

AP-SH4A-2A(SH7763 CPU BOARD)

サンプルプログラム解説

2版 2013年 6月27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.....	3
1.4 ワークスペースについて.....	3
2. サンプルプログラムの構成.....	4
2.1 フォルダ構成.....	4
2.2 ファイル構成.....	7
2.2.1 ビッグエンディアン・SHCのファイル構成.....	7
2.2.2 ビッグエンディアン・GCCのファイル構成.....	10
2.2.3 リトルエンディアン・SHCのファイル構成.....	13
2.2.4 リトルエンディアン・GCCのファイル構成.....	16
3. USBホストサンプルプログラム.....	19
3.1 ビルド・デバッグ方法（USBホスト）.....	19
3.2 動作説明（USBホスト）.....	21
3.2.1 サンプルプログラム概要（USBホスト）.....	21
3.2.2 USBホスト動作.....	22
3.3 RAM動作時のメモリマップ（USBホスト）.....	24
3.4 ROM動作時のメモリマップ（USBホスト）.....	25
4. USBファンクションサンプルプログラム.....	26
4.1 ビルド・デバッグ方法（USBファンクション）.....	26
4.2 動作説明（USBファンクション）.....	28
4.2.1 サンプルプログラム概要（USBファンクション）.....	28
4.2.2 USBファンクション動作.....	29
4.3 RAM動作時のメモリマップ（USBファンクション）.....	30
4.4 ROM動作時のメモリマップ（USBファンクション）.....	31

5. ネットワークサンプルプログラム	32
5.1 ビルド・デバッグ方法（ネットワーク）	32
5.2 ネットワークサンプルプログラムの動作説明	34
5.2.1 ネットワークサンプルプログラム概要	34
5.2.2 ネットワーク動作	35
5.2.3 ネットワークサンプルプログラム注意事項	36
5.3 RAM 動作時のメモリマップ（ネットワーク）	37
5.4 ROM 動作時のメモリマップ（ネットワーク）	38

1. 概要

1.1 概要

本アプリケーションノートでは、AP-SH4A-2A に付属するサンプルプログラムについて解説します。
AP-SH4A-2A には、Renesas 社製 SHC 用サンプルプログラムと KPIT 社製 GCC 用サンプルプログラムが付属しています。
本サンプルプログラムの概要を以下に示します

サンプルプログラム	動作内容
USB ホストサンプルプログラム	・ USB ホスト デバイス情報表示 ・ SCIF0 シリアル通信 ・ タイマ割り込み
USB ファンクションサンプルプログラム	・ USB ファンクション 仮想シリアル通信 ・ SCIF0 シリアル通信 ・ タイマ割り込み
ネットワークサンプルプログラム	・ ネットワーク通信 ・ SCIF0 シリアル通信 ・ タイマ割り込み

1.2 動作モード

本サンプルプログラムは、AP-SH4A-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-SH4A-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。

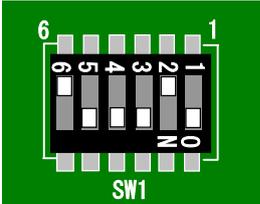
エリア 0 バス幅	: 16bit
エンディアン	: 使用するサンプルプログラムに合わせて下さい。
PCI モード	: ノーマルモード
外部 CPU インターフェース	: 外部 CPU 接続なし
FLASHROM 設定	: FLASHROM を使用する
FLASHROM ライトプロテクト設定	: ライトプロテクト解除
USB チャネル設定	: 使用するサンプルプログラムに合わせて下さい。
動作モード	: XrossFinder 使用時は DEBUG に、未使用時は NORMAL にして下さい

CPU ボードの設定を製品出荷時の状態とし、使用方法に合わせて以下の各スイッチの設定を行って下さい。

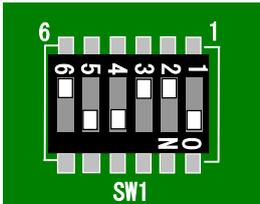
・ SW1

<SW1 設定>

エリア 0 バス幅	: 16 ビット
PCI 動作モード	: ノーマルモード
外部 CPU 接続	: 外部 CPU 接続なし



ビッグエンディアン使用時

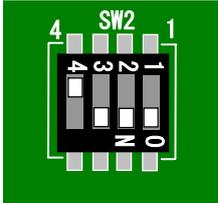


リトルエンディアン使用時

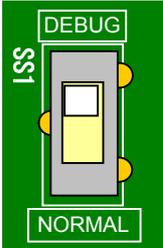
・ SW2

<SW2 設定>

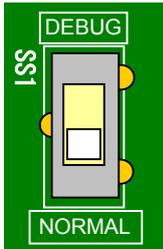
FLASHROM 設定	: FLASHROM を使用する
FLASHROM ライトプロテクト設定	: ライトプロテクト解除
USB チャンネル設定	: Host/Function 固定



・ SS1

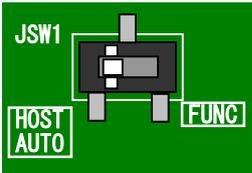


XrossFinder 使用時
DEBUG モード

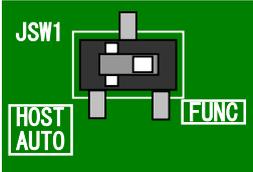


XrossFinder 未使用時
NORMAL モード

・ JSW1



USB ホスト使用時



USB ファンクション使用時

Fig1.2-1 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 High-performance Embedded Workshop を用いて開発されております。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	High-performance Embedded Workshop	V 3.01 (release 1) 以降	SHC ※1 (ルネサス エレクトロニクス)	V8.0.0.0 以降	SuperH ファミリ用 C/C++コンパイラ パッケージに付属
gcc	High-performance Embedded Workshop	V4.05.00.059 以降	GNUSH ※2 (KPIT)	V0902 以降	ELF 形式 High-performance Embedded Workshop 対応

※1： ルネサスエレクトロニクス社製「SuperH ファミリ用 C/C++コンパイラパッケージ」です。ルネサスエレクトロニクス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 High-performance Embedded Workshop 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。

1.4 ワークスペースについて

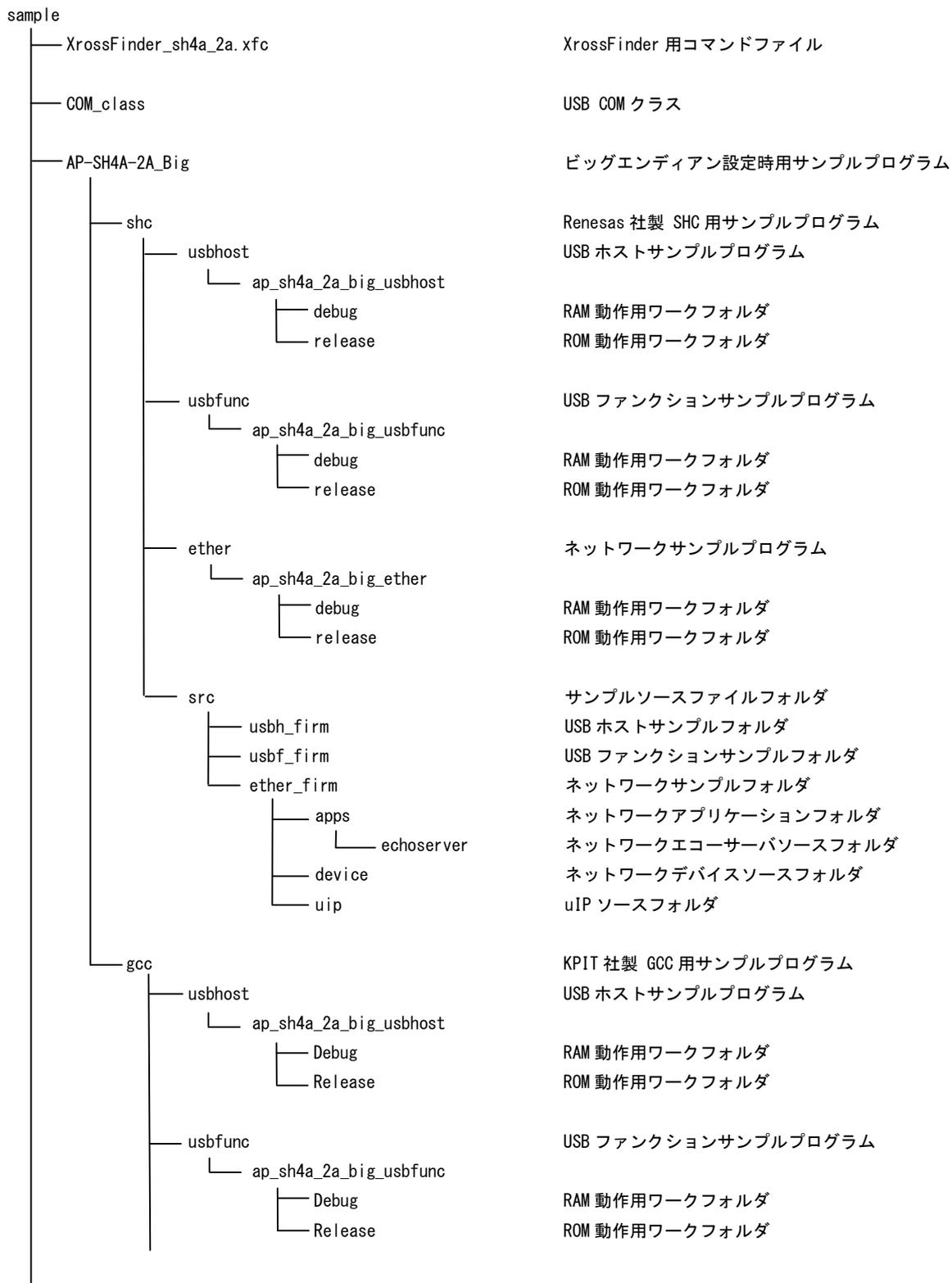
本サンプルプログラムの総合開発環境 High-performance Embedded Workshop ワークスペースは次のフォルダに格納されています。

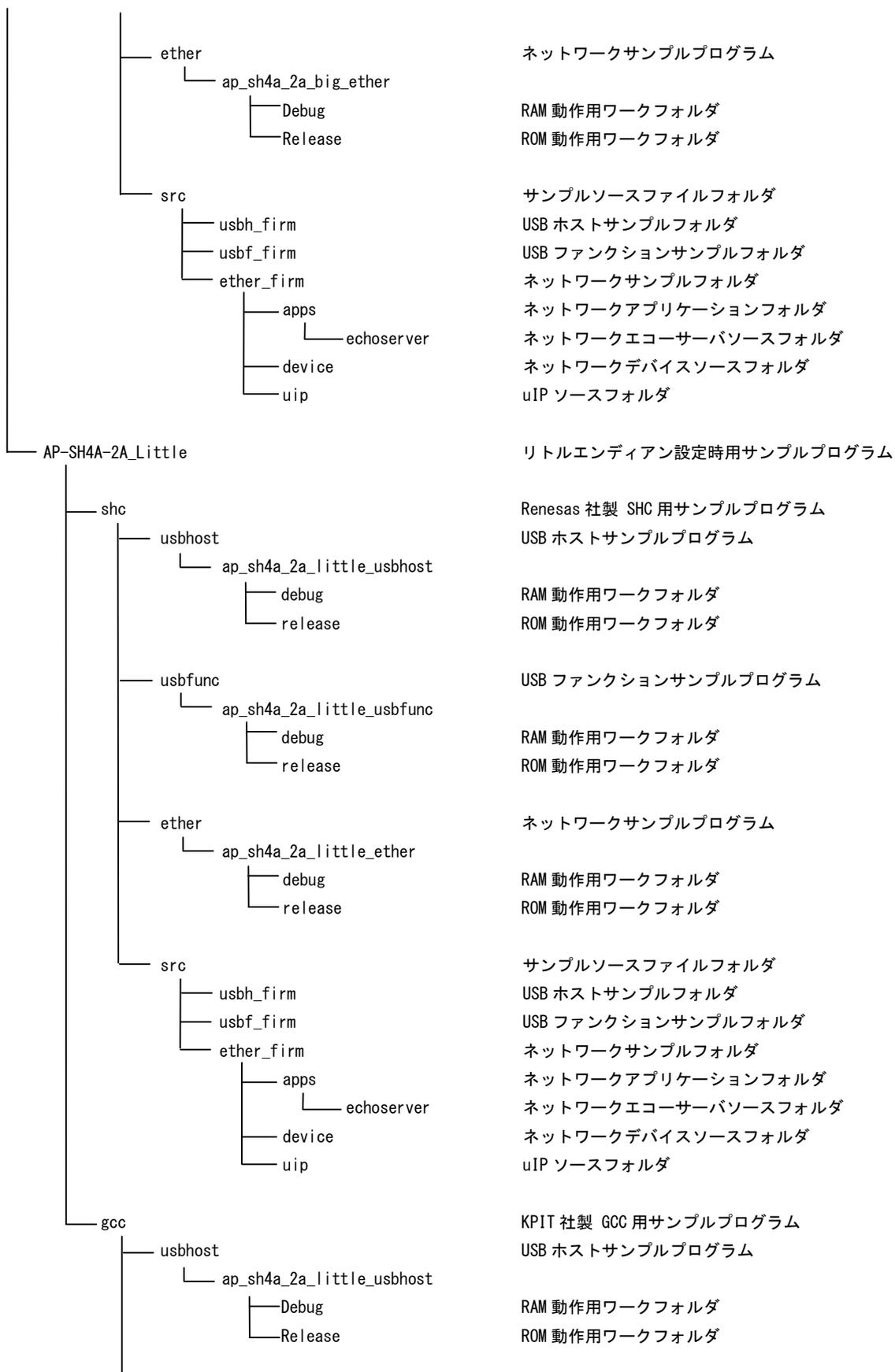
サンプルプログラム	エンディアン	コンパイラ	フォルダ
USB ホスト サンプルプログラム	ビッグ	SHC	¥sample¥AP-SH4A-2A_Big¥shc¥usbhost¥ap_sh4a_2a_big_usbhost.hws
		GCC	¥sample¥AP-SH4A-2A_Big¥gcc¥usbhost¥ap_sh4a_2a_big_usbhost.hws
	リトル	SHC	¥sample¥AP-SH4A-2A_Little¥shc¥usbhost¥ap_sh4a_2a_little_usbhost.hws
		GCC	¥sample¥AP-SH4A-2A_Little¥gcc¥usbhost¥ap_sh4a_2a_little_usbhost.hws
USB ファンクション サンプルプログラム	ビッグ	SHC	¥sample¥AP-SH4A-2A_Big¥shc¥usbfunc¥ap_sh4a_2a_big_usbfunc.hws
		GCC	¥sample¥AP-SH4A-2A_Big¥gcc¥usbfunc¥ap_sh4a_2a_big_usbfunc.hws
	リトル	SHC	¥sample¥AP-SH4A-2A_Little¥shc¥usbfunc¥ap_sh4a_2a_little_usbfunc.hws
		GCC	¥sample¥AP-SH4A-2A_Little¥gcc¥usbfunc¥ap_sh4a_2a_little_usbfunc.hws
ネットワーク サンプルプログラム	ビッグ	SHC	¥sample¥AP-SH4A-2A_Big¥shc¥ether¥ap_sh4a_2a_big_ether.hws
		GCC	¥sample¥AP-SH4A-2A_Big¥gcc¥ether¥ap_sh4a_2a_big_ether.hws
	リトル	SHC	¥sample¥AP-SH4A-2A_Little¥shc¥ether¥ap_sh4a_2a_little_ether.hws
		GCC	¥sample¥AP-SH4A-2A_Little¥gcc¥ether¥ap_sh4a_2a_little_ether.hws

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・SHC のファイル構成

ビッグエンディアン設定時は「¥sample¥AP-SH4A-2A_Big」以下にファイルが構成されます。
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>

usbhost	...	USB ホストサンプルプログラムフォルダ
usbfunc	...	USB ファンクションサンプルプログラムフォルダ
ether	...	ネットワークサンプルプログラムフォルダ
src	...	サンプルソースファイルフォルダ

<shc¥usbhost フォルダ内>

ap_sh4a_2a_big_usbhost.hws	...	USB ホスト HEW 用ワークスペースファイル
----------------------------	-----	--------------------------

<shc¥usbhost¥ap_sh4a_2a_big_usbhost フォルダ内>

ap_sh4a_2a_big_usbhost.hwp	...	USB ホスト HEW 用プロジェクトファイル
----------------------------	-----	-------------------------

<shc¥usbhost¥ap_sh4a_2a_big_usbhost¥debug フォルダ内>

ap_sh4a_2a_big_usbhost.abs	...	USB ホスト RAM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_big_usbhost.mot	...	USB ホスト RAM 動作用モトローラ S フォーマット形式 ファイル
ap_sh4a_2a_big_usbhost.map	...	USB ホスト 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

<shc¥usbhost¥ap_sh4a_2a_big_usbhost¥release フォルダ内>

ap_sh4a_2a_big_usbhost.abs	...	USB ホスト ROM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_big_usbhost.mot	...	USB ホスト ROM 動作用モトローラ S フォーマット形式 ファイル
ap_sh4a_2a_big_usbhost.map	...	USB ホスト 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

<shc¥usbfunc フォルダ内>

ap_sh4a_2a_big_usbfunc.hws	...	USB ファンクション HEW 用ワークスペースファイル
----------------------------	-----	------------------------------

<shc¥usbfunc¥ap_sh4a_2a_big_usbfunc フォルダ内>

ap_sh4a_2a_big_usbfunc.hwp	...	USB ファンクション HEW 用プロジェクトファイル
----------------------------	-----	-----------------------------

<shc¥usbfunc¥ap_sh4a_2a_big_usbfunc¥debug フォルダ内>

ap_sh4a_2a_big_usbfunc.abs	...	USB ファンクション RAM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_big_usbfunc.mot	...	USB ファンクション RAM 動作用モトローラ S フォーマット 形式ファイル
ap_sh4a_2a_big_usbfunc.map	...	USB ファンクション 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

次ページへ (ビッグエンディアン・SHC)

前ページより (ビッグエンディアン・SHC)

<shc%usbfunc%ap_sh4a_2a_big_usbfunc%release フォルダ内>

ap_sh4a_2a_big_usbfunc. abs	...	USB ファンクション ROM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_big_usbfunc. mot	...	USB ファンクション ROM 動作用モトローラ S フォーマット形式ファイル
ap_sh4a_2a_big_usbfunc. map	...	USB ファンクション 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

<shc%ether フォルダ内>

ap_sh4a_2a_big_ether. hws	...	ネットワーク HEW 用ワークスペースファイル
---------------------------	-----	-------------------------

<shc%ether%ap_sh4a_2a_big_ether フォルダ内>

ap_sh4a_2a_big_ether. hwp	...	ネットワーク HEW 用プロジェクトファイル
---------------------------	-----	------------------------

<shc%ether%ap_sh4a_2a_big_ether%debug フォルダ内>

ap_sh4a_2a_big_ether. abs	...	ネットワーク RAM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_big_ether. mot	...	ネットワーク RAM 動作用モトローラ S フォーマット形式ファイル
ap_sh4a_2a_big_ether. map	...	ネットワーク 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

<shc%ether%ap_sh4a_2a_big_ether%release フォルダ内>

ap_sh4a_2a_big_ether. abs	...	ネットワーク ROM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_big_ether. mot	...	ネットワーク ROM 動作用モトローラ S フォーマット形式ファイル
ap_sh4a_2a_big_ether. map	...	ネットワーク 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

<shc%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
7763. h	...	SH7763 内部レジスタ定義ヘッダファイル
vector. c	...	割込ベクタテーブル
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

<shc%src%usbh_firm フォルダ内>

Usbh. c	...	USB ホスト処理
Usbh. h	...	USB ホストヘッダファイル

次ページへ (ビッグエンディアン・SHC)

前ページより (ビッグエンディアン・SHC)

<shc¥src¥usbf_firm 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
DoRequestVenderCommand.c	...	ベンダーコマンド実行処理
DoSerial.c	...	シリアル転送処理
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
SysMemMap.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
SH7763.h	...	SH7763 USB ファンクションレジスタ定義ヘッダファイル

<shc¥src¥ether_firm フォルダ内>

apps	...	ネットワークアプリケーションフォルダ
device	...	ネットワークデバイスソースフォルダ
uip	...	uIP ソースフォルダ
ether_main.c	...	ネットワークメイン処理

<shc¥src¥ether_firm¥apps フォルダ内>

echoserver	...	ネットワークエコーサーバソースフォルダ
------------	-----	---------------------

2.2.2 ビッグエンディアン・GCC のファイル構成

ビッグエンディアン設定時は「¥sample¥AP-SH4A-2A_Big」以下にファイルが構成されます。

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>

usbhost	…	USB ホストサンプルプログラムフォルダ
usbfunc	…	USB ファンクションサンプルプログラムフォルダ
ether	…	ネットワークサンプルプログラムフォルダ
src	…	サンプルソースファイルフォルダ

<gcc¥usbhost フォルダ内>

ap_sh4a_2a_big_usbhost.hws	…	USB ホスト HEW 用ワークスペースファイル
----------------------------	---	--------------------------

<gcc¥usbhost¥ap_sh4a_2a_big_usbhost フォルダ内>

ap_sh4a_2a_big_usbhost.hwp	…	USB ホスト HEW 用プロジェクトファイル
----------------------------	---	-------------------------

<gcc¥usbhost¥ap_sh4a_2a_big_usbhost¥Debug フォルダ内>

ap_sh4a_2a_big_usbhost.abs	…	USB ホスト RAM 動作用オブジェクトファイル(elf 形式)
ap_sh4a_2a_big_usbhost.mot	…	USB ホスト RAM 動作用モトローラ S フォーマット形式 ファイル
ap_sh4a_2a_big_usbhost.map	…	USB ホスト RAM 動作用マップファイル コンパイル後は、.o,.a 等のファイルが生成されます

<gcc¥usbhost¥ap_sh4a_2a_big_usbhost¥Release フォルダ内>

ap_sh4a_2a_big_usbhost.abs	…	USB ホスト ROM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_big_usbhost.mot	…	USB ホスト ROM 動作用モトローラ S フォーマット形式 ファイル
ap_sh4a_2a_big_usbhost.map	…	USB ホスト ROM 動作用マップファイル コンパイル後は、.o,.a 等のファイルが生成されます

<gcc¥usbfunc フォルダ内>

ap_sh4a_2a_big_usbfunc.hws	…	USB ファンクション HEW 用ワークスペースファイル
----------------------------	---	------------------------------

<gcc¥usbfunc¥ap_sh4a_2a_big_usbfunc フォルダ内>

ap_sh4a_2a_big_usbfunc.hwp	…	USB ファンクション HEW 用プロジェクトファイル
----------------------------	---	-----------------------------

<gcc¥usbfunc¥ap_sh4a_2a_big_usbfunc¥Debug フォルダ内>

ap_sh4a_2a_big_usbfunc.abs	…	USB ファンクション RAM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_big_usbfunc.mot	…	USB ファンクション RAM 動作用モトローラ S フォーマット 形式ファイル
ap_sh4a_2a_big_usbfunc.map	…	USB ファンクション RAM 動作用マップファイル コンパイル後は、.o,.a 等のファイルが生成されます

次ページへ (ビッグエンディアン・GCC)

前ページより (ビッグエンディアン・GCC)

<gcc¥usbfunc¥ap_sh4a_2a_big_usbfunc¥Release フォルダ内>

ap_sh4a_2a_big_usbfunc.abs … USB ファンクション ROM 動作用オブジェクトファイル (elf 形式)
 ap_sh4a_2a_big_usbfunc.mot … USB ファンクション ROM 動作用モトローラ S フォーマット形式ファイル
 ap_sh4a_2a_big_usbfunc.map … USB ファンクション ROM 動作用マップファイル
 コンパイル後は、.o,.a 等のファイルが生成されます

<gcc¥ether フォルダ内>

ap_sh4a_2a_big_ether.hws … ネットワーク HEW 用ワークスペースファイル

<gcc¥ether¥ap_sh4a_2a_big_ether フォルダ内>

ap_sh4a_2a_big_ether.hwp … ネットワーク HEW 用プロジェクトファイル

<gcc¥ether¥ap_sh4a_2a_big_ether¥Debug フォルダ内>

ap_sh4a_2a_big_ether.abs … ネットワーク RAM 動作用オブジェクトファイル(elf 形式)
 ap_sh4a_2a_big_ether.mot … ネットワーク RAM 動作用モトローラ S フォーマット形式ファイル
 ap_sh4a_2a_big_ether.map … ネットワーク RAM 動作用マップファイル
 コンパイル後は、.o,.a 等のファイルが生成されます

<gcc¥ether¥ap_sh4a_2a_big_ether¥Release フォルダ内>

ap_sh4a_2a_big_ether.abs … ネットワーク ROM 動作用オブジェクトファイル (elf 形式)
 ap_sh4a_2a_big_ether.mot … ネットワーク ROM 動作用モトローラ S フォーマット形式ファイル
 ap_sh4a_2a_big_ether.map … ネットワーク ROM 動作用マップファイル
 コンパイル後は、.o,.a 等のファイルが生成されます

<gcc¥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
 7763.h … SH7763 内部レジスタ定義ヘッダファイル
 vector.c … 割込ベクタテーブル
 crt0.s … スタートアップ処理
 cpu.h … CPU 定義ファイル

<gcc¥src¥usbh_firm フォルダ内>

Usbh.c … USB ホスト処理
 Usbh.h … USB ホストヘッダファイル

次ページへ (ビッグエンディアン・GCC)

前ページより (ビッグエンディアン・GCC)

<gcc¥src¥usbf_firm 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
DoRequestVenderCommand.c	...	ベンダーコマンド実行処理
DoSerial.c	...	シリアル転送処理
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
SysMemMap.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
SH7763.h	...	SH7763 USB ファンクションレジスタ定義ヘッダファイル

<gcc¥src¥ether_firm フォルダ内>

apps	...	ネットワークアプリケーションフォルダ
device	...	ネットワークデバイスソースフォルダ
uip	...	uIP ソースフォルダ
ether_main.c	...	ネットワークメイン処理

<gcc¥src¥ether_firm¥apps フォルダ内>

echoserver	...	ネットワークエコーサーバソースフォルダ
------------	-----	---------------------

2.2.3 リトルエンディアン・SHC のファイル構成

リトルエンディアン設定時は「¥sample¥AP-SH4A-2A_Little」以下にファイルが構成されます。

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>

usbhost	…	USB ホストサンプルプログラムフォルダ
usbfunc	…	USB ファンクションサンプルプログラムフォルダ
ether	…	ネットワークサンプルプログラムフォルダ
src	…	サンプルソースファイルフォルダ

<shc¥usbhost フォルダ内>

ap_sh4a_2a_little_usbhost.hws	…	USB ホスト HEW 用ワークスペースファイル
-------------------------------	---	--------------------------

<shc¥usbhost¥ap_sh4a_2a_little_usbhost フォルダ内>

ap_sh4a_2a_little_usbhost.hwp	…	USB ホスト HEW 用プロジェクトファイル
-------------------------------	---	-------------------------

<shc¥usbhost¥ap_sh4a_2a_little_usbhost¥debug フォルダ内>

ap_sh4a_2a_little_usbhost.abs	…	USB ホスト RAM 動作用オブジェクトファイル(elf 形式)
ap_sh4a_2a_little_usbhost.mot	…	USB ホスト RAM 動作用モトローラ S フォーマット形式 ファイル
ap_sh4a_2a_little_usbhost.map	…	USB ホスト 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

<shc¥usbhost¥ap_sh4a_2a_little_usbhost¥release フォルダ内>

ap_sh4a_2a_little_usbhost.abs	…	USB ホスト ROM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_little_usbhost.mot	…	USB ホスト ROM 動作用モトローラ S フォーマット形式 ファイル
ap_sh4a_2a_little_usbhost.map	…	USB ホスト 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

<shc¥usbfunc フォルダ内>

ap_sh4a_2a_little_usbfunc.hws	…	USB ファンクション HEW 用ワークスペースファイル
-------------------------------	---	------------------------------

<shc¥usbfunc¥ap_sh4a_2a_little_usbfunc フォルダ内>

ap_sh4a_2a_little_usbfunc.hwp	…	USB ファンクション HEW 用プロジェクトファイル
-------------------------------	---	-----------------------------

<shc¥usbfunc¥ap_sh4a_2a_little_usbfunc¥debug フォルダ内>

ap_sh4a_2a_little_usbfunc.abs	…	USB ファンクション RAM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_little_usbfunc.mot	…	USB ファンクション RAM 動作用モトローラ S フォーマット 形式ファイル
ap_sh4a_2a_little_usbfunc.map	…	USB ファンクション 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

次ページへ (リトルエンディアン・SHC)

前ページより (リトルエンディアン・SHC)

<shc%usbfunc%ap_sh4a_2a_little_usbfunc%release フォルダ内>

ap_sh4a_2a_little_usbfunc.abs	...	USB ファンクション ROM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_little_usbfunc.mot	...	USB ファンクション ROM 動作用モトローラ S フォーマット 形式ファイル
ap_sh4a_2a_little_usbfunc.map	...	USB ファンクション 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

<shc%ether フォルダ内>

ap_sh4a_2a_little_ether.hws	...	ネットワーク HEW 用ワークスペースファイル
-----------------------------	-----	-------------------------

<shc%ether%ap_sh4a_2a_little_ether フォルダ内>

ap_sh4a_2a_little_ether.hwp	...	ネットワーク HEW 用プロジェクトファイル
-----------------------------	-----	------------------------

<shc%ether%ap_sh4a_2a_little_ether%debug フォルダ内>

ap_sh4a_2a_little_ether.abs	...	ネットワーク RAM 動作用オブジェクトファイル(elf 形式)
ap_sh4a_2a_little_ether.mot	...	ネットワーク RAM 動作用モトローラ S フォーマット形式 ファイル
ap_sh4a_2a_little_ether.map	...	ネットワーク 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

<shc%ether%ap_sh4a_2a_little_ether%release フォルダ内>

ap_sh4a_2a_little_ether.abs	...	ネットワーク ROM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_little_ether.mot	...	ネットワーク ROM 動作用モトローラ S フォーマット形式 ファイル
ap_sh4a_2a_little_ether.map	...	ネットワーク 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

<shc%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
7763.h	...	SH7763 内部レジスタ定義ヘッダファイル
vector.c	...	割込ベクタテーブル
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

<shc%src%usbh_firm フォルダ内>

Usbh.c	...	USB ホスト処理
Usbh.h	...	USB ホストヘッダファイル

次ページへ (リトルエンディアン・SHC)

前ページより (リトルエンディアン・SHC)

<shc¥src¥usbf_firm 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
DoRequestVenderCommand.c	...	ベンダーコマンド実行処理
DoSerial.c	...	シリアル転送処理
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
SysMemMap.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
SH7763.h	...	SH7763 USB ファンクションレジスタ定義ヘッダファイル

<shc¥src¥ether_firm フォルダ内>

apps	...	ネットワークアプリケーションフォルダ
device	...	ネットワークデバイスソースフォルダ
uip	...	uIP ソースフォルダ
ether_main.c	...	ネットワークメイン処理

<shc¥src¥ether_firm¥apps フォルダ内>

echoserver	...	ネットワークエコーサーバソースフォルダ
------------	-----	---------------------

2.2.4 リトルエンディアン・GCC のファイル構成

リトルエンディアン設定時は「¥sample¥AP-SH4A-2A_Little」以下にファイルが構成されます。

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>

usbhost	…	USB ホストサンプルプログラムフォルダ
usbfunc	…	USB ファンクションサンプルプログラムフォルダ
ether	…	ネットワークサンプルプログラムフォルダ
src	…	サンプルソースファイルフォルダ

<gcc¥usbhost フォルダ内>

ap_sh4a_2a_little_usbhost.hws	…	USB ホスト HEW 用ワークスペースファイル
-------------------------------	---	--------------------------

<gcc¥usbhost¥ap_sh4a_2a_little_usbhost フォルダ内>

ap_sh4a_2a_little_usbhost.hwp	…	USB ホスト HEW 用プロジェクトファイル
-------------------------------	---	-------------------------

<gcc¥usbhost¥ap_sh4a_2a_little_usbhost¥Debug フォルダ内>

ap_sh4a_2a_little_usbhost.abs	…	USB ホスト RAM 動作用オブジェクトファイル(elf 形式)
ap_sh4a_2a_little_usbhost.mot	…	USB ホスト RAM 動作用モトローラ S フォーマット形式 ファイル
ap_sh4a_2a_little_usbhost.map	…	USB ホスト RAM 動作用マップファイル コンパイル後は、.o..a 等のファイルが生成されます

<gcc¥usbhost¥ap_sh4a_2a_little_usbhost¥Release フォルダ内>

ap_sh4a_2a_little_usbhost.abs	…	USB ホスト ROM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_little_usbhost.mot	…	USB ホスト ROM 動作用モトローラ S フォーマット形式 ファイル
ap_sh4a_2a_little_usbhost.map	…	USB ホスト ROM 動作用マップファイル コンパイル後は、.o..a 等のファイルが生成されます

<gcc¥usbfunc フォルダ内>

ap_sh4a_2a_little_usbfunc.hws	…	USB ファンクション HEW 用ワークスペースファイル
-------------------------------	---	------------------------------

<gcc¥usbfunc¥ap_sh4a_2a_little_usbfunc フォルダ内>

ap_sh4a_2a_little_usbfunc.hwp	…	USB ファンクション HEW 用プロジェクトファイル
-------------------------------	---	-----------------------------

<gcc¥usbfunc¥ap_sh4a_2a_little_usbfunc¥Debug フォルダ内>

ap_sh4a_2a_little_usbfunc.abs	…	USB ファンクション RAM 動作用オブジェクトファイル (elf 形式)
ap_sh4a_2a_little_usbfunc.mot	…	USB ファンクション RAM 動作用モトローラ S フォーマット 形式ファイル
ap_sh4a_2a_little_usbfunc.map	…	USB ファンクション RAM 動作用マップファイル コンパイル後は、.o..a 等のファイルが生成されます

次ページへ (リトルエンディアン・GCC)

前ページより (リトルエンディアン・GCC)

<gcc¥usbfunc¥ap_sh4a_2a_little_usbfunc¥Release フォルダ内>

ap_sh4a_2a_little_usbfunc.abs	...	USB ファンクション ROM 動作オブジェクトファイル (elf 形式)
ap_sh4a_2a_little_usbfunc.mot	...	USB ファンクション ROM 動作モトローラ S フォーマット形式ファイル
ap_sh4a_2a_little_usbfunc.map	...	USB ファンクション ROM 動作マップファイル コンパイル後は、.o,.a 等のファイルが生成されます

<gcc¥ether フォルダ内>

ap_sh4a_2a_little_ether.hws	...	ネットワーク HEW 用ワークスペースファイル
-----------------------------	-----	-------------------------

<gcc¥ether¥ap_sh4a_2a_little_ether フォルダ内>

ap_sh4a_2a_little_ether.hwp	...	ネットワーク HEW 用プロジェクトファイル
-----------------------------	-----	------------------------

<gcc¥ether¥ap_sh4a_2a_little_ether¥Debug フォルダ内>

ap_sh4a_2a_little_ether.abs	...	ネットワーク RAM 動作オブジェクトファイル (elf 形式)
ap_sh4a_2a_little_ether.mot	...	ネットワーク RAM 動作モトローラ S フォーマット形式ファイル
ap_sh4a_2a_little_ether.map	...	ネットワーク RAM 動作マップファイル コンパイル後は、.o,.a 等のファイルが生成されます

<gcc¥ether¥ap_sh4a_2a_little_ether¥Release フォルダ内>

ap_sh4a_2a_little_ether.abs	...	ネットワーク ROM 動作オブジェクトファイル (elf 形式)
ap_sh4a_2a_little_ether.mot	...	ネットワーク ROM 動作モトローラ S フォーマット形式ファイル
ap_sh4a_2a_little_ether.map	...	ネットワーク ROM 動作マップファイル コンパイル後は、.o,.a 等のファイルが生成されます

<gcc¥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
7763.h	...	SH7763 内部レジスタ定義ヘッダファイル
vector.c	...	割込ベクタテーブル
crt0.s	...	スタートアップ処理
cpu.h	...	CPU 定義ファイル

<gcc¥src¥usbh_firm フォルダ内>

Usbh.c	...	USB ホスト処理
Usbh.h	...	USB ホストヘッダファイル

次ページへ (リトルエンディアン・GCC)

前ページより (リトルエンディアン・GCC)

<gcc¥src¥usbf_firm 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
DoRequestVenderCommand.c	...	ベンダーコマンド実行処理
DoSerial.c	...	シリアル転送処理
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
SysMemMap.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
SH7763.h	...	SH7763 USB ファンクションレジスタ定義ヘッダファイル

<gcc¥src¥ether_firm フォルダ内>

apps	...	ネットワークアプリケーションフォルダ
device	...	ネットワークデバイスソースフォルダ
uip	...	uIP ソースフォルダ
ether_main.c	...	ネットワークメイン処理

<gcc¥src¥ether_firm¥apps フォルダ内>

echoserver	...	ネットワークエコーサーバソースフォルダ
------------	-----	---------------------

3. USB ホストサンプルプログラム

3.1 ビルド・デバッグ方法（USB ホスト）

注意：本項ではビッグエンディアンでのビルド・デバッグ方法を記載しています。リトルエンディアンで 사용되는場合は、「Big」を「Little」、「big」を「little」、「ビッグ」を「リトル」に読み替えて作業を行ってください。

(1) ビルド

Renesas 社製 SHC

- ① HEW を起動し、¥sample¥AP-SH4A-2A-Big¥shc¥usbhost¥ap_sh4a_2a_big_usbhost.hws を読み込みます。
- ② 最初の読み込みを行ったときに、「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_sh4a_2a_big_usbhost.mot、ap_sh4a_2a_big_usbhost.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。

KPIT 社製 GCC

- ① HEW を起動し、¥sample¥AP-SH4A-2A-Big¥gcc¥usbhost¥ap_sh4a_2a_big_usbhost.hws を読み込みます。
- ② 最初の読み込みを行ったときに、「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「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。
- ④ 最初の読み込みを行ったときに、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。
- ⑤ [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。
- ⑥ メニューの [Build] - [Build] を実行して下さい。ap_sh4a_2a_big_usbhost.mot、ap_sh4a_2a_big_usbhost.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-SH4A-2A のスイッチを、「1.2 動作モード」 「Fig1.2-1 動作モード設定」を参考に設定します。なお、SW1 はビッグエンディアン使用時、SS1 は DEBUG モード、JSW1 は USB ホスト使用時に設定して下さい。
- ② XrossFinder を起動し、¥sample フォルダ直下にある XrossFinder_sh4a_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_sh4a_2a_big_usbhost.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-SH4A-2A のスイッチを、「1.2 動作モード」 「Fig1.2-1 動作モード設定」を参考に設定します。なお、SW1 はビッグエンディアン使用時、SW2 は FLASHROM を使用する・ライトプロテクト解除、SS1 は DEBUG モード、JSW1 は USB ホスト使用時に設定して下さい。
- ② ¥release フォルダ内の ap_sh4a_2a_big_usbhost.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。

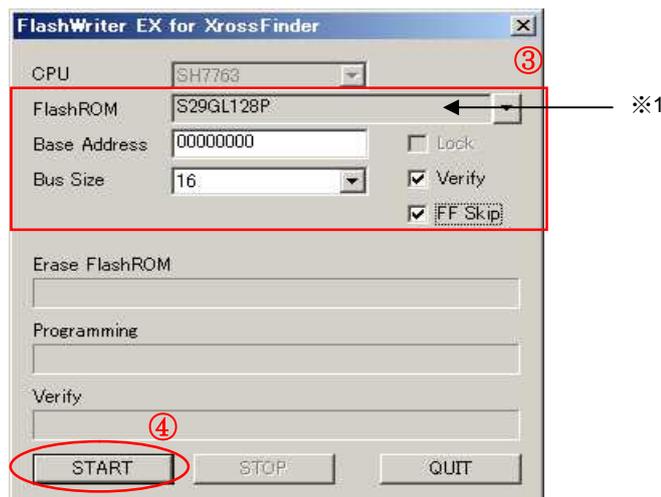


Fig3.1-1 FlashWriter EX for XrossFinder の設定

(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 AP-SH4A-2A のスイッチを、「1.2 動作モード」 「Fig1. 2-1 動作モード設定」を参考に設定します。SW1 はビッグエンディアン使用時、SW2 は FLASHROM を使用する・ライトプロテクト解除、SS1 は DEBUG モード、JSW1 は USB ホスト使用時に設定して下さい。
- ③ FlashWriterEX を起動して、「Table3. 1-1 FlashWriterEX の設定」を参考に設定を行って下さい。
- ④ FlashWriterEX で、¥sample フォルダ直下にある XrossFinder_sh4a_2a.xfc コマンドファイルを使用するように設定して下さい。
- ⑤ ¥release フォルダ内の ap_sh4a_2a_big_usbhost.mot をボードに書き込みます。
- ⑥ AP-SH4A-2A のスイッチを、「1.2 動作モード」 「Fig1. 2-1 動作モード設定」を参考に設定します。なお、SW1 はビッグエンディアン使用時、SW2 は FLASHROM を使用する、SS1 は NORMAL モード、JSW1 は USB ホスト使用時に設定して下さい。
- ⑦ 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」 HJ-LINK 使用時は「HJ-LINK」
JTAG クロック (XrossFinder 使用時のみ)	10MHz 以下
CPU	SH7763
BaseAddress	00000000
FLASHROM	S29GL128P ※1
Bus Size	16

Table3.1-1 FlashWriter EX の設定

※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下さい。

3.2 動作説明 (USB ホスト)

3.2.1 サンプルプログラム概要 (USB ホスト)

USB ホストサンプルプログラムは、下記の動作を行います。

- SCIF0 でエコーバックを行います。(送受信割り込み使用)
SCIF0 から受信した値をそのまま、SCIF0 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、フロー制御なしです。
動作確認は、パソコン上のターミナルソフト (ハイパーターミナルなど) 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 1sec 間隔で ON/OFF します。(TMU1 割り込み使用)
 - USB ホストに USB メモリを接続すると、SCIF0 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、SCIF0 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
- ※1 USB ホスト動作の詳細は、「3.2.2 USB ホスト動作」を参照してください。

3.2.2 USB ホスト動作

(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 RS232C コンバータ (PC-RS-04 等) を介してシリアルケーブルで接続します。
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。
- ② 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 (CN8) 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 ※1
Transfer End !!
```

※1. 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。

```
DisConnected !!
```

- ⑦ 以上で USB メモリ接続時の動作は終了です。

(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 RS232C コンバータ (PC-RS-04 等) を介して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 (CN8) 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。

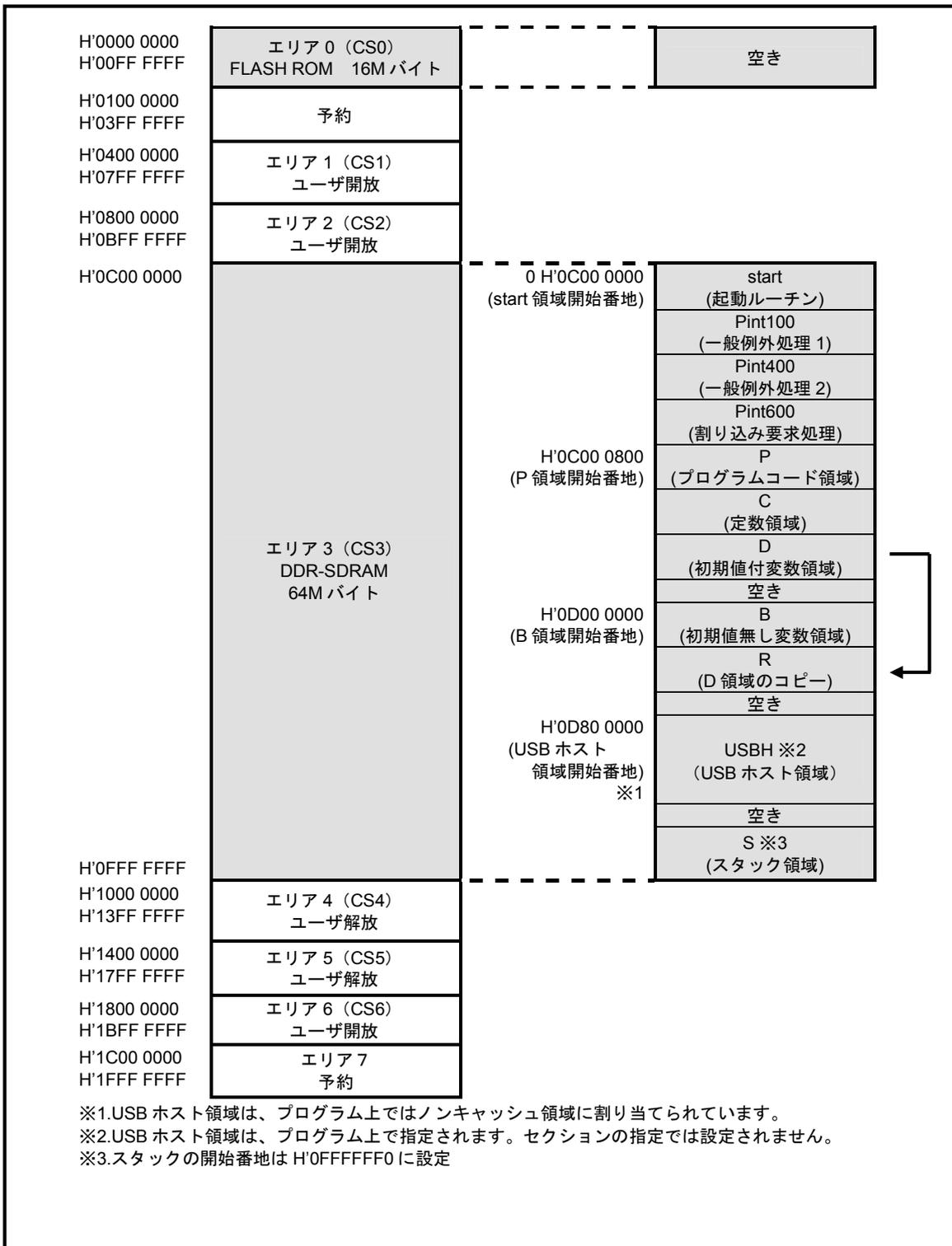
```
DisConnected !!
```

- ⑧ 以上で USB マウス接続時の動作は終了です。

3.3 RAM 動作時のメモリマップ (USB ホスト)

メモリマップを以下に示します。

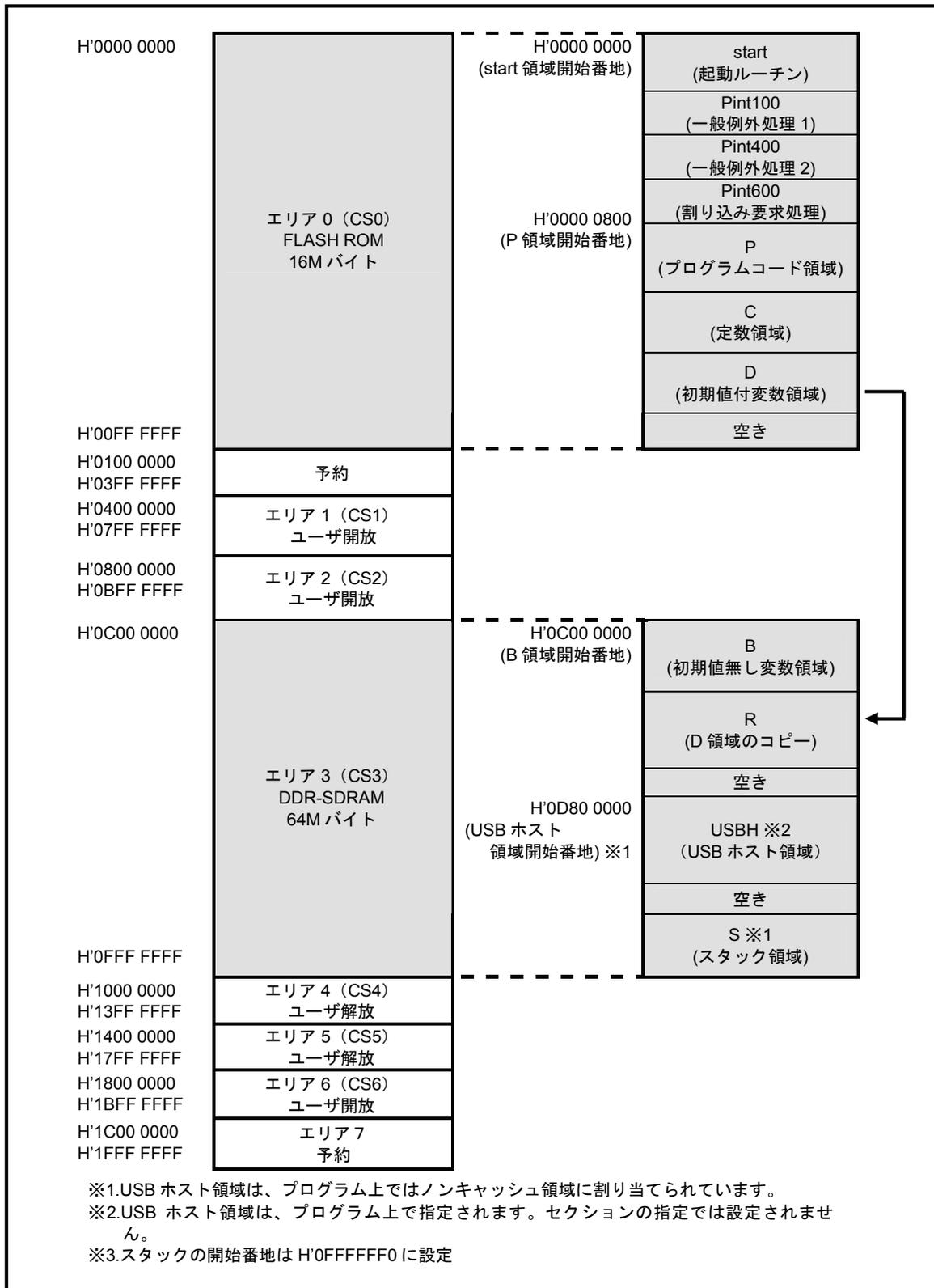
shc 版、gcc 版共にプログラム領域の割り当ては同じです。(図に示す領域は shc 版の名称を使用しています。)



3.4 ROM 動作時のメモリマップ (USB ホスト)

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. USB ファンクションサンプルプログラム

4.1 ビルド・デバッグ方法（USB ファンクション）

注意：本項ではビッグエンディアンでのビルド・デバッグ方法を記載しています。リトルエンディアンで 사용되는場合は、「Big」を「Little」、「big」を「little」、「ビッグ」を「リトル」に読み替えて作業を行ってください。

(1) ビルド

Renesas 社製 SHC

- ① HEW を起動し、¥sample¥AP-SH4A-2A-Big¥shc¥usbfunc¥ap_sh4a_2a_big_usbfunc. hws を読み込みます。
- ② 最初の読み込みを行ったときに、「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_sh4a_2a_big_usbfunc. mot、ap_sh4a_2a_big_usbfunc. 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。

KPIT 社製 GCC

- ① HEW を起動し、¥sample¥AP-SH4A-2A-Big¥gcc¥usbfunc¥ap_sh4a_2a_big_usbfunc. hws を読み込みます。
- ② 最初の読み込みを行ったときに、「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「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。
- ④ 最初の読み込みを行ったときに、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。
- ⑤ [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。
- ⑥ メニューの [Build] - [Build] を実行して下さい。ap_sh4a_2a_big_usbfunc. mot、ap_sh4a_2a_big_usbfunc. 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-SH4A-2A のスイッチを、「1.2 動作モード」 「Fig1. 2-1 動作モード設定」を参考に設定します。なお、SW1 はビッグエンディアン使用時、SS1 は DEBUG モード、JSW1 は USB ファンクション使用時に設定して下さい。
- ② XrossFinder を起動し、¥sample フォルダ直下にある XrossFinder_sh4a_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_sh4a_2a_big_usbfunc. 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-SH4A-2A のスイッチを、「1.2 動作モード」 「Fig1. 2-1 動作モード設定」を参考に設定します。なお、SW1 はビッグエンディアン使用時、SW2 は FLASHROM を使用する・ライトプロテクト解除、SS1 は DEBUG モード、JSW1 は USB ファンクション使用時に設定して下さい。
- ② ¥release フォルダ内の ap_sh4a_2a_big_usbfunc. 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 Fig4. 1-2 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。

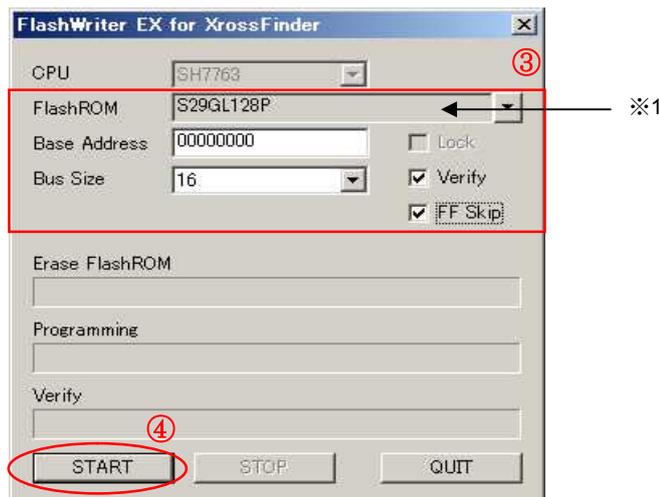


Fig4.1-2 FlashWriter EX for XrossFinder の設定

(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 AP-SH4A-2A のスイッチを、「1.2 動作モード」 「Fig1. 2-1 動作モード設定」を参考に設定します。SW1 はビッグエンディアン使用時、SW2 は FLASHROM を使用する・ライトプロテクト解除、SS1 は DEBUG モード、JSW1 は USB ファンクション使用時に設定して下さい。
- ③ FlashWriterEX を起動して、「Table4. 1-2 FlashWriterEX の設定」を参考に設定を行って下さい。
- ④ FlashWriterEX で、¥sample フォルダ直下にある XrossFinder_sh4a_2a. xfc コマンドファイルを使用するように設定して下さい。
- ⑤ ¥release フォルダ内の ap_sh4a_2a_big_usbfunc. mot をボードに書き込みます。
- ⑥ AP-SH4A-2A のスイッチを、「1.2 動作モード」 「Fig1. 2-1 動作モード設定」を参考に設定します。なお、SW1 はビッグエンディアン使用時、SW2 は FLASHROM を使用する、SS1 は NORMAL モード、JSW1 は USB ファンクション使用時に設定して下さい。
- ⑦ 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」 HJ-LINK 使用時は「HJ-LINK」
JTAG クロック (XrossFinder 使用時のみ)	10MHz 以下
CPU	SH7763
BaseAddress	00000000
FLASHROM	S29GL128P ※1
Bus Size	16

Table4.1-2 FlashWriter EX の設定

※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下さい。

4.2 動作説明 (USB ファンクション)

4.2.1 サンプルプログラム概要 (USB ファンクション)

USB ファンクションサンプルプログラムは、下記の動作を行います。

- SCIF0 でエコーバックを行います。(送受信割り込み使用)
SCIF0 から受信した値をそのまま、SCIF0 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、フロー制御なしです。
動作確認は、パソコン上のターミナルソフト (ハイパーターミナルなど) 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 1sec 間隔で ON/OFF します。(TMU1 割り込み使用)
 - 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。 ※1
- ※1. USB ファンクション動作の詳細は、「4.2.2 USB ファンクション動作」を参照してください。

4.2.2 USB ファンクション動作

以下の手順に従い、USB シリアル動作を確認してください。

USB ファンクションの動作確認は、あらかじめ USB 仮想シリアルドライバを PC にインストールしておく必要があります。

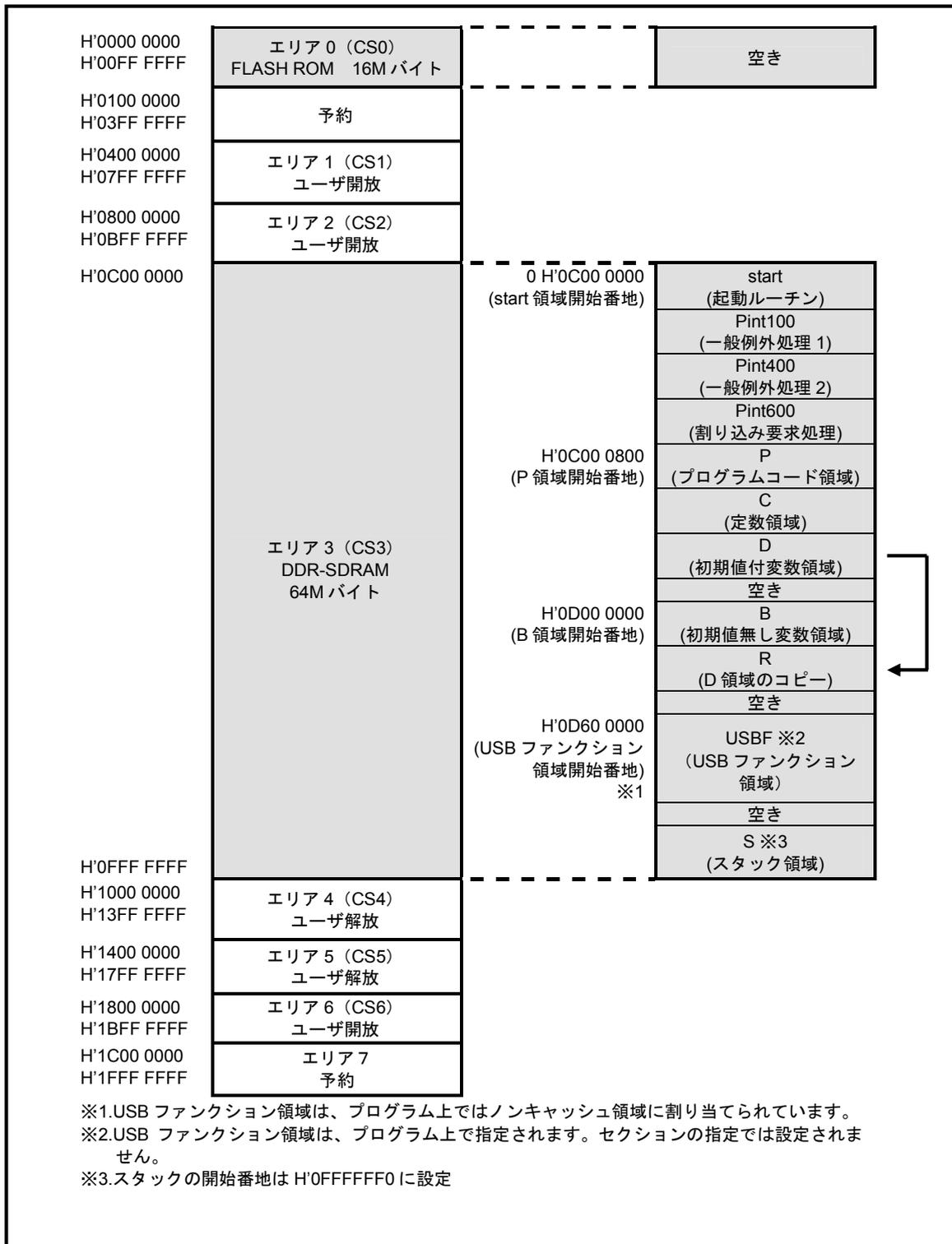
USB 仮想シリアルドライバのインストール方法につきましては、「AN178 USB 仮想シリアルドライバ インストールガイド」を参照してください。

- 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 (CN9) 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(ハイパーターミナルなど)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「AN178 USB 仮想シリアルドライバ 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、フロー制御なしです。
- 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 シリアル動作は終了です。

4.3 RAM 動作時のメモリマップ (USB ファンクション)

メモリマップを以下に示します。

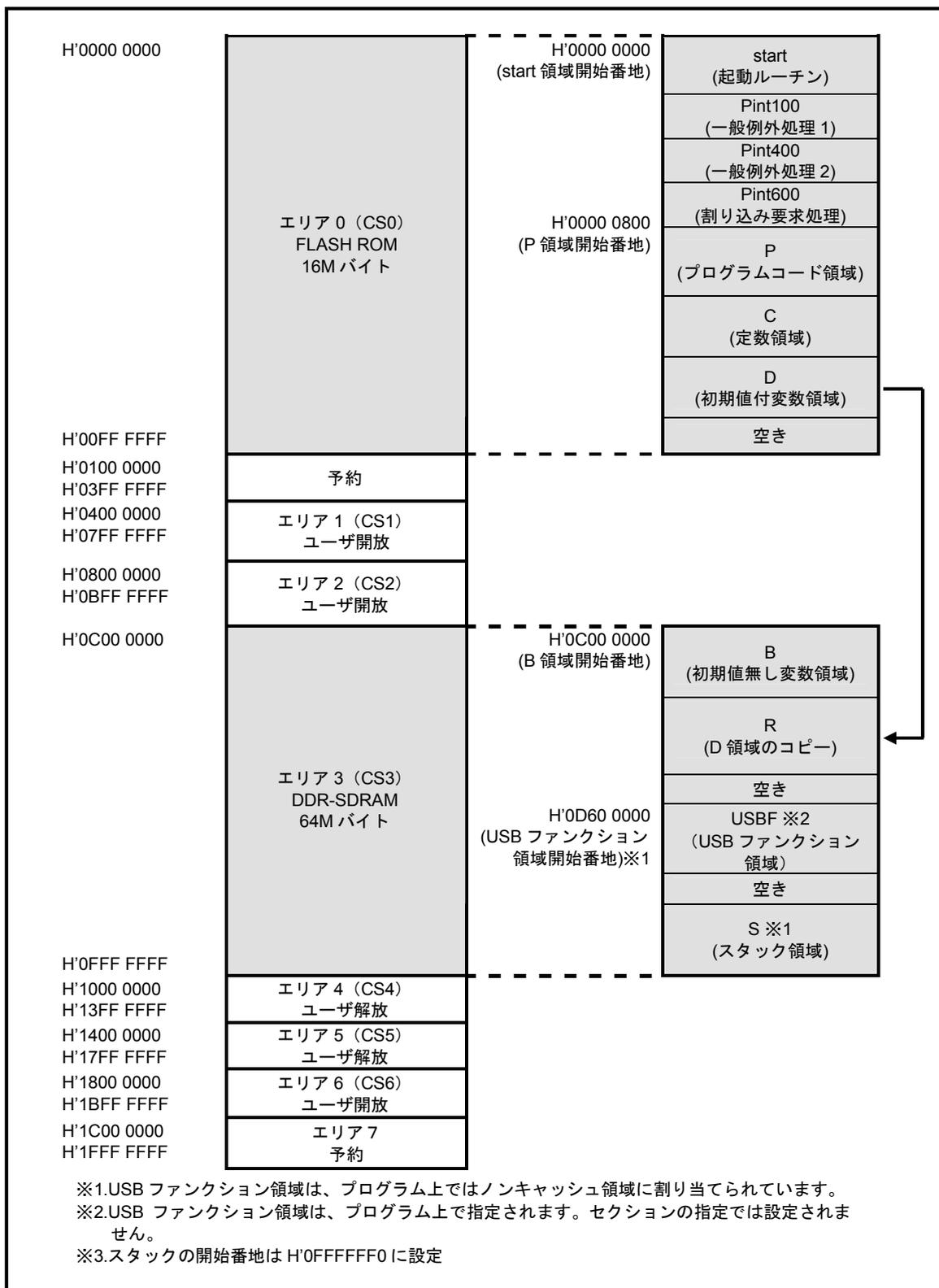
shc 版、gcc 版共にプログラム領域の割り当ては同じです。(図に示す領域は shc 版の名称を使用しています。)



4.4 ROM 動作時のメモリマップ (USB ファンクション)

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。)



5. ネットワークサンプルプログラム

5.1 ビルド・デバッグ方法（ネットワーク）

注意：本項ではビッグエンディアンでのビルド・デバッグ方法を記載しています。リトルエンディアンで 사용되는場合は、「Big」を「Little」、「big」を「little」、「ビッグ」を「リトル」に読み替えて作業を行ってください。

(1) ビルド

Renesas 社製 SHC

- ① HEW を起動し、¥sample¥AP-SH4A-2A_Big¥shc¥ether¥ap_sh4a_2a_big_ether.hws を読み込みます。
- ② 最初の読み込みを行ったときに、「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_sh4a_2a_big_ether.mot、ap_sh4a_2a_big_ether.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。

KPIT 社製 GCC

- ① HEW を起動し、¥sample¥AP-SH4A-2A_Big¥gcc¥ether¥ap_sh4a_2a_big_ether.hws を読み込みます。
- ② 最初の読み込みを行ったときに、「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「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。
- ④ 最初の読み込みを行ったときに、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。
- ⑤ [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。
- ⑥ メニューの [Build] - [Build] を実行して下さい。ap_sh4a_2a_big_ether.mot、ap_sh4a_2a_big_ether.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-SH4A-2A のスイッチを、「1.2 動作モード」 「Fig1.2-1 動作モード設定」を参考に設定します。なお、SW1 はビッグエンディアン使用時、SS1 は DEBUG モードに設定して下さい。
- ② XrossFinder を起動し、¥sample フォルダ直下にある XrossFinder_sh4a_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_sh4a_2a_big_ether.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-SH4A-2A のスイッチを、「1.2 動作モード」 「Fig1.2-1 動作モード設定」を参考に設定します。なお、SW1 はビッグエンディアン使用時、SW2 は FLASHROM を使用する・ライトプロテクト解除、SS1 は DEBUG モードに設定して下さい。
- ② ¥release フォルダ内の ap_sh4a_2a_big_ether.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 Fig5.1-3 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。

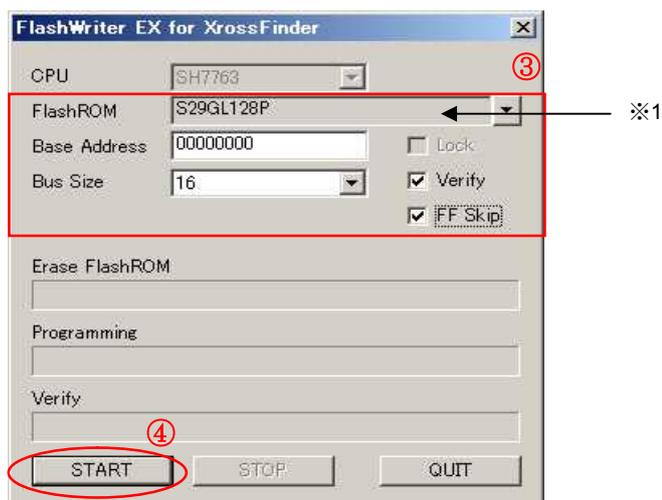


Fig5.1-3 FlashWriter EX for XrossFinder の設定

(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 AP-SH4A-2A のスイッチを、「1.2 動作モード」 「Fig1.2-1 動作モード設定」を参考に設定します。SW1 はビッグエンディアン使用時、SW2 は FLASHROM を使用する・ライトプロテクト解除、SS1 は DEBUG モードに設定して下さい。
- ③ FlashWriterEX を起動して、「Table5.1-3 FlashWriterEX の設定」を参考に設定を行って下さい。
- ④ FlashWriterEX で、¥sample フォルダ直下にある XrossFinder_sh4a_2a.xfc コマンドファイルを使用するように設定して下さい。
- ⑤ ¥release フォルダ内の ap_sh4a_2a_big_ether.mot をボードに書き込みます。
- ⑥ AP-SH4A-2A のスイッチを、「1.2 動作モード」 「Fig1.2-1 動作モード設定」を参考に設定します。なお、SW1 はビッグエンディアン使用時、SW2 は FLASHROM を使用する、SS1 は NORMAL モードに設定して下さい。
- ⑦ 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」 HJ-LINK 使用時は「HJ-LINK」
JTAG クロック (XrossFinder 使用時のみ)	10MHz 以下
CPU	SH7763
BaseAddress	00000000
FLASHROM	S29GL128P ※1
Bus Size	16

Table5.1-3 FlashWriter EX の設定

※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下さい。

5.2 ネットワークサンプルプログラムの動作説明

5.2.1 ネットワークサンプルプログラム概要

ネットワークサンプルプログラムは、TCP/IP プロトコルスタックに uIP(マイクロアイピー)を使用しています。uIP に関する詳細は、uIP ホームページ「http://www.sics.se/~adam/uip/index.php/Main_Page」を参照してください。

ネットワークサンプルプログラムは、下記の動作を行います。

- SCIF0 でエコーバックを行います。(送受信割り込み使用)
SCIF0 から受信した値をそのまま、SCIF0 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、フロー制御なしです。
動作確認は、パソコン上のターミナルソフト (ハイパーターミナルなど) 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 1sec 間隔で ON/OFF します。(TMU1 割り込み使用)
- Ethernet (CN6 および CN7) でエコーバックを行います※1
※1. ネットワーク動作の詳細は、「5.2.2 ネットワーク動作」を参照してください。

5.2.2 ネットワーク動作

ネットワーク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
LAN ポート	10/100BASE-TX 以上対応の LAN ポート
スイッチングハブ	10/100BASE-TX 以上対応のスイッチングハブ
LAN ケーブル	ストレートケーブル 3 本

(1) ネットワーク設定

本 CPU ボードは Ethernet インターフェースを 2 ポート備えています。ネットワーク設定は以下のようになっています。

ポート 0 (CN6)	IP アドレス	192.168.1.200
	サブネットマスク	255.255.255.0
	ゲートウェイ	192.168.1.254
	MAC アドレス	00-0C-7B-28-XX-XX ※1
ポート 1 (CN7)	IP アドレス	192.168.1.201
	サブネットマスク	255.255.255.0
	ゲートウェイ	192.168.1.254
	MAC アドレス	00-0C-7B-28-YY-YY ※1

※1. XX-XX、YY-YY の値はボードごとに異なります

上記設定のうち、IP アドレス・サブネットマスク・ゲートウェイの設定はサンプルプログラム内の「src\ether_firm\ether_main.c」で行われています。

また、MAC アドレスは EEPROM の先頭 6 ワード(12 バイト)に格納されています。格納イメージを以下に示します。

アドレス	ポート	格納値
先頭アドレス	ポート 0 (CN6)	0x000C
先頭アドレス+1 ワード		0x7B28
先頭アドレス+2 ワード		0xXXXX ※2
先頭アドレス+3 ワード	ポート 1 (CN7)	0x000C
先頭アドレス+4 ワード		0x7B28
先頭アドレス+5 ワード		0xYYYY ※2

※2. XXXX、YYYY の値はボードごとに異なります

本製品の MAC アドレスは、弊社が米国電気電子学会 (IEEE) より取得したアドレスになります。MAC アドレスを変更される際は、お客様にて IEEE より MAC アドレスを取得してください。

(2) ネットワーク動作内容

以下の手順に従い、ネットワークの動作を確認してください。

- ① CPU ボードの Ethernet コネクタ (CN6、CN7) をそれぞれスイッチングハブと LAN ストレートケーブルで接続します。
- ② スwitchングハブとパソコンを LAN ストレートケーブルで接続します。

注意： ネットワークサンプルプログラムは LAN ケーブルの活栓挿抜に対応していないため、CPU ボードの電源投入前に必ず LAN 接続を完了してください。

LAN ケーブルが接続されていない場合、プログラムは正常に動作しません。

- ③ パソコン上でネットワークの設定を行います。
以下に示す内容で設定を行ってください。

IP アドレス	192.168.1.202
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.1.254

- ④ 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。

注意： LAN ケーブルが 1 本しか接続されていない場合、プログラムの起動に 5 秒以上時間がかかります。

- ⑤ パソコン上でターミナルソフト(ハイパーターミナルなど)を起動し、ポート 0 (CN6) の TCP/IP の設定を行います。
TCP/IP の設定は、ホストアドレス「192.168.1.200」、ポート番号「50000」です。
- ⑥ パソコン上でターミナルソフト(ハイパーターミナルなど)を起動し、ポート 1 (CN7) の TCP/IP の設定を行います。
TCP/IP の設定は、ホストアドレス「192.168.1.201」、ポート番号「50000」です。
- ⑦ ターミナルソフト上で接続が確認できましたら、任意のパケットを送信してください。
エコーバック動作が確認できれば終了です。

5.2.3 ネットワークサンプルプログラム注意事項

(1) 2 ポート対応

TCP/IP プロトコルスタック uIP は通常 Ethernet インターフェースを 1 ポートのみサポートします。ネットワークサンプルプログラムでは 2 ポートの Ethernet インターフェースに対応するため、ポート番号取得・設定関数を追加しています。

API 名	内容説明
uip_getifnum	現在設定されているポート番号の取得
uip_setifnum	ポート番号の設定

(2) 待ち時間処理

サンプルプログラムのソース「ネットワークデバイスソースフォルダ」内に 100us 間待ち処理を行う「LOOP_100us」定義がありますが、この定義は正確に 100us の待ち処理を行うものではありません。

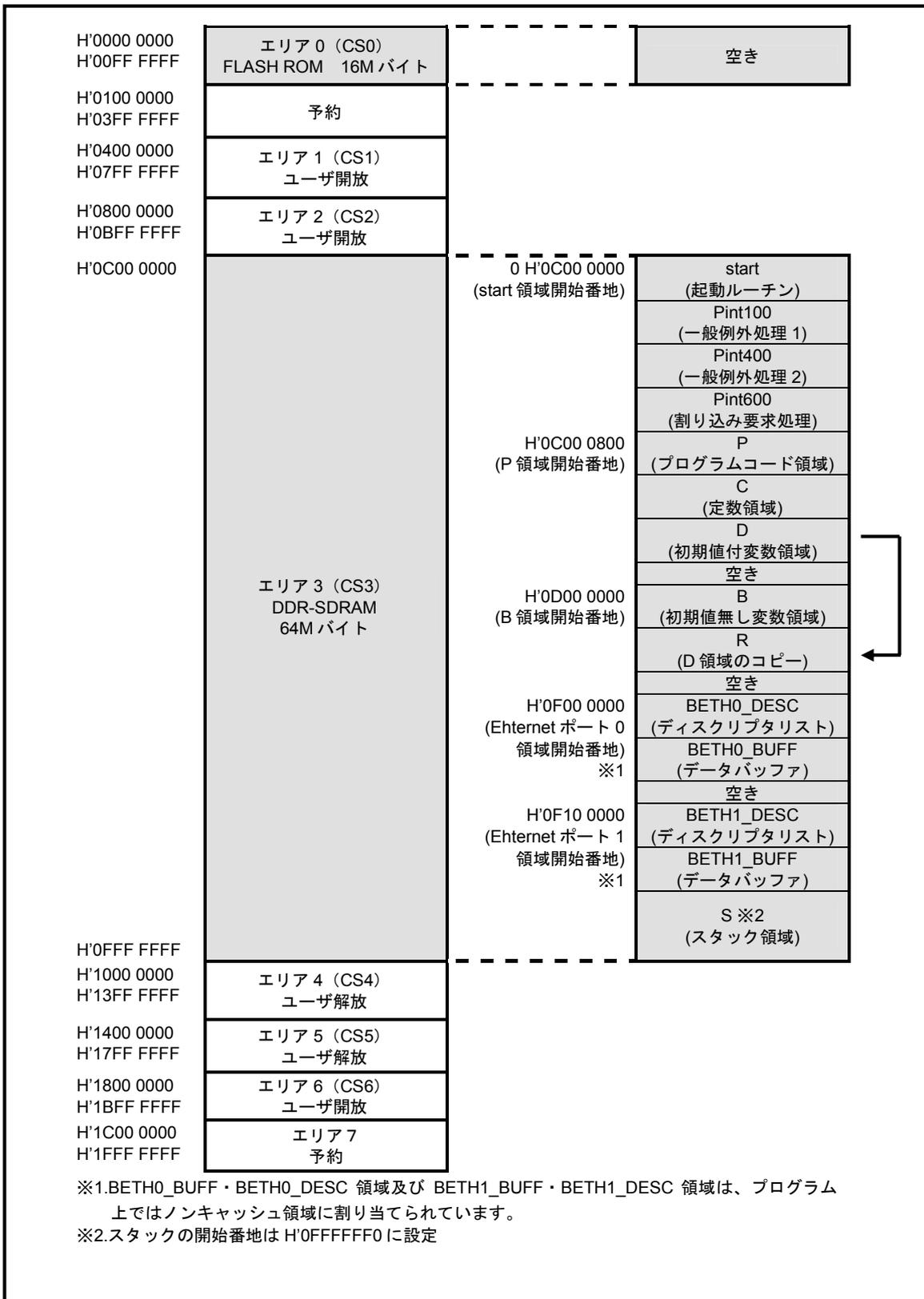
本サンプルプログラムでは、あくまで、指定した時間以上の待ち時間を得るために使用しておりますので、ご注意ください。

正確に 100us の待ち時間が必要な場合には、上記の定義の値を調整していただくか、タイマ (CMT 等) をご使用ください。

5.3 RAM 動作時のメモリマップ (ネットワーク)

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。)



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 USB および Ethernet デバイスドライバのサンプルソースの著作権は株式会社ルネサスエレクトロニクスが保有します。
- ・本文書に記載されているネットワークサンプルプログラム内の uIP(TCP/IP プロトコルスタック)には、BSD ライセンスが規定されています。
BSD ライセンスは、無保証であることの明記と著作権表示だけを再配布の条件とするライセンス規定です。
上記ライセンス規定に従い、uIP は無保証であり、著作権は「Adam Dunkels and the Swedish Institute of Computer Science」が保有します。
- ・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アルファプロジェクトが保有します。
- ・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、ルネサスエレクトロニクス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エレクトロニクス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。

商標について

- ・SH7763 は、ルネサスエレクトロニクス株式会社の登録商標、商標または商品名称です
- ・Windows®の正式名称は Microsoft®Windows®Operating System です。
- ・Microsoft、Windows 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 は、米国 Microsoft Corporation.の商品名称です。
- ・SuperH は、ルネサスエレクトロニクス株式会社の登録商標、商標または商品名称です。

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®7 は Windows 7 もしくは Win7
- ・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
- ・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