

付録

1. ユーザマニュアル

1.1. OS 操作マニュアル

(1) ハードウェアの準備

以下のものを用意する.

- PC
- シリアルと USB のコンバータ (/dev/ttyUSB0 へ接続する)
- シリアルケーブル
- H8/3069f マイコンボードと電源ケーブル

(2) OS 操作ツール類の準備

以下のものを用意する.

- ホスト OS に Debian GNU/Linux である ubuntu または Fedora
- ブートローダを書き込むための ROM ライタである, H8write 参考文献[36]または kz_h8write(参考文献[37])のどちらか使用
- エミュレータである minicom(参考文献[38])または kermi(参考文献[39])のどちらか使用
- OS 実行ファイルを転送する lrzsz(参考文献[40])

以下の(2-1)～(2-5)にインストール方法, 各ツールの設定方法を説明する. ubuntu または Fedora のインストール方法については説明しないものとする. ちなみに, ホスト OS に ubuntu, ツール ROM ライタに H8write, エミュレータに minicom, を使用方法が最も簡単である.

(2-1) H8write のインストール

H8write のインストールは, 参考文献[36]のソースコードまたは, H8/3069f マイコンボード付属の CD-ROM のバイナリーイメージからインストールできる. 参考文献[36]の最新版の方がフラッシュ ROM 書き込みエラーが少ない.

(2-1-1) ソースコードからビルドする場合

```
% gcc h8write.c -o h8write -Wall
```

より, ビルドする.

(2-1-2) バイナリーイメージを使用する場合

```
%chmod +x h8write
```

により, 実行フラグを立てておく

(2-2) kz_h8write

(2-1)の ROM ライタでうまく書き込めない場合は, kz_h8write を使用する. 参考文献[36]のソースコードをビルドする

```
% gcc kz_h8write.c -o kz_h8write -Wall
```

(2-3) minicom のインストール

ホスト OS が ubuntu であれば、インターネットに接続した状態で Synaptic を起動し、minicom のインストールを行うか、参考文献[38]のソースコードからビルドを行う。

(2-3-1) minicom の設定

スーパーユーザになり、以下で minicom の設定を行う。

```
# minicom -s -o
```

-s オプションで minicom 設定メニューを表示できる。ここでシリアルポートを /dev/ttyUSB0 設定 (ホスト PC に /dev/ttyUSB0 へ接続している事を確認する) をし、ボーレートを 9600bps (9600bps 以上はエラーになる)、データ長を 8 ビット、ストップビットを 1、パリティ無しに設定する (96008N1)。さらに、ハードウェアフロー制御、ソフトウェアフロー制御、XOFF/XON を無効にする。完了後、設定を保存する。

minicom の起動は以下で行う。

```
minicom -o
```

minicom の終了は以下で行う。

[Ctrl]+[a]を入力し、続けて[x]を押す。

(2-4) kermit のインストール

インストール方法は(2-3)と同様となる。

(2-4-1) kermit の設定

スーパーユーザとなり、以下で kermit の設定を行う。

```
# kermit
```

より、kermit の起動ができる。プロンプトが出力され、以下でシリアルデバイスの設定を行う

```
.....> set line [シリアルデバイス名]
```

完了後、以下でシリアル接続を行う。

```
.....> connet
```

(2-5) lrzsz のインストール

(2-3)と同様の手順でインストールを行う。エミュレータに minicom を使用する場合は lrzsz 設定はない。

(3) H8/3069f マイコンボードのフラッシュ ROM へブートロード書込み

まず、ハードウェアの準備を行う。H8/3069f マイコンと電源ケーブルを接続し、H8/3069f マイコンボードと PC を接続する。

マイコンの CPU 動作モードをフラッシュ ROM 書込みモードに設定する。フラッシュ ROM 書込みモードは、マイコンボードのディップスイッチを ON, ON, OFF, ON にする。

次に、ROM ライタのソースコードと実行イメージをカレント bootload フォルダから

.././tools/h8write/に指定しておく。なお、Makefile の設定を変更すれば、指定はい

らない。以下のように bootload.mot ファイルをブートローダを書き込む

```
・・・>% make write          //フラッシュ ROM へ書き込み
```

なお、PC の USB コンバータを利用して PC とマイコンボードを接続してる場合は、ホスト OS 起動後一回しか書き込みができない。よって書き込みに失敗した場合は、再度 OS を再起動して書き込み操作を行う。

(4) ブートローダの起動

H8/3069f マイコンと電源ケーブルを接続し、H8/3069f マイコンボードと PC を接続する。マイコンの CPU 動作モードを実行モードに設定する。実行モードはマイコンボードのディップスイッチを ON, OFF, ON, OFF にする。次にエミュレータを起動し、ブートローダを起動する。ブートローダの起動はマイコンボードのリセットベクタで行う(リセットベクタを押す度にブートローダが起動される)。ブートローダが起動するとブートローダのプロンプトが表示され、ブートローダのコマンドが体験できる。ブートローダのコマンドと内容を以下に示す。

- load

[説明]

OS 実行ファイルを XMODEM プロトコルによってマイコンボードへ送信する。

- run

[説明]

OS の起動をするコマンドである(OS のエン트리ポイントにプログラムを切り替える)。

- dump

メモリ 16 進ダンプ表示をする。

- ramchk

DRAM への read/wirte チェックコマンドである。一括書き込み後、一括読み込みを行う。

- ramchk2

DRAM への read/wirte チェックコマンドである。段階的に書き込み後、読み込みを行う。これは、DRAM のリフレッシュレートチェック用となる。

- ramclr

DRAM のメモリを掃く

dump, ramchk, ramchk2, ramclr の実行結果を図 1 に示す。

```
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
minicom へようこそ 2.4

オプション: I18n
コンパイルされた日時は: Jun  3 2010, 13:46:31.
Port /dev/ttyUSB0

CTRL-A Z を押すと、説明画面になります。

kzload (kozoz boot loader) started.
kzload> ramclr ← ramclrコマンド
DRAM clearing...
DRAM cleared.
kzload> dump ← dumpコマンド
size: ffffffff
no data.
kzload> ramchk ← ramchkコマンド
DRAM checking...
00403990kzload (kozoz boot loader) started. ← リセットベクタで
kzload> kzload (kozoz boot loader) started.  中断
kzload> ramchk2 ← ramchk2コマンド
DRAM check pattern: 0
DRAM setting...
005ff000
DRAM checking...
005ff000
DRAM check pattern: 1
DRAM setting...
004af000kzload (kozoz boot loader) started. ← リセットベクタで
kzload> ram  中断
```

図1 ブートローダのコマンド実行結果

(5) OS の起動

エミュレータが起動するディレクトリにOS 実行ファイルである kernel を置く．次にエミュレータを(2-3-1)の方法で起動し，リセットベクタでブートローダを起動する．起動後，load コマンドを入力し，[Ctrl+a]に続けて[s]を入力し，ファイル転送画面に入る．XMODEM を選択し，kernel と入力する．図2 に示す．

```
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
minicom へようこそ 2.4

オプション: I18n
コンパイルされた日時は: Jun  3 2010, 13:46:31.
Port /dev/ttyUSB0

CTRL-A Z を押すと、説明画面になります。      ド]--+

kzload (kozoz boot loader) started.
kzload> load
+-----+
| ファイル名を入力して下さい |
|> kernel|
+-----+

説明画面 CTRL-A Z | 9600 8N1 | NORminicom 2.4 | VT102 | 切断
```

図2 転送するOS実行ファイル選択画面

[Enter]を入力すると、OS 実行ファイル転送処理に入る。転送処理画面を図 3 に示す。

```
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)

minicom へようこそ 2.4

オプション: I18n
コンパイルされた日時は: Jun  3 2010, 13:46:31.
Port /dev/ttyUSB0

+-----[xmodem アップロード - CTRL-C で終了します]-----+
CTRL-A Z | Sending kernel, 549 blocks: Give your local XMODEM receive co
          | mmand now.
kzload (k | Xmodem sectors/kbytes sent: 74/ 9k
kzload> l
```

図3 OS実行ファイル転送処理画面

なお、load コマンド入力から OS 実行ファイル転送処理までのユーザ入力操作が遅いとファイルが転送できないので、素早く行う。

OS が転送されると、XMODEM receive succeeded メッセージが表示され、ブートローダのプロンプトが表示されるので、(4)の run コマンドより OS を起動する。ここまですを示す。

```
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)

minicom へようこそ 2.4

オプション: I18n
コンパイルされた日時は: Jun  3 2010, 13:46:31.
Port /dev/ttyUSB0

CTRL-A Z を押すと、説明画面になります。

kzload (kzos boot loader) started.
kzload> load

XMODEM receive succeeded.
kzload> run
starting from entry point: 400100
kernel boot succeed!
  syscall handler ok
  softerr handler ok
  timer handler handler ok
  nmi handler ok
init task started.

```

図4 OS起動要求画面(runコマンドの応答結果)

OSのプロンプト表示後は、ユーザタスクセットの選択を行う(タスクセットは1~26).
タスクセットの選択し, [Enter]を押すとユーザタスクセットのレスポンスを体験できる.
図5に示す.

```
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
minicom へようこそ 2.4

オプション: I18n
コンパイルされた日時は: Jun  3 2010, 13:46:31.
Port /dev/ttyUSB0

CTRL-A Z を押すと、説明画面になります。

kzload (kozoz boot loader) started.
kzload> load

XMODEM receive succeeded.
kzload> run
starting from entry point: 400100
kernel boot succeed!
  syscall handler ok
  softerr handler ok
  timer handler handler ok
  nmi handler ok
init task started.

unknown.
> task set1
2 tsk next counter
> sample_task3 started.
test5_7_5 create task.
sample_task3 create running in (sample_task2).
sample_task2 started.
4 tsk next counter
sample_task1 create task.
sample_task2 create running in (sample_task1).
sample_task1 started.
1 sample_task1 task priority
tskid 3 4 change priority activ(context syscall) for interrput handler.
sample_task2 create running out (sample_task1).
not get kernel object for task dormant or non exsitent.
sample_task1 DORMANT.
sample_task1 termination task.
delete task contorol block for interrput handler
sample_task1 delete task.
sample_task2 EXIT.
sample_task3 create running out (sample_task2).
sample_task3 DORMANT.

unknown. CTRL-A Z | 9600 8N1 NORinicom 2.4 | VT102 切断
> 
```

図5 ユーザタスクセットの選択と応答結果画面

なお, どのような内容のタスクセットか実装されているかは, 可読マニュアルを参照して頂きたい.