義ヘッダファイル                               |
| setssystemswitch.h                  | … システム設定ヘッダファイル                              |
| setusbinfo.h                        | … USB ファンクション情報設定ヘッダファイル                     |
| systemmap.h                         | … USB ファンクション関係メモリマップ設定ヘッダファイル               |
| <gcc¥ap_sh3d_2a_big¥Debug フォルダ内>    |  |
| ap_sh3d_2a_big.x                    | … RAM 動作用オブジェクトファイル(elf 形式)                  |
| ap_sh3d_2a_big.mot                  | … RAM 動作用モトローラ S フォーマット形式ファイル                |
| ap_sh3d_2a_big.map                  | … RAM 動作用マップファイル                             |
| gnuconfig.ini                       | … GNUSH 構成設定ファイル<br>コンパイル後は、.o 等のファイルが生成されます |
| <gcc¥ap_sh3d_2a_big¥Release フォルダ内>  |  |
| ap_sh3d_2a_big.x                    | … ROM 動作用オブジェクトファイル(elf 形式)                  |
| ap_sh3d_2a_big.mot                  | … ROM 動作用モトローラ S フォーマット形式ファイル                |
| ap_sh3d_2a_big.map                  | … ROM 動作用マップファイル                             |
| gnuconfig.ini                       | … GNUSH 構成設定ファイル<br>コンパイル後は、.o 等のファイルが生成されます |

## 2. 2. 2 リトルエンディアン設定時 (AP-SH3D-2A\_Little) のファイル構成

## (1) SHC 用サンプルプログラム

SHC 用サンプルプログラムは以下のファイルで構成されています。

## &lt;shc¥フォルダ内&gt;

ap\_sh3d\_2a\_little.hws … HEW 用ワークスペースファイル

## &lt;shc¥ap\_sh3d\_2a\_little¥フォルダ内&gt;

ap\_sh3d\_2a\_little.hwp … HEW 用プロジェクトファイル

## &lt;gcc¥ap\_sh3d\_2a\_little¥src フォルダ内&gt;

main.c … メイン処理  
 tmr.c … タイマ処理  
 sci.c … シリアル処理  
 boot.c … CPU 初期化処理  
 common.h … 共通ヘッダファイル  
 7727.H … SH7727 内部レジスタ定義ヘッダファイル  
 Vector.c … 例外処理  
 dsp.src … DSP 処理  
 startup.src … スタートアップ処理  
 section.src … セクション定義  
 cpu.h … CPU 定義ファイル  
 Usbh.c … USB ホスト処理  
 Usbh.h … USB ホストヘッダファイル

## &lt;gcc¥ap\_sh3d\_2a\_little¥src¥USB フォルダ内&gt;

dobulk.c … バルク転送処理  
 docontrol.c … コントロール転送処理  
 dorequest.c … 標準コマンドのデコード処理  
 dorequestcomcommand.c … コミュニケーションコマンド実行処理  
 usbmain.c … USB ファンクションメイン処理  
 catprotype.h … 関数・変数のプロトタイプヘッダファイル  
 cattypedef.h … 構造体定義ヘッダファイル  
 setmacro.h … マクロ定義ヘッダファイル  
 setsystemswitch.h … システム設定ヘッダファイル  
 setusbinfo.h … USB ファンクション情報設定ヘッダファイル  
 systemmap.h … USB ファンクション関係メモリマップ設定ヘッダファイル

## &lt;shc¥ap\_sh3d\_2a\_little¥Debug フォルダ内&gt;

ap\_sh3d\_2a\_little.abs … RAM 動作用オブジェクトファイル (elf 形式)  
 ap\_sh3d\_2a\_little.mot … RAM 動作用モトローラ S フォーマット形式ファイル  
 ap\_sh3d\_2a\_little.map … RAM 動作用マップファイル  
 コンパイル後は、.obj,.lib 等のファイルが生成されます

## &lt;shc¥ap\_sh3d\_2a\_little¥Release フォルダ内&gt;

ap\_sh3d\_2a\_little.abs … ROM 動作用オブジェクトファイル (elf 形式)  
 ap\_sh3d\_2a\_little.mot … ROM 動作用モトローラ S フォーマット形式ファイル  
 ap\_sh3d\_2a\_little.map … ROM 動作用マップファイル  
 コンパイル後は、.obj,.lib 等のファイルが生成されます

## (2) GCC 用サンプルプログラム

GCC 用サンプルプログラムは以下のファイルで構成されています。

## &lt;gcc¥フォルダ内&gt;

ap\_sh3d\_2a\_little.hws … HEW 用ワークスペースファイル

## &lt;gcc¥ap\_sh3d\_2a\_little¥フォルダ内&gt;

ap\_sh3d\_2a\_little.hwp … HEW 用プロジェクトファイル  
DefaultSession.hsf … セッション情報ファイル

## &lt;gcc¥ap\_sh3d\_2a\_little¥src フォルダ内&gt;

main.c … メイン処理  
tmr.c … タイマ処理  
sci.c … シリアル処理  
boot.c … CPU 初期化处理  
common.h … 共通ヘッダファイル  
7727.H … SH7727 内部レジスタ定義ヘッダファイル  
Vector.c … 例外処理  
crt0.s … スタートアップ処理  
cpu.h … CPU 定義ファイル  
Usbh.c … USB ホスト処理  
Usbh.h … USB ホストヘッダファイル

## &lt;gcc¥ap\_sh3d\_2a\_little¥src¥USB フォルダ内&gt;

dobulk.c … バルク転送処理  
docontrol.c … コントロール転送処理  
dorequest.c … 標準コマンドのデコード処理  
dorequestcomcommand.c … コミュニケーションコマンド実行処理  
usbmain.c … USB ファンクションメイン処理  
catprototype.h … 関数・変数のプロトタイプヘッダファイル  
cattypedef.h … 構造体定義ヘッダファイル  
setmacro.h … マクロ定義ヘッダファイル  
setsystemswitch.h … システム設定ヘッダファイル  
setusbinfo.h … USB ファンクション情報設定ヘッダファイル  
systememmap.h … USB ファンクション関係メモリマップ設定ヘッダファイル

## &lt;gcc¥ap\_sh3d\_2a\_little¥Debug フォルダ内&gt;

ap\_sh3d\_2a\_little.x … RAM 動作用オブジェクトファイル(elf 形式)  
ap\_sh3d\_2a\_little.mot … RAM 動作用モトローラ S フォーマット形式ファイル  
ap\_sh3d\_2a\_little.map … RAM 動作用マップファイル  
gnuconfig.ini … GNUSH 構成設定ファイル  
コンパイル後は、.o 等のファイルが生成されます

## &lt;gcc¥ap\_sh3d\_2a\_little¥Release フォルダ内&gt;

ap\_sh3d\_2a\_little.x … ROM 動作用オブジェクトファイル(elf 形式)  
ap\_sh3d\_2a\_little.mot … ROM 動作用モトローラ S フォーマット形式ファイル  
ap\_sh3d\_2a\_little.map … ROM 動作用マップファイル  
gnuconfig.ini … GNUSH 構成設定ファイル  
コンパイル後は、.o 等のファイルが生成されます

### 3. ビルド・デバッグ方法

#### 3. 1 ビッグエンディアン設定時

##### 3. 1. 1 Renesas 社製 SHC 用サンプルプログラム (Ver6 以降)

###### (1) ビルド

HEW (Renesas SHC Ver6 以降) を使用した場合のサンプルプログラムのビルド方法を説明します。

- ① HEW を起動し、ap\_sh3d\_2a\_big.hws を読み込みます。
- ② 最初の読み込みを行なったときに、「ワークスペース (Workspace) が移動しました」という内容の確認メッセージが表示されますので「はい」を選択して下さい。
- ③ 最初の読み込みを行なったときに、コンパイラバージョンによって、バージョンの選択を行なうダイアログが表示されることがあります。表示された場合には、使用するコンパイラバージョンを選択して下さい。
- ④ [Build] ボタン横のリストボックス [Configuration Section] から、[Debug] または [Release] を選択します。  
[Debug] を選択した場合、¥Debug ワークフォルダ内に RAM 動作用のオブジェクトが生成されます。  
[Release] を選択した場合、¥Release ワークフォルダ内に ROM 動作用のオブジェクトが生成されます。
- ⑤ メニューの [Build] - [Build] を実行して下さい。ap\_sh3d\_2a\_big.mot、ap\_sh3d\_2a\_big.abs が出力されます。  
このとき、マップファイルは、ワークフォルダに作成されます。

HEW の詳細な使用方法につきましては、HEW のマニュアルを参照して下さい。

###### (2) RAM 上でのデバッグ

- ① AP-SH3D-2A のスイッチを「1.2 動作モード」「Fig1.2-1 ビッグエンディアン時の動作モード設定」に合わせて設定します。
- ② XrossFinder を起動し、¥sample フォルダ直下にある XrossFinder\_sh3d\_2a.xfc コマンドファイルを読み込みます。
- ③ ¥Debug フォルダ内の ap\_sh3d\_2a\_big.abs を XrossFinder でダウンロードして動作を確認して下さい。

###### (3) ROM 上でのデバッグ

- ① AP-SH3D-2A のスイッチを「Fig1.2-1 ビッグエンディアン時の動作モード設定」に合わせて設定します。
- ② ¥Release フォルダ内の ap\_sh3d\_2a\_big.abs を XrossFinder で読み込みます。
- ③ XrossFinder のメニューから FlashWriter EX を選択し、下図 Fig3.1-1 のように設定を行ってください。
- ④ START ボタンを押してプログラムの書き込みを行い、動作を確認して下さい。

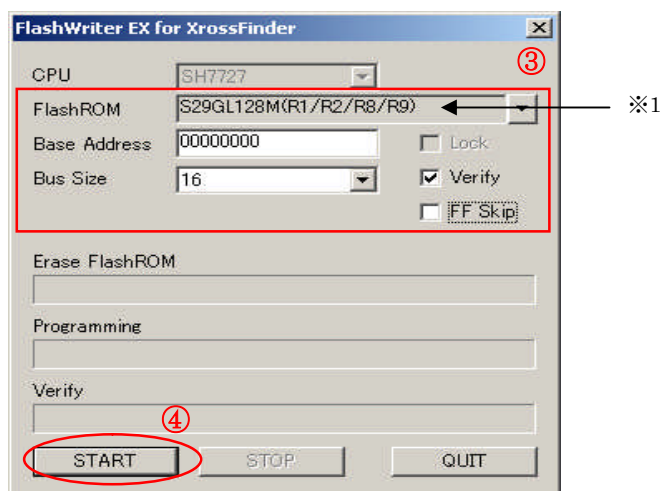


Fig3.1-1 FlashWriter EX for XrossFinder の設定

## (4) XrossFinder 未使用時の確認方法

## ・FlashWriterEX を使用する場合

- ① アダプタ (HJ-LINK または XrossFinder) を使用して PC とボードを繋ぎます。
- ② FlashWriterEX を起動して、「Table3.1-1 FlashWriterEX の設定 (ビッグエンディアン)」を参考に設定を行ってください。
- ③ ¥Release フォルダ内の ap\_sh3d\_2a\_big.mot をボードに書き込み、動作確認を行ってください。

FlashWriter EX の使用方法の詳細につきましては、FlashWriter EX のマニュアルを参照して下さい。

|                               |  |
|-------------------------------|--|
| アダプタ設定                        | XrossFinder 使用時は「XrossFinder」<br>HJ-LINK 使用時は「HJ-LINK」 |
| JTAG クロック (XrossFinder 使用時のみ) | 20MHz 以下   |
| CPU                           | SH7727   |
| BaseAddress                   | 00000000   |
| FlashROM                      | S29GL128M (R1/R2/R8/R9) ※1                             |
| Bus Size                      | 16   |
| Endian                        | Big  |

Table3.1-1 FlashWriter EX の設定 (ビッグエンディアン)

※1. 本ボードに実装されている FlashROM は、生産中止等の理由により変更することがございます。本アプリケーションノートでの設定は、「S29GL128P90TFIR20 (SPANSION)」が実装されているボードでの設定となります。お手元の CPU ボードに実装されている FlashROM の型番と異なっている場合には、お手元のボードに実装されている FlashROM の型番にあわせて設定を行ってください。

## 3. 1. 2 KPIT 社製 GNU GCC 用サンプルプログラム (GNUSH v0701 以降)

## (1) ビルド・デバッグ

HEW (KPIT GNUSH v0701 以降) を使用した場合のサンプルプログラムのビルド・デバッグ方法も 3.1.1 と同様です。

¥gcc フォルダ 直下にある ap\_sh3d\_2a\_big.hws を読み込んでビルドを実行して下さい。また、最初の読み込みを行なったときに「セッションが見つかりません」という内容の確認メッセージが表示されますので、「はい」を選択して下さい。その後、ビルド後生成されたオブジェクトファイル ap\_sh3d\_2a\_big.x を XrossFinder で読み込んで動作を確認して下さい。

もしくは、ビルド実行後に FlashWriter EX を使用して、ap\_sh3d\_2a\_big\_dsp.mot をボードに書き込み、動作を確認して下さい。

コンパイラならびに HEW は、KPIT 社のホームページよりダウンロード可能です。詳しくは弊社ホームページの技術コラムを参照して下さい。また、GCC の詳細な使用方法につきましては、GCC コンパイラのマニュアル、もしくは、関連書籍、ホームページなどを参照して下さい。

### 3. 2 リトルエンディアン設定時

#### 3. 2. 1 Renesas 社製 SHC 用サンプルプログラム (Ver6 以降)

##### (1) ビルド

HEW (Renesas SHC Ver6 以降) を使用した場合のサンプルプログラムのビルド方法を説明します。

- ① HEW を起動し、ap\_sh3d\_2a\_little.hws を読み込みます。
- ② 最初の読み込みを行なったときに、「ワークスペース (Workspace) が移動しました」という内容の確認メッセージが表示されますので「はい」を選択して下さい。
- ③ 最初の読み込みを行なったときに、コンパイラバージョンによって、バージョンの選択を行なうダイアログが表示されることがあります。表示された場合には、使用するコンパイラバージョンを選択して下さい。
- ④ [Build] ボタン横のリストボックス [Configuration Section] から、[Debug] または [Release] を選択します。  
[Debug] を選択した場合、¥Debug ワークフォルダ内に RAM 動作のオブジェクトが生成されます。  
[Release] を選択した場合、¥Release ワークフォルダ内に ROM 動作のオブジェクトが生成されます。
- ⑤ メニューの [Build] - [Build] を実行して下さい。ap\_sh3d\_2a\_little.mot、ap\_sh3d\_2a\_little.abs が出力されます。このとき、マップファイルは、ワークフォルダに作成されます。

HEW の詳細な使用方法につきましては、HEW のマニュアルを参照して下さい。

##### (2) RAM 上でのデバッグ

- ① AP-SH3D-2A のスイッチを「1.2 動作モード」 「Fig1.2-2 リトルエンディアン時の動作モード設定」に合わせて設定します。
- ② XrossFinder を起動し、¥sample フォルダ直下にある XrossFinder\_sh3d\_2a.xfc コマンドファイルを読み込みます。
- ③ ¥Debug フォルダ内の ap\_sh3d\_2a\_little.abs を XrossFinder でダウンロードして動作を確認して下さい。

##### (3) ROM 上でのデバッグ

- ① AP-SH3D-2A のスイッチを「Fig1.2-1 ビッグエンディアン時の動作モード設定」に合わせて設定します。
- ② ¥Release フォルダ内の ap\_sh3d\_2a\_little.abs を XrossFinder で読み込みます。
- ③ XrossFinder のメニューから FlashWriter EX を選択し、下図 Fig3.2-1 のように設定を行ってください。
- ④ START ボタンを押してプログラムの書き込みを行い、動作を確認して下さい。

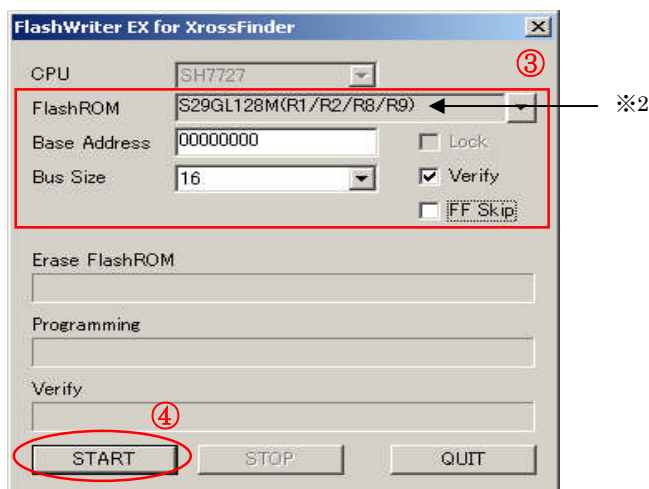


Fig3.2-1 FlashWriter EX for XrossFinder の設定

## (4) XrossFinder 未使用時の確認方法

## ・FlashWriterEX を使用する場合

- ① アダプタ (HJ-LINK または XrossFinder) を使用して PC とボードを繋ぎます。
- ② FlashWriterEX を起動して、「Table3.2-1 FlashWriterEX の設定 (リトルエンディアン)」を参考に設定を行ってください。
- ③ ¥Release フォルダ内の ap\_sh3d\_2a\_little.mot をボードに書き込み、動作確認を行ってください。

FlashWriter EX の使用方法の詳細につきましては、FlashWriter EX のマニュアルを参照して下さい。

|                               |  |
|-------------------------------|--|
| アダプタ設定                        | XrossFinder 使用時は「XrossFinder」<br>HJ-LINK 使用時は「HJ-LINK」 |
| JTAG クロック (XrossFinder 使用時のみ) | 20MHz 以下   |
| CPU                           | SH7727   |
| BaseAddress                   | 00000000   |
| FlashROM                      | S29GL128M (R1/R2/R8/R9) ※2                             |
| Bus Size                      | 16   |
| Endian                        | Little   |

Table3.2-1 FlashWriter EX の設定 (リトルエンディアン)

※2. 本ボードに実装されている FlashROM は、生産中止等の理由により変更することがございます。本アプリケーションノートでの設定は、「S29GL128P90TFIR20 (SPANSION)」が実装されているボードでの設定となります。お手元の CPU ボードに実装されている FlashROM の型番と異なっている場合には、お手元のボードに実装されている FlashROM の型番にあわせて設定を行ってください。

## 3. 2. 2 KPIT 社製 GNU GCC 用サンプルプログラム (GNUSH v0701 以降)

## (1) ビルド・デバッグ

HEW (KPIT GNUSH v0701 以降) を使用した場合のサンプルプログラムのビルド・デバッグ方法も 3.2.1 と同様です。

¥gcc フォルダ 直下にある ap\_sh3d\_2a\_little.hws を読み込んでビルドを実行して下さい。また、最初の読み込みを行なったときに「セッションが見つかりません」という内容の確認メッセージが表示されますので、「はい」を選択して下さい。その後、ビルド後生成されたオブジェクトファイル ap\_sh3d\_2a\_little.x を XrossFinder で読み込んで動作を確認して下さい。

もしくは、ビルド実行後に FlashWriter EX を使用して、ap\_sh3d\_2a\_little.dsp.mot をボードに書き込み、動作を確認して下さい。

コンパイラならびに HEW は、KPIT 社のホームページよりダウンロード可能です。詳しくは弊社ホームページの技術コラムを参照して下さい。また、GCC の詳細な使用方法につきましては、GCC コンパイラのマニュアル、もしくは、関連書籍、ホームページなどを参照して下さい。

## 4. 動作説明

### 4. 1 SHC 用サンプルプログラムの動作説明

SHC 用サンプルプログラムは、下記の動作を行いません。

- SCIF (SC12) から入力された数値の移動平均を算出し SCIF へ返します。(送受信割り込み、DSP 使用)  
SCIF から三桁の数値と合わせて、改行・クリア・コンマ (ASCII コードでそれぞれ、0x0a・0x0d・0x2c) のいずれかを入力すると、DSP を使用して移動平均を行ないその結果を SCIF から出力します。5つの入力値で移動平均を行いません。  
上記以外の入力は、エコーバックを行います。  
COM ポートの設定は、38400bps、ビット長 8、パリティなし、ストップビット 1 です。  
動作確認は、パソコン上のターミナルソフト (telnet など) を使用して行って下さい。
- LD1 (緑の LED) を 500msec 間隔で ON/OFF します。(TMU0 割り込み使用)
- LD2 (緑の LED) を 1000msec 間隔で ON/OFF します。(TMU0 割り込み使用)
- CN2 のポートより方形波を出力します。周期とピン番号を以下に示します。
- USB ホストに USB メモリを接続すると、SCIF (SC12) から接続状況とデバイス情報を出力します。 ※1
- USB ホストに USB マウスを接続すると、SCIF (SC12) から接続状況とマウスのボタン情報や位置情報を出力します。  
※1
- USB ファンクションをパソコンに接続すると、仮想 COM ポートとして認識され、USB シリアルとしてエコーバックを行います。 ※2  
※1. USB ホスト動作の詳細は、「4.5 USB ホスト動作」を参照してください。  
※2. USB ファンクション動作の詳細は、「4.6 USB ファンクション動作」を参照してください。

CN2 方形波出力端子一覧

| ピン番号 | ピン名              | 周期     | 備考      |
|------|------------------|--------|---------|
| 7    | LCD0/PTD0        | 10msec | TMU0 使用 |
| 8    | LCD1/PTD1        | 10msec | TMU0 使用 |
| 9    | LCD2/PTC0        | 20msec | TMU0 使用 |
| 10   | LCD3/PTC1        | 20msec | TMU0 使用 |
| 11   | LCD4/PTC2        | 20msec | TMU0 使用 |
| 12   | LCD5/PTC3        | 20msec | TMU0 使用 |
| 13   | LCD6/PTD2        | 10msec | TMU0 使用 |
| 14   | LCD7/PTD3        | 10msec | TMU0 使用 |
| 15   | LCD8/PTC4/PINT0  | 20msec | TMU0 使用 |
| 16   | LCD9/PTC5/PINT1  | 20msec | TMU0 使用 |
| 17   | LCD10/PTC6/PINT2 | 20msec | TMU0 使用 |
| 18   | LCD11/PTC7/PINT3 | 20msec | TMU0 使用 |
| 45   | PTD7/DON         | 10msec | TMU0 使用 |
| 46   | PTD5/CL1         | 10msec | TMU0 使用 |
| 49   | #CE2B/PTE5       | 10msec | TMU0 使用 |
| 50   | #CE2A/PTE4       | 10msec | TMU0 使用 |
| 51   | PTE3/FLM         | 10msec | TMU0 使用 |

信号名に#がついているものは負論理を表します。



## 4. 2 GCC 用サンプルプログラムの動作説明

GCC 用サンプルプログラムは、下記の動作を行いません。

- SCIF (SC12) でエコーバックを行いません。(送受信割り込み使用)  
SCIF から受信をした値をそのまま、SCIF へ送信します。  
COM ポートの設定は、38400bps、ビット長 8、パリティなし、ストップビット 1 です。  
動作確認は、パソコン上のターミナルソフト (telnet など) を使用して行って下さい。
- LD1 (緑の LED) を 500msec 間隔で ON/OFF します。(TMU0 割り込み使用)
- LD2 (緑の LED) を 1000msec 間隔で ON/OFF します。(TMU0 割り込み使用)
- CN2 のポートより方形波を出力します。周期とピン番号を以下に示します。
- USB ホストに USB メモリを接続すると、SCIF (SC12) から接続状況とデバイス情報を出力します。 ※1
- USB ホストに USB マウスを接続すると、SCIF (SC12) から接続状況とマウスのボタン情報や位置情報を出力します。  
※1
- USB ファンクションをパソコンに接続すると、仮想 COM ポートとして認識され、USB シリアルとしてエコーバックを行います。 ※2  
※1. USB ホスト動作の詳細は、「4.5 USB ホスト動作」を参照してください。  
※2. USB ファンクション動作の詳細は、「4.6 USB ファンクション動作」を参照してください。

CN2 方形波出力端子一覧

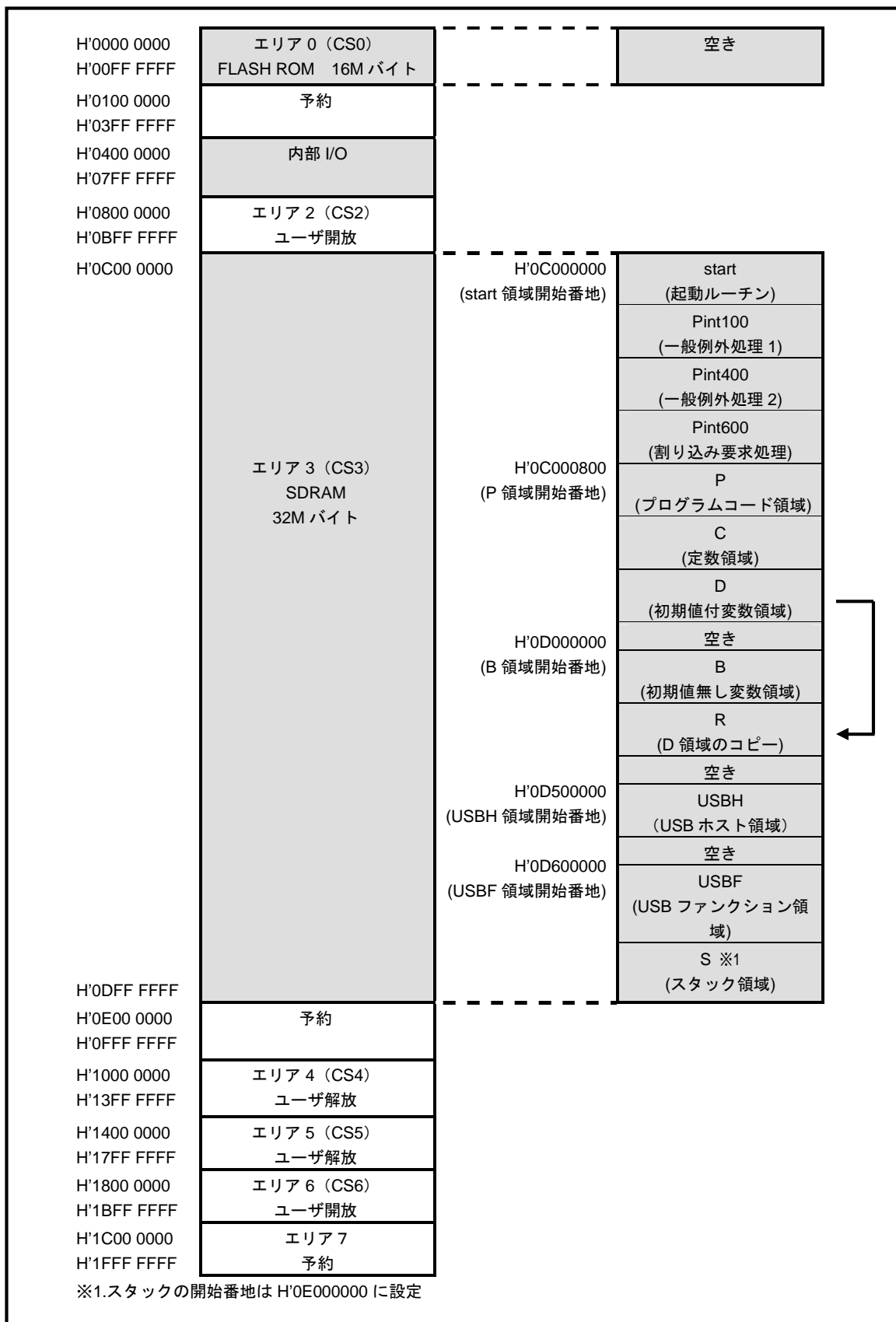
| ピン番号 | ピン名              | 周期     | 備考      |
|------|------------------|--------|---------|
| 7    | LCD0/PTD0        | 10msec | TMU0 使用 |
| 8    | LCD1/PTD1        | 10msec | TMU0 使用 |
| 9    | LCD2/PTC0        | 20msec | TMU0 使用 |
| 10   | LCD3/PTC1        | 20msec | TMU0 使用 |
| 11   | LCD4/PTC2        | 20msec | TMU0 使用 |
| 12   | LCD5/PTC3        | 20msec | TMU0 使用 |
| 13   | LCD6/PTD2        | 10msec | TMU0 使用 |
| 14   | LCD7/PTD3        | 10msec | TMU0 使用 |
| 15   | LCD8/PTC4/PINT0  | 20msec | TMU0 使用 |
| 16   | LCD9/PTC5/PINT1  | 20msec | TMU0 使用 |
| 17   | LCD10/PTC6/PINT2 | 20msec | TMU0 使用 |
| 18   | LCD11/PTC7/PINT3 | 20msec | TMU0 使用 |
| 45   | PTD7/DON         | 10msec | TMU0 使用 |
| 46   | PTD5/CL1         | 10msec | TMU0 使用 |
| 49   | #CE2B/PTE5       | 10msec | TMU0 使用 |
| 50   | #CE2A/PTE4       | 10msec | TMU0 使用 |
| 51   | PTE3/FLM         | 10msec | TMU0 使用 |

信号名に#がついているものは負論理を表します。

### 4. 3 RAM 動作時のメモリマップ

メモリマップを以下に示します。

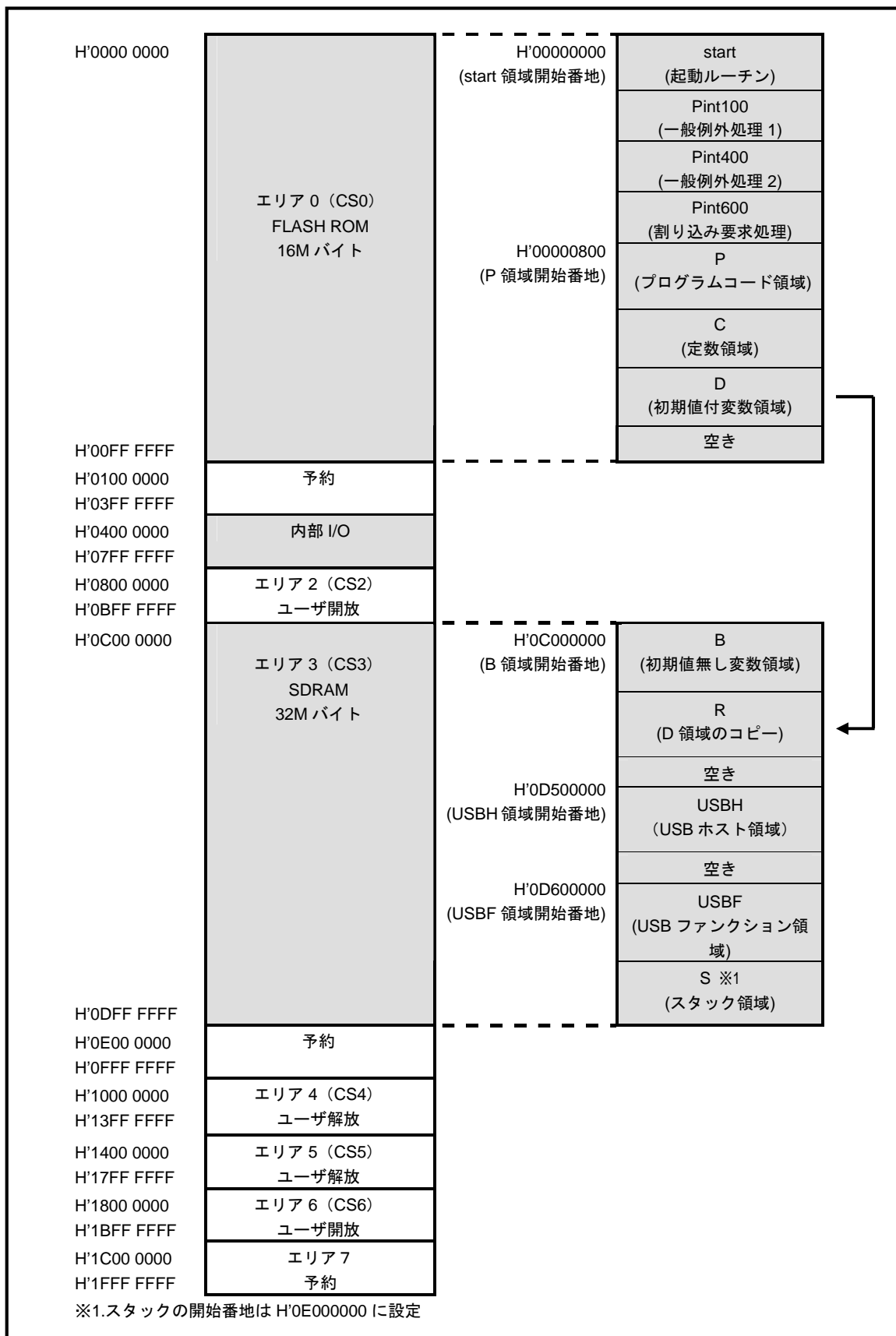
shc 版、gcc 版共にプログラム領域の割り当ては同じです。(図に示す領域は shc 版の名称を使用しています。)



#### 4. 4 ROM 動作時のメモリマップ

メモリマップを以下に示します。

shc 版、gcc 版共にプログラム領域の割り当ては同じです。(図に示す領域は shc 版の名称を使用しています。)



## 4. 5 USB ホスト動作

### 4. 5. 1 USB メモリ接続時の動作内容

以下の手順に従い、USB メモリ接続時の動作を確認してください。

- ① CPU ボードとパソコンをシリアルケーブルで接続します。
- ② パソコン上でターミナルソフト (telnet など) を起動し、COM ポートの設定を行います。  
COM ポートの設定は、38400bps、ビット長 8、パリティなし、ストップビット 1 です。
- ③ CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ④ USB ホストポート (CN6) に USB メモリを挿入します。
- ⑤ USB メモリを挿入すると、自動的に USB メモリの接続状況とデバイス情報がターミナルソフト上に表示されます。以下のような表示が出ていることを確認してください。

```
Connected !!  
TD.ConditionCode = 0 SET ADDRESS OK !!  
TD.ConditionCode = 0 GET DESCRIPTOR (DEVICE) OK !!  
TD.ConditionCode = 0 GET DESCRIPTOR (CONFIG) OK !!  
TD.ConditionCode = 0 SET CONFIG OK !!  
MSC BOT Inquiry Command OK !!  
Received String Data : XXXXXXXX ※  
Transfer End !!
```

※. XXXXXXXX には、USB メモリのデバイス情報が表示されます。

- ⑥ 上記の表示が出たことを確認した後に、USB メモリを抜いてください。ターミナルソフト上に切断状況が表示されず。以下のような表示が出ていることを確認してください。

```
DisConnected !!
```

- ⑦ 以上で USB メモリ接続時の動作は終了です。

## 4. 5. 2 USB マウス接続時の動作内容

以下の手順に従い、USB マウス接続時の動作を確認してください。

- ① CPU ボードとパソコンをシリアルケーブルで接続します。
- ② パソコン上でターミナルソフト (telnet など) を起動し、COM ポートの設定を行います。  
COM ポートの設定は、38400bps、ビット長 8、パリティなし、ストップビット 1 です。
- ③ CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ④ USB ホストポート (CN6) に USB マウスを挿入します。
- ⑤ USB マウスを挿入すると、自動的に USB マウスの接続状況がターミナルソフト上に表示されます。以下のような表示が出ていることを確認してください。

```
Connected !!
TD.ConditionCode = 0 SET ADDRESS OK !!
TD.ConditionCode = 0 GET DESCRIPTOR (DEVICE) OK !!
TD.ConditionCode = 0 GET DESCRIPTOR (CONFIG) OK !!
TD.ConditionCode = 0 SET CONFIG OK !!
```

- ⑥ 上記の表示が出たことを確認した後に、USB マウスを動作させると、マウスから取得したデータがターミナルソフト上に表示されます。以下に表示されるデータの一例を示します。

```
Interrupt Transfer OK !!
Received Interrupt Data : 01000000
Interrupt Transfer OK !!
Received Interrupt Data : 00000000
Interrupt Transfer OK !!
Received Interrupt Data : 02000000
Interrupt Transfer OK !!
Received Interrupt Data : 00000000
Interrupt Transfer OK !!
Received Interrupt Data : 00010000
Interrupt Transfer OK !!
Received Interrupt Data : 0002ff00
```

- ⑦ 上記の表示が出たことを確認した後に、USB マウスを抜いてください。ターミナルソフト上に切断状況が表示されません。以下のような表示が出ていることを確認してください。

```
DisConnected !!
```

- ⑧ 以上で USB マウス接続時の動作は終了です。

#### 4. 6 USB ファンクション動作

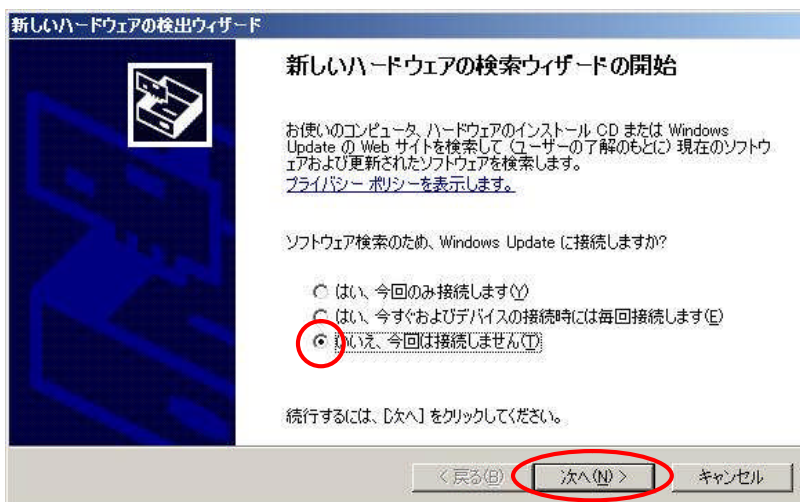
USB ファンクション動作に必要な推奨環境は以下のとおりです。

|             |  |
|-------------|--|
| パーソナルコンピュータ | PC/AT 互換機                                |
| OS          | Windows2000/XP/Vista                     |
| CD ドライブ     | CD-ROM 読み込み可能なドライブ (USB 仮想シリアルインストール時のみ) |
| USB ポート     | USB1.1 または 2.0 1 ポート                     |

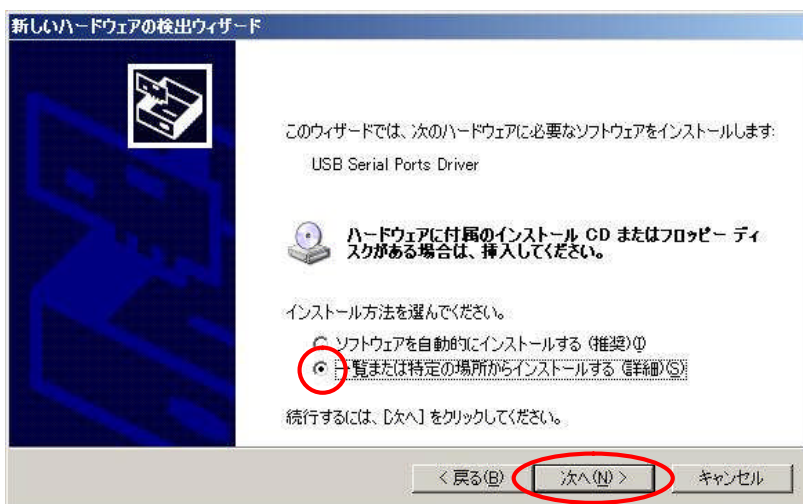
##### 4. 6. 1 USB 仮想シリアルインストール方法 (WindowsXP の場合)

USB ファンクションでの USB シリアル動作を行うには、パソコンに USB 仮想 COM ポートの設定を行う必要があります。以下の手順に従い、USB 仮想シリアルをインストールしてください。

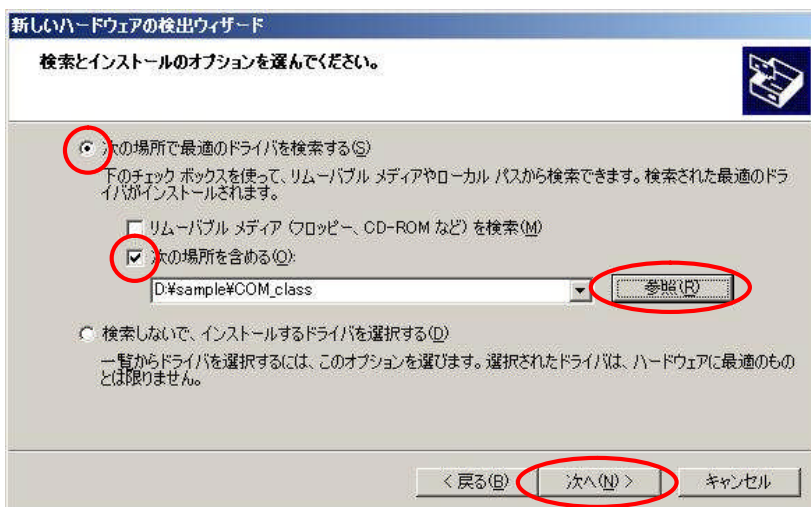
- ① CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ② パソコンの CD ドライブに添付 CD を挿入します。
- ③ USB ケーブルを使い、パソコンの USB ポートと CPU ボードの USB ファンクションポート (CN7) を接続します。
- ④ パソコン上で以下のような画面が表示されたら、「いいえ、今回は接続しません」にチェックをして「次へ」を選択してください。



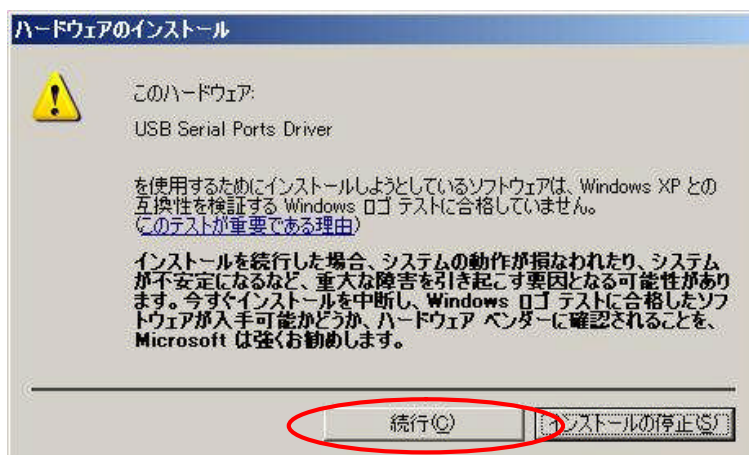
- ⑤ 次の画面が表示されたら、「一覧または特定の場所からインストールする」にチェックをして「次へ」を選択してください。



- ⑥ 次の画面が表示されたら、「次の場所で最適なドライバを検索する」と「次の場所を含める」にチェックをして「参照」ボタンから添付 CD 内の「¥sample¥COM\_class」を選び、「次へ」を選択してください。



- ⑦ 次の画面が表示されたら、「続行」を選択してください。



- ⑧ 次の画面が表示されたら、「完了」を選択してください。



- ⑨ 「スタート」→「コントロールパネル」→「パフォーマンスとメンテナンス」→「システム」→「ハードウェア」→「デバイスマネージャ」を順に選択し、ポート (COM と LPT) を見て設定された仮想 COM ポート番号を確認します。以下に一例を表示しますが、実際に設定される仮想 COM ポート番号はパソコンの環境により異なりますのでご注意ください。



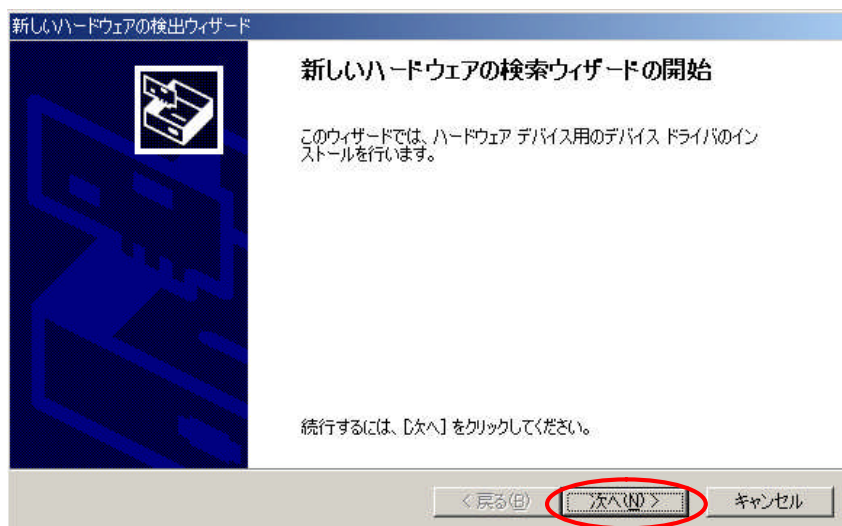
- ⑩ 以上で仮想シリアルインストールは終了です。



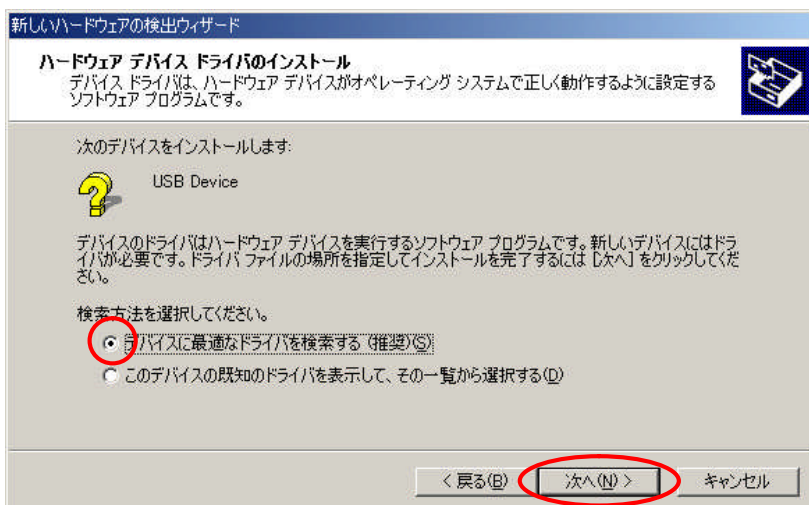
## 4. 6. 2 USB 仮想シリアルインストール方法 (Windows2000 の場合)

USB ファンクションでの USB シリアル動作を行うには、パソコンに USB 仮想 COM ポートの設定を行う必要があります。以下の手順に従い、USB 仮想シリアルをインストールしてください。

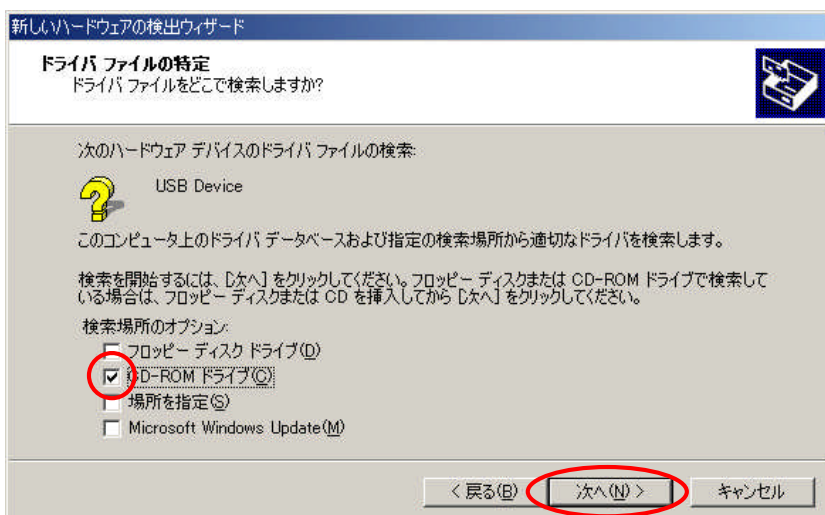
- ① CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ② パソコンの CD ドライブに添付 CD を挿入します。
- ③ USB ケーブルを使い、パソコンの USB ポートと CPU ボードの USB ファンクションポート (CN7) を接続します。
- ④ パソコン上で以下のような画面が表示されたら、「次へ」を選択してください。



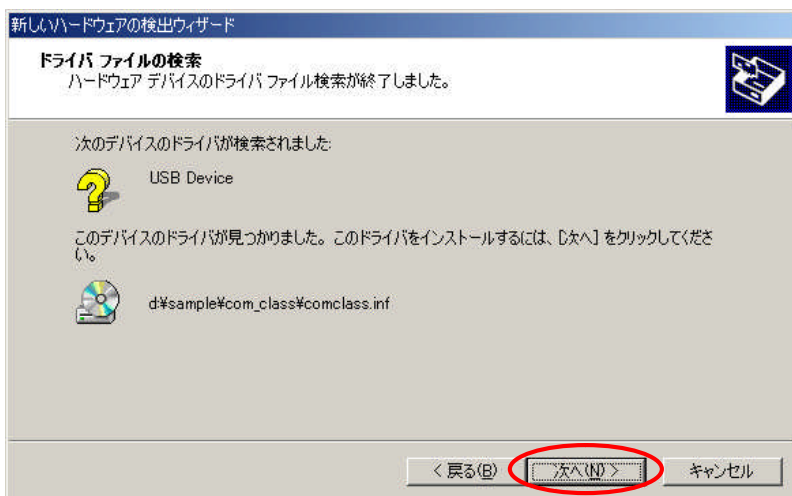
- ⑤ 次の画面が表示されたら、「デバイスに最適なドライバを検索する」にチェックをして「次へ」を選択してください。



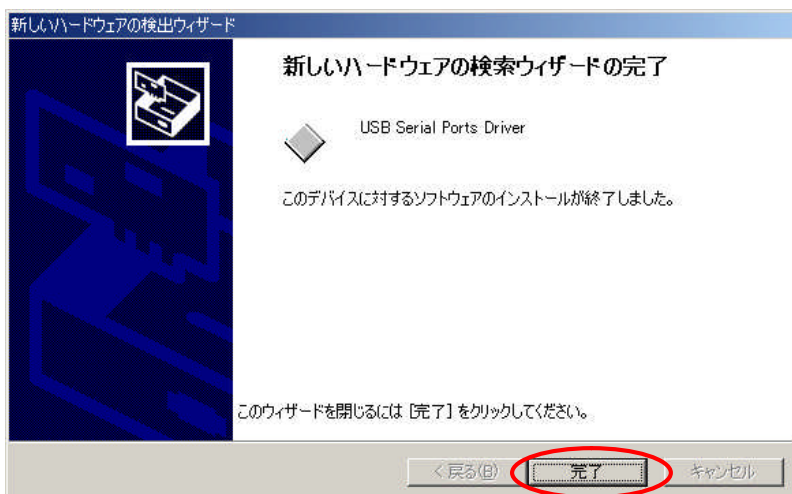
- ⑥ 次の画面が表示されたら、「CD-ROM ドライブ」にチェックをして「次へ」を選択してください。



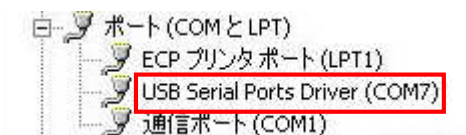
- ⑦ 次の画面が表示されたら、「次へ」を選択してください。



- ⑧ 次の画面が表示されたら、「完了」を選択してください。



- ⑨ 「マイコンピュータ」→「コントロールパネル」→「システム」→「ハードウェア」→「デバイスマネージャ」を順に選択し、ポート (COM と LPT) を見て設定された仮想 COM ポート番号を確認します。  
以下に一例を表示しますが、実際に設定される仮想 COM ポート番号はパソコンの環境により異なりますのでご注意ください。



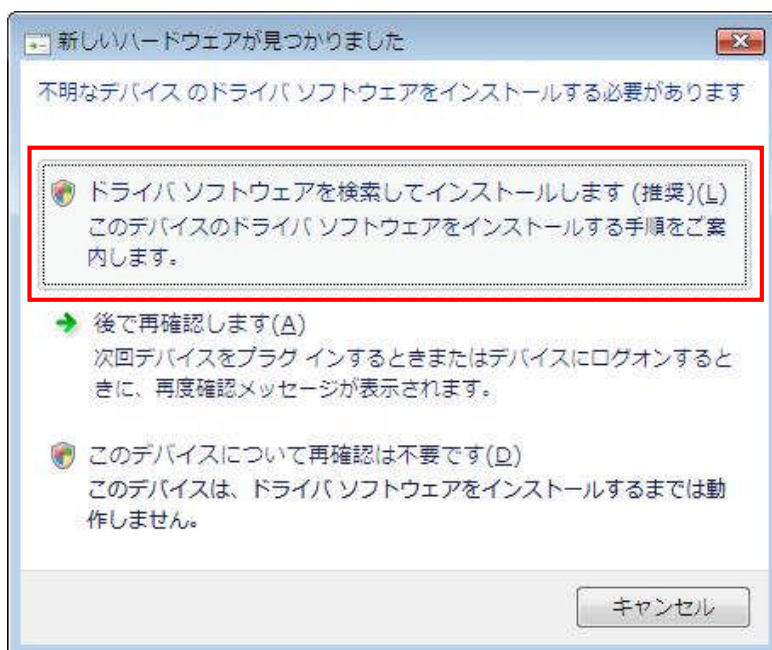
- ⑩ 以上で仮想シリアルインストールは終了です。

## 4. 6. 3 USB 仮想シリアルインストール方法 (WindowsVista の場合)

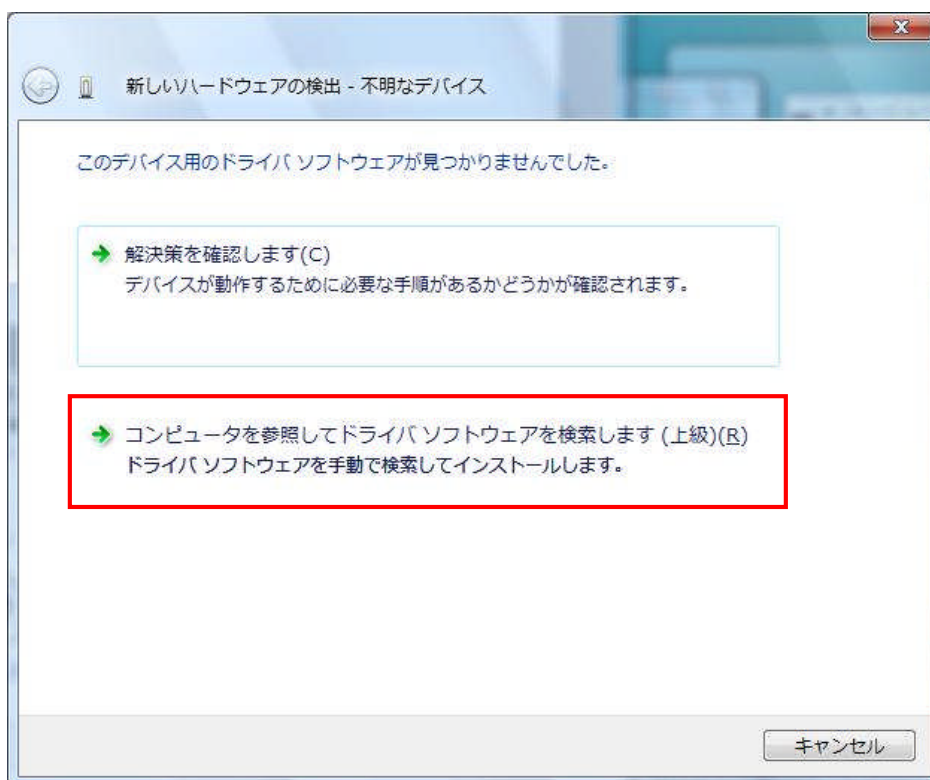
USB ファンクションでの USB シリアル動作を行うには、パソコンに USB 仮想 COM ポートの設定を行う必要があります。以下の手順に従い、USB 仮想シリアルをインストールしてください。

**注意：**WindowsVista を使用する際は、WindowsVistaSP1 にアップデートした後に以下の手順を行ってください。

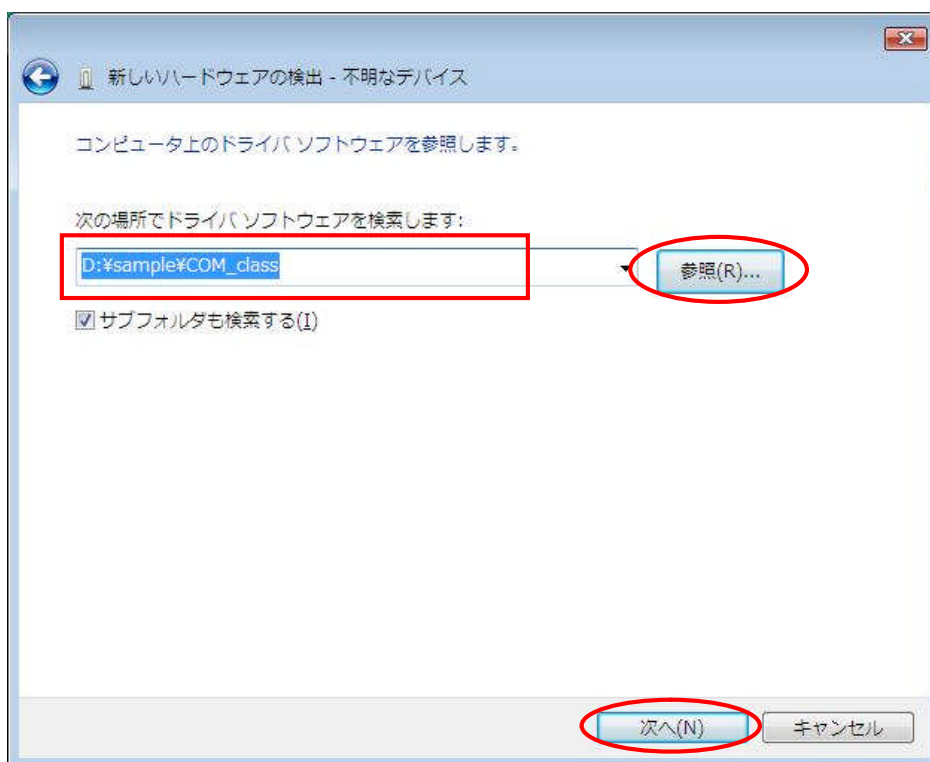
- ① CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ② パソコンの CD ドライブに添付 CD を挿入します。
- ③ USB ケーブルを使い、パソコンの USB ポートと CPU ボードの USB ファンクションポート (CN7) を接続します。
- ④ パソコン上で以下のような画面が表示されたら、「ドライバソフトウェアを検索してインストールします」を選択してください。



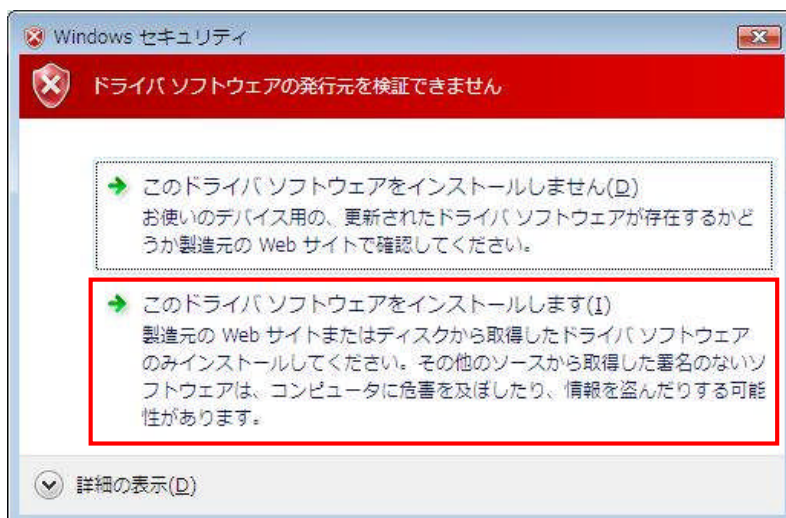
- ⑤ 次の画面が表示されたら、「コンピュータを参照してドライバソフトウェアを検索します」を選択してください。



- ⑥ 次の画面が表示されたら、「参照」ボタンから添付 CD 内の「¥sample¥COM\_class」を選び、「次へ」を選択してください。



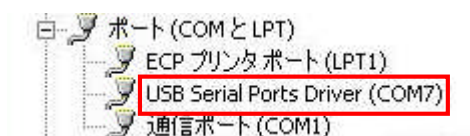
- ⑦ 次の画面が表示されたら、「このドライバソフトウェアをインストールします」を選択してください。



- ⑧ 次の画面が表示されたら、「閉じる」を選択してください。



- ⑨ 「スタート」→「コントロールパネル」→「システムとメンテナンス」→「システム」→「デバイスマネージャ」を順に選択し、ポート (COM と LPT) を見て設定された仮想 COM ポート番号を確認します。  
以下に一例を表示しますが、実際に設定される仮想 COM ポート番号はパソコンの環境により異なりますのでご注意ください。



- ⑩ 以上で仮想シリアルインストールは終了です。

#### 4. 6. 4 USB シリアル動作内容

以下の手順に従い、USB シリアルの動作を確認してください。

- ① USB ケーブルを使い、パソコンの USB ポートと CPU ボードの USB ファンクションポート (CN7) を接続します。
- ② CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ③ パソコン上でターミナルソフト (telnet など) を起動し、COM ポートの設定を行います。  
その際、使用する COM ポートは「4. 6. 1 USB シリアルインストール方法」で確認した仮想 COM ポートを選択してください。  
COM ポートの設定は、38400bps、ビット長 8、パリティなし、ストップビット 1 です。
- ④ ターミナルソフトを使用し、エコーバックが行われることを確認してください。
- ⑤ 以上で USB シリアルの動作は終了です。

## 参考文献

### ■Interface 2007 年 1 月号

「USB ターゲット & ホスト機器設計の完全理解 第 6 章 組み込み向け OHCI ホスト・コントローラの制御事例」  
音堂栄良／関根隆弘（株）ルネサステクノロジ

### ■TECHI USB ターゲット機器開発の全て

「USB コントローラ内蔵 SuperH&H8S マイコンの使い方」  
音堂栄良／池谷貴之

## 参考 HP

### ■Interface ホームページ

<http://www.cqpub.co.jp/interface/default.asp>

## ご注意

- ・本文書の著作権は株式会社アルファプロジェクトが保有します。
  - ・本文書の内容を無断で転載することは一切禁止します。
  - ・本文書に記載されているサンプルプログラムの著作権は株式会社ルネサステクノロジが保有します。
  - ・本文書に記載されている内容およびサンプルプログラムについての技術サポートは一切受け付けておりません。
  - ・本サンプルプログラムに関して、ルネサステクノロジへのお問い合わせはご遠慮ください。
  - ・本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社およびルネサステクノロジでは一切責任を負いませんのでご了承下さい。
  - ・本文書の内容については、万全を期して作成いたしました。が、万一ご不審な点、誤りなどお気づきの点がありましたら弊社までご連絡下さい。
  - ・本文書の内容は、将来予告なしに変更されることがあります。
- 
- ・Windows®の正式名称は Microsoft®Windows®Operating System です。  
Microsoft、Windows、Windows NT は、米国 Microsoft Corporation.の米国およびその他の国における商標または登録商標です。  
Windows®Vista、Windows®XP、Windows®2000 Professional、Windows®Millennium Edition、Windows®98 は、米国 Microsoft Corporation.の商品名称です。  
本文書では下記のように省略して記載している場合がございます。ご了承下さい。  
Windows®Vista は Windows Vista もしくは WinVista  
Windows®XP は Windows XP もしくは WinXP  
Windows®2000 Professional は Windows 2000 もしくは Win2000  
Windows®Millennium Edition は Windows Me もしくは WinMe  
Windows®98 は Windows 98 もしくは Win98
- 
- ・IBM-PC/AT は、米国 IBM 社の登録商標、商標または商品名称です。
  - ・DOS/V は、日本 IBM 社の登録商標、商標または商品名称です。
  - ・SuperH は、(株)Renesas の登録商標、商標または商品名称です。
- 
- ・その他の会社名、製品名は、各社の登録商標または商標です。



株式会社アルファプロジェクト  
〒431-3114  
静岡県浜松市東区積志町 834  
<http://www.apnet.co.jp>  
E-MAIL : [query@apnet.co.jp](mailto:query@apnet.co.jp)