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.
User’s Manual

AZ850 Ver. 3.30
System Performance Analyzer

Target Tool
RX850 Ver. 3.20
RX850 Pro Ver. 3.21
Windows, Windows NT, and Windows XP are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
Green Hills Software is a trademark of Green Hills Software, Inc.
The information in this document is current as of July, 2006. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.

- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC 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 NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

*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.

*Special*: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-criminal systems, safety equipment and medical equipment (not specifically designed for life support).

*Specific*: Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)
(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).
[MEMO]
INTRODUCTION

Target Readers
This manual is intended for users who wish to design and develop application systems using the V850 microcontrollers.

Purpose
This manual’s purpose is to help the user understand the functions and the operation method of the AZ850.

Organization
This manual is organized as follows.

- General
- Installation and startup
- AZ850 functions
- Constructing application program
- AZ monitor (soft trace only)
- Debugging with AZ850
- Window reference
- Error messages

How to Use this Manual
This manual assumes that the reader has general knowledge of microcontrollers, C language, assembly language, and debugging as well as basic knowledge of the Windows operation method.

To learn about the hardware functions and instruction functions of the V850 microcontrollers:
→ Read the user’s manual for each product.

“RTOS” is used as the representative product name in descriptions that are the same for the RX850 and the RX850 Pro.
If using the RX850, read “RTOS” as “RX850,” and if using the RX850 Pro, read “RTOS” as “RX850 Pro.”

Conventions
< >: Indicates a window or dialog box title.
[ ]: Indicates a menu.
text: Indicates a button in a window or dialog box.
<< >>: Indicates a character string displayed in a window or dialog box.

Data significance: Higher digits on the left and lower digits on the right
Memory map addresses: Higher addresses on the top and lower addresses on the bottom
Active low representation: xxx (overscore over pin or signal name)

Note:
Footnote for item marked with Note in the text
Caution:
Information requiring particular attention

Numerical representation:
Binary...XXXX or XXXXB
Decimal...XXXX
Hexadecimal...0XXXX

Prefix indicating power of 2 (address space, memory capacity)
K (Kilo) \(2^{10} = 1024\)
M (Mega) \(2^{20} = 1024^2\)
Related Documents

Please use the following documents in conjunction with this manual. The related documents listed below may include preliminary versions. However, preliminary versions are not marked as such.

### Documents Related to V850 microcontrollers Development Tools (User’s Manuals)

<table>
<thead>
<tr>
<th>Document Name</th>
<th>Document Number</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CA850 Ver. 3.00 C Compiler Package</strong></td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>U17293E</td>
</tr>
<tr>
<td>C Language</td>
<td>U17291E</td>
</tr>
<tr>
<td>Assembly Language</td>
<td>U17292E</td>
</tr>
<tr>
<td>Link Directives</td>
<td>U17294E</td>
</tr>
<tr>
<td><strong>ID850 Ver. 3.00 Integrated Debugger</strong></td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>U17358E</td>
</tr>
<tr>
<td><strong>ID850NW Ver. 3.00, 3.10 Integrated Debugger</strong></td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>U17369E</td>
</tr>
<tr>
<td><strong>ID850NWC Ver. 2.51 Integrated Debugger</strong></td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>U16525E</td>
</tr>
<tr>
<td><strong>ID850QB Ver. 3.20 Integrated Debugger</strong></td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>U17964E</td>
</tr>
<tr>
<td><strong>SM+ System Simulator</strong></td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>U18010E</td>
</tr>
<tr>
<td>User Open Interface</td>
<td>U18212E</td>
</tr>
<tr>
<td><strong>SM850 Ver. 2.50 System Simulator</strong></td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>U16218E</td>
</tr>
<tr>
<td><strong>SM850 Ver. 2.00 or later System Simulator</strong></td>
<td></td>
</tr>
<tr>
<td>External Part User Open Interface Specifications</td>
<td>U14873E</td>
</tr>
<tr>
<td><strong>RX850 Ver. 3.20 Real-Time OS</strong></td>
<td></td>
</tr>
<tr>
<td>Basics</td>
<td>U13430E</td>
</tr>
<tr>
<td>Installation</td>
<td>U17419E</td>
</tr>
<tr>
<td>Technical</td>
<td>U13431E</td>
</tr>
<tr>
<td>Task Debugger</td>
<td>U17420E</td>
</tr>
<tr>
<td><strong>RX850 Pro Ver. 3.21 Real-Time OS</strong></td>
<td></td>
</tr>
<tr>
<td>Basics</td>
<td>U18165E</td>
</tr>
<tr>
<td>In-Structure</td>
<td>U18164E</td>
</tr>
<tr>
<td>Task Debugger</td>
<td>U17422E</td>
</tr>
<tr>
<td><strong>AZ850 Ver. 3.30 System Performance Analyzer</strong></td>
<td></td>
</tr>
<tr>
<td>This manual</td>
<td></td>
</tr>
<tr>
<td><strong>PG-FP4 Flash Memory Programmer</strong></td>
<td></td>
</tr>
<tr>
<td>U15260E</td>
<td></td>
</tr>
<tr>
<td><strong>TW850 Ver. 2.00 Performance Analysis Tuning Tool</strong></td>
<td></td>
</tr>
<tr>
<td>U17421E</td>
<td></td>
</tr>
<tr>
<td><strong>PM+ Ver. 6.20 Project Manager</strong></td>
<td></td>
</tr>
<tr>
<td>U17990E</td>
<td></td>
</tr>
</tbody>
</table>
CONTENTS

CHAPTER 1 GENERAL ... 14

1. 1 Overview ... 14
1. 2 Functions and Features ... 15
1. 3 System configuration ... 16
   1. 3. 1 Using debug monitor ... 16
   1. 3. 2 Using ROM emulator ... 16
   1. 3. 3 Using in-circuit emulator ... 17
   1. 3. 4 Using simulator ... 17
1. 4 Operating Environment ... 18
1. 5 Resource ... 19
   1. 5. 1 Soft trace form ... 19
   1. 5. 2 Hard trace form ... 19

CHAPTER 2 INSTALLATION AND STARTUP ... 20

2. 1 Installing AZ850 ... 20
2. 2 Folder Configuration ... 20
   2. 2. 1 AZ850 system performance analyzer V3.xx components ... 21
   2. 2. 2 AZ850 monitor program (CA850 version) components ... 21
   2. 2. 3 AZ850 monitor program (GHS compiler version) components ... 22
   2. 2. 4 AZ850 (V3.xx) document components ... 22
2. 3 Starting and Exiting ... 23
   2. 3. 1 Starting ... 23
   2. 3. 2 Exiting ... 23
2. 4 Uninstalling AZ850 ... 23

CHAPTER 3 AZ850 FUNCTIONS ... 24

3. 1 Soft Trace Form and Hard Trace Form ... 24
3. 2 About AZ Trace Data ... 25
   3. 2. 1 AZ trace data detection contents ... 25
   3. 2. 2 About trace buffer ... 25
   3. 2. 3 Regarding time accuracy ... 26
   3. 2. 4 Guide to time trace data can be obtained ... 26
3. 3 Debugging Possible with AZ850 ... 27

CHAPTER 4 CONSTRUCTING APPLICATION PROGRAM ... 29

4. 1 Using the AZ850 in Soft Trace Form ... 29
4. 2 Using the AZ850 in Hard Trace Form ... 32
CHAPTER 5  AZ MONITOR (SOFT TRACE ONLY) ... 33

5.1 About AZ Monitor ... 33
5.2 Creating AZ Monitor ... 34
   5.2.1 Creating user own coding block ... 34
   5.2.2 Initializing AZ monitor ... 36

CHAPTER 6 DEBUGGING WITH AZ850 ... 37

6.1 Debugging Procedure Using AZ850 ... 37

CHAPTER 7 WINDOW REFERENCE ... 43

7.1 Outline of Windows and Dialog Boxes of AZ850 ... 43
7.2 Relationship of AZ850 Windows and Dialog Boxes ... 44
7.3 Explanation of Windows and Dialog Boxes ... 45
   Main Window ... 46
   [AZ Option] dialog box ... 51
   [Open]/[Save As] dialog box ... 54
   Analyze Window ... 57
   [Object Select] dialog box ... 74
   [Pattern Search] dialog box ... 77
   Cpu Window ... 81
   [Pattern Set] dialog box ... 87
   Pattern Window ... 92
   Trace View Window ... 98
   [Trace Search] dialog box ... 103
   [About] dialog box ... 106

CHAPTER 8 ERROR MESSAGES ... 107

INDEX ... 109
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure No.</th>
<th>Title and Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>System Configuration (When Using Debug Monitor) ... 16</td>
</tr>
<tr>
<td>1-2</td>
<td>System Configuration (When Using ROM Emulator) ... 16</td>
</tr>
<tr>
<td>1-3</td>
<td>System Configuration (When Using In-Circuit Emulator) ... 17</td>
</tr>
<tr>
<td>1-4</td>
<td>System Configuration (When Using Simulator) ... 17</td>
</tr>
<tr>
<td>2-1</td>
<td>Startup Screen: Main Window ... 23</td>
</tr>
<tr>
<td>4-1</td>
<td>Example of AZ Monitor Section Definition for RX850 (When Using CA850) ... 30</td>
</tr>
<tr>
<td>4-2</td>
<td>Example of AZ Monitor Section Definition for RX850 (When Using GHS's Compiler) ... 30</td>
</tr>
<tr>
<td>4-3</td>
<td>Example of AZ Monitor Section Definition for RX850 Pro (When Using CA850) ... 30</td>
</tr>
<tr>
<td>4-4</td>
<td>Example of AZ Monitor Section Definition for RX850 Pro (When Using GHS's Compiler) ... 31</td>
</tr>
<tr>
<td>5-1</td>
<td>Example of Timer Counter Operation (In Case of Up Counter) ... 34</td>
</tr>
<tr>
<td>5-2</td>
<td>Example of Use Own Coding Block ... 35</td>
</tr>
<tr>
<td>5-3</td>
<td>Example of AZ Monitor Initialization Description (When Using RX850) ... 36</td>
</tr>
<tr>
<td>6-1</td>
<td>Checking on Main Window ... 37</td>
</tr>
<tr>
<td>6-2</td>
<td>Setting in [AZ Option] Dialog Box ... 38</td>
</tr>
<tr>
<td>6-3</td>
<td>AZ Trace Mode Switching ... 39</td>
</tr>
<tr>
<td>6-4</td>
<td>Uploading of Trace Data ... 39</td>
</tr>
<tr>
<td>6-5</td>
<td>Display Example of the Analyze Window ... 40</td>
</tr>
<tr>
<td>6-6</td>
<td>Display Example of Cpu Window ... 41</td>
</tr>
<tr>
<td>6-7</td>
<td>Display Example of Pattern Window ... 41</td>
</tr>
<tr>
<td>6-8</td>
<td>Display Example of Trace View Window ... 42</td>
</tr>
<tr>
<td>7-1</td>
<td>Relationship of AZ850 Windows and Dialog Boxes ... 44</td>
</tr>
<tr>
<td>7-2</td>
<td>Main Window ... 46</td>
</tr>
<tr>
<td>7-3</td>
<td>Status Bar of Main Window ... 49</td>
</tr>
<tr>
<td>7-4</td>
<td>[AZ Option] Dialog Box ... 51</td>
</tr>
<tr>
<td>7-5</td>
<td>[Open]/[Save As] Dialog Box ... 54</td>
</tr>
<tr>
<td>7-6</td>
<td>Analyze Window ... 57</td>
</tr>
<tr>
<td>7-7</td>
<td>Analyze Window (Simple Mode) ... 67</td>
</tr>
<tr>
<td>7-8</td>
<td>Analyze Window (Standard Mode) ... 67</td>
</tr>
<tr>
<td>7-9</td>
<td>Analyze Window (Detailed Mode) ... 68</td>
</tr>
<tr>
<td>7-10</td>
<td>Example of Transition Map When Equal Mode Is Not Specified ... 71</td>
</tr>
<tr>
<td>7-11</td>
<td>Example of Transition Map When Equal Mode Is Specified ... 71</td>
</tr>
<tr>
<td>7-12</td>
<td>[Object Select] Dialog Box ... 74</td>
</tr>
<tr>
<td>7-13</td>
<td>[Pattern Search] Dialog Box ... 77</td>
</tr>
<tr>
<td>7-14</td>
<td>Cpu Window ... 81</td>
</tr>
<tr>
<td>7-15</td>
<td>[Pattern Set] Dialog Box ... 87</td>
</tr>
<tr>
<td>7-16</td>
<td>Difference between [Valid] and [Invalid] ... 89</td>
</tr>
<tr>
<td>7-17</td>
<td>Pattern Window ... 92</td>
</tr>
<tr>
<td>7-18</td>
<td>Pattern Distribution Viewing Method ... 97</td>
</tr>
<tr>
<td>7-19</td>
<td>Trace View Window ... 98</td>
</tr>
<tr>
<td>7-20</td>
<td>AZ Trace Data Viewing Method ... 101</td>
</tr>
<tr>
<td>7-21</td>
<td>[Trace Search] Dialog Box ... 103</td>
</tr>
</tbody>
</table>
7-22  Example of Trace Search Setting 1 ... 105
7-23  Example of Trace Search Setting 2 ... 105
7-24  Example of Trace Search Setting 3 ... 105
7-25  [About] Dialog Box ... 106
8-1   [Error] Dialog Box ... 107
<table>
<thead>
<tr>
<th>Table No.</th>
<th>Title and Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1</td>
<td>Relationship Between Trace Form and Debugging Environment ... 24</td>
</tr>
<tr>
<td>3-2</td>
<td>Detectable Contents as AZ Trace Data ... 25</td>
</tr>
<tr>
<td>3-3</td>
<td>Trace Buffer Area and Timing of Clearing the Area ... 25</td>
</tr>
<tr>
<td>5-1</td>
<td>Data/Functions of User Own Coding Block for AZ Monitor ... 34</td>
</tr>
<tr>
<td>7-1</td>
<td>List of Windows and Dialog Boxes of AZ850 ... 43</td>
</tr>
<tr>
<td>7-2</td>
<td>Tool Bar of Main Window ... 48</td>
</tr>
<tr>
<td>7-3</td>
<td>Function Buttons of the [AZ Option] Dialog Box ... 53</td>
</tr>
<tr>
<td>7-4</td>
<td>Function Buttons of the [Open]/[Save As] Dialog Box ... 55</td>
</tr>
<tr>
<td>7-5</td>
<td>Display File Extensions and Formats ... 55</td>
</tr>
<tr>
<td>7-6</td>
<td>Tool Bar of the Analyze Window ... 64</td>
</tr>
<tr>
<td>7-7</td>
<td>List of the Marks in the Analyze Window (RX850) ... 69</td>
</tr>
<tr>
<td>7-8</td>
<td>List of the Marks in the Analyze Window (RX850 Pro) ... 70</td>
</tr>
<tr>
<td>7-9</td>
<td>Function Buttons of the [Object Select] Dialog Box ... 75</td>
</tr>
<tr>
<td>7-10</td>
<td>Sort Mode for Objects ... 76</td>
</tr>
<tr>
<td>7-11</td>
<td>Search Modes ([Pattern Search] Dialog Box) ... 78</td>
</tr>
<tr>
<td>7-12</td>
<td>Search Conditions ([Pattern Search] Dialog Box: When [Task Switch] Is Selected) ... 78</td>
</tr>
<tr>
<td>7-13</td>
<td>Search Conditions ([Pattern Search] Dialog Box: When [System Call(xxxx)] Is Selected) ... 79</td>
</tr>
<tr>
<td>7-14</td>
<td>Function Buttons of the [Pattern Search] Dialog Box ... 80</td>
</tr>
<tr>
<td>7-15</td>
<td>Tool Bar of Cpu Window ... 85</td>
</tr>
<tr>
<td>7-16</td>
<td>Pattern Modes ([Pattern Set] Dialog Box) ... 88</td>
</tr>
<tr>
<td>7-17</td>
<td>Pattern Conditions ([Pattern Set] Dialog Box: When [Task Switch] Is Selected) ... 88</td>
</tr>
<tr>
<td>7-18</td>
<td>Search Conditions ([Pattern Set] Dialog Box: When [System Call (xxxx)] Is Selected) ... 88</td>
</tr>
<tr>
<td>7-19</td>
<td>Function Buttons of the [Pattern Set] Dialog Box ... 90</td>
</tr>
<tr>
<td>7-20</td>
<td>Tool Bar of the Pattern Window ... 96</td>
</tr>
<tr>
<td>7-21</td>
<td>Types of Events In AZ Trace Data ... 99</td>
</tr>
<tr>
<td>7-22</td>
<td>Function Buttons of the [Trace Search] Dialog Box ... 104</td>
</tr>
<tr>
<td>7-23</td>
<td>Function Buttons of the [About] Dialog Box ... 106</td>
</tr>
</tbody>
</table>
1.1 Overview

As the performance of microprocessors has increased, application programs have grown in scale and complexity. With conventional debuggers, theoretical debugging of such application programs is simple, but time-related analysis is not. It is difficult and takes a very long time, for example, to analyze errors such as those caused by incorrect processing timing, or to evaluate the performance of the entire system.

To solve these problems, NEC Electronics Corporation has developed powerful microprocessors such as the V850 Microcontrollers. NEC Electronics Corporation has also developed and released “System Performance Analyzer: AZ850”, to support the quantitative performance analysis of application programs.

The AZ850 is a performance analysis tool for analyzing the execution transition statuses and the execution efficiency of the CPU for application programs that embed the RX850/RX850 Pro real-time operating system for the V850 Microcontrollers.

The AZ850 has a function for tracing the occurrence of events (issue of a system call, occurrence of an interrupt, etc.) and presenting the trace data graphically, in cooperation with a debugger. By using this function, the execution transition states of tasks, access states to objects such as an acquisition/release of resource, and execution time for a task can be easily analyzed.

The specifications of the AZ850 conform to the Tool Interface Protocol (TIP), and as long as a debugger (even one not made by NEC Electronics Corporation) that supports this interface is used, the functions of the AZ850 can easily be used.
1. 2  Functions and Features

The major functions and features of the AZ850 are described below.

- **Graphical display of the task execution transition statuses**
  The graphically displayed the task/interrupt execution transition status of application program in which the RX850/RX850 Pro is embedded (horizontal axis = time, vertical axis = task name, etc.) make it easy to analyze system status changes caused by task switching, the occurrence of interrupts, etc. Since all accesses to an object caused by system call are marked, the order of task execution and status transition can easily be understood.

- **Graphical display of CPU occupation time**
  Based on the display of the CPU usage within a specified time range, the execution efficiency of the entire system can be estimated.

- **Statistical analysis of processing time**
  The time required for an application program to execute a certain operation is calculated and the result is displayed as a histogram. The worst, average, and other values are also displayed.

- **Linked operation with debugger window**
  A jump can be made from the AZ850 window to debugger's windows displaying the source text, disassemble text, and memory contents. This makes it easy to identify any problem.

The AZ850 supports the following two trace forms. Either of these trace forms can be selected for the user’s debugging environment. Refer to the "3. 1 Soft Trace Form and Hard Trace Form" for details on the trace forms.

- **Soft trace form**
  A monitor function is provided on the target system, and the monitor program collects AZ trace data.
  The monitor program must be created for use in the user’s environment. Refer to the "CHAPTER 5 AZ MONITOR (SOFT TRACE ONLY)" for details on how to create the monitor program.

- **Hard trace form**
  AZ trace data is collected using the trace function of an in-circuit emulator or simulator.
  AZ trace data can be collected without modifying the application program.
1.3 System configuration

The AZ850 extends the function of the debugger by communicating with it on the TIP specification. Various AZ850 system configurations are shown for the debugging environments that can be used. The trace form that can be used differs depending on the debug environment used. Refer to the "3.1 Soft Trace Form and Hard Trace Form" for details.

1.3.1 Using debug monitor

[Caution] One unused timer counter is required on the target system.

1.3.2 Using ROM emulator

[Caution] One unused timer counter is required on the target system.
1.3.3 Using in-circuit emulator

Figure 1-3 System Configuration (When Using In-Circuit Emulator)

[Caution] When using the AZ850 with the soft trace form, the AZ monitor must be linked to the load module. Moreover, one unused timer counter is required on the target system.

1.3.4 Using simulator

Figure 1-4 System Configuration (When Using Simulator)

[Caution] When using the AZ850 with the soft trace form, the AZ monitor must be linked to the load module. Moreover, one unused timer counter is required on the target system.
1. 4 Operating Environment

The AZ850 requires an environment in which a debugger can operate.

(1) Host machine

One of the following OSs must be operated.

Windows XP Home Edition, Windows XP Professional

[Caution] It is recommended that the newest Service Pack be installed in any of the above OSs.

(2) Software

C compiler package

<table>
<thead>
<tr>
<th>CA850 (NEC Electronics Corporation : Version 3.00 or later)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCV850/CCV850E (Green Hills® Software, Inc., USA : Multi®2000 Version 2000/3.5 Release 6.5.3)</td>
</tr>
<tr>
<td>CCV850/CCV850E (Green Hills Software, Inc., USA : Multi2000 Version 2000/4.0 Release 7.0.1)</td>
</tr>
</tbody>
</table>

Real-time OS

<table>
<thead>
<tr>
<th>RX850 (uITRON3.0 : Version 3.20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX850 Pro (uITRON3.0 : Version 3.20)</td>
</tr>
</tbody>
</table>

Debugger or Simulator

<table>
<thead>
<tr>
<th>ID850 (NEC Electronics Corporation : Version 3.00 or later)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID850NW (NEC Electronics Corporation : Version 3.00 or later)</td>
</tr>
<tr>
<td>ID850QB (NEC Electronics Corporation : Version 3.00 or later)</td>
</tr>
<tr>
<td>SM850 (NEC Electronics Corporation : Version 3.00 or later)</td>
</tr>
<tr>
<td>PARTNER-ETII V850E/Win (Kyoto Micro Computer Co., Ltd. : Version 2.83a or later)</td>
</tr>
<tr>
<td>850EServ (Green Hills Software, Inc., USA : Multi2000 Version 2000/3.5 Release 6.5.3)</td>
</tr>
<tr>
<td>RTEServ (Green Hills Software, Inc., USA : Multi2000 Version 2000/3.5 Release 6.5.3)</td>
</tr>
<tr>
<td>850EServ (Green Hills Software, Inc., USA : Multi2000 Version 2000/4.0 Release 7.0.1)</td>
</tr>
<tr>
<td>RTEServ (Green Hills Software, Inc., USA : Multi2000 Version 2000/4.0 Release 7.0.1)</td>
</tr>
</tbody>
</table>

[Caution] Any of the above debugger or simulator must support TIP.

(3) ROM emulator

ROM emulator that can be connected to TIP debugger

(4) In-circuit emulator

In-circuit emulator that can be connected to TIP debugger
1. 5 Resource

This section describes the resource which is used by the AZ850 for the debugger and target system.

1. 5. 1 Soft trace form

In the soft trace form, the AZ850 is embedded into the application system and a trace buffer that stores trace data is located on the memory of the target system.

Therefore, the AZ850 uses the following target system memory.

To collect trace data, a timer counter for the AZ850 is required to obtain time information from the timer counter on the target system.

| Text/data area of the AZ850 (TEXT attribute) | For RX850: Approx. 800 bytes For RX850 Pro: Approx. 1200 bytes |
| Work area of the AZ850 (BSS attribute) | For RX850: 44 bytes For RX850 Pro: 64 bytes |
| Trace buffer area | 4K bytes to 4M bytes (64K bytes to 1M byte recommended) The size of this area can be changed with the [AZ Option] dialog box. |
| Timer counter | 1 |

1. 5. 2 Hard trace form

The AZ850 uses the following trace conditions on the debugger to collect the trace data.

Care is required regarding the number of trace conditions that the debugger can use. If the resource for the AZ850 is insufficient, it may not be possible to set the AZ trace mode.

| Trace condition | Qualify trace condition (for write-access) : 1 |
2.1 Installing AZ850

The procedure to install the AZ850 is as follows:

(1) Start up Windows.

(2) If the supply medium is CD-ROM
    Insert the CD-ROM into CD drive. The setup program is automatically started. At this time, if the startup program is not started, manually start it by starting Windows Explorer and double-click "Install.exe" icon in the CD drive.
    If the supply medium is ODS
    Download the install program (az703000_v3xx.exe) which is the self-extracting file from ODS. Then, double-click the install program.

(3) Perform installation according to the message displayed.
    Select either the component of the monitor program for CA850 or that for GHS compiler, or the components for both, in accordance with the environment to be used.

[Caution] Once the AZ850 has been installed once, to install it again, it is necessary to uninstall it first.

2.2 Folder Configuration

This section describes the folders and the files to be copied as a result of installing the AZ850.

The following components are copied:

- AZ850 system performance analyzer V3.xx components
- AZ850 monitor program (CA850 version) components
- AZ850 monitor program (GHS compiler version) components
- AZ850 (V3.xx) document components
2.2.1 AZ850 system performance analyzer V3.xx components

In addition to the above, the following library files are copied.

- When using Windows98 or WindowsMe

  Windows system folder\
  tipxdbg.dll     - Library file of TIP processing for the AZ850 and RD850
  tipdbg.dll     - Library file of TIP processing for the debugger
  tipcmm.dll     - Library file of TIP processing for common

- When using WindowsNT, Windows2000 or WindowsXP

  Windows system32 folder\
  tipxdbg.dll     - Library file of TIP processing for the AZ850 and RD850
  tipdbg.dll     - Library file of TIP processing for the debugger
  tipcmm.dll     - Library file of TIP processing for common

2.2.2 AZ850 monitor program (CA850 version) components

In addition to the above, the following library files are copied.

- When using Windows98 or WindowsMe

  Windows system folder\
  tipxdbg.dll     - Library file of TIP processing for the AZ850 and RD850
  tipdbg.dll     - Library file of TIP processing for the debugger
  tipcmm.dll     - Library file of TIP processing for common

- When using WindowsNT, Windows2000 or WindowsXP

  Windows system32 folder\
  tipxdbg.dll     - Library file of TIP processing for the AZ850 and RD850
  tipdbg.dll     - Library file of TIP processing for the debugger
  tipcmm.dll     - Library file of TIP processing for common

2.2.2 AZ850 monitor program (CA850 version) components

In addition to the above, the following library files are copied.

- When using Windows98 or WindowsMe

  Windows system folder\
  tipxdbg.dll     - Library file of TIP processing for the AZ850 and RD850
  tipdbg.dll     - Library file of TIP processing for the debugger
  tipcmm.dll     - Library file of TIP processing for common

- When using WindowsNT, Windows2000 or WindowsXP

  Windows system32 folder\
2. 2. 3 AZ850 monitor program (GHS compiler version) components

Installation folder (default : C:\Program Files\NEC Electronics Tools\AZ850\V3.xx)

```
lib850_ghs\ [Note1]  r22\  azcrc.o  . . . . Core of AZ monitor (for RX850)
  r26\  azcrc.o  . . . . Core of AZ monitor (for RX850)
  r32\  azcrc.o  . . . . . Core of AZ monitor (for RX850)

smp850_ghs\ [Note2]  azusr_v853.850  . . User own coding block of AZ monitor
  (Sample: for V853)
  azusr_MS1.850  . . User own coding block of AZ monitor
  (Sample: for V850E/MS1)
  azusr_MA1.850  . . User own coding block of AZ monitor
  (Sample: for V850E/MA1)
  azusr_SG2.850  . . User own coding block of AZ monitor
  (Sample: for V850ES/SG2)
```

[Note1] "*.o" file in the lib850 or lib850_ghs folder is used in common with V850/V850E1/V850E2/V850ES.
[Note2] smp850 folder or smp850_ghs folder is a sample folder for V850/V850E1/V850E2/V850ES.

2. 2. 4 AZ850 (V3.xx) document components

Installation folder (default : C:\Program Files\NEC Electronics Tools\AZ850\V3.xx)

```
doc\  ZUD-CD-xx-xxxx.pdf  . . . . . Modifications from preceding version, restrictions, and cautions
    Uxxxxxxxxxxxxxxxx.pdf  . . . User’s manual
```

User’s Manual U17423EJ2V0UM
2. 3  Starting and Exiting

2. 3. 1  Starting

Since the AZ850 operates by communicating with and obtaining information from a debugger, the debugger must operate at the same time.

- Starting form Windows start menu

- Starting from PM+ (when using PM+)
  The procedure to start the AZ850 is as follows:
  
  (1) Open a workspace file (*.prw) after starting the PM+.

  (2) Select the [Tool] menu -> [Start up AZ850] on the Main Window of the PM+.

  [Remark] The version of the AZ850 to be used can be selected with the Project Settings dialog box of the PM+. Refer to the PM+ user's manual for details.

The following Main Window will be displayed after the AZ850 is started.

![Startup Screen: Main Window](image)

2. 3. 2  Exiting

To exit the AZ850, select the [File] menu -> [Exit] or the \[X\] button on the Main Window.

2. 4  Uninstalling AZ850

The procedure to uninstall the AZ850 is as follows:

(1) Start up Windows.

(2) Activate "Add/Remove Application" ("Add or Remove Programs" in the case of WindowsXP) in the Control Panel.

(3) Select the item to be uninstalled ("NEC EL AZ850 V3.xx"), then click [Remove] button.

(4) Perform uninstallation according to the message displayed.
CHAPTER 3    AZ850 FUNCTIONS

3. 1 Soft Trace Form and Hard Trace Form

The AZ850 supports the following two trace forms.
Either of these trace forms can be selected for the user’s debugging environment in the [AZ Option] dialog box.

- Soft trace form
  The soft trace form provides a monitor function on the target system, and collects AZ trace data using a monitor program. AZ trace data is collected by linking the AZ monitor (monitor program) to the application program.
  In the soft trace form, AZ trace data is stored into the user memory area which is specified in the [AZ Option] dialog box.
  The monitor program must be created according to the user’s environment. For details on how to create the monitor program, refer to the "CHAPTER 5  AZ MONITOR (SOFT TRACE ONLY)".

- Hard trace form
  The hard trace form uses the trace function of an in-circuit emulator or simulator to collect AZ trace data. AZ trace data can be collected without adding monitor or other programs to the application program.
  In the hard trace form, AZ trace data is stored into a trace buffer area of the debugger.

The trace form that can be used differs depending on the user’s debugging environment.

Table 3-1  Relationship Between Trace Form and Debugging Environment

<table>
<thead>
<tr>
<th>Debugging Environment</th>
<th>Soft Trace Form</th>
<th>Hard Trace Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug monitor</td>
<td>OK</td>
<td>-</td>
</tr>
<tr>
<td>ROM emulator</td>
<td>OK</td>
<td>-</td>
</tr>
<tr>
<td>In-circuit emulator</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Simulator</td>
<td>OK</td>
<td>OK</td>
</tr>
</tbody>
</table>
3. 2 About AZ Trace Data

3. 2. 1 AZ trace data detection contents

The locations that can be detected as AZ trace data and their contents are described below.

Table 3-2 Detectable Contents as AZ Trace Data

<table>
<thead>
<tr>
<th>Detected Location</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>System call entry</td>
<td>Time&lt;br&gt;Number of a system call&lt;br&gt;Target object ID of a system call&lt;br&gt;Issue address of a system call</td>
</tr>
<tr>
<td>System call exit</td>
<td>Time&lt;br&gt;Return value of a system call</td>
</tr>
<tr>
<td>Interrupt entry</td>
<td>Time&lt;br&gt;Exception code&lt;br&gt;Occurrence address of a interrupt</td>
</tr>
<tr>
<td>Interrupt exit</td>
<td>Time</td>
</tr>
<tr>
<td>Task switch</td>
<td>Time&lt;br&gt;ID of the task to which operation will shift (or ID that indicates Idle status)</td>
</tr>
<tr>
<td>Task starter</td>
<td>Time&lt;br&gt;Entry address of a task</td>
</tr>
</tbody>
</table>

Note that the following items cannot be detected as AZ trace data.
- RESET, NMI and exceptions (software exception, exception trap)
- Interrupts that have not been registered into the RX850/RX850 Pro
- Start and end of a cyclic startup handler
- Issue address of a system call that is ext_tsk, exd_tsk (only for RX850 Pro), ret_int or ret_wup

3. 2. 2 About trace buffer

The trace buffer area that is used for collecting AZ trace data, and the timing of clearing the area are shown in the table below.

Table 3-3 Trace Buffer Area and Timing of Clearing the Area

<table>
<thead>
<tr>
<th>Trace Form</th>
<th>Trace Buffer Area</th>
<th>Timing of Clearing the Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soft trace form</td>
<td>Arbitrarily specified in trace buffer area specification area in the [AZ Option] dialog box</td>
<td>If AZ trace mode is set to on again after AZ trace mode was switched off</td>
</tr>
<tr>
<td>Hard trace form</td>
<td>Trace memory of the debugger</td>
<td>Every start of an application program</td>
</tr>
</tbody>
</table>
3. 2. 3  Regarding time accuracy

- Soft trace form
  Because time information is obtained from a timer counter on the target system, the time information of the AZ850 accords with the operation of the timer counter. How to control the timer counter used with the AZ850 and read its value is explained in the description of the user own coding block of the AZ monitor. Refer to the "5. 2. 1 Creating user own coding block" for details.

- Hard trace form
  The unit of the time information displayed as AZ trace data is the same as that of the time tag of the trace data in the trace function of the debugger. Also note that the time information may not be correct when the system clock is set to either STOP mode or IDLE mode (The task execution time is calculated from the CPU system clock.).

[Caution] When using the AZ850 in the hard trace form with the ID850, clear the [ADD Up Timetag] check box in the ID850 Extended Option dialog box. Unless this check box is cleared, the correct time information cannot be obtained.

[Remark] Some in-circuit emulators do not support the timetag of trace data. In this case, the transition status of the task can be checked by selecting the "equal mode" as the display mode of the Analyze Window.

3. 2. 4  Guide to time trace data can be obtained

A rough guide to the time AZ trace data can be obtained is shown in the table below. The time shown in this table may vary significantly depending on the configuration of the application program and the operating speed of the CPU.

<table>
<thead>
<tr>
<th>Assumed Configuration of Application</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency of system call occurrence</td>
<td>Once every 1000 instructions</td>
</tr>
<tr>
<td>Frequency of task switch occurrence</td>
<td>Once every 5 system calls</td>
</tr>
<tr>
<td>Frequency of interrupt occurrence</td>
<td>Once every 1 [msec]</td>
</tr>
</tbody>
</table>

- Soft trace form
  If 64K bytes of the memory on the target system are used, AZ trace data of about 100 [msec] can be obtained.

- Hard trace form
  If all the 32K frames of the trace memory of the in-circuit emulator are used, AZ trace data of about 300 [msec] can be obtained.
3. 3 Debugging Possible with AZ850

This section describes whether it is available of what debugging by using the AZ850 functions. For details about the procedure to operate, refer to the window/dialog box descriptions.

Verification using the Analyze Window

This window displays the collected AZ trace data using execution transition map. Using this window and each of the windows that can be opened from this window, system timing misses as well as the entire system itself can be checked.

- Checking the system call issue status
  In addition to execution transition map showing the system flow with the task name/interrupt name as the vertical axis, the system calls supplied by the real-time OS are shown using various marks. Furthermore, it is possible to select specific locations within AZ trace data by opening the Trace View Window from any marked point.

- Checking object operation
  The operation of specified objects (tasks/interrupts, etc.) can be searched on execution transition map.

- Searching the occurrence location of specific events
  The "An Event" set in the [Pattern Set] dialog box can be searched on execution transition map. Searching the switching location for particular tasks, locations where an interrupt has occurred, and locations where a particular system call has been issued can easily be done by using this function.

- Jumping to debugger window from a given location
  The windows displaying source text, disassemble text or memory list corresponding to a task/interrupt specified with up temporary cursor in the execution transition map can be opened on the debugger.

Verification using the Pattern Window

The pattern (particular processing) set with the [Pattern Set] dialog box is searched in the execution transition map and the distribution status of that pattern is displayed in this window. It is possible to obtain the worst execution time value and average execution time value for application program processing from this window.

- Histogram display of task/interrupt processing time
  The tabulated results for the searched pattern are displayed in histogram from with the execution time interval as a parameter.

- Search using task/interrupt processing time
  Pattern locations where the execution time interval is maximum or minimum are searched in the processing time of the searched pattern and the results of this search are displayed in the execution transition map. This enables the easy search of locations where particular patterns are processed.
CHAPTER 3   AZ850 FUNCTIONS

Verification using the Cpu Window

This window displays the CPU usage within the specified period in the execution transition map. From this window, it is possible to check the execution time of given tasks/interrupts and evaluate the performance of the target data from the idle time and interrupt time.

- Check of execution time of given task/interrupt, or check of CPU usage percentage
- Check of task execution time percentages within time interval for which data is collected
- Check of system execution time percentages within time interval for which data is collected

Verification using the Trace View Window

This window displays in list form the collected AZ trace data contents.

- Checking AZ trace data
  In addition to the processing time, execution address, and executed task/interrupt name, the system call issue status provided by the real-time OS is displayed, enabling even finer debugging in task units.

- Verification of task/interrupt name, event type or system call argument name
  It is possible to perform searches using conditions such as task/interrupt name, event type, system call argument name, etc.
CHAPTER 4    CONSTRUCTING APPLICATION PROGRAM

4. 1 Using the AZ850 in Soft Trace Form

This section describes the procedure of constructing an application program when the AZ850 is used in the soft trace form.

In the soft trace form, a monitor function is provided on the target system, and the monitor program (AZ monitor) collects AZ trace data (refer to the "3. 1 Soft Trace Form and Hard Trace Form" for details on the soft trace form).

(1) Creating a task source program
  Create the task source program (including boot block and interrupt handler) that will form the processing to realize the application system.

(2) Creating table with configurator
  Using the configurator (CF850/CF850 Pro) supplied by the RX850/RX850 Pro, create a system information table (SIT file) from a system configuration file.
  When the RX850 is used, the trace information must be defined in the system configuration file (refer to the RX850 user's manual for details).

(3) Creating AZ monitor
  Create the user own coding block of AZ monitor (refer to the "CHAPTER 5  AZ MONITOR (SOFT TRACE ONLY)" for details).

(4) Creating object files
  Compile and assemble the created source programs (including the user own coding block for the real-time OS and the system information table) to generate object files.
  At this time, note that the user own coding block for AZ monitor (CA850: azusr_XXX.s, GHS compiler: azusr_xxx.850) must be assembled to generate a object file.

(5) Creating link directive file
  Create a link directive file (section map file) by which the user fixes address allocation performed by the link editor. At this time, note that specify the following additional section definitions for the AZ monitor in the data area and text area of the link directive file.

<table>
<thead>
<tr>
<th>Section Name</th>
<th>Description</th>
<th>Attribute</th>
<th>Occupancy Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>.azmon_b</td>
<td>Work area of the AZ monitor</td>
<td>bss</td>
<td>For RX850: 44 bytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For RX850 Pro: 64 bytes</td>
</tr>
<tr>
<td>.azmon_t</td>
<td>Text/data area of the AZ monitor</td>
<td>text</td>
<td>For RX850: Appro. 800 bytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For RX850 Pro: Appro. 1200 bytes</td>
</tr>
</tbody>
</table>
CHAPTER 4 CONSTRUCTING APPLICATION PROGRAM

Figure 4-1  Example of AZ Monitor Section Definition for RX850 (When Using CA850)

```
IDATA : !LOAD ?RW {
    .pool0 = $NOBITS ?AW .pool0;
    .data = $PROGBITS ?AW .data;
    .sdata = $PROGBITS ?AWG .sdata;
    .sbss = $NOBITS ?AWG .sbss;
    .bss = $NOBITS ?AW .bss;
    .azmon_b = $NOBITS ?AW .azmon_b;  <!--Additional description
};

TEXT : !LOAD ?RX {
    .sit = $PROGBITS ?AX .sit;
    .text = $PROGBITS ?AX .text;
    .azmon_t = $PROGBITS ?AX .azmon_t;  <!--Additional description
};
```

Figure 4-2  Example of AZ Monitor Section Definition for RX850 (When Using GHS's Compiler)

```
.data  0x00001000 :
.sdata :
.sbss :
.bss :
.azmon_b :  <!--Additional description
;
.text  0x00ffe000 :
.system :
.system_int :
.system_sys :
.azmon_t :  <!--Additional description
;
```

Figure 4-3  Example of AZ Monitor Section Definition for RX850 Pro (When Using CA850)

```
EDATA :!LOAD ?RW  V0x00100000{
    .data = $PROGBITS ?AW .data;
    .sdata = $PROGBITS ?AWG .sdata;
    .sbss = $NOBITS ?AWG .sbss;
    .bss = $NOBITS ?AW .bss;
    .azmon_b = $NOBITS ?AW .azmon_b;  <!--Additional description
};

TEXT : !LOAD ?RX  V0x000010000{
    .sit = $PROGBITS ?A .sit;
    .system = $PROGBITS ?AX .system;
    .system_cmn = $PROGBITS ?AX .system_cmn;
    .system_int = $PROGBITS ?AX .system_int;
    .text = $PROGBITS ?AX .text;
    .azmon_t = $PROGBITS ?AX .azmon_t;  <!--Additional description
};
```
Figure 4-4 Example of AZ Monitor Section Definition for RX850 Pro (When Using GHS’s Compiler)

[Caution] Locate the .azmon_b section within a RAM area that can be read and written. The area should not be written (or initialized) by an application program or the like.

(6) Creating a load module file
Link all of the following files to create a load module file.
- Object files
- Core of AZ Monitor (RX850: azcore.o, RX850 Pro: azcore_p.o)
- All library files provided by the RX850/RX850 Pro
- Link directive file

[Caution] If the AZ monitor is not linked, the AZ850 cannot be used in the soft trace form.

(7) Debugging and performance analysis using debugger and AZ850
Download the load module file created above to the debugger to execute debugging and performance analysis.
4. 2 Using the AZ850 in Hard Trace Form

This section describes the procedure of organizing an application program when the AZ850 is used in the hard trace form.

In the hard trace form, trace data is collected using the trace function of an in-circuit emulator or simulator (refer to the "3. 1 Soft Trace Form and Hard Trace Form" for details on the hard trace form).

Since AZ monitor is not used in the hard trace form, it is not necessary to link the AZ monitor.

(1) Creating a source program
Create the source program that will form the processing to realize the application system.

(2) Creating table with configurator
Using the configurator (CF850/CF850 Pro) supplied by the RX850/RX850 Pro, create a system information table (SIT file) from a system configuration file.
When the RX850 is used, the trace information must be defined in the system configuration file (refer to the RX850 user's manual for details).

(3) Creating object files
Compile and assemble the created source program and the other files to create object files.

(4) Creating link directive file
Create a link directive file (section map file) by which the user fixes address allocation performed by the link editor.

(5) Creating a load module file
Link the object files created above and all the library files provided by the RX850/RX850 Pro to create a load module file.

(6) Debugging and performance analysis using debugger and AZ850
Download the load module file created above to the debugger to execute debugging and performance analysis.

[Caution1] The AZ850 uses the debugger's trace conditions to collect AZ trace data. Thus, if many conditions are used on the debugger side, it may not be possible to perform AZ trace ON state setting (refer to "1. 5 Resource").

[Caution2] When using the AZ850 in the hard trace form with the ID850, clear the [ADD Up Timetag] check box in the ID850 Extended Option dialog box. Unless this check box is cleared, the correct time information cannot be obtained.
CHAPTER 5   AZ MONITOR (SOFT TRACE ONLY)

This chapter describes the details of how to create the AZ monitor (monitor program) that is necessary for using the AZ850 in the soft trace form.
This chapter may be skipped if the AZ850 is used only in the hard trace form.

5. 1 About AZ Monitor

AZ monitor is a monitor program which consists of the user own coding block (CA850: azusr_XXX.s, GHS's compiler: azusr_XXX.850) and the core block (RX850: azcore.o, RX850 Pro: azcore_p.o), that collects AZ trace data when the AZ850 is used in the soft trace form.
AZ monitor acquires the time information necessary for AZ trace data by using the timer counter of the user's target device. This requires that the control block of the timer counter be coded according to the target device of user.
Therefore, the user own coding block that controls the timer counter must be created to complete creation of the AZ monitor.
5. 2 Creating AZ Monitor

5. 2. 1 Creating user own coding block

Code the control block of the timer counter of the target to be used. Create the control block by referring to the provided sample program (CA850: azusr_XXX.s, GHS’s compiler: azusr_XXX.850).

Figure 5-1 shows the data and functions necessary for the user own coding block, taking a timer counter as an example.

Figure 5-1 Example of Timer Counter Operation (In Case of Up Counter)

![Figure 5-1 Example of Timer Counter Operation (In Case of Up Counter)]

Table 5-1 Data/Functions of User Own Coding Block for AZ Monitor

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>__AZMON_MaxCount</td>
<td>Data</td>
<td>Specify the maximum number of timer counts. (&quot;n&quot; value in Figure 5-1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data type: Specify with 4-byte integer format (.word)</td>
</tr>
<tr>
<td>__AZMON_CountMode</td>
<td>Data</td>
<td>Specify timer count mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data type: Specify with 1-byte integer format (.byte)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data setting values: 0x00 … Up counter, 0x01 … Down counter</td>
</tr>
<tr>
<td>__AZMON_TimePerCount</td>
<td>Data</td>
<td>Specify the time per count in microsecond units (usec) (&quot;tpc&quot; value in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Figure 5-1).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data type: Specify with 4-byte floating-point format (.float)</td>
</tr>
<tr>
<td>__AZMON_InitTimer</td>
<td>Function</td>
<td>Specify the timer initialization processing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input register: lp … Return address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Output register: None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Destructible register: Depends on C language calling restrictions</td>
</tr>
<tr>
<td>__AZMON_GetCounter</td>
<td>Function</td>
<td>Specify the processing for acquiring the timer counter value. Note that a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stack overflow in this function cannot be detected. Therefore, creation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>without using a stack is recommended.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input register: lp … Return address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Output register: r10 … Counter value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Destructible registers: r1, r11</td>
</tr>
</tbody>
</table>
[Caution1] The unit of time displayed as AZ trace data depends on the __AZMON_TimePerCount value (tpc value in Figure 5-1). If the event interval of AZ trace data to be collected is longer than the interval time of the timer counter, the correct time cannot be displayed.

[Caution2] If the ld.h instruction is used to acquire the timer counter value (__AZMON_GetCounter), the r10 value is sign-extended to 4-byte value. Therefore, this value must be masked as the example shown below. When the V850E/V850ES is used, however, it is not necessary to mask it with the andi instruction by using the ld.hu instruction instead of the ld.h instruction.

\[
\begin{align*}
&\text{ld.h} \quad 0[r1], \quad r10 \\
&\text{andi} \quad 0xffff, \quad r10, \quad r10
\end{align*}
\]

[Caution3] Do not share the timer to be used for AZ monitor and the timer to be used for the RX850/RX850 Pro.
5. 2. 2 Initializing AZ monitor

The AZ monitor must be initialized before it is operated. "AzInit()" that is a initialization routine for AZ monitor is prepared in AZ monitor. Issue AzInit() within the initialization handler of the RX850/RX850 Pro (RX850: init_handler(), RX850 Pro: varfunc() (default name)). AzInit() initializes the timer counter and trace control. The specification of AzInit() function is as follows. This function has no arguments.

```c
void AzInit(void);
```

Figure 5-3 shows an example of description for initialization of the AZ monitor.

**Figure 5-3 Example of AZ Monitor Initialization Description (When Using RX850)**

```c
void init_handler()
{
    ..................

    /*
     * Initialize AZ Monitor
     */
    AzInit();    <-- Additional description

    return;
}
```
6.1 Debugging Procedure Using AZ850

This section describes the procedure for debugging using the AZ850.

(1) Starting the debugger
Start the debugger to be used.

(2) Starting the AZ850
Start the AZ850 and open the Main Window.
At this time, check whether "Connected" is displayed on the status bar in the Main Window. It indicates that the debugger is connected (if it is not displayed, the chances are that the debugger does not support the TIP interface).

(3) Downloading a load module
Download a load module created for the AZ850 to the debugger.
Check the following status on the status bar in the Main Window after the downloading is completed (if not displayed, the chances are that the symbol information is not included in the load module).

[Using in soft trace form]

| RX+AZ Loaded | Both of the RX850/RX850 Pro and AZ monitor are loaded. |

[Using in hard trace form]

| RX Loaded | The RX850/RX850 Pro is loaded. |

Figure 6-1 Checking on Main Window
(4) Setting in the [AZ Option] dialog box
Select the [Option] menu -> [AZ Option...] on the Main Window to open the [AZ Option] dialog box.
Specify the following options in this dialog box. For details on the specification, refer to the [AZ Option] dialog box.

Figure 6-2 Setting in [AZ Option] Dialog Box

- **Address Mask**
  Specify the maximum physical address of CPU.

- **RTOS Select**
  Select the real-time OS type embedded in the application program.

- **Trace Form**
  Select the trace form to be used.
  If the debugger to be used supports only one trace form, however, the form is fixed to either of the two form.

- **Soft Form Buffer Type**
  Select the type of trace buffer to be used when the soft trace form has been selected as the trace form. This item is invalid when the hard trace form has been selected.

- **Soft Form Buffer Region**
  Specify the start and end address of the trace buffer region when the soft trace form has been selected as the trace form.
  The trace data collected by the AZ850 is acquired once into the target memory. This means that it is necessary to specify the unused memory area of the target memory as the trace buffer region.
  Specify the unused memory area in the range of 4K bytes to 4M bytes.
  If this setting is not correctly made when the AZ850 used in the soft trace form, AZ trace data cannot be collected.
(5) Switching the AZ trace mode
Set the AZ trace ON state to enable the AZ850 trace function by clicking the button on the Main Window. Note that the AZ trace ON state cannot be set until after a load module has been downloaded.

Figure 6-3  AZ Trace Mode Switching

(6) Executing an application program
Execute the application program on the debugger. The trace data starts being collected into the trace buffer along with the application program execution.

(7) Stopping the application program
Stop the application program on the debugger. If a breakpoint has been set, wait until a break occurs. Stopping the application program also stops the collection of trace data.

(8) Uploading of trace data
Convert the trace data collected in the trace buffer to AZ trace data for the AZ850 through uploading by clicking the button on the Main Window.

Figure 6-4  Uploading of Trace Data

(9) Check using the Analyze Window
Open the Analyze Window to display the execution transition map based on the uploaded AZ trace data. Using this window and each of the windows that can be opened from this window, system timing misses as well as the entire system itself can be checked.
- To display the Analyze Window:
  Click the button on the Main Window.

- To check the operation of each object:
  Check the operation of the specified object using the button (the simple search button) displayed by clicking the object name.

- To check the transition state of a task/interrupt:
  By using the [Pattern Search] dialog box opened by clicking the button, a search can be made for the position of a task switching, occurrence of an interrupt, or issue of a system call.

- To analyze the task/interrupt processing time:
  Search/analyze the pattern (specific processing) set in the [Pattern Set] dialog box opened by the button on this window.
(10) Check using the CPU Window

Check the CPU usage status for the defined range by using the **CPU Window**.

Using this window, the execution time of a specified task/interrupt can be confirmed, and the target data performance can be evaluated, base on the idle time and interrupt time.

![Figure 6-6 Display Example of CPU Window](image)

- To display the **CPU Window**:
  1. Specify the range for which the total CPU usage is to be calculated, using the up cursor and down cursor in the Analyze Window.
  2. Click the ![button](image) button on the Analyze Window.

  **Remark** If the CPU Window is in the active status and either the up cursor or down cursor is repositioned in the execution transition map, the contents of CPU usage is automatically updated.

(11) Check using the Pattern Window

Check the distribution of the specified pattern for given execution duration by using the **Pattern Window**.

It is possible to obtain the worst execution time value and average execution time value for application program processing from this window.

![Figure 6-7 Display Example of Pattern Window](image)
- To display the **Pattern Window**:

  **Displaying a pattern distribution**

  1. Specify the range for which pattern distribution is to be calculated, using the up cursor and down cursor in the **Analyze Window**.

  2. Click the button on the **Analyze Window**. In the [Pattern Set] dialog box that opens automatically, set the conditions of the pattern to be calculated. Then, click the [OK] button in this dialog box.

**Displaying a pattern distribution from the Cpu Window**

  1. Specify the range for which pattern distribution is to be calculated, using the up cursor and down cursor in the **Analyze Window**.

  2. In the **Cpu Window**, select an object in the list.

  3. Click the button on the **Cpu Window**.

  **[Remark]** If the Pattern Window is in the active status and either the up cursor or down cursor is repositioned in the execution transition map, the contents of pattern distribution is automatically updated.

(12) **Check using the Trace View Window**

Check the collected AZ trace data in detail by using the **Trace View Window**.

![Figure 6-8 Display Example of Trace View Window](image)

- To display the **Trace View Window**:

  1. Move the up temporary cursor to the desired position in the **Analyze Window**. When the up temporary cursor is not displayed, the beginning of the trace data becomes the display start position.

  2. Click the button on the **Analyze Window**.

  **[Caution]** If the Trace View Window is in the active status and either the up temporary cursor is repositioned in the execution transition map, the contents of AZ trace data is automatically updated.

Debug the application program, repeating steps (6) and after as required.

For details on the procedure to operate on a each window or dialog box, refer to the following.
CHAPTER 7    WINDOW REFERENCE

7. 1   Outline of Windows and Dialog Boxes of AZ850

The following shows the list of windows and dialog boxes of the AZ850.

Table 7-1  List of Windows and Dialog Boxes of AZ850

<table>
<thead>
<tr>
<th>Window/Dialog Box Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Window</td>
<td>Window for performing basic operation of the AZ850.</td>
</tr>
<tr>
<td>[AZ Option] dialog box</td>
<td>Sets options such as the trace form and the trace level to use the AZ850.</td>
</tr>
<tr>
<td>[Open][Save As] dialog box</td>
<td>Loads and saves of the contents displayed in windows to the display file.</td>
</tr>
<tr>
<td>Analyze Window</td>
<td>Displays the execution transition map based on the AZ trace data collected.</td>
</tr>
<tr>
<td>[Object Select] dialog box</td>
<td>Selects the objects to be displayed in execution transition map and modifies the order in which those objects are displayed.</td>
</tr>
<tr>
<td>[Pattern Search] dialog box</td>
<td>Sets the search conditions to search for an object in the execution transition map.</td>
</tr>
<tr>
<td>Cpu Window</td>
<td>Displays the CPU usage within the specified period in the task-level execution transition map.</td>
</tr>
<tr>
<td>[Pattern Set] dialog box</td>
<td>Sets the pattern to be displayed in the Pattern Window.</td>
</tr>
<tr>
<td>Pattern Window</td>
<td>Displays the histogram representing the number of times the specified pattern appears for given execution duration in the execution transition map.</td>
</tr>
<tr>
<td>Trace View Window</td>
<td>Displays the list of the contents of AZ trace data collected.</td>
</tr>
<tr>
<td>[Trace Search] dialog box</td>
<td>Sets the search conditions applied when searching for AZ trace data with the Trace View Window.</td>
</tr>
<tr>
<td>[About] dialog box</td>
<td>Displays the information about the AZ850.</td>
</tr>
</tbody>
</table>
7. 2 Relationship of AZ850 Windows and Dialog Boxes

The following shows the relationship of the AZ850 windows and dialog boxes.

Figure 7-1 Relationship of AZ850 Windows and Dialog Boxes
7. 3 Explanation of Windows and Dialog Boxes

This section describes each window/dialog box of the AZ850 as follows:

**Window/Dialog box name**

Shown in the frame are the window or dialog box name. In addition, the display image of the window or dialog box, functional outline, and how to open the window or dialog box are also explained.

**Explanation of each area**

Explains items to be set to or displayed in each area of the window or dialog box.

**Menu bar**

Lists the menu items and explains the operation of each menu item.

**Tool bar**

Explains the operation of each button on the tool bar.

**Function buttons**

Explains the operation of each function button in the window or dialog box.

**Caution**

Enumerates points that require care during operation in the window or dialog box.

**Error**

Enumerates errors that may occur during operation in the window or dialog box.

**Other**

Explains the special functions of the window or dialog box, such as the operating method.
Main Window

This window, which opens automatically after the AZ850 is started up, is used to perform basic operations of the AZ850. To use the AZ850, start operation from this window.

This section describes the following items:

- Menu bar
- Tool bar
- Status bar
- Caution
- Error
Menu bar

(1) [File] menu

- **[Exit]**
  Terminates the AZ850.

(2) [Option] menu

- **[Tool Bar]**
  Displays (default) or hides the tool bars in windows of the AZ850.
  The tool bars are displayed while this item is prefixed with a check mark.

- **[Status Bar]**
  Displays (default) or hides the status bars in windows of the AZ850.
  The status bars are displayed while this item is prefixed with a check mark.

- **[AZ Option...]**
  Opens the [AZ Option] dialog box.
  In this dialog box, the AZ trace options are specified.
  The function of this item is same as that of the button.

(3) [Operation] menu

- **[AZ Trace ON]**
  Sets the AZ trace ON state. While this item is prefixed with a check mark, the AZ trace ON state is set, so that executing the application program collects AZ trace data into the trace buffer.
  This menu item cannot be selected until after both of the RX850/RX850 Pro and AZ monitor (in the case of the hard trace form, until after the RX850/RX850 Pro) have been downloaded. In addition, this menu item cannot be also selected while the application program is being executed.
  The function of this item is same as that of the button.

- **[AZ Trace OFF]**
  Sets the AZ trace OFF state. While this item is prefixed with a check mark, the AZ trace OFF state is set, so that executing the application program does not collect AZ trace data into the trace buffer.
  AZ trace OFF state is set as AZ trace mode when the AZ850 is started.
  The function of this item is same as that of the button.

- **[Upload]**
  Uploads the collected AZ trace data into the AZ850.
  This item cannot be selected if the trace buffer contains no trace data.
  The function of this item is same as that of the button.
(4) [Browse] menu

[Analyze...] Opens the Analyze Window.

(5) [Window] menu

[Close All] Closes all the AZ850 windows and dialog boxes except the Main Window.

(6) [Help] menu

[This Window] Displays the help window for this window.
[Help Topics] Starts the AZ850 online help.
[About...] Opens the [About] dialog box.

The product name, version number of the AZ850, [date of product build] and copyright year are displayed.

Tool bar

The tool bar consists of buttons that can execute frequently used menu items with a single action.

Table 7-2 Tool Bar of Main Window

<table>
<thead>
<tr>
<th>Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>![AZ Option icon]</td>
<td>Opens the [AZ Option] dialog box. In this dialog box, the AZ trace options are specified. Same operation as [Option] -&gt; [AZ Option...].</td>
</tr>
<tr>
<td>![Trace On/Off icon]</td>
<td>Sets AZ trace data to on if it is off, and vice versa. In the AZ trace ON state, trace data is collected into the trace buffer as the application program is executed. This item cannot be selected until after both of the RX850/RX850 Pro and AZ monitor (in the case of the hard trace form, until after the RX850/RX850 Pro) have been downloaded. In addition, this item cannot be also selected while the application program is being executed. In the AZ trace OFF state, trace data is not collected even if the application program is executed. Same operation as [Operation] -&gt; [AZ Trace ON] or [Operation] -&gt; [AZ Trace OFF].</td>
</tr>
<tr>
<td>![Upload icon]</td>
<td>Uploads the collected AZ trace data into the AZ850. This item cannot be selected if the trace buffer contains no trace data. Same operation as [Operation] -&gt; [Upload].</td>
</tr>
<tr>
<td>![Analyze icon]</td>
<td>Opens the Analyze Window. If AZ trace data has been uploaded, the execution transition map based on that data is displayed.</td>
</tr>
</tbody>
</table>
Status bar

Figure 7-3 shows the name of each area on the status bar.

Figure 7-3  Status Bar of Main Window

(a) Connection status area
This area indicates the status of connection with the debugger.
Connected  Connected to the debugger
Not Connected  Not connected to the debugger

(b) Load module status area
This area indicates the state of the load module to be loaded onto the debugger.
This area is not displayed if the AZ850 is not connected to the debugger.
RX+AZ Loaded  Both of the RX850/RX850 Pro and AZ monitor are loaded. (Soft trace form)
RX+AZ None  Both or either of the RX850/RX850 Pro and AZ monitor is not loaded. (Soft trace form)
RX Loaded  The RX850/RX850 Pro is loaded. (Hard trace form)
RX None  The RX850/RX850 Pro is not loaded. (Hard trace form)

(c) Load module execution status area
This area indicates the state of the application program execution.
This area is not displayed if the AZ850 is not connected to the debugger.
Running  Status of application program being execution
Breaked  Status of application program operation undergoing break

(d) AZ trace mode status area
This area indicates the current state of AZ trace mode.
This area is not displayed if the AZ850 is not connected to the debugger or a load module (the required symbol information) has not been downloaded.
AZ Trace ON  Status where AZ trace ON has been set
AZ Trace OFF  Status where AZ trace OFF has been set
Caution

- The following state of the AZ850 is required to set the AZ trace ON.
  
  In case of soft trace form
  Both of the RX850/RX850 Pro and AZ monitor are loaded ([RX+AZ Loaded] is displayed on the status bar).
  
  In case of hard trace form
  The RX850/RX850 Pro is loaded ([RX Loaded] is displayed on the status bar).

- In the hard trace form, the AZ850 sets a trace condition to the debugger when the AZ trace ON state is set. Therefore, care is required regarding the number of trace conditions that the debugger can use. If the resources for the AZ850 is insufficient, the AZ trace ON state cannot be set. Refer to the "1. 5 Resource" for details.

- The contents of Main Window dose not change even if AZ trace data is uploaded. To check the contents of AZ trace data, open the Analyze Window after selecting the [Operation] menu -> [Upload].

Error

In the following cases, the ERROR MESSAGES is opened to display error messages.

- If an error occurs during the switching processing of the AZ trace mode
- If an error occurs during the uploading processing of the AZ trace data
- If the AZ trace ON is set without set anything with [Soft Form Buffer Region] in the [AZ Option] dialog box
- If an attempt is made to upload AZ trace data when no trace data is collected
- If AZ trace data obtained by uploading processing is not correct
This dialog box is used to specify various options for using the AZ850. This dialog box can be opened from the Main Window by any of the following:

- Select the [Option] menu -> [AZ Option...]
- Click the button on the tool bar.
- Press the [Alt], [P] and [O] keys in that order.
- Press the [Ctrl]+[O] keys at the same time.

Figure 7-4 [AZ Option] Dialog Box

This section describes the following items:

- Explanation of each area
- Function buttons
- Error
Explanation of each area

(a) Address Mask
This area is used to specify the maximum physical address of the using CPU.
Select the according to the CPU used as shown below.

0x00ffffff  V850/SA1, V850/SB1, V850/SB2, V850/SC1, V850/SC2, V850/SC3, V850/SV1, and so on
0x03ffffff  V850ES/KF1, V850ES/KG1, V850ES/KJ1, V850ES/SJ2, V850ES/SG2, V850E/MS1, V850E/MS2, and so on
0x0fffffff  V850E/MA1, V850E/ME2, V850E/IA1, V850E/IA2, and so on (default)

For except for the above, refer to the user’s manual of using CPU and specify the value in hexadecimal from the key board.

(b) RTOS Select
This area is used to specify the real-time OS that is embedded in the downloaded load module.
Select either [RX850] or [RX850 Pro].

(c) Trace Form
This area is used to specify the trace form of the AZ850.
Select the option button corresponding to the operating environment.

Soft Form  The AZ850 is used in the soft trace form.
Select this button when a monitor function is provided on the target system, and the monitor program collects AZ trace data.

Hard Form  The AZ850 is used in the hard trace form (default).
Select this button when trace data is collected using the trace function of an in-circuit emulator or simulator.

[Caution] Either the soft trace form or hard trace form of the AZ850 must be specified.
If the debugger to be used dose not support the hard trace form, the default form is the soft trace form.

(d) Soft Form Buffer Type
This area is used to specify the trace buffer type when the [Soft Form] is selected with "Trace Form" area (This area is invalid when the [Hard Form] is selected).
Select the option button corresponding to the operating environment.

Fixed Buffer  The trace buffer is fixed type buffer.
The trace data is collected until the trace buffer is filled. Therefore, all the trace data up to the point where the application program stops is not always collected.

Ring Buffer  The trace buffer is ring type buffer.
The oldest trace data is overwritten when the trace buffer is filled.

(e) Soft Form Buffer Region
This area is used to specify the trace buffer area when the [Soft Form] is selected with "Trace Form" area (This area is invalid when the [Hard Form] is selected).
Directly enter the start address and end address for the trace buffer in the text box.
In the soft trace form, the trace data collected by the AZ850 is acquired into the target memory once. This means that it is necessary to specify the unused memory area of the target memory as the trace buffer area.
Specify the unused memory area in the range of 4K bytes to 4M bytes.
Function buttons

Table 7-3  Function Buttons of the [AZ Option] Dialog Box

<table>
<thead>
<tr>
<th>Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>Validates the specified settings.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Closes this dialog box.</td>
</tr>
<tr>
<td>Help</td>
<td>Displays the help window for this dialog box.</td>
</tr>
</tbody>
</table>

Error

In the following cases, the ERROR MESSAGES is opened to display error messages.

- If the OK button is clicked when the value specified in the [Address Mask] is not correct.
- If the OK button is clicked when the address range specified in the [Soft Form Buffer Region] is not correct.
[Open]/[Save As] dialog box

This dialog box is used to select files when loading or saving display files. This dialog box can be opened from each window (excluding the Main Window) by any of the following.

- Select the [File] menu -> [Open...].
- Select the [File] menu -> [Save...].
- Click the button on the tool bar.
- Click the button on the tool bar.
- Press the [Alt], [F] and [O] keys in that order.
- Press the [Alt], [F] and [S] keys in that order.
- Press the [Ctrl]+[O] keys at the same time.
- Press the [Ctrl]+[S] keys at the same time.

This section describes the following items:

- Explanation of each area
- Function buttons
- Other
- Error
Explanation of each area

(a) Look in:
Select the drive or folder in which the specified file exists, or the file is to be saved.

(b) List of files
This area displays the list of display files.

(c) File name:
Specify the name of the file to be opened or saved.

(d) Files of type:
Select the type of the file to be opened or saved.

Function buttons

Table 7-4 Function Buttons of the [Open]/[Save As] Dialog Box

<table>
<thead>
<tr>
<th>Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Opens the specified file, or saves the file with specified name.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Closes this dialog box.</td>
</tr>
<tr>
<td>Help</td>
<td>Displays the help window for this dialog box.</td>
</tr>
</tbody>
</table>

Other

(1) Display file extensions and file formats
Display files vary from window to window, and are differentiated by their extension. The default extensions and formats of display files corresponding to the various windows of the AZ850 are as follows:

Table 7-5 Display File Extensions and Formats

<table>
<thead>
<tr>
<th>Window Name</th>
<th>Extension</th>
<th>File Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyze Window</td>
<td>.az</td>
<td>Binary</td>
</tr>
<tr>
<td>Cpu Window</td>
<td>.azc</td>
<td>CSV</td>
</tr>
<tr>
<td>Pattern Window</td>
<td>.azp</td>
<td>CSV</td>
</tr>
<tr>
<td>Trace View Window</td>
<td>.azt</td>
<td>CSV</td>
</tr>
</tbody>
</table>
(2) Display information reproduction

Reproduction of display information by loading a display file differs as follows for each window.

- **Analyze Window**
  The display information is reproduced in the currently open Analyze Window. To maintain the currently displayed execution transition map, therefore, set the current Analyze Window to the hold status, then open a new Analyze Window and load a display file.

- **Other windows**
  The display information is reproduced in a new window that is opened in the hold status. The display information of the currently displayed window is, therefore, maintained as is on the screen.

**Error**

In the following cases, the **ERROR MESSAGES** is opened to display error messages.

- If the specified file name format is not correct
- If loading a file that does not exist is attempt
- If saving data to a write-protected file is attempt
- If loading data that cannot be handled in the current window is attempt
Analyze Window

This window is used to display AZ trace data, collected as a result of executing an application program, in the execution transition map for each task.

Using this window and each of the windows that can be opened from this window, system timing misses as well as the entire system itself can be checked and analyzed in units of tasks.

This window can be opened from the Main Window by any of the following.

• Select the [Browse] menu -> [Analyze...].
• Click the button on the tool bar.
• Press the [Alt], [B] and [A] keys in that order.
• Press the [Ctrl] + [A] keys at the same time.

Figure 7-6 Analyze Window

(a) Up cursor  (j) Time display of up temporary cursor position
(b) Down cursor  (k) Data length
(c) Up temporary cursor  (l) Jump to beginning button
(d) Down temporary cursor  (m) Jump to end button
(e) Object button  (n) Scale modify button
(f) Analysis result (transition map) display area  (o) Simple search button
(g) Time display of up cursor position  (p) Time area
(h) Time display of down cursor position  (q) Pop-up display
(i) Inter-cursor time display  (r) Sort button
This section describes the following items:

- Explanation of each area
- Menu bar
- Tool bar
- About objects
- Understanding the execution transition map
- Verification method in execution transition map
- Caution
- Error

**Explanation of each area**

(a) **Up cursor**
This cursor is used to specify the range for measuring a processing time in AZ trace data or for calculating other data (specify the beginning of the range for which a calculation is to be done).
This cursor can be manipulated by clicking on the transition map while pressing the [Shift] key.

(b) **Down cursor**
This cursor is used to specify the range for measuring a processing time in AZ trace data or for calculating other data (specify the end of the range for which a calculation is to be done).
This cursor can be manipulated by clicking on the transition map while pressing the [Ctrl] key.

(c) **Up temporary cursor**
This cursor indicates a position resulting from a search by using the "simple search button" or pattern search, or the start position from which a jump is made to the debugger.
When the search result is displayed by using the Pattern Window, this cursor indicates the start position of the searched pattern.
When the up cursor is moved, the up temporary cursor also moves to the same position.
This cursor can be manipulated by clicking the left mouse button.

(d) **Down temporary cursor**
When the search result is displayed by using the Pattern Window, this cursor indicates the end position of the searched pattern.
When the down cursor is moved, the down temporary cursor also moves to the same position.

**[Remark]** Each cursor can be moved by dragging it with the mouse. When the horizontal scroll bar is valid, the entire screen can be scrolled in the horizontal direction by moving the mouse on the left or right of the transition map, or out of the horizontal frame, while dragging each cursor.

(e) **Object button**
A button group that is used to indicate objects within the collected AZ trace data.
Objects indicate the RX850/RX850 Pro resources such as tasks, event flags and mailboxes. If the object name is too long, part of it is omitted. The correct object name can be displayed by putting the mouse pointer on the button.
The display sequence of each object can be changed freely, by dragging it with the mouse. For details on the objects, refer to the "About objects" below.
The simple search button (button) appears by clicking an object name. Clicking the object name once more makes the simple search button disappear.
(f) Analysis result (transition map) display area
This area is used to display the result of analyzing an application program.
For details on the marks in the transition map, refer to "Understanding the execution transition map" below.

(g) Time display of up cursor position
This box displays the relative time from the start of AZ trace data collection until the up cursor position.
The units are milliseconds [msec].

(h) Time display of down cursor position
This box displays the relative time from the start of AZ trace data collection until the down cursor position.
The units are milliseconds [msec].

(i) Inter-cursor time display
This box displays the time interval between the up cursor and the down cursor.
The units are milliseconds [msec].

(j) Time display of up temporary cursor position
This box displays the absolute time from the start of AZ trace data collection until the up temporary cursor position. The units are milliseconds [msec].

(k) Data length
This box displays the time interval from the start to the end of AZ trace data.

(l) Jump to beginning button
This button is used to move the up cursor and the display screen to the beginning of the AZ trace data.

(m) Jump to end button
This button is used to move the down cursor and display screen to the end of the AZ trace data.

(n) Scale modify button
This button is used to change the display scale for the execution transition map. A drop-down list with a button that expands the scale by double each time and a button that reduces the scale by 1/2 each time is available.

(o) Simple search button
This button appears by clicking an object name, is used to determine the operation of selected object.
When the button is clicked, the operation of the object selected is searched in the time axis direction in the execution transition map, and the up temporary cursor moves to the search position.
When the button is clicked, the operation of the object selected is searched in the opposite direction as the time axis, and the up temporary cursor moves to the search position.

(p) Time area
This area indicates the standard time between events displayed on the execution transition map.
The unit is displayed at the rightmost position of this area and displayed in milliseconds [msec], microsseconds [usec], or nanoseconds [nsec].
When this window is displayed as the equal mode, a guide to the number of events on the transition map is displayed. In this mode, the unit is the number of events, and the number increases from 1 to 10, 100, and so on, along with the scale. At this time, [unit] is not displayed at the rightmost position of the area.
(q) **Pop-up display**

The detailed information can be displayed by putting the mouse pointer on the following position.

- **Object button**
  - RX850: The name of the object
  - RX850 Pro: The name of the object (the ID number of the object)

- **Vertical line that indicates a task switch**
  - Time at the task switch
  - The name of the task/interrupt before switching
  - The name of the task/interrupt after switching

- **System call (orange horizontal line)**
  - The time at issue of the system call
  - The time at termination of the system call (The time required for the system call)
  - Syscall (the name of the system call)
  - The name of the target object (for some system calls, this information cannot be displayed.)

- **Interrupt (orange horizontal line)**
  - The time at occurrence of the interrupt
  - The time at termination of the interrupt (the time required for the interrupt)
  - Int (interrupt name)

- **Vertical line that indicates an access to an object**
  - The time at issue/termination of the system call
  - The name of a task at the occurrence of an event
  - Syscall/SysRet (system call name)
  - Target object name

- **The mark indicating system call error**
  - The time at termination of the system call
  - Return value

(r) **Sort button**

This button is used to change the sequence in which object buttons are to be displayed. The following specifications can be made.

- **Appr** Displays the objects in the sequence in which they were detected as AZ trace data.
- **Name** Displays the objects according to ASCII code.
- **ID** Displays the objects according to their ID number.
- **Pri** Displays the objects according to their priority (valid only for tasks).

[Caution] The objects are sorted by type and displayed, in the order of interrupt, task, idle, event flag, semaphore, mailbox, and memory pool.
Menu bar

(1) [File] menu

- **[Open...]** Opens the [Open][Save As] dialog box. A previously saved display file, obtained using this window, is read. The default extension for the display file of this window is ".az". The file name is displayed on the title bar after the display file has been read.
- **[Save...]** Opens the [Open][Save As] dialog box. The current display information for this window is saved to a display file, which can either be created newly or by copying and renaming an existing file. The default extension for the display file of this window is ".az".
- **[Close]** Closes this window.

(2) [View] menu

- **[Grid mode]** Sets whether grid lines are displayed within the execution transition map. The cascade menu contains the following.
  - **[Grid]** Displays grid lines (default).
  - **[Ungrid]** Does not display grid lines.
- **[View mode]** Specifies the view mode for the execution transition map. The cascade menu contains the following.
  - **[Simple]** Does not display vertical lines on the execution transition map. The function of this item is same as that of the button.
  - **[Standard]** Displays only the CPU execution transition. The function of this item is same as that of the button.
  - **[Detail]** Displays the "mark" of the system call access state of the RX850/RX850 Pro (default). The function of this item is same as that of the button.
(3) [Operation] menu

[Equal] The length of the horizontal line on the transition map becomes equal between events regardless of the execution time of the CPU (if not selected this menu item, a graph in proportion to the execution time of the CPU is displayed).

The function of this item is same as that of the \(\text{SCALE}\) button.

[Small] Reduces the scale of the execution transition map. The function of this item is the same as that of the "scale modify button".

[Large(T)] Expands the scale of the execution transition map. The function of this item is the same as that of the "scale modify button".

[Find...] Opens the [Pattern Search] dialog box.

The function of this item is same as that of the \(\text{FIND}\) button.

[Sort Object] Specifies the sequence in which object buttons are to be displayed. The cascade menu contains the following.

[Appear] Displays the objects in the sequence in which they were detected as AZ trace data (default).

[Name] Displays the objects according to ASCII code.

[ID] Displays the objects according to their ID number.

[Priority] Displays the objects according to their priority (valid only for tasks).

[Select Object...] Opens the [Object Select] dialog box.

The function of this item is same as that of the \(\text{SELECT}\) button.

[Active] Switches this window from the hold status to the active status (default).

[Hold] Switches this window from the active status to the hold status.

(4) [Browse] menu

[CPU...] Opens the Cpu Window to display the CPU usage between the up cursor and the down cursor. While the Cpu Window is already opened and active, the contents of the window is updated.

The function of this item is same as that of the \(\text{CPU}\) button.

[Pattern...] Opens the [Pattern Set] dialog box. Setting pattern conditions using this dialog box opens the Pattern Window. While the Pattern Window is already opened and active, the contents of the window is updated.

The function of this item is same as that of the \(\text{PAT}\) button.

[Trace View...] Opens the Trace View Window.

While the Trace View Window is already opened and active, the contents of the window is updated.

The function of this item is same as that of the \(\text{TRACE}\) button.
(5) [Jump] menu

<table>
<thead>
<tr>
<th>File</th>
<th>View</th>
<th>Operation</th>
<th>Browse</th>
<th>Jump</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SourceText...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Assemble...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Memory...</td>
<td></td>
</tr>
</tbody>
</table>

[Source Text...] Displays the source text on the debugger for the range starting at the position set with the up temporary cursor. While the source text is already displayed on an active status window, the contents of the window is updated.

The function of this item is same as that of the button.

[Assemble...] Displays the disassemble text on the debugger for the range starting at the position set with the up temporary cursor. While the disassemble text is already displayed on an active status window, the contents of the window is updated.

The function of this item is same as that of the button.

[Memory...] Displays the memory list on the debugger for the range starting at the position set with the up temporary cursor. While the memory list is already displayed on an active status window, the contents of the window is updated.

The function of this item is same as that of the button.

[Caution] If there is no address information at the position specified with the up temporary cursor, none of the items on the [Jump] menu can be selected.

(6) [Help] menu

<table>
<thead>
<tr>
<th>File</th>
<th>View</th>
<th>Operation</th>
<th>Browse</th>
<th>Jump</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>This Window</td>
<td>F1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Help Topics</td>
<td></td>
</tr>
</tbody>
</table>

[This Window] Displays the help window for this window.

[Help Topics] Displays the AZ850 help window.
## Tool bar

The tool bar consists of buttons that can execute frequently used menu items with a single action.

<table>
<thead>
<tr>
<th>Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ]</td>
<td>Opens the [Open]/[Save As] dialog box. A previously saved display file, obtained using this window, is read. The default extension for the display file of this window is &quot;.az&quot;. The file name is displayed on the title bar after the display file has been read. Same operation as [File] -&gt; [Open...].</td>
</tr>
<tr>
<td>![ ]</td>
<td>Opens the [Open]/[Save As] dialog box. The current display information for this window is saved to a display file, which can either be created newly or by copying and renaming an existing file. The default extension for the display file of this window is &quot;.az&quot;. Same operation as [File] -&gt; [Save...].</td>
</tr>
<tr>
<td>![ ]</td>
<td>Toggles the grid mode of the execution transition map on and off.</td>
</tr>
<tr>
<td>![ ]</td>
<td>Changes the view mode of the execution transition map to the simple mode. Same operation as [View] menu -&gt; [View mode] -&gt; [Simple].</td>
</tr>
<tr>
<td>![ ]</td>
<td>Changes the view mode of the execution transition map to the standard mode. Same operation as [View] -&gt; [View mode] -&gt; [Standard].</td>
</tr>
<tr>
<td>![ ]</td>
<td>Changes the view mode of the execution transition map to the detailed mode. Same operation as [View] -&gt; [View mode] -&gt; [Detail].</td>
</tr>
<tr>
<td>![ ]</td>
<td>Toggles the equal mode of the execution transition map on and off. Same operation as [View] -&gt; [View mode] -&gt; [Equal].</td>
</tr>
<tr>
<td>![ ]</td>
<td>Opens the [Object Select] dialog box. Same operation as [View] -&gt; [Select Object...].</td>
</tr>
<tr>
<td>![ ]</td>
<td>Opens the [Pattern Search] dialog box. Same operation as [File] -&gt; [Find...].</td>
</tr>
<tr>
<td>![ ]</td>
<td>Opens the Cpu Window to display the CPU usage between the up cursor and the down cursor. While the Cpu Window is already opened and active, the contents of the window is updated. Same operation as [Browse] -&gt; [CPU...].</td>
</tr>
<tr>
<td>![ ]</td>
<td>Opens the [Pattern Set] dialog box. Setting pattern conditions using this dialog box opens the Pattern Window. While the Pattern Window is already opened and active, the contents of the window is updated. Same operation as [Browse] -&gt; [Pattern...].</td>
</tr>
<tr>
<td>![ ]</td>
<td>Opens the Trace View Window for the range starting at the position set with the up temporary cursor. While the Trace View Window is already opened and active, the contents of the window is updated. Same operation as [Browse] -&gt; [Trace View...].</td>
</tr>
<tr>
<td>![ ]</td>
<td>Displays the source text on the debugger for the range starting at the position set with the up temporary cursor. While the source text is already displayed on an active status window, the contents of the window is updated. If there is no address information at the position specified with the up temporary cursor, this button cannot be selected. Same operation as [Jump] -&gt; [Source Text...].</td>
</tr>
<tr>
<td>![ ]</td>
<td>Displays the disassemble text on the debugger for the range starting at the position set with the up temporary cursor. While the disassemble text is already displayed on an active status window, the contents of the window is updated. If there is no address information at the position specified with the up temporary cursor, this button cannot be selected. Same operation as [Jump] -&gt; [Assemble...].</td>
</tr>
</tbody>
</table>
About objects

(1) Object names

The buttons used to indicate objects are defined as follows:

- **RX850 objects**

  The name of the object specified by the system configuration file (CF definition file) of the RX850, such as the name of a task, event flag, 1-bit event flag, semaphore, mailbox, fixed-length memory pool, or variable-length memory pool, is displayed.

  If the object name is too long, part of it is omitted. The correct object name can be displayed by putting the mouse pointer on the object button.

  However, an idle task or a task whose ID is unknown (unidentified task) is displayed as follows:

  - If the program is executed in the middle of a task.
  - If the trace buffer is specified in the "ring buffer" type ([Ring Buffer] is specified in the [AZ Option] dialog box), and if a task appears at the beginning of the buffer after trace data has gone around the buffer.

- **RX850 Pro objects**

  The name of the object for a task is indicated by the function name and the task ID number that are specified by user, and the names of the other objects are indicated by resources and ID numbers.

  If the object name is too long, part of it is omitted. The correct object name can be displayed by putting the mouse pointer on the object button.

  ![Button](image)

  - Function name (xxxx: Task ID number)
  - Event flag (xxxx: Event flag ID number)
  - Semaphore (xxxx: Semaphore ID number)
  - Mailbox (xxxx: Mailbox ID number)
  - Memory pool (xxxx: Memory pool ID number)

  **[Caution]** Because the information of a function name is obtained from the target memory where the system management table of the RX850 Pro is located when uploading processing, the function name and the task ID number may not be correctly displayed.
However, an idle task or a task whose ID is unknown (unidentified task) is displayed as follows:

- **Idle**
  - Idle task

- **Task(????)**
  - Unidentified task

The task ID may not be able to be identified due to the following causes.

- If the program is executed in the middle of a task.
- When the trace buffer is specified in the “ring buffer” type ([Ring Buffer] is specified in the [AZ Option] dialog box), and if a task appears at the beginning of the buffer after trace data has gone around the buffer.

- **Interrupt objects**
  - The interrupt handler name obtained from the debugger is indicated. If the interrupt handler name cannot be obtained, the following default names are indicated.

- **Int(xxx)**
  - Maskable interrupt (xxx: Exception code (hexadecimal))

- **Other objects**
  - Object group that is set as hidden using the [Object Select] dialog box or mouse right-click menu.

(2) **Modifying the order in which objects are displayed**

The display sequence of each object can be changed by dragging the object with the mouse (the display sequence can also be changed by the Sort button on this window).

By right-clicking the object button, the following pop-up menu is displayed, and the selected object can be specified as hidden.

- **[Hide]**
  - Hides the selected object.

If the **Etc.** button that indicates an object group to be hidden is right-clicked, the following pop-up menu is displayed.

- **[Show All]**
  - Displays all objects.
- **[Show]**
  - Specifies the object to be displayed.
  - The cascade menu contains the list of the hidden objects.

**[Remark]** Modifying the order in which objects are displayed is also available in the [Object Select] dialog box.
Understanding the execution transition map

The execution transition map is displayed in three view modes: "simple mode", "standard mode", "detailed mode". In each of these display modes, "equal mode" can also be specified.

These view modes can be specified by selecting the [View] menu -> [View mode].

1. Simple mode
2. Standard mode
3. Detailed mode (default)
4. Equal mode

[Remark] Whether to display grid lines in the execution transition map can be specified by selecting the [View] menu -> [Grid mode].

(1) Simple mode

Only those points at which the CPU is operating are indicated, using horizontal lines. This mode is well suited to analyzing AZ trace data over a wide range.

The solid orange line indicates the processing of a system call of the RX850/RX850 Pro.

Figure 7-7 Analyze Window (Simple Mode)

(2) Standard mode

CPU state transitions are indicated using continuous, joined lines.

The solid orange line indicates the processing of a system call of the RX850/RX850 Pro.

Figure 7-8 Analyze Window (Standard Mode)
(3) Detailed mode (default)

In addition to the CPU state transitions, those RX850/RX850 Pro system calls that are related to managing memory pools and synchronous communications are indicated using "marks (refer to Table 7-7 or Table 7-8)". The detailed mode is selected when this window is opened. This mode is well suited to analyzing the accesses made to each object.

Figure 7-9 Analyze Window (Detailed Mode)

(a) The CPU accepts the interrupt in the Idle state (Halt state) and processing shifts to the handling of the interrupt.
(b) Processing shifts to the handling of Task2.
(c) Processing shifts to the handling of Task3.
(d) The CPU accepts an interrupt.
(e) Task2 issues a wai_flg system call to event flag EventFlag1.
   Because Task2 enters the wait state, processing shifts to the handling of Task3.
(f) Task3 issues a set_flg system call to event flag EventFlag1.
(g) The wai_flg system call issued by Task2 returns, so that the wait state is cleared.
(h) Task2 issues a get_blf system call to memory pool Mempool1.
(i) An error is returned in response to the system call issued in (h).
(j) Task3 issues a sig_sem system call to semaphore Semaphore1.
<table>
<thead>
<tr>
<th>Mark</th>
<th>Object to be Accessed</th>
<th>Issued System Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>light blue</td>
<td>Indicates that any of the following system call was issued to each object, and the corresponding returned.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Event flag</td>
<td>wai_flg, pol_flg, twai_flg, vwai_flg1, vtwai_flg1</td>
</tr>
<tr>
<td></td>
<td>Semaphore</td>
<td>wai_sem, twai_sem, preq_sem</td>
</tr>
<tr>
<td></td>
<td>Mailbox</td>
<td>rcv_msg, trcv_msg, prcv_msg</td>
</tr>
<tr>
<td></td>
<td>Variable-length memory block</td>
<td>get_blk, tget_blk, pget_blk</td>
</tr>
<tr>
<td></td>
<td>Fixed-length memory block</td>
<td>get_blk, tget_blk, pget_blk</td>
</tr>
<tr>
<td>green</td>
<td>Indicates that any of the following system call was issued to each object.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Event flag</td>
<td>set_flg, clr_flg, vset_flg1, vclr_flg1</td>
</tr>
<tr>
<td></td>
<td>Semaphore</td>
<td>sig_sem</td>
</tr>
<tr>
<td></td>
<td>Mailbox</td>
<td>snd_msg</td>
</tr>
<tr>
<td></td>
<td>Variable-length memory block</td>
<td>rel_blk</td>
</tr>
<tr>
<td></td>
<td>Fixed-length memory block</td>
<td>rel_blf</td>
</tr>
<tr>
<td>Red</td>
<td>Indicates that the issue of a system call results in an error.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Indicates that the system call that was issued has returned a timeout. This mark is displayed together with the red error mark above.</td>
<td></td>
</tr>
</tbody>
</table>
## Table 7-8  List of the Marks in the Analyze Window (RX850 Pro)

<table>
<thead>
<tr>
<th>Mark</th>
<th>Object to be Accessed</th>
<th>Issued System Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>light blue</td>
<td></td>
<td>Indicates that any of the following system call was issued to each object, and the corresponding returned.</td>
</tr>
<tr>
<td></td>
<td>Event flag</td>
<td>wai_flg, pol_flg, twai_flg</td>
</tr>
<tr>
<td></td>
<td>Semaphore</td>
<td>wai_sem, twai_sem, preq_sem</td>
</tr>
<tr>
<td></td>
<td>Mailbox</td>
<td>rcv_msg, trcv_msg, prcv_msg</td>
</tr>
<tr>
<td></td>
<td>Memory pool</td>
<td>get_blk, tget_blk, pget_blk</td>
</tr>
<tr>
<td>green</td>
<td></td>
<td>Indicates that any of the following system call was issued to each object.</td>
</tr>
<tr>
<td></td>
<td>Event flag</td>
<td>set_flg, clr_flg</td>
</tr>
<tr>
<td></td>
<td>Semaphore</td>
<td>sig_sem</td>
</tr>
<tr>
<td></td>
<td>Mailbox</td>
<td>snd_msg</td>
</tr>
<tr>
<td></td>
<td>Memory pool</td>
<td>rel_blk</td>
</tr>
<tr>
<td>Yellow</td>
<td></td>
<td>Indicates that any of the following system call was issued to each object.</td>
</tr>
<tr>
<td></td>
<td>Event flag</td>
<td>del_flg</td>
</tr>
<tr>
<td></td>
<td>Semaphore</td>
<td>del_sem</td>
</tr>
<tr>
<td></td>
<td>Mailbox</td>
<td>del_mbx</td>
</tr>
<tr>
<td></td>
<td>Memory pool</td>
<td>del_mpl</td>
</tr>
<tr>
<td>Red</td>
<td></td>
<td>Indicates that the issue of a system call results in an error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Indicates that the system call that was issued has returned a timeout. This mark is displayed together with the red error mark above.</td>
</tr>
</tbody>
</table>
(4) Equal mode

If the equal mode is not specified (default), the horizontal line between events, such as the entrance/exit of a system call, the entrance/exit of an interrupt or a task switch, are displayed in proportion to the execution time of the CPU, as shown in Figure 7-10.

Figure 7-10  Example of Transition Map When Equal Mode Is Not Specified

In contrast, if the equal mode is specified, the horizontal line between events, such as the entrance/exit of a system call, the entrance/exit of an interrupt or a task switch, are displayed not in proportion to the execution time of the CPU, but at a regular interval, as shown in Figure 7-11.

Figure 7-11  Example of Transition Map When Equal Mode Is Specified

This mode is well suited to analyzing the transition relationship between a task and interrupt processing. Even in an environment in which time information cannot be obtained (such as when an in-circuit emulator without time tag is used), the transition relationship between a task and interrupt processing can be analyzed by specifying this mode.
Verification method in execution transition map

To check the collected AZ trace data using this window, the following search methods are supported:

1. Search using the simple search button
2. Search using the [Pattern Search] dialog box
3. Search using the Pattern Window

(1) Search using the simple search button

Using the simple search button (button), a search can be made for the processing of a specified object. The simple search button is displayed at both ends of the transition map by clicking an object name. The operation method is as follows:

1. If the up temporary cursor is displayed in the execution transition map, move the up temporary cursor to the search start position. If the up temporary cursor is not displayed, move the up cursor to the search start position.
2. Clicking the button corresponding to the object whose operation is to be confirmed causes the simple search button to be displayed. The simple search button is displayed immediately to the right of the object name, and at the right edge of the execution transition map.
3. Click the simple search button.

If the identified point does not fall within the range currently displayed by the execution transition map, the map is shifted with the identified position being used as the origin.

When the object name is clicked again, the simple search button disappears from the screen.

(2) Search using the [Pattern Search] dialog box

A search can be made for a pattern set by the [Pattern Search] dialog box, in the execution transition map. The search result is indicated by the up temporary cursor on the execution transition map.

By using this dialog box, the transition state of a task/interrupt can be checked.

Refer to the [Pattern Search] dialog box for details.

(3) Search using the Pattern Window

A search can be made for a pattern set by the Pattern Window, in the execution transition map. The search result is indicated by the up/down temporary cursor on the execution transition map.

By using this window, the processing interval of a task/interrupt can be checked.

Refer to the Pattern Window for details.
Caution

- The data for the execution transition map is not updated automatically. Once trace data has been newly collected by executing an application program, therefore, uploading (selecting the [Operation] menu -> [Upload] on the Main Window) must be performed to update the execution transition map.

- The name of a task of the RX850 Pro is displayed as "function name (task ID number)", however, because the information of a function name is obtained from the target memory where the system management table of the RX850 Pro is located when uploading processing, the function name and the task ID number may not be correctly displayed when the dynamic generation processing of a task is performed.

- If there is no address information in AZ trace data at the position specified with the up temporary cursor, none of the items on the [Jump] menu can be selected.

Error

In the following cases, the ERROR MESSAGES is opened to display error messages.

- If an attempt is made to set the Analyze Window in the hold status to the active status when another Analyze Window in the active status exists.

- If the uploaded AZ trace data is different from the real-time OS set by the [AZ Option] dialog box
This dialog box is used to select the objects to be displayed in the Analyze Window and to modify the order in which those objects are displayed.

The function of this dialog box can also be realized by "sort button" in the Analyze Window (for details, refer to "About objects" on the Analyze Window).

This dialog box can be opened from the Analyze Window by any of the following.

- Select the [View] menu -> [Select Object...].
- Click the button on the tool bar.
- Press the [Alt], [V] and [O] keys in that order.

Figure 7-12  [Object Select] Dialog Box

This section describes the following items:

- Explanation of each area
- Function buttons
- Operating method
**Explanation of each area**

(a) **Hide objects**
   This area is used to display a list of those objects, from among AZ trace data, that are not to be displayed on the execution transition map.
   The horizontal scroll bar is available when a object name is too long. The vertical scroll bar is available when too many objects are listed.

(b) **Show objects**
   This area is used to display a list of those objects that are to be displayed on the execution transition map.
   In the default, all of the objects contained in AZ trace data are displayed.
   The display order specified in this area is applied to the execution transition map.
   The horizontal scroll bar is available when a object name is too long. The vertical scroll bar is available when too many objects are listed.

(c) **Sort mode**
   This area is used to specify the order into which the objects in the [Hide objects] or [Show objects] will be sorted.

**Function buttons**

<table>
<thead>
<tr>
<th>Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Show &gt;&gt;]</td>
<td>Moves those objects selected in the [Hide objects] to the [Show objects]. If the destination has already been specified, the object is inserted immediately ahead of the specified position. If the destination has not been specified, the object is inserted at the end of the list.</td>
</tr>
<tr>
<td>![&lt;&lt; Hide]</td>
<td>Moves those objects selected in the [Show objects] to the [Hide objects]. If the destination has already been specified, the object is inserted immediately ahead of the specified position. If the destination has not been specified, the object is inserted at the end of the list.</td>
</tr>
<tr>
<td>![OK]</td>
<td>Updates the execution transition map based on the objects listed in the [Show objects].</td>
</tr>
<tr>
<td>![Cancel]</td>
<td>Closes this dialog box.</td>
</tr>
<tr>
<td>![Help]</td>
<td>Displays the help window for this dialog box.</td>
</tr>
</tbody>
</table>
Operating method

(1) Restricting the objects to be displayed

The objects to be displayed in the execution transition map can be restricted by the following method.
From the list displayed in the [Show objects], select those objects that need not be displayed in the execution transition map.

Click the \[\text{Hide} \] button. The selected object is moved to the [Hide objects]. If the destination has already been specified, the object is inserted immediately ahead of the specified position. If the destination has not been specified, the object is inserted at the end of the list.

Click the \[\text{OK} \] button to update the execution transition map.

In the execution transition map, those objects placed in the [Hide objects] are displayed collectively as a [Etc.] object.

(2) Adding objects to be displayed

The objects to be displayed in the execution transition map can be added by the following method.
From those objects listed in the [Hide objects], select objects that need to add to the execution transition map.

Click the \[\text{Show} \] button. The selected objects are moved into the [Show objects]. If the destination has already been specified, the object is inserted immediately ahead of the specified position. If the destination has not been specified, the object is inserted at the end of the list.

Click the \[\text{OK} \] button to update the execution transition map.

(3) Modifying the order in which objects are displayed

Objects listed in the [Hide objects] and [Show objects] are classified into interrupts, tasks, idle, event flags, semaphores, mailboxes and memory pools (default).

To modify the order in which these objects are displayed, select the desired sort order from the drop-down list of the [Sort mode].

In the execution transition map, the display order specified in the [Show objects] is applied.

The following sort modes can be select:

<table>
<thead>
<tr>
<th>Sort mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No-sort mode (default)</td>
</tr>
<tr>
<td>Appear</td>
<td>In the order in which AZ trace data is detected</td>
</tr>
<tr>
<td>Name</td>
<td>According to ASCII code, within each object class</td>
</tr>
<tr>
<td>ID</td>
<td>According to ID, within each object class</td>
</tr>
<tr>
<td>Priority</td>
<td>According to priority (valid only when tasks are to be displayed)</td>
</tr>
</tbody>
</table>
**[Pattern Search] dialog box**

This dialog is used to set the search conditions that are applied when a search is made for an event (such as issue of a system call, occurrence of an interrupt, switching point of a task, and so on) in the execution transition map of the Analyze Window.

This dialog box can be opened from the Analyze Window by any of the following.

- Select the [View] menu -> [Find...].
- Click the button on the tool bar.
- Press the [Alt], [V] and [F] keys in that order.
- Press the [Ctrl] + [F] keys at the same time.

![Figure 7-13  [Pattern Search] Dialog Box](image)

This section describes the following items:

- Explanation of each area
- Function buttons
- Search method
Explanation of each area

(a) Search Mode
This area is used to select the search mode in which a pattern is to be searched.
The search mode indicates the type of an event to be searched. The contents displayed in the search condition setting area vary depending on this search mode.
Select the search mode from the following:

<table>
<thead>
<tr>
<th>Search Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Switch</td>
<td>Searches for task switching positions.</td>
</tr>
<tr>
<td>System Call (Call)</td>
<td>Searches for those positions where a system call was called.</td>
</tr>
<tr>
<td>System Call (Return)</td>
<td>Searches for those positions where a system call was returned.</td>
</tr>
<tr>
<td>System Call (Error)</td>
<td>Searches for those positions where an error was returned in response to a system call.</td>
</tr>
</tbody>
</table>

(b) Search condition setting area
This area is used to specify the search conditions corresponding to the specified search mode.
The items to be set depend on the specified search mode.
Select the search condition for each item from the drop-down list.
The following search conditions are available:

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning of Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task (upper)</td>
<td>Specify the pre-switching task name or interrupt name. When the specification of the task or interrupt is not critical, specify [* ANY *].</td>
</tr>
<tr>
<td>Task (lower)</td>
<td>Specify the post-switching task name or interrupt name. When the specification of the task or interrupt is not critical, specify [* ANY *].</td>
</tr>
<tr>
<td>Interrupt</td>
<td>Specify whether switching to (or from) an interrupt is to be included in the search object. Specifying [Valid] causes the interrupt transition to be used as a search object. When [Invalid] is specified, the interrupt transition is not used as a search object.</td>
</tr>
</tbody>
</table>
[Remark] The position for which a search is made vary with the specification of [Valid]/[Invalid] for the Interrupt item, as follows, where switching from [* ANY *] to [task2] is assumed to be specified as the search conditions:

(a)
Valid  Found as switching from [task1] to [task2]
Invalid Found as switching from [task1] to [task2]

(b)
Valid  Found as switching from [interrupt] to [task2]
Invalid Switching from interrupt handling is ignored but this position is found as switching from [task1] to [task2].

(c)
Valid  Found as switching from [interrupt] to [task2]
Invalid Switching from interrupt handling is ignored, so that this position is assumed to be switching from [task2] to [task2] and, therefore, is not found as switching which matches the conditions.

Table 7-13  Search Conditions ([Pattern Search] Dialog Box: When [ System Call(xxxx)] Is Selected)

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning of Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Specify the name of the task or interrupt that issues a system call. When the specification of the task or interrupt is not critical, specify [* ANY *].</td>
</tr>
<tr>
<td>System Call</td>
<td>Specify the name of the system call. When the specification of the system call is not critical, specify [* ANY *].</td>
</tr>
<tr>
<td>Object</td>
<td>Specify the target object of the system call. When the specification of the object is not critical, specify [* ANY *].</td>
</tr>
</tbody>
</table>
Function buttons

Table 7-14 Function Buttons of the [Pattern Search] Dialog Box

<table>
<thead>
<tr>
<th>Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search[Fore]</td>
<td>A search is made for those positions that correspond to the specified conditions, starting from the position indicated by the up cursor or up temporary cursor on the execution transition map, in the direction of the time axis. Then, the up temporary cursor is moved to the positions determined by the search.</td>
</tr>
<tr>
<td>Search[Back]</td>
<td>A search is made for those positions that correspond to the specified conditions, starting from the position indicated by the up cursor or up temporary cursor on the execution transition map, in the direction opposite to the time axis. Then, the up temporary cursor is moved to the positions determined by the search.</td>
</tr>
<tr>
<td>Close</td>
<td>Closes this dialog box.</td>
</tr>
<tr>
<td>Help</td>
<td>Displays the help window for this dialog box.</td>
</tr>
</tbody>
</table>

Search method

A search can be made for the desired position on the execution transition map by the following method.

1. Specifying the position from which a search is to be started:
   When the up temporary cursor is displayed in the execution transition map, move the up temporary cursor to the start position. Otherwise, move the up cursor to the start position.

2. Selecting a search mode:
   Select a desired search mode in the [Search Mode] on this dialog box.

3. Selecting search conditions:
   Select the search condition for each item form the drop-down list in the search condition setting area on this dialog box.

4. Clicking the function button:
   When the Search[Fore] button is clicked:
   Search starts from the specified position, in the direction of the time axis. The up temporary cursor is shifted to the position identified by the search.

   When the Search[Back] button is clicked:
   Search starts from the specified position, in the direction opposite to the time axis. The up temporary cursor is shifted to the position identified by the search.
This window is used to display the CPU usage status within the period specified with the up cursor and down cursor in the execution transition map of the Analyze Window.

Using this window, the execution time of a specified task or interrupt can be confirmed, and the target data performance can be evaluated, based on the idle time and interrupt time.

This window can be opened from the Analyze Window by any of the following:

- Select the [Browse] menu -> [CPU...].
- Click the button on the tool bar.
- Press the [Alt], [B] and [C] keys in that order.
- Press the [Ctrl] + [C] keys at the same time.

Figure 7-14 Cpu Window
This section describes the following items:

- Explanation of each area
- Menu bar
- Tool bar
- CPU usage display method
- Caution
- Error

**Explanation of each area**

(a) **Time display of up cursor position**
   This box indicates the time at which calculation of the CPU usage was started. After the start of AZ trace data collection, the relative time that has elapsed is indicated by the position of the up cursor on the execution transition map. The units are milliseconds [msec].

(b) **Time display of down cursor position**
   This box indicates the time at which calculation of the CPU usage was ended. After the start of AZ trace data collection, the relative time that has elapsed is indicated by the position of the down cursor on the execution transition map. The units are milliseconds [msec].

(c) **Inter-cursor time display**
   This box indicates the duration for which the CPU usage was calculated. The duration is indicated by the distance between the up and down cursors. The units are milliseconds [msec].

(d) **Object name**
   This area is used to display a list of the names of the objects that exist in the total time range. The horizontal scroll bar is available when the object name is too long.

(e) **Time percentage display of task execution**
   This area is used to display the time required for task execution as a proportion of the total time.

(f) **Total execution time for an object**
   This area is used to display the total of the system processing time and the time required to execute a task or that of a task related to interrupt processing in the total time range.

(g) **Graphical display of CPU usage**
   This area is used to display the CPU usage for the objects that exist in the total time range. The displayed graphs are broken down into two part: [Task Run:] and [System:].

(h) **Time percentage display of system execution**
   This area is used to display the proportion of the system execution time (including the execution time of the interrupt handler) to the total time.
Menu bar

(1) [File] menu

- **[Open…]** Opens the [Open]/[Save As] dialog box. A previously saved display file, obtained using this window, is read. The default extension for the display file of this window is ".azc". The file name is displayed on the title bar after the display file has been read.
- **[Save…]** Opens the [Open]/[Save As] dialog box. The current display information for this window is saved to a display file, which can either be created newly or by copying and renaming an existing file. The default extension for the display file of this window is ".azc".
- **[Close]** Closes this window.

(2) [View] menu

- **[Sort Appear]** Displays the objects in the sequence in which they are detected as AZ trace data. The function of this item is same as that of the button.
- **[Sort Name]** Displays the objects in alphabetical order. The function of this item is same as that of the button.
- **[Sort Time]** Displays the objects in ascending order of execution time (default). The function of this item is same as that of the button.
- **[Sort Analyze]** Displays the objects in same order as the execution transition map. The function of this item is same as that of the button.

(3) [Operation] menu

- **[Active]** Switches this window from the hold status to the active status (default).
- **[Hold]** Switches this window from the active status to the hold status.
(4) [Jump] menu

[Pattern...] Opens the Pattern Window corresponding to the specified object. While the Pattern Window in the active status is already opened, the contents of the window is updated.

The function of this item is same as that of the button. The pattern calculated with this menu item is automatically set as follows:
<Start point>: Task Switch, [* ANY *] -> [Specified object]
<End point>: Task Switch, [Specified object] -> [* ANY *]
<Interrupt>: Valid

(5) [Help] menu

[This Window] Displays the help window for this window.
[Help Topics] Displays the AZ850 help window.
Tool bar

The tool bar consists of buttons that can execute frequently used menu items with a single action.

Table 7-15 Tool Bar of Cpu Window

<table>
<thead>
<tr>
<th>Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Open/Save As]</td>
<td>Opens the [Open]/[Save As] dialog box. A previously saved display file, obtained using this window, is read. The default extension for the display file of this window is &quot;.azc&quot;. The file name is displayed on the title bar after the display file has been read. Same operation as [File] -&gt; [Open...].</td>
</tr>
<tr>
<td>![Open/Save As]</td>
<td>Opens the [Open]/[Save As] dialog box. The current display information for this window is saved to a display file, which can either be created newly or by copying and renaming an existing file. The default extension for the display file of this window is &quot;.azc&quot;. Same operation as [File] -&gt; [Save...].</td>
</tr>
<tr>
<td>![List Objects]</td>
<td>Displays the objects in the sequence in which they are detected as AZ trace data. Same operation as [View] -&gt; [Sort Appear].</td>
</tr>
<tr>
<td>![List Objects]</td>
<td>Displays the objects in alphabetical order. Same operation as [View] -&gt; [Sort Name].</td>
</tr>
<tr>
<td>![List Objects]</td>
<td>Displays the objects in ascending order of execution time (default). Same operation as [View] -&gt; [Sort Time].</td>
</tr>
<tr>
<td>![List Objects]</td>
<td>Displays the objects in the same order as the Analyze Window. Same operation as [View] -&gt; [Sort Analyze].</td>
</tr>
<tr>
<td>![Pattern Window]</td>
<td>Opens the Pattern Window corresponding to the specified object. While the Pattern Window in the active status is already opened, the contents of the window is updated. &lt;Start point&gt;: Task Switch, [* ANY <em>] -&gt; [Specified object] &lt;End point&gt;: Task Switch, [Specified object] -&gt; [</em> ANY *] &lt;Interrupt&gt;: Valid Same operation as [Jump] -&gt; [Pattern...].</td>
</tr>
</tbody>
</table>

CPU usage display method

The CPU usage is calculated within the period specified with the up cursor and down cursor in the execution transition map, as follows:

(1) Setting the up and down cursors:
   In the execution transition map, specify the range for which the total CPU usage is to be calculated, using the up cursor and down cursor.

(2) Opening the Cpu Window:
   Select the [Browse] menu -> [CPU...] on the Analyze Window to open this window.

   If the Cpu Window is left open and either the up cursor or down cursor is repositioned in the execution transition map, the displayed CPU usage is automatically updated.
Caution

- Closing the Analyze Window also closes this window.
- If an [Etc.] object is selected, the [Jump] menu -> [Pattern...] cannot be selected.

Error

In the following cases, the ERROR MESSAGES is opened to display error messages.

- If an attempt is made to set the Cpu Window in the hold status to the active status when another Cpu Window in the active status exists.
This dialog box is used to set the pattern modes and pattern conditions for the pattern to be displayed in the Pattern Window.

The pattern means the duration of the CPU processing from the point at which an event (such as a task switching, an interrupt occurrence, a system call status, etc.) occurs to the point at which the event ends or another event occurs.

This dialog box can be opened as follows:

In the Analyze Window

- Select the [Browse] menu -> [Pattern...].
- Click the button on the tool bar.
- Press the [Alt], [B] and [P] keys in that order.
- Press the [Ctrl]+[P] keys at the same time.

In the Pattern Window

- Select the [Option] menu -> [Pattern Set...]
- Click the button on the tool bar.
- Press the [Alt], [P] and [S] keys in that order.
- Press the [Ctrl]+[P] keys at the same time.

[Caution] The contents of the pattern condition setting area depend on the specified pattern mode.
This section describes the following items:

- **Explanation of each area**
- **Function buttons**
- **Pattern search/calculation method**
- **Error**

**Explanation of each area**

(a) **Pattern mode selection area**

This area is used to select the pattern mode. Select the type of the event that is to act as the start ([From:]) and end ([To:]) points of the pattern displayed in the **Pattern Window**, from the following:

<table>
<thead>
<tr>
<th>Pattern Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Switch</td>
<td>Task switching positions are used as the start/end point.</td>
</tr>
<tr>
<td>System Call (Call)</td>
<td>Those positions where a system call was called are used as the start/end point.</td>
</tr>
<tr>
<td>System Call (Return)</td>
<td>Those positions where a system call was returned are used as the start/end point.</td>
</tr>
<tr>
<td>System Call (Error)</td>
<td>Those positions where an error was returned in response to a system call are used as the start/end point.</td>
</tr>
</tbody>
</table>

(b) **Pattern condition setting area**

This area is used to specify the pattern conditions corresponding to the pattern modes specified for the start and end points. The items to be set depend on the specified pattern modes. Select the pattern condition for each item from the drop-down list. The following search conditions are available:

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning of Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task (upper)</td>
<td>Specify the pre-switching task name or interrupt name. When the specification of the task or interrupt is not critical, specify [* ANY *].</td>
</tr>
<tr>
<td>Task (lower)</td>
<td>Specify the post-switching task name or interrupt name. When the specification of the task or interrupt is not critical, specify [* ANY *].</td>
</tr>
<tr>
<td>Interrupt</td>
<td>Specify whether switching to (or from) an interrupt is to be included in the search object. Specifying [Valid] causes the interrupt transition to be used as a search object. When [Invalid] is specified, the interrupt transition is not used as a search object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning of Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Specify the name of the task or interrupt that issues a system call. When the specification of the task or interrupt is not critical, specify [* ANY *].</td>
</tr>
<tr>
<td>System Call</td>
<td>Specify the name of the system call. When the specification of the system call is not critical, specify [* ANY *].</td>
</tr>
<tr>
<td>Object</td>
<td>Specify the target object name of the system call. When the specification of the object is not critical, specify [* ANY *].</td>
</tr>
</tbody>
</table>
[Remark] The position for which a search is made vary with the specification of [Valid]/[Invalid] for the Interrupt item, as follows:


Figure 7-16 Difference between [Valid] and [Invalid]

When [Valid] is specified
Interrupts are detected as switching objects, so that (a) and (b) are detected as the pattern.

When [Invalid] is specified
Interrupts are not detected as switching objects, so that (c) is detected as the pattern.
In this case, the time at the exit of the interrupt is assumed as the pattern start/end time, and the calculated time required for executing the pattern will include the time required for handling the interrupt.
Examples of setting typical patterns are shown below.

- **Analyzing the interrupt handling time**
  Analyzing the interrupt "Interrupt1"

<table>
<thead>
<tr>
<th>Area</th>
<th>Pattern Mode</th>
<th>Pattern condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>Task Switch</td>
<td>[* ANY *] -&gt; [Interrupt1] ; [Valid]</td>
</tr>
<tr>
<td>To</td>
<td>Task Switch</td>
<td>[Interrupt1] -&gt; [* ANY *] ; [Valid]</td>
</tr>
</tbody>
</table>

- **Analyzing the time required for processing a system call**
  Analyzing the processing time between "Task1" issuing a wai_sem system call and "semaphore1" being acquired

<table>
<thead>
<tr>
<th>Area</th>
<th>Pattern Mode</th>
<th>Pattern Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>System Call (Call)</td>
<td>[Task1] -&gt; [wai_sem] ; [Semaphore1]</td>
</tr>
<tr>
<td>To</td>
<td>System Call (Return)</td>
<td>[Task1] -&gt; [wai_sem] ; [Semaphore1]</td>
</tr>
</tbody>
</table>

- **Analyzing the processing time between a system call being issued and another task being woken up**
  Analyzing the processing time between "Task1" issuing a wai_tsk system call and processing being passed to "Task2"

<table>
<thead>
<tr>
<th>Area</th>
<th>Pattern Mode</th>
<th>Pattern Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>System Call (Call)</td>
<td>[Task1] -&gt; [wai_tsk]</td>
</tr>
<tr>
<td>To</td>
<td>Task Switch</td>
<td>[* ANY *] -&gt; [Task2]</td>
</tr>
</tbody>
</table>

- **Analyzing the interval between error returns**
  Analyzing the interval between the locations from which an error is returned, by "Task1"

<table>
<thead>
<tr>
<th>Area</th>
<th>Pattern Mode</th>
<th>Pattern Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>System Call (Error)</td>
<td>[Task1] -&gt; [* ANY <em>] ; [</em> ANY *]</td>
</tr>
<tr>
<td>To</td>
<td>System Call (Error)</td>
<td>[Task1] -&gt; [* ANY <em>] ; [</em> ANY *]</td>
</tr>
</tbody>
</table>

**Function buttons**

<table>
<thead>
<tr>
<th>Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>Searches for the set pattern and calculates its distribution within the range specified by the up cursor and down cursor in the execution transition map, then opens the Pattern Window.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Closes this dialog box.</td>
</tr>
<tr>
<td>Help</td>
<td>Displays the help window for this dialog box.</td>
</tr>
</tbody>
</table>
Pattern search/calculation method

By the following method, a search can be made for the set pattern within the specified range to open the Pattern Window.

(1) Specify the range for which the pattern distribution state is calculated, using the up cursor and down cursor on the execution transition map. Then, open this dialog box.

(2) Select the pattern mode corresponding to the start point of the pattern to be calculated, from the [From:] items.

(3) Select the pattern mode corresponding to the end point of the pattern to be calculated, from the [To:] items.

(4) Specify the pattern conditions corresponding to the selected pattern mode from each drop-down list.

(5) Click the button.

Error

In the following cases, the ERROR MESSAGES is opened to display error messages.

- If a pattern which satisfies the set pattern conditions is not found in AZ trace data
Pattern Window

This window is used to search for the pattern specified using the [Pattern Set] dialog box within the range set with the up cursor and down cursor on the execution transition map, and display a histogram representing the number of times the pattern appears for given execution duration.

The pattern means the duration of the CPU processing from the point at which an event (such as a task switching, an interrupt occurrence, a system call status, etc.) occurs to the point at which the event ends or another event occurs. Using this window, therefore, the worst/average execution time for the processing performed by an application program can be obtained.

This window can be opened as follows:

[In the Analyze Window]
- Select the [Browse] menu -> [Pattern...], and specify the pattern conditions in the [Pattern Set] dialog box.
- Click the button on the tool bar, and specify the pattern conditions in the [Pattern Set] dialog box.
- Press the [Alt], [B] and [P] keys in that order, and specify the pattern conditions in the [Pattern Set] dialog box.
- Press the [Ctrl]+[P] keys at the same time, and specify the pattern conditions in the [Pattern Set] dialog box.

[In the Cpu Window]
- Select the [Jump] menu -> [Pattern...].
- Click the button on the tool bar.
- Press the [Ctrl]+[P] keys at the same time.

![Pattern Window Diagram]

Figure 7-17 Pattern Window

(a) Time display of up cursor position  (e) Indication of average value
(b) Time display of down cursor position  (f) Indication of number of histogram divisions
(c) Inter-cursor time display  (g) Processing time display area
(d) Indication of set pattern conditions  (h) Pattern distribution display area
This section describes the following items:

- Explanation of each area
- Menu bar
- Tool bar
- Pattern distribution display method
- Pattern distribution viewing method
- Search from Pattern Window
- Caution
- Error

**Explanation of each area**

(a) Time display of up cursor position
   This box indicates the time at which calculation of the pattern distribution started.
   After the start of AZ trace data collection, the relative time that has elapsed is indicated by the position of the up cursor on the execution transition map. The units are milliseconds [msec].

(b) Time display of down cursor position
   This box indicates the time at which calculation of the pattern distribution ended.
   After the start of AZ trace data collection, the relative time that has elapsed is indicated by the position of the down cursor on the execution transition map. The units are milliseconds [msec].

(c) Inter-cursor time display
   The up and down cursor indicate the duration for which the pattern distribution was calculated.
   The duration is indicated by the distance between the up and down cursors. The units are milliseconds [msec].

(d) Indication of set pattern conditions
   This area is used to display the contents of the pattern conditions set with the [Pattern Set] dialog box.
   Putting the mouse pointer on this area displays pop-up window that shows the name of the object (if the object name is too long, however, part of it is omitted).

(e) Indication of average value
   This area is used to display the average time required to execute the set pattern. The units are milliseconds [msec].

(f) Indication of number of histogram divisions
   This area is used to display the number of histogram divisions. By clicking the button on the right, the number of divisions can be changed to any value between 1 and 100.

(g) Processing time display area
   This area is used to display the duration in which the set pattern is processed.
   When the number of the histogram divisions is changed, the time width is also changed.

(h) Pattern distribution display area
   This area is used to display a histogram of the obtained processing times for the set pattern that specified with the up cursor and down cursor of the execution transition map.
Menu bar

(1) [File] menu

<table>
<thead>
<tr>
<th>File</th>
<th>View</th>
<th>Option</th>
<th>Operation</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open..</td>
<td>Ctrl+O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save..</td>
<td>Ctrl+S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Close</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Open...]  Opens the [Open]/[Save As] dialog box. A previously saved display file, obtained using this window, is read.
The default extension for the display file of this window is ".azp".
The file name is displayed on the title bar after the display file has been read.

[Save...]  Opens the [Open]/[Save As] dialog box. The current display information for this window is saved to a display file, which can either be created newly or by copying and renaming an existing file.
The default extension for the display file of this window is ".azp".

[Close]  Closes this window.

(2) [View] menu

<table>
<thead>
<tr>
<th>File</th>
<th>View</th>
<th>Option</th>
<th>Operation</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Divide(+)(I)</td>
<td>Ctrl+[</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Divide(-)(D)</td>
<td>Ctrl+]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Divide(+)(I)]  Increases the number of histogram divisions by 1.

[Divide(-)(D)]  Reduces the number of histogram divisions by 1.

(3) [Option] menu

<table>
<thead>
<tr>
<th>File</th>
<th>View</th>
<th>Option</th>
<th>Operation</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pattern Set..</td>
<td>Ctrl+P</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Pattern Set..]  Opens the [Pattern Set] dialog box.
The function of this item is same as that of the button.
(4) [Operation] menu

<table>
<thead>
<tr>
<th>File</th>
<th>View</th>
<th>Option</th>
<th>Operation</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Active</td>
<td>Ctrl+A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hold</td>
<td>Ctrl+H</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Search(Min)</td>
<td>Ctrl+N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Search(Max)</td>
<td>Ctrl+K</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Search(Fore)</td>
<td>Ctrl+F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Search(Back)</td>
<td>Ctrl+B</td>
</tr>
</tbody>
</table>

[Active] Switches this window from the hold status to the active status (default).

[Hold] Switches this window from the active status to the hold status.

[Search(Min)] Indicates, on the execution transition map, the location where the execution duration for the specified pattern is minimum. The up temporary cursor is moved to the identified pattern start point, while the down temporary cursor is moved to the identified pattern end point. The function of this item is same as that of the button.

[Search(Max)] Indicates, on the execution transition map, the location where the execution duration for the specified pattern is maximum. The up temporary cursor is moved to the identified pattern start point, while the down temporary cursor is moved to the identified pattern end point. The function of this item is same as that of the button.

[Search(Fore)] Searches the pattern with the next longer execution time compared to the currently searched pattern after selecting the [Search(Min)] or the button on the tool bar. The up temporary cursor is moved to the identified pattern start point, while the down temporary cursor is moved to the identified pattern end point. The function of this item is same as that of the button.

[Search(Back)] Searches the pattern with the next shorter execution time compared to the currently searched pattern after selecting the [Search(Max)] or the button on the tool bar. The up temporary cursor is moved to the identified pattern start point, while the down temporary cursor is moved to the identified pattern end point. The function of this item is same as that of the button.

(5) [Help] menu

<table>
<thead>
<tr>
<th>File</th>
<th>View</th>
<th>Option</th>
<th>Operation</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>This Window</td>
<td>F1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Help Topics</td>
<td></td>
</tr>
</tbody>
</table>

[This Window] Displays the help window for this window.

[Help Topics] Displays the AZ850 help window.
Tool bar

The tool bar consists of buttons that can execute frequently used menu items with a single action.

Table 7-20 Tool Bar of the Pattern Window

<table>
<thead>
<tr>
<th>Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Open][Save] dialog box</td>
<td>Opens the [Open]/[Save As] dialog box. A previously saved display file, obtained using this window, is read. The default extension for the display file of this window is &quot;.azp&quot;. The file name is displayed on the title bar after the display file has been read. Same operation as [File] -&gt; [Open...].</td>
</tr>
<tr>
<td>![Open][Save] dialog box</td>
<td>Opens the [Open]/[Save As] dialog box. The current display information for this window is saved to a display file, which can either be created newly or by copying and renaming an existing file. The default extension for the display file of this window is &quot;.azp&quot;. Same operation as [File] -&gt; [Save...].</td>
</tr>
<tr>
<td>![Pattern Set] dialog box</td>
<td>Opens the [Pattern Set] dialog box. Same operation as [Option] -&gt; [Pattern Set...].</td>
</tr>
<tr>
<td>![Search(Min)]</td>
<td>Indicates, on the execution transition map, the location where the execution duration for the specified pattern is minimum. The up temporary cursor is moved to the identified pattern start point, while the down temporary cursor is moved to the identified pattern end point. Same operation as [Operation] -&gt; [Search(Min)].</td>
</tr>
<tr>
<td>![Search(Back)]</td>
<td>Searches for the location where the execution duration for the pattern is the largest, after that identified by the previous search. The up temporary cursor is moved to the identified pattern start point, while the down temporary cursor is moved to the identified pattern end point. Same operation as [Operation] -&gt; [Search(Back)].</td>
</tr>
<tr>
<td>![Search(Fore)]</td>
<td>Searches for the location where the execution duration for the pattern is the smallest, after that identified by the previous search. The up temporary cursor is moved to the identified pattern start point, while the down temporary cursor is moved to the identified pattern end point. Same operation as [Operation] -&gt; [Search(Fore)].</td>
</tr>
<tr>
<td>![Search(Max)]</td>
<td>Indicates, on the execution transition map, the location where the execution duration for the specified pattern is maximum. The up temporary cursor is moved to the identified pattern start point, while the down temporary cursor is moved to the identified pattern end point. Same operation as [Operation] -&gt; [Search(Max)].</td>
</tr>
</tbody>
</table>

Pattern distribution display method

The distribution of a pattern is calculated for the period specified with the up cursor and down cursor in the execution transition map, as follows:

(1) In the execution transition map, specify the range for which pattern distribution is to be calculated, using the up cursor and down cursor.

(2) Select the [Browse] menu -> [Pattern...] on the Analyze Window to open the [Pattern Set] dialog box.

(3) Specify the pattern conditions and click the ![Ok] button on the [Pattern Set] dialog box.

If the Pattern Window is left open and either the up cursor or down cursor is repositioned in the execution transition map, the displayed distribution of the pattern is automatically updated.
Pattern distribution viewing method

Figure 7-18 shows the example for analysis of the processing time for "Task1".

**Figure 7-18 Pattern Distribution Viewing Method**

Indicates that the average processing time for "Task1" is 0.155 [msec].

<table>
<thead>
<tr>
<th>From Task Switch</th>
<th>Task Switch</th>
<th>To Task Switch</th>
<th>Interrupt Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Switch</td>
<td>ANY</td>
<td>Task1</td>
<td>Interrupt Valid</td>
</tr>
</tbody>
</table>

Indicates that the execution duration by dividing the range between the maximum and minimum processing times for "Task1" by the value of [Divide].

Indicates that there are 12 positions at which the processing for "Task1" required a time of between 0.110350 and 0.200940 [msec].

Search from Pattern Window

Using the Tool bar on this window, a search can be made in the calculated pattern distribution, based on the processing time (such as a search for those locations where the pattern execution duration is a maximum or a minimum). The result of the search is displayed in the execution transition map.

The up temporary cursor is moved to the identified pattern start point, while the down temporary cursor is moved to the identified pattern end point.

Caution

- Closing the Analyze Window also closes this window.

- If this window is in the active status and either the up cursor or down cursor is repositioned in the Analyze Window, the contents of this window is automatically updated.

Error

In the following cases, the ERROR MESSAGES is opened to display error messages.

- If an attempt is made to set the Pattern Window in the hold status to the active status when another Pattern Window in the active status exists.
Trace View Window

This window is used to list the contents of AZ trace data, displayed as an execution transition map in the Analyze Window.

The detailed information about AZ trace data can be obtained by using this window.

This window can be opened from the Analyze Window by any of the following.

- Select the [Browse] menu -> [Trace View...].
- Click the button on the tool bar.
- Press the [Alt], [B] and [T] keys in that order.
- Press the [Ctrl]+[T] keys at the same time.

Figure 7-19 Trace View Window

This section describes the following items:

- Explanation of each area
- Menu bar
- AZ trace data display method
- AZ trace data viewing method
- AZ trace data search method
- Caution
- Error
Explanation of each area

(a) Time
This area is used to display the relative time from the AZ trace start time. The units are milliseconds [msec].

(b) Address
This area is used to display the execution address of the application program.

(c) Task
This area is used to display the name of a task (a function in the case of the RX850 Pro) or the name of an interrupt upon occurrence of an event.
Refer to "About objects" in the Analyze Window for details on the displaying of names.

(d) Event
This area is used to display the type of an event in AZ trace data.
The types of events that can be displayed are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syscall</td>
<td>Indicates the issue of a system call. The name of the issued system call is indicated in parentheses.</td>
</tr>
<tr>
<td>SysRet</td>
<td>Indicates a return form a system call. The name of the returned system call is indicated in parentheses. If the event issued by corresponding system call dose not exist in the trace buffer, however, inside the parentheses is blank. This happens if the application program is executed from the middle.</td>
</tr>
<tr>
<td>Int</td>
<td>Indicates the entry of an interrupt. The name of the interrupt request is indicated in parentheses.</td>
</tr>
<tr>
<td>IntRet</td>
<td>Indicates the exit of an interrupt. The name of the interrupt request is indicated in parentheses.</td>
</tr>
<tr>
<td>TaskStart</td>
<td>Indicates the initial start up of a task.</td>
</tr>
<tr>
<td>Idle</td>
<td>Indicates transition to the Idle state.</td>
</tr>
</tbody>
</table>

(e) Parameter
This area is used to display the name of the target object of the system call when the event is a system call (Call).

(f) Return
This area is used to display the return value form the system call in a macro name when the event is a system call (Return).
For details of the return value, refer to the user's manual of the real-time OS used.
Menu bar

(1) [File] menu

[Save...] Opens the [Open]/[Save As] dialog box. The current display information for this window is saved to a display file, which can either be created newly or by copying and renaming an existing file. The default extension for the display file of this window is ".azt".

[Close] Closes this window.

(2) [View] menu

[Find...] Opens the [Trace Search] dialog box.

[Time tag] Toggles the display of the "Time" area between on and off. Select either the [Show] (default) or [Hide], displayed in the cascade menu.

[Address] Toggles the display of the "Address" area between on and off. Select either the [Show] (default) or [Hide], displayed in the cascade menu.

[Task] Toggles the display of the "Task" area between on and off. Select either the [Show] (default) or [Hide], displayed in the cascade menu.

[Event] Toggles the display of the "Event" area between on and off. Select either the [Show] (default) or [Hide], displayed in the cascade menu.

[Parameter] Toggles the display of the "Parameter" area between on and off. Select either the [Show] (default) or [Hide], displayed in the cascade menu.

[Return] Toggles the display of the "Return" area between on and off. Select either the [Show] (default) or [Hide], displayed in the cascade menu.

(3) [Operation] menu

[Active] Switches this window from the hold status to the active status (default).

[Hold] Switches this window from the active status to the hold status.
(4) [Help] menu

<table>
<thead>
<tr>
<th>File</th>
<th>View</th>
<th>Operation</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This Window</td>
<td>F1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Help Topics</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[This Window] Displays the help window for this window.
[Help Topics] Displays the AZ850 help window.

AZ trace data display method

The list of AZ trace data is displayed from the position specified on the execution transition map.
The operation method is as follows:

(1) Specify the start position from which AZ trace data is to be displayed.
   In the Analyze Window, move the up temporary cursor to the desired position. When the up temporary cursor is
   not displayed, the beginning of AZ trace data becomes the display start position.

(2) Select the [Browse] menu -> [Trace View...] on the Analyze Window to open this window.
   If the Trace View Window is left open and the up temporary cursor is repositioned in the execution transition
   map, the displayed AZ trace data is automatically updated.

AZ trace data viewing method

Figure 7-20 describes how to read the AZ trace data.

Figure 7-20 AZ Trace Data Viewing Method

<table>
<thead>
<tr>
<th>Time</th>
<th>Address</th>
<th>Task</th>
<th>Event</th>
<th>Parameter</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.209</td>
<td>00000204</td>
<td>TSK_MAIN</td>
<td>Syscall [ref_tsk]</td>
<td>TSK_MAIN</td>
<td></td>
</tr>
<tr>
<td>0.211</td>
<td>00000204</td>
<td>TSK_MAIN</td>
<td>SysRet [ref_tsk]</td>
<td>TSK_MAIN</td>
<td>E_OK</td>
</tr>
<tr>
<td>0.292</td>
<td>00000212</td>
<td>TSK_MAIN</td>
<td>Syscall [sta_tsk]</td>
<td>TSK_SUB</td>
<td></td>
</tr>
<tr>
<td>0.297</td>
<td>00000212</td>
<td>TSK_MAIN</td>
<td>SysRet [sta_tsk]</td>
<td>TSK_SUB</td>
<td>E_OK</td>
</tr>
<tr>
<td>0.384</td>
<td>00000220</td>
<td>TSK_MAIN</td>
<td>SysRet [chg_prf]</td>
<td>TSK_SUB</td>
<td>E_OK</td>
</tr>
<tr>
<td>0.465</td>
<td>00000222</td>
<td>TSK_MAIN</td>
<td>Syscall [rot_rdq]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) When 0.292 [msec] has elapsed after the start of the system, task "TSK_MAIN" issued a "sta_tsk" system call
to task "TSK_SUB" at address 0x000212.

(b) When 0.297 [msec] has elapsed after the start of the system, the previously issued "sta_tsk" system call
returned with a return value of "E_OK(0000)" at address 0x00212.

[Caution] Regarding "Event" area, if no event for which a system call was issued exists in the trace buffer, the
parentheses include no data. This is caused by execution from a location other than the beginning of the
application program, etc.
AZ trace data search method

Using the [Trace Search] dialog box, opened from this window, a search can be made according to conditions such as "execution task/interrupt names, event types, and system call arguments". The operation method is as follows:

(1) Select the [View] menu -> [Find...] to open the [Trace Search] dialog box. Then, specify the search conditions in the dialog box.

(2) The search in the new trace time direction is started by clicking the [Search(Forw)] button in the dialog box, and the search in the old trace time direction is started by clicking the [Search(Back)] button. The display starts from the position identified by the search.

Caution

- Closing the Analyze Window also closes this window.

- If this window is in the active status and the up temporary cursor is repositioned in the Analyze Window, the displayed AZ trace data is shifted automatically.

Error

In the following cases, the ERROR MESSAGES is opened to display error messages.

- If an attempt is made to set the Trace View Window in the hold status to the active status when another Trace View Window in the active status exists.
This dialog box is used to specify the search conditions applied when searching for AZ trace data with the Trace View Window.

This dialog box can be opened from the Trace View Window by any of the following.

- Select the [View] menu -> [Find...].
- Press the [Alt], [V] and [F] keys in that order.
- Press the [Ctrl]+[F] keys at the same time.

This section describes the following items:

- Explanation of each area
- Function buttons
- AZ trace data search method

**Explanation of each area**

**(a) Search item setting area**

This area is used to specify the search items for AZ trace data in the Trace View Window.
Select the check box corresponding to the desired search items. Any number of boxes can be selected.

- Task: Searches for a task/ interrupt name.
- Event: Searches for an event type.
- Parameter: Searches for a system call argument.

**(b) Search condition setting area**

This area is used to specify the search conditions for each search item.
The drop-down list displays the task/interrupt names, event types, and system call arguments that exist in the collected AZ trace data.
Function buttons

Table 7-22 Function Buttons of the [Trace Search] Dialog Box

<table>
<thead>
<tr>
<th>Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search(Fore)</td>
<td>A search is made for a position which satisfies the set search conditions, in the direction of the trace time axis, and the Trace View Window is displayed with the search position as the beginning.</td>
</tr>
<tr>
<td>Search(Back)</td>
<td>A search is made for a position which satisfies the set search conditions, in the direction opposite to the trace time axis, and the Trace View Window is displayed with the search position as the beginning.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Closes this dialog box.</td>
</tr>
<tr>
<td>Help</td>
<td>Displays the help window for this dialog box.</td>
</tr>
</tbody>
</table>

AZ trace data search method

A search can be made for the desired position in the Trace View Window by the following method.

(1) In the search item setting area, select the items to be searched. At this time, any number of items can be selected.

(2) In the search condition setting area, specify the search conditions for each search item by using the drop-down lists.

(3) Click the function button.

When the Search(Fore) button is clicked:
A search is made for a position which satisfies the set search conditions, in the direction of the trace time axis, and the Trace View Window is displayed with the search position as the beginning.

When the Search(Back) button is clicked:
A search is made for a position which satisfies the set search conditions, in the direction opposite to the trace time axis, and the Trace View Window is displayed with the search position as the beginning.
Examples of setting search conditions in the [Trace Search] dialog box is shown below.

Figure 7-22 Example of Trace Search Setting 1
(To search for the task "TSK_MAIN")

<table>
<thead>
<tr>
<th>Task</th>
<th>Event</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSK_MAIN</td>
<td>sta_tsk</td>
<td>TSK_MAIN</td>
</tr>
</tbody>
</table>

Figure 7-23 Example of Trace Search Setting 2
(To search for those locations where the task "TSK_MAIN" has issued a "sta_tsk" system call.)

<table>
<thead>
<tr>
<th>Task</th>
<th>Event</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSK_MAIN</td>
<td>sta_tsk</td>
<td>TSK_MAIN</td>
</tr>
</tbody>
</table>

Figure 7-24 Example of Trace Search Setting 3
(To search for those locations where the task "TSK_MAIN" has issued a "sta_tsk" system call for the task "TSK_SUB").

<table>
<thead>
<tr>
<th>Task</th>
<th>Event</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSK_MAIN</td>
<td>sta_tsk</td>
<td>TSK_SUB</td>
</tr>
</tbody>
</table>
This dialog box is used to display the version information of the AZ850. This dialog box can be opened from the Main Window by any of the following:

- Select the [Help] menu -> [About...].
- Press the [Ctrl]+[A] keys at the same time.

Figure 7-25 [About] Dialog Box

This section describes the following items:

- Explanation of each area
- Function buttons

### Explanation of each area

(a) Version information display
This area is used to display "the product name, version number of the AZ850, [date of product build] and copyright year".

### Function buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>Closes this dialog box.</td>
</tr>
</tbody>
</table>
CHAPTER 8  ERROR MESSAGES

If an error occurs during an operation in a window/dialog box, the following error dialog box that displays the error number and error message is displayed.

Figure 8-1  [Error] Dialog Box

The error number, error message, description of the error, and action to be taken are explained below.

1000: Not enough memory.
The memory in the machine is insufficient.
After closing unneeded windows, perform the operation again.

1001: Internal error.
An error that could not be anticipated has occurred.
Perform the operation again from the beginning.

1010: The active window already exists.
Since an active window already exists, the status of the window cannot be changed to active.
Either change the status of the active window to hold or close the active window before making the desired window active.

1021: The file does not exist.
The specified file name does not exist.
Specify a correct file name.

1022: Fail to write the file.
Either there is not sufficient free disk space or the specified file is a read-only file, so that it cannot be written to.
Specify a file in a separate disk, or a file that is not read-only.

1023: The file format is illegal.
The file cannot be read because its format is incorrect.
Check if it is a file for AZ trace data or not.

1031: The RX850 (uITRON3.0) is downloaded.
The RX850 Pro (uITRON3.0) is not embedded into the load module.
Embed the RX850 Pro (uITRON3.0) into the load module.

1032: The RX850 Pro(uITRON3.0) is downloaded.
The RX850 (uITRON3.0) is not embedded into the load module.
Embed the RX850 (uITRON3.0) into the load module.
CHAPTER 8  ERROR MESSAGES

1033: The RX850V4 (uITRON4.0) is downloaded.
    The RX850 (uITRON3.0) or the RX850 Pro (uITRON3.0) is not embedded into the load module.
    Embed the RX850 (uITRON3.0) or the RX850 Pro (uITRON3.0) into the load module.

1100: The debugger does not support AZ interface.
    Connection with the debugger is not possible because the AZ interface is not mounted on the debugger side.
    Check whether the debugger supports TIP.

1110: Fail to switch AZ trace mode.
    An error occurred during the switching processing of the AZ trace mode. The address mask value may not be
    correct or communication between the debugger and the AZ850 may have failed.
    Check that the address mask value is set correctly in the [AZ Option] dialog box. Otherwise restart the debugger
    and the AZ850.

1118: The buffer region not specified.
    AZ Trace ON was set without a trace buffer area being specified.
    Specify the trace buffer area in the [AZ Option] dialog box, and then set AZ Trace ON.

1120: Fail to load the trace data.
    An error occurred during trace data uploading. The settings of the trace buffer area may not be correct.
    Check if correct addresses have been specified as the trace buffer area.

1121: The trace data does not exist.
    There is no trace data in the executed data.
    Execute the application program in the AZ Trace ON mode and upload the trace data again.

1122: The trace data is illegal.
    The time tag in the trace data may not be correct.
    Check if the user own coding block is correctly coded.

1128: Task level data not included.
    Task-level trace data was not included in the collected trace data.
    Use the AZ850 that supports the function-level AZ trace data.

1210: The trace buffer address is illegal.
    The address range specified for the trace buffer is incorrect.
    Specify a correct address range for the trace buffer area specification in the [AZ Option] dialog box.

1220: The address mask is illegal.
    An illegal value that cannot be handled as a mask value was input for the address mask specification in the [AZ
    Option] dialog box.
    Specify a hexadecimal number for the mask value in the [AZ Option] dialog box.

1400: The pattern does not exist.
    The specified pattern does not exist in the trace data between the up cursor and the down cursor.
    Expand the interval between the up cursor and the down cursor, and perform the operation again. If this error
    occurs even with the maximum interval, the specified pattern does not exist in the trace data.

1700: The specified file has an invalid RX type.
    An attempt was made to read an AZ file whose real-time OS differs form the real-time OS specified in the [AZ
    Option] dialog box.
    Check if the appropriate real-time OS (RX850 or rx850 Pro) is specified in the [AZ Option] dialog box.
INDEX

A
[About] dialog box ... 106
  Function buttons ... 106
About objects ... 65
Address ... 99
Address mask ... 52
An event ... 87, 92
Analyze Window ... 57
  About objects ... 65
  Caution ... 73
  Error ... 73
  Explanation of each area ... 58
  Menu bar ... 61
  Tool bar ... 64
Understanding the execution transition map ... 67
  Verification method in execution transition map ... 72
AZ Monitor ... 33
[AZ Option] dialog box ... 51
  Error ... 53
  Explanation of each area ... 52
  Function buttons ... 53
AZ trace mode status area ... 49
AZ trace OFF ... 47
AZ trace ON ... 47
AzInit() ... 36
  __AZMON_CountMode ... 34
  __AZMON_GetCounter ... 34
  __AZMON_InitTimer ... 34
  __AZMON_MaxCount ... 34
  __AZMON_TimePerCount ... 34

C
C compiler package ... 18
Connection status area ... 49
CPU usage ... 82
Cpu Window ... 81
  Caution ... 86
  CPU usage display method ... 85
  Error ... 86
  Explanation of each area ... 82
  Menu bar ... 83
  Tool bar ... 85

D
Debugger ... 18
Detailed mode ... 68
Down cursor ... 58
Down temporary cursor ... 58

E
Equal mode ... 71
Error messages ... 107
Event ... 99
Extension of display file ... 55

G
Grid mode ... 61

H
Hard trace form ... 15, 32
Hide objects ... 75
Show objects ... 75
Host machine ... 18

I
Idle ... 99
In-circuit emulator ... 18
init_handler() ... 36
Int ... 99
IntRet ... 99

J
Jump to beginning button ... 59
Jump to end button ... 59

L
Load module execution status area ... 49

M
Main Window ... 46
  Menu bar ... 47
  Status bar ... 49
  Tool bar ... 48
  Marks of the execution transition map ... 69, 70

User’s Manual  U17423EJ2V0UM  109
N
Number of histogram divisions ... 93

O
Object ... 58
[Object Select] dialog box ... 74
  Adding objects to be displayed ... 76
  Explanation of each area ... 75
  Function buttons ... 75
  Modifying the order in which objects are displayed ... 76
  Restricting the objects to be displayed ... 76
[Open]/[Save As] dialog box ... 54
  Display file extensions and file formats ... 55
  Display information reproduction ... 56
  Error ... 56
  Explanation of each area ... 55
  Function buttons ... 55

P
Parameter ... 99
Pattern condition ... 88
Pattern Mode ... 88
[Pattern Search] dialog box ... 77
  Explanation of each area ... 78
  Function buttons ... 80
  Search method ... 80
[Pattern Set] dialog box ... 87
  Error ... 91
  Explanation of each area ... 88
  Function buttons ... 90
  Pattern search/calculation method ... 91
Pattern Window ... 92
  Caution ... 97
  Error ... 97
  Explanation of each area ... 93
  Menu bar ... 94
  Pattern distribution display method ... 96
  Pattern distribution viewing method ... 97
  Search method ... 97
  Tool bar ... 96
Procedure for debugging ... 37

R
Resource ... 19
Return ... 99
ROM emulator ... 18
RX850 ... 18
RX850 Pro ... 18

S
Scale modify button ... 59
Search condition ... 78, 103
Search item ... 103
Search Mode ... 78
Simple mode ... 67
Simple search button ... 59
Simulator ... 18
Soft trace form ... 15, 24, 29
Sort mode ... 75, 76
SysCall ... 99
SysRet ... 99

T
Task ... 99
TaskStart ... 99
Time ... 99
Time accuracy ... 26
Trace buffer ... 25
Trace buffer area ... 52
Trace buffer type ... 52
Trace form ... 15, 24, 52
[Trace Search] dialog box ... 103
  Explanation of each area ... 103
  Function buttons ... 104
  Search method ... 104
Trace View Window ... 98
  AZ trace data display method ... 101
  AZ trace data viewing method ... 101
  Caution ... 102
  Error ... 102
  Explanation of each area ... 99
  Menu bar ... 100
  Search method ... 102

U
Up cursor ... 58
Up temporary cursor ... 58
Upload ... 47

V
varfunc( ) ... 36
Version information ... 106
View mode ... 67
For further information, please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa 211-8668, Japan
Tel: 044-435-5111
http://www.necel.com/

[America]
NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]
NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbelski-Strasse 166 B
30177 Hanover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Sucursale Française
9, rue Paul Dautier, B.P. 52180
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Limburglaan 5
5616 HR Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]
NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
TEL: 010-8235-1155
http://www.cn.necel.com/

NEC Electronics Shanghai Ltd.
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai P.R. China P.C:200120
Tel: 021-5888-5400
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
12/F., Cityplaza 4,
12 Taikoo Wan Road, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

Seoul Branch
11F., Samik Lavied’or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-2719-2377

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/