

AP-SH2A-6A(SH7269 CPU BOARD)

サンプルプログラム解説

3.1版 2023年10月02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
1.4 ワークスペースについて.....	4
1.5 対応 OS 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
3. USB ファンクションサンプルプログラム.....	10
3.1 ビルド・デバッグ方法（USB ファンクション）	10
3.2 動作説明（USB ファンクション）	12
3.2.1 サンプルプログラム概要（USB ファンクション）	12
3.2.2 USB ファンクション動作.....	14
3.3 RAM 動作時のメモリマップ（USB ファンクション）	15
3.4 ROM 動作時のメモリマップ（USB ファンクション）	16
4. USB ホストサンプルプログラム	17
4.1 ビルド・デバッグ方法（USB ホスト）	17
4.2 動作説明（USB ホスト）	19
4.2.1 サンプルプログラム概要（USB ホスト）	19
4.2.2 USB ホスト動作	21
4.3 RAM 動作時のメモリマップ（USB ホスト）	22
4.4 ROM 動作時のメモリマップ（USB ホスト）	23

1. 概要

1.1 概要

本アプリケーションノートでは、AP-SH2A-6A に付属するサンプルプログラムについて解説します。
AP-SH2A-6A には、SHC 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

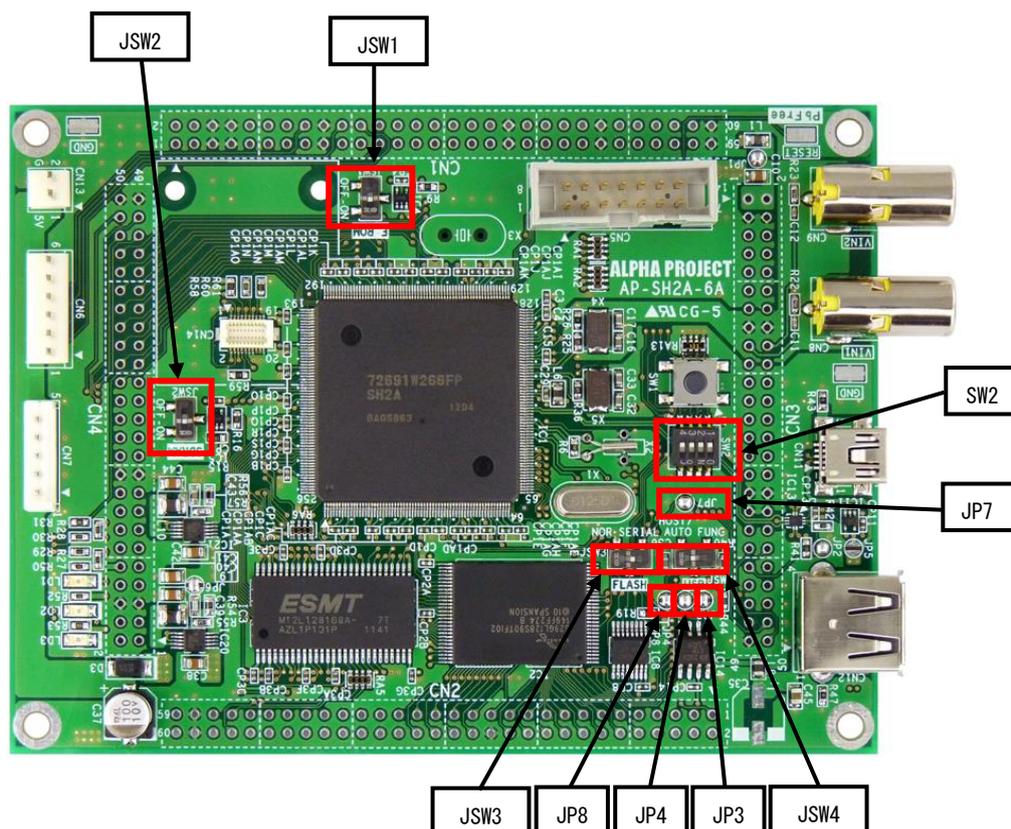
サンプルプログラム	動作内容
USB ファンクションサンプルプログラム	<ul style="list-style-type: none">・ USB ファンクション 仮想シリアル通信・ シリアル通信・ CAN 通信・ タイマ割り込み
USB ホストサンプルプログラム	<ul style="list-style-type: none">・ USB ホスト・ シリアル通信・ CAN 通信・ タイマ割り込み

1.2 動作モード

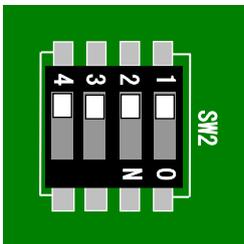
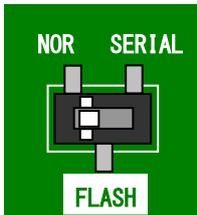
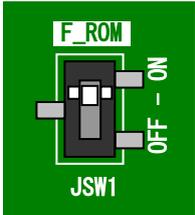
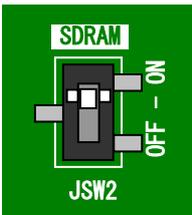
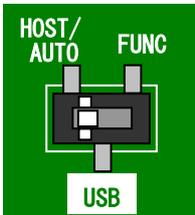
本サンプルプログラムは、AP-SH2A-6A で動作します。CPU 動作モード、各メモリ設定は下記のようになっています。
モードの設定方法等につきましては、「AP-SH2A-6A ハードウェアマニュアル」をご覧ください。
なお、下記以外の条件で動作させる場合には、ソースファイルやコンパイラオプションなどを変更する必要があります。

ブートモード : ブートモード0 (CS0 16bit ブート)
SSCG : SSCG OFF
SDRAM 設定 : SDRAM を使用する
FLASHROM 設定 : NOR FLASHROM を使用する

f



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ください。
JP3、JP4、JP7、JP8 は短絡されている状態とします。

<p>・ SW2</p>		<p><SW2 設定> ブートモード : CS0(16bit)ブート SSCG : SSCG 動作 OFF</p>
<p>・ JSW3</p>		<p><JSW3 設定> FLASHROM 選択 : NOR FlashROM を使用</p>
<p>・ JSW1</p>		<p><JSW1 設定> ボード上の NOR FLASHROM : 使用する</p>
<p>・ JSW2</p>		<p><JSW2 設定> ボード上の SDRAM : 使用する</p>
<p>・ JSW4</p>		<p><JSW4 設定> USB ポートの選択 : AUTO (PJ28 を用いる)</p>

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います。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。

開発環境	バージョン	コンパイラ名	バージョン	備考
HEW	V 4.09.01.007 以降	SHC ^{※1}	V9.4.0.0 以降	SuperH ファミリー用 C/C++コンパイラ パッケージに付属

※1 「SuperH ファミリー用 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。

1.4 ワークスペースについて

本サンプルプログラムの統合開発環境 HEW 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	フォルダ
USB ファンクションサンプルプログラム	¥Sample¥ap_sh2a_6a_usbfunc¥ap_sh2a_6a_usbfunc. hws
USB ホストサンプルプログラム	¥Sample¥ap_sh2a_6a_usbhost¥ap_sh2a_6a_usbhost. hws

1.5 対応 OS について

本サンプルプログラムは以下の OS に対応しています。

- ・ Windows10 (32Bit / 64Bit)
- ・ Windows11 (32Bit / 64Bit)

それぞれの OS に対応する USB ファンクションサンプルプログラム用 USB シリアルドライバのインストール方法に関しては、「3.2.2 USB ファンクション動作」の該当する項目を参照してください。

2. サンプルプログラムの構成

2.1 フォルダ構成

サンプルプログラムは下記のようなフォルダ構成になっています。



2.2 ファイル構成

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。

<¥Sample フォルダ内>

ap_sh2a_6a_usbfunc	...	USB ファンクションサンプルプログラムフォルダ
ap_sh2a_6a_usbhost	...	USB ホストサンプルプログラムフォルダ
COM_Class	...	COM クラスドライバフォルダ
XrossFinder_sh2a_6a.xfc	...	XrossFinder 用コマンドファイル

<¥Sample¥ap_sh2a_6a_usbfunc フォルダ内>

ap_sh2a_6a_usbfunc.hws	...	USB ファンクション HEW 用ワークスペースファイル
------------------------	-----	------------------------------

<¥Sample¥ap_sh2a_6a_usbfunc¥ap_sh2a_6a_usbfunc フォルダ内>

ap_sh2a_6a_usbfunc.hwp	...	USB ファンクション HEW 用プロジェクトファイル
------------------------	-----	-----------------------------

<¥Sample¥ap_sh2a_6a_usbfunc¥ap_sh2a_6a_usbfunc¥Debug フォルダ内>

ap_sh2a_6a_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_sh2a_6a_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_sh2a_6a_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

<¥Sample¥ap_sh2a_6a_usbfunc¥ap_sh2a_6a_usbfunc¥Release フォルダ内>

ap_sh2a_6a_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_sh2a_6a_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_sh2a_6a_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

<¥Sample¥ap_sh2a_6a_usbfunc¥ap_sh2a_6a_usbfunc¥src フォルダ内>

main.c	...	メイン処理
timer.c	...	タイマ処理
sci.c	...	シリアル処理
scif.h	...	シリアル処理ヘッダファイル
can.c	...	CAN 処理
boot.c	...	CPU 初期化処理
common.h	...	共通ヘッダファイル
iodef.h	...	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
section.src	...	セクション定義
boarddepend.h	...	ボード依存ファイル
typedef.h	...	型定義ファイル

- <¥Sample¥ap_sh2a_6a_usbfunc¥ap_sh2a_6a_usbfunc¥src¥usb_firm フォルダ内>
- c_version.h … バージョン情報定義ファイル
- <¥Sample¥ap_sh2a_6a_usbfunc¥ap_sh2a_6a_usbfunc¥src¥usb_firm¥src フォルダ内>
- c_datatbl.c … データテーブル定義ファイル
 - p_main.c … ペリフェラルメイン処理
- <¥Sample¥ap_sh2a_6a_usbfunc¥ap_sh2a_6a_usbfunc¥src¥usb_firm¥inc フォルダ内>
- c_debugprint.h … デバッガヘッダファイル
 - c_def7269.h … SH7269 USB モジュールレジスタ定義ファイル
 - cfg_usb_sh7269.h … ユーザオプション定義ファイル
 - usb.h … USB ドライバヘッダファイル
 - usb_firm.h … USB モジュール値定義ファイル
- <¥Sample¥ap_sh2a_6a_usbfunc¥ap_sh2a_6a_usbfunc¥src¥usb_firm¥src¥common フォルダ内>
- c_dataio.c … データテーブル定義ファイル
 - c_global.c … グローバル変数定義
 - c_intrn.c … パイプ割り込み処理
 - c_lib7269.c … SH7269 USB レジスタルーチン処理
 - c_libint.c … USB 割り込みレジスタルーチン処理
 - c_usb_dly.c … USB ソフトウェアウェイト処理定義
 - c_usb_dma.c … USB DMA コントローラ
 - c_usbint.c … USB 割り込み処理
- <¥Sample¥ap_sh2a_6a_usbfunc¥ap_sh2a_6a_usbfunc¥src¥usb_firm¥src¥peri フォルダ内>
- p_changeep.c … ペリフェラル設定処理
 - p_classvendor.c … ペリフェラルクラス/ベンダ要求処理
 - p_controlrw.c … ペリフェラルリード/ライト処理
 - p_def_ep.h … ペリフェラルエンドポイント定義
 - p_descrip.c … ペリフェラルデスクリプタ定義処理
 - p_intrn.c … ペリフェラルパイプ割り込み処理
 - p_lib7269.c … ペリフェラル SH7269USB レジスタ定義
 - p_libint.c … ペリフェラル USB 割り込み管理
 - p_status.c … ペリフェラル内部ステータス関連処理
 - p_stdreqget.c … ペリフェラルスタンダードデバイス取得要求処理
 - p_stdreqset.c … ペリフェラルスタンダードデバイス設定要求処理
 - p_usbint.c … ペリフェラル USB 割り込み処理
 - p_usbsig.c … ペリフェラルバスドライバ
- <¥Sample¥ap_sh2a_6a_usbhost フォルダ内>
- ap_sh2a_6a_usbhost.hws … USB ホスト HEW 用ワークスペースファイル
- <¥Sample¥ap_sh2a_6a_usbhost¥ap_sh2a_6a_usbhost フォルダ内>
- ap_sh2a_6a_usbhost.hwp … USB ホスト HEW 用プロジェクトファイル
- <¥Sample¥ap_sh2a_6a_usbhost¥ap_sh2a_6a_usbhost¥Debug フォルダ内>
- ap_sh2a_6a_usbhost.abs … USB ホスト RAM 動作用オブジェクトファイル
(elf 形式)
 - ap_sh2a_6a_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_sh2a_6a_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

<¥Sample¥ap_sh2a_6a_usbhost¥ap_sh2a_6a_usbhost¥Release フォルダ内>

ap_sh2a_6a_usbhost.abs	…	USB ホスト ROM 動作用オブジェクトファイル (elf 形式)
ap_sh2a_6a_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_sh2a_6a_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

<¥Sample¥ap_sh2a_6a_usbhost¥ap_sh2a_6a_usbhost¥src フォルダ内>

main.c	…	メイン処理
timer.c	…	タイマ処理
sci.c	…	シリアル処理
scif.h	…	シリアル処理ヘッダファイル
can.c	…	CAN 処理
boot.c	…	CPU 初期化処理
common.h	…	共通ヘッダファイル
iodef.h	…	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
section.src	…	セクション定義
boarddepend.h	…	ボード依存ファイル
typedef.h	…	型定義ファイル

<¥Sample¥ap_sh2a_6a_usbhost¥ap_sh2a_6a_usbhost¥src¥usb_firm フォルダ内>

c_version.h	…	バージョン情報定義ファイル
-------------	---	---------------

<¥Sample¥ap_sh2a_6a_usbhost¥ap_sh2a_6a_usbhost¥src¥usb_firm¥src フォルダ内>

c_datatbl.c	…	データテーブル定義ファイル
h_main.c	…	ホストメイン処理

<¥Sample¥ap_sh2a_6a_usbhost¥ap_sh2a_6a_usbhost¥src¥usb_firm¥inc フォルダ内>

c_debugprint.h	…	デバッグヘッダファイル
c_def7269.h	…	SH7269 USB モジュールレジスタ定義ファイル
cfg_usb_sh7269.h	…	ユーザオプション定義ファイル
usb.h	…	USB ドライバヘッダファイル
usb_firm.h	…	USB モジュール値定義ファイル

<¥Sample¥ap_sh2a_6a_usbhost¥ap_sh2a_6a_usbhost¥src¥usb_firm¥src¥common フォルダ内>

c_dataio.c	…	データテーブル定義ファイル
c_global.c	…	グローバル変数定義
c_intrn.c	…	パイプ割り込み処理
c_lib7269.c	…	SH7269 USB レジスタルーチン処理
c_libint.c	…	USB 割り込みレジスタルーチン処理
c_usb_dly.c	…	USB ソフトウェアウェイト処理定義
c_usb_dma.c	…	USB DMA コントローラ
c_usbint.c	…	USB 割り込み処理

<¥Sample¥ap_sh2a_6a_usbhost¥ap_sh2a_6a_usbhost¥src¥usb_firm¥src¥host フォルダ内>

h_controlrw.c	...	ホストリード/ライト処理
h_debugprint.c	...	デバッグ用ターミナル出力処理
h_host_Desc.c	...	ホストパイプ定義処理
h_intrn.c	...	ホストパイプ割り込み処理
h_lib7269.c	...	SH7269 ホストレジスタ処理
h_libint.c	...	ホスト割り込み管理
h_usb_elt.c	...	ホスト電気的テスト処理
h_usbdrv.c	...	ホスト USB ドライバメイン処理
h_usbdrv_api.c	...	ホスト USB ドライバサブ処理
h_usbdrv_el.c	...	ホスト電気的サンプルプログラムメイン処理
h_usbdrv_sub.c	...	ホスト USB ドライバサブ処理
h_usbint.c	...	ホスト USB 割り込み処理
h_usbsig.c	...	ホスト USB バスドライバ

3. USB ファンクションサンプルプログラム

3.1 ビルド・デバッグ方法（USB ファンクション）

（1）ビルド

- ① HEW を起動し、¥Sample¥ap_sh2a_6a_usbfunc¥ap_sh2a_6a_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 [ビルド] 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 [ビルド] - [ビルド] を実行してください。ap_sh2a_6a_usbfunc.mot、ap_sh2a_6a_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ください。

（2）RAM 上でのデバッグ

- ① XsSight を起動し、¥Sample フォルダ直下にある XrossFinder_sh2a_6a.xfc コマンドファイルを読み込みます。
- ② ¥Sample¥ap_sh2a_6a_usbfunc¥ap_sh2a_6a_usbfunc¥Debug フォルダ内の ap_sh2a_6a_usbfunc.abs を XsSight からダウンロードして動作を確認してください。

（3）ROM 上でのデバッグ

- ① SP-SH2A-6A のスイッチを、「1.2 動作モード」を参考に設定します。
- ② ¥Sample フォルダ内の XrossFinder_sh2a_6a.xfc と ¥Sample¥ap_sh2a_6a_usbfunc¥ap_sh2a_6a_usbfunc¥Release フォルダ内の ap_sh2a_6a_usbfunc.abs を XsSight で読み込みます。
- ③ 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。
- ④ START ボタンを押してプログラムの書き込みを行い、動作を確認してください。

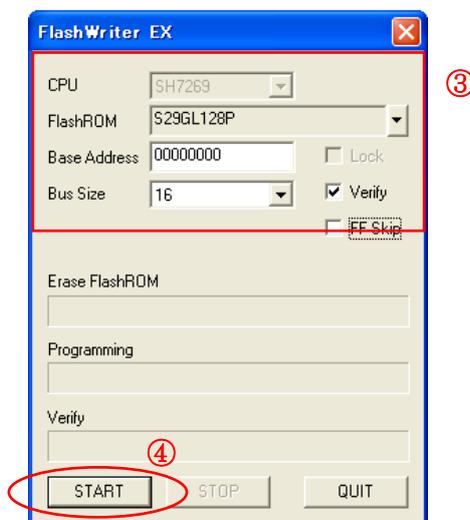


Fig3.1-1 FlashWriter EX for XsSight の設定

(4) XsSight 不使用时の確認方法

・FlashWriter EXを使用する場合

- ① アダプタ (HJ-LINK / XrossFinder / XrossFinder Evo) を使用して PC とボードを繋ぎます。
- ② AP-SH2A-6A 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。
- 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_sh2a_6a.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 ¥Release フォルダ内の ap_sh2a_6a_usbfunc.mot をボードに書き込みます。

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)	20MHz 以下
CPU	SH7269
BaseAddress	00000000
FLASHROM	S29GL128P (SPANSION)
BUS SIZE	16

Table3.1-1 FlashWriter EX の設定

- ※ 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、「S29GL128P (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や拡張バスを用いて他の FLASHROM を接続している場合には、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。
- ※ FlashWriter EX はシリアル FLASHROM への書き込みに対応しておりません。
- ※ AP-SH2A-6A は標準ではシリアル FLASHROM が実装されていません。シリアル FLASHROM の実装に関しましては、AP-SH2A-6A のハードウェアマニュアルをご覧ください。

3.2 動作説明 (USB ファンクション)

3.2.1 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。

- 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。
- SCIF3 でエコーバックを行います。(送受信割り込み使用)
SCIF3 から受信した値をそのまま、SCIF3 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、ホスト PC 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。
- 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 1sec 間隔で ON/OFF します。(MTU20 割り込み使用)
- 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 500kbps (TSG1=5 (6tq), TSE2=2 (3tq), SJW=0, BSP=0, BRP=4) です。
- CN1 のポートより方形波を出力します。周期とピン番号を下記の表に示します。
- USB ファンクションをホスト PC 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、「3.2.2 USB ファンクション動作」を参照してください。

方形波出力端子一覧 1

ピン番号	ピン名	周期	備考
CN1.13	PF4/#CE5/#CE1A/SSISCK0//SGOUT0/	20msec	MTU20 使用
CN1.14	PF5/#CE6/#CE1B/SSIWS0//SGOUT1/	20msec	MTU20 使用
CN1.15	PF6/#CE2A/SSITxD0//SGOUT2/	20msec	MTU20 使用
CN1.21	PF10/#CS1/SSISCK1/DV_DATA1/SCK1/MMC_D5/	20msec	MTU20 使用
CN1.24	PF15/A0/SSIDATA2/#WDTOVF/TxD2/#UBCTRG/	20msec	MTU20 使用
CN4.18	PF16/SD_CD_0//#FCE/IRQ4/MMC_CD/	20msec	MTU20 使用
CN4.16	PF18/SD_D1_0/SSISCK3//IRQ6/MMC_D1/	20msec	MTU20 使用
CN4.15	PF19/SD_D0_0/SSIWS3//IRQ7/MMC_D0/	20msec	MTU20 使用
CN4.14	PF20/SD_CLK_0/SSIDATA3//MMC_CLK/	20msec	MTU20 使用
CN4.13	PF21/SD_CMD_0//SCK3/MMC_CMD/	20msec	MTU20 使用
CN4.12	PF22/SD_D3_0//RxD3/MMC_D3/	20msec	MTU20 使用
CN4.11	PF23/SD_D2_0//TxD3/MMC_D2/	20msec	MTU20 使用

信号名に#がついているものは負論理を表します。

方形波出力端子一覧2

ピン番号	ピン名	周期	備考
CN1.56	PG0/D16/LCD_DATA0/IRQ0/TIOC0A	10msec	CMT0 使用
CN1.55	PG1/D17/LCD_DATA1/IRQ1/TIOC0B	10msec	CMT0 使用
CN1.54	PG2/D18/LCD_DATA2/IRQ2/TIOC0C	10msec	CMT0 使用
CN1.53	PG3/D19/LCD_DATA3/IRQ3/TIOC0D	10msec	CMT0 使用
CN1.52	PG4/D20/LCD_DATA4/IRQ4/TIOC1A	10msec	CMT0 使用
CN1.51	PG5/D21/LCD_DATA5/IRQ5/TIOC1B	10msec	CMT0 使用
CN1.50	PG6/D22/LCD_DATA6/IRQ6/TIOC2A	10msec	CMT0 使用
CN1.49	PG7/D23/LCD_DATA7/IRQ7/TIOC2B	10msec	CMT0 使用
CN1.46	PG8/D24/LCD_DATA8/PINT0/ TIOC3A	10msec	CMT0 使用
CN1.45	PG9/D25/LCD_DATA9/PINT1/ TIOC3B	10msec	CMT0 使用
CN1.44	PG10/D26/LCD_DATA10/PINT2/ TIOC3C	10msec	CMT0 使用
CN1.43	PG11/D27/LCD_DATA11/PINT3/ TIOC3D	10msec	CMT0 使用
CN1.42	PG12/D28/LCD_DATA12/PINT4/	10msec	CMT0 使用
CN1.41	PG13/D29/LCD_DATA13/PINT5/	10msec	CMT0 使用
CN1.40	PG14/D30/LCD_DATA14/PINT6/	10msec	CMT0 使用
CN1.39	PG15/D31/LCD_DATA15/ PINT7/	10msec	CMT0 使用
CN1.36	PG16/#WE2 #ICIORD DQMUL/LCD_DATA16//	10msec	CMT0 使用
CN1.35	PG17/#WE3 #ICIOWR #AH DQMUU/LCD_DATA17//	10msec	CMT0 使用
CN1.34	PG18/DV_DATA4/LCD_DATA18/SPDIF_IN/SCK4	10msec	CMT0 使用
CN1.33	PG19/DV_DATA5/LCD_DATA19/SPDIF_OUT/SCK5	10msec	CMT0 使用
CN1.32	PG20/DV_DATA6/LCD_DATA20/LCD_TCON3/RxD4	10msec	CMT0 使用
CN1.31	PG21/DV_DATA7/LCD_DATA21/LCD_TCON4/TxD4	10msec	CMT0 使用
CN4.33	PG22//LCD_DATA22/LCD_TCON5/RxD5	10msec	CMT0 使用
CN4.34	PG23//LCD_DATA23/LCD_TCON6/TxD5	10msec	CMT0 使用
CN1.29	PG25//LCD_TCON0//	10msec	CMT0 使用
CN1.28	PG26//LCD_TCON1//	10msec	CMT0 使用
CN1.27	PG27//LCD_TCON2/LCD_EXTCLK/	10msec	CMT0 使用

信号名に#がついているものは負論理を表します。

3.2.2 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。

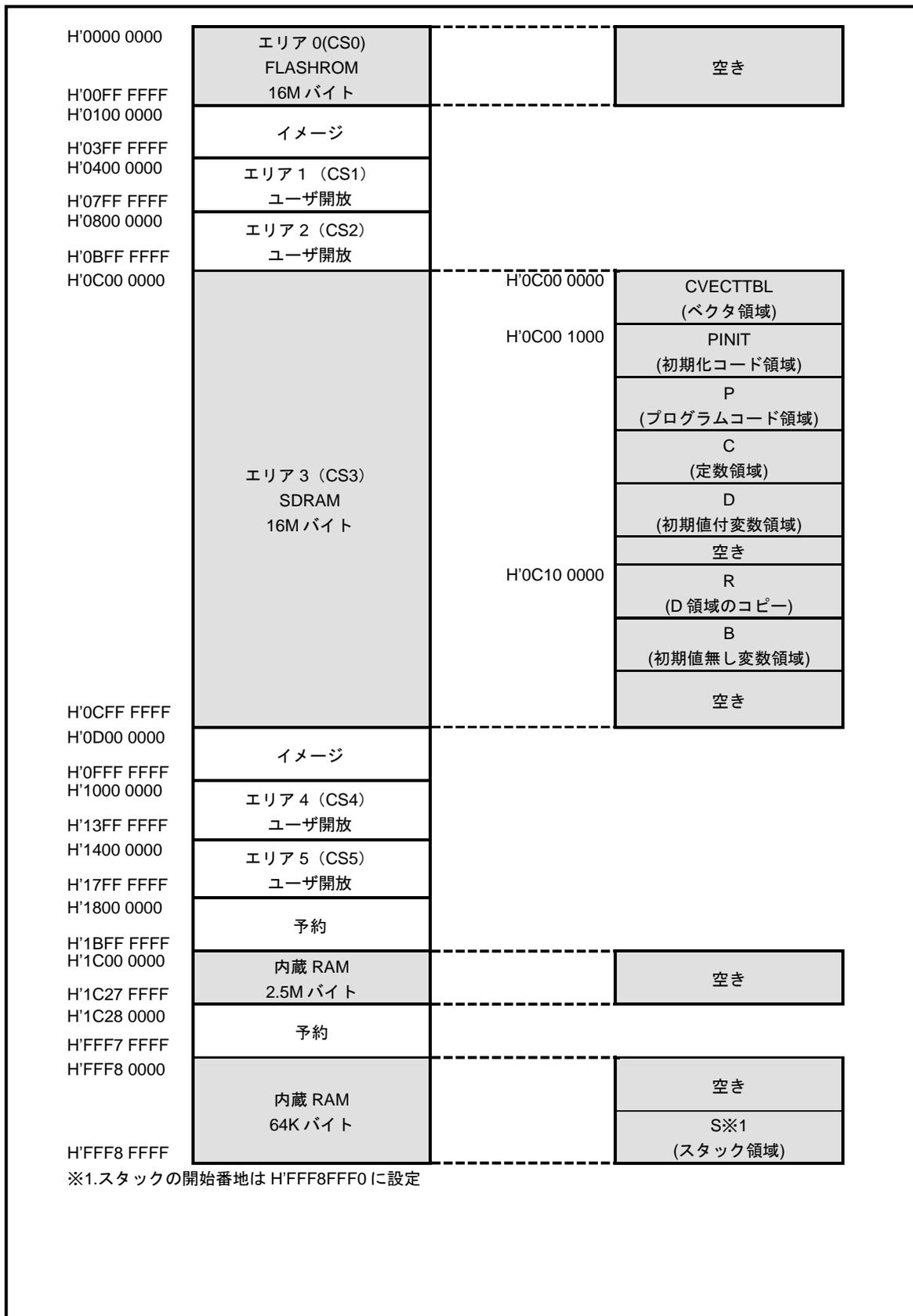
Win10 よりも前の OS での USB ファンクション動作確認は、あらかじめ USB 仮想シリアルドライバを PC にインストールしておく必要があります。

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、ホスト PC の USB ポートと CPU ボードの USB ファンクションポート (CN11) 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 ホスト PC 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。

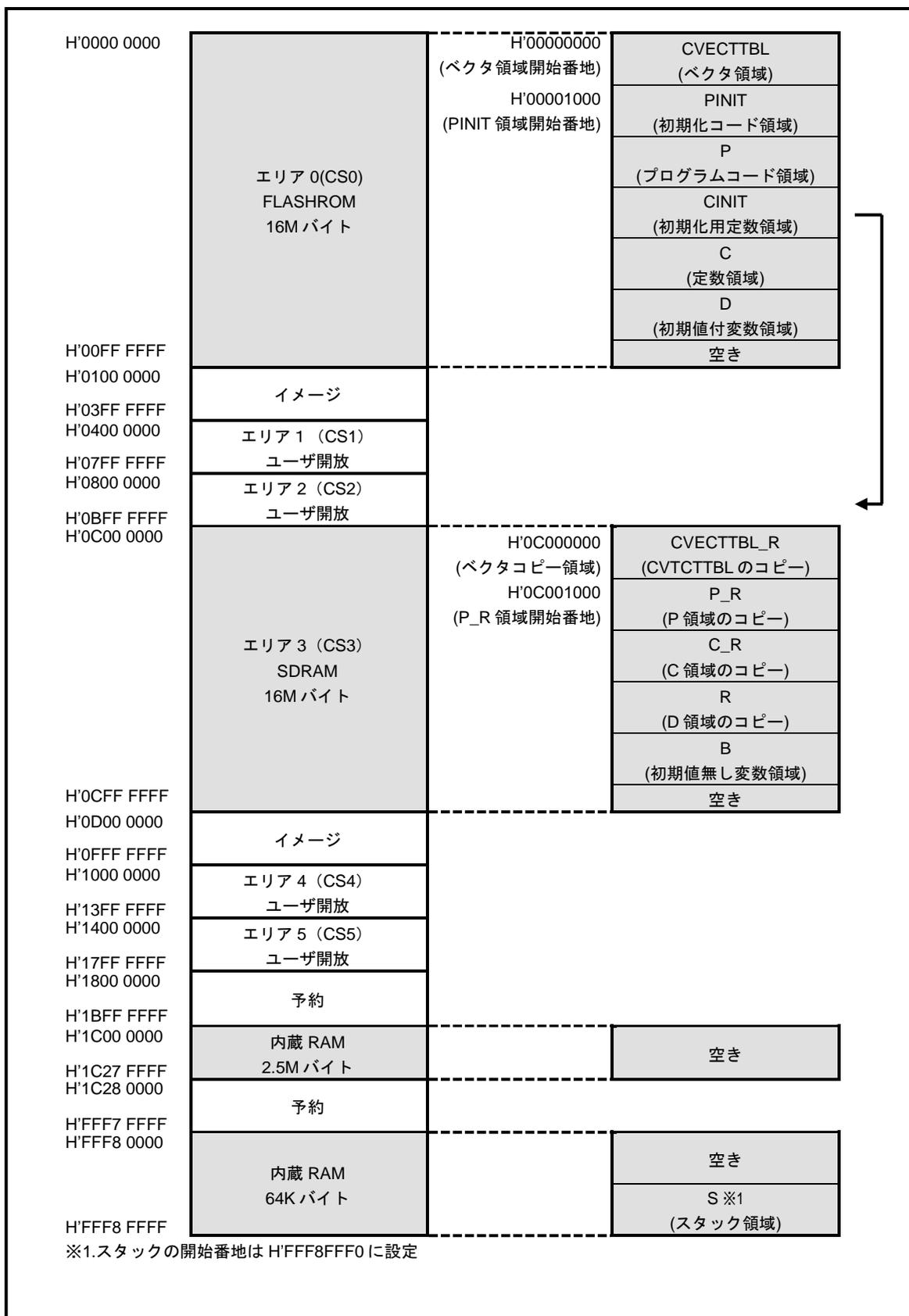
3.3 RAM 動作時のメモリマップ (USB ファンクション)

メモリマップを以下に示します。



3.4 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。



4. USB ホストサンプルプログラム

4.1 ビルド・デバッグ方法（USB ホスト）

（1）ビルド

- ① HEW を起動し、¥Sample¥ap_sh2a_6a_usbhost¥ap_sh2a_6a_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 [ビルド] 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 [ビルド] - [ビルド] を実行してください。ap_sh2a_6a_usbhost.mot、ap_sh2a_6a_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ください。

（2）RAM 上でのデバッグ

- ① XsSight を起動し、¥Sample フォルダ直下にある XrossFinder_sh2a_6a.xfc コマンドファイルを読み込みます。
- ② ¥Sample¥ap_sh2a_6a_usbhost¥ap_sh2a_6a_usbhost¥Debug フォルダ内の ap_sh2a_6a_usbhost.abs を XsSight でダウンロードして動作を確認してください。

（3）ROM 上でのデバッグ

- ① SP-SH2A-6A のスイッチを、「1.2 動作モード」を参考に設定します。
- ② ¥Sample フォルダ内の XrossFinder_sh2a_6a.xfc と ¥Sample¥ap_sh2a_6a_usbhost¥ap_sh2a_6a_usbhost¥Release フォルダ内の ap_sh2a_6a_usbhost.abs を XsSight で読み込みます。
- ③ 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。
- ④ START ボタンを押してプログラムの書き込みを行い、動作を確認してください。

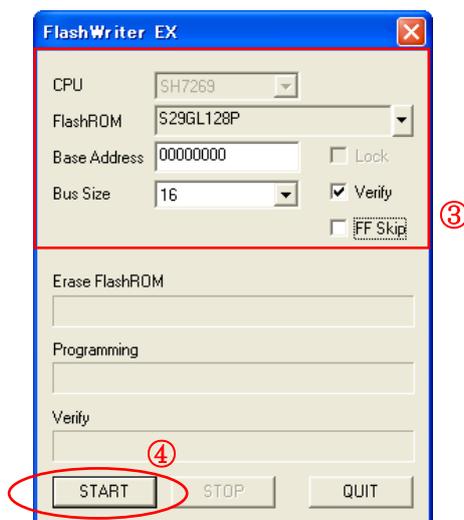


Fig4.1-1 FlashWriter EX for XsSight の設定

(4) XsSight 不使用时の確認方法

・FlashWriter EXを使用する場合

- ① アダプタ (HJ-LINK / XrossFinder / XrossFinder Evo) を使用して PC とボードを繋ぎます。
- ② AP-SH2A-6A 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。
- 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ください。
- ④ FlashWriter EX で、¥Sample フォルダ直下にある XrossFinder_sh2a_6a.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 ¥Release フォルダ内の ap_sh2a_6a_usbhost.mot をボードに書き込みます。
- ⑥ AP-SH2A-6A 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。
- 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)	20MHz 以下
CPU	SH7269
BaseAddress	00000000
FLASHROM	S29GL128P (SPANSION)
BUS SIZE	16

Table4.1-1 FlashWriter EX の設定

- ※ 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、「S29GL128P (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や拡張バスを用いて他の FLASHROM を接続している場合には、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。
- ※ FlashWriter Ex はシリアル FLASHROM への書き込みに対応しておりません。
- ※ AP-SH2A-6A は標準ではシリアル FLASHROM が実装されていません。シリアル FLASHROM の実装に関しましては、AP-SH2A-6A のハードウェアマニュアルをご覧ください。

4.2 動作説明 (USB ホスト)

4.2.1 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。

- 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。
- SCIF3 でエコーバックを行います。(送受信割り込み使用)
SCIF3 から受信した値をそのまま、SCIF3 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、ホスト PC 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。
- 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 1sec 間隔で ON/OFF します。(MTU20 割り込み使用)
- 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 500kbps (TSG1=5 (6tq), TSE2=2 (3tq), SJW=0, BSP=0, BRP=4) です。
- CN1 のポートより方形波を出力します。周期とピン番号を下記の表に示します。
- USB ホストに USB メモリを接続すると、SCIF3 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、「4.2.2 USB ホスト動作」を参照してください。

方形波出力端子一覧 1

ピン番号	ピン名	周期	備考
CN1.13	PF4/#CE5/#CE1A/SSISCK0//SGOUT0/	20msec	MTU20 使用
CN1.14	PF5/#CE6/#CE1B/SSIWS0//SGOUT1/	20msec	MTU20 使用
CN1.15	PF6/#CE2A/SSITxD0//SGOUT2/	20msec	MTU20 使用
CN1.21	PF10/#CS1/SSISCK1/DV_DATA1/SCK1/MMC_D5/	20msec	MTU20 使用
CN1.24	PF15/A0/SSIDATA2/#WDTOVF/TxD2/#UBCTRG/	20msec	MTU20 使用
CN4.18	PF16/SD_CD_0/#FCE/IRQ4/MMC_CD/	20msec	MTU20 使用
CN4.16	PF18/SD_D1_0/SSISCK3//IRQ6/MMC_D1/	20msec	MTU20 使用
CN4.15	PF19/SD_D0_0/SSIWS3//IRQ7/MMC_D0/	20msec	MTU20 使用
CN4.14	PF20/SD_CLK_0/SSIDATA3//MMC_CLK/	20msec	MTU20 使用
CN4.13	PF21/SD_CMD_0//SCK3/MMC_CMD/	20msec	MTU20 使用
CN4.12	PF22/SD_D3_0/RxD3/MMC_D3/	20msec	MTU20 使用
CN4.11	PF23/SD_D2_0//TxD3/MMC_D2/	20msec	MTU20 使用

信号名に#がついているものは負論理を表します。

方形波出力端子一覧 2

ピン番号	ピン名	周期	備考
CN1.56	PG0/D16/LCD_DATA0/IRQ0/TIOC0A	10msec	CMT0 使用
CN1.55	PG1/D17/LCD_DATA1/IRQ1/TIOC0B	10msec	CMT0 使用
CN1.54	PG2/D18/LCD_DATA2/IRQ2/TIOC0C	10msec	CMT0 使用
CN1.53	PG3/D19/LCD_DATA3/IRQ3/TIOC0D	10msec	CMT0 使用
CN1.52	PG4/D20/LCD_DATA4/IRQ4/TIOC1A	10msec	CMT0 使用
CN1.51	PG5/D21/LCD_DATA5/IRQ5/TIOC1B	10msec	CMT0 使用
CN1.50	PG6/D22/LCD_DATA6/IRQ6/TIOC2A	10msec	CMT0 使用
CN1.49	PG7/D23/LCD_DATA7/IRQ7/TIOC2B	10msec	CMT0 使用
CN1.46	PG8/D24/LCD_DATA8/PINT0/ TIOC3A	10msec	CMT0 使用
CN1.45	PG9/D25/LCD_DATA9/PINT1/ TIOC3B	10msec	CMT0 使用
CN1.44	PG10/D26/LCD_DATA10/PINT2/ TIOC3C	10msec	CMT0 使用
CN1.43	PG11/D27/LCD_DATA11/PINT3/ TIOC3D	10msec	CMT0 使用
CN1.42	PG12/D28/LCD_DATA12/PINT4/	10msec	CMT0 使用
CN1.41	PG13/D29/LCD_DATA13/PINT5/	10msec	CMT0 使用
CN1.40	PG14/D30/LCD_DATA14/PINT6/	10msec	CMT0 使用
CN1.39	PG15/D31/LCD_DATA15/ PINT7/	10msec	CMT0 使用
CN1.36	PG16/#WE2 #ICIORD DQMUL/LCD_DATA16//	10msec	CMT0 使用
CN1.35	PG17/#WE3 #ICIOWR #AH DQMUU/LCD_DATA17//	10msec	CMT0 使用
CN1.34	PG18/DV_DATA4/LCD_DATA18/SPDIF_IN/SCK4	10msec	CMT0 使用
CN1.33	PG19/DV_DATA5/LCD_DATA19/SPDIF_OUT/SCK5	10msec	CMT0 使用
CN1.32	PG20/DV_DATA6/LCD_DATA20/LCD_TCON3/RxD4	10msec	CMT0 使用
CN1.31	PG21/DV_DATA7/LCD_DATA21/LCD_TCON4/TxD4	10msec	CMT0 使用
CN4.33	PG22//LCD_DATA22/LCD_TCON5/RxD5	10msec	CMT0 使用
CN4.34	PG23//LCD_DATA23/LCD_TCON6/TxD5	10msec	CMT0 使用
CN1.29	PG25//LCD_TCON0//	10msec	CMT0 使用
CN1.28	PG26//LCD_TCON1//	10msec	CMT0 使用
CN1.27	PG27//LCD_TCON2/LCD_EXTCLK/	10msec	CMT0 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 ホスト動作

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 ボードとホスト PC 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。
ホスト PC 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 (CN12) 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。

```
Attach Port -----  
  
Hi-Speed Device----  
  
USB Address is X--- ※1  
  
Get configuration (Device descriptor) ----- PASS! Device descriptor  
dump: YY ※2  
(後略)
```

※1. X には、USB デバイスのアドレスが表示されます。

※2. YY には USB のデバイス情報が表示されます。

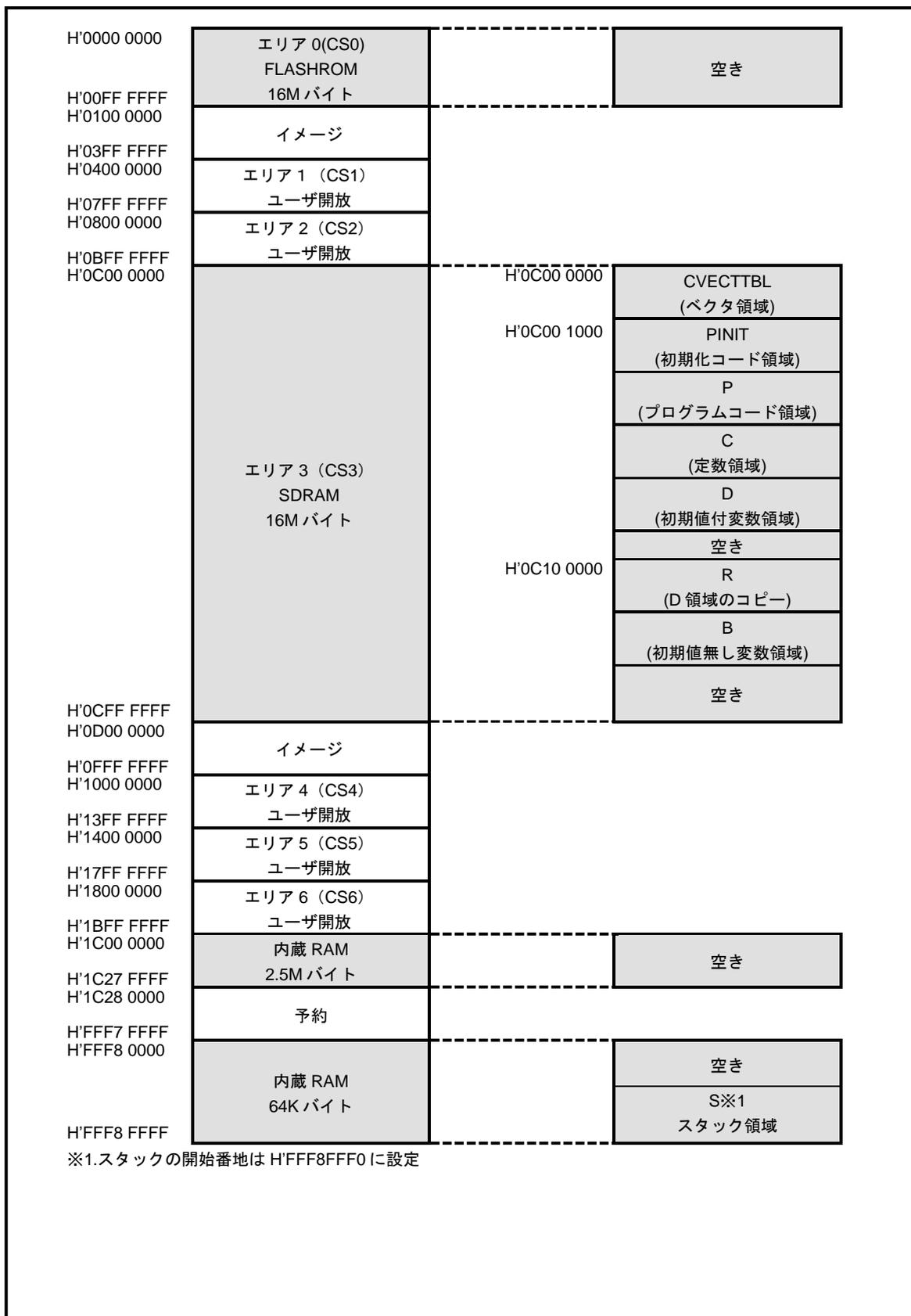
- ⑥ 上記の表示が出たことを確認した後、USB メモリを抜いてください。ターミナルソフト上に切断状況が表示されます。以下のような表示が出ていることを確認してください。

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

- ⑦ 以上で USB メモリ接続時の動作は終了です。

4.3 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。



4.4 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。

H'0000 0000	エリア 0 (CS0) FLASHROM 16M バイト	H'00000000 (ベクタ領域開始番地)	CVECTTBL (ベクタ領域)
		H'00001000 (PINIT 領域開始番地)	PINIT (初期化コード領域)
			P (プログラムコード領域)
			CINIT (初期化用定数領域)
			C (定数領域)
			D (初期値付変数領域)
H'00FF FFFF			空き
H'0100 0000	イメージ		
H'03FF FFFF H'0400 0000	エリア 1 (CS1)		
H'07FF FFFF H'0800 0000	ユーザ開放		
H'0BFF FFFF H'0C00 0000	エリア 2 (CS2)		
	ユーザ開放		
	エリア 3 (CS3) SDRAM 16M バイト	H'0C00 0000 (ベクタコピー領域)	CVECTTBL_R (CVECTTBL のコピー)
		H'0C00 1000 (P_R 領域開始番地)	P_R (P 領域のコピー)
			C_R (C 領域のコピー)
			R (D 領域のコピー)
			B (初期値無し変数領域)
			空き
H'0CFF FFFF			
H'0D00 0000	イメージ		
H'0FFF FFFF H'1000 0000	エリア 4 (CS4)		
H'13FF FFFF H'1400 0000	ユーザ開放		
H'17FF FFFF H'1800 0000	エリア 5 (CS5)		
	ユーザ開放		
H'1BFF FFFF H'1C00 0000	エリア 6 (CS6)		
	ユーザ開放		
H'1BFF FFFF H'1C00 0000	内蔵 RAM 2.5M バイト		空き
H'1C27 FFFF H'1C28 0000	予約		
H'FFF7 FFFF H'FFF8 0000	内蔵 RAM 64K バイト		空き
			S ※1 (スタック領域)
H'FFF8 FFFF			

※1.スタックの開始番地は H'FFF8FFF0 に設定

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 ドライバ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。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

- ・ 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。
- ・ SH7269 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®10、Windows®11 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®11 は Windows 11 もしくは Win11
High-performance Embedded Workshop は HEW

- ・ 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
静岡県浜松市中央区積志町 834
<https://www.apnet.co.jp>
E-Mail: query@apnet.co.jp