

# MIPS ベース組み込みマイコンを 使ってみよう！

川本 泰久

この章では MIPS を搭載した組み込みマイコン・ボードを使って課題を作成する。今回使用する E!Kit-1100 には ROM モニタとして YAMON が搭載されており、プログラムをロードして実行する機能がある。そこで、課題プログラムは YAMON を使用してロードして実行することとする。  
(編集部)

入社してから数週間が過ぎ会社にも慣れてきたころ、先輩に課題を渡されました。

- 使用する評価ボードは自由に選べる
- LED をタイマ割り込みで周期的に点灯制御する
- タクト・スイッチの入力を判定して LED の点灯パターンを切り替える

新人「先輩！評価ボードを自由に選べと言われても、何を選べばよいのかさっぱりわかりません」

先輩「うちの会社は、MIPS プロセッサを使用したボードの開発実績があるから、MIPS プロセッサのものを選ぶのはどうかな。とりあえず、代表的なボードの一覧表(表1)を渡しておくから参考にしてくれ」

いろいろ悩みましたが、今回は、(株)デバイスドライバーズが開発、販売している E!Kit-1100 という Au1100(AMD)を採用した組み込み Linux ボードと、専用の拡張ユニバーサル・ボードを使用することにします。

表1 MIPS プロセッサ組み込み評価ボード一覧

ボード・メーカー	製品名	CPU
(株)デバイスドライバーズ <a href="http://www.devdrv.co.jp/">http://www.devdrv.co.jp/</a>	E!Kit-1100	Au1100-400MHz
LASER5 <a href="http://www.laser5.co.jp/">http://www.laser5.co.jp/</a>	L-Card+16M	V <sub>R</sub> 4181-66MHz
(有)ハンブルソフト <a href="http://www.humblesoft.com/">http://www.humblesoft.com/</a>	N-Card	V <sub>R</sub> 4131-200MHz
ぷらっとホーム(株) <a href="http://www.plathome.co.jp/">http://www.plathome.co.jp/</a>	OpenMicro Server	Au1550-400MHz
東亜ディーケーケー(株)	TCS-8000	V <sub>R</sub> 4181A-131MHz

注1：MIPS Technologies が公開している。

## 1 「Hello World」を表示する

先輩「MIPS プロセッサを使用したボードには、YAMON<sup>注1</sup> という ROM モニタ(ブートローダ)を採用しているものがあり、E!Kit-1100 にも採用されているようだ。そこで今回の課題は、YAMON から直接起動して動作するソフトウェアとして作ってみよう」

新人「せっかくの Linux 対応ボードなので、Linux 上で動作するものではダメなのですか？」

先輩「まあそれでもいいんだが、今回はよりハードウェアに近い部分についても理解するために、OS を介さないソフトウェアにしたほうが良いと思う。そこでまずは開発環境を整え、その確認もかねて Hello World をコンパイル、実行してくれ」

### 開発環境の準備

今回の機器構成を図1に示します。今回使用している JTAG デバッガには Windows が動作している PC が必要なので、PC2

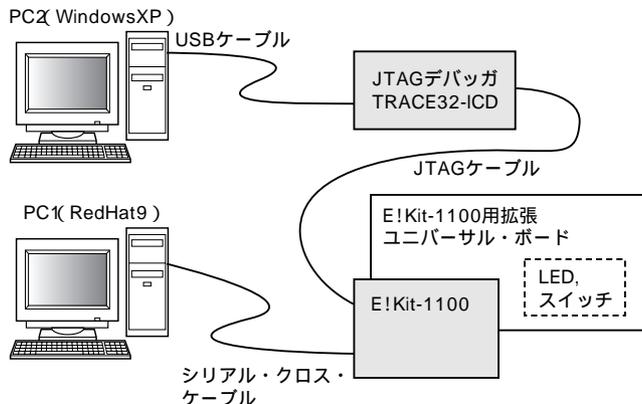


図1 開発機器の構成

```
$ tar zxvf yamon2.6.tgz
$ cd yamon-02.06
$ unzip yamon-sampleappl-src-02.03.zip
```

図2 サンプル・ソフトウェアの展開

```
$ cd yamon_appl/helloworld
$ make depend
$ make
```

図3 HelloWorld のコンパイル

## リスト1 サンプル・アプリケーション用パッチ

```
diff -cNr yamon_appl/helloworld/makefile yamon_appl-ekit/
                                helloworld/makefile
*** yamon_appl/helloworld/makefile
                                2002-09-18 22:14:02.000000000 +0900
--- yamon_appl-ekit/helloworld/makefile
                                2006-02-02 14:51:25.000000000 +0900
*****
*** 91,97 ****
# to your specific environment.
# *****
ifndef TOOLCHAIN
! TOOLCHAIN = sde
endif

ifeq ($(TOOLCHAIN),cygnus)
--- 91,111 ----
# to your specific environment.
# *****
ifndef TOOLCHAIN
! TOOLCHAIN = ekit
! endif
!
! ifeq ($(TOOLCHAIN),ekit)
! CC      = /export/local/bin/mipsel-linux-gcc
! LD      = /export/local/bin/mipsel-linux-ld
! OBJ2BIN = /export/local/bin/mipsel-linux-objcopy -O binary
! OBJ2REC = /export/local/bin/mipsel-linux-objcopy -O srec
! OBJDUMP = /export/local/bin/mipsel-linux-objdump -S
! ifeq ($(ENDIAN),EL)
! OFORMAT = elf32-tradlittlemips
! else
! OFORMAT = elf32-tradbigmips
! endif
! endif

!
endif

ifeq ($(TOOLCHAIN),cygnus)
*****
*** 166,171 ****
--- 180,192 ----

INCLUDE = -I$(ROOT)/../include

+ ifeq ($(TOOLCHAIN),ekit)
+ CC_OPTS = -c -g -O2 -$(ENDIAN) $(INCLUDE) -D$(ENDIAN)
                                -mips32 -fno-pic -mno-abicalls -fno-builtin
+ CC_OPTS_A = $(CC_OPTS) -D_ASSEMBLER_
+ LD_SCRIPT = $(ROOT)/link.xn
+ LD_OPTS = -T $(LD_SCRIPT) -o $(IMAGE_ELF) -Map $(IMAGE_MAP)
                                --oformat $(OFORMAT)
+ else
+
+ ifeq ($(TOOLCHAIN),ghs)
+ CC_OPTS = -c -cpu=mips32_4kc -O2 -embedded_calling_sequence
+ inline_prologue -sda=all -nostartfiles $(INCLUDE) -D$(ENDIAN)
+ CC_OPTS_A = $(CC_OPTS) -D_ASSEMBLER_ -preprocess_assembly_files
*****
*** 185,190 ****
--- 206,212 ----

endif

+ endif

# *****
# Files to be compiled
```

を別途用意しています(JTAG デバッガを使用しない場合は不要)。クロスコンパイラについてはMIPS Technologiesのサイトからダウンロードしたり、自分でコンパイルすることもできますが、今回はE!Kit-1100に添付されているものをマニュアルどおりにPC1にインストールし、開発環境の準備を行います。

ソフトウェアのサンプルは、MIPS TechnologiesのサイトからダウンロードできるYAMONのソース・ファイルに含まれているようです。しかし最新版の02.11には含まれていないようですので、02.06をダウンロードする必要があります。

ダウンロードしたファイルは図2のように展開しますが、ファイルをダウンロードするとき拡張子はtgzのままgzipで展開されてしまうことがあるようです。その場合はzオプションなしでyamon2.6.tgzを解凍します。

## ソフトウェアのコンパイル

HelloWorldは図3のようにコンパイルしますが、そのままではコンパイル・エラーが発生したのでmakefileを変更します。GCCに以下のオプションを追加しました。

- -fno-pic-mno-abicalls : 位置独立なコード(PIC : Position Independent Code)関連の抑制をする
- -fno-builtin : 最適化でprintf関数がputs関数に置き換わったりする可能性があるのをそれを抑止する

変更内容をリスト1にパッチ・ファイルとして掲載しておきます。コンパイルに成功するとSレコード形式のファイル(app1.rec)ができあがります。

新人「ところで先輩、このSレコード形式って何ですか？」  
先輩「モトローラSレコード形式とも言って、アドレスとデータ、チェック・サムをASCII形式(テキスト形式)で表したフォーマットだよ。同じようなものにインテルHEX形式というものがある。YAMONはSレコード形式のファイルの読み込みをサポートしているからそのままロードして、実行できるよ」

## ソフトウェアの実行

コンパイルが終了したら早速実行してみます。Linuxカーネルなどのサイズの大きなソフトウェアをロードする場合はネットワーク経由で行うと速くて便利ですが、今回はソフトウェアのサイズが小さいのでシリアル・ポートからロードしてみます。

PC1上でminicomを起動してE!Kit-1100の電源をONにすると図4のようなメッセージが表示されてYAMONが起動します。コンパイルしたソフトウェアをロードするにはYAMONのloadコマンドを使用します。loadコマンド実行後、キーボードのctrl+A, Sキーを押すと図5のような画面になるので、app1.recファイルをasciiモードでアップロード(送信)します。図6