



# EZ-CUBE

具有编程功能的片上调试仿真器

本资料所记载的内容，均为本资料发行时的信息，瑞萨电子对于本资料所记载的产品或者规格可能会作改动，恕不另行通知。  
请通过瑞萨电子(中国)的主页确认发布的最新信息。

#### 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; anticrime 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 majorityowned subsidiaries.

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

---

## 前言

衷心感谢您使用瑞萨电子（中国）有限公司的EZ-CUBE 仿真器。

EZ-CUBE 仿真器（YRCNEZCUBE01）是用于瑞萨电子产MCU 的仿真器。

这些手册的最新版能从本公司开发环境的主页（<http://www.cn.renesas.com/tools>）下载。

## 重要事项

在使用本仿真器前，请务必仔细阅读并理解用户手册。

务必保管好用户手册，在使用中如有不明之处，请再次阅读。

### 仿真器：

本资料中的仿真器是指瑞萨电子（中国）有限公司的EZ-CUBE 仿真器，不包含客户的用户系统和主机。

### 仿真器的使用目的：

本仿真器是支援使用瑞萨单片机进行系统开发的装置，从软件和硬件两方面支援系统开发，但是不保证生产线上的使用。

请遵循此使用目的正确使用本仿真器，坚决拒绝本目的以外的使用。

### 仿真器的使用对象：

本仿真器仅供仔细阅读并理解用户手册的对象使用。在使用本仿真器时，需要电子电路、逻辑电路和单片机的基本知识。

### 使用仿真器时：

1. 本仿真器是用于程序开发和评价阶段的开发支援装置。在批量生产已开发的程序时，请客户务必事先进行安装评价和试验等判断是否适用。

2. 对于因使用本仿真器而产生的用户开发结果，本公司不承担任何责任。

3. 本公司努力提供有关本仿真器的缺陷对策。但是，不保证任何情况都能提供。

4. 本仿真器是为实验室用于程序开发和评价而准备的产品。

5. 本公司无法预测所有存在的潜在危险而可能引发的诸多状况和错误使用。因此，此用户手册中和贴在本仿真器上的并非所有的警告。请客户正确安全使用本仿真器。

6. 对于因本产品的偶发故障或者误动作而引起的客户直接和间接的损害，本公司不承担责任。

### 关于废弃：

在废弃本仿真器时，请当作工业废弃物，根据法令规定进行处理。

### 使用限制：

本仿真器是作为开发支持工具而开发的产品。因此，请不要用作嵌入式设备，也不要用于以下所示的开发用途：

1. 运输、交通车辆
2. 医疗（用于涉及生命安全的装置）
3. 航空宇宙
4. 原子能控制
5. 海底中继器
6. 军事产品、大规模杀伤性武器

### 关于产品的更改：

本公司采取不断改良本仿真器的设计和性能的方针。因此，在更改规格、设计和用户手册时，恕不另行通知。

### 关于权利:

1. 对于因使用本资料记载的信息、产品或者电路而引起的损害或者专利权等其他权利的侵犯，本公司不承担任何责任。
2. 本资料不承诺第三者或者本公司的专利权以及其他权利的实施权。
3. 本用户手册和本仿真器享有著作权保护，所有权利归属本公司。事先未经本公司的书面许可，不得翻印、复制和转载本用户手册的部分或者全部内容。

### 关于图:

本用户手册的部分图可能和实物有差异。

### 术语

本手册使用的术语的含义请参照下表。

| 项目                      | 说明  |
|-------------------------|---|
| EZ-CUBE                 | EZ-CUBE 的通用名称                                   |
| 目标设备                    | 用于仿真的设备   |
| 目标系统                    | 用于调试的系统(用户创建的系统)<br>包括用户建立的软件和硬件。               |
| 固件                      | 嵌入设备用来控制 EZ-CUBE 的程序                            |
| CubeSuite+<br>for China | 集成开发环境 CubeSuite+ 中国版本                          |
| RFP                     | Renesas Flash Programmer, 用于进行 Flash 编程的 GUI 软件 |
| WriteEZ5                | 用于进行 Flash 编程的 GUI 软件                           |

## 注意事项

### 关于异常工作:

如果因外来噪声等干扰而引起仿真器工作异常，请按以下步骤处理：

1. 结束仿真调试程序，切断仿真器和用户系统的电源。
2. 在经过10 秒以后再次接通电源，启动仿真调试程序。

### 操作本产品的注意事项

瑞萨电子（中国）有限公司对于用户未遵照用户手册进行的非正常及非惯例使用造成的产品失效和其他相关损失不承担任何责任。

# 目 录

|                                     |    |
|-------------------------------------|----|
| 第 1 章 概述 .....                      | 8  |
| 1.1 特点 .....                        | 8  |
| 1.2 使用EZ-CUBE前的注意事项 .....           | 8  |
| 1.3 硬件规格 .....                      | 9  |
| 1.4 固件更新 .....                      | 9  |
| 1.5 标准配置 .....                      | 10 |
| 第 2 章 硬件名称和功能 .....                 | 11 |
| 2.1 EZ-CUBE各部件名称和功能 .....           | 11 |
| 第 3 章 如何在 78K0 微控制器上使用EZ-CUBE ..... | 13 |
| 3.1 目标系统设计 .....                    | 14 |
| 3.1.1 引脚配置 .....                    | 14 |
| 3.1.2 电路连接示例 .....                  | 15 |
| 3.1.3 reset 引脚的连接 .....             | 16 |
| 3.1.4 目标系统设计的注意事项 .....             | 17 |
| 3.1.5 在目标系统上安装连接器 .....             | 17 |
| 3.2 片上调试 .....                      | 18 |
| 3.2.1 调试功能 .....                    | 18 |
| 3.2.2 系统配置 .....                    | 18 |
| 3.2.3 系统启动顺序 .....                  | 19 |
| 3.2.4 系统关闭顺序 .....                  | 20 |
| 3.2.5 时钟设置 .....                    | 20 |
| 3.2.6 用户资源的保留和安全ID的设置 .....         | 21 |
| 3.2.7 调试的注意事项 .....                 | 25 |
| 3.3 Flash编程 .....                   | 26 |
| 3.3.1 编程功能的规格定义 .....               | 26 |
| 3.3.2 系统配置 .....                    | 26 |
| 3.3.3 系统启动顺序 .....                  | 27 |
| 3.3.4 应用样例 .....                    | 28 |
| 3.3.5 系统关闭顺序 .....                  | 32 |
| 3.3.6 Flash编程的注意事项 .....            | 33 |
| 第 4 章 如何在 78K0R微控制器上使用EZ-CUBE ..... | 34 |
| 4.1 目标系统设计 .....                    | 35 |
| 4.1.1 引脚配置 .....                    | 36 |
| 4.1.2 电路连接示例 .....                  | 37 |
| 4.1.3 reset 引脚的连接 .....             | 37 |
| 4.1.4 在目标系统上安装连接器 .....             | 38 |
| 4.2 片上调试 .....                      | 39 |
| 4.2.1 调试功能 .....                    | 39 |
| 4.2.2 系统配置 .....                    | 39 |
| 4.2.3 系统启动顺序 .....                  | 40 |
| 4.2.4 系统关闭顺序 .....                  | 41 |

|       |                                |    |
|-------|--------------------------------|----|
| 4.2.5 | 用户资源的保留，设置安全ID 以及片上调试选项字节..... | 41 |
| 4.2.6 | 调试的注意事项.....                   | 45 |
| 4.3   | Flash 编程.....                  | 46 |
| 4.3.1 | 编程功能的规格定义.....                 | 46 |
| 4.3.2 | 系统配置.....                      | 46 |
| 4.3.3 | 系统启动顺序.....                    | 47 |
| 4.3.4 | 应用样例.....                      | 48 |
| 4.3.5 | 系统关闭顺序.....                    | 52 |
| 4.3.6 | Flash 编程的注意事项.....             | 52 |
| 第 5 章 | 如何在R8C微控制器上使用EZ-CUBE.....      | 53 |
| 5.1   | 目标系统设计.....                    | 54 |
| 5.1.1 | 引脚配置.....                      | 54 |
| 5.1.2 | 电路连接示例.....                    | 55 |
| 5.1.3 | 在目标系统上安装连接器.....               | 56 |
| 5.2   | 片上调试.....                      | 57 |
| 5.2.1 | 调试功能.....                      | 57 |
| 5.2.2 | 系统配置.....                      | 57 |
| 5.2.3 | 系统启动顺序.....                    | 58 |
| 5.2.4 | 系统关闭顺序.....                    | 59 |
| 5.2.5 | 调试的注意事项.....                   | 59 |
| 5.3   | Flash 编程.....                  | 69 |
| 5.3.1 | 编程功能的规格定义.....                 | 69 |
| 5.3.2 | 系统配置.....                      | 69 |
| 5.3.3 | 系统启动顺序.....                    | 70 |
| 5.3.4 | 应用样例.....                      | 71 |
| 5.3.5 | 系统关闭顺序.....                    | 76 |
| 5.3.6 | Flash 编程的注意事项.....             | 76 |
| 第 6 章 | 如何在RL78 微控制器上使用EZ-CUBE.....    | 77 |
| 6.1   | 目标系统设计.....                    | 78 |
| 6.1.1 | 引脚配置.....                      | 78 |
| 6.1.2 | 电路连接举例.....                    | 79 |
| 6.1.3 | reset引脚的连接.....                | 80 |
| 6.1.4 | 在目标系统上安装连接器.....               | 81 |
| 6.2   | 片上调试.....                      | 82 |
| 6.2.1 | 调试功能.....                      | 82 |
| 6.2.2 | 系统配置.....                      | 82 |
| 6.2.3 | 系统启动顺序.....                    | 83 |
| 6.2.4 | 系统关闭顺序.....                    | 84 |
| 6.2.5 | 保留用户资源和设置安全ID.....             | 84 |
| 6.2.6 | 调试时的注意事项.....                  | 88 |
| 6.3   | Flash 编程.....                  | 89 |
| 6.3.1 | 编程功能的规格定义.....                 | 89 |
| 6.3.2 | 系统配置.....                      | 89 |
| 6.3.3 | 系统启动顺序.....                    | 90 |
| 6.3.4 | 应用样例.....                      | 91 |

---

|                                       |            |
|---------------------------------------|------------|
| 6.3.5 系统关闭顺序 .....                    | 95         |
| 6.3.5 系统关闭顺序 .....                    | 96         |
| 6.3.6 Flash编程时的注意事项.....              | 96         |
| <b>第7章 如何在V850微控制器上使用EZ-CUBE.....</b> | <b>97</b>  |
| <b>7.1 目标系统设计.....</b>                | <b>98</b>  |
| 7.1.1 引脚配置.....                       | 98         |
| 7.1.2 电路连接举例 .....                    | 99         |
| 7.1.3 reset引脚的连接 .....                | 100        |
| 7.1.4 在目标系统上安装连接器 .....               | 100        |
| <b>7.2 片上调试.....</b>                  | <b>101</b> |
| 7.2.1 调试功能.....                       | 101        |
| 7.2.2 系统配置.....                       | 101        |
| 7.2.3 系统启动顺序 .....                    | 102        |
| 7.2.4 系统关闭顺序 .....                    | 103        |
| 7.2.5 保留用户资源和设置安全ID .....             | 103        |
| 7.2.6 调试时的注意事项 .....                  | 107        |
| <b>7.3 Flash 编程.....</b>              | <b>109</b> |
| 7.3.1 编程功能的规格定义 .....                 | 109        |
| 7.3.2 系统配置.....                       | 109        |
| 7.3.3 系统启动顺序 .....                    | 110        |
| 7.3.4 应用样例.....                       | 111        |
| 7.3.5 系统关闭顺序 .....                    | 116        |
| 7.3.6 Flash编程时的注意事项.....              | 116        |

# 第 1 章 概述

EZ-CUBE 是具有 flash 存储器编程功能的片上调试仿真器，可以用于调试程序或将程序烧写到内置 Flash 存储器微控制器中。

## 1.1 特点

- 片上调试  
可以调试已连接到目标板上的微控制器。
- Flash编程  
可以将程序写入控制器的内置flash存储器中。
- USB 连接  
可以通过USB 2.0接口和主机连接。  
工作时使用USB供电，无需外部电源。
- 支持产品范围广  
EZ-CUBE能支持Renesas Electronics 8位到32位的内置Flash存储器微控制器。
  - 78K0微控制器
  - 78K0R微控制器
  - RL78 微控制器
  - R8C 微控制器
  - V850 微控制器

## 1.2 使用 EZ-CUBE 前的注意事项

第 1 章和第 2 章是有关 EZ-CUBE 的概要和基本规格说明，以下的章节针对不同的目标设备及其用途分别进行描述。

为了有效地使用本手册，您可以参照下表。

表 1-1. 用途对应章节

| 目标设备  | 用途       | 参照章节                              |
|-------|----------|-----------------------------------|
| 78K0  |          | 第 3 章 如何在 78K0 系列微控制器上使用 EZ-CUBE  |
|       | 目标系统设计   | 3.1 目标系统设计                        |
|       | 片上调试     | 3.2 片上调试                          |
| 78K0R | Flash 编程 | 3.3 Flash 编程                      |
|       |          | 第 4 章 如何在 78K0R 系列微控制器上使用 EZ-CUBE |
|       | 目标系统设计   | 4.1 目标系统设计                        |
| R8C   | 片上调试     | 4.2 片上调试                          |
|       | Flash 编程 | 4.3 Flash 编程                      |
|       |          | 第 5 章 如何在 R8C 系列微控制器上使用 EZ-CUBE   |
|       | 目标系统设计   | 5.1 目标系统设计                        |
|       | 片上调试     | 5.2 片上调试                          |
|       | Flash 编程 | 5.3 Flash 编程                      |



|      |          |                                |
|------|----------|--------------------------------|
| RL78 |          | 第6章 如何在 RL78 系列微控制器上使用 EZ-CUBE |
|      | 目标系统设计   | 6.1 目标系统设计                     |
|      | 片上调试     | 6.2 片上调试                       |
|      | Flash 编程 | 6.3 Flash 编程                   |
| V850 |          | 第7章 如何在 V850 系列微控制器上使用 EZ-CUBE |
|      | 目标系统设计   | 7.1 目标系统设计                     |
|      | 片上调试     | 7.2 片上调试                       |
|      | Flash 编程 | 7.3 Flash 编程                   |

### 1.3 硬件规格

本节将说明 EZ-CUBE 的硬件规格。  
相应片上调试和 Flash 存储器编程功能的相关规格在随后的章节中介绍。

表 1-2. 硬件配置

| 分类      | 项目      | 规格  |
|---------|---------|---|
| EZ-CUBE | 工作电压    | USB 接口供电 (5 V)  |
|         | 工作环境    | 温度: $\pm 0 \sim +40^{\circ}\text{C}$<br>湿度: 40 - 80% RH (无结露) |
|         | 保存      | 温度: $-15 \sim +60^{\circ}\text{C}$<br>湿度: 40 - 80% RH (无结露)   |
|         | 外形尺寸    | 59 × 35 × 13 mm   |
|         | 重量      | 约 40 g  |
| 主机界面    | 目标主机    | 兼容 IBM PC/AT/ATM  |
|         | 目标操作系统  | Windows XP, Vista, Windows 7 <sup>注</sup>                     |
|         | USB     | 2.0   |
|         | USB 连接线 | 1.0 m   |
|         | 消耗电流    | 最大 500 mA   |
| 目标界面    | 目标线缆    | 8 引脚线缆  |
|         | 供电电压    | 5 V $\pm$ 0.3V (典型值)  |
|         | 供电电流    | 最大 100 mA   |
|         | 电压范围    | 2.7 - 5.5 V   |

注: WriteEZ5 不支持 Windows 7。

### 1.4 固件更新

- (1) **驱动程序安装**  
如果没有安装用于 EZ-CUBE 的 USB 驱动程序, 请先安装它。(CD 光盘)
- (2) **USB 连接线**  
将 EZ-CUBE 连接到主机, **不要将 EZ-CUBE 与目标板相连**。连接成功后状态指示灯显示为红色。

(3) **EZ-CUBE 固件更新程序的启动**

启动 EZ-CUBE 固件更新程序【QBEZUTL.exe】，如下所示。

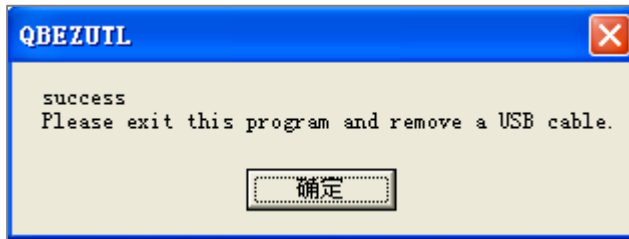


(4) **选择固件**

点击菜单栏中的[...]按钮，选择固件 (\*.hex)。

(5) **更新固件**

点击[Start]按钮，开始更新固件。固件更新完成，显示以下对话框。



(6) **EZ-CUBE 固件更新程序的关闭**

固件更新完成，点击[EXIT]按钮，退出固件更新程序。

(7) **拔除 USB 线缆**

从 EZ-CUBE 或主机上拔除 USB 线缆。

## 1.5 标准配置

EZ-CUBE 本体一个、USB 电缆一根、8pin 目标线缆一根和软件光盘一张(包含: 安装软件、用户手册电子版)。

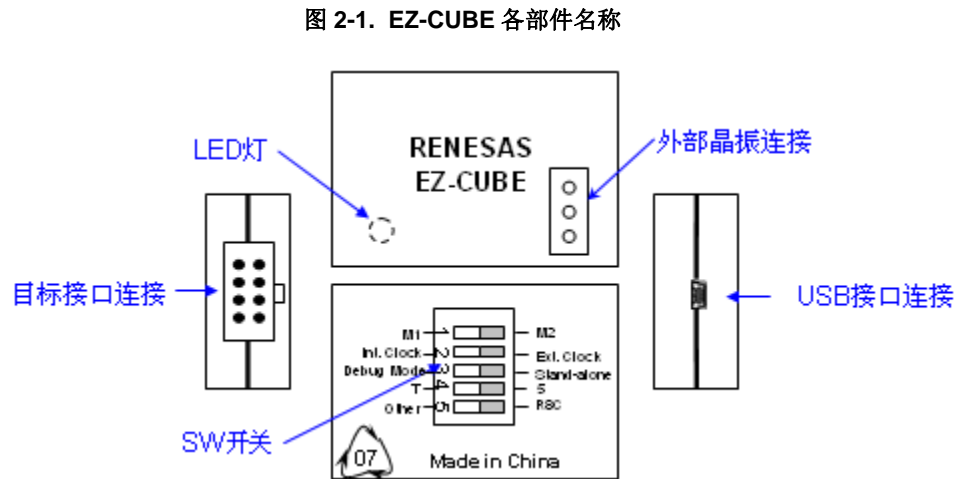
## 第 2 章 硬件名称和功能

本章介绍 EZ-CUBE 的部件名称和功能及其附件。

本章所使用的部件名称将在后续的章节继续使用。本章介绍各种功能的概述。阅读本章后，读者将对 EZ-CUBE 有一个基本的认识，有利于理解接下来的章节。阅读本章内容同时，可以用于检查硬件设备是否存在缺损。

### 2.1 EZ-CUBE 各部件名称和功能

图 2-1 标明了 EZ-CUBE 各部件名称。具体功能请参照 (1) ~ (9)。



(1) **SW-1 开关**

此开关取决于目标设备，详情请参见各系列章节。出厂时默认设定为“M2”。

(2) **SW-2 开关**

此开关用来设定目标系统振荡器，表 2-1 说明了具体含义，出厂时默认设定为“Int. Clock”。

**表 2-1. 振荡器选择开关设定**

| 设定         | 说明  |
|------------|---|
| Int. Clock | 当选择“Int. Clock”，时钟以 8MHz 固定频率提供给目标板。（出厂设定）  |
| Ext. Clock | 如需要改变频率，必须选择“Ext. Clock”（在这种情况下，需使用外部晶振连接）。 |

(3) **SW-3 开关**

此开关选择目标系统的运行方式，表 2-2 说明了具体含义，出厂时默认设定为“Debug Mode”。

**表 2-2. 运行模式选择开关设定**

| 设定          | 说明  |
|-------------|---|
| Debug Mode  | 当选择“Debug Mode”，运行命令从 GUI 调试器发出时，用户程序将运行。（出厂设定） |
| Stand Alone | 当选择“Stand Alone”，EZ-CUBE 连接到目标板，产生复位，用户程序将自动运行。 |

(4) **SW-4 开关**

此开关用来设定目标系统电源模式，表 2-3 说明了具体含义，出厂时默认设定为“5”。

表 2-3. 电源选择开关设定

| 设定 | 说明  |
|----|---|
| 5  | EZ-CUBE 向目标系统提供+5±0.3 V电源 <sup>注</sup> 。（出产设定）  |
| T  | 使用目标系统的电源，电压范围+2.7V~+5.5V。<br>即使在这种情况下，EZ-CUBE的VDD引脚线也必须与目标板的VDD连接，以调整目标板和EZ-CUBE之间的电压电平。 |

**注** 最大允许电流 100mA，所以请勿将 EZ-CUBE 连接到电流很大的目标系统上。在 EZ-CUBE 与主机连通后，主机始终保持向 EZ-CUBE 供电。

(5) **SW-5 开关**

此开关取决于目标设备，详情请参见各系列章节。出厂时默认设定为“Other”。

(6) **USB 连接器**

通过 USB 连接线，和主机进行连接的连接器。  
使用 USB 2.0 对应的 mini-B 连接器。

(7) **目标连接器**

通过 2\*4pin 的 8 引脚线缆，和目标系统进行连接的连接器。

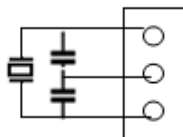
(8) **外部晶振连接**

**SW-2 开关选为“Ext. Clock ”。**

在 EZ-CUBE 上的外部晶振位置安装振荡器或者振荡电路。（在调试器的配置对话框中，选择外部晶振为主时钟）。详细信息，请参考 CubeSuite+的用户手册。

安装振荡器，参考图 2-2。

图 2-2 在 EZ-CUBE 的外接晶振上安装振荡电路



(9) **模式灯**

根据硬件和软件的变化，模式灯会有不同的显示状态，请参照表 2-4。

表 2-4. 模式灯说明

| 模式灯颜色 | 显示状态 | 说明       |              |
|-------|------|----------|--------------|
|       |      | USB 连接状态 | 软件工作状态       |
| -     | 熄灭   | 未连接      | 未启动          |
| Red   | 点亮   | 连接       | 上电或用户程序调试时中断 |
| Green | 点亮   |          | 用户程序调试正在运行   |

## 第 3 章 如何在 78K0 微控制器上使用 EZ-CUBE

本章描述了如何使用 EZ-CUBE 在 78K0 微控制器上进行片上调试和 Flash 编程。

片上调试是指通过芯片内置的调试功能对已经装载在目标系统上的微控制器进行调试的方法。由于这种调试方法是通过板上的目标设备实行的，所以非常适合现场调试。

Flash 编程是将程序写入设备内置的 Flash 存储器的方法。可以在板上对设备进行擦除，写入和校验。

请先将 EZ-CUBE 的固件更新为用于 78K0，具体请参照 (1) - (3)。详细内容请参见 **1.4 固件更新**。

- (1) 使用 USB 缆连接 EZ-CUBE 和 PC。**不要将 EZ-CUBE 与目标板相连。**
- (2) 启动【QBEZUTL.exe】，并指定“78K0\_OCD\_FW.hex”固件文件。
- (3) 点击[Start]按钮，然后指定用于 78K0 固件下载到 EZ-CUBE。

如果首次用 78K0 微控制器作为目标设备来使用 EZ-CUBE，那么请仔细阅读以下内容。

### 3.1 目标系统设计

要使用 EZ-CUBE 和目标系统进行通信，必须在目标系统上安装对应的通信电路。本节描述了相关电路的设计和连接插头的安装。

### 3.2 片上调试

本节描述了用 EZ-CUBE 进行片上调试的系统配置和启动方法。

### 3.3 Flash 编程

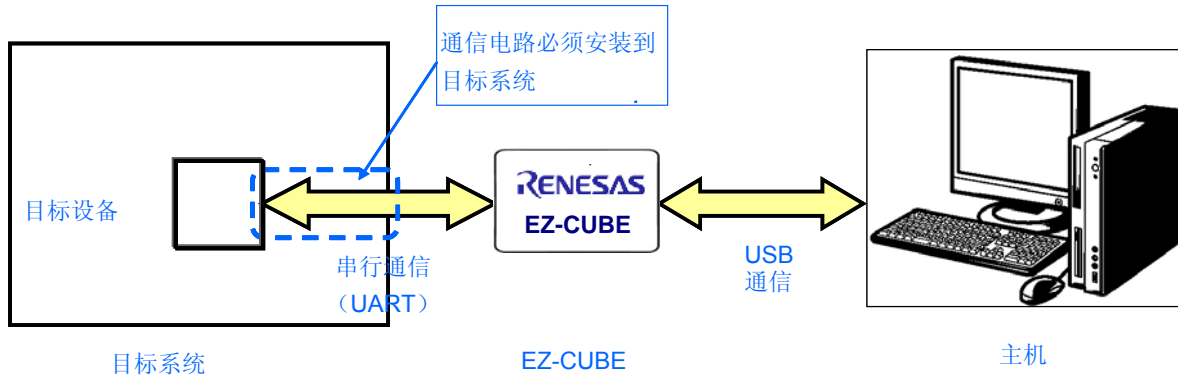
本节描述了用 EZ-CUBE 进行 Flash 编程的系统配置和启动方法。

### 3.1 目标系统设计

要使用 EZ-CUBE 和目标系统进行通信，必须在目标系统上安装对应的通信电路。本节描述了相关电路的设计和连接插头的安装。

图 3-1 显示了 EZ-CUBE 通信的整体接口框图。为了 EZ-CUBE 和目标系统之间的通信，目标系统上必须要安装通信电路，如图中左侧所示。请参考本节的内容进行合适的电路设计。

图 3-1 通信接口框图



#### 3.1.1 引脚配置

本节将说明 EZ-CUBE 和目标系统之间使用的接口信号。表 3-1 中列出了引脚的配置情况。表 3-2 说明了各个引脚的功能。

表 3-1. 引脚配置

| 引脚编号 | 引脚名称 <sup>注</sup> |
|------|-------------------|
| 1    | GND               |
| 2    | RESET_IN          |
| 3    | Vdd               |
| 4    | FLMD0             |
| 5    | CLK               |
| 6    | RxD.              |
| 7    | RESET_OUT         |
| 8    | TxD               |

注 EZ-CUBE 内的信号名称。

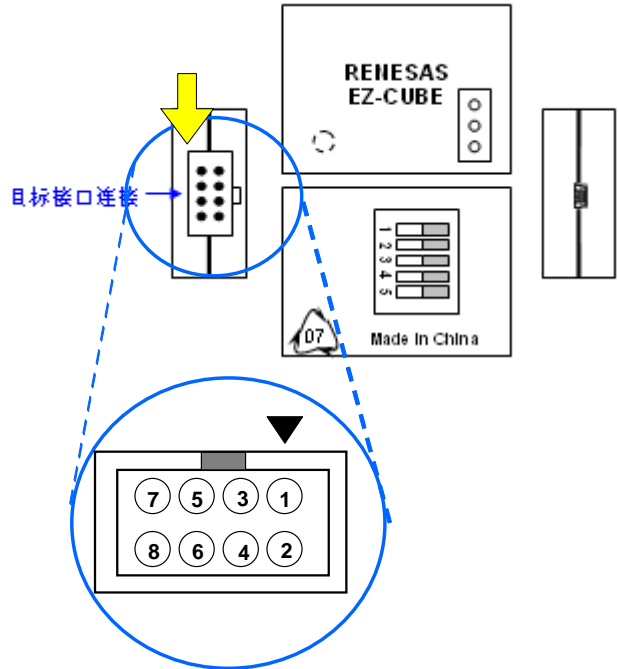


表 3-2. 引脚功能

| 引脚名称      | IN/OUT <sup>注</sup> | 描述                     |
|-----------|---------------------|------------------------|
| RESET_IN  | IN                  | 该引脚用于从目标系统输入 reset 信号  |
| RESET_OUT | OUT                 | 该引脚用于向目标系统输出 reset 信号  |
| CLK       | OUT                 | 该引脚用于向目标系统输出时钟信号       |
| FLMD0     | OUT                 | 该引脚用于设定目标设备进入调试模式或编程模式 |
| RxD       | IN                  | 该引脚用于接收来自目标设备的命令/数据    |
| TxD       | OUT                 | 该引脚用于向目标设备发送命令/数据      |

注 以 EZ-CUBE 为参照基准。

### 3.1.2 电路连接示例

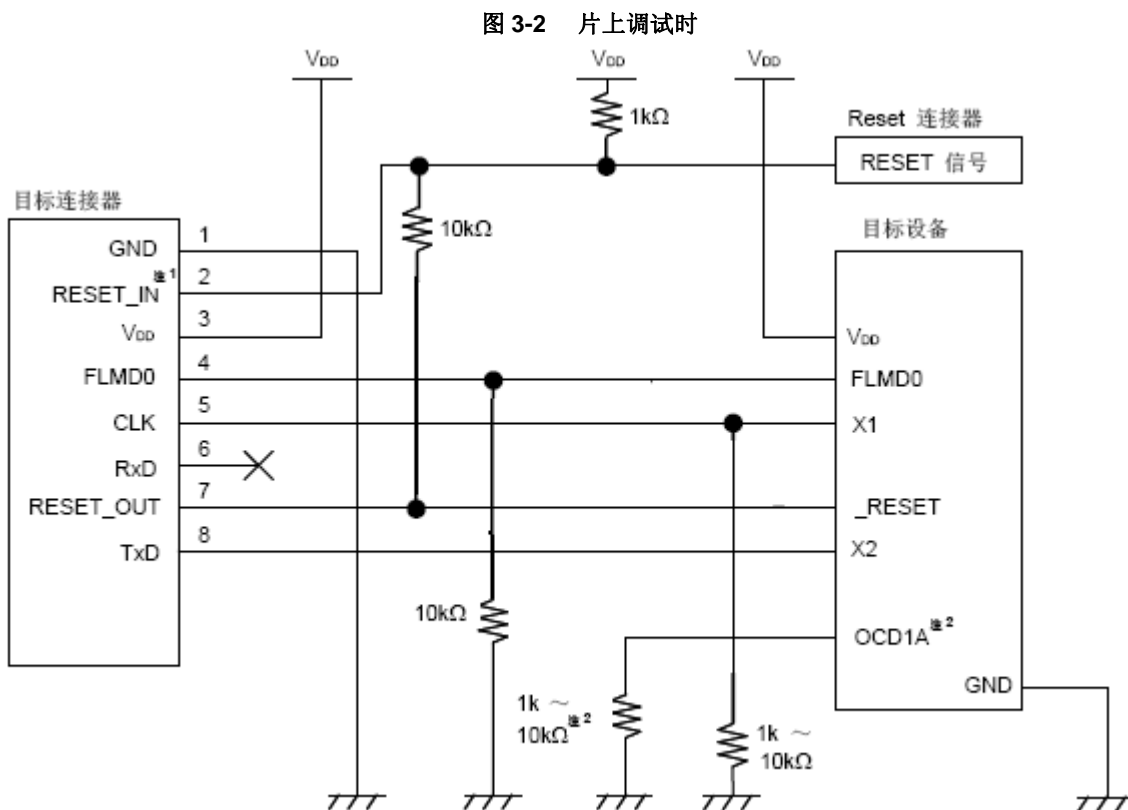
在调试和编程的时候，引脚配置有很大区别，所以请根据描述的电路连接示例来设计合适的电路。

目标系统上的电路设计会根据接口信号而有所不同。接口的引脚名称可能会因目标设备有所不同。详细信息请参考目标设备的用户手册。

下面的 (1) 至 (2) 项是针对不同使用情况的主要用途。请确认使用目的，参照表 3-1 和相关的电路连接示例。

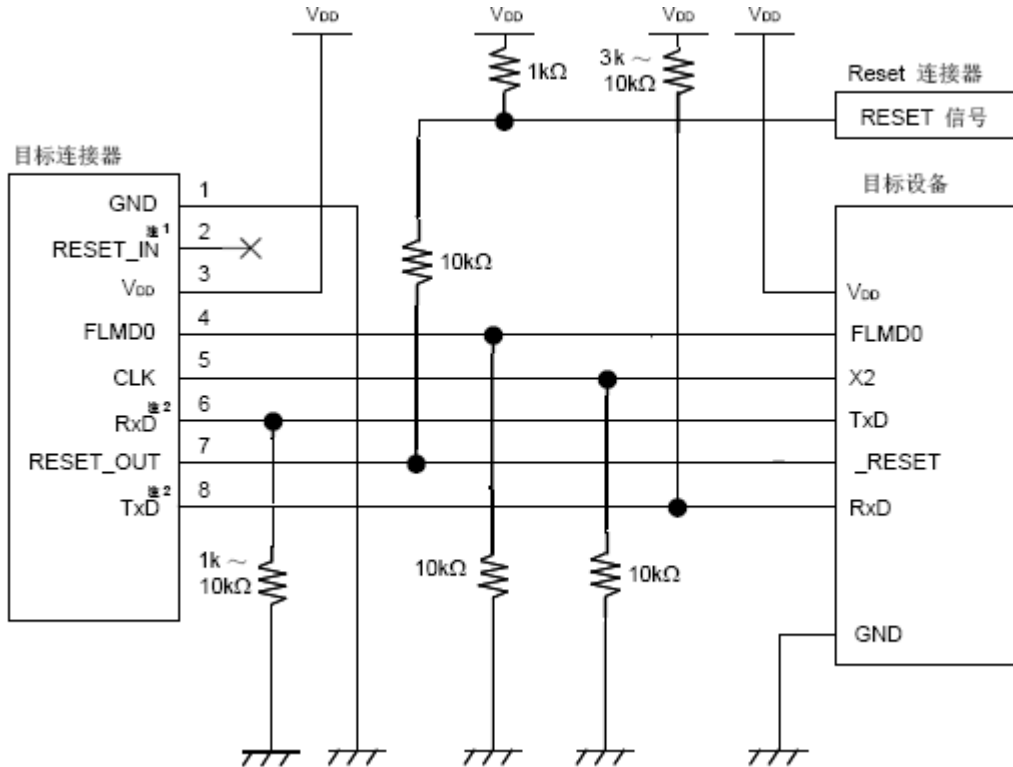
**注意事项** 在电路连接示例中出现的常数值为参考值。

- (1) 进行片上调试，参考图 3-2
- (2) 进行 Flash 编程，参考图 3-3



- 注:**
- 1. 这种连接的设计是认为 RESET 信号是从 N 沟道开漏缓冲器（输出阻抗：100 欧姆或更小）输出的。详细信息请参考 3.1.3 reset 引脚的连接
  - 2. 各种不同设备上的 OCD1A 引脚名称可能不同，详细信息请参考目标设备的用户手册。

图 3-3 flash 编程时



- 注： 1. 这种连接的设计是认为 RESET 信号是从 N 沟道开漏缓冲器（输出阻抗：100 欧姆或更小）输出的。详细信息请参考 3.1.3 reset 引脚的连接。
2. 将目标设备的 TxD（发送方）连接到目标连接器的 RxD（接收方），同时将目标连接器的 TxD（发送方）连接到目标设备的 RxD（接收方）。

### 3.1.3 reset 引脚的连接

本节将介绍上一节的电路连接示例中提到的需要特别注意的 reset 引脚的连接方法。

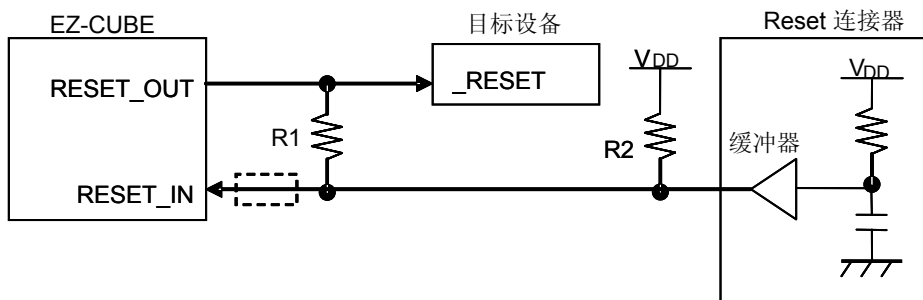
片上调试时，来自目标系统的 reset 信号输入 EZ-CUBE，被屏蔽后，再输出到目标设备。于是 reset 信号的连接因为 EZ-CUBE 是否连接而有所不同。

编程时，电路的设计必须保证目标系统的 reset 信号和 EZ-CUBE 的 reset 信号不冲突。

推荐通过电阻自动切换 reset 信号。图 3-4 说明了 3.1.2 电路连接示例中描述的 reset 引脚连接情况。

这种连接的设计是认为目标系统的 RESET 电路包含有 N 沟道开漏缓冲器（输出阻抗：100 欧姆或更小）。当 EZ-CUBE 的 RESET\_IN/OUT 逻辑反转时，VDD 或 GND 的电平可能不稳定，所以请注意备注中所描述的条件。

图 3-4 含有缓冲器的复位电路连接

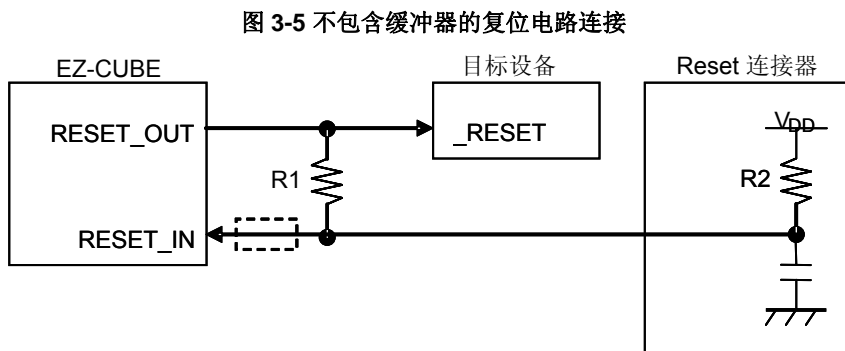




**备注：** 需确保R1的阻值至少是R2阻值的十倍，R1为10k或更大。

如果复位电路的缓冲器为CMOS输出，就不再需要R2上拉电阻。当仅进行Flash编程时，虚线框内的电路可以略去。

图 3-5 所示的电路连接示例描述了目标系统的复位电路中不包含缓冲器以及 reset 信号仅通过电阻或电容产生的情况。设计电路时需要注意备注中描述的条件。



**备注：** 需确保R1的阻值至少是R2阻值的十倍，R1为10 k或更大。  
只有在进行 Flash 编程时，虚线框内的电路才可以略去。

### 3.1.4 目标系统设计的注意事项

在设计目标系统时，请务必注意以下的事项。

- 目标系统电路布局时请不要将 X1/OCD1A 和 X2/OCD1B 放在相邻的位置。如果必须这样布局，他们的连线要尽量短。
- 目标连接器和目标设备之间的距离要尽量短。
- 由于 X1/OCD1A 和 X2/OCD1B 用于提供时钟，请注意屏蔽，比如使用 GND 通路。
- 在产品出货之前，请使用跳线或者其他方式将 X1/OCD1A 和 X2/OCD1B 的引脚同目标连接器分离开来，以确保正常的时钟振荡。
- 在调试时，请将电容、反馈电阻以及其他用于振荡的元器件移去，以防止 X1/OCD1A 和 X2/OCD1B 引脚上所连接的负载引发信号失真。

### 3.1.5 在目标系统上安装连接器

在设计目标系统时，必须安装 EZ-CUBE 和目标系统的连接器。可以选择 2.54mm 间距的 8 引脚通用型连接器。

除了专用的连接线之外，同时也支持散线连接。

### 3.2 片上调试

本节描述了使用 EZ-CUBE 进行片上调试的系统配置，启动/关闭顺序和调试中的若干注意事项。

#### 3.2.1 调试功能

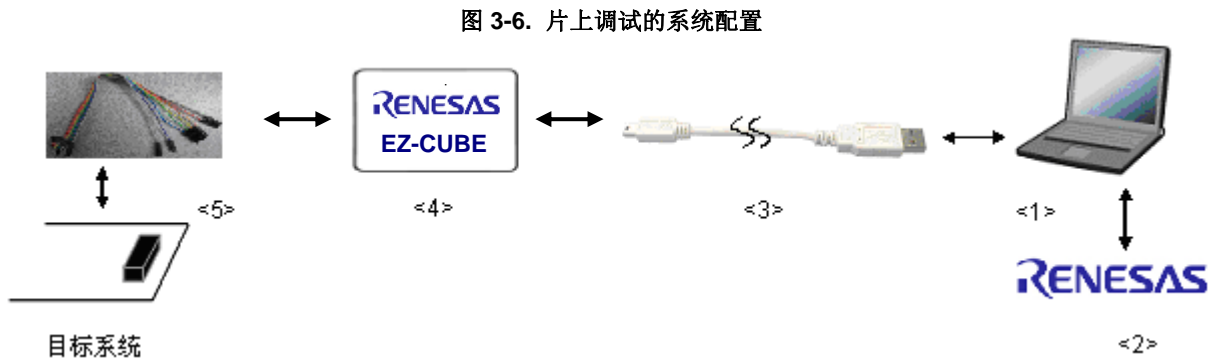
以 78K0 微控制器作为目标设备，进行调试时的调试功能。

表3-3 调试功能

| 功能        | 规格定义  |
|-----------|---|
| 目标MCU     | 78K0系列<br>78K0/ Kx2, Lx3                        |
| 安全        | 10字节ID码验证                                       |
| 下载        | 支持  |
| 执行        | 全速执行，忽略断点执行，单步执行，下一步执行，运行至此，重启                  |
| 硬件断点      | 1个  |
| 软件断点      | 多个  |
| 调试所用的用户空间 | 内部 ROM: 256~ 400 字节<br>内部RAM: 7~9 字节<br>根据设备而不同 |
| 用于调试的功能引脚 | X1, X2  |

#### 3.2.2 系统配置

图 3-6 所示为片上调试的系统配置。



- <1> 主机  
要求有 USB 接口。
- <2> 软件  
包含 CubeSuite+ for China，USB 驱动等。
- <3> USB 线缆（附件）
- <4> EZ-CUBE（本产品）
- <5> 8 引脚目标线缆（附件）

### 3.2.3 系统启动顺序

本节描述了系统启动的顺序。请注意按次序操作。

#### (1) 软件的准备和安装

以下软件是片上调试所必须的。关于软件的准备和安装，请参考 EZ-CUBE 附带的安装手册。

- CubeSuite+ for China
- USB 驱动

#### (2) 开关设置

SW-1 开关：请选择 "M2"

SW-2 开关：请根据实际目标设备使用选择

SW-3 开关：请选择 " Debug Mode"

SW-4 开关：请根据实际目标设备使用选择

SW-5 开关：请选择 "Other"

#### 注意事项

1. 在 USB 线缆处于连接状态时，请勿改变开关的设置。
2. 最大允许电流 100mA，所以请勿将 EZ-CUBE 连接到电流很大的目标系统上。在 EZ-CUBE 与主机连通后，主机始终保持向 EZ-CUBE 供电。

#### (3) 连接目标系统

将 EZ-CUBE 连接到目标系统，在目标系统打开前确认完成连接。

#### (4) 连接 USB 线缆

请在目标系统上电之前将 EZ-CUBE 连接到主机。

当 USB 缆与 PC 机间连接时，模式灯点亮为红色。

#### (5) 目标系统上电

打开目标系统的电源。如果电源选择为 "5"，则这一步并不是必需的。

#### (6) 启动调试器

打开调试器。

此步骤及之后的操作，请参考 CubeSuite+ 的用户手册。

如果调试器无法正常开启或者操作不稳定，可能是由以下问题引发的。

- EZ-CUBE 和目标系统之间通信错误
  - 固件是否更新；
  - 开关是否选择正确；
  - 目标连线是否正确。
- 未预留用户资源或者未设置安全 ID
  - 使用 EZ-CUBE 进行调试时，必须预留出一部分用户资源作为调试监控区域和串行通信接口区域，安全 ID 也必须设置。详细信息请参考 3.2.6 用户资源的保留和设置安全 ID。
- 使用了不支持的软件（调试器，设备文件或者固件）
  - 所使用的软件可能不支持目标设备的调试。
- EZ-CUBE 损坏
  - EZ-CUBE 可能被损坏。

### 3.2.4 系统关闭顺序

请按照下列操作顺序终止调试并关闭系统。

如果没有按照下列顺序操作，目标系统或者 EZ-CUBE 可能被损坏。

(1) 停止调试器

停止调试器的运行并断开调试器连接。

(2) 目标系统电源关闭

关闭目标系统的电源。如果电源选择开关放在“5”的位置，则无需本步操作。

(3) 拔除 USB 线缆

从 EZ-CUBE 或主机上将 USB 线拔下。

(4) 拔除目标线缆

从 EZ-CUBE 或目标系统上将目标线缆拔下。

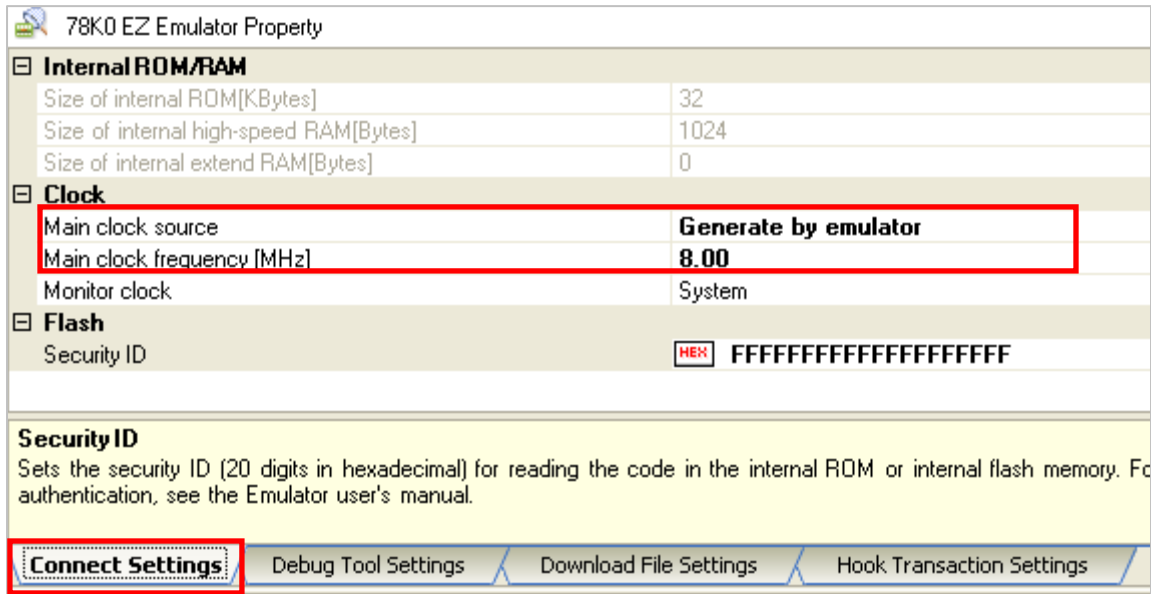
### 3.2.5 时钟设置

EZ-CUBE 可以提供时钟信号用作高速系统时钟（4、8 或 16MHz）。

**SW-2 开关选为“Int. Clock”。**

移除外部晶振的振荡器或振荡电路。（在 CubeSuite+ 的“Debug Tool”的[Connect Settings]选项卡中，时钟选择“Generate by emulator”为主时钟）。

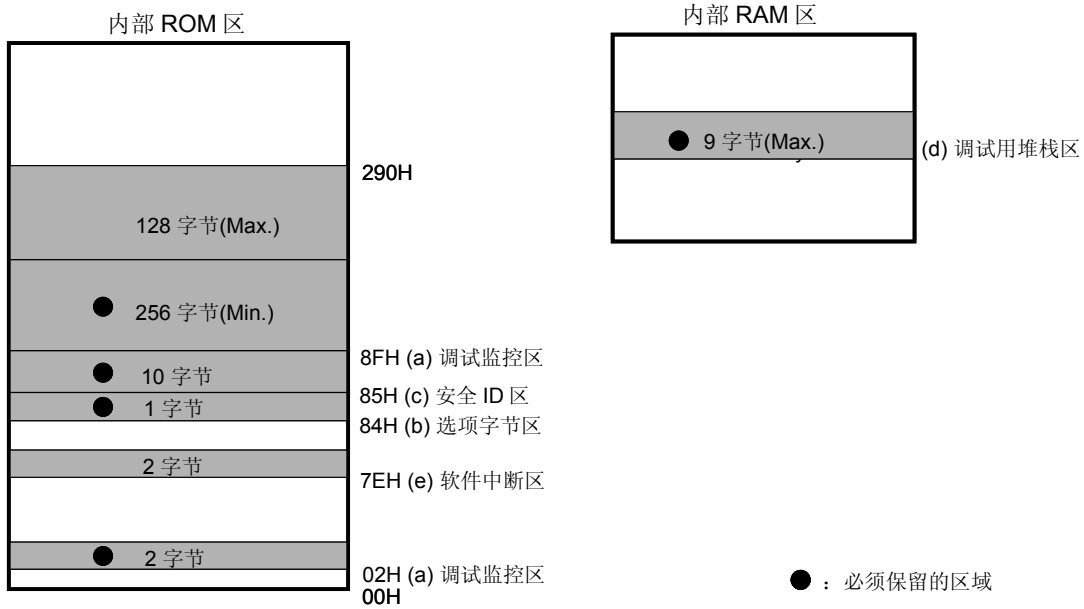
这些设置的具体含义和详细内容请参考 CubeSuite+ 的用户手册。



### 3.2.6 用户资源的保留和安全 ID 的设置

EZ-CUBE 使用用户存储区域（图 3-7 中阴影表示的区域）来完成自身和目标设备的通信，或者完成各种调试功能。用点（●）表示的区域在调试过程中会一直被占用，其他区域的占用对应实现各个调试功能。请参考后面描述的（a）至（e）项，并在用户程序中预留这些区域，或使用编译参数加以指定。

图3-7 使用EZ-CUBE时的保留区域



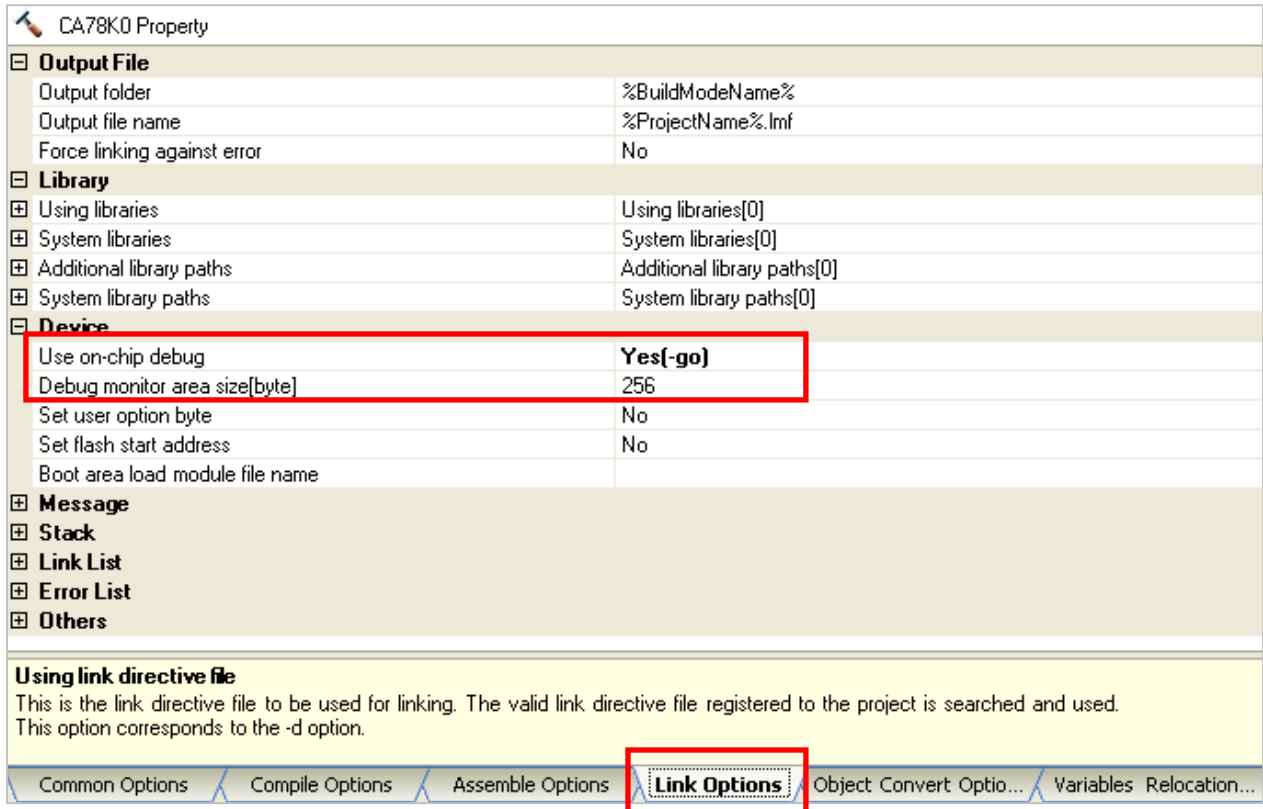
**(a) 调试监控区域（必须保留此区域）**

地址 0x02、0x03 以及 0x8F 开始的地址必须预留，以存放调试监控程序。  
如果这块区域被 Flash 自编程重写，则片上调试无法再进行。

**[如何预留区域]**

当使用 Renesas Electronics 的编译器时，选用连接参数（-go）可以预留地址 0x02、0x03 以及从 0x8F 开始的地址，用于调试监控，如图 3-8 显示了在 CubeSuite+中的“Build Tool”的[Link Options]选项卡。选中下图中的“Use on-chip debug”选择框，保留区域大小设置为 256 字节（实际上，从 0x8F 到 0x18F 的 257 字节都预留出来了）。

图3-8 连接参数设置（调试监控区域）



(b) 选项字节区域（必须）

本区域用于安全设置，以防止对 Flash 存储器未经授权的读取。  
根据设置值的不同，目标设备的操作有所不同，具体如下表所示。

表3-4 选项字节设置和操作

| 可选值  | 描述                               | 备注              |
|------|----------------------------------|-----------------|
| 0x00 | 即使连接了片上调试的仿真器（比如EZ-CUBE），仍然无法调试  | 这个设置仅对Flash编程有效 |
| 0x02 | 不论安全ID鉴别发生多少次错误，片上Flash存储器都不会被擦除 | -               |
| 0x03 | 如果安全ID鉴别错误，片上Flash存储器的所有内容都会被擦除  | -               |
| 其它   | 禁止设置                             | -               |

[如何设置]

描述如何在用户程序实现选项字节的设置。请参考下面的汇编代码。

例 设置为 0x02

```
SSS CSEG AT 084H; “SSS”是任意符号名（长度不超过 8 个字符）
DB 2H
```

**注意事项** 如果地址 0x84 的内容在自编程中被重写为 0x00，EZ-CUBE 和主机之间的通信就被禁止，此时即使重启调试器，调试和连接也无法进行。此时，只能通过 Flash 编程的方法擦除存储器内容。

(c) 安全 ID 区域（必须）

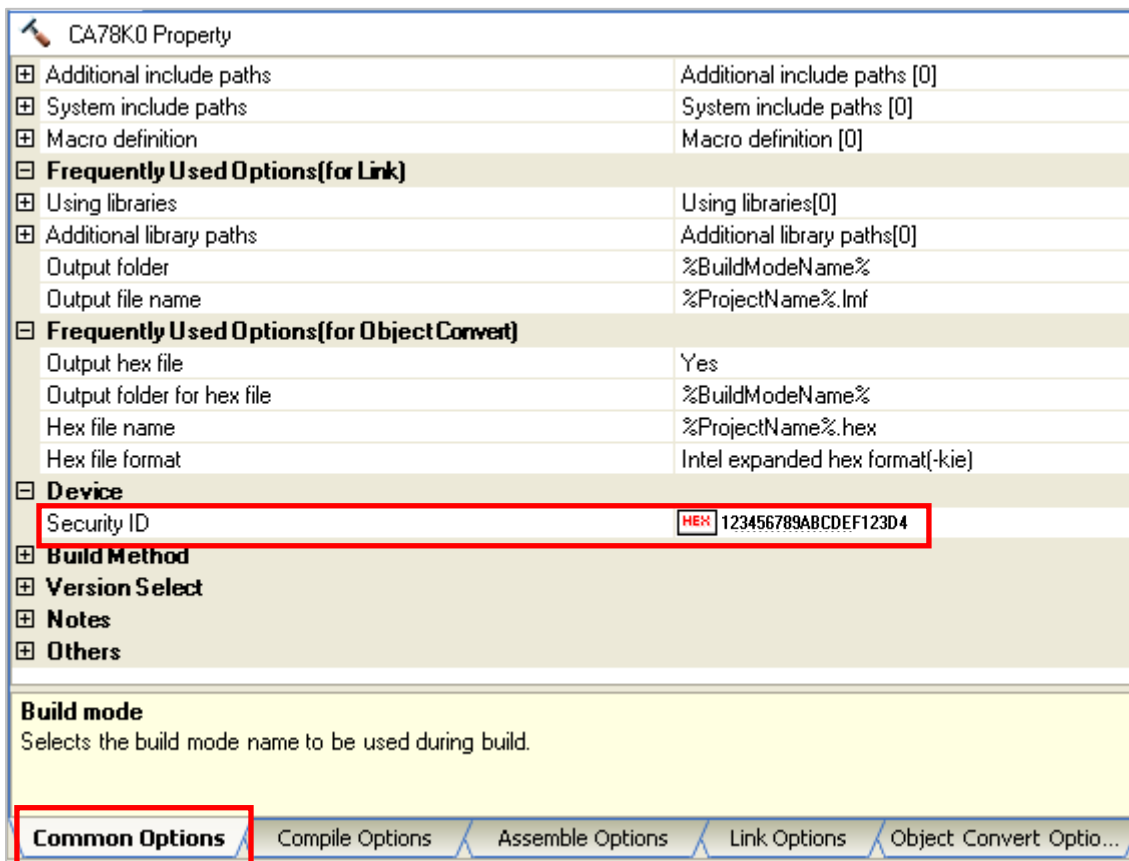
本区域用于安全设置，以防止对 Flash 存储器未经授权的读取。

安全 ID 功能作为启动调试器的密码。调试器启动之初，只有当输入的安全 ID 和本区域中写入的安全 ID 相符合时，调试器才可以启动。

[如何设置]

如图 3-9 显示了 CubeSuite+ 中的“Build Tool”的[Common Option]选项卡。请选中图中的“Security ID”选择框，可以设置为任意 ID 值（10 字节长度）。

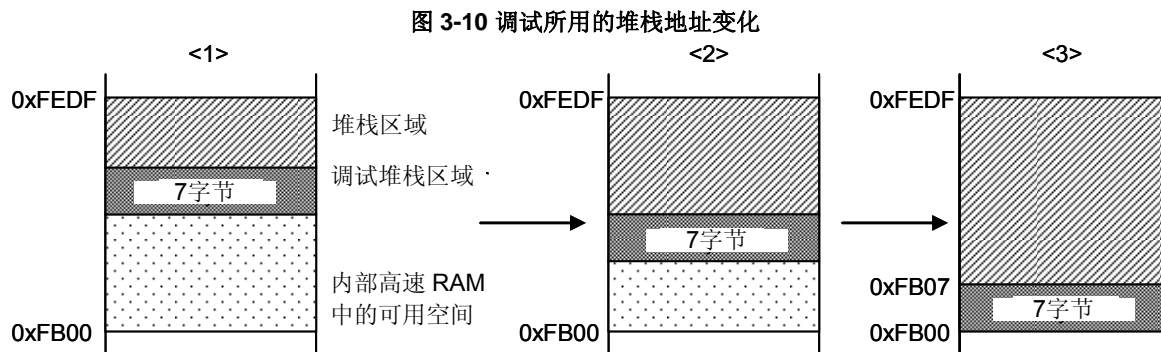
图3-9 连接参数设置（安全ID）



(d) 调试所需的堆栈区域（必须保留此区域）

需要 7~9 个字节作为调试的堆栈区域。由于本区域被分配在紧邻堆栈区域之前，所以本区域的实际地址会根据堆栈的增加和减少而改变。

图 3-10 说明了堆栈区域增加时的情况，图中的内部高速 RAM 从 0xFB00 开始。



根据是否使用软件断点功能而有所不同，本区域所占用的大小也会改变。

表3-5 调试所用的堆栈区域大小

| 选项     | 调试所需的堆栈区域大小 |
|--------|-------------|
| 标准     | 7字节         |
| 使用软件中断 | 9字节         |

[如何预留这段区域]

请参考下面的地址范围来设置堆栈指针。

例 当内部高速 RAM 从 0xFB00 开始时

- 当使用硬件断点时  
在 0xFB07 到 0xFEDF 范围内<sup>注</sup>
- 当使用软件断点时（参见（e）项）  
在 0xFB09 到 0xFEDF 范围内<sup>注</sup>

注 因为 EZ-CUBE 在中断时占用了 0xFEDF，请将堆栈指针初始值指向 0xFEDF，而不要指向 0xFEE0（当初始值设置为 0xFEDF，堆栈实际使用的值为 0xFEDE，所以就不会产生冲突）。另外，0xFEDF 被禁止用作堆栈区域，但是可以被用作变量区域。

(e) 软件断点区域

本区域用于软件断点。

[如何预留这段区域]

参考如下代码来预留区域。

```
SSS    CSEG AT      07EH; "SSS"是任意符号名（长度不超过 8 个字符）
        DB    0FFH, 0FFH
```



### 3.2.7 调试的注意事项

本节描述了在 78K0 微控制器上进行片上调试的注意事项。

请认真阅读下列事项，以保证 EZ-CUBE 的正常使用。

#### (1) 处理用于调试的设备

不要在批量生产的产品上安装曾用于调试的设备，因为 Flash 存储器在调试过程中曾被多次重写，Flash 存储器的重写次数无法保证。

#### (2) 片上调试时改写 Flash 存储器

如果片上调试时进行了如下步骤，则设备中的 Flash 存储器会被改写。

<1> 下载程序

<2> 在存储器填充对话框中操作

<3> 在存储器拷贝对话框中操作

<4> 在存储器观察窗口进行修改操作

<5> 在汇编窗口中在线汇编

<6> 使用软件断点时，设置软件断点

从 Flash 存储器编程的完成到转入 GUI 控制需要一段时间。

#### (3) 软件断点

在程序运行中，不要重写软件断点地址上的数据。这包括自编程和 RAM 重写。如果重写了，那么存放在这个地址上的指令可能失效。

#### (4) 自编程期间的引导交换

无法仿真引导交换功能。因为引导交换移动了用于调试的存储器空间，于是调试通信就无法再进行。

#### (5) 堆栈指针初始化失败的中断功能

如果没有正确初始化堆栈指针，则在中断发生时，或者执行 PUSH 指令时，这个功能会执行一个断点。

如果在 reset 操作之后，紧接着有如下操作或者指令，堆栈指针初始化失败的断点功能就会失效。

- 设置一个软件断点
- 从寄存器窗口写堆栈指针
- 从存储器窗口写 Flash 存储器，等等

如果在堆栈指针没有正确初始化的时候发生了一个软件中断，在任务栏就会显示“Uninitialized Stack Pointer”。

如果发生了这种情况，后续的操作将无法正常进行，所以，请确保在用户程序中正确设置 SP。

#### (6) 下载 HEX 文件时的注意事项

当下载 HEX 文件时，请不要将目标转换参数 (-U) 的填充值指定为 0xFF 之外的值。

#### (7) 单步执行时的注意事项

在单步执行时，某些 SFRs（特殊功能寄存器）的内容不能被更新。这种情况下，用实时运行来执行指令。

#### (8) 在 EZ-CUBE 上安装时钟的注意事项

在 EZ-CUBE 上安装时钟时，即使目标系统的供电已经关闭，电源的电平也可能不等于 GND。

#### (9) POC 功能的仿真

当调试时目标系统的电源被关闭时，由于 EZ-CUBE 的 RESET\_OUT 引脚，目标设备会进入复位状态。因此，目标设备的 POC 功能执行伪仿真功能。注意 POC 的检测电压和目标系统的电压有  $\pm 1$  V 的误差。

### 3.3 Flash 编程

本节描述了使用 EZ-CUBE 对 78K0 微控制器进行 Flash 编程时的系统配置和启动/关闭顺序。

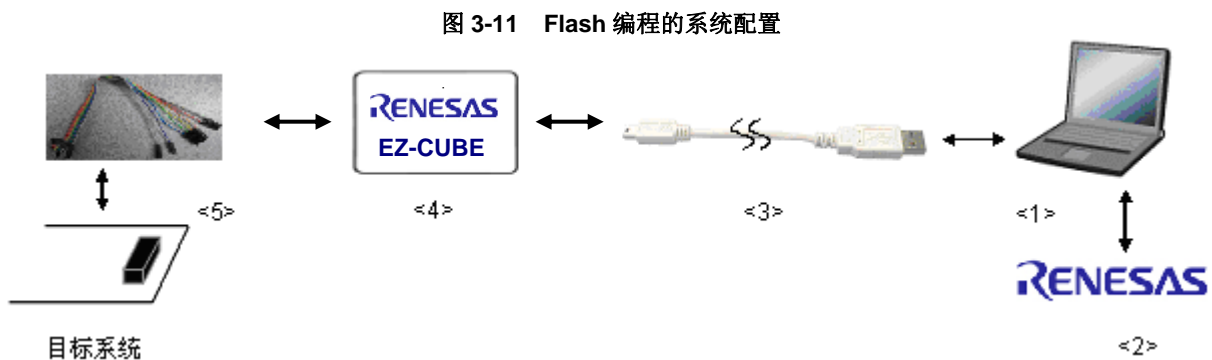
#### 3.3.1 编程功能的规格定义

表3-6 编程功能的规格定义

| 功能       | 规格定义说明                         |
|----------|--------------------------------|
| 主机接口     | USB 2.0                        |
| 主机接口     | UART                           |
| 目标系统电压   | 2.7到5.5 V（根据目标设备而定）            |
| 时钟源      | 可以提供8 MHz时钟<br>可以使用外部振荡器       |
| 电源       | 5 V±0.3V（最大电流：100mA）           |
| 获取设备特定信息 | 使用RENESAS ELECTRONICS的78K0参数文件 |
| 安全标识设置   | 支持                             |
| 脱机操作     | 不支持（必须连接主机）                    |

#### 3.3.2 系统配置

图 3-11 显示了 Flash 编程的系统配置



<1> 主机

要求有 USB 接口

<2> 软件

包含 PRF（Renesas Flash Programmer），USB 驱动等。

<3> USB 线缆（附件）

<4> EZ-CUBE（本产品）

<5> 8 引脚目标连接线（附件）

### 3.3.3 系统启动顺序

本节描述了系统启动顺序。请注意先后次序。

#### (1) 软件的准备和安装

要进行 Flash 编程，以下软件都是必需的。软件的准备和安装详细信息，请参考 EZ-CUBE 附带的配置手册。

- RFP
- USB 驱动

#### (2) 开关设置

SW-1 开关：请选择 "M1"

SW-2 开关：请根据实际目标设备使用选择

SW-3 开关：请选择 "Debug Mode"

SW-4 开关：请根据实际目标设备使用选择

SW-5 开关：请选择 "Other"

**注意事项**            在 **USB 线缆** 处于连接状态时，请勿改变开关的设置。

#### (3) 连接到目标系统

请务必在目标系统上电之前，将 EZ-CUBE 连接到目标系统。

#### (4) 连接 USB 线缆

在目标系统上电之前，将 EZ-CUBE 连接到主机。

通过 USB 将 EZ-CUBE 与主机连通后，EZ-CUBE 始终保持供电。

#### (5) 目标系统上电

打开目标系统的供电电源。如果电源选择开关设置在 "5"，则无需进行本步骤操作。

#### (6) 启动 RFP (Renesas Flash Programmer)

打开 RFP 软件。

### 3.3.4 应用样例

本节描述了使用RFP（Renesas Flash Programmer）的一系列基本操作，使用UPD78F0513D 作为示例设备。描述了执行[Autoprocedure(EP)] 命令到为目标设备编程的整个过程。 命令的详细内容，请参加RFP的用户手册。

本节系列操作所需要的条件如下。

<目标系统>

目标设备: UPD78F0513D  
电源电压: 5 V±0.3V (EZ-CUBE 提供)  
通信通道: UART, 115,200 bps

<EZ-CUBE>

开关: SW-1 开关: M1  
SW-2 开关: Int. Clock  
SW-3 开关: Debug Mode  
SW-4 开关: 5  
SW-5 开关: Other

<RFP >

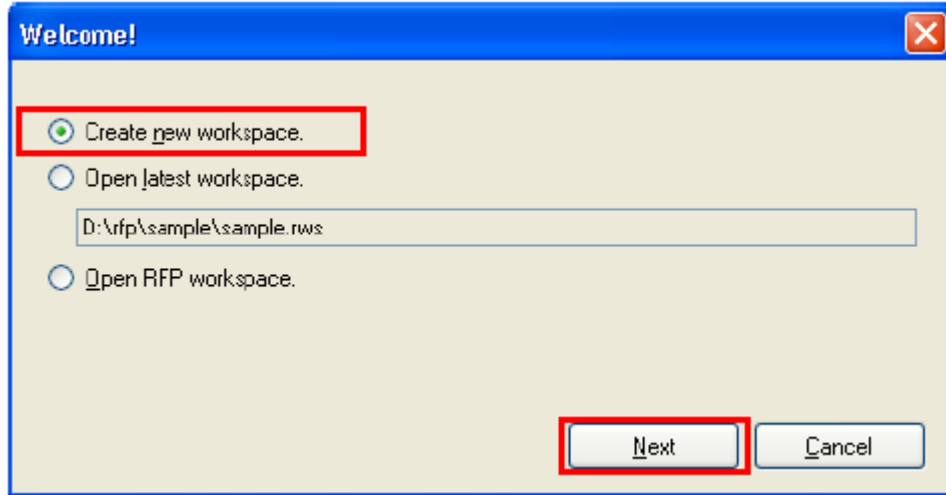
时钟设置: 8 MHz, x1  
操作模式: chip  
程序文件: sample.hex  
命令参数: 选中[Blank check before Erase]  
选中[Verify after Program]

(1) 设置编程环境

请按照下面描述的步骤<1>至<6>顺序设置编程环境。

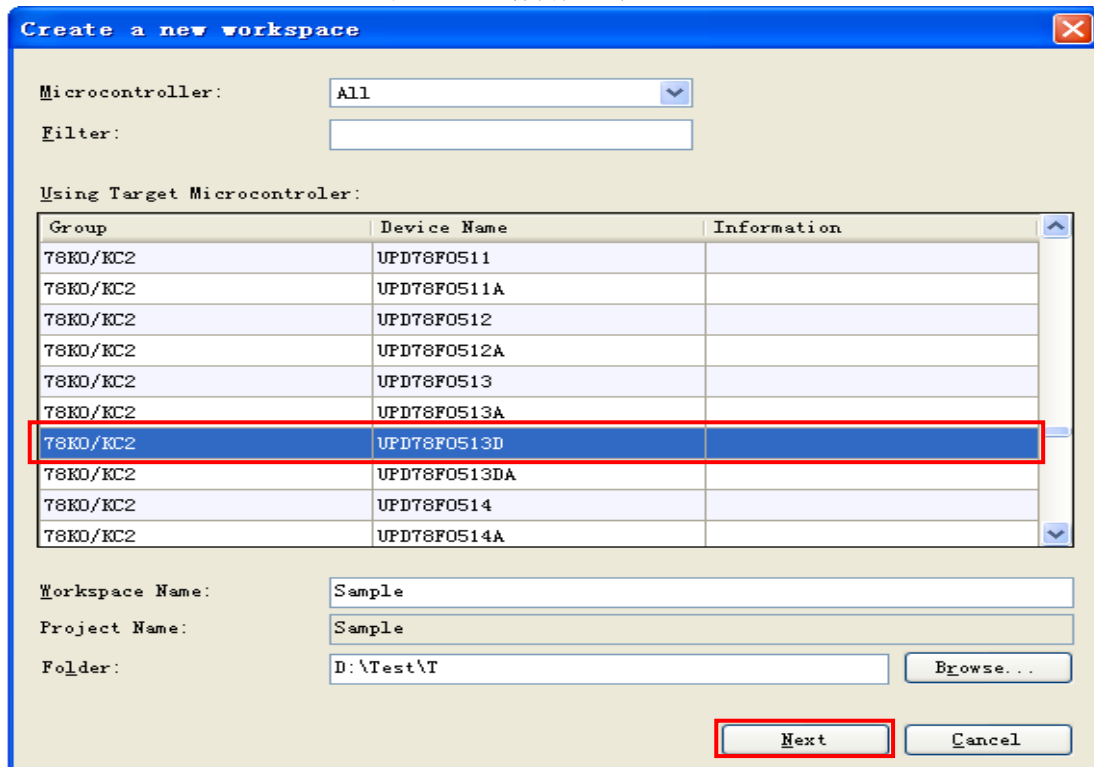
- <1> 点击【开始】按钮，指向[程序]，[Renesas Electronics CubeSuite+]，[Programming Tools]，[Renesas Flash Programmer Vx.xx]，点击[Renesas Flash Programmer Vx.xx [Basic mode]]，启动 RFP 软件。
- <2> 创建[ Create new workspace. ]，如图 3-12 所示

图 3-12 创建



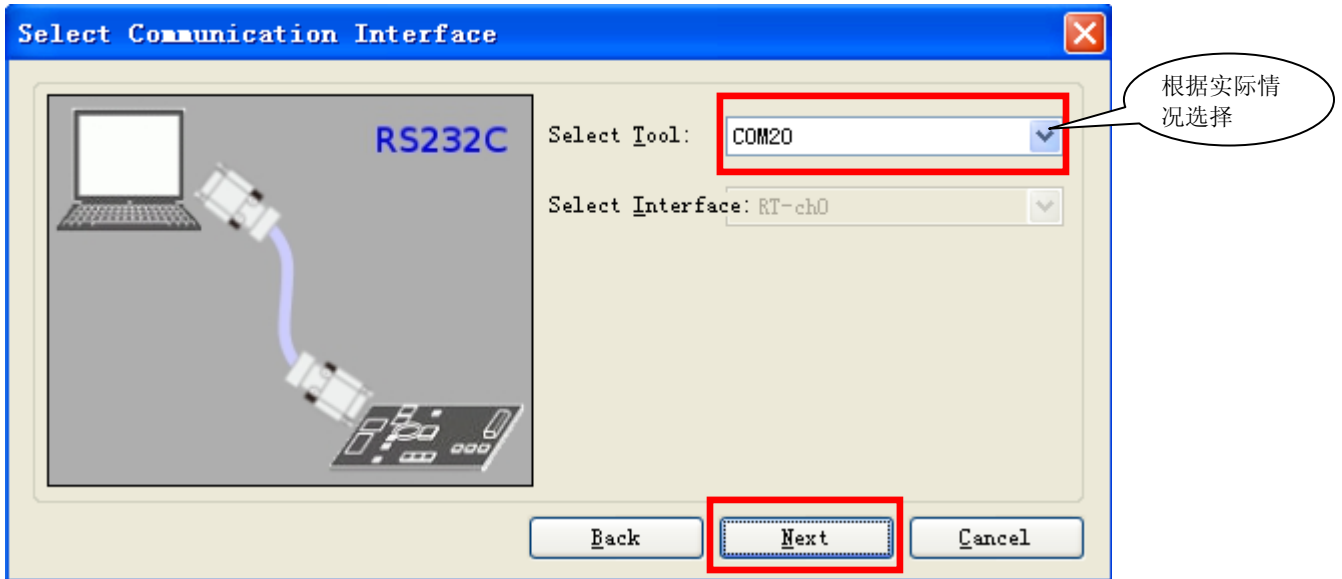
- <3> 选择编程芯片，如 78F0513D。如图 3-13 所示。

图 3-13 选择编程芯片



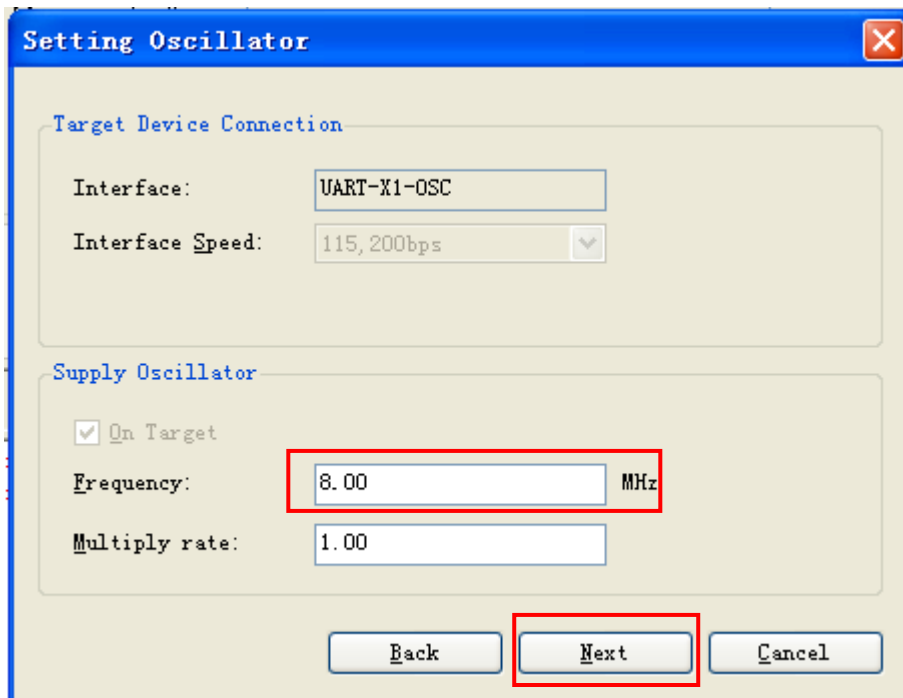
<4> 选择通信接口[ Select Communication Interface ], 如图 3-14 所示。

图3-14 [Select Communication Interface] 对话框



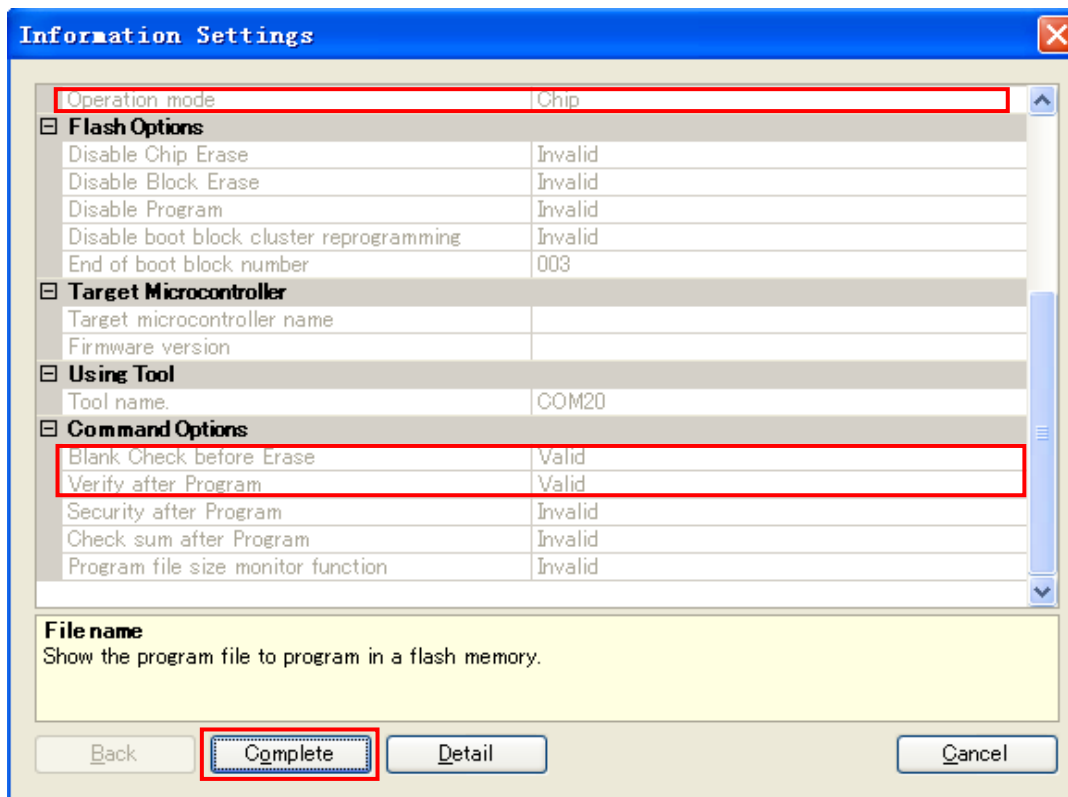
<5> 选择时钟[Setting Oscillator ], 如图 3-15 所示。

图3-15 [Setting Oscillator] 对话框



<6> 根据实际需要设置[Information Settings] 选项卡中的选项。详细内容，请参考 RFP 的用户手册。

图 3-16 [Information Settings] 选项卡



## (2) 程序文件的选择

请按照下述的步骤<1>至<3>选择程序文件。

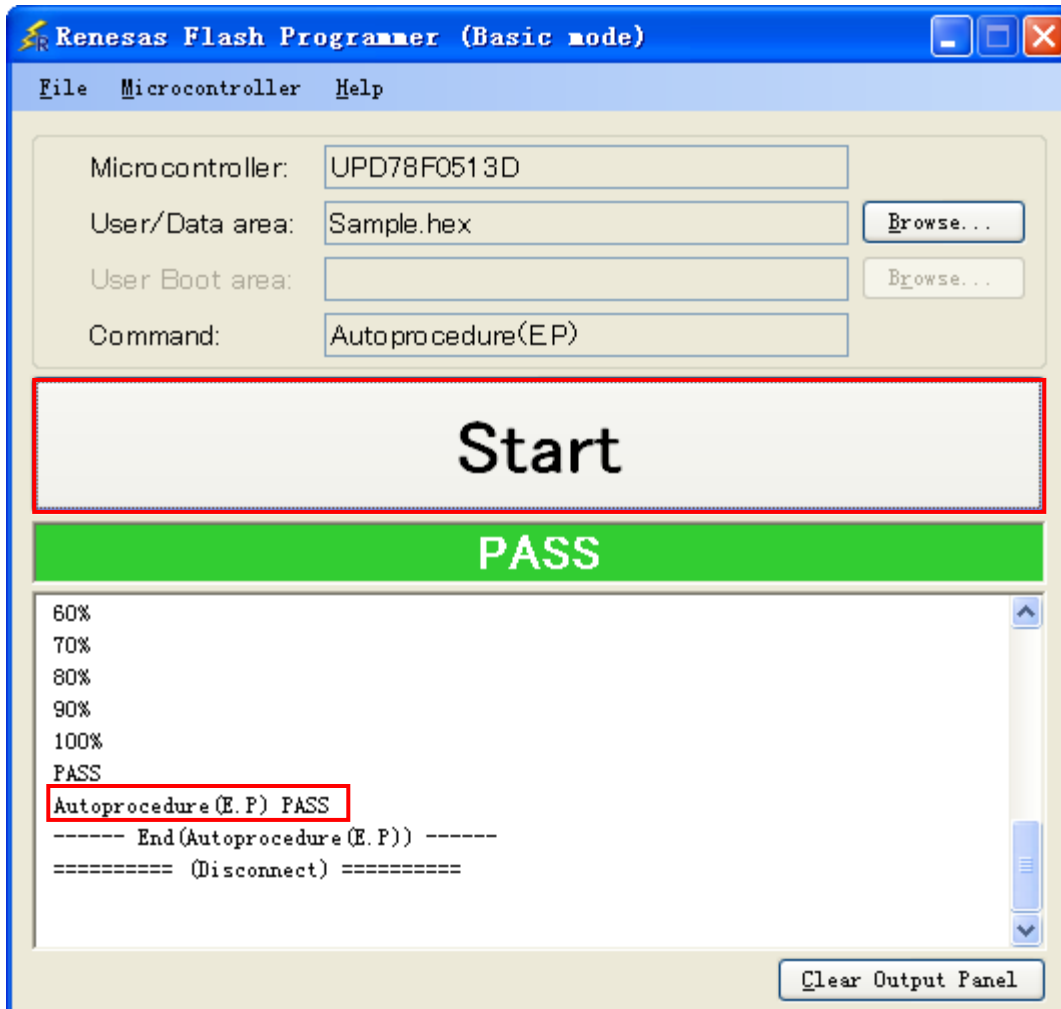
- <1> 在[User/Data area]区域点击 **Browse...** 按钮。
- <2> 弹出程序文件选择对话框。
- <3> 选择要写入目标设备的程序文件，然后点击 **打开** 按钮。

## (3) 执行[Autoprocedure(EP)]命令

- <1> 点击菜单栏的[Microcontroller]，然后选择[Autoprocedure(EP)]命令。
- <2> 点击 **Start** 按钮

[Blank Check]、[Erase]（如果目标设备上的Flash 存储器非空）、[Program]、[Verify]命令将会在目标设备上依次执行。当本命令正常执行完成后，弹出如下窗口。

图3-17 [Autoprocedure(EP)]命令执行完成之后



### 3.3.5 系统关闭顺序

请按照下列顺序终止 Flash 编程和关闭系统。

如果未按照本文的操作顺序，目标系统或者 EZ-CUBE 可能会被损坏。

(1) **终止 RFP 软件**

如果你不需要对其它设备进行编程，则请终止 RFP。

(2) **关闭目标系统电源**

关闭目标系统的电源，如果电源选择开关设置在“5”，则此步骤可以略过。

(3) **移除 USB 线缆**

从 EZ-CUBE 或者主机上拔掉 USB 连线。

如果电源选择开关设置为“T”时，可以在此步骤之前先执行 (4) **移除目标线缆**。

(4) **移除目标线缆**

从 EZ-CUBE 或者主机上拔掉目标线缆。



### 3.3.6 Flash 编程的注意事项

本节描述了 Flash 编程的注意事项。请认真阅读，以保证 EZ-CUBE 的正确使用。

为了提高写入质量，请在使用 EZ-CUBE 之前对下列问题进行充分了解，并加以校验评估。

- 电路的设计要符合 EZ-CUBE 用户手册中对设备的描述。
- 设备，RFP 和 EZ-CUBE 的使用都要遵循各自手册的描述。
- 目标系统的供电电源要稳定。

## 第 4 章 如何在 78K0R 微控制器上使用 EZ-CUBE

本章描述了如何使用 EZ-CUBE 对 78K0R 微控制器进行片上调试和 Flash 编程。

片上调试是指通过芯片内置的调试功能对已经装载在目标系统上的微控制器进行调试的方法。由于这种调试方法是通过板上的目标设备实行的，所以非常适合现场调试。

Flash 编程是将程序写入设备内置的 Flash 存储器中的方法。可以在板上对设备进行擦除，写入和校验。

请先将 EZ-CUBE 的固件更新为用于 78K0R，具体请参照 (1) - (3)。详细内容请参见 **1.4 固件更新**。

- (1) 使用 USB 缆连接 EZ-CUBE 和 PC，**不要将 EZ-CUBE 与目标板相连**。
- (2) 启动【QBEZUTL.exe】，并指定“78K0R\_OCD\_FW.hex”固件文件。
- (3) 点击[Start]按钮，然后指定用于 78K0R 固件下载到 EZ-CUBE。

如果首次用 78K0R 微控制器作为目标设备来使用 EZ-CUBE，那么请仔细阅读以下内容。

### 4.1 目标系统设计

要用 EZ-CUBE 和目标系统进行通信，必须在目标系统上安装对应的通信电路。本节描述了相关电路的设计和连接插头的安装。

### 4.2 片上调试

本节描述了用 EZ-CUBE 进行片上调试的系统配置和启动方法。

### 4.3 Flash 编程

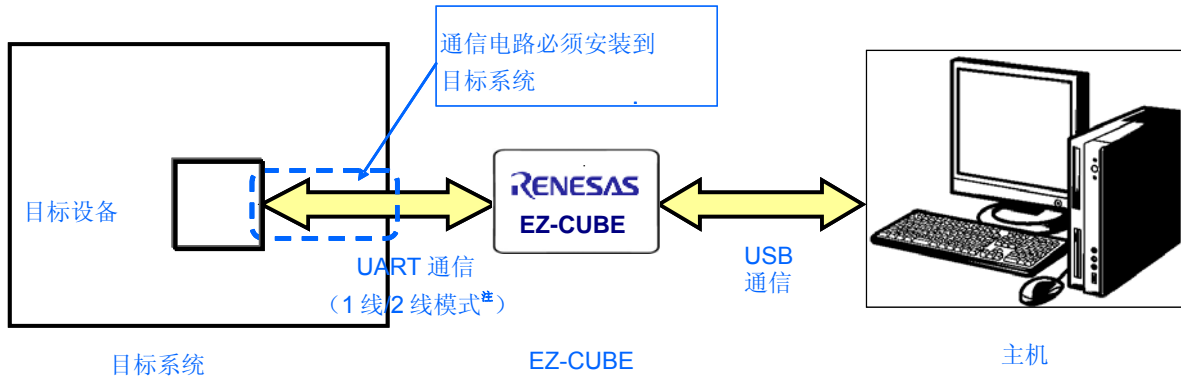
本节描述了用 EZ-CUBE 进行 Flash 编程的系统配置和启动方法。

## 4.1 目标系统设计

本节描述了片上调试和 Flash 编程所需目标系统的电路设计。

图 4-1 显示了 EZ-CUBE 通信的整体接口框图。如图所示，EZ-CUBE 和目标系统的目标设备之间进行串行通讯。为了通讯，目标系统上必须安装用于通信的电路。请参考本节的内容设计合适的电路。

图 4-1. 通信接口框图



注 单线模式：使用TOOL0引脚进行单线UART通信  
 双线模式：使用TOOL0和TOOL1引脚进行双线UART通信

表 4-1. 单线模式和双线模式的不同之处

| 通信模式 | Flash编程功能 | 调试功能  |
|------|-----------|---|
| 单线模式 | 可用        | 调试时所用的用户资源<br>内部ROM:1036字节<br>内部RAM:6字节(堆栈) |
| 双线模式 | 可用        | 调试时所用的用户资源<br>内部ROM:100字节<br>内部RAM:6字节(堆栈)  |

### 4.1.1 引脚配置

本节描述了 EZ-CUBE 和目标系统之间使用的接口信号。表 4-2 列出了使用 8 引脚线缆时的引脚配置。表 4-3 描述了每个引脚的功能。在调试和编程的时候，引脚配置有很大区别，所以请根据后面的章节中描述的电路连接例程设计合适的电路。

表 4-2. 引脚配置

| 引脚编号 | 引脚名称 <sup>注</sup> |
|------|-------------------|
| 1    | GND               |
| 2    | RESET_IN          |
| 3    | Vdd               |
| 4    | FLMD0             |
| 5    | CLK               |
| 6    | RxD.              |
| 7    | RESET_OUT         |
| 8    | TxD               |

注 EZ-CUBE 内的信号名称。

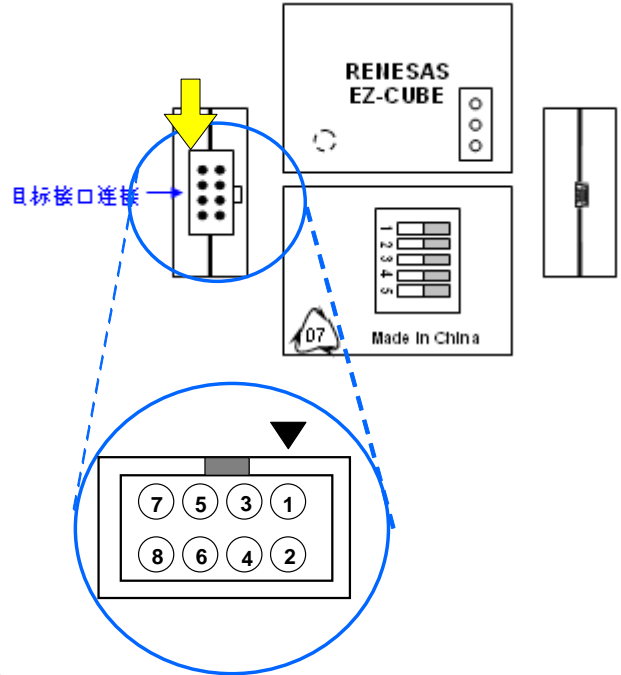


表 4-3. 引脚功能

| 引脚名称      | IN/OUT <sup>注</sup> | 描述                     |
|-----------|---------------------|------------------------|
| RESET_IN  | IN                  | 该引脚用于从目标系统输入 reset 信号  |
| RESET_OUT | OUT                 | 该引脚用于向目标系统输出 reset 信号  |
| CLK       | OUT                 | 该引脚用于向目标系统输出时钟信号       |
| FLMD0     | OUT                 | 该引脚用于设定目标设备进入调试模式或编程模式 |
| RxD       | IN                  | 该引脚用于接收来自目标设备的命令/数据    |
| TxD       | OUT                 | 该引脚用于向目标设备发送命令/数据      |

注 以 EZ-CUBE 为参照基准。

### 4.1.2 电路连接示例

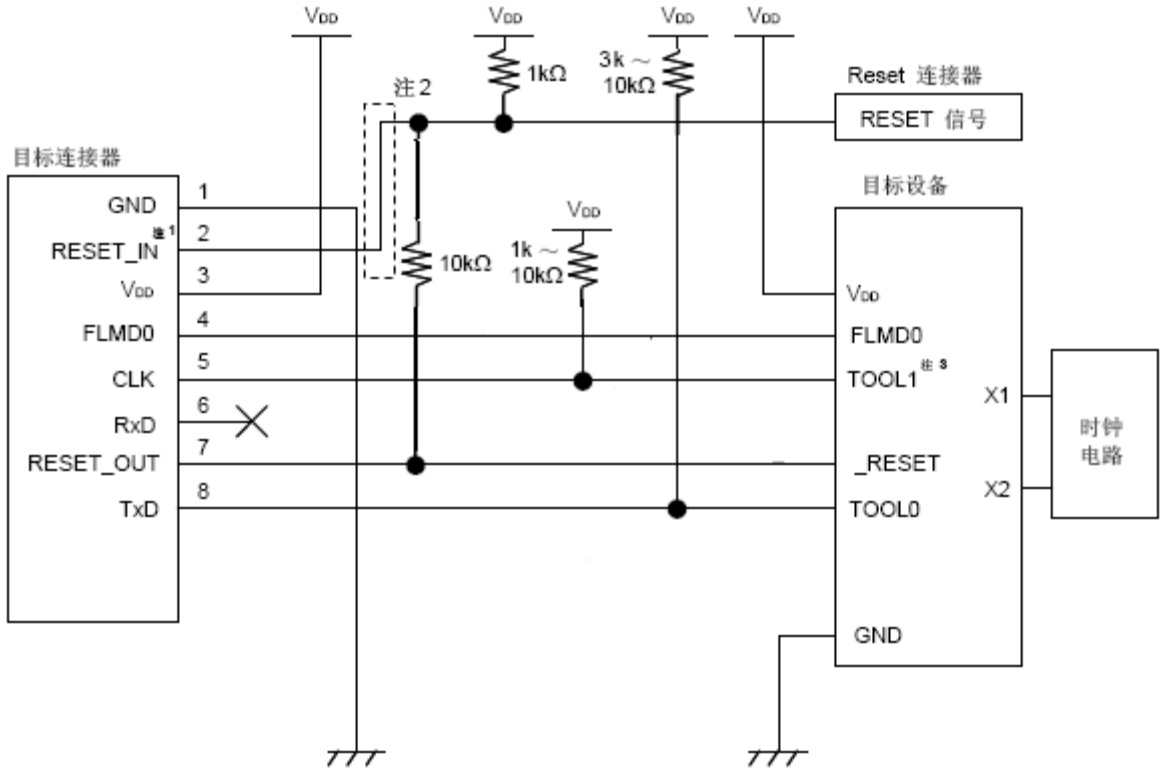
在调试和编程的时候，请根据描述的电路连接示例来设计合适的电路。

目标系统上的电路设计会根据接口信号而有所不同。接口的引脚名称可能会因目标设备有所不同。详细信息请参考目标设备的用户手册。

请确认使用目的，参照表 4-1 和相关的电路连接示例。

**注意事项** 在电路连接示例中出现的常数值为参考值。如果进行 flash 编程旨在量产时，请彻底评估目标设备的规格是否满足要求。

图 4-2. 电路连接示例



- 注
1. 这种连接的设计是假定目标系统的复位电路包含有N沟道开漏缓冲器（输出阻抗：100欧姆或更小）。详细信息请参考4.1.3 reset 引脚的连接。
  2. 仅当flash编程时，不需要虚线所画的电路。
  3. 这个连接在双线通信模式下是必须的，但在单线模式下不需要。

### 4.1.3 reset 引脚的连接

本节描述了电路中 reset 引脚的连接，此处必须特别注意，电路连接示例见前面的章节。

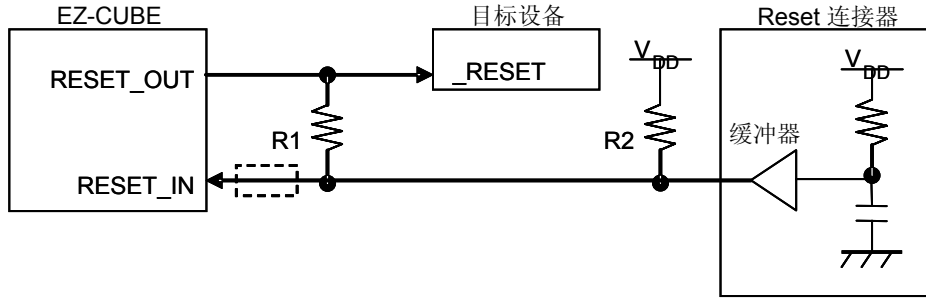
片上调试时，来自目标系统的 reset 信号输入 EZ-CUBE，被屏蔽后，再输入到目标设备。于是 reset 信号的连接根据是否连接 EZ-CUBE 而有所不同。

Flash 编程时，电路的设计必须保证目标系统的 reset 信号和 EZ-CUBE 的 reset 信号不冲突。

**推荐通过电阻自动切换 reset 信号。**图 4-3 说明了 4.1.2 电路连接示例中描述的 reset 引脚连接情况。

这种连接的设计是假定目标系统的复位电路包含有 N 沟道开漏缓冲器（输出阻抗：100 欧姆或更小）。当 EZ-CUBE 的 RESET\_IN/OUT 逻辑反转时，VDD 或 GND 的电平可能会不稳定，所以请注意备注中所描述的条件。

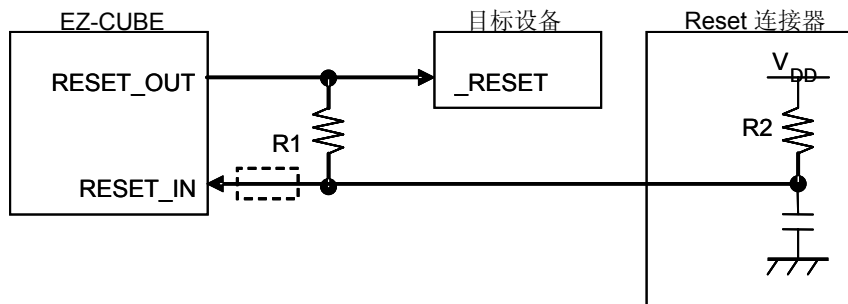
图 4-3. 包含缓冲器的复位电路的连接



**备注** 需确保R1的阻值至少是R2阻值的十倍，R1为10 kΩ或更大。  
如果复位电路的缓冲器是CMOS 输出，则无需上拉电阻R2。  
当只用于Flash编程时，可以略去虚线框内的电路。

图 4-4 所示的电路连接情形是目标系统的复位电路不包含缓冲器，并且 reset 信号仅通过电阻或电容产生。设计电路时注意备注中描述的条件。

图 4-4. 不包含缓冲器的复位电路的连接



**备注** 需确保R1的阻值至少是R2阻值的十倍，R1为10 kΩ或更大。  
当只用于Flash编程时，可以略去虚线框内的电路。

#### 4.1.4 在目标系统上安装连接器

在设计目标系统时，必须安装 EZ-CUBE 和目标系统的连接器。可以选择 2.54mm 间距的 8 引脚通用型连接器。

除了专用的连接线以外，同时也支持散线连接。

## 4.2 片上调试

本节描述了使用 EZ-CUBE 进行片上调试的系统配置，启动/关闭顺序和调试中的若干注意事项。

### 4.2.1 调试功能

表 4-4 列出了以 78K0R 微控制器为目标设备时的调试功能。

表 4-4. 调试功能

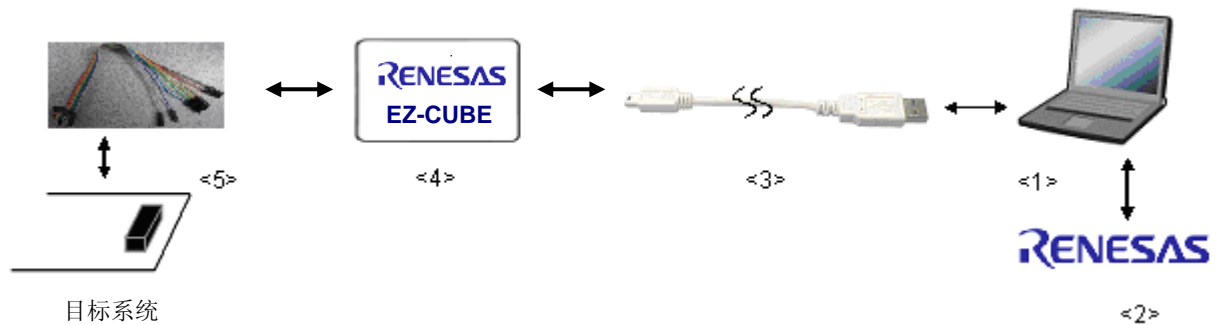
| 功能         | 规格  |
|------------|---|
| 目标MCU      | 78K0R系列<br>78K0R/ Kx3,Lx3   |
| 安全功能       | 10字节ID码验证   |
| 下载         | 支持  |
| 运行         | 全速执行，忽略断点执行，单步执行，下一步执行，运行至此，重启  |
| 硬件断点       | 1个  |
| 软件断点       | 多个  |
| 调试时所用的用户区域 | 单线模式：内部ROM：1036字节，内部RAM：6字节 <sup>※</sup><br>双线模式：内部ROM：100字节，内部RAM：6字节 <sup>※</sup> |
| 用于调试的功能引脚  | 单线模式：TOOL0<br>双线模式：TOOL0，TOOL1  |

注 详细情况请参考 4.2.5 用户资源的保留，设置安全 ID 以及片上调试选项字节。

### 4.2.2 系统配置

图 4-5 显示片上调试的系统配置。

图 4-5. 片上调试的系统配置



<1> 主机

要求有 USB 接口

<2> 软件

包括 CubeSuite+ for China，USB 驱动等。

<3> USB 线缆（附带）

<4> EZ-CUBE (本产品)

<5> 8 引脚目标线缆（附带）

### 4.2.3 系统启动顺序

本节描述了系统启动的顺序。请注意按次序操作。

#### (1) 软件的准备和安装

以下的软件是片上调试所必需的。关于软件的准备和安装的详细信息，请参考 EZ-CUBE 附带的安装手册。

- CubeSuite+
- USB 驱动

#### (2) 开关设置

- SW-1 开关：请选择 "M2"
- SW-2 开关：请根据实际目标设备使用选择
- SW-3 开关：请选择 " Debug Mode"
- SW-4 开关：请根据实际目标设备使用选择
- SW-5 开关：请选择 "Other"

**注意事项**

1. 在 **USB 线缆处于连接状态时，请勿改变开关的设置。**
2. **最大允许电流 100mA，所以请勿将 EZ-CUBE 连接到电流很大的目标系统上。在 EZ-CUBE 与主机连通后，主机始终保持向 EZ-CUBE 供电。**

#### (3) 连接目标系统

将 EZ-CUBE 到目标系统，在开启目标系统电源之前建立连接。

#### (4) 连接 USB 线缆

请在目标系统上电之前，将 EZ-CUBE 连接到主机，模式灯点亮为红色。

#### (5) 目标系统上电

打开目标系统的电源。如果电源选择为“5”，则这一步并不是必需的。

#### (6) 启动调试器

打开调试器。

此步骤及之后的操作，请参考 CubeSuite+的用户手册。

如果调试器无法正常开启或者操作不稳定，可能是由以下问题引发的。

- EZ-CUBE 和目标系统之间通信错误
  - 固件是否更新；
  - 开关是否选择正确；
  - 目标连线是否正确。
- 用户资源没有预留或者安全 ID 没有设置
  - 为了使用 EZ-CUBE 进行调试，必须预留出调试监控程序区，也必须设置安全 ID 和片上调试选项字节。详细信息请参考 **4.2.5 节用户资源的保留，设置安全 ID 以及片上调试选项字节。**
- 使用了不支持的软件（调试器，设备文件或者固件）
  - 所使用的软件可能不支持目标设备的调试。
- EZ-CUBE 损坏
  - EZ-CUBE 可能被损坏。



#### 4.2.4 系统关闭顺序

停止调试并关闭系统时，请按照下列顺序操作。

如果没有按照下列顺序操作，目标系统或者 EZ-CUBE 可能被损坏。

**(1) 停止调试器**

停止调试器的运行并断开调试器连接。

**(2) 关闭目标系统电源**

关闭目标系统的电源。如果电源选择开关设置为“5”的位置，则无需本步操作。

**(3) 移除 USB 线缆**

从 EZ-CUBE 或主机上将 USB 线缆拔下。

**(4) 移除目标线缆**

从 EZ-CUBE 或目标系统上将目标线缆拔下。

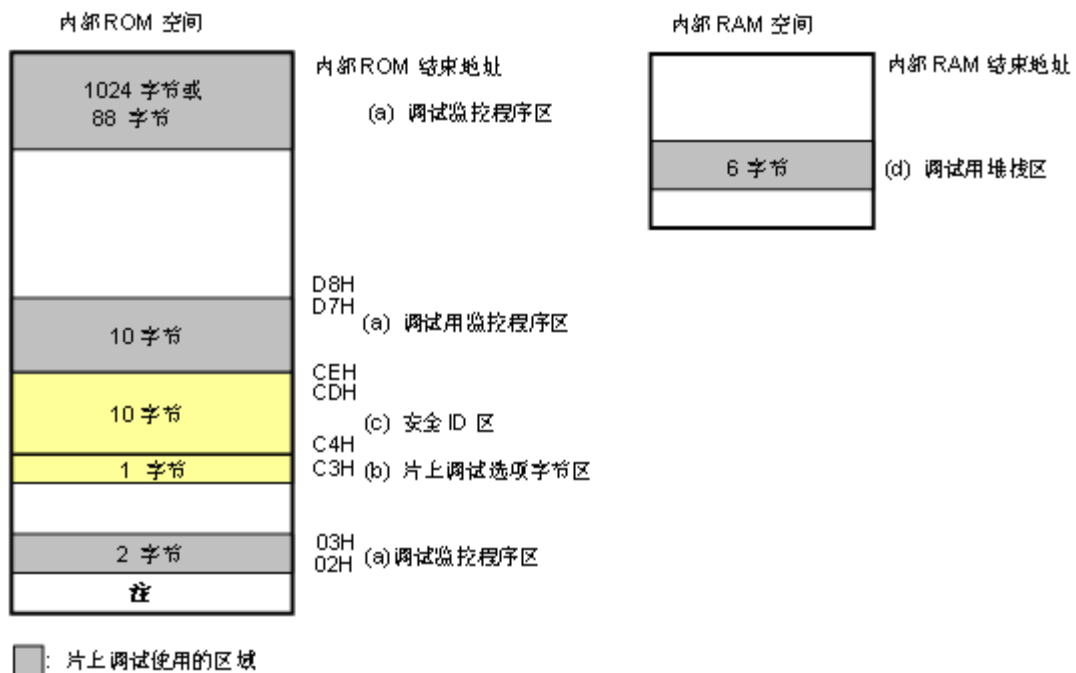
#### 4.2.5 用户资源的保留，设置安全 ID 以及片上调试选项字节

用户必须做好下面的保留以保证 EZ-CUBE 和目标设备之间的通信以及实现每一个调试功能。使用 Renesas ELECTRONICS 的编译器，这些都可以通过链接选项设置。请参考下面的描述并设置。

- 内存空间的保留

图4-6中的阴影部分是存放调试监控程序的保留区域，所以用户程序或数据不能存放在这些空间。这些空间必须保证不被用户程序使用。另外，这些区域也不能被用户程序重写。

图 4-6. 分配给调试监控程序的内存空间



注 调试过程中，复位向量被重写，指向监控程序区。

- 设置安全 ID 和片上调试选项字节

安全 ID 或者片上调试选项字节必须存放在图 4-6 的黄色区域中 (0xC3 和 0xC4 到 0xCD)，以防止没有认证的人读取存储器。

**(a) 调试监控程序的保留区域**

这是一个放置调试监控程序的区域。监控程序为调试通信接口和CPU的运行或者断点做初始化操作。用户程序或数据不能存放在片上调试选项字节周边的 22 个字节的区域以及内部ROM底端的 1,024 字节区域内。另外，复位向量会被重写并指向放置监控程序的地址。

**注** 双线模式时不使用伪 RRM 功能时的 88 个字节区域。如果内部 ROM 的结束地址是 0x3FFFF，88 个字节的监控程序被定位到 0x3FFA8 ~ 0x3FFFF 区。

[如何预留区域]

下列示例用于说明使用 RENESAS ELECTRONICS 的汇编器或编译器的预留区域。在如图 4-7 中选择“Use on-chip debug”复选框。选项字节周边的 22 个字节区域被预备。通过设置起始地址和大小，内部 ROM 结束地址之前的 1,024 个字节区域也被预留。

**(b) 片上调试选项字节区域**

这个区域用来存放安全设置以防无认证的人读取 Flash 存储器。调试器根据设置的值对目标设备进行操作，如下表所示。

**表 4-5. 片上调试选项字节设置和操作**

| 设置值  | 描述                             | 备注                   |
|------|--------------------------------|----------------------|
| 0x04 | 即使连上了片上调试仿真器（如EZ-CUBE）也不能进行调试。 | 这个设置只在Flash编程或自编程时有效 |
| 0x85 | 不管安全ID码认证失败多少次都不会擦除片上Flash存储器  | -                    |
| 0x84 | 如果安全ID码认证失败擦除所有片上Flash存储器区域    | -                    |
| 其它   | 禁止设置                           | -                    |

**注意事项** 只能设置第7和第0位（OCDENSET 和 OCDERSD）。保证设置第6到第0位为000010B。

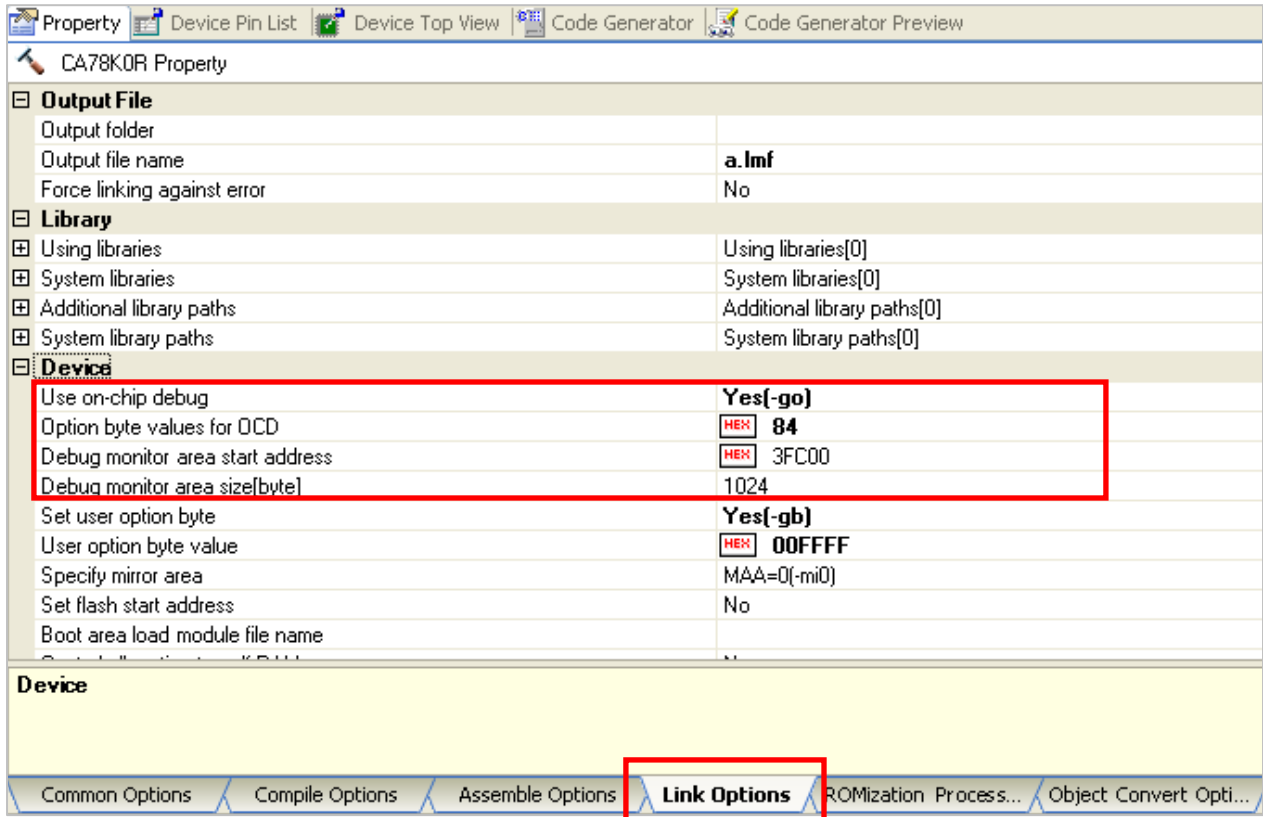
**备注** 第3位到第1位的值在片上调试的时候会被改变，所以设置后这些值会不确定。但不管怎么样，确保设置为初始值（0, 1, 0）。

[如何设置]

下面说明了用 RENESAS ELECTRONICS 的汇编器或编译器时的设置样例。如图 4-7 中显示了在 CubeSuite+ 中的“Build Tool”的[Link Options]选项卡。选中下图中的“Use on-chip debug”选择框，保留区域大小设置为 1024 字节。

**例** 把控制值设置为 0x84。

图 4-7. 预留调试监测区域和片上调试选项字节样例



**注意事项** 如果片上调试选项字节区域（地址 0xC3）的第 7 位（OCDENSET）被 Flash 编程或自编程设置为“0”，调试被禁止并且调试器不能被启动。如要允许调试，通过 Flash 编程擦除 Flash 存储器。

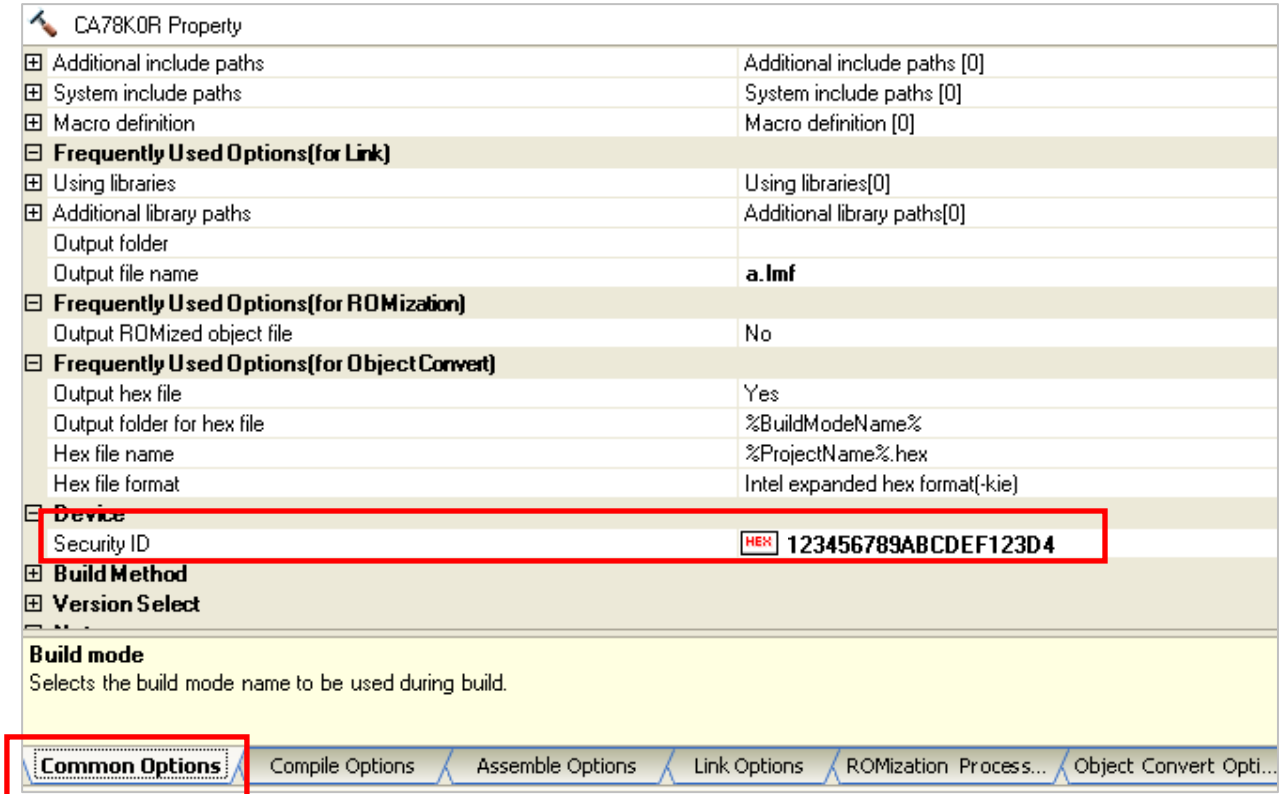
### (c) 安全 ID 区域

这个区域用来存放安全设置以防无认证的人读取 Flash 存储器。安全 ID 功能就像启动调试器的密码。只有在调试器启动时输入的安全 ID 和存放在这个区域的安全 ID 匹配时，调试器才能正常启动。如果 ID 码不匹配，调试器根据片上调试选项字节区域设置的值对目标设备进行操作（参考表 4-5）。

#### [如何设置]

当使用 RENESAS ELECTRONICS 的汇编器或编程器时，设置安全 ID 就可以在安全 ID 区域产生一个 ID 码。如图显示了 CubeSuite+ 中的“Build Tool”的[Common Option]选项卡。请选中图中的“Security ID”选择框，可以设置为任意 ID 值（10 字节长度）。

图 4-8. 安全ID设置样例



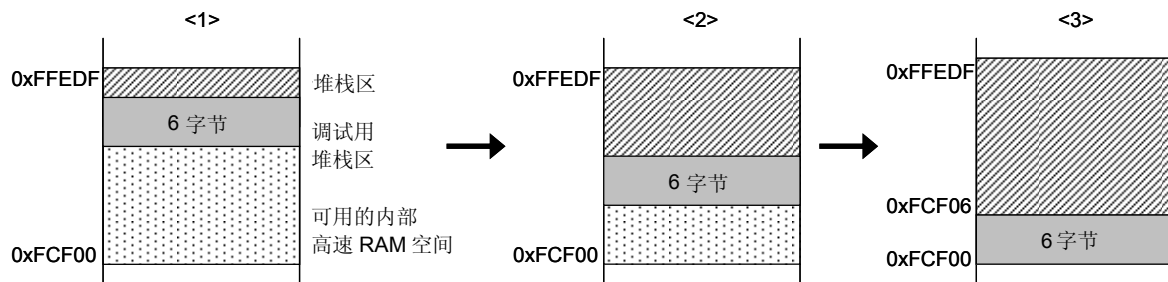
**注意事项** 如果已经忘记安全 ID 码，请通过 flash 编程擦除 flash 存储器，然后再次设置安全 ID。

(d) 调试用堆栈区域的保留

这个区域需要 6 个字节作为调试的堆栈区域。因为这个区域分配在紧邻堆栈区域之前，所以这个区域的地址根据堆栈的增加和减少而有所改变。就是说，在使用的堆栈区域之外有 6 个额外的字节。

图 4-9 说明了堆栈区域增加时的情况，图中的内部高速 RAM 从 0xFCF00 开始。

图 4-9. 调试所用的堆栈地址的变化



[如何预留区域]

通过在用户程序使用堆栈之外再多定义 6 个字节来设置堆栈指针。确保堆栈指针不会超过内部高速 RAM 的起始地址。

**备注** 关于如何预留自编程序的堆栈区域请参考自编程序手册。

### 4.2.6 调试的注意事项

本节描述了在 78K0R 微控制器上进行片上调试的注意事项。

请认真阅读下列事项，以保证 EZ-CUBE 的正常使用。

(1) 处理用于调试的设备

不要在批量生产的产品上安装曾用于调试的设备，因为 Flash 存储器在调试过程中被多次重写，Flash 存储器的重写次数无法保证。另外，不要把调试监控程序写入批量生产的产品。

(2) Flash 自编程

如果存放调试监控程序的区域被 Flash 自编程重写，调试器就不能正常操作。这个注意事项对这个区域的引导交换操作也适用。

(3) 复位后的操作

在外部复位或内部复位后，监控程序会进行调试初始化处理。因此，从产生复位到用户程序执行的时间和实际设备的操作有所不同。

(4) 使用实机调试而不使用 EZ-CUBE

如果使用实机调试而不使用 EZ-CUBE，请使用 RFP 写入用户程序。通过调试器下载的程序包括监控程序，这样的程序如果在有使 TOOL0 引脚变成低电平的操作时会出现误操作。

(5) 调试器启动后的操作

调试器启动后，如果调试器配置对话框中的目标设备连接设置和上次的调试设置不同，内部 Flash 存储器会被擦除。

(6) 通过 Flash 编程写入程序后的调试

如果通过 RFP 把程序写入内部 Flash 存储器，即使在片上调试选项字节设置中允许了片上调试，也不能进行片上调试。如果要对这样的目标设备要进行调试，可以先用 RFP 擦除内部 Flash 存储器后通过调试器下载程序。

(7) 默认 LVI 起始功能设置(地址 C1H)

调试过程中，调试监控程序停止了 C1H 地址处 LVI 的默认启动功能。因此，即使调试结束，LVI 默认启动功能仍保持被停止状态，通过 flash 编程功能改变地址 C1H 的设置。

(8) 片上调试选项字节设置（地址 C3H）

片上调试选项字节设置被调试器重写为任意值。

(9) 在单线模式下调试

在单线模式下执行调试的情况下（通过在调试器配置对话框的 Connection with Target Board 区 Communication method 选择 1 line type(TOOL0)），当内部高速振荡器被用于 CPU 的工作时钟时，如果在调试器启动和中断发生之间频率变化太大（改变寄存器时除外），中断可能不会正常发生。这种情况在工作电压或温度改变太大时也可能发生。

### 4.3 Flash 编程

本节描述了使用 EZ-CUBE 对 78K0R 微控制器进行 Flash 编程时的系统配置和启动/关闭顺序。

#### 4.3.1 编程功能的规格定义

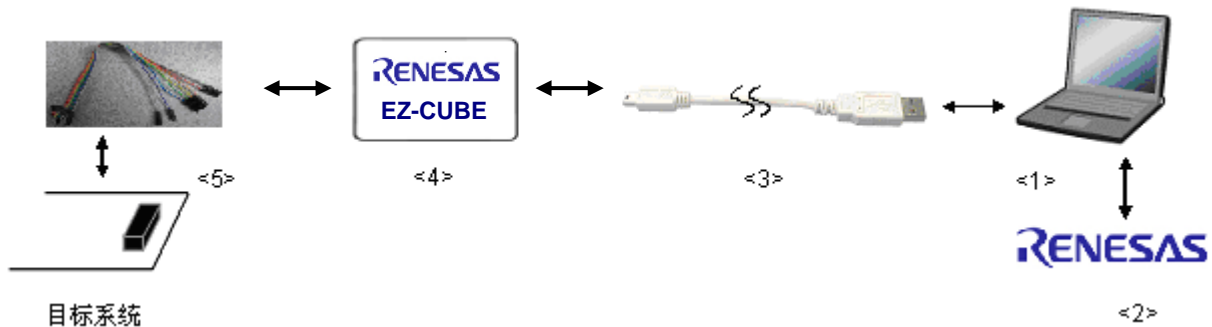
表 4-6. 编程功能的规格定义

| 功能       | 规格定义说明                          |
|----------|---------------------------------|
| 主机接口     | USB2.0                          |
| 目标接口     | UART（单线模式）                      |
| 目标系统电压   | 2.7到5.5 V（根据目标设备而定）             |
| 时钟提供     | 使用内部高速振荡器                       |
| 电源       | 5 V±0.3V（最大电流：100mA）            |
| 获取设备特定信息 | 使用RENESAS ELECTRONICS的78K0R参数文件 |
| 安全标志设置   | 支持                              |
| 脱机操作     | 不支持（必须连接主机）                     |

#### 4.3.2 系统配置

图 4-10 说明了 Flash 编程的系统配置。

图 4-10. Flash 编程的系统配置



- <1> 主机  
带有 USB 接口
- <2> 软件  
包括 RFP（Renesas Flash Programmer），USB 驱动等。
- <3> USB 线缆（附带）
- <4> EZ-CUBE（本产品）
- <5> 8 引脚目标线缆（附带）

### 4.3.3 系统启动顺序

本节描述了系统启动顺序。请注意先后次序。

#### (1) 软件的准备和安装

要进行 Flash 编程，以下软件都是必需的。有关软件的准备和安装详细信息，请参阅 EZ-CUBE 附带的配置手册。

- RFP
- USB 驱动

#### (2) 开关设置

SW-1 开关：请选择 "M2"

SW-2 开关：请根据实际目标设备使用选择

SW-3 开关：请选择 " Debug Mode"

SW-4 开关：请根据实际目标设备使用选择

SW-5 开关：请选择 "Other"

**注意事项**            在 USB 线缆处于连接状态时，请勿改变开关的设置。

#### (3) 连接目标系统

在目标系统上电之前，将 EZ-CUBE 连接到目标系统。

#### (4) 连接 USB 线缆

在目标系统上电之前，将 EZ-CUBE 连接到主机。

#### (5) 目标系统上电

打开目标系统的电源。如果电源选择开关设置为 "5"，则这一步不是必需的。

#### (6) 启动 RFP (Renesas Flash Programmer)

打开 RFP 软件。

#### 4.3.4 应用样例

本节讲述了使用RFP（Renesas Flash Programmer）的一系列基本操作，用UPD78F1166 作一个样例。说明执行[Autoprocedure (EP)] 命令为目标设备进行编程。命令的详细内容，请参加RFP的用户手册。

本节系列操作所需要的条件如下。

##### <目标系统>

目标设备: UPD78F1166  
电源电压: 5 V±0.3V (EZ-CUBE 提供)  
通信通道: UART, 115,200 bps

##### <EZ-CUBE>

开关:  
SW-1 开关: M2  
SW-2 开关: Int. Clock  
SW-3 开关: Debug Mode  
SW-4 开关: 5  
SW-5 开关: Other

##### <RFP>

操作模式: chip  
待写文件: sample.hex  
命令参数: 选中[Blank check before Erase]  
选中[Verify after Program]

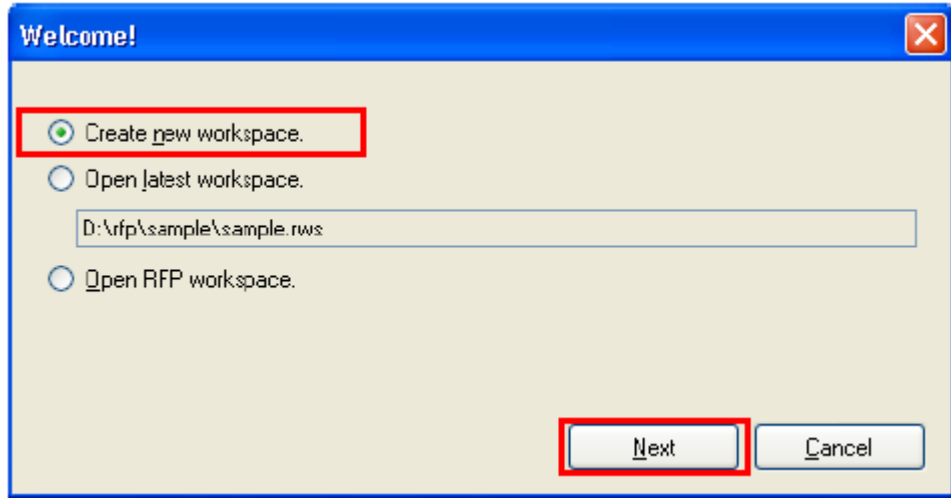


(1) 设置编程环境

请按照下面描述的步骤<1>至<5>顺序设置编程环境。

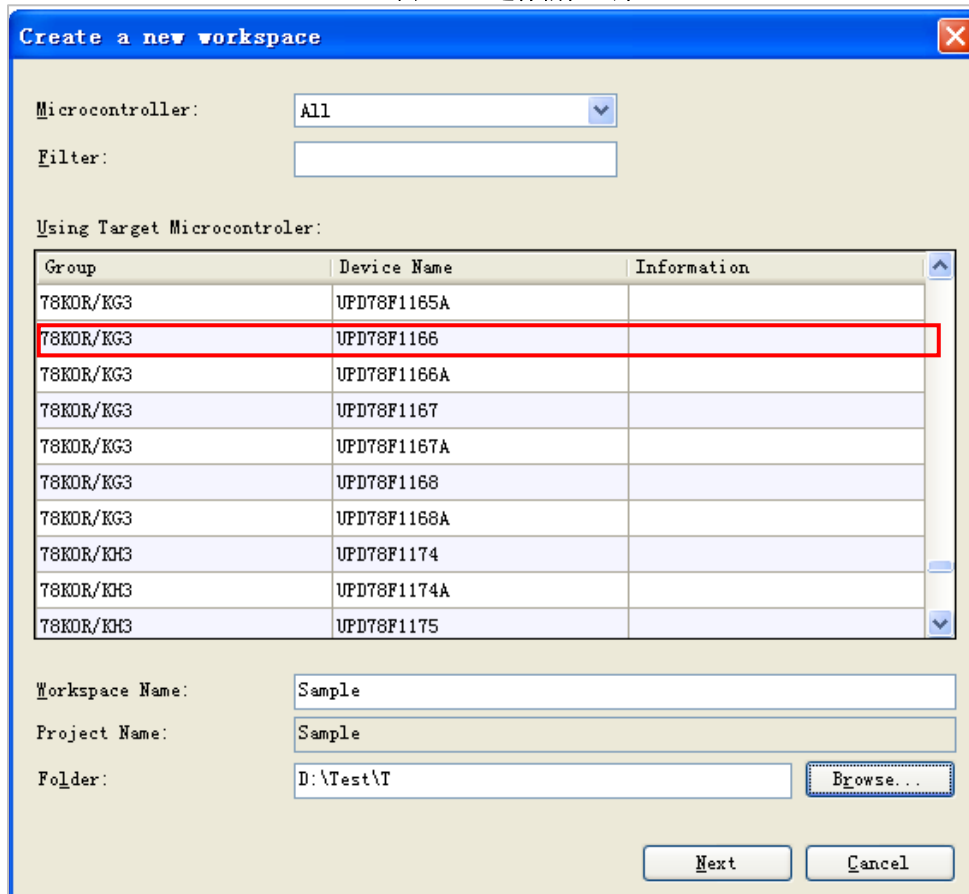
- <1> 点击【开始】按钮，指向[程序]，[Renesas Electronics CubeSuite+]，[Programming Tools]，[Renesas Flash Programmer Vx.xx]，点击[Renesas Flash Programmer Vx.xx [Basic mode]]，启动 RFP 软件。
- <2> 创建[ Create new workspace. ]，如图 4-11 所示

图 4-11 创建



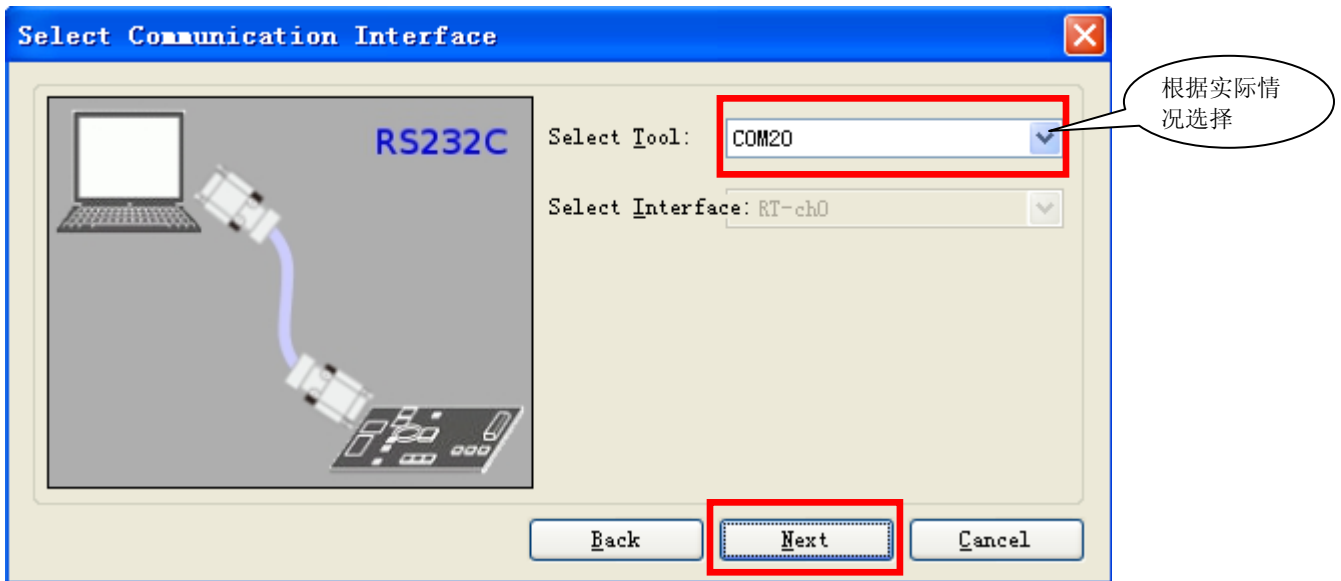
- <3> 选择编程芯片，如 78F1166。如图 4-12 所示。

图 4-12 选择编程芯片



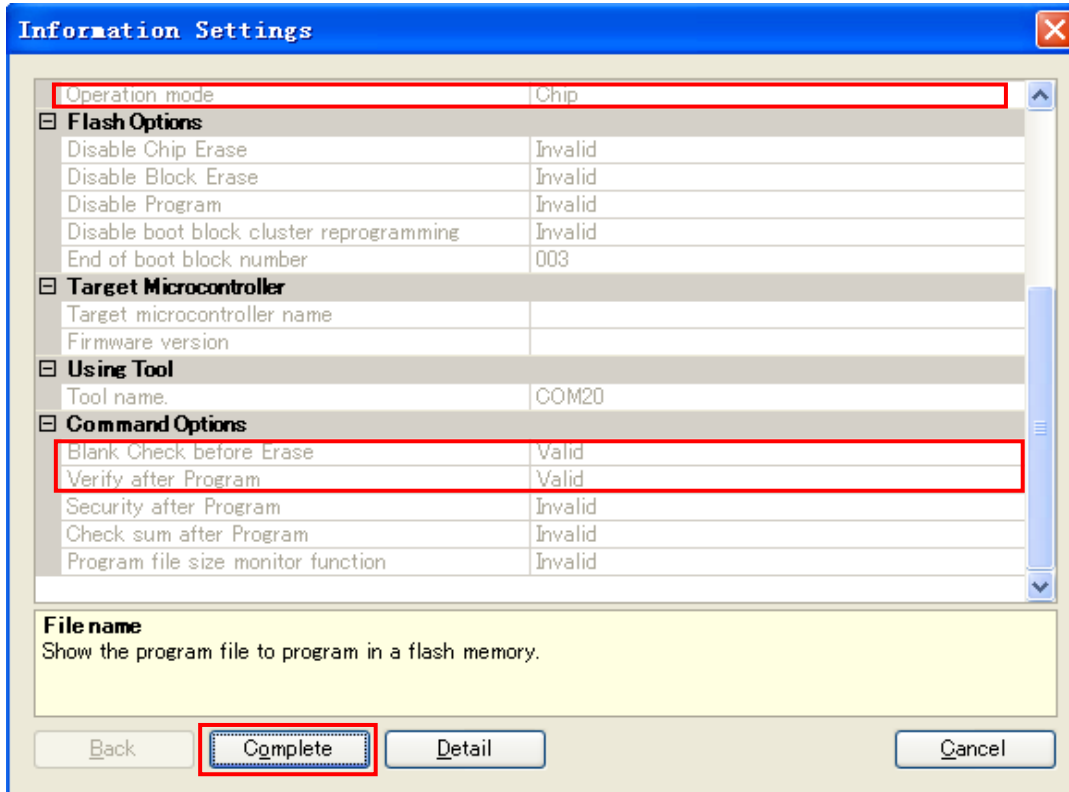
<4> 选择通信接口[ Select Communication Interface ], 如图 4-13 所示。

图4-13 [Select Communication Interface] 对话框



<5> 根据实际需要设置[Information Settings] 选项卡中的选项。详细内容，请参考 RFP 的用户手册。

图 4-14 [Information Settings] 选项卡



(2) 程序文件的选择

请按照下述的步骤<1>至<3>选择程序文件。

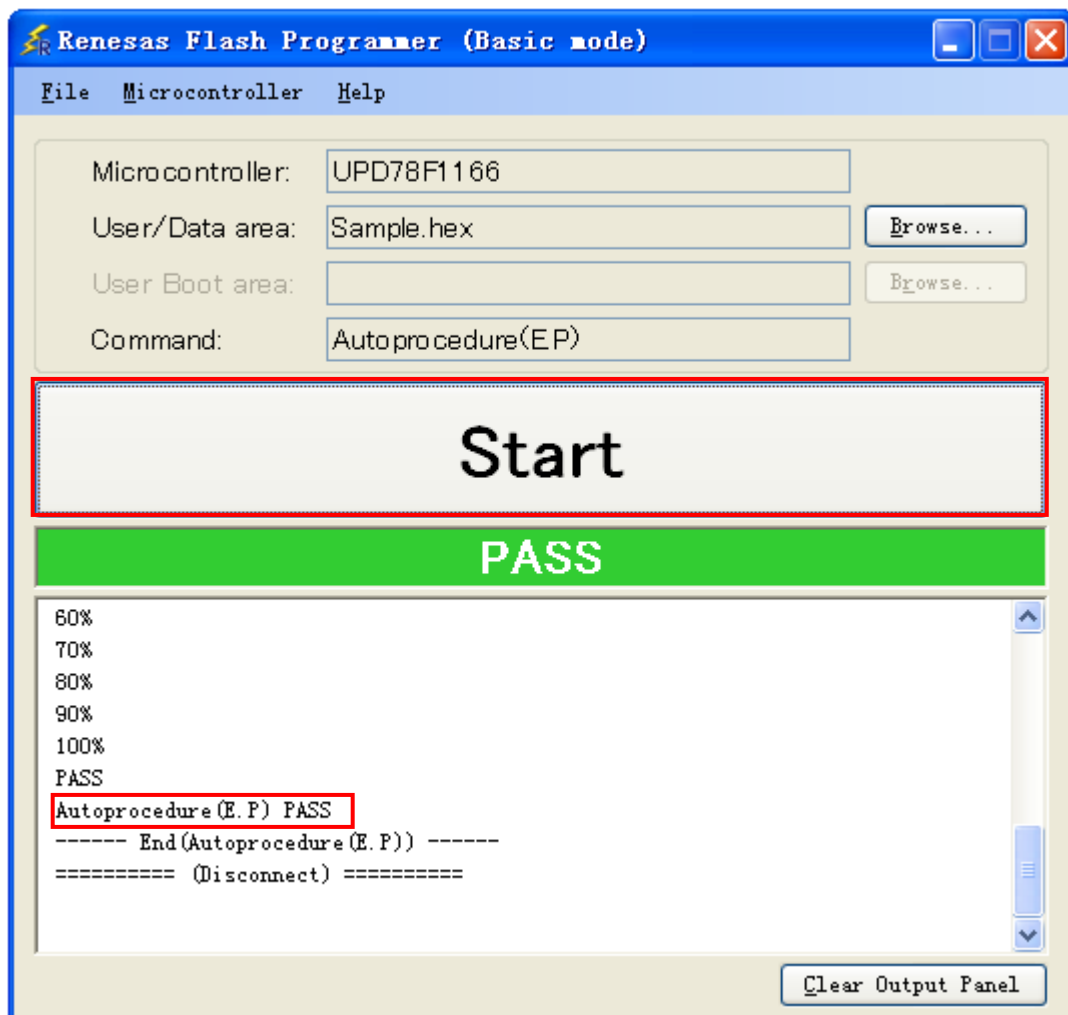
- <1> 在[User/Data area]区域点击 **Browse...**按钮。
- <2> 弹出程序文件选择对话框。
- <3> 选择要写入目标设备的程序文件，然后点击**打开**按钮。

(3) 执行[Autoprocedure(EP)]命令

- <1> 点击菜单栏的[Microcontroller]，然后选择[Autoprocedure(EP)]命令。
- <2> 点击 **Start** 按钮

[Blank Check]、[Erase]（如果目标设备上的Flash 存储器非空）、[Program]、[Verify]命令将会在目标设备上依次执行。当本命令正常执行完成后，弹出如下窗口。

图4-15 [Autoprocedure(EP)]命令执行完成之后



### 4.3.5 系统关闭顺序

按照下面的顺序来结束 Flash 编程并关闭系统。

如果未按照这个顺序操作，目标系统或者 EZ-CUBE 可能会被损坏。

**(1) 关闭 RFP**

如果不需要对其它设备进行编程，则请关闭 RFP 软件。

**(2) 关闭目标系统电源**

关闭目标系统的电源。如果电源选择开关设置在“5”，则此步骤可以略过。

**(3) 移除 USB 线缆**

从 EZ-CUBE 或者主机上移除 USB 线缆。

如果电源选择开关设置为“T”时，可以在此步骤之前先执行（4）移除目标线缆。

**(4) 移除目标线缆**

从 EZ-CUBE 或者目标系统上拔掉目标线缆。

### 4.3.6 Flash 编程的注意事项

本节描述了 Flash 编程的注意事项。请认真阅读，以保证 EZ-CUBE 的正确使用。

- 为了提高写入质量，请在使用 EZ-CUBE 之前对下列问题进行充分了解，并加以校验和评估。
  - 电路的设计要符合 EZ-CUBE 用户手册对设备的描述
  - 设备，RFP 和 EZ-CUBE 的使用都要遵循各自手册的描述
  - 提供给目标系统的电源要稳定

## 第 5 章 如何在 R8C 微控制器上使用 EZ-CUBE

本章描述了如何使用 EZ-CUBE 对 R8C 微控制器进行片上调试和 Flash 编程。

片上调试是指通过芯片内置的调试功能对已经装载在目标系统上的微控制器进行调试的方法。由于这种调试方法是通过板上的目标设备实行的，所以非常适合现场调试。

Flash 编程是将程序写入设备内置的 Flash 存储器中的方法。可以在板对设备进行擦除，写入和校验。

请先将 EZ-CUBE 的固件更新为用于 R8C，具体请参照 (1) - (3)。详细内容请参见 **1.5 固件更新**。

- (1) 使用 USB 缆连接 EZ-CUBE 和 PC。**不要将 EZ-CUBE 与目标板相连。**
- (2) 启动【QBEZUTL.exe】，并指定“R8C\_OCD\_FW.hex”固件文件。
- (3) 点击[Start]按钮，然后指定用于 R8C 固件下载到 EZ-CUBE。

如果首次用 R8C 微控制器作为目标设备来使用 EZ-CUBE，那么请仔细阅读以下内容。

### 5.1 目标系统设计

要用 EZ-CUBE 和目标系统进行通信，必须在目标系统上安装对应的通信电路。本节描述了相关电路的设计和连接插头的安装。

### 5.2 片上调试

本节描述了用 EZ-CUBE 进行片上调试的系统配置和启动方法。

### 5.3 Flash 编程

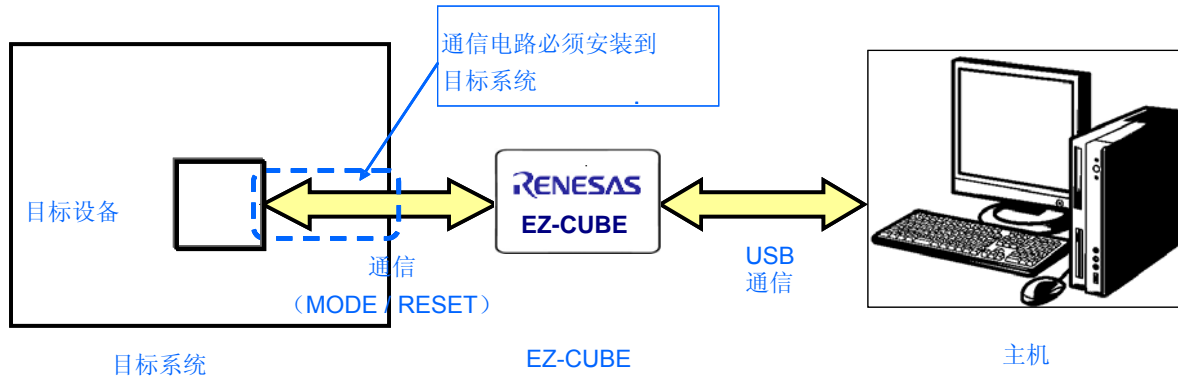
本节描述了用 EZ-CUBE 进行 Flash 编程的系统配置和启动方法。

## 5.1 目标系统设计

本节描述了片上调试和 Flash 编程所需目标系统的电路设计。

图 5-1 显示了 EZ-CUBE 通信的整体接口框图。如图所示，EZ-CUBE 和目标系统的目标设备之间进行串行通讯。为了通讯，目标系统上必须安装用于通信的电路。请参考本节的内容设计合适的电路。

图 5-1. 通信接口框图



### 5.1.1 引脚配置

本节描述了 EZ-CUBE 和目标系统之间使用的接口信号。表 5-1 列出了使用 8 引脚线缆时的引脚配置。表 5-2 描述了每个引脚的功能。在调试和编程的时候，引脚配置是相同的。

表 5-1. 引脚配置

| 引脚编号 | 引脚名称 <sup>注</sup> |
|------|-------------------|
| 1    | GND               |
| 2    | RESET_IN          |
| 3    | Target Vcc (5V)   |
| 4    | N.C               |
| 5    | N.C               |
| 6    | N.C               |
| 7    | N.C               |
| 8    | MODE              |

注 EZ-CUBE 内的信号名称。

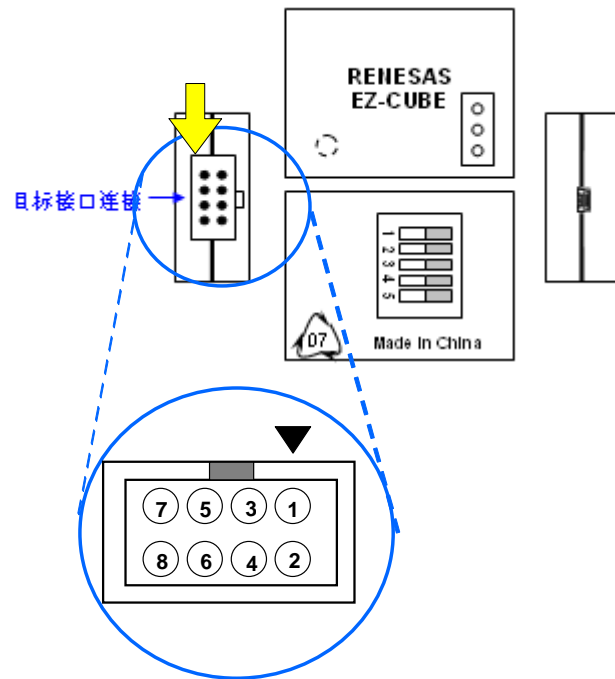


表 5-2. 引脚功能

| 引脚名称            | IN/OUT <sup>#</sup> | 描述                                   |
|-----------------|---------------------|--------------------------------------|
| GND             | P                   | 接地                                   |
| Target Vcc (5V) | P (OUT)             | 5V±0.3V 从 EZ-CUBE 仿真器 输出, 最大电流 100mA |
| MODE            | IN/OUT              | 双向通信, 仿真器与 MCU 的通信                   |
| RESET_IN        | IN/OUT              | 复位信号, 仿真器输出 / 从目标板输入                 |

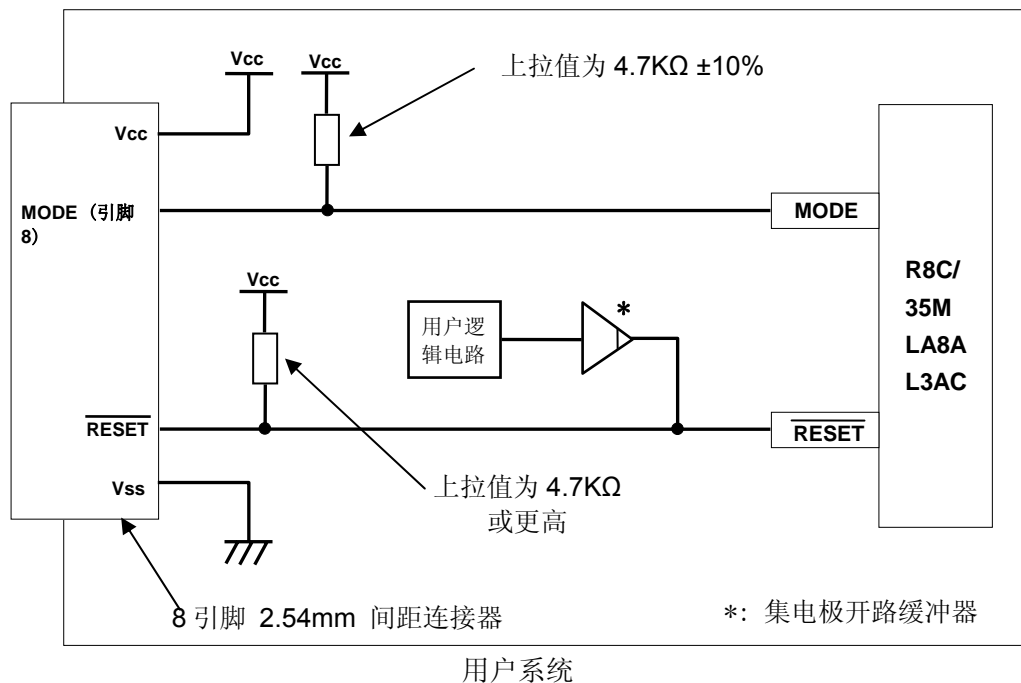
注 以 EZ-CUBE 为参照基准。

### 5.1.2 电路连接示例

图 5-2 给出了连接示例。在调试和编程的时候，连接是相同的。

**注意事项** 在电路连接示例中出现的常数值为参考值。如果进行 flash 编程旨在量产时，请彻底评估目标设备的规格是否满足要求。

图 5-2. 片上调试时 / Flash编程

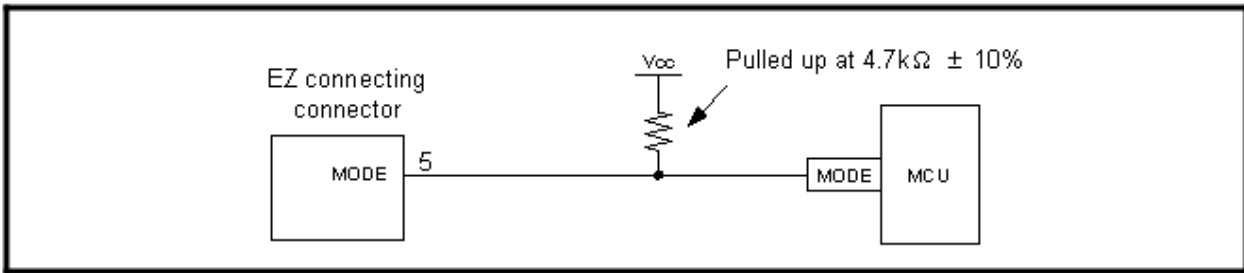


### 5.1.3 管脚连接

#### (1) MODE 管脚

EZ-CUBE 仿真器将 MODE 管脚作为 MCU 控制和强制中断控制。不要在此管脚连接电容等。

图 5-3. MODE 管脚和仿真器连接

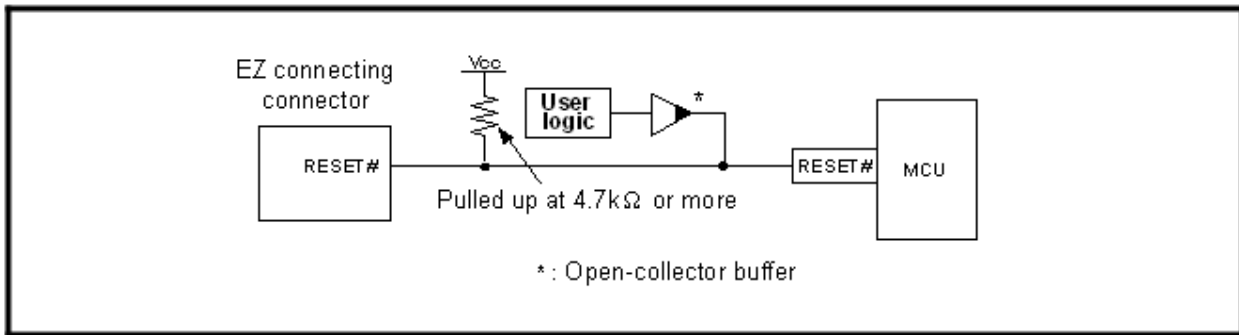


(2) RESET 管脚

RESET 管脚用于仿真器输出“L”并监视管脚状态。因此，用集电极开路输出缓冲器或 CR 复位电路作为用户系统的复位电路。推荐的上拉阻值是 4.7kΩ 或以上。

MCU 可以由仿真器输出“L”进行复位。然而，复位 IC 输出为“H”，则用户系统复位电路不能设为“L”。因此，仿真器不能正常操作。

图 5-4. MODE 管脚和仿真器连接



(3) 其他管脚

- 分别连接 Vss 和 Vcc 到 MCU 的 Vss 和 Vcc。
- Vcc 输入量必须在 MCU 指定范围之内。
- 对于 N.C 管脚不要连接任何内容。

**⚠ WARNING**

When supplying power, ensure that there are no shorts between Vcc and GND. Only connect the EZ emulator after confirming that there are no mismatches in pin assignments of the EZ connecting connector. Incorrect connection will result in the host computer, the emulator, and the user system emitting smoke or catching fire.

5.1.4 在目标系统上安装连接器

在设计目标系统时，必须安装 EZ-CUBE 和目标系统的连接器。可以选择 2.54mm 间距的 8 引脚通用型连接器。

除了专用的连接线以外，同时也支持散线连接。



## 5.2 片上调试

本节描述了使用 EZ-CUBE 进行片上调试的系统配置，启动/关闭顺序和调试中的若干注意事项。

### 5.2.1 调试功能

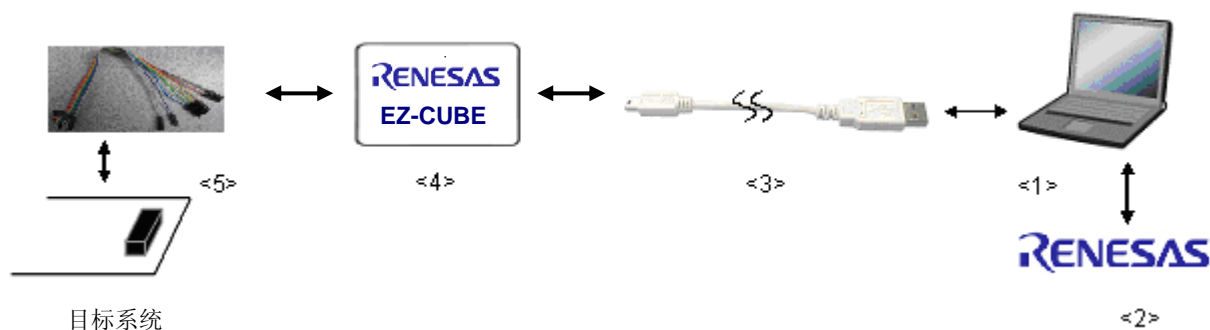
表 5-3 列出了以 R8C 微控制器为目标设备时的调试功能。

| 功能          | 规格   |
|-------------|--|
| 目标 MCU      | R8C/Tiny 系列<br>R8C/ L3AM,35M,LA8A                              |
| 可用工作模式单片模式  | 单片模式   |
| 断点功能        | — 地址匹配断点, 8个点<br>— 数据断点, 2 个点<br>— PC 断点 (最多 255 个点)<br>— 强制中断 |
| 跟踪功能        | 4 个转移指令  |
| 用户接口        | 1 线时钟异步串行接口 (通过 MODE 引脚进行通信)                                   |
| 要使用的 MCU 资源 | — ROM 容量: 2 KB [*1]<br>— 堆栈 8 字节<br>— 地址匹配中断                   |
| 电源电压        | 2.7~5.5V   |

### 5.2.2 系统配置

图 5-3 显示片上调试的系统配置。

图 5-5. 片上调试的系统配置



<1> 主机

要求有 USB 接口

<2> 软件

包括 CubeSuite+ for China, USB 驱动等。

<3> USB 线缆 (附带)

<4> EZ-CUBE (本产品)

<5> 8 引脚目标线缆 (附带) 或 R8C 专用转接板 (另购)

### 5.2.3 系统启动顺序

本节描述了系统启动的顺序。请注意按次序操作。

#### (1) 软件的准备和安装

以下的软件是片上调试所必需的。关于软件的准备和安装的详细信息，请参考 EZ-CUBE 附带的安装手册。

- CubeSuite+
- USB 驱动

#### (2) 开关设置

SW-1 开关：请选择 "M2"

SW-2 开关：请选择 "Int. Clock"

SW-3 开关：请选择 " Debug Mode"

SW-4 开关：请根据实际目标设备使用选择

SW-5 开关：请选择 " R8C"

- 注意事项**
1. 在 USB 线缆处于连接状态时，请勿改变开关的设置。
  2. 最大允许电流 100mA，所以请勿将 EZ-CUBE 连接到电流很大的目标系统上。在 EZ-CUBE 与主机连通后，主机始终保持向 EZ-CUBE 供电。

#### (3) 连接目标系统

将 EZ-CUBE 连接到目标系统的连接，在开启目标系统电源之前建立连接。

#### (4) 连接 USB 线缆

请在目标系统上电之前，将 EZ-CUBE 连接到主机，模式灯点亮为红色。

#### (5) 目标系统上电

打开目标系统的电源。如果电源选择为 "5"，则这一步并不是必需的。

#### (6) 启动调试器

打开调试器。

此步骤及之后的操作，请参考 CubeSuite+的用户手册。

如果调试器无法正常开启或者操作不稳定，可能是由以下问题引发的。

- EZ-CUBE 和目标系统之间通信错误
  - 固件是否更新；
  - 开关是否选择正确；
  - 目标连线是否正确。
- 使用了不支持的软件（调试器，设备文件或者固件）
  - 所使用的软件可能不支持目标设备的调试。
- EZ-CUBE 损坏
  - EZ-CUBE 可能被损坏。

### 5.2.4 系统关闭顺序

停止调试并关闭系统时，请按照下列顺序操作。

如果没有按照下列顺序操作，目标系统或者 EZ-CUBE 可能被损坏。

**(1) 停止调试器**

停止调试器的运行并断开调试器连接。

**(2) 关闭目标系统电源**

关闭目标系统的电源。如果电源选择开关设置为“5”的位置，则无需本步操作。

**(3) 移除 USB 线缆**

从 EZ-CUBE 或主机上将 USB 线缆拔下。

**(4) 移除目标线缆**

从 EZ-CUBE 或目标系统上将目标线缆拔下。

### 5.2.5 调试的注意事项

#### 5.2.5.1 仿真器占用的 MCU 资源

**(1) 仿真器的程序区**

表 5-4 列出了仿真器占用的程序区。不要更改此区域分配，否则仿真器将无法控制 MCU。

表 5-4. 仿真器的程序区

| Group    | Part No.  | Internal ROM Size |                    | Program Area for the Emulator <sup>Note 1</sup>  |  |
|----------|-----------|-------------------|--------------------|--|--|
|          |           | Program ROM       | Data Flash         | Vector Area  | ROM Area                               |
| R8C/35M  | R5F21354M | 16 KB             | 1 KB<br>(4 blocks) | FFE4h - FFE7h,<br>FFE8h - FFEBh,<br>FFEC h - FFEFh,<br>FFF4h - FFF7h,<br>FFFCh - FFFEh | -                                      |
|          | R5F21355M | 24 KB             |                    |  | -                                      |
|          | R5F21356M | 32 KB             |                    |  | 2 KB of the ROM area <sup>Note 2</sup> |
|          | R5F21357M | 48 KB             |                    |  | -                                      |
|          | R5F21358M | 64 KB             |                    |  | -                                      |
|          | R5F2135AM | 96 KB             |                    |  | -                                      |
|          | R5F2135CM | 128 KB            |                    |  | 2 KB of the ROM area <sup>Note 2</sup> |
| R8C/L3AM | R5F2L3A7M | 48 KB             | 1 KB<br>(2 blocks) | FFE4h - FFE7h,<br>FFE8h - FFEBh,<br>FFEC h - FFEFh,<br>FFF4h - FFF7h,<br>FFFCh - FFFEh | -                                      |
|          | R5F2L3A8M | 64 KB             |                    |  | -                                      |
|          | R5F2L3AAM | 96 KB             |                    |  | -                                      |
|          | R5F2L3ACM | 128 KB            |                    |  | 2 KB of the ROM area <sup>Note 2</sup> |
| R8C/LA8A | R5F2LA84A | 16 KB             | 1 KB<br>(2 blocks) | FFE4h - FFE7h,<br>FFE8h - FFEBh,<br>FFEC h - FFEFh,<br>FFF4h - FFF7h,<br>FFFCh - FFFEh | -                                      |
|          | R5F2LA86A | 32 KB             |                    |  | -                                      |
|          | R5F2LA87A | 48 KB             |                    |  | -                                      |
|          | R5F2LA88A | 64 KB             |                    |  | 2 KB of the ROM area <sup>Note 2</sup> |

注

1. 仿真器需要重写 OFS, OFS2 和 ID code。
2. 启动调试器，在调试工具设置处点击右键选择 [R8C EZ-CUBE Emulator (Debugging Tool)]。指定将不会在用户系列中使用的区域。

**(2) EZ-CUBE 仿真器使用的管脚**

仿真器控制 MCU 需使用以下管脚。

- RESET# 管脚和 MODE 管脚

**(3) Interrupts (不可用)**

EZ-CUBE 仿真器使用 BRK 指令中断，地址匹配中断，single-step 中断和地址中断。

因此，用户程序不能使用这些中断。仿真器会改变这些中断向量值用于自身使用。如果在用户程序中编写中断向量值不会发生问题。这些向量地址不会在[Memory]面板中被重写。

**(4) EZ-CUBE 仿真器使用的堆栈区**

仿真器在用户程序中中断期间使用多达 8 个字节的堆栈指针。因此，需为堆栈区预留 8 个字节。

**(5) EZ-CUBE 仿真器使用的 SFRs**

表 5-5 列出的 SFRs 由 EZ-CUBE 仿真器程序使用。

- 其他用户程序不要改变[Memory]面板上的这些值。如果引用寄存器内容，将读出在 EZ-CUBE 仿真器程序中设置的值。

**表 5-5 EZ-CUBE 仿真器使用的 SFRs (1)**

| Address | Register   | Symbol | Bit                         |
|---------|--|--------|-----------------------------|
| 000Ah   | Protect register                                 | PRCR   | Bit 0                       |
| 0023h   | High-speed on-chip oscillator control register 0 | FRA0   | Bit 0 <small>Note 1</small> |
| 0080h   | DTC Activation Control Register                  | DTCTL  | Bit 0 <small>Note 2</small> |

表 5-6 列出的 SFRs 由 EZ-CUBE 仿真器程序使用。不要更改其中的任何值，否则 EZ-CUBE 将无法控制 MCU。

**表 5-6 EZ-CUBE 仿真器使用的 SFRs (2)**

| Address       | Register                                  | Symbol | Bit                         |
|---------------|---|--------|-----------------------------|
| 0004h         | Processor Mode Register 0                 | PM0    | Bit 0 <small>Note 3</small> |
| 01C0h - 01C2h | Address match interrupt register 0        | RMAD0  | All bits                    |
| 01C3h         | Address match interrupt enable register 0 | AIER0  | All bits                    |
| 01C4h - 01C6h | Address match interrupt register 1        | RMAD1  | All bits                    |
| 01C7h         | Address match interrupt enable register 1 | AIER1  | All bits                    |

**注** 1. EZ-CUBE 的仿真器调试时，即使高速片上振荡器的选项始能位有效和 FRA00 可以设置为“高速片上振荡器关闭”，高速片上振荡器也不会停止。检查低功耗，高速片上振荡器关闭，评估最终产品或系统等功，仅在仿真器断开并且用户程序写入 MCU。

如果高速片上振荡器的频率或频率控制器需要更改，请在用户程序中进行更改。如果在[Memory]面板更改，EZ-CUBE 将变为不可控。

- 2. 不适用 R8C/LA8A。
- 3. 仅适用 R8C/L3AM。当寄存器用于 EZ-CUBE 仿真时，此位始终为“1”。

**(6) 选项功能选择区域**

EZ-CUBE 设置选项功能寄存器(OFS: 0FFFh)的位 0 和位 7 为 1b。虽然这些地址可以改写，并且能在 [Memory]面板更改值，位 0 的更改值是无效的。

- b0: 看门狗定时器启动选择位 1: 复位后看门狗定时器停止
- b7: 复位后计数源保护模式选择位 1: 复位后计数源保护模式无效

此外，仿真器设置选项功能选择寄存器 2(OFS2: 0FFDBh)低 4 位为 1111b。

- b1, b0: 看门狗定时器下溢期间设置位 11: 3FFFh
- b3, b2: 看门狗定时器刷新确认期间设置位 11: 100%

看门狗定时器:

用户程序 halt 期间, EZ-CUBE 仿真器程序将在程序操作过程中刷新看门狗定时器。如果用户程序使用看门狗定时器, 用户程序 halt 期间看门狗定时器将由 EZ-CUBE 仿真器程序刷新, 此刷新时间与实际的操作定时不同。

此外, 请注意, 用户程序执行期间不会刷新看门狗定时器。

计数源保护模式:

EZ-CUBE 仿真器不能调试计数源保护模式。

### (7) EZ-CUBE 仿真器初始化寄存器

系统启动后, 仿真器会初始化寄存器, 如表 5-7 所示。

表 5-7 EZ-CUBE 仿真器寄存器初始值

| Status                 | Register             | Initial Value                                     |
|------------------------|----------------------|---|
| EZ Emulator Activation | PC                   | Reset vector value in the vector address table    |
|                        | R0 to R3 (bank 0, 1) | 0000h   |
|                        | A0, A1 (bank 0, 1)   | 0000h   |
|                        | FB (bank 0, 1)       | 0000h   |
|                        | INTB                 | 0000h   |
|                        | USP                  | 0000h   |
|                        | ISP                  | 05FFh (differs from the specification of the MCU) |
|                        | SB                   | 0000h   |
|                        | FLG                  | 0000h   |

### (8) RAM 初始化

注意 EZ-CUBE 仿真器初始化部分 MCU 内部 RAM 区域(00400h-004FFh) 启动时为"00h"。

### (9) MCU 保留区

在 MCU 硬件手册没有定义的地址为保留区域。不要变更保留区的内容。

如果变更, EZ-CUBE 将无法控制 MCU。

- 在[Memory]面板此区域的值不确定。
- 此区域, 在[Memory]面板的搜索、比较和移动功能失常。

### (10) 用户程序 halt 期间 DTC (不包括 R8C/LA8A group)

当用户程序 halts 时, 禁止使用 DTC 进行数据传送。

然而, DTC 中断请求不会自己停止。因此, 当用户程序重启时, 用户程序 halt 期间会产生 DTC 中断请求。

### (11) 内部低功耗注意事项

确保电压检测寄存器 2(VCA2)的位 0 设置为 0: 低功耗禁止。

如果选择为"1", 仿真器则无法控制 MCU。

### (12) 调试功能以降低功耗

当调试低功耗读取模式时, 禁止进行窗口操作, 直到模式取消后程序在断点处停止。

调试功能以停止 Flash 存储器时, 不要操作窗口, 直到功能取消后程序在断点处停止。

### (13) 访问 SFR 区域的注意事项

可以在[Memory]或[IO]面板引用或设置 SFR 区域内容。请注意以下内容。

-当访问特殊寄存器, 在用户程序停止期间\*, 您不能存取一些特殊寄存器  
这些寄存器包括:

- 存取禁止地址

- 存取顺序从高字节到低字节指定的寄存器
- 只由特殊指令访问的寄存器
- 总线宽度与[Memory]面板设置的总线宽度不匹配的寄存器
- 按条件访问的寄存器（其中之一是，fOCO-F 必须比 CPU 时钟更快存取寄存器）

注 按照目标MCU硬件手册访问SFR区域

### 5.2.5.2 Reset

#### (1) Reset 功能

不能使用上电复位和电压监视器 0。如果执行了任一复位，EZ-CUBE 仿真器将不可控。当复位管脚为“L”时，不要停止用户程序。将产生超时错误。

#### (2) Reset 向量地址

EZ-CUBE 仿真器调试期间，仿真器将使用复位向量地址。用户程序停止，在下载程序或 [Memory]面板设置的值将被引用。但是，用户程序运行时，将显示仿真器程序设置的值。如果用户程序停止设置复位向量地址的值，仅仿真器内部缓存被更新。

#### (3) Reset 后

如果当用户程序运行时MCU复位，在用户程序重启前EZ-CUBE仿真器程序将临时进入（如表 5-8）。所以，有时间延迟直到用户程序复位重启。因此，请注意硬件复位（RESET#）与其他复位表现不同，如表所示<sup>1</sup>。

表 5-8 用户程序执行期间 MCU 复位并重启它的表现

| Reset   | Behavior when the MCU is reset   |
|---|--|
| Hardware reset (RESET#)                             | (1) A hardware reset (RESET#) is detected by the EZ emulator.                                |
|   | (2) The emulator is reconnected to the MCU.  |
|   | (3) The MCU is reset and the EZ emulator program is entered into.                            |
|   | (4) The address jumps to the reset vector address which is retained in the emulator program. |
| Watchdog timer reset or S/W reset <sup>Note 2</sup> | (1) The MCU is reset and the EZ emulator program is entered into.                            |
|   | (2) The address jumps to the reset vector address which is retained in the emulator program. |

- 注
1. 可以执行用户程序中设置或删除硬件断点（中断前），但复位将用户程序执行中设置的硬件断点（中断前）初始化。
  2. 如果在执行用户程序看门狗定时器复位或 S/W 复位，因为 SFR 的初始化,硬件中断（中断前）（通道 0 到通道 3）被禁用(不发生中断)。

#### (4) 看门狗定时器 reset

看门狗定时器持续计数直到用户程序停止，包括仿真器程序计数刷新。

因此，如果用户程序中断后立即发生下溢或其他异常状况，看门狗定时器刷新前引起其复位，仿真器可能会变得无法控制。

#### (5) Reset 相关限制

用户程序停止时不执行硬件复位（RESET#）。EZ-CUBE 仿真器将无法控制 MCU。请注意，如果用户程序停止(Halt)时从 MCU 引脚复位，CPU 寄存器的内容将没有初始化，并且他们的值作为时间用户程序停止或设置在 CPU 寄存器面板的值为优先考虑。

#### (6) 冲突

如果复位(硬件复位(RESET#), 看门狗定时器复位等)和 EZ-CUBE 仿真器操作(在[Memory]面板存储器存取等)发生冲突时, EZ-CUBE 仿真器运行将不可控。将显示例如超时错误的错误信息。

此外, 一个消息对话框将询问是否发布系统复位或不显示。点击[Yes]按钮将停止用户程序并初始化仿真器。点击[No]按钮则既不初始化仿真器也不会停止用户程序。点击任一按钮, 您都可以继续调试。

**警告** 如果激活在[Memory]或[Watch]面板存储器自动更新, 不要执行 MCU 硬件复位。否则, 将丢失 MCU 通信并且 EZ-CUBE 仿真器运行将不可控。

### 5.2.5.3 内部ROM区域 (Flash存储器)

#### (1)更改内部 ROM 区域

当更改 MCU 内部 ROM 内容, 没有从调试器下载, 仅 EZ-CUBE 仿真器内部缓存更新, 并且更改的内容在用户程序启动运行前立刻映射到 MCU。

#### (2) CPU 改写模式调试的注意事项

##### (a) CPU 改写模式的不可重写区域

当调试 CPU 改写模式, 不要改写以下 CPU 内部 ROM 区域。如果这些区域被改写, EZ-CUBE 仿真器将无法控制 MCU。CPU 改写仅适用于数据区域。

- 固定的中断向量区
- 仿真器程序区
- 程序 ROM 区的 Block 3 (如下表所示, 不包括 R8C/LA8A。)

| MCUs               | Applicable areas       |
|--------------------|------------------------|
| 4 KB ROM version   | -                      |
| 8 KB ROM version   | -                      |
| 16 KB ROM version  | -                      |
| 24 KB ROM version  | Block 3: 0A000h-0BFFFh |
| 32 KB ROM version  | -                      |
| 48 KB ROM version  | Block 3: 08000h-0BFFFh |
| 64 KB ROM version  | Block 3: 08000h-0BFFFh |
| 96 KB ROM version  | -                      |
| 128 KB ROM version | -                      |

##### (b) CPU 改写模式的操作

- 当调试 CPU 改写模式时, 不能停止用户程序而 CPU 改写模式以启动或在擦除挂起状态。并且不能执行单步指令启动 CPU 改写模式或进入擦除挂起状态。如果停止, EZ-CUBE 仿真器将无法控制 MCU。此外, 运行程序前禁止在[Watch]面板自动更新或[Memory]面板固定显示, 所以执行期间不会发生存储器访问。

-在 CPU 改写模式下重写内部 ROM 后, 检查内部 ROM 区, CPU 改写模式停用和引用[Memory]面板等后停止用户程序。如果执行 CPU 改写可用于数据 flash 区执行, 和擦除过程可以暂停, 不要使用 SW 中断。

- 当在用户区重写 flash 存储器, 启动调试器。在[Project Tree]面板右击[R8C EZ -CUBE Emulator (Debugging Tool)], 选择 R8C EZ-CUBE 仿真器的[Property]。选择[Debug Tool Setting]选项卡, 位于[System]下的"Debugging the program re-writing the internal flash"。然后选择"Yes"。

- 警告**
1. 当选择[Debugging the program re-writing the internal flash.]，不要在[Memory]面板等改写内部 ROM 区。同样，注意当选择[Debugging the program re-writing the internal flash.]不能使用 S/W 中断。
  2. 当在 CPU 改写模式时调试程序，可以使用存储器引用或修改功能。然而，不能在以下条件使用这些功能。
    - 写指令正在执行的寄存器，需要连续写入（如 FMR13 位）。如果写指令被中断的存储器引用或修改处理，MCU 不承认连续写入执行。

**(3) EZ-CUBE 仿真器改写 flash 存储器的注意事项**

EZ-CUBE 仿真器写入内部 ROM (flash 存储器)时，MCU 不能复位或执行调试操作。如果 MCU 复位或重写 flash 存储器调试，用户程序或 EZ-CUBE 仿真器程序将可能被打乱。

发生 Flash 存储器改写：

- 当下载用户程序
- 在内部 ROM 区设置软件断点和执行用户操作后
- 在内部 ROM 区删除软件断点和执行用户操作后
- 在[Memory]面板改写内部 ROM 区值和执行用户操作后

**(4) 执行用户程序期间 Flash 存储器**

当用户程序运行时，内部 ROM 区不能被用户程序以外(在存储器面板等)的更改。

**(5) 调试 MCUs 的使用**

当调试时，Flash 存储器频繁被 EZ-CUBE 仿真器重写。因此，不要在产品中使用用于调试的 MCU。此外，当 EZ-CUBE 程序被写入 MCU 调试时，不保存用于调试的 MCUFlash 存储器内容，也不使用产品的 ROM 数据。

**(6) Flash 存储器 ID 码**

此 MCU 功能可防止除用户之外的任何人读取 Flash 存储器。

写入到 MCU 的 ID 码区的 ID 码（参加表 5-9）必须与[Connect Settings]选项卡输入的"ID code"匹配，否则无法启动调试器。注意，当 ID 码为 FFh, FFh, FFh, FFh, FFh, FFh, FFh, 将视为未定义 ID 码。此时，ID 码将自动验证。

写入 ID 码区的值依据模式不同而不同。

- [Debugging mode] : FFh, FFh, FFh, FFh, FFh, FFh, FFh (无论下载用户程序的内容)

**注意事项** 如果写入 MCU 的 ID 码区的 ID 码与一个输入在[Flash]的[ID Code]的匹配，EZ-CUBE 仿真器写入 FFh, FFh, FFh, FFh, FFh, FFh, FFh 到 ID 码区。

表 5-9 ID 码存储区

| Address | Description             |
|---------|-------------------------|
| FFDFh   | First byte of ID code   |
| FFE3h   | Second byte of ID code  |
| FFE3h   | Third byte of ID code   |
| FFE3h   | Fourth byte of ID code  |
| FFF3h   | Fifth byte of ID code   |
| FFF7h   | Sixth byte of ID code   |
| FFF7h   | Seventh byte of ID code |



#### 5.2.5.4 在用户程序 halt 的操作

##### (1) 用户程序 halt 操作时钟

当用户程序停止，仿真器将 cpu 时钟转换为操作用内部高速振荡器时钟。然而，外设操作由用户程序指定的时钟。

##### (2) 用户程序 halt 期间外设 I/O

用户程序 halt 期间，因为仿真器禁止中断，可匹配中断请求不被接受。然而，外设 I/O 仍然运行，中断请求接受后立即启动用户程序。

例如，当用户程序定时器启动后再被中断停止，虽然定时器仍然在计数，但定时器中断不被接受。

##### (3) 用户程序 halt 期间[CPU Register] 面板和[Memory] 面板

如果 CPU 寄存器的值被[CPU Register]面板更改或 flash 存储器内容被[Memory]面板更改，而用户程序停止时，只反映仿真器内部缓存的值。更改寄存器的值或存储器内容立即反映在 MCU 用户程序启动前。

#### 5.2.5.5 程序的最终评价

在进入批量生产阶段前，确保进行不连接 EZ-CUBE 仿真器，单独执行程序的最终评价。

#### 5.2.5.6 调试功能

##### (1) 存储器存取

校验和

向量区，ID 码和 OFS，OFS2 区的值由仿真器重写，要知道他们的校验和不匹配。向量地址复位，尽管下载的值在[Memory]面板等上显示，要知道他们是仿真器缓存内部值，在实际 flash 存储器中设置的值是仿真器指定的。

##### (2) 软件中断功能

###### (a) 设置 inhibited 区

软件中断不能被指定在 MCU 内部 RAM 和内部 ROM 区外。

###### (b) 软件断点的注意事项

当您在[Editor]面板更改程序并且下载程序到 MCU，取决于改变的内容，您设置的软件断点地址可能不予纠正正常。在这种情况下，设置的软件断点的位置可能移动。

在程序下载后，请再次检查设置软件中断的地址。

###### (c) 用户程序执行期间存储器

放置 S/W 断点地址的指令在[Memory]面板显示的不是“BRK”指令(00h)。

###### (d) 从设置的软件断点地址的用户程序执行

当用户程序从设定的软件断点地址执行，用户程序运行前有关的地址(即用户程序开始地址)强制执行。因此，关于执行时间和中断，与没有设置软件断点前相比有所不同。

#### (3) 单步执行

##### (a) 单步执行的限制

在单步执行期间，软件中断和片上中断无效。

**(b) 强制执行**

强制执行使用一个跳转指令（即不是一个子程序返回指令）返回到调用源是不可能的。

**(c) 使用存储器自动更新的注意事项**

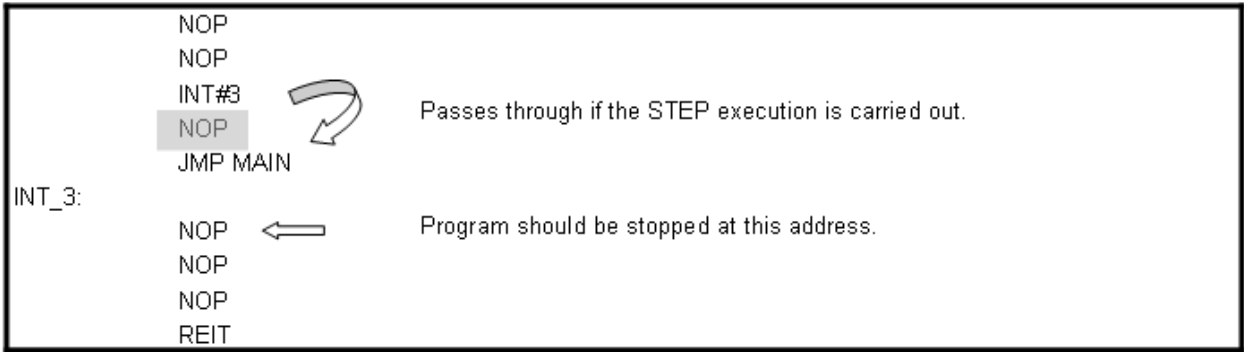
当在[Memory]和[Watch]面板的自动更新有效时，不执行单步或多步。否则，会需要很长时间来更新存储器数据和操作延时。

**(d) 执行特殊的单步**

i) 软件中断指令

程序中内部处理指令(undefined, overflow, BRK 和 INT)产生一个持续的软件中断时，单步运行不能执行(参见图 5-6)。

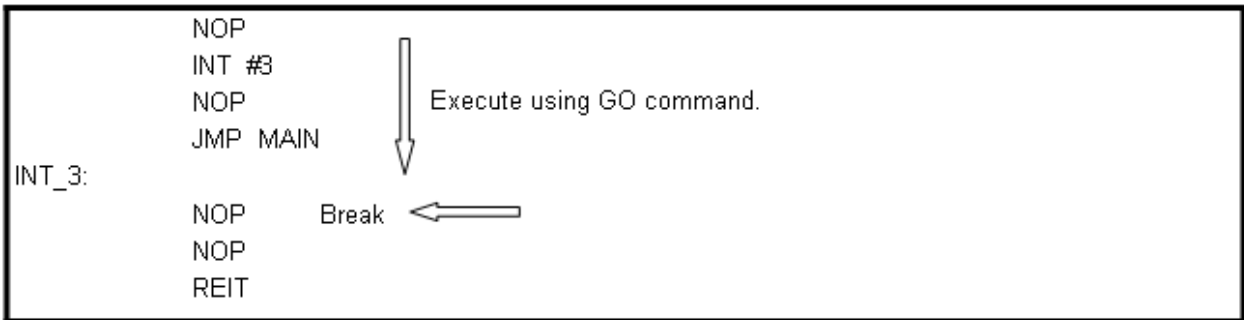
图 5-6 软件中断指令举例



ii) INT 指令

要使用 INT 指令调试用户程序，应当为 INT 指令的内部处理设置一个软件断点，并用 GO 命令执行程序(参见图 5-7)。

图 5-7 INT 指令示例



iii) 其他：标志控制指令

以下指令，单步时，仅操作在 EZ-CUBE 仿真器的标志，没有 MCU 的操作。因此，当执行这些指令时，要知道 Start/Stop 功能不起作用。

```

LDC src, FLG
STC FLG, dest
LDINTB src
    
```

**(1) 其他调试功能****(a) "Go to Here" 功能**

"Go to Here" 功能是使用硬件断点（中断前）。当执行"Go to Here"命令，所有硬件断点（中断前）和片上中断无效。

**<限制 1>**

从程序在执行[Go To Here]功能后停止的地址进行单步运行，并且地址中存在一个硬件断点，单步运行可能不能执行（PC 将不提示）。

执行单步前取消硬件断点。

除[Go To Here]功能以外，在从程序停止的地址开始进行单步执行之前，必须取消硬件断点。这些情况包括以下两个功能的使用。

- 在[Execute to the specified symbol after CPU Reset]属性选择[Yes]时，执行 CPU 复位。
- [Step Over]

如下所示，如果一个硬件断点被设置到地址在一个 JSR 指令（或 JSRI, INT, INTO, UND）之后，step over 指令将被执行，后面的单步执行失败。

|         |  |
|---------|--|
| JSR SUB | <-- (a): "JSR" 可以是 JSR, JSRI, INT, INTO,或 UND 中的任一指令 |
| nop     | <-- (b): "nop" 可以为除上述以外的任意指令                         |

**<限制 2>**

如果一个硬件中断被设为 RTS 指令（子程序返回指令），并且[Return Out]功能被用在子程序，程序可能在 RTS 指令时终止。

使用[Return Out]时，不能把硬件中断设为 RTS 指令。

**(b) 在停止模式或等待模式时调试**

在停止模式或等待模式时调试，不要操作窗口直到程序停止在程序行中设定的断点，停止模式或等待模式取消后程序将继续执行。

不能通过单步执行指令，切换到停止模式或等待模式过程中。

同时，在停止模式或等待模式时，如果存储器内容被参考或改变，在退出停止或等待模式，存取 MCU 存储器后，程序从设置在停止等待模式中的指令的下一个指令开始重新启动。如果在存储器存取中程序进入停止等待模式，存储器内容的参考或改变有可能发生异常。

因此，在程序运行前，不可以自动更新在[Memory]面板或[Watch] 面板或确定[Memory]面板的显示。同时，不要进行刷新操作，否则，用户程序执行中，存储器不能存取。

当程序在停止等待模式时被强制停止，这些模式将被取消。

**(c) 用户系统的看门狗定时器**

如果用户系统的重置电路中有看门狗定时器，在使用仿真器时要禁用。

**(d) 硬件断点改变程序执行**

在程序执行中，不要为了重置目标板管脚而设置或删除硬件中断（中断前）。如果程序执行过程中引脚复位，设置的硬件断点不会发生，会删除硬件中断。

如果在程序执行中管脚被重置，新设置的硬件断点将不发生，被删除的硬件断点将被执行。

如果在程序执行中管脚没有重置，硬件断点将按照设定发生。

一旦程序在设定或删除硬件断点后停止，即使管脚被重置，硬件断点也按照设置发生。

#### 5.2.5.7 通讯波特率

- 当您选择[Data flash area]调试监视器的位置，使用 250000 bps 或更低的通信波特率。

- 57600bps 或更低的波特率被指定为检查的目的的情况下，仿真器连接失败。

这么低的波特率，它需要较长的时间写入到目标 MCU 的 flash 存储器，并且仿真器调试可能会没有反应。

并且，注意如果进行 1024 字节或更大的数据处理，显示存储器的内容或存储器填充功能，因为占用太多的时间，可能会发生超时错误。

## 5.3 Flash 编程

本节描述了使用 EZ-CUBE 对 R8C 微控制器进行 Flash 编程时的系统配置和启动/关闭顺序。

### 5.3.1 编程功能的规格定义

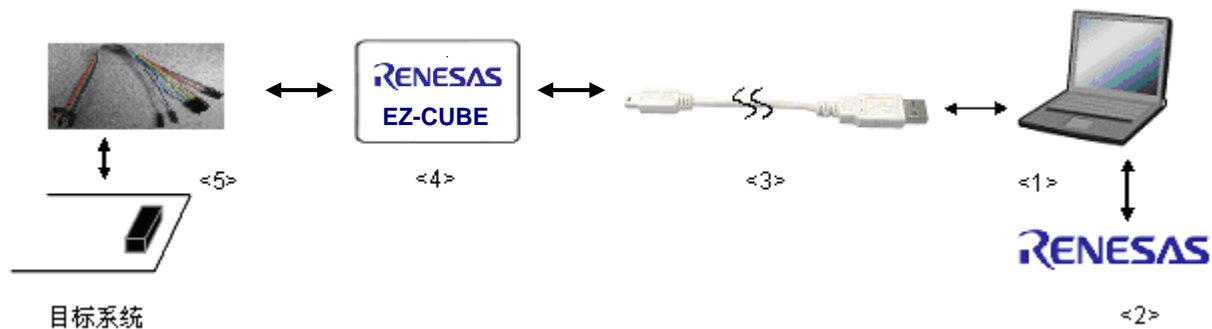
表 5-10. 编程功能的规格定义

| 功能       | 规格定义说明                                |
|----------|---------------------------------------|
| 主机接口     | USB2.0                                |
| 目标接口     | UART（单线模式）                            |
| 目标系统电压   | 2.7到5.5 V（根据目标设备而定）                   |
| 时钟提供     | 使用内部高速振荡器                             |
| 电源       | 5 V ±0.3V（最大电流：100mA）                 |
| 获取设备特定信息 | 使用RENESAS ELECTRONICS 的R8C参数文件（*.PRR） |
| 安全标志设置   | 支持                                    |
| 脱机操作     | 不支持（必须连接主机）                           |

### 5.3.2 系统配置

图 5-8 说明了 Flash 编程的系统配置。

图 5-8. Flash 编程的系统配置



<1> 主机

带有 USB 接口

<2> 软件

包括 WriteEZ5, USB 驱动, 参数文件等。

<3> USB 线缆（附带）

<4> EZ-CUBE（本产品）

<5> 8 引脚目标线缆（附带）或 R8C 专用转接板（另购）

### 5.3.3 系统启动顺序

本节描述了系统启动顺序。请注意先后次序。

#### (1) 软件的准备和安装

要进行 Flash 编程，以下软件都是必需的。有关软件的准备和安装详细信息，请参阅 EZ-CUBE 附带的配置手册。

- WriteEZ5
- USB 驱动
- 参数文件

#### (2) 开关设置

SW-1 开关：请选择 "M2"

SW-2 开关：请选择 "Int. Clock"

SW-3 开关：请选择 " Debug Mode"

SW-4 开关：请根据实际目标设备使用选择

SW-5 开关：请选择 " R8C"

**注意事项** 在 USB 线缆处于连接状态时，请勿改变开关的设置。

#### (3) 连接目标系统

在目标系统上电之前，将 EZ-CUBE 连接到目标系统。

#### (4) 连接 USB 线缆

在目标系统上电之前，将 EZ-CUBE 连接到主机。

#### (5) 目标系统上电

打开目标系统的电源。如果电源选择开关设置为 "5"，则这一步不是必需的。

#### (6) 启动 WriteEZ5

打开 WriteEZ5 软件。

### 5.3.4 应用样例

本节讲述了使用WriteEZ5 的一系列基本操作，用R5F21356M作一个样例。说明执行[Autoprocedure (EPV)] 命令为目标设备进行编程。命令的详细内容，请参考WriteEZ5 的用户手册。

本节系列操作所需要的条件如下。

#### <目标系统>

目标设备: R5F21356M  
电源电压: 5 V±0.3V (EZ-CUBE 提供)

#### <EZ-CUBE>

开关: SW-1 开关: M2  
SW-2 开关: Int. Clock  
SW-3 开关: Debug Mode  
SW-4 开关: 5  
SW-5 开关: R8C

#### <WriteEZ5>

参数文件: R5F21356M.prr  
时钟设置: 无 (使用内部高速振荡器)  
操作模式: chip  
待写文件: sample.hex  
命令参数: 选中[Blank check before Erase]  
选中[Read Verify after Program]

(1) 设置编程环境

设置编程环境，请按照下面描述的步骤<1>至<6>顺序进行。

<1> 点击菜单栏的[Device]，然后再点击[Setup...]。

<2> 出现 Device Setup 对话框，默认选择的是[Standard]选项。

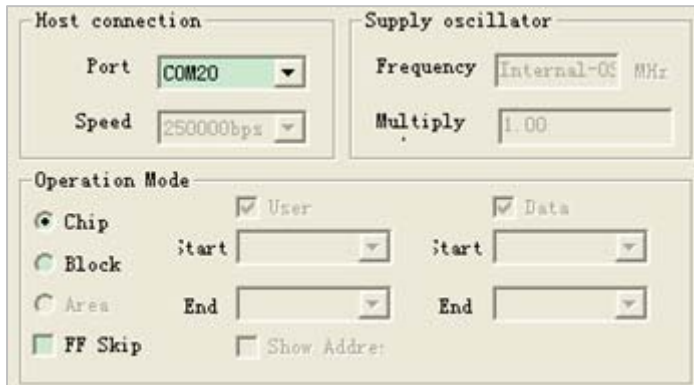
图 5-9. Device Setup对话框的[Standard]选项



<3> 选中 **PRM File Read** 按钮打开参数文件选择对话框。选中目标设备所需的参数文件，然后点击 **打开(O)** 按钮。

<4> 根据编程环境，设置[Host Connection]、[Operation Mode]以及[Supply Oscillator]区域的内容。下面给出了一个例子。

图 5-10. 设置后的Device Setup对话框[Standard]选项

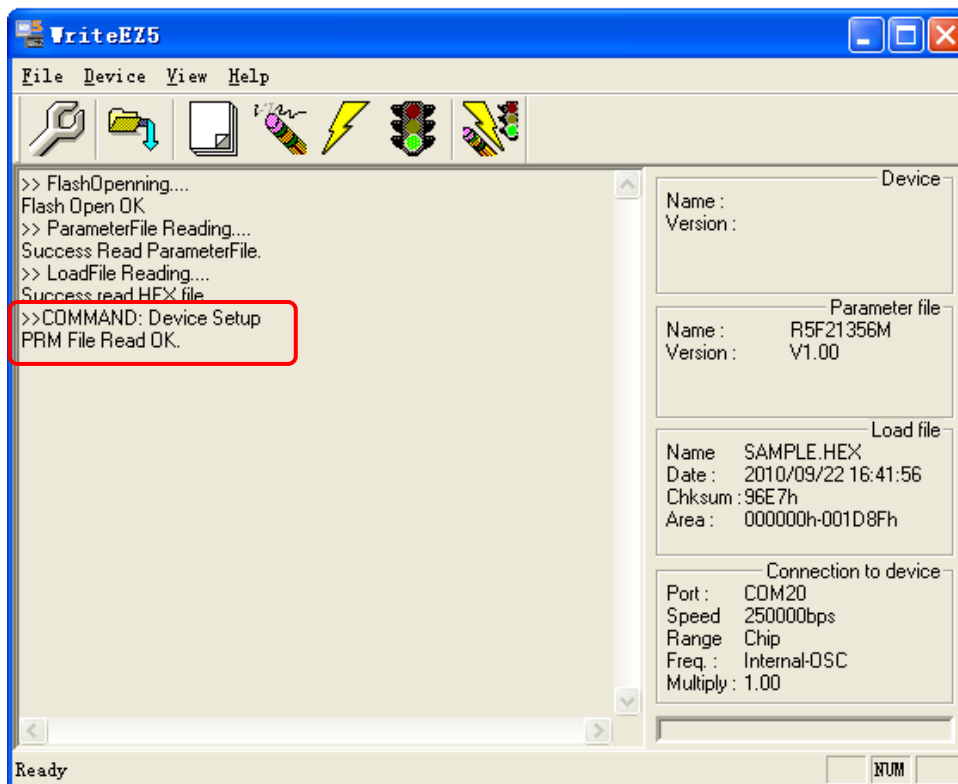




<5> 根据编程环境设置[Advance] 选项卡的[Command options]、 [Security flag settings]和[ID Code settings]中的选项内容。详细内容，请参考 WriteEZ5 的用户手册及 WEZ5\_readme。

<6> 点击确定按钮，完成编程环境的设置，并自动关闭 Device Setup 对话框。主窗口的界面如下所示。

图5-11. 完成编程环境的设置



(2) 程序文件的选择

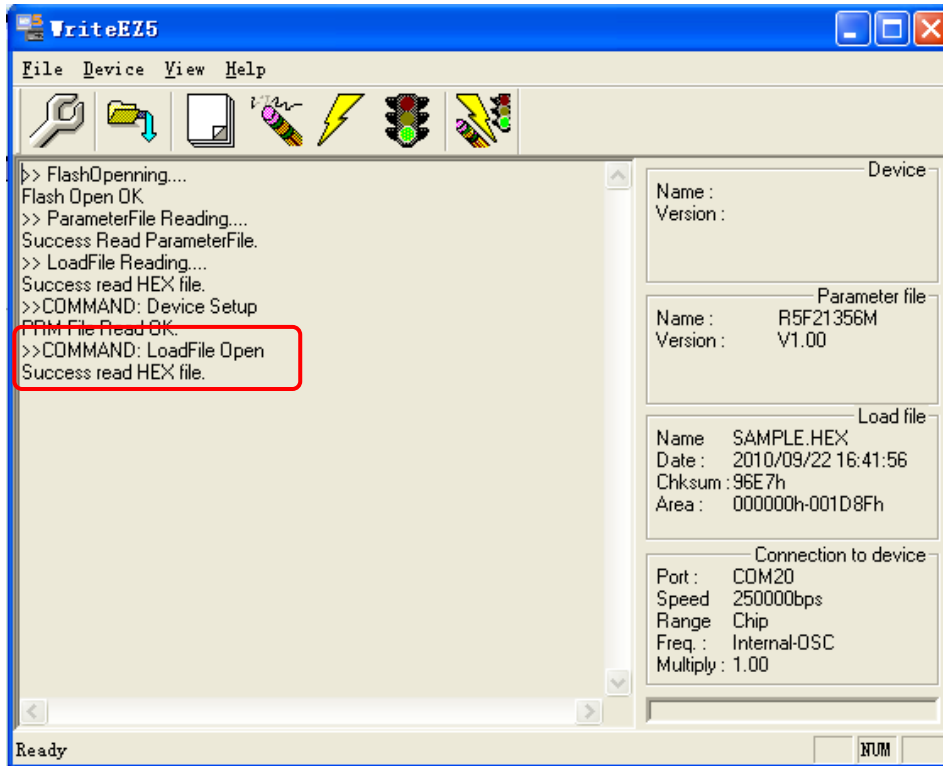
选择程序文件，请按照下述的步骤<1>至<3>来进行。

<1> 选择菜单栏的[File]，然后点击[Load...]

<2> 弹出程序文件选择对话框。

<3> 选择要写入目标设备的程序文件，然后点击 **Open** 按钮。

图5-12. 完成程序文件的选择

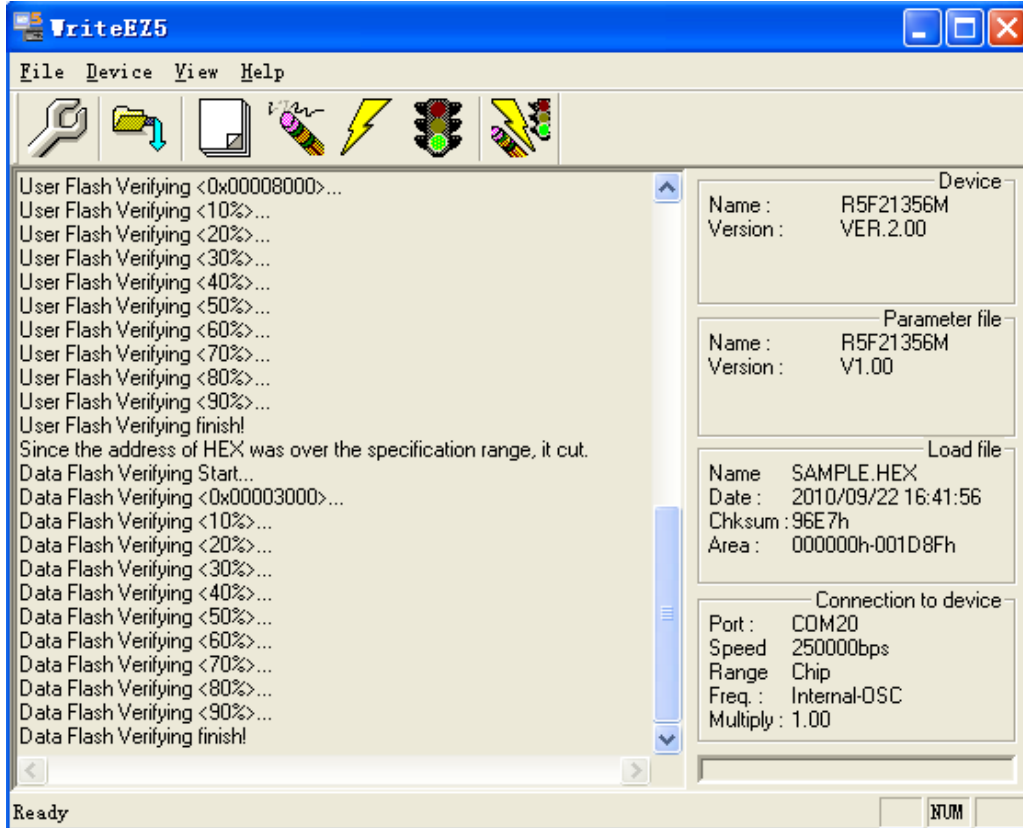


## (3) 执行[Autoprocedure (EPV)]命令

点击菜单栏的[Device]，然后再点击[Autoprocedure (EPV)]。

[Blank Check]，[Erase]（如果目标设备上的 Flash 存储器非空），[Program]，和[Verify]命令会在目标设备上依次执行。当本命令被正常执行完成后，出现如下窗口。

图5-13. [Autoprocedure (EPV)]命令执行完成



### 5.3.5 系统关闭顺序

按照下面的顺序来结束 Flash 编程并关闭系统。

如果未按照这个顺序操作，目标系统或者 EZ-CUBE 可能会被损坏。

#### (1) 关闭 WriteEZ5

如果不需要对其它设备进行编程，则请关闭 WriteEZ5。

**备注** WriteEZ5被关闭后，相关的设置信息被保存。因此下次打开WriteEZ5时会保持同样的设置。

#### (2) 关闭目标系统电源

关闭目标系统的电源。如果电源选择开关设置在“5”，则此步骤可以略过。

#### (3) 移除 USB 线缆

从 EZ-CUBE 或者主机上移除 USB 线缆。

如果电源选择开关设置为“T”时，可以在此步骤之前先执行（4）**移除目标线缆**。

#### (4) 移除目标线缆

从 EZ-CUBE 或者目标系统上拔掉目标线缆。

### 5.3.6 Flash 编程的注意事项

本节描述了 Flash 编程的注意事项。请认真阅读，以保证 EZ-CUBE 的正确使用。

- 为了提高写入质量，请在使用 EZ-CUBE 之前对下列问题进行充分了解，并加以校验和评估。
  - 电路的设计要符合 EZ-CUBE 用户手册对设备的描述
  - 设备，WriteEZ5 和 EZ-CUBE 的使用都要遵循各自手册的描述
  - 提供给目标系统的电源要稳定

## 第 6 章 如何在 RL78 微控制器上使用 EZ-CUBE

本章描述了如何使用 EZ-CUBE 在 RL78 微控制器上进行片上调试和 Flash 编程。

片上调试是指通过芯片内置的调试功能对已经装载在目标系统上的微控制器进行调试的方法。由于这种调试方法是对板上的目标设备进行操作的，所以非常适合现场调试。

Flash 编程是将程序写入设备内置的 Flash 存储器的方法。可以在板上对设备进行擦除、写入和校验。

请先将 EZ-CUBE 的固件更新为用于 RL78，具体请参照 (1) - (3)。详细内容请参见 **1.4 固件更新**。

- (1) 使用 USB 缆连接 EZ-CUBE 和 PC。**不要将 EZ-CUBE 与目标板相连。**
- (2) 启动[QBEZUTL.exe]，并指定“RL78\_OCD\_FW.hex”固件文件。
- (3) 点击“Start”按钮，然后指定用于 RL78 固件下载到 EZ-CUBE。

如果首次用 RL78 微控制器作为目标设备来使用 EZ-CUBE，那么请仔细阅读以下内容。

### 6.1 目标系统设计

要用EZ-CUBE和目标系统进行通信，必须在目标系统上安装对应的通信电路。本节描述了相关电路的设计以及接口的安装。

### 6.2 片上调试

本节描述了用EZ-CUBE进行片上调试的系统配置和启动方法。

### 6.3 Flash 编程

本节描述了用 EZ-CUBE 进行 Flash 编程的系统配置和启动方法。

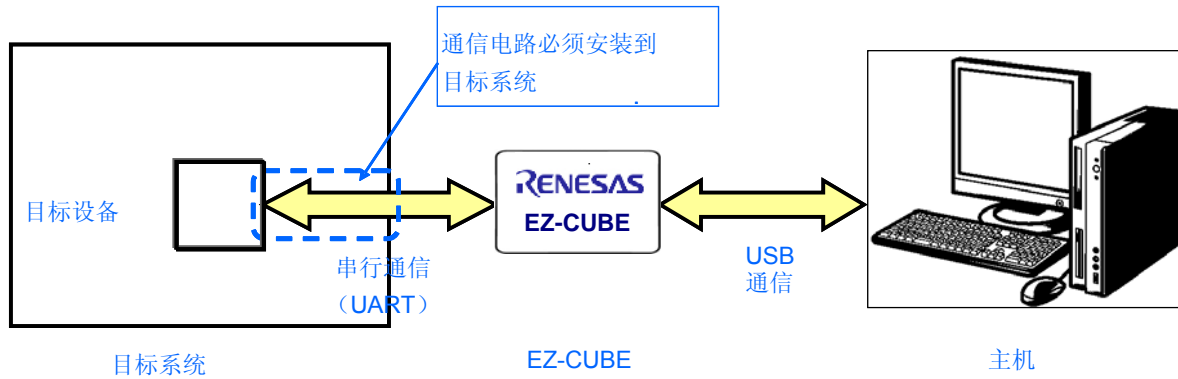
## 6.1 目标系统设计

为了实现片上调试和Flash编程功能，需要设计目标系统的电路，本节描述了相关的电路设计。

图6-1为EZ-CUBE通信的整体接口框图。如图所示，EZ-CUBE与目标系统的目标设备之间进行串行通讯。为了实现通讯，目标系统上必须安装用于通信的电路。请参考本节的内容设计合适的电路。

这里用于串行通信的引脚基本上和使用 Flash 编程器（例如 EZ-CUBE）时使用的串行通信引脚相同，但是有一些设备并不支持这些引脚。

图6-1. 通信接口概况



### 6.1.1 引脚配置

本节将说明 EZ-CUBE 和目标系统之间使用的接口信号。表 6-1 中列出了引脚的配置情况。表 6-2 说明了各个引脚的功能。

表 6-1. 引脚配置

| 引脚编号 | 引脚名称 <sup>注</sup> |
|------|-------------------|
| 1    | GND               |
| 2    | RESET_IN          |
| 3    | Vdd               |
| 4    | FLMD0             |
| 5    | CLK               |
| 6    | RxD.              |
| 7    | RESET_OUT         |
| 8    | TOOL0             |

注 EZ-CUBE 内的信号名称。

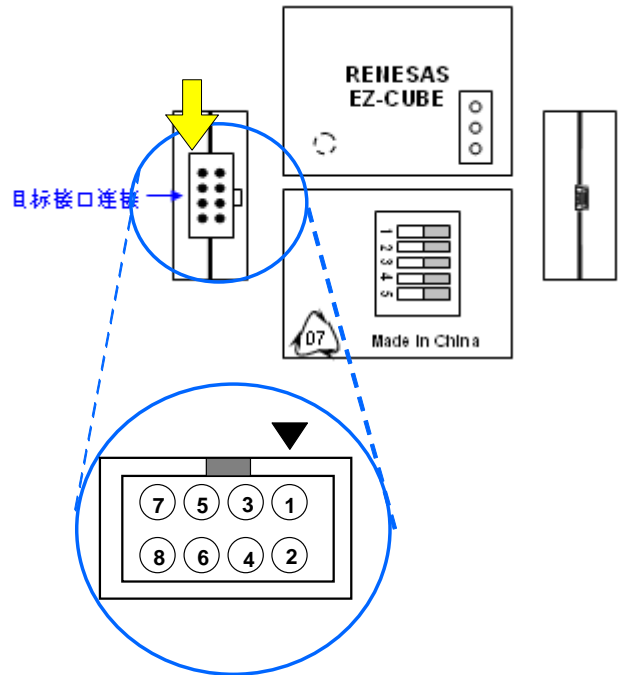


表 6-2. 引脚功能

| 引脚名称      | IN/OUT <sup>注</sup> | 描述                     |
|-----------|---------------------|------------------------|
| RESET_IN  | IN                  | 该引脚用于从目标系统输入 reset 信号  |
| RESET_OUT | OUT                 | 该引脚用于向目标系统输出 reset 信号  |
| FLMD0     | OUT                 | 该引脚用于设定目标设备进入调试模式或编程模式 |
| TOOL0     | OUT                 | 该引脚用于向目标设备发送命令/数据      |

注 以 EZ-CUBE 为参照基准。

### 6.1.2 电路连接举例

目标系统上的电路设计会根据接口信号而有所不同。接口的引脚名称可能会因目标设备有所不同。详细信息请参考目标设备的用户手册。参照表 6-1 和相关的电路连接示例。

**注意事项** 在电路连接示例中出现的常数值为参考值。如果进行 flash 编程旨在量产，请彻底评估目标设备的规格是否满足要求。

图 6-2. RL78/G12(20pin,24pin)电路连接示例

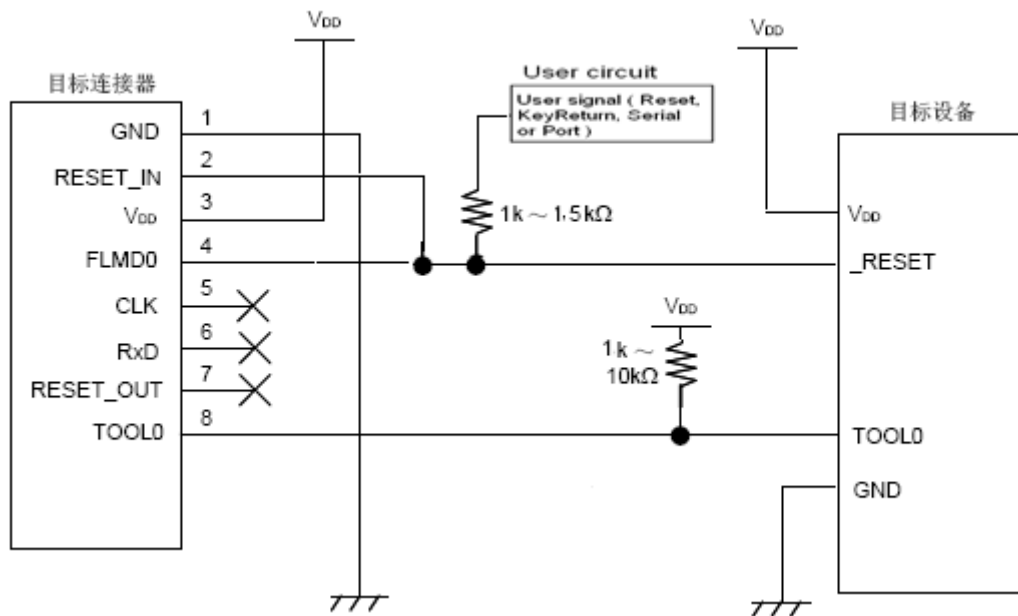
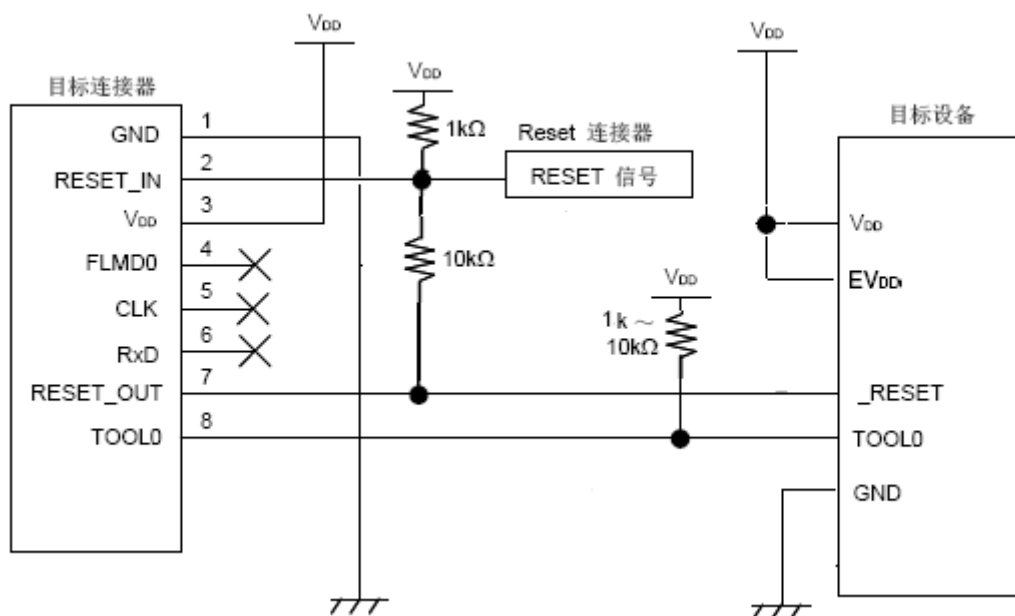


图 6-3. RL78/G12(30pin), RL78/G13, RL78/G14, RL78/I1A 电路连接示例



### 6.1.3 reset 引脚的连接

本节描述了电路中 reset 引脚的连接，此处必须特别注意，电路连接示例见前面的章节。

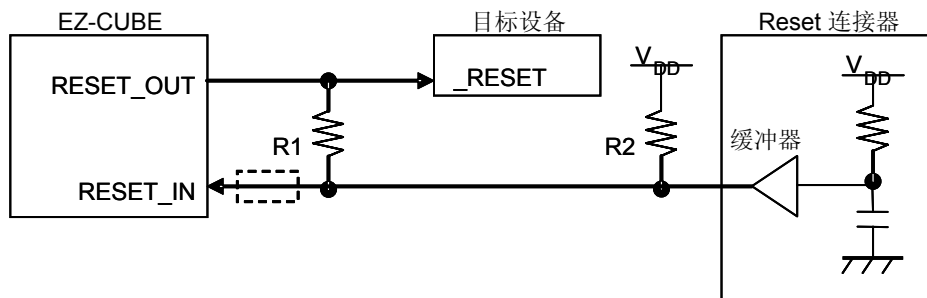
片上调试时，来自目标系统的 reset 信号输入 EZ-CUBE，被屏蔽后，再输入到目标设备。于是 reset 信号的连接根据是否连接 EZ-CUBE 而有所不同。

Flash 编程时，电路的设计必须保证目标系统的 reset 信号和 EZ-CUBE 的 reset 信号不冲突。

**推荐通过电阻自动切换 reset 信号。**图 6-4 说明了 6.1.2 电路连接示例中描述的 reset 引脚连接情况。

这种连接的设计是假定目标系统的复位电路包含有 N 沟道开漏缓冲器（输出阻抗：100 欧姆或更小）。当 EZ-CUBE 的 RESET\_IN/OUT 逻辑反转时，VDD 或 GND 的电平可能会不稳定，所以请注意备注中所描述的条件。

图 6-4. 包含缓冲器的复位电路的连接

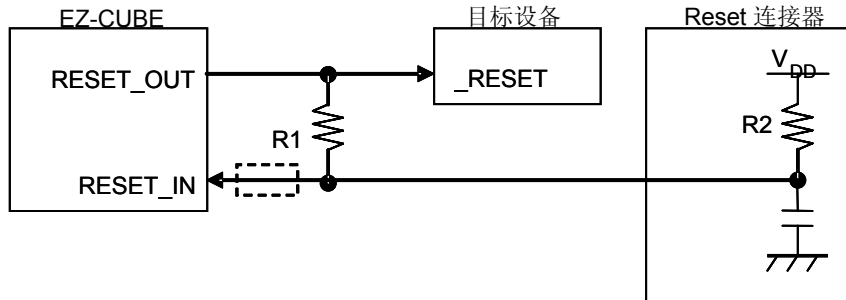


**备注** 需确保 R1 的阻值至少是 R2 阻值的十倍，R1 为 10 kΩ 或更大。  
如果复位电路的缓冲器是 CMOS 输出，则无需上拉电阻 R2。  
当只用于 Flash 编程时，可以略去虚线框内的电路。



图 6-5 所示的电路连接情形是目标系统的复位电路不包含缓冲器，并且 reset 信号仅通过电阻或电容产生。设计电路时注意备注中描述的条件。

图 6-5. 不包含缓冲器的复位电路的连接



**备注** 需确保R1的阻值至少是R2阻值的十倍，R1为10 kΩ或更大。  
当只用于Flash编程时，可以略去虚线框内的电路。

#### 6.1.4 在目标系统上安装连接器

在设计目标系统时，必须安装 EZ-CUBE 和目标系统的连接器。可以选择 2.54mm 间距的 8 引脚通用连接器。

除了专用的连接线之外，同时也支持散线连接。

## 6.2 片上调试

本节描述了使用 EZ-CUBE 进行片上调试的系统配置，启动/关闭顺序和调试中的若干注意事项。

### 6.2.1 调试功能

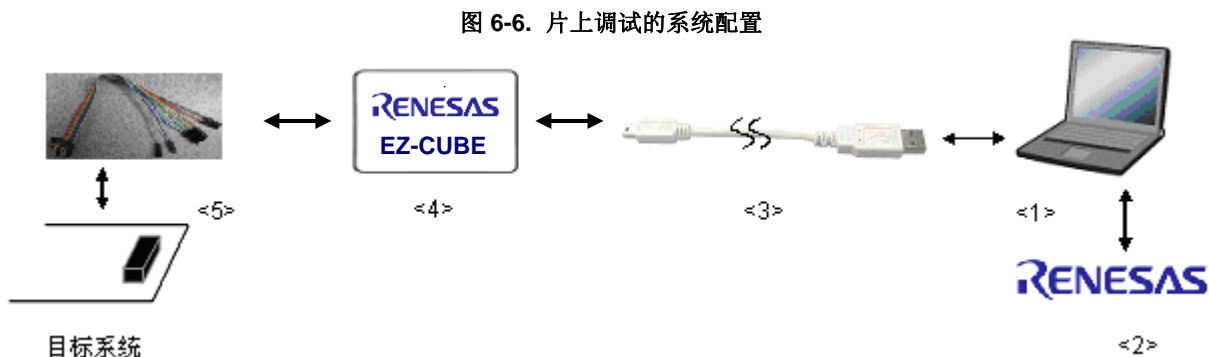
表 6-3 列出了以 RL78 微控制器作为目标设备，进行调试时的调试功能。

表 6-3. 调试功能

| 功能        | 规格定义   |
|-----------|--|
| 目标 MCU    | RL78系列<br>RL78/G12, RL78/G13, RL78/G14, RL78/I1A |
| 安全        | 10 字节 ID 码验证                                     |
| 下载        | 支持   |
| 执行        | 全速执行，忽略断点执行，单步执行，下一步执行，运行至此，重启                   |
| 硬件断点      | 1 个  |
| 软件断点      | 多个   |
| RAM 监视    | 支持   |
| 调试所用的功能引脚 | TOOL0  |

### 6.2.2 系统配置

图 6-6 所示为片上调试的系统配置。



- <1> 主机  
要求有 USB 接口。
- <2> 软件  
包含 CubeSuite+ for China, USB 驱动等。
- <3> USB 线缆 (附件)
- <4> EZ-CUBE (本产品)
- <5> 8 引脚目标线缆 (附件)

### 6.2.3 系统启动顺序

本节描述了系统启动的顺序。请注意按次序操作。

#### (1) 软件的准备和安装

以下软件是片上调试所必须的。关于软件的准备和安装的信息，请参考 EZ-CUBE 附带的安装手册。

- CubeSuite+
- USB 驱动

#### (2) 开关设置

##### For RL78/G12(20pin, 24pin)

- SW-1 开关: 请选择 "M2"
- SW-2 开关: 请选择 "Int. Clock"
- SW-3 开关: 请选择 " Debug Mode"
- SW-4 开关: 请根据实际目标设备使用选择
- SW-5 开关: 请选择 "R8C"

##### For RL78/G12(30pin), RL78/G13, RL78/G14, RL78/I1A

- SW-1 开关: 请选择 "M2"
- SW-2 开关: 请选择 "Int. Clock"
- SW-3 开关: 请选择 " Debug Mode"
- SW-4 开关: 请根据实际目标设备使用选择
- SW-5 开关: 请选择 "Other"

**注意事项 1.** 在 USB 线缆处于连接状态时，请勿改变开关的设置。

**2.** 最大允许电流 100mA，所以请勿将 EZ-CUBE 连接到电流很大的目标系统上。在 EZ-CUBE 与主机连通后，主机始终保持向 EZ-CUBE 供电。

#### (3) 连接目标系统

EZ-CUBE 连接到目标系统，此时**请确保目标系统的电源处于关闭状态。**

#### (4) 连接 USB 线缆

将 EZ-CUBE 与主机相连接，**连接时请务必保持目标系统电源关闭。**

当 USB 缆与 PC 机间连接时，模式灯点亮为红色。

#### (5) 目标系统上电

打开目标系统的电源。如果电源选择为“5”，则这一步并不是必需的。

#### (6) 启动调试器

打开调试器。

此步骤及之后的操作，请参考 CubeSuite+的用户手册。

如果调试器无法正常开启或者操作不稳定，可能是由以下问题引发的。

- EZ-CUBE 和目标系统之间通信错误

固件是否更新；

开关是否选择正确；

目标连线是否正确。

- 未预留用户资源或者未设置安全 ID

使用 EZ-CUBE 进行调试时，必须预留出一部分用户资源作为调试监控区域和串行通信接口区域，安全 ID

也必须设置。详细信息请参考 **6.2.5 用户资源的保留和设置安全 ID**。

- 使用了不支持的软件（调试器，设备文件或者固件）  
所使用的软件可能不支持目标设备的调试。
- EZ-CUBE 损坏  
EZ-CUBE 可能被损坏。

#### 6.2.4 系统关闭顺序

请按照下列操作顺序停止调试并关闭系统。

如果没有按照下列顺序操作，目标系统或者 EZ-CUBE 可能被损坏。

##### (1) 停止调试器

停止调试器的运行并断开调试器连接。

##### (2) 关闭目标系统电源

关闭目标系统的电源。如果电源选择开关放在“5”的位置，则无需本步操作。

##### (3) 拔除 USB 线缆

从 EZ-CUBE 或主机上拔除 USB 线缆。

##### (4) 拔除目标线缆

从 EZ-CUBE 或目标系统上将目标线缆拔下。

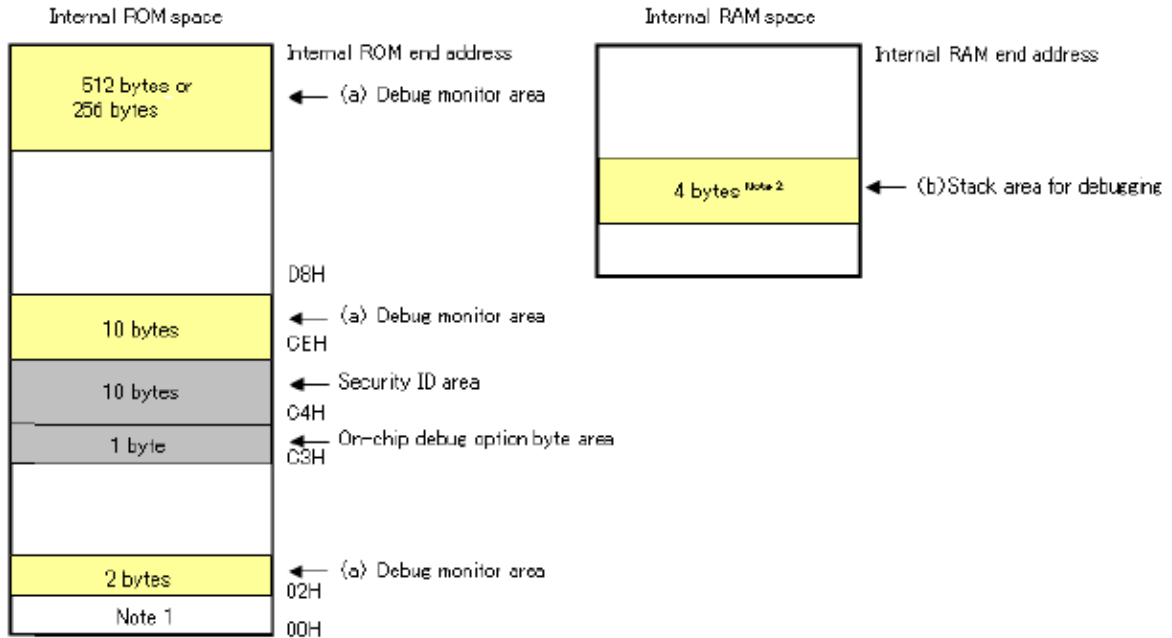
#### 6.2.5 保留用户资源和设置安全 ID

用户必须做好下面的保留以保证 EZ-CUBE 和目标设备之间的通信以及实现每一个调试功能。使用 Renesas ELECTRONICS 的编译器，这些都可以通过 CubeSuite+ 选项设置。请参考下面的描述并设置，详细内容请查阅 CubeSuite+ 相关用户手册。

- 内存空间的保留

图6-7中的阴影部分是存放调试监控程序的保留区域，所以用户程序或数据不能存放在这些空间。这些空间必须保证不被用户程序使用。另外，这些区域也不能被用户程序重写。

图 6-7. 分配给调试监控程序的内存空间



- 注
1. 调试过程中，复位向量被重写，指向监控程序区。
  2. 当执行自编程时，将是 12 字节。

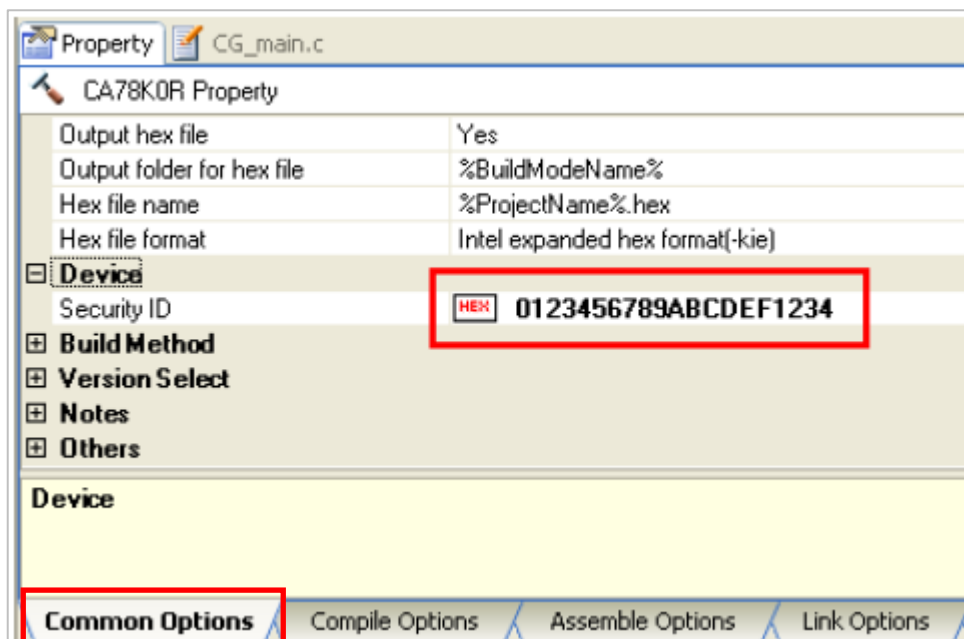
• 设置安全 ID 和片上调试选项字节

这个区域用来存放安全设置以防无认证的人读取 Flash 存储器。安全 ID 功能就像启动调试器的密码。只有在调试器启动时输入的安全 ID 和存放在这个区域的安全 ID 匹配时，调试器才能正常启动。如果 ID 码不匹配，调试器根据片上调试选项字节区域设置的值对目标设备进行操作（参考表 6-4）。

[如何设置]

当使用 RENESAS ELECTRONICS 的编程器时，设置安全 ID 就可以在安全 ID 区域产生一个 ID 码。如下图所示，在[Build Tool]的 [Common Options]选项卡的“Device”处进行设置一个任意的 ID 码（10 字节）。

图 6-8. 安全ID设置样例



**注意事项** 如果已经忘记安全 ID 码，请通过 flash 编程擦除 flash 存储器，然后再次设置安全 ID。

#### 片上调试选项字节区域

这个区域用来存放安全设置以防无认证的人读取 Flash 存储器。调试器根据设置的值对目标设备进行操作，如下表所示。

表 6-4. 片上调试选项字节设置和操作

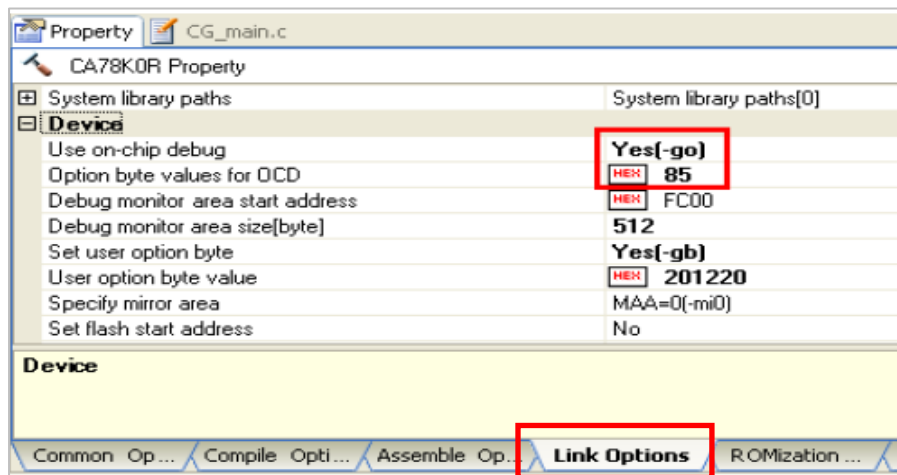
| 设置值  | 描述                             | 备注                   |
|------|--------------------------------|----------------------|
| 0x04 | 即使连上了片上调试仿真器（如EZ-CUBE）也不能进行调试。 | 这个设置只在Flash编程或自编程时有效 |
| 0x85 | 不管安全ID码认证失败多少次都不会擦除片上Flash存储器  | -                    |
| 0x84 | 如果安全ID码认证失败擦除所有片上Flash存储器区域    | -                    |
| 其它   | 禁止设置                           | -                    |

[如何设置]

下面说明了用RENASAS ELECTRONICS的汇编器或编译器时的设置样例。如图6-9中所示，使用CubeSuite+，在[Build Tool]的[Link Options]选项卡的“Device”处进行设置。

例 把控制值设置为 0x85。

图 6-9. 片上调试选项字节样例



#### (a) 调试监控程序的保留区域

这是一个放置调试监控程序的区域。监控程序为调试通信接口和 CPU 的运行或者断点做初始化操作。用户程序或数据不能存放在片上调试选项字节周边的 22 个字节的区域以及内部 ROM 底端的 512 字节区域内。另外，复位向量会被重写并指向放置监控程序的地址。

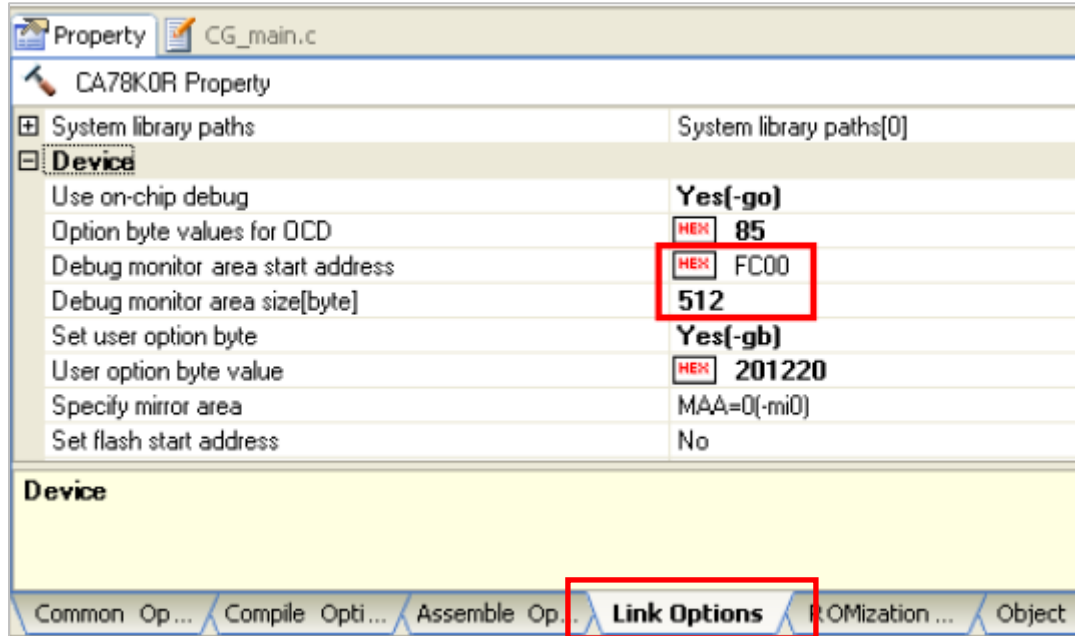
[如何预留区域]

如果用户程序不使用这个区域，可以不进行预留。

为了避免在调试时出现问题，建议提前预留区域。

如图 6-10 所示，使用 CubeSuite+，在[Link Options]选项卡的”Device”处进行设置。

图 6-10. 预留调试监测区域

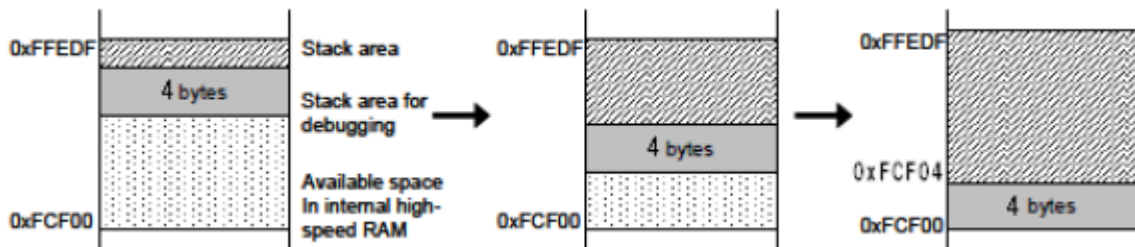


(b) 调试用堆栈区域的保留

这个区域需要 4 个字节作为调试的堆栈区域。因为这个区域分配在紧邻堆栈区域之前，所以这个区域的地址根据堆栈的增加和减少而有所改变。就是说，在使用的堆栈区域之外有 4 个额外的字节。

图 6-11 说明了堆栈区域增加时的情况，图中的内部高速 RAM 从 0xFCF00 开始。

图 6-11. 调试所用的堆栈地址的变化



[如何预留区域]

通过在用户程序使用堆栈之外再多定义 4 个字节来设置堆栈指针。确保堆栈指针不会超过内部高速 RAM 的起始地址。

**备注** 关于如何预留自编程序的堆栈区域请参考自编程手册。

### 6.2.6 调试时的注意事项

本节介绍使用 RL78 微控制器的片上调试功能时需要注意的事项。  
请仔细阅读以下事项以正确使用 EZ-CUBE。

- (1) 处理用于调试的设备  
不要在批量生产的产品上安装曾用于调试的设备，因为 Flash 存储器在调试过程中被多次重写，Flash 存储器的重写次数无法保证。另外，不要把调试监控程序写入批量生产的产品。
- (2) Flash 自编程  
如果存放调试监控程序的区域被 Flash 自编程重写，调试器就不能正常操作。这个注意事项对这个区域的引导交换操作也适用。
- (3) 复位后的操作  
在外部复位或内部复位后，监控程序会进行调试初始化处理。因此，从产生复位到用户程序执行的时间和实际设备的操作有所不同。
- (4) 使用实机调试而不使用 EZ-CUBE  
如果使用实机调试而不使用 EZ-CUBE，请使用 RFP 写入用户程序。通过调试器下载的程序包括监控程序，这样的程序如果在有使 TOOL0 引脚变成低电平的操作时会出现误操作。
- (5) 调试器启动后的操作  
调试器启动后，如果调试器配置对话框中的目标设备连接设置和上次的调试设置不同，内部 Flash 存储器会被擦除。
- (6) 通过 Flash 编程写入程序后的调试  
如果通过 RFP 把程序写入内部 Flash 存储器，即使在片上调试选项字节设置中允许了片上调试，也不能进行片上调试。如果要对这样的目标设备要进行调试，可以先用 RFP 擦除内部 Flash 存储器后通过调试器下载程序。



## 6.3 Flash 编程

本节将介绍使用 EZ-CUBE 对 RL78 微控制器的 Flash 进行编程时的系统配置以及启动/关闭顺序。

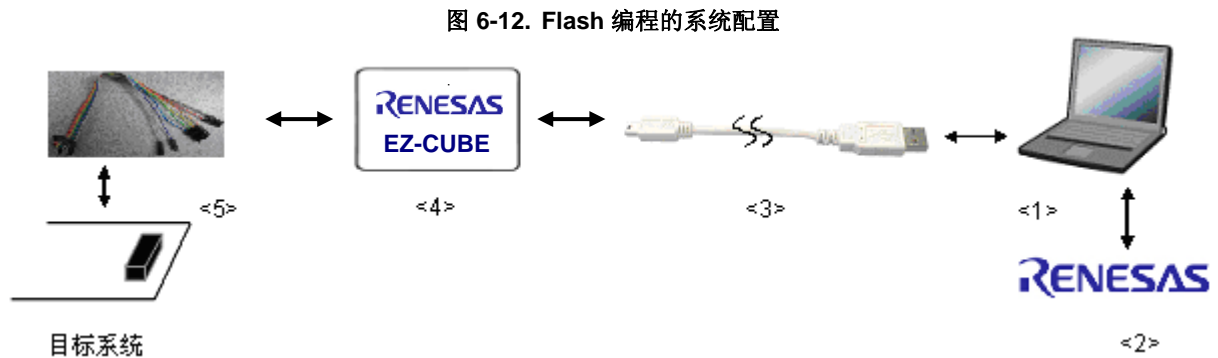
### 6.3.1 编程功能的规格定义

表6-5. 编程功能的规格定义

| 功能       | 规格定义说明                           |
|----------|----------------------------------|
| 主机接口     | USB 2.0                          |
| 目标接口     | UART                             |
| 目标系统电压   | 2.7到5.5 V（根据目标设备而定）              |
| 时钟源      | 可以提供8 MHz时钟信号<br>可以使用安装在目标系统上的时钟 |
| 电源       | 5 V $\pm$ 0.3V（最大电流：100 mA）      |
| 获取设备特定信息 | 使用RENESAS ELECTRONICS的RL78参数文件   |
| 安全标志的设置  | 有                                |
| 独立操作     | 无（必须连接主机）                        |

### 6.3.2 系统配置

图 6-12 显示了 Flash 编程时的系统配置。



<1> 主机

要求有 USB 接口

<2> 软件

包括 RFP（Renesas Flash Programmer），USB 驱动等。

<3> USB 线缆（附件）

<4> EZ-CUBE（本产品）

<5> 8 引脚目标连接线（附件）

### 6.3.3 系统启动顺序

本章将介绍系统的启动顺序。请按照以下顺序进行操作。

#### (1) 软件的准备和安装

以下软件是片上调试所必须的。有关软件的准备和安装的详细信息，请参考 EZ-CUBE 附带的安装手册。

- RFP
- USB 驱动

#### (2) 开关设置

##### **For RL78/G12(20pin, 24pin)**

SW-1 开关: 请选择 "M2"

SW-2 开关: 请选择 "Int. Clock"

SW-3 开关: 请选择 " Debug Mode"

SW-4 开关: 请根据实际目标设备使用选择

SW-5 开关: 请选择 "R8C"

##### **For RL78/G12(30pin), RL78/G13, RL78/G14, RL78/I1A,**

SW-1 开关: 请选择 "M2"

SW-2 开关: 请选择 "Int. Clock"

SW-3 开关: 请选择 " Debug Mode"

SW-4 开关: 请根据实际目标设备使用选择

SW-5 开关: 请选择 "Other"

**注意事项**            在 USB 线缆处于连接状态时，请勿改变开关的设置。

#### (3) 连接目标系统

将 EZ-CUBE 连接到目标系统，请务必在目标系统上电之前连接好连线。

#### (4) 连接 USB 线缆

将 EZ-CUBE 与主机相连接，连接时请务必保持目标系统电源关闭。

#### (5) 目标系统上电

打开目标系统的电源。如果电源选择为“5”，则这一步并不是必需的。

#### (6) 启动 RFP (Renesas Flash Programmer)

打开 RFP 软件。

### 6.3.4 应用样例

本节描述了使用RFP（Renesas Flash Programmer）的一系列基本操作，使用R5F100LE作为示例设备。描述了执行[Autoprocedure (EP)]命令为目标设备编程的整个过程。命令的详细内容，请参加RFP的用户手册。

本节中操作举例的条件如下。

<目标系统>

目标设备: R5F100LE

<EZ-CUBE>

开关: SW-1 开关: M2  
SW-2 开关: Int. Clock  
SW-3 开关: Debug Mode  
SW-4 开关: T  
SW-5 开关: Other

<RFP>

电源电压: 5V  
操作模式: 芯片  
程序文件: sample.hex  
命令参数: 选中[Blank check before Erase]

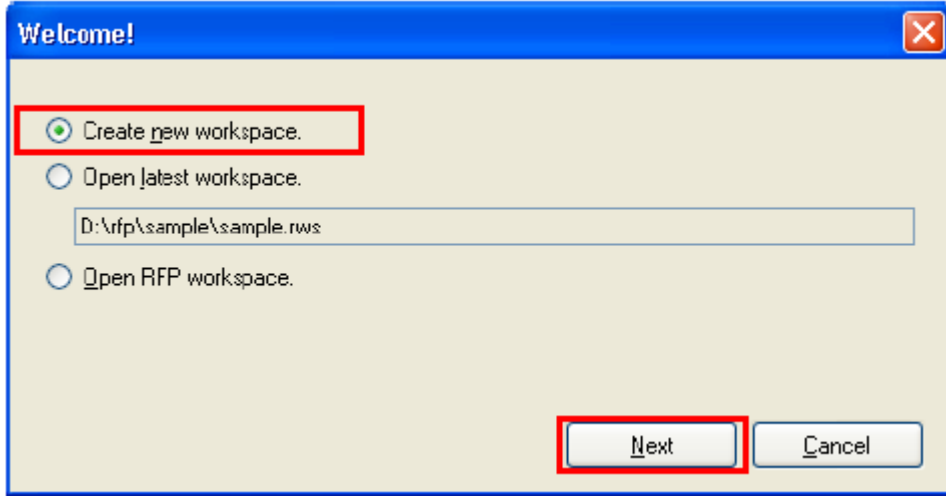
(1) 设置编程环境

请按照下述<1>到<6>的步骤进行编程环境的设置。

<1> 点击【开始】按钮，指向[程序]，[Renesas Electronics CubeSuite+]，[Programming Tools]，[Renesas Flash Programmer Vx.xx]，点击[Renesas Flash Programmer Vx.xx [Basic mode]]，启动 RFP 软件。

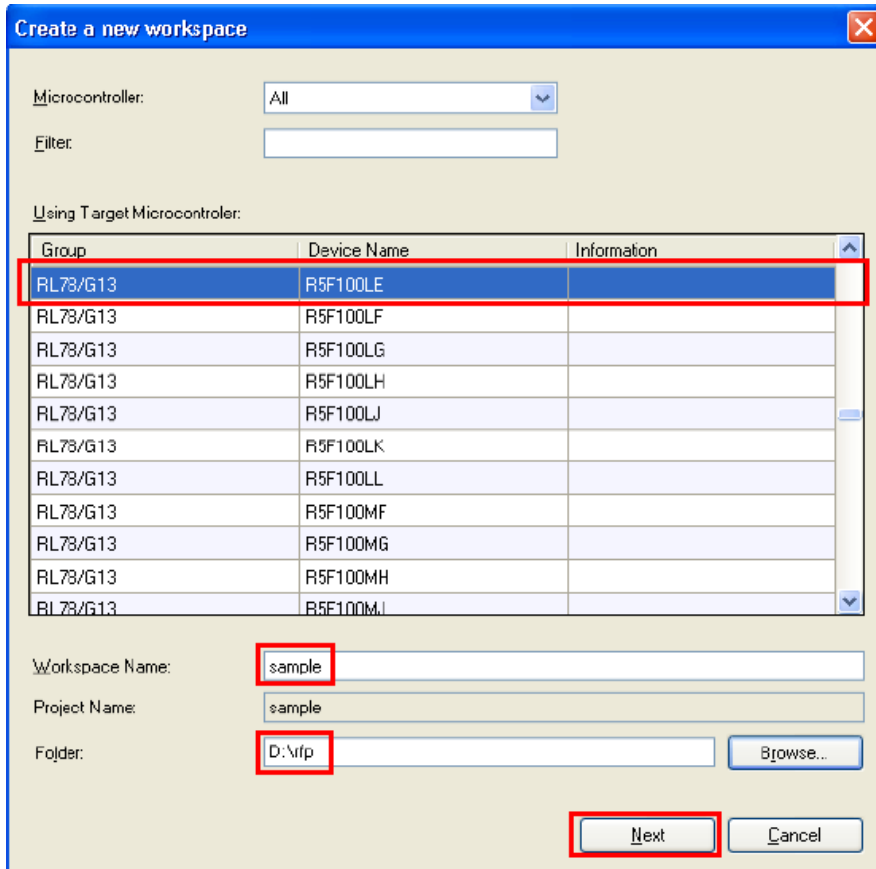
<2> 创建[ Create new workspace. ]

图 6-13 创建



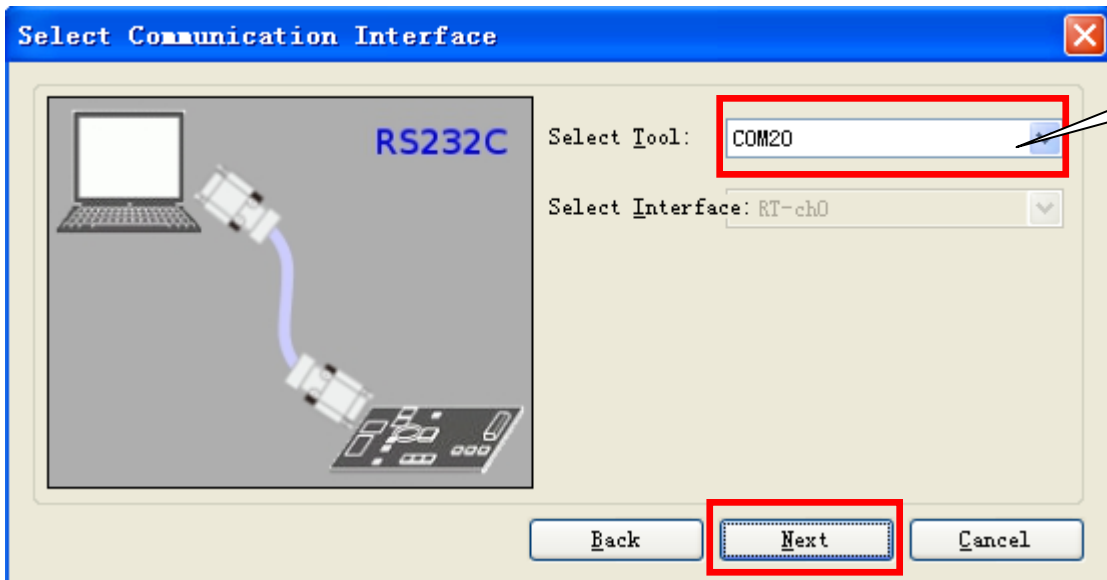
<3> 选择编程芯片，如 R5F100LE

图 6-14 选择编程芯片



<4> 选择通讯接口[ Select Communication Interface ], 请选择实际的编程接口, 这里为 COM20。

图6-15. Select Communication Interface 选项卡



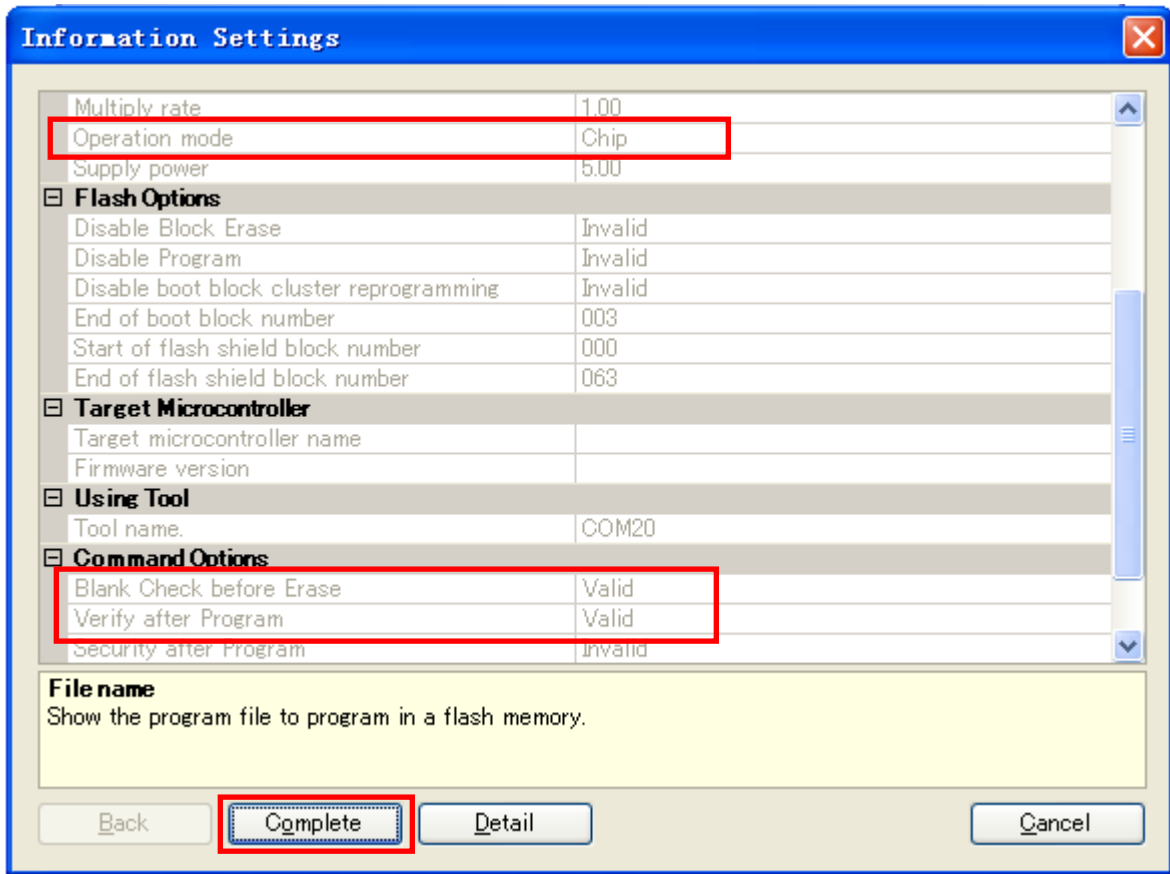
<5> 设置电源。

图6-16. 设置 [Setting Power Supply]选项卡



<6> 根据编程环境设置[Information Setting]选项卡的各项选项。这里设置[Target]和[Command Options]中的选项。详细内容, 请参考 RFP 的用户手册。点击确定按钮, 完成编程环境的设置。

图6-17. 设置 [Information Settings]选项卡



## (2) 程序文件的选择

请按照下述的步骤<1>至<3>选择程序文件。

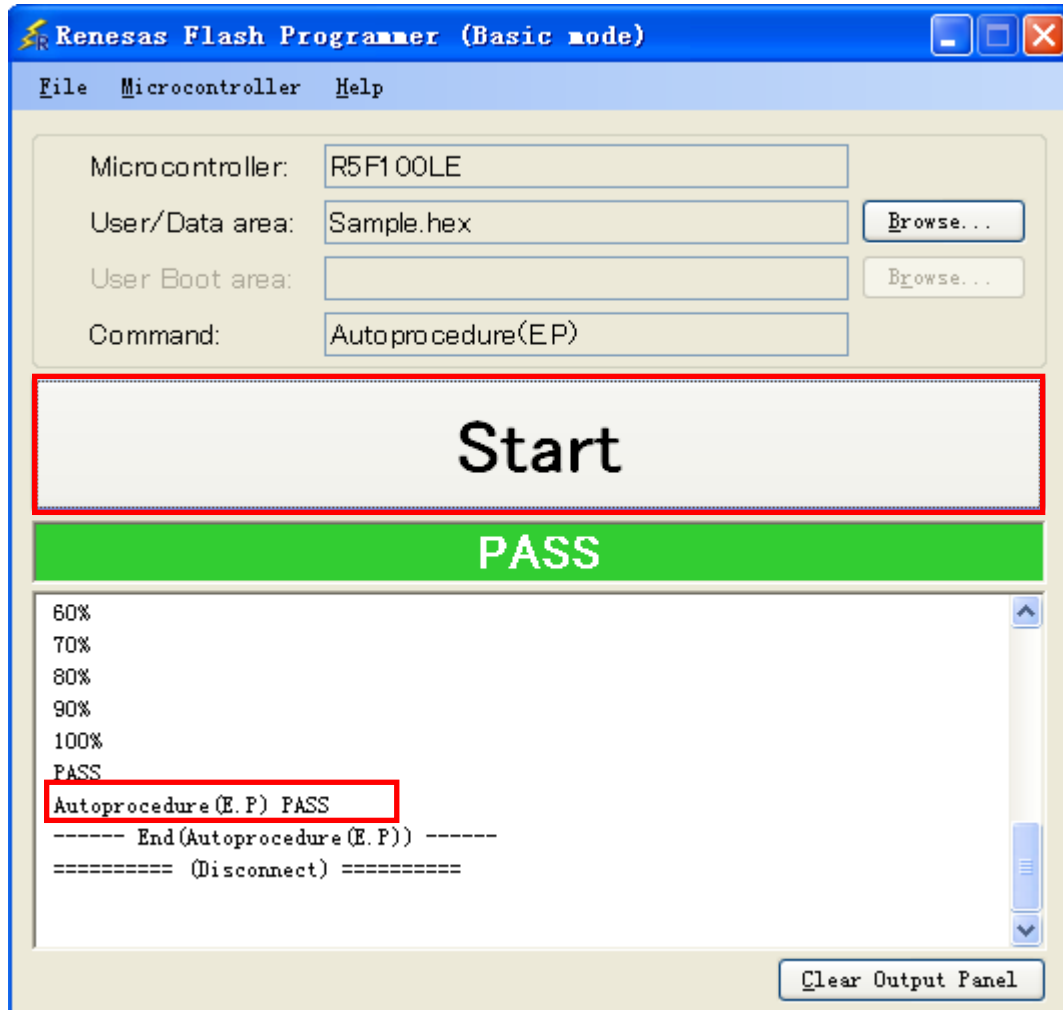
- <1> 在[User/Data area]区域点击 **Browse...** 按钮。
- <2> 弹出程序文件选择对话框。
- <3> 选择要写入目标设备的程序文件，然后点击 **打开** 按钮。

## (3) 执行[Autoprocedure(EP)]命令

- <1> 点击菜单栏的[Microcontroller]，然后选择[Autoprocedure(EP)]命令。
- <2> 点击 **Start** 按钮

[Blank Check]、[Erase]（如果目标设备上的Flash 存储器非空）、[Program]、[Verify]命令将会在目标设备上依次执行。当本命令正常执行完成后，弹出如下窗口。

图6-18 [Autoprocedure(EP)]命令执行完成之后



### 6.3.5 系统关闭顺序

请按以下步骤结束 Flash 的编程并关闭系统。

如果未按照以下顺序进行操作，那么 EZ-CUBE 将有可能被损坏。

(1) **终止 RFP 软件**

如果不继续对其他设备进行编程，则终止 RFP。

(2) **关闭目标系统的电源**

关闭目标系统的电源。如果电源选择开关放在“5”的位置，则无需本步操作。

(3) **移除 USB 线缆**

从 EZ-CUBE 或主机上拔除 USB 线缆。

电源选择开关设置为“T”时，(4) **移除目标线缆**的操作可以在本操作之前进行。

(4) **移除目标线缆**

从 EZ-CUBE 或目标系统上拔除目标线缆。

### 6.3.6 Flash 编程时的注意事项

本节将介绍进行 Flash 编程时需要注意的事项。请务必仔细阅读本节内容以正确使用 EZ-CUBE。

- 为提高写入的编程的质量，请在使用 EZ-CUBE 之前了解以下内容并队列出的各项进行校验和评估。
  - 务必按照设备及 EZ-CUBE 的用户手册中给出的方法进行目标系统的电路的设计。
  - 务必按照设备、RFP 及 EZ-CUBE 各自的用户手册中介绍的方法进行操作。
  - 务必向目标系统提供稳定的电源。



## 第 7 章 如何在 V850 微控制器上使用 EZ-CUBE

本章描述了如何使用 EZ-CUBE 在 V850 微控制器上进行片上调试和 Flash 编程。

片上调试是指通过芯片内置的调试功能对已经装载在目标系统上的微控制器进行调试的方法。由于这种调试方法是对板上的目标设备进行操作的，所以非常适合现场调试。

Flash 编程是将程序写入设备内置的 Flash 存储器的方法。可以在板上对设备进行擦除、写入和校验。

请先将 EZ-CUBE 的固件更新为用于 V850，具体请参照 (1) - (3)。详细内容请参见 **1.4 固件更新**。

- (1) 使用 USB 缆连接 EZ-CUBE 和 PC。**不要将 EZ-CUBE 与目标板相连。**
- (2) 启动[QBEZUTL.exe]，并指定“V850\_OCD\_FW.hex”固件文件。
- (3) 点击“Start”按钮，然后指定用于 V850 固件下载到 EZ-CUBE。

如果首次用 V850 微控制器作为目标设备来使用 EZ-CUBE，那么请仔细阅读以下内容。

### 6.1 目标系统设计

要用 EZ-CUBE 和目标系统进行通信，必须在目标系统上安装对应的通信电路。本节描述了相关电路的设计以及接口的安装。

### 6.2 片上调试

本节描述了用 EZ-CUBE 进行片上调试的系统配置和启动方法。

### 6.3 Flash 编程

本节描述了用 EZ-CUBE 进行 Flash 编程的系统配置和启动方法。

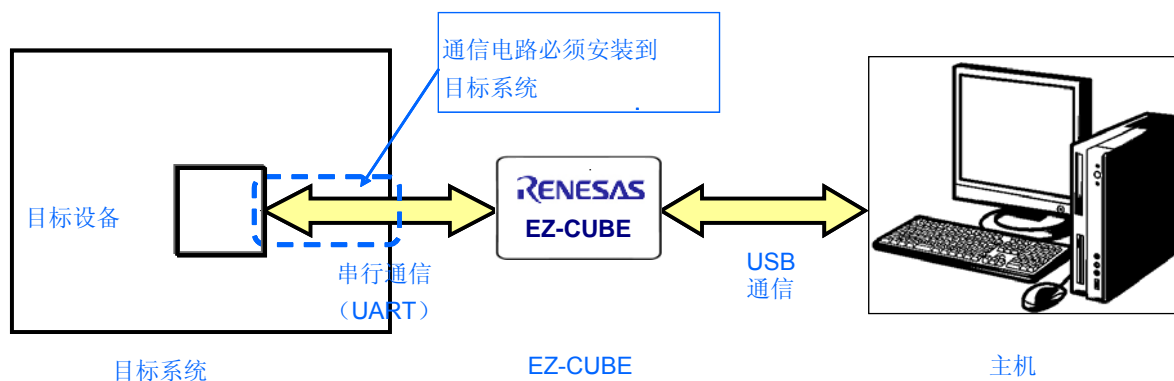
## 7.1 目标系统设计

为了实现片上调试和Flash编程功能，需要设计目标系统的电路，本节描述了相关的电路设计。

图6-1为EZ-CUBE通信的整体接口框图。如图所示，EZ-CUBE与目标系统的目标设备之间进行串行通讯。为了实现通讯，目标系统上必须安装用于通信的电路。请参考本节的内容设计合适的电路。

这里用于串行通信的引脚基本上和使用 Flash 编程器（例如 EZ-CUBE）时使用的串行通信引脚相同，但是有一些设备并不支持这些引脚。

图7-1. 通信接口概况



### 7.1.1 引脚配置

本节将说明 EZ-CUBE 和目标系统之间使用的接口信号。表 7-1 中列出了引脚的配置情况。表 7-2 说明了各个引脚的功能。

表 7-1. 引脚配置

| 引脚编号 | 引脚名称 <sup>注</sup> |
|------|-------------------|
| 1    | GND               |
| 2    | RESET_IN          |
| 3    | Vdd               |
| 4    | FLMD0             |
| 5    | CLK               |
| 6    | RxD.              |
| 7    | RESET_OUT         |
| 8    | TxD               |

注 EZ-CUBE 内的信号名称。

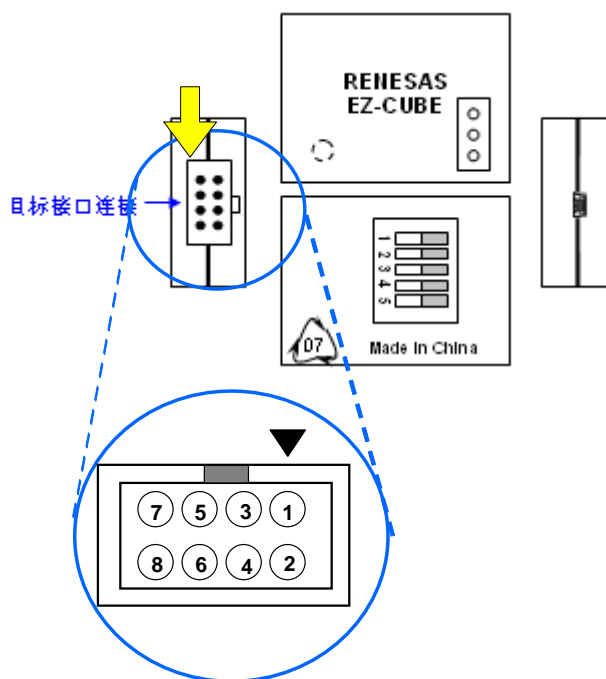


表 7-2. 引脚功能

| 引脚名称      | IN/OUT <sup>注</sup> | 描述                     |
|-----------|---------------------|------------------------|
| RESET_IN  | IN                  | 该引脚用于从目标系统输入 reset 信号  |
| RESET_OUT | OUT                 | 该引脚用于向目标系统输出 reset 信号  |
| CLK       | OUT                 | 该引脚用于向目标系统输出时钟信号       |
| FLMD0     | OUT                 | 该引脚用于设定目标设备进入调试模式或编程模式 |
| RxD       | IN                  | 该引脚用于接收来自目标设备的命令/数据    |
| TxD       | OUT                 | 该引脚用于向目标设备发送命令/数据      |

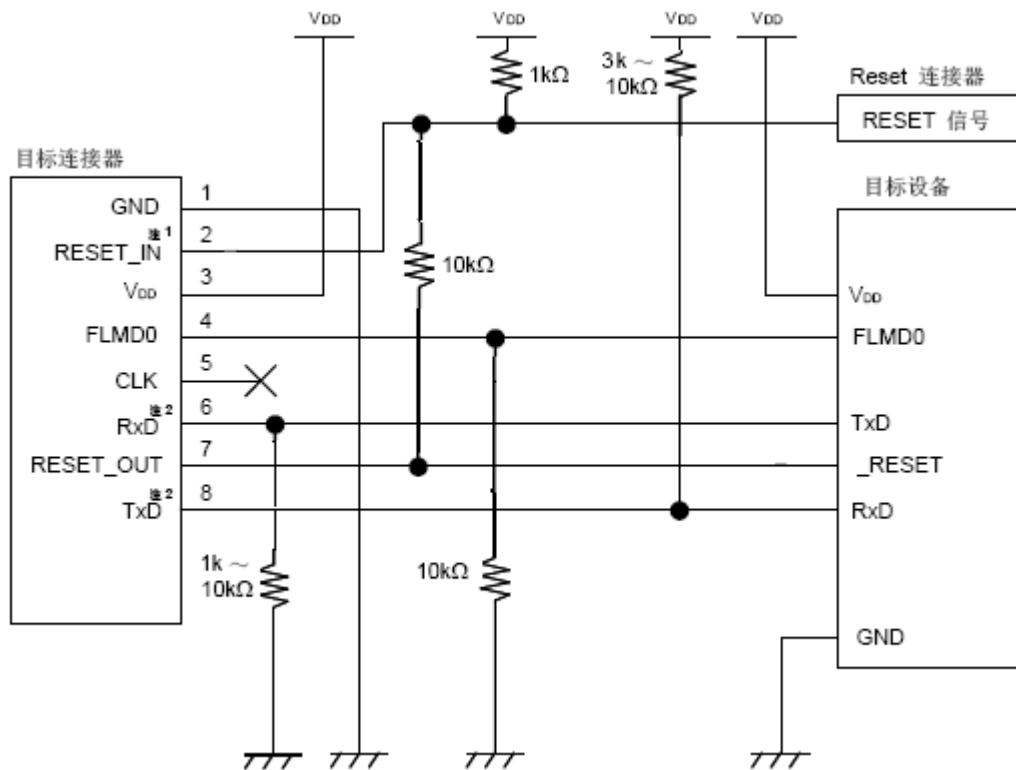
注 以 EZ-CUBE 为参照基准。

### 7.1.2 电路连接举例

目标系统上的电路设计会根据接口信号而有所不同。接口的引脚名称可能会因目标设备有所不同。详细信息请参考目标设备的用户手册。参照表 7-1 和相关的电路连接示例。

**注意事项** 在电路连接示例中出现的常数值为参考值。如果进行 flash 编程旨在量产，请彻底评估目标设备的规格是否满足要求。

图 7-2. 电路连接示例



- 注：
1. 该连接的设计是假定 RESET 信号从 N 沟道开漏缓冲器（输出阻抗：100 kΩ 或更小）输出的。详细信息请参考 7.1.3 reset 引脚的连接。
  2. 将目标设备的 TxD（发送方）连接到目标连接器的 RxD（接收方），同时将目标连接器的 TxD（发送方）连接到目标设备的 RxD（接收方）。

### 7.1.3 reset 引脚的连接

本节将介绍上一节的电路连接举例中提到的需要特别注意的reset引脚的连接方法。

片上调试时，来自目标系统的reset信号输入EZ-CUBE被屏蔽后，再输出到目标设备。因此reset信号的连接根据是否连接EZ-CUBE而有所不同。

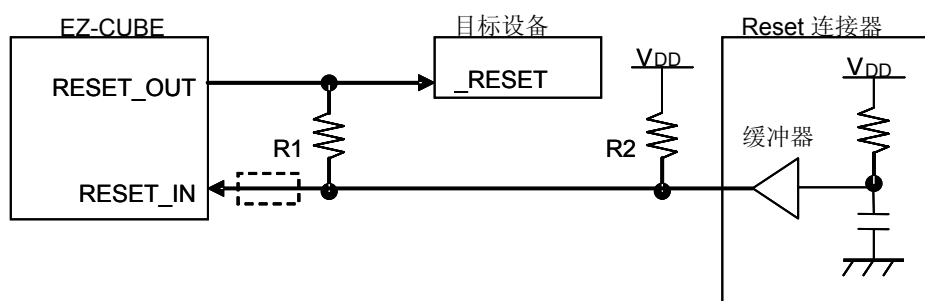
Flash编程时，电路的设计必须保证目标系统的reset信号和EZ-CUBE的reset信号不冲突。

推荐通过电阻自动切换 reset 信号。图 7-3 说明了 7.1.2 电路连接示例中描述的 reset 引脚连接情况。

图7-3说明了7.1.2 电路连接示例中描述的reset引脚的连接情况。

这种连接的设计是假定目标系统的复位电路包含有N沟道开漏缓冲器（输出阻抗：100Ω或更小）。当EZ-CUBE的RESET\_IN/OUT逻辑反转时，VDD或GND的电平可能会不稳定，所以请注意备注中所描述的条件。

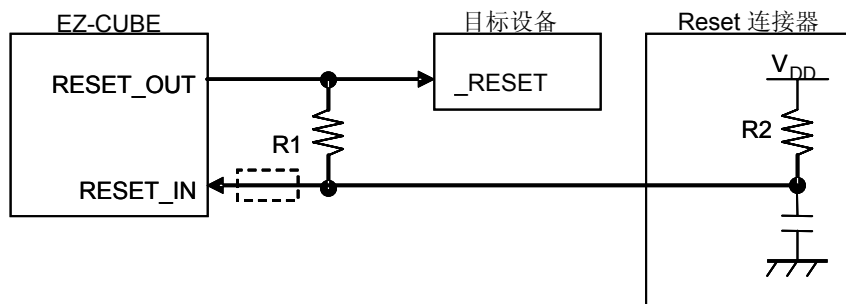
图 7-3. 包含缓冲器的复位电路的连接



**备注** 需确保R1的阻值至少是R2阻值的十倍，R1为10 kΩ或更大。  
如果复位电路的缓冲器是CMOS输出，则无需R2上拉电阻。  
当仅进行Flash编程时，虚线框内的电路就可以略去。

图7-4 所示的电路为目标系统的复位电路中不包含缓冲器，并且reset信号仅通过电阻或电容产生。设计电路时注意备注中描述的条件。

图 7-4. 不包含缓冲器的复位电路的连接



**备注** 需确保R1的阻值至少是R2阻值的十倍，R1为10 kΩ或更大。  
当仅进行Flash编程时，虚线框内的电路就可以略去。

### 7.1.4 在目标系统上安装连接器

在设计目标系统时，必须安装EZ-CUBE和目标系统的连接器。可以选择2.54mm间距的8引脚通用连接器。

除了专用的连接线之外，同时也支持散线连接。

## 7.2 片上调试

本节描述了使用 EZ-CUBE 进行片上调试的系统配置，启动/关闭顺序和调试中的若干注意事项。

### 7.2.1 调试功能

表 7-3 列出了以 V850 微控制器作为目标设备，进行调试时的调试功能。

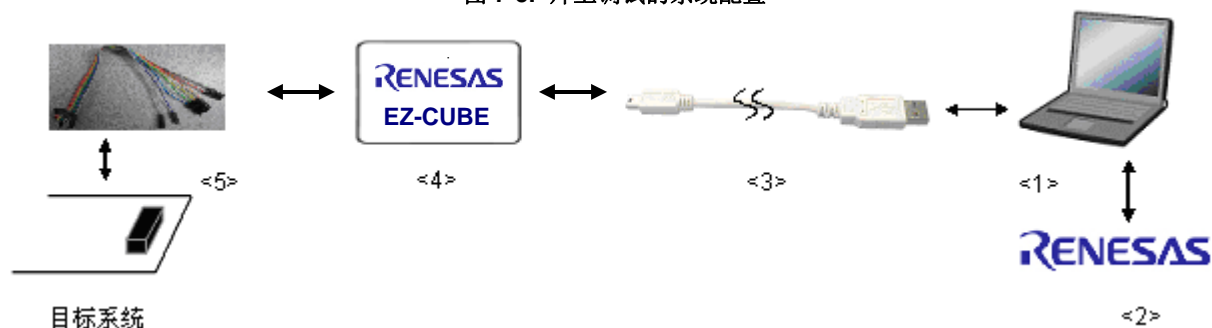
表 7-3. 调试功能

| 功能        | 规格定义                               |
|-----------|------------------------------------|
| 目标 MCU    | V850系列<br>V850ES/Jx3, V850ES/Jx3-L |
| 安全        | 10 字节 ID 码验证                       |
| 下载        | 支持                                 |
| 执行        | 全速执行，忽略断点执行，单步执行，下一步执行，运行至此，重启     |
| 硬件断点      | 1 个                                |
| 软件断点      | 多个                                 |
| RAM 监视    | 支持                                 |
| 调试所用的功能引脚 | RxD, TxD                           |

### 7.2.2 系统配置

图 6-5 所示为片上调试的系统配置。

图 7-5. 片上调试的系统配置



- <1> 主机  
要求有 USB 接口。
- <2> 软件  
包含 CubeSuite+ for China, USB 驱动等。
- <3> USB 线缆 (附件)
- <4> EZ-CUBE (本产品)
- <5> 8 引脚目标线缆 (附件)

### 7.2.3 系统启动顺序

本节描述了系统启动的顺序。请注意按次序操作。

#### (1) 软件的准备和安装

以下软件是片上调试所必须的。关于软件的准备和安装的详细信息，请参考 EZ-CUBE 附带的安装手册。

- CubeSuite+
- USB 驱动

#### (2) 开关设置

SW-1 开关：请选择 "M1"

SW-2 开关：请根据实际目标设备使用选择

SW-3 开关：请选择 " Debug Mode"

SW-4 开关：请根据实际目标设备使用选择

SW-5 开关：请选择 "Other"

**注意事项 1.** 在 USB 线缆处于连接状态时，请勿改变开关的设置。

2. 最大允许电流 100mA，所以请勿将 EZ-CUBE 连接到电流很大的目标系统上。在 EZ-CUBE 与主机连通后，主机始终保持向 EZ-CUBE 供电。

#### (3) 连接目标系统

EZ-CUBE 连接到目标系统，此时**请确保目标系统的电源处于关闭状态。**

#### (4) 连接 USB 线缆

将 EZ-CUBE 与主机相连接，**连接时请务必保持目标系统电源关闭。**

当 USB 缆与 PC 机间连接时，模式灯点亮为红色。

#### (5) 目标系统上电

打开目标系统的电源。如果电源选择为“5”，则这一步并不是必需的。

#### (6) 启动调试器

打开调试器。

此步骤及之后的操作，请参考 CubeSuite+的用户手册。

如果调试器无法正常开启或者操作不稳定，可能是由以下问题引发的。

- EZ-CUBE 和目标系统之间通信错误

固件是否更新；

开关是否选择正确；

目标连线是否正确。

- 未预留用户资源或者未设置安全 ID

使用 EZ-CUBE 进行调试时，必须预留出一部分用户资源作为调试监控区域和串行通信接口区域，安全 ID 也必须设置。详细信息请参考 **7.2.5 用户资源的保留和设置安全 ID。**

- 使用了不支持的软件（调试器，设备文件或者固件）

所使用的软件可能不支持目标设备的调试。

- EZ-CUBE 损坏

EZ-CUBE 可能被损坏。

## 7.2.4 系统关闭顺序

请按照下列操作顺序停止调试并关闭系统。

如果没有按照下列顺序操作，目标系统或者 EZ-CUBE 可能被损坏。

### (1) 停止调试器

停止调试器的运行并断开调试器连接。

### (2) 关闭目标系统电源

关闭目标系统的电源。如果电源选择开关放在“5”的位置，则无需本步操作。

### (3) 拔除 USB 线缆

从 EZ-CUBE 或主机上拔除 USB 线缆。

### (4) 拔除目标线缆

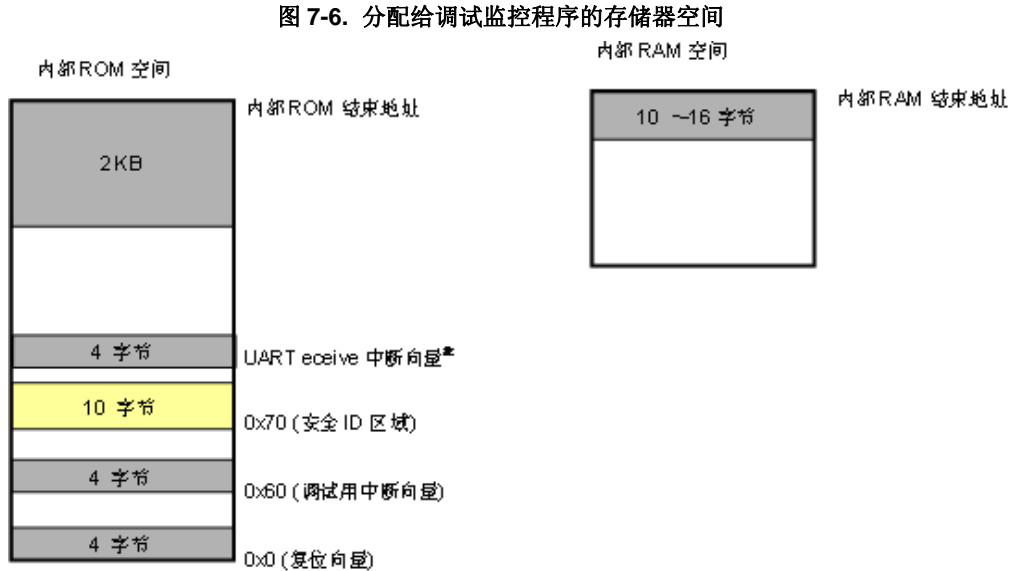
从 EZ-CUBE 或目标系统上将目标线缆拔下。

## 7.2.5 保留用户资源和设置安全 ID

为实现目标设备与 EZ-CUBE 之间的通信以及各种调试功能，用户必须做以下几点准备。请参照本小节的内容修改用户程序或编译选项，以完成准备工作。

- 预留给调试器空间

图7-6中的阴影部分是存放调试监控程序的保留区域，因此，用户程序和数据不能分配到这些空间。



■：调试区域

注： 如果存在接收错误中断的向量或接收状态中断的向量，必须保证是安全的。

- 通讯用的串行接口保留

不能改变调试监控程序执行时，用作 EZ-CUBE 和目标设备间通讯的 UART 的状态。

- 设置安全 ID

图 7-6 中涂黄色的区域中（0x70 到 0x79）必须写入安全 ID 代码，以防止存储器内容被他人盗读。

(a) 复位向量

复位向量中保存着向调试监控程序跳转的跳转指令信息。

[如何预留区域]

如果用户程序不使用这块区域就没有必要一定要保留这块区域。下载程序时，调试器会自动以以下几种形式将复位向量改写。如果改写的形式与以下几种形式不符，那么调试器将产生一个错误警告。

- 从地址 0 开始连续放置两条 nop 指令

|          |   |                   |
|----------|---|-------------------|
| 写入之前     |   | 写入之后              |
| 0x0 nop  | → | 跳至 0x0 地址上的调试监控程序 |
| 0x2 nop  |   | 0x4 xxxx          |
| 0x4 xxxx |   |                   |

- 从地址 0 开始连续放置两条 0xFFFF 数据（被擦除过的设备）

|            |   |                   |
|------------|---|-------------------|
| 写入之前       |   | 写入之后              |
| 0x0 0xFFFF | → | 跳至 0x0 地址上的调试监控程序 |
| 0x2 0xFFFF |   | 0x4 xxxx          |
| 0x4 xxxx   |   |                   |

- 地址 0 处放置 jr 指令（当使用 Renesas Electronics 的编译器时）

|               |   |                   |
|---------------|---|-------------------|
| 写入之前          |   | 写入之后              |
| 0x0 jr disp22 | → | 跳至 0x0 地址上的调试监控程序 |
|               |   | 0x4 jr disp22 - 4 |

- 地址 0 处放置向调试监控程序跳转的指令

|                   |   |      |
|-------------------|---|------|
| 写入之前              |   | 写入之后 |
| 跳至 0x0 地址上的调试监控程序 | → | 不变化  |



## (b) 调试监控程序的保留区域

在目标设备的内部 ROM 中，图 7-6 的阴影部分是分配给调试监控程序的区域。调试监控程序执行对调试通讯接口的初始化处理和对 CPU 的运行或中断处理。内部 ROM 区域必须填满 0xFF。这个区域千万不要被用户程序重写。

## [如何预留区域]

如果用户程序并未使用这一区域，那么用户也不必特意进行预留操作。

但是为了确保调试器启动时不会产生问题，我们还是推荐用户使用编译器将此区域预留。

下面是使用 Renesas Electronics 编译器预留该方法举例。请在工程中加入如下所示的汇编语言源文件和链接指示代码。

- 汇编语言源文件（将以下代码写入汇编源文件中并加入到工程）

```
--为监控 ROM 节预留 2 KB 空间
.section "MonitorROM", const
.space 0x800, 0xff

--预留调试用的中断向量
.section "DBG0"
.space 4, 0xff

--预留串行通信用的中断向量
--请根据所使用的串行通信模式更改节名称
.section "INTCB0R"
.space 4, 0xff

--为监控 RAM 节预留 16 字节的空间
.section "MonitorRAM", bss
.lcomm monitorramsym, 16, 4 /* 定义 monitorramsym 符号 */
```

注 如果用语句“monitorramsym:”替换该行，仅执行一个符号定义，则下载速度会被提升。如果数值被填入一个洞（没有编码的区域），该效果不再显现。填入时，填入值必须是 0xFF 用于预留空间。

- 链接伪指令（将以下代码加入链接指示文件中）

下面的代码是以内部 ROM 结尾地址为 0xffff、内部 RAM 结尾地址为 0x3ffff 为例。

```
MROMSEG : !LOAD ?R V0x0ff800{
  MonitorROM = $PROGBITS ?A MonitorROM;
};

MRAMSEG : !LOAD ?RW V0x03ffff0{
  MonitorRAM = $NOBITS ?AW MonitorRAM;
};
```

(c) 预留串行通信接口

UART 用作 EZ-CUBE 和目标系统间的通讯。和 UART 相关的设置通过调试监控程序来执行，但是如果这个设置被用户程序改变，会产生通讯错误。

为了防止这种问题的发生，在用户程序里通讯串行接口必须保留。

[如何对串行通信接口进行预留]

请参考以下内容创建用户程序。

- 串行接口寄存器

不要在用户程序中对 UART 的相关寄存器进行设置。

- 中断屏蔽寄存器

使用 UART 时，不要屏蔽接收完成中断<sup>※</sup>。

**注** 不要屏蔽接收错误中断或者接收状态中断。

- 端口寄存器

使用 UART 时，不要将端口寄存器设置为 TxD 和 RxD 以外的模式。

(d) 安全 ID 的设置

该功能用于防止存储器被盗读。内部 Flash 的地址 0x70 到 0x79 上会嵌入一个安全 ID。只有在调试器启动时输入的安全 ID 与内部 Flash 中地址 0x70 到 0x79 的安全 ID 一致时，调试器才可以正常启动。

但是，如果地址 0x79 的第 7 位的值为“0”，那么调试功能将被禁止使用。这种情况下，调试器将无法启动。对于量产的设备而言，通常调试功能都是被禁止的。

如果用户忘记了安全 ID 的值或希望重新打开调试功能，那么只有将 Flash 全部擦除后再次设置安全 ID。

[如何设置安全 ID]

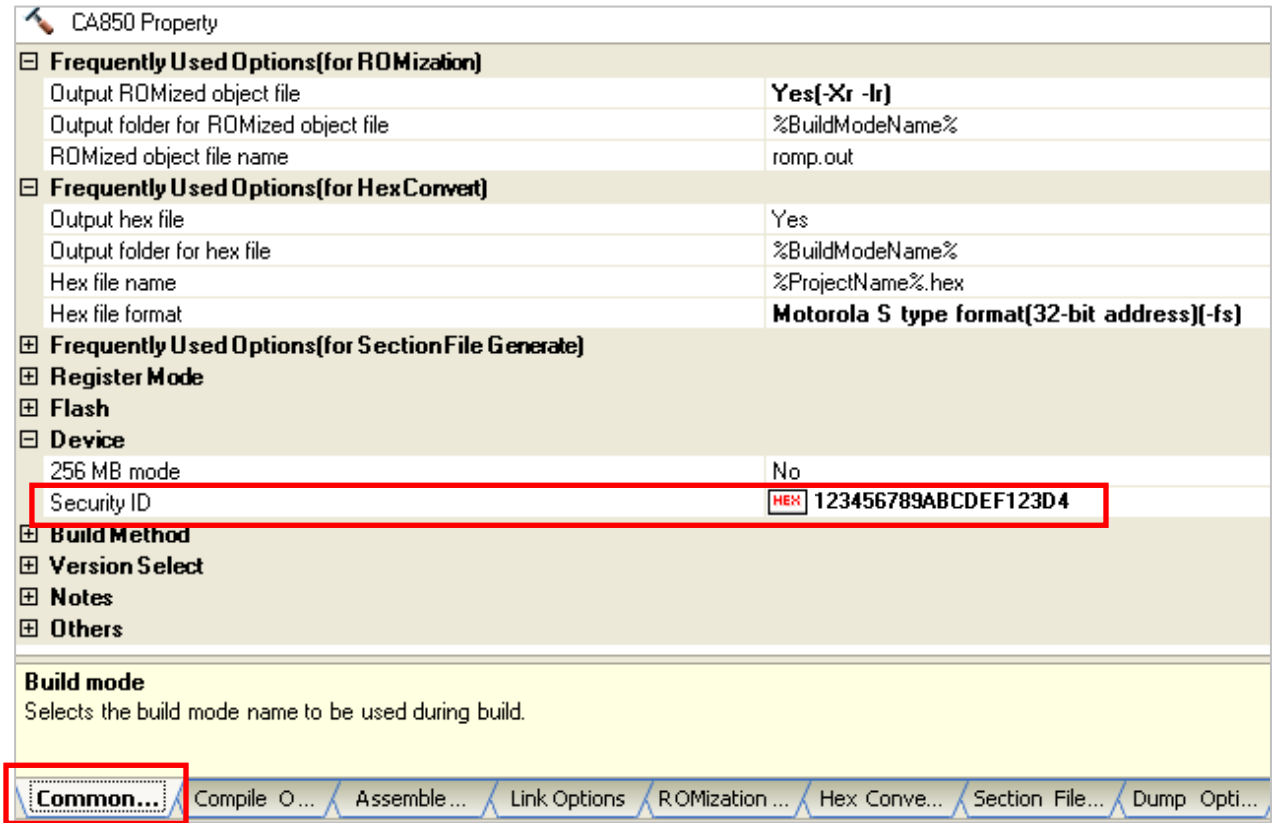
用户需要在用户程序中向地址 0x70 到 0x79 嵌入安全 ID。

如果嵌入的安全 ID 为“123456789ABCDEF123D4”，那么调试器中需要设置的安全 ID 也应该相同（不区分大小写）。

| 地址   | 数值 [7:0] |
|------|----------|
| 0x70 | 0x12     |
| 0x71 | 0x34     |
| 0x72 | 0x56     |
| 0x73 | 0x78     |
| 0x74 | 0x9A     |
| 0x75 | 0xBC     |
| 0x76 | 0xDE     |
| 0x77 | 0xF1     |
| 0x78 | 0x23     |
| 0x79 | 0xD4     |

使用 Renesas Electronics 的编译器时，如图所示使用 CubeSuite+，在[Build Tool]的[Common Options]选项卡的“Device”处进行设置一个任意的 ID 码（10 字节）。

图7-7. 设置安全ID



## 7.2.6 调试时的注意事项

本节介绍使用 V850 微控制器的片上调试功能时需要注意的事项。

请仔细阅读以下事项以正确使用 EZ-CUBE。

### (1) 对用于调试的设备的处理

由于调试时设备内部的 Flash 会被多次擦写并且无法确保擦写的次数，因此，不要将用于调试的设备安装在量产的产品中。此外，也不要将调试监控程序编程至量产的产品中。

### (2) 当无法执行断点操作时

以下情况发生时，将无法执行强制断点功能。

- 禁止中断时 (DI)
- 用于 EZ-CUBE 与目标设备通信的串行接口中断被屏蔽
- 当设置了“禁止可屏蔽中断唤醒待机模式”时进入了待机模式
- EZ-CUBE 与目标设备的通信模式为 UART，但主时钟信号停止时

### (3) 调试器的启动会被减缓的设备

首次启动调试器时会进行芯片的擦写和调试监控程序的写入操作。在进行上述操作时需要十几秒的时间。

### (4) 写入调试监控程序

如果调试器改变了 CPU 的工作时钟设置，调试器会重新写入调试监控程序。此时，调试器需要十几秒的时间进行该操作。使用 Renesas Electronics 的调试器时，更改 CPU 工作时钟是在配置对话框中的时钟区域进行的。

(5) Flash 自编程

如果调试监控程序区域的代码被 Flash 自编程操作改写，那么调试器将无法继续正常工作。

(6) POC 功能仿真

目标设备的 POC 功能不能被仿真。要确保调试期间目标系统的电源不能被断掉。

(7) 复位后的操作

外部或内部复位发生后，监控程序会进行调试初始化操作。因此，这里复位至用户程序执行的时间与真实的设备操作时间是不同的。

(8) 使用实机调试而不使用 EZ-CUBE

如果使用实机调试而不使用 EZ-CUBE，请使用 RFP 写入用户程序。通过调试器下载的程序包括监控程序，如果不通过 EZ-CUBE 控制可能发生程序故障。

## 7.3 Flash 编程

本节将介绍使用 EZ-CUBE 对 V850 微控制器的 Flash 进行编程时的系统配置以及启动/关闭顺序。

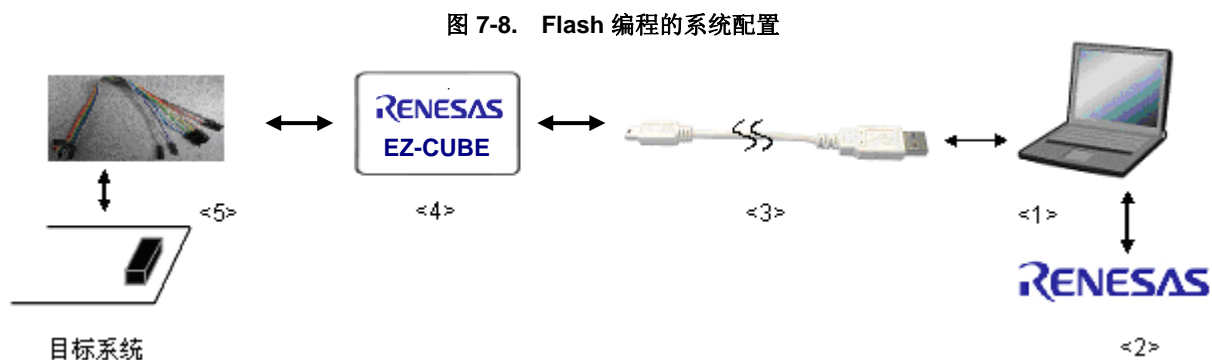
### 7.3.1 编程功能的规格定义

表7-4. 编程功能的规格定义

| 功能       | 规格定义说明                           |
|----------|----------------------------------|
| 主机接口     | USB 2.0                          |
| 目标接口     | UART                             |
| 目标系统电压   | 2.7到5.5 V（根据目标设备而定）              |
| 时钟源      | 可以提供8 MHz时钟信号<br>可以使用安装在目标系统上的时钟 |
| 电源       | 5 V $\pm$ 0.3V（最大电流：100 mA）      |
| 获取特定设备信息 | 使用Renesas Electronics的V850参数文件   |
| 安全标志的设置  | 有                                |
| 独立操作     | 无（必须连接主机）                        |

### 7.3.2 系统配置

图 7-8 显示了 Flash 编程时的系统配置。



<1> 主机

要求有 USB 接口

<2> 软件

包含 RFP（Renesas Flash Programmer），USB 驱动等。

<3> USB 线缆（附件）

<4> EZ-CUBE（本产品）

<5> 8 引脚目标连接线（附件）

### 7.3.3 系统启动顺序

本章将介绍系统的启动顺序。请按照以下顺序进行操作。

#### (1) 软件的准备和安装

以下软件是片上调试所必须的。有关软件的准备和安装的详细信息，请参考 EZ-CUBE 附带的安装手册。

- RFP
- USB 驱动

#### (2) 开关设置

SW-1 开关：请选择 "M1"

SW-2 开关：请根据实际目标设备使用选择

SW-3 开关：请选择 " Debug Mode"

SW-4 开关：请根据实际目标设备使用选择

SW-5 开关：请选择 "Other"

**注意事项**            在 USB 线缆处于连接状态时，请勿改变开关的设置。

#### (3) 连接目标系统

将 EZ-CUBE 连接到目标系统，请务必在目标系统上电之前连接好连线。

#### (4) 连接 USB 线缆

将 EZ-CUBE 与主机相连接，连接时请务必保持目标系统电源关闭。

#### (5) 目标系统上电

打开目标系统的电源。如果电源选择为“5”，则这一步并不是必需的。

#### (6) 启动 RFP (Renesas Flash Programmer)

打开 RFP 软件。

### 7.3.4 应用样例

本节描述了使用RFP（Renesas Flash Programmer）的一系列基本操作，使用UPD70F3717 作为示例设备。描述了执行[Autoprocedure (EP)]命令为目标设备编程的整个过程。命令的详细内容，请参加RFP的用户手册。

本节中操作举例的条件如下。

<目标系统>

目标设备: UPD70F3717  
电源电压: 3.3V

<EZ-CUBE>

开关: SW-1 开关: M1  
SW-2 开关: Int. Clock  
SW-3 开关: Debug Mode  
SW-4 开关: T  
SW-5 开关: Other

<RFP>

时钟设置: 5 MHz, x4  
操作模式: 芯片  
程序文件: sample.hex  
命令参数: 选中[Blank check before Erase]  
选中[Verify after Program]

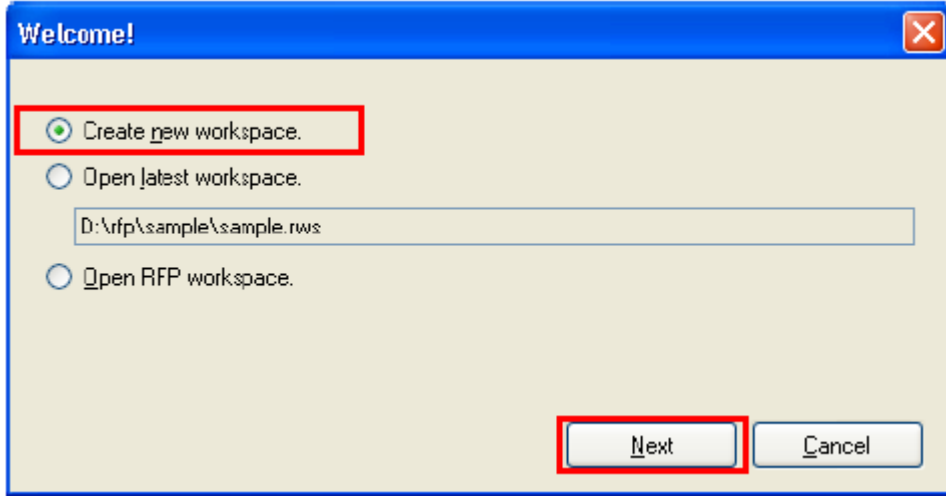
(1) 设置编程环境

请按照下述<1>到<6>的步骤进行编程环境的设置。

<1> 点击【开始】按钮，指向[程序]，[Renesas Electronics CubeSuite+]，[Programming Tools]，[Renesas Flash Programmer Vx.xx]，点击[Renesas Flash Programmer Vx.xx [Basic mode]]，启动 RFP 软件。

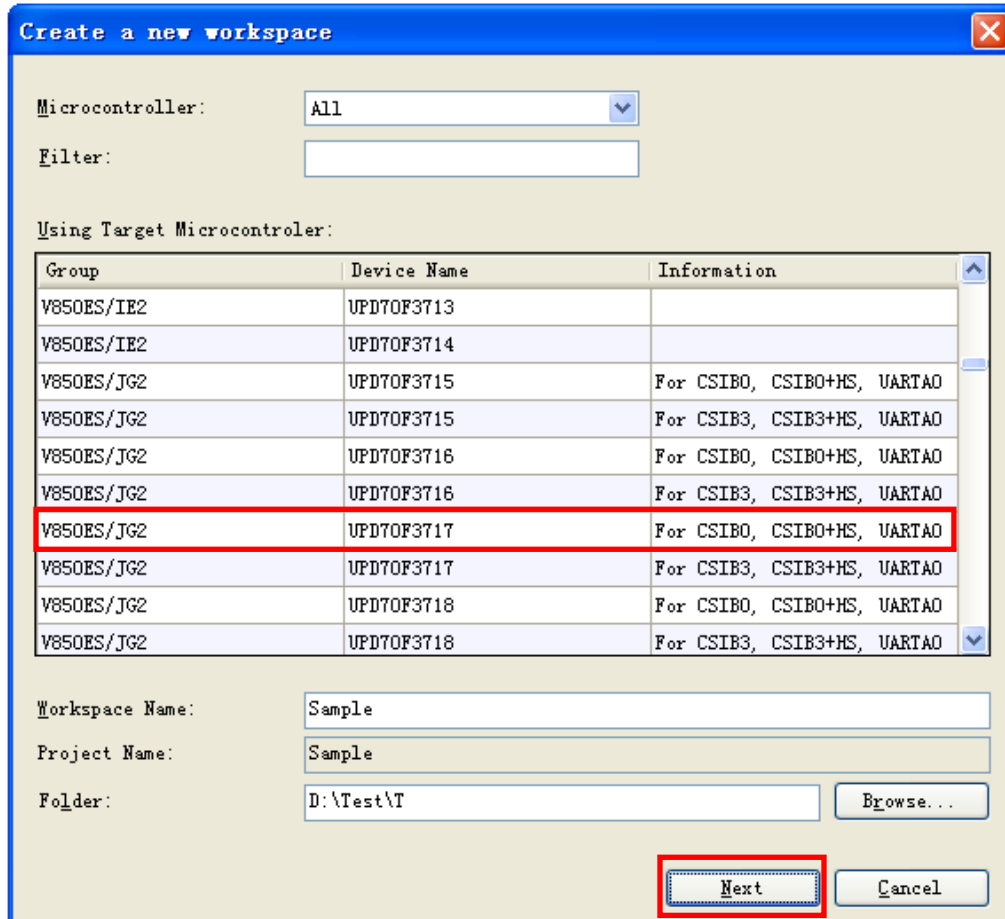
<2> 创建[ Create new workspace. ]

图 7-9 创建



<3> 选择编程芯片，如 70F3717

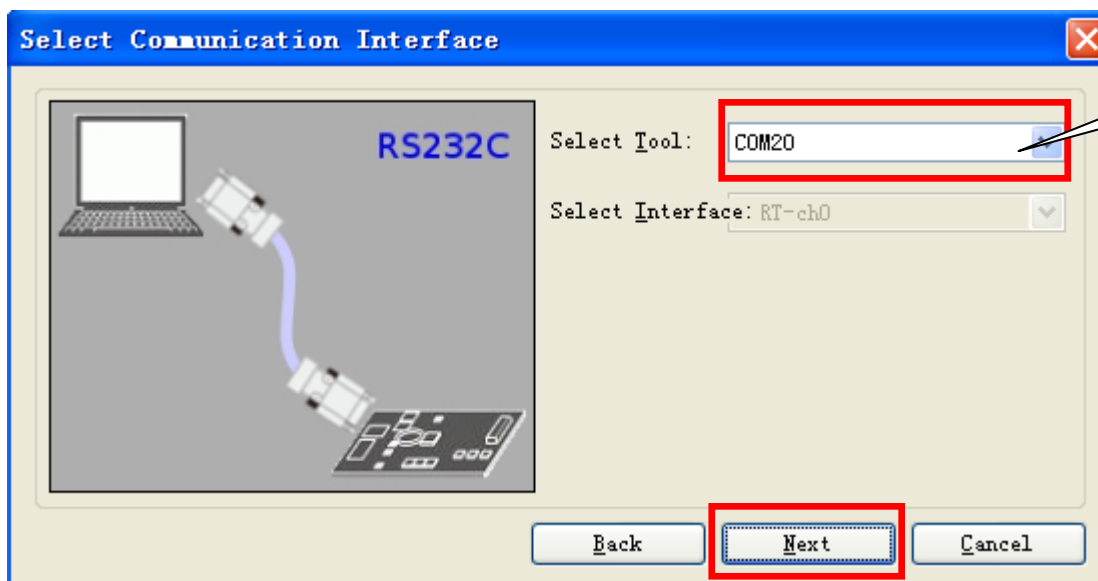
图 7-10 选择编程芯片





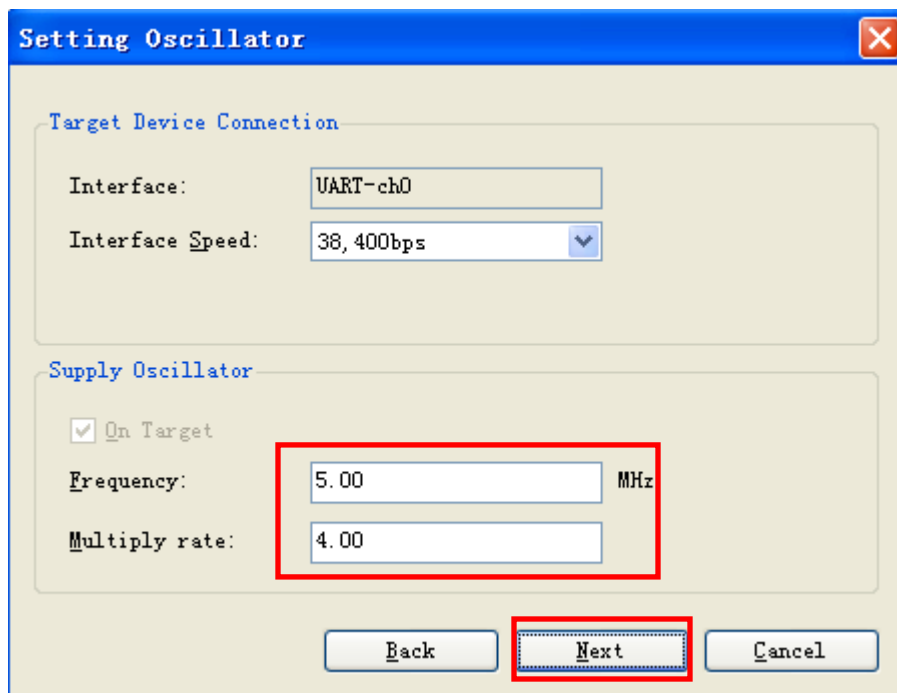
<4> 选择通讯接口[ Select Communication Interface ], 请选择实际的编程接口, 这里为 COM20。

图6-11. Select Communication Interface 选项卡



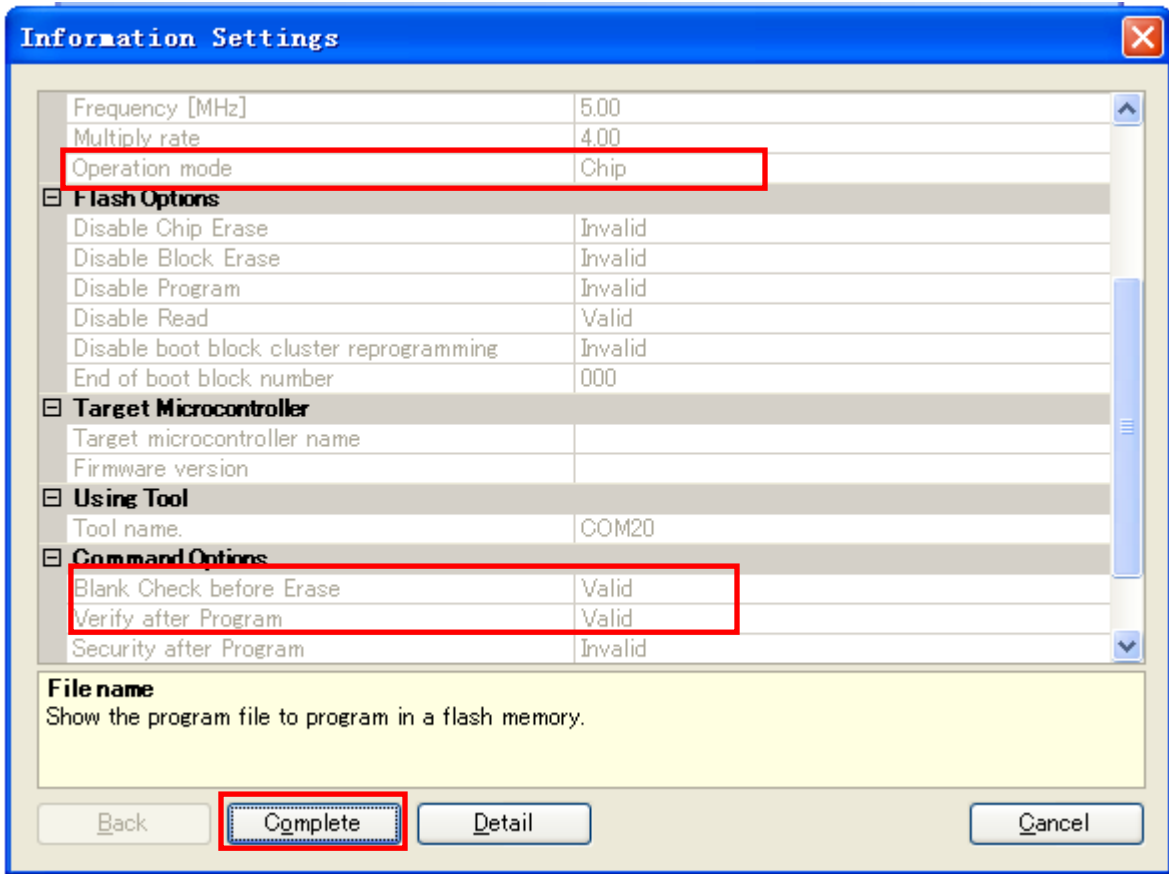
<5> 设置时钟。

图7-13. 设置 [Setting Oscillator]选项卡



<6> 根据编程环境设置[Information Setting]选项卡的各项选项。这里设置[Target]和[Command Options]中的选项。详细内容, 请参考 RFP 的用户手册。点击确定按钮, 完成编程环境的设置。

图7-14. 设置 [Information Settings]选项卡



## (2) 程序文件的选择

请按照下述的步骤<1>至<3>选择程序文件。

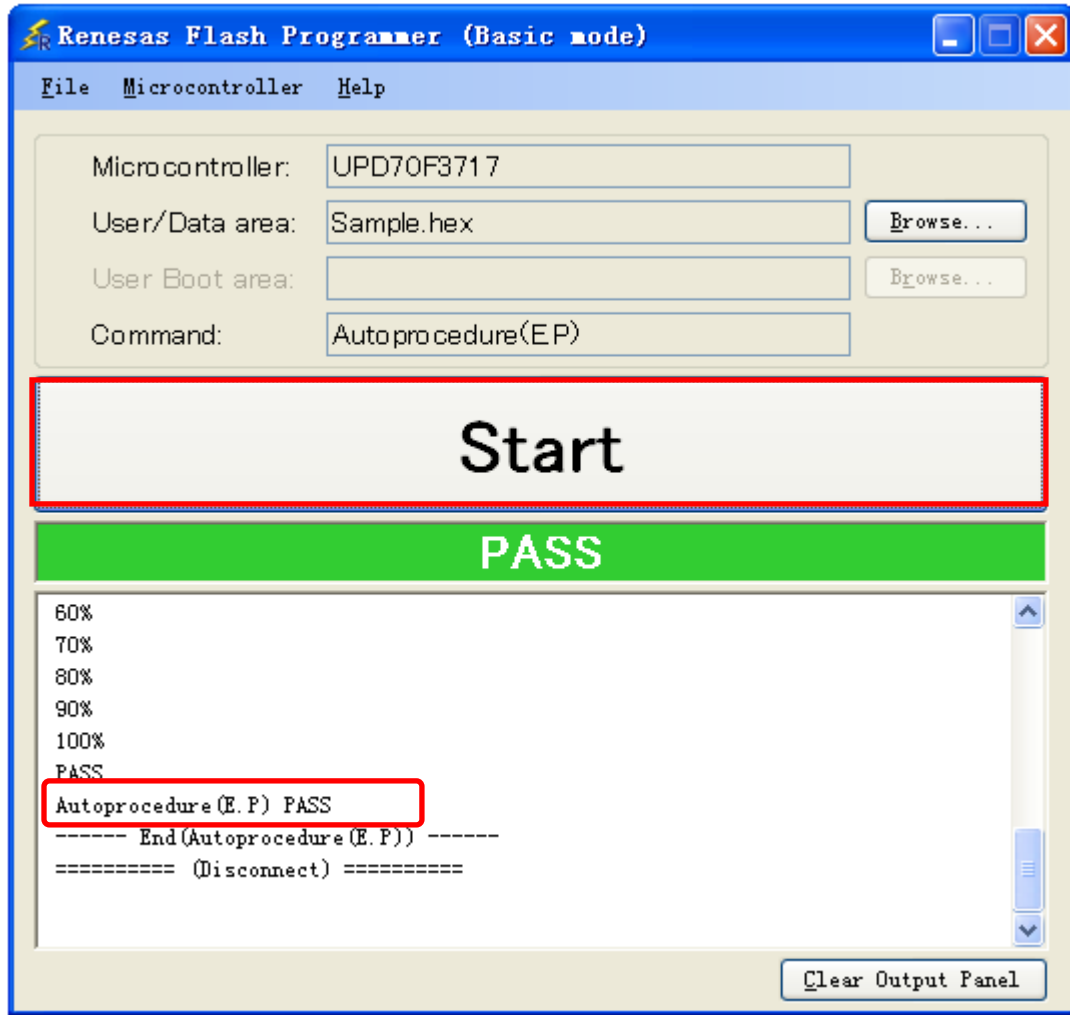
- <1> 在[User/Data area]区域点击 **Browse...**按钮。
- <2> 弹出程序文件选择对话框。
- <3> 选择要写入目标设备的程序文件，然后点击**打开**按钮。

## (3) 执行[Autoprocedure(EP)]命令

- <1> 点击菜单栏的[Microcontroller]，然后选择[Autoprocedure(EP)]命令。
- <2> 点击 **Start** 按钮

[Blank Check]、[Erase]（如果目标设备上的Flash 存储器非空）、[Program]、[Verify]命令将会在目标设备上依次执行。当本命令正常执行完成后，弹出如下窗口。

图7-15. [Autoprocedure (EP)]命令执行完成



### 7.3.5 系统关闭顺序

请按以下步骤结束 Flash 的编程并关闭系统。

如果未按照以下顺序进行操作，那么 EZ-CUBE 将有可能被损坏。

(1) **终止 RFP 软件**

如果你不继续对其他设备进行编程，则终止 RFP。

(2) **关闭目标系统的电源**

关闭目标系统的电源。如果电源选择开关放在“5”的位置，则无需本步操作。

(3) **移除 USB 线缆**

从 EZ-CUBE 或主机上拔除 USB 线缆。

电源选择开关设置为“T”时，(4) **移除目标线缆**的操作可以在本操作之前进行。

(4) **移除目标线缆**

从 EZ-CUBE 或目标系统上拔除目标线缆。

### 7.3.6 Flash 编程时的注意事项

本节将介绍进行 Flash 编程时需要注意的事项。请务必仔细阅读本节内容以正确使用 EZ-CUBE。

- 为提高写入的编程的质量，请在使用 EZ-CUBE 之前了解以下内容并队列出的各项进行校验和评估。
  - 务必按照设备及 EZ-CUBE 的用户手册中给出的方法进行目标系统的电路的设计。
  - 务必按照设备、RFP 及 EZ-CUBE 各自的用户手册中介绍的方法进行操作。
  - 务必向目标系统提供稳定的电源。