

PC-HID-01

サンプルプログラム解説

第1版 2009年11月10日

目次

| | |
|--|----|
| 1. 概要..... | 1 |
| 1. 1 概要..... | 1 |
| 1. 2 接続方法..... | 1 |
| 1. 3 動作モード..... | 2 |
| 1. 4 開発環境について..... | 5 |
| 1. 5 ファイルシステム及びドライバについて..... | 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 SHC用Mass Storage Class Host サンプルプログラム..... | 8 |
| 2. 2. 2 SHC用HID Class Host サンプルプログラム..... | 10 |
| 2. 2. 3 SHC用HID Class Device サンプルプログラム..... | 12 |
| 2. 2. 4 GCC用Mass Storage Class Host サンプルプログラム..... | 14 |
| 2. 2. 5 GCC用HID Class Host サンプルプログラム..... | 16 |
| 2. 2. 6 GCC用HID Class Device サンプルプログラム..... | 18 |
| 3. ビルド・デバッグ方法..... | 20 |
| 3. 1 SHC用サンプルプログラム (Ver9以降)..... | 20 |
| 3. 2 KPIT社製 GNU GCC サンプルプログラム (GNUSH v0903以降)..... | 21 |
| 3. 3 サンプルプログラムの移植について..... | 21 |
| 4. 動作説明..... | 22 |
| 4. 1 Mass Storage Class Host用サンプルプログラムの動作説明..... | 22 |
| 4. 1. 1 HID Storage Class Host用サンプルプログラム概要..... | 22 |
| 4. 1. 2 HID Storage Class Host用サンプルプログラム実行方法..... | 22 |
| 4. 2 HID Class Host用サンプルプログラムの動作説明..... | 23 |
| 4. 2. 1 HID Class Host用サンプルプログラム概要..... | 23 |
| 4. 2. 2 HID Class Host用サンプルプログラム実行方法..... | 23 |
| 4. 3 HID Class Device用サンプルプログラムの動作説明..... | 24 |
| 4. 3. 1 HID Class Device用サンプルプログラム概要..... | 24 |
| 4. 3. 2 HID Class Device用サンプルプログラム実行方法..... | 24 |
| 5. HID Class データフォーマット..... | 25 |
| 5. 1 keyboard 入力のデータフォーマット..... | 25 |
| 5. 2 キーコード変換..... | 25 |

1. 概要

1. 1 概要

本アプリケーションノートでは、PC-HID-01 に付属するサンプルプログラムについて解説します。

PC-HID-01 には、弊社製 CPU ボード AP-SH2A-0A 上で動作する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 マスストレージファイルシステム実装例 • USB ホスト HID 実装例 |
| 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 HID 実装例 |

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。

CPU ボード AP-SH2A-0A の詳細に関しては弊社 HP <http://apnet.co.jp/index.html> からマニュアルがダウンロードできますので、そちらをご覧ください。

1. 2 接続方法

PC-HID-01 と AP-SH2A-0A は PC-HID-01 に付属のシリアルインターフェースケーブルで接続します。

PC-HID-01 の CN1 と AP-SH2A-0A の CN3 をシリアルインターフェースケーブルで接続して使用してください。

PC-HID-01 の動作電源は AP-SH2A-0A から供給されます。

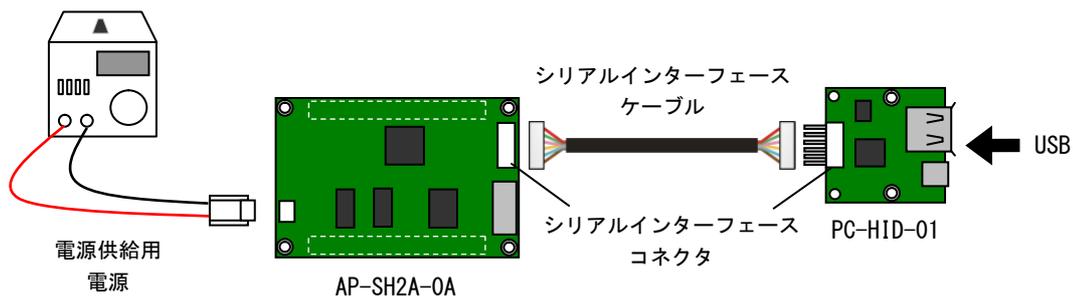


Fig 1.2-1 PC-HID-01 と AP-SH2A-0A との接続例

1. 3 動作モード

本サンプルプログラムは、AP-SH2A-0A で動作します。クロックモード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-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。

CPUモード : モード2、クロック CPU = 160MHz、周辺 = 40MHz、バス = 40MHz
CS3 バスサイズ 16bit、ビッグエンディアン

デバッグモード : XrossFinder 使用時は SW1 の 6 を ON に、未使用の場合は OFF として下さい。

XrossFinder をご使用の場合、以下の DIPSW の設定を行って下さい。

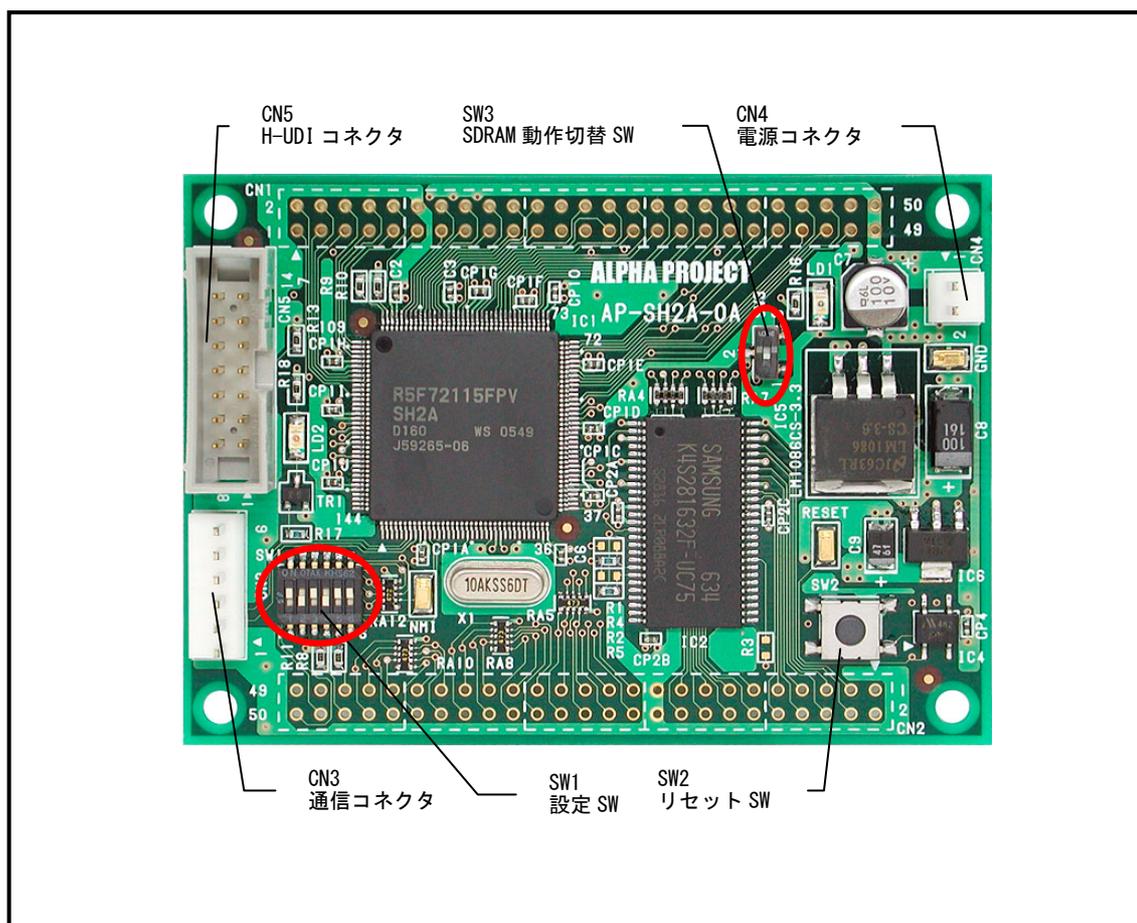


Fig 1.3-1 AP-SH2A-01 外形図

デバッグ時

- ・ SW1 - 1~6 - 101011 (1=ON, 0=OFF とします)
- ・ SW3 - 外付け SDRAM を使用する場合は、シルク表記の 1 側に合わせ、R7 を実装状態とする。

SW1 の設定

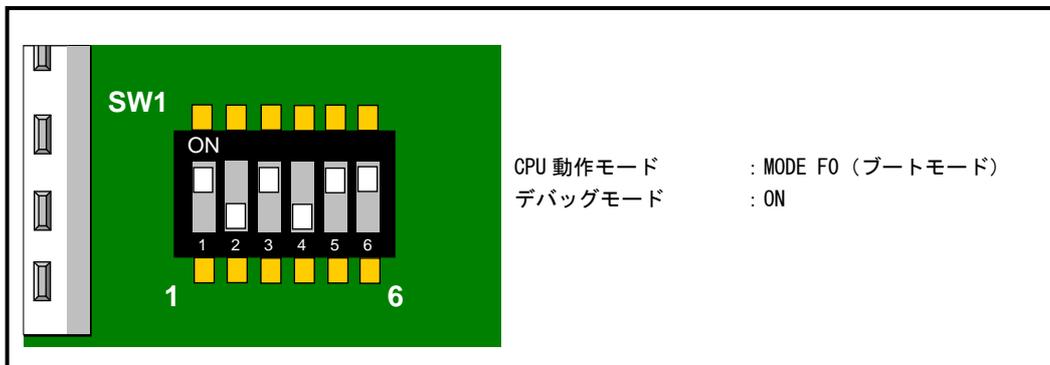


Fig 1.3-2 デバッグ時の DIPSW1 設定

SW3 外付け SDRAM 使用時の設定



Fig 1.3-3 SDRAM 動作切り替え SW

R7 外付け SDRAM 使用時の設定 (R7 は AP-SH2A-0A の裏面に実装されています。)



Fig 1.3-4 SDRAM 動作切り替え抵抗

内蔵フラッシュ ROM 書き込み時

- ・ SW1 - 1~6 - 101001 (1=ON, 0=OFF とします)

※この設定でもデバッグ可能ですが、クロックが CPU = 40MHz 固定になります。

SW1 の設定

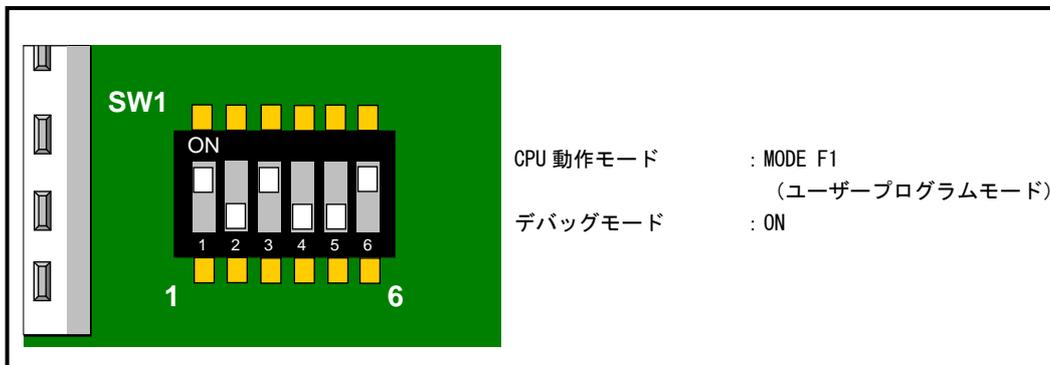


Fig 1.3-5 デバッグ時の DIPSW1 設定

1. 4 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。使用しているコンパイラにより、「shc」、「gcc」とフォルダ別に収録しています。各フォルダに収録されているサンプルプログラムに対応する開発環境、コンパイラのバージョンは次になります。

また、サンプルプログラムは弊社 JTAG デバッガ『XrossFinder』及びフラッシュ書き込みツール『FlashWriter EX』での使用を前提に記載します。

| 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 3.01.08.000 (release 1) 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 |
| gcc | High-performance Embedded Workshop | V 4.04.01.001 以降 | GNUSH ※2 (KPIT) | V0903 以降 | ELF 形式 High-performance Embedded Workshop 対応 |

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

※2: KPIT 社製「GCC」です。GNUSH 及び 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. 5 ファイルシステム及びドライバについて

本サンプルプログラムはフリーウェアである FAT 使用準拠のファイルシステム「Fatfs」及び EPSON より提供されている S1R72U06 サンプルドライバを使用しています。これらに関するサポートはお受け出来ませんので宜しくお願いします。

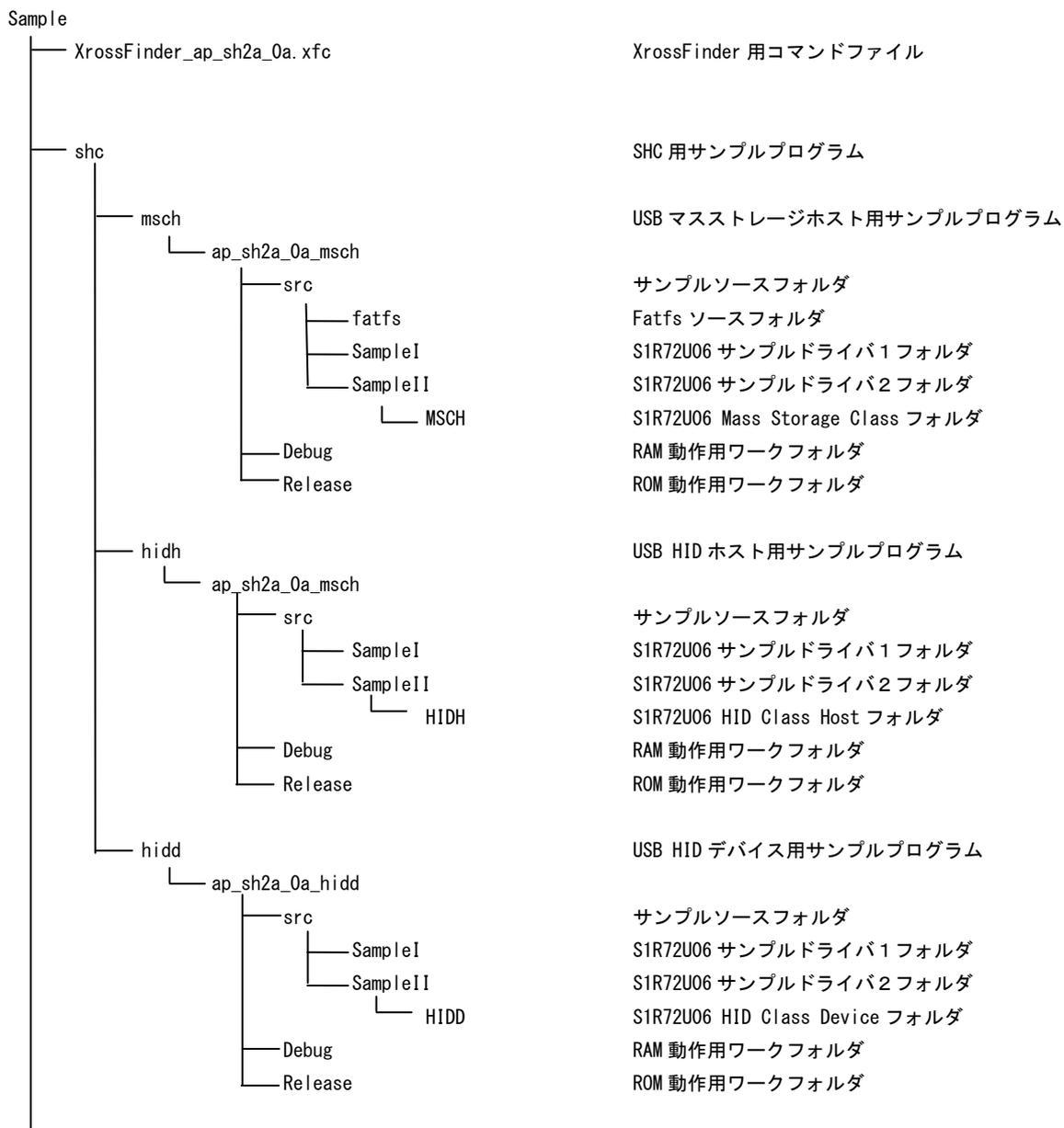
※ Fatfs 入手先 http://elm-chan.org/fsw/ff/00index_j.html

※ S1R72U06 サンプルドライバ及びマニュアル入手先
<https://www.epsondevice.com/support/interface/product/72u06.html>
(ユーザー登録が必要となります)

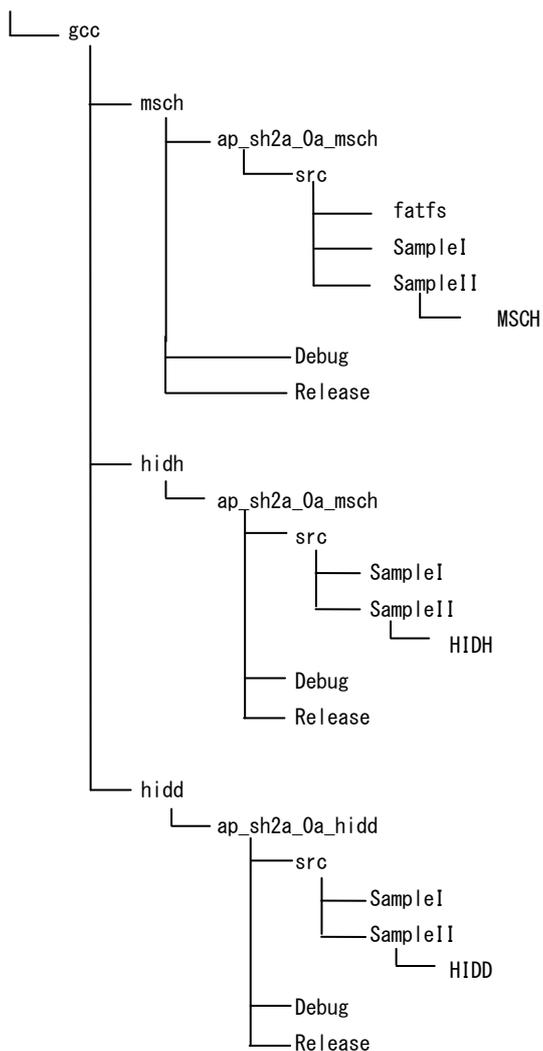
2. サンプルプログラムの構成

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。



次ページへ続きます。



KPIT 社製 GCC 用サンプルプログラム

USB マスストレージホスト用サンプルプログラム

- サンプルソースフォルダ
- Fatfs ソースフォルダ
- S1R72U06 サンプルドライバ1 フォルダ
- S1R72U06 サンプルドライバ2 フォルダ
- MSCH
- RAM 動作用ワークフォルダ
- ROM 動作用ワークフォルダ

USB HID ホスト用サンプルプログラム

- サンプルソースフォルダ
- S1R72U06 サンプルドライバ1 フォルダ
- S1R72U06 サンプルドライバ2 フォルダ
- S1R72U06 HID Class Host フォルダ
- RAM 動作用ワークフォルダ
- ROM 動作用ワークフォルダ

USB HID デバイス用サンプルプログラム

- サンプルソースフォルダ
- S1R72U06 サンプルドライバ1 フォルダ
- S1R72U06 サンプルドライバ2 フォルダ
- S1R72U06 HID Class Device フォルダ
- RAM 動作用ワークフォルダ
- ROM 動作用ワークフォルダ

2. 2 ファイル構成

2. 2. 1 SHC用 Mass Storage Class Host サンプルプログラム

SHC用 Mass Storage Class Host 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¥msch フォルダ内>

ap_sh2a_0a_msch. hws … High-performance Embedded Workshop 用
ワークスペースファイル

<shc¥msch¥ap_sh2a_0a_msch フォルダ内>

ap_sh2a_0a_msch. hwp … High-performance Embedded Workshop 用
プロジェクトファイル

<shc¥msch¥ap_sh2a_0a_msch¥src フォルダ内>

vector. c … 割込ベクタテーブル
boot. c … CPU 初期化処理
main. c … メイン処理
sci. c … シリアル処理
timer. c … タイマ処理
section. src … セクション定義ファイル
common. h … 共通ヘッダファイル
BoardDepend. h … ボード依存定義ファイル
7211. h … SH7211 内部レジスタ定義ヘッダファイル
UART. c … サンプルドライバ用シリアル入出力処理
UART. h … サンプルドライバ用シリアル入出力処理ヘッダファイル
MSCH. c … サンプルドライバ用 MSCH API 呼び出し処理
MSCH. h … サンプルドライバ用 MSCH API 呼び出し処理ヘッダファイル
USB. c … サンプルドライバ用 fatfs API 呼び出し処理
USB. h … サンプルドライバ用 fatfs API 呼び出し処理ヘッダファイル

<shc¥msch¥ap_sh2a_0a_msch¥src¥fatfs¥フォルダ内>

Diskio. c … Fatfs 低レベル I/O ソース
Diskio. h … Fatfs 低レベル I/O ヘッダ
ff. c … Fatfs アプリケーションソース
ff. h … Fatfs アプリケーションヘッダ
integer. h … 型宣言ヘッダファイル

<shc¥msch¥ap_sh2a_0a_msch¥src¥SampleI¥フォルダ内>

UOX_User. h … S1R72U06 サンプルドライバユーザー定義ヘッダ
UOX_IF_API. h … S1R72U06 サンプルドライバ API 定義ヘッダ
UOX_IF_API_UART. c … S1R72U06 サンプルドライバ UART API ソース
UOX_IF_SIO. h … S1R72U06 サンプルドライバ SIO 定義ヘッダ
UOX_IF_SIO_UART. c … S1R72U06 サンプルドライバ UART ドライバソース
UOX_IF_Port. h … S1R72U06 サンプルドライバ I/O 定義ヘッダ
UOX_IF_Port. c … S1R72U06 サンプルドライバ I/O ドライバソース

<shc¥msch¥ap_sh2a_0a_msch¥src¥SampleII¥フォルダ内>

UOX_DEF. h … S1R72U06 サンプルドライバユーザー定義ヘッダ
UOX_Common. h … S1R72U06 サンプルドライバ API 定義ヘッダ
UOX. c … S1R72U06 サンプルドライバ UART API ソース
UOX. h … S1R72U06 サンプルドライバ I/O 定義ヘッダ

<shc¥msch¥ap_sh2a_0a_msch¥src¥MSCH¥フォルダ内>

| | | |
|-------------|-----|-------------------------------|
| UOX_MSCH. c | ... | S1R72U06 サンプルドライバマスタレージクラスソース |
| UOX_MSCH. h | ... | S1R72U06 サンプルドライバマスタレージクラスヘッダ |

<shc¥msch¥ap_sh2a_0a_msch¥Debug フォルダ内>

| | | |
|----------------------|-----|-----------------------------|
| ap_sh2a_0a_msch. abs | ... | RAM 動作用オブジェクトファイル(elf 形式) |
| ap_sh2a_0a_msch. mot | ... | RAM 動作用モトローラ S フォーマット形式ファイル |
| ap_sh2a_0a_msch. 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¥msch¥ap_sh2a_0a_msch¥Release フォルダ内>

| | | |
|----------------------|-----|-----------------------------|
| ap_sh2a_0a_msch. abs | ... | ROM 動作用オブジェクトファイル (elf 形式) |
| ap_sh2a_0a_msch. mot | ... | ROM 動作用モトローラ S フォーマット形式ファイル |
| ap_sh2a_0a_msch. 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 SHC 用 HID Class Host サンプルプログラム

SHC 用 HID Class Host 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\hid フォルダ内>

ap_sh2a_0a_hidh.hws … High-performance Embedded Workshop 用
ワークスペースファイル

<shc\hid\ap_sh2a_0a_hidh フォルダ内>

ap_sh2a_0a_hidh.hwp … High-performance Embedded Workshop 用
プロジェクトファイル

<shc\hid\ap_sh2a_0a_hidh\src フォルダ内>

vector.c … 割込ベクタテーブル
boot.c … CPU 初期化処理
main.c … メイン処理
sci.c … シリアル処理
timer.c … タイマ処理
section.src … セクション定義ファイル
common.h … 共通ヘッダファイル
BoardDepend.h … ボード依存定義ファイル
7211.h … SH7211 内部レジスタ定義ヘッダファイル
UART.c … サンプルドライバ用シリアル入出力処理
UART.h … サンプルドライバ用シリアル入出力処理ヘッダファイル
HIDH.c … サンプルドライバ用 HIDH API 呼び出し処理
HIDH.h … サンプルドライバ用 HIDH API 呼び出し処理ヘッダファイル

<shc\hid\ap_sh2a_0a_hidh\src\SampleI フォルダ内>

UOX_User.h … S1R72U06 サンプルドライバユーザー定義ヘッダ
UOX_IF_API.h … S1R72U06 サンプルドライバ API 定義ヘッダ
UOX_IF_API_UART.c … S1R72U06 サンプルドライバ UART API ソース
UOX_IF_S10.h … S1R72U06 サンプルドライバ S10 定義ヘッダ
UOX_IF_S10_UART.c … S1R72U06 サンプルドライバ UART ドライバソース
UOX_IF_Port.h … S1R72U06 サンプルドライバ IO 定義ヘッダ
UOX_IF_Port.c … S1R72U06 サンプルドライバ IO ドライバソース

<shc\hid\ap_sh2a_0a_hidh\src\SampleII フォルダ内>

UOX_DEF.h … S1R72U06 サンプルドライバユーザー定義ヘッダ
UOX_Common.h … S1R72U06 サンプルドライバ API 定義ヘッダ
UOX.c … S1R72U06 サンプルドライバ UART API ソース
UOX.h … S1R72U06 サンプルドライバ IO 定義ヘッダ

<shc\hid\ap_sh2a_0a_hidh\src\MSCH フォルダ内>

UOX_HIDH.c … S1R72U06 サンプルドライバマスタトレージクラスソース
UOX_HIDH.h … S1R72U06 サンプルドライバマスタトレージクラスヘッダ

<shc\hid\ap_sh2a_0a_hidh\Debug フォルダ内>

ap_sh2a_0a_hidh.abs … RAM 動作用オブジェクトファイル(elf 形式)
ap_sh2a_0a_hidh.mot … RAM 動作用モトローラ S フォーマット形式ファイル
ap_sh2a_0a_hidh.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%hidh%ap_sh2a_0a_hidh%Release フォルダ内>

- ap_sh2a_0a_hidh.abs … ROM 動作用オブジェクトファイル (elf 形式)
 - ap_sh2a_0a_hidh.mot … ROM 動作用モトローラ S フォーマット形式ファイル
 - ap_sh2a_0a_hidh.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. 3 SHC 用 HID Class Device サンプルプログラム

SHC 用 HID Class Device 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¥hidd フォルダ内>

ap_sh2a_0a_hidd.hws … High-performance Embedded Workshop 用
ワークスペースファイル

<shc¥hidd¥ap_sh2a_0a_hidd フォルダ内>

ap_sh2a_0a_hidd.hwp … High-performance Embedded Workshop 用
プロジェクトファイル

<shc¥hidd¥ap_sh2a_0a_hidd¥src フォルダ内>

vector.c … 割込ベクタテーブル
boot.c … CPU 初期化処理
main.c … メイン処理
sci.c … シリアル処理
timer.c … タイマ処理
section.src … セクション定義ファイル
common.h … 共通ヘッダファイル
BoardDepend.h … ボード依存定義ファイル
7211.h … SH7211 内部レジスタ定義ヘッダファイル
UART.c … サンプルドライバ用シリアル入出力処理
UART.h … サンプルドライバ用シリアル入出力処理ヘッダファイル
HIDD.c … サンプルドライバ用 HIDD API 呼び出し処理
HIDD.h … サンプルドライバ用 HIDD API 呼び出し処理ヘッダファイル

<shc¥hidd¥ap_sh2a_0a_hidd¥src¥SampleI¥フォルダ内>

UOX_User.h … S1R72U06 サンプルドライバユーザー定義ヘッダ
UOX_IF_API.h … S1R72U06 サンプルドライバ API 定義ヘッダ
UOX_IF_API_UART.c … S1R72U06 サンプルドライバ UART API ソース
UOX_IF_S10.h … S1R72U06 サンプルドライバ S10 定義ヘッダ
UOX_IF_S10_UART.c … S1R72U06 サンプルドライバ UART ドライバソース
UOX_IF_Port.h … S1R72U06 サンプルドライバ IO 定義ヘッダ
UOX_IF_Port.c … S1R72U06 サンプルドライバ IO ドライバソース

<shc¥hidd¥ap_sh2a_0a_hidd¥src¥SampleII¥フォルダ内>

UOX_DEF.h … S1R72U06 サンプルドライバユーザー定義ヘッダ
UOX_Common.h … S1R72U06 サンプルドライバ API 定義ヘッダ
UOX.c … S1R72U06 サンプルドライバ UART API ソース
UOX.h … S1R72U06 サンプルドライバ IO 定義ヘッダ

<shc¥hidd¥ap_sh2a_0a_hidd¥src¥MSCH¥フォルダ内>

UOX_HIDD.c … S1R72U06 サンプルドライバマスタトレージクラスソース
UOX_HIDD.h … S1R72U06 サンプルドライバマスタトレージクラスヘッダ

<shc¥hidd¥ap_sh2a_0a_hidd¥Debug フォルダ内>

ap_sh2a_0a_hidd.abs … RAM 動作用オブジェクトファイル(elf 形式)
ap_sh2a_0a_hidd.mot … RAM 動作用モトローラ S フォーマット形式ファイル
ap_sh2a_0a_hidd.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%hidd%ap_sh2a_0a_hidd%Release フォルダ内>

- ap_sh2a_0a_hidd.abs … ROM 動作用オブジェクトファイル (elf 形式)
 - ap_sh2a_0a_hidd.mot … ROM 動作用モトローラ S フォーマット形式ファイル
 - ap_sh2a_0a_hidd.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. 4 GCC用 Mass Storage Class Host サンプルプログラム

GCC用 Mass Storage Class Host 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¥msch フォルダ内>

ap_sh2a_0a_msch.hws … High-performance Embedded Workshop 用
ワークスペースファイル

<gcc¥msch¥ap_sh2a_0a_msch フォルダ内>

ap_sh2a_0a_msch.hwp … High-performance Embedded Workshop 用
プロジェクトファイル

<gcc¥msch¥ap_sh2a_0a_msch¥src フォルダ内>

vector.c … 割込ベクタテーブル
boot.c … CPU 初期化处理
main.c … メイン処理
sci.c … シリアル処理
timer.c … タイマ処理
common.h … 共通ヘッダファイル
BoardDepend.h … ボード依存定義ファイル
7211.h … SH7211 内部レジスタ定義ヘッダファイル
UART.c … サンプルドライバ用シリアル入出力処理
UART.h … サンプルドライバ用シリアル入出力処理ヘッダファイル
MSCH.c … サンプルドライバ用 MSCH API 呼び出し処理
MSCH.h … サンプルドライバ用 MSCH API 呼び出し処理ヘッダファイル
USB.c … サンプルドライバ用 fatfs API 呼び出し処理
USB.h … サンプルドライバ用 fatfs API 呼び出し処理ヘッダファイル

<gcc¥msch¥ap_sh2a_0a_msch¥src¥fatfs¥フォルダ内>

Diskio.c … Fatfs 低レベル I/O ソース
Diskio.h … Fatfs 低レベル I/O ヘッダ
ff.c … Fatfs アプリケーションソース
ff.h … Fatfs アプリケーションヘッダ
integer.h … 型宣言ヘッダファイル

<gcc¥msch¥ap_sh2a_0a_msch¥src¥SampleI¥フォルダ内>

UOX_User.h … S1R72U06 サンプルドライバユーザー定義ヘッダ
UOX_IF_API.h … S1R72U06 サンプルドライバ API 定義ヘッダ
UOX_IF_API_UART.c … S1R72U06 サンプルドライバ UART API ソース
UOX_IF_SIO.h … S1R72U06 サンプルドライバ SIO 定義ヘッダ
UOX_IF_SIO_UART.c … S1R72U06 サンプルドライバ UART ドライバソース
UOX_IF_Port.h … S1R72U06 サンプルドライバ I/O 定義ヘッダ
UOX_IF_Port.c … S1R72U06 サンプルドライバ I/O ドライバソース

<gcc¥msch¥ap_sh2a_0a_msch¥src¥SampleII¥フォルダ内>

UOX_DEF.h … S1R72U06 サンプルドライバユーザー定義ヘッダ
UOX_Common.h … S1R72U06 サンプルドライバ API 定義ヘッダ
UOX.c … S1R72U06 サンプルドライバ UART API ソース
UOX.h … S1R72U06 サンプルドライバ I/O 定義ヘッダ

<gcc¥msch¥ap_sh2a_0a_msch¥src¥MSCH¥フォルダ内>

UOX_MSCH.c … S1R72U06 サンプルドライバマスタストレージクラスソース
UOX_MSCH.h … S1R72U06 サンプルドライバマスタストレージクラスヘッダ

<gcc¥msch¥ap_sh2a_0a_msch¥Debug フォルダ内>

| | | |
|----------------------|-----|-----------------------------|
| ap_sh2a_0a_msch. x | ... | RAM 動作用オブジェクトファイル(elf 形式) |
| ap_sh2a_0a_msch. mot | ... | RAM 動作用モトローラ S フォーマット形式ファイル |
| ap_sh2a_0a_msch. map | ... | RAM 動作用マップファイル |

コンパイル後は、.o 等のファイルが生成されます

<gcc¥msch¥ap_sh2a_0a_msch¥Release フォルダ内>

| | | |
|----------------------|-----|-----------------------------|
| ap_sh2a_0a_msch. x | ... | ROM 動作用オブジェクトファイル (elf 形式) |
| ap_sh2a_0a_msch. mot | ... | ROM 動作用モトローラ S フォーマット形式ファイル |
| ap_sh2a_0a_msch. map | ... | ROM 動作用マップファイル |

コンパイル後は、.o 等のファイルが生成されます

2. 2. 5 GCC 用 HID Class Host サンプルプログラム

GCC 用 HID Class Host 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¥hid¥フォルダ内>

ap_sh2a_0a_hidh.hws … High-performance Embedded Workshop 用
ワークスペースファイル

<gcc¥hid¥ap_sh2a_0a_hidh フォルダ内>

ap_sh2a_0a_hidh.hwp … High-performance Embedded Workshop 用
プロジェクトファイル

<gcc¥hid¥ap_sh2a_0a_hidh¥src フォルダ内>

vector.c … 割込ベクタテーブル
boot.c … CPU 初期化処理
main.c … メイン処理
sci.c … シリアル処理
timer.c … タイマ処理
common.h … 共通ヘッダファイル
BoardDepend.h … ボード依存定義ファイル
7211.h … SH7211 内部レジスタ定義ヘッダファイル
UART.c … サンプルドライバ用シリアル入出力処理
UART.h … サンプルドライバ用シリアル入出力処理ヘッダファイル
HIDH.c … サンプルドライバ用 HIDH API 呼び出し処理
HIDH.h … サンプルドライバ用 HIDH API 呼び出し処理ヘッダファイル

<gcc¥hid¥ap_sh2a_0a_hidh¥src¥SampleI¥フォルダ内>

UOX_User.h … S1R72U06 サンプルドライバユーザー定義ヘッダ
UOX_IF_API.h … S1R72U06 サンプルドライバ API 定義ヘッダ
UOX_IF_API_UART.c … S1R72U06 サンプルドライバ UART API ソース
UOX_IF_S10.h … S1R72U06 サンプルドライバ S10 定義ヘッダ
UOX_IF_S10_UART.c … S1R72U06 サンプルドライバ UART ドライバソース
UOX_IF_Port.h … S1R72U06 サンプルドライバ IO 定義ヘッダ
UOX_IF_Port.c … S1R72U06 サンプルドライバ IO ドライバソース

<gcc¥hid¥ap_sh2a_0a_hidh¥src¥SampleII¥フォルダ内>

UOX_DEF.h … S1R72U06 サンプルドライバユーザー定義ヘッダ
UOX_Common.h … S1R72U06 サンプルドライバ API 定義ヘッダ
UOX.c … S1R72U06 サンプルドライバ UART API ソース
UOX.h … S1R72U06 サンプルドライバ IO 定義ヘッダ

<gcc¥hid¥ap_sh2a_0a_hidh¥src¥MSCH¥フォルダ内>

UOX_HIDH.c … S1R72U06 サンプルドライバマスストレージクラスソース
UOX_HIDH.h … S1R72U06 サンプルドライバマスストレージクラスヘッダ

<gcc¥hid¥ap_sh2a_0a_hidh¥Debug フォルダ内>

ap_sh2a_0a_hidh.x … RAM 動作用オブジェクトファイル(elf 形式)
ap_sh2a_0a_hidh.mot … RAM 動作用モトローラ S フォーマット形式ファイル
ap_sh2a_0a_hidh.map … RAM 動作用マップファイル
コンパイル後は、.o 等のファイルが生成されます

<gcc¥hidh¥ap_sh2a_0a_hidh¥Release フォルダ内>

- ap_sh2a_0a_hidh.x … ROM 動作用オブジェクトファイル (elf 形式)
 - ap_sh2a_0a_hidh.mot … ROM 動作用モトローラ S フォーマット形式ファイル
 - ap_sh2a_0a_hidh.map … ROM 動作用マップファイル
- コンパイル後は、.o 等のファイルが生成されます

2. 2. 6 GCC 用 HID Class Device サンプルプログラム

GCC 用 HID Class Device 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¥hidd フォルダ内>

ap_sh2a_0a_hidd.hws … High-performance Embedded Workshop 用
ワークスペースファイル

<gcc¥hidd¥ap_sh2a_0a_hidd フォルダ内>

ap_sh2a_0a_hidd.hwp … High-performance Embedded Workshop 用
プロジェクトファイル

<gcc¥hidd¥ap_sh2a_0a_hidd¥src フォルダ内>

vector.c … 割込ベクタテーブル
boot.c … CPU 初期化処理
main.c … メイン処理
sci.c … シリアル処理
timer.c … タイマ処理
common.h … 共通ヘッダファイル
BoardDepend.h … ボード依存定義ファイル
7211.h … SH7211 内部レジスタ定義ヘッダファイル
UART.c … サンプルドライバ用シリアル入出力処理
UART.h … サンプルドライバ用シリアル入出力処理ヘッダファイル
HIDD.c … サンプルドライバ用 HIDD API 呼び出し処理
HIDD.h … サンプルドライバ用 HIDD API 呼び出し処理ヘッダファイル

<gcc¥hidd¥ap_sh2a_0a_hidd¥src¥SampleI¥フォルダ内>

UOX_User.h … S1R72U06 サンプルドライバユーザー定義ヘッダ
UOX_IF_API.h … S1R72U06 サンプルドライバ API 定義ヘッダ
UOX_IF_API_UART.c … S1R72U06 サンプルドライバ UART API ソース
UOX_IF_S10.h … S1R72U06 サンプルドライバ S10 定義ヘッダ
UOX_IF_S10_UART.c … S1R72U06 サンプルドライバ UART ドライバソース
UOX_IF_Port.h … S1R72U06 サンプルドライバ IO 定義ヘッダ
UOX_IF_Port.c … S1R72U06 サンプルドライバ IO ドライバソース

<gcc¥hidd¥ap_sh2a_0a_hidd¥src¥SampleII¥フォルダ内>

UOX_DEF.h … S1R72U06 サンプルドライバユーザー定義ヘッダ
UOX_Common.h … S1R72U06 サンプルドライバ API 定義ヘッダ
UOX.c … S1R72U06 サンプルドライバ UART API ソース
UOX.h … S1R72U06 サンプルドライバ IO 定義ヘッダ

<gcc¥hidd¥ap_sh2a_0a_hidd¥src¥MSCH¥フォルダ内>

UOX_HIDD.c … S1R72U06 サンプルドライバマスストレージクラスソース
UOX_HIDD.h … S1R72U06 サンプルドライバマスストレージクラスヘッダ

<gcc¥hidd¥ap_sh2a_0a_hidd¥Debug フォルダ内>

ap_sh2a_0a_hidd.x … RAM 動作用オブジェクトファイル(elf 形式)
ap_sh2a_0a_hidd.mot … RAM 動作用モトローラ S フォーマット形式ファイル
ap_sh2a_0a_hidd.map … RAM 動作用マップファイル
コンパイル後は、.o 等のファイルが生成されます

<gcc¥hidd¥ap_sh2a_0a_hidd¥Release フォルダ内>

| | | |
|---------------------|-----|-----------------------------|
| ap_sh2a_0a_hidd.x | ... | ROM 動作用オブジェクトファイル (elf 形式) |
| ap_sh2a_0a_hidd.mot | ... | ROM 動作用モトローラ S フォーマット形式ファイル |
| ap_sh2a_0a_hidd.map | ... | ROM 動作用マップファイル |

コンパイル後は、.o 等のファイルが生成されます

3. ビルド・デバッグ方法

各モードのビルド・デバッグ方法は共通です。

ここで表記されているファイル名は「ap_sh2a_0a_XXXX.hws」のようになっていますので
使用するモードによって「_XXXX」を「_msch」、「_hidh」、「_hidd」に当てはめて使用してください。

3. 1 SHC 用サンプルプログラム (Ver9 以降)

(1) ビルド

- ① High-performance Embedded Workshop を起動し、ap_sh2a_0a_XXXX.hwp を読み込みます。
- ② 最初の読み込みを行なったときに、「ワークスペース (Workspace) が移動しました」という内容の確認メッセージが表示されますので「はい」を選択してください。
- ③ 最初の読み込みを行なったときに、コンパイラバージョンによって、バージョンの選択を行なうダイアログが表示されることがあります。表示された場合には、使用するコンパイラバージョンを選択してください。
- ④ [Build] ボタン横のリストボックス [Configuration Section] から、[Debug] または [Release] を選択します。
[Debug] を選択した場合、¥Debug ワークフォルダ内に RAM 動作用のオブジェクトが生成されます。
[Release] を選択した場合、¥Release ワークフォルダ内に ROM 動作用のオブジェクトが生成されます。
- ⑤ メニューの [Build] - [Build] を実行してください。ap_sh2a_0a_XXXX.mot、ap_sh2a_0a_XXXX.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。

(2) RAM 上でのデバッグ

- ① AP-SH2A-0A のスイッチを「1.3 動作モード」の「デバッグ時」に設定します。
- ② XrossFinder を起動します。
このとき、System Setup ダイアログの CPU の欄には、7211F を選択します。
- ③ ¥Sample フォルダ直下にある XrossFinder_sh2a_0a.xfc コマンドファイルを読み込みます。
- ④ ¥Debug フォルダ内のオブジェクトファイル ap_sh2a_0a.abs を XrossFinder で読み込んで動作を確認して下さい。

(3) ROM 上でのデバッグ

- ① AP-SH2A-0A のスイッチを「1.3 動作モード」の「内蔵フラッシュ ROM 書き込み時」に設定します。
- ② XrossFinder を起動します。
このとき、System Setup ダイアログの CPU の欄には、7211F を選択します。
- ③ ¥Sample フォルダ直下にある XrossFinder_sh2a_0a.xfc コマンドファイルを読み込みます。
(コマンドファイルの PC、VBR 設定は RAM 動作設定になっていますので、ROM 動作設定に変更しておいて下さい)
- ④ ¥Release フォルダ内のオブジェクトファイル ap_sh2a_0a.abs を XrossFinder で読み込みます。
- ⑤ XrossFinder のメニューから FlashWriter EX を起動し、ボードにプログラムを書込みます。
FlashWriter EX for XrossFinder ダイアログの FlashROM の設定は、On-Chip を選択して下さい。
- ⑥ AP-SH2A-0A のスイッチを「1.3 動作モード」の「デバッグ時」に設定し、動作確認を行なって下さい。

(4) XrossFinder 未使用時の確認方法

FlashWriter EX を使用して ¥Release フォルダ内の ap_sh2a_0a_XXXX.mot をボードに書き込み、動作確認を行ってください。
XrossFinder、FlashWriter EX の使用方法につきましては、各製品のマニュアルを参照して下さい。
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。

3. 2 KPIT 社製 GNU GCC サンプルプログラム (GNUSH v0903 以降)

(1) ビルド・デバッグ

High-performance Embedded Workshop (KPIT GNUSH v0903 以降) を使用した場合のサンプルプログラムのビルド・デバッグ方法も 3. 1 章と同様です。

¥gcc フォルダ 直下にある ap_sha_0a_XXXX.hwp を読み込んでビルドを実行して下さい。また、ビルド後生成されたオブジェクトファイル ap_sh2a_0a_XXXX.x を XrossFinder で読み込んで動作を確認して下さい。

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

3. 3 サンプルプログラムの移植について

(1) 弊社 CPU ボードへの移植

弊社 CPU ボードへのサンプルプログラムへの移植には各 CPU ボードに添付されているサンプルプログラムを流用することが可能です。

シリアル通信プログラム (sci. c 等) とタイマ割り込み処理 (timer. c 等) のソースを変更してご使用ください。ただし、CPU 能力やメモリ容量等によってはうまく動作しない場合がありますのでご注意ください。

(2) その他の機器への移植

PC-HID-01 を動作させるためにはシリアル通信ソフトウェアとタイマ割り込みソフトウェアが必要となります。また、シリアル通信ではフロー制御が無いため十分なデータ受信能力が必要となります。

4. 動作説明

4. 1 Mass Storage Class Host 用サンプルプログラムの動作説明

4. 1. 1 HID Storage Class Host 用サンプルプログラム概要

USB マスストレージクラスホスト用サンプルプログラムは、下記の動作を行います。

- USB メモリにファイルを作成します。
ファイルの新規作成を行い、10KByte のダミーデータ書き込み終了します。

4. 1. 2 Mass Storage Class Host 用サンプルプログラム実行方法

1. PC-HID-01 の JSW1 を「HOST」に設定します。
2. PC-HID-01 に USB メモリを接続します。
3. AP-SH2A-0A 及び PC-HID-01 の電源を投入し、プログラムを実行します。
4. USB メモリの認識後、AP-SH2A-0A の LED が点灯しファイルを作成します。(ファイル名” FILE.TXT”)
5. ファイルアクセスが終了後、LED が消灯します。

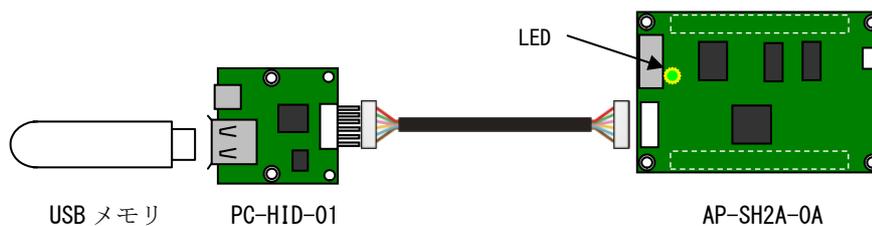


Fig 4.1-1 Mass Storage Class Host 用サンプルプログラム実行例

4. 2 HID Class Host 用サンプルプログラムの動作説明

4. 2. 1 HID Class Host 用サンプルプログラム概要

USB HID クラスホスト用サンプルプログラムは、下記の動作を行います。

- USB キーボードからのキー情報を受信します。
キー情報が受信されるごとにLED が点灯します。

4. 2. 2 HID Class Host 用サンプルプログラム実行方法

1. PC-HID-01 の JSW1 を「HOST」に設定します。
2. PC-HID-01 の CN2 に USB キーボードを接続します。
3. AP-SH2A-0A 及びPC-HID-01 の電源を投入し、プログラムを実行します。
4. USB キーボードの認識後、キーが押されると AP-SH2A-0A のLED が点灯します。

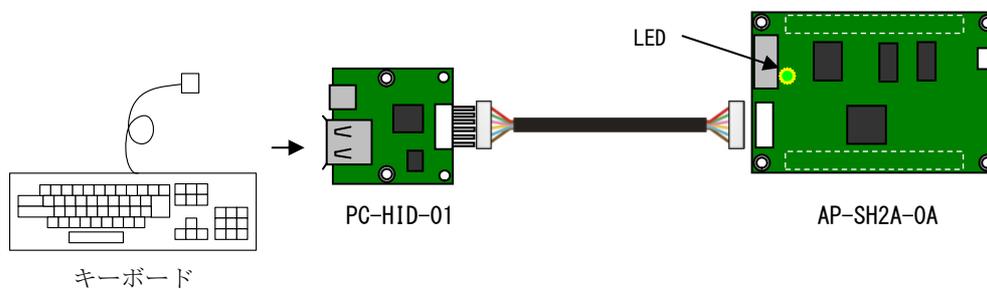


Fig 4.2-1 HID Class Host 用サンプルプログラム実行例

4. 3 HID Class Device 用サンプルプログラムの動作説明

4. 3. 1 HID Class Device 用サンプルプログラム概要

USB HID クラスデバイス用サンプルプログラムは、下記の動作を行います。

- PC から USB キーボードとして認識されます。
1 秒間隔で PC にキー情報(数字)が出力されます。

4. 3. 2 HID Class Device 用サンプルプログラム実行方法

1. PC-HID-01 の JSW1 を「DEVICE」に設定します。
2. PC-HID-01 の CN3 を使用し PC に USB ケーブルで接続します。(USB A コネクタ、USB miniB コネクタのケーブル使用)
3. PC にてメモ帳等のテキスト入力可能なアプリケーションを起動しておきます。
4. AP-SH2A-0A 及び PC-HID-01 の電源を投入し、プログラムを実行します。
5. PC のアプリケーションに文字情報が入力されます。(入力を確認するアプリケーションをアクティブにしてください)

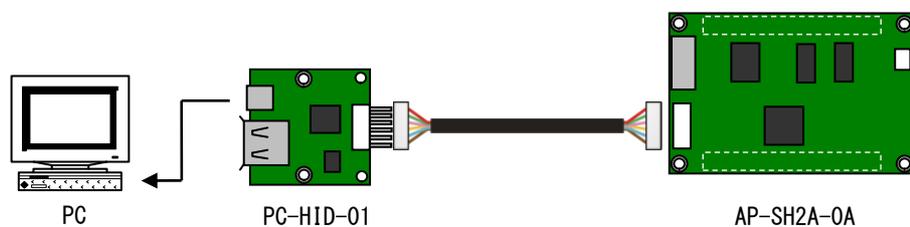


Fig 4.3-1 HID Class Device 用サンプルプログラム実行例

5. HID Class データフォーマット

5. 1 keyboard 入力データのフォーマット

Keyboard のデータフォーマット

| BYTE | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|--------|------|------|------|------|------|------|------|
| 内容 | 特殊 KEY | 0 固定 | KEY1 | KEY2 | KEY3 | KEY4 | KEY5 | KEY6 |

Table 5.1-1 Keyboard のデータフォーマット

特殊キーの割り当て

| Bit | 入力キー |
|-----|---------------|
| 7 | 右 Windows KEY |
| 6 | 右 Alt KEY |
| 5 | 右 Shift KEY |
| 4 | 右 Ctrl KEY |
| 3 | 左 Windows KEY |
| 2 | 左 Alt KEY |
| 1 | 左 Shift KEY |
| 0 | 左 Ctrl KEY |

Table 5.1-2 特殊キーの割り当て

Keyboard のデータフォーマットは 8Byte のデータで構成されており、先頭の 1Byte に Alt、Shift、Ctrl 等のキー情報が割り当てられており、3~8Byte の情報が通常のキー情報となります。

キー情報は最大 6 つのキーの同時押しまで対応し、現在押されているキーコードが KEY1 から KEY6 に格納されます。

また、キーが押された状態から離された場合にはキーコードが 0 のデータが通信されます。

5. 2 キーコード変換

キーコードはキーボードの位置情報であるため、通常の ASCII コードのような文字コードとは異なります。

キーボードのタイプごとにキーコードの配置が異なるため接続するキーボードによってキーコードの変換を行ってください。

ご注意

.

- ・本文書の著作権は（株）アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書に記載されているサンプルプログラムの著作権は（株）アルファプロジェクトが保有しますが、お客様のアプリケーションで使用される場合には、ご自由にご利用いただけます。
- ・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。
- ・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質問等のサポートは一切受け付けており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。万が一不審な点、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、万が一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。

- ・ Windows®の正式名称は Microsoft®Windows®Operating System です。
Microsoft、Windows、Windows NT は、米国 Microsoft Corporation.の米国およびその他の国における商標または登録商標です。
Windows®Vista、Windows®XP、Windows®2000 Professional は、米国 Microsoft Corporation.の商品名称です。
本文書では下記のように省略して記載している場合がございます。ご了承下さい。
Windows®Vista は Windows Vista もしくは WinVista
Windows®XP は Windows XP もしくは WinXP
Windows®2000 Professional は Windows 2000 もしくは Win2000
- ・ IBM-PC/AT は、米国 IBM 社の登録商標、商標または商品名称です。
- ・ DOS/V は、日本 IBM 社の登録商標、商標または商品名称です。
- ・ SuperH は、（株）Renesas の登録商標、商標または商品名称です。
- ・ その他の会社名、製品名は、各社の登録商標または商標です。



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