

SHマイコン開発支援パッケージ

SH3 Starter KIT

ユーザズマニュアル

4 版 2001 / 01 / 22

ALPHA PROJECT Co., LTD

SH3 Starter KIT User's Manual

この度は、当社製品「SH3 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SH3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 - SH3 - 0A CPUボード	1枚
* PC - RS - 03	1枚
* RS232Cケーブル	1本
* SH3 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
〒433-8120 静岡県浜松市上島4-4-24

FAX (053)464-3737 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	2
1.4 CD-ROMの内容	3
1.5 各ソフトウェアの機能	3
1.5.1 VisualMonitor	3
1.5.2 ダウンローダ	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ugg(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uggの動作確認	5
2.3 ダウンローダのインストール	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	6
3.2 リモートデバugg	10
3.2.1 デバugg環境の準備	10
3.2.2 リモートデバuggの起動と使用方法	10
3.2.3 その他の機能	16
3.2.4 サンプルプログラムのメモリマップ	17
3.2.5 VisualMonitor正規版との違い	18

第4章. フラッシュROM	19
----------------------	-----------

4.1 フラッシュROM書き込み方法	19
--------------------	----

第5章. ROM化	22
------------------	-----------

5.1 ユーザプログラムのROM化	22
5.1.1 ROM化用ファイル	22
5.1.2 ROM化プログラムの構造	23
5.1.3 ROM化の手順	23

第6章. その他	24
-----------------	-----------

6.1 RS232C変換アダプタ	24
6.1.1 概要	24
6.1.2 仕様	24
6.1.3 設定	24
6.1.4 接続方法	26
6.1.5 SHUTDOWN機能	26
6.1.6 外形寸法	26
6.1.7 回路図	27
6.2 RS232Cケーブル	28
6.1.1 結線	28

第7章. サポート	29
------------------	-----------

7.1 質問の受け付け	29
7.2 バージョンアップサービス	29

第8章. Q&A	30
---------------------	-----------

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。

- ・SH7709 CPUボード 「AP-SH3-0A」
- ・RS232Cアダプタ 「PC-RS-03」
- ・リモートデバッグ 「VisualMonitor Ver.2 パッケージ限定版」
- ・フラッシュダウンローダ
- ・GNU Cコンパイラ

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。

- 「SH3 Stater KIT ユーザーズマニュアル」 コンパイルからデバッグの使用法までの簡易マニュアル。
- 「SH3 Stater KIT CPUボード編」 CPUボードのハードウェアマニュアル
- 「SH2/SH3 Stater KIT Cコンパイラ編」 gccの解説マニュアル
- 「SH2/SH3 Stater KIT リモートデバッグ編」 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	16Mバイト以上を推奨
	ハードディスク	15M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-SH3-0A (SH7709ボード)
	CPUクロック	80MHz
	モニタコード	オンボードフラッシュROMに書込済み
	ダウンローダ	EPROMで搭載済み
	モニタ使用RAMサイズ	約4Kバイト
	シリアルポート	SCIFを使用、ボード上のシリアルコネクタに接続
	RS232C	PC-RS-03と添付のクロスケーブルを使用
	電源	5V / 400mA以上の安定化電源

- ・VisualMonitor 及びダウンローダはWIN95 / 98 / NT 4.0 / 2000上で動作します。
- ・GCCコンパイラは、WINDOWS上の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。

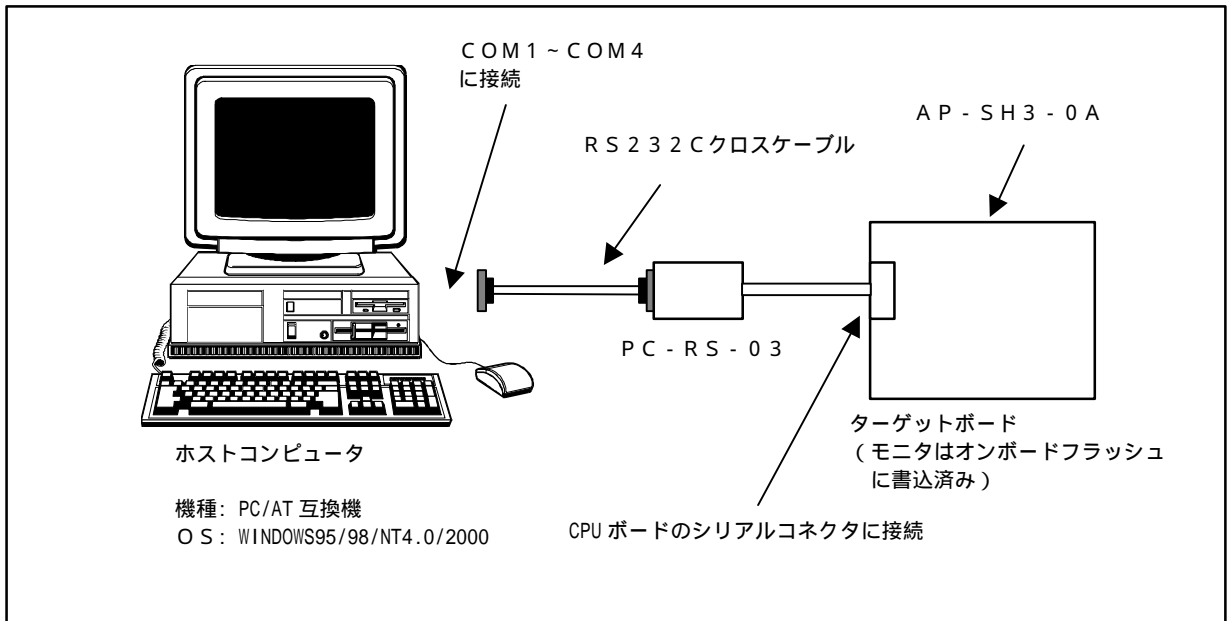


図1.3 本キットでのデバッグ環境の構成例

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) (モニタプログラム) (サンプルプログラム)
	DOWNLOAD		専用ダウンローダ
	GCC	BIN LIB INCLUDE SOURCE GPL	GNU C コンパイラ(gcc), ツール ライブラリ インクルードファイル GNU ソース GPL 規約
	SAMPLE		ROM サンプルプログラム
	SH_MAN		SH7709 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ーです。

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-SH3-0Aでのみ動作します。

メモリマッピングは、オンボードフラッシュに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 ダウンローダ

AP-SH3-0A専用のオンボードフラッシュ書き込みソフトです。

PCのダウンロードソフトとターゲット上のダウンローダの2つから構成されます。

1.5.4 GCC

フリーのCコンパイラです。

GCCは特に機能制限はなく、ROM化まで可能です。

ただし、本製品に添付されるGNUツール及びライブラリは全てGPL及びGPLに従います。

したがって、お客様が作成したプログラムもこれらの規約に従う必要があります。

※GCC※GPL 以下に GNU 一般公有使用許諾書と GNU ライブラリ一般公有使用許諾書が収録されていますので、ご使用前によくお読みください。なお、ライブラリはSH3用のみが添付されています。

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。

パーソナルコンピュータ

前項の使用環境を御覧下さい。

ターゲット

CPUボード「AP-SH3-0A」、RS232Cアダプタ「PC-RS-03」

インストールディスク

本キットに添付されているCD-ROM

RS232Cケーブル

添付のクロスケーブルをご使用ください。

電源

5V/300mA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

CD-ROM内の¥Vm¥Install¥SETUP.EXEを実行して下さい。

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。

インストール方法を3種類用意してありますので希望の方法を選択して下さい。

インストール時にセットアップ方法のウィンドウが開いて選択します。

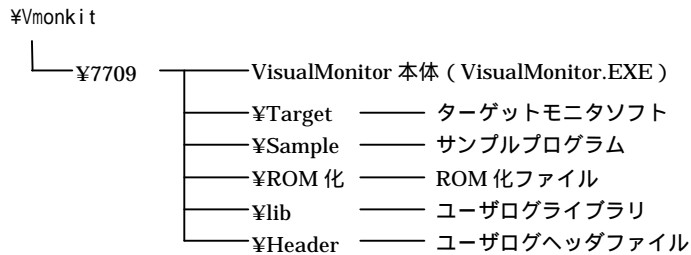
2.2.2 ターゲットモニタソフトのインストール

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¥7709kit.sr(ビッグエンディアン用)もしくは¥Target¥7709lkit.sr(リトルエンディアン用)です。

ターゲットモニタソフトは、ビッグエンディアン用がご購入時に既にオンボードフラッシュROMに書き込まれています。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や、リトルエンディアンで使用する場合には、付属の専用ダウンロードカードを使用してオンボード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¥7709 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

PC とターゲットを添付の PC - RS - 03 と RS 232C クロスケーブルで接続します。

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。

[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。

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 ダウンローダのインストール

特にインストーラは用意されておりませんので、¥download¥pc 以下の sh3dl.EXE を適当なフォルダにコピーして下さい。

2.4 コンパイラ (gcc) 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、¥GCC 以下を適当なフォルダにコピーしてください。

コンパイラが必要とするファイルは ¥GCC¥BIN 以下のファイルと ¥GCC¥LIB 以下のファイルと

¥GCC¥INCLUDE 以下のファイルのみです。

それ以外のファイルは、特に必要はありません。

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グの使い方までを簡単に説明します。

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

VisualMnitor を¥Vmonkit¥7709 ディレクトリにインストールしたものと説明します。

サンプルプログラムは g c c 用は¥Vmonkit¥7709¥Sample¥gcc、日立 C 用は¥Vmonkit¥7709¥Sample¥hitachi にインストールされます。(ビッグエンディアン用は¥big, リトルエンディアン用は¥little 以下になります)

サンプルプログラムは以下のファイルから構成されています。

<ソースファイル>

sh7709a.h	SH 7 7 0 9 シリーズレジスタ定義
gmachine3.h	SH3 MPU 依存命令ヘッダファイル
gmachine3.c	SH3 MPU 依存命令
UserLog.h	USERLOG 関数ライブラリヘッダ
Sh3UL.a(lib)	USERLOG 組み込み関数ライブラリ (¥lib ディレクトリに収録)
boot.c	スタートアッププログラム、メインプログラム

<その他>

makeall.bat	コンパイル、リンクのバッチファイル
gcc_sh3.out(abs)	実行ファイル (デバック情報有り)

最初はサンプルプログラムをベースにプログラムを変更していくのが一番理解が早いでしょう。

なお、各内蔵 I / O のサンプルプログラムは、以下の日立半導体事業部のサイトからダウンロードできますので、参考にしてください。(日立 C 版のみです)

<http://www.hitachi.co.jp/Sicd/Japanese/Seminar/down.htm>

3.1.2 サンプルプログラムのコンパイル

サンプルプログラムのコンパイル、リンクをおこなうには、収録されている makeall.bat を実行します。

まず、実行する前に、インストールした環境に合わせて、makeall.bat のパスの記述を変更します。

makeall.bat をエディタで開いて、以下の箇所を変更してください。

環境変数の詳細については、「コンパイラ編」を参照してください。

makeall.bat の変更(gcc)

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

```
set GCC_SH_PATH=c:\gcc
set path=%GCC_SH_PATH%\bin
```

← gcc をインストールしたディレクトリを指定してください。

```
set C_INCLUDE_PATH=%GCC_SH_PATH%\include
set LIBRARY_PATH=%GCC_SH_PATH%\lib
set COMPILER_PATH=%GCC_SH_PATH%\bin
set GCC_EXEC_PREFIX=%GCC_SH_PATH%\bin
set TMPDIR=%GCC_SH_PATH%
set GCCDIR=%GCC_SH_PATH%
```

echo OFF

```
echo -t -v -warn-common > gcc_sh3.lnk
echo -Map gcc_sh3.map >> gcc_sh3.lnk
echo -o gcc_sh3.out >> gcc_sh3.lnk
echo boot.o >> gcc_sh3.lnk
echo gmachine3.o >> gcc_sh3.lnk
echo C:\VMonKit\7709\lib\SH3UL.a >> gcc_sh3.lnk
```

← VisualMonitor をインストールしたディレクトリを指定してください。

```
echo SECTIONS { > gcc_sh3.cmd
echo .text 0x10001000 : >> gcc_sh3.cmd
echo { >> gcc_sh3.cmd
echo __stext = . ; >> gcc_sh3.cmd
echo *(.text) >> gcc_sh3.cmd
echo __etext = . ; >> gcc_sh3.cmd
echo } >> gcc_sh3.cmd
echo .data : >> gcc_sh3.cmd
echo { >> gcc_sh3.cmd
echo __sdata = . ; >> gcc_sh3.cmd
echo *(.data) >> gcc_sh3.cmd
echo __edata = . ; >> gcc_sh3.cmd
echo } >> gcc_sh3.cmd
echo .bss : >> gcc_sh3.cmd
echo { >> gcc_sh3.cmd
echo __sbss = . ; >> gcc_sh3.cmd
echo *(.bss) >> gcc_sh3.cmd
echo *(COMMON) >> gcc_sh3.cmd
echo __ebss = . ; >> gcc_sh3.cmd
echo __heap = . ; >> gcc_sh3.cmd
echo } >> gcc_sh3.cmd
echo } >> gcc_sh3.cmd
echo ON
```

```
gcc -c -v -Wall -g -m3 -O0 boot.c
gcc -c -v -Wall -g -m3 -O0 gmachine3.c
```

```
ld @gcc_sh3.lnk -Tgcc_sh3.cmd
```

makeall.bat の変更(日立c)

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 debug                > shc_sh3.sub
echo input boot.obj       >> shc_sh3.sub
echo input gmachine3.obj  >> shc_sh3.sub
echo input defsect.obj    >> shc_sh3.sub
echo library C:\VMonKit\lib\SH3UL.lib >> shc_sh3.sub
echo entry _boot         >> shc_sh3.sub
echo output shc_sh3.abs  >> shc_sh3.sub
echo print shc_sh3.map   >> shc_sh3.sub
echo ROM (D,R)           >> shc_sh3.sub
echo START P,C,D,R,B(10001000) >> shc_sh3.sub
echo exit                >> shc_sh3.sub
echo ON

asmsh defsect.src -DEBUG -NOLIST -CR -SE

shc /op=0 /cpu=SH3 /debug /show=ob,noso boot.c
shc /op=0 /cpu=SH3 /debug /show=ob,noso gmachine3.c

lnk /subcommand=shc_sh3.sub

```

VisualMonitor をインストールしたディレクトリを指定してください。

日立Cの場合、バージョンによって環境変数が違いますので、各環境変数を適宜設定してください。
 なお、HI Mを使用する場合には、上記の設定を参考にプロジェクトを設定してください。

次に makeall.bat を実行します。

WINDOWS の DOS プロンプトで %Vmonkit%¥7709¥sample¥gcc¥big¥(¥little) (日立 C の場合は ¥hitachi) にカレントディレクトリを移動し、makeall.bat を実行してください。

実行すると次のように経過が表示されます。(環境によっては若干異なります)

コンパイルの経過表示例 (gcc)

```
D:¥VMoniKit¥7709¥sample¥gcc¥big>set GCC_SH_PATH=d:¥gcc
D:¥VMoniKit¥7709¥sample¥gcc¥big>set path=d:¥gcc¥bin
D:¥VMoniKit¥7709¥sample¥gcc¥big>
D:¥VMoniKit¥7709¥sample¥gcc¥big>set C_INCLUDE_PATH=d:¥gcc¥include
D:¥VMoniKit¥7709¥sample¥gcc¥big>set LIBRARY_PATH=d:¥gcc¥lib
D:¥VMoniKit¥7709¥sample¥gcc¥big>set COMPILER_PATH=d:¥gcc¥bin
D:¥VMoniKit¥7709¥sample¥gcc¥big>set GCC_EXEC_PREFIX=d:¥gcc¥bin
D:¥VMoniKit¥7709¥sample¥gcc¥big>set TMPDIR=c:¥tmp
D:¥VMoniKit¥7709¥sample¥gcc¥big>set GCCDIR=c:¥gcc
D:¥VMoniKit¥7709¥sample¥gcc¥big>echo OFF
D:¥VMoniKit¥7709¥sample¥gcc¥big>
D:¥VMoniKit¥7709¥sample¥gcc¥big>gcc -c -v -Wall -g -m3 -O0 boot.c
Using builtin specs.
gcc version egcs-2.91.60 19981201 (egcs-1.1.1 release)
d:¥gcc¥bin¥cpp.exe -lang-c -v -iprefix d:¥gcc¥bin¥sh3¥2_91_60¥ -isystem d:¥gcc¥bin¥include -undef
-D_GNUC__=2 -D_GNUC_MINOR__=91 -D_sh__ -D_sh__ -Acpu(sh) -Amachine(sh) -g -Wall -D_sh3__ boot.c
C:¥WINDOWS¥TEMP¥ccSzoXQe.i
GNU CPP version egcs-2.91.60 19981201 (egcs-1.1.1 release) (Hitachi SH)
#include "..." search starts here:
#include <...> search starts here:
d:¥gcc¥include
End of search list.

省略

D:¥VMoniKit¥7709¥sample¥gcc¥big>gcc -c -v -Wall -g -m3 -O0 gmachine3.c
Using builtin specs.
gcc version egcs-2.91.60 19981201 (egcs-1.1.1 release)
d:¥gcc¥bin¥cpp.exe -lang-c -v -iprefix d:¥gcc¥bin¥sh3¥2_91_60¥ -isystem d:¥gcc¥bin¥include -undef
-D_GNUC__=2 -D_GNUC_MINOR__=91 -D_sh__ -D_sh__ -Acpu(sh) -Amachine(sh) -g -Wall -D_sh3__ gmachine3.c
C:¥WINDOWS¥TEMP¥ccOZ6UVK.i
GNU CPP version egcs-2.91.60 19981201 (egcs-1.1.1 release) (Hitachi SH)
#include "..." search starts here:
#include <...> search starts here:
d:¥gcc¥include
End of search list.

GNU ld version 2.9.1 (with BFD 2.9.1)
d:/gcc/bin/ld.exe: mode sh
boot.o
gmachine3.o
(d:¥VMoniKit¥7709¥lib¥SH3UL.a)userlog.o
(d:¥VMoniKit¥7709¥lib¥SH3UL.a)ul.o
```

以上で、実行形式ファイル gcc_sh3.out (日立 c の場合は shc_sh3.abs) が生成されます。

途中でエラーが発生した場合にはパスの指定を再度見直してください。

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。

まず、パソコンとCPUボードを添付のPC-RS-03と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を参考にしてください。

CPUボードに電源を接続します。

電源は5V/300mA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

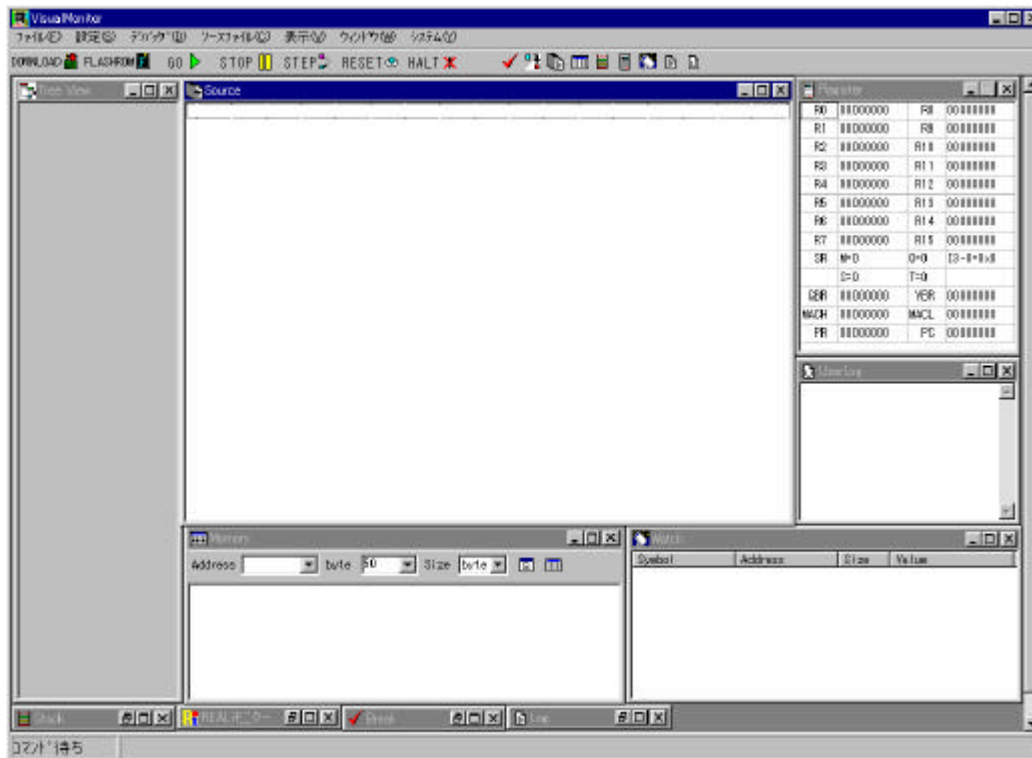
リモートデバッガでサンプルプログラムを動作させてみます。

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SH3 7709 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3.2.2.1 VisualMonitor 初期画面

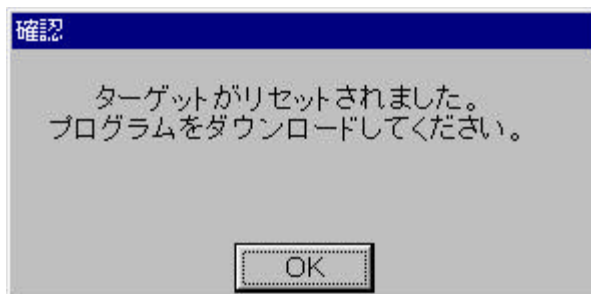


次にCPUボードの電源を入れます。

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。

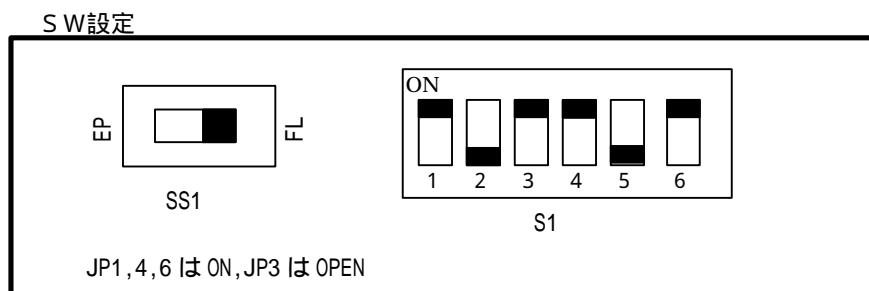
図3.2.2.2 リセットメッセージ



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3.2.2.3 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、ターゲットモニタを再度書きこんでください。

書き込み方法は後述の第4章のフラッシュROMの書き込み方法をご覧ください。

書き込んだ後、 の手順より再度行ってください。

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 3.2.2.4 コンパイラの選択



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。

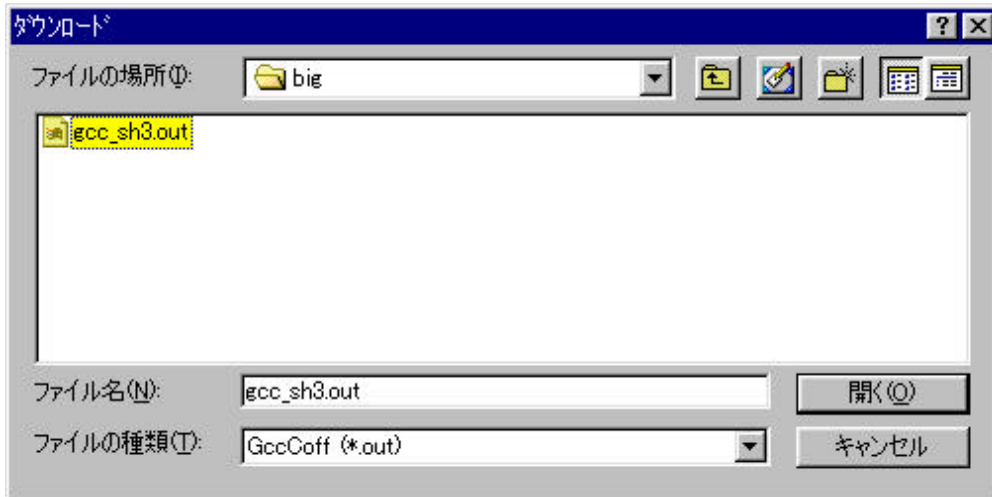
図 3.2.2.5 CPUの選択



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¥7709¥sample ¥gcc¥big¥(¥little)の gcc_sh2.out (日立 c の場合は¥monkit¥7709¥sample¥hitachi) を選択して開くをクリックします。

図 3.2.2.6 ダウンロードファイルの選択



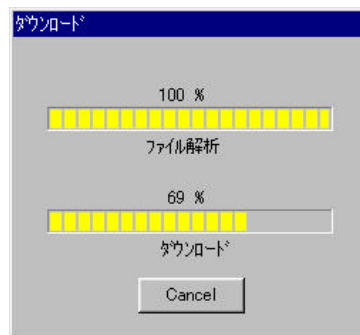
一番最初にダウンロードをする場合には、以下のダイアログが表示されるので、TRAP優先を選択し、OKを押してください。

図 3.2.2.7 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。

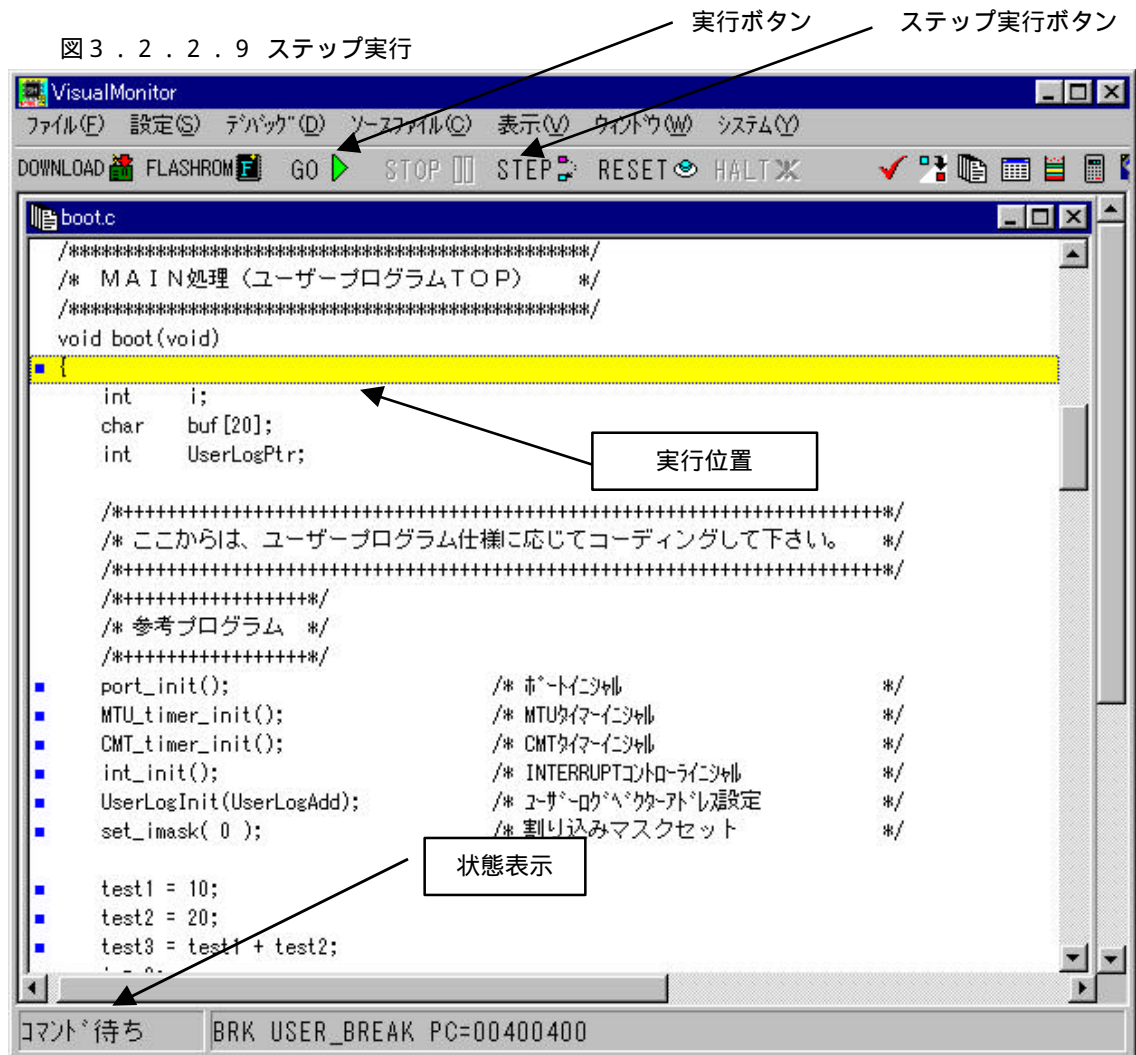
図 3.2.2.8 ダウンロード表示



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。



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。

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。

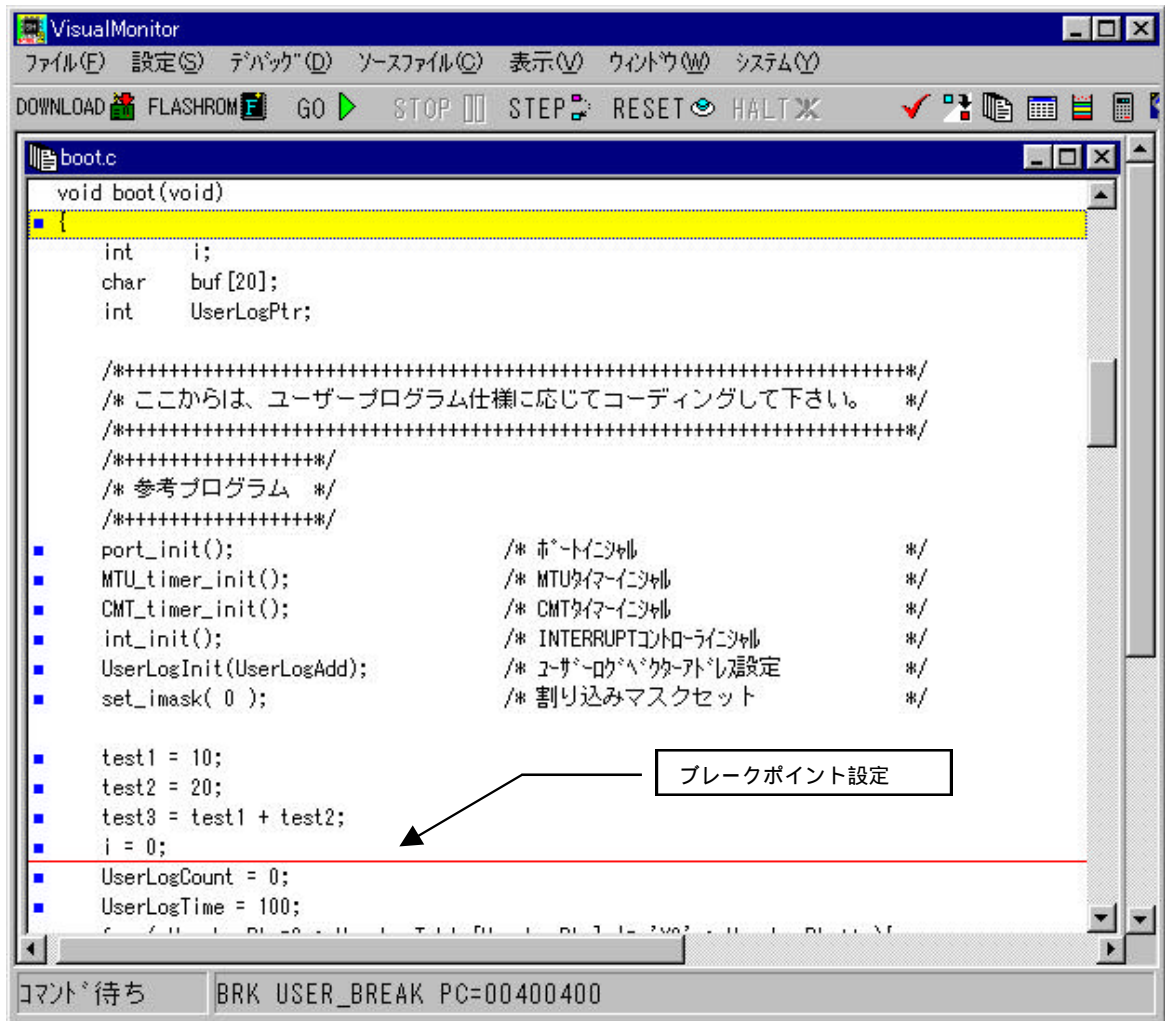
まず、ソース上の停止させたい行でマウスを右クリックします。

すると、「ブレーク設定」という文字が表示されるので、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 3.2.2.10 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「リモートデバッガ編」を参照してください。

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。

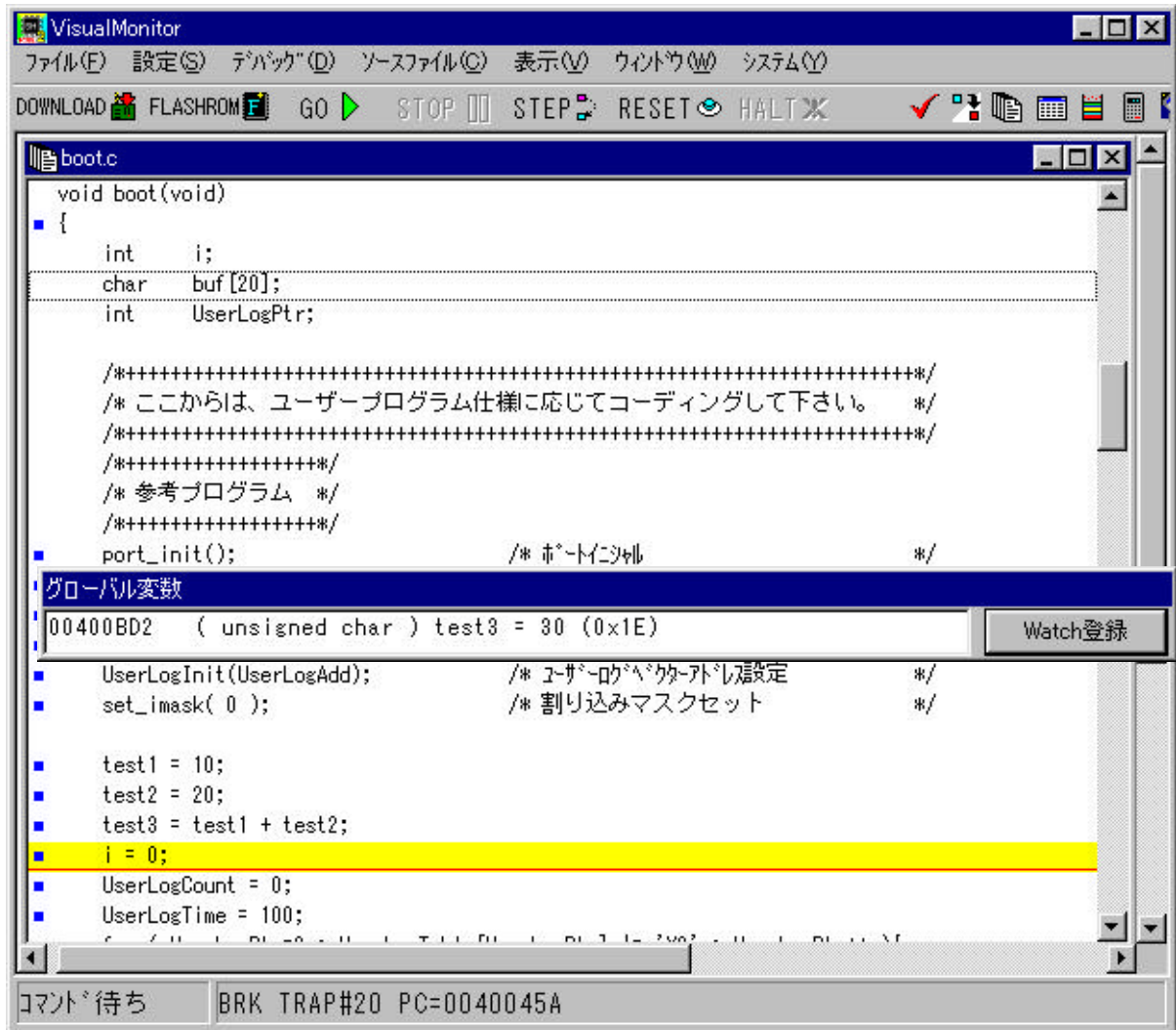
すると、変数の内容が10進数と16進数で表示されます。

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右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 3.2.2.11 変数表示



3.2.3 その他の機能

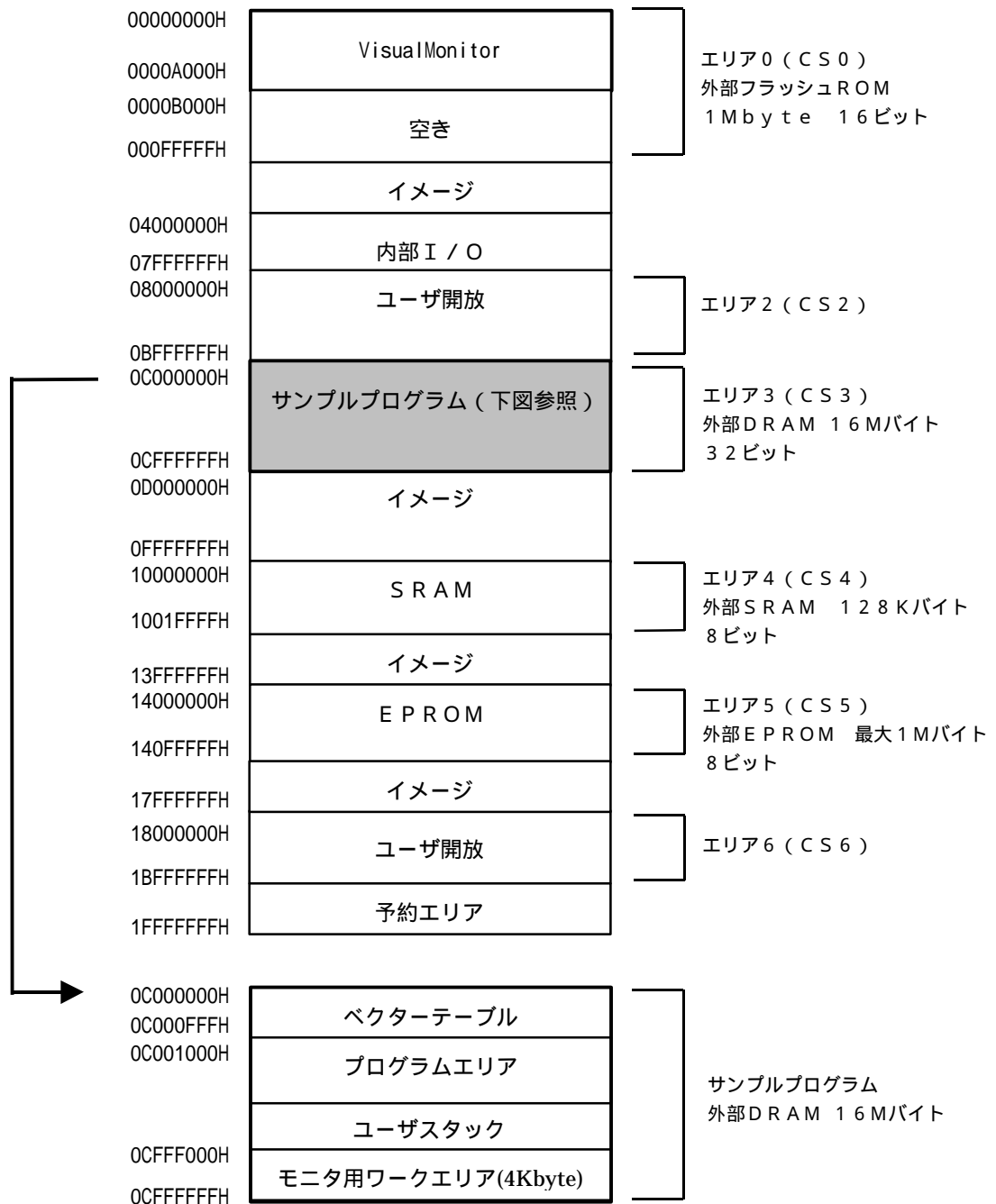
VisualMonitor には、その他の機能としてウォッチ、メモリダンプ、メモリ編集、ユーザログ出力等、様々な機能があります。それらの機能については別冊の「リモートデバッガ編」を参照してください。

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、オンボードフラッシュROM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。

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。



テストプログラムは、タイマ割り込みのプログラムです。

3.2.5 VisualMonitor 正規版との違い

本キットに付属する VisualMonitor / SH は AP - SH3 - 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. AP - SH3 - 0A (SH7709) 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. フラッシュROM

4.1 フラッシュROMの書き込み方法

AP-SH3-0A では添付のソフトを利用してオンボードでプログラムの書き込みが可能です。

添付ソフトはW I N D O W S 9 5 / 9 8 / N T 4 . 0 / 2 0 0 0 対応で、モトローラSフォーマットに対応しています。
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) 書き込みソフトの準備

PC側のダウンロードソフトをインストールします。

特にインストーラはありませんので、サンプルディスクの¥download¥pc¥sh3dl.EXE を適当なフォルダにコピーしてください。

次にターゲット側のダウンローダプログラムを用意します。

サンプルディスクのダウンローダプログラムをE P R O Mに書き込み、ボード上にソケットに実装します。

使用するROMは「CPUボード編」のROMサイズの設定をご覧ください。

通常は弊社出荷時に既にダウンローダを書き込んだROM（ビッグエンディアン用）が実装されていますので必要はありません。（リトルエンディアンで使用される場合には、リトルエンディアン用をE P R O Mに書き込んでご使用ください）

g c c と日立 c は記述が異なるだけで、動作は全く同一なのでどちらを使用しても構いません。

【ダウンローダプログラム】

ビッグエンディアン用	¥download¥target¥hitachi¥big¥flash2.mot（日立c）
	¥download¥target¥gcc¥big¥flash2.sr（gcc）
リトルエンディアン用	¥download¥target¥hitachi¥little¥flash2.mot（日立c）
	¥download¥target¥gcc¥little¥flash2.sr（gcc）

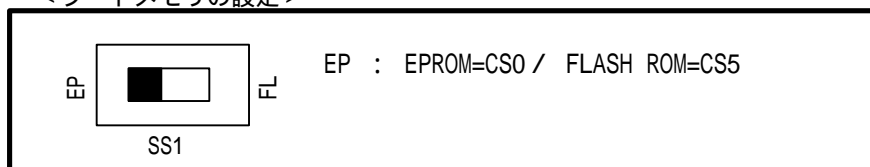
2) ボードの準備

モードの設定

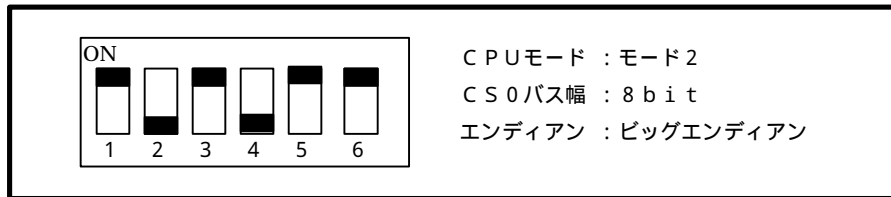
C P Uのモードを以下の設定に合わせます。設定は電源を切った状態でおこなってください。

ブートメモリ : E P R O M C P Uモード : モード2

<ブートメモリの設定>



<ディップSW1 (S1) の設定 (ビッグエンディアンの場合)>



3) 書き込み手順

パソコンとボードをクロスケーブルで接続します。(接続は図4-1参照)

sh3dl.exe をダブルクリックして起動します。

[ポート]メニューより使用するパソコンのCOMポートを選択します。

CPUボードの電源を投入します。

[ファイル]メニューの[ダウンロード]を選択し、ダウンロードするファイルを選択します。

ファイルを選択すると自動的にダウンロードを開始します。ダウンロードが終了すると、転送終了が表示されます。(この時点ではまだフラッシュROMには書き込まれていません)

次に[ファイル]メニューの[書き込み]を選択します。

すると、書き込み開始が表示され、フラッシュROMに書き込みが開始されます。

終了すると書き込み終了が表示されます。

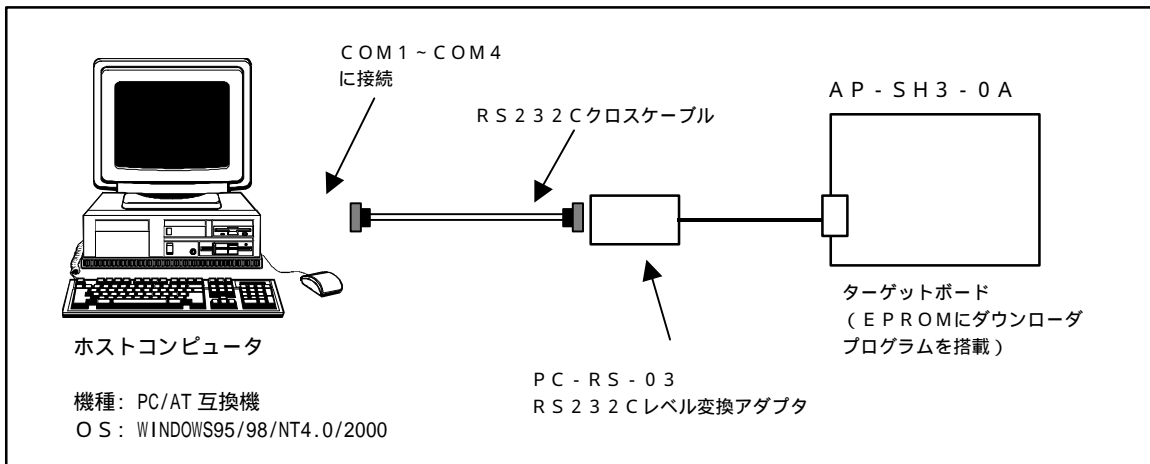


図 4 - 1 ダウンローダ使用時の接続

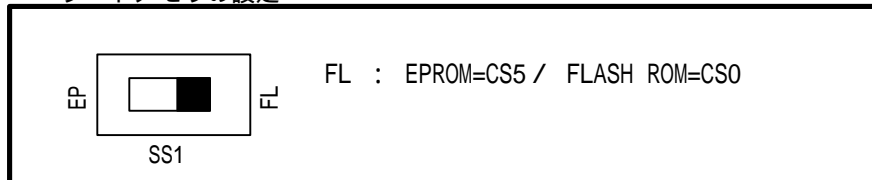
4) 動作の確認

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行ってください

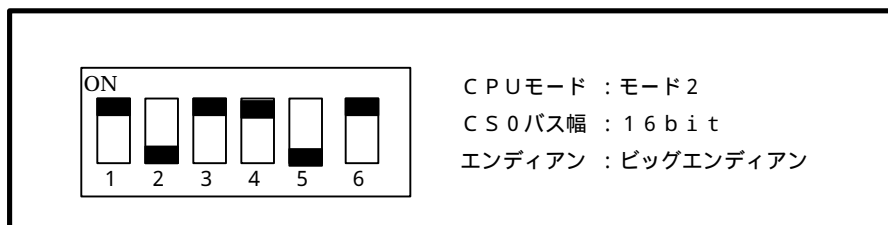
ブートメモリ : フラッシュROM

CPUモード : モード2

<ブートメモリの設定>



<ディップSW1 (S1の) 設定 (ビッグエンディアンの場合)>



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。

gcc用ROM化ファイル インストールフォルダ¥ROM化¥gcc¥
日立C用ROM化ファイル インストールフォルダ¥ROM化¥hitachi¥

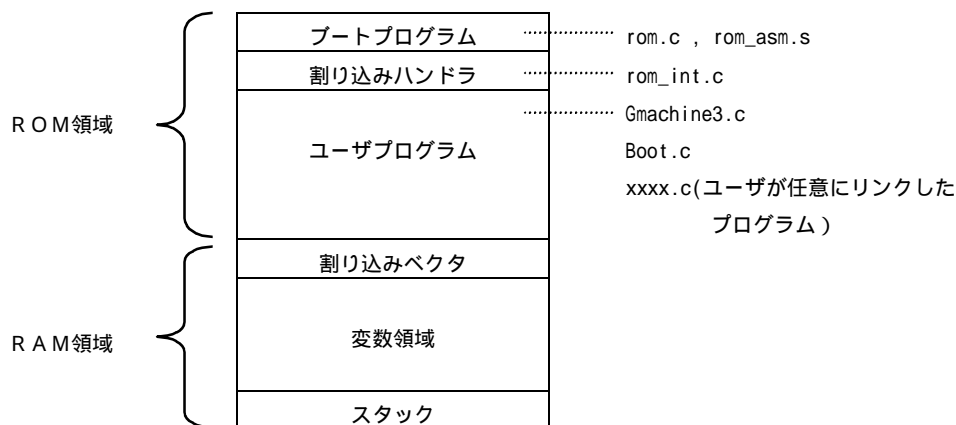
フォルダ内には以下のファイルが収められています。

• 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
• makeall.bat	サンプル生成用バッチファイル
• set.h	各種設定ファイル

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

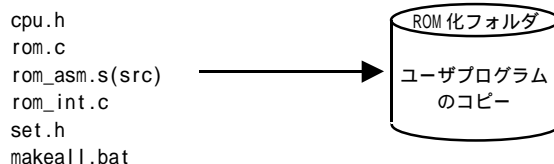
ブートプログラム	モニタプログラムの代わりとなるブートプログラムを配置します。サイズは1Kバイト以下になります。(ブートプログラム+割り込みハンドラ)
ユーザプログラム	ROM領域に配置します。
スタック	ユーザプログラム用のスタックのみとなります。
ユーザログ	ユーザログは出力されません。

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。

<ROM化手順>

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。



makeall.bat を必要に応じて編集します。ユーザが任意に作成したファイル等をリンクする場合には修正が必要です。

makeall.bat 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。

生成された*.sr(gcc)もしくは*.mot(日立C)を専用ダウンローダでオンボードフラッシュROM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 RS232C変換アダプタ(PC-RS-03)

6.1.1 概要

PC-RS-03は、アルファボードシリーズのインターフェースコネクタに接続して使用するRS232Cレベル変換ユニットです。3V～5.5Vまでの電圧範囲に対応しています。

6.1.2 仕様

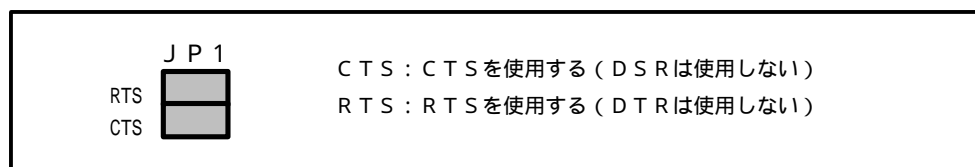
PC-RS-03仕様

ドライバ	MAX3222ECWN (MAXIM社) もしくは SP3222ECT (SIPEX社)
最大通信速度	235 Kbit / 秒
入出力電圧	RS232C入力電圧 : MAX ± 2.5 V RS232C出力電圧 : MIN ± 5.0 V
外部接続	TTL側 : 6PINコネクタ 日圧 NHコネクタ RS232C側: Dサブ9PIN オス
電源電圧	+3.0V～5.5V
自己消費電流	1.0mA (無負荷時) 10μA (SHUTDOWN時)
使用環境条件	0～70 20～80%RH 結露なし
寸法	38mm×64mm
添付品	CPUボード接続用ハーネス (6P)

6.1.3 設定

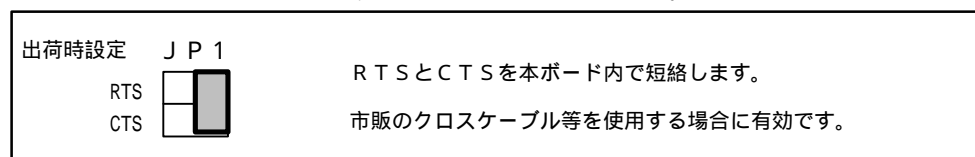
1) JP1

CPUボード側と接続する信号を設定します。



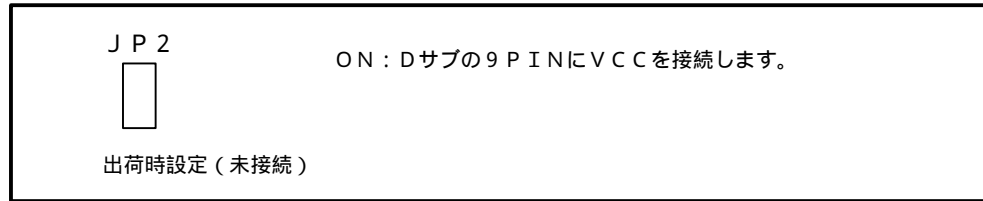
通常はCTSとRTSの組み合わせ、もしくはDSRとDTRの組み合わせで使用してください。

CTSとRTSを使用しない場合は以下のように短絡してご使用いただけます。



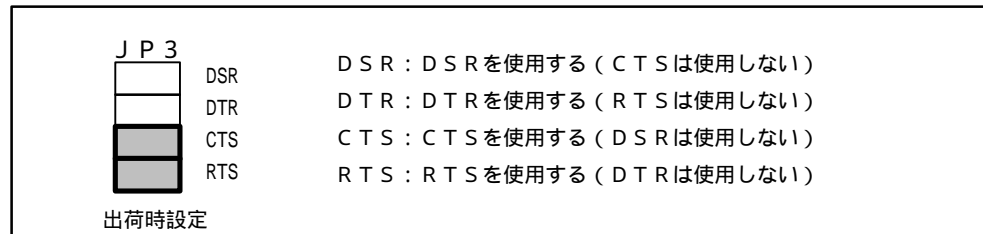
2) J P 2

相手側外部機器に電源を供給する場合に使用します。
通常はOFFにしてください。



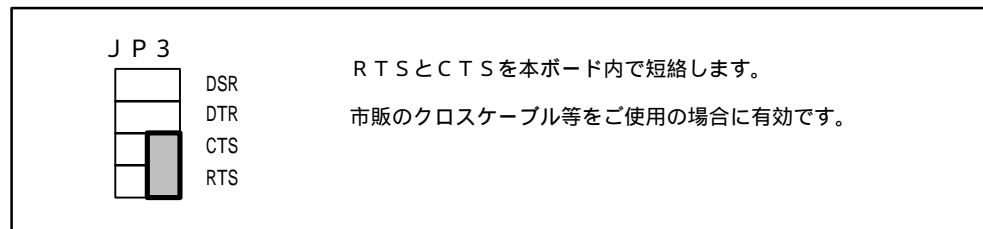
3) J P 3

相手側機器 (P C 等) と R S 2 3 2 C レベルで接続する信号を設定します。



通常は C T S と R T S の組み合わせ、もしくは D S R と D T R の組み合わせで使用してください。

C T S と R T S を使用しない場合は以下のように短絡して御使用いただけます。

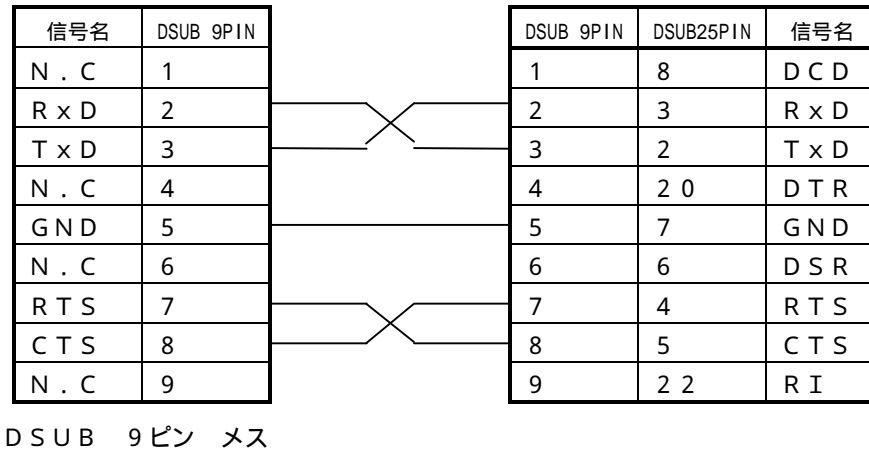


上記の設定は前述の J P 1 で R T S と C T S を短絡した場合と等価です。

6.1.4 接続方法

本ユニットをパソコン等と接続する場合のケーブル結線例を以下に示します。
市販のRS232Cクロスケーブルをご使用いただけます。

ケーブル結線例

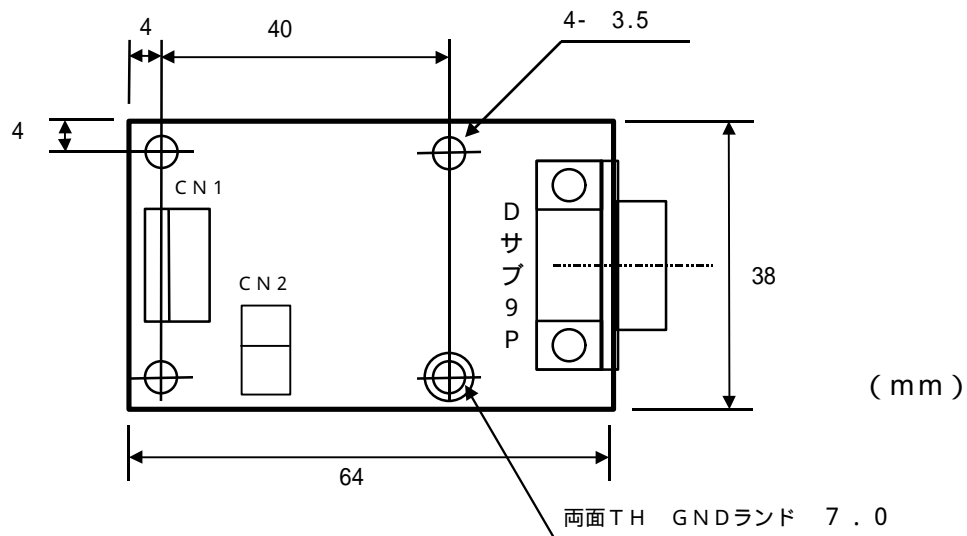


6.1.5 SHUTDOWN機能

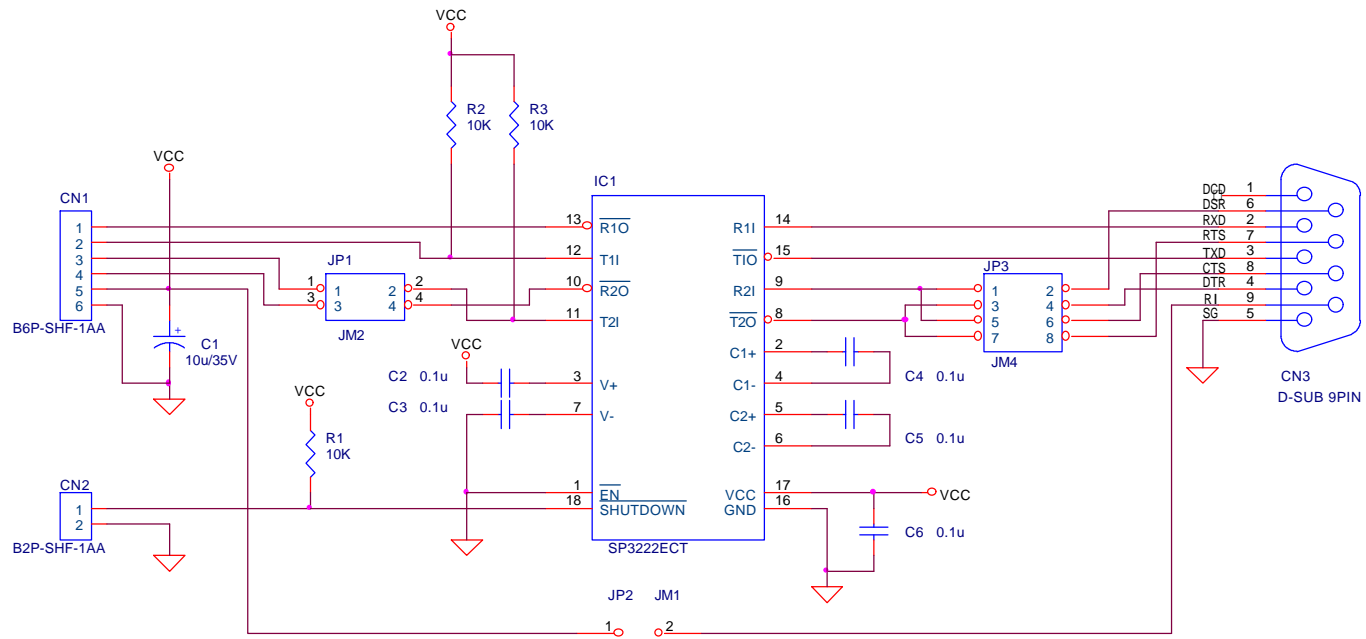
本ユニットにはSHUTDOWN機能があります。
CN2（未実装）の1pをGNDレベルにすることで、RS232Cドライバは、SHUTDOWN状態となり、消費電流が低下します。CPUのポートなどからコントロールできます。

6.1.6 外形寸法

図6.1 PC-RS-03基板寸法



6.1.1.7 回路图



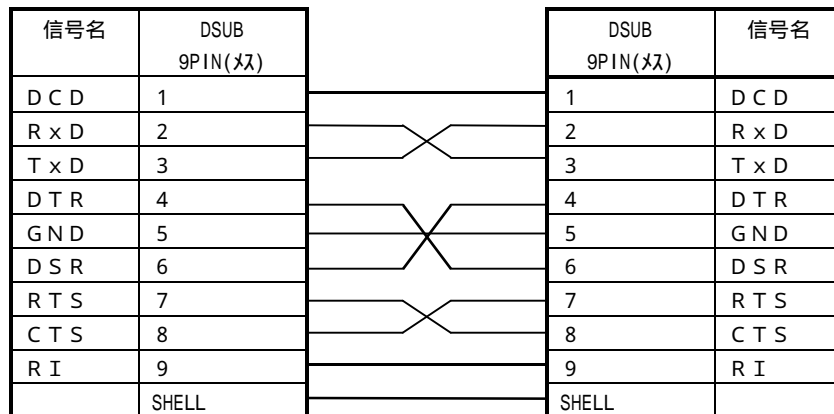
ALPHA PROJECT INC.		
Title PC-RS-03 RS232C ADAPTER BOARD		
Size A	Document Number CD-RS-0030-001	Rev 1.0
Date:	Monday, July 12, 1999	Sheet 1 of 1

6.2 RS232Cケーブル

6.2.1 結線

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、Visual Monitor、ダウンロード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コンパイラについては、サポート及び保証はおこなっておりません。

もし、技術的な内容で質問がある場合には、FAXもしくはE-MAILのみ受け付けますので、下記の連絡先までお送りください。回答の責務を負うものではありませんが、弊社で回答可能な範囲でお答えいたします。

なお、コンパイラの使用法、プログラムの作成方法等の技術指導的な質問については御遠慮ください。

各サンプルプログラムについては、何ら動作を保証するもの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

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

Visual Monitor パッケージ限定版、ダウンロードについては、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は一切おこないません。

お問い合わせ先

株式会社	アルファプロジェクト
〒433-8120	静岡県浜松市上島4-4-24
FAX	(053)464-3737 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。

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 など間違え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。

添付のMAKEALL.BAT にライブラリパス指定行がありますので、その部分をお客様の環境に合わせて変更してください。

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 8 . オンボードのフラッシュROM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10万回となっていますのでプログラムの書き換えならば、ほぼ恒久的と考えていいでしょう

Q 9 . EPROMは3.3V品が必要ですか？

A 9 . 3.3V品と5V品の両方が使用可能です。

3.3V品のEPROMはほとんどメーカーより供給されていないので、5V品をお使いいただけるようになっています。

Q 1 0 . リトルエンディアンのプログラムをデバッグしたいのですが、正常に動作しません。

A 1 0 . 本 K I T の出荷時の状態はビッグエンディアンの設定になっています。

リトルエンディアンのプログラムをデバッグするには以下の準備が必要です。

ボード上の D I P - S W をリトルエンディアンに設定する。

リトルエンディアン用のダウンロードを E P R O M に書きこみ、ボードに載せかえる。

V i s u a l M o n i t o r のリトルエンディアン用のターゲットモニタをフラッシュ R O M に書き込む。

ユーザプログラムをリトルエンディアンで作成する。

本 K I T ではリトルエンディアン用のプログラムが全て添付されていますので、それらをご使用ください。

Q 1 1 . モトローラ S フォーマットは S 1 、 S 2 、 S 3 がありますが、ダウンロード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 1 . 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 2 . 添付の g c c は、他の S H マイコンでも使用可能でしょうか？

A 1 2 . 使用可能です。S H 7 7 0 9 シリーズや S H 7 7 0 8 シリーズ等の S H 3 マイコンのプログラムを作成できます。

S H 1 や S H 2 では、ライブラリを使用しないプログラムであれば作成できます。

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、誤りなどお気付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（SH3 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