

## AP-SH2F-7A (SH7144 CPU BOARD)

## サンプルプログラム解説

第2版 2009年11月30日

## 1. 概要

## 1. 1 概要

本サンプルプログラムは、AP-SH2F-7A の動作を確認する簡単なプログラムです。また、弊社製 XrossFinder 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。サンプルプログラムは、周辺機能の中で使用頻度の高いと思われるシリアル通信、タイマ割り込み処理を行なっています。

## 1. 2 動作モード

本サンプルプログラムは、AP-SH2F-7A で動作します。

CPU モード、クロックモード、SRAM バス幅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-SH2F-7A 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                                      |
| クロックモード    | モード3(システムクロック=49.152MHz 周辺クロック=24.576MHz) |
| デバッグモード    | XrossFinder 使用時は ON に、未使用の場合は OFF として下さい。 |
| SRAM バス幅設定 | 8bit/16bit (出荷時設定は 16bit)                 |

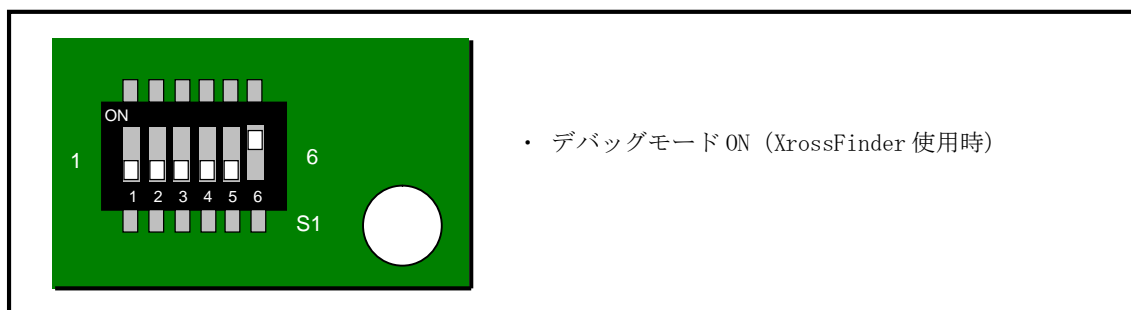


Fig 1.2-1 デバッグモード ON

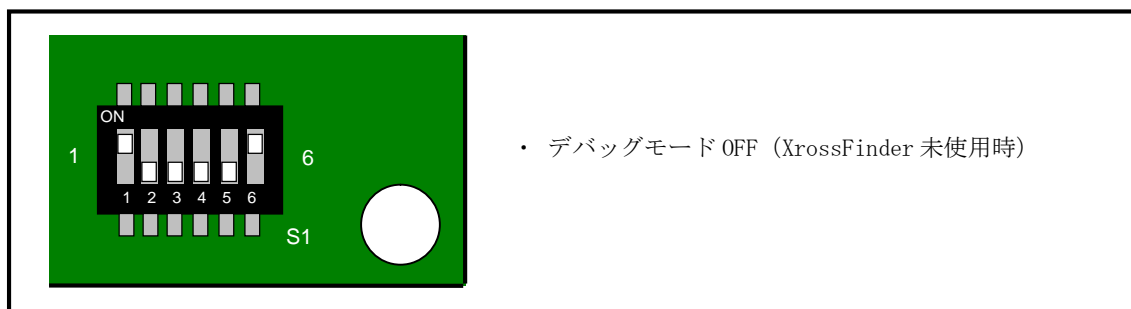


Fig 1.2-2 デバッグモード OFF

## 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 HEW を用いて開発されており、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 1.2 (release 9)<br>以降 | *1SHC<br>(ルネサステクノロジ) | V6.0.0<br>以降 | SuperH ファミリ用 C/C++ コンパイラパッケージに付属 |
| gcc  | HEW  | V4.05.01.001<br>以降      | *2GNUSH<br>(KPIT)    | V0702<br>以降  | ELF 形式、HEW 対応                    |

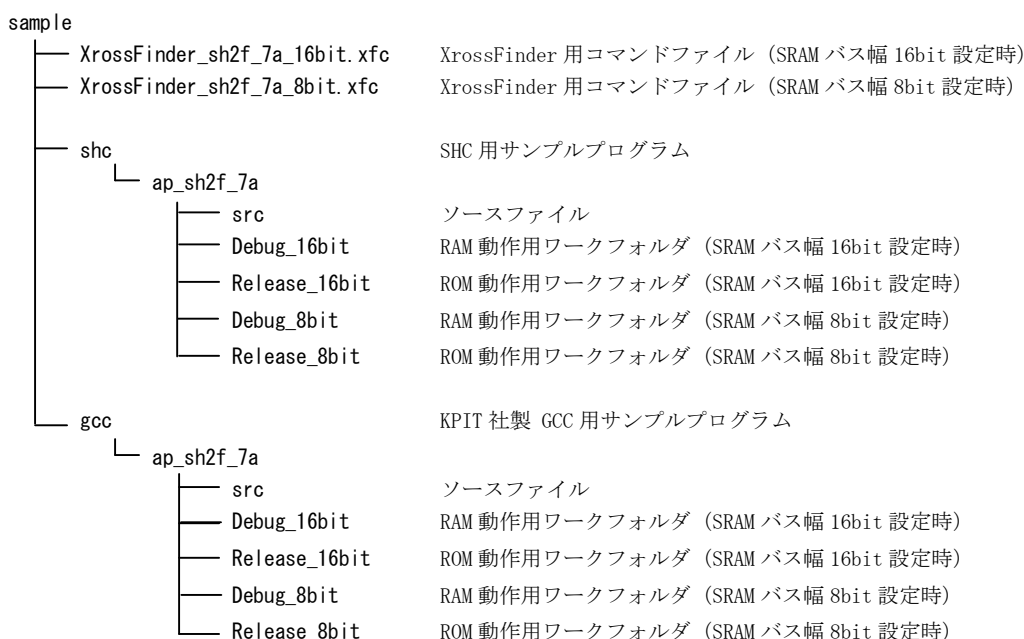
※1: 「SuperH ファミリ用 C/C++ コンパイラパッケージ」です。ルネサステクノロジ社のウェブサイトより評価版をダウンロードできます。

※2: KPIT 社製「GCC」です。GNUSH 及び HEW 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。

## 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

### (1) SHC 用サンプルプログラム

SHC 用サンプルプログラムは以下のファイルで構成されています。

#### <shc¥フォルダ内>

ap\_sh2f\_7a.hws … HEW 用ワークスペースファイル

#### <shc¥ap\_sh2f\_7a¥フォルダ内>

ap\_sh2f\_7a.hwp … HEW 用プロジェクトファイル

#### <shc¥ap\_sh2f\_7a¥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  
 vector.c … 割込みベクタテーブル  
 common.h … 共通ヘッダファイル  
 7144S.H … SH7144 シリーズ内部レジスタ定義ヘッダファイル  
 section.src … セクション定義

#### <shc¥ap\_sh2f\_7a¥Debug\_16bit フォルダ内>

ap\_sh2f\_7a.abs … RAM 動作用デバッグ情報ファイル(ELF 形式)  
 ap\_sh2f\_7a.mot … RAM 動作用モトローラ S フォーマット形式ファイル  
 ap\_sh2f\_7a.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¥ap\_sh2f\_7a¥Release\_16bit フォルダ内>

ap\_sh2f\_7a.abs … ROM 動作用デバッグ情報ファイル(ELF 形式)  
 ap\_sh2f\_7a.mot … ROM 動作用モトローラ S フォーマット形式ファイル  
 ap\_sh2f\_7a.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

#### <shc¥ap\_sh2f\_7a¥Debug\_8bit フォルダ内>

ap\_sh2f\_7a.abs … RAM 動作用デバッグ情報ファイル(ELF 形式)  
 ap\_sh2f\_7a.mot … RAM 動作用モトローラ S フォーマット形式ファイル  
 ap\_sh2f\_7a.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¥ap\_sh2f\_7a¥Release\_8bit フォルダ内>

ap\_sh2f\_7a.abs … ROM 動作用デバッグ情報ファイル(ELF 形式)  
 ap\_sh2f\_7a.mot … ROM 動作用モトローラ S フォーマット形式ファイル  
 ap\_sh2f\_7a.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) GCC 用サンプルプログラム

GCC 用サンプルプログラムは以下のファイルで構成されています。

## &lt;gcc¥フォルダ内&gt;

ap\_sh2f\_7a.hws … HEW 用ワークスペースファイル

## &lt;gcc¥ap\_sh2f\_7a¥フォルダ内&gt;

ap\_sh2f\_7a.hwp … HEW 用プロジェクトファイル  
 Debug session.hsf … Debug session 情報ファイル  
 Release session.hsf … Release session 情報ファイル

## &lt;gcc¥ap\_sh2f\_7a¥src フォルダ内&gt;

main.c … メイン処理  
 tmr.c … タイマ処理  
 sci.c … シリアル処理  
 vector.c … 割込みベクタテーブル  
 common.h … 共通ヘッダファイル  
 7144S.H … SH7144 シリーズ内部レジスタ定義ヘッダファイル  
 crt0.s … セクション定義

## &lt;gcc¥ap\_sh2f\_7a¥Debug\_16bit フォルダ内&gt;

ap\_sh2f\_7a.x … RAM 動作用デバッグ情報ファイル(ELF 形式)  
 ap\_sh2f\_7a.mot … RAM 動作用モトローラ S フォーマット形式ファイル  
 ap\_sh2f\_7a.map … RAM 動作用マップファイル  
 gnuconfig.ini … GNUSH 構成設定ファイル  
 コンパイル後は「.o」等のファイルが生成されます

## &lt;gcc¥ap\_sh2f\_7a¥Release\_16bit フォルダ内&gt;

ap\_sh2f\_7a.x … ROM 動作用デバッグ情報ファイル(ELF 形式)  
 ap\_sh2f\_7a.mot … ROM 動作用モトローラ S フォーマット形式ファイル  
 ap\_sh2f\_7a.map … ROM 動作用マップファイル  
 gnuconfig.ini … GNUSH 構成設定ファイル  
 コンパイル後は「.o」等のファイルが生成されます

## &lt;gcc¥ap\_sh2f\_7a¥Debug\_8bit フォルダ内&gt;

ap\_sh2f\_7a.x … RAM 動作用デバッグ情報ファイル(ELF 形式)  
 ap\_sh2f\_7a.mot … RAM 動作用モトローラ S フォーマット形式ファイル  
 ap\_sh2f\_7a.map … RAM 動作用マップファイル  
 gnuconfig.ini … GNUSH 構成設定ファイル  
 コンパイル後は「.o」等のファイルが生成されます

## &lt;gcc¥ap\_sh2f\_7a¥Release\_8bit フォルダ内&gt;

ap\_sh2f\_7a.x … ROM 動作用デバッグ情報ファイル(ELF 形式)  
 ap\_sh2f\_7a.mot … ROM 動作用モトローラ S フォーマット形式ファイル  
 ap\_sh2f\_7a.map … ROM 動作用マップファイル  
 gnuconfig.ini … GNUSH 構成設定ファイル  
 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

ご注意：CD-ROM からサンプルプログラムをコピーする場合、コピー先のファイル属性が読み取り専用となっています。  
すべてのファイルの読み取り専用属性を解除してからビルドを実行して下さい。

#### 3. 1 SRAM バス幅 16bit 設定時の SHC 用サンプルプログラム (Ver6 以降)

##### (1) ビルド

HEW (Renesas SHC Ver6 以降) を使用した場合のサンプルプログラムのビルド方法を説明します。

- ① HEW を起動し、ap\_sh2f\_7a.hws 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\_16bit] または [Release\_16bit] を選択します。  
[Debug\_16bit] を選択した場合、¥Debug\_16bit 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\_16bit] を選択した場合、¥Release\_16bit 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] を実行してください。ap\_sh2f\_7a.mot、ap\_sh2f\_7a.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-SH2F-7A のスイッチを「1.2 動作モード」「Fig1.2-1 デバッグモード ON」に合わせて設定します。
- ② XrossFinder を起動し、¥sample フォルダ直下にある XrossFinder\_sh2f\_7a\_16bit.xfc コマンドファイルを読み込みます。
- ③ ¥Debug\_16bit フォルダ内の ap\_sh2f\_7a.abs を XrossFinder でダウンロードして動作を確認して下さい。

##### (3) ROM 上でのデバッグ

- ① AP-SH2F-7A のスイッチを「1.2 動作モード」「Fig1.2-1 デバッグモード ON」に合わせて設定します。
- ② XrossFinder を起動し、¥sample フォルダ直下にある XrossFinder\_sh2f\_7a\_16bit.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)
- ③ ¥Release\_16bit フォルダ内の ap\_sh2f\_7a.abs を XrossFinder で読み込みます。
- ④ FlashWriter EX を起動し、ボードにプログラムを書込んで動作を確認して下さい。

##### (4) XrossFinder 未使用時の確認方法

FlashWriter EX を使用して ¥Release\_16bit フォルダ内の ap\_sh2f\_7a.mot をボードに書き込み、動作確認を行ってください。

FlashWriter EX の使用方法につきましては、アプリケーションノート「AN141 ボード付属 FlashWriterEX を使った内蔵 FlashROM への書き込み方法」を参照して下さい。

なお、HEW の詳細な使用方法につきましては、HEW のマニュアルを参照して下さい。

### 3. 2 SRAM バス幅 16bit 設定時の KPIT 社製 GNU GCC サンプルプログラム (GNUSH v0702 以降)

#### (1) ビルド・デバッグ

HEW (KPIT GNUSH v0702 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 節と同様です。  
¥gcc フォルダ 直下にある ap\_sh2f\_7a.hws を読み込んでビルドを実行して下さい。また、最初の読み込みを行なったときに「以下のセッションは、見つけることができないか、またはこれよりも新しいバージョンの HEW で作成されています」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。その後、ビルド後生成された ap\_sh2f\_7a.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 HEW 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下さい。

### 3. 3 SRAM バス幅 8bit 設定時の SHC 用サンプルプログラム (Ver6 以降)

#### (1) ビルド

HEW (Renesas SHC Ver6 以降) を使用した場合のサンプルプログラムのビルド方法を説明します。

- ① HEW を起動し、ap\_sh2f\_7a.hws 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\_8bit] または [Release\_8bit] を選択します。  
[Debug\_8bit] を選択した場合、¥Debug\_8bit 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\_8bit] を選択した場合、¥Release\_8bit 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] を実行してください。ap\_sh2f\_7a.mot、ap\_sh2f\_7a.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-SH2F-7A のスイッチを「1.2 動作モード」「Fig1.2-1 デバッグモード ON」に合わせて設定します。
- ② XrossFinder を起動し、¥sample フォルダ直下にある XrossFinder\_sh2f\_7a\_8bit.xfc コマンドファイルを読み込みます。
- ③ ¥Debug\_8bit フォルダ内の ap\_sh2f\_7a.abs を XrossFinder でダウンロードして動作を確認して下さい。

#### (3) ROM 上でのデバッグ

- ① AP-SH2F-7A のスイッチを「1.2 動作モード」「Fig1.2-1 デバッグモード ON」に合わせて設定します。
- ② XrossFinder を起動し、¥sample フォルダ直下にある XrossFinder\_sh2f\_7a\_8bit.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)
- ③ ¥Release\_8bit フォルダ内の ap\_sh2f\_7a.abs を XrossFinder で読み込みます。
- ④ FlashWriter EX を起動し、ボードにプログラムを書込んで動作を確認して下さい。

#### (4) XrossFinder 未使用時の確認方法

FlashWriter EX を使用して ¥Release\_8bit フォルダ内の ap\_sh2f\_7a.mot をボードに書き込み、動作確認を行ってください。

FlashWriter EX の使用方法につきましては、アプリケーションノート「AN141 ボード付属 FlashWriterEX を使った内蔵 FlashROM への書き込み方法」を参照して下さい。

なお、HEW の詳細な使用方法につきましては、HEW のマニュアルを参照して下さい。

### 3. 4 SRAM バス幅 8bit 設定時の KPIT 社製 GNU GCC サンプルプログラム (GNUSH v0702 以降)

#### (1) ビルド・デバッグ

HEW (KPIT GNUSH v0702 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 節と同様です。  
 %gcc フォルダ 直下にある ap\_sh2f\_7a.hws を読み込んでビルドを実行して下さい。また、最初の読み込みを行なったときに「以下のセッションは、見つけることができないか、またはこれよりも新しいバージョンの HEW で作成されています」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。その後、ビルド後生成された ap\_sh2f\_7a.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 HEW 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下さい。

## 4. 動作説明

### 4. 1 サンプルプログラムの動作

本サンプルプログラムは、下記の動作を行ないます。

- SC11 でエコーバックを行ないます。(送受信割り込み使用)  
 SC11 から受信をした値をそのまま、SC11 へ送信します。  
 シリアルの設定は、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等) を使用して下さい。
- LD1 (緑の LED) を 1sec 周期で ON/OFF します。
- CN1、CN2 の次のポートより方形波を出力します。周期とピン番号を次に示します。

CN1 方形波出力端子一覧

| ピン番号 | ピン名       | 周期     | 備考     |
|------|-----------|--------|--------|
| 23   | IREQ3/PB5 | 10msec | MTU 使用 |
| 24   | PB4       | 10msec | MTU 使用 |
| 25   | SDA0/PB3  | 10msec | MTU 使用 |
| 26   | SCL0/PB2  | 10msec | MTU 使用 |

CN2 方形波出力端子一覧

| ピン番号 | ピン名         | 周期     | 備考     |
|------|-------------|--------|--------|
| 1    | PE15/TIOC4D | 1sec   | MTU 使用 |
| 31   | PA15/CK     | 20msec | CMT 使用 |
| 36   | PA10/CS0    | 20msec | CMT 使用 |
| 37   | PA9/IRQ3    | 20msec | CMT 使用 |
| 38   | PA8/IRQ2    | 20msec | CMT 使用 |
| 41   | PA7/CS3     | 20msec | CMT 使用 |
| 42   | PA6/CS2     | 20msec | CMT 使用 |
| 43   | PA5/IRQ1    | 20msec | CMT 使用 |
| 46   | PA2/IRQ0    | 20msec | CMT 使用 |
| 47   | PA1/TXD0    | 20msec | CMT 使用 |
| 48   | PA0/RXD0    | 20msec | CMT 使用 |



### 4. 2 RAM 動作時のメモリマップ (SRAM バス幅 16bit 設定時)

SRAM バス幅 16bit 時のメモリマップを次に示します。

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

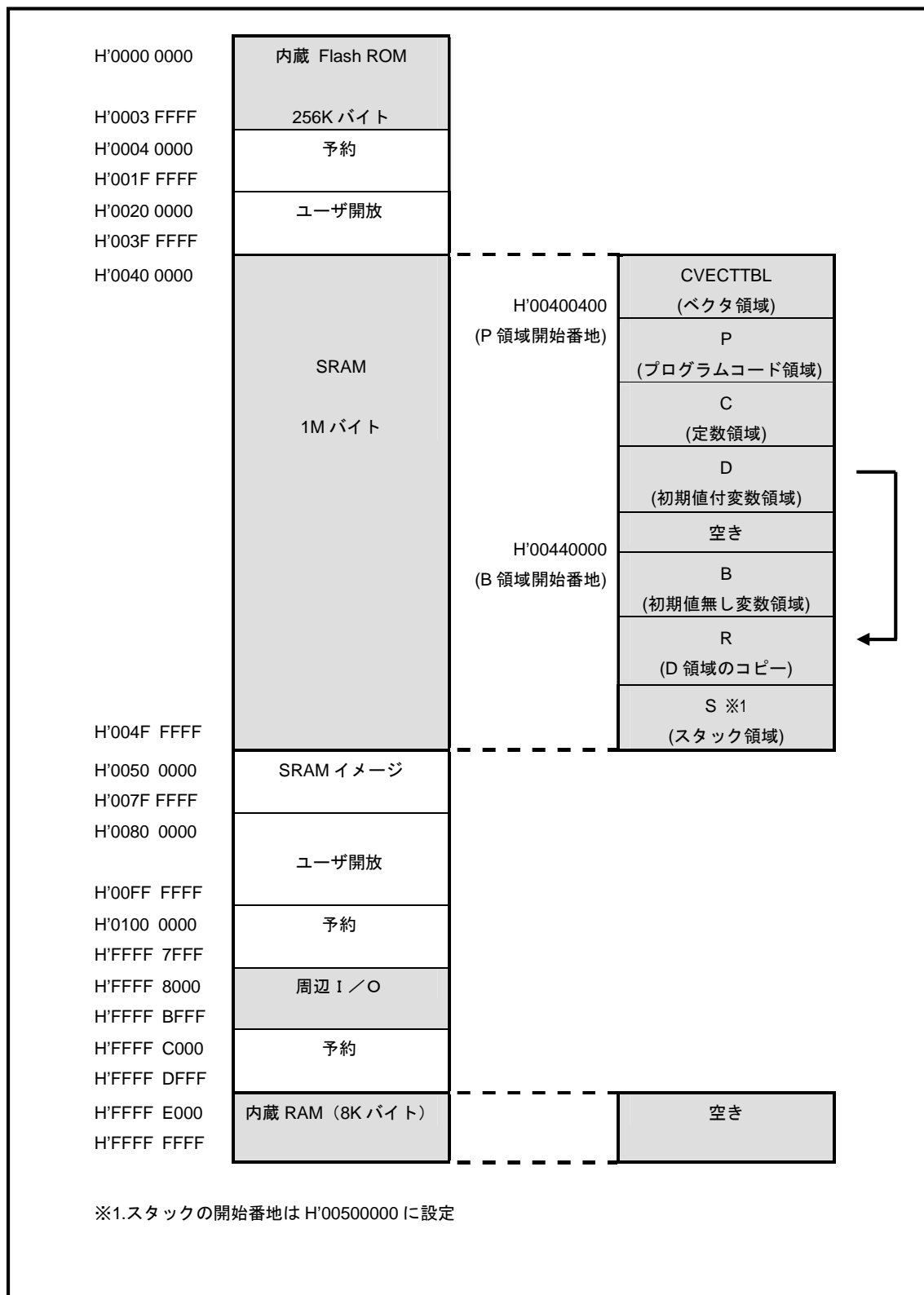


Fig 4.2-1 RAM 動作時のメモリマップ (16bit)

### 4. 3 ROM 動作時のメモリマップ (SRAM バス幅 16bit 設定時)

SRAM バス幅 16bit 時のメモリマップを次に示します。

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

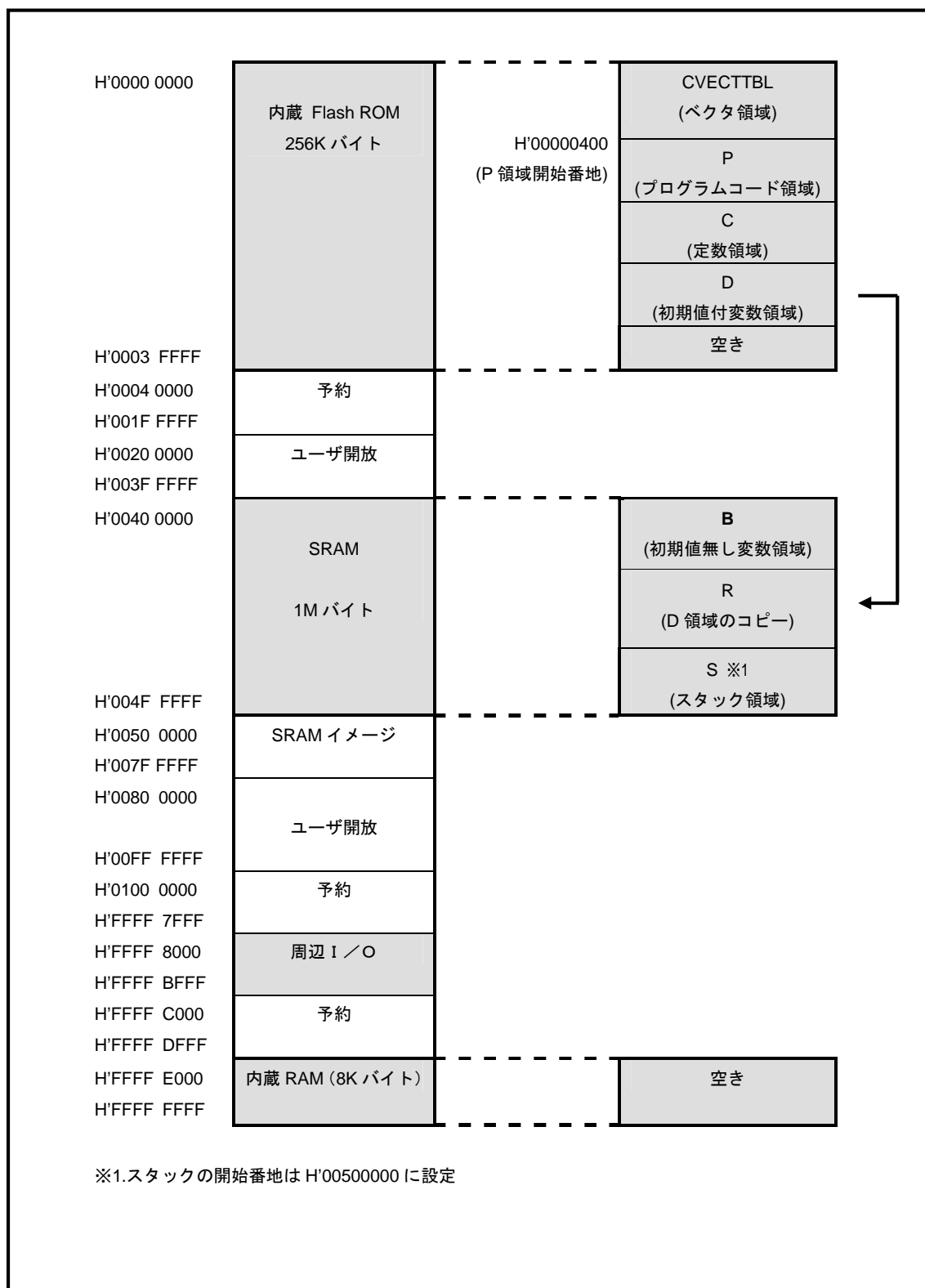


Fig 4.3-1 ROM 動作時のメモリマップ(16bit)

#### 4. 4 RAM 動作時のメモリマップ (SRAM バス幅 8bit 設定時)

SRAM バス幅 8bit 時のメモリマップを次に示します。

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

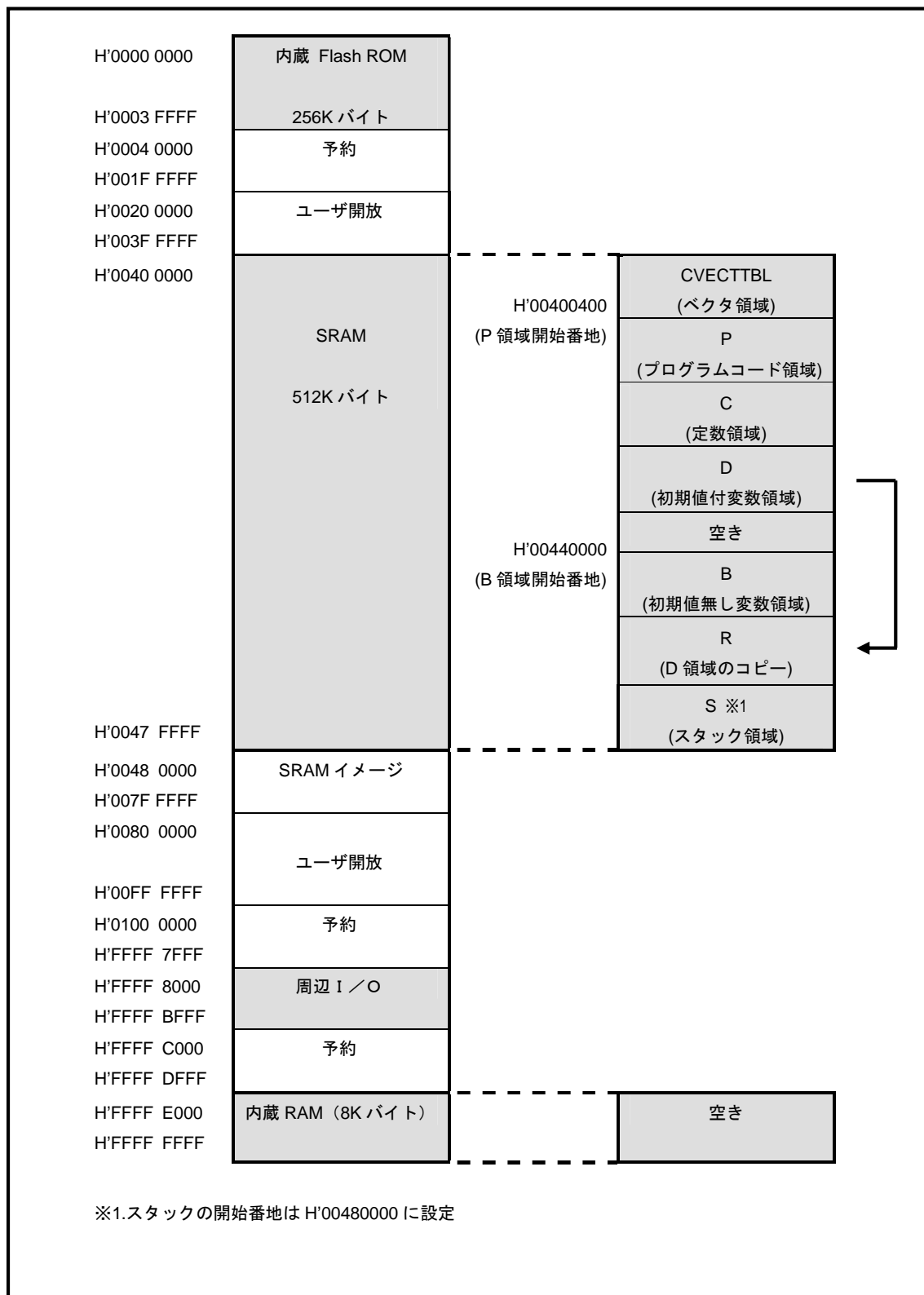


Fig 4.4-1 RAM 動作時のメモリマップ(8bit)

### 4. 5 ROM 動作時のメモリマップ (SRAM バス幅 8bit 設定時)

SRAM バス幅 8bit 時のメモリマップを次に示します。

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

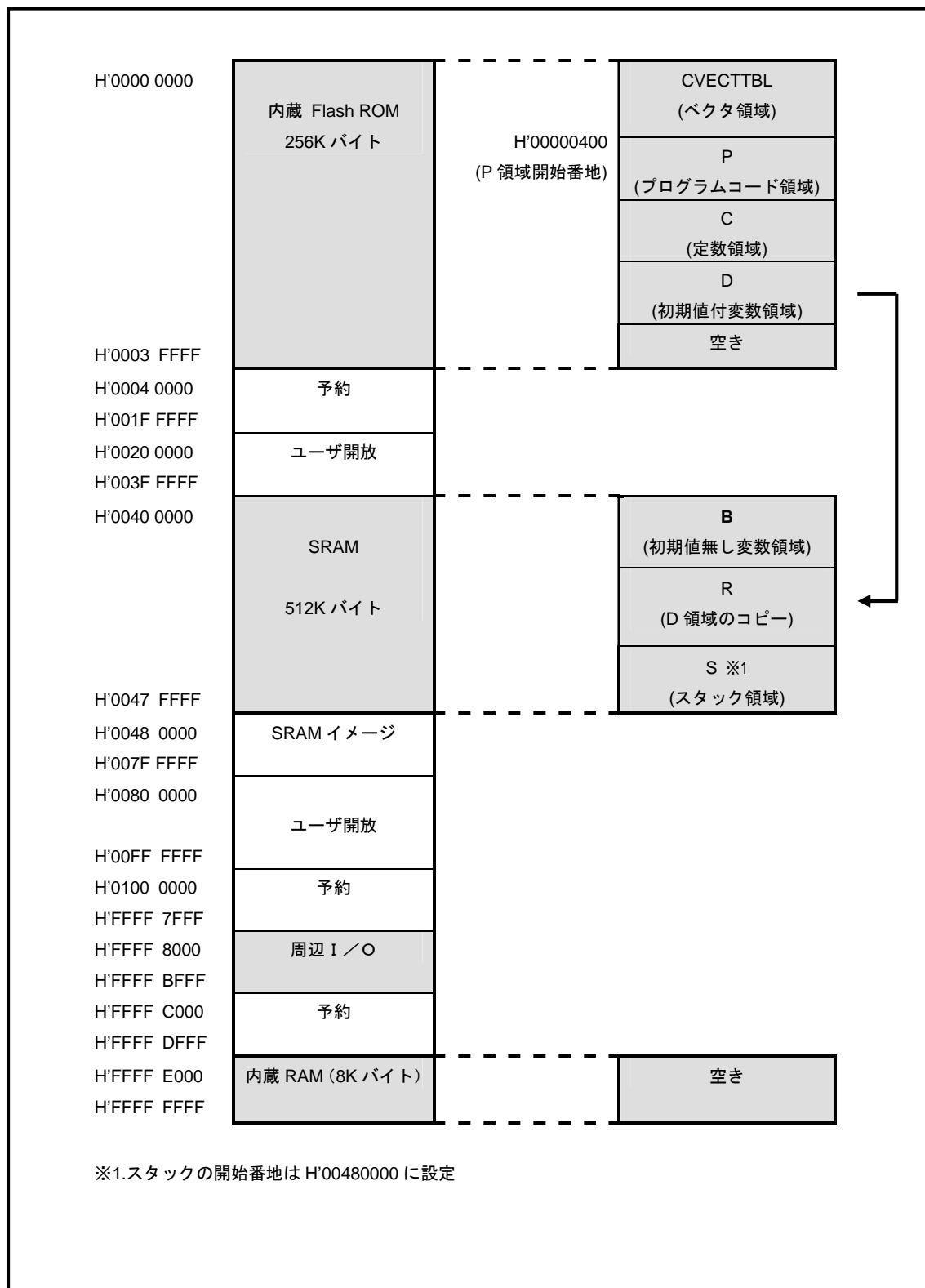


Fig 4.5-1 ROM 動作時のメモリマップ(8bit)

## 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、Windows®Millennium Edition、Windows®98 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®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
- ・ Windows®Millennium Edition は Windows Me もしくは WinMe
- ・ Windows®98 は Windows 98 もしくは Win98
- ・ High-performance Embedded Workshop は HEW
- ・ 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](mailto:query@apnet.co.jp)