

APPENDIX A. RTE-V850E/MA1-IE INTERNAL COMMANDS

This document describes the internal commands of the RTE-V850E/MA1-IE. These commands can be used as through commands in each debug monitor. For information about entering through commands, refer to the manual supplied with the debug monitor.

(Example) PARTNER

- >& << Selects the through command mode.
- >#ENV << Internal command
- >& << Exits the through command mode.

Commands

APPENDIX A. RTE-V850E/MA1-IE INTERNAL COMMANDS.....	A-1
Commands.....	A-1
Command syntax.....	A-2
Access break point setting : abp, abp1, abp2, abp3, and abp4 commands	A-3
Access trace trigger setting : atp, atp1, and atp2 commands	A-4
Access trace trigger setting (range) : atp3 command.....	A-5
Environment setting : env command	A-7
Help : help command.....	A-9
Initialization : init command	A-10
Memory mapping specification : map command	A-11
Debugger cache area cancellation : nc command	A-12
Debugger cache area setting : ncd command.....	A-13
CPU resetting : reset command	A-14
Real-time RAM monitor base : rrm command	A-15
Real-time RAM monitor read : rrm command	A-16
SFR : sfr command.....	A-17
Symbol : symfile and sym commands.....	A-19
Execution time display : time command.....	A-20
Execution trace trigger setting : tp, tp1, and tp2 commands	A-21
Trace start and setting : tron command.....	A-22
Trace termination : troff command.....	A-23
Trace result display : trace command.....	A-24
Version information display : ver command	A-26

Caution Do not use these commands when equivalent functions are supported by the debugger. If you issue a command when the debugger in use has an equivalent function, contention will occur between them, and either the command or the equivalent function is likely to malfunction.

Command syntax

The internal commands of the RTE-V850E/MA1-IE are explained using the following notation.

command-name parameter

* A parameter enclosed in [and] is omissible. The horizontal line (|) separates optional parameters.

The command name is an alphabetic character string. The command name string is separated from the first parameter by a space or tab character. A parameter is specified using an alphabetic character string or hexadecimal number. Parameters are separated by a space or tab character. (No operator can be used for hexadecimal numbers.)

abp, abp1, abp2, abp3, and abp4 commands**[Format]**

```

abp [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
abp1 [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
abp2 [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
abp3 [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
abp4 [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]

```

[Parameters]

ADDR : An address value is specified in hexadecimal.

DATA : Data to be accessed is specified in hexadecimal.

MASK : A data mask is specified in hexadecimal.

read|write|access A type of status is specified.

read : Data read

write : Data write

access : Data access

byte|hword|word|nosize The size of data to be accessed is specified.

byte : Byte access

hword : Halfword access

word : Word access

nosize : No access size (In this specification, DATA is ignored.)

/del : Any previous setting by the command is canceled.

[Function]

The abp, abp1, abp2, abp3, and abp4 commands set or cancel an access break point. Four access break points can be set. The abp command automatically specifies a channel of an unused break point.

- The abp1, abp2, abp3, and abp4 commands are used to specify a channel explicitly.
- The data mask is used to specify invalid bits in the data.
- If a mask bit is 1, the corresponding data bit will not be compared. If the mask bits are ffffffff, for example, the accessed data will be utterly ignored.

[Example]

```
abp 1020 0 ffffffff access hword
```

A break occurs if a halfword at address 1020H is accessed (regardless of the bit configuration of the data).

```
abp 1020 100 0 write word
```

A break occurs if the word 100H is written to address 1020H.

```
abp2 /del
```

The abp2 command is canceled.

atp, atp1, and atp2 commands**[Format]**

```
atp [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
atp1 [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
atp2 [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
```

[Parameters]

ADDR : An address value is specified in hexadecimal.

DATA : Data to be accessed is specified in hexadecimal.

MASK : A data mask is specified in hexadecimal.

read|write|access A type of status is specified.

read : Data read

write : Data write

access : Data access

byte|hword|word|nosize The size of data to be accessed is specified.

byte : Byte access

hword : Halfword access

word : Word access

nosize : No access size

/del : Any previous setting by the command is canceled.

[Function]

The atp, atp1, and atp2 commands set or cancel an access trigger point. Two access trigger points can be set. The atp command automatically specifies a channel of an unused trigger point.

- The atp1 and atp2 commands are used to specify a channel explicitly.
- The data mask is used to specify invalid bits in the data.
- If a mask bit is 1, the corresponding data bit will not be compared.

Example) If the mask bits are ffffffff, the accessed data will be utterly ignored.

When any of these commands is issued, the trace buffer will be cleared, and tracing will newly begin.

[Example]

```
atp 1020 0 ffffffff access hword
```

Triggering occurs if a halfword at address 1020H is accessed (regardless of the bit configuration of the data).

```
atp 1020 100 0 write word
```

Triggering occurs if the word 100H is written to address 1020H.

```
atp2 /del
```

The atp2 command is canceled.

atp3 command**[Format]**

```
atp3 [in|out] [ADDR [HADDR [DATA [MASK]]]] [read|write|access] [byte|hword|word|nosize] [/del]
atp3 [<|>|<>|<=>|=eq] [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
atp3 noaddr [DATA [MASK]] [read|write|access] [byte|hword|word|nosize] [/del]
```

[Parameters]

in|out An address range is specified.

in : Addresses in a specified range are assumed to be valid.
 ADDR <= trigger address <= HADDR

out: Addresses out of a specified range are assumed to be valid.
 Trigger address < ADDR, HADDR < trigger address

ADDR : A lower address value is specified in hexadecimal.

HADDR : An upper address value is specified in hexadecimal.

<|>|<>|<=>|=eq An address condition is specified.

< : Addresses lower than the specified address are made valid.
 Trigger address < ADDR

> : Addresses higher than the specified address are made valid.
 Trigger address > ADDR

<> : Any address not equal to the specified address is made valid.
 Trigger address <> ADDR

<= : Addresses lower than or equal to the specified address are made valid.
 Trigger address <= ADDR

>= : Addresses higher than or equal to the specified address are made valid.
 Trigger address >= ADDR

eq : Any address equal to the specified address is made valid.
 Trigger address = ADDR

noaddr: Address conditions are ignored.

DATA : Data to be accessed is specified in hexadecimal.

MASK : A data mask is specified in hexadecimal.

read|write|access A type of status is specified.

read : Data read

write : Data write

access : Data access

byte|hword|word|nosize The size of data to be accessed is specified.

byte : Byte access

hword : Halfword access

word : Word access

nosize : No access size

/del : Any previous setting by the command is canceled.

[Function]

The atp3 command sets or cancels a range-of-address access break point. If in or out is specified as a address range condition, two address values, upper and lower, can be specified. In any other address range specification, only one address value can be specified.

When the command is issued, the trace buffer will be cleared, and tracing will newly begin.

- The data mask is used to specify invalid bits in the data.
- If a mask bit is 1, the corresponding data bit will not be compared. If the mask bits are ffffffff, for example, the accessed data will be utterly ignored.

[Example]

atp3 in 1020 1300 0 ffffffff access hword

Triggering occurs if a halfword at an address between 1020H and 1300H is accessed (regardless of the bit configuration of the data).

atp3 <= 1020 100 0 write word

Triggering occurs if the word 100H is written to address 1020H or lower.

env command

[Format]

```
env [romXX] [ramXX] [romless0|romless1|single0|single1]
env [[!]reset] [[!]nmi] [[!]hldrq] [[!]stop] [waitmode{0|1|2|3}] [w0|w16|w32|w64|w128|w256]
env [[!]dmatrace]
```

[Parameters]

rom[32|64|128|256]:

The size of internal ROM is specified. The sizes that can be specified include 32K, 64K, 128K, 256K, and 512K. ROM256 should be specified for the RTE-V850E/MA1-IE.

ram[4|12|28|60]:

The size of internal RAM is specified. The sizes that can be specified include 4K, 12K, 28K, and 60K. RAM12 should be specified for the RTE-V850E/MA1-IE.

romless0|romless1|single0|single1:

The CPU mode is specified.

[[!]reset]:

Whether to mask the RESET pin is specified. Special character ! specifies not to mask the pin.

[[!]nmi]:

Whether to mask the NMI pin is specified. Special character ! specifies not to mask the pin.

[[!]hldrq]:

Whether to mask the HLDRQ pin is specified. Special character ! specifies not to mask the pin.

[[!]stop]:

For the RTE-V850E/MA1-IE, keep this parameter at the initial value (to mask).

waitmode{0|1|2|3}:

The wait mode is specified.

The following combinations are provided. Generally, waitmode1 should be used.

wait mode	External	Emulation
	Memory	Memory
waitmode0	EXT-RDY	EXT-RDY
waitmode1	EXT-RDY	2WAIT
waitmode2	0WAIT	2WAIT
waitmode3	0WAIT	0WAIT

[w0|w16|w32|w64|w128|w256]:

A wait timeout value is specified.

- w0 specifies to wait permanently.
- The other values specify that bus operation is forcibly terminated (timed out) after 16 to 256 clock cycles.

[!]dmatrace:

This parameter specifies the tracing of DMA cycles. Special character ! specifies that DMA cycles are not to be traced.

[Function]

The env command is used to specify various environmental values for the emulation CPU of the RTE-V850E/MA1-IE. A parameter may be omitted if it is unnecessary.

- Parameters may be written in any order.
- If parameters that are mutually exclusive are coded, the last to appear is effective.
- The initial values immediately after startup are as follows:

Built-in ROM capacity : 256 KB

Built-in RAM capacity : 12 KB

Pin masks : Pins (other than the stop pin) are not masked.

Wait mode : waitmode1

Wait time : 256 clock pulses

DMA trace : Disabled

[Example]

```
env !nmi
```

The NMI pin is not masked.

```
env rom512 RAM28
```

The built-in ROM capacity is set to 512 Kbytes while the built-in RAM capacity is set to 28 Kbytes.

help command

[Format]

help [command]

[Parameters]

command: The name of the command you want to be helped with. When the command name is omitted, the command list is displayed.

[Function]

The help command displays the help message for each specified command.

[Example]

help map

The information about the map command is displayed.

init command

[Format]

init

[Parameters]

None

[Function]

The init command initializes the RTE-V850E/MA1-IE. All but the rejection area of the memory cache are initialized.

map command**[Format]**

map [ADDR LENGTH] [guard|ram|rom|target]

[Parameters]

ADDR : The address of a location where mapping is to begin

LENGTH : The number of bytes to be mapped

[guard|ram|rom|target] The mapping attributes are specified.

guard : Specifies a guarded area. An attempt to access the guarded area results in an error being detected.

ram : Specifies that the emulation memory be treated as RAM. This is a read/write parameter.

rom : Specifies that the emulation memory be treated as ROM. This is a read-only parameter.

target : Specifies that the memory of the target system be used.

[Function]

The map command carries out memory mapping. Memory is assigned in 1-Mbyte units.

[Caution]

When data is written to the area mapped for ROM, a break operation is forcibly performed but the written address data is corrupted.

[Example]

map 100000 100000 ram

1 Mbyte starting at 100000h is assigned as emulation RAM.

map 0 100000 guard

1 Mbyte starting at address 0h is assigned as a guard area.

nc command**[Format]**

```
nc [[ADDR [LENGTH]]
```

[Parameters]

[ADDR] : The start address of a memory cache rejection area is specified.

[LENGTH]: The number of bytes in the memory cache rejection area is specified. The default value is 32 bytes. The minimum value that can be specified is 32 bytes.

[Function]

The nc command specifies an I/O space, shared memory, or another area that cannot be used as a memory cache. The RTE-V850E/MA1-IE has an 8-bit block x 32-byte memory read cache in the debugger to speed up memory reference. When memory reference is repeated at the same address, data access is performed to the cache instead of memory to speed up the read operation. This cache operation conflicts with the actual operation of an I/O unit in the space where I/O is mapped. Therefore, the nc command can be used to specify the I/O space as a memory cache rejection area. A maximum of eight blocks can be specified as a memory cache rejection area. The minimum block size is 32 bytes. Since the following spaces are generally specified as the SFR space of the CPU, their addresses are set as initial values. Therefore, these spaces do not need to be specified as a no-memory cache area.

No Memory Cache Area

No.	Address	Length
1	0fff000	f0001000
2	03fff000	00001000

[Example]

```
nc 10000 1000
```

A 1000-byte area starting at address 10000 is specified as a memory cache rejection area.

```
>nc 10000 1000
```

No Memory Cache Area

No.	Address	Length
1	00010000	00001000
2	0fff000	f0001000
3	03fff000	0c001000

ncd command**[Format]**

ncd block-number

[Parameters]

block-number: The block number for a memory cache rejection area to be deleted is specified.

[Function]

The ncd command deletes a memory cache rejection area. The memory cache rejection area to be deleted is specified using a block number.

[Example]

ncd 1

Block 1 is deleted from a memory cache rejection area.

>nc 10000 1000

No Memory Cache Area

No.	Address	Length
-----	---------	--------

1	00010000	00001000
---	----------	----------

2	0fff000	f0001000
---	---------	----------

3	03fff000	0c001000
---	----------	----------

>ncd 1

No Memory Cache Area

No.	Address	Length
-----	---------	--------

1	0fff000	f0001000
---	---------	----------

2	03fff000	0c001000
---	----------	----------

reset command

[Format]

reset

[Parameters]

None

[Function]

The reset command resets the emulation CPU of the RTE-V850E/MA1-IE.

rrmb command

[Format]

rrmb ADDR

[Parameters]

ADDR: The base address for the real-time RAM monitor is specified.

[Function]

The rrmb command specifies the base address for the real-time RAM monitor. A 1-Kbyte area starting at the address specified here can be referenced using the rrm command, even when the CPU is running.

[Example]

rrmb 10000

A 1-Kbyte area starting at address 10000 is specified as the real-time RAM monitor area.

rrm command

[Format]

rrm [ADDR[LENGTH]]

[Parameters]

ADDR : The memory reference start address in the real-time RAM monitor is specified.

LENGTH : The number of bytes (up to 256 bytes) to be read is specified.

[Function]

The rrm command accesses a location in the real-time RAM monitor memory area. The length can be up to 256 bytes.

[Example]

rrm 10000 30

30H bytes are read from the real-time RAM monitor, starting at address 10000.

sfr command**[Format]**

sfr [reg] [VAL]

[Parameters]

VAL : An SFR register value is specified in hexadecimal.

reg : An SFR register name is specified. The names that can be used for registers are listed below.

Read/write registers

PAL PAH PDL PCS PCT PCM PCD PBD
 PMAL PMAH PMDL PMCS PMCT PMCM PMCD PMBD
 PMCAL PMCAH PMCDL PMCCS PFCCS PMCCT PMCCM PFCCM PMCCD PFCCD PMCBS PMCBD
 CSC0 CSC1 BSC VSWC DSAL0 DSAH0
 DDAL0 DDAH0 DSAL1 DSAH1 DDAL1 DDAH1 DSAL2 DSAH2 DDAL2 DDAH2 DSAL3 DSAH3
 DDAL3 DDAH3 DBC0 DBC1 DBC2 DBC3
 DADC0 DADC1 DADC2 DADC3 DCHC0 DCHC1 DCHC2 DCHC3
 DRST IMR0 IMR0L IMR0H IMR1 IMR1L IMR1H IMR2 IMR2L IMR2H IMR3 IMR3L
 IMR3H OVIC00 OVIC01 OVIC02 OVIC03 P00IC0 P00IC1 P01IC0 P01IC1 P02IC0
 P02IC1 P03IC0 P03IC1 P10IC0 P10IC1 P10IC2 P10IC3 P11IC0 P11IC1 P11IC2
 P11IC3 P12IC0 P12IC1 P12IC2 P12IC3 P13IC0 P13IC1 P13IC2 P13IC3 CMIC40
 CMIC41 CMIC42 CMIC43 DMAIC0 DMAIC1 DMAIC2 DMAIC3 CSIC0 SEIC0 SRIC0
 STIC0 CSIC1 SEIC1 SRIC1 STIC1 CSIC2 SEIC2 SRIC2 STIC2 ADIC
 PSC ADM0 ADM1 ADM2 P0 P1 P2 P3
 P4 P5 P7 PM0 PM1 PM2 PM3 PM4
 PM5 PMC0 PMC1 PMC2 PMC3 PMC4
 PMC5 PFC0 PFC2 PFC3 PFC4 BCT0
 BCT1 DWC0 DWC1 BCC ASC BCP PRC RWC
 DRC1 SCR1 RFC1 RFS1 DRC3 SCR3 RFC3 RFS3 DRC4 SCR4 RFC4 RFS4 DRC6 SCR6
 RFC6 RFS6 CMD0 TMCD0 CMD1 TMCD1
 CMD2 TMCD2 CMD3 TMCD3 CCC00 CCC01
 TMCC00 TMCC01 SESC0 CCC10 CCC11 TMCC10 TMCC11 SESC1 CCC20 CCC21
 TMCC20 TMCC21 SESC2 CCC30 CCC31 TMCC30 TMCC31 SESC3
 PHS DTFR0 DTFR1 DTFR2 DTFR3 PSMR
 CKC INTM0 INTM1 INTM2 INTM3 INTM4
 CSIM0 CSIC0 SOTB0 CSIM1 CSIC1 SOTB1
 CSIM2 CSIC2 SOTB2 ASIM0 TXB0
 CKSR0 BRGC0 ASIM1 TXB1 CKSR1 BRGC1 ASIM2 TXB2
 CKSR2 BRGC2 PWMC0 PWMB0 PWMC1
 PWMB1

Write-only registers

PRCMD PHCMD

Read-only registers

DDIS ISPR ADCR0 ADCR1 ADCR2 ADCR3

ADCR4 ADCR5 ADCR6 ADCR7 ADCR0H ADCR1H ADCR2H ADCR3H ADCR4H ADCR5H

ADCR6H

ADCR7H TMD0 TMD1 TMD2 TMD3 TMC0

TMC1 TMC2 TMC3 UNLOCK SIO0 SIO1

SIO2 RXB0 ASIS0 ASIF0 ASISE0

RXB1 ASIS1 ASIF1 ASISE1 RXB2 ASIS2 ASIF2 ASISE2

[Function]

The sfr command sets and displays a value in an SFR register.

[Example]

sfr P3

The value in the P3 register is displayed.

sfr PMC3 07

The value 07H is set in the PMC3 register.

symfile and sym commands**[Format]**

symfile FILENAME : Reads symbols from an elf file (.elf) of GHS.
sym [NAME] : Displays up to 30 symbols.

[Parameters]

symfile : File name
sym : The first character string in symbols.

[Function]

The symfile command reads symbols from the elf file specified as FILENAME. This command targets only global symbols. The sym command displays up to 30 symbols.

[Example]

```
symfile c:\test\dry\dry.elf
    The elf file (dry.elf) is read from the directory c:\test\dry.
sym m
    Up to 30 symbols starting with character m are displayed.
```

time command**[Format]**

time [sysclk]

[Parameters]

sysclk: The system clock of the CPU is specified in MHz. Up to two decimal places are valid.
The default is 50 MHz.

[Function]

The time command displays the result of execution time measurement. Each time the CPU starts running, the execution time measurement timer is initialized and started to count. It continues to count, as long as the CPU is running. The timer counts a clock cycle of the CPU as one.

[Remark]

The measurement value contains the overhead time (several clock cycles) needed in starting and breaking measurement.

[Example]

time 40

Execution time measured at a system clock of 40 MHz is displayed.

tp, tp1, and tp2 commands**[Format]**

```
tp [ADDR] [/del]
tp1 [ADDR] [/del]
tp2 [ADDR] [/del]
```

[Parameters]

ADDR : An address value is specified in hexadecimal.
/del : Any previous setting by the command is canceled.

[Function]

The tp, tp1, and tp2 commands set and cancel execution trigger points. There are two execution trigger points. The tp command specifies an unused trigger point automatically.

- The tp1 and tp2 commands are used to specify a trigger point explicitly.

When any of these commands is issued, the trace buffer will be cleared, and tracing will newly begin.

[Caution]

Do not specify execution trigger points at an execution starting address. Doing so causes the trigger points immediately after a starting position to be ignored.

[Example]

```
tp 1020
```

Triggering occurs when an instruction at address 1020H is executed.

tron command**[Format]**

```
tron [DELAY][add|cycle][all|qualify|qualify2]
      [x1|x2|x4|x8|x16|x32|x64|x128|x256|x512|x1k|x4k|x16k|x64k|x256k|x512k
      |x1m|x2m|x4m|x8m|x16m|x32m|x64m|x128m|x256m|x512m|x1g|x2g]
```

[Parameters]

DELAY=0..07fff Delay counter

The number of instruction cycles (delay counter) during which tracing should continue since the occurrence of triggering is specified in hexadecimal. The trace buffer can hold data for up to 32K cycles.

```
[add|cycle] [x1|x2|x4|x8|x16|x32|x64|x128|x256|x512|x1k|x4k|x16k|x64k|x256k|x512k
|x1m|x2m|x4m|x8m|x16m|x32m|x64m|x128m|x256m|x512m|x1g|x2g]
```

These parameters are the things for a mode setup of the timer for trace timer tag.

Add The addition mode is specified.

In this mode, a value from the previous cycle is accumulated.

Cycle The cycle mode is specified.

The trace timer tag counter value is displayed for an individual cycle.

x1..x2g Trace timer tag counter frequency division ratio

The frequency division ratio for the time tag counter clock is specified.

The setting x16 means that the timer unit is multiplied by 16.

[all|qualify|qualify2] The mode which trace takes in is specified.

All All cycles are taken in is specified.

Qualify Trigger cycles are taken in with delay counter is specified.

qualify2 Trigger cycles are taken in without delay counter is specified.

[Function]

The tron command clears the trace buffer and begins recording trace data.

[Caution]

1. Even when the addition mode is specified for the timer counter, the time tag will be cleared at restart if a break occurs.

[Example]

```
tron 100 x16 cycle
```

Tracing is terminated after data for 100h (256) cycles has been recorded since the occurrence of triggering. The timer unit is multiplied by 16. Execution results are written to the trace buffer in every cycle.

troff command

[Format]

troff

[Parameters]

None

[Function]

The troff command forcibly terminates tracing.

trace command**[Format]**

trace [POS] [asm]

[Parameters]

POS= $\pm 0..07ff$ Read start position (with the trigger point or end point as 0)
 The number of cycles counted from the beginning of the trace buffer is
 specified in hexadecimal.

asm Display type (assembler)...displayed in a disassembly form

[Function]

The trace command displays the contents of the trace buffer.

- Tracing is terminated when this command is issued.
- To restart tracing, it is necessary to issue the tron command.

[Display]: Assembly mode

Frame	neis	Time	Ext	Address	Code	Operand
_start:						
-0003	--i-	0003	0000	00000800	401e0000	movehi 0000h, zero, sp
-0002	--i-	0003	0000	00000804	231efcef	movea -1004h, sp, sp
-0001	--i-	0001	0000	00000808	40360000	movehi 0000h, zero, r6
+0000	--i-	0001	0000	0000080c	26365c11	movea +115ch, r6, r6
+0001	--i-	0001	0000	00000810	6600	jmp [r6]
main:						
+0002	--i-	0003	0000	0000115c	5c1a	add -04h, sp
+0003	--i-	0001	0000	0000115e	63ff0100	st.w lp, +00h [sp]
				00000246	Write	00000246h-> [00feff8h]
main+0006h:						
+0004	--i-	0001	0000	00001162	bff64f8	jarl RegChkInit (000009c6h)
RegChkInit:						
+0005	--i-	0003	0000	000009c6	501a	add +10h, sp
+0006	--i-	0001	0000	000009c8	63ff0d00	st.w lp, +0ch [sp]
				00001166	Write	00001166h-> [00feff4h]
+0007	--i-	0001	0000	000009cc	63b70900	st.w r22, +08h [sp]
				00000000	Write	00000000h-> [00feff0h]
+0008	--i-	0001	0000	000009d0	63af0500	st.w r21, +04h [sp]
				00000908	Write	00000908h-> [00fefeh]

- Frame : A trigger position is specified in hexadecimal with the first trigger cycle as 0.
- neis : The PSW flag is displayed.
- n : NMI flag
 - e : External interrupt flag
 - i : Exception flag
 - s : Saturation flag
- Time : The trace timer tag counter is displayed in hexadecimal.
- Ext : External data is displayed in bit units (in sequence of EXT0, 1, 2, 3 from the right).
- Address : An instruction execution address is displayed in hexadecimal.
- Code : For execution tracing, an instruction code is displayed in hexadecimal. For data tracing, data is displayed in hexadecimal.
- Operand : An instruction is displayed in a disassembly form.
- Data is displayed in the following format:
- Read [address] <- data
 - Write data -> [address]
- The number of digits in the data means the data size.

[Caution]

The times displayed for the following frames are invalid: frame during single-step execution, two frames immediately after real-time execution, and frame immediately before break operation.

ver command

[Format]

ver

[Parameters]

None

[Function]

The ver command indicates the version of the RTE-V850E/MA1-IE.