# 付録．Ａ　ＫＩＴ－ＶＲ４１２０－ＴＰ　内部コマンド

本書は、ＫＩＴ－ＶＲ４１２０－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各デバッガのマニュアルを参照ください。

ＰＡＲＴＮＥＲ／Ｗｉｎの場合

```
＞&          ＜＜ スルーコマンドへの移行します。
＞＃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。
```

ＧＨＳ－Ｍｕｌｔｉの場合

ＲＴＥＳＥＲＶを接続後、ターゲット・ウインドウで直接入力できます。

## コマンド一覧


付録．Ａ　ＫＩＴ－ＶＲ４１２０－ＴＰ　内部コマンド..... A-1
コマンド一覧..... A-1
コマンド書式..... A-2
オプションブレーク　：ＢＰＯＰＴコマンド..... A-3
キャッシュ操作　：ＣＡＣＨＥＩＮＩＴ，ＣＡＳＨＥＦＬＵＳＨコマンドA-4
環境設定　：ＥＮＶコマンド..... A-5
アクセス系イベント　：ＡＢＰ１，ＡＢＰ２コマンド..... A-7
実行系イベント　：ＩＢＰ１，ＡＢＰ２コマンド..... A-8
ヘルプ　：ＨＥＬＰコマンド..... A-9
ポート入力　：ＩＮＢ，ＩＮＨ，ＩＮＷ，ＩＮＤコマンド.....A-10
初期化　：ＩＮＩＴコマンド.....A-11
ＪＴＡＧリード　：ＪＲＥＡＤコマンド.....A-12
デバッガキャッシュの解除　：ＮＣコマンド.....A-13
デバッガキャッシュの設定　：ＮＣＤコマンド.....A-14
ソフトブレーク禁止領域の設定：ＮＳＢＰコマンド.....A-15
ソフトブレーク禁止領域の設定：ＮＳＢＰＤコマンド.....A-16
強制ユーザ領域の設定　：ＮＲＯＭコマンド.....A-17
強制ユーザ領域の設定　：ＮＲＯＭＤコマンド.....A-18
ポート出力　：ＯＵＴＢ，ＯＵＴＨ，ＯＵＴＷ，ＯＵＴＤコマンドA-19
ＣＰＵリセット　：ＲＥＳＥＴコマンド.....A-20
Ｅ．ＲＯＭの環境設定　：ＲＯＭコマンド.....A-21
ＴＬＢ　：ＴＬＢ３２，ＴＬＢ６４コマンド.....A-22
シンボル　：ＳＹＭＦＩＬＥ，ＳＹＭコマンド.....A-23
バージョン表示　：ＶＥＲコマンド.....A-24


ご注意：これらのコマンドは、ご使用になりたい機能がデバッガ本体に有していない場合にのみ補助的にご使用ください。ご使用になるデ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

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。）

## ｂｐｏｐｔコマンド

［書式］

bpopt [seq | [aand|aor] [iand|ior]]

［パラメータ］

seq: シーケンシャル条件を設定します。シーケンシャル条件は、abp1または、ibp1が発生後、abp2または、ibp2の条件成立でブレークします。

[aand|aor]: abp1,abp2の条件を設定します。

aand: abp1とabp2が同時に成立した時にブレークします。

ior : abp1または、abp2のどちらかが成立した時にブレークします。

[iand|ior]: ibp1,ibp2の条件を設定します。

iand: ibp1とibp2が同時に成立した時にブレークします。

ior : ibp1または、ibp2のどちらかが成立した時にブレークします。

［機能］

ブレーク用のイベントの条件を設定します。

ibp1,2は実行系のイベント、abp1,2はアクセス系のイベントです。

ibp1,2, abp1,2の設定方法はそれぞれのコマンドを参照ください。

［入力例］

bpopt aor

abp1とabp2をオアの条件で使用します。

bpopt seq

abp1,2,ibp1,2をシーケンシャル条件で使用します。

## ｃａｃｈｅｉｎｉｔ，ｃａｃｈｅｆｌｕｓｈコマンド

［書式］

cacheinit

cacheflush [ADDRESS [LENGTH]]

［パラメータ］

cacheinit キャッシュの初期化を行います。ライトバックは行いませんので、キャッシュの内容は破棄されます。

cacheflush 指定した範囲のキャッシュのフラッシュを行います。ライトバックが指定されている場合は、ライトバックサイクルが発生します。

ADDR: 開始アドレスを１６進数で指定します。

LENGTH: フラッシュする空間のバイト数を１６進数で指定します。

［機能］

キャッシュ操作の為のコマンドです。

［入力例］

cacheflush 80000000 1000

flush cache addr=80000000 len=00001000

0x80000000 0x1000バイトのキャッシュの内容をフラッシュします。

## ｅｎｖコマンド

［書式］

env [[!]auto] [[!]nmi] [jtag{25|12|5|2|1|500|250|100}] [[!]verify]
[[!]int0] [[!]int1] [[!]int2] [[!]int3] [[!]int4] [[!]timer]
[[!]cresetb] [[!]resetb] [pclock1|pclock2|pclock4]
[io_nouse|io_brkout|io_brkin|io_trgout|io_trgin]

［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。

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

jtag{25|12|5|2|1|500|250|100}: N-WireのJTAGクロック指定します。それぞれ以下に対応します。
[25MHz|12.5MHz|5MHz|2MHz|1MHz|500KHz|250KHz|100KHz]

> 備考：通常は、25MHzまたは、12.5MHzでご使用ください。1MHzより低い周波数を指定した場合は、デバッガの動作が著しく遅くなったり、異常になる場合があります。

[!]verify:メモリへの書き込み時にリードアウトしてベリファイするかどうか指定します。!はベリファイしないを意味します。

> 備考：ROMをエミュレーションしている領域に対しても、CPUからリード(jread相当)しますので、ダウンロード時のテストにも有効です。但し、処理速度が遅くなります。

[[!]int0] [[!]int1] [[!]int2] [[!]int3] [[!]int4] [[!]timer]:
外部割込みのマスク条件を指定します。!はマスクしないを意味します。

[[!]cresetb] [[!]resetb]: RESET端子のマスク条件を指定します。!はマスクしないを意味します。
CresetbがコColdResetB端子を、resetbがResetB端子を意味します。

[pclock1|pclock2|pclock4]: 常にデフォルトのpclock4でご使用ください。

[io_nouse|io_brkout|io_brkin|io_trgout|io_trgin]: BKTGIO_L端子のモードを指定します。

- io_nouse: 使用しません。
- io_brkout: ブレーク出力
- io_brkin: ブレーク入力
- io_trgout: 設定しないでください。
- io_trgin: 設定しないでください。

［機能］

envコマンドは、エミュレーション環境の設定とDCUの状態を表示します。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。

```
Probe:
 Unit       : RTE-1000-TP     << 接続している本体を表示します。
 Rom Probe  : Extend Type     << 接続しているROMプローブをタイプを表示します。
 Emem Size  : 32Mbyte         << 実装しているエミュレーションメモリの容量を表示します。
CPU:
 BKTGIO_L      = Present
 Cotrol Unit   = Present
CPU Settings:
 Auto Run      = ON (auto)
 JTAGCLOCK     = 12.5MHz (jtag12)
 Verify        = verify off (!verify)
Signals Mask:
 INT0          = NO MASK (!int0)
 INT1          = NO MASK (!int1)
 INT2          = NO MASK (!int2)
 INT3          = NO MASK (!int3)
 INT4          = NO MASK (!int4)
 TIMER         = NO MASK (!timer)
 NMI           = NO MASK (!nmi)
 COLDRESETB    = NO MASK (!cresetb)
 RESETB        = NO MASK (!resetb)
Trace UNIT:
 TRCCLK Mode   = PClock 1/4 (pclock4)
 BKTGIO_L Mode= not use (io_nouse)
```

［入力例］

```
env nmi verify timer
```

NMIとタイマ割込みをマスク、verify ONを指定をします。

## ａｂｐ１，ａｂｐ２コマンド

［書式］

```
abp{1|2} [ADDR [AMASK]] [data DATA [DMASK]] [asid ASID|noasid]
     [aeq|aneq] [deq|dneq] [read|write|accs]
     [nosize|byte|hword|word|dword]
abp{1|2} /del
```

［パラメータ］

ADDR:　　アドレスを１６進数で指定します。

AMASK:　　ADDRに対するマスクを指定します。ビット単位で'1'でマスクします。

data DATA [DMASK]|nodata:　　データ条件を指定します。

　　data DATA:　　dataを前置きし、データを１６進数で指定します。

　　DMASK:　　DATAに対するマスクを指定します。ビット単位で'1'でマスクします。

　　nodata:　　データの指定をしません。

asid ASID|noasid : asidを指定します。

　　asid ASID:　　ASIDを比較対象に含めます。

　　noasid:　　ASIDを比較対象にしません。

aeq|aneq: アドレスの条件を指定します。aeqで指定値を、aneqで指定値の否定を条件にします。

deq|dneq: データの条件を指定します。deqで指定値を、dneqで指定値の否定を条件にします。

read|write|acc:　　ステータス条件を指定します。

　　read:　　ステータス条件としてリードサイクルを指定します。

　　write:　　ステータス条件としてライトサイクルを指定します。

　　acc:　　ステータスの指定を条件から削除します。

nosize|byte|hword|word|dword:アクセスサイズ条件を指定します。

　　nosize:　　アクセスサイズは比較しません。

　　byte:　　アクセスサイズとしてバイト条件を指定します。

　　hword:　　アクセスサイズとしてハーフワード条件を指定します。

　　word:　　アクセスサイズとしてワード条件を指定します。

　　dword:　　アクセスサイズとしてダブルワード条件を指定します。

abp{1|2} /del: それぞれの条件を削除します。

［機能］

アクセスサイクルのブレーク用のイベントを指定します。

［使用例］

```
abp1 1000 0 deta 5555 0 hword read
```

　　1000h番地からハーフワードで5555hをリードしたサイクルをブレーク条件に指定します。

［備考］

abp1とabp2の組み合わせ条件はbpoptで指定します。

## ｉｂｐ１，ｉｂｐ２コマンド

［書式］

ibp{1|2} [ADDR [AMASK]] [asid ASID|noasid] [aeq|aneq]
ibp{1|2} /del

［パラメータ］

ADDR:　　アドレスを１６進数で指定します。
AMASK:　　ADDRに対するマスクを指定します。ビット単位で'1'でマスクします。
asid ASID|noasid: ASIDを指定します。
　　asid ASID:　　ASIDを比較対象に含めます。
　　noasid:　　ASIDを比較対象にしません。
aeq|aneq: アドレスの条件を指定します。aeqで指定値を、aneqで指定値の否定を条件にします。
ibp{1|2} /del: それぞれの条件を削除します。

［機能］

実行アドレスのイベントを指定します。

［使用例］

ibp1 1000 0
　　1000hの命令実行をマスクなしでブレーク条件として指定します。
1bp2 1000 0ff
　　1000hの下位8bitをマスクした実行アドレスをブレーク条件として指定します。
1bp2 1000 0 asid 10
　　asid=10hで1000hの命令実行をマスクなしでブレーク条件として指定します。

［備考］

ibp1とibp2の組み合わせ条件はbpoptで指定します。

## 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。

## ｉｎｂ，ｉｎｈ，ｉｎｗ，ｉｎｄコマンド

［書式］

inb [ADDR]
inh [ADDR]
inw [ADDR]
ind [ADDR]

［パラメータ］

ADDR: 入力ポートのアドレスを16進数で指定します。

［機能］

inb.inh,inw,indは、アクセスサイズを区別して、リードを行ないます。
inbはバイト、inhはハーフ・ワード、inwはワード、indはロングワード単位でアクセスします。

［使用例］

inb b0000000
　　b0000000Hからバイト(8-bit)でリードします。
inh 0000000
　　b0000000Hからハーフワード(16-bit)でリードします。
inw 0000000
　　b0000000Hからワード(32-bit)でリードします。
ind 0000000
　　b0000000Hからロングワード(64-bit)でリードします。

## ｉｎｉｔコマンド

［書式］
init

［パラメータ］
なし

［機能］
ＫＩＴ－ＶＲ４１２０－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。

## ｊｒｅａｄコマンド

［書式］

jread [ADDR [LENGTH]]

［パラメータ］

ADDR: アドレスを１６進数で指定します。

LENGTH: 読み出すバイト数を１６進数で指定します。(max 100h)

［機能］

ROMコマンドで割り付けたROMエミュレーション領域をJTAG(CPU)から読み出すためのコマンドです。
通常のコマンドではROMエミュレーション領域へのアクセスは内部のメモリに対し直接行っています。

［使用例］

jread a0000000 100

a0000000hから100hバイトをJTAG経由で読み出します。

## 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］

メモリ参照の高速化を図るため、デバッガ内部に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下さい。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 b8000000 100000

b8000000hから100000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 b8000000 100000
 No Memory Cache Area
 No. Address  Length
  1  b8000000   00100000
```

## 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 1

ブロック番号１をメモリキャッシュの除外エリアから削除します。

```
>nc bf000000 100
 No Memory Cache Area
 No. Address  Length
  1  bf000000   00000100
  2  b8000000   00100000

>ncd 1
 No Memory Cache Area
 No. Address  Length
  1  b8000000   00100000
```

## ｎｓｂｐコマンド

［書式］

nsbp [[ADDR [LENGTH]]

［パラメータ］

ADDR: ソフトウェアブレーク禁止領域の開始アドレスを指定します。

LENGTH: ソフトウェアブレーク禁止領域のバイト数を指定します。
指定領域の最小単位はワードバウンダリです。
また、指定できる領域の数は最大4ヶ所です。

［機能］

ソフトウェアブレークを禁止したい領域を指定します。
ブレークポイントを指定した場合、デバッガは暗黙的に対象アドレスに対し、メモリテスト（ライトアクセス）を行います。
一部のフラッシュROM等、ライトアクセスを行うことでメモリの状態が変り、正しいデータの読み出しが行えなくなる場合等に、ライトサイクルを禁止する目的で指定してください。
通常は、指定する必要はありません。

［使用例］

nsbp a0010000 20000
a0010000h番地から20000バイトの領域をソフトウェアブレーク禁止領域に指定します。

```
>nsbp a0010000 20000
Num Address  Length
01  a0010000 00020000
```

## ｎｓｂｐｄコマンド

［書式］

nspbd [ブロック番号|/all]

［パラメータ］

ブロック番号: 削除するソフトウェアブレーク禁止領域のブロック番号を指定します。
/all : 全てのソフトウェアブレーク禁止領域を削除します。

［機能］

nsbpで指定したソフトウェアブレーク禁止領域を削除します。

［使用例］

nsbpd 1

ブロック番号１をソフトウェアブレーク禁止領域から削除します。

```
nsbp
Num Address  Length
01  a0100000 00200000
02  a0400000 00010000

>nsbpd 1
Num Address  Length
01  a0400000 00010000
```

## ｎｒｏｍコマンド

［書式］

nrom [[ADDR [LENGTH]]

［パラメータ］

ADDR: 強制ユーザ領域の開始アドレスを指定します。

LENGTH: 強制ユーザ領域のバイト数を指定します。
指定領域の最小単位は、ロングワードバウンダリです。
また、指定できる領域の数は最大4ヶ所です。

［機能］

ROMコマンドで指定したROMエミュレーション領域内の一部が、ユーザシステム上の他の資源にマップされていた場合にその領域を指定します。
指定領域は、デバッガからのアクセスは強制的にユーザシステムに対し行われるようになります。
通常は、指定する必要はありません。

［使用例］

nrom a0000000 2000
a0000000h番地から2000バイトを強制ユーザ領域に指定します。

```
>nrom a0000000 1000
No. Address    Length
 1  a0000000   00001000

>nrom a010000 100
No. Address    Length
 1  a0000000   00001000
 2  a0010000   00000100
```

## ｎｒｏｍｄコマンド

［書式］

nromd [ブロック番号|/all]

［パラメータ］

ブロック番号: 削除する強制ユーザ領域のブロック番号を指定します。
/all　　: 全ての強制ユーザ領域のブロックを削除します。

［機能］

nromで指定した強制ユーザ領域を削除します。

［使用例］

ncd 1

ブロック番号１を強制ユーザ領域から削除します。

```
>nrom a0010000 8000
No. Address    Length
 1  a0000000   00001000
 2  a0010000   00008000

>nromd 1
No. Address    Length
 1  a0010000   00008000
```

## ｏｕｔｂ，ｏｕｔｈ，ｏｕｔｗ，ｏｕｔｄコマンド

［書式］

outb [[ADDR] DATA]
outh [[ADDR] DATA]
outw [[ADDR] DATA]
outd [[ADDR] DATA]

［パラメータ］

ADDR:　　出力ポートのアドレスを16進数で指定します。
DATA:　　出力するデータを１６進数で指定します。

［機能］

outb,outh,outwは、アクセスサイズを区別して、ライトを行ないます。
outbはバイト、outhはハーフ・ワード、outwはワード、outdはロングワード単位でアクセスします。

［使用例］

outb b800000 12
　　bfc00000hへバイトデータ：12hをIライトします。
outh b800000 1234
　　bfc00000hへハーフワードデータ：1234hをライトします。
outh b800000 12345678
　　bfc00000hへワードデータ：12345678hをライトします。
outd b800000 123456789abcdef0
　　bfc00000hへワードデータ：123456789abcdef0hをライトします。

## ｒｅｓｅｔコマンド

［書式］
reset

［パラメータ］
なし

［機能］
ＫＩＴ－ＶＲ４１２０－ＴＰの対象エミュレーションＣＰＵをリセットします。

## ｒｏｍコマンド

［書式］

rom [ADDR [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16]
[bus8|bus16|bus32] [little|big]

［パラメータ］

ADDR [LENGTH]: エミュレーションする領域を指定します。

ADDR: 開始アドレスを指定します。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|32m|64m|128m|256m: １本のROMプローブでエミュレートするROMのBit容量を指定します。512K-bitから256M-bitまでの値が指定できます。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、16bit-標準ROMケーブルを使用する場合は、rom16を指定します。

bus8|bus16|bus32: 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,16bit,32bitが指定できます。

little|big: romデータのエンディアンを指定します。ダウンロード時、little指定時は、ファイルのバイナリイメージをそのままの形で書き込みます。big指定時は、ROMのバスサイズに応じて、上位バイトと下位バイトのデータを入れ替えて書き込みます。

［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してください。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（使用しない）になっています。

［入力例］

rom bfc0000 40000 1m rom16 bus32 little

27C1024(1M-bitの16bit-ROM)をbfc00000hから256Kバイト(40000h)エミュレートします。この場合、16bit-romを１個をエミュレートします。Romのエンディアンはlittleです。（バイナリのイメージをそのままロードします。）

rom bfc00000 40000 2m rom rom16 bus16 big

27c2048(2M-bitの16bit-ROM)をbfc00000hから256Kバイト(40000h)エミュレートします。この場合、16bit-romを１個をエミュレートします。Romのエンディアンはbigです。（バイナリのイメージを上位と下位のバイトを入れ替えてロードします。）

＜備考＞

ｒｏｍコマンドで指定した範囲へのアクセスは、ツール内部のエミュレーションメモリに対し直接アクセスします。したがって、表示時、正しく見えていても、プロセッサから正しくROMにアクセスできない場合もあります。この場合、jreadコマンドを使用して確認するか、envコマンドでverifyをONにして書き込み（ダウンロード）を行って下さい。そうすることにより、エミュレーションしているROMの内容をCPUのバスを介して読み出すことでチェックできます。

## ｔｌｂ３２，ｔｌｂ６４コマンド

［書式］

tlb32 [all|INDEX [MASK HI LO0 LO1]]
tlb64 [all|INDEX [MASK HI LO0 LO1]]

［パラメータ］

all:　　全てのインデックスの表示を指定します。
INDEX:　特定のインデックスを指定します。
MASK HI LO0 LO1: 変更時、INDEXで指定したインデックスの内容を指定します。
　　４つセットで入力してください。
　MASK: PageMaskを指定します。
　HI:　EntryHiを指定します。
　LO0:　EntryLo0を指定します。
　LO1:　EntryLo1を指定します。

［機能］

TLBの内容の表示と変更を行います。
tlb32は、CPUが32bitの時の内容です。
Tlb64は、CPUが64bitの時の内容です。

［使用例］

tlb32 all
　全インデックスを内容を表示します。
Tlb32 10
　TLB#=10の内容を表示します。

## ｓｙｍｆｉｌｅ，ｓｙｍコマンド

［書式］

symfile FILENAME
sym [NAME]

［パラメータ］

symfile:　ファイル名を指定します。
sym:　　シンボルの先頭文字列を指定します。

［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。
Symコマンドは、読み込んだシンボルの表示（最大３０個）ができます。

［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。

## ｖｅｒコマンド

［書式］
ver

［パラメータ］
なし

［機能］
ＫＩＴ－ＶＲ４１２０－ＴＰのバージョンを表示します。