# 付録．Ａ　ＫＩＴ－Ｖ３０ＭＺ－ＴＰ　内部コマンド

本書は、ＫＩＴ－Ｖ３０ＭＺ－ＴＰの内部コマンドについて記述しています。これらのコマンドは、ディバッガの中でスルーコマンドとして使用できます。スルーコマンドの使用方法はディバッガのマニュアルを参照ください。

（例）ＰＡＲＴＮＥＲ／Ｗｉｎの場合

```
＞＆          ＜＜　スルーコマンドへ移行します。
＞＃ＥＮＶ    ＜＜　内部コマンドの入力です。
＞＆          ＜＜　スルーコマンドモードを終了します。
```

## コマンド一覧



ご注意：これらのコマンドは、ご使用になりたい機能がディバッガ本体に有していない場合にのみ補助的にご使用下さい。ご使用になるディバッガで同等の機能を有している場合にこれらのコマンドを発行した場合、ディバッガとの間で競合をおこし、いずれかの動作が異常になる場合があります。

## コマンド書式

ＫＩＴ－Ｖ３０ＭＺ－ＴＰの内部コマンドの基本書式を以下に示します。

コマンド名　パラメータ

＊パラメータ書式で［］は省略可能を示し、｜　は択一を意味します。

コマンド名はアルファベットの文字列でパラメータとの間はスペースまたはタブで区切ります。パラメータはアルファベットの文字列または１６進数を指定し、各パラメータ間はスペースまたはタブで区切ります。（１６進数には演算子は使用できません。）

## ｅｎｖコマンド

[書式]

env [[!]auto] [[!]reset] [[!]int] [[!]nmi] [[!]hldrq] [[!]poll]
[[!]fready] [jtag12|jtag25] [clk|clk2]

[パラメータ]

[!]auto
実行中にブレークポイントを設定した場合一時的にブレークしますが、その後の実行を自動的に行う場合に[Auto],行わない場合に[!auto]を指定します。

[!]reset
RESET端子のマスク指定を指定します。!はマスクしないを意味します。

[!]int
INT端子のマスク指定を指定します。!はマスクしないを意味します。

[!]nmi
NMI端子のマスク指定を指定します。!はマスクしないを意味します。

[!]hldrq
HLDRQ端子のマスク指定を指定します。!はマスクしないを意味します。

[!]poll
POLL端子のマスク指定を指定します。!はマスクしないを意味します。

[!]fready
バスサイクル中、一定時間たってもREADYが返されない場合のタイムアウトを指定します。!はタイムアウトしないを意味します。

jtag[12|25]
N-WireのJTAGクロック指定します。通常、jtag25で使用ください。

clk|clk2
トレースクロックの指定です。通常、clk(sama clk)で使用ください。

[機能]

envコマンドは、エミュレーション環境の設定を行います。設定は変更が必要なパラメータだけを入力ください。入力の順序は任意です。但し、同じパラメータを２回入力した場合は、後から入力した値が有効です。初期値は、以下の通りです。

```
CPU Settings:
 Auto Run     = ON (auto)
 JTAGCLOCK    = 25MHz (jtag25)
Signals Mask:
 NMI          = NO MASK (!nmi)
 RESET        = NO MASK (!reset)
 HLDRQ        = NO MASK (!hldrq)
 INT          = NO MASK (!int)
 POLL         = NO MASK (!poll)
 ForceREADY   = Enable  (fready)
Trace Settings:
 Trace Clock  = Same Clock of CPU CLOCK (clk)
```

＜入力例＞

env reset !nmi

RESETをマスクし、NMIをマスクしません。

## ｅｖ１，ｅｖ２コマンド

[書式]

ev{1|2} [ADDR [AMASK [DATA [DMASK]]]] [byte|word] [exe|data]
[intack|ioread|iowrite|ioacc|memread|memwrite|memacc|all] [noext|nega|posi]

[パラメータ]

ADDR: アドレス値を１６進数で指定します。
AMASK: アドレスのマスク値を１６進数で指定します。１でマスクします。
DATA: データ値を１６進数で指定します。
DMASK: データのマスク値を１６進数で指定します。１でマスクします。
byte|word: データのサイズを指定します。
byte: DATA,DMASKを8ビットの値として取り扱います。
word: DATA,DMASKを16ビットの値として取り扱います。
exe|data: 検出サイクルの種類を指定します。
exe: 実行アドレスを検出します。ADDR,AMASKのみが有効です。
data: バスサイクルを検出します。
intack|ioread|iowrite|ioacc|memread|memwrite|memacc|all:バスサイクルを指定します。
intack: intackサイクル(ADDR,DATAは、無視します)
ioread: io readサイクル
iowrite: io writeサイクル
ioacc: io readとio writeの両方のサイクル
memread: memory readサイクル（命令フェッチは除く）
memwrite:memory writeサイクル
memacc: memory readとwriteの両方のサイクル（命令フェッチは除く）
all: 全てのサイクル（命令フェッチを含む）
noext|nega|posi:外部入力信号(EXI0)のエッジの検出を指定します。
noext: エッジの検出を無効にします。
nega: 立ち下がりエッジの検出
posi: 立ち上がりエッジの検出

[機能]

イベントの設定を行います。イベントは、以下の用途に使用できます。
トレースのトリガ
トレースのアクセスサイクルの取り込み条件の指定
ブレークの条件

[使用例]

ev1 80000 0 5555 0 data word memread
0x80000から0x5555をリードしたサイクルを検出します。下線の0は、マスクをしない指定のために入力しています。トレースのトリガ検出等で指定すると有効です。

ev1 0 fffff 0 ffff data memacc
memoryへのアクセスサイクルを全て検出します。トレースのアクセスの取り込み条件の指定等に使用すると有効です。

ev1 fe000 0 exe
fe000番地の命令実行を検出します。トレースのトリガ検出等で指定すると有効です。

[備考]

アドレスに奇数番地を指定した場合のワード指定はできません。

## ｈｅｌｐコマンド

[書式]
help [command]

[パラメータ]
command: コマンド名を指定します。
コマンド名を省略した場合、コマンドの一覧が表示されます。

[機能]
各コマンドのヘルプメッセージを表示します。

[使用例]
help map
mapコマンドのヘルプを表示します。

## ｉｎｉｔコマンド

［書式］
　　init

［パラメータ］
　　なし

［機能］
　　ＲＴＥ－１００－ＴＰを初期化します。全ての環境設定値は初期化されます。
　　メモリキャッシュの除外エリアは初期化されません。

## ｎｃコマンド

[書式]

nc [[ADDR [LENGTH]]

[パラメータ]

| | |
|---|---|
| ADDR: | メモリキャッシュの除外エリアの開始アドレスを指定します。 |
| LENGTH: | メモリキャッシュの除外エリアのバイト数を指定します。<br>デフォルト値３２バイト、最少値３２バイト |

[機能]

ＫＩＴ－Ｖ３０ＭＺ―ＴＰではメモリ参照の高速化を図るため、８ブロック＊３２バイトのメモリリードキャッシュを持っています。同一アドレスのメモリ参照などは実際にはメモリをリードしません。メモリにＩ／Ｏを割り付けている場合は、このキャッシュ機能は実際の動作と矛盾してしまいますので、このコマンドでメモリキャッシュの除外エリアを指定してください。メモリキャッシュの除外エリアは最大８ブロック指定でき、最少のブロックサイズは３２バイトです。

[使用例]

nc 10000 1000

１００００番地から１０００バイトの領域をメモリキャッシュの除外エリアに指定します。

```
>nc 10000 1000
 No Memory Cache Area
 No. Address  Length
 1  010000   001000
 2  fff000   001000
```

## ｎｃｄコマンド

[書式]

ncd ブロック番号

[パラメータ]

ブロック番号: 削除するメモリキャッシュの除外エリアのブロック番号を指定します。

[機能]

メモリキャッシュの除外エリアを削除します。削除は各メモリキャッシュの除外エリアのブロック番号を指定します。

[使用例]

ncd 2

ブロック番号２をメモリキャッシュの除外エリアから削除します。

```
>nc
 No Memory Cache Area
 No. Address  Length
 1  020000   000100
 2  010000   001000

>ncd 2
 No Memory Cache Area
 No. Address  Length
 1  020000   000100
```

## ｒｅｓｅｔコマンド

［書式］
reset

［パラメータ］
なし

［機能］
ＲＴＥ－１００－ＴＰのエミュレーションＣＰＵをリセットします。

## ｒomコマンド

[書式]

rom [ADDRESS [LENGTH]] [512k|1m|2m|4m|8m|16m] [rom8|rom16] [bus8|bus16]

[パラメータ]

ADDRESS [LENGTH]

ADDRESS: エミュレートするROMの最下位のアドレス（ROMのバウンダリに合致していない場合、エラーになります。）

LENGTH: エミュレートするROMのバイト数（4バイトの境界単位で指定）

512k|1m|2m|4m|8m|16m

エミュレートするROMのBitサイズを指定します。512K-bitから16M-bitまでのサイズが指定できます。例えば、27C1024の場合は、1Mを指定します。

rom8|rom16

エミュレートするROMのデータビット数を指定します。8bitと16bitが指定できます。DIP32-ROMケーブルを使用する場合はrom8、DIP-40/42-ROMケーブルを使用する場合は、rom16を指定します。

bus8|bus16

エミュレートするシステムの中でのROMのバスサイズを指定します。
8bitと16bitが指定できます。

[機能]

ROMのエミュレーション環境の設定を行います。設定は変更が必要なパラメータだけを入力ください。入力の順序は任意です。但し、同じパラメータを２回入力した場合は、後から入力した値が有効です。初期値は、LENGTH = 0で使用しないになっています。

[入力例]

rom C0000 40000 1m rom16 bus16

27C1024(1M-bitの16bit-ROM)を0xc0000から256Kバイト(40000)エミュレートします。
この場合、結果的に16bit-romを２個エミュレートします。

rom 80000 80000 1m rom rom8 bus16

27C010(1M-bitの8bit-ROM)を0x80000から512Kバイト(80000)エミュレートします。
この場合、結果的に8bit-ROMを４個エミュレートします。

＜備考＞

ｒomコマンドで指定した範囲へのアクセス方法は、実行中とブレーク中で異なります。
ブレーク中：内部のエミュレションメモリに対し直接アクセスします。
実行中　：プロセッサのバスを介してアクセスします。但し、実行中、ROMエミュレーション領域への書き込みは行えません。必ず、ブレーク中に行ってください。

## ｓｙｍｆｉｌｅ，ｓｙｍコマンド

[書式]

| | |
|---|---|
| symfile FILENAME | >>elfファイル(.elf)の読み込みを行います |
| sym [NAME] | >>シンボルの表示（３０個）を行います |

[パラメータ]

| | |
|---|---|
| symfile: | ﾌｧｲﾙ名 |
| sym: | シンボルの先頭文字列 |

[機能]

symfile コマンドは、FILENAMEで指定したelfファイルからシンボルを読み込みます。対象となるのはグローバルシンボルだけです。また、読み込んだシンボルの表示（最大３０個）もできます。

[使用例]

```
symfile c:\test\dry\dry.elf
```

c:\test\dryのディレクトリからelfファイル:dry.elfのシンボルを読み込みます。

```
sym m
```

mから始まるシンボルを最大30個表示します。

## ｔｒｏｎ，ｔｍｏｄｅコマンド

[書式]

tron [DELAY] [[!]delay] [[!]startev1] [[!]ev1trg] [[!]ev2trg] [noext|nega|posi]
[dt1off|dt1eq|dt1ty|dt1all][dt2off|dt2eq|dt2ty|dt2all][pcoff|pcjmp|pcal

tmode :設定状態の表示コマンド

[パラメータ]

DELAY = 0..1ffff ディレイカウンタ
トリガ成立後にメモリの取り込むフレーム数を十進数で指定します。

[!]delay : 強制ディレイモードを指定します。!で通常のモードの指定に戻ります。
強制ディレイモードでは、トレース開始後、ディレイカウンタ数分のトレースをした時点で強制的にトレースを終了するモードです。
このモード中は、トリガイベントは無視されます。

[!]startev1: ev1をトレースの取り込みの開始条件に指定します。!で無条件に開始します。

[!]ev1trg: ev1をトレーストリガ条件に指定します。!で無効です。

[!]ev2trg: ev2をトレーストリガ条件に指定します。!で無効です。

noext|nega|posi: 外部入力端子(exi0)をトリガ条件に指定します。
- noext: 無効
- nega: 立ち下がりエッジを検出
- posi: 立ち上がりエッジを検出

real|noreal|sample:トレースの取り込みのモードを指定します。
- real: リアルタイム実行中にトレースします。V30MZ内のトレースバッファがフルになった場合は、トレース情報の取りこぼしが発生します。
- noreal: V30MZ内のトレースバッファがフルになった場合、CPUの実行を一時的に停止させてトレース情報の取りこぼしがないようにするモードです。
- sample: 256 CPUCLKに１回PCの値を出力するモードです。

pcoff|pcjmp|pcall:実行アドレスに関するトレース取り込み情報を指定します。
- pcoff: 実行アドレスに関する情報はトレースしません。
- pcjmp: PCのみトレースします。
この場合、実行過程のトレースは正しく表示できません。
- pcall: 全ての情報をトレースします。

dt1off|dt1eq|dt1ty|dt1all:ev1にマッチした時に取り込むトレース情報を指定します。
- dt1off: トレースしません。
- dt1eq: ev1を検出したという情報のみをトレースします。
- dt1ty: ev1を検出し、サイクルの種類とデータをトレースします。
- dt1all: ev1を検出し、アドレス、データ、サイクルの全てをトレースします。

dt2off|dt2eq|dt2ty|dt2all:ev1にマッチした時に取り込むトレース情報を指定します。
- dt2off: トレースしません。
- dt2eq: ev2を検出したという情報のみをトレースします。
- dt2ty: ev2を検出し、サイクルの種類とデータをトレースします。
- dt2all: ev2を検出し、アドレス、データ、サイクルの全てをトレースします。

[機能]

トレースの諸設定とトレースバッファをクリアしトレースの取り込みを開始します。

[使用例]

delayモードで無条件に1ffffサイクル分トレースします。

```
rte>tron delay 1ffff
Trace Settings:
 Delay Count  = 01ffff
 Trace Mode   = Real Time (real)
 PC Mode      = Jump and Execution (pcall)
 Delay Mode   = Enable (delay)
 Data Trace 1 = ALL Trace (dt1all)
 Data Trace 2 = ALL Trace (dt2all)
 Start EV1    = Disable (!startev1)
 EV1 Trigger  = Disable (!ev1trg)
 EV2 Trigger  = Disable (!ev2trg)
 Ext Trigger  = Disable (noext)
     Addr  Mask  Data Mask Kind Status   Size Aux
EV1: -     -     -    -    -    -        -    -
EV2: -     -     -    -    -    -        -    -
```

memory cycleの全てを取り込みながら、0x100番地へのiowriteでトリガをかけ、トリガ後ffffサイクル分トレースします。

```
rte>ev1 data memacc                              << ev1の設定
     Addr  Mask  Data Mask Kind Status   Size Aux
EV1: 00000 fffff 0000 ffff data memacc   byte noext
EV2: 00100 00000 0000 ffff data ioacc    byte noext

rte>ev2 100 0 data iowrite                       << ev2の設定
     Addr  Mask  Data Mask Kind Status   Size Aux
EV1: 00000 fffff -    -    exe  -        -    noext
EV2: 00100 00000 0000 ffff data iowrite  byte noext

rte>tron !delay ffff ev2trg                      << tronの設定
Trace Settings:
 Delay Count  = 00ffff
 Trace Mode   = Real Time (real)
 PC Mode      = Jump and Execution (pcall)
 Delay Mode   = Disable (!delay)
 Data Trace 1 = ALL Trace (dt1all)
 Data Trace 2 = ALL Trace (dt2all)
 Start EV1    = Disable (!startev1)
 EV1 Trigger  = Disable (!ev1trg)
 EV2 Trigger  = Enable (ev2trg)
 Ext Trigger  = Disable (noext)
     Addr  Mask  Data Mask Kind Status   Size Aux
EV1: 00000 fffff 0000 ffff data memacc   byte noext
EV2: 00100 00000 0000 ffff data ioacc    byte noext
```

## ｔｒｃｏｆｆコマンド

[書式]

troff

[パラメータ]

なし

[機能]

トレースの取り込みを強制的に終了します。

## ｔｒａｃｅコマンド

[書式]

```
trace [POS] [all|pc|ev1|ev2|ev12] [asm]
```

[パラメータ]

| | |
|---|---|
| POS=±0..1ffff | トリガサイクル近辺または終了サイクルを０として、トレースの表示開始位置を１６進数で指定します。 |
| [all\|pc\|ev1\|ev2] | 取り込んだトレース情報の中から選択して表示するサイクルの指定 |
| all: | 全てのサイクル |
| pc: | 実行サイクルのみ |
| ev1: | ev1にマッチしたサイクルのみ |
| ev2: | ev2にマッチしたサイクルのみ |
| ev12 | ev1,ev2のマッチしたサイクルのみ |
| asm | 表示種別（アセンブル）...逆アセンブル表示で表示します。 |

[機能]

・トレースバッファの内容を表示します。
・トレース中にこのコマンドを発行した場合、強制的に取り込みを終了します。

[表示内容] ：アセンブラモード

```
rte>trace -10
   Cycle  Sub  Addr   Code       Instruction                    EXT  Stat
  -000011 0000 02304  7000       [mem read  ]                   0000 EV1
  -000006 0000 011d6  90         NOP                            0000 ISC
  -000006 0001 011d7  90         NOP                            0000 ---
  -000006 0002 011d8  a9ff0f     TEST  AX,0FFFH                 0000 ---
  -000006 0003 011db  7517       JNZ   11F4H                    0000 ---
  -000006 0004 011dd  90         NOP                            0000 ---
  -000006 0005 011de  90         NOP                            0000 ---
  -000006 0006 011df  90         NOP                            0000 ---
  -000006 0007 011e0  89c3       MOV   BX,AX                    0000 ---
  -000006 0008 011e2  d0e1       SHL   CL,1                     0000 ---
rte>
  -000006 0009 011e4  80c901     OR    CL,01H                   0000 ---
  -000006 000a 011e7  80f9bf     CMP   CL,BFH                   0000 ---
  -000006 000b 011ea  7502       JNZ   11EEH                    0000 ---
  -000004 0000 011ee  88c8       MOV   AL,CL                    0000 NEW
  -000001 0000 011f0  ee         OUT   DX,AL                    0000 ISC
*  000000 0000 00000  fd         [io write  ]                   0000 EV2
   00000b 0000 011f2  d8909090   ESC   BYTE PTR [BX+SI+9090H]   0000 ISC
   00000b 0001 011f6  90         NOP                            0000 ---
   00000b 0002 011f7  90         NOP                            0000 ---
   00000b 0003 011f8  90         NOP                            0000 ---
rte>
   00000b 0004 011f9  90         NOP                            0000 ---
   00000b 0005 011fa  90         NOP                            0000 ---
   00000b 0006 011fb  90         NOP                            0000 ---
   00000b 0007 011fc  90         NOP                            0000 ---
```

| | |
|---|---|
| Cycle: | トレースバッファ内の位置を16進数で相対的に表示しています。トリガポイント位置の近辺または、トレースの最終フレームを０としています。 |

| | |
|---|---|
| Sub: | 分岐や実行命令数などの情報から解析して生成したサイクルの番号です。 |
| Addr: | 実行アドレスまたは、バスサイクルのアドレスを表示します。 |
| Code: | 命令コードまたは、バスサイクルのデータを表示します。 |
| Instruction: | 命令のニーモニックまたは、バスの種類を表示します。 |
| EXT: | 外部入力端子EXI3..0の状態をビット列で表示します。 |
| Stat: | 表示にもとになるトレースパケットの種別を表示します。 |
| NEW: | 分岐や割込みがあった時のアドレスを示すパケット。 |
| ISC: | 何等かのパケットがあった時にそこまでに実行した命令数を示すパケット。 |
| EV1: | ev1を検出したパケット。 |
| EV2: | ev2を検出したパケット。 |
| OVF: | オーバフローが発生したことを示すパケット。このパケットが発生した場合、次のサイクルまでの間に取りこぼしが発生しています。 |

## ｖｅｒコマンド

［書式］
ver

［パラメータ］
なし

［機能］
ＫＩＴ－Ｖ３０ＭＺ－ＴＰのバージョンを表示します。