

SHマイコン開発支援パッケージ

SH2 7045F Starter KIT

ユーザーズマニュアル

6 版 2003 / 4 / 10

ALPHA PROJECT Co., LTD

SH2 7045F Starter KIT User's Manual

この度は、当社製品「SH2 7045F Starter KIT」をご購入いただき、誠にありがとうございます。

本製品はSH2の性能評価、開発環境等をお試しいただく為に、必要なツール類をパッケージした製品です。

本製品をお役立ていただくために添付マニュアルを十分お読みくださいますようお願いいたします。

★本製品の内容及び仕様は予告なしに変更されることがありますのでご了承ください。

★本マニュアルに記載される会社名、商品名は各社の商標及び登録商標です。

お使いになる前に

<梱包の確認>

次のものが揃っているかどうかを確認して下さい。

万一足りないものがあれば、販売店まですみやかにご連絡下さい。

*AP-SH2F-0A CPUボード	1枚
*RS232Cケーブル	1本
*SH2 7045F Starter Kit CD-ROM	1枚
*ユーザー登録カード	1枚

お問い合わせ先

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

FAX (053) 401-0035 技術部 担当者宛

E-MAIL query@apnet.co.jp

目次

第1章 概要

1

1. 1 概要	1
1. 2 マニュアル構成	1
1. 3 使用環境	1
1. 4 CD-ROMの内容	2
1. 5 各ソフトウェアの機能	3
1. 5. 1 VisualMonitor	3
1. 5. 2 FlashWriter	3
1. 5. 3 GCC	3

第2章 ソフトウェアのインストール

4

2. 1 準備するもの	4
2. 2 デバッガ(VisualMonitor)のインストール	4
2. 2. 1 コントロールソフトのインストール	4
2. 2. 2 ターゲットモニタソフトのインストール	4
2. 2. 3 ディレクトリ構成	5
2. 2. 4 デバッガの動作確認	5
2. 3 F-Z T A T 書込ソフト(FlashWriter)のインストール	5
2. 4 コンパイラ(gcc)のインストール	5

第3章 チュートリアル

6

3. 1 サンプルプログラム	6
3. 1. 1 サンプルプログラムの構成	6
3. 1. 2 サンプルプログラムのコンパイル (GCC)	8
3. 1. 3 サンプルプログラムのコンパイル (日立C)	13
3. 2 リモートデバッガ	14
3. 2. 1 デバッグ環境の準備	14
3. 2. 2 リモートデバッガの起動と使用方法	14
3. 2. 3 その他の機能	20
3. 2. 4 サンプルプログラムのメモリマップ	21
3. 2. 5 VisualMonitor 正規版との違い	22

第4章. 内蔵フラッシュROM	23
------------------------	-----------

4. 1 内蔵フラッシュROM書き込み方法	-----	23
-----------------------	-------	----

第5章. ROM化	25
------------------	-----------

5. 1 ユーザプログラムのROM化	-----	25
5. 1. 1 ROM化用ファイル	-----	25
5. 1. 2 ROM化プログラムの構造	-----	26
5. 1. 3 ROM化の手順	-----	26

第6章. その他	27
-----------------	-----------

6. 1 付属品	-----	27
6. 1. 1 RS232Cケーブル	-----	27

第7章. サポート	28
------------------	-----------

7. 1 質問の受け付け	-----	28
7. 2 バージョンアップサービス	-----	28

第8章. Q&A	29
---------------------	-----------

第1章 概要

1.1 概要

本キットは、主に以下の製品から構成されます。

- ・ SH7045F CPUボード「AP-SH2F-0A」
- ・ リモートデバッガ 「VisualMonitor Ver.2 パッケージ限定版」
- ・ F-Z T A T 書込ツール 「Flash Writer for 7045」
- ・ GNU Cコンパイラ 「GNUPro99r1p1 for SH」

1.2 マニュアル構成

本キットのマニュアルは、以下のマニュアルから構成されます。

- | | |
|------------------------------------|----------------------------|
| 「SH2 7045F Starter KIT ユーザーズマニュアル」 | コンパイルからデバッガの使用法までの簡易マニュアル。 |
| 「SH2 7045F Starter KIT CPUボード編」 | CPUボードのハードウェアマニュアル |
| 「SH2/SH3 Starter KIT Cコンパイラ編」 | gccの解説マニュアル |
| 「SH2/SH3 Starter KIT リモートデバッガ編」 | VisualMonitorの解説マニュアル |

1.3 使用環境

	使用機器等	環 境
ホ ス ト	パーソナルコンピュータ	PC/AT 互換機
	OS	WINDOWS 95 / 98 / NT 4.0 / 2000
	メモリ	32Mバイト以上を推奨
	ハードディスク	VisualMonitor 15Mバイト以上の空き領域 GNUPro 50Mバイト以上の空き領域
	表示	800×600 以上
	CDドライブ	CD-ROM読み込み可能なドライブ
	その他	シリアルポート1CH
タ ー ゲ ッ ト	ターゲットボード	AP-SH2F-0A (SH7045Fボード)
	CPUクロック	28.6MHz
	モニタコード	内蔵フラッシュに書込済み
	モニタ使用RAMサイズ	約4Kバイト
	シリアルポート	ボード上のD-SUBコネクタに接続 38.4Kbps固定
そ の 他	RS232Cケーブル	添付のクロスケーブルを使用
	電源	5V / 500mA以上の安定化電源

・GCCコンパイラは、WINDOWSのDOSプロンプトで動作します。

注1) 一部のCDROMドライブではCD-Rメディアを安定して読み出せないものがあります。
その場合には、他のパソコンでMOやFDにバックアップして利用するなどしてください。

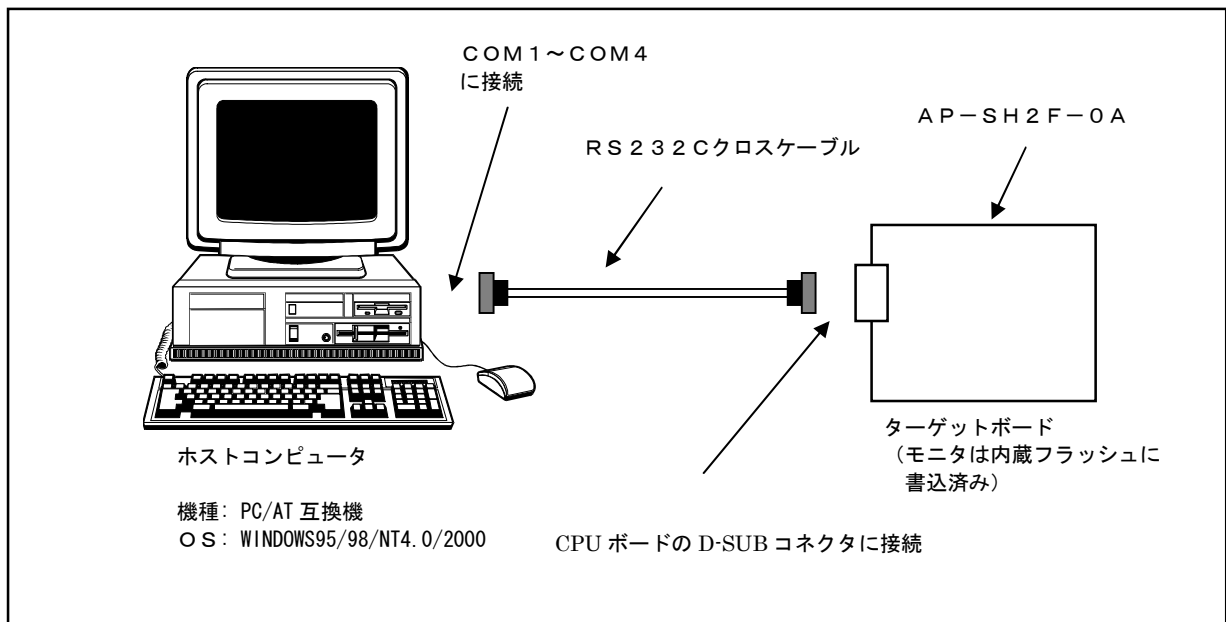


図1. 3-1 本キットでのデバッグ環境の構成例

1.4 CD-ROMの内容

添付のCD-ROMには以下のものが収録されています。

¥	VM	INSTALL	VisualMonitor セットアップ (VisualMonitor 本体) (モニタープログラム) (サンプルプログラム)
		REALi	REALi サンプルプログラム
	FW		FlashWriter for 7045
	GNUPro		GNU C コンパイラ (GCC), GNU ツール
	DJGPP		DJGPP 関連ツール
	SAMPLE		ROM 化サンプルプログラム
	SH_MAN		SH7040 シリーズ各種マニュアル
	MAN		本キットの各種マニュアル
	SCH		回路図
	ACROREAD		Adobe Acrobat Reader

1. 5 各ソフトウェアの機能

1. 5. 1 VisualMonitor

SH用に開発されたリモートモニタです。

gccと日立Cに対応し、従来にない高機能デバッグ環境を提供します。

本製品に添付されているVisualMonitorは、AP-SH2F-0Aでのみ動作します。

メモリマッピングは、内蔵フラッシュにVisualMonitorが配置され、RAM上にユーザプログラムが配置されます。

1. 5. 2 FlashWriter

F-ZTAT対応SHマイコンの内蔵フラッシュ書き込みソフトです。

本製品に添付されているFlashWriterはSH7045Fのみに対応しています。

本ソフトウェアは、お客様のターゲットボードでも使用可能ですが、動作保証はいたしません。

1. 5. 3 GCC

本製品には、レッドハット社のGNUPro@が収録されています。

GNUProは、GNUソフトウェアをベースとしたアプリケーション開発ツールキットです。

GNUProには、gccのほかにgdb（デバッグ）やオリジナルのツール類などが含まれています。

本製品に添付されているパッケージは、ホストがWindowsでターゲットはSH1/SH2/SH3/SH3eとなっています。機能的な制限は全くなくROM化まで可能です。

ただし、サポート権はありませんので、サポートが必要なお客様はレッドハット社より直接GNUProを購入していただく必要があります。

第2章. ソフトウェアのインストール

2. 1 準備するもの

インストールを始める前に以下のものを準備して下さい。

- ① パーソナルコンピュータ
前項の使用環境を御覧下さい。
- ② ターゲット
CPUボード「AP-SH2F-0A」
- ③ インストールディスク
本キットに添付されているCD-ROM
- ④ RS232Cケーブル
添付のクロスケーブルをご使用ください。
- ⑤ 電源
5V / 500mA以上の電源

2. 2 デバッガ(VisualMonitor)のインストール

VisualMonitor は、PC上で動作するコントロールソフトと、ターゲットボード上のROMに載せるターゲットモニタソフトの2種類のソフトウェアで構成されます。

2. 2. 1 コントロールソフトのインストール

- ① CD-ROM内の¥Vm¥ Install¥SETUP.EXE を実行して下さい。
- ② セットアップウィザードに従って、プログラムをインストールして下さい。

2. 2. 2 ターゲットモニタソフトのインストール

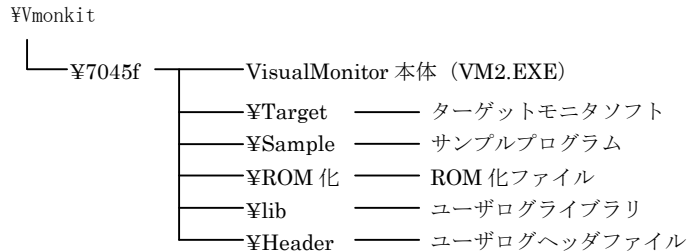
ターゲットモニタソフトは、VisualMonitor のインストールフォルダ以下の¥Target¥7045fkit.sr です。

ターゲットモニタソフトは、ご購入時に既に内蔵フラッシュROMに書き込まれていますので、特にインストール等は必要ありません。

誤ってターゲットモニタソフトを消去してしまった場合には、付属のF l a s h W r i t e r を使用して 7045fkit.sr を内蔵フラッシュに書き込んでください。

2. 2. 3 ディレクトリ構成

¥Vmonkit¥7045f 以下にインストールした場合、ディレクトリ構成は以下のようになります。



2. 2. 4 デバッガの動作確認

- ① PCとターゲットを添付のRS232Cクロスケーブルで接続します。
- ② VM2.exe を実行すると、VisualMonitor が起動しメインウィンドウが表示されます。
- ③ [システム]メニューから[ポート設定]を選択し、使用するCOMポート番号を設定します。
- ④ ターゲットの電源を投入するかリセットをすると、正常であればメッセージウィンドウが開き、「ターゲットがリセットされました」のメッセージが表示されます。メッセージが表示されない場合には、COMポート番号を再度確認して下さい。

2. 3 F-Z T A T書込ソフト(FlashWriter)のインストール

特にインストーラは用意されておりませんので、¥FW以下のFW7045.EXEを適当なフォルダにコピーしてください。

2. 4 コンパイラ(gcc)のインストール

¥GNUPro以下のSetup.exeを実行してください。

インストール方法の詳細は、「コンパイラ編」をご覧ください。

第3章. チュートリアル

本章では、サンプルプログラムを使用して、コンパイルからデバッガの使い方までを簡単に説明します。

サンプルプログラムの内容については、プログラム中のコメントを参考にしてください。

3. 1 サンプルプログラム

3. 1. 1 サンプルプログラムの構成

サンプルプログラムは、以下の処理をおこなっています。

- ・ ユーザログの出力
- ・ cmt0 で 10ms の割り込みを発生させ、PE0～PE15 を 10msec 間隔で ON/OFF します。
- ・ mtu0 で 5ms の割り込みを発生させ、PB2～5 を 5msec 間隔で ON/OFF します。

サンプルプログラムはGCC用と日立C用の2種類が用意されています。

サンプルプログラムの構成は次のようになっています。(VisualMnitorを¥Vmonikit¥7045fディレクトリにインストールした場合)

<GCC用> ¥Vmonikit¥7045f¥Sample¥gcc

プログラムソースファイル	
sh7040s.h	SH7040 シリーズレジスタ定義
gmachine.h	SH2 MPU 依存命令ヘッダファイル定義
gmachine.c	SH2 MPU 依存命令
crt0.s	スタートアップルーチン
main.c	メイン処理ルーチン
UserLog.h	USERLOG 関数ライブラリヘッダ
Sh2ul.a	USERLOG 組み込み関数ライブラリ (¥Vmonikit¥7045f¥lib に格納)
コンパイル用ファイル	
sh2f_0a.bat	コンパイル、リンクのバッチファイル
makefile	make 用定義ファイル
sh2f_0a.p	コンパイル用パラメータファイル
sh2f_0a.x	リンカスクリプトファイル
コンパイル出力ファイル	
sh2f_0a.out	実行ファイル (デバック情報有り)
sh2f_0a.map	マップファイル

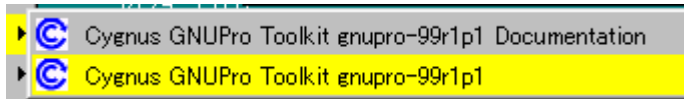
<日立C用> ¥Vmonikit¥7045f¥Sample¥hitachi

プログラムソースファイル	
sh7040s.h	SH7040Aシリーズレジスタ定義
gmachine.h	SH2 MPU 依存命令ヘッダファイル定義
gmachine.c	SH2 MPU 依存命令
section.src	セクション定義ファイル
main.c	メイン処理ルーチン
UserLog.h	USERLOG 関数ライブラリヘッダ
Sh2ul.lib sh2uld.lib (Ver6 用)	USERLOG 組み込み関数ライブラリ (¥Vmonikit¥7045f¥lib に格納)
コンパイル用ファイル	
makeall.bat makeall6.bat (Ver6 用)	コンパイル、リンクのバッチファイル
コンパイル出力ファイル	
sh2f_0a.abs	実行ファイル (デバック情報有り)
sh2f_0a.map	マップファイル

3. 1. 2 サンプルプログラムのコンパイル (GCC)

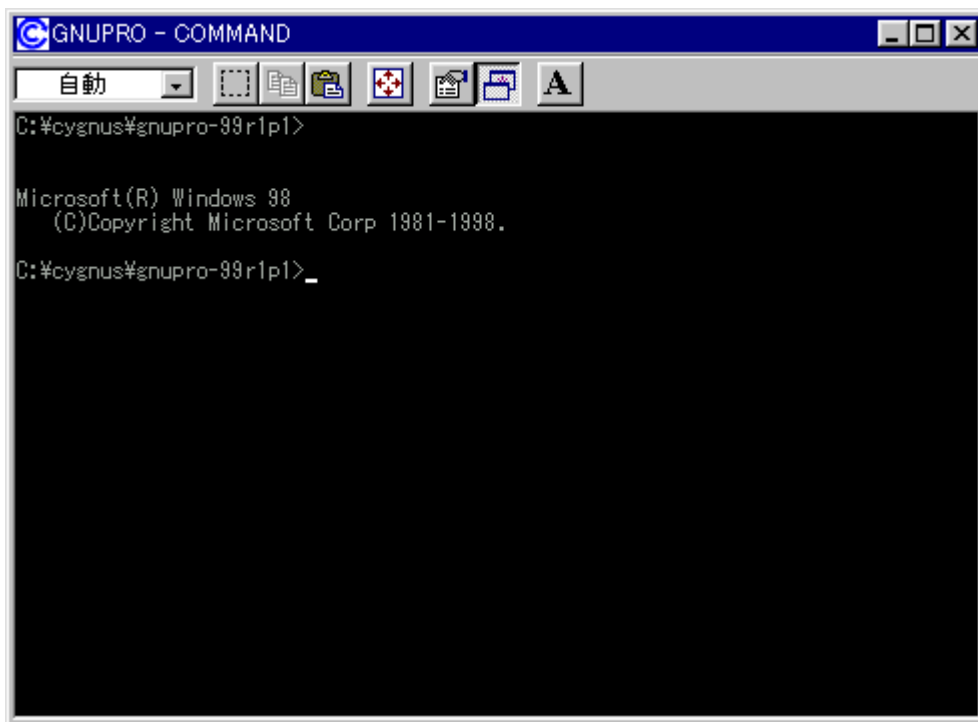
それでは、GCCを使用して実際にサンプルプログラムをコンパイルする手順を説明します。
VisualMoitor は C:\¥monikit に、GNUPro は C:\¥Cygnus にインストールしたものとして説明します。

- ①スタートメニューから Cygnus GNUPro Toolkit gnupro-99r1p1 を起動します。



- ②すると次のようなDOSプロンプト画面が表示されます。

このとき、「環境変数のための領域が足りません」というメッセージが表示された場合には、環境変数領域サイズを設定する必要があります。詳しくはマニュアル「Cコンパイラ編」をご覧ください。



- ③サンプルプログラムフォルダに移動します。次のようにコマンドを入力してください。
サンプルプログラムを他のフォルダにコピーしてあった場合には該当するフォルダに移動してください。

```
>CD C:\¥monikit¥7045f¥sample¥gcc
```

- ④次に gcc を起動してコンパイルをおこなってみます。
次のように入力します。

```
>sh-hms-gcc @sh2f_0a.p
```


③ それでは、バッチファイルを実行してコンパイルをおこなってみます。

次のように実行します。

>sh2f_0a.bat

```

GNUPRO - COMMAND
自動
Reading specs from /CYGNUS/GNUPRO~1/H-I686~1/BIN/..../lib/gcc-lib/sh-hms/2.9-gnupro-99r1p1/specs
gcc version 2.9-gnupro-99r1p1
/CYGNUS/GNUPRO~1/H-I686~1/BIN/..../lib/gcc-lib/sh-hms/2.9-gnupro-99r1p1/collect2.exe -o sh2f_0a.out -L/CYGNUS/GNUPRO~1/H-I686~1/BIN/..../lib/gcc-lib/sh-hms/2.9-gnupro-99r1p1/m2 -L/CYGNUS/GNUPRO~1/H-I686~1/BIN/..../lib/gcc-lib/sh-hms/2.9-gnupro-99r1p1 -L/CYGNUS/GNUPRO~1/H-I686~1/BIN/..../lib/gcc-lib -L/CYGNUS/GNUPRO~1/H-I686~1/BIN/..../lib/gcc-lib/sh-hms/2.9-gnupro-99r1p1/..../sh-hms/lib/m2 -L/CYGNUS/GNUPRO~1/H-I686~1/BIN/..../lib/gcc-lib/sh-hms/2.9-gnupro-99r1p1/..../sh-hms/lib -L/CYGNUS/GNUPRO~1/H-I686~1/BIN/..../lib/gcc-lib/sh-hms/2.9-gnupro-99r1p1/..../sh-hms/bin/ld: mode sh
crt0.o
main.o
gmachine.o
(c:%VMonikit%7045f%lib%sh2ul.a)userlog.o
(c:%VMonikit%7045f%lib%sh2ul.a)ul.o
/CYGNUS/GNUPRO~1/H-I686~1/BIN/..../lib/gcc-lib/sh-hms/2.9-gnupro-99r1p1/..../sh-hms/lib/m2/libc.a)memcpy.o

C:%Vmonikit%7045f%sample%gcc>
C:%Vmonikit%7045f%sample%gcc>

```

上記のようにパラメータファイルを使用した場合とほとんど変わらない表示がされ、同様に sh2f_0a.out と sh2f_0a.map が作成されます。

③ それでは、make でコンパイルを実行してみます。

>make

make では、同一フォルダ内にある makefile を参照してコンパイルが実行されます。結果は、先の2つの方法と同様です。

リスト3. 1-1~3. 1-3に各ファイルの簡単な説明を記載します。変更する場合の参考にしてください。各パラメータの詳細については、マニュアル「Cコンパイラ編」を参照してください。

リスト3. 1-1 sh2f_0a.p

```

-v
-g
-m2
-nostartfiles
-Wl, -t, -Map sh2f_0a.map
-o
sh2f_0a.out
}
 crt0.s
}
 main.c
}
 gmachine.c
-T sh2f_0a.x
c:¥VMonikit¥7045f¥lib¥sh2ul.a
    
```

- 出力ファイル名指定 (Output file name specification) - points to `-o sh2f_0a.out`
- コンパイルするファイル名を指定 (Specify file names to compile) - points to the source files `crt0.s`, `main.c`, and `gmachine.c`
- リンクスクリプトファイル指定 *1 (Specify link script file *1) - points to `-T sh2f_0a.x`
- VisualMonitor のインストールディレクトリを指定 (Specify VisualMonitor installation directory) - points to `c:¥VMonikit¥7045f¥lib¥sh2ul.a`

*1 リンカスクリプトの詳細は [Cコンパイラ編] をご覧ください。

リスト3. 1-2 makefile

```

INCPATH =
LIBPATH = C:/VMonikit/7045f/lib/
AS      = sh-hms-gcc
CC      = sh-hms-gcc
LD      = sh-hms-gcc
OBJCOPY = sh-hms-objcopy

TARGET = sh2f_0a

CFLAGS = -c -v -g -m2
LDFLAGS = -v -m2 -nostartfiles -Wl, -t, -Map ${TARGET}.map -Tsh2f_0a.x

SRCS = crt0.s main.c gmachine.c
OBSJ = crt0.o main.o gmachine.o
LIBS = ${LIBPATH}SH2UL.a

${TARGET}.out:${OBSJ} ${TARGET}.x makefile
        ${LD} ${LDFLAGS} -o ${TARGET}.out ${OBSJ} ${LIBS}

crt0.o:crt0.s
        ${CC} ${CFLAGS} crt0.s -o crt0.o

main.o:main.c
        ${CC} ${CFLAGS} main.c -o main.o

gmachine.o:gmachine.c
        ${CC} ${CFLAGS} gmachine.c -o gmachine.o
    
```

- VisualMonitor のインストールディレクトリを指定 (Specify VisualMonitor installation directory) - points to `LIBPATH = C:/VMonikit/7045f/lib/`
- コンパイルするソースファイルを指定 (Specify source files to compile) - points to `SRCS = crt0.s main.c gmachine.c`
- リンクするオブジェクトファイルを指定 (Specify object files to link) - points to `OBSJ = crt0.o main.o gmachine.o`
- オブジェクトファイルを生成するファイル名を指定 (Specify file names to generate object files) - points to the three object file rules at the bottom

リスト 3.1-3 sh2f_0a.bat

rem 環境に合わせて各実行ファイルやライブラリのパスの設定を変更してください

```

echo OFF
echo -v -m2 -nostartfiles                > sh2f_0a.lnk
echo -Wl,-t,-Map sh2f_0a.map             >> sh2f_0a.lnk
echo -o sh2f_0a.out                       >> sh2f_0a.lnk
echo crt0.o                               >> sh2f_0a.lnk
echo main.o                               >> sh2f_0a.lnk
echo gmachine.o                           >> sh2f_0a.lnk
echo -Tsh2f_0a.x                           >> sh2f_0a.lnk
echo c:\¥VMonikit¥7045f¥lib¥sh2ul.a       >> sh2f_0a.lnk
echo ON

```

sh-hms-gcc -c -v -t -g -m2 main.c -o main.o
sh-hms-gcc -c -v -t -g -m2 crt0.s -o crt0.o
sh-hms-gcc -c -v -g -m2 gmachine.c -o gmachine.o
sh-hms-gcc @sh2f_0a.lnk

出力ファイル名を指定

リンクファイル名を指定

リンクスクリプトファイル指定

VisualMonitor のインストールディレクトリを指定

オブジェクトファイルを生成するファイル名を指定

* マップファイル名 (sh2f_0a.map) 等は適宜修正してください。

3. 2 リモートデバッガ

3. 2. 1 デバッグ環境の準備

デバッグをおこなう為に環境を準備します。

事前に、CPUボードの基板端の4箇所を取り付け穴を利用してスペーサ等を取り付けておいてください。

- ① まず、パソコンとCPUボードを添付のRS232Cケーブルで接続してください。
この時、使用するパソコンのCOMポート番号（COM1～COM4）を確認します。
接続は、概要の図1. 3-1を参考にしてください。

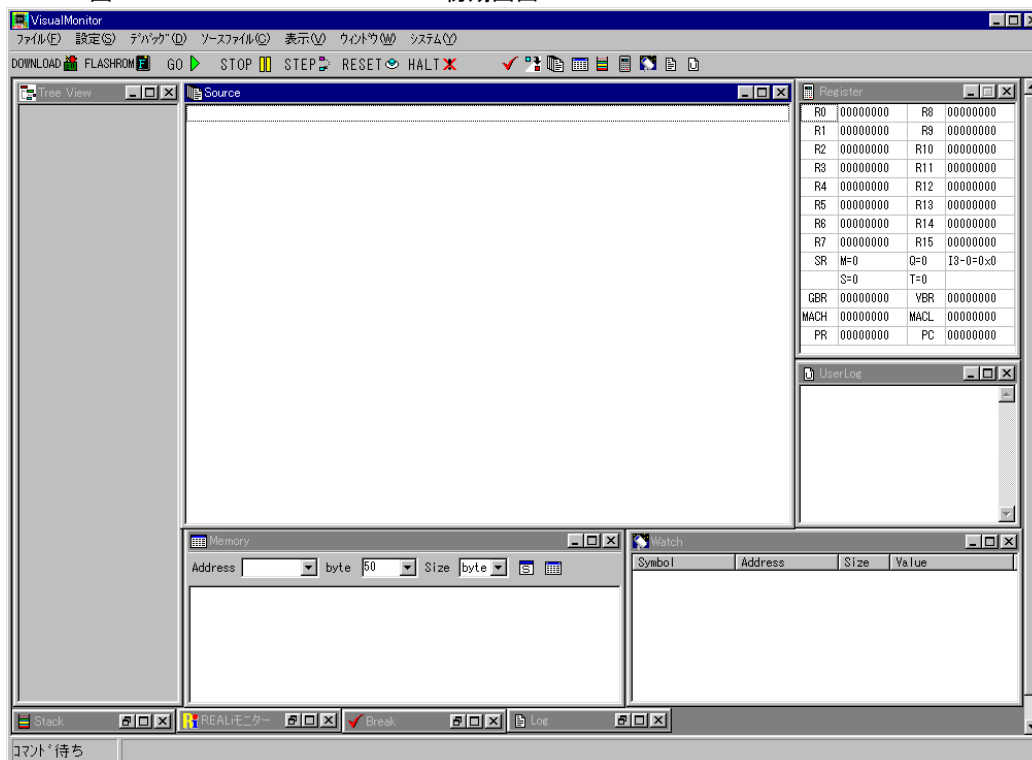
- ② CPUボードに電源を接続します。
電源は5V/500mA以上のもので電源SWが付いているものを用意してください。
添付の電源用ハーネスで接続します。

3. 2. 2 リモートデバッガの起動と使用方法

リモートデバッガでサンプルプログラムを動作させてみます。

- ① 最初にVisualMonitorを起動します。
Windowsの「スタートメニュー」の「プログラム」からSH2 7045F Starter Kit→VM2を選択し、実行します。
実行すると、VisualMonitorの初期画面が表示されます。

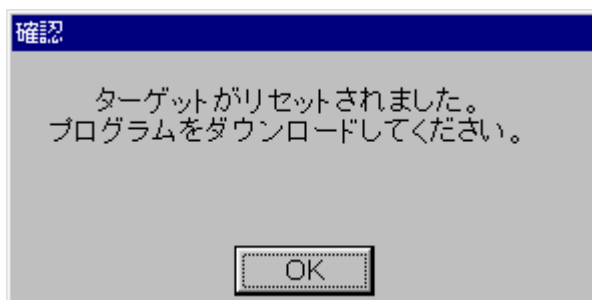
図 3. 2. - 1 VisualMonitor 初期画面



② 次にCPUボードの電源を入れます。

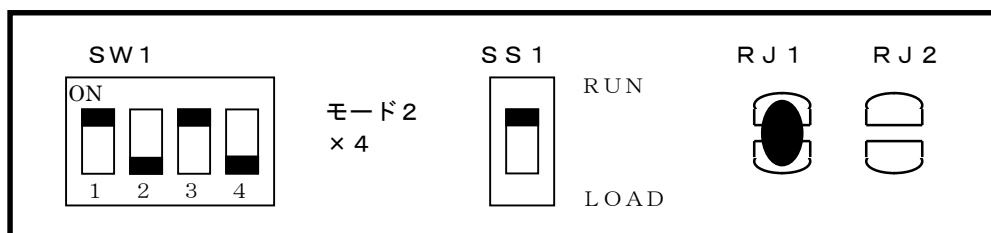
既にCPUボードの電源が入っている場合には、CPUボード上の白いプッシュSW（リセットボタン）を押します。
VisualMonitorの画面に以下のメッセージダイアログが表示されるので、OKをクリックしてください。

図3. 2-2 リセットメッセージ



上記メッセージが表示されない場合には、SWが以下の設定になっているか確認してください。
なっていない場合には、設定を修正して、リセットSWを押してください。

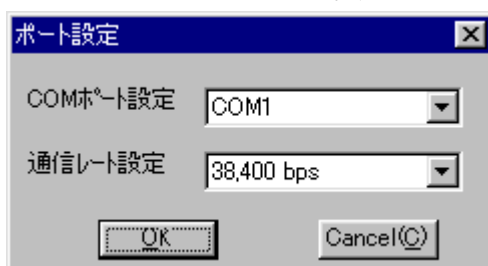
< SW設定 >



上記のDIP-SWの設定も間違いなく、メッセージが表示されない場合には、COMポートの設定を確認します。
デフォルトではCOM1になっているので、COM1以外を使用している場合には設定が必要です。

「システム」メニューの「ポート設定」を選択すると、以下のダイアログが表示されるので、使用するCOMポートを選択して変更してください。

図3. 2-3 COMポートの設定



以上の設定でもメッセージが出力されない場合には、何らかの原因でVisualMonitorのモニタ部が消去されている可能性があります。

その場合は、¥Vmonkit¥7045¥target¥の7045fkit.srをFlashWriterで再度CPUの内蔵フラッシュに書き込みます。
書き込み方法は後述の第4章の内蔵フラッシュROMの書き込み方法をご覧ください。

書き込んだ後、①の手順より再度行ってください。

③ 次にコンパイラを設定します。

「システム」メニューより「コンパイラ」を選択し、該当するコンパイラを選択します。

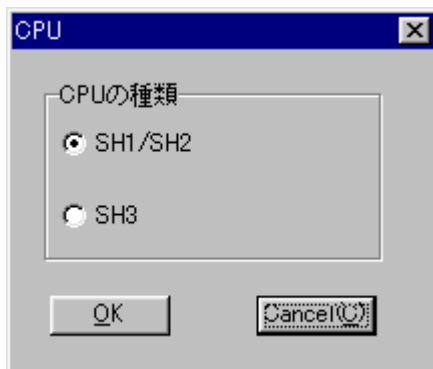
図3. 2-4 コンパイラを選択



④ 次にCPUを設定します。

「システム」メニューより「CPU」を選択し、該当するCPUを選択します。

図3. 2-5 CPUを選択



⑤ 次にサンプルプログラムをロードします。

「ファイル」メニューより「ダウンロード」を選択すると以下のダイアログが表示されるので、¥monkit¥7045f¥sample ¥gcc の sh2f_0a.out（日立 c の場合は¥monkit¥7045f¥sample¥hitachi¥sh2f_0a.abs）を選択して開くをクリックします。

図 3. 2-6 ダウンロードファイルの選択



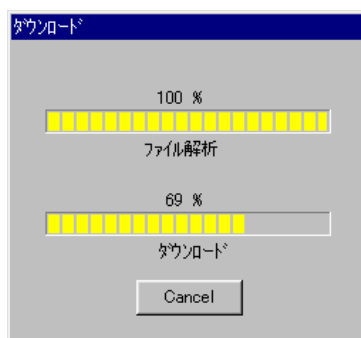
一番最初にダウンロードをする場合には、以下のダイアログが表示されるので、TRAP 優先を選択し、OKを押してください。

図 3. 2-7 ブレーク設定



ダウンロードが開始されると、以下のダイアログが表示されます。ダウンロードは数秒で終了します。

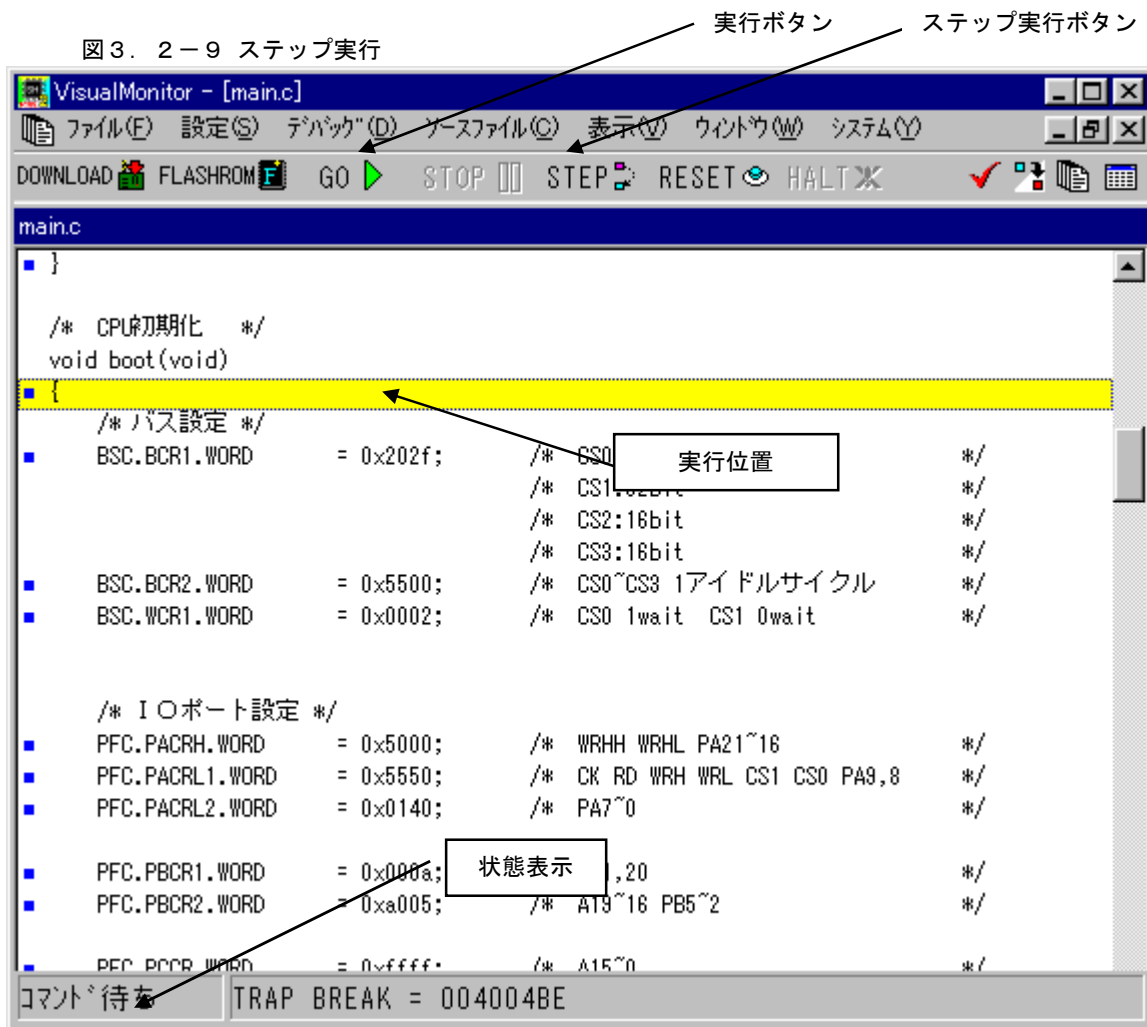
図 3. 2-8 ダウンロード表示



⑥ 次にサンプルプログラムを実行します。

ダウンロードすると、プログラムカウンタの値が、ユーザプログラムの先頭になると同時にソースプログラムが表示され、プログラムの先頭に黄色いバーが表示されます。

続けてステップ実行ボタンを押すと、黄色いバーが移動していきます。これは次に実行するプログラム位置を示しています。また、レジスタウィンドウを確認すると各レジスタ値が変化するのが判ります。



⑦ ステップ実行でプログラムが動作していることが確認できれば、次にプログラムを連続実行してみます。

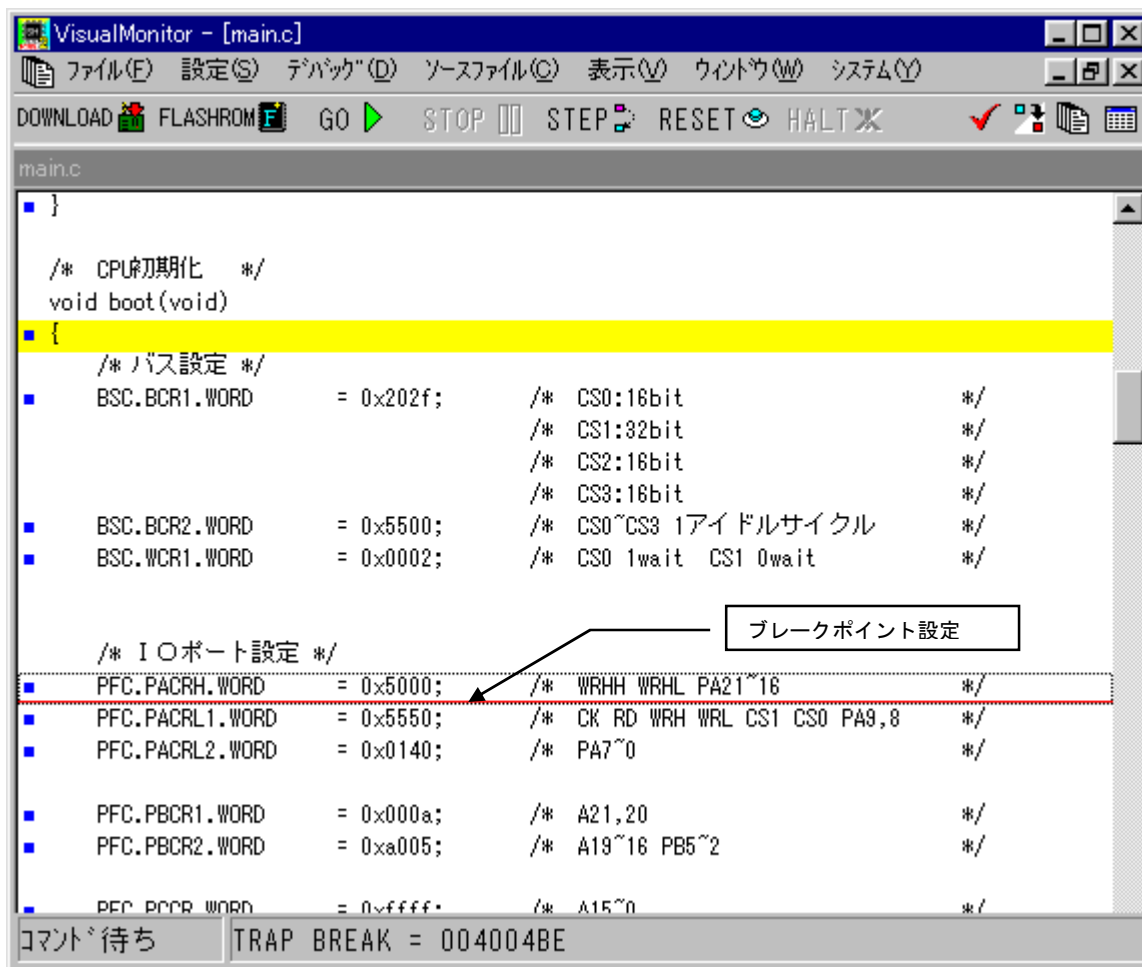
上図の実行ボタンをクリックすると、状態表示が実行中になり、停止ボタンがアクティブになります。

この状態でプログラムは実行されています。実行中は VisualMonitor の制御は関わらないため、速度の低下等はありません。停止ボタンを押すとプログラムは停止状態となり、停止位置が表示されます。

- ⑧ 次にブレークポイントを設定して任意の位置でプログラムを停止させてみます。
 まず、ソース上の停止させたい行でマウスを右クリックします。
 すると、ブレークメニューが表示されるので「ブレーク設定」表示部分にマウスを移動し、選択状態にします。
 その状態で左クリックすると、選択した行に赤いアンダーバーが表示されます。
 これでブレーク位置が設定されたので、実行ボタンを押すと設定した位置でプログラムが停止します。

図3. 2-10 ブレーク設定

ブレーク設定の解除も、右クリックで行います。詳細は「リモートデバッグ編」を参照してください。

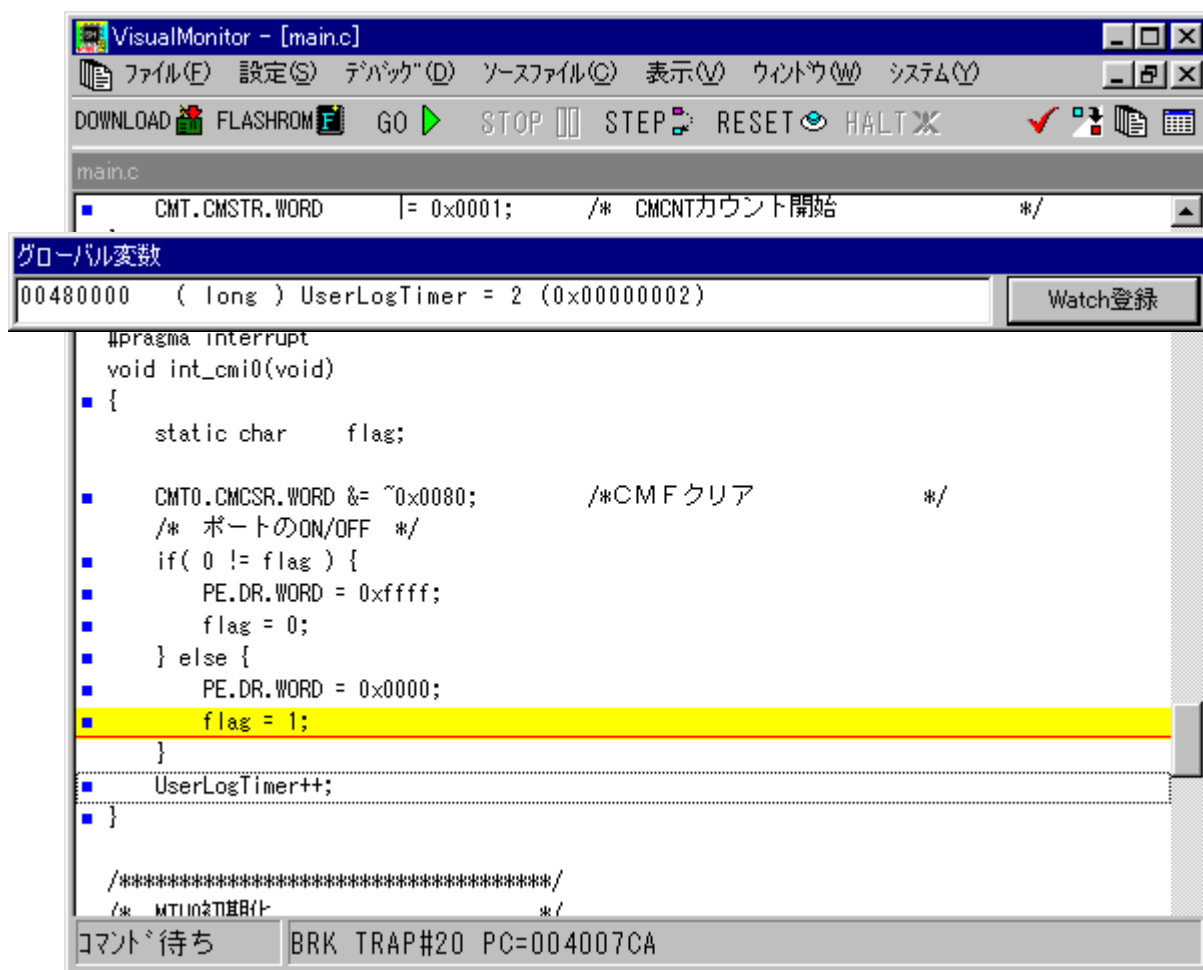


⑨ 次に変数の値を表示してみます。

ソース上の値を表示させたい変数名にマウスカーソルを位置付けて左ダブルクリックします。すると、変数の内容が10進数と16進数で表示されます。

さらにグローバル変数の場合は、Watch 登録ボタンが表示されるのでボタンを左クリックするとウォッチウィンドウに変数が自動登録されます。変数表示はグローバル変数とローカル変数に対応しています。配列変数はアドレス表示しかされませんので、メモリウィンドウと組み合わせてご使用ください。変数表示ウィンドウは、別の操作をすると自動的に消えます。

図3. 2-11 変数表示



3. 2. 3 その他の機能

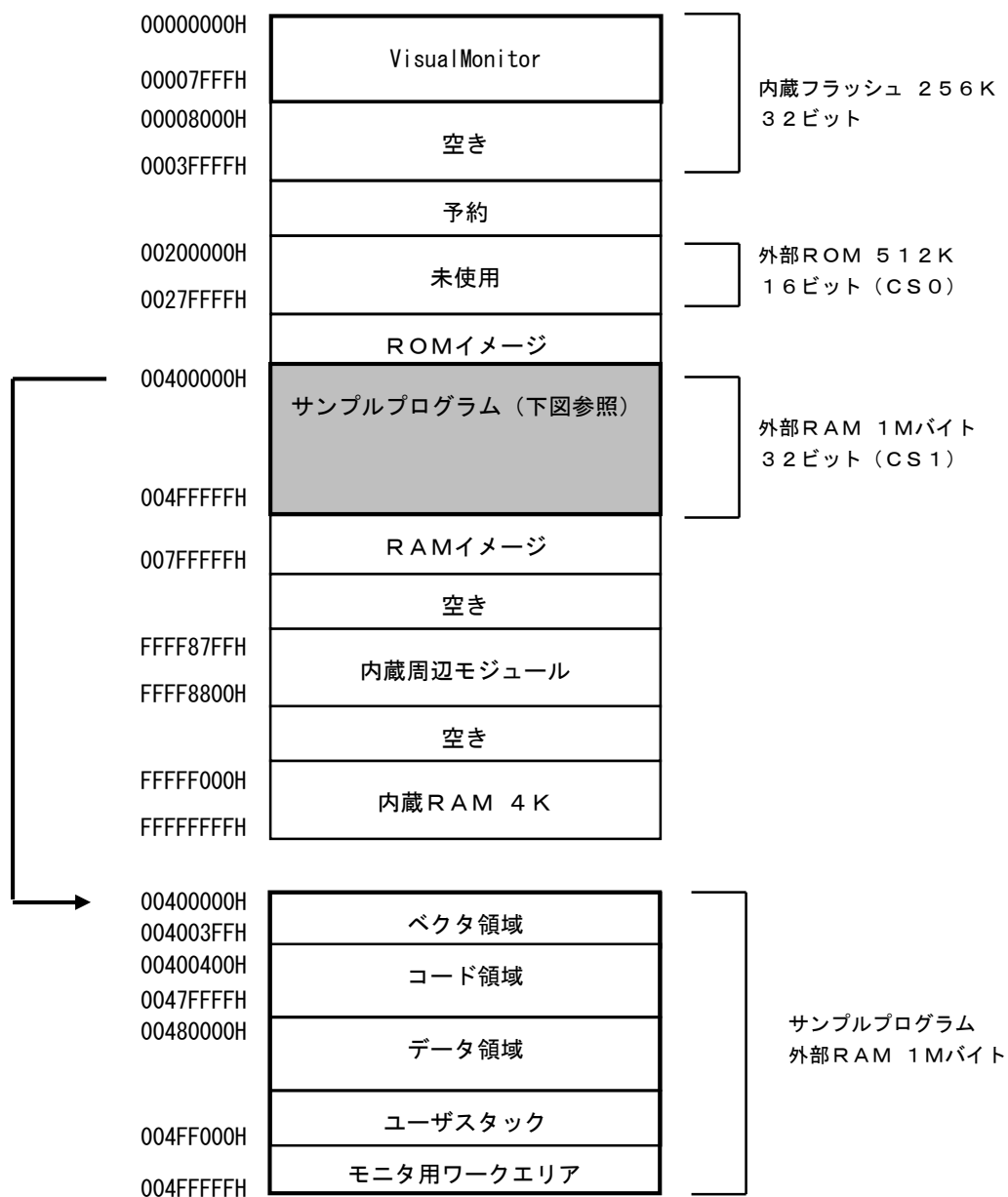
VisualMonitorには、その他の機能としてウォッチ、メモリダンプ、メモリ編集、ユーザログ出力等、様々な機能があります。それらの機能については別冊の「リモートデバッグ編」を参照してください。

3. 2. 4 サンプルプログラムのメモリマップ

サンプルプログラムのマッピングは下図のようになっています。

本キットのマッピングは固定となっており、内蔵フラッシュの先頭から、VisualMonitorが配置されRAMの先頭から、ベクタ、ユーザプログラムの順番となっています。

本製品では、ユーザプログラムを組む際には基本的に同一の配置にしてプログラムを作成する必要があります。



3. 2. 5 VisualMonitor 正規版との違い

本キットに付属する VisualMonitor / SH は AP-SH2F-0A ボードでのみ動作し、正規版と以下の点が異なります。

1. アドレスマップが固定

正規版では、ユーザーターゲットに合わせ、カスタマイズが可能となっています。

また正規版ではフラッシュROMへのダウンロードもサポートしています。

2. SH7045のみ対応

正規版では、SH1、SH2、SH3の各シリーズに対応しています。

3. 通信レート

正規版では、9600bps～115.2kbpsまで通信速度を選択できます。

また、限定版のターゲットモニタは正規版と互換性はありませんので、注意してください。

第4章. 内蔵フラッシュROM

4. 1 内蔵フラッシュROMの書き込み方法

AP-SH2F-0A では 256 Kbyte のフラッシュROM を内蔵した SH7045F を採用しています。

内蔵フラッシュROM には添付のソフトを利用してオンボードでユーザプログラムの書き込みが可能です。

なお、添付ソフトはWINDOWS 95/98/NT4.0/2000 対応で書き込みモードはブートモードのみです。

VisualMonitor のターゲットプログラムやユーザプログラムを書き込む場合に使用してください。

1) 書き込みソフト (FLASH WRITER) の準備

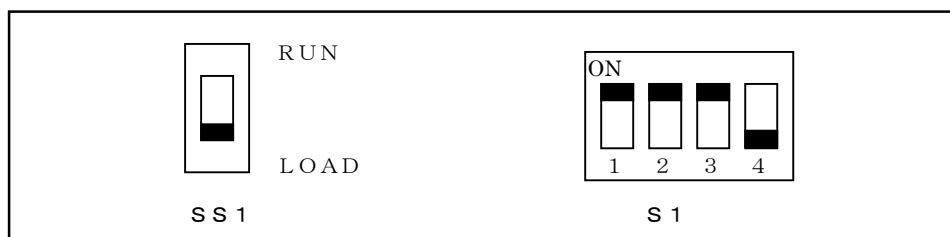
- ① CD-ROM 内の ¥FW¥FW7045.EXE を適当なフォルダにコピーします。

2) ボードの準備

- ① モードの設定

CPU のモードを以下の設定に合わせます。設定は電源を切った状態でおこなってください。

FWP : L
CPUモード : モード0
クロックモード : ×4



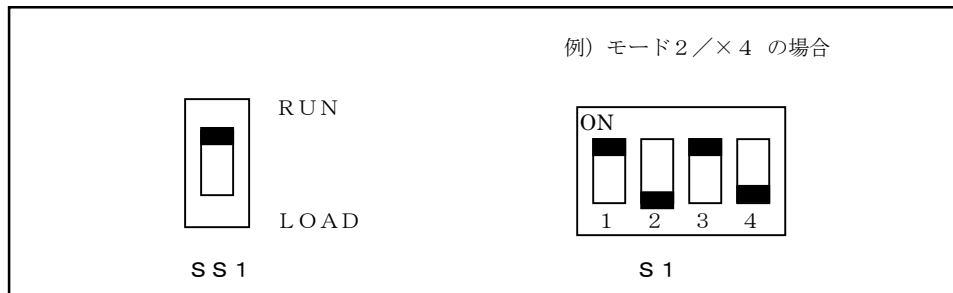
3) 書き込み手順

- ① パソコンとボードをクロスケーブルで接続します。
- ② FW7045.EXE をダブルクリックしてFLASH WRITER を起動します。
- ③ [設定]メニュー→[CPU 設定]でダイアログを開き、CPU 動作クロックに 28.636 を入力し、適用ボタンを押して設定します。
- ④ [設定]メニュー→[COM ポート選択]より使用するパソコンのCOMポートを選択します。
- ⑤ CPU ボードの電源を投入します。
- ⑥ [ファイル]メニュー→[開く]より、ダウンロードするファイルを選択します。(ダウンロード可能なファイルはモトローラSフォーマットのみです)
- ⑦ 自動的に書き込みを開始します

4) 動作の確認

① CPUのモードを以下の設定に合わせます。

FWP : H
CPUモード : モード2もしくはモード3
クロックモード : ×4



② 電源を投入すると、プログラムが動作します。

第5章. ROM化

5. 1 ユーザプログラムのROM化

デバッグ後のプログラムをROM化する方法について説明します。

また、以下に説明するROM化プログラムは、ROM化の手法一例であり、ユーザがブートプログラムや割り込みベクタ等を別プログラムで記述することによってもROM化は可能です。それらの方法によるサンプルは、CD-ROM内の¥sample フォルダに収録されていますので参考にしてください。

5. 1. 1 ROM化用ファイル

本製品には、デバッグ後のユーザプログラムをROM化するためのファイル群を添付しています。ファイルは、インストールフォルダ内の以下のフォルダに収められています。

<u>g c c</u> 用ROM化ファイル	インストールフォルダ¥ROM化¥g c c¥
<u>日立C</u> 用ROM化ファイル	インストールフォルダ¥ROM化¥h i t a c h i ¥

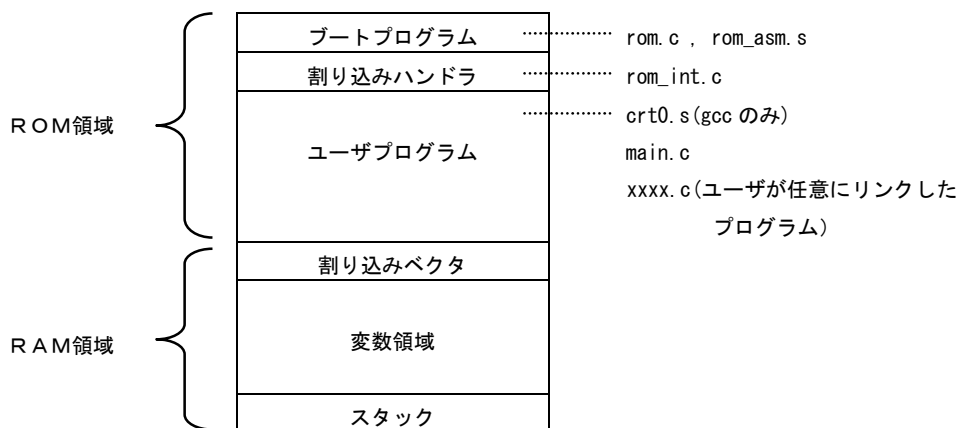
フォルダ内には以下のファイルが収められています。

• cpu. h	ダミーヘッダファイル
• rom. c	ブートプログラム
• rom_asm. s (src)	割り込み処理
• rom_int. c	割り込みハンドラ
• set. h	各種設定ファイル
• sh2f_0a. bat	生成用バッチファイル (GCC)
• sh2f_0a. x	生成用リンクスクリプトファイル (GCC)
• makefile	m a k e 定義バッチファイル (GCC)
• makeall. bat	生成用バッチファイル (日立SHC Ver6 以前)
• makeall6. bat	生成用バッチファイル (日立SHC Ver6 以降)

これらのファイル群を利用すれば、デバッグしたユーザプログラムを変更することなくROM化が可能です。なお、ROM化用ファイルは全てソースコードで提供しておりますので参考にしてください。

5. 1. 2 ROM化プログラムの構造

本製品に添付されているROM化プログラムは、基本的にデバッグしたユーザプログラムを変更することなくROM化することを前提としています。添付のサンプルプログラムを使用した場合を例に、構造を以下に示します。



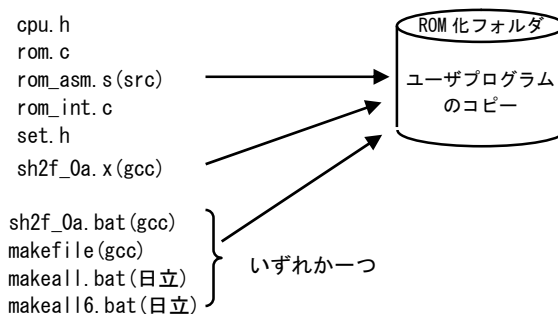
デバッグ用プログラムとの違い

- ① **ブートプログラム** モニタプログラムの代わりとなるブートプログラムを配置します。サイズは1Kバイト以下になります。(ブートプログラム+割り込みハンドラ)
- ② **ユーザプログラム** ROM領域に配置します。
- ③ **スタック** ユーザプログラム用のスタックのみとなります。
- ④ **ユーザログ** ユーザログは出力されません。

5. 1. 3 ROM化の手順

サンプルプログラムを例にROM化手順を説明します。

- ① ユーザプログラムのフォルダをコピーし、前述のROM化ファイルをコピーします。



- ② GCCの場合は、sh2f_0a.bat (バッチファイル) か makefile を使用します。日立Cの場合は、バージョンに合わせて makeall.bat か makeall6.bat を使用します。それぞれを必要に応じて編集します。編集方法は前述の「サンプルプログラムのコンパイル」を参照してください。
- ③ バッチやmakeを実行します。エラーが発生した場合にはエラーリストを出力して適宜修正してください。
- ④ 生成されたモトローラ s フォーマットファイル (*.sr もしくは*.mot) をFlashWriter で内蔵ROMに書きこむか、EEPROMに書きこんで動作を確認してください。

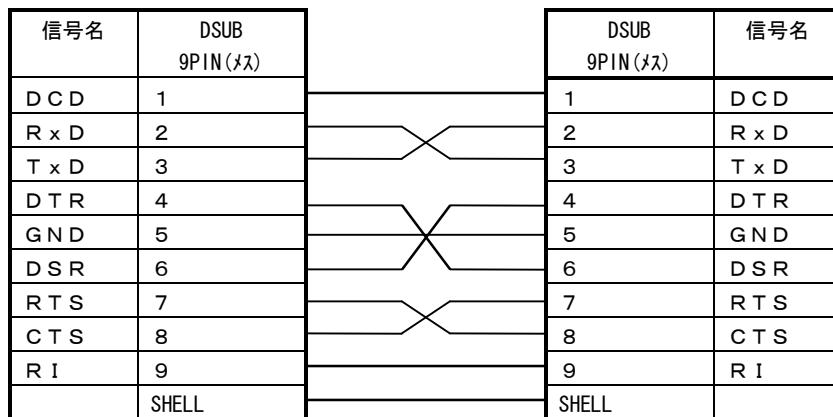
第6章. その他

6. 1 付属品

6. 1. 1 RS232Cケーブル

添付のRS232Cケーブルの結線を以下に示します。

ケーブル結線 (クロスケーブル)



第7章. 本製品のサポートについて

7. 1 質問の受け付け

CPUボード、VisualMonitor、FlashWriterの質問については弊社ホームページのQ&Aをご覧ください。

また、それ以外の質問についても随時受け付けておりますので、下記のFAXもしくはE-MAILで受け付けております。なお、正確な回答を行うために、電話による受け付けは基本的にお断りしておりますので、御了解ください。

CPUの使用方法及びユーザアプリケーションに関する質問等、技術指導的な質問については御遠慮ください。

本キットに付属するGCCコンパイラ（GNUPro）は、弊社ではサポート及び保証はおこなっておりません。これらのサポートを受ける場合にはレッドハット株式会社より直接GNUProを購入していただく必要があります。

レッドハット社URL : <http://www.jp.redhat.com>

なお、弊社作成の各サンプルプログラムについては動作の確認をおこなっておりますが、製品保証範囲ではありませんので御了承ください。

7. 2 バージョンアップサービス

VisualMonitor正規版へのアップグレードは随時受け付けております。

VisualMonitorパッケージ限定版、FlashWriterについては、弊社で致命的な不具合があると判断した場合のみ、バージョンアップデータの提供をいたします。

バージョンアップデータは弊社ホームページにて提供いたします。

GCCについてはバージョンアップサービスをご希望のお客様はレッドハット株式会社よりご購入ください。

お問い合わせ先

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

FAX (053) 401-0035 技術部 担当者宛

E-MAIL query@apnet.co.jp

HOME-PAGE <http://www.apnet.co.jp>

第8章. Q & A

Q & Aは、弊社ホームページで随時更新しておりますので、トラブルシューティングにお役立てください。

Q 1. リアルタイムOSなどでユーザプログラムがスタックを切り替えているが、問題なく動作しますか？

A 1. Visual Monitorとユーザプログラムの使用するスタックは別エリアとなっておりますので、特に問題なく動作します。（弊社製REALiの場合）。ただし、OSの種類によってはスタックポインタを直接操作するものや割り込みベクタを操作するものがありますので、それらについては正しくデバッグできない場合があります。

Q 2. GCCでサンプルプログラムをコンパイルしたが、コンパイルエラーが大量に出ってしまった。

A 2. コンパイルで問題が発生するのは、ほとんどの場合コンパイラオプションの指定ミスによるものです。
特に-m1, -m2, -m3 指定誤り及び-ansi 指定などは注意してください。
また、コンパイラオプションは環境変数でも指定されるので、AUTOEXEC.BATなどで間違ったオプションを指定していないかなどもチェックしてください。

Q 3. GCCでサンプルプログラムをコンパイルしたが、リンクエラーが出てしまう。

A 3. ほとんどがPATH指定ミスによるものです。
ご使用の環境に合わせて正しく変更されているかご確認ください。

Q 4. Visual Monitorの通信レートは変更できないのでしょうか？

A 4. 限定版は38.4Kbps固定となっていて変更できません。正規版では9600bps~115.2Kbpsまで対応しています。

Q 5. デバッグ中にVisual Monitorへ制御が戻らなくなってしまう。

A 5. 原因としては、

- ・ユーザプログラムで割り込みマスクを15に設定している
- ・ユーザプログラムでVisual Monitorが使用しているシリアルポートの設定を変更している。
- ・ユーザプログラムでVisual Monitorが使用するRAMエリアを書き換えている。

その他にも様々な原因が考えられます。

Visual Monitorは、ターゲット上の実チップで動作するため、ユーザプログラムで不正な処理をしても検出が困難です。ステップ実行などを利用して原因を特定してください。

Q 6. プログラムで#PRAGMAを使用するとブレークポイントの設定がずれる。

A 6. #PRAGMAなどの擬似命令を使用した場合、GCCから正しいデバッグ情報が出力されない場合があります。回避方法としては、同一行にコメントを入れる等すれば、改善されるようです。

Q 7. ROMデバッガを使用したいのですが。？

A 7. CS0エリアに外付けで16ビットバス幅のROMソケットを搭載しておりますので内蔵ROM無効モード（モード0）で動作させればROMデバッガの使用が可能です。

Q 8. 内蔵FROMは何回ぐらい書き換え可能ですか？

A 8. メーカー標準値は100回となっています。ただし、当社の実績からするとその数倍は問題ないようです。

Q 9. 内蔵FROMと外部ROMを両方使用することはできますか？

A 9. 可能です。内蔵ROM有効モードのマッピングは内蔵ROM：00000000H-0003FFFFH 外部ROM：00200000-0027FFFFH となり内蔵ROM無効モードの場合は外部ROM：00000000-0007FFFFHとなります。

Q 1 0. 外部EPROMに書き込むと動かないのですが？

A 1 0. SH7040シリーズはデータバスの並びがビッグエンディアンになっています。すなわち、D15-D8が0番地(EVEN)となります。ところが16ビットバス幅のEPROMをライターで書き込む場合、機種によってはリトルエンディアンで書き込みます。この場合、上位バイトと下位バイトをスワップさせる必要があります。通常はライターにスワップ機能がありますのでこれを利用して書き込んでください。

その他の原因としては以下のことも考えられます。

- ① 内蔵ROM有効モードの場合はデータバス、アドレスバスとも入力ポートとなっている為、バスとして使用する為にはPFCの設定が必要です。外部メモリアクセスの前に初期化処理を行ってください。

Q 1 1. 内蔵フラッシュのみで動作させたいのですが可能ですか？

A 1 1. 可能です。外部ROMはデバッグ用途や拡張用に設けてあるので、使用しない場合は搭載する必要はありません。

Q 1 2. VisualMonitorでREALiのモニタウィンドウがアクティブになりません。

A 1 2. REALiのモニタウィンドウを有効にするためには、REALiのデバッグ情報付きライブラリをリンクする必要があります。REALiにはソースコードが添付されておりますので、デバッグオプション付きで再生成してライブラリを作成してください。わからない場合にはE-mailにて送付しますので、ご連絡ください。

Q 1 3. FlashWriterで内蔵フラッシュの一部を消さないで書き込みが可能ですか？

A 1 3. できません。FLASH WRITER PROはブート書き込みのみをサポートしているため、チップの仕様によりすべての内蔵フラッシュのエリアが消去されてしまいます。

Q 1 4. モトローラSフォーマットはS1、S2、S3がありますが、FlashWriterで書き込む場合はどれを使用したらよいのでしょうか？

A 1 4. 全てのフォーマットに対応していますので問題ありません。

Q 1 5. 添付のgccは、他のSHマイコンでも使用可能でしょうか？

A 1 5. 使用可能です。使用方法は変わりませんのでSH1/SH2/SH3/SH3Eのプログラムを作成することができます。

ご注意

- (1) 本書に記載されている、MPUなどの製品名は各社の登録商標です。
- (2) 本書の内容を無断で転載することは一切禁止します。
- (3) 本書の内容および本資料に記載された製品に関しては、将来予告なしに変更されることがあります。
- (4) 本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。
- (5) 運用した結果については(4)項にかかわらず責任を負いませんのでご了承下さい。

本製品 (**SH2 7045F Starter Kit**) は日本国内仕様であり、外国の規格等には準拠していません。本製品は日本国外で使用された場合、当社では一切責任を負いかねます。また当社は本製品に関し、海外での保守サービス及び技術サポート等はおこなっていません。

Alpha Project Co.,LTD