

AP-SH4AD-0A(SH7786 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.....	2
1.3 開発環境について.....	5
1.4 ワークスペースについて.....	5
2. サンプルプログラムの構成.....	6
2.1 フォルダ構成.....	6
2.2 ファイル構成.....	8
2.2.1 ビッグエンディアン・SHCのファイル構成.....	8
2.2.2 リトルエンディアン・SHCのファイル構成.....	12
3. USBホストサンプルプログラム.....	16
3.1 ビルド・デバッグ方法（USBホスト）.....	16
3.2 動作説明（USBホスト）.....	19
3.2.1 サンプルプログラム概要（USBホスト）.....	19
3.2.2 USBホスト動作.....	20
3.3 RAM動作時のメモリマップ（USBホスト）.....	23
3.4 ROM動作時のメモリマップ（USBホスト）.....	24
4. USBファンクションサンプルプログラム.....	25
4.1 ビルド・デバッグ方法（USBファンクション）.....	25
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 動作説明（ネットワーク）.....	35
5.2.1 サンプルプログラム概要（ネットワーク）.....	35
5.2.2 ネットワーク動作.....	36
5.3 RAM動作時のメモリマップ（ネットワーク）.....	38
5.4 ROM動作時のメモリマップ（ネットワーク）.....	39

1. 概要

1.1 概要

本サンプルプログラムは、AP-SH4AD-0A の動作を確認する簡単なプログラムです。また、弊社製 XrossFinderEvo デバッガ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。

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	<ul style="list-style-type: none">・ USB ホスト デバイス情報表示・ シリアル通信・ タイマ割り込み・ PCI-Express カードコンフィグレーション情報表示・ ディスプレイ表示
USB ファンクションサンプルプログラム	<ul style="list-style-type: none">・ USB ファンクション 仮想シリアル通信・ シリアル通信・ タイマ割り込み・ PCI-Express カードコンフィグレーション情報表示・ ディスプレイ表示
ネットワークサンプルプログラム	<ul style="list-style-type: none">・ ネットワーク通信・ シリアル通信・ タイマ割り込み・ PCI-Express カードコンフィグレーション情報表示・ ディスプレイ表示

詳細な動作内容に関しては、後述の「3. USB ホストサンプルプログラム」、「4. USB ファンクションサンプルプログラム」、「5. ネットワークサンプルプログラム」を参照してください。

1.2 動作モード

本サンプルプログラムは、AP-SH4AD-0A 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-SH4AD-0A 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。

クロック動作モード	: モード3
エリア0 バス幅	: SRAM インターフェース 16bit
エリア0 拡張領域	: 通常モード
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。
クロック入力	: 水晶発振子
アドレス拡張モード	: 29 ビットアドレスモード
PCI Express モード	: ルートポートモード
PCI Express PHY	: 4 レーン + 1 レーン
AUD モード	: AUD 有効
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	: 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

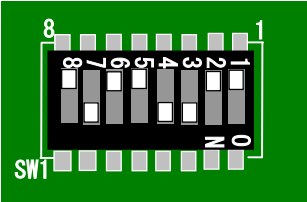
<SW1 設定>

クロック動作モード : モード3

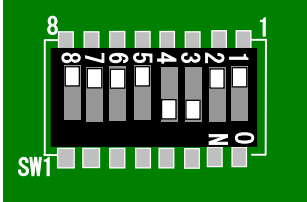
エリア0バス幅 : SRAM インターフェース 16ビット

エリア0領域拡張 : 通常モード

クロック入力 : 水晶発振子



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>

アドレス拡張モード : 29ビットアドレスモード

PCI Express モード : ルートポートモード

PCI Express PHY モード : 4レーン + 1レーン

AUD モード : AUD有効

・ SW4

<SW4 設定>

USB チャンネル : JSW1 に依存

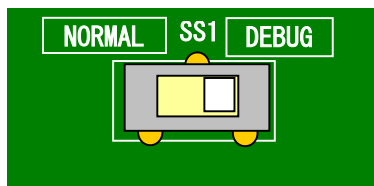
FLASHROM ライトプロテクト設定 : ライトプロテクト解除

PCI Express WAKE 設定 : WAKE を使用しない

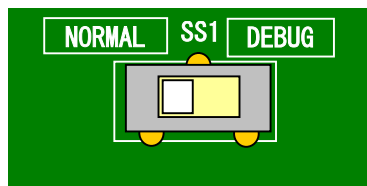
PCI Express PRSNT 設定 : PRSNT を使用しない

Fig1.2-1 動作モード設定

・ SS1

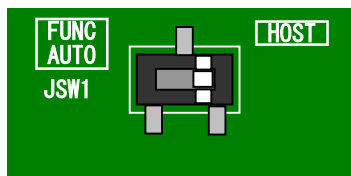


XrossFinder 使用時
DEBUG モード

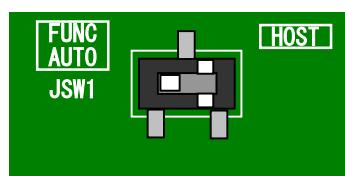


XrossFinder 未使用時
NORMAL モード

・ JSW1

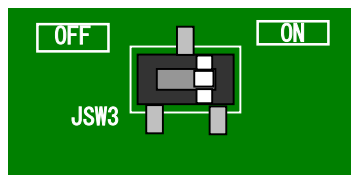


USB ホスト使用時



USB ファンクション使用時

・ JSW3



FLASHROM 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	4.08.00.011以降	SHC ※1 (ルネサスエレクトロニクス)	V9.4.0.0以降	SuperH ファミリ用 C/C++コンパイラパッケージに付属

※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。

1.4 ワークスペースについて

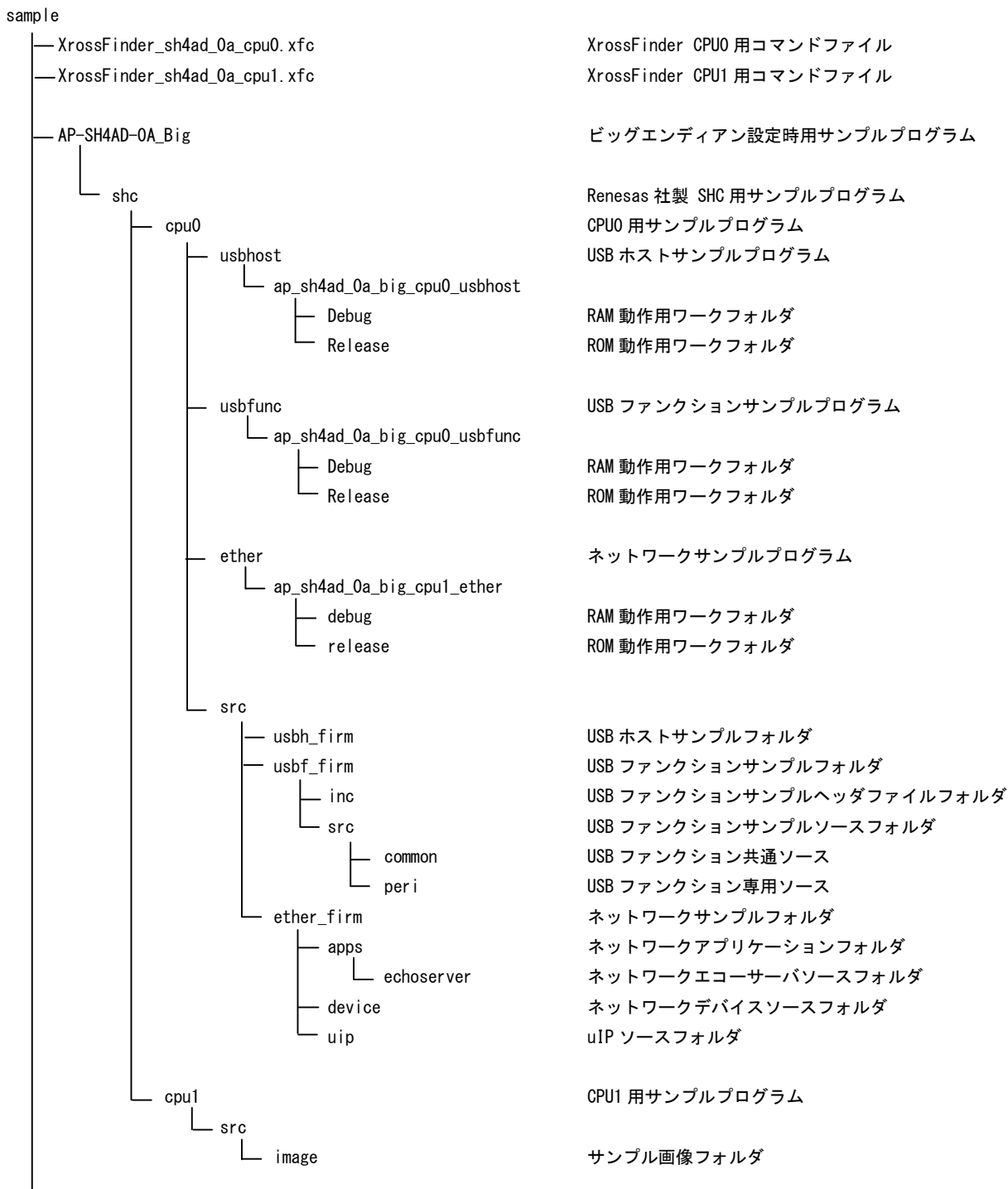
本サンプルプログラムの総合開発環境 High-performance Embedded Workshop ワークスペースは次のフォルダに格納されています。

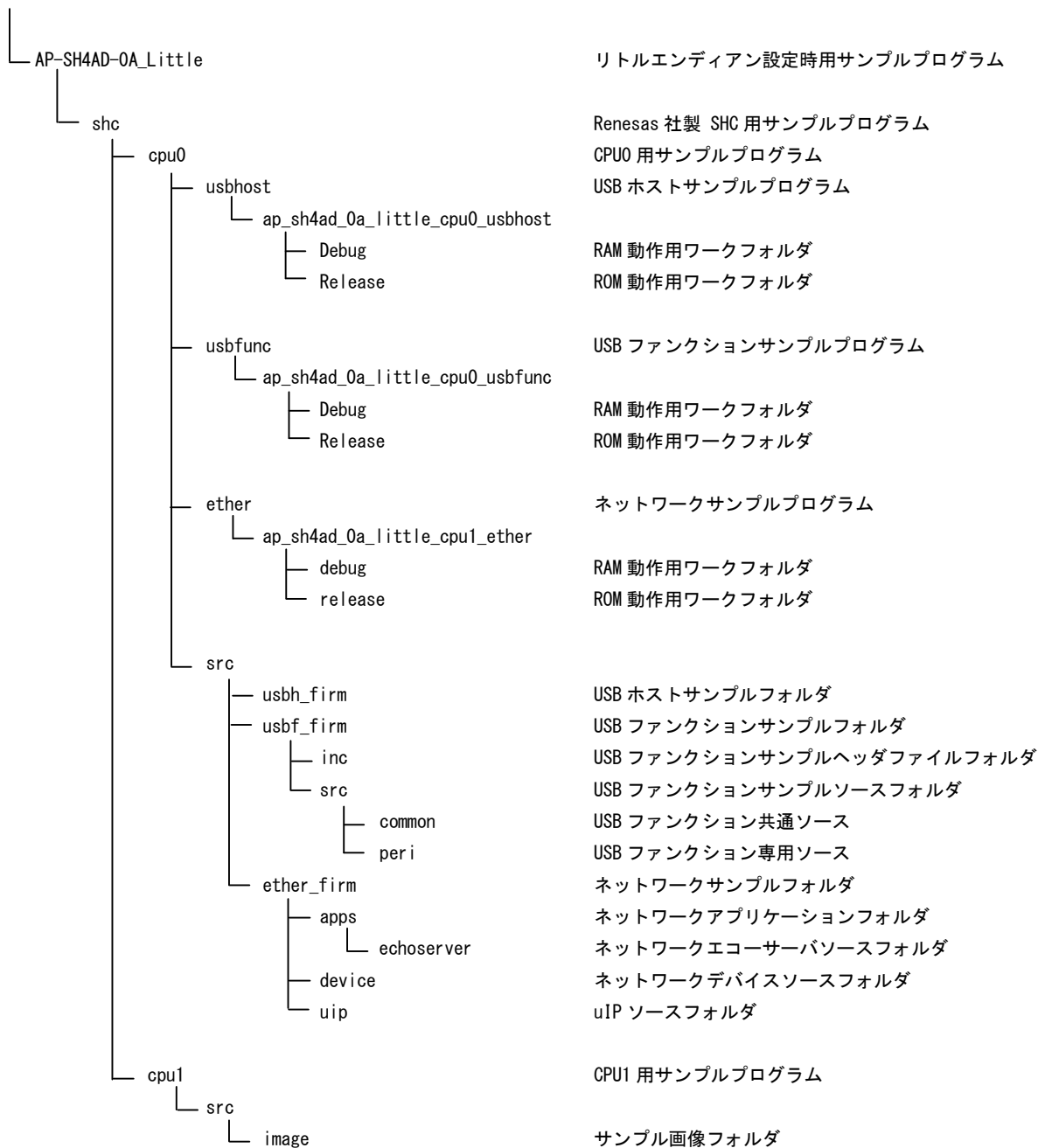
サンプルプログラム	CPU	エンディアン	ワークスペース
USB ホストサンプルプログラム	CPU0	ビッグ	¥sample¥AP-SH4AD-0A_Big¥shc¥cpu0¥usbhost¥ap_sh4ad_0a_big_cpu0_usbhost.hws
		リトル	¥sample¥AP-SH4AD-0A_Little¥shc¥cpu0¥usbhost¥ap_sh4ad_0a_little_cpu0_usbhost.hws
USB ファンクションサンプルプログラム	CPU0	ビッグ	¥sample¥AP-SH4AD-0A_Big¥shc¥cpu0¥usbfunc¥ap_sh4ad_0a_big_cpu0_usbfunc.hws
		リトル	¥sample¥AP-SH4AD-0A_Little¥shc¥cpu0¥usbfunc¥ap_sh4ad_0a_little_cpu0_usbfunc.hws
ネットワークサンプルプログラム	CPU0	ビッグ	¥sample¥AP-SH4AD-0A_Big¥shc¥cpu0¥ether¥ap_sh4ad_0a_big_cpu0_ether.hws
		リトル	¥sample¥AP-SH4AD-0A_Little¥shc¥cpu0¥ether¥ap_sh4ad_0a_little_cpu0_ether.hws
CPU1 用サンプルプログラム	CPU1	ビッグ	¥sample¥AP-SH4AD-0A_Big¥shc¥cpu1¥ap_sh4ad_0a_big_cpu1.hws
		リトル	¥sample¥AP-SH4AD-0A_Little¥shc¥cpu1¥ap_sh4ad_0a_little_cpu1.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-SH4AD-0A_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¥cpu0 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
src	...	サンプルソースファイルフォルダ

<shc¥cpu0¥usbhost フォルダ内>

ap_sh4ad_0a_big_cpu0_usbhost.hws	...	USB ホスト HEW 用ワークスペースファイル
----------------------------------	-----	--------------------------

<shc¥cpu0¥usbhost¥ap_sh4ad_0a_big_cpu0_usbhost フォルダ内>

ap_sh4ad_0a_big_cpu0_usbhost.hwp	...	USB ホスト HEW 用プロジェクトファイル
----------------------------------	-----	-------------------------

<shc¥cpu0¥usbhost¥ap_sh4ad_0a_big_cpu0_usbhost¥Debug フォルダ内>

ap_sh4ad_0a_big_cpu0_usbhost.abs	...	USB ホスト RAM 動作用オブジェクトファイル(elf 形式)
ap_sh4ad_0a_big_cpu0_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_sh4ad_0a_big_cpu0_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¥cpu0¥usbhost¥ap_sh4ad_0a_big_cpu0_usbhost¥Release フォルダ内>

ap_sh4ad_0a_big_cpu0_usbhost.abs	...	USB ホスト ROM 動作用オブジェクトファイル (elf 形式)
ap_sh4ad_0a_big_cpu0_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_sh4ad_0a_big_cpu0_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¥cpu0¥usbfunc フォルダ内>

ap_sh4ad_0a_big_cpu0_usbfunc.hws	...	USB ファンクション HEW 用ワークスペースファイル
----------------------------------	-----	------------------------------

<shc¥cpu0¥usbfunc¥ap_sh4ad_0a_big_cpu0_usbfunc フォルダ内>

ap_sh4ad_0a_big_cpu0_usbfunc.hwp	...	USB ファンクション HEW 用プロジェクトファイル
----------------------------------	-----	-----------------------------

<shc¥cpu0¥usbfunc¥ap_sh4ad_0a_big_cpu0_usbfunc¥Debug フォルダ内>

ap_sh4ad_0a_big_cpu0_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_sh4ad_0a_big_cpu0_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_sh4ad_0a_big_cpu0_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¥cpu0¥usbfunc¥ap_sh4ad_0a_big_cpu0_usbfunc¥Release フォルダ内>

ap_sh4ad_0a_big_cpu0_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_sh4ad_0a_big_cpu0_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_sh4ad_0a_big_cpu0_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¥cpu0¥ether フォルダ内>

ap_sh4ad_0a_big_cpu0_ether.hws	...	ネットワーク HEW 用ワークスペースファイル
--------------------------------	-----	-------------------------

<shc¥cpu0¥ether¥ap_sh4ad_0a_big_cpu0_ether フォルダ内>

ap_sh4ad_0a_big_cpu0_ether.hwp	...	ネットワーク HEW 用プロジェクトファイル
--------------------------------	-----	------------------------

<shc¥cpu0¥ether¥ap_sh4ad_0a_big_cpu0_ether¥Debug フォルダ内>

ap_sh4ad_0a_big_cpu0_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_sh4ad_0a_big_cpu0_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_sh4ad_0a_big_cpu0_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¥cpu0¥ether¥ap_sh4ad_0a_big_cpu0_ether¥Release フォルダ内>

ap_sh4ad_0a_big_cpu0_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_sh4ad_0a_big_cpu0_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_sh4ad_0a_big_cpu0_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¥cpu0¥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
pcie.c	...	PCI Express ルートポート処理
boot.c	...	CPU 初期化処理
vector.c	...	割込ベクタテーブル
common.h	...	共通ヘッダファイル
pcie.c	...	PCI Express ルートポート定義ヘッダファイル
SH7786.h	...	SH7786 内部レジスタ定義ヘッダファイル
cpu.h	...	CPU 定義ファイル
startup.src	...	スタートアップ処理
section.src	...	セクション定義

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¥cpu0¥src¥usbh_firm フォルダ内>

usbh. c	...	USB ホスト処理
ohci. c	...	OHCI ホスト処理
ehci. c	...	EHCI ホスト処理
usbh. h	...	USB ホストヘッダファイル
ohci. h	...	OHCI ホストヘッダファイル
ehci. h	...	EHCI ホストヘッダファイル

<shc¥cpu0¥src¥usbf_firm フォルダ内>

c_version. h	...	バージョン情報ファイル
--------------	-----	-------------

<shc¥cpu0¥src¥usbf_firm¥src フォルダ内>

c_datatbl. c	...	USB ファンクション用データバッファ
p_main. c	...	USB ファンクション用メインソース
common	...	USB ファンクション共通ソースフォルダ
per i	...	USB ファンクション専用ソースフォルダ

<shc¥cpu0¥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¥cpu0¥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
------------	-----	---------------------

<shc¥cpu1 フォルダ内>

ap_sh4ad_0a_big_cpu1. hws	...	CPU1 HEW 用ワークスペースファイル
---------------------------	-----	-----------------------

<shc¥cpu1¥ap_sh4ad_0a_big_cpu1 フォルダ内>

ap_sh4ad_0a_big_cpu1. hwp	...	CPU1 HEW 用プロジェクトファイル
---------------------------	-----	----------------------

<shc¥cpu1¥ap_sh4ad_0a_big_cpu1¥Debug フォルダ内>

ap_sh4ad_0a_big_cpu1. abs	...	CPU1 RAM 動作用オブジェクトファイル(elf 形式)
ap_sh4ad_0a_big_cpu1. mot	...	CPU1 RAM 動作用モトローラ S フォーマット形式ファイル
ap_sh4ad_0a_big_cpu1. map	...	CPU1 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¥cpu1¥ap_sh4ad_0a_big_cpu1¥Release フォルダ内>

ap_sh4ad_0a_big_cpu1. abs	...	CPU1 ROM 動作用オブジェクトファイル (elf 形式)
ap_sh4ad_0a_big_cpu1. mot	...	CPU1 ROM 動作用モトローラ S フォーマット形式ファイル
ap_sh4ad_0a_big_cpu1. map	...	CPU1 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

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¥cpu1¥src フォルダ内>

main.c	...	メイン処理
tmr.c	...	タイマ処理
du.c	...	ディスプレイ表示処理
boot.c	...	CPU 初期化処理
vector.c	...	割込ベクタテーブル
common.h	...	共通ヘッダファイル
du.h	...	ディスプレイ設定ヘッダファイル
SH7786.h	...	SH7786 内部レジスタ定義ヘッダファイル
cpu.h	...	CPU 定義ファイル
startup.src	...	スタートアップ処理
section.src	...	セクション定義

<shc¥cpu1¥src¥image フォルダ内>

image1.c	...	サンプル画像 1
image2.c	...	サンプル画像 2
image3.c	...	サンプル画像 3

2.2.2 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-SH4AD-0A_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¥cpu0 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
src	...	サンプルソースファイルフォルダ

<shc¥cpu0¥usbhost フォルダ内>

ap_sh4ad_0a_little_cpu0_usbhost.hws	...	USB ホスト HEW 用ワークスペースファイル
-------------------------------------	-----	--------------------------

<shc¥cpu0¥usbhost¥ap_sh4ad_0a_little_cpu0_usbhost フォルダ内>

ap_sh4ad_0a_little_cpu0_usbhost.hwp	...	USB ホスト HEW 用プロジェクトファイル
-------------------------------------	-----	-------------------------

<shc¥cpu0¥usbhost¥ap_sh4ad_0a_little_cpu0_usbhost¥Debug フォルダ内>

ap_sh4ad_0a_little_cpu0_usbhost.abs	...	USB ホスト RAM 動作用オブジェクトファイル(elf 形式)
ap_sh4ad_0a_little_cpu0_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_sh4ad_0a_little_cpu0_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¥cpu0¥usbhost¥ap_sh4ad_0a_little_cpu0_usbhost¥Release フォルダ内>

ap_sh4ad_0a_little_cpu0_usbhost.abs	...	USB ホスト ROM 動作用オブジェクトファイル (elf 形式)
ap_sh4ad_0a_little_cpu0_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_sh4ad_0a_little_cpu0_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¥cpu0¥usbfunc フォルダ内>

ap_sh4ad_0a_little_cpu0_usbfunc.hws	...	USB ファンクション HEW 用ワークスペースファイル
-------------------------------------	-----	------------------------------

<shc¥cpu0¥usbfunc¥ap_sh4ad_0a_little_cpu0_usbfunc フォルダ内>

ap_sh4ad_0a_little_cpu0_usbfunc.hwp	...	USB ファンクション HEW 用プロジェクトファイル
-------------------------------------	-----	-----------------------------

<shc¥cpu0¥usbfunc¥ap_sh4ad_0a_little_cpu0_usbfunc¥Debug フォルダ内>

ap_sh4ad_0a_little_cpu0_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_sh4ad_0a_little_cpu0_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_sh4ad_0a_little_cpu0_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¥cpu0¥usbfunc¥ap_sh4ad_0a_little_cpu0_usbfunc¥Release フォルダ内>

ap_sh4ad_0a_little_cpu0_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_sh4ad_0a_little_cpu0_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_sh4ad_0a_little_cpu0_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¥cpu0¥ether フォルダ内>

ap_sh4ad_0a_little_cpu0_ether.hws	...	ネットワーク HEW 用ワークスペースファイル
-----------------------------------	-----	-------------------------

<shc¥cpu0¥ether¥ap_sh4ad_0a_little_cpu0_ether フォルダ内>

ap_sh4ad_0a_little_cpu0_ether.hwp	...	ネットワーク HEW 用プロジェクトファイル
-----------------------------------	-----	------------------------

<shc¥cpu0¥ether¥ap_sh4ad_0a_little_cpu0_ether¥Debug フォルダ内>

ap_sh4ad_0a_little_cpu0_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_sh4ad_0a_little_cpu0_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_sh4ad_0a_little_cpu0_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¥cpu0¥ether¥ap_sh4ad_0a_little_cpu0_ether¥Release フォルダ内>

ap_sh4ad_0a_little_cpu0_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_sh4ad_0a_little_cpu0_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_sh4ad_0a_little_cpu0_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¥cpu0¥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
pcie.c	...	PCI Express ルートポート処理
boot.c	...	CPU 初期化処理
vector.c	...	割込ベクタテーブル
common.h	...	共通ヘッダファイル
pcie.c	...	PCI Express ルートポート定義ヘッダファイル
SH7786.h	...	SH7786 内部レジスタ定義ヘッダファイル
cpu.h	...	CPU 定義ファイル
startup.src	...	スタートアップ処理
section.src	...	セクション定義

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¥cpu0¥src¥usbh_firm フォルダ内>

usbh. c	...	USB ホスト処理
ohci. c	...	OHCI ホスト処理
ehci. c	...	EHCI ホスト処理
usbh. h	...	USB ホストヘッダファイル
ohci. h	...	OHCI ホストヘッダファイル
ehci. h	...	EHCI ホストヘッダファイル

<shc¥cpu0¥src¥usbf_firm フォルダ内>

c_version. h	...	バージョン情報ファイル
--------------	-----	-------------

<shc¥cpu0¥src¥usbf_firm¥src フォルダ内>

c_datatbl. c	...	USB ファンクション用データバッファ
p_main. c	...	USB ファンクション用メインソース
common	...	USB ファンクション共通ソースフォルダ
peri	...	USB ファンクション専用ソースフォルダ

<shc¥cpu0¥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¥cpu0¥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
------------	-----	---------------------

<shc¥cpu1 フォルダ内>

ap_sh4ad_0a_little_cpu1. hws	...	CPU1 HEW 用ワークスペースファイル
------------------------------	-----	-----------------------

<shc¥cpu1¥ap_sh4ad_0a_little_cpu1 フォルダ内>

ap_sh4ad_0a_little_cpu1. hwp	...	CPU1 HEW 用プロジェクトファイル
------------------------------	-----	----------------------

<shc¥cpu1¥ap_sh4ad_0a_little_cpu1¥Debug フォルダ内>

ap_sh4ad_0a_little_cpu1. abs	...	CPU1 RAM 動作用オブジェクトファイル(elf 形式)
ap_sh4ad_0a_little_cpu1. mot	...	CPU1 RAM 動作用モトローラ S フォーマット形式ファイル
ap_sh4ad_0a_little_cpu1. map	...	CPU1 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¥cpu1¥ap_sh4ad_0a_little_cpu1¥Release フォルダ内>

ap_sh4ad_0a_little_cpu1. abs	...	CPU1 ROM 動作用オブジェクトファイル (elf 形式)
ap_sh4ad_0a_little_cpu1. mot	...	CPU1 ROM 動作用モトローラ S フォーマット形式ファイル
ap_sh4ad_0a_little_cpu1. map	...	CPU1 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		

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¥cpu1¥src フォルダ内>

main. c	...	メイン処理
tmr. c	...	タイマ処理
du. c	...	ディスプレイ表示処理
boot. c	...	CPU 初期化処理
vector. c	...	割込ベクタテーブル
common. h	...	共通ヘッダファイル
du. h	...	ディスプレイ設定ヘッダファイル
SH7786. h	...	SH7786 内部レジスタ定義ヘッダファイル
cpu. h	...	CPU 定義ファイル
startup. src	...	スタートアップ処理
section. src	...	セクション定義

<shc¥cpu1¥src¥image フォルダ内>

image1. c	...	サンプル画像 1
image2. c	...	サンプル画像 2
image3. c	...	サンプル画像 3

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 High-performance Embedded Workshop を起動し、¥sample¥AP-SH4AD-0A_Big¥shc¥cpu0¥usbhost¥ap_sh4ad_0a_big_cpu0_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_sh4ad_0a_big_cpu0_usbhost.mot、ap_sh4ad_0a_big_cpu0_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。
- ⑥ High-performance Embedded Workshop を一度終了して、¥sample¥AP-SH4AD-0A_Big¥shc¥cpu1¥ap_sh4ad_0a_big_cpu1.hws を読み込みます。
- ⑦ 上記①～⑤と同様にビルドを行って下さい。

High-performance Embedded Workshop の詳細な使用方法につきましては、High-performance Embedded Workshop 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-SH4AD-0A 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 XsSight を起動し、CPU0 ウィンドウを選択します。
- ③ ¥sample フォルダ直下にある XrossFinder_sh4ad_0a_cpu0.xfc コマンドファイルを読み込みます。
- ④ ¥cpu0¥usbhost¥ap_sh4ad_0a_big_cpu0_usbhost¥Debug フォルダ内の ap_sh4ad_0a_big_cpu0_usbhost.abs または ap_sh4ad_0a_big_cpu0_usbhost.x を CPU0 にダウンロードします。
- ⑤ XsSight 上で、CPU1 ウィンドウを選択します。
- ⑥ ¥sample フォルダ直下にある XrossFinder_sh4ad_0a_cpu1.xfc コマンドファイルを読み込みます。
- ⑦ ¥cpu1¥ap_sh4ad_0a_big_cpu1¥Debug フォルダ内の ap_sh4ad_0a_big_cpu1.abs または ap_sh4ad_0a_big_cpu1.x を CPU1 にダウンロードして、動作を確認して下さい。

(3) ROM上でのデバッグ

- ① AP-SH4AD-0A 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 XsSight を起動し、CPU0 ウィンドウを選択します。
- ③ ¥cpu0¥usbhost¥ap_sh4ad_0a_big_cpu0_usbhost¥Release フォルダ内の ap_sh4ad_0a_big_cpu0_usbhost.abs または ap_sh4ad_0a_big_cpu0_usbhost.x を CPU0 に読み込みます。
- ④ ¥sample フォルダ直下にある XrossFinder_sh4ad_0a_cpu0.xfc コマンドファイルを読み込みます。
- ⑤ XsSight 上で、CPU1 ウィンドウを選択します。
- ⑥ ¥cpu1¥ap_sh4ad_0a_big_cpu1¥Release フォルダ内の ap_sh4ad_0a_big_cpu1.abs または ap_sh4ad_0a_big_cpu1.x を CPU1 に読み込みます。
- ⑦ XsSight 上で、CPU0 ウィンドウを選択します。
- ⑧ XsSight 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。
- ⑨ 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。

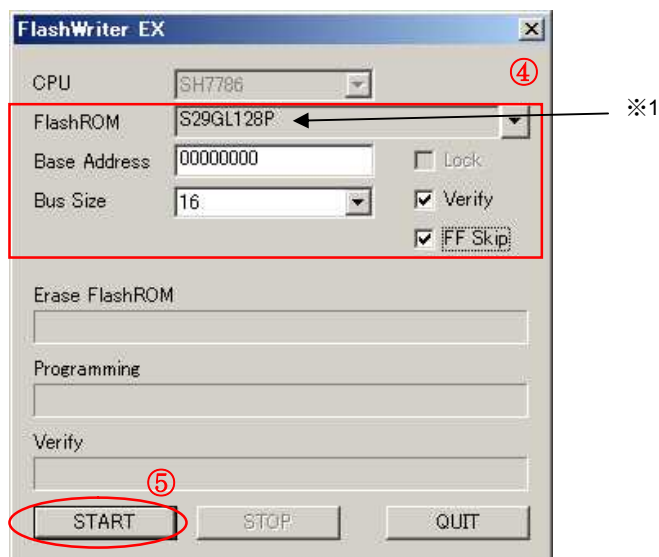


Fig3.1-1 FlashWriter EX の設定

- ※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ください。

(4) XsSight 未使用時の確認方法

・FlashWriter EXを使用する場合

- ① アダプタ (XrossFinderEvo、XrossFinder または HJ-LINK) を使用して PC とボードを繋ぎます。
- ② AP-SH4AD-0A 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 FlashWriter EX を起動して、「Table3.1-1 FlashWriter EX の設定」を参考に設定を行って下さい。
- ④ FlashWriter EX で、¥sample フォルダ直下にある XrossFinder_sh4ad_0a_cpu0.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 ¥cpu0¥usbhost¥ap_sh4ad_0a_big_cpu0_usbhost¥Release フォルダ内の ap_sh4ad_0a_big_cpu0_usbhost.mot を「File」メニューの「Open」から開きます。その後、¥cpu1¥ap_sh4ad_0a_little_cpu1¥Release フォルダ内の ap_sh4ad_0a_big_cpu1.mot を、「File」メニューの「Merge」から開き、ボードに書き込みを行います。(cpu0 と cpu1 の両方の mot ファイルを「Open」で開くと、先に読み込んだファイルの情報が消去されてしまいます。両方の mot ファイルをボードに書き込む際は、上記の方法に従い FlashWriter EX のマージ機能を使用するようにして下さい。)
- ⑥ AP-SH4AD-0A 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。

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	XrossFinderEvo 使用時は「XrossFinderEvo」 XrossFinder 使用時は「XrossFinder」 HJ-LINK 使用時は「HJ-LINK」
JTAG クロック (XrossFinderEvo/XrossFinder 使用時)	20MHz 以下
CPU	SH7786
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。

CPU0 側の動作内容

- 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。
- PCI-Express カードスロットに接続した PCI-Express アドインカードのコンフィグレーション情報を SCIF0 から出力します。
PCI-Express アドインカードによって電源の配給方法が異なります。ご使用になられる PCI-Express アドインカードに合わせて、必要な場合はオプション電源より+12V と+3.3Vaux を配給してください。
- USB ホストポート 0 (CN7A) に USB メモリを接続すると EHCI ドライバ (ハイスピード) が動作し、SCIF0 から接続状況とデバイス情報を出力します。 ※1
- USB ホストポート 1 (CN7B) に USB メモリを接続すると OHCI ドライバ (フルスピード) が動作し、SCIF0 から接続状況とデバイス情報を出力します。 ※1
- USB ホストポート 2 (CN7B) 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。

CPU1 側の動作内容

- LD1 (緑の LED) を 500msec 間隔で ON/OFF します。(TMU1 割り込み使用)
- LD2 (緑の LED) を 1sec 間隔で ON/OFF します。(TMU2 割り込み使用)
- DVI で接続したモニタに 5sec 間隔で画像を切り替えて表示します。表示画像は XGA サイズ (1024×768) で、切替画面数は 4 画面になります。

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 ボードのシリアルインターフェース (CN6) 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 (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。
- 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 0 (CN7A: 上段) に USB メモリを挿入します。
- ⑤ USB メモリを挿入すると、EHCI ドライバが自動的に 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 EHCI !!
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 EHCI !!
```

※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 EHCI!!
```

- ⑦ USB ホストポート 1 (CN7B: 下段) に USB メモリを挿入します。
- ⑧ USB メモリを挿入すると、OHCI ドライバが自動的に 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 OHCI !!
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 ※2
Transfer End OHCI !!
```

※2. XXXXXXXX には、USB メモリのデバイス情報が表示されます。

- ⑨ 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 OHCI!!
```

- ⑩ 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 (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。
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 1 (CN7B: 下段) 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 OHCI!!  
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
```

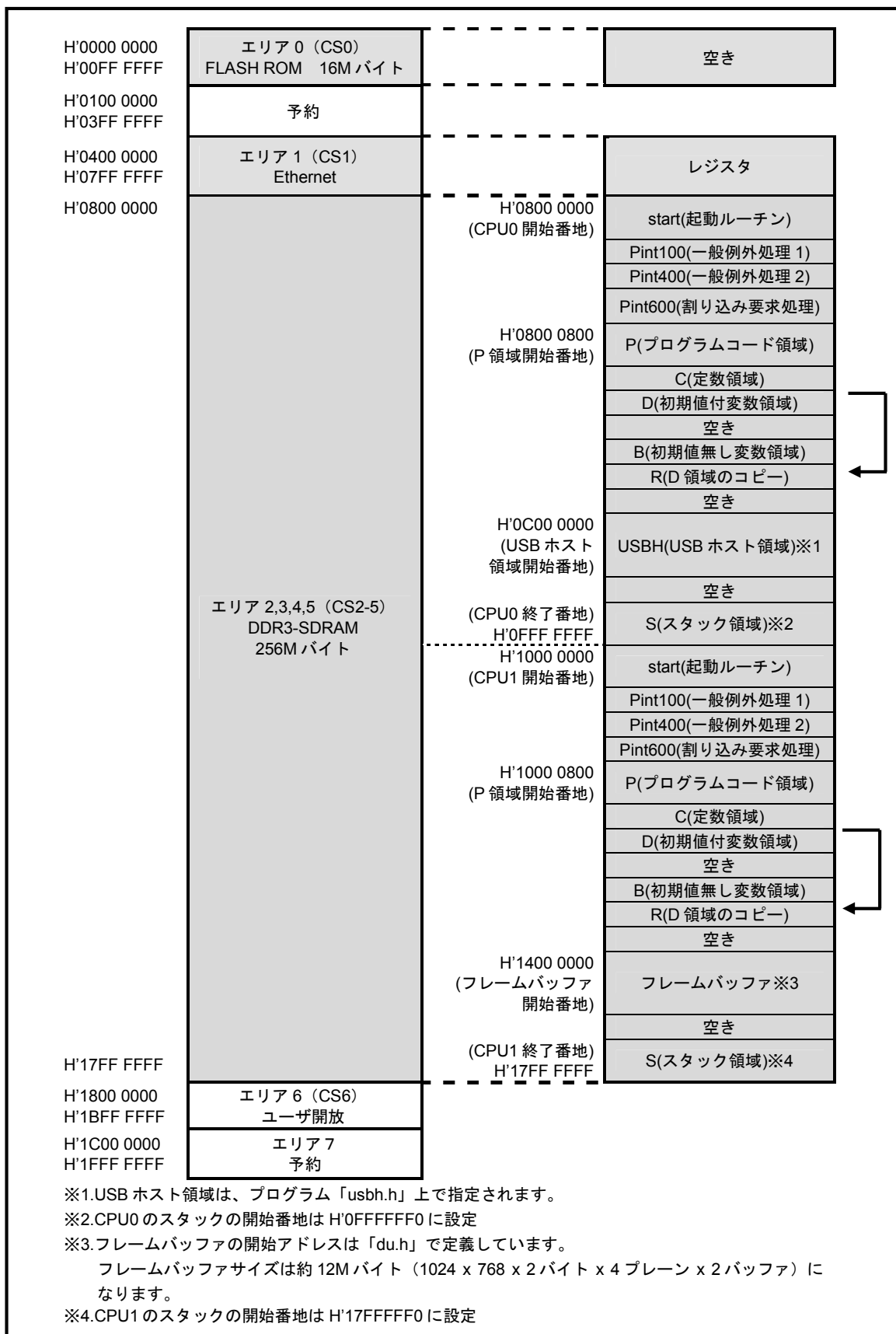
- ⑦ 上記の表示が出たことを確認した後に、USB マウスを抜いてください。ターミナルソフト上に切断状況が表示されません。以下のような表示が出ていることを確認してください。

```
DisConnected OHCI!!
```

- ⑧ 以上で USB マウス接続時の動作は終了です。

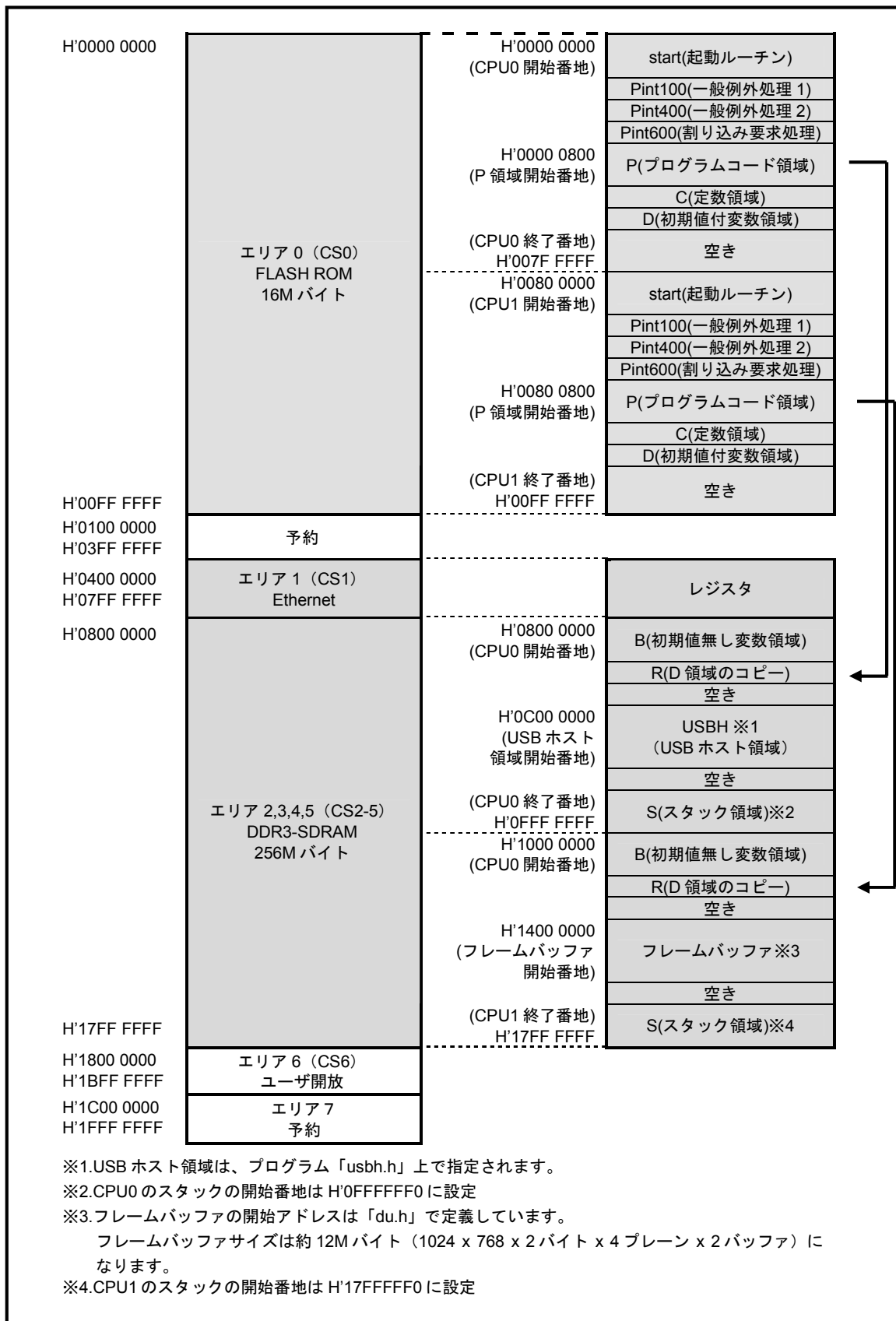
3.3 RAM 動作時のメモリマップ (USB ホスト)

メモリマップを以下に示します。



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。



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使用される場合は、「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 High-performance Embedded Workshop を起動し、¥sample¥AP-SH4AD-0A_Big¥shc¥cpu0¥usbfunc¥ap_sh4ad_0a_big_cpu0_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_sh4ad_0a_big_cpu0_usbfunc.mot、ap_sh4ad_0a_big_cpu0_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。
- ⑥ High-performance Embedded Workshop を一度終了して、¥sample¥AP-SH4AD-0A_Big¥shc¥cpu1¥ap_sh4ad_0a_big_cpu1.hws を読み込みます。
- ⑦ 上記①～⑤と同様にビルドを行って下さい。

High-performance Embedded Workshop の詳細な使用方法につきましては、High-performance Embedded Workshop 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-SH4AD-0A 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 XsSight を起動し、CPU0 ウィンドウを選択します。
- ③ ¥sample フォルダ直下にある XrossFinder_sh4ad_0a_cpu0.xfc コマンドファイルを読み込みます。
- ④ ¥cpu0¥usbfunc¥ap_sh4ad_0a_big_cpu0_usbfunc¥Debug フォルダ内の ap_sh4ad_0a_big_cpu0_usbfunc.abs または ap_sh4ad_0a_big_cpu0_usbfunc.x を CPU0 にダウンロードします。
- ⑤ XsSight 上で、CPU1 ウィンドウを選択します。
- ⑥ ¥sample フォルダ直下にある XrossFinder_sh4ad_0a_cpu1.xfc コマンドファイルを読み込みます。
- ⑦ ¥cpu1¥ap_sh4ad_0a_big_cpu1¥Debug フォルダ内の ap_sh4ad_0a_big_cpu1.abs または ap_sh4ad_0a_big_cpu1.x を CPU1 にダウンロードして、動作を確認して下さい。

(3) ROM上でのデバッグ

- ① AP-SH4AD-0A 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 はデバ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 XsSight を起動し、CPU0 ウィンドウを選択します。
- ③ ¥cpu0¥usbfunc¥ap_sh4ad_0a_big_cpu0_usbfunc¥Release フォルダ内の ap_sh4ad_0a_big_cpu0_usbfunc.abs または ap_sh4ad_0a_big_cpu0_usbfunc.x を CPU0 に読み込みます。
- ④ ¥sample フォルダ直下にある XrossFinder_sh4ad_0a_cpu0.xfc コマンドファイルを読み込みます。
- ⑤ XsSight 上で、CPU1 ウィンドウを選択します。
- ⑥ ¥cpu1¥ap_sh4ad_0a_big_cpu1¥Release フォルダ内の ap_sh4ad_0a_big_cpu1.abs または ap_sh4ad_0a_big_cpu1.x を CPU1 に読み込みます。
- ⑦ XsSight 上で、CPU0 ウィンドウを選択します。
- ⑧ XsSight のメニューから FlashWriter EX を選択し、下図 Fig4.1-1 のように設定を行ってください。
- ⑨ 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。

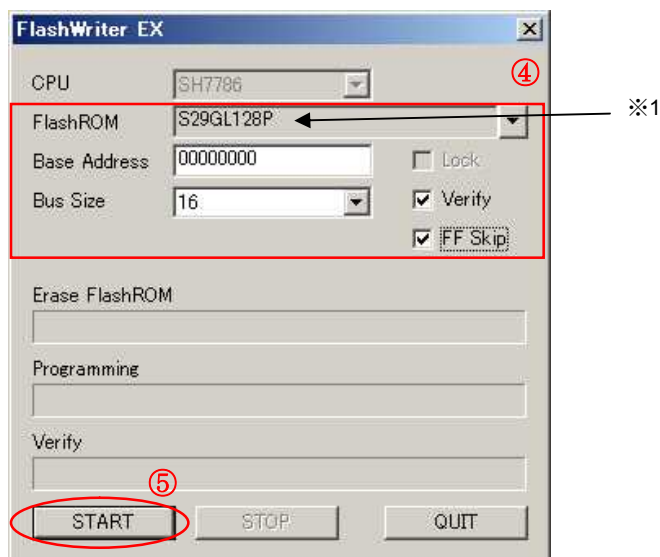


Fig4.1-1 FlashWriter EX の設定

- ※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ください。

(4) XsSight 未使用時の確認方法

・FlashWriter EXを使用する場合

- ① アダプタ (XrossFinderEvo、XrossFinder または HJ-LINK) を使用して PC とボードを繋ぎます。
- ② AP-SH4AD-0A 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 FlashWriter EX を起動して、「Table4.1-1 FlashWriter EX の設定」を参考に設定を行って下さい。
- ④ FlashWriter EX で、¥sample フォルダ直下にある XrossFinder_sh4ad_0a_cpu0.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 ¥cpu0¥usbfunc¥ap_sh4ad_0a_big_cpu0_usbfunc¥Release フォルダ内の ap_sh4ad_0a_big_cpu0_usbfunc.mot を「File」メニューの「Open」から開きます。その後、¥cpu1¥ap_sh4ad_0a_little_cpu1¥Release フォルダ内の ap_sh4ad_0a_big_cpu1.mot を、「File」メニューの「Merge」から開き、ボードに書き込みを行います。(cpu0 と cpu1 の両方の mot ファイルを「Open」で開くと、先に読み込んだファイルの情報が消去されてしまいます。両方の mot ファイルをボードに書き込む際は、上記の方法に従い FlashWriter EX のマージ機能を使用するようにして下さい。)
- ⑥ AP-SH4AD-0A 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。

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	XrossFinderEvo 使用時は「XrossFinderEvo」 XrossFinder 使用時は「XrossFinder」 HJ-LINK 使用時は「HJ-LINK」
JTAG クロック (XrossFinderEvo/XrossFinder 使用時)	20MHz 以下
CPU	SH7786
BaseAddress	00000000
FLASHROM	S29GL128P ※1
Bus Size	16

Table4.1-1 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。

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。

CPU0 側の動作内容

- 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。
- PCI-Express カードスロットに接続した PCI-Express アドインカードのコンフィグレーション情報を SCIF0 から出力します。
PCI-Express アドインカードによって電源の配給方法が異なります。ご使用になられる PCI-Express アドインカードに合わせて、必要な場合はオプション電源より+12V と+3.3Vaux を配給してください。
- 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。

CPU1 側の動作内容

- LD1 (緑の LED) を 500msec 間隔で ON/OFF します。(TMU1 割り込み使用)
- LD2 (緑の LED) を 1sec 間隔で ON/OFF します。(TMU2 割り込み使用)
- DVI で接続したモニタに 5sec 間隔で画像を切り替えて表示します。表示画像は XGA サイズ (1024×768) で、切替画面数は 4 画面になります。

4.2.2 USB ファンクション動作

(1) 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。

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 (CN8) 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 シリアルの動作は終了です。

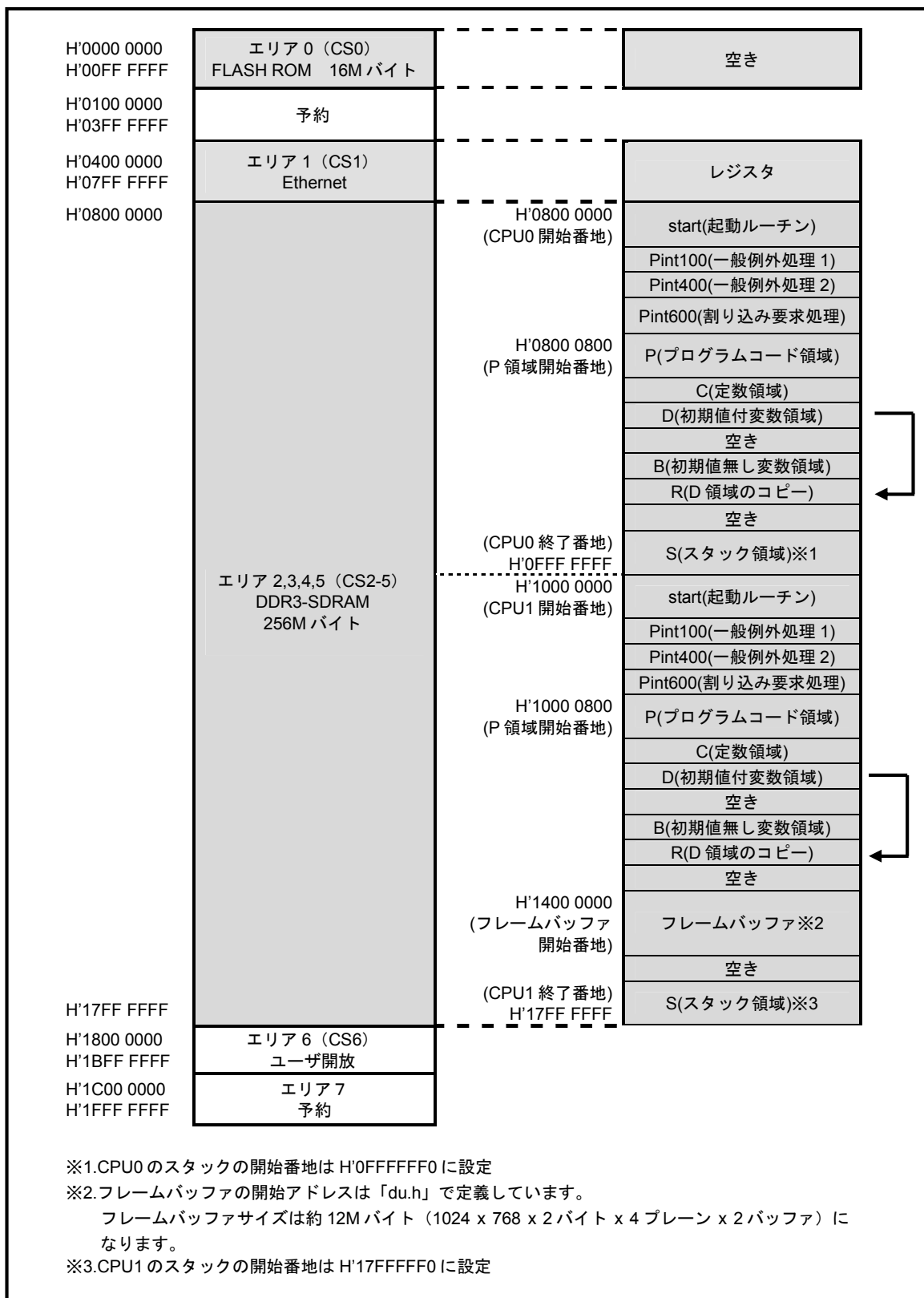
(2) USB サンプルプログラム注意事項

サンプルプログラムのソース「USB サンプル ファンクション共通ソース common フォルダ」内に 1msec 間待ち処理を行う「usb_delay_1ms」関数がありますが、この関数は正確に 1msec 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 1msec の待ち時間が必要な場合には、上記の関数のループ回数を調整していただくか、タイマ (TMU 等) をご使用ください。

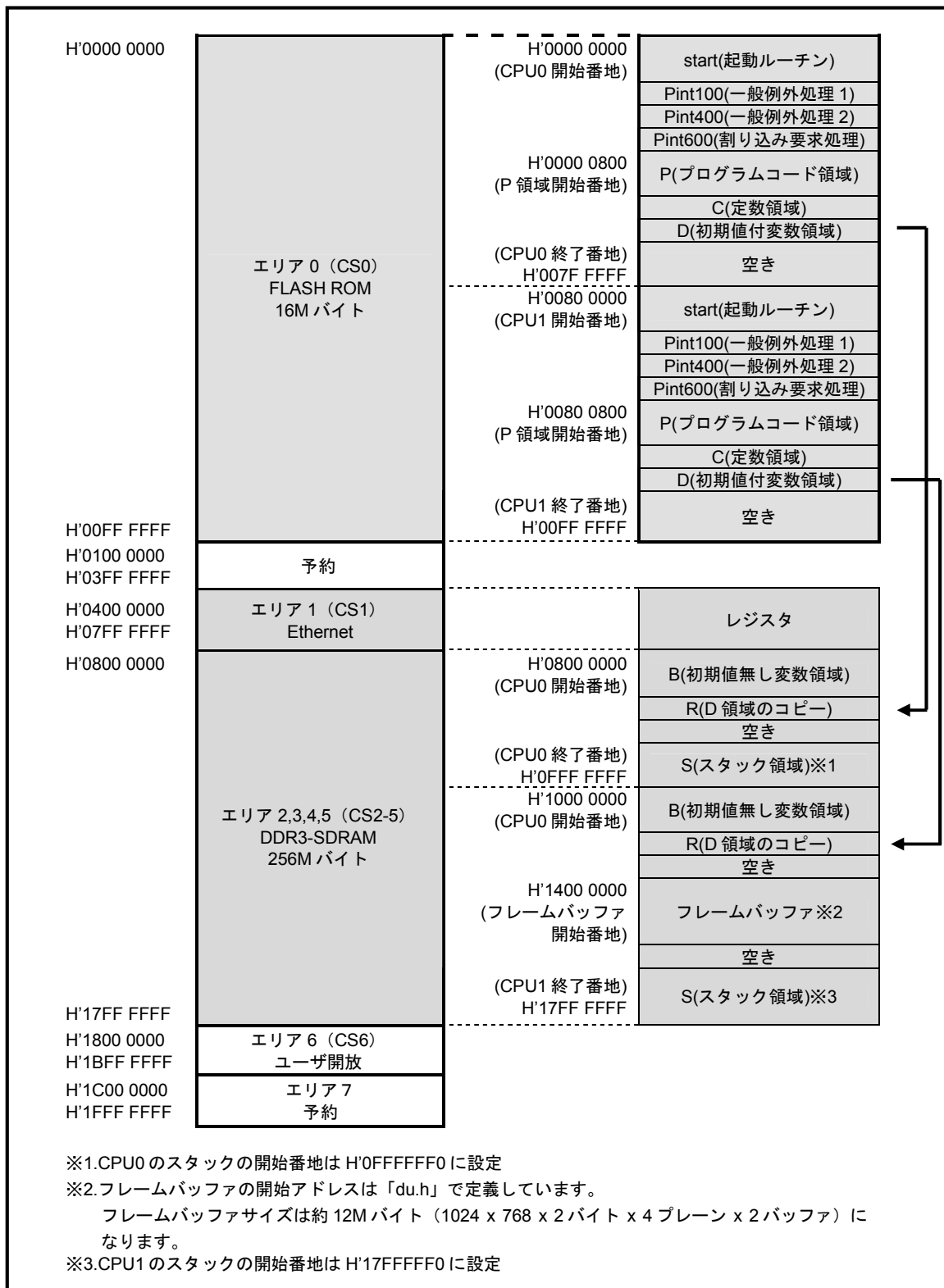
4.3 RAM 動作時のメモリマップ (USB ファンクション)

メモリマップを以下に示します。



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。



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使用される場合は、「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 High-performance Embedded Workshop を起動し、¥sample¥AP-SH4AD-0A_Big¥shc¥cpu0¥ether¥ap_sh4ad_0a_big_cpu0_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_sh4ad_0a_big_cpu0_ether.mot、ap_sh4ad_0a_big_cpu0_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。
- ⑥ High-performance Embedded Workshop を一度終了して、¥sample¥AP-SH4AD-0A_Big¥shc¥cpu1¥ap_sh4ad_0a_big_cpu1.hws を読み込みます。
- ⑦ 上記①～⑤と同様にビルドを行って下さい。

High-performance Embedded Workshop の詳細な使用方法につきましては、High-performance Embedded Workshop 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-SH4AD-0A 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 XsSight を起動し、CPU0 ウィンドウを選択します。
- ③ ¥sample フォルダ直下にある XrossFinder_sh4ad_0a_cpu0.xfc コマンドファイルを読み込みます。
- ④ ¥cpu0¥ether¥ap_sh4ad_0a_big_cpu0_ether¥Debug フォルダ内の ap_sh4ad_0a_big_cpu0_ether.abs または ap_sh4ad_0a_big_cpu0_ether.x を CPU0 にダウンロードします。
- ⑤ XsSight 上で、CPU1 ウィンドウを選択します。
- ⑥ ¥sample フォルダ直下にある XrossFinder_sh4ad_0a_cpu1.xfc コマンドファイルを読み込みます。
- ⑦ ¥cpu1¥ap_sh4ad_0a_big_cpu1¥Debug フォルダ内の ap_sh4ad_0a_big_cpu1.abs または ap_sh4ad_0a_big_cpu1.x を CPU1 にダウンロードして、動作を確認して下さい。

(3) ROM上でのデバッグ

- ① AP-SH4AD-0A 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 XsSight を起動し、CPU0 ウィンドウを選択します。
- ③ ¥cpu0¥ether¥ap_sh4ad_0a_big_cpu0_ether¥Release フォルダ内の ap_sh4ad_0a_big_cpu0_ether.abs または ap_sh4ad_0a_big_cpu0_ether.x を CPU0 に読み込みます。
- ④ ¥sample フォルダ直下にある XrossFinder_sh4ad_0a_cpu0.xfc コマンドファイルを読み込みます。
- ⑤ XsSight 上で、CPU1 ウィンドウを選択します。
- ⑥ ¥cpu1¥ap_sh4ad_0a_big_cpu1¥Release フォルダ内の ap_sh4ad_0a_big_cpu1.abs または ap_sh4ad_0a_big_cpu1.x を CPU1 に読み込みます。
- ⑦ XsSight 上で、CPU0 ウィンドウを選択します。
- ⑧ XsSight のメニューから FlashWriter EX を選択し、下図 Fig5.1-1 のように設定を行ってください。
- ⑨ 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。

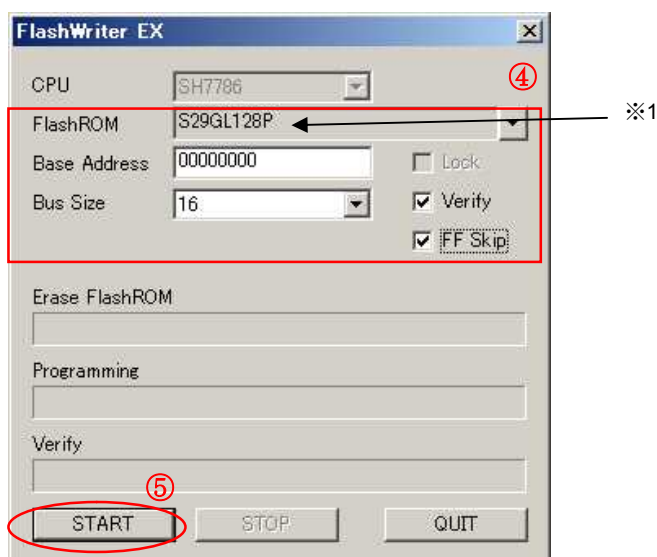


Fig5.1-1 FlashWriter EX の設定

- ※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。

(4) XsSight 未使用時の確認方法

・FlashWriter EXを使用する場合

- ① アダプタ (XrossFinderEvo、XrossFinder または HJ-LINK) を使用して PC とボードを繋ぎます。
- ② AP-SH4AD-0A 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 FlashWriter EX を起動して、「Table5.1-1 FlashWriter EX の設定」を参考に設定を行って下さい。
- ④ FlashWriter EX で、¥sample フォルダ直下にある XrossFinder_sh4ad_0a_cpu0.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 ¥cpu0¥ether¥ap_sh4ad_0a_big_cpu0_ether¥Release フォルダ内の ap_sh4ad_0a_big_cpu0_ether.mot を「File」メニューの「Open」から開きます。その後、¥cpu1¥ap_sh4ad_0a_little_cpu1¥Release フォルダ内の ap_sh4ad_0a_big_cpu1.mot を、「File」メニューの「Marge」から開き、ボードに書き込みを行います。
(cpu0 と cpu1 の両方の mot ファイルを「Open」で開くと、先に読み込んだファイルの情報が消去されてしまいます。両方の mot ファイルをボードに書き込む際は、上記の方法に従い FlashWriter EX のマージ機能を使用するようにしてください。)
- ⑥ AP-SH4AD-0A 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。

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	XrossFinderEvo 使用時は「XrossFinderEvo」 XrossFinder 使用時は「XrossFinder」 HJ-LINK 使用時は「HJ-LINK」
JTAG クロック (XrossFinderEvo/XrossFinder 使用時)	20MHz 以下
CPU	SH7786
BaseAddress	00000000
FLASHROM	S29GL128P ※1
Bus Size	16

Table5.1-1 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。

5.2 動作説明（ネットワーク）

5.2.1 サンプルプログラム概要（ネットワーク）

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。

CPU0 側の動作内容

- 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。
- PCI-Express カードスロットに接続した PCI-Express アドインカードのコンフィグレーション情報を SCIF0 から出力します。
PCI-Express アドインカードによって電源の配給方法が異なります。ご使用になられる PCI-Express アドインカードに合わせて、必要な場合はオプション電源より+12V と+3.3Vaux を配給してください。
- Ethernet 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、「4.2.2 ネットワーク動作」を参照してください。

CPU1 側の動作内容

- LD1（緑の LED）を 500msec 間隔で ON/OFF します。（TMU1 割り込み使用）
- LD2（緑の LED）を 1sec 間隔で ON/OFF します。（TMU2 割り込み使用）
- DVI で接続したモニタに 5sec 間隔で画像を切り替えて表示します。表示画像は XGA サイズ（1024×768）で、切替画面数は 4 画面になります。

4.2.2 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	WindowsXP/Vista/7
LAN ポート	10/100BASE 以上対応の LAN ポート
LAN ケーブル	クロスケーブル 1 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 1 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。

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-2A-XX-XX ※1

※1. XX-XX 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 Ethernet コントローラに接続された EEPROM に格納されています。

EEPROM のデータ構造については Ethernet コントローラ (SMSC 社製 LAN9221) のデータシートをご覧ください。

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。

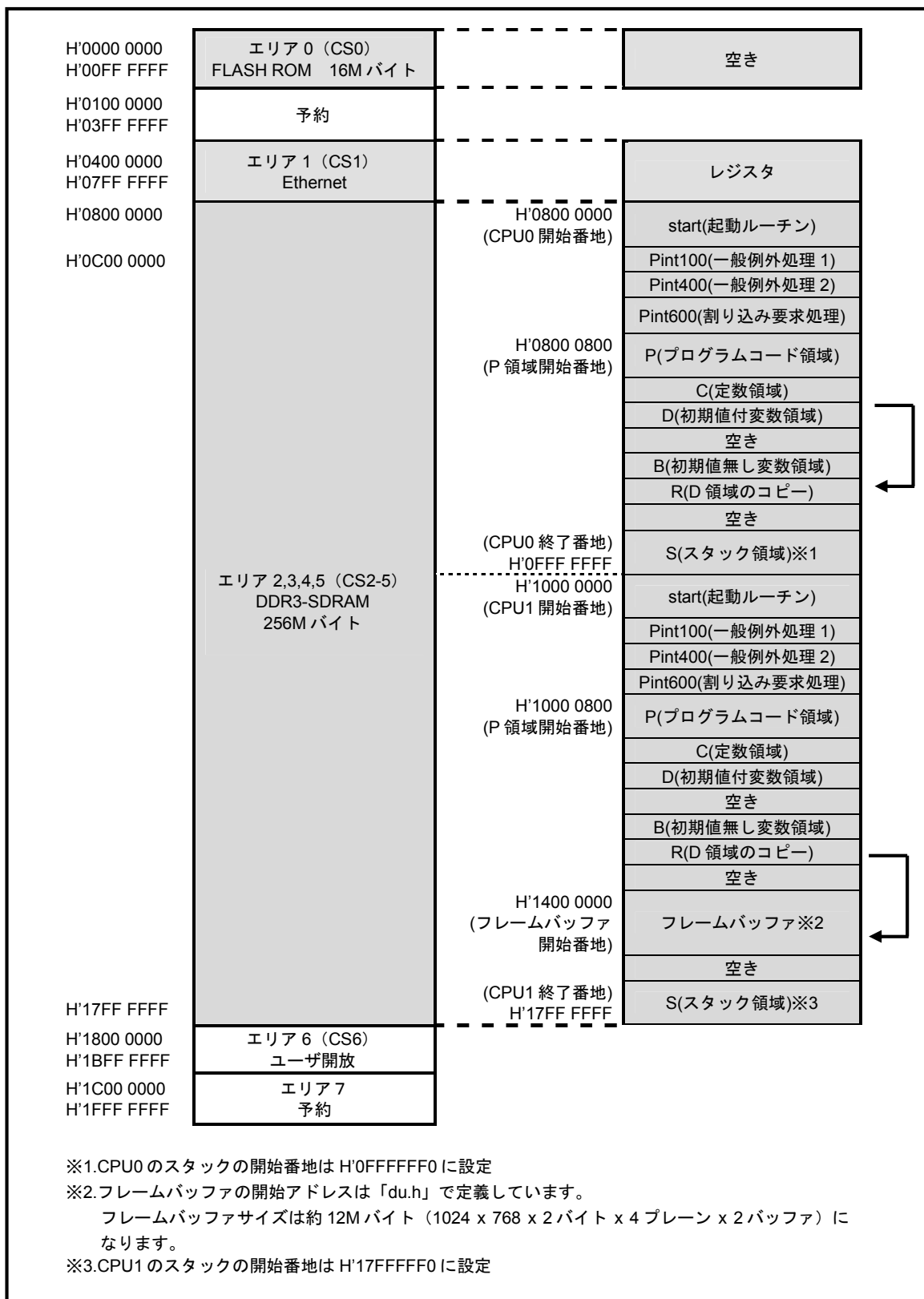
- ① CPU ボードの LAN コネクタ (CN5) とパソコンを LAN クロスケーブルで接続します。
- ② パソコン上でネットワークの設定を行います。
以下に示す内容で設定を行ってください。

IP アドレス	192.168.1.201
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.1.254

- 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 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、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。
- ⑤ 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。

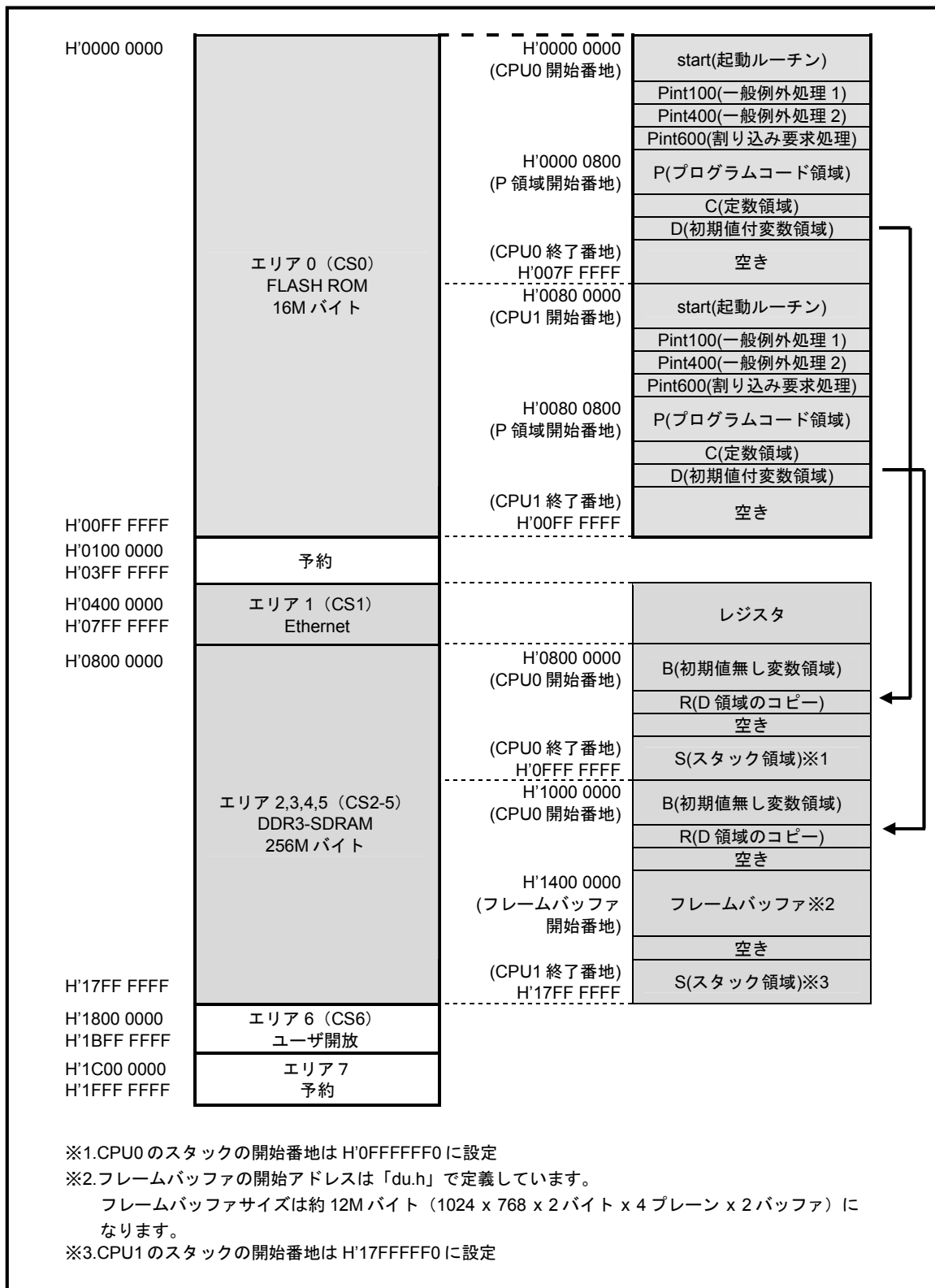
5.3 RAM 動作時のメモリマップ (ネットワーク)

メモリマップを以下に示します。



5.4 ROM 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。



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 OHCI ホスト・USB ファンクションサンプルソースの著作権はルネサスエレクトロニクス株式会社が保有します。
- ・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

- ・ 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。
- ・ SH7786 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。
- ・ High-performance Embedded Workshop 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®7、Windows®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®7 は Windows7 もしくは Win7
- ・ 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