

AP-SH2AD-0A(SH7205 CPU BOARD)

サンプルプログラム解説

6版 2021年02月05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 USB ホスト用サンプルプログラムのファイル構成	6
2.2.2 USB ファンクション用サンプルプログラムのファイル構成	8
3. ビルド・動作確認方法	10
3.1 USB ホスト用サンプルプログラム	10
3.2 USB ファンクション用サンプルプログラム	12
4. 動作説明	14
4.1 USB ホスト用サンプルプログラムの動作説明	14
4.1.1 USB ホスト用サンプルプログラム概要	14
4.1.2 USB ホスト動作	16
4.1.3 USB ホスト用サンプルプログラム API 一覧	17
4.1.4 USB サンプルプログラム注意事項	18
4.2 USB ファンクション用サンプルプログラムの動作説明	19
4.2.1 USB ファンクション用サンプルプログラム概要	19
4.2.2 USB ファンクション動作	21
4.2.3 USB サンプルプログラム注意事項	21
4.3 RAM 動作時のメモリマップ	22
4.4 ROM 動作時のメモリマップ	24

1. 概要

1.1 概要

本アプリケーションノートでは、AP-SH2AD-0A に付属するサンプルプログラムについて解説します。
AP-SH2AD-0A には、下記のサンプルプログラムが付属しています。

サンプルプログラム	動作内容
USB ホスト用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• CAN 通信
USB ファンクション用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• CAN 通信

詳細な動作内容に関しては、後述の「4. 動作説明」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-SH2AD-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-SH2AD-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。

クロックモード	: MODE0
エリア 0 空間バス幅	: 16bit
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
USB ホスト 0 VBUS 設定	: PH7 で行う
USB ホスト 1 VBUS 設定	: PH9 で行う
USB ホスト 0 OVERCURRENT 設定	: PH8 で行う
USB ホスト 1 OVERCURRENT 設定	: PH10 で行う
動作モード	: FlashROM 書き込み時は DEBUG に、 動作確認時は NORMAL に設定してください
USB チャンネル設定	: 使用するサンプルプログラムに合わせてください

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ください。

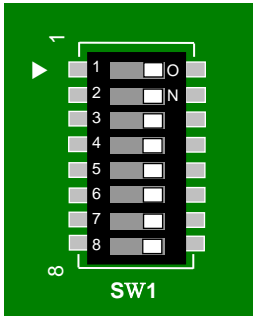
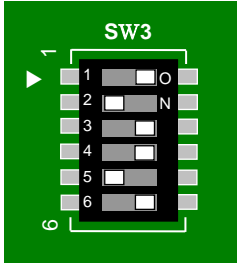
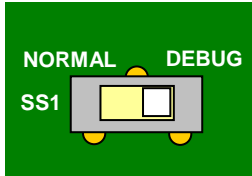
 <p>SW1</p>	<p>クロックモード : MODE0 エリア 0 空間バス幅 : 16bit FlashROM 設定 : FlashROM 使用 FlashROM ライトプロテクト設定 : ライトプロテクト解除 SDRAM 設定 : SDRAM 使用</p>
 <p>SW3</p>	<p>USB ホスト 0 VBUS 設定 : PH7 で行う USB ホスト 1 VBUS 設定 : PH9 で行う USB ホスト 0 OVERCURRENT 設定 : PH8 で行う USB ホスト 1 OVERCURRENT 設定 : PH10 で行う</p>
 <p>SS1</p>	<p>動作モード設定 : DEBUG</p>

Fig1.2-1 FlashROM 書き込み時及び JTAG デバッガ使用時の動作モード設定

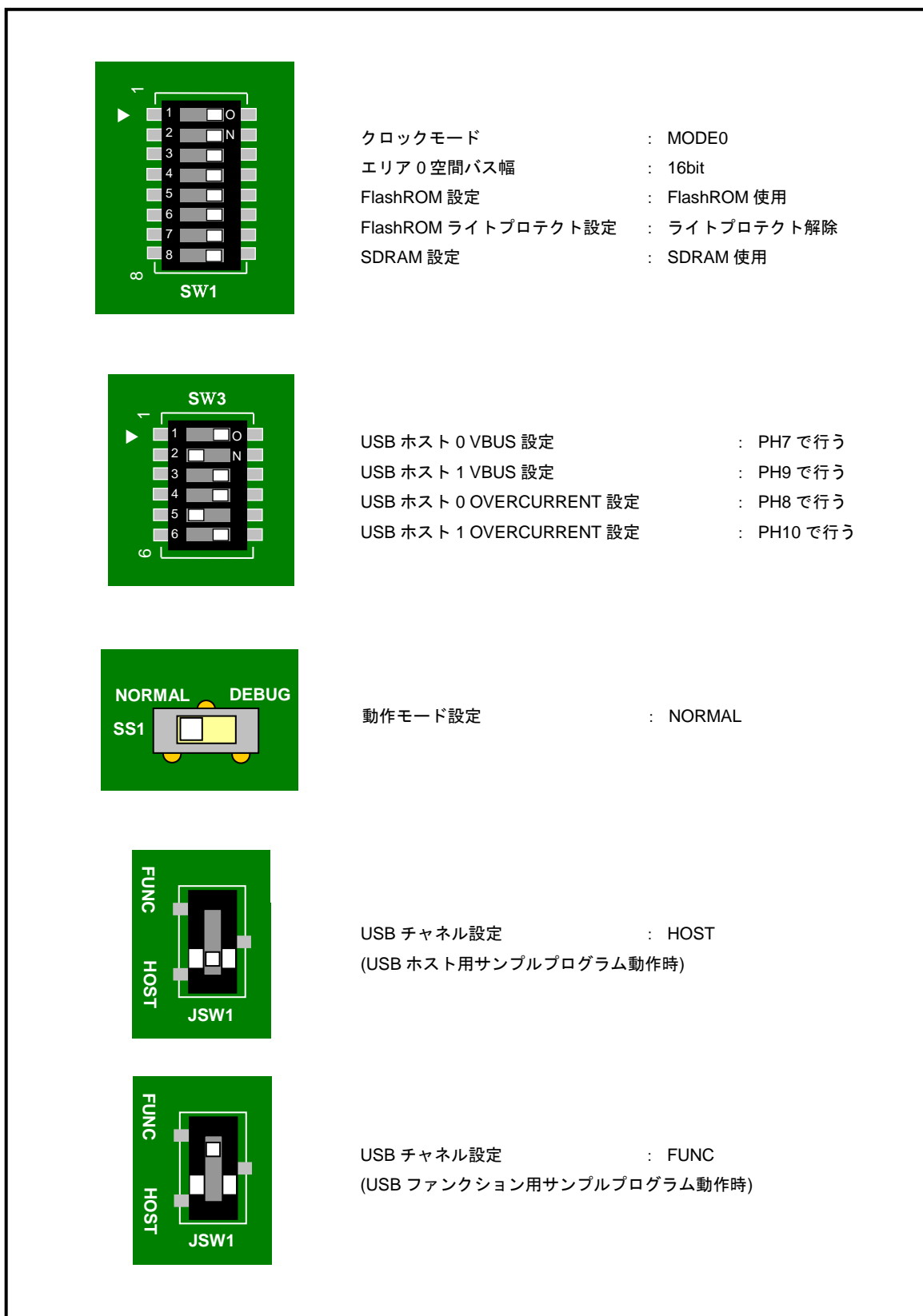


Fig1.2-2 動作確認時の動作モード設定

1.3 開発環境について

本サンプルプログラムは統合開発環境 High-performance Embedded Workshop(以下、「HEW」という)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	HEW	V 3.01.08.000 以降	SHC ^{※1}	V9.0.1.0 以降	SuperHファミリ用C/C++コンパイラパッケージに付属

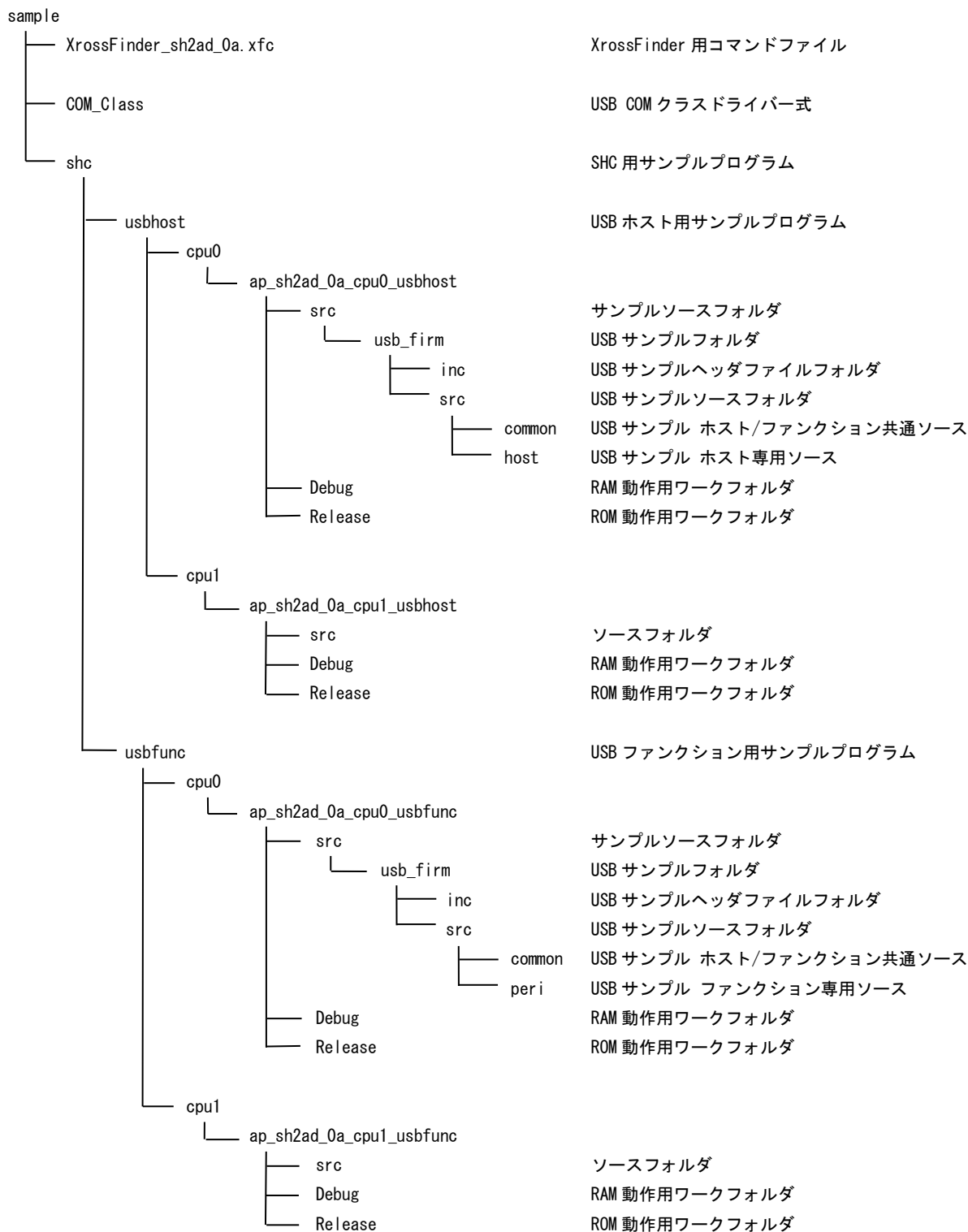
※1 「SuperHファミリ用C/C++コンパイラ」です。ルネサス エレクトロニクス株式会社のウェブサイトより評価版をダウンロードできます。

2. 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。

SH7205 はデュアルコアアーキテクチャを採用しています。その為、サンプルプログラムでは2つのCPU(CPU0 と CPU1)ごとにプロジェクトを作成しています。CPUごとの動作内容に関しては、「4. 動作説明」を参照してください。



2.2 ファイル構成

2.2.1 USB ホスト用サンプルプログラムのファイル構成

(1) CPU0 側

CPU0 側の USB ホスト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%usbhost%cpu0 フォルダ内>

ap_sh2ad_0a_cpu0_usbhost. hws … HEW 用ワークスペースファイル

<shc%usbhost%cpu0%ap_sh2ad_0a_cpu0_usbhost フォルダ内>

ap_sh2ad_0a_cpu0_usbhost. hwp … HEW 用プロジェクトファイル

<shc%usbhost%cpu0%ap_sh2ad_0a_cpu0_usbhost%src フォルダ内>

main. c	…	メイン処理
sci. c	…	シリアル処理
can. c	…	CAN 処理
boot. c	…	CPU 初期化処理
common. h	…	共通ヘッダファイル
7205. h	…	SH7205 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
BordDepend. h	…	ボード依存ヘッダファイル
typedefine. h	…	タイプ定義ファイル
section. src	…	セクション定義ファイル

<shc%usbhost%cpu0%ap_sh2ad_0a_cpu0_usbhost%src%usb_firm フォルダ内>

c_version. h … バージョン情報ファイル

<shc%usbhost%cpu0%ap_sh2ad_0a_cpu0_usbhost%src%usb_firm%src フォルダ内>

c_datatbl. c	…	USB サンプル用データバッファ
h_main. c	…	USB サンプル用メインソース
common	…	USB サンプル ホスト/ファンクション共通ソースフォルダ
host	…	USB サンプル ホスト専用ソースフォルダ

<shc%usbhost%cpu0%ap_sh2ad_0a_cpu0_usbhost%Debug フォルダ内>

ap_sh2ad_0a_cpu0_usbhost. abs	…	RAM 動作用オブジェクトファイル(elf 形式)
ap_sh2ad_0a_cpu0_usbhost. mot	…	RAM 動作用モトローラ S フォーマット形式ファイル
ap_sh2ad_0a_cpu0_usbhost. 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

<shc%usbhost%cpu0%ap_sh2ad_0a_cpu0_usbhost%Release フォルダ内>

ap_sh2ad_0a_cpu0_usbhost. abs	…	ROM 動作用オブジェクトファイル (elf 形式)
ap_sh2ad_0a_cpu0_usbhost. mot	…	ROM 動作用モトローラ S フォーマット形式ファイル
ap_sh2ad_0a_cpu0_usbhost. 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) CPU1 側

CPU1 側の USB ホスト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%usbhost%cpu1 フォルダ内>

ap_sh2ad_0a_cpu1_usbhost. hws … HEW 用ワークスペースファイル

<shc%usbhost%cpu1%ap_sh2ad_0a_cpu1_usbhost フォルダ内>

ap_sh2ad_0a_cpu1_usbhost. hwp … HEW 用プロジェクトファイル

<shc%usbhost%cpu1%ap_sh2ad_0a_cpu1_usbhost%src フォルダ内>

main. c … メイン処理
 timer. c … タイマ処理
 boot. c … CPU 初期化処理
 common. h … 共通ヘッダファイル
 7205. h … SH7205 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
 BordDepend. h … ボード依存ヘッダファイル
 typedefine. h … タイプ定義ファイル
 section. src … セクション定義ファイル

<shc%usbhost%cpu1%ap_sh2ad_0a_cpu1_usbhost%Debug フォルダ内>

ap_sh2ad_0a_cpu1_usbhost. abs … RAM 動作用オブジェクトファイル(elf 形式)
 ap_sh2ad_0a_cpu1_usbhost. mot … RAM 動作用モトローラ S フォーマット形式ファイル
 ap_sh2ad_0a_cpu1_usbhost. 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

<shc%usbhost%cpu1%ap_sh2ad_0a_cpu1_usbhost%Release フォルダ内>

ap_sh2ad_0a_cpu1_usbhost. abs … ROM 動作用オブジェクトファイル (elf 形式)
 ap_sh2ad_0a_cpu1_usbhost. mot … ROM 動作用モトローラ S フォーマット形式ファイル
 ap_sh2ad_0a_cpu1_usbhost. 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

2.2.2 USB ファンクション用サンプルプログラムのファイル構成

(1) CPU0 側

CPU0 側の USB ファンクション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¥usbfunc¥cpu0 フォルダ内>

ap_sh2ad_0a_cpu0_usbfunc. hws … HEW 用ワークスペースファイル

<shc¥usbfunc¥cpu0¥ap_sh2ad_0a_cpu0_usbfunc フォルダ内>

ap_sh2ad_0a_cpu0_usbfunc. hwp … HEW 用プロジェクトファイル

<shc¥usbfunc¥cpu0¥ap_sh2ad_0a_cpu0_usbfunc¥src フォルダ内>

main. c … メイン処理
 sci. c … シリアル処理
 can. c … CAN 処理
 boot. c … CPU 初期化処理
 common. h … 共通ヘッダファイル
 7205. h … SH7205 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
 BordDepend. h … ボード依存ヘッダファイル
 typedefine. h … タイプ定義ファイル
 section. src … セクション定義ファイル

<shc¥usbfunc¥cpu0¥ap_sh2ad_0a_cpu0_usbfunc¥src¥usb_firm フォルダ内>

c_version. h … バージョン情報ファイル

<shc¥usbfunc¥cpu0¥ap_sh2ad_0a_cpu0_usbfunc¥src¥usb_firm¥src フォルダ内>

c_datatbl. c … USB サンプル用データバッファ
 p_main. c … USB サンプル用メインソース
 common … USB サンプル ホスト/ファンクション共通ソースフォルダ
 peri … USB サンプル ファンクション専用ソースフォルダ

<shc¥usbfunc¥cpu0¥ap_sh2ad_0a_cpu0_usbfunc¥Debug フォルダ内>

ap_sh2ad_0a_cpu0_usbfunc. abs … RAM 動作オブジェクトファイル(elf 形式)
 ap_sh2ad_0a_cpu0_usbfunc. mot … RAM 動作モトローラ S フォーマット形式ファイル
 ap_sh2ad_0a_cpu0_usbfunc. map … 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

<shc¥usbfunc¥cpu0¥ap_sh2ad_0a_cpu0_usbfunc¥Release フォルダ内>

ap_sh2ad_0a_cpu0_usbfunc. abs … ROM 動作オブジェクトファイル(elf 形式)
 ap_sh2ad_0a_cpu0_usbfunc. mot … ROM 動作モトローラ S フォーマット形式ファイル
 ap_sh2ad_0a_cpu0_usbfunc. map … 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

(1) CPU1 側

CPU1 側の USB ファンクション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%usbfunc%cpu1 フォルダ内>

ap_sh2ad_0a_cpu1_usbfunc. hws … HEW 用ワークスペースファイル

<shc%usbfunc%cpu1%ap_sh2ad_0a_cpu1_usbfunc フォルダ内>

ap_sh2ad_0a_cpu1_usbfunc. hwp … HEW 用プロジェクトファイル

<shc%usbfunc%cpu1%ap_sh2ad_0a_cpu1_usbfunc%src フォルダ内>

main. c … メイン処理
 timer. c … タイマ処理
 boot. c … CPU 初期化処理
 common. h … 共通ヘッダファイル
 7205. h … SH7205 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
 BordDepend. h … ボード依存ヘッダファイル
 typedefine. h … タイプ定義ファイル
 section. src … セクション定義ファイル

<shc%usbfunc%cpu1%ap_sh2ad_0a_cpu1_usbfunc%Debug フォルダ内>

ap_sh2ad_0a_cpu1_usbfunc. abs … RAM 動作用オブジェクトファイル(elf 形式)
 ap_sh2ad_0a_cpu1_usbfunc. mot … RAM 動作用モトローラ S フォーマット形式ファイル
 ap_sh2ad_0a_cpu1_usbfunc. 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

<shc%usbfunc%cpu1%ap_sh2ad_0a_cpu1_usbfunc%Release フォルダ内>

ap_sh2ad_0a_cpu1_usbfunc. abs … ROM 動作用オブジェクトファイル (elf 形式)
 ap_sh2ad_0a_cpu1_usbfunc. mot … ROM 動作用モトローラ S フォーマット形式ファイル
 ap_sh2ad_0a_cpu1_usbfunc. 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

3. ビルド・動作確認方法

3.1 USB ホスト用サンプルプログラム

(1) ビルド

- ① HEW を起動し、ap_sh2ad_0a_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」を選択してください。
- 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ください。
- ④ [Build]ボタン横のリストボックス [Configuration Section]から、[Release]を選択します。
¥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] を実行してください。ap_sh2ad_0a_cpu0_usbhost.mot、ap_sh2ad_0a_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 HEW を一度終了して、再度起動した後に ap_sh2ad_0a_cpu1_usbhost.hws を読み込みます。
- ⑦ 上記②～⑤と同様にビルドを行ってください。

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ください。

JTAG デバッガを使用し、RAM 上でデバッグをされる場合には、上記④の際に、リストボックス「Configuration Section」から、「Debug」を選択し、ビルドしてください。これにより、RAM 上で動作するオブジェクトが生成されます。
なお、この場合、オブジェクトは、¥Release フォルダ内ではなく、¥Debug フォルダ内に出力されます。

(2) プログラムの書き込み

- ① AP-SH2AD-0A のスイッチを「1.2 動作モード」「Fig1.2-1 FlashROM 書き込み時の動作モード設定」に合わせて設定します。
- ② アダプタ (HJ-LINK / XrossFinder / XrossFinder Evo) を使用して PC とボードを繋ぎます。
- 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ください。
- ④ FlashWriter EX で、¥sample フォルダ直下にある XrossFinder_sh2ad_0a.xfc コマンドファイルを使用するように設定してください。
- ⑤ shc¥usbhost¥cpu0¥ap_sh2ad_0a_cpu0_usbhost¥Release フォルダ内の ap_sh2ad_0a_cpu0_usbhost.mot を「File」メニューの「Open」から開きます。その後、「File」メニューの「Merge」から shc¥usbhost¥cpu1¥ap_sh2ad_0a_cpu1_usbhost¥Release フォルダ内の ap_sh2ad_0a_cpu1_usbhost.mot を開き、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。)

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ください。

アダプタ設定	XrossFinder Evo 使用時は「XrossFinder Evo」 XrossFinder 使用時は「XrossFinder」 HJ-LINK 使用時は「HJ-LINK」
JTAG クロック (XrossFinder / XrossFinder Evo 使用時)	10MHz 以下
CPU	SH7205
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ください。

(3) 動作確認

- ① AP-SH2AD-0A のスイッチを「1.2 動作モード」「Fig1.2-2 動作確認時の動作モード設定」に合わせて設定します。
- ② 電源を投入し、サンプルプログラムの動作を確認してください。

3.2 USB ファンクション用サンプルプログラム

(1) ビルド

- ① HEW を起動し、ap_sh2ad_0a_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」を選択してください。
- 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ください。
- ④ [Build] ボタン横のリストボックス [Configuration Section] から、[Release] を選択します。
¥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] を実行してください。ap_sh2ad_0a_cpu0_usbfunc.mot、ap_sh2ad_0a_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 HEW を一度終了して、再度起動した後に ap_sh2ad_0a_cpu1_usbfunc.hws を読み込みます。
- ⑦ 上記②～⑤と同様にビルドを行ってください。

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ください。

JTAG デバッガを使用し、RAM 上でデバッグをされる場合には、上記④の際に、リストボックス「Configuration Section」から、「Debug」を選択し、ビルドしてください。これにより、RAM 上で動作するオブジェクトが生成されます。
なお、この場合、オブジェクトは、¥Release フォルダ内ではなく、¥Debug フォルダ内に出力されます。

(2) プログラムの書き込み

- ① AP-SH2AD-0A のスイッチを「1.2 動作モード」「Fig1.2-1 FlashROM 書き込み時の動作モード設定」に合わせて設定します。
- ② アダプタ (HJ-LINK / XrossFinder / XrossFinder Evo) を使用して PC とボードを繋ぎます。
- ③ FlashWriter EX を起動して、「Table3.2-1 FlashWriter EX の設定」を参考に設定を行ってください。
- ④ FlashWriter EX で、¥sample フォルダ直下にある XrossFinder_sh2ad_0a.xfc コマンドファイルを使用するように設定してください。
- ⑤ shc¥usbfunc¥cpu0¥ap_sh2ad_0a_cpu0_usbfunc¥Release フォルダ内の ap_sh2ad_0a_cpu0_usbfunc.mot を「File」メニューの「Open」から開きます。その後、「File」メニューの「Merge」から shc¥usbfunc¥cpu1¥ap_sh2ad_0a_cpu1_usbfunc¥Release フォルダ内の ap_sh2ad_0a_cpu1_usbfunc.mot を開き、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。)

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ください。

アダプタ設定	XrossFinder Evo 使用時は「XrossFinder Evo」 XrossFinder 使用時は「XrossFinder」 HJ-LINK 使用時は「HJ-LINK」
JTAG クロック (XrossFinder / XrossFinder Evo 使用時)	10MHz 以下
CPU	SH7205
BaseAddress	00000000
FlashROM	S29GL128P ※1
Bus Size	16

Table3.2-1 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。

(3) 動作確認

- ① AP-SH2AD-0A のスイッチを「1.2 動作モード」「Fig1.2-2 動作確認時の動作モード設定」に合わせて設定します。
- ② 電源を投入し、サンプルプログラムの動作を確認してください。

4. 動作説明

4.1 USB ホスト用サンプルプログラムの動作説明

4.1.1 USB ホスト用サンプルプログラム概要

USB ホスト用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

- ROM 動作の場合、起動時に Flash ROM 内のプログラムやデータを読み出して SDRAM にコピーします。
その後、SDRAM 上でアプリケーションを開始します。
- SCIF (SCIO) でエコーバックを行います。(送受信割り込み使用)
SCIF から受信した値をそのまま、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、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) を使用して行ってください。
- CAN I/F でエコーバックを行います。以下の設定で、受信したデータをそのまま送信します。
CAN の設定は、送信 ID:B' 10101010100、受信 ID:B' 10101010101、スタンダードフォーマット、
データフレーム、データ長 1byte、通信速度 490kbps (TSG1=10 (11tq), TSE2=4 (5tq), SJW=0, BSP=0, BRP=1) です。
- USB ホストに USB メモリを接続すると、SCIF (SCIO) から接続状況とデバイス情報を出力します。 ※1
※1. USB ホスト動作の詳細は、「4.1.2 USB ホスト動作」を参照してください。

CPU1 側の動作内容

- ROM 動作の場合、起動時に Flash ROM 内のプログラムやデータを読み出して SDRAM にコピーします。
その後、SDRAM 上でアプリケーションを開始します。
- LD1 (緑の LED) を 500msec 間隔で ON/OFF します。(CMT0 割り込み使用)
- LD2 (緑の LED) を 1000msec 間隔で ON/OFF します。(CMT1 割り込み使用)
- CN2、CN4 のポートより方形波を出力します。周期とピン番号を次のページに示します。

CN2 方形波出力端子一覧

ピン番号	ピン名	周期	備考
19	PJ12/VIDATA7/#SCS1/FRB	20msec	CMT0 使用
20	PJ11/VIDATA6/SS01	20msec	CMT0 使用
21	PJ10/VIDATA5/SSI1/AUDATA3	20msec	CMT0 使用
22	PJ9/VIDATA4/SSCK1/AUDATA2	20msec	CMT0 使用
23	PJ8/VIDATA3/TIOC1B/NAF7/AUDATA1	20msec	CMT0 使用
24	PJ7/VIDATA2/TIOC1A/NAF6/AUDATA0	20msec	CMT0 使用
25	PJ6/VIDATA1/#TEND3/FCE/AUDSYNC	20msec	CMT0 使用
26	PJ5/VIDATA0/#DACK3/#DACT3/FSC/ TxD4/AUDCK	20msec	CMT0 使用
27	PJ4/VICLK/DREQ3/FOE/RxD4	20msec	CMT0 使用
28	PJ3/IRQ7/TIOC0D/TxD3	20msec	CMT0 使用
29	PJ2/IRQ6/TIOC0C/RxD3	20msec	CMT0 使用
30	PJ1/IRQ5/TIOC0B	20msec	CMT0 使用
31	PJ0/IRQ4/TIOC0A	20msec	CMT0 使用

信号名に#がついているものは負論理を表します。

CN4 方形波出力端子一覧

ピン番号	ピン名	周期	備考
25	PC10/DIRECTION/#TEND1/#FCE/PINT6	10msec	CMT1 使用
26	PC9/IDERST#/#DACK1/NAF7 /PINT5/#DACT1	10msec	CMT1 使用
27	PC8/IDEINT/DREQ1/NAF6/PINT4	10msec	CMT1 使用
28	PC7/IDEIORDY/TIOC4D/NAF5/PINT3	10msec	CMT1 使用
29	PC6/IDEIORD#/#TIOC4C/NAF4/PINT2	10msec	CMT1 使用
30	PC5/IDEIOWR#/#TIOC4B/NAF3/PINT1	10msec	CMT1 使用
31	PC4/IODREQ/TIOC4A/NAF2/PINT0	10msec	CMT1 使用
32	PC3/IODACK#/#TCLKD/NAF1/IRQ3	10msec	CMT1 使用
33	PC2/IDEA2/TCLKC/NAF0/IRQ2	10msec	CMT1 使用
34	PC1/IDEA1/TCLKB/FSC/IRQ1	10msec	CMT1 使用
35	PC0/IDEA0/TCLKA/FOE/IRQ0	10msec	CMT1 使用

信号名に#がついているものは負論理を表します。

4.1.2 USB ホスト動作

以下の手順に従い、USB ホスト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(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、フロー制御なしです。
- 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 電源を投入後、ターミナルソフト上に現在の接続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。

```
Detach Port 0-----
Detach Port 1-----
```

- ⑤ USB ホストポート(CN8)のポート 0 (上側) に USB メモリを挿入します。
- ⑥ 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。

```
Attach Port 0-----
Hi-Speed Device----
USB Address is 1----

Get configuration (Device descriptor) -----
PASS!
Device descriptor dump :
    12 01 00 02 00 00 00 40 8f 05 87 63 41 01 01 02
    03 01

Device descriptor fields :
    bLength      : 0x12
                :
                :
```

※. 以下に、デバイスのディスクリプタ情報が表示されます。

上記の表示も含め、表示される情報は接続する USB メモリにより異なります。

- ⑦ 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。

```
Detach Port 0-----
```

- ⑧ USB ホストポート(CN8)のポート1（下側）にUSBメモリを挿入します。
- ⑨ 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。

```

Attach Port 1-----
Hi-Speed Device----
USB Address is 2----

Get configuration (Device descriptor) -----
PASS!
Device descriptor dump :
    12 01 00 02 00 00 00 40 8f 05 87 63 41 01 01 02
    03 01

Device descriptor fields :
    bLength      : 0x12      .
                  .
                  .
    
```

※. 以下に、デバイスのディスクリプタ情報が表示されます。
上記の表示も含め、表示される情報は接続するUSBメモリにより異なります。

- ⑩ 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。

```

Detach Port 1-----
    
```

- ⑪ 以上でUSBホストの動作は終了です。

4.1.3 USBホスト用サンプルプログラムAPI一覧

USBホスト用サンプルプログラムの主要API一覧は以下の通りです。

API名	内容説明
H_ModeInit	USBホストドライバ初期化
H_DataIn	データイントランザクション実行
H_DataOut	データアウトトランザクション実行
H_CtrlTransfer	コントロール転送実行

4.1.4 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 ホスト/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（CMT、MTU2 等）をご使用ください。

4.2 USB ファンクション用サンプルプログラムの動作説明

4.2.1 USB ファンクション用サンプルプログラム概要

USB ファンクション用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

- ROM 動作の場合、起動時に Flash ROM 内のプログラムやデータを読み出して SDRAM にコピーします。
その後、SDRAM 上でアプリケーションを開始します。
- SCIF (SCIO) でエコーバックを行います。(送受信割り込み使用)
SCIF から受信した値をそのまま、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、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) を使用して行ってください。
- CAN I/F でエコーバックを行います。以下の設定で、受信したデータをそのまま送信します。
CAN の設定は、送信 ID:B' 10101010100、受信 ID:B' 10101010101、スタンダードフォーマット、
データフレーム、データ長 1byte、通信速度 490kbps (TSG1=10 (11tq), TSE2=4 (5tq), SJW=0, BSP=0, BRP=1) です。
- 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

- ROM 動作の場合、起動時に Flash ROM 内のプログラムやデータを読み出して SDRAM にコピーします。
その後、SDRAM 上でアプリケーションを開始します。
- LD1 (緑の LED) を 500msec 間隔で ON/OFF します。(CMT0 割り込み使用)
- LD2 (緑の LED) を 1000msec 間隔で ON/OFF します。(CMT1 割り込み使用)
- CN2、CN4 のポートより方形波を出力します。周期とピン番号を次のページに示します。

CN2 方形波出力端子一覧

ピン番号	ピン名	周期	備考
19	PJ12/VIDATA7/#SCS1/FRB	20msec	CMT0 使用
20	PJ11/VIDATA6/SS01	20msec	CMT0 使用
21	PJ10/VIDATA5/SSI1/AUDATA3	20msec	CMT0 使用
22	PJ9/VIDATA4/SSCK1/AUDATA2	20msec	CMT0 使用
23	PJ8/VIDATA3/TIOC1B/NAF7/AUDATA1	20msec	CMT0 使用
24	PJ7/VIDATA2/TIOC1A/NAF6/AUDATA0	20msec	CMT0 使用
25	PJ6/VIDATA1/#TEND3/FCE/AUDSYNC	20msec	CMT0 使用
26	PJ5/VIDATA0/#DACK3/#DACT3/FSC/ TxD4/AUDCK	20msec	CMT0 使用
27	PJ4/VICLK/DREQ3/FOE/RxD4	20msec	CMT0 使用
28	PJ3/IRQ7/TIOC0D/TxD3	20msec	CMT0 使用
29	PJ2/IRQ6/TIOC0C/RxD3	20msec	CMT0 使用
30	PJ1/IRQ5/TIOC0B	20msec	CMT0 使用
31	PJ0/IRQ4/TIOC0A	20msec	CMT0 使用

信号名に#がついているものは負論理を表します。

CN4 方形波出力端子一覧

ピン番号	ピン名	周期	備考
25	PC10/DIRECTION/#TEND1/#FCE/PINT6	10msec	CMT1 使用
26	PC9/IDERST#/#DACK1/NAF7 /PINT5/#DACT1	10msec	CMT1 使用
27	PC8/IDEINT/DREQ1/NAF6/PINT4	10msec	CMT1 使用
28	PC7/IDEIORDY/TIOC4D/NAF5/PINT3	10msec	CMT1 使用
29	PC6/IDEIORD#/#TIOC4C/NAF4/PINT2	10msec	CMT1 使用
30	PC5/IDEIOWR#/#TIOC4B/NAF3/PINT1	10msec	CMT1 使用
31	PC4/IODREQ/TIOC4A/NAF2/PINT0	10msec	CMT1 使用
32	PC3/IODACK#/#TCLKD/NAF1/IRQ3	10msec	CMT1 使用
33	PC2/IDEA2/TCLKC/NAF0/IRQ2	10msec	CMT1 使用
34	PC1/IDEA1/TCLKB/FSC/IRQ1	10msec	CMT1 使用
35	PC0/IDEA0/TCLKA/FOE/IRQ0	10msec	CMT1 使用

信号名に#がついているものは負論理を表します。

4.2.2 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。

Win10 よりも前の OS での USB ファンクション動作確認は、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 (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 (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。

4.2.3 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 ホスト/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。

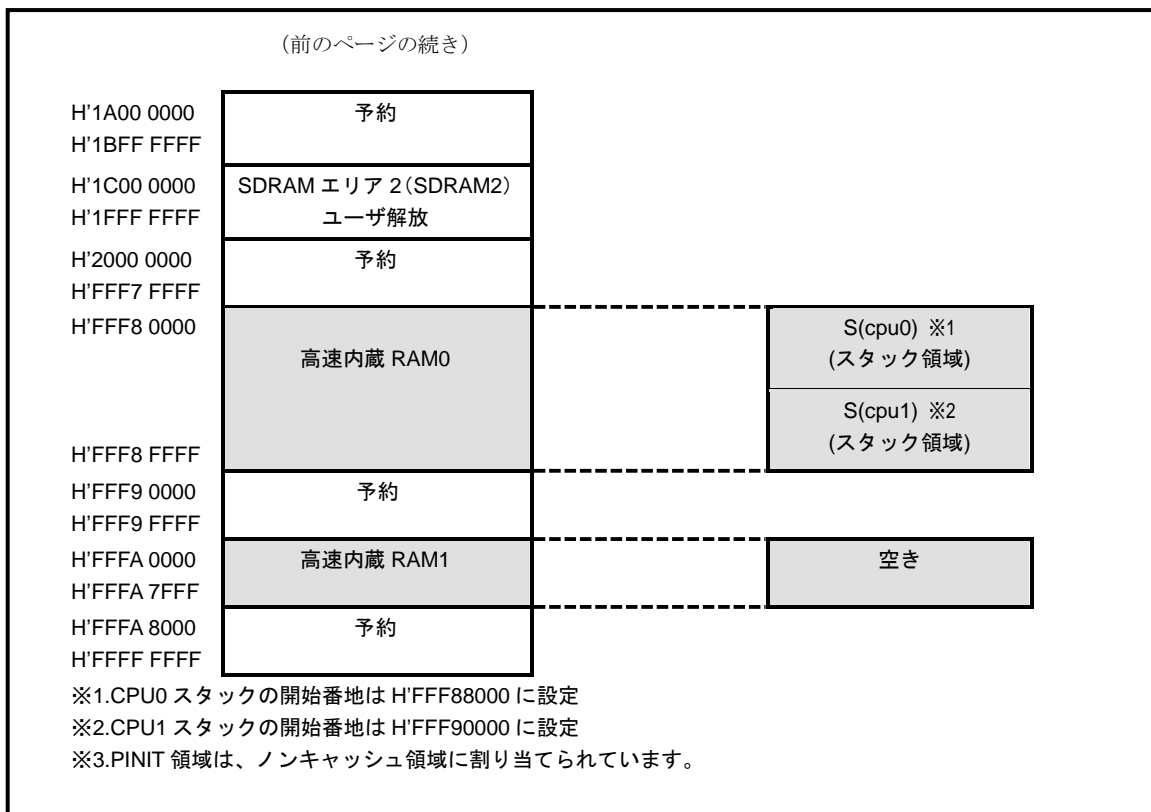
正確に 1msec の待ち時間が必要な場合には、上記の関数のループ回数を調整していただくか、タイマ (CMT、MTU2 等) をご使用ください。

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。

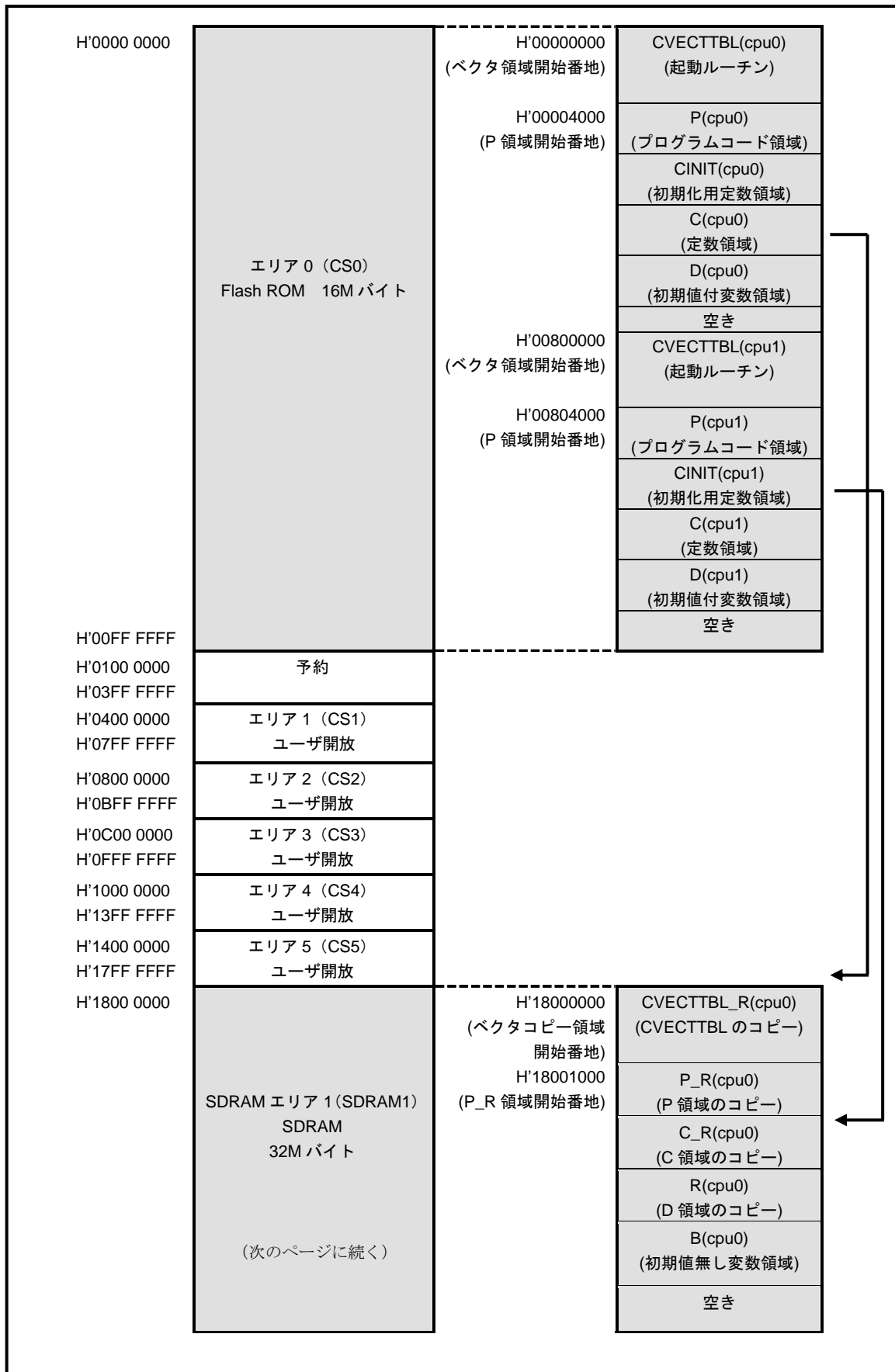
H'0000 0000	エリア 0 (CS0)		空き
H'00FF FFFF	Flash ROM 16M バイト		
H'0100 0000	予約		
H'03FF FFFF			
H'0400 0000	エリア 1 (CS1)		
H'07FF FFFF	ユーザ開放		
H'0800 0000	エリア 2 (CS2)		
H'0BFF FFFF	ユーザ開放		
H'0C00 0000	エリア 3 (CS3)		
H'0FFF FFFF	ユーザ開放		
H'1000 0000	エリア 4 (CS4)		
H'13FF FFFF	ユーザ開放		
H'1400 0000	エリア 5 (CS5)		
H'17FF FFFF	ユーザ開放		
H'1800 0000	SDRAM エリア 1 (SDRAM1) SDRAM 32M バイト	H'18000000 (ベクタ領域開始番地)	CVECTTBL(cpu0) (起動ルーチン)
		H'18004000 (P 領域開始番地)	P(cpu0) (プログラムコード領域)
			C(cpu0) (定数領域)
			D(cpu0) (初期値付変数領域)
			空き
		H'18800000 (B 領域開始番地)	R(cpu0) (D 領域のコピー)
			B(cpu0) (初期値無し変数領域)
			空き
		H'19000000 (ベクタ領域開始番地)	CVECTTBL(cpu1) (起動ルーチン)
		H'19004000 (P 領域開始番地)	P(cpu1) (プログラムコード領域)
			C(cpu1) (定数領域)
			D(cpu1) (初期値付変数領域)
			空き
		H'19800000 (B 領域開始番地)	R(cpu1) (D 領域のコピー)
			B(cpu1) (初期値無し変数領域)
H'19FF FFFF			

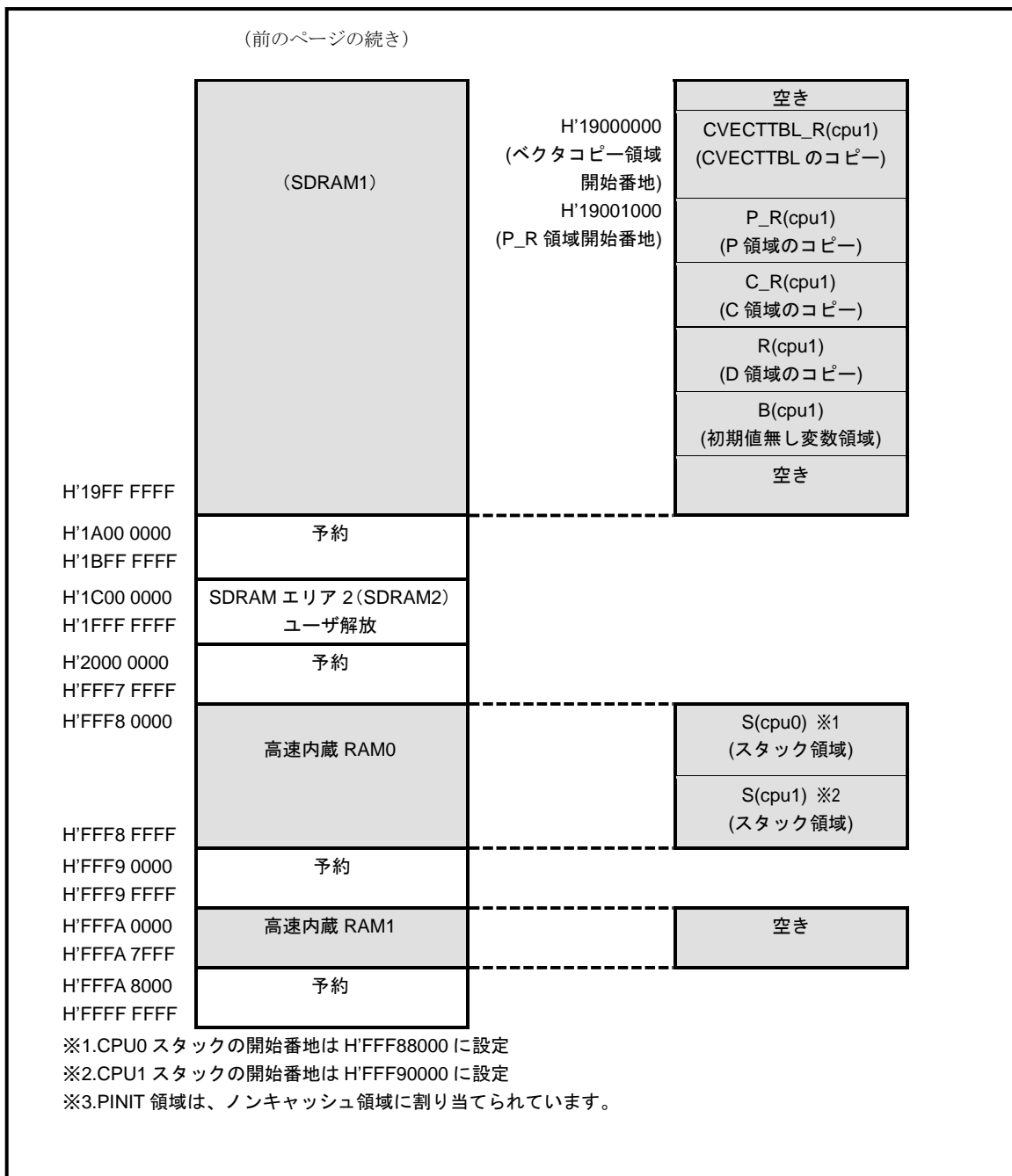
(次のページに続く)



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。





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 部分の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 エレクトロニクス株式会社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 エレクトロニクス株式会社では一切責任を負いませんのでご了承ください。
- ・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、誤りなどお気付きの点がありましたら弊社までご連絡ください。
- ・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

- ・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®10、Windows®8、Windows®7、Windows®XP 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。ご了承ください。

Windows®10 は Windows 10 もしくは Win10

Windows®8 は Windows 8 もしくは Win8

Windows®7 は Windows 7 もしくは Win7

Windows®XP は Windows XP もしくは WinXP

High-performance Embedded Workshop は HEW

- ・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、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。

- ・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。

 **ALPHA PROJECT**
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 8 3 4
<https://www.apnet.co.jp>
E-Mail : query@apnet.co.jp