To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

E30A Emulator for the R32C/100 Series

How to Use Trace and Section Time Measurement Events

Summary

The E30A emulator for the R32C/100 series incorporates two debug functions—the trace function and the section time measurement function. By selecting trace or time measurement for the event and then specifying access conditions in the event setup screen of High-performance Embedded Workshop (HEW), it is possible to perform a trace or section time measurement.

The results of trace and section time measurement can each be confirmed in the trace window and the time section measurement window of HEW.

The access conditions selectable for each event are listed below.

Event	Access condition				
	EXECUTION	BRANCH	READ	WRITE	R/W
Trace event (trace extraction event)	×	○	○	○	○
Trace event (trace start event/ trace end event) [*1]	○	×	○	○	○
Section time measurement event	×	×	○	○	○

*1. Can be assigned to only events E0 and E4.

Limitations on access conditions of each event are shown below.

[Limitations of trace events]

If EXECUTION is specified for access condition of a trace, the emulator, owing to NSD specifications, may erroneously assume that the condition is met when the set address is prefetched [*2] (prefetched in size of about 48 bytes). This will result in excessive trace information being acquired or a failure to acquire the intended trace information depending on trace measurement range (After or Before).

*2. Although it is possible to set an address for the trace event condition by taking prefetch into consideration, such is not a much practiced case because there are various factors such as interrupt process, branch process, loop process, or compiler optimization that make it difficult to identify the address.

[Limitations of section time measurement events]

Since the section time measurement is an operation to measure the execution time between data accesses based on trace information, the data access trace information (address, data, data size, read/write) is detected in the emulator as the measurement start event and measurement end event.

Therefore, it is only data access (READ, WRITE, or R/W) that can be specified for access condition, and in no case can such a time measurement that uses the start and end of a function as condition be performed (i.e., EXECUTION cannot be specified for access condition).

This application note explains the measures to be taken to cope with the above limitations.

Table of Contents

1. Environment

Microcomputers used: R32C/118 group (R5F64189PFB)

HEW: V.4.05.01.001

Emulator software: R32C/100 E30A Emulator Debugger V.1.00 Release 00

Compiler: R32C/100 Series C Compiler V.1.01 Release 00

2. Flow of This Document

The corrective measures are described below.

[Corrective measures for limitations on trace events]

- Embed a sequence in place that will have a data access (read process [*3]) generated at the start position of a trace and specify the data access for access condition (alteration of the source required).

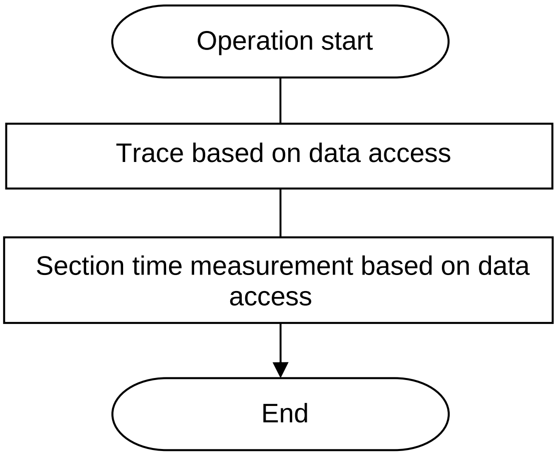
*3. If a write process is used for condition, because a trace event cannot have data comparison set for condition (e.g., a condition like the one that will cause a trace to start when variable ‘a’ = 1), a trace may erroneously be started at the time the data (variable) set for access condition is initialized (by writing “00”), making it impossible to trace the intended range.

(For details, see paragraph (4) in Section 3, “Frequently Asked Questions.”)

[Corrective measures for limitations on section time measurement events]

- Embed a sequence in place that will have a data access generated in a section of the program (start to end) in which a time measurement is to be performed and specify the data access for access condition (alteration of the source required).

The flow of this document is shown below.

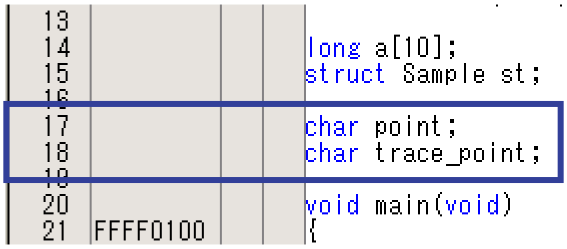

Operation start
Trace based on data access
Section time measurement based on data access
End

2.1 Trace Based on Data Access

This section explains the method for performing a trace unaffected by prefetch by first embedding a global variable read process at the start position of a trace event and then specifying the global variable read process (data access) for the event's access condition.

Shown here is an example in which a trace is started from the beginning of the sort function by using the Tutorial program included with the E30A emulator software.

(1) Add global variables point and trace_point in the source file.

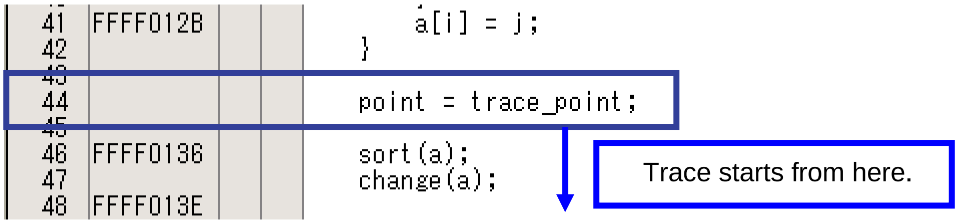


```
          long a[10];
          struct Sample st;

          char point;
          char trace_point;

          void main(void)
FFFF0100  {
```

(2) Add point = trace_point at a place preceding the sort function. Perform a build after adding and then download the program.



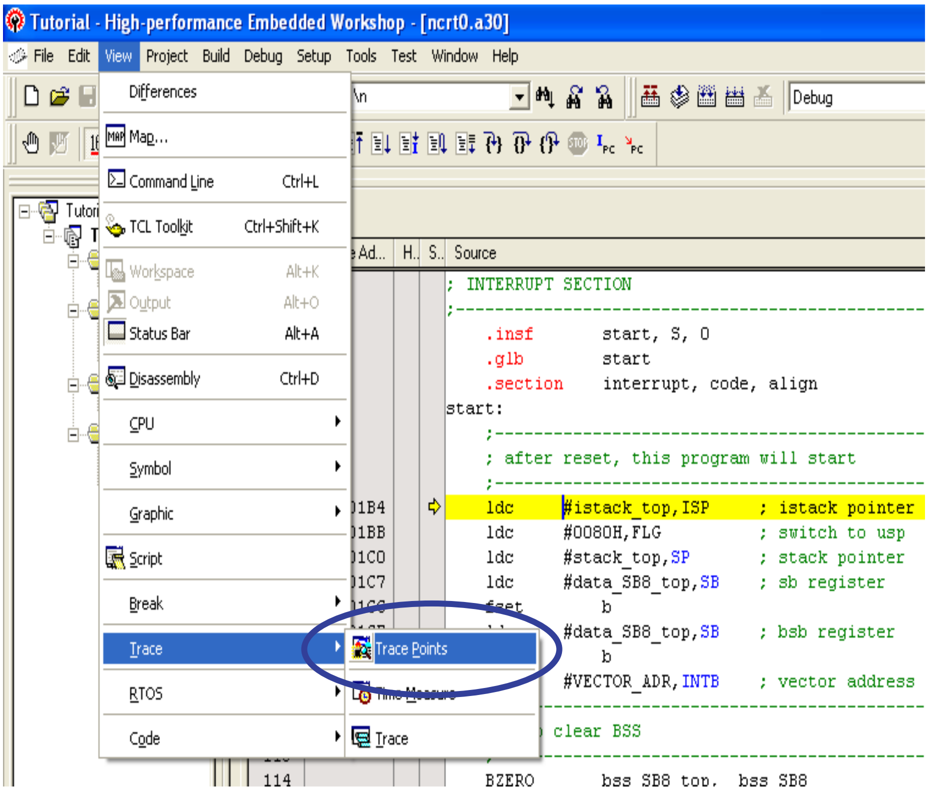
```
FFFF012B        a[i] = j;
            }

            point = trace_point;

FFFF0136    sort(a);
            change(a);
FFFF013E
```

Trace starts from here.

(3) Choose Trace from the View menu and then select Trace Point.

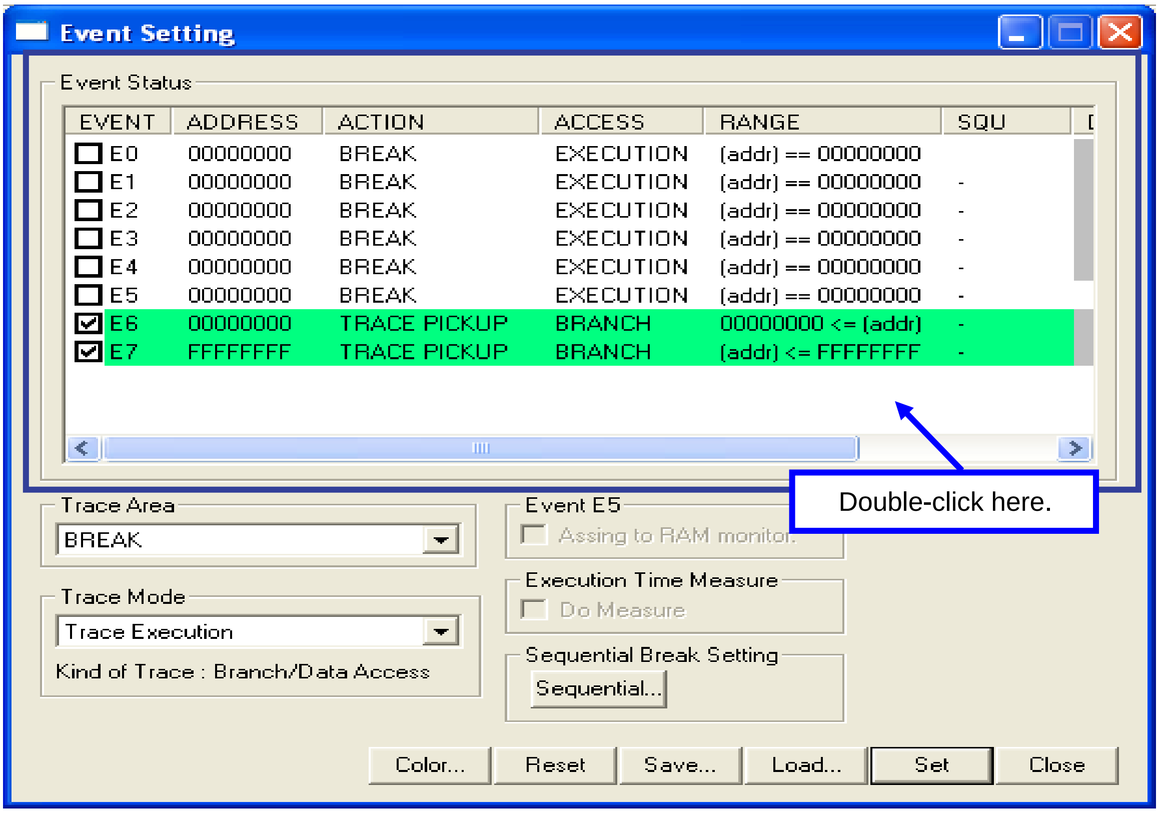


```
; INTERRUPT SECTION
;---------------------------------------------------
    .insf       start, S, 0
    .glb        start
    .section    interrupt, code, align
start:
    ;-----------------------------------------------
    ; after reset, this program will start
    ;-----------------------------------------------
    ldc     #istack_top,ISP     ; istack pointer
    ldc     #0080H,FLG          ; switch to usp
    ldc     #stack_top,SP       ; stack pointer
    ldc     #data_SB8_top,SB    ; sb register
    fset        b
    ldc     #data_SB8_top,SB    ; bsb register
                b
            #VECTOR_ADR,INTB    ; vector address

            ; clear BSS

    BZERO       bss_SB8_top,  bss_SB8
```

(4) The Event Settings dialog box is displayed. Double-click in the Event Status column.



Event Setting

Event Status

EVENT	ADDRESS	ACTION	ACCESS	RANGE	SQU	[
□ E0	00000000	BREAK	EXECUTION	(addr) == 00000000		
□ E1	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
□ E2	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
□ E3	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
□ E4	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
□ E5	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
☑ E6	00000000	TRACE PICKUP	BRANCH	00000000 <= (addr)	-	
☑ E7	FFFFFFFF	TRACE PICKUP	BRANCH	(addr) <= FFFFFFFF	-	

Double-click here.

Trace Area: BREAK

Trace Mode: Trace Execution

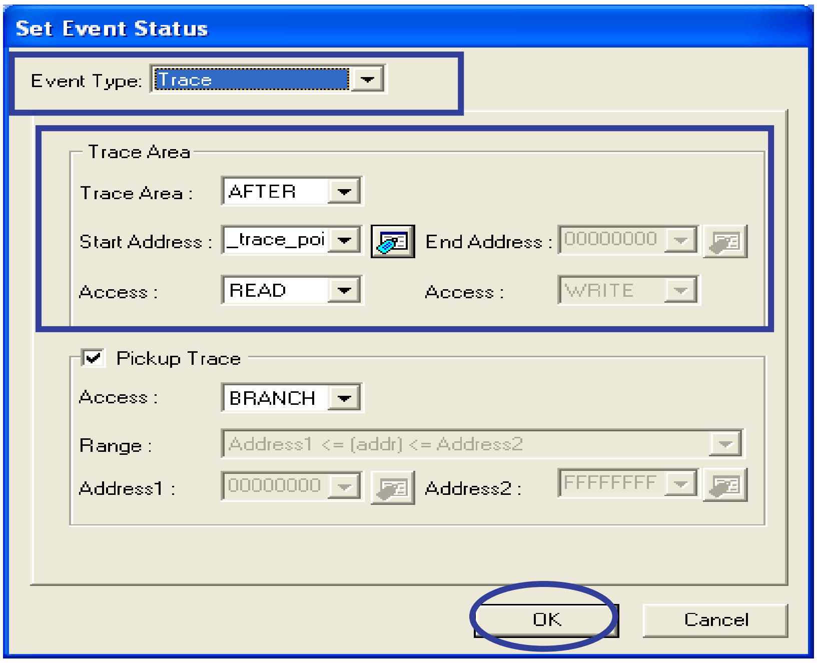
Kind of Trace : Branch/Data Access

Event E5: □ Assing to RAM monitor

Execution Time Measure: □ Do Measure

Sequential Break Setting

(5) The Set Event Status dialog box is displayed. Select Trace for Event Type, AFTER for Trace Measurement Range, trace_point for Start Address, and READ for Access Condition. Then click the OK button.



Set Event Status

Event Type: Trace

Trace Area

Trace Area : AFTER

Start Address : _trace_poi

End Address : 00000000

Access : READ

Access : WRITE

☑ Pickup Trace

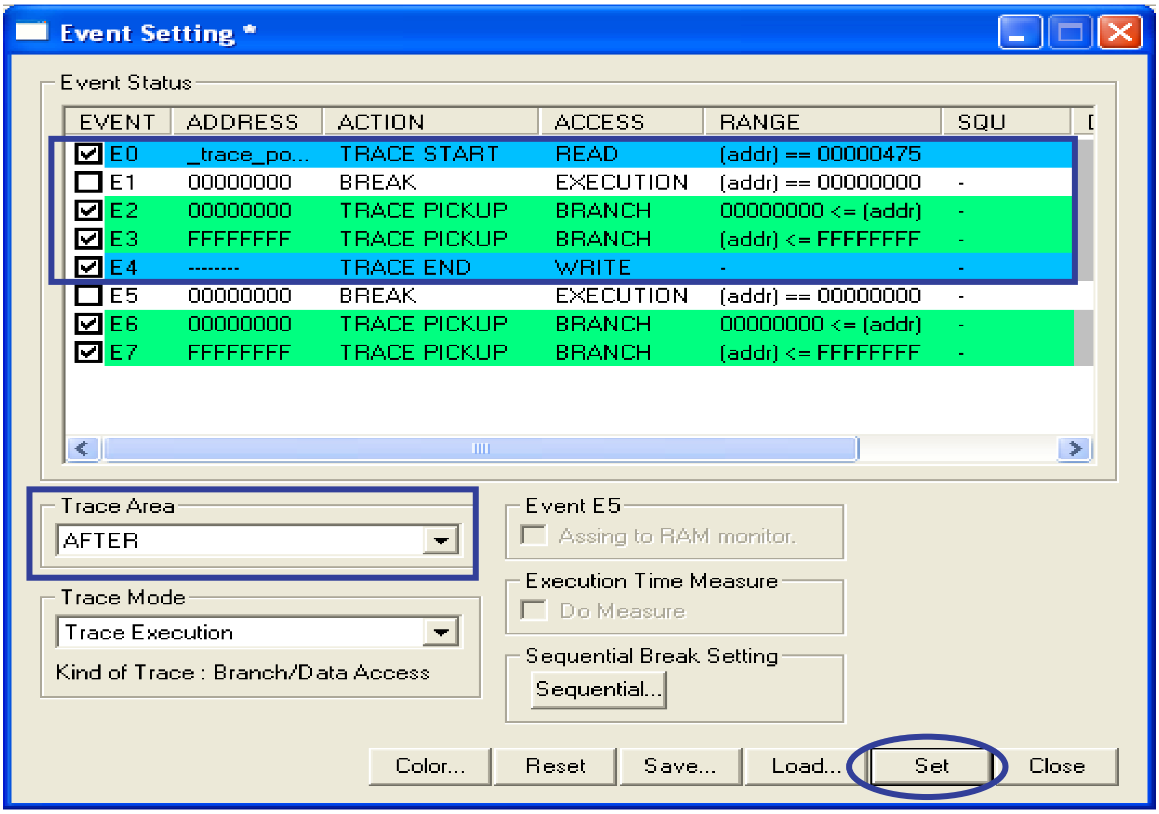
Access : BRANCH

Range : Address1 <= (addr) <= Address2

Address1 : 00000000

Address2 : FFFFFFFF

(6) The contents you've set are displayed in the Event Settings dialog box. Click the Set button to make the changes take effect.



EVENT	ADDRESS	ACTION	ACCESS	RANGE	SQU
☑ E0	_trace_po...	TRACE START	READ	(addr) == 00000475	
☐ E1	00000000	BREAK	EXECUTION	(addr) == 00000000	-
☑ E2	00000000	TRACE PICKUP	BRANCH	00000000 <= (addr)	-
☑ E3	FFFFFFFF	TRACE PICKUP	BRANCH	(addr) <= FFFFFFFF	-
☑ E4	--------	TRACE END	WRITE	-	-
☐ E5	00000000	BREAK	EXECUTION	(addr) == 00000000	-
☑ E6	00000000	TRACE PICKUP	BRANCH	00000000 <= (addr)	-
☑ E7	FFFFFFFF	TRACE PICKUP	BRANCH	(addr) <= FFFFFFFF	-

Trace Area: AFTER

Trace Mode: Trace Execution

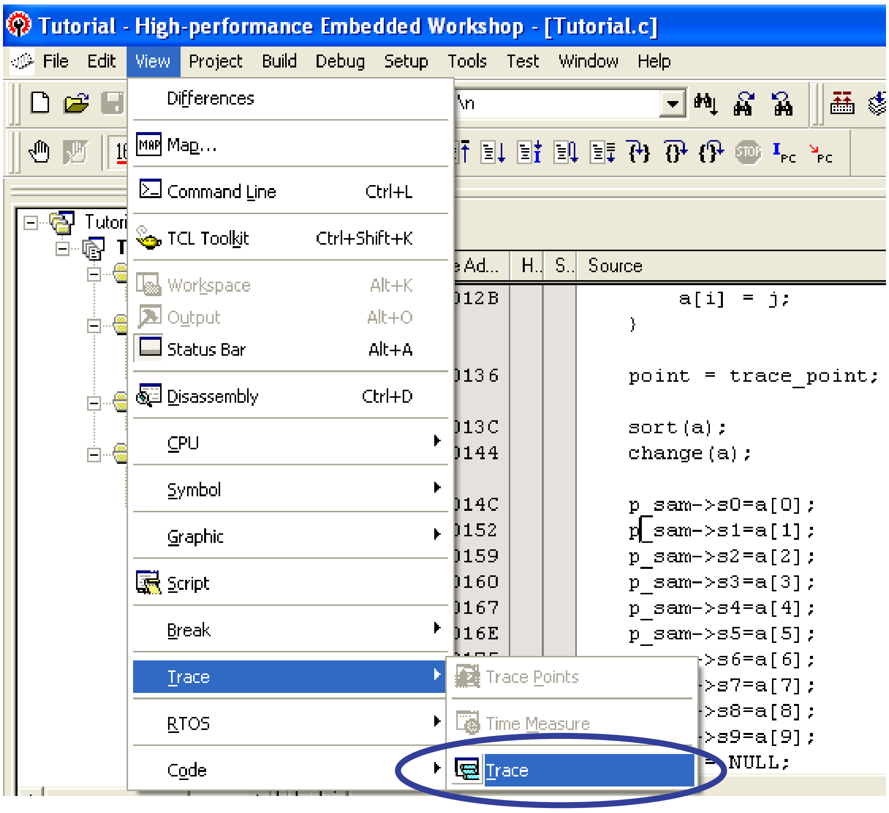
Kind of Trace : Branch/Data Access

Event E5: ☐ Assing to RAM monitor.

Execution Time Measure: ☐ Do Measure

Sequential Break Setting: Sequential...

(7) Choose Trace from the View menu and then select Trace.



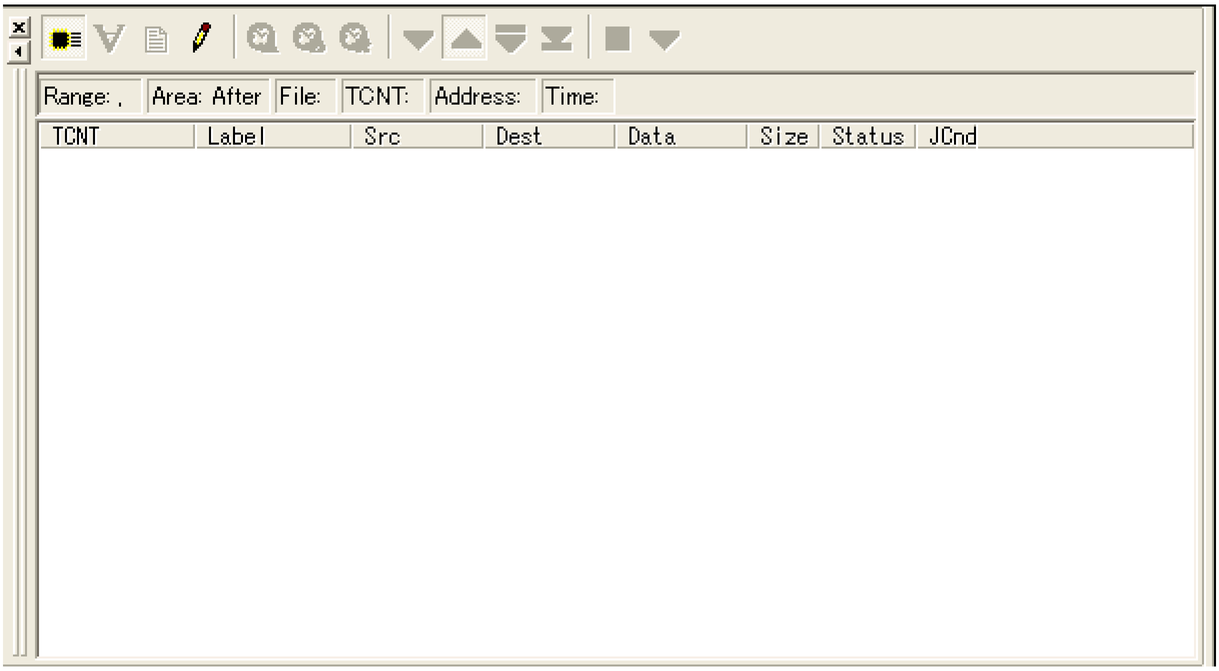
```
        a[i] = j;
    }

    point = trace_point;

    sort(a);
    change(a);

    p_sam->s0=a[0];
    p_sam->s1=a[1];
    p_sam->s2=a[2];
    p_sam->s3=a[3];
    p_sam->s4=a[4];
    p_sam->s5=a[5];
```

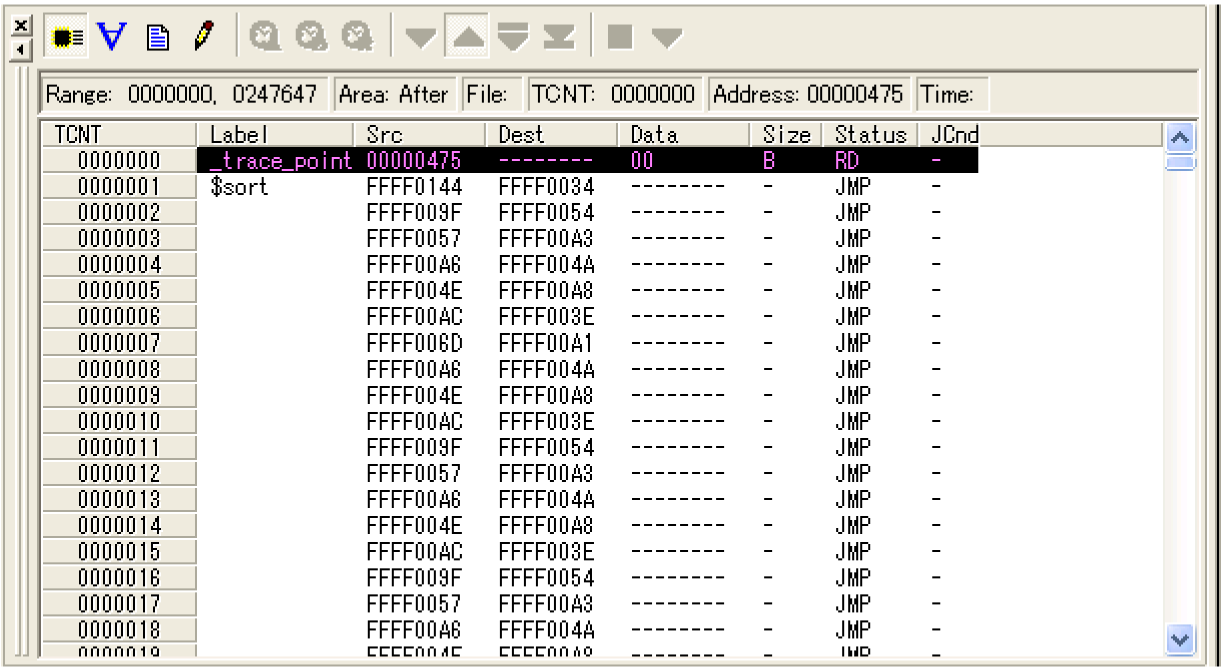
(8) The Trace window is displayed.



Range: , | Area: After | File: | TCNT: | Address: | Time:

TCNT	Label	Src	Dest	Data	Size	Status	JCnd

(9) Run the program and stop, and the result of a trace will be displayed. Although one event has just been used in the global variable process (data access) specified for access condition, a trace can be started from the beginning of the sort function.



Range: 0000000, 0247647 | Area: After | File: | TCNT: 0000000 | Address: 00000475 | Time:

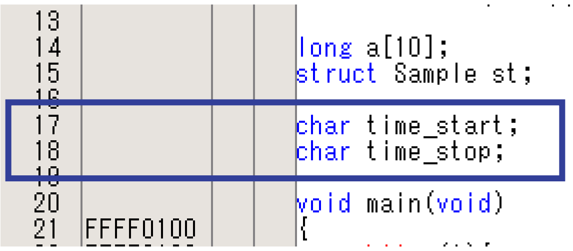
TCNT	Label	Src	Dest	Data	Size	Status	JCnd
0000000	_trace_point	00000475	--------	00	B	RD	-
0000001	$sort	FFFF0144	FFFF0034	--------	-	JMP	-
0000002		FFFF009F	FFFF0054	--------	-	JMP	-
0000003		FFFF0057	FFFF00A3	--------	-	JMP	-
0000004		FFFF00A6	FFFF004A	--------	-	JMP	-
0000005		FFFF004E	FFFF00A8	--------	-	JMP	-
0000006		FFFF00AC	FFFF003E	--------	-	JMP	-
0000007		FFFF006D	FFFF00A1	--------	-	JMP	-
0000008		FFFF00A6	FFFF004A	--------	-	JMP	-
0000009		FFFF004E	FFFF00A8	--------	-	JMP	-
0000010		FFFF00AC	FFFF003E	--------	-	JMP	-
0000011		FFFF009F	FFFF0054	--------	-	JMP	-
0000012		FFFF0057	FFFF00A3	--------	-	JMP	-
0000013		FFFF00A6	FFFF004A	--------	-	JMP	-
0000014		FFFF004E	FFFF00A8	--------	-	JMP	-
0000015		FFFF00AC	FFFF003E	--------	-	JMP	-
0000016		FFFF009F	FFFF0054	--------	-	JMP	-
0000017		FFFF0057	FFFF00A3	--------	-	JMP	-
0000018		FFFF00A6	FFFF004A	--------	-	JMP	-
0000019		FFFF004E	FFFF00A8	--------	-	JMP	-

2.2 Section Time Measurement Based on Data Access

This section explains the method for measuring the execution time (from start to end) of a function by embedding a global variable write process at the start and end positions of a section time measurement event and then specifying the global variable write process (data access) for the event's access condition.

Shown here is an example in which measurement is made of the execution time of the sort function by using the Tutorial program included with the E30A emulator software.

(1) Add global variables time_start and time_stop in the source file.

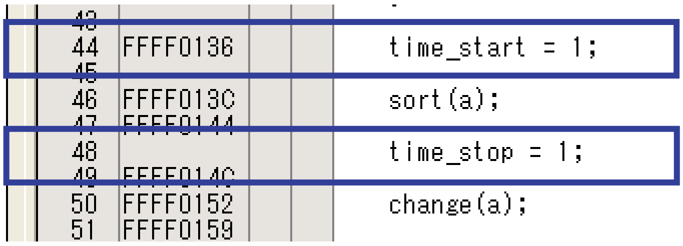


```
long a[10];
struct Sample st;

char time_start;
char time_stop;

void main(void)
FFFF0100    {
```

(2) Add time_start = 1 and time_stop = 1 before and after the sort function. Perform a build after adding and then download the program.

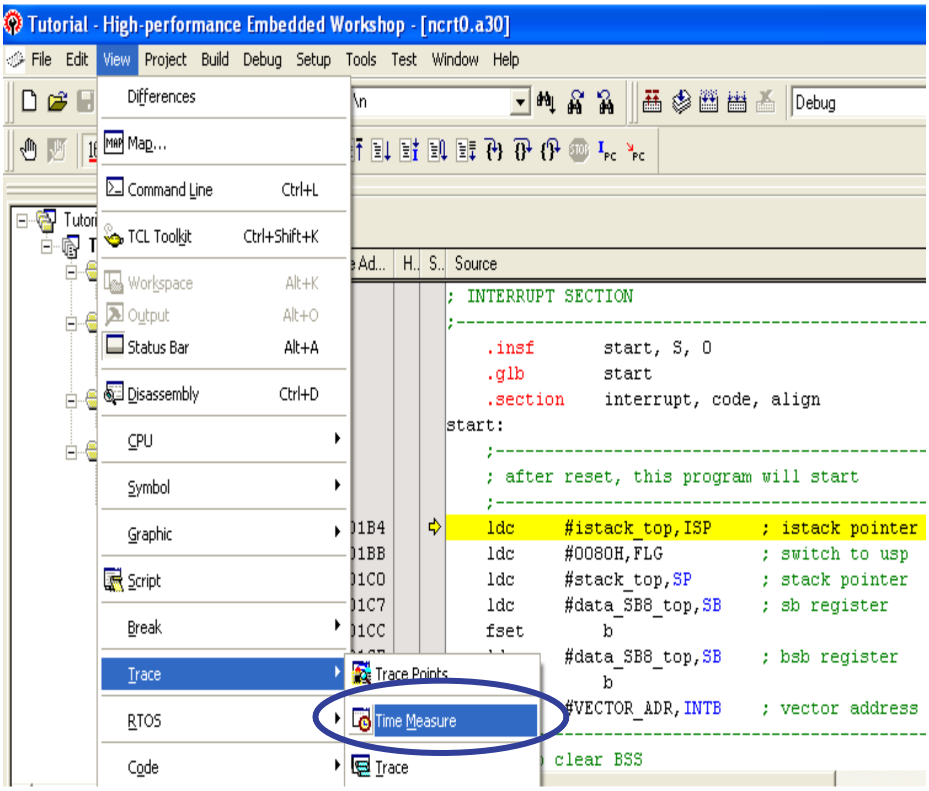


```
FFFF0136    time_start = 1;

FFFF013C    sort(a);
FFFF0144
            time_stop = 1;
FFFF014C
FFFF0152    change(a);
FFFF0159
```

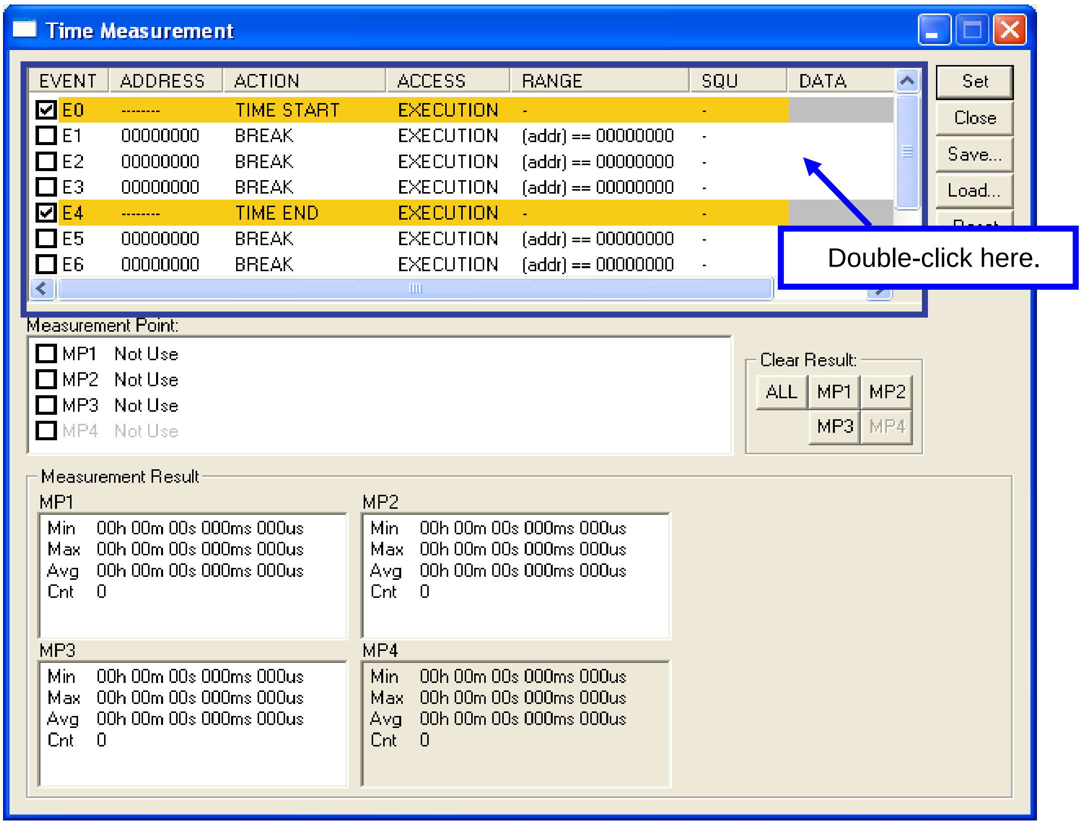
Measurement is taken of this section.

(3) Choose Trace from the View menu and then select Section Time Measurement.



```
; INTERRUPT SECTION
;------------------------------------------------
        .insf       start, S, 0
        .glb        start
        .section    interrupt, code, align
start:
        ;-----------------------------------------
        ; after reset, this program will start
        ;-----------------------------------------
        ldc     #istack_top,ISP     ; istack pointer
        ldc     #0080H,FLG          ; switch to usp
        ldc     #stack_top,SP       ; stack pointer
        ldc     #data_SB8_top,SB    ; sb register
        fset        b
                #data_SB8_top,SB    ; bsb register
                    b
                #VECTOR_ADR,INTB    ; vector address
        ;-----------------------------------------
        ; clear BSS
```

(4) The Section Time Measurement dialog box is displayed. Double-click in the Event Status column.



Time Measurement

EVENT	ADDRESS	ACTION	ACCESS	RANGE	SQU	DATA
☑ E0	--------	TIME START	EXECUTION	-	-	
□ E1	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
□ E2	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
□ E3	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
☑ E4	--------	TIME END	EXECUTION	-	-	
□ E5	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
□ E6	00000000	BREAK	EXECUTION	(addr) == 00000000	-	

Measurement Point:
□ MP1 Not Use
□ MP2 Not Use
□ MP3 Not Use
□ MP4 Not Use

Measurement Result

MP1
Min 00h 00m 00s 000ms 000us
Max 00h 00m 00s 000ms 000us
Avg 00h 00m 00s 000ms 000us
Cnt 0

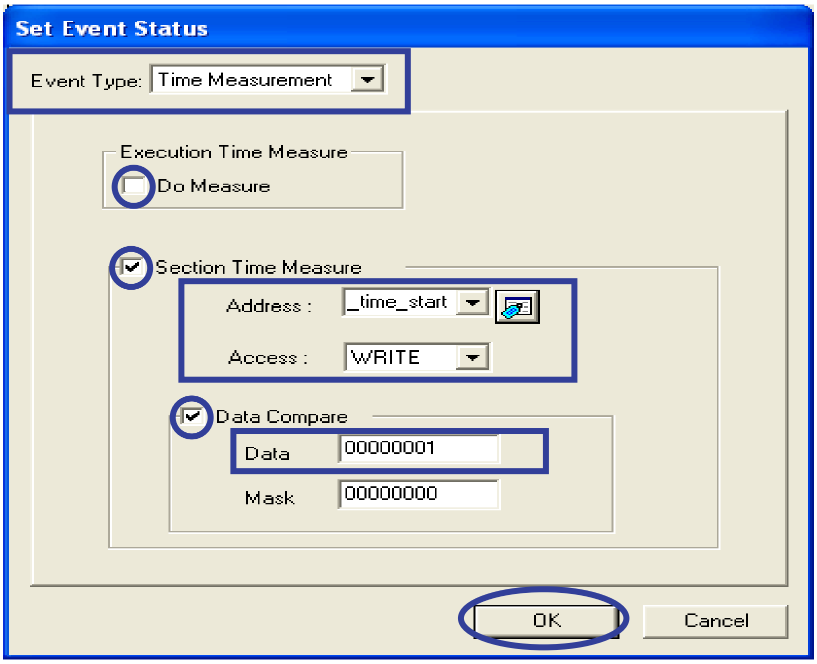
MP2
Min 00h 00m 00s 000ms 000us
Max 00h 00m 00s 000ms 000us
Avg 00h 00m 00s 000ms 000us
Cnt 0

MP3
Min 00h 00m 00s 000ms 000us
Max 00h 00m 00s 000ms 000us
Avg 00h 00m 00s 000ms 000us
Cnt 0

MP4
Min 00h 00m 00s 000ms 000us
Max 00h 00m 00s 000ms 000us
Avg 00h 00m 00s 000ms 000us
Cnt 0

Double-click here.

(5) The Set Event Status dialog box is displayed. Select Time Measurement for Event Type, check the Section Time Measurement and the Data Comparison check boxes and uncheck the Execution Time Measurement check box. Select _time_start for Address and WRITE for Access Condition and set Data to “00000001." Then click the OK button.



Set Event Status

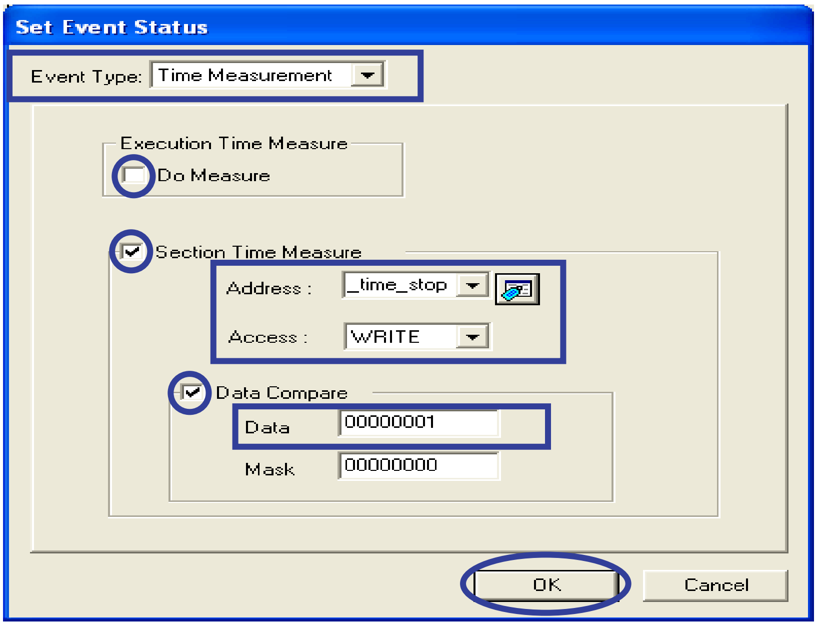
Event Type: Time Measurement

Execution Time Measure
□ Do Measure

☑ Section Time Measure
Address : _time_start
Access : WRITE

☑ Data Compare
Data 00000001
Mask 00000000

In the same way, complete the dialog box by selecting _time_stop for Address etc. as shown below, and then click the OK button.



Set Event Status

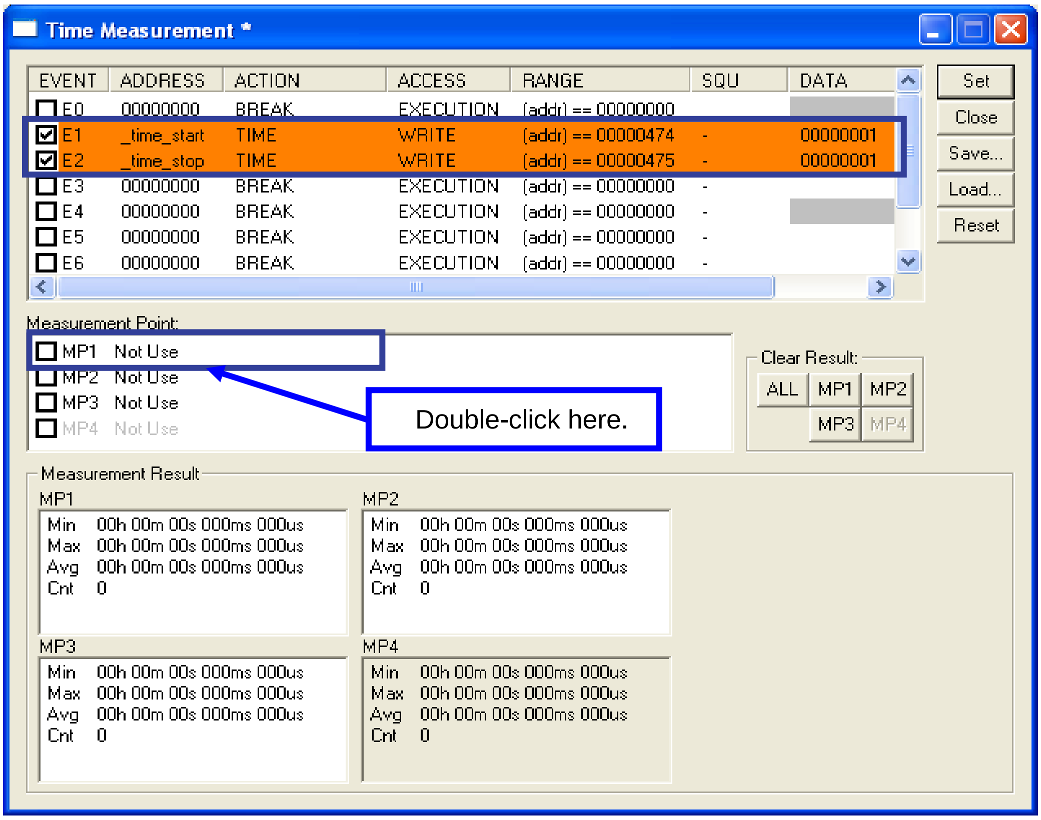
Event Type: Time Measurement

Execution Time Measure
□ Do Measure

☑ Section Time Measure
Address : _time_stop
Access : WRITE

☑ Data Compare
Data 00000001
Mask 00000000

(6) The contents you've set are displayed in the Section Time Measurement dialog box. At this time, check to see if _time_start and _time_stop are set for events E1 and E2, respectively. Next, select MP1 in the Measurement Points column and double-click in it.



Time Measurement *

EVENT	ADDRESS	ACTION	ACCESS	RANGE	SQU	DATA
□ E0	00000000	BREAK	EXECUTION	(addr) == 00000000		
☑ E1	_time_start	TIME	WRITE	(addr) == 00000474	-	00000001
☑ E2	_time_stop	TIME	WRITE	(addr) == 00000475	-	00000001
□ E3	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
□ E4	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
□ E5	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
□ E6	00000000	BREAK	EXECUTION	(addr) == 00000000	-	

Measurement Point:
□ MP1 Not Use
□ MP2 Not Use
□ MP3 Not Use
□ MP4 Not Use

Double-click here.

Measurement Result

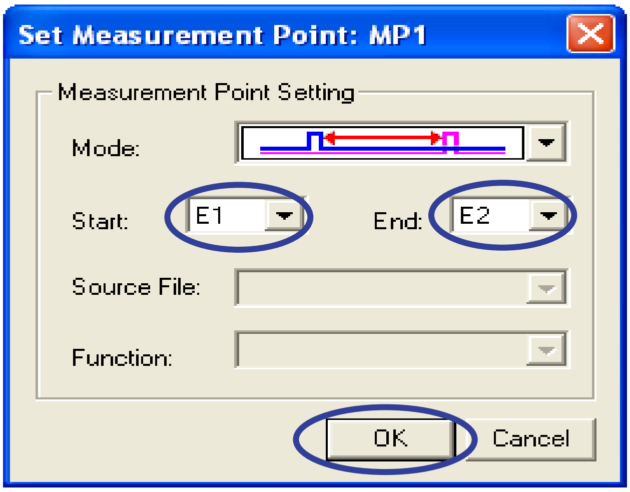
MP1
Min 00h 00m 00s 000ms 000us
Max 00h 00m 00s 000ms 000us
Avg 00h 00m 00s 000ms 000us
Cnt 0

MP2
Min 00h 00m 00s 000ms 000us
Max 00h 00m 00s 000ms 000us
Avg 00h 00m 00s 000ms 000us
Cnt 0

MP3
Min 00h 00m 00s 000ms 000us
Max 00h 00m 00s 000ms 000us
Avg 00h 00m 00s 000ms 000us
Cnt 0

MP4
Min 00h 00m 00s 000ms 000us
Max 00h 00m 00s 000ms 000us
Avg 00h 00m 00s 000ms 000us
Cnt 0

(7) The Set Measurement Point: MP1 dialog box is displayed. Select E1 for Start and E2 for End, and then click the OK button. This setting causes a measurement to be made from when event E1 occurs till when E2 occurs.



Set Measurement Point: MP1

Measurement Point Setting

Mode:

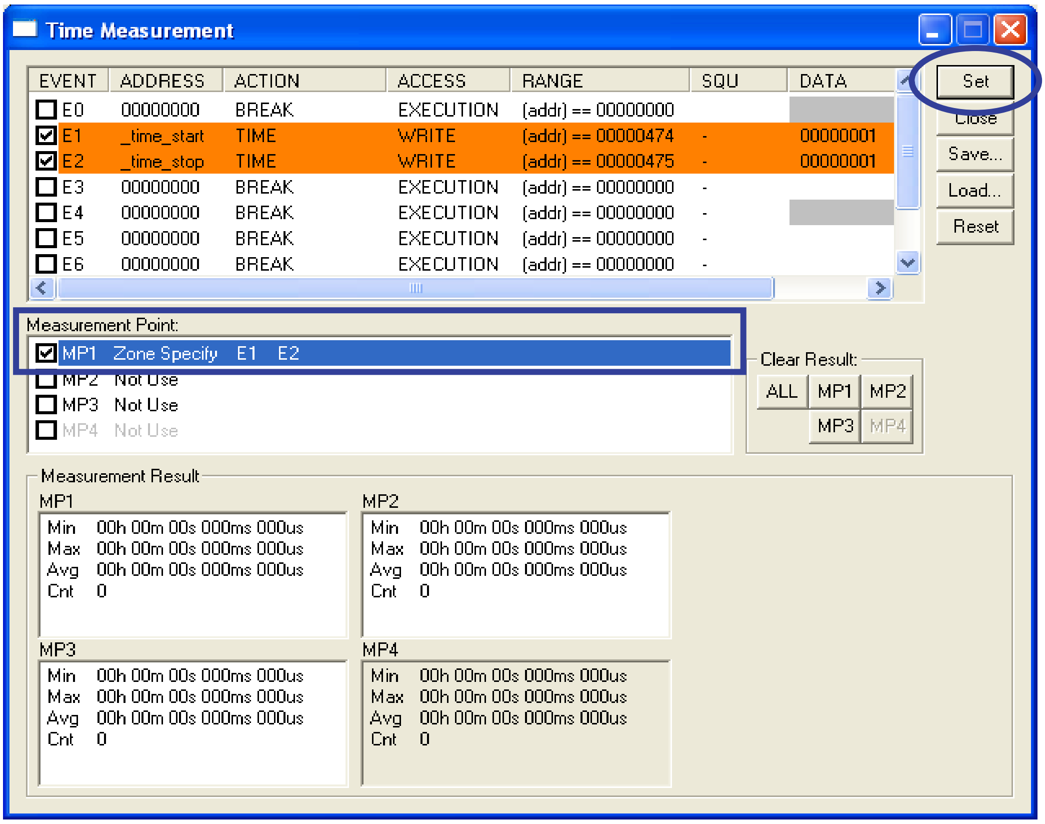
Start: E1

End: E2

Source File:

Function:

(8) The contents you've set are displayed in the Measurement Points column of the Section Time Measurement dialog box. Click the Set button to make the changes take effect.



Time Measurement

EVENT	ADDRESS	ACTION	ACCESS	RANGE	SQU	DATA
☐ E0	00000000	BREAK	EXECUTION	(addr) == 00000000		
☑ E1	_time_start	TIME	WRITE	(addr) == 00000474	-	00000001
☑ E2	_time_stop	TIME	WRITE	(addr) == 00000475	-	00000001
☐ E3	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
☐ E4	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
☐ E5	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
☐ E6	00000000	BREAK	EXECUTION	(addr) == 00000000	-	

Measurement Point:

☑ MP1 Zone Specify E1 E2
☐ MP2 Not Use
☐ MP3 Not Use
☐ MP4 Not Use

Measurement Result

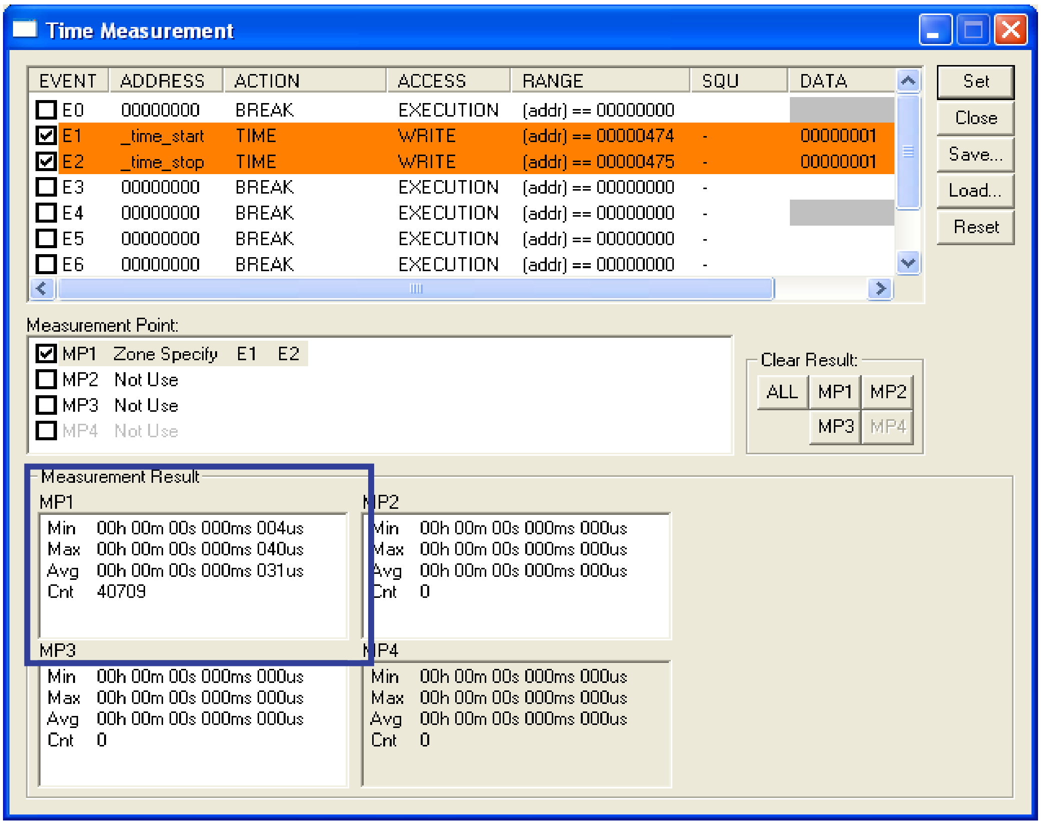
MP1
Min 00h 00m 00s 000ms 000us
Max 00h 00m 00s 000ms 000us
Avg 00h 00m 00s 000ms 000us
Cnt 0

MP2
Min 00h 00m 00s 000ms 000us
Max 00h 00m 00s 000ms 000us
Avg 00h 00m 00s 000ms 000us
Cnt 0

MP3
Min 00h 00m 00s 000ms 000us
Max 00h 00m 00s 000ms 000us
Avg 00h 00m 00s 000ms 000us
Cnt 0

MP4
Min 00h 00m 00s 000ms 000us
Max 00h 00m 00s 000ms 000us
Avg 00h 00m 00s 000ms 000us
Cnt 0

(9) Run the program and stop, and the result of a section time measurement will be displayed. That way, it is possible to measure the execution time of the sort function.



Time Measurement

EVENT	ADDRESS	ACTION	ACCESS	RANGE	SQU	DATA
□ E0	00000000	BREAK	EXECUTION	(addr) == 00000000		
☑ E1	_time_start	TIME	WRITE	(addr) == 00000474	-	00000001
☑ E2	_time_stop	TIME	WRITE	(addr) == 00000475	-	00000001
□ E3	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
□ E4	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
□ E5	00000000	BREAK	EXECUTION	(addr) == 00000000	-	
□ E6	00000000	BREAK	EXECUTION	(addr) == 00000000	-	

Measurement Point:

☑ MP1 Zone Specify E1 E2
□ MP2 Not Use
□ MP3 Not Use
□ MP4 Not Use

Measurement Result

```
MP1                                   MP2
Min  00h 00m 00s 000ms 004us          Min  00h 00m 00s 000ms 000us
Max  00h 00m 00s 000ms 040us          Max  00h 00m 00s 000ms 000us
Avg  00h 00m 00s 000ms 031us          Avg  00h 00m 00s 000ms 000us
Cnt  40709                            Cnt  0

MP3                                   MP4
Min  00h 00m 00s 000ms 000us          Min  00h 00m 00s 000ms 000us
Max  00h 00m 00s 000ms 000us          Max  00h 00m 00s 000ms 000us
Avg  00h 00m 00s 000ms 000us          Avg  00h 00m 00s 000ms 000us
Cnt  0                                Cnt  0
```

3. Frequency Asked Questions

(1) Are there any precautions to take when using the trace function and the section time measurement function?

[Precautions to taken when using the trace function]

- The trace function can only be used when the trace mode is selected on the operation mode tab of the Init dialog.
- In the Before and the After modes, the trace events that occurred in the trace rage, as well as the trace start events that occurred during that time, are recorded. Note, however, that trace end events are not recorded.
- In a conditional branch trace, display for the disassemble mode and source mode of the trace window may take time.
- If MCU execution priority is selected for the trace mode, the trace range becomes 512 cycles. If a trace range of 8M cycles is desired, select trace priority for the trace mode. In the trace priority mode, trace data output has priority and MCU execution is delayed. Program processing requires a longer timer than in the MCU execution priority mode. Note that for a conditional branch trace, the trace priority mode is the default mode of trace.
- Even when a function is specified when selecting the file to be displayed in the SRC mode, if the specified function exists in the currently displayed file, it is always the top of the file that is displayed.
- When a trace result is saved in text form, there is a possibility of the header and data strings getting their vertical positions displaced. Select “save in tab separated text” when saving a trace result and open it in spreadsheet software etc. to get correct display.
- Displaying each cycle in time (hours, minutes, and seconds) is not supported.
- The trace image files (*.rtt files) saved with the M3T-PDxx debugger cannot be loaded. Nor can the trace image files saved with other targets be loaded.
- The word data (16-bit) located beginning with an odd-numbered address or longword data (32-bit) not located at 4-byte boundaries cannot be traced normally.
- Data access events by DMA are not detected.
- If a runtime debug, RAM monitor, or data comparison break is used in a branch trace to record jump addresses, branch information (debug monitor program processing address) that is unlikely to occur in the user program may be included. So be careful.

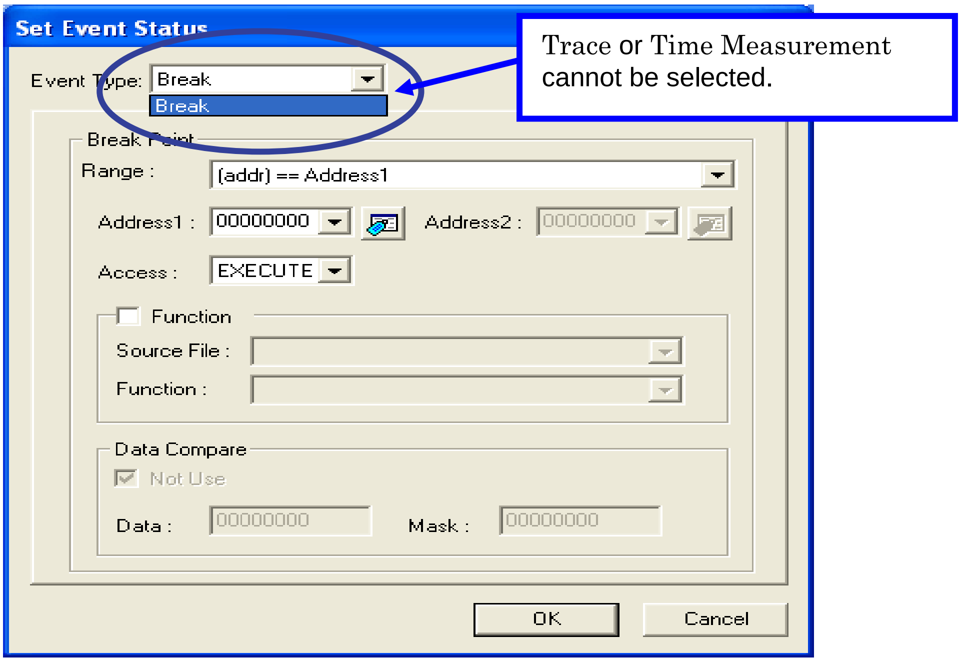
[Precautions to taken when using the section time measurement function]

- The event setting window and the section time measurement window share the same resource of the emulator. If an event is changed in the section time measurement window, the contents set in the event setting window are also changed.
- In a section time measurement performed, the data access trace information output from the MCU is compared in the emulator to detect the start and end events. If a large number of data access times is specified, trace output will take a lot of time, leading to errors in section time measurement or affecting execution of the MCU.

  For the case where Xin = 8 MHz, PLL = 128 MHz (× 16), and BASE = 64 MHz, for example, it takes 5 µs to output the data access trace information. If a data access of less than 5 µs occurs, a trace output time-dependent error will occur. Furthermore, if such a short data access occurs frequently, execution of the MCU will be delayed.

  We recommend write-only settings with least possible events.

(2) Why is it impossible to select Trace or Time Measurement for Event Type in the Set Event Status dialog box?



Set Event Status

Event Type: Break

Break Point

Range : (addr) == Address1

Address1 : 00000000 Address2 : 00000000

Access : EXECUTE

□ Function

Source File :

Function :

Data Compare

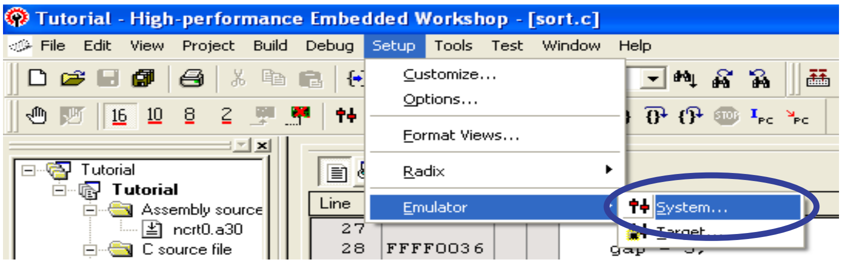
☑ Not Use

Data : 00000000 Mask : 00000000

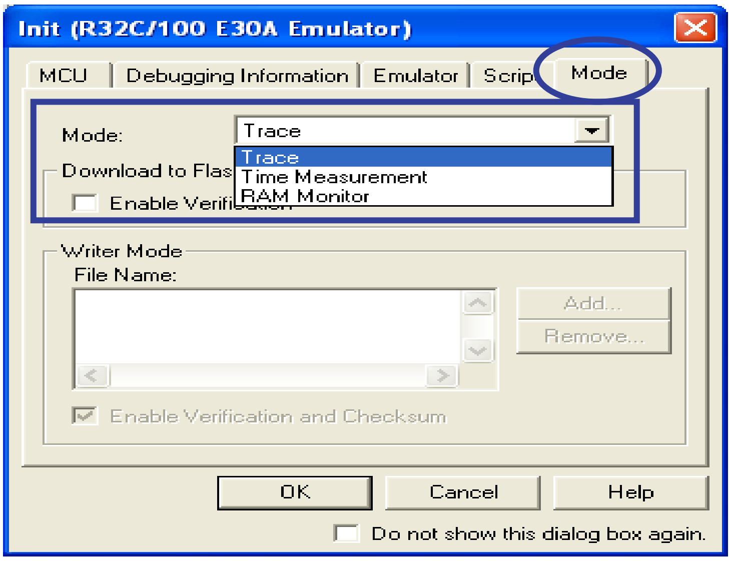
Trace or Time Measurement cannot be selected.

E30A uses the same internal resource of the emulator for trace, time measurement, and RAM monitor, so that to use the respective functions, it is necessary to change operation modes.

Choose Emulator from the Basic Settings menu of HEW and then select System to open the Init dialog box.



On the Operation Mode tab of the Init dialog box, select Trace to use the trace function or Time Measurement to use the section time measurement function.



Init (R32C/100 E30A Emulator)

Mode: Trace

Download to Flash

□ Enable Verification

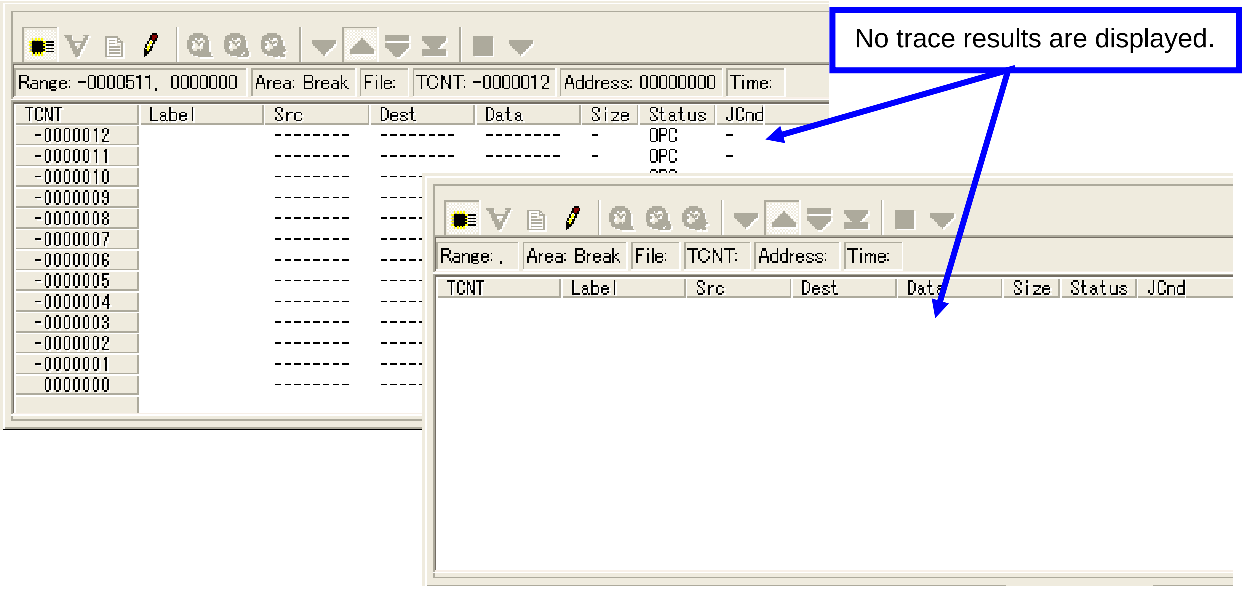
Writer Mode

File Name:

☑ Enable Verification and Checksum

□ Do not show this dialog box again.

(3) Although a trace was executed, no results are displayed in the trace window. Why?



```
Range: -0000511, 0000000  Area: Break  File:  TCNT: -0000012  Address: 00000000  Time:

TCNT       Label     Src        Dest       Data       Size  Status  JCnd
-0000012             --------   --------   --------   -     OPC     -
-0000011             --------   --------   --------   -     OPC     -
-0000010             --------   -----
-0000009             --------   -----
-0000008             --------   -----
-0000007             --------   -----
-0000006             --------   -----
-0000005             --------   -----
-0000004             --------   -----
-0000003             --------   -----
-0000002             --------   -----
-0000001             --------   -----
 0000000             --------   -----
```

```
Range: ,  Area: Break  File:  TCNT:  Address:  Time:

TCNT       Label     Src        Dest       Data       Size  Status  JCnd
```

No trace results are displayed.

This is because the operation mode is set to other than trace (time measurement or RAM monitor).

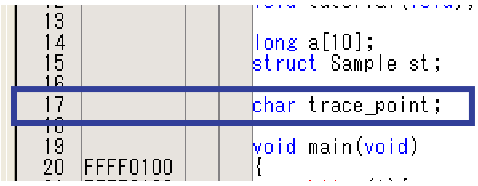
Check to see if Trace is selected on the Operation Mode tab of the Init dialog box.

(For details on how to set the operation mode, see paragraph (2) in this section.)

(4) Although a data access (write process) is embedded for the access condition of a trace event, the intended range cannot be traced. Why?

Example: To trace the sort function, embed a process for write to a global variable trace_point = 1 in front of the sort function and perform a trace with a data access (trace_point write process) specified as access condition.

Define the global variable trace_point.

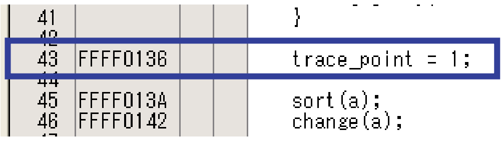


```
                  long a[10];
                  struct Sample st;

                  char trace_point;

                  void main(void)
         FFFF0100 {
```

Embed trace_point = 1 in front of the sort function and, after performing a build, download the program.

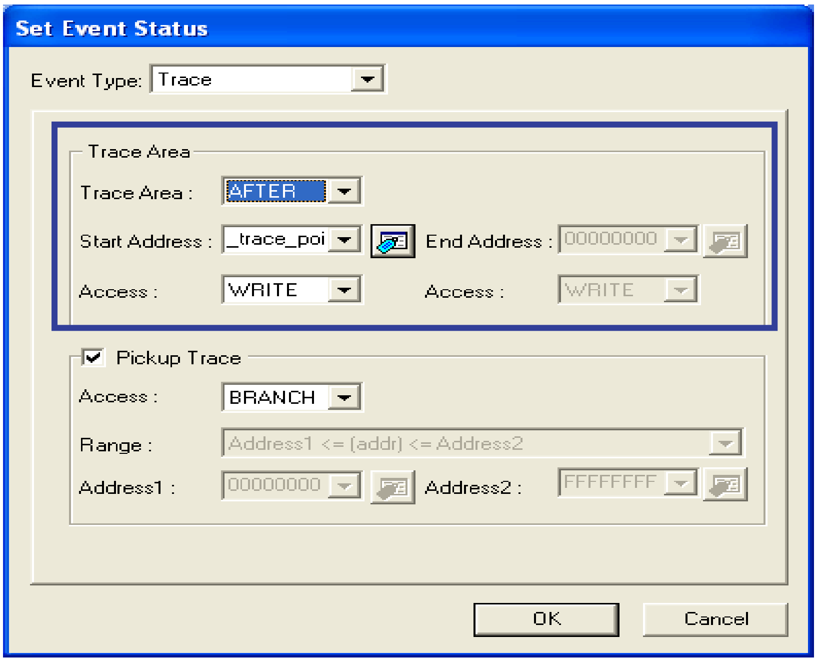


```
                      }

         FFFF0136     trace_point = 1;

         FFFF013A     sort(a);
         FFFF0142     change(a);
```

In the Set Event Status dialog box, set an event that will cause a trace to start upon WRITE to trace_point. And then perform a trace.



Set Event Status

Event Type: Trace

Trace Area

Trace Area : AFTER

Start Address : _trace_poi   End Address : 00000000

Access : WRITE   Access : WRITE

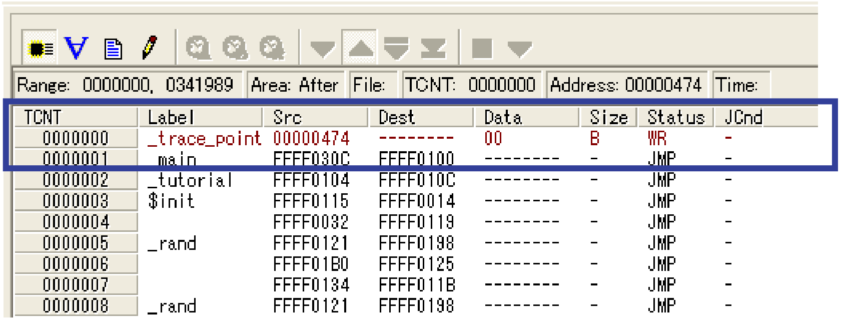
☑ Pickup Trace

Access : BRANCH

Range : Address1 <= (addr) <= Address2

Address1 : 00000000   Address2 : FFFFFFFF

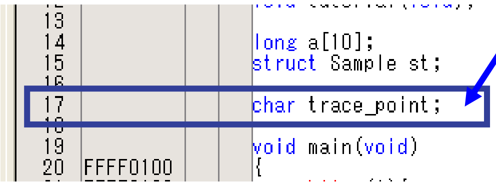
Check the trace result in the trace window after program execution, and it will be found that a trace was started by a process trace_point = 00, and that the main function, not the sort function, was traced.



Range: 0000000, 0341989 | Area: After | File: | TCNT: 0000000 | Address: 00000474 | Time:

TCNT	Label	Src	Dest	Data	Size	Status	JCnd
0000000	_trace_point	00000474	--------	00	B	WR	-
0000001	main	FFFF030C	FFFF0100	--------	-	JMP	-
0000002	_tutorial	FFFF0104	FFFF010C	--------	-	JMP	-
0000003	$init	FFFF0115	FFFF0014	--------	-	JMP	-
0000004		FFFF0032	FFFF0119	--------	-	JMP	-
0000005	_rand	FFFF0121	FFFF0198	--------	-	JMP	-
0000006		FFFF01B0	FFFF0125	--------	-	JMP	-
0000007		FFFF0134	FFFF011B	--------	-	JMP	-
0000008	_rand	FFFF0121	FFFF0198	--------	-	JMP	-

This is because initialization was performed (by writing “00”) at the time the global variable trace_point was declared, by which a trace was erroneously started.



```
                    void tutorial(void);

                    long a[10];
                    struct Sample st;

                    char trace_point;

                    void main(void)
FFFF0100            {
```

Initialized at here (by writing “00”)

To set a data access for the access condition of a trace event, we recommend using a read process for the condition.

4. Related Documents

The E30A emulator and HEW have many other useful functions than those presented in this application note. Along with this application note, please see the following related documents also, in which you'll find detailed specifications of each product, as well as technical information, limitations, and other helpful information.

[E30A emulator related documents]

- E30A Emulator User's Manual
- R32C/100 E30A Emulator Debugger V.1.00 User's Manual
- R32C/100 E30A Emulator Debugger V.1.00.00 Release Notes

[High-performance Embedded Workshop related documents]

- High-performance Embedded Workshop User's Manual
- High-performance Embedded Workshop Release Notes

[CPU related documents]

- R32C/118 Group Short Sheet
- R32C/100 Series Software Manual

[R32C/100 series C compiler package related documents]

- R32C/100 Series C Compiler Package V.1.01 C Compiler User's Manual
- R32C/100 Series C Compiler Package V.1.01 Release 00 Release Notes
- R32C/100 Series C Compiler Package V.1.01 Assembler User's Manual

For information on this product, visit the Renesas websites below.

Japan site:	http://japan.renesas.com/e30a
Global site:	http://www.renesas.com/e30a

Homepage and Where to Contact for Support

Renesas Technology homepage
http://www.renesas.com/

Where to contact
http://www.renesas.com/inquiry

Revision Record

Rev.	Issue date	Content of revision	
		Page	Points
1.00	2009.3.25	—	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.