

目次

NuttShell コマンド 1

NuttShell commands 1

date 1

dmesg 1

free 2

ls 2

ps 3

reboot 3

Builtin Applications 3

cu 3

i2c 4

NuttShell コマンド

[ファームウェアの書き込みと動作](#) で書き込みしたファームウェアには、いくつかのコマンドが用意されています。

```
nsh> help
help usage:  help [-v] [<cmd>]

[      dirname  false  mkfatfs  pwd      time
?      date     free   mkfifo   reboot   true
basename dd      help   mkrd     rm       uname
break   df      hexdump mh       rmdir    umount
cat      dmesg  kill   mount    set      unset
cd       echo   ls     mv       sh       usleep
cp       exec  mb     mw       sleep    xd
cmp      exit  mkdir  ps       test

Builtin Apps:
  cu
  i2c
  sudoku
nsh>
```

このうち、デバッグなどでもよく使用するコマンドについて紹介します。

NuttShell commands

date

RTC への日時設定や、現在日時の表示を行います。

```
nsh> help date
date usage:  date [-s "MMM DD HH:MM:SS YYYY"]
nsh> date -s "Mar 08 16:32:20 2018"
nsh> date
Thu, Mar 08 16:32:21 2018
nsh>
```

dmesg

[ramlog](#) に記録された syslog を表示します。

実行して出力すると、出力された分は RAM から消去されます。

```
nsh> dmesg
[    0.000000] stm32l4_rng_initialize: Initializing RNG
[    0.000000] registered UART4 as /dev/console.
[    0.000000] registered UART4 as /dev/ttyS0.
[    0.000000] registered USART2 as /dev/ttyS1.
[    0.000000] registered USART3 as /dev/ttyS2.
[    0.001000] Mounting procfs to /proc

nsh>
```

free

メモリの使用状況を表示します。

```
nsh> free
              total        used        free      largest
Umem:         105088         14576         90512         69744

nsh>
```

ls

ファイルのリストを表示します。

```
nsh> ls
/:
dev/
proc/
nsh> ls /proc
/proc:
0/
1/
2/
meminfo
fs/
mtd
uptime
nsh> ls /dev
/dev:
console
i2c1
mtdblock0
```

```

null
ptmx
ramlog
random
rtc0
timer0
ttyS0
ttyS1
ttyS2
zero
nsh>

```

ps

動作しているタスク、スレッドの状況を表示します。

```

nsh> ps
  PID GROUP  PRI POLICY   TYPE      NPX STATE   EVENT          SIGMASK  STACK
COMMAND
   0    0    0  FIFO    Kthread N-- Ready          00000000 000000 Idle
Task
   1    1   50  FIFO    Kthread --- Waiting  Signal      00000000 002028
lpwork
   2    1  100  FIFO    Task     --- Running          00000000 004076 init
   5    5  200  FIFO    Task     --- Waiting  MQ empty    00000000 004068
bp35_task
   7    7  100  FIFO    Task     --- Waiting  MQ empty    00000000 004068
sensor_task
   9    9   50  FIFO    Task     --- Waiting  MQ empty    00000000 004076 main
nsh>

```

reboot

再起動します。

Builtin Applications

cu

シリアル端末です。

i2c

I2C バスのスキャンやデバイスの読み書きを行います。

```
nsh> i2c
Usage: i2c <cmd> [arguments]
Where <cmd> is one of:

  Show help      : ?
  List busses    : bus
  List devices   : dev [OPTIONS] <first> <last>
  Read register  : get [OPTIONS] [<repetitions>]
  Show help      : help
  Write register: set [OPTIONS] <value> [<repetitions>]
  Verify access  : verf [OPTIONS] [<value>] [<repetitions>]

Where common "sticky" OPTIONS include:
  [-a addr] is the I2C device address (hex).  Default: 03 Current: 03
  [-b bus] is the I2C bus number (decimal).  Default: 0 Current: 0
  [-r regaddr] is the I2C device register address (hex).  Default: 00
Current: 00
  [-w width] is the data width (8 or 16 decimal).  Default: 8 Current: 8
  [-s|n], send/don't send start between command and data.  Default: -n
Current: -n
  [-i|j], Auto increment|don't increment regaddr on repetitions.  Default:
NO Current: NO
  [-f freq] I2C frequency.  Default: 400000 Current: 400000

NOTES:
o An environment variable like $PATH may be used for any argument.
o Arguments are "sticky".  For example, once the I2C address is
  specified, that address will be re-used until it is changed.

WARNING:
o The I2C dev command may have bad side effects on your I2C devices.
  Use only at your own risk.
nsh>
```

```
nsh> i2c dev -b 1 0 7f
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: 20 -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- 51 -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- --
70: -- -- -- -- -- -- 76 -- -- -- -- -- -- -- --
nsh>
```

From:

<https://centurysys.jp/> - **MA-X/MA-S/MA-E/IP-K Developers' WiKi**

Permanent link:

https://centurysys.jp/doku.php?id=xg_series_devel:nsh_commands:start

Last update: **2018/03/08 17:04**