

U-Boot コマンドの使用

● コマンドライン

uboot のコマンドラインは、**BusyBox** の **Hush** を **uboot** 用にしたものです。スクリプトを作成する事もできます。簡単なスクリプトは環境変数に設定して保存しておき、設定した文字列を **run** コマンドで実行します。

● 変数

シェル変数と環境変数があります。

環境変数は、**saveenv** コマンドでフラッシュに保存可能ですが、シェル変数は保存されません。環境変数の設定は **setenv** コマンドを使います。**setenv name value** シェル変数への設定は、**name=value** とします。

変数を参照するには、**\${name}** もしくは、**\$name** とします。

環境変数は、シェル変数より優先され、同じ名称のシェル変数は使えなくなります。環境変数に設定したコマンドは **run** コマンドで実行できますが、シェル変数は **run** コマンドに渡せません。

環境変数には **uboot** のコマンドが利用する予約された変数がたくさんあります。

シェル変数 **\$?** には、最後に実行したコマンドの終了ステータスが入ります。

● Hush 文法

リスト： コマンドを **;**、**&&**、**||** のどれかで区切って並べたもの

リスト自体の終了ステータスは最後に実行したコマンドの結果になります。

コマンド **1** ; コマンド **2**

コマンド **1** を実行してからコマンド **2** を実行します。

コマンド **1** && コマンド **2**

コマンド **1** の終了ステータスが **0** の場合のみコマンド **2** が実行されます。

コマンド **1** || コマンド **2**

コマンド **1** の終了ステータスが **0** 以外の場合のみコマンド **2** が実行されます。

for name in 単語のリスト; **do** リスト; **done**

単語のリスト一つずつについて、その単語をシェル変数 **name** に設定してからリストを実行します。

if リスト; **then** リスト; [**elif** リスト; **then** リスト;] ... [**else** リスト;] **fi**

if のリストと **elif** のリスト部を順番に実行し、終了ステータスが **0** のリストに出会ったら、対応する **then** のリスト部を実行して終了します。出会わなかった場合には **else** 部のリストが実行されます。

while リスト; **do** リスト; **done**

while のリスト部が終了ステータス **0** 以外を返すまで、**do** のリストを繰り返し実行します。

until リスト; **do** リスト; **done**

until のリスト部が終了ステータス **0** を返すまで、**do** のリストを繰り返し実行します。

- クオート処理

\$ や **;** をエスケープするには、**** を使います。

" で括った文字列内に **\${name}** 等変数の参照がある場合、変数は展開されますが、' で括った場合、展開されずにそのままの文字列になります。

2.6.4 改行だけの入力行の扱い改行だけの入力は、最後のコマンドの再実行です。注意して下さい。

- **TAB** による補完とコマンドの省略形

コマンドの最初の数文字入力したあと、**TAB** キーを入力する事により、補完可能な範囲で補完され、候補が複数あれば、候補の一覧が表示されます。

コマンドは最初の数文字だけでも、それがどのコマンドであるか判別可能な範囲であれば、残りは省略できます。

例えば、**tftpboot** コマンドは、**tftp** や **tf** でも **tftpboot** と解釈されます。

- **^C** による中断

CtrlC で処理の中断が出来ますが、コマンドによっては中断できないので、中断されるまでに時間がかかります。例えば、**erase** コマンドは、処理中には止められません。

u-boot コマンド一覧

uboot での数値の入力は基本的に **16** 進数として解釈されます。

引数省略時のデフォルトや、省略可能な引数が複数ある場合に一つだけ引数を与えた場合の動作(解釈方法)はコマンドによって異なりますし、コンパイル時の機能の選択(特に、**CFG_HUSH_PARSER**)によって、書き方が変わったりしますので、出荷時の状態では、こういう使い方になるはずであるという説明です。また、全てのコマンドの確認をしておりますが、未確認な部分は、未確認と明記してあります。

- **? - 'help'** の別名

文法 : **? [command...]**

command: コマンド名

command が省略された場合、全てのコマンドとその概略の一覧を表示します。

command が指定された場合、そのコマンドの説明と使い方を表示します。

- **askenv** - 標準入力からの入力で環境変数を設定

文法 : **askenv name [size]**

askenv name [message] size

name: 環境変数の名称

size : 環境変数を読みこむ際の最大文字数の指定(**10** 進数で指定します)

1 以上 **255** 以下が可能です。

ユーザは **255** 文字まで入力可能ですが、入力された文字列は指定したサイズに切り詰められます。

message:入力をうながすプロンプト文字列を指定します。

環境変数を設定するには、**setenv** もありますが、**askenv** の利点は、特殊な文字列をクオートする必要がなく、入力した文字列がそのまま設定できる所にあります。

関連： **setenv printenv**

- **autoscr** - メモリ上のスクリプトの実行

文法： **autoscr [addr]**

addr: スクリプトの置いてあるアドレスです。省略した場合は、**80400000** です

スクリプトは、**mkimage** で作成されたものでなくてはなりません。

環境変数 **verify** が **n** で始まらない場合、チェックサムを検査し、正しいときのみ実行します。(デフォルトでは **verify=n** に設定しています。)

```
# iminfo 80400000
```

```
## Checking Image at 80400000 ...
```

```
Image Name: hello
```

```
Created: 2004-12-04 9:24:20 UTC
```

```
Image Type: MIPS U-Boot Script (uncompressed)
```

```
Data Size: 24 Bytes = 0 kB
```

```
Load Address: 80200000
```

```
Entry Point: 80200000
```

```
Verifying Checksum ... OK
```

```
# autoscr
```

```
## Executing script at 80400000
```

```
hello
```

```
#
```

関連： **run**

- **base** - メモリ関連のコマンドのベースアドレスの設定と表示

文法： **base [offset]**

offset: 設定したいオフセット

offset が省略されると、現在の設定値を表示します。

md mm nm mw cmp cp crc32 コマンドの引数にアドレスを指定したときに、**base** コマンド

で設定された値が自動的に加算されます。

再起動すると **0** に初期化されます。

デフォルトで環境変数に設定してあるスクリプトは、**base** コマンドで値を設定すると、動かなくなりますので、気を付けて下さい。

関連： **md mm nm mw cmp cp crc32**

- **bdinfo** - ボードの情報を表示

文法： **bdinfo**

RAM やフラッシュのアドレス、サイズなどのボードに関する情報を表示します。

- **boot** - デフォルトのブートコマンドの実行

文法： **boot**

bootd 環境変数 **bootcmd** は、電源投入時に自動で実行すべきコマンドが保存されています。

boot コマンドはこれを実行するコマンドです。

bootd というコマンドもありますが、**bootd** は過去互換の為の名前であり、**boot** とまったく同じコマンドです。

- **bootelf** - ELF イメージの **u-boot** 用アプリケーションの実行

文法： **bootelf [addr]**

addr: **ELF** ファイルが置いてあるアドレス。

省略した場合は、最後にファイルをメモリ上にロードしたアドレス

例：

```
# nfs \"192.168.3.91:/usr/src/mlldbox/u-boot/examples/hello_world\"
```

```
Using i82559#0 device
```

```
File transfer via NFS from server 192.168.3.91; our IP address is  
192.168.3.202
```

```
Filename '/usr/src/mlldbox/u-boot/examples/hello_world'.
```

```
Load address: 0x80400000
```

```
Loading: ##
```

```
done
```

```
Bytes transferred = 6798 (1a8e hex)
```

```
# bootelf 80400000
```

```
Loading .text @ 0x80200000 (672 bytes)
```

```
Loading .rodata @ 0x802002a0 (160 bytes)
```

```
Loading .reginfo @ 0x80200340 (24 bytes)
```

```
Loading .got @ 0x80200360 (56 bytes)
```

```
## Starting application at 0x80200000 ...
```

```
Example expects ABI version 2
```

Actual U-Boot ABI version 2

Hello World

argc = 1

argv[0] = "80400000"

argv[1] = "<NULL>"

Hit any key to exit ...

Application terminated, rc = 0x0

#

関連： bootm go

● bootm - OS とアプリケーションの起動

文法： bootm [addr [initrdaddr]]

addr: カーネルが置いてあるアドレスです。

省略した場合は、最後にファイルをメモリ上にロードしたアドレスになります。

initrdaddr: イニシャルラムディスクが置いてあるアドレスです。

省略した場合は、イニシャルラムディスクに関するオプションはカーネルに渡されません。**Linux** 以外の **OS** を起動する際は、**initrdaddr** を指定してはいけません。

mkimage で作られたファイルには、**OS** の種類等が埋め込まれています。その種類に応じたアプリケーションの起動を行います。

環境変数 **verify** が **n** で始まらない場合、チェックサムを検査し、正しいときのみ実行します。(デフォルトでは **verify=n** に設定しています。)

イメージタイプが **Linux** カーネルイメージの場合、カーネル引数は環境変数 **bootargs** に設定しておく、自動的に渡されます。

Linux カーネルと **uboot** 用アプリケーション以外での動作は不明です。

関連： bootelf bootvx iminfo

● bootp - BOOTP プロトコルで IPv4 アドレスを取得

文法： bootp [addr] [filename]

addr: ロードするアドレスです。省略した場合、環境変数 **loadaddr** の値、**loadaddr** も設定されていなければ、**80400000** です。

filename: ロードするファイルです。省略した場合は、**bootp** サーバの応答によって指定されたもの。サーバから指定も無かった場合は、環境変数 **bootfile** の値が使われます。

addr を省略して、**filename** を指定する場合、**filename** は" で括られていなくてもはいけませんが、パーサが" をはずしてしまうため、**bootp** コマンドに" が渡るように、エスケープする必要があります。

最初に **Bootp** プロトコルで **IP** アドレス、サーバアドレス、ダウンロードするファイル等を取得し、以下の環境変数を更新します。

serverip: bootp サーバの **IP** アドレス

ipaddr: 割り当てられた **IP** アドレス

bootfile : 読みこむべきファイル

(**filename** が指定されている場合、引数で渡したものがコピーされます)

gatewayip,netmask,hostname,rootpath,dnsip,dnsip2,domain:

サーバの応答で指定があった場合に更新され、取得できなかったものは以前の値が保持されます。

次に、環境変数の **autoload** をチェックして、

autoload の値が **n** で始まっていたら終了します。

autoload の値が **NFS** の場合、**nfs** コマンドを同じ引数で呼び出します。

autoload が上記以外の場合は **tftpboot** コマンドを同じ引数で呼び出します。

関連 : **dhcp nfs rarpboot tftpboot**

- **bootvx** - **ELF** イメージの **vxWorks** を起動

文法 : **bootvx [addr]**

addr: **vxWorks** の **ELF** イメージが置いてあるアドレスです。

bootm でも **vxWorks** は起動できますが、こちらは **ELF** ファイル版です。

このコマンドの動作は未確認です。

関連 : **bootm**

- **chpart** - アクティブパーティションの変更

文法 : **chpart num**

num:パーティション番号(02)

ls,fsload,fsinfo コマンドの対称にするパーティション番号を指定します
フラッシュ上のパーティションです。

再起動すると、**0**に戻ります。

関連 : **ls fsload fsinfo**

- **cmp** - メモリの比較

文法 : **cmp[.b, .w, .l] addr1 addr2 count**

[.b, .w, .l]: バス幅

演算するときに、バイト単位か(.b)、ハーフワード(2バイト)単位(.w)か、ワード(4バイト)単位(.l)かを指定します。省略した場合、.lになります。

addr1: 比較するアドレス

addr2: 比較するアドレス

count: 比較する数

[.b, .w, .l]の単位で数えます。また、16進で指定します。

addr1 から **count** 個のメモリ範囲と、**addr2** から **count** 個のメモリ範囲の内容が同じかどうか検査します。**count** はバイト数ではない事に注意して下さい。

例 :

```
# nfs 80400000 192.168.3.91:/usr/src/mldbox/u-boot.bin
```

```
Using i82559#0 device
```

```
File transfer via NFS from server 192.168.3.91; our IP address is  
192.168.3.202
```

```
Filename '/usr/src/mldbox/u-boot.bin'.
```

```
Load address: 0x80400000
```

```
Loading: #####
```

```
done
```

```
Bytes transferred = 199948 (30d0c hex)
```

```
# cmp.b BFC00000 80400000 $filesize
```

```
Total of 199948 bytes were the same
```

```
#
```

関連 : base md mm nm mw cp crc32

● coninfo - コンソールデバイスの表示

文法 : coninfo

コンソールデバイスの一覧を表示します。

● cp - メモリ間のコピー

文法 : cp[.b, .w, .l] source target count

[.b, .w, .l]: バス幅

演算するときに、バイト単位(.b)、ハーフワード(2バイト)単位(.w)か、ワード(4バイト)単位(.l)かを指定します。省略した場合、.lになります。

source: コピー元アドレス

target: コピー先アドレス

count: コピーする数

[.b, .w, .l]の単位で数えます。また、16進で指定します。

使い方の詳細は、**cmp** を参照して下さい。

関連 : base md mm nm mw cmp crc32

- **dhcp** - DHCP プロトコルで IPv4 アドレスを取得

文法 : **dhcp**

DHCP プロトコルでアドレスを取得し、以下の環境変数を更新します。

serverip: DHCP サーバの IP アドレス

ipaddr: 割り当てられた IP アドレス

bootfile : 読みこむファイル

gatewayip, netmask, hostname, rootpath, dnsip, dnsip2, domain:

サーバの応答で指定があった場合に更新され、取得できなかったものは以前の値が保持されます。

次に、環境変数の **autoload** をチェックして、

autoload の値が **n** で始まっていたら終了します。

autoload の値が **NFS** の場合、**nfs** コマンドを引数なしで呼び出します。

autoload が上記以外の場合は **tftpboot** コマンドを引数なしで呼び出します。

関連 : **bootp nfs rarpboot tftpboot**

- **echo** - テキストを表示

文法 : **echo [args ...]**

args: 出力する文字列

args を標準出力に出力します。出力する文字列に '\c' が含まれていると、改行は出力されません。

- **erase** - フラッシュメモリの消去

文法 : **erase start end**

erase N:SF[SL]

erase bank N

26

erase all

start: 開始アドレス(最初のイレースブロックの先頭アドレス)

end: 終了アドレス(最後のイレースブロックの最終アドレス)

N : バンク番号(常に 1 を指定します)

SF: イレースブロック番号

SL: フラッシュ内のイレースブロックを先頭から数えた番号(0 から数え始めます)

省略した場合、**SF** と同じ値になります。

指定された範囲内で、プロテクトされていないイレースブロックを消去します。

イレースブロック番号は、フラッシュ内のイレースブロックを先頭から数えた番号です。

(0 から数え始めます)

もし 1 バンクで構成されていた場合、**erase bank 1** と **erase all** は同じ意味になります
本機に搭載されているフラッシュは先頭と末尾の **128kbyte** はイレースブロックサイズ
が **16kbyte** で残りの部分は **128kbyte** です。場所によってサイズが変わりますので、イレ
ースブロック番号の計算には注意が必要です。アドレス指定を勧めます。

例

```
# erase BFC40000 BFFDFFFF
..... done
Erased 29 sectors
#
関連： protect
```

- **exit** - スクリプトの終了

文法： **exit**

スクリプトを終了します

mkimage で作るスクリプトは、最後を **NULL** ターミネートしづらいので、最後の行は
exit で終了する様にしておくと安全です。

- **flinfo** - フラッシュの情報表示

文法： **flinfo [N]**

N: バンク番号(省略時は全バンク)

本機は 1 バンク構成ですから、バンク番号を指定しても何もかわりません。

フラッシュチップの概略、各イレースブロックの開始アドレス、プロテクトの状態が一覧
出来ます。

関連： **protect**

- **fsinfo** - ファイルシステム情報の表示

文法： **fsinfo**

アクティブパーティションのファイルシステム情報を表示します。

ファイルシステムは自動で認識されます

サポートされているファイルシステムは、**cramfs** と **jffs2** です。

アクティブパーティションを変更するには、**chpart** を使います。

関連： **chpart ls fsload**

- **fsload** - フラッシュ上のファイルシステムからファイルのロード

文法： **fsload [addr] [filename]**

addr: ロードするアドレス

省略した場合、最後に **loadb,fsload,nfs,tftpboot** 等のコマンドでファイルを読み込んだアドレス。起動後始めてファイルを読み込む場合、**80400000** になります。

filename: ロードするファイル名

省略した場合、環境変数 **bootfile** の値。

bootfile が設定されていなければ、” **uImage**”

アクティブパーティションのファイルシステムからメモリ上にファイルを読み込みます。ファイルシステムは自動で認識されます。

環境変数 **filesize** に読みこんだファイルサイズが設定されます。

サポートされているファイルシステムは、**cramfs** と **jffs2** です。

関連 : **chpart ls fsinfo loadb loads nfs tftpboot**

● go - u-boot 用アプリケーションの実行

文法 : **go addr [arg ...]**

add: 実行するアドレス

arg: プログラムに渡す引数

uboot

の内部関数を利用したアプリケーションを実行する場合に使います

以下に実行例を載せます

```
# nfs 80200000 $nfsbase/u-boot/examples/hello_world.bin
```

```
Using i82559#0 device
```

```
File transfer via NFS from server 192.168.1.91; our IP address is  
192.168.1.201
```

```
Filename '/usr/src/mldbox//u-boot/examples/hello_world.bin'.
```

```
Load address: 0x80200000
```

```
Loading: #
```

```
done
```

```
28
```

```
Bytes transferred = 920 (398 hex)
```

```
# go 80200000 myip $serverip
```

```
## Starting application at 0x80200000 ...
```

```
Example expects ABI version 2
```

```
Actual U-Boot ABI version 2
```

```
Hello World
```

```
argc = 3
```

```
argv[0] = "80200000"
```

```
argv[1] = "myip"
argv[2] = "192.168.3.29"
argv[3] = "<NULL>"
Hit any key to exit ...
## Application terminated, rc = 0x0
#
関連： bootelf
```

- **help** - オンラインヘルプ

文法： **help** [command...]

command: コマンド名

command が省略された場合、全てのコマンドとその概略の一覧を表示します。

command が指定された場合、そのコマンドの説明と使い方を表示します。

2.8.24 **iminfo** - アプリケーションイメージヘッダの表示

文法： **iminfo** [addr...]

addr: アドレス

省略した場合、最後にファイルを読み込んだアドレス。

ヘッダ情報表示と、チェックサム確認を行います。

例：

```
# iminfo
## Checking Image at 80400000 ...
Image Name: Linux Kernel Image
Created: 2004-12-02 7:52:37 UTC
Image Type: MIPS Linux Kernel Image (gzip compressed)
Data Size: 764307 Bytes = 746.4 kB
Load Address: 80000000
Entry Point: 8019a040
Verifying Checksum ... OK
#
```

- **imls** - フラッシュの中にあるイメージを探す

文法： **imls**

フラッシュの全てのイレースブロックの先頭を調べて、アプリケーションイメージヘッダが

見付かる毎に、見付けた場所を表示し、**iminfo** を実行します。

- **itest** - 整数と文字列の比較テスト

文法 : **itest**[.b, .w, .l, .s] [*]**value1** <**op**> [*]**value2**

[.b, .w, .l, .s]:バイトで比較するか、ハーフワード(2 バイト)か、ワード(4 バイト)か、文字列として比較するかを指定します。省略された場合、.l と同じです。

[*]**value1**:*がある場合、**value1** はアドレスとして解釈され、比較対象はそのアドレスにある値になります。*がない場合比較対象は **value1** そのものとなります。

op :lt,<,

gt,>,

eq,==,

ne,!=<,>,<=,

ge,>=,

le,<=

のどれか

[*]**value2**:*がある場合、**value2** はアドレスとして解釈され、比較対象はそのアドレスにある値になります。*がない場合比較対象は **value2** そのものとなります。

比較を行い結果を返します。結果は表示されません。if や &&, || とともに用います。

関連 : **test**

- **loadb** - シリアル経由でファイルのダウンロード(kermit モード)

文法 : **loadb** [**addr**] [**baudrate**]

addr: ダウンロードするアドレス

省略した場合、環境変数 **loadaddr** の値、**loadaddr** が設定されていないければ、80400000 です。

baudrate: ダウンロードするボーレート

省略した場合現在のボーレートが使われます。

バイナリファイルをダウンロードします。**Ckermit**

を使ってダウンロードする場合に使いま

す。ダウンロードする間だけボーレートを変更する事もできます。

正常にダウンロードされると、ダウンロードされたファイルのサイズが環境変数 **filesize** に設定されます。その後、環境変数 **autoscript** が **yes** に設定されていると、**autoscr** を呼び出します。

関連 : **fsload loads nfs tftpboot**

- **loads** - シリアル経由で S レコード形式のファイルのダウンロード

文法 : **loads** [**offset**]

offset: S レコーで示されているアドレスに加算する値

省略した場合 0 です。

S レコード形式のファイルをダウンロードします。

環境変数 **loads_echo** を 1 に設定すると、100 行読み込む毎に '!' が表示されます。

正常にダウンロードされると、ダウンロードされたファイルのサイズが環境変数 **filesize** に設定されます。

関連 : **fsload**, **loadb nfs tftpboot**

- **loop** - 指定した範囲のアドレスを読み続ける無限ループ

文法 : **loop**[.b, .w, .l] **addr** **count**

[.b, .w, .l]: バス幅

アクセスするときに、バイト単位か(.b)、ハーフワード(2 バイト)単位(.w)か、ワード(4 バイト)単位(.l)かを指定します。省略した場合、.l になります。

addr: 開始アドレス

count: 指定した単位での数を 16 進で指定します

開始アドレスから、**count** 個までの範囲を読み続けます。

動作確認はしていません。

- **ls** - ファイルシステムの中身を一覧表示

文法 : **ls** [**name**]

name: 表示したいディレクトリもしくはファイル名

省略時は / になります。

アクティブパーティションのファイルシステム内のリストを表示します。

ファイルシステムは自動で認識されます

サポートされているファイルシステムは、**cramfs** と **jffs2** です。

アクティブパーティションを変更するには、**chpart** を使います。

関連 : **chpart** **fsinfo** **fsload**

- **md** - メモリ内容の表示

文法 : **md**[.b, .w, .l] **addr** [**count**]

[.b, .w, .l]: バス幅

アクセスするときに、バイト単位か(.b)、ハーフワード(2 バイト)単位(.w)か、ワード(4 バイト)単位(.l)かを指定します。省略した場合、.l になります。

addr: 表示するアドレス

count: 指定した単位での数を 16 進で指定します。

コマンド実行後に、改行だけを入力すると、自動的にアドレスがインクリメントされて実行されます。

例 :

```
# md.b 80400000 8
80400000: 27 05 19 56 fc 21 36 93 '..V.!6.
#
80400008: 41 ae c9 c5 00 0b a9 93 A.....
#
80400010: 80 00 00 00 80 19 a0 40 .....@
#
80400018: 69 9f c3 da 05 05 02 01 i.....
#
80400020: 4c 69 6e 75 78 20 4b 65 Linux Ke
#
```

関連 : base mm nm mw cmp cp crc32

- mm - 連続するアドレスのメモリ内容を対話的に変更

文法 : mm[.b, .w, .l] addr

[.b, .w, .l]: バス幅

アクセスするときに、バイト単位か(.b)、ハーフワード(2 バイト)単位(.w)か、ワード(4 バイト)単位(.l)かを指定します。省略した場合、.l になります。

addr: 変更するアドレス

現在の値を表示し、新しい値を入力する様に促されますので、変更したい場合新しい値を 16 進で入力します。そのまま良ければそのまま改行を入力します。

"

を入力すると、一つ前のアドレスに戻ります。

アドレスを自動的に指定単位分増やして再度聞いて来ます。

^C (Ctrl キーを押しながら C)で終了します。

例 :

```
# md.w 80400020 4
80400020: 694c 756e 2078 654b Linux Ke
# mm.w 80400020
80400020: 694c ? 494c
80400022: 756e ?
80400024: 2078 ? -
80400022: 756e ? # <INTERRUPT>
# md.w 80400020 4
80400020: 494c 756e 2078 654b LInux Ke
```

#

関連： **base md nm mw cmp cp crc32**

- **mtest** - 簡単なメモリのテスト

文法： **mtest [start [end [pattern]]]**

start: 開始アドレス

省略時 **80000000** になります。

end: 終了アドレス

省略時 **81F70000** になります。

pattern:0

パターンの値を書いて確認したあと、パターンの値をビット反転したもので確認します。

パターンの値を **1** 増やしながら、上記を永久に繰り返します。

^C (Ctrl キーを押しながら **C**)で終了します。

例：

```
# mtest 80000000 81F70000 55555555
```

```
Pattern 55555556 Writing... Reading...
```

```
#
```

- **mw** - メモリ内容を指定した値で埋める

文法： **mw[.b, .w, .l] addr val [count]**

[.b, .w, .l]: バス幅

アクセスするときに、バイト単位か(**.b**)、ハーフワード(**2 バイト**)単位(**.w**)か、ワード(**4 バイト**)単位(**.l**)かを指定します。省略した場合、**.l** になります。

addr: 変更するアドレス

val: 値

count: 指定した単位での数を **16** 進で指定します。

省略した場合、**1** とみなされます。

関連： **base md nm mw cmp cp crc32**

- **nfs** - NFS プロトコルでファイルをダウンロード

文法： **nfs [addr] [[ip:]filename]**

addr: ロードするアドレスです。

省略した場合、環境変数 **loadaddr** の値、**loadaddr** も設定されていなければ、**80400000** です。

ip: **nfs** サーバの IP アドレスです。

省略した場合、環境変数 **serverip** の値が使われます。

filename: ロードするファイルです。

省略した場合は、環境変数 **bootfile** の値、環境変数 **bootfile** が設定されてい

ない場合、/nfsroot/<自分の IP アドレス>.img です。

<自分の IP アドレス>の部分は、IP アドレスを 16 進で表記して、並べたものです。例えば、192.168.3.202 を 16 進で表記すると、C0.A8.03.CA となりますので、並べると、CA03A8C0 となりますので、” /nfsroot/C0A803CA.img”を探しに行きます。addr を省略して、filename を指定する場合、filename は" で括られていなくてははいけませんが、パーサが" をはずしてしまうため、コマンドに" が渡るように、エスケープする必要があります。ファイルのダウンロードが正常に終了すると、環境変数 fileaddr にロードしたアドレスが設定され、環境変数 filesize にダウンロードしたファイルのサイズが設定されます。その後、環境変数 autostart が yes に設定されていると、続けて bootm を呼び出し、環境変数 autoscript が yes に設定されていると、さらに続けて autoscr を呼び出します。

関連： fsload loadb loads tftpboot

- nm - 同一アドレスのメモリ内容を対話的に変更

文法： nm[.b, .w, .l] addr

[.b, .w, .l]: バス幅

アクセスするときに、バイト単位か(.b)、ハーフワード(2 バイト)単位(.w)か、ワード(4 バイト)単位(.l)かを指定します。省略した場合、.l になります。

addr: 変更するアドレス

現在の値を表示し、新しい値を入力する様に促されますので、新しい値を入力します。

リターンだけの入力、書き込みしません。

^C (Ctrl キーを押しながら C)で終了します。

GPIO のレジスタや IO ポートを操作する際には便利です。

2.8.37 pci - PCI バスの一覧と、PCI コンフィグレーションスペースのアクセス

文法： pci [bus] [long]

bus: バス番号を指定します。省略時は 0

long: 詳細を出したいときには、long を指定します。

指定されなかった場合は、簡単な表示になります。

PCI バスをリストします。

文法： pci header <b.d.f>

b.d.f: PCI のデバイスを指定します。

b,d,f はそれぞれ、バス、デバイス、ファンクションの番号を指定します。

指定されたデバイスの詳細(PCI コンフィグレーションスペースのヘッダ情報)が出力されます。

文法： pci display[.b, .w, .l] b.d.f [addr] [count]

[.b, .w, .l]: バス幅

アクセスするときに、バイト単位か(.b)、ハーフワード(2 バイト)単位(.w)か、ワード(4

バイト)単位(.l)かを指定します。省略した場合、.lになります。

b.d.f: PCI のデバイスを指定します。

b,d,fはそれぞれ、バス.デバイス.ファンクションの番号を指定します。

addr: 表示するアドレス

count: 指定した単位での数を 16 進で指定します

PCI コンフィグレーションスペースの内容を表示します。

文法 : **pci modify**[.b, .w, .l] **b.d.f addr**

[.b, .w, .l]: バス幅

アクセスするときに、バイト単位か(.b)、ハーフワード(2 バイト)単位(.w)か、ワード(4 バイト)単位(.l)かを指定します。省略した場合、.lになります。

addr: 変更するアドレス

mm コマンドの **PCI** 用です。現在の値を表示し、新しい値を入力する様に促されますので、変更したい場合新しい値を 16 進で入力します。そのままであればそのまま改行を入力します。

"

を入力すると、一つ前のアドレスに戻ります。

アドレスを自動的に指定単位分増やして再度聞いて来ます。

^C (Ctrl キーを押しながら C)で終了します。

文法 : **pci next**[.b, .w, .l] **b.d.f addr**

[.b, .w, .l]: バス幅

アクセスするときに、バイト単位か(.b)、ハーフワード(2 バイト)単位(.w)か、ワード(4 バイト)単位(.l)かを指定します。省略した場合、.lになります。

b.d.f: PCI のデバイスを指定します。

b,d,fはそれぞれ、バス.デバイス.ファンクションの番号を指定します。

addr: 変更するアドレス

nm コマンドの **PCI** 用です。現在の値を表示し、新しい値を入力する様に促されますので、新しい値を入力します。リターンだけの入力の場合は、書き込みしません。

^C (Ctrl キーを押しながら C)で終了します。

文法 : **pci write**[.b, .w, .l] **b.d.f addr value**

[.b, .w, .l]: バス幅

アクセスするときに、バイト単位か(.b)、ハーフワード(2 バイト)単位(.w)か、ワード(4 バイト)単位(.l)かを指定します。省略した場合、.lになります。

b.d.f: PCI のデバイスを指定します。

b,d,fはそれぞれ、バス.デバイス.ファンクションの番号を指定します。

addr: 変更するアドレス

value: 書き込む値

long header display modify next write はそれぞれ、l h d m n w と略する事も可能です。

- **ping** - ICMP ECHO_REQUEST パケットを指定ホストに送る

文法 : **ping ip**

ip: IP アドレス。

指定したアドレスに 1 回 **ping** パケットを送出し、返事を待ちます。

- **printenv** - 環境変数の一覧と内容表示

文法 : **printenv [name...]**

name: 環境変数の名前

name が指定されなかった場合、全ての環境変数を表示します。

name が指定された場合その環境変数だけを表示します。

- **protect** - フラッシュメモリのプロテクトの設定

文法 : **protect op start end**

protect op N:SF[SL]

protect op bank N

protect op all

op: on もしくは off

start: 開始アドレス(最初のイレースブロックの先頭アドレス)

end: 終了アドレス(最後のイレースブロックの最終アドレス)

N : バンク番号(常に 1 を指定します)

SF: イレースブロック番号

SL: フラッシュ内のイレースブロックを先頭から数えた番号(0 から数え始めます)

省略した場合、**SF** と同じ値になります。

指定された範囲内で、プロテクトの状態を変更します。

イレースブロック番号は、フラッシュ内のイレースブロックを先頭から数えた番号です。
(0 から数え始めます)

関連 : **erase**

- **rarpboot**- RARP プロトコルで IPv4 アドレスを取得

文法 : **rarpboot [addr] [filename]**

addr: ロードするアドレスです。

省略された場合、環境変数 **loadaddr** の値、**loadaddr** も設定されていなければ、**80400000** です。

filename : ロードするファイルです。

addr を省略して、**filename** を指定する場合、**filename** は" で括られていなくてもはいけませんが、パーサが" をはずしてしまうため、**rarpboot** コマンドに" が渡るように、エスケープする必要があります。

RARP プロトコルで **IPv4** アドレスを取得し、環境変数 **ipaddr** を更新します。

また、環境変数 **serverip** が設定されていない場合、**serverip** を **RARP** サーバの **IP** アドレスで更新します。

次に、環境変数の **autoload** をチェックして、

autoload の値が **n** で始まっていたら終了します。

autoload の値が **NFS** の場合、**nfs** コマンドを同じ引数で呼び出します。

autoload が上記以外の場合は **tftpboot** コマンドを同じ引数で呼び出します。

このコマンドの動作は未確認です。

関連 : **bootp dhcp nfs tftpboot**

- **reset** - CPU のリセット

文法 : **reset**

再起動します。

37

2.8.43 **run** - 環境変数に設定されている文字列の実行

文法 : **run name [...]**

name: 環境変数の名称

環境変数に設定されている文字列を実行します

run a b は、**run a && run b** と同等です。

- **saveenv**- 現在の環境変数の値を全てフラッシュに保存

文法 : **saveenv**

現在の環境変数の値を全てフラッシュに保存します。

自動的に生成される環境変数も保存されてしまいます。特に **dhcp** 等をお使いの際は割り振られたアドレスもセーブしてしまいますので注意して下さい。

- **setenv** - 環境変数の設定と削除

文法 : **setenv name value**

name: 環境変数の名称

value: 設定する文字列

value を指定しなかった場合、その環境変数を削除します。

value が指定された場合、その文字列に設定します。

- **sleep** - 指定した秒数遅延させる

文法 : **sleep** N

N: 遅延させる秒数(10進で指定します)

2.8.47 **test** - シェルライクな **test** の最小限の実装

o,

a,

z,

n,

=, !=, >, <, eq,

ne,

lt,

le,

gt,

ge

が使えます。

数値は、10進数として扱われます。

if **&&** **||** 等とともに利用します。

補足 :

Hush パーサでは**n**

z

は事実上使えないので、

if test x\$env = x; then echo env not set; fi の様に使って下さい。

- **tftpboot** - TFTP プロトコルでファイルをダウンロード

文法 : **tftpboot** [addr] [filename]

addr: ロードするアドレスです。

省略された場合、環境変数 **loadaddr** の値、**loadaddr** も設定されていなければ、**80400000** です。

filename : ロードするファイルです。

省略された場合は、環境変数 **bootfile** の値、環境変数 **bootfile** が設定されていない場合、<自分の IP アドレス>.img を使います。

<自分の IP アドレス>の部分は、IP アドレスを 16 進で表記して、並べたものです。例えば、**192.168.3.202** を 16 進で表記すると、**C0.A8.03.CA** となりますので、” **CA03A8C0.img**”を探しに行きます。

addr を省略して、**filename** を指定する場合、**filename** は" で括られていなくて

はいけませんが、パーサが" をはずしてしまうため、コマンドに" が渡るように、エスケープする必要があります。

ファイルのダウンロードが正常に終了すると、環境変数 **fileaddr** にロードしたアドレスが設定され、環境変数 **filesize** にダウンロードしたファイルのサイズが設定されます。その後、環境変数 **autostart** が **yes** に設定されていると、続けて **bootm** を呼び出し、環境変数 **autoscript** が **yes** に設定されていると、さらに続けて **autoscr** を呼び出します。

関連： **fsload loadb loads nfs**

- **version - u-boot** のバージョンの表示

文法： **version**

バージョンと、ビルドした日付と時刻が表示されます。

u-boot で特殊な意味を持つ環境変数の一覧

- **IFS**

値： 文字列

初期値： なし

Hush パーサのトークンのセパレータです。

設定していなければ、スペース、タブ、改行になります。

設定しないで下さい。

- **autoload**

値： **n** | **NFS**

初期値： **no**

bootp, **dhcp**, **rarpboot** コマンドで IP アドレスを取得したあと、自動的にファイルをダウンロードするかどうかを決めます。

n で始まる文字列であれば何もしません。

NFS なら **nfs** コマンドを実行します。

上記以外もしくは、設定されていない場合は、**tftpboot** コマンドを実行します。

- **autoscript**

値： **yes**

初期値： なし

loadb, **nfs**, **tftpboot** でファイルをダウンロードしたあと、自動的に **autoscr** を呼び出すかどうかを決めます。

yes 以外、もしくは設定されていない場合は、呼び出されません。

- **autostart**

値： **yes**

初期値： なし

loadb, nfs, tftpbboot でファイルをダウンロードしたあと、自動的に **bootm** を呼び出すかどうかを決めます。

yes 以外、もしくは設定されていなければ、呼び出されません。

- **baudrate**

値： **9600 | 19200 | 38400 | 57600 | 115200**

初期値： **15200**

コンソールのボーレートを設定します。

変更するとすぐに反映されます。

- **bootaddr**

値： **16** 進の値

初期値： なし

VxWorks を起動する際のブートアドレス。

- **bootargs**

値： 任意の文字列

初期値： **root=/dev/mtdblock1** などはプログラムソースで変更可能です。

OS に渡す起動パラメータを設定します。

- **bootcmd**

値： 任意の文字列

初期値： **run tryboot**

デフォルトのカーネル起動方法を設定します。

(**boot** コマンドや、自動起動に設定されている時に参照されます)

- **bootdelay**

値：

もしくは、負でない整数(10 進数)

初期値： **5**

電源投入時にデフォルトのコマンドを実行するまでの待ち時間を設定します。

1

を指定すると、自動起動しません。

- **bootfile**

値： 任意の文字列

初期値： なし

起動に使うデフォルトのファイルを指定します。自動的に更新される場合がありますので、フラッシュへ保存する際には注意が必要です。

- **dnsip**

値： IP アドレス

初期値： なし

bootp,dhcp で DNS サーバの IP アドレスがもらえた場合に設定されます。

この変数を参照するコマンドはありません。

- **dnsip2**

値： IP アドレス

初期値： なし

bootp,dhcp で DNS サーバの二つ目の IP アドレスがもらえた場合に設定されます。

この変数を参照するコマンドはありません。

- **domain**

値： 文字列

初期値： なし

bootp,dhcp でドメイン名がもらえた場合に設定されます。

この変数を参照するコマンドはありません。

- **ethact**

値： **i82559#0** | **i82559#1**

初期値： **i82559#0**

ネットワークを利用するコマンドが使用するデバイスを指定します。

i82559#0 は本体に **1** と書かれた方を使い、**i82559#1** は本体に **2** と書かれた方を使います。

環境変数 **netretry** が **once** に設定されている場合、この変数に指定されたデバイスでアクセスに失敗するとこの変数は次のデバイスに書き換えられ、再試行されます。

成功した場合はこの変数には成功したデバイス名が保持されています。

- **ethaddr**

値： MAC アドレス

初期値： なし

本機 1 番のポートのイーサネットデバイスを初期化する際に指定された **MAC** アドレスを使うようにします。

設定した場合の動作は未確認です。

- **eth1addr**

値： **MAC** アドレス

初期値： なし

本機 2 番のポートのイーサネットデバイスを初期化する際に指定された **MAC** アドレスを使うようにします。

設定した場合の動作は未確認です。

- **ethprime**

値： **i82559#0** | **i82559#1**

初期値： なし

ネットワークを利用するコマンドが優先的に使用するデバイスを指定します。

電源投入時、環境変数 **ethact** の値を **ethprime** の値で初期化します。

fileaddr

値： **16** 進の値

初期値： なし

nfs,tftpboot コマンドが、ファイルをダウンロードしたメモリ上のアドレスを設定します。
この変数を参照するコマンドはありません。

- **filesize**

値： **16** 進の値

初期値： なし

fsload,loadb,loads,nfs,tftpboot コマンドが、ファイルをダウンロードした時にダウンロードしたファイルサイズを設定します。

この変数を参照するコマンドはありません。

- **gatewayip**

値： **IP** アドレス

初期値： なし

デフォルトゲートウェイの **IP** アドレスを設定します。

bootp,dhcp でゲートウェイの **IP** アドレスがもたらえた場合は上書きされます。

- **hostname**

値： 文字列

初期値： なし

dhcp で IP アドレスを要求する時に、現在の値を送信し、**DHCP** の応答で得られた値で更新されます。

2.9.22 ipaddr

値： **IP** アドレス

初期値： なし

本機の **IP** アドレスを設定します。

bootp,dhcp,rarpboot コマンドで **IP** アドレスが取得できた場合は上書きされます。

- **loadaddr**

値： **16** 進の値

初期値： なし

ファイルをダウンロードする際のデフォルトのアドレスを指定します。

2.9.24 loads_echo

値： **1**

初期値： なし

1 に設定すると、**loads** コマンドで、**100** 行読み込む毎に **!** が表示されます。

cu コマンドで **S** レコードファイルをダウンロードするには、**1** に設定する必要があるでしょう。

- **netmask**

値： **IP** アドレス

初期値： なし

bootp,dhcp でネットマスクがもらえた場合に設定されます。

この変数を参照するコマンドはありません。

- **netretry**

値： **no** | **once**

初期値： **once**

no が設定されていると、**bootp,dhcp** が失敗したときに、デバイスを切り替えて再試行しません。**once** が設定されていると、**bootp,dhcp** が失敗したときに、**1** 回だけ試します。

- **nvlan**

native VLAN の略です。

- **preboot**

値： 任意の文字

初期値： "echo;" \

"echo Type \"boot\" for boot normal way;" \

"echo Type \"run cram\" to boot with root filesystem on mtdblock1" \

"echo Type \"run usb\" to boot with root filesystem on USB;" \

"echo Type \"run nfs\" to boot with NFS root filesystem;" \

"echo"

自動起動の前に実行するコマンドを設定します。

- **rootpath**

値： 任意の文字

初期値： なし

bootp,dhcp で **rootpath** がもらえた場合には上書きされます。

この変数を参照するコマンドはありません。

- **serial#**

値： 任意の文字

初期値： なし

製品のシリアルナンバーを設定します。

一度設定すると、変更できなくなります。

この変数を参照するコマンドはありません。

- **serverip**

値： IP アドレス

初期値： なし

tftp サーバや、**nfs** のデフォルトサーバを設定します。

rarpboot で設定されたり、**bootp,dhcp** で上書きされたりします。

- **stdin**

値： デバイス名

初期値： **serial**

標準入力に使うデバイスを設定します。

- **stdout**

値： デバイス名

初期値： **serial**

標準出力に使うデバイスを設定します。

- **stderr**

値： デバイス名

初期値： **serial**

標準エラー出力に使うデバイスを設定します。

- **verify**

値： **n**

初期値： **no**

autoscr, **bootm** を実行する時、チェックサムを検査するかどうかを指定します。

- **vlan**

値： **4095** 未満の正の整数

初期値： 未設定

802.1q の **VLAN** タグを設定します。

最後に

U-Boot のような非常に移植性の優れたツールでも実際にインプリしようとするれば時間がかかるものです。ましてや初めて行うには工数なども読めないことが普通です。このような場合にメーリングリストに入るのは非常に有効な方法であり活用すべきことはいうまでもありません。**U-boot** には英語によるオリジナルのものから日本語での **uboot-jp** のものまであります。またコンサルティングサービスも下記の一部の会社でおこなっているので利用することも可能です。

この記事の執筆に協力いただいた方々

uboot-jp プロジェクトの方々

uboot-jp メーリングリストの方々

(以下 5 0 音順) 株式会社アイウェア <http://www.aiware.co.jp/> NEC エレクトロニクス株式会社 http://www.necel.com/index_j.html 株式会社メディアラボ <http://www.mlb.co.jp/> 株式会社タンバック <http://www.tanbac.co.jp/>

関連サイト

ppcboot Project <http://ppcboot.sourceforge.net/>

Das U-Boot" - the Universal Boot Loader <http://sourceforge.net/projects/u-boot/>
uboot-jp Project <http://www.uboot-jp.org/> mirror <http://ntools.net/pub/uboot-jp/> コンパ
イル済みバイナリも入手可能