RENESAS

QB-MINI2

On-Chip Debug Emulator with Programming

Function   User's Manual

Target Devices

V850 Microcontrollers

78K0R Microcontrollers

78K0 Microcontrollers

78K0S Microcontrollers

Renesas Electronics
www.renesas.com

Rev.6.00   Mar, 2011

General Precautions for Handling This Product

1. **Circumstances not covered by product guarantee**
   - If the product was disassembled, altered, or repaired by the customer
   - If it was dropped, broken, or given another strong shock
   - Use at overvoltage, use outside guaranteed temperature range, storing outside guaranteed temperature range
   - If power was turned on while the USB cable or connection to the target system was in an unsatisfactory state
   - If the cable of the USB cable, the target cable, or the like was bent or pulled excessively
   - If the product got wet
   - If this product is connected to the target system when there is a potential difference between the GND of this product and GND of the target system.
   - If the connectors or cables are plugged/unplugged while this product is in the power-on state.
   - If excessive load is applied to the connectors or sockets.
   - If a metal part of the interface connector or another such part comes in contact with an electrostatic charge
   - If specifications other than those specified are used for the USB port

2. **Safety precautions**
   - The product may become hot depending on the status of use.   Be careful of low temperature burns and other dangers due to the product becoming hot.
   - Be careful of electrical shock.   There is a danger of electrical shock if the product is used as described above in **1 Circumstances not covered by product guarantee**.
   - Do not use a device that was used for on-chip debugging as the official product.

How to Use This Manual

Readers This manual is intended for users who wish to perform debugging using the QB-MINI2 (generic name: MINICUBE®2).

The readers of this manual are assumed to be familiar with the device functions and usage, and to have knowledge of debuggers and flash programming.

Purpose This manual is intended to give users an understanding of the basic specifications and correct use of MINICUBE2.

Organization This manual is divided into the following sections.

- General
- Names and functions of hardware
- Self-testing
- Firmware update

How to Read This Manual It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

This manual describes the basic setup procedures.

To understand the basic specifications and usages of MINICUBE2

→ Read this manual according to the **CONTENTS**. The mark <R> shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.

To know the manipulations, command functions, and other software-related settings of MINICUBE2

→ See the user's manual of the debugger to be used.

Conventions

Note:	Footnote for item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numeric representation:	Binary ... xxxx or xxxxB
	Decimal ... xxxx
	Hexadecimal ... xxxxH
Prefix indicating power of 2 (address space, memory capacity):	K (kilo): $2^{10} = 1,024$
	M (mega): $2^{20} = 1,024^2$

Terminology   The meanings of the terms used in this manual are described in the table below.

Term	Meaning
MINICUBE2	Generic name of QB-MINI2
Target device	This is the device to be emulated.
Target system	This is the system to be debugged (user-created system). It includes software and hardware created by the user.
OCD unit	OCD is an abbreviation of On-Chip Debug The OCD unit means the debug function block embedded in the target device.
Firmware	Program embedded in the device for controlling MINICUBE2
QB-Programmer	GUI software used to perform flash programming

Related Documents   Please use the following documents in conjunction with this manual.

The related documents listed below may include preliminary versions. However, preliminary versions are not marked as such.

Documents Related to Development Tools (User's Manuals)

Document Name		Document Number
QB-MINI2 On-Chip Debug Emulator with Programming Function		This document
QB-MINI2 Setup Manual	Partner Tool	U19158E
ID850QB Ver. 3.40 Integrated Debugger	Operation	U18604E
ID78K0-QB Ver. 3.10 Integrated Debugger	Operation	U19611E
ID78K0S-QB Ver. 2.90 Integrated Debugger	Operation	U18247E
ID78K0R-QB Ver. 3.20 Integrated Debugger	Operation	U17839E
QB-Programmer Programming GUI Operation		U18527E
MINICUBE2 Diagnostic Tool		U18588E
MINICUBE OCD Checker		U18591E

`<R>` ID850QB Ver. 3.40 Integrated Debugger
`<R>` ID78K0-QB Ver. 3.10 Integrated Debugger
`<R>` ID78K0S-QB Ver. 2.90 Integrated Debugger

Caution   The related documents listed above are subject to change without notice.   Be sure to use the latest version of each document for designing, etc.

Documents Related to Development Tools (Documents except User's Manuals)

Document Name	Document Number
QB-MINI2 Operating Precautions	ZUD-CD-07-1212-E

Caution   The related documents listed above are subject to change without notice.   Be sure to use the latest version of each document for designing, etc.

IECUBE is a registered trademark of Renesas Electronics Corporation in Japan and Germany.

Windows is either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

PC/AT is a trademark of International Business Machines Corporation.

CONTENTS

CHAPTER 1 GENERAL

MINICUBE2 is an on-chip debug emulator with flash programming function, which is used for debugging and programming a program to be embedded in on-chip flash memory microcontrollers.

1.1 Features

- On-chip debugging

  Can debug with the target microcontroller connected to the target system.

- Flash memory programming

  Can write programs to the on-chip flash memory of microcontrollers.

- USB connection

  Can be connected to the host machine via USB interface 2.0 or 1.1.

  Since MINICUBE2 operates on power supplied via USB, an external power supply is unnecessary.

- Variety of supported devices and expandability

  MINICUBE2 supports a wide variety of Renesas Electronics 8- to 32-bit on-chip flash memory

  microcontrollers. More devices can be supported by upgrading firmware embedded in MINICUBE2.

- Self-testing function

  MINICUBE2 has a self-testing function, which is used to detect faults in MINICUBE2 and perform

  maintenance.

1.2 Notes Before Using MINICUBE2

Chapters 1 and 2 present an overview and the basic specifications of MINICUBE2, and the following chapters provide separate descriptions for the target devices and the purpose of use. To utilize this manual effectively, refer to the following table and see the relevant chapter for your target device and purpose of use.

Table 1-1. Chapters Corresponding to Usage

Target Device	Purpose of Use	Relevant Chapter
V850		CHAPTER 3 HOW TO USE MINICUBE2 WITH V850 MICROCONTROLLER
	Target system design	3.1 Target System Design
	On-chip debugging	3.2 On-Chip Debugging
	Flash memory programming	3.3 Flash Programming
78K0		CHAPTER 4 HOW TO USE MINICUBE2 WITH 78K0 MICROCONTROLLER
	Target system design	4.1 Target System Design
	On-chip debugging	4.2 On-Chip Debugging
	Flash memory programming	4.3 Flash Programming
78K0S		CHAPTER 5 HOW TO USE MINICUBE2 WITH 78K0S MICROCONTROLLER
	Target system design	5.1 Target System Design
	On-chip debugging	5.2 On-Chip Debugging
	Flash memory programming	5.3 Flash Programming
78K0R		CHAPTER 6 HOW TO USE MINICUBE2 WITH 78K0R MICROCONTROLLER
	Target system design	6.1 Target System Design
	On-chip debugging	6.2 On-Chip Debugging
	Flash memory programming	6.3 Flash Programming
-	MINICUBE2 self-testing	CHAPTER 7 SELF-TESTING
-	Expansion of supported devices	CHAPTER 8 FIRMWARE UPDATE

1.3 Supported Devices

Access the following MINICUBE2 information site or refer to the document for details on the devices that support debugging and flash programming with MINICUBE2. The document describes the development tools that are required as the support conditions, serial interface and operating precautions, so be sure to read the document, in addition to this manual.

- MINICUBE2 information site

  URL: http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html

- Document

  Document name: QB-MINI2 Operating Precautions

  Remark The above-mentioned document is posted on the MINICUBE2 information Web site.

1.4 Hardware Specifications

This section describes the MINICUBE2 hardware specifications.

The specifications related to the on-chip debug and flash memory programming functions are described in the following chapters.

Table 1-2. Hardware Specifications

Classification	Item	Specifications
MINICUBE2	Operating power supply	Supplied via USB interface (5 V)
	Operating environment conditions	Temperature: ±0 to +40°C Humidity: 10 to 80% RH (no condensation)
	Storage environment conditions	Temperature: -15 to +60°C Humidity: 10 to 80% RH (no condensation)
	External dimensions	48 × 48 × 13.9 mm
	Weight	Approximately 40 g
Host machine interface	Target host machine	PC98-NX Series, IBM PC/AT^{TM} compatibles
	Target OS	Windows XP, Windows Vista, Windows7
	USB	2.0 (1.1 compatible)
	USB cable	2 m
	Current consumption	500 mA max.
Target interface	Target cable length	16-pin cable: 10 cm 10-pin cable: 10 cm
	Supported connector	2.54 mm pitch general-purpose connector
	Supply voltage	3.1 V (typ.) or 5.0 V (typ.)
	Supply current	100 mA max.
	Current consumption	8 mA (typ.)
	Voltage range	1.65 to 5.5 V

(Note: the `<R>` marks appear in the left margin beside "Host machine interface" row and beside the Note below.)

<R> Note   When the power is supplied from MINICUBE2 to the target system, the host machine is required to have 500 mA current supply capability.

RENESAS

CHAPTER 2   NAMES AND FUNCTIONS OF HARDWARE

This chapter describes the part names and functions of MINICUBE2 and its accessories.

The part names described in this chapter are used throughout this document. This chapter provides an overview of the various functions. Reading it through, the reader will gain a basic grasp of MINICUBE2 and the self-check board that will facilitate reading of subsequent chapters. While reading this chapter, also check if the hardware has a defect.

2.1 Names of Supplied Hardware

Figure 2-1 shows the names of hardware supplied with MINICUBE2.

Figure 2-1. Names of Supplied Hardware

MINICUBE2

USB cable

16-pin target cable

10-pin target cable

78K0-OCD board

(A 20 MHz oscillator mounted)
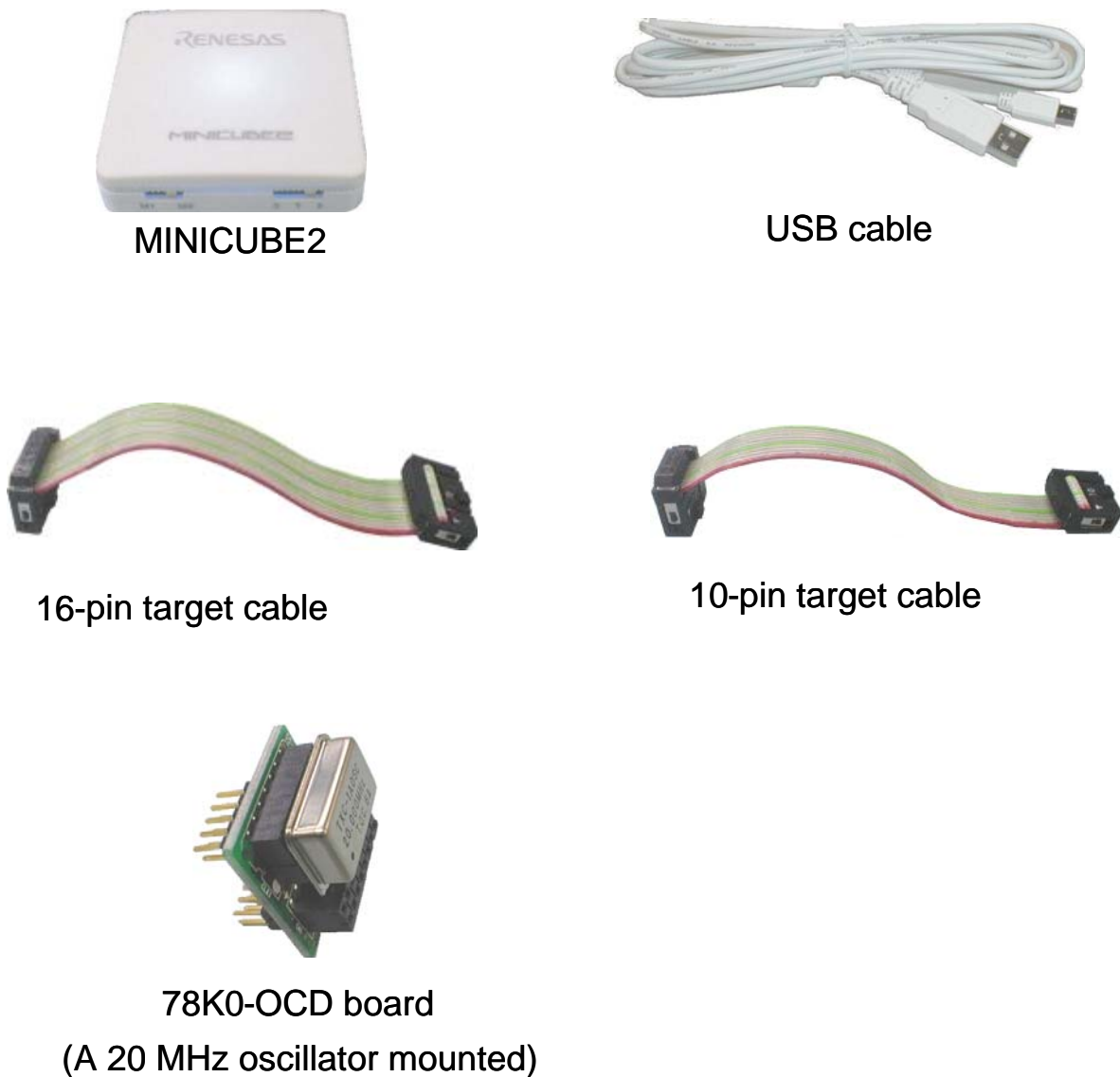
2.2 Part Names and Functions of MINICUBE2

Figure 2-2 shows the part names of the MINICUBE2 main unit. For their functions, refer to (1) to (5) below.

Figure 2-2. Part Names of MINICUBE2
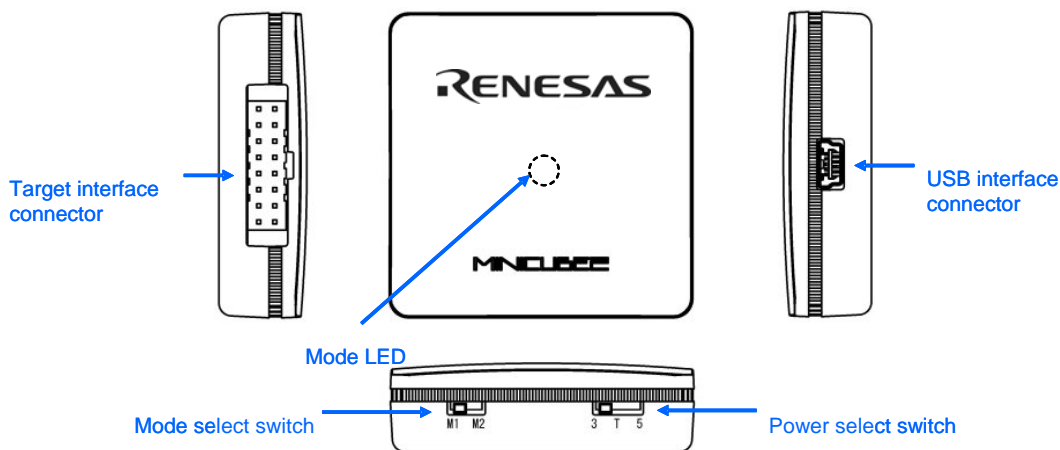
(1) Mode select switch
This switch is used to select the target device. Table 2-1 describes the setting details. This switch is set to ″M2″ at shipment.

Table 2-1. Description of Mode Select Switch

Setting	Description
M1	Sets 78K0S or 78K0R microcontroller as target device
M2	Sets V850 or 78K0 microcontroller as target device (factory setting)

(2) Power select switch
This switch is used to set the power supplied to the target system. Table 2-2 describes the setting details. This switch is set to ″3″ at shipment.

Caution Do not change the switch setting while the USB cable is connected.

Table 2-2. Setting of Power Select Switch

Setting	Description
3	3 V is supplied from MINICUBE2 to the target system[Note]. (Factory setting) The supplied power is fed back to MINICUBE2 and used only for power detection.
5	5 V is supplied from MINICUBE2 to the target system[Note]. The supplied power is fed back to MINICUBE2 and used only for power detection.
T	Power supply of the target system is used. MINICUBE2 only detects the power for the target system.

Note   The maximum rating of the current is 100mA, so do not use MINICUBE2 with the target system with the higher current rating. The power is always supplied after MINICUBE2 is connected to the host machine.

(3) USB interface connector

This is a connector used to connect MINICUBE2 with the host machine, via a USB cable.

A USB 2.0 compliant mini-B connector (UX60A-MB-5ST: Made by Hirose Electric Co., Ltd.) is employed.

(4) Target interface connector

This is a connector used to connect MINICUBE2 with the target system, via a 16-pin target cable. Connect the 78K0-OCD board for debugging a 78K0 microcontroller. To prevent the 78K0-OCD board from being inserted by mistake, pin 11 is removed at shipment.

Since this connector is compatible with 16-core 2.54 mm pitch general-purpose connectors (female), a commercially available cable can be substituted.

(5) Mode LED

The appearance of the mode LED changes according to the status of hardware and software, as shown in Table 2-3.

Table 2-3. Mode LED Status

Mode LED Color	Appearance	Description		
		USB Connection	Target VDD Detection	Software Operation Status
-	Extinguished	Not connected	Not detected	Not yet started
White	Flashing	Connected	Not detected	Not yet started
	Glowing		Detected	Not yet started
Blue	Glowing			A debugger for V850 microcontrollers has been started and CPU is in the break mode.
	Flashing			A debugger for V850 microcontrollers has been started and the CPU is running, or a program is being downloaded.
Orange	Glowing			A debugger for 78K0 microcontrollers has been started and the CPU is in the break mode.
	Flashing			A debugger for 78K0 microcontrollers has been started and the CPU is running, or a program is being downloaded.
Green	Glowing			A debugger for 78K0S microcontrollers has been started and the CPU is in the break mode. The QB-Programmer has been started and is waiting for execution processing. Alternatively, execution processing has been completed.
	Flashing			A debugger for 78K0S microcontrollers has been started and the CPU is running, or a program is being downloaded.
Yellow	Glowing			A debugger for 78K0R microcontrollers has been started and the CPU is in the break mode.
	Flashing			A debugger for 78K0R microcontrollers has been started and the CPU is running, or a program is being downloaded. A command from the QB-Programmer is being executed for the flash memory.
Red	Glowing			A command from the QB-Programmer is ended abnormally (results in error)
Illumination		The illumination mode is entered when MINICUBE2 is connected to a USB interface and 15 seconds or longer have elapsed. This mode is also entered when the debugger is terminated normally.		

2.3 Part Names and Functions of 78K0-OCD Board

The 78K0-OCD board is used for debugging a 78K0 microcontroller (not used for flash programming). Figure 2-3 illustrates the external view of the 78K0-OCD board. The name of each part is printed on the 78K0-OCD board. For each function, refer to **(1)** to **(4)**, below.

Figure 2-3. Part Names of 78K0-OCD Board
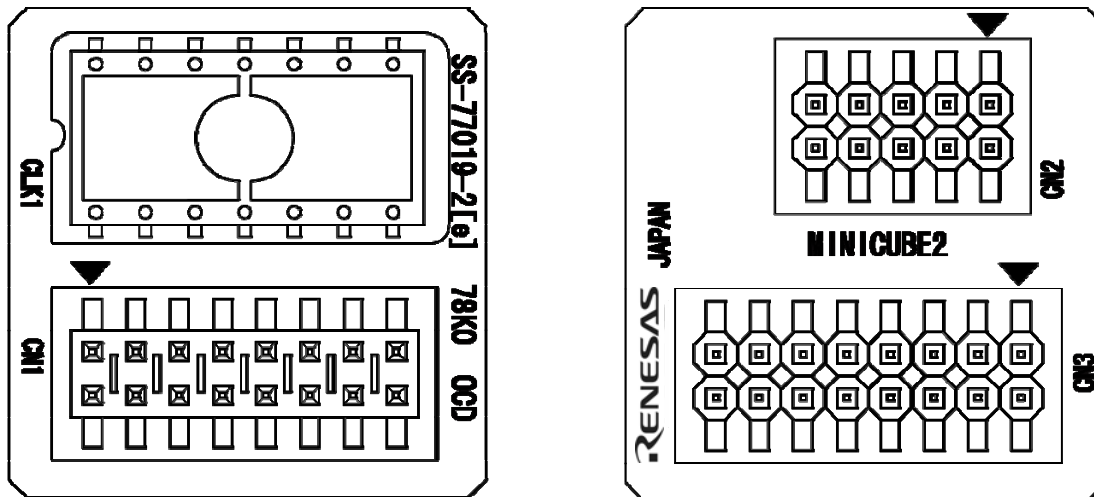
(1) CLK1

For clock supply to the target device, a 14-pin DIP socket is implemented as CLK1. A 14-pin type oscillator (5 V) or a parts board that can configure an oscillation circuit can be mounted on the 14-pin DIP socket. A parts board such as the 160-90-314 (manufactured by PRECI-DIP) can be used with this socket to configure oscillation circuit.

(2) CN1

This is a connector used to connect MINICUBE2 with the target interface connector of MINICUBE2. To prevent the 78K0-OCD board from being inserted by mistake, a dummy pin is mounted in the place of pin 11 in CN1.

(3) CN2

This is a connector used to connect MINICUBE2 with the target system, via a 10-pin target cable.
A 10-core 2.54 mm pitch general-purpose connector (TSM-105-01-L-DV) is employed.

(4) CN3

This is a connector used to connect MINICUBE2 with the target system, via a 16-pin target cable.
A 16-core 2.54 mm pitch general-purpose connector (TSM-108-01-L-DV) is employed.

CHAPTER 3   HOW TO USE MINICUBE2 WITH V850 MICROCONTROLLER

This chapter describes how to use MINICUBE2 when performing on-chip debugging and flash programming for a V850 microcontroller.

On-chip debugging is a method to debug a microcontroller mounted on the target system, using a debug function implemented in the device. Since debugging is performed with the target device operating on the board, this method is suitable for field debugging.

Flash programming is a method to write a program to the flash memory embedded in a device. Erasing, writing and verifying the program can be performed on-board with the device.

Read the following chapters if you are using MINICUBE2 for the first time with a V850 microcontroller as the target device.

- 3.1 Target System Design

  For communication between MINICUBE2 and the target system, communication circuits must be mounted on the target system. This section describes the circuit design and mounting of connectors.

- 3.2 On-Chip Debugging

  This section describes the system configuration and startup method to perform on-chip debugging with MINICUBE2.

- 3.3 Flash Programming

  This section describes the system configuration and startup method to perform flash programming with MINICUBE2.

3.1 Target System Design

This section describes the target system circuit design required for on-chip debugging and flash programming.

Figure 3-1 presents an overview of the MINICUBE2 communication interface. As shown on the left side of the figure, MINICUBE2 performs serial communication with the target device on the target system. For this communication, communication circuits must be mounted on the target system. Refer to this section to design circuits appropriately.

UART and CSI-H/S are supported as communication modes. Note that CSI-H/S is not available if there is no H/S pin connection. The H/S pin is the name of the pin for the programmer and this pin might not be described as an alternate-function pin in the user's manual of the device. In such a case, connect the relevant port that is assigned to the H/S pin. (In the V850ES/JJ, for example, the PCM0 pin is assigned to the H/S pin.)

The pins used for serial communication are basically the same as those of the flash memory programmer (such as PG-FP5), but some devices do not support some of them. Refer to the document ″QB-MINI2 Operating Precautions″ and check the supported pins.

Figure 3-1. Outline of Communication Interface
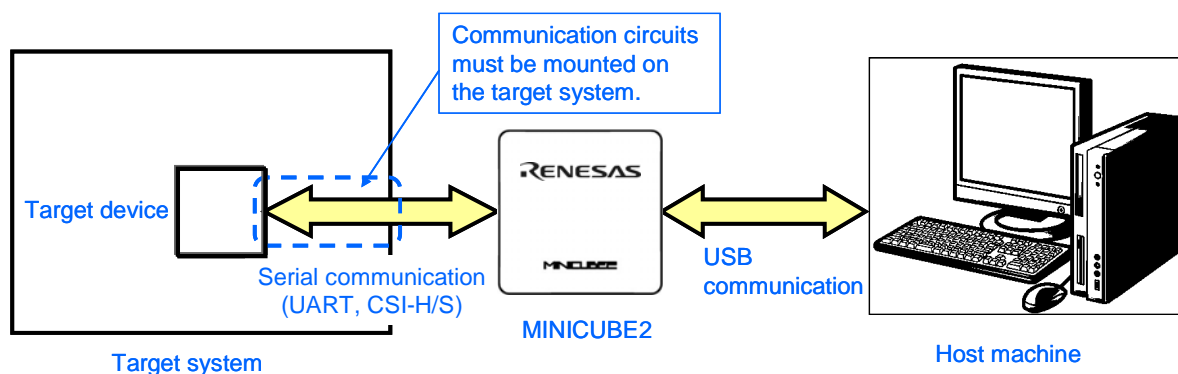
3.1.1 Pin assignment

This section describes the interface signals used between MINICUBE2 and the target system. Table 3-1 lists the pin assignment. Table 3-2 describes the functions of each pin. The pin assignment varies depending on whether CSI-H/S or UART is used, so design the circuit appropriately according to the circuit connection examples described on the following sections.

Table 3-1. Pin Assignment

Pin No.	Pin Name[Note]	
	CSI-H/S Used	UART Used
1	GND	
2	RESET_OUT	
3	SI	RxD
4	VDD	
5	SO	TxD
6	R.F.U.	
7	SCK	R.F.U.
8	H/S	R.F.U.
9	CLK	
10	R.F.U.	
11	R.F.U.	
12	FLMD1	
13	R.F.U.	
14	FLMD0	
15	RESET_IN	
16	R.F.U.	

Note   Signal names in MINICUBE2

Table 3-2. Pin Functions

Pin Name	IN/OUT[Note]	Description
RESET_IN	IN	Pin used to input reset signal from the target system
RESET_OUT	OUT	Pin used to output reset signal to the target device
CLK	OUT	Pin used to output clock signal to the target device
FLMD0	OUT	Pin used to set the target device to debug mode or programming mode
FLMD1	OUT	Pin used to set programming mode
RxD/SI	IN	Pin used to receive command/data from the target device
TxD/SO	OUT	Pin used to transmit command/data to the target device
SCK	OUT	Pin used to output clock signal for 3-wire serial communication
H/S	IN	Pin used to input handshake signal from the target device
R.F.U.	-	This pin is reserved. For the connection of the reserved pins, see each circuit related to the pins.

\<R>

Note   As seen from MINICUBE2

3.1.2 Circuit connection examples

The circuit design on the target system varies depending on the communication interface mode. Refer to the following table and see the relevant circuit connection example.

Caution The constants described in the circuit connection example are reference values. If you perform flash programming aiming at mass production, thoroughly evaluate whether the specifications of the target device are satisfied.

Table 3-3. Corresponding Circuit Connection Diagrams

Communication Interface Mode	Refer to:
UART	Figure 3-2
CSI-H/S	Figure 3-3

<R> **Caution The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.**

Figure 3-2. Recommended Circuit Connection When UART Is Used for Communication Interface

Notes 1. Connect TxD (transmit side) of the target device to RxD (receive side) of the target connector, and TxD (transmit side) of the target connector to RxD (receive side) of the target device. Read the serial interface pin names on the target device side as those for flash programming supported by the target device.

   2. This pin may be used to supply an external clock during flash programming (4, 8 or 16 MHz). For the connection, refer to the user's manual for the target device.

   3. The alternate function of this pin can be used because this pin changes to input (unused) when debugging. In MINICUBE2, it is pulled down at 100 kΩ.

   4. This connection is designed assuming that the RESET signal is output from the N-ch open-drain buffer (output resistance: 100Ω or less). For details, refer to **3.1.3 Connection of reset pin**.

   5. The circuit enclosed by a dashed line is not required when only flash programming is performed.

   6. The circuit enclosed by a dashed line is designed for flash self programming, which controls the FLMD0 pin via ports. Use the port for inputting or outputting the high level. When flash self programming is not performed, a pull-down resistance for the FLMD0 pin can be within 1 to 10 kΩ.

   7. This is for pin processing when not used as a device.
   To change this value to another resistance value, see **APPENDIX A EQUIVALENT CIRCUIT**.

RENESAS

<R> **Caution The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.**

Figure 3-3. Recommended Circuit Connection When CSI-H/S Is Used for Communication Interface
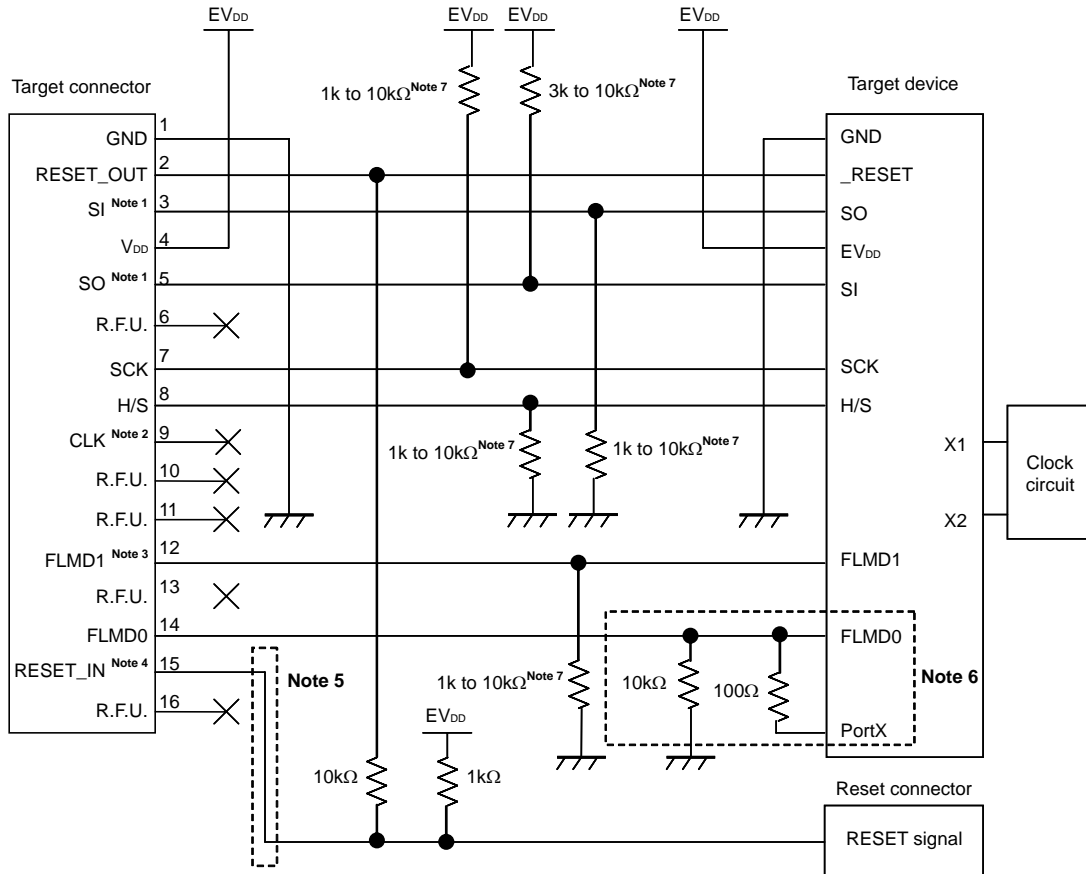
Notes 1. Connect SO (transmit side) of the target device to SI (receive side) of the target connector, and SO (transmit side) of the target connector to SI (receive side) of the target device. Read the serial interface pin names on the target device side as those for flash programming supported by the target device.

    2. This pin may be used to supply an external clock during flash programming (4, 8 or 16 MHz). For the connection, refer to the user's manual for the target device.

    3. The alternate function of this pin can be used because this pin changes to input (unused) when debugging. In MINICUBE2, it is pulled down at 100 kΩ.

    4. This connection is designed assuming that the RESET signal is output from the N-ch open-drain buffer (output resistance: 100Ω or less). For details, refer to **3.1.3 Connection of reset pin**.

    5. The circuit enclosed by a dashed line is not required when only flash programming is performed.

    6. The circuit enclosed by a dashed line is designed for flash self programming, which controls the FLMD0 pin via ports. Use the port for inputting or outputting the high level. When flash self programming is not performed, a pull-down resistance for the FLMD0 pin can be within 1 to 10 kΩ.

    7. This is for pin processing when not used as a device.
       To change this value to another resistance value, see **APPENDIX A EQUIVALENT CIRCUIT**.

3.1.3 Connection of reset pin

This section describes the connection of the reset pin, for which special attention must be paid, in circuit connection examples shown in the previous section.

During on-chip debugging, a reset signal from the target system is input to MINICUBE2, masked, and then output to the target device. Therefore, the reset signal connection varies depending on whether MINICUBE2 is connected.

For flash programming, the circuit must be designed so that the reset signals of the target system and MINICUBE2 do not conflict.

Select one of the following methods and connect the reset signal in the circuit. The details of each method are described on the following pages.

(1)   Automatically switching the reset signal via series resistor (recommended; described in recommended circuit connection in the previous section)

(2)   Manually switching the reset signal with jumper

(3)   Resetting the target device by power-on clear (POC) only

(1) Automatically switching the reset signal via series resistor

Figure 3-4 illustrates the reset pin connection described in **3.1.2 Circuit connection examples**.

This connection is designed assuming that the reset circuit on the target system contains an N-ch open-drain buffer (output resistance: 100Ω or less). The VDD or GND level may be unstable when the logic of RESET_IN/OUT of MINICUBE2 is inverted, so observe the conditions described below in **Remark**.

Figure 3-4. Circuit Connection with Reset Circuit That Contains Buffer
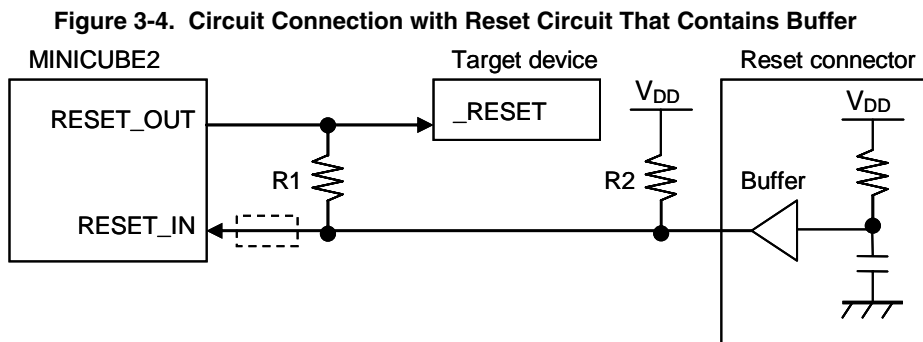
Remark   Make the resistance of at least R1 ten times that of R2, R1 being 10 kΩ or more.
Pull-up resistor R2 is not required if the buffer of the reset circuit consists of CMOS output.
The circuit enclosed by a dashed line is not required when only flash programming is performed.

Figure 3-5 illustrates the circuit connection for the case where the reset circuit on the target system contains no buffers and the reset signal is only generated via resistors or capacitors. Design the circuit, observing the conditions described below in **Remark**.

Figure 3-5. Circuit Connection with Reset Circuit That Contains No Buffers
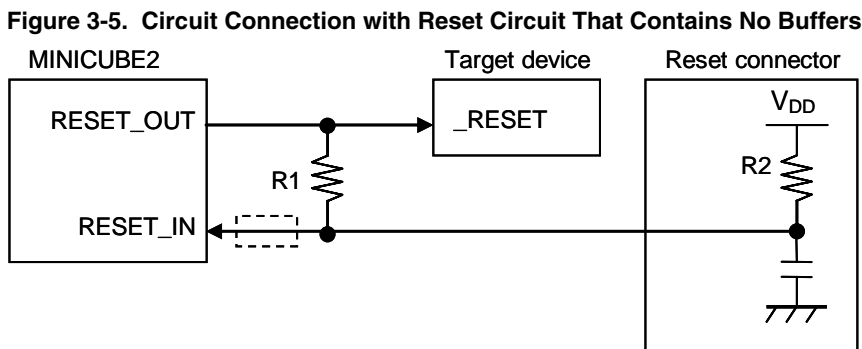
Remark   Make the resistance of at least R1 ten times that of R2, R1 being 10 kΩ or more.
The circuit enclosed by a dashed line is not required when only flash programming is performed.

(2) Manually switching the reset signal with jumper

Figure 3-6 illustrates the circuit connection for the case where the reset signal is switched using the jumper, with or without MINICUBE2 connected. This connection is simple, but the jumper must be set manually.

Figure 3-6. Circuit Connection for Switching Reset Signal with Jumper


```
*  Jumper setting
   When MINICUBE2 is connected:        1-2 shorted
   When MINICUBE2 is not connected:    2-3 shorted
```
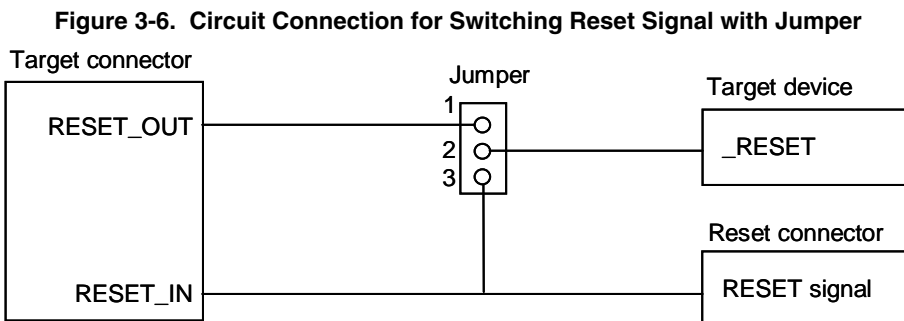
(3) Resetting the target device by power-on clear (POC) only

Figure 3-7 illustrates the circuit connection for the case where the target device is only reset via POC without using the reset pin. RESET_OUT is valid only when the debugger is running or during flash programming.
The operation is not guaranteed if the power to the target system is shut down during debugging. Note that the POC function cannot be emulated.

Figure 3-7. Circuit Connection for the Case Where Target Device Is Only Reset via POC

3.1.4 Mounting connector onto target system

    The connector must be mounted on the target system for connecting MINICUBE2 with the target system. A 2.54 mm pitch 16-pin general-purpose connector can be used. The following products are recommended.

- HIF3FC-16PA-2.54DS (right-angle type, made by Hirose Electric Co., Ltd.)
- HIF3FC-16PA-2.54DSA (straight type, made by Hirose Electric Co., Ltd.)
- 7616-5002PL (right-angle type, made by Sumitomo 3M, Ltd.)
- 7616-6002PL (straight type, made by Sumitomo 3M, Ltd.)

    Compact connectors and loose-wire cables are also available as options. For more information on these products, access the MINICUBE2 information site.

    URL: http:// http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html

Figure 3-8. 2.54 mm Pitch 16-Pin General-Purpose Connector Pin Layout
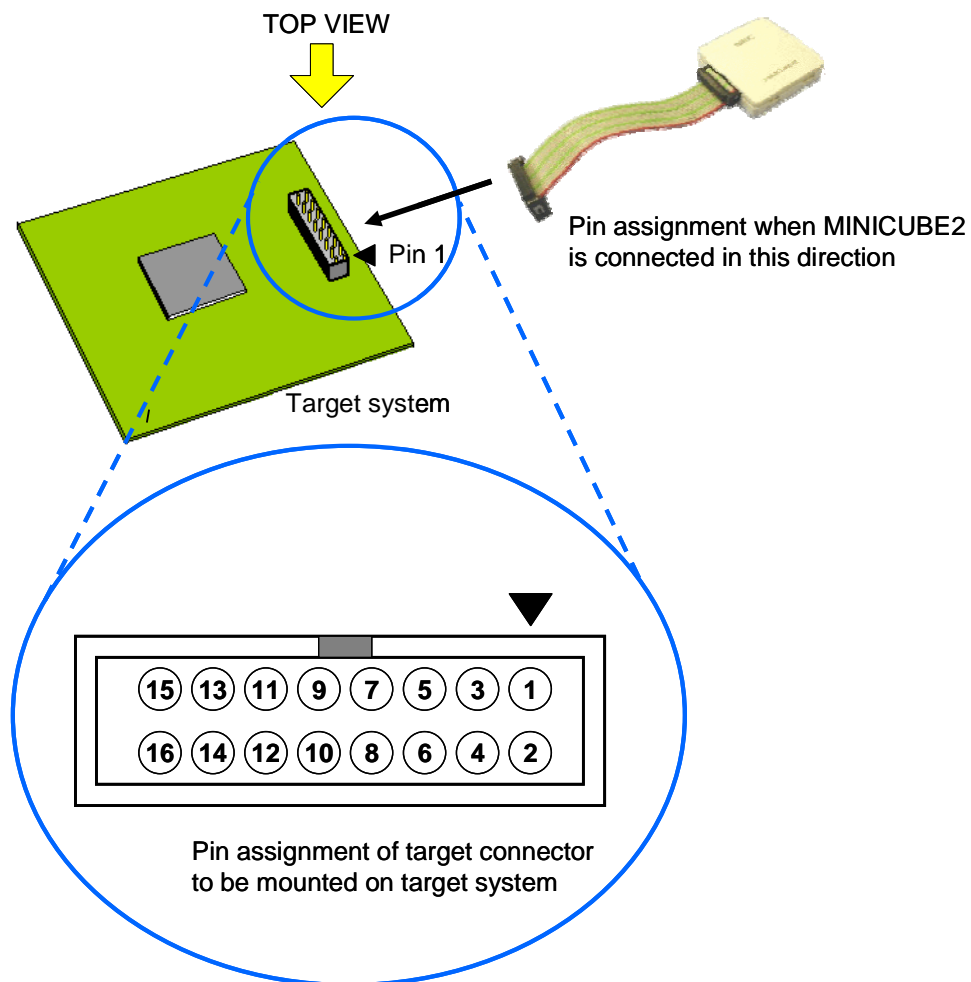
3.2 On-Chip Debugging

This section describes the system configuration, startup/shutdown procedure and cautions for debugging when on-chip debugging is performed with MINICUBE2.

3.2.1 Debug functions

Table 3-4 lists the debug functions when a V850 microcontroller is the target device and the ID850QB is used. If a debugger other than the ID850QB (made by partner manufacturer) is used, the available functions may differ, so check the specifications of the debugger used.

Table 3-4. Debug Functions

Functions		Specifications
Security		10-byte ID code authentication
Download		Available
Execution		Go & Go, Start from Here, Come Here, Restart, step execution
Hardware break		2 points[Note 1] (commonly used by execution and access)
Software break	ROM area	4 points
	RAM area	2,000 points
Forced break		Available[Note 2]
RAM monitoring		Available
DMM (writing to memory during RUN)		Available
Pin masking		Available (reset pin only)
Time measurement (from execution start to break)		Measurement resolution: 100 μs
		Max. measurement time: Approximately 100 hours
User spaces used for debugging		Internal ROM: 2 K bytes+12 bytes
		Internal RAM: 16 bytes max.
Function pins used for debugging		UART selected: RxD, TxD
		CSI-H/S selected: SI, SO, SCK, H/S

Notes 1.   The following devices are not yet supported.

V850ES/KE2, V850ES/KF2, V850ES/KG2, μPD70F3733, V850ES/IE2

2.   Available with restrictions. For details, refer to **3.2.6 Cautions on debugging**.

RENESAS

3.2.2 System configuration

Figure 3-9 illustrates the system configuration for on-chip debugging.

Figure 3-9. System Configuration for On-Chip Debugging
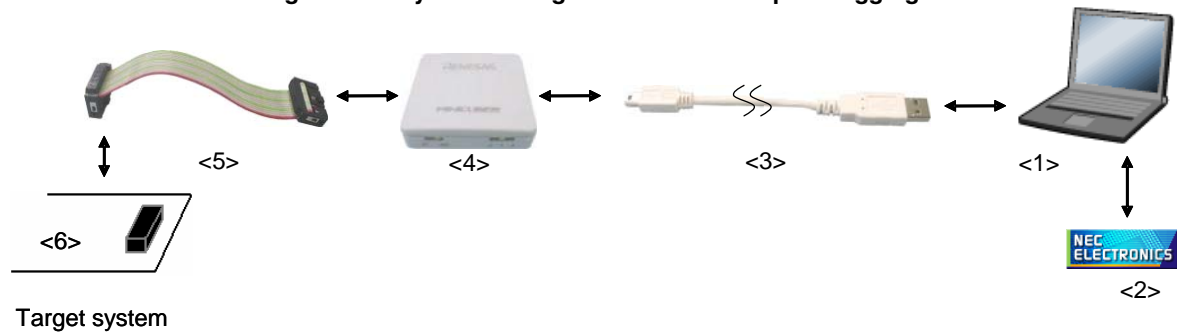
<5>   <4>   <3>   <1>

<6>

Target system

<2>

<1> Host machine

Products with USB ports

<2> Software

Includes the debugger, USB driver, device files, and so on.

Download these Renesas Electronics software from the following website (ODS).

URL: http://www2.renesas.com/micro/en/ods/index.html

<3> USB cable (accessory)

<4> MINICUBE2 (this product)

<5> 16-pin target cable (accessory)

<6> Target connector (sold separately)

3.2.3　System startup procedure

This section describes the system startup procedure. Observe the following order.

(1) Preparation and installation of software

The following software is required to perform on-chip debugging. For details on preparation and installation of software, refer to the setup manual supplied with MINICUBE2.

- Debugger
- USB driver
- Device file (may be unnecessary when using a partner manufacture's debugger)

(2) Switch setting

Set the mode select switch to ″M2″. Set the power select switch by referring to Table 3-5.

Caution　　Do not change the switch setting while the USB cable is connected.

Table 3-5. Setting of Power Select Switch

Setting	Description
3	3 V is supplied from MINICUBE2 to the target system[Note].
5	5 V is supplied from MINICUBE2 to the target system[Note].
T	Power supply of the target system is used. MINICUBE2 detects the power for the target system. The detected power is also used as the power supply for the communication interface.

Note The maximum rating of the current is 100 mA, so do not use MINICUBE2 with the target system with the higher current rating. The power is always supplied after MINICUBE2 is connected to the host machine.

(3) Connecting the target system

Connect MINICUBE2 to the target system as shown in Figure 3-10, **before the power to the target system is turned on**.

Figure 3-10. Connecting MINICUBE2 to Target System

16-pin target cable　　　　MINICUBE2
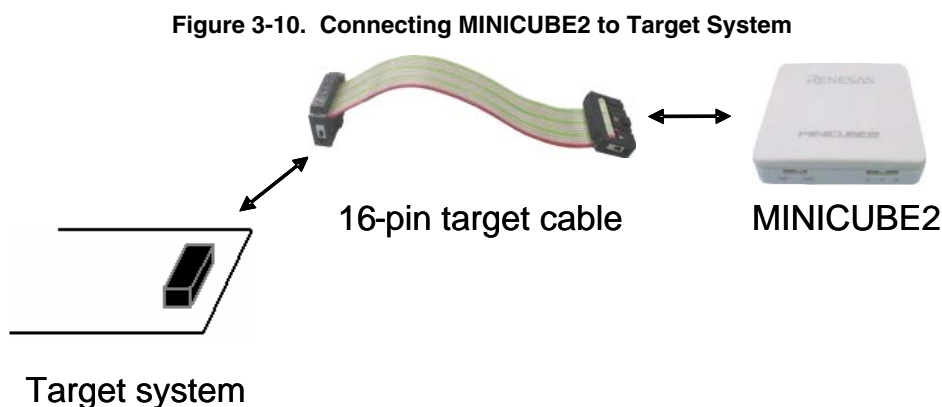
Target system

(4) Connecting the USB cable

Connect MINICUBE2 to the host machine as shown in Figure 3-11, **before the power to the target system is turned on**.

When the power select switch is set to ″T″, the mode LED **flashes** white after connection.

When the power select switch is set to ″3″ or ″5″, the mode LED **glows** white after connection.

Figure 3-11. Connecting MINICUBE2 to Host Machine

MINICUBE2   USB cable   Host machine
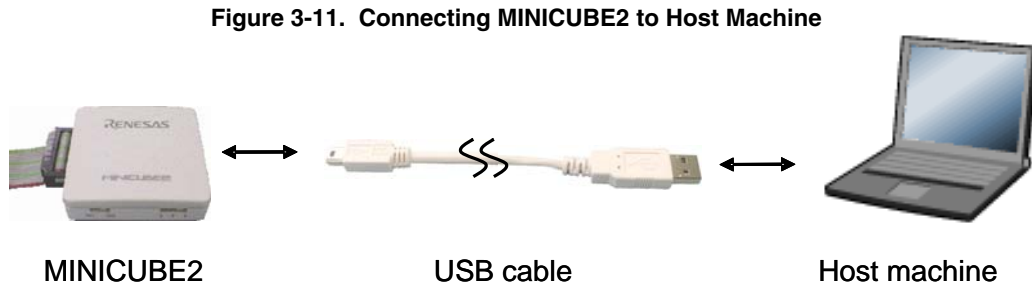
(5) Applying power to target system

Turn on power to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″. The mode LED glows white after power application.

(6) Debugger startup

Start the debugger. The mode LED glows blue after debugger startup.

For the operation after this step, refer to the user's manual for the debugger.

If the debugger does not start normally or the operation is unstable, the possible causes may be the following.

• Communication error between MINICUBE2 and target system

  Whether communication is performed normally can be checked using the OCD Checker. For details, refer to the user's manual for the OCD Checker.

• The user resource has not been secured or the security ID has not been set

  To perform debugging with MINICUBE2, the debug monitor area and communication serial interface for communication must be secured and the security ID must be set. For details, refer to **3.2.5 Securing of user resources and setting of security ID**.

• Unsupported software (debugger, device file, or firmware) is used

  The software used may not support debugging of the target device. Refer to the document ″QB-MINI2 Operating Precautions″ and check the supported version. When using software provided by a Renesas Electronics partner, refer to the documents prepared by the partner company.

• Defect of MINICUBE2

  MINICUBE2 may have a defect. Refer to **CHAPTER 7 SELF-TESTING**.

3.2.4 System shutdown procedure

Terminate debugging and shutdown the system in the following order.

If the following order is not observed, the target system or MINICUBE2 may be damaged.

(1) Debugger termination

Terminate the debugger.

(2) Target system power shutdown

Shutdown the power supplied to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″.

(3) Unplugging the USB cable

Unplug the USB cable from MINICUBE2 or the host machine.

(4) Unplugging the target cable

Unplug the target cable from MINICUBE2 or the target system.

3.2.5 Securing of user resources and setting of security ID

The user must prepare the following to perform communication between MINICUBE2 and the target device and implement each debug function. Refer to the descriptions on the following pages and set these items in the user program or using the compiler options.

When a partner tool is used, read also the following material.
○ When using MULTI manufactured by Green Hills Software
    QB-MINI2 Setup Manual User's Manual Partner Tool (U19158E)
○ When using C-SPY™ manufactured by IAR Systems
    IAR C-SPY Hardware Debugger Systems User Guide issued by IAR Systems

Securement of memory space

The shaded portions in Figure 3-12 are the areas reserved for placing the debug monitor program, so user programs and data cannot be allocated in these spaces. These spaces must be secured so as not to be used by the user program.

Figure 3-12. Memory Spaces Where Debug Monitor Programs Are Allocated
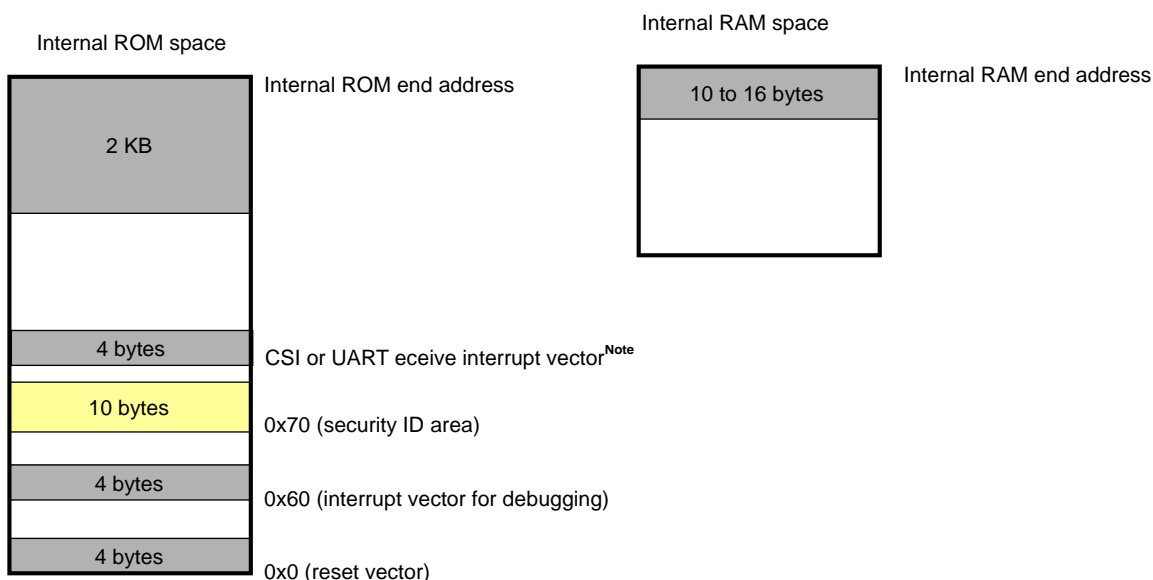
□ : Debugging area
Note The vectors for the receive error interrupt or receive status interrupt, if any, must be secured.

- Securement of serial interface for communication
  The register settings, concerning UART or CSI-H/S used for communication between MINICUBE2 and the target device, performed by the debug monitor program must not be changed.

- Security ID setting
  The ID code must be embedded in the area filled with yellow (0x70 to 0x79) in Figure 3-12, to prevent the memory from being read by an unauthorized person.

(a) Reset vector

A reset vector includes the jump instruction for the debug monitor program.

[How to secure areas]

It is not necessary to secure this area intentionally. When downloading a program, however, the debugger rewrites the reset vector in accordance with the following cases. If the rewritten pattern does not match the following cases, the debugger generates an error (F0c34 when using the ID850QB).

- When two nop instructions are placed in succession from address 0

    Before writing   After writing
    0x0 nop   →   Jumps to debug monitor program at 0x0
    0x2 nop   0x4 xxxx
    0x4 xxxx

- When two 0xFFFF are successively placed from address 0 (already erased device)

    Before writing   After writing
    0x0 0xFFFF   →   Jumps to debug monitor program at 0x0
    0x2 0xFFFF   0x4 xxxx
    0x4 xxxx

- The *jr* instruction is placed at address 0 (when using Renesas Electronics compiler CA850)

    Before writing   After writing
    0x0 jr disp22   →   Jumps to debug monitor program at 0x0
                             0x4 jr disp22 - 4

- *mov32* and *jmp* are placed in succession from address 0 (when using IAR compiler ICCV850)

    Before writing   After writing
    0x0 mov imm32,reg1   →   Jumps to debug monitor program at 0x0
    0x6 jmp [reg1]   0x4 mov imm32,reg1
                             0xa jmp [reg1]

- The jump instruction for the debug monitor program is placed at address 0

    Before writing   After writing
    Jumps to debug monitor program at 0x0   →   No change

(b) Securement of area for debug monitor program

The shaded portions in Figure 3-12 are the areas where the debug monitor program is allocated. The monitor program performs initialization processing for debug communication interface and RUN or break processing for the CPU. The internal ROM area must be filled with 0xFF. This area must not be rewritten by the user program.

[How to secure areas]

It is not necessarily required to secure this area if the user program does not use this area.

To avoid problems that may occur during the debugger startup, however, it is recommended to secure this area in advance, using the compiler.

The following shows examples for securing the area, using the Renesas Electronics compiler CA850. Add the assemble source file and link directive code, as shown below.

• Assemble source (Add the following code as an assemble source file.)

```
-- Secures 2 KB space for monitor ROM section
.section  "MonitorROM", const
.space    0x800, 0xff^Note

-- Secures interrupt vector for debugging
.section  "DBG0"
.space    4, 0xff

-- Secures interrupt vector for serial communication for receive
-- Secures vector for receive error interrupt and receive status interrupt, if any
-- Change the section name according to serial communication mode used
.section  "INTCSI00"
.space    4, 0xff

-- Secures 16 byte space for monitor ROM section
.section  "MonitorRAM", bss
.lcomm   monitorramsym, 16, 4      /* defines monitorramsym symbol */
```

Note The downloading speed can be increased by replacing this line with the statement "monitorromsym:" to perform a symbol definition only. This effect is not applicable if values are filled into a hole (area without a code). When performing filling, the filling value must be 0xFF for securing the area.

• Link directive (Add the following code to the link directive file.)

The following shows an example when the internal ROM end address is 0x3ffff and internal RAM end address is 0x3ffefff.

```
MROMSEG : !LOAD ?R V0x03f800{
        MonitorROM      = $PROGBITS    ?A MonitorROM;
};

MRAMSEG : !LOAD ?RW V0x03ffeff0{
        MonitorRAM      = $NOBITS        ?AW MonitorRAM;
};
```

RENESAS

(c) Securement of communication serial interface

UART or CSI-H/S is used for communication between MINICUBE2 and the target system. The settings related to the serial interface modes are performed by the debug monitor program, but if the setting is changed by the user program, a communication error may occur.

To prevent such a problem from occurring, communication serial interface must be secured in the user program.

[How to secure communication serial interface]

Create the user program observing the following points.

- Serial interface registers

  Do not set the registers related to UART and CSI-H/S in the user program.

- Interrupt mask register

  When UART is used, do not mask receive end interrupts[Note]. When CSI-H/S is used, do not mask transmit end interrupts. The following shows an example.

  Example Setting other than below is prohibited when the target device is the V850ES/KJ2 and CSI00 is used.

	7	6	5	4	3	2	1	0
CSI0IC0	x	0	x	x	x	x	x	x

  x: Any

  Note Do not mask receive these interrupts when there is a receive error interrupt or a receive status interrupt.

- Port registers

  When UART is used, do not set port registers to make the TxD and RxD pins invalid. When CSI-H/S is used, do not set port registers to make the SI, SO, SCK and H/S pins invalid. The H/S pin is used as the port output for debugging. The following shows two examples.

  Example 1 Setting other than below is prohibited when the target device is the V850ES/KJ2 and UART0 is used.

	7	6	5	4	3	2	1	0
PFC3	x	x	x	x	x	x	0	0

	7	6	5	4	3	2	1	0
PMC3L	x	x	x	x	x	x	1	1

  x: Any

Example 2 Setting other than below is prohibited when the target device is the V850ES/HG2 and CSIB0 is used.

	7	6	5	4	3	2	1	0
PMC4	x	x	x	x	x	1	1	1

	7	6	5	4	3	2	1	0
PMCCM	x	x	x	x	x	x	x	0

	7	6	5	4	3	2	1	0
PMCM	x	x	x	x	x	x	x	0

	7	6	5	4	3	2	1	0
PCM	x	x	x	x	x	x	x	Read-only[Note]

x: Any

Note The port values corresponding to the H/S pin are changed by the monitor program according to the debugger status. To perform port register settings in 8-bit units, usually the user program can use read-modify-write. If an interrupt for debugging occurs before writing, however, an unexpected operation may be performed.

(d) Security ID setting

This setting is required to prevent the memory from being read by an unauthorized person. Embed a security ID at addresses 0x70 to 0x79 in the internal flash memory. The debugger starts only when the security ID that is set during debugger startup and the security ID set at addresses 0x70 to 0x79 match.

If bit 7 of address 0x79 is ″0″, however, debugging is disabled. In such a case, there are no methods to start the debugger. Debugging is mainly disabled for mass-produced devices.

If the user has forgotten the security ID or to enable debugging, erase the flash memory and set the security ID again.

[How to set security ID]
Embed a security ID at addresses 0x70 to 0x79 in the user program.
If the security ID is embedded as follows, for example, the security ID set by the debugger is
″123456789ABCDEF123D4″ (not case-sensitive).

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

If Renesas Electronics compiler CA850 is used, the security ID can be set using the Compiler Common Options menu.

Figure 3-13. Security ID Setting with CA850
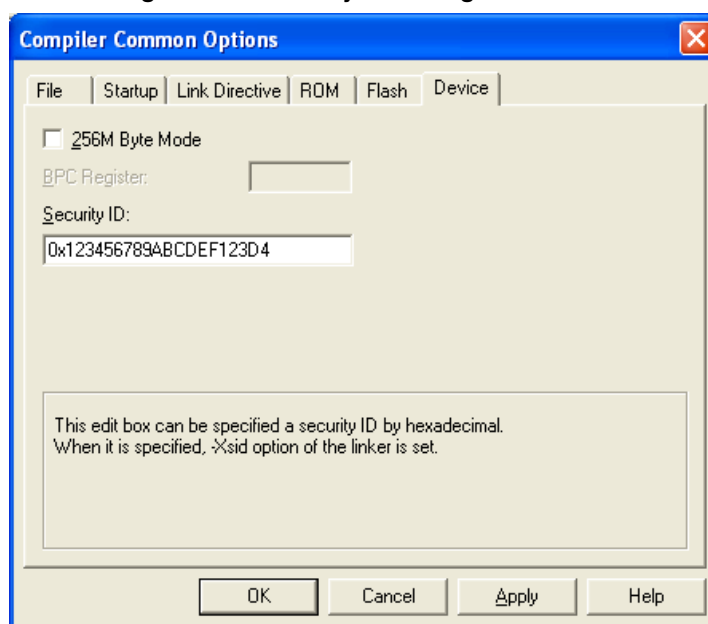
3.2.6 Cautions on debugging

This section describes cautions on performing on-chip debugging for a V850 microcontroller.

Be sure to read the following to use MINICUBE2 properly.

(1) Handling of device that was used for debugging

Do not mount a device that was used for debugging on a mass-produced product, because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed. Moreover, do not embed the debug monitor program into mass-produced products.

(2) When breaks cannot be executed

Forced breaks cannot be executed if one of the following conditions is satisfied.
- Interrupts are disabled (DI)
- Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the target device, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the target device is UART, and the main clock has been stopped

(3) When pseudo real-time RAM monitor (RRM) function and DMM function do not operate

The pseudo RRM function and DMM function do not operate if one of the following conditions is satisfied.
- Interrupts are disabled (DI)
- Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the target device, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the target device is UART, and the main clock has been stopped
- Mode for communication between MINICUBE2 and the target device is UART, and a clock different from the one specified in the debugger is used for communication

(4) Standby release with pseudo RRM and DMM functions enabled

The standby mode is released by the pseudo RRM function and DMM function if one of the following conditions is satisfied.
- Mode for communication between MINICUBE2 and the target device is CSI-H/S
- Mode for communication between MINICUBE2 and the target device is UART, and the main clock has been supplied

(5) Writing to peripheral I/O registers that requires a specific sequence, using DMM function

Peripheral I/O registers that requires a specific sequence cannot be written with the DMM function.

(6) Device with which the debugger startup is slowed

When the debugger is started for the first time, chip erasure and writing of debug monitor program are performed. The following devices require about a dozen seconds for these operations.

V850ES/JG2, V850ES/JJ2

V850ES/HE2, V850ES/HF2, V850ES/HG2, V850ES/HJ2

V850ES/IE2

(7) Writing of debug monitor program

If the CPU operating clock setting is changed by the debugger, the debugger rewrites the debug monitor program. At this time, it takes about a dozen seconds for this operation. When using Renesas Electronics debugger ID850QB, this specification applies when the setting in the Clock area in the Configuration dialog box.

(8) Flash self programming

If a space where the debug monitor program is allocated is rewritten by flash self programming, the debugger can no longer operate normally.

(9) Emulation of POC function

The POC function of the target device cannot be emulated. Make sure that the power to the target system is not shut down during debugging.

(10) Operation after reset

After an external pin reset or internal reset, the monitor program performs debug initialization processing. Consequently, the time from reset occurrence until user program execution differs from that in the actual device operation.

(11) Debugging with real machine running without using MINICUBE2

If debugging is performed with a real machine running, without using MINICUBE2, write the user program using the QB-Programmer. Programs downloaded by the debugger include the monitor program, and such a program malfunctions if it is not controlled via MINICUBE2.

(12) Cautions unique to devices

• V850ES/Fx3

Note the following two points when debugging the V850ES/Fx3 as the target microcontroller.

<1> When the MINICUBE2 is connected to the target system via UARTD0, fRH cannot be used as the CPU clock. The debug monitor program will change the MCM register setting so as to select the main clock oscillator clock after the reset. If the MCM register setting is changed by the user program, the debugger operation will become invalid.

<2> When using the pseudo RRM function or the DMM function with the ID850QB, set the result of the following calculation as the "Multiply rate" in the Configuration dialog box.

Multiply rate = $8 \times$ PLLO $\times$ PLLI

Example   When PLLO = 1/2 and PLLI = 1/2,

Multiply rate = $8 \times 1/2 \times 1/2 = 2$

- V850E/MA3
    Note the following two points when debugging the V850E/MA3 as the target microcontroller.
    <1> If the "Multiply rate" is set to 1.25 or 2.5 in the Configuration dialog box when using the ID850QB, these
        values will be an integer 1 or 2 when the Configuration dialog box is opened the next time. This is only a
        problem of the debugger's display, so the actual CPU clock is not affected.
    <2> When the CKSEL pin is low level (PLL mode) and the target interface is CSI, set the "Multiply rate" in the
        Configuration dialog box to 10, and then start the debugger. This process is required because the clock is
        fixed to one multiplied by 10 in the device specifications for the instance of writing the debug monitor
        program in the programming mode. Once the debugger starts, restoring the original multiply rate will not
        affect the debugging operation.
- V850ES/IE2, V850ES/IK1, V850E/IA3, V850E/IA4, V850E/IF3 and V850E/IG3
    Note the following point when debugging the V850ES/IE2, V850ES/IK1, V850E/IA3 or V850E/IA4 as the target
    microcontroller.
    <1> When the target interface is CSI, set the "Multiply rate" in the Configuration dialog box to 8 and then start the
        debugger. This process is required because the clock is fixed to one multiplied by 8 in the device
        specifications for the instance of writing the debug monitor program in the programming mode. Once the
        debugger starts, restoring the original multiply rate will not affect the debugging operation.

(13) Processing of monitoring program for debugging

An overview of the processing of the monitoring program for debugging when MINICUBE2 has been connected
and the debugger has been started (including after the debugger has been used to perform a CPU reset), and
related cautions are described below.

<1> Stopping watchdog timer that starts operating immediately after reset

> **Caution**   **When an operation has been started in a user program for a watchdog timer whose
> operation is not started immediately after reset, the debugger may hang up. In this case,
> use the debugger to perform a CPU reset and restore the CPU.**

<2> Clock operations
    The V850ES/Fx3, V850ES/Fx3-L, and V850ES/Hx3 can use an internal oscillator as the CPU clock. For such
    devices, an internal clock is switched to an external clock during UART communication. (The PLL and
    frequency division are the same as during operation of a program created by the user.)

> **Caution**   **With MINICUBE2, an internal clock is not supported during UART communication. (The
> internal low-speed clock when the clock has been stopped is also not supported during
> UART communication.) Both clocks can be used in CSI.**

<3> Changing settings to enable serial communication, according to debugger communication pin settings
    For example, in the case of UART, the settings of UART control, the data format, and the baud rate are
    changed, and the alternate-function ports are changed to control mode. Also, masking of UART receive
    interrupts is released.

> **Caution**   **When the values of the registers related to the communication pins are changed from
> those set by the monitoring program, the debugger may hang up, because a
> communication failure will occur. In this case, use the debugger to perform a CPU reset
> and restore the CPU.**

3.3 Flash Programming

This section describes the system configuration and startup/shutdown procedure when flash programming is performed for a V850 microcontroller, using MINICUBE2.

3.3.1 Specifications of programming function

Table 3-6. Specifications of Programming Function

Functions	Specifications
Host interface	USB 2.0 (compatible with 1.1)
Target interface	UART or CSI-H/S
Target system voltage	1.65 to 5.5 V (depends on the target device)
Clock supply	16, 8, or 4 MHz clock can be supplied Clock mounted on the target system can be used
Power supply	3 or 5 V (maximum current rating: 100 mA)
Acquisition of device-specific information	Parameter file for MINICUBE2 made by Renesas Electronics is used
Security flag setting	Available
Standalone operation	Unavailable (must be connected to host machine)

<R> (Target system voltage row)

<R> (Acquisition of device-specific information row)

3.3.2 System configuration

Figure 3-14 illustrates the system configuration for flash programming.

Figure 3-14. System Configuration for Flash Programming
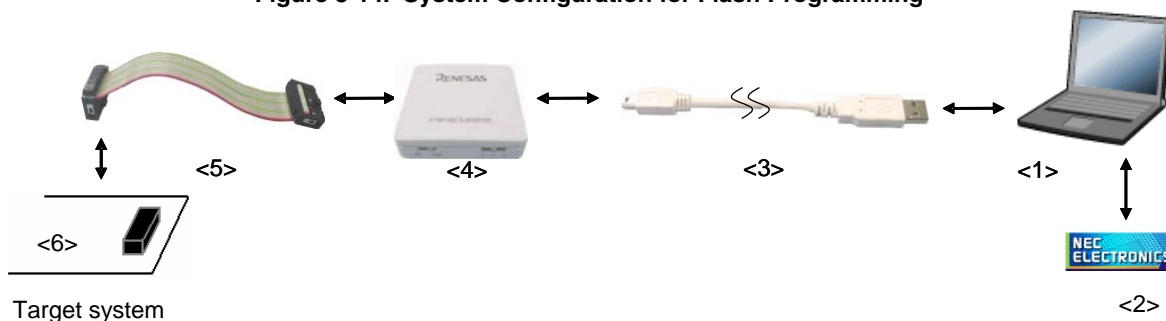
Target system

<1> Host machine
      Products with USB ports
<2> Software
      Includes the QB-Programmer USB driver, parameter files, and so on.
      Download these Renesas Electronics software from the following website (ODS).
          URL: http://www2.renesas.com/micro/en/ods/index.html
<3> USB cable (accessory)
<4> MINICUBE2 (this product)
<5> 16-pin target cable (accessory)
<6> Target connector (sold separately)

3.3.3 System startup procedure
This section describes the system startup procedure. Observe the following order.

(1) Preparation and installation of software
The following software is required to perform flash programming. For details on preparation and installation of software, refer to the setup manual supplied with MINICUBE2.
- QB-Programmer
- USB driver
- Parameter file

(2) Switch setting
Set the mode select switch to ″M2″. Set the power select switch by referring to Table.

Caution   Do not change the switch setting while the USB cable is connected.

Table 3-7. Setting of Power Select Switch

Setting	Description
3	3 V is supplied from MINICUBE2 to the target system[Note]
5	5 V is supplied from MINICUBE2 to the target system[Note].
T	Power supply of the target system is used.
	MINICUBE2 detects the power for the target system.
	The detected power is also used as the power supply for the communication interface.

Note The maximum rating of the current is 100 mA, so do not use MINICUBE2 with the target system with the higher current rating. The power is always supplied after MINICUBE2 is connected to the host machine.

(3) Connecting the target system
Connect MINICUBE2 to the target system as shown in Figure 3-15, **before the power to the target system is turned on**. When the power select switch is set to ″T″, the step of **(4) Connecting the USB cable** can be performed before this step.

Figure 3-15. Connecting MINICUBE2 to Target System

16-pin target cable   MINICUBE2
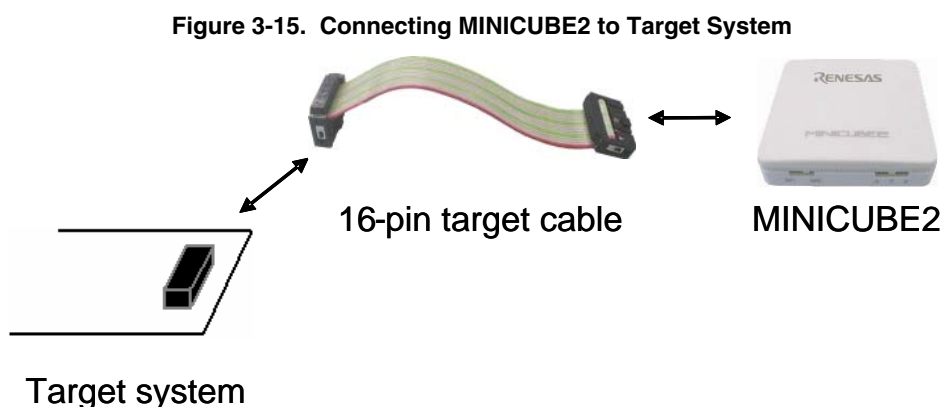
Target system

(4) Connecting the USB cable

Connect MINICUBE2 to the host machine as shown in Figure 3-16, **before the power to the target system is turned on**.

When the power select switch is set to ″T″, the mode LED **flashes** white after connection.

When the power select switch is set to ″3″ or ″5″, the mode LED **glows** white after connection. The power is always supplied after MINICUBE2 is connected to the host machine with the USB cable.

Caution   Do not change the switch setting while the USB cable is connected.

Figure 3-16. Connecting MINICUBE2 to Host Machine

MINICUBE2   USB cable   Host machine

(5) Applying power to target system

Turn on power to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″. The mode LED glows white after power application.

(6) QB-Programmer startup

Start the QB-Programmer from the Start menu of Windows. The mode LED glows green after QB-Programmer startup.

A series of basic operations after this step is described in the following section. For detailed usage of the QB-Programmer, refer to the **QB-Programmer Programming GUI Operation User's Manual**.

3.3.4 Usage examples

This section describes a series of basic operations using the QB-Programmer, using the μPD70F3732 as an example. The operations from execution of the [Autoprocedure (EPV)] command to programming for the target device are described.   For other commands and their applications, refer to the **QB-Programmer Programming GUI Operation User's Manual**.

The conditions used for a series of operations in this section are as follows.

<Target system>
Target device:	μPD70F3732
Power supply voltage:	5 V (supplied from MINICUBE2)
Communication channel:	CSI-H/S, 2 MHz

<MINICUBE2>
Switches:	Mode select switch: M2
	Power select switch: 5

<QB-Programmer>
Parameter file:	70F3732.PRM
Clock setting:	5 MHz, x4 (oscillator on the target system side)
Operation mode:	chip
Program file:	sample.hex
Command options:	[Blank check before Erase] selected
	[Read verify after Program] selected
	[Security flag after Program] selected
Security function:	[Disable Read] selected

(1) Setting of programming environment

Set the programming environment, following the steps <1> to <7> described below.

<1> Click the [Device] menu on the menu bar and then click [Setup...].

<2> The Device Setup dialog box appears, in which the [Standard] tab is active.

Figure 3-17. [Standard] Tab of Device Setup Dialog Box
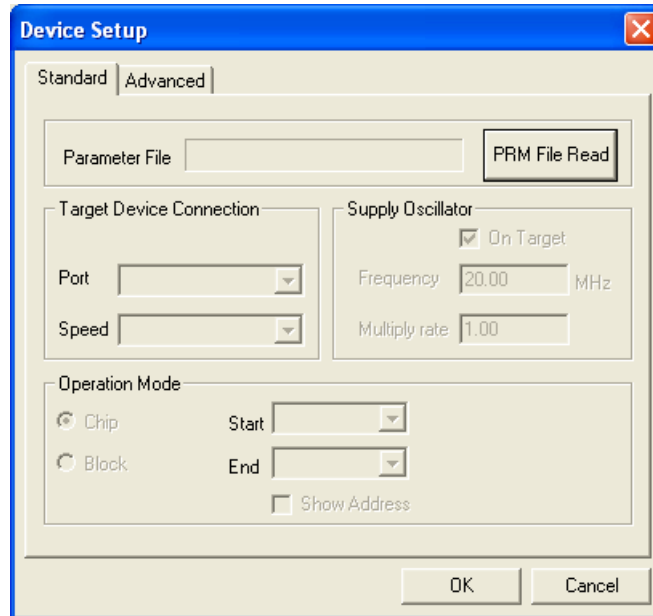
<3> Click the ⎡PRM File Read⎤ button to open the Parameter File Select dialog box. Select the parameter file for the target device and then click the ⎡Open⎤ button.

Figure 3-18. Parameter File Select Dialog Box
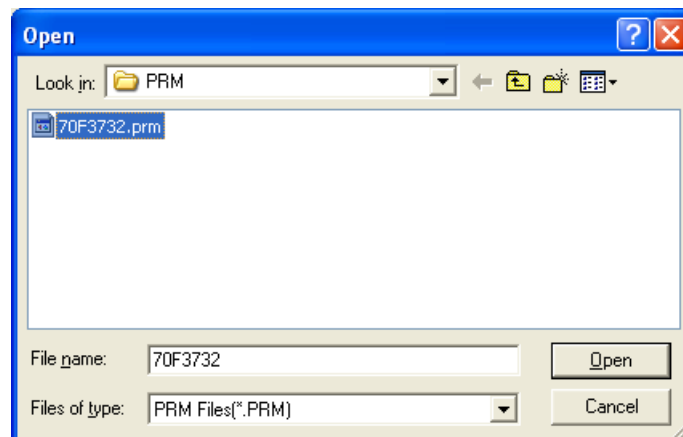
<4> Set the items in the [Target Device Connection], [Operation Mode] and [Supply Oscillator] areas, in accordance with your programming environment. The following figure shows an example.

Figure 3-19. [Standard] Tab of Device Setup Dialog Box After Setting

<5> The µPD70F3732 can use the [Get Security Settings] command, so click the OK button to return to the main window, click the [Device] menu and then click [Get Security Settings...]. After execution of this command, the security functions and boot area settings for the target device can be checked in the [Security flag settings] area on the [Advanced] tab in the Device Setup dialog box. For details, refer to the **QB-Programmer Programming GUI Operation User's Manual**.

Figure 3-20. [Advanced] Tab of Device Setup Dialog Box After [Get Security Settings...] Execution
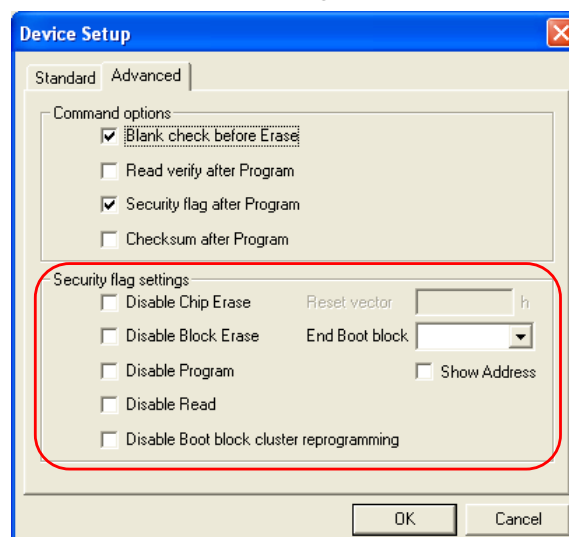
<6> Next, set the items in the [Command options] and [Security flag settings] areas, in accordance with your programming environment. The following figure shows an example.

Figure 3-21. [Advanced] Tab of Device Setup Dialog Box
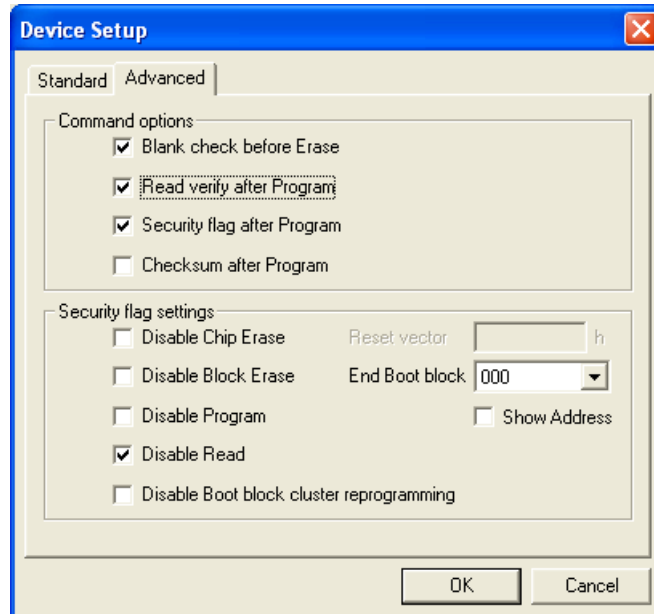
<7> By clicking the $\boxed{\text{OK}}$ button, the programming environment is set and the Device Setup dialog box is closed. The main window appears as follows.

Figure 3-22. Completion of Programming Environment Setting
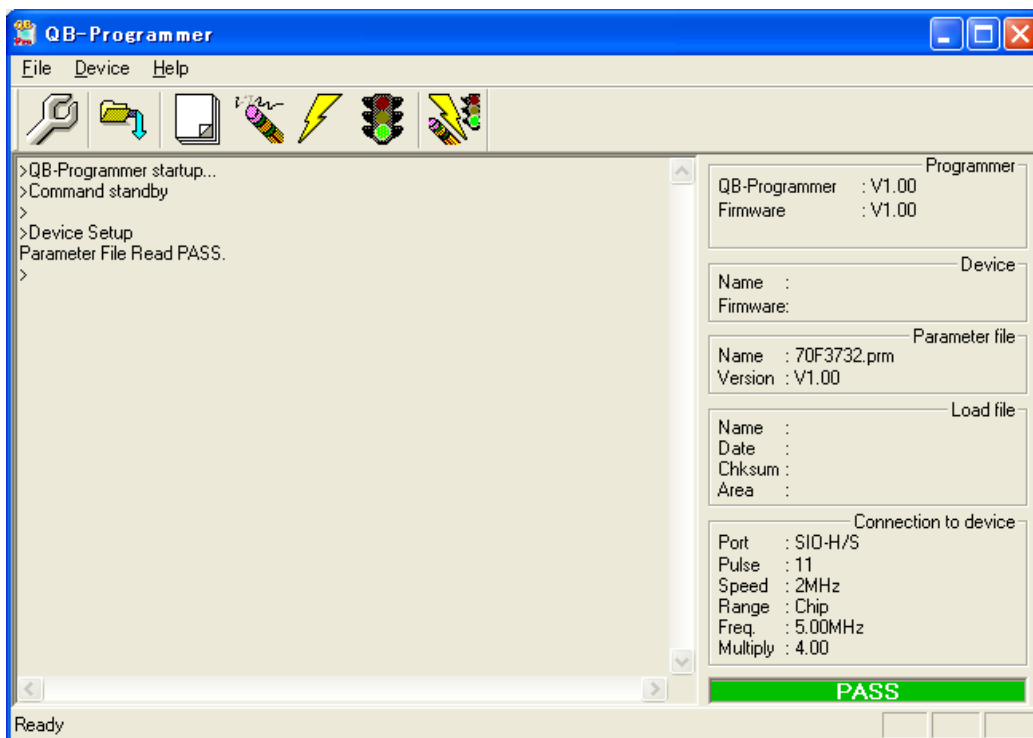
(2) Selection of program file

Select the program file, following the steps <1> to <3> described below.

<1> Click the [File] menu on the menu bar and then click [Load...].

<2> The program file select dialog box appears.

Figure 3-23. Program File Select Dialog Box
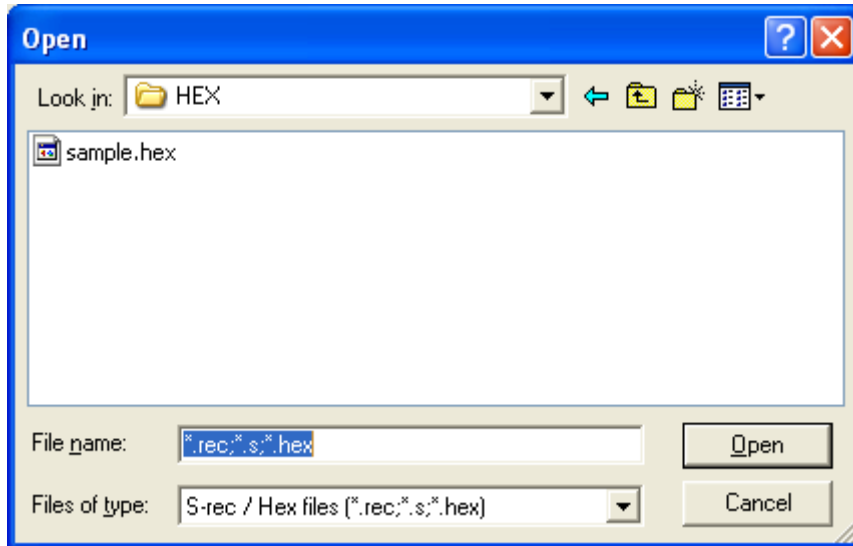
<3> Select the program file for the target device and then click the ⬚Open button.

Figure 3-24. Completion of Program File Selection
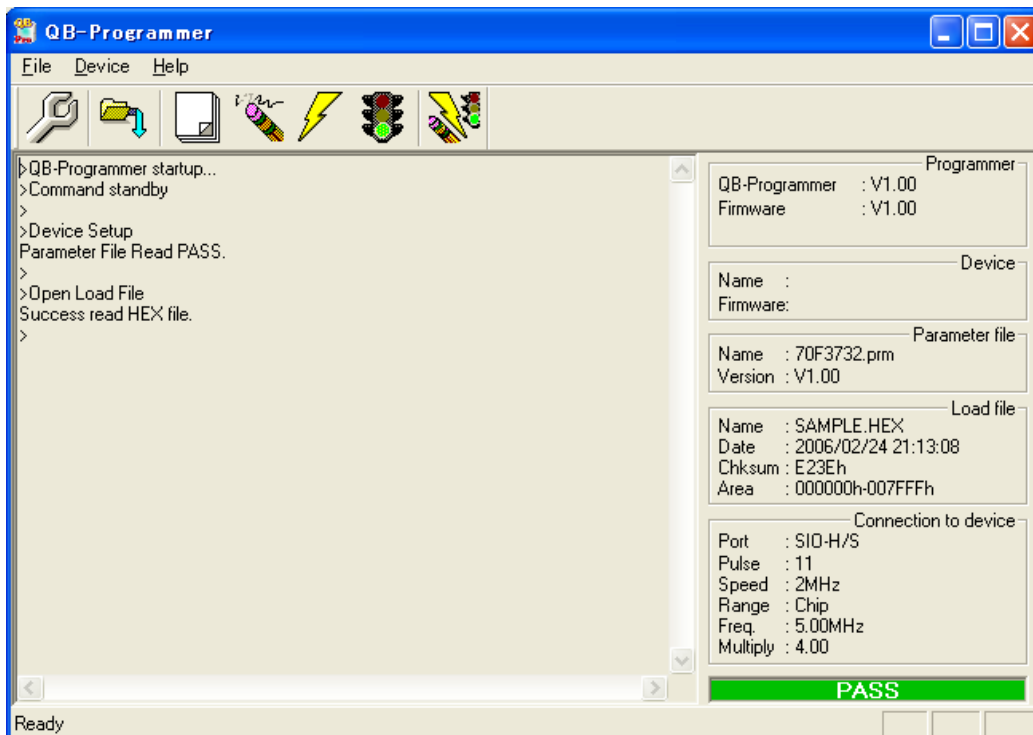
(3) Execution of [Autoprocedure (EPV)] command

Click the [Device] menu on the menu bar and then click [Autoprocedure (EPV)].

The [Blank Check], [Erase] (if the flash memory in the target device is not blank), [Program], [Verify] are executed for the target device sequentially. The mode LED on MINICUBE2 flashes yellow during execution. When the command execution is completed normally, the mode LED glows green and the following window appears.

Figure 3-25. Completion of [Autoprocedure (EPV)] Command Execution

3.3.5 System shutdown procedure

Terminate flash programming and shutdown the system in the following order.

If the following order is not observed, the target system or MINICUBE2 may be damaged.

(1) Terminating the QB-Programmer

Terminate the QB-Programmer if you are not going to perform programming to other devices.

Remark   When the QB-Programmer is terminated, the setting information is saved in the *qbp.ini* file. Due to this file, the same setting is maintained when the QB-Programmer is started the next time.

(2) Target system power shutdown

Shutdown the power supplied to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″.

(3) Unplugging the USB cable

Unplug the USB cable from MINICUBE2 or the host machine.

When the power select switch is set to ″T″, the step of **(4) Unplugging the target cable** can be performed before this step.

(4) Unplugging the target cable

Unplug the target cable from MINICUBE2 or the target system.

3.3.6 Cautions on flash programming

This section describes the cautions for flash programming. Be sure to read the following for the proper use of MINICUBE2.

• To improve the writing quality, fully understand, verify, and evaluate the following items before using MINICUBE2.
  - Circuits are designed as described in the user's manuals for the device and MINICUBE2.
  - The device, QB-Programmer and MINICUBE2 are used as described in each user's manual.
  - The power supplied to the target system is stable.

CHAPTER 4 HOW TO USE MINICUBE2 WITH 78K0 MICROCONTROLLER

This chapter describes how to use MINICUBE2 when performing on-chip debugging and flash programming for a 78K0 microcontroller.

On-chip debugging is a method to debug a microcontroller mounted on the target system, using a debug function implemented in the device. Since debugging is performed with the target device operating on the board, this method is suitable for field debugging.

Flash programming is a method to write a program to the flash memory embedded in a device. Erasing, writing and verifying the program can be performed on-board with the device.

Read the following chapters if you are using MINICUBE2 for the first time with a 78K0 microcontroller as the target device.

Caution   ID78K0-QB Ver.3.10 or later is used for the descriptions in this chapter. If using an earlier version, some settings might differ.

- 4.1 Target System Design
  For communication between MINICUBE2 and the target system, communication circuits must be mounted on the target system. This section describes the circuit design and mounting of connectors.

- 4.2 On-Chip Debugging
  This section describes the system configuration and startup method to perform on-chip debugging with MINICUBE2.

- 4.3 Flash Programming
  This section describes the system configuration and startup method to perform flash programming with MINICUBE2.

4.1 Target System Design

This section describes the target system circuit design required for on-chip debugging and flash programming.

Because the communication mode for on-chip debugging differs depending on the target device, the target system configuration varies. Check the device specifications when designing the target system.

Example When 78k0/Kx2, 78K0/Lx2, 78K0/Fx2

OCDxA, OCDxB, FLMD0 are communication for on-chip debugging

RxD, TxD, FLMD0 are communication for programming

When 78K0/Kx2-L, 78K0/Ix

TOOLCx, TOOLDx are communication for on-chip debugging and programming.

If you cannot determine the communication mode of the target device, see the manual for the target device to check whether the OCDxA and OCDxB pins or TOOLCx and TOOLDx pins are used for on-chip debugging.

Figure 4-1 and Figure 4-2 present overviews of the MINICUBE2 communication interface. For communication between MINICUBE2 and the target system, communication circuits must be mounted on the target system, as shown on the left side of the figure. Refer to this section to design circuits appropriately.

Figure 4-1. Outline of Communication Interface for On-Chip Debugging
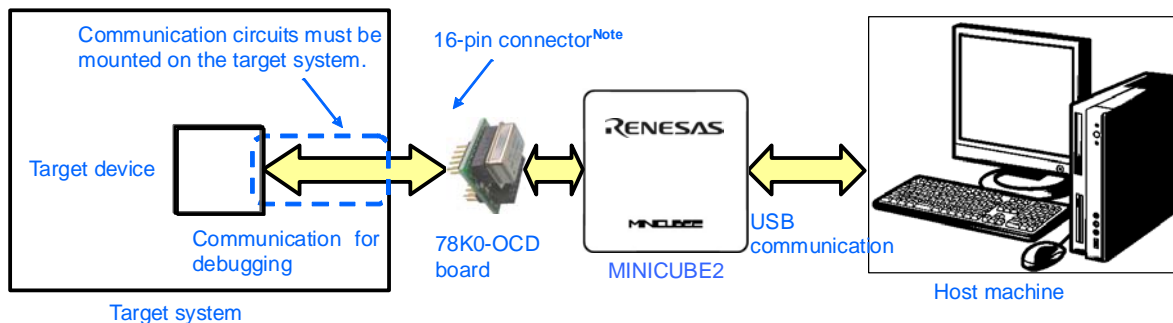
Note   To reduce the footprint of the target system, SICA can be used to reduce the size of connectors.
When replacing QB-78K0MINI (obsolete product) with MINICUBE2, a 10-pin connector dedicated to debugging can be used.

SICA is made by Tokyo Eletech Corporation.

Figure 4-2. Outline of Communication Interface for Flash Programming
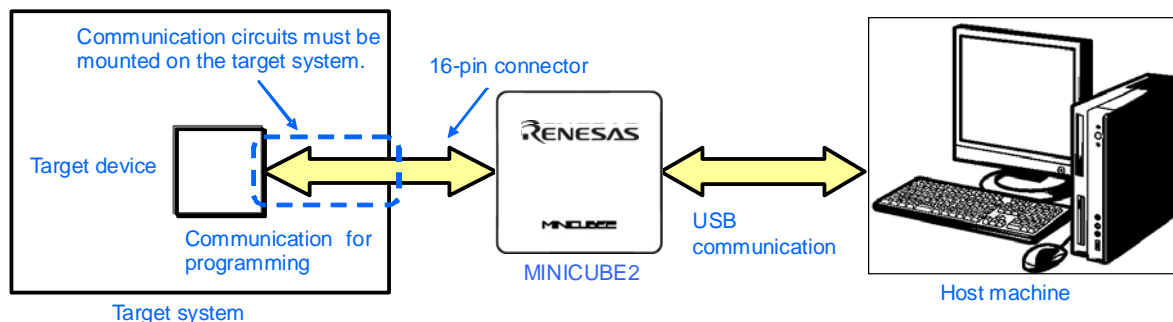
4.1.1 Pin assignment

This section describes the interface signals used between MINICUBE2 and the target system. Table 4-1 lists the pin assignment when a 16-pin target cable is used. Table 4-2 describes the functions of each pin. The pin assignment varies depending on whether the operation is debugging or programming, so design the circuit appropriately according to the circuit connection examples described on the following sections.

Table 4-1. Pin Assignment When Using 16-Pin Target Cable

	Pin No.	Pin Name[Note]	
		For Debugging (with 78K0-OCD Board Connected)	For Programming
	1	GND	
	2	RESET_OUT	
<R>	3	R.F.U.	RxD
	4	V_{DD}	
<R>	5	R.F.U.	TxD
	6	R.F.U.	
	7	R.F.U.	
	8	R.F.U.	
	9	CLK	
	10	R.F.U.	
<R>	11	R.F.U.	
	12	R.F.U.	FLMD1
	13	DATA	R.F.U.
	14	FLMD0	
<R>	15	RESET_IN	R.F.U.
	16	R.F.U.	

TOP VIEW

78K0-OCD board is not required for programming.

Pin 1

Target system

Pin assignment of target connector to be mounted on target system
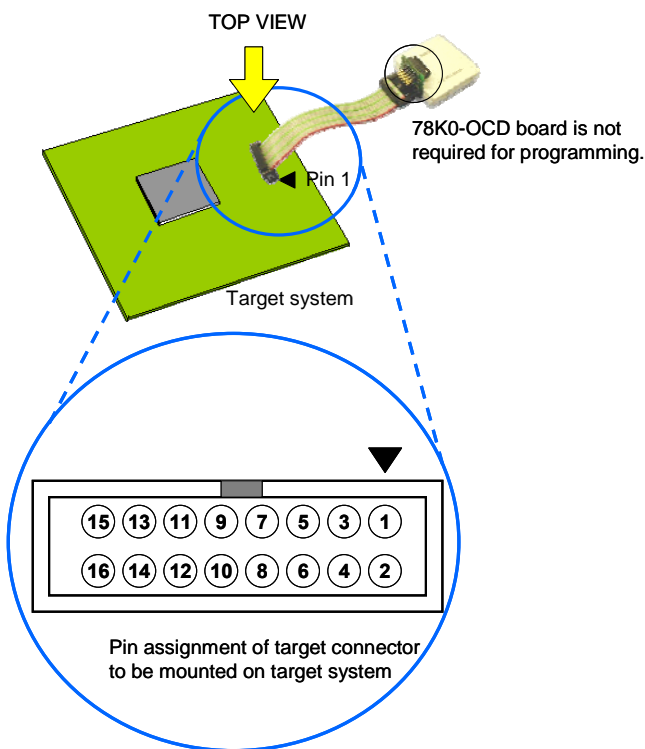
Note   Signal names in MINICUBE2

Table 4-2. Pin Functions

	Pin Name	IN/OUT[Note 1]	Description
	RESET_IN	IN	Pin used to input reset signal from the target system
	RESET_OUT	OUT	Pin used to output reset signal to the target device
	CLK	OUT	Pin used to output clock signal to the target device
	DATA	IN/OUT	Pin used to input/output for data communication during debugging
<R>	FLMD0	OUT	Pin used to set the target device to debug mode or programming mode (This pin is used by only the devices that have OCDxA and OCDxB pins.)
	FLMD1	OUT	Pin used to set programming mode[Note 2]
	RxD	IN	Pin used to receive command/data from the target device
	TxD	OUT	Pin used to transmit command/data to the target device
<R>	R.F.U.	–	This pin is reserved. For the connection of the reserved pins, see each circuit related to the pins.
<R>	N.C	–	This pin is not used, so leave it open.

Notes 1. As seen from MINICUBE2.
      2. Not used in some devices.

RENESAS

Table 4-3 lists the pin assignment when a 10-pin target cable is used. The 10-pin target cable uses the OCDxA and OCDxB pins and is dedicated to debugging. Use this cable if the target system is designed for MINICUBE (QB-78K0MINI) or in order to limit the usage of the connector mounted on the target system only for debugging.

Table 4-3. Pin Assignment When Using 10-Pin Target Cable

Pin No.	Pin Name[Note]
1	RESET_IN
2	RESET_OUT
3	FLMD0
4	V_{DD}
5	DATA
6	GND
7	CLK
8	GND
9	R.F.U.
10	N.C

Note Signal names in MINICUBE2
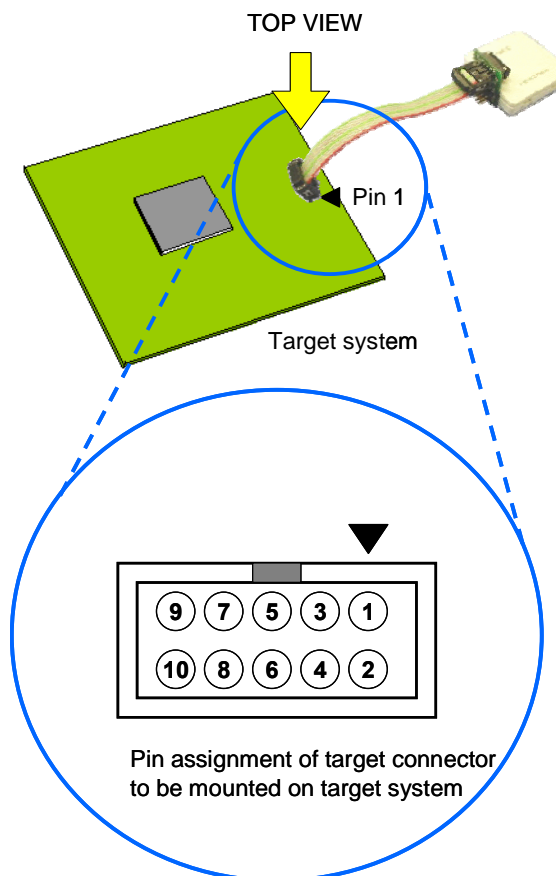
Table 4-4. Pin Functions

Pin Name	IN/OUT[Note]	Description
RESET_IN	IN	Pin used to input reset signal to the target system
RESET_OUT	OUT	Pin used to output reset signal to the target device
CLK	OUT	Pin used to output clock signal to the target device
DATA	IN/OUT	Pin used to input/output for data communication during debugging
FLMD0	OUT	Pin used to set the target device to debug mode or programming mode
R.F.U.	–	This pin is reserved. For the connection of the reserved pins, see each circuit related to the pins.
N.C	–	This pin is not used, so leave it open.

<R>

Note As seen from MINICUBE2.

4.1.2 Circuit connection examples

The circuit design on the target system varies depending on the used connector and interface signals.

The following (1) to (3) are the major purpose of use. Confirm the purpose, refer to Table 4-5 and see the relevant circuit connection example for specifications. **Figure 4-3** Flowchart for Selecting the Circuit also shows a circuit connection example.

Caution　The constants described in the circuit connection example are reference values. If you perform flash programming aiming at mass production, thoroughly evaluate whether the specifications of the target device are satisfied.

(1)　Used to perform on-chip debugging and flash programming via one 16-pin connector.
(2)　Used to perform on-chip debugging via a 10-pin connector.
　　　* This type of circuit connection can be used with on-chip debugging emulator QB-78K0MINI.
(3)　Used to perform flash programming.

Table 4-5. Corresponding Circuit Connection Diagrams

Usage	Connector Used	Programming Interface	Debug Interface	X1 oscillator circuit	Refer to:
(1)	16-pin	UART[Note1]	OCD0A(X1), OCD0B(X2)	Do not use	Figure 4-4
				Use	Figure 4-5
			OCD1A (P31), OCD1B (P32)[Note2]	Do not use	Figure 4-6
				Use	Figure 4-7
		TOOLC0, TOOLD0		Do not use	Figure 4-8
				Use	Figure 4-9
		TOOLC1, TOOLD1		–	Figure 4-10
(2)	10-pin	–	OCD0A(X1), OCD0B(X2)	–	Figure 4-11[Note3]
		–	OCD1A (P31), OCD1B (P32)[Note2]	–	Figure 4-12[Note3]
(3)	16-pin	UART[Note1]		–	Figure 4-13
		TOOLC0, TOOLD0		–	Figure 4-8
		TOOLC1, TOOLD1		–	Figure 4-10

Notes　1.　Refer to the target device's user's manual to confirm whether an FLMD1 pin is provided.
**　　　　2.**　The debug interface pin name varies depending on the target device. For details, confirm the target device's user's manual.
**　　　　3.**　If the security ID is lost (including when the area is corrupt), it can only be recovered by erasing the flash memory of the device.

Figure 4-3. Flow Chart for Selecting the Circuit
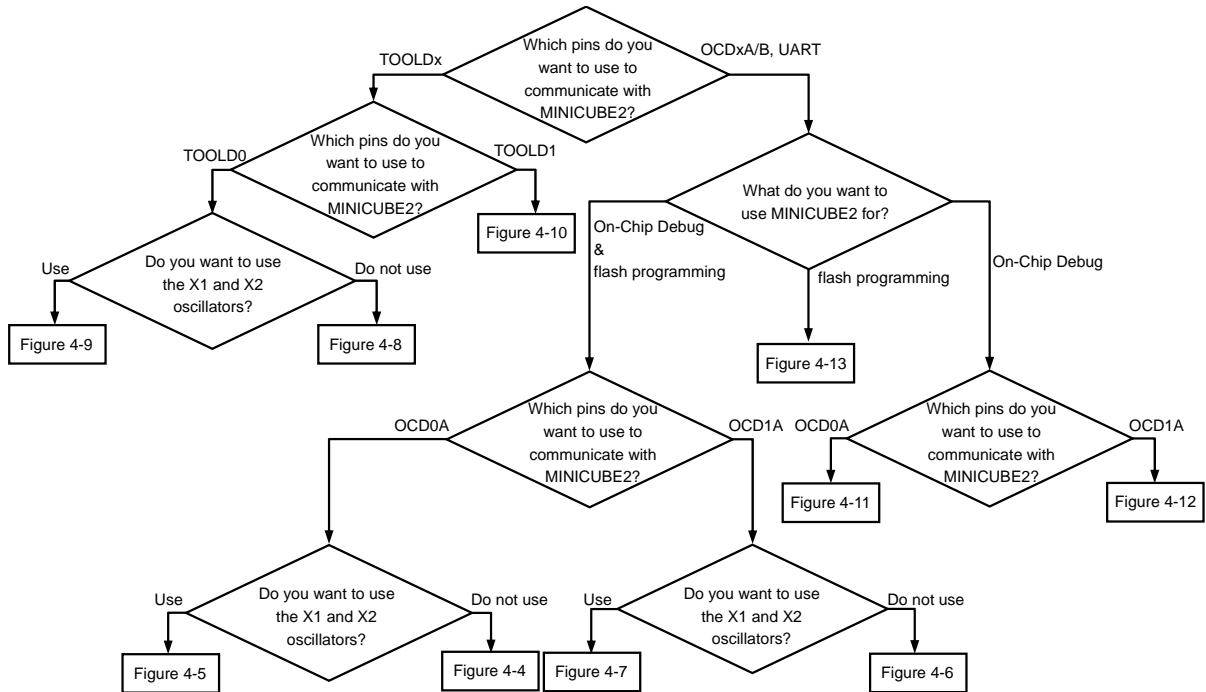
<R> **Caution The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.**

Figure 4-4. When Both Debugging and Programming Are Performed
(Not communicating by using the OCD0A and OCD0B pins or not using the X1 oscillator)
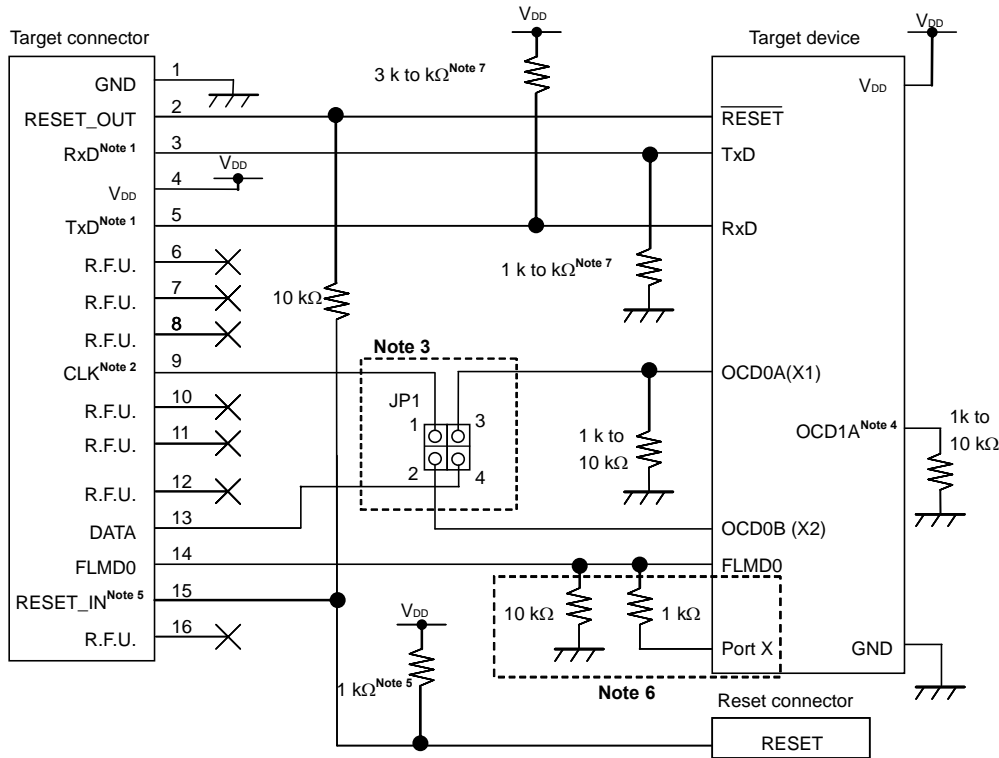
Notes 1. Connect TxD (transmit side) of the target device to RxD (receive side) of the target connector, and TxD (transmit side) of the target connector to RxD (receive side) of the target device.

2. Circuits other than the X1 oscillator can be used to generate the clock signal for the target device during on-chip debugging.

Only a 4, 8, or 16 MHz clock signal generated in MINICUBE2 can be used during flash programming.

3. Short 1 to 3 and 2 to 4 of JP1 during on-chip debugging.

Short 1 to 2 of JP1 and leave 3 to 4 of JP1 open during flash programming.

Leave JP1 open when using the target device (when MINICUBE2 is not connected).

4. The name of the pin shared with OCD1A might be invalid or might not exist depending on the device.

For details, refer to the user's manual for the target device.

5. This connection is designed assuming that the RESET signal is output from the N-ch open-drain buffer (output resistance: 100Ω or less). For details, refer to **4.1.3 Connection of reset pin**.

6. The circuit enclosed by a dashed line is designed for flash self programming, which controls the FLMD0 pin via ports.

Connect any port that can output data to FLMD0 via a resistor.

When not using flash self programming, process the pins according to the device specifications.

7. This is the processing for the pins that are unused (the inputs are left open) when the target device operates (when MINICUBE2 is not connected).

When changing the resistance, refer to **APPENDIX A EQUIVALENT CIRCUIT** and make sure that the change does not affect operation before using the device.

<R>   **Caution The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.**

Figure 4-5. When Both Debugging and Programming Are Performed
(Communicating by using the OCD0A and OCD0B pins or using the X1 oscillator)

Notes 1. Connect TxD (transmit side) of the target device to RxD (receive side) of the target connector, and TxD (transmit side) of the target connector to RxD (receive side) of the target device.

2. The X1 oscillator of the target system cannot be used to generate the clock signal for the target device during on-chip debugging.

Use the clock signal supplied from MINICUBE2. (The clock signal generated by an oscillator provided on the 78K0-OCD board or a 4, 8, or 16 MHz clock that can be selected as the system clock can also be used.)

3. Short the jumper pins and disconnect the clock circuit during on-chip debugging.

Leave the jumper pins open and connect the clock circuit when the target device operates (when MINICUBE2 is not connected) during flash programming.

4. OCD1A may be a different name, such as P31, depending on the device used. For details, refer to the user's manual for the target device.

5. This connection is designed assuming that the RESET signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **4.1.3 Connection of reset pin**.

6. The circuit enclosed by a dashed line is designed for flash self programming, which controls the FLMD0 pin via ports.

Connect any port that can output data to FLMD0 via a resistor.

When not using flash self programming, process the pins according to the device specifications.

7. This is the processing for the pins that are unused (the inputs are left open) when the target device operates (when MINICUBE2 is not connected).

When changing the resistance, refer to **APPENDIX A EQUIVALENT CIRCUIT** and make sure that the change does not affect operation before using the device.

8. To use FLMD1, connect it to FLMD1 of MINICUBE2.

<R> **Caution The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.**

**Figure 4-6. When Both Debugging and Programming Are Performed
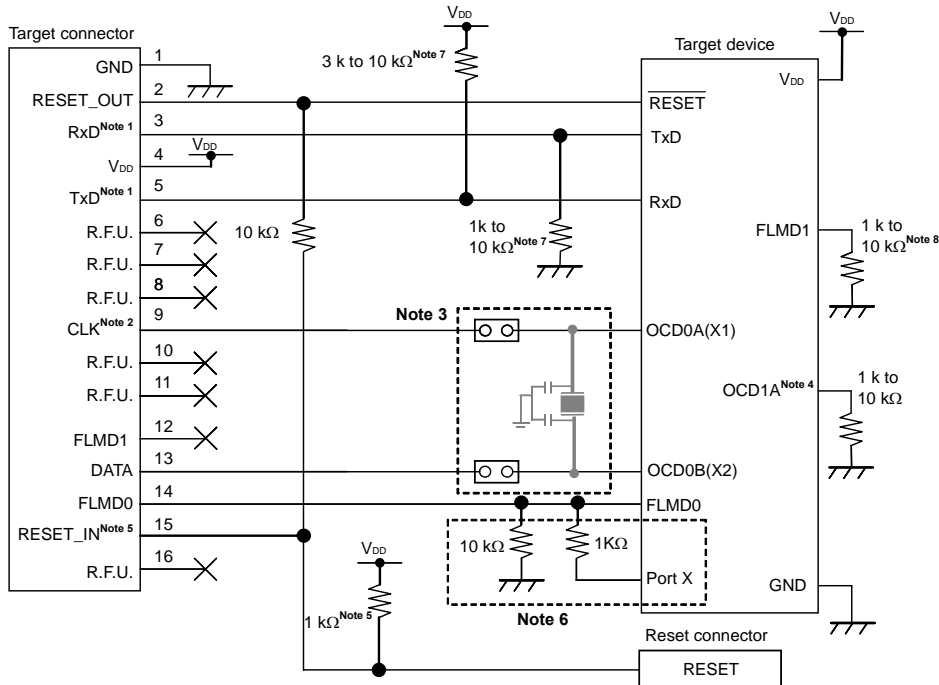(Not communicating by using the OCD1A and OCD1B pins or using the X1 oscillator)**
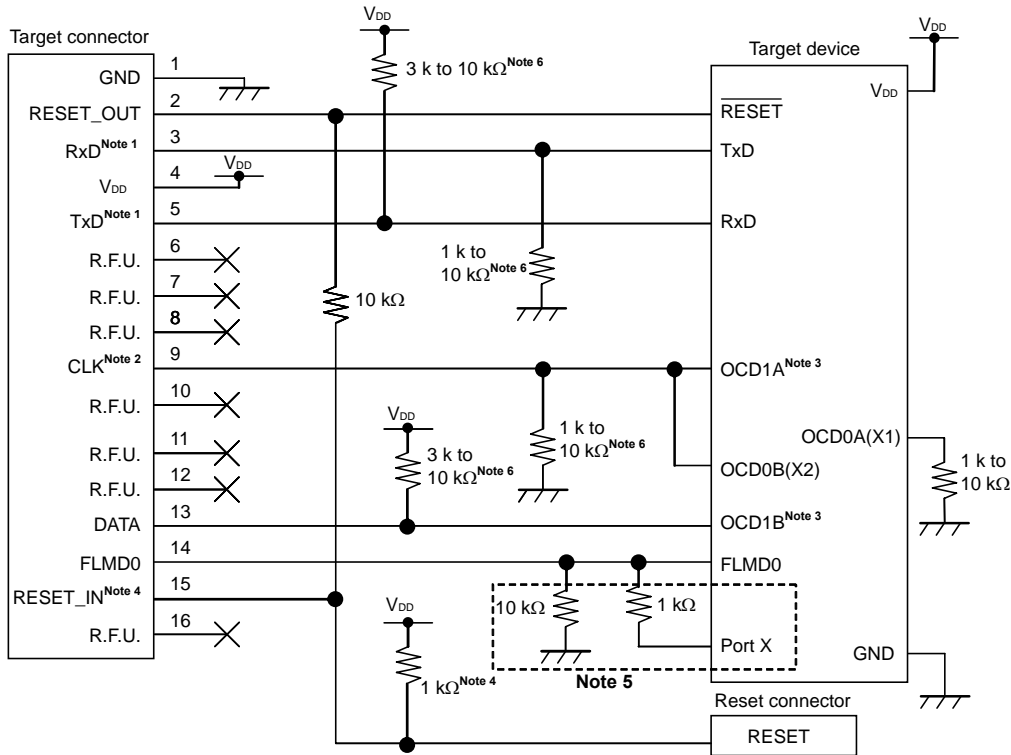
Notes 1. Connect TxD (transmit side) of the target device to RxD (receive side) of the target connector, and TxD (transmit side) of the target connector to RxD (receive side) of the target device.

    2. Circuits other than the X1 oscillator can be used to generate the clock signal for the target device during on-chip debugging.

    3. The name of the pin shared with OCD1A (OCD1B) might be invalid or might not exist depending on the device. For details, see the user's manual for the target device.

    During on-chip debugging, the settings specified by the user program are ignored, because these pins are used as pins dedicated to on-chip debugging. However, if the pins are specified as input pins, the pins must be processed (because they are left open when MINICUBE2 is not connected.)

    4. This connection is designed assuming that the RESET signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **4.1.3 Connection of reset pin**.

    5. The circuit enclosed by a dashed line is designed for flash self programming, which controls the FLMD0 pin via ports.

    When not using flash self programming, process the pins according to the device specifications.

    6. This is the processing for the pins that are unused (the inputs are left open) when the target device operates (when MINICUBE2 is not connected).

    When changing the resistance, refer to **APPENDIX A EQUIVALENT CIRCUIT** and make sure that the change does not affect operation before using the device.

<R>　　**Caution The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.**

Figure 4-7. When Both Debugging and Programming Are Performed
(Communicating by using the OCD1A and OCD1B pins or using the X1 oscillator)
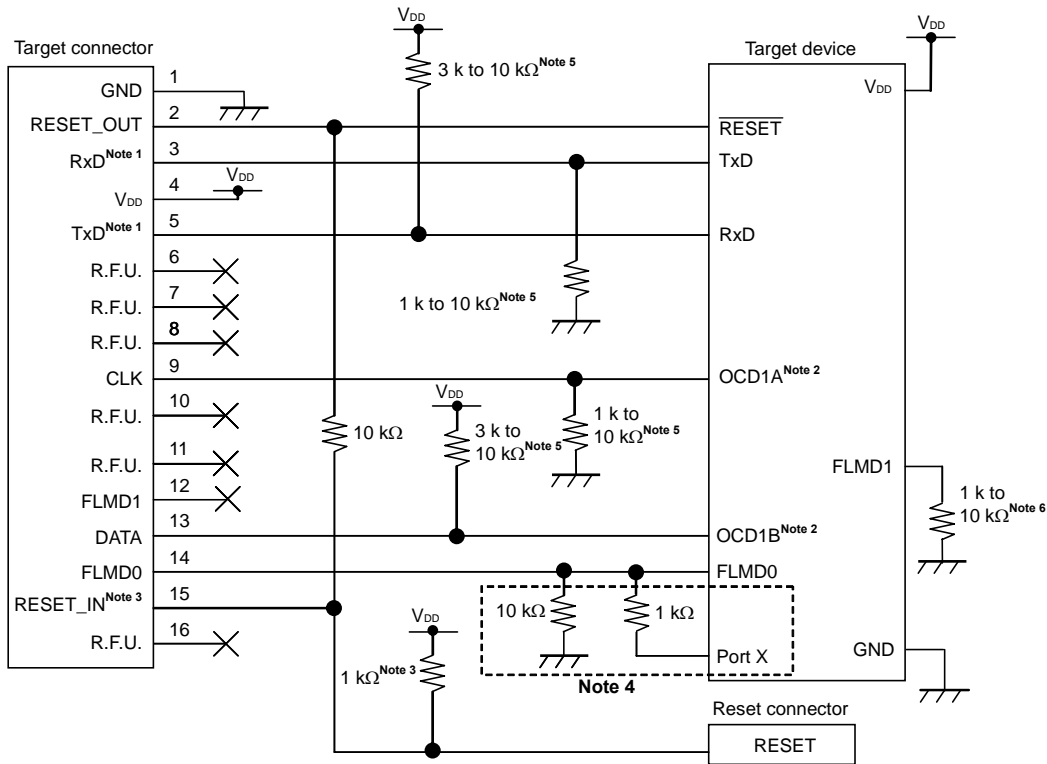
Notes 1. Connect TxD (transmit side) of the target device to RxD (receive side) of the target connector, and TxD (transmit side) of the target connector to RxD (receive side) of the target device.

  2. OCD1A (OCD1B) may be a different name, such as P31, depending on the device used. Check the pin name with the user's manual for the target device. Since this pin is dedicated to debugging during debugging, port settings made by the user program are ignored. If this pin is assigned to an input port, the pin must be connected as shown in this figure, because it is open when MINICUBE2 is not connected.

  3. This connection is designed assuming that the RESET signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **4.1.3 Connection of reset pin**.

  4. The circuit enclosed by a dashed line is designed for flash self programming, which controls the FLMD0 pin via ports.

  When not using flash self programming, process the pins according to the device specifications.

  5. This is the processing for the pins that are unused (the inputs are left open) when the target device operates (when MINICUBE2 is not connected).

  When changing the resistance, refer to **APPENDIX A EQUIVALENT CIRCUIT** and make sure that the change does not affect operation before using the device.

  6. To use FLMD1, connect it to FLMD1 of MINICUBE2.

<R>   **Caution The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.**

Figure 4-8. When Both Debugging and Programming Are Performed
(with X1/X2 oscillator is not used , with TOOLC0/TOOLD0 communication)

Notes 1. If there are capacitance elements such as capacitors, on-chip debugging might not operate normally.
   2. A clock signal provided on the 78K0-OCD board, a 4, 8, or 16 MHz clock signal generated in MINICUBE2, or the clock signal generated by the internal high-speed oscillator of the device can be used for the clock signal of the target device during on-chip debugging.

   Only the internal high-speed oscillator of the device can be used during flash programming.
   3. During on-chip debugging, the settings specified by the user program are ignored, because these pins are used as pins dedicated to on-chip debugging. However, if the pins are specified as input pins, the pins must be processed (because they are left open when MINICUBE2 is not connected.)
   4. This is the processing for the pins that are unused (the inputs are left open) when the target device operates (when MINICUBE2 is not connected). Output setting or an oscillator circuit is not used.

   When changing the resistance, refer to **APPENDIX A EQUIVALENT CIRCUIT** and make sure that the change does not affect operation before using the device.
   5. This connection is designed assuming that the Reset connector is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **4.1.3 Connection of reset pin**.

<R>

Caution The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.
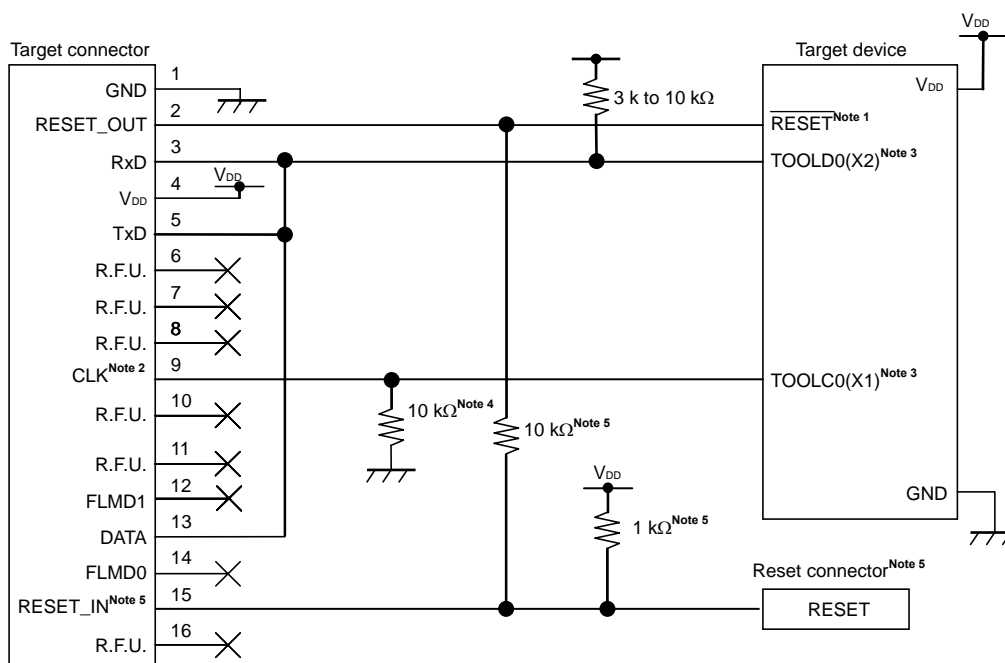
Figure 4-9. When Both Debugging and Programming Are Performed
(with X1/X2 oscillator is used, with TOOLC0/TOOLD0 communication)

Notes 1. If there are capacitance elements such as capacitors, on-chip debugging might not operate normally.
   2. A clock signal provided on the 78K0-OCD board, a 4, 8, or 16 MHz clock signal generated in MINICUBE2, or the clock signal generated by the internal high-speed oscillator of the device can be used for the clock signal of the target device during on-chip debugging.
   Only the internal high-speed oscillator of the device can be used during flash programming.
   3. During on-chip debugging, the settings specified by the user program are ignored, because these pins are used as pins dedicated to on-chip debugging. However, if the pins are specified as input pins, the pins must be processed (because they are left open when MINICUBE2 is not connected.)
   4. This connection is designed assuming that the RESET signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **4.1.3 Connection of reset pin**.
   5. Never connect an oscillation circuit to the 78K0-OCD board during on-chip debugging and flash programming.
   To prevent an oscillation circuit from not oscillating due to wiring capacitance when the target device operates (when MINICUBE2 is not connected), also consider countermeasures such as disconnecting the oscillation circuit from the target connectors by setting the jumpers.
   A program that was downloaded using the debugger does not operate when MINICUBE2 is not connected.

Caution   The red paths in the figure (TOOLD0 and TOOLC0) must be designed so that the device pins are less than 30 mm from the MINICUBE2 connectors or the paths must be shielded by connecting them to GND.

<R>   **Caution The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.**

Figure 4-10. When Both Debugging and Programming Are Performed
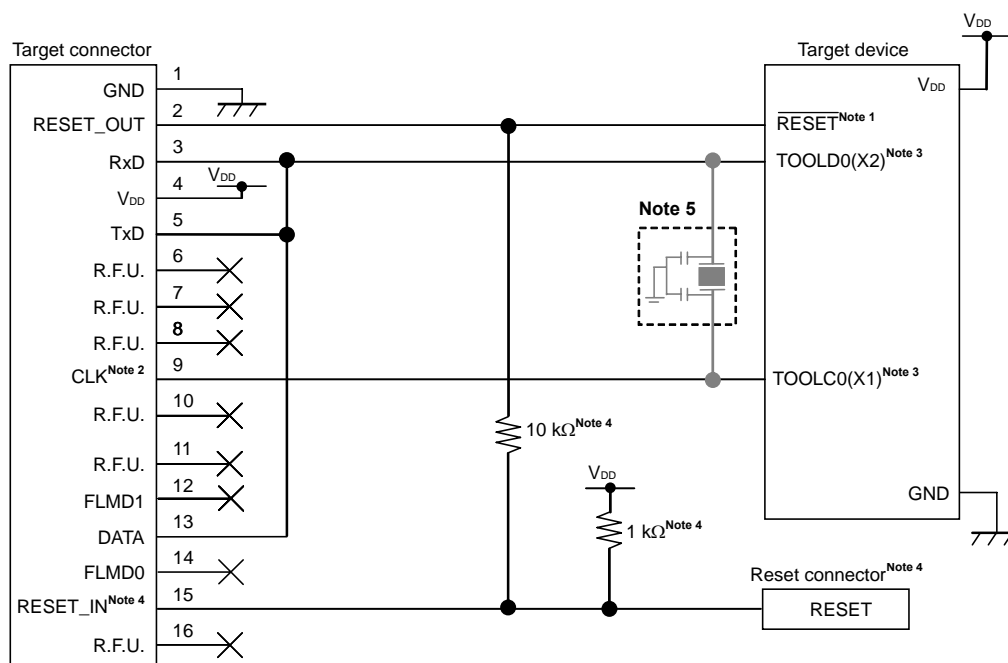**(with TOOLC0/TOOLD0 communication)**
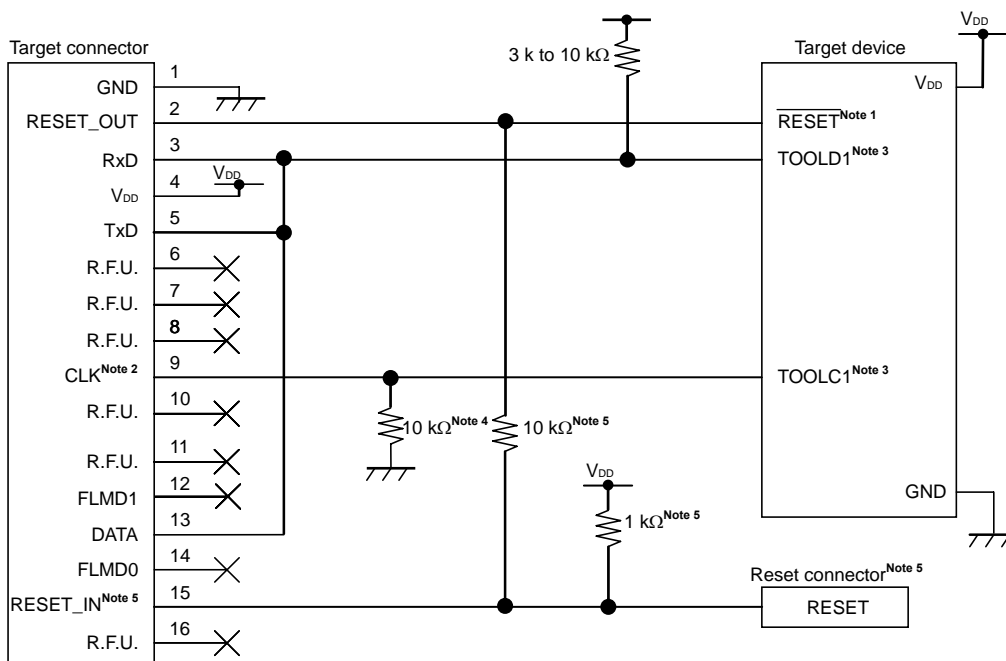
Notes **1.** If there are capacitance elements such as capacitors, on-chip debugging might not operate normally.

2. The clock signal generated by the clock circuit on the target system or by the internal high-speed oscillator of the device can be used for the clock signal of the target device during on-chip debugging.

Only the internal high-speed oscillator of the device can be used during flash programming.

3. During on-chip debugging, the settings specified by the user program are ignored, because these pins are used as pins dedicated to on-chip debugging. However, if the pins are specified as input pins, the pins must be processed (because they are left open when MINICUBE2 is not connected.)

4. This is the processing for the pins that are unused (the inputs are left open) when the target device operates (when MINICUBE2 is not connected). (This processing is not required if the pins are set to output.)

When changing the resistance, refer to **APPENDIX A EQUIVALENT CIRCUIT** and make sure that the change does not affect operation before using the device.

5. This connection is designed assuming that the RESET signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **4.1.3 Connection of reset pin**.

<R> **Caution The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.**

Figure 4-11. When Only Debugging Is Performed (with OCD0A/OCD0B communication)

Notes **1.** This connection is designed assuming that the RESET signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **4.1.3 Connection of reset pin**.

2. For debugging, mount no clock circuit. For flash programming, mount a clock circuit.

3. The clock signal provided on the 78K0-OCD board or a 4, 8, or 16 MHz clock signal generated in MINICUBE2 can be used for the clock signal of the target device during on-chip debugging.

4. The circuit enclosed by a dashed line is designed for flash self programming, which controls the FLMD0 pin via ports. (This circuit is used to control the port during flash programming.)
Connect any port that can output data to FLMD0 via a resistor.
When not using flash self programming, process the pins according to the device specifications.

5. The name of the pin shared with OCD1A (OCD1B) might be invalid or might not exist depending on the device. For details, refer to the user's manual for the target device.

6. This is the processing for the pins that are unused (the inputs are left open) when the target device operates (when MINICUBE2 is not connected).
When changing the resistance, refer to **APPENDIX A EQUIVALENT CIRCUIT** and make sure that the change does not affect operation before using the device.

<R>   **Caution The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.**
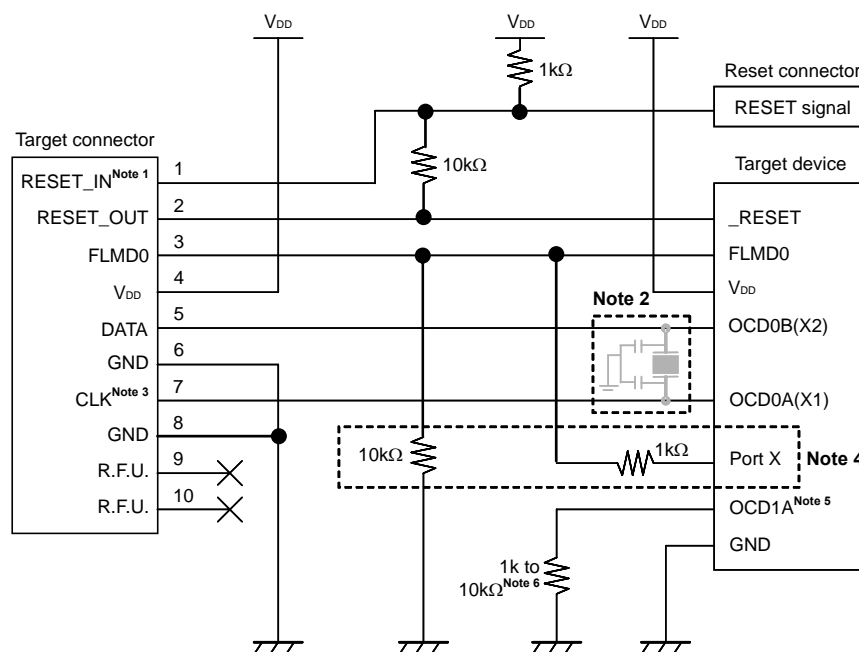
Figure 4-12. When Only Debugging Is Performed (with OCD1A/OCD1B communication)

Notes 1. This connection is designed assuming that the RESET signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **4.1.3 Connection of reset pin**.

     2. The name of the pin shared with OCD1A (OCD1B) might be invalid or might not exist depending on the device. For details, see the user's manual for the target device.

        During on-chip debugging, the settings specified by the user program are ignored, because these pins are used as pins dedicated to on-chip debugging. However, if the pins are specified as input pins, the pins must be processed (because they are left open when MINICUBE2 is not connected.)

     3. The clock signal generated by the internal high-speed oscillator of the device or a 4, 8, or 16 MHz clock signal generated in MINICUBE2 can be used for the clock signal of the target device during on-chip debugging.

     4. The circuit enclosed by a dashed line is designed for flash self programming, which controls the FLMD0 pin via ports. (This circuit is used to control the port during flash programming.)

        Connect any port that can output data to FLMD0 via a resistor.

        When not using flash self programming, process the pins according to the device specifications.

     5. This is the processing for the pins that are unused (the inputs are left open) when the target device operates (when MINICUBE2 is not connected).

        When changing the resistance, refer to **APPENDIX A EQUIVALENT CIRCUIT** and make sure that the change does not affect operation before using the device.

<R> **Caution The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.**
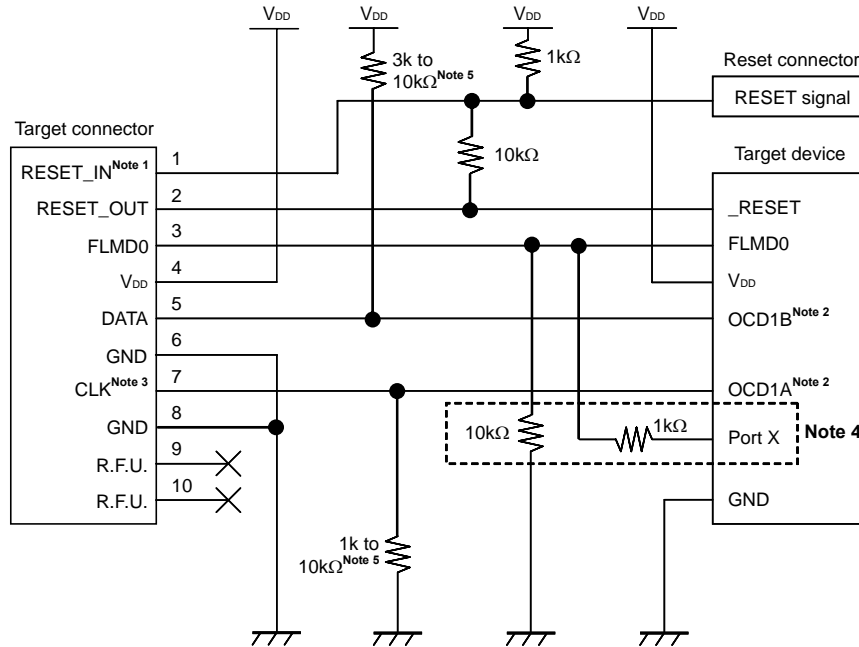
Figure 4-13. When Only Programming Is Performed
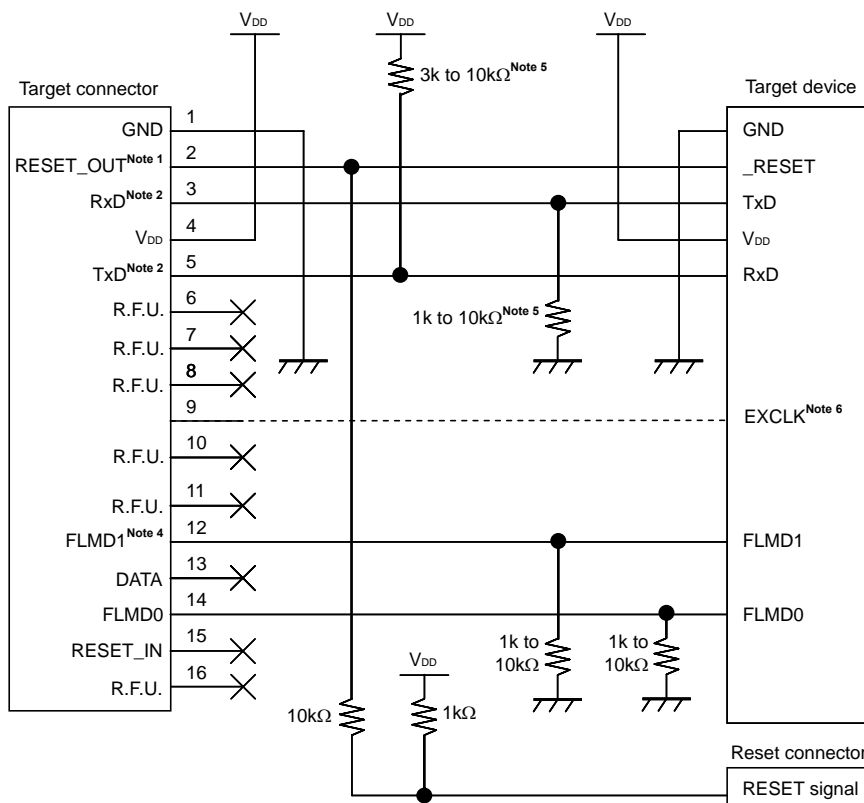
Notes **1.** This connection is designed assuming that the RESET signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **4.1.3 Connection of reset pin**.

    2. Connect TxD (transmit side) of the target device to RxD (receive side) of the target connector, and TxD (transmit side) of the target connector to RxD (receive side) of the target device.

    3. When the input clock to the target device is generated on the target system, CLK does not need to be

<R>   connected. Please leave it open when no connection is made. When no input clock is generated, or the dedicated program adapter (FA series) is used, be sure to connect CLK as external clock supply to the target device

    4. If FLMD1 pin is provided on the target device, the connection must be made. If FLMD1 pin is not provided on the target device, leave it open.

    5. This is for pin processing when not used as a device.

         To change this value to another resistance value, refer to **APPENDIX A EQUIVALENT CIRCUIT**.

<R>   **6.** In some cases, a pin name of EXCLK may differ depending on devices. For the details, see the user's manual related to the target device.

4.1.3 Connection of reset pin

This section describes the connection of the reset pin, for which special attention must be paid, in circuit connection examples shown in the previous section.

During on-chip debugging, a reset signal from the target system is input to MINICUBE2, masked, and then output to the target device. Therefore, the reset signal connection varies depending on whether MINICUBE2 is connected.

For flash programming, the circuit must be designed so that the reset signals of the target system and MINICUBE2 do not conflict.

Select one of the following methods and connect the reset signal in the circuit. The details of each method are described on the following pages.

(1) Automatically switching the reset signal via resistor (recommended; described in recommended circuit connection in the previous section)

(2) Automatically switching the reset signal via selector logic

(3) Manually switching the reset signal with jumper

(4) Resetting the target device by power-on clear (POC) only

<R>   **Caution   Do not connect a capacitor to the RESET_OUT pin.**

<R>   **Figure 4-14. Cautions for using the RESET-OUT pin**
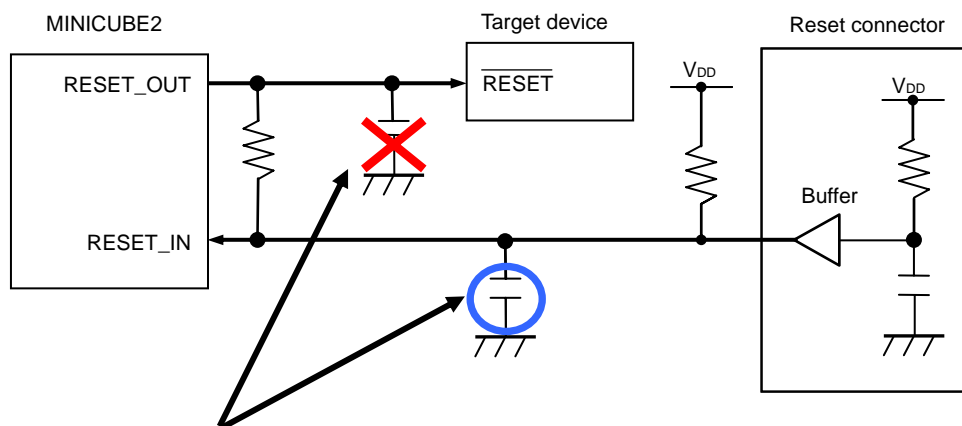
If a capacitance element such as a capacitor or resistor is
connected to the RESET_OUT pin, the target device might not
be able to communicate with MINICUBE2.
If necessary, connect a capacitor to the RESET_IN pin.

(1) Automatically switching the reset signal via resistor

Figure 4-15 illustrates the reset pin connection described in **4.1.2 Circuit connection examples**.
This connection is designed assuming that the reset circuit on the target system contains an N-ch open-drain
buffer (output resistance: 100Ω or less). The VDD or GND level may be unstable when the logic of
RESET_IN/OUT of MINICUBE2 is inverted, so observe the conditions described below in **Remark**.

Figure 4-15. Circuit Connection with Reset Circuit That Contains Buffer
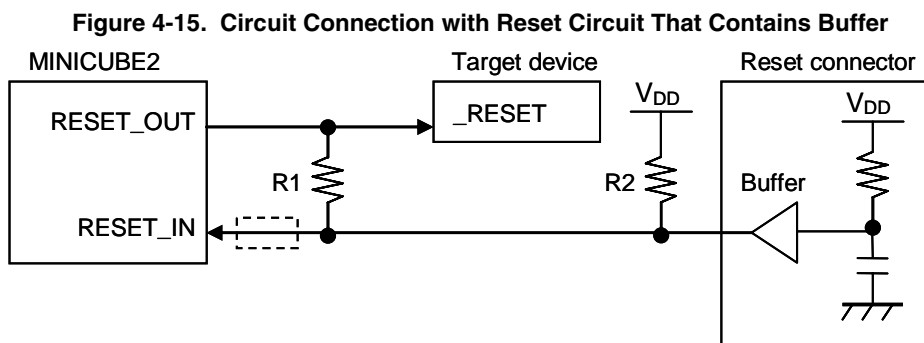
Remark   Make the resistance of at least R1 ten times that of R2, R1 being 10 kΩ or more.
Pull-up resistor R2 is not required if the buffer of the reset circuit consists of CMOS output.
The circuit enclosed by a dashed line is not required when only flash programming is performed.

Figure 4-16 illustrates the circuit connection for the case where the reset circuit on the target system contains no
buffers and the reset signal is only generated via resistors or capacitors. Design the circuit, observing the
conditions described below in **Remark**.

Figure 4-16. Circuit Connection with Reset Circuit That Contains No Buffers
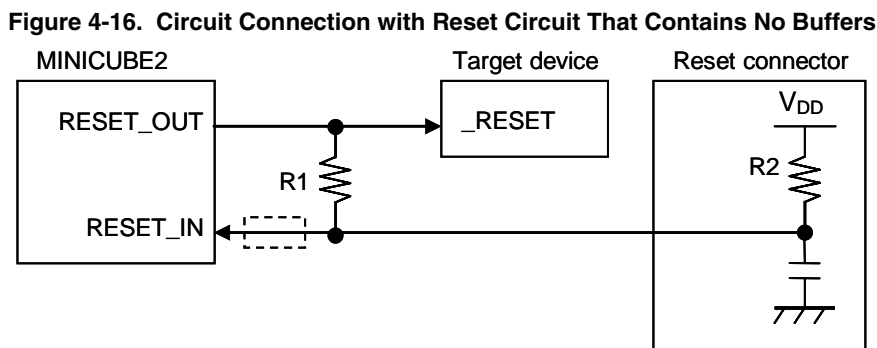
Remark   Make the resistance of at least R1 ten times that of R2, R1 being 10 kΩ or more.
The circuit enclosed by a dashed line is not required when only flash programming is performed.

<R>   **(2) Automatically switching the reset signal via selector logic (Only for devices that have no FLMDPUP register but have a FLMD0 pin)**
        Figure 4-17 illustrates the circuit connection for the case where the reset signal is switched automatically using the selector logic, with or without MINICUBE2 connected. When using flash self programming, refer to Figure 4-18.

Caution   The reset circuit is disconnected if flash self programming is performed (FLMD0 = HI) without MINICUBE2 connected, so no signals can be input to the $\overline{\text{RESET}}$ pin.

Figure 4-17. Circuit Connection for Automatically Switching Reset Signal with Selector Logic (When Not Using Flash Self Programming)
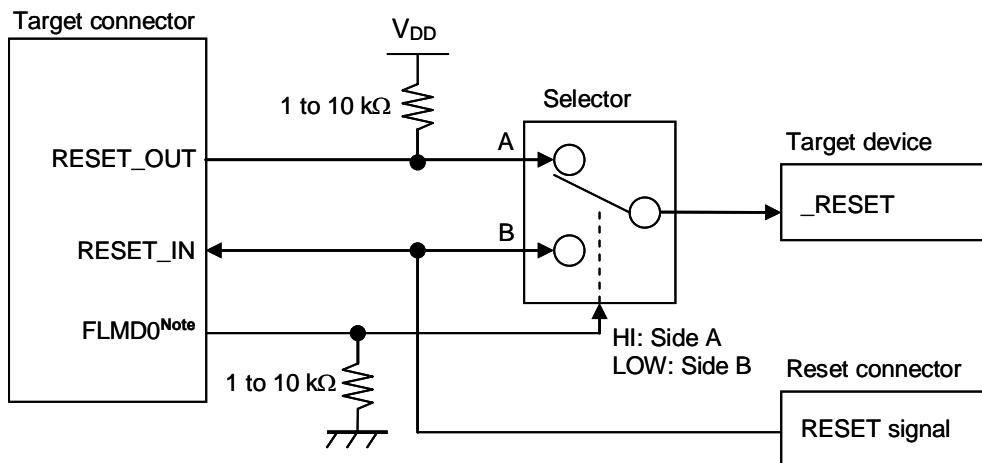
Note   FLMD0 is high level when MINICUBE2 is connected, and FLMD0 is pulled down when MINICUBE2 is not connected.

Figure 4-18. Circuit Connection for Automatically Switching Reset Signal with Selector Logic (When Using Flash Self Programming)
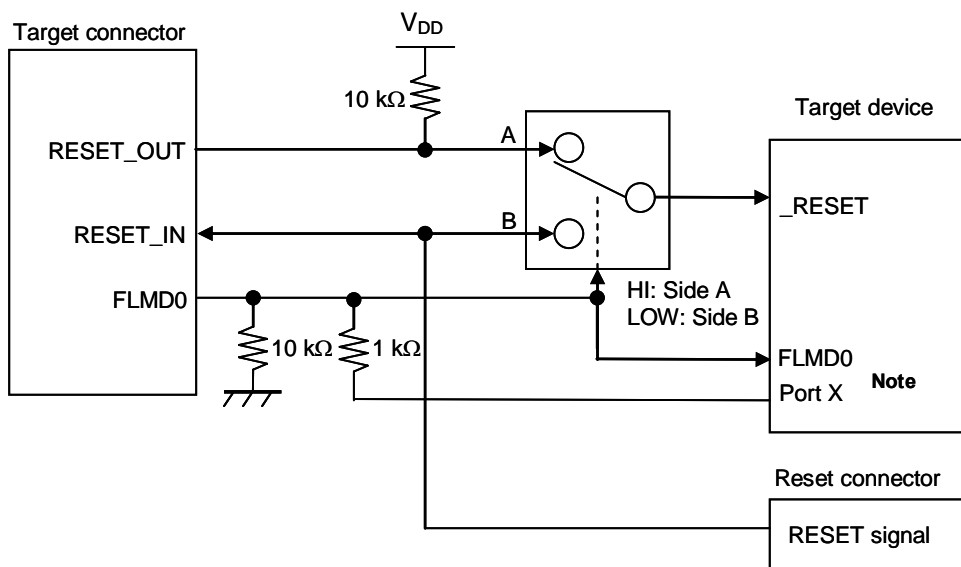
Note   This circuit is designed for flash self programming, which controls the FLMD0 pin via ports.

(3) Manually switching the reset signal with jumper

Figure 4-19 illustrates the circuit connection for the case where the reset signal is switched using the jumper, with or without MINICUBE2 connected. This connection is simple, but the jumper must be set manually.

Figure 4-19. Circuit Connection for Switching Reset Signal with Jumper
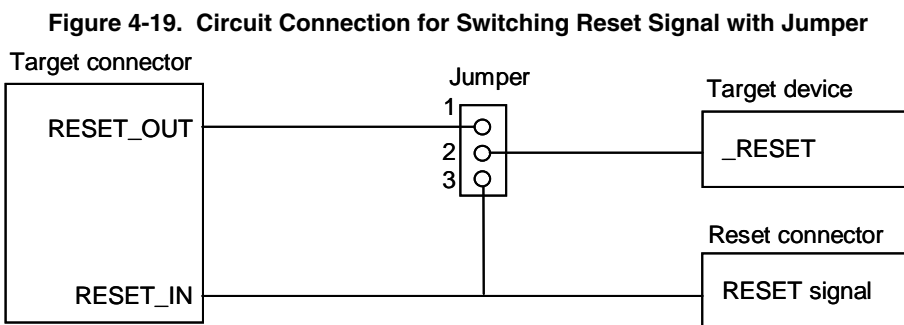
* Jumper setting

When MINICUBE2 is connected: 1-2 shorted

When MINICUBE2 is not connected: 2-3 shorted

When a 10-pin connector is used, pins 1 and 2 of the target connector can also be used as the jumper, as shown in Figure 4-20.

Figure 4-20. Jumper Connection When 10-Pin Connector Is Used
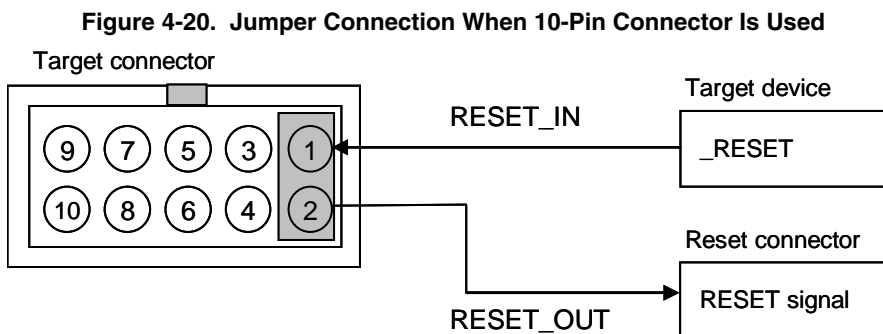
(4) Resetting the target device by power-on clear (POC) only

Figure 4-21 illustrates the circuit connection for the case where the target device is only reset via POC without using the reset pin. RESET_OUT becomes active when power is applied to MINICUBE2.

<R>　Even if power supply to the target system is turned off during debugging, pseudo POC function emulation is available because RESET_OUT becomes active. Note that the POC detection voltage is about 1 V higher than that of the target device.

Figure 4-21. Circuit Connection for the Case Where Target Device Is Only Reset via POC
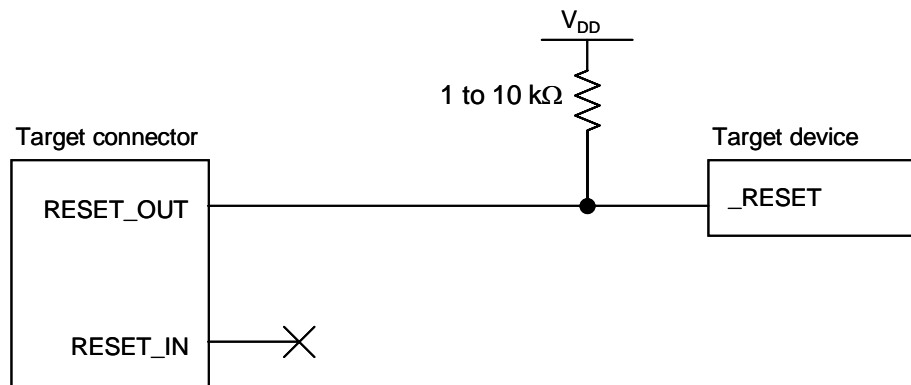
4.1.4 Cautions on target system design

Note the following cautions when designing the target system.

<R>
- If possible, do not create sections in which the communication lines for debugging (OCDxA, OCDxB, TOOLCx, and TOOLDx) run in parallel in the target system. If this cannot be prevented, shorten the sections as much as possible. When using TOOLC0 and TOOLD0 as communication lines for debugging by using X1 or X2 oscillation, be sure to make the section where TOOLC0 and TOOLD0 run in parallel less than 30 mm.

<R>
- Use a GND pattern to shield the communication lines for debugging (OCDxA, OCDxB, TOOLCx, and TOOLDx) to reduce their capacitive load, because the lines are used for high-speed communication.
- Make the distance between the target connector and the target device as short as possible.
- Before shipping the product, use jumpers or other means to physically separate the X1 and X2 pins from the target connector in order to ensure normal clock oscillation.

<R>
- To use OCD0A, OCD0B, TOOLC0, and TOOLD0 as communication pins for debugging, remove elements such as resonator capacitance and feedback resistors, so that the signals do not degrade due to capacitive load.

Figure 4-22. Reference Diagram of Target System
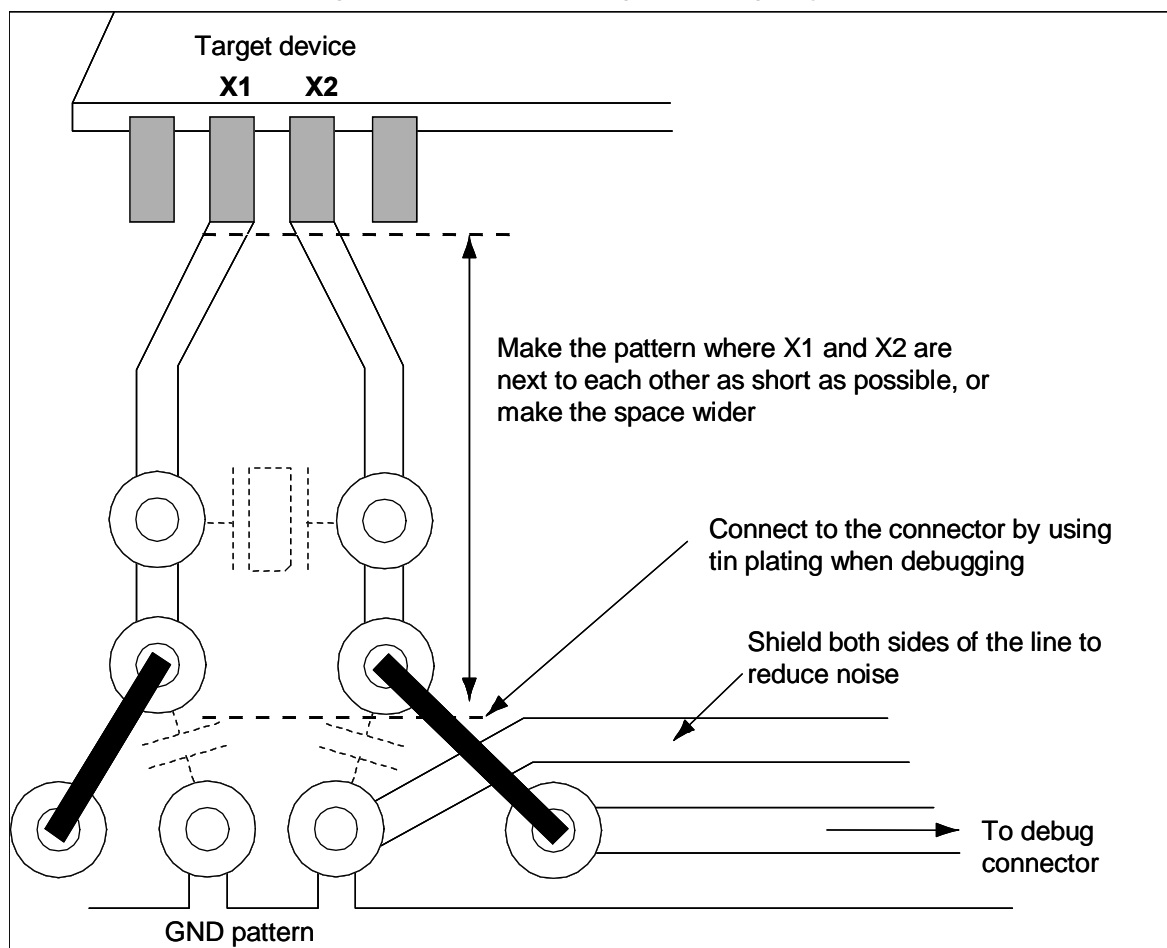
4.1.5 Mounting connector onto target system

The connector must be mounted on the target system for connecting MINICUBE2 with the target system. A 2.54 mm pitch 10- or 16-pin general-purpose connector can be used. The following products are recommended.

• HIF3FC-10PA-2.54DS (10-pin right-angle type, made by Hirose Electric Co., Ltd.)

• HIF3FC-10PA-2.54DSA (10-pin straight type, made by Hirose Electric Co., Ltd.)

• HIF3FC-16PA-2.54DS (16-pin right-angle type, made by Hirose Electric Co., Ltd.)

• HIF3FC-16PA-2.54DSA (16-pin straight type, made by Hirose Electric Co., Ltd.)

• 7616-5002PL (16-pin right-angle type, made by Sumitomo 3M, Ltd.)

• 7616-6002PL (16-pin straight type, made by Sumitomo 3M, Ltd.)

Compact connectors and loose-wire cables are also available as options. For more information on these products, access the MINICUBE2 information site.

URL: http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html

Figure 4-23. 2.54 mm Pitch 10-/16-Pin General-Purpose Connector Pin Layout
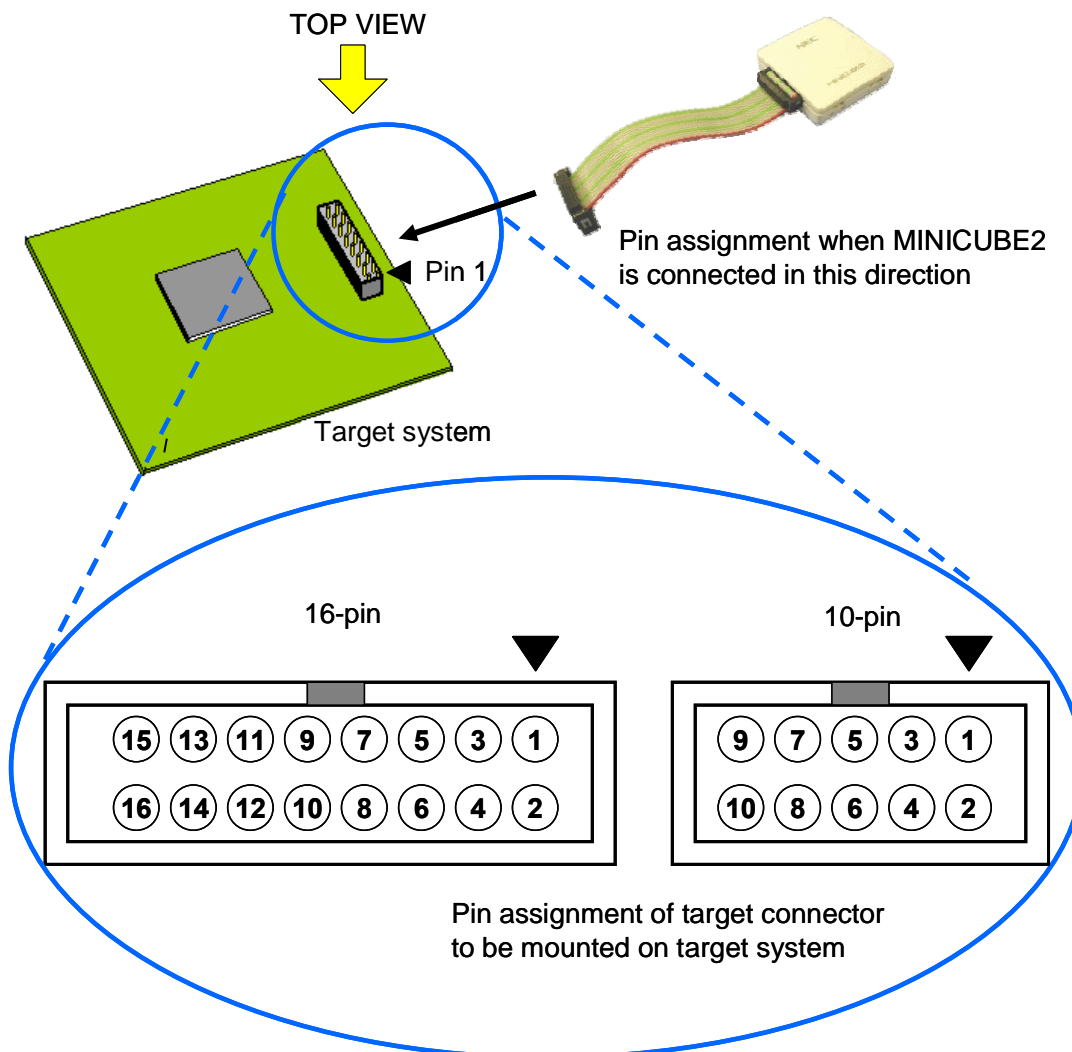
4.2 On-Chip Debugging

This section describes the system configuration, startup/shutdown procedure and cautions for debugging when on-chip debugging is performed with MINICUBE2.

4.2.1 Debug functions

Table 4-6 lists the debug functions when a 78K0 microcontroller is the target device and the ID78K0-QB is used. If a debugger other than the ID78K0-QB (made by partner manufacturer) is used, the available functions may differ, so check the specifications of the debugger used

Table 4-6. Debug Functions

Functions	Specifications
Security	10-byte ID code authentication
Download	Available
Execution	Go & Go, Start from Here, Come Here, Restart, step execution
Hardware break	Break before execution: 1 point (unavailable when using software break)
	Access break: 1 point
Software break	2,000 points
Forced break	Available
Pseudo real-time RAM monitoring	Available
DMM (writing to memory during RUN)	Available
Pin masking	Available (reset pin only)
Time measurement	Resolution: 100 μs
(from execution start to break)	Max. measurement time: Approximately 100 hours
User spaces used for debugging	Internal ROM: 256 to 400 bytes
	Internal RAM: 7 to 9 bytes
	Differs depending on the device.
	For details, refer to the document ″QB-MINI2 Operating Precautions″.
Function pins used for debugging	The pin shared with OCDxA and OCDxB, the pin shared with TOOLCx and TOOLDx, or the pin shared with $\overline{\text{RESET}}$.

<R> User spaces used for debugging

<R> Function pins used for debugging

4.2.2 System configuration

Figure 4-24 illustrates the system configuration for on-chip debugging.

Figure 4-24. System Configuration for On-Chip Debugging
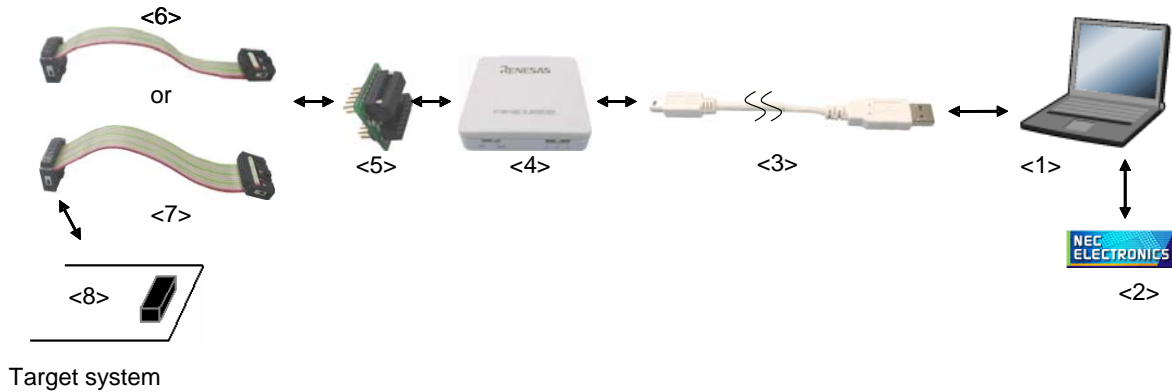
Target system

<1> Host machine

   Products with USB ports

<2> Software

   Includes the debugger, USB driver, device files, and so on.

   Download these Renesas Electronics software from the following website (ODS).

      URL: http://www2.renesas.com/micro/en/ods/index.html

<3> USB cable (accessory)

<4> MINICUBE2 (this product)

<5> 78K0-OCD board (accessory)**Note**

<6> 10-pin target cable (accessory)

<7> 16-pin target cable (accessory)

<8> Target connector (sold separately)

<R> **Note** On-chip debugging using TOOLCx and TOOLDx can be performed without using the 78K0-OCD board.

4.2.3 System startup procedure

This section describes the system startup procedure. Observe the following order.

(1) Preparation and installation of software

The following software is required to perform on-chip debugging. For details on preparation and installation of software, refer to the setup manual supplied with MINICUBE2.

- Debugger
- USB driver
- Device file (may be unnecessary when using a partner manufacture′s debugger)

(2) Switch setting

Set the mode select switch to ″M2″. Set the power select switch by referring to Table 4-7.

Caution   Do not change the switch setting while the USB cable is connected.

Table 4-7. Setting of Power Select Switch

Setting	Description
3	3 V is supplied from MINICUBE2 to the target system[Note].
5	5 V is supplied from MINICUBE2 to the target system[Note].
T	Power supply of the target system is used. MINICUBE2 detects the power for the target system. The detected power is also used as the power supply for the communication interface.

Note The maximum rating of the current is 100 mA, so do not use MINICUBE2 with the target system with the higher current rating. The power is always supplied after MINICUBE2 is connected to the host machine.

(3) 78K0-OCD board setting

The settings of the 78K0-OCD board vary depending on how clock is supplied to the target device. A 20 MHz oscillator is mounted on the 78K0-OCD board at shipment. If either of the following conditions applies, the factory setting must be changed. Refer to **4.2.5 Clock setting** and set the clock.

- The target device does not operate with a 20 MHz clock, or the target voltage that does not operate with 20 MHz clock is used
- A clock frequency other than 20 MHz is needed as the high-speed system clock.

(4) Connecting the target system

Connect MINICUBE2 to the target system as shown in Figure 4-25. Select a 10- or 16-pin target cable according to the target system circuit (refer to **Remark** below).

Perform connection **before the power to the target system is turned on**

Figure 4-25. Connecting MINICUBE2 to Target System
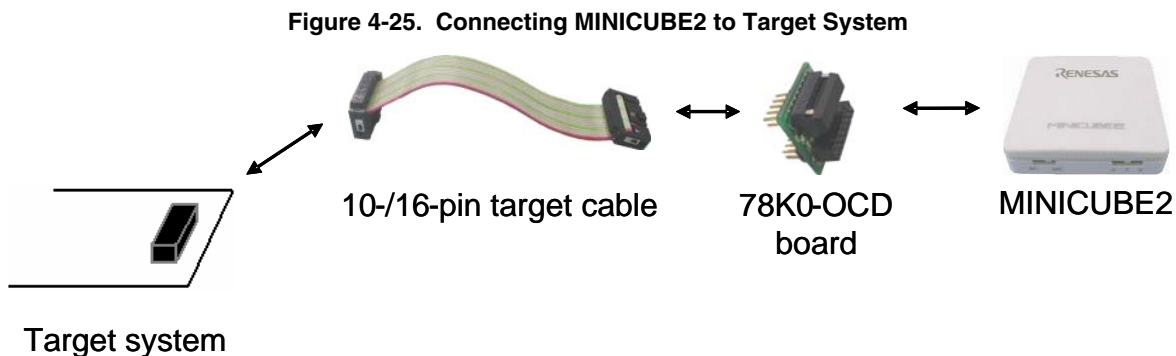
<R> **Caution   On-chip debugging using TOOLCx and TOOLDx can be performed without using the 78K0-OCD board. When TOOLC0 and TOOLD0 are used, a 4, 8, or 16 MHz clock signal generated by MINICUBE2 can be used for the clock signal of the microcontroller.**

Remark   Connect the 78K0-OCD board and the 10-/16-pin target cable as shown below. Do not connect both the 10-pin target cable and 16-pin target cable.
With 10-pin cable: Connect CN2 on the 78K0-OCD board to the 10-pin target cable (Figure 4-26).
With 16-pin cable: Connect CN3 on the 78K0-OCD board to the 16-pin target cable (Figure 4-27).

Figure 4-26. With 10-Pin Connector   **Figure 4-27. With 16-Pin Connector**

Caution   Connect the 78K0-OCD board to the target cable so that pin 1 position (▲) on each side match.
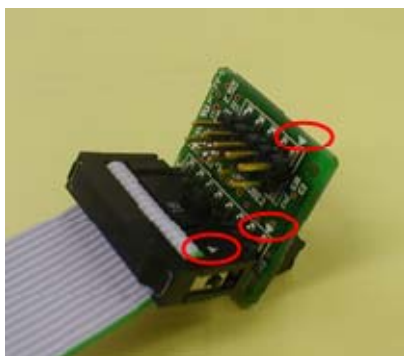
(5) Connecting the USB cable

Connect MINICUBE2 to the host machine as shown in Figure 4-28, **before the power to the target system is turned on**.

When the power select switch is set to ″T″, the mode LED **flashes** white after connection.

When the power select switch is set to ″3″ or ″5″, the mode LED **glows** white after connection.

Figure 4-28. Connecting MINICUBE2 to Host Machine

MINICUBE2   USB cable   Host machine
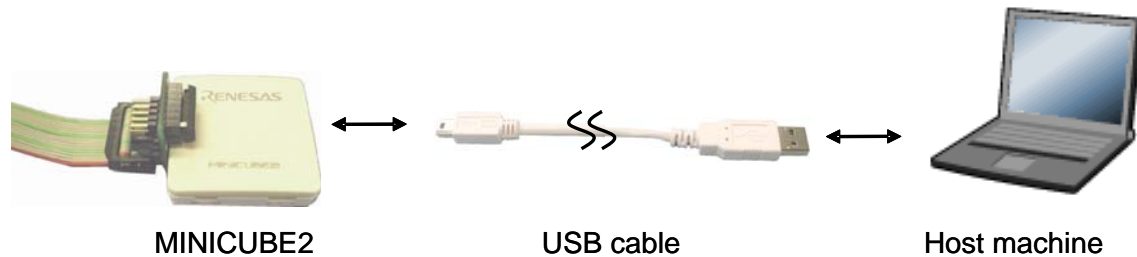
(6) Applying power to target system

Turn on power to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″. The mode LED glows white after connection after power application.

(7) Debugger startup

Start the debugger. The mode LED glows orange after debugger startup.

For the operation after this step, refer to the user′s manual for the debugger.

If the debugger does not start normally or the operation is unstable, the possible causes may be the following.

• Communication error between MINICUBE2 and target system

  Whether communication is performed normally can be checked using the OCD Checker. For details, refer to the user′s manual for the OCD Checker.

• The user resource has not been secured or the security ID has not been set

  To perform debugging with MINICUBE2, the user resource must be secured and the security ID must be set. For details, refer to **4.2.6 Securing of user resources and setting of security ID**.

• Unsupported software (debugger, device file, or firmware) is used

  The software used may not support debugging of the target device. Refer to the document ″QB-MINI2 Operating Precautions″ and check the supported version. When using software provided by a Renesas Electronics partner, refer to the documents prepared by the partner company.

• Defect of MINICUBE2

  MINICUBE2 may have a defect. Refer to **CHAPTER 7 SELF-TESTING**.

4.2.4 System shutdown procedure

Terminate debugging and shutdown the system in the following order.

If the following order is not observed, the target system or MINICUBE2 may be damaged.

(1) Debugger termination

Terminate the debugger.

(2) Target system power shutdown

Shutdown the power supplied to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″.

(3) Unplugging the USB cable

Unplug the USB cable from MINICUBE2 or the host machine.

(4) Unplugging the target cable

Unplug the target cable from MINICUBE2 or the target system.

<R>   **4.2.5 Clock setting**

The clock signal generated by the X1 oscillator, internal high-speed oscillator, or subsystem clock oscillator can be used for the clock signal of the target device during on-chip debugging. Setting up each is described below.

> **Remark**   MINICUBE2 can supply the clock signal for the target device instead of the X1 oscillator. Supplying a clock signal to the subsystem clock is not supported.

(1)   Using the high-speed system clock
(a) Using the clock provided in the target system
      (only when OCD1A and OCD1B or TOOLD1 and TOOLC1 are used)
(b) Using the clock generated in MINICUBE2
      (only when OCD0A and OCD0B or TOOLD0 and TOOLC0 are used)
(c) Using the oscillator connected to the 78K0-OCD board
      (only when OCD0A and OCD0B or TOOLD0 and TOOLC0 are used)
(d) Using the oscillator circuit connected to the 78K0-OCD board
      (only when OCD0A and OCD0B or TOOLD0 and TOOLC0 are used)
(2)   Using the internal high-speed clock (when no restrictions are applied by the occupied pins)
(3)   Using the subclock (when no restrictions are applied by the occupied pins)

The settings for (1) to (3) are described below.

(1) Using the high-speed system clock

(a) Using the clock provided in the target system (only when OCD1A and OCD1B or TOOLD1 and TOOLC1 are used)

Remove the 78K0-OCD board oscillator or the oscillation circuit.

("System" is automatically selected for "Main Clock" in the Configuration dialog box of the debugger.)
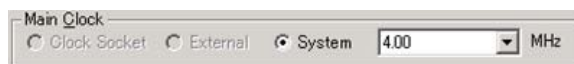
In this case, the selected system is used for communication between MINICUBE2 and the target device. Therefore, the download time, which depends on the communication speed, can be shortened if the 16 MHz clock for the system is specified instead of the 4 MHz clock for the system.

(b) Using the clock generated in MINICUBE2 (only when OCD0A and OCD0B or TOOLD0 and TOOLC0 are used)

Remove the 78K0-OCD board oscillator or the oscillation circuit.

("System" is automatically selected for "Main Clock" in the Configuration dialog box of the debugger.)

For details about the setting, see the debugger user's manual.
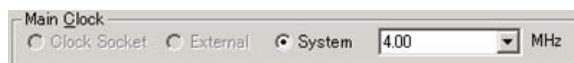
(c) Using the oscillator connected to the 78K0-OCD board (only when OCD0A and OCD0B or TOOLD0 and TOOLC0 are used)

Connect an oscillator or oscillation circuit to CLK1 on the 78K0-OCD board.

("Clock Board" is automatically selected for "Main Clock" in the Configuration dialog box of the debugger.)

The same frequency as that of the target device can be used.
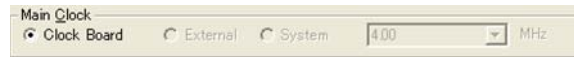
Mount the 14-pin oscillator (5 V, CMOS type) in CLK1 of the 78K0-OCD board, as shown in Figure 4-29.

TXC-1, manufactured by Tokyo Quartz Co., Ltd. is recommended as an oscillator.

Figure 4-29. Mounting Oscillator
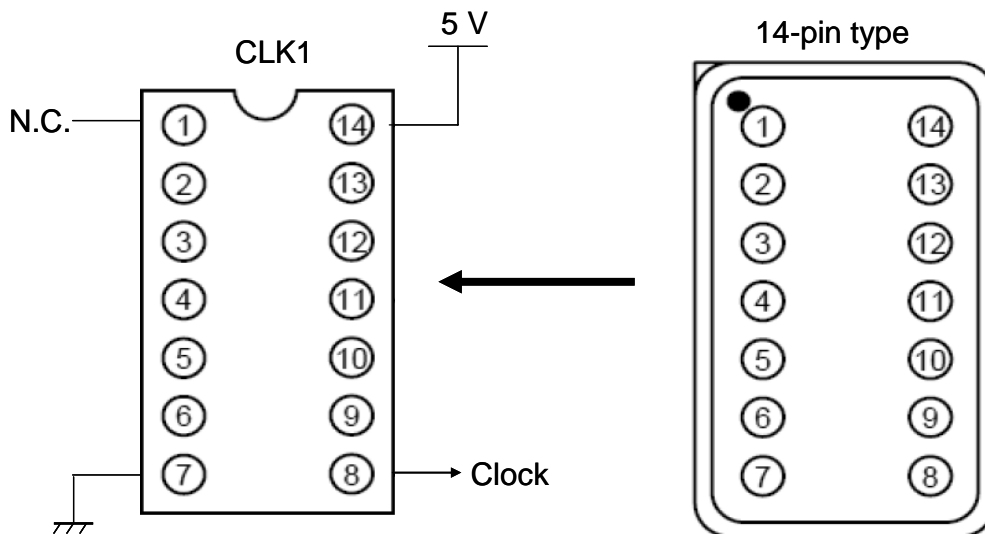
(d) Using the oscillator circuit connected to the 78K0-OCD board (only when OCD0A and OCD0B or TOOLD0 and TOOLC0 are used)

Connect an oscillator or oscillation circuit to CLK1 on the 78K0-OCD board.

("Clock Board" is automatically selected for "Main Clock" in the Configuration dialog box of the debugger.)

The same frequency as that of the target device can be used.
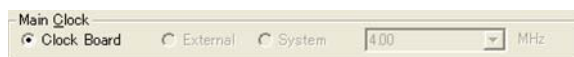
Refer to Figure 4-30 and mount an oscillator in CLK1 of the 78K0-OCD board. Figure 4-27 shows the appearance after mounting the oscillator.

″160-90-314″ (made by PRECI-DIP) can be used as the parts board to be mounted in the socket.

Figure 4-30. 78K0-OCD Board Mounting Diagram and CLK1 Equivalent Circuit

Parts Mounting Diagram   **CLK1 Equivalent**
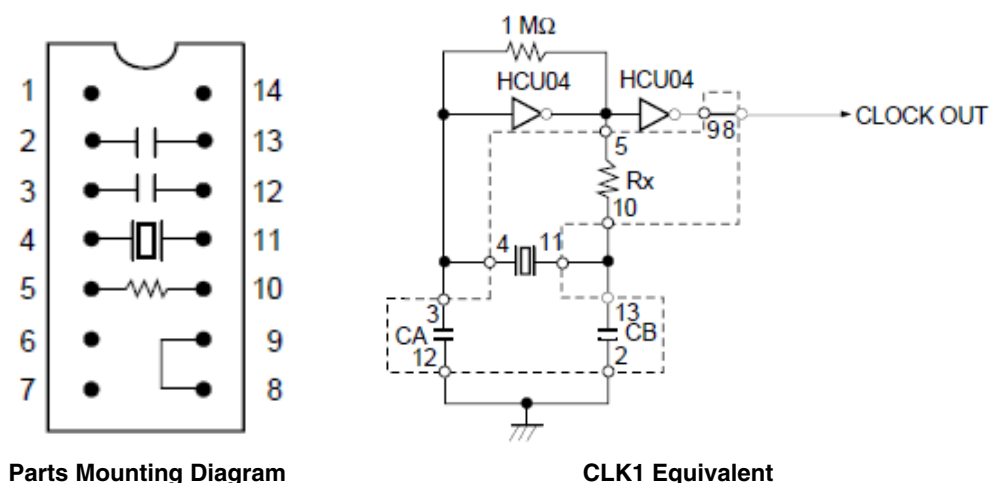
Pin Number	Connection
2-13	Capacitor CB
3-12	Capacitor CA
4-11	Ceramic resonator or crystal resonator
5-10	Resistance Rx
8-9	Shorted

Figure 4-31. Appearance After Mounting Oscillator

(2) Using the internal high-speed clock (when no restrictions are applied by the occupied pins)

MINICUBE2 operates on the internal high-speed oscillator of the target device, regardless of the setting specified for "Main Clock" in the Configuration dialog box of the debugger.

However, the clock specified by the debugger is used for downloading a program when the debugger starts.

Therefore, the download time can be shortened if, for example, the 16 MHz clock for the system is specified instead of the 4 MHz clock for the system.

(3) Using the subclock (when no restrictions are applied by the occupied pins)

MINICUBE2 operates on the subclock of the target system, regardless of the setting specified for "Sub Clock" in the Configuration dialog box of the debugger. (This setting cannot be specified.)
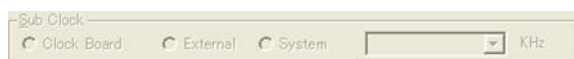
4.2.6 Securing of user resources and setting of security ID

MINICUBE2 uses the user memory spaces (shaded portions in Figure 4-32) to implement communication with the target device, or each debug functions. The areas marked with a dot (•) are always used for debugging, and other areas are used for each debug function used. Refer to the descriptions of (a) to (f) on the following pages and secure these spaces in the user program or using the compiler options.

When C-SPY manufactured by IAR Systems is used, read also the following material.
- IAR C-SPY Hardware Debugger Systems User Guide issued by IAR Systems

<R>   **Figure 4-32. Reserved Area Used by MINICUBE2**
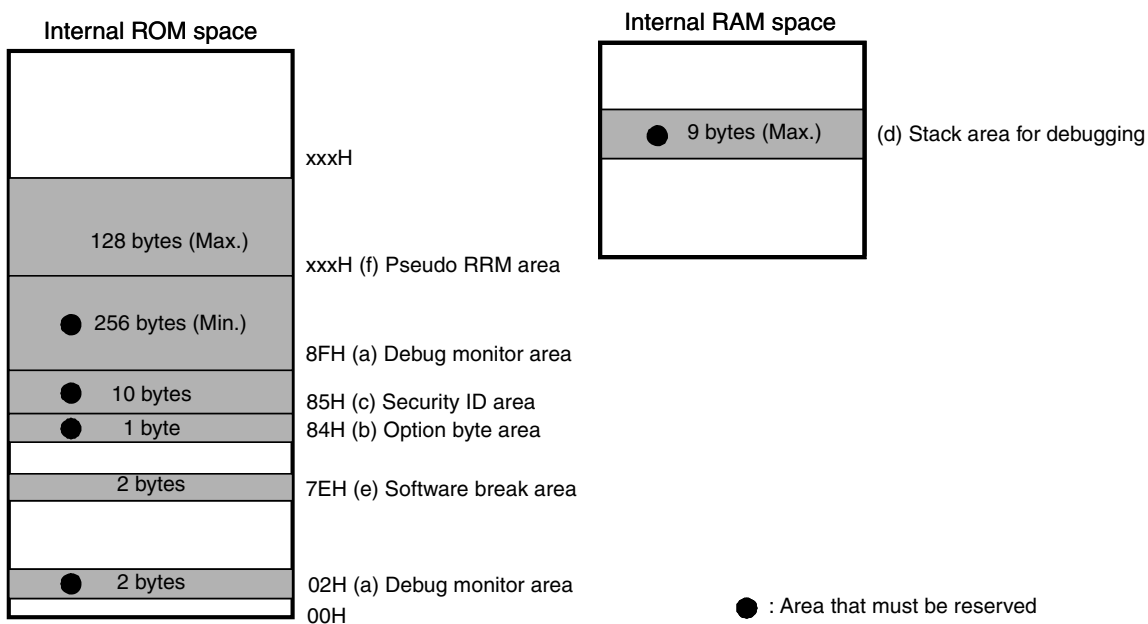
<R>   **Note**   When ID78K0-QB Ver.3.00 or later is used, be sure to check following the cautions.
For the products having the internal extended RAM:
By allocating the area, F7F0H to F7FFH, in the internal extended RAM, the pseudo PRM function can be used.
For the products not having the internal extended RAM:
The pseudo PRM function cannot be used.

(a) Debug monitor area (this area must be secured)

Addresses 0x02, 0x03 and area starting from address 0x8F must be secured to embed the debug monitor program. If this area is rewritten by flash self programming, on-chip debugging can no longer be performed.

[How to secure areas]

<R> When using Renesas Electronics compiler CC78K0, addresses 0x02, 0x03 and area starting from address can be secured for debug monitoring, using the linker option (-go). Figure 4-33 shows the Linker Options dialog box of the project manager. Select the ″On-Chip Debug″ check box as shown in the figure below. If the pseudo RRM function is not used, set the size of the debug monitor area (in actuality, 257 bytes from 0x8F to 0x18F are secured). For the required size of the debug monitor area, see the document "QB-MIN2 Operating Precautions.". The area to allocate differs depending on the target device.) If the pseudo RRM function is used, refer to **(f)**.

Figure 4-33. Linker Option Setting (Debug Monitor Area)
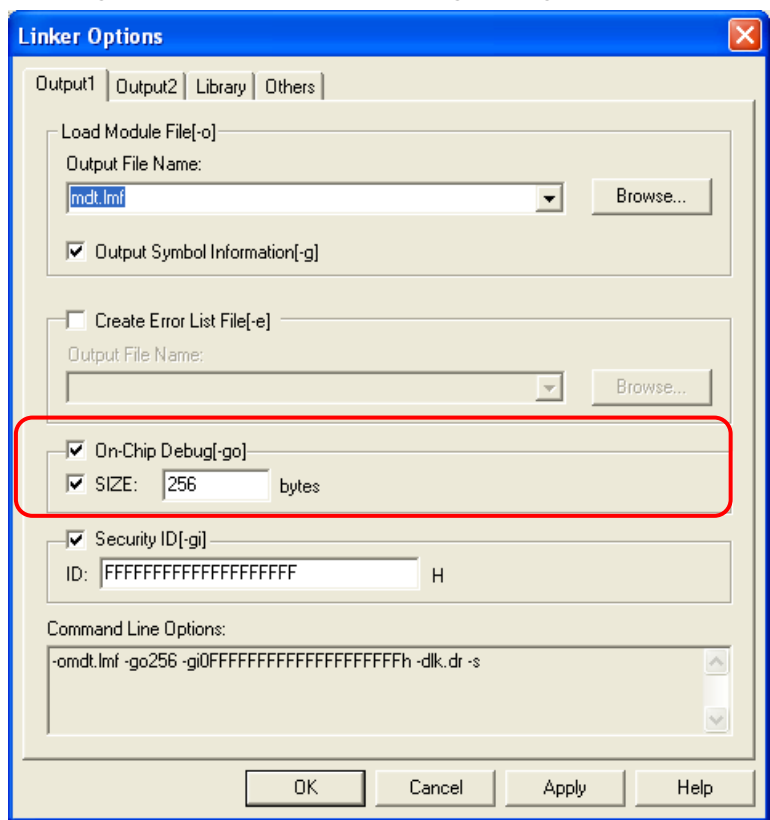
(b) Option byte area (essential)

This is the area for the security setting to prevent the flash memory from being read by an unauthorized person. The target device operates in accordance with the set value, as shown below.

Table 4-8. Option Byte (84H) Settings and Descriptions

FLMD0	Value[Note 1]	Description
Products that have an option byte	00H	The debugger cannot be started even if MINICUBE2 is connected.
	02H	The internal flash memory is not erased regardless of how many times authenticating the security ID code fails.
	03H	The entire internal flash memory area is erased if authenticating the security ID code fails.
Products that do not have an option byte	00H	The entire internal flash memory area is erased if MINICUBE2 is connected.
	02H	The internal flash memory is not erased regardless of how many times authenticating the security ID code fails.[Note 2]
	03H	The entire internal flash memory area is erased if authenticating the security ID code fails.[Note 2]
Other than the above		Setting prohibited

Notes 1. This value is to be specified in the assembler source code of a user-created program. When the program is downloaded by the debugger, the debugger changes the value as required.

2. If the value at address 83H of the option bytes is illegal, the entire internal flash memory area is erased and then the debugger starts. If authenticating the security ID code fails after the debugger starts, the entire internal flash memory area is specified to be erased.

Table 4-9. Option Byte (84H) Settings and Descriptions

FLMD0	Value[Note 1]	Description
Products that have an option byte	00H	Be sure to set 0083H to 00H, because it is a reserved area.
Products that do not have an option byte	Device dependent	Specify a value according to the on-chip debugging specifications in the device user's manual.[Note 2]

Notes 1. This value is to be specified in the assembler source code of a user-created program. When the program is downloaded by the debugger, the debugger changes the value as required.

2. For PM+, if this value is not specified in the assembler source code, 1EH is automatically specified. If the value is illegal, the entire internal flash memory area is erased and then the debugger starts.   In STOP mode, the internal oscillator does not stop even if specified to do so.

[How to set]

Describe the code for setting the option byte in the user program. Refer to the following and add a code to the assembler source.

Example When setting 0x02

```
      SSS     CSEG    AT      084H;    ″SSS″ is an arbitrary symbol name (eight characters or less)
              DB      2H
```

Caution   If address 0x84 is overwritten by 0x00 by self programming, communication is disabled, and debugging and connection can no longer be performed even if the debugger is restarted. In such as case, erase the memory via flash programming.

(c) Security ID area (essential)

This is the area for the security setting to prevent the flash memory from being read by an unauthorized person. The security ID functions as a password for starting the debugger. The debugger starts only when the security ID that is input during debugger startup and the security ID embedded in this area match.

[How to set]

When using Renesas Electronics assembler RA78K0, an ID code is generated in this area, using the linker option (-gi). Figure 4-34 shows the Linker Options dialog box of the project manager. Select the ″Security ID″ check box as shown in the figure below and set an arbitrary ID code (10 bytes).

Figure 4-34. Linker Option Setting (Security ID)
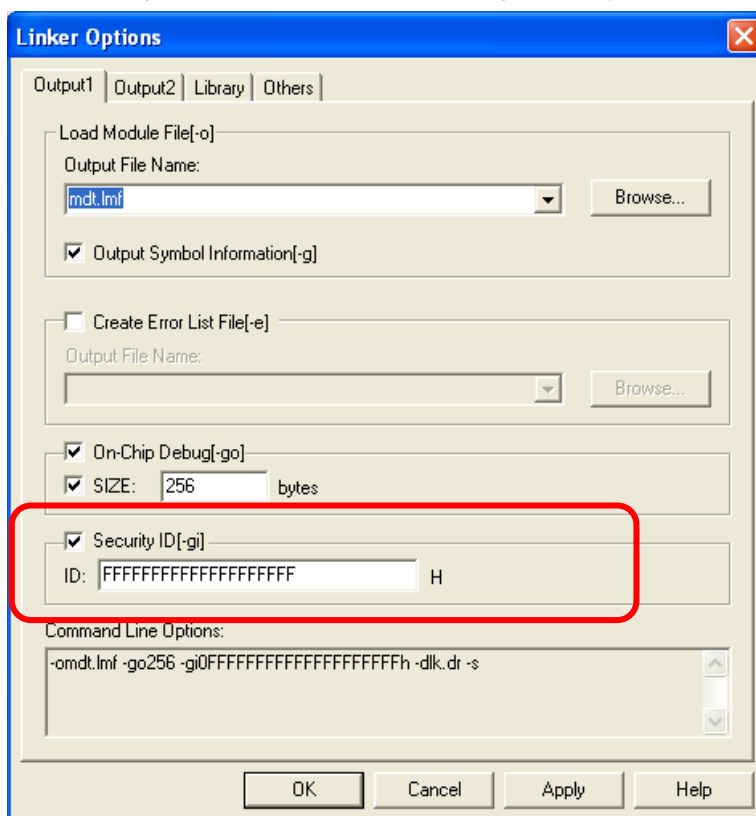
(d) Stack area for debugging (this area must be secured)

This area requires 7 to 9 bytes as the stack area for debugging. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease.

Figure 4-35 illustrates the case where the stack area is increased when the internal high-speed RAM starts from 0xFB00.

Figure 4-35. Variation of Address of Stack Area for Debugging
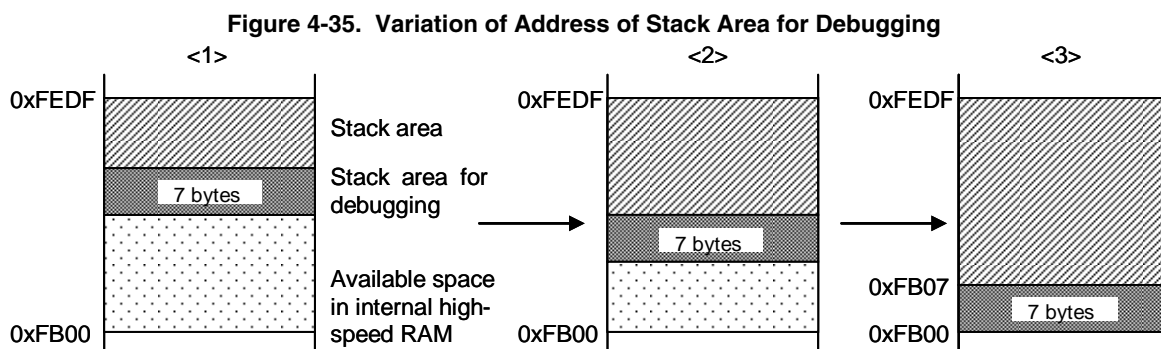
The size of this area also varies depending on whether software breaks or pseudo real-time RAM monitor is used.

Table 4-10. Size of Stack Area for Debugging

Item	Size of Stack Area for Debugging
When hardware breaks are used	7 bytes
When software breaks are used	9 bytes

[How to secure areas]

Refer to the address range shown below and set the stack pointer.

Example When internal high-speed RAM starts from 0xFB00

- When hardware breaks are used

    Within the range 0xFB07 to 0xFEDF[Note]

- When software breaks are used (also refer to **(e)**)

    Within the range 0xFB09 to 0xFEDF[Note]

Note Because the area 0xFEDF is used by MINICUBE2 during a break, set the initial value of the stack pointer to 0xFEDF, not 0xFEE0 (when the initial value is set to 0xFEDF, the actual value used for the stack is 0xFEDE, so there will be no problems).

Furthermore, 0xFEDF is prohibited for use as stack area, but it can be used as variable area.

(e) Software break area
This area is used for software breaks.

[How to secure areas]
Refer to the following and secure the area.
  SSS   CSEG   AT   07EH;   ″SSS″ is an arbitrary symbol name (eight characters or less)
           DB   0FFH, 0FFH

<R>   **(f) Pseudo real-time RAM monitor (RRM) area**
To use pseudo real-time RAM monitoring, an area of "Debug monitor area (see (a)) + up to 128 bytes" is required.

[How to secure ″debug monitor area + 256 bytes″]
When using Renesas Electronics compiler CC78K0, the area used for pseudo real-time RAM monitoring can be secured using the linker option (-go). Figure 4-36 shows the Linker Options dialog box of the project manager. Add the size of the area for RAM monitor to the area specified in (a). For details about the RAM monitoring area, refer to the document ″QB-MINI2 Operating Precautions″.

Figure 4-36. Linker Option Setting (When Pseudo RAM Monitor and Software Break Are Used)
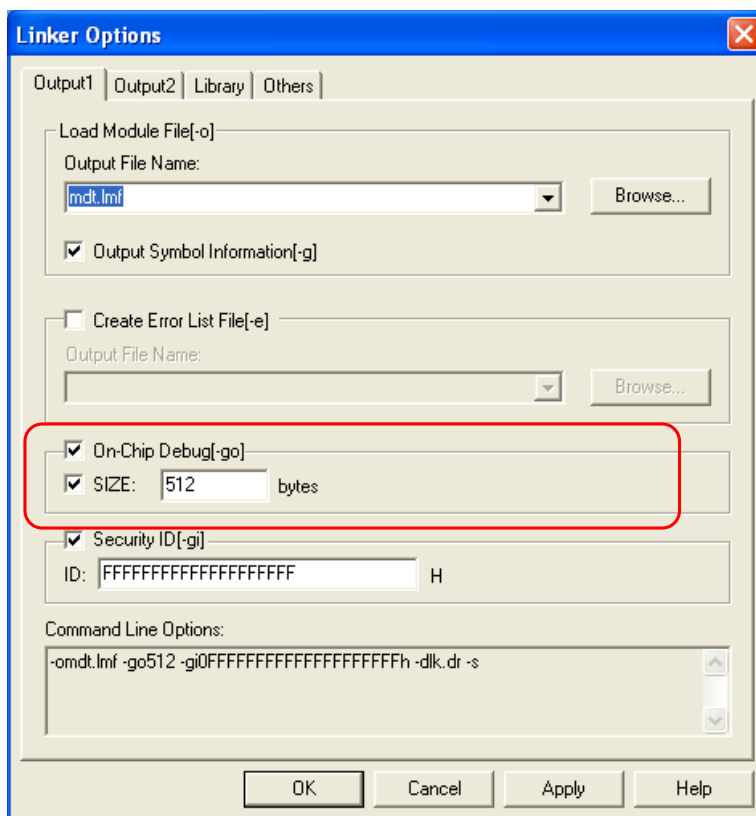
4.2.7 Cautions on debugging

This section describes cautions on performing on-chip debugging for a 78K0 microcontroller.

Be sure to read the following to use MINICUBE2 properly.

(1) Handling of device that was used for debugging

Do not mount a device that was used for debugging on a mass-produced product, because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed.

(2) Overwriting flash memory during on-chip debugging

If the following debugger operations <1> to <7> are performed on-chip, the flash memory in the device is overwritten.

If any of the following debugger operations <1> to <7>, which involve flash memory rewriting, is performed while flash memory cannot be rewritten, the debugger automatically changes the register setting so as to enable flash memory rewriting, and restores the register setting after the operation is completed. If any of the following operations <1> to <7> is performed while flash memory rewriting is disabled or operation is performed at a voltage at which flash memory cannot be rewritten, however, the debugger outputs an error and the operation is ignored.

<1> Writing to internal flash memory

<2> Program execution after specifying or canceling software breakpoints

<3> Step-over execution, Return Out execution

<4> Come Here

<5> If Permit is selected in the Target Power off area in the Configuration dialog box, the following operations cannot be performed:

   a) Specifying, changing, or canceling hardware breakpoints

   b) Masking/unmasking internal resets

   c) Switching peripheral breakpoints

   d) Program execution

   e) Software reset (a reset performed by the debugger)

<6> Adding, changing, or deleting the monitor address when using the pseudo real-time RAM monitor function

<7> Performing operations without using breakpoints when software breakpoints are specified

It takes a while from the completion of flash memory programming until control is passed to the GUI.

(3) Software break (1)

During program running, do not rewrite the data at the address where a software break is set. This includes self programming and rewriting to RAM. If performed, the instruction placed at the address may be invalid.

(4) Boot swapping during self programming

The boot swapping function cannot be emulated. This is because boot swapping moves the memory spaces used for debugging, and thus the debug communication can no longer be performed.

(5) Emulation in self programming mode

For self programming, the mode is switched from normal mode to self programming mode.

```
MOV PFCMD,#0A5H
MOV FLPMC,#1H        ← (1)
MOV FLPMC,#0FEH
MOV FLPMC,#1H
                ---------------------- From this position
CALL !08100H         ← (2)                  ↑
MOV PFCMD,#0A5H
MOV FLPMC,#0H                       Mode A1
MOV FLPMC,#0FFH
MOV FLPMC,#0H        ← (3)                   ↓
                ---------------------- To this position
```

The section between (2) and (3) is in self programming mode A1. Step execution or debug operations, such as stopping program execution, cannot be performed in this section.

In addition, do not set a software break in the section between (1) and (3); otherwise the execution continues without a break but emulation is not performed normally.

(6) Break function for stack pointer initialization failure

This function executes a break when an interrupt occurs or a PUSH instruction is executed while the initial setting has not been made for the stack pointer.

If the manipulation or instruction shown below is executed immediately after a reset operation, the break function for stack pointer initialization failure becomes invalid.

- Setting a software break
- Write to the stack pointer from the Register window
- Write to the flash memory from the Memory window, etc

If a software break occurs while the initial setting has not been made for the stack pointer, the message "Uninitialized Stack Pointer" is displayed on the status bar.

The subsequent operations are not performed normally, so make sure to set the SP value in the user program.

(7) Caution on downloading a HEX file

When downloading a HEX file, do not set specify a filling value other than 0xFF for the object converter option (-U).

(8) Software break (2)

If the debugger does not terminate normally due to a factor such as a freeze while a software break is set, the instruction for which the software break is set and that has been substituted by a debug instruction remains as is. To avoid this, select ″Object″ in the Download dialog box and download the load module. In particular, take care regarding this point when starting the debugger using a project file, as well as when using PM+.

Select the Object check box and
then download the project file
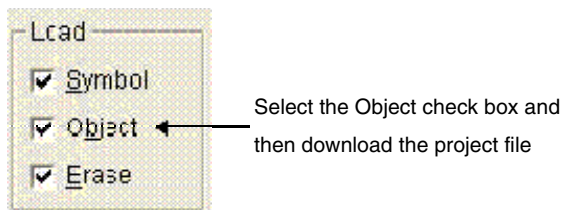
(9) Cautions when stepping into code

Some SFRs (special function registers) might remain stopped while stepping into code. If the operation of the SFRs does not advance while stepping into code, operate the microcontroller by continuously executing the instructions instead of executing them in steps. Stepping into code: Instructions in the user-created program are executed one by one. Continuous execution: The user-created program is executed from the current PC value.

(10) Cautions when using pseudo real-time RAM monitor function

<1> The user program is stopped while the pseudo real-time RAM monitor function processing is being performed.

Example

When 78K0Kx2

30 x (n + 1) [μs] per n bytes when operating at 20 MHz

When 78K0Kx2-L

120 x (n + 1) [μs] per n bytes when operating at 20 MHz

<2> If the pseudo real-time RAM monitor function is executed during a standby state, the standby state is released.

<3> If the clock signal switches to that generated by the internal low-speed oscillator[Note] or to the subsystem clock signal by the user-created program when pseudo real-time RAM monitoring is performed, pseudo real-time RAM monitoring stops. The user-created program is stopped for about 200 clock cycles by pseudo real-time RAM monitoring.

<4> Pseudo real-time RAM monitoring updates RAM information in byte units. Therefore, even if information is displayed in word units in the debugger, there will be a difference of a few ms in updating the higher and lower byte data.

Note   Devices with OCD control code V1.xx. Version can be checked by selecting [About…] from the [Help] menu in the debugger.

(11)Caution on mounting a clock on 78K0-OCD board

When a clock is mounted on the 78K0-OCD board, power supply may not become the GND level even if the power to the target system is shut down.

(12) Emulation of POC function

If power supply to the target system is turned off during debugging, the target device enters the reset state by the RESET_OUT pin of MINICUBE2. Consequently, the target device's POC function performs pseudo emulation. Note that there is an error of ±1 V between the POC detection voltage and the target system voltage. Note that the POC detection voltage might be 1 V higher than that of the target device.

(13) Device with/without on-chip debug function

Only the devices with the on-chip debug function can be debugged. See the user's manual of the target device for checking whether it is equipped with the on-chip debug function.

To debug a device without the on-chip debug function, perform debugging by using an upper-compatible product. The following shows an example when using the 78K0/KF2.

Target Device	On-Chip Debug Function	Device for Debugging	Device File (DF) Used for Debugging
μPD78F0544	Not equipped	μPD78F0547D (Set the IMS and IXS registers to values according to the target device.)	DF for μPD78F0544
μPD78F0545			DF for μPD78F0545
μPD78F0546			DF for μPD78F0546
μPD78F0547			DF for μPD78F0547

\<R\> **(15) Cautions when debugging with only the real device (without using MINICUBE2)**

When debugging with only the real device (i.e. without using MINICUBE2), it is recommended that the user program be written by QB-Programmer. The operation of the program that is downloaded by the debugger differs from the actual operation because the monitor program performs the processing after reset.

\<R\> **(16) Cautions when reading the location and the reserved area where SW/HW breaks are set**

When the addresses where the software breaks are set are read, the data read from the addresses differ from the actual data.

When the addresses where the hardware breaks are set are read, breaks occur.

When the addresses where are reserved for MINICUBE2, the operations differ depending on the following usages by the software breaks.

- When software breaks are not used:
    The data read from the addresses may differ from the actual data.

- When software breaks are used:
    The data read from the addresses may differ from the actual data, or breaks may occur.

4.3 Flash Programming

This section describes the system configuration and startup/shutdown procedure when flash programming is performed for a 78K0 microcontroller, using MINICUBE2.

4.3.1 Specifications of programming function

Table 4-11. Specifications of Programming Function

Functions	Specifications
Host interface	USB 2.0 (compatible with 1.1)
Target interface	UART
Target system voltage	1.65 to 5.5 V (depends on the target device)
Clock supply	16, 8, or 4 MHz clock can be supplied Clock mounted on the target system can be used
Power supply	3 or 5 V (maximum current rating: 100 mA)
Acquisition of device-specific information	Parameter file for Renesas Electronics MINICUBE2 is used
Security flag setting	Available
Standalone operation	Unavailable (must be connected to host machine)

<R>

4.3.2 System configuration

Figure 4-37 illustrates the system configuration for flash programming.

Figure 4-37. System Configuration for Flash Programming
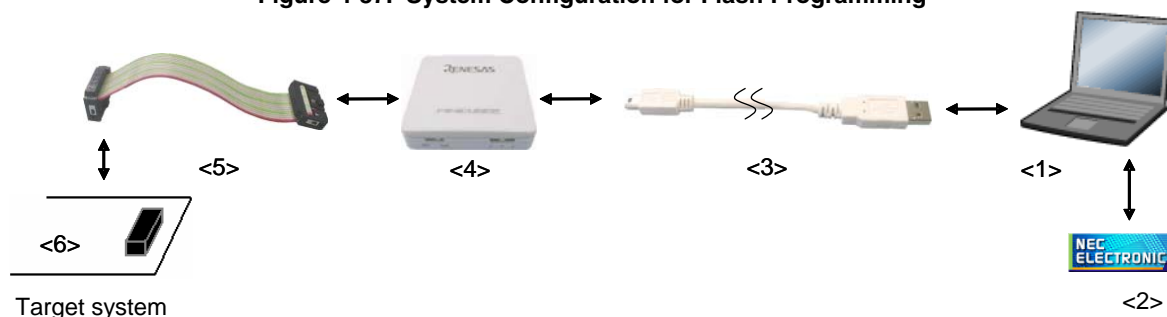
<1> Host machine
   Products with USB ports
<2> Software
   Includes the QB-Programmer USB driver, parameter files, and so on.
   Download these Renesas Electronics software from the following website (ODS).
       URL: http://www2.renesas.com/micro/en/ods/index.html
<3> USB cable (accessory)
<4> MINICUBE2 (this product)
<5> 16-pin target cable (accessory)
<6> Target connector (sold separately)

Remark Flash programming using the TOOLDx and TOOLCx pins can be performed even if the 78K0-OCD board is connected.

4.3.3 System startup procedure

This section describes the system startup procedure. Observe the following order.

(1) Preparation and installation of software

The following software is required to perform flash programming. For details on preparation and installation of software, refer to the setup manual supplied with MINICUBE2.

- QB-Programmer
- USB driver
- Parameter file

(2) Switch setting

Set the mode select switch to ″M2″. Set the power select switch by referring to the following table.

Caution   Do not change the switch setting while the USB cable is connected.

Table 4-12. Setting of Power Select Switch

Setting	Description
3	3 V is supplied from MINICUBE2 to the target system[Note].
5	5 V is supplied from MINICUBE2 to the target system[Note].
T	Power supply of the target system is used. MINICUBE2 detects the power for the target system. The detected power is also used as the power supply for the communication interface.

Note The maximum rating of the current is 100 mA, so do not use MINICUBE2 with the target system with the higher current rating. The power is always supplied after MINICUBE2 is connected to the host machine.

(3) Connecting the target system

Connect MINICUBE2 to the target system as shown in Figure 4-38, **before the power to the target system is turned on**. When the power select switch is set to ″T″, the step of **(4) Connecting the USB cable** can be performed before this step.

Figure 4-38. Connecting MINICUBE2 to Target System

16-pin target cable   MINICUBE2
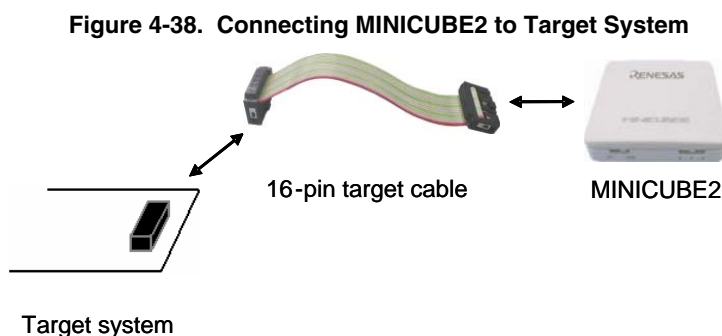
Target system

(4) Connecting the USB cable

Connect MINICUBE2 to the host machine as shown in Figure 4-39, **before the power to the target system is turned on**.

When the power select switch is set to ″T″, the mode LED **flashes** white after connection.

When the power select switch is set to ″3″ or ″5″, the mode LED **glows** white after connection.

The power is always supplied after MINICUBE2 is connected to the host machine with the USB cable.

Caution   Do not change the switch setting while the USB cable is connected.

Figure 4-39. Connecting MINICUBE2 to Host Machine
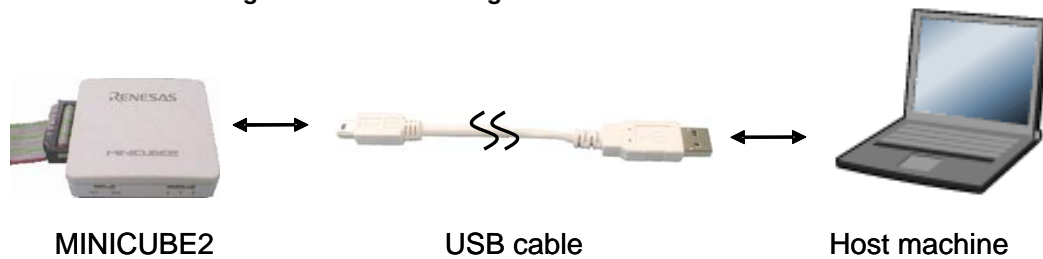
MINICUBE2   USB cable   Host machine

(5) Applying power to target system

Turn on power to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″. The mode LED glows white after power application.

(6) QB-Programmer startup

Start the QB-Programmer from the Start menu of Windows. The mode LED glows green after QB-Programmer startup.

A series of basic operations after this step is described in the following section. For detailed usage of the QB-Programmer, refer to the **QB-Programmer Programming GUI Operation User's Manual**.

4.3.4 Usage examples

This section describes a series of basic operations using the QB-Programmer, using the μPD78F0547D as an example. The operations from execution of the [Autoprocedure (EPV)] command to programming for the target device are described. For other commands and their applications, refer to the **QB-Programmer Programming GUI Operation User's Manual**.

The conditions used for a series of operations in this section are as follows.

<Target system>

    Target device: μPD78F0547D

    Power supply voltage: 5 V (supplied from MINICUBE2)

    Communication channel: UART, 115,200 bps

<MINICUBE2>

    Switches: Mode select switch: M2

               Power select switch: 5

<QB-Programmer>

    Parameter file: 78F0547D.PRM

    Clock setting: 20 MHz, x1 (oscillator on the target system side)

    Operation mode: chip

    Program file: sample.hex

    Command option: [Blank check before Erase] selected

                      [Read Verify after Program] selected

(1) Setting of programming environment

Set the programming environment, following the steps <1> to <6> described below.

<1> Click the [Device] menu on the menu bar and then click [Setup...].

<2> The Device Setup dialog box appears, in which the [Standard] tab is active.

Figure 4-40. [Standard] Tab of Device Setup Dialog Box
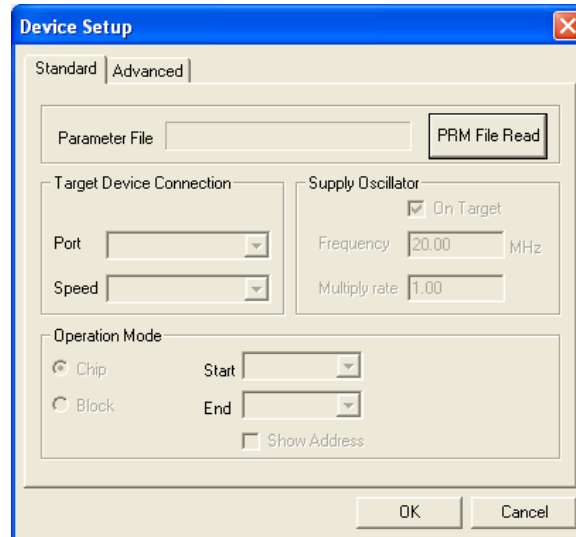
<3> Click the PRM File Read button to open the Parameter File Select dialog box. Select the parameter file for the target device and then click the Open button.

Figure 4-41. Parameter File Select Dialog Box
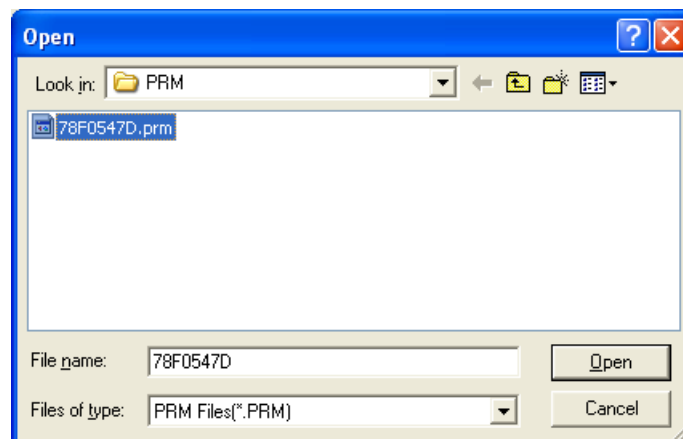
<4> Set the items in the [Target Device Connection], [Operation Mode] and [Supply Oscillator] areas, in accordance with your programming environment. The following figure shows an example.

Figure 4-42. [Standard] Tab of Device Setup Dialog Box After Setting
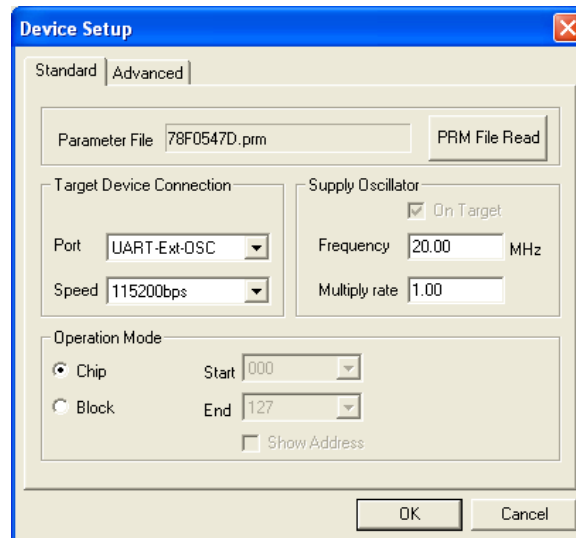
<5> Switch to the [Advanced] tab. Set the items in the [Command options] and [Security flag settings] areas, in accordance with your programming environment. The following figure shows an example.

Figure 4-43. [Advanced] Tab of Device Setup Dialog Box
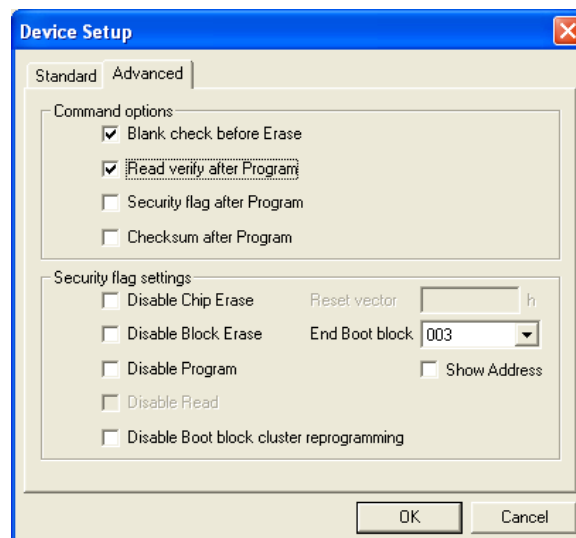
<6> By clicking the OK button, the programming environment is set and the Device Setup dialog box is closed.
      The main window appears as follows.

Figure 4-44. Completion of Programming Environment Setting
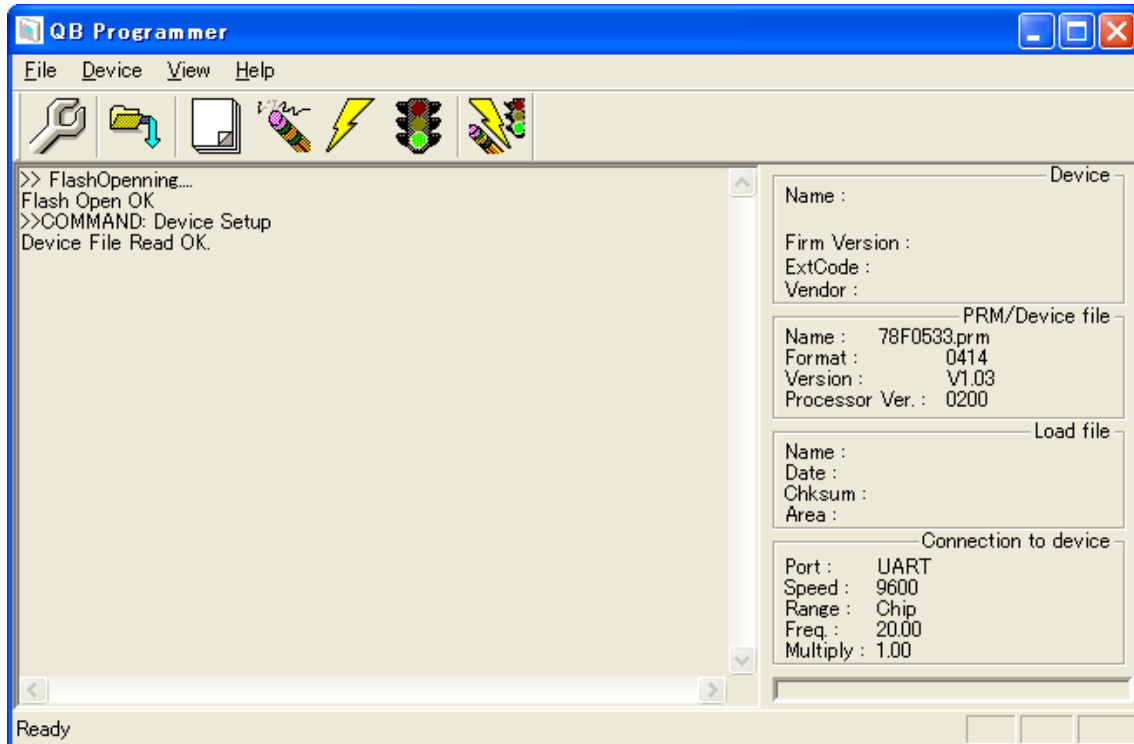
(2) Selection of program file
Select the program file, following the steps <1> to <3> described below.
<1> Click the [File] menu on the menu bar and then click [Load...].
<2> The program file select dialog box appears.

Figure 4-45. Program File Select Dialog Box
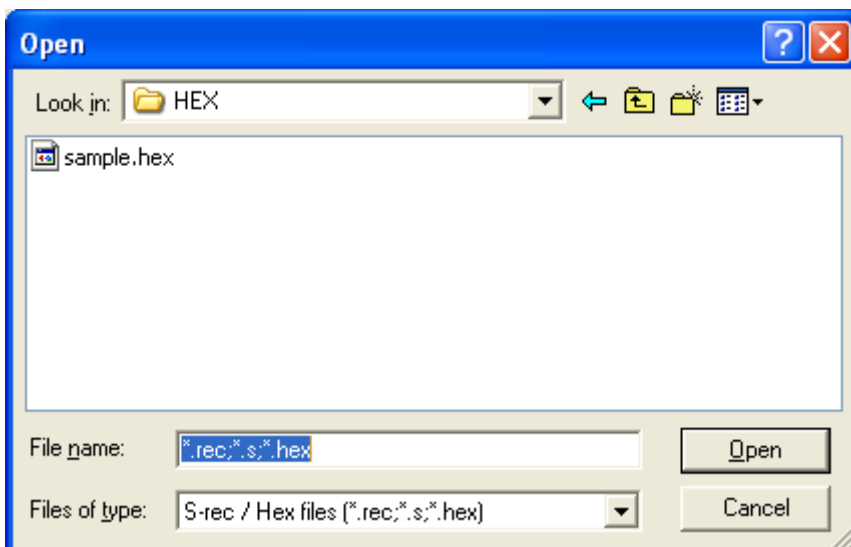
<3> Select the program file for the target device and then click the Open button.

Figure 4-46. Completion of Program File Selection
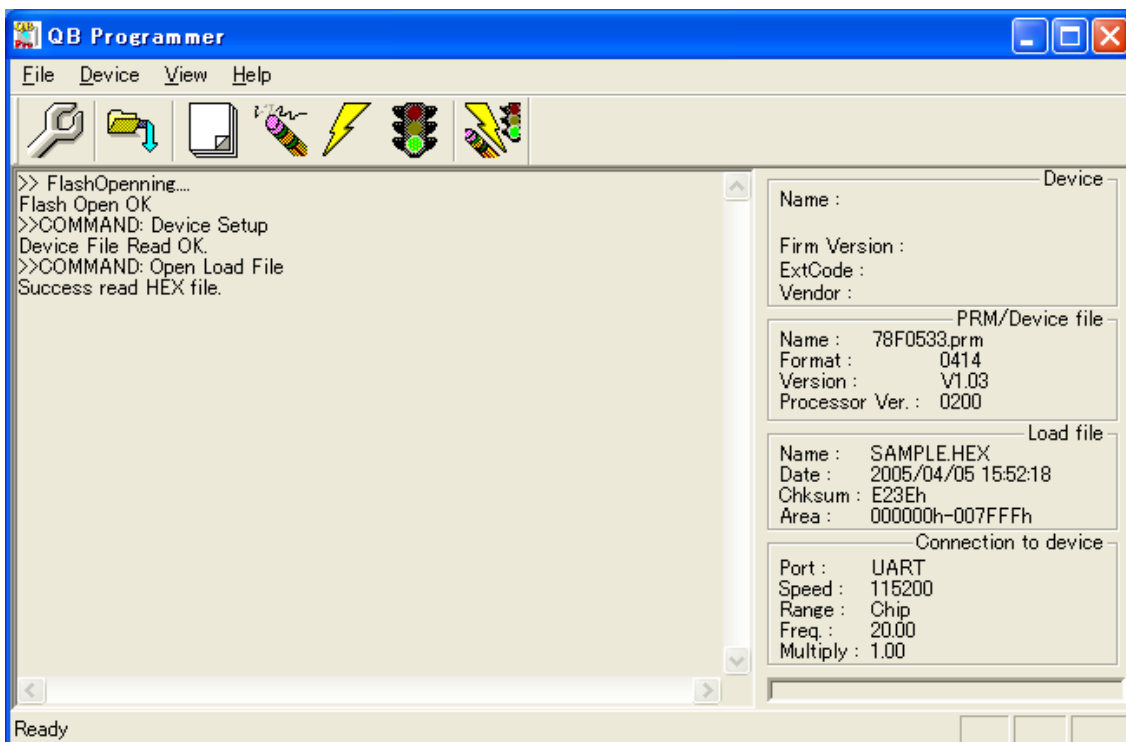
(3) Execution of [Autoprocedure (EPV)] command

Click the [Device] menu on the menu bar and then click [Autoprocedure (EPV)].

The [Blank Check], [Erase] (if the flash memory in the target device is not blank), [Program], [Verify] are executed for the target device sequentially. The mode LED on MINICUBE2 flashes yellow during execution. When the command execution is completed normally, the mode LED glows green and the following window appears.

Figure 4-47. Completion of [Autoprocedure (EPV)] Command Execution

4.3.5 System shutdown procedure

Terminate flash programming and shutdown the system in the following order.

If the following order is not observed, the target system or MINICUBE2 may be damaged.

(1) Terminating the QB-Programmer

Terminate the QB-Programmer if you are not going to perform programming to other devices.

Remark When the QB-Programmer is terminated, the setting information is saved in the *qbp.ini* file. Due to this file, the same setting is maintained when the QB-Programmer is started the next time.

(2) Target system power shutdown

Shutdown the power supplied to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″.

(3) Unplugging the USB cable

Unplug the USB cable from MINICUBE2 or the host machine.

When the power select switch is set to ″T″, the step of **(4) Unplugging the target cable** can be performed before this step.

(4) Unplugging the target cable

Unplug the target cable from MINICUBE2 or the target system.

4.3.6 Cautions on flash programming

This section describes the cautions for flash programming. Be sure to read the following for the proper use of MINICUBE2.

• To improve the writing quality, fully understand, verify, and evaluate the following items before using MINICUBE2.
   - Circuits are designed as described in the user's manuals for the device and MINICUBE2.
   - The device, QB-Programmer and MINICUBE2 are used as described in each user's manual.
   - The power supplied to the target system is stable.

CHAPTER 5   HOW TO USE MINICUBE2 WITH 78K0S MICROCONTROLLER

a 78K0S microcontroller.

To perform on-chip debugging for a 78K0S microcontroller, a specific program (debug function) must be downloaded to the device, and then debug the device mounted on the target system. Since debugging is performed with the target device operating on the board, this method is suitable for field debugging.

Flash programming is a method to write a program to the flash memory embedded in a device. Erasing and writing the program can be performed on-board with the device.

Read the following chapters if you are using MINICUBE2 for the first time with a 78K0S microcontroller as the target device.

- 5.1 Target System Design
  For communication between MINICUBE2 and the target system, communication circuits must be mounted on the target system. This section describes the circuit design and mounting of connectors.

- 5.2 On-Chip Debugging
  This section describes the system configuration and startup method to perform on-chip debugging with MINICUBE2.

- 5.3 Flash Programming
  This section describes the system configuration and startup method to perform flash programming with MINICUBE2.

5.1 Target System Design

   This section describes the target system circuit design required for on-chip debugging and flash programming.
   Figure 5-1 presents an overview of the MINICUBE2 communication interface. As shown on the left side of the figure, MINICUBE2 performs serial communication with the target device on the target system. For this communication, communication circuits must be mounted on the target system. Refer to this section to design circuits appropriately.

Figure 5-1. Outline of Communication Interface
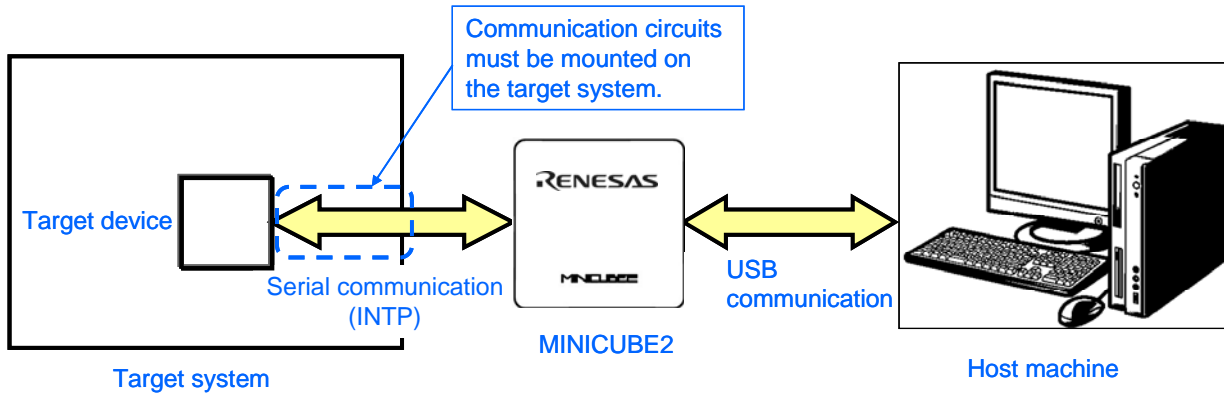
5.1.1 Pin assignment

This section describes the interface signals used between MINICUBE2 and the target system. Table 5-1 lists the pin assignment. Table 5-2 describes the functions of each pin. The pin assignment varies depending on whether the operation is debugging or programming, so design the circuit appropriately according to the circuit connection examples described on the following sections.

Table 5-1. Pin Assignment

Pin No.	Pin Name[Note]	
	For Debugging	For Programming
1	GND	
2	RESET_OUT	
3	DATA	
4	VDD	
5	DATA	
6	R.F.U.	
7	R.F.U.	
8	H/S	
9	CLK	
10	R.F.U.	
11	R.F.U.	
12	INTP	R.F.U.
13	R.F.U.	
14	CLK	
15	RESET_IN	R.F.U.
16	R.F.U.	

Note Signal names in MINICUBE2

TOP VIEW

Pin 1

Target system

Pin assignment of target connector to be mounted on target system
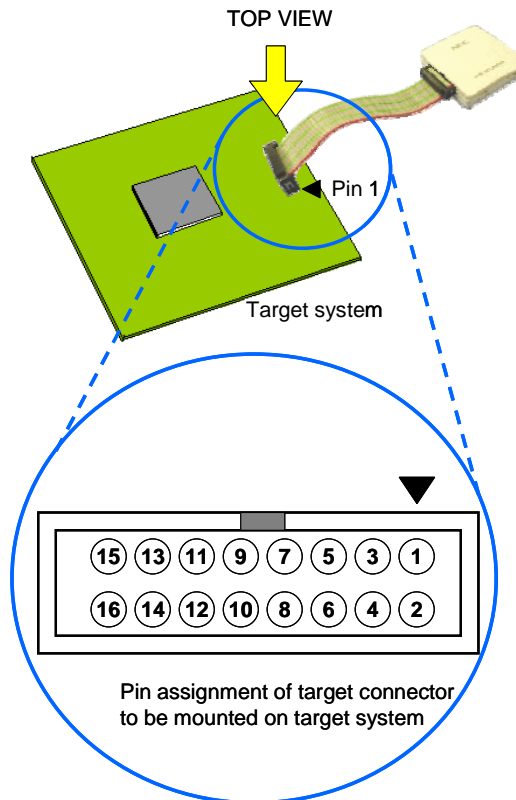
Table 5-2. Pin Functions

Pin Name	IN/OUT[Note]	Description
RESET_IN	IN	Pin used to input reset signal from the target system
RESET_OUT	OUT	Pin used to output reset signal to the target device
CLK	OUT	Pin used to output clock signal to the target device and to set the target device to programming mode
DATA	IN/OUT	Pin used to receive commands or data from the target device and to transmit commands or data to the target device
H/S	IN	Pin used to input handshake signal from the target device
INTP	OUT	Pin used to set the target device to debug mode
R.F.U.	-	This pin is not used, so leave it open.

Note As seen from MINICUBE2

5.1.2 Circuit connection example

Refer to Figure 5-2 and design an appropriate circuit.

Caution 1. The constants described in the circuit connection example are reference values. If you perform flash programming aiming at mass production, thoroughly evaluate whether the specifications of the target device are satisfied.

\<R\>   **2. The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.**

\<R\>   **Figure 5-2. Recommended Circuit Connection**
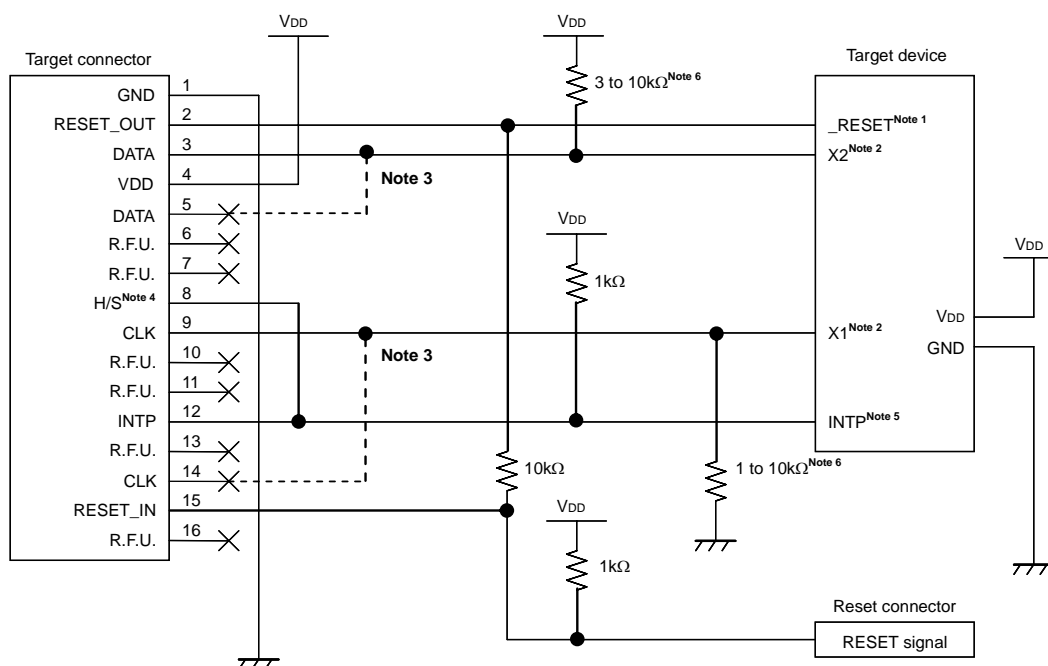
Notes 1. The $\overline{RESET}$ pin is used to download the monitor program at debugger startup or to implement forced reset. Therefore, a pin that alternately functions as the $\overline{RESET}$ pin cannot be used. For reset pin connection, refer to **5.1.3 Connection of reset pin**.

2. This is the pin connection when the X1 and X2 pins are not used in the target system. When using the X1 and X2 pins, refer to **5.1.5 Connection of X1 and X2 pins**.

The X2 pin is gently pulled down in the device during the reset period. With MINICUBE2, it is pulled up at 1.5 kΩ, so use a resistance of about 100 kΩ when pulling it down on the substrate.

3. No problem will occur if the dashed line portions are connected.

4. This pin is connected to enhance the accuracy of time measurement between run and break during debugging. Debugging is possible even if this pin is left open, but measurement error occurs in several ms units.

5. The INTP pin is used for communication between MINICUBE2 and the target device during debugging. When debugging is performed with MINICUBE2, therefore, the INTP pin and its alternate-function pin cannot be used. For INTP pin connection, refer to **5.1.4 Connection of INTP pin**.

For details on the INTP pin to be used, refer to the document indicated in **1. 3 Supported Devices**.

6. This is for pin processing when not used as a device.

To change this value to another resistance value, see **APPENDIX A EQUIVALENT CIRCUIT**.

5.1.3 Connection of reset pin

   This section describes the connection of the reset pin, for which special attention must be paid, in the circuit connection example shown in the previous section.

   During on-chip debugging, a reset signal from the target system is input to MINICUBE2, masked, and then output to the target device. Therefore, the reset signal connection varies depending on whether MINICUBE2 is connected.

   For flash programming, the circuit must be designed so that the reset signals of the target system and MINICUBE2 do not conflict.

   Select one of the following methods and connect the reset signal in the circuit. The details of each method are described on the following pages.

   (1) Automatically switching the reset signal via series resistor (recommended; described in recommended circuit connection in the previous section)
   (2) Manually switching the reset signal with jumper
   (3) Resetting the target device by power-on clear (POC) only

(1) Automatically switching the reset signal via series resistor

Figure 5-3 illustrates the reset pin connection described in **5.1.2 Circuit connection example**.

This connection is designed assuming that the reset circuit on the target system contains an N-ch open-drain buffer (output resistance: 100Ω or less). The VDD or GND level may be unstable when the logic of RESET_IN/OUT of MINICUBE2 is inverted, so observe the conditions described below in **Remark**.

Figure 5-3. Circuit Connection with Reset Circuit That Contains Buffer
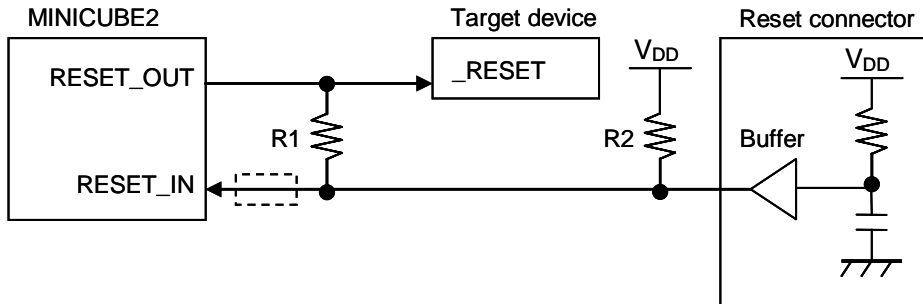
Remark   Make the resistance of at least R1 ten times that of R2, R1 being 10 kΩ or more.

Pull-up resistor R2 is not required if the buffer of the reset circuit consists of CMOS output.

The circuit enclosed by a dashed line is not required when only flash programming is performed.

Figure 5-4 illustrates the circuit connection for the case where the reset circuit on the target system contains no buffers and the reset signal is only generated via resistors or capacitors. Design the circuit, observing the conditions described below in **Remark**.

Figure 5-4. Circuit Connection with Reset Circuit That Contains No Buffers
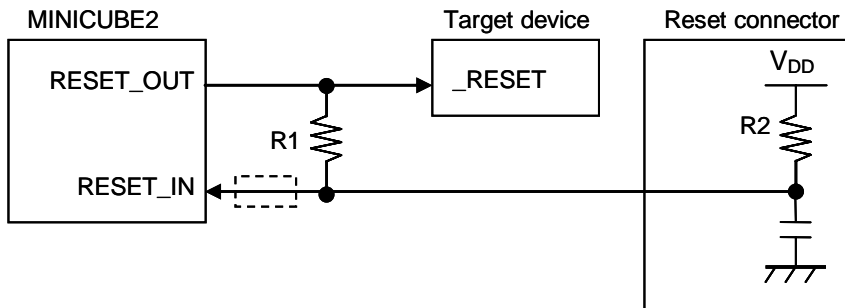
Remark   Make the resistance of at least R1 ten times that of R2, R1 being 10 kΩ or more.

The circuit enclosed by a dashed line is not required when only flash programming is performed.

(2) Manually switching the reset signal with jumper

Figure 5-5 illustrates the circuit connection for the case where the reset signal is switched using the jumper, with or without MINICUBE2 connected. This connection is simple, but the jumper must be set manually.

Figure 5-5. Circuit Connection for Switching Reset Signal with Jumper
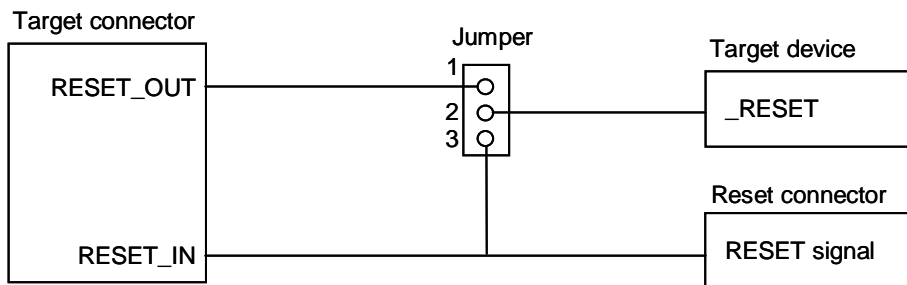
* Jumper setting
  When MINICUBE2 is connected: 1-2 shorted
  When MINICUBE2 is not connected: 2-3 shorted

(3) Resetting the target device by power-on clear (POC) only

Figure 5-6 illustrates the circuit connection for the case where the target device is only reset via POC without using the reset pin. RESET_OUT is valid only when the debugger is running or during flash programming.
The operation is not guaranteed if the power to the target system is shut down during debugging. Note that the POC function cannot be emulated.

Figure 5-6. Circuit Connection for the Case Where Target Device Is Only Reset via POC
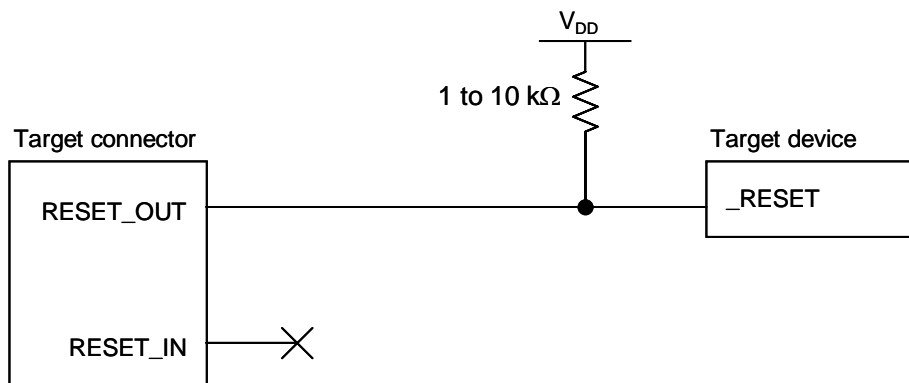
5.1.4 Connection of INTP pin

The INTP pin is used only for communication between MINICUBE2 and the target device during debugging. Design circuits appropriately according to the relevant case among the cases shown below.

(1) INTP pin is not used in target system (as is illustrated in **Figure 5-2**. **Recommended Circuit Connection**)
→ See **Figure 5-7**.

(2) MINICUBE2 is used only for programming, not for debugging
→ See **Figure 5-8**.

(3) MINICUBE2 is used for debugging and debugging of the INTP pin is performed only with a real machine
→ See **Figure 5-9**.

Figure 5-7. Circuit Connection for the Case Where INTP Pin Is Not Used in Target System
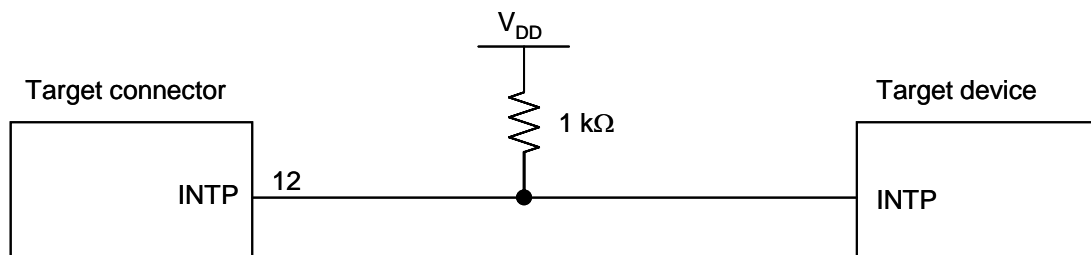
Figure 5-8. Circuit Connection for the Case Where MINICUBE2 Is Used Only for Programming

**Figure 5-9. Circuit Connection for the Case Where MINICUBE2 Is Used for Debugging and
Debugging of INTP Pin Is Performed Only with Real Machine**
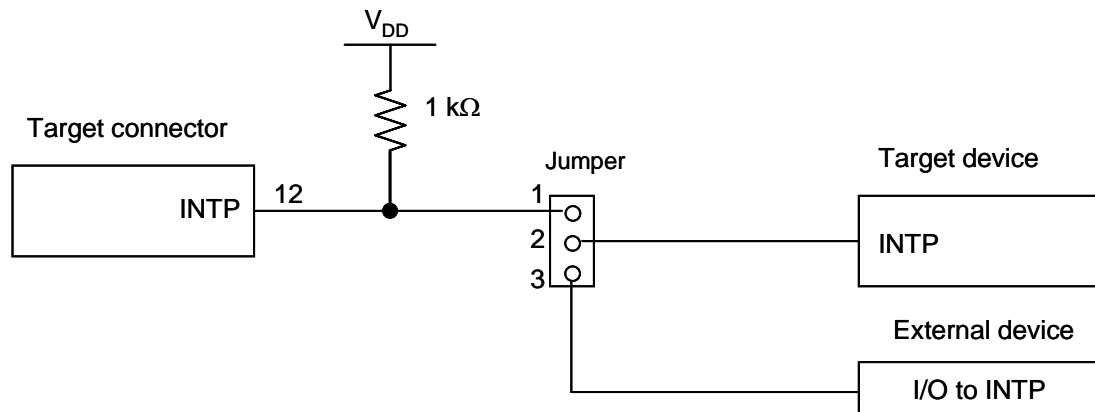
* Jumper setting

When debugging with MINICUBE2 connected: 1-2 shorted

Other than above: 2-3 shorted

Caution   **If debugging is performed with a real machine running, without using MINICUBE2, write the user
program using the QB-Programmer. Programs downloaded by the debugger include the monitor
program, and such a program malfunctions if it is not controlled via MINICUBE2.**

5.1.5 Connection of X1 and X2 pins

The X1 and X2 pins are used when the debugger is started for the first time (when downloading the monitor program) and when programming is performed with the QB-Programmer. Design circuits appropriately according to the relevant case among the cases shown below.

(1) X1 and X2 pins are not used in target system
→ See **Figure 5-10**.
(2) X1 and X2 pins are used in target system
→ See **Figure 5-11** and **5.2.6 Debugging of alternate-functions of X1 and X2 pins**.

Figure 5-10. Circuit Connection for the Case Where X1 and X2 Pins Are Not Used in Target System
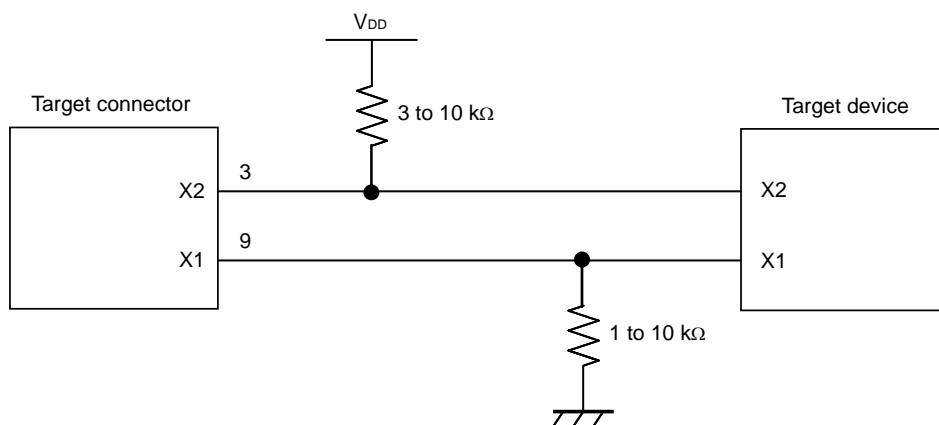
Figure 5-11. Circuit Connection for the Case Where X1 and X2 Pins Are Used in Target System
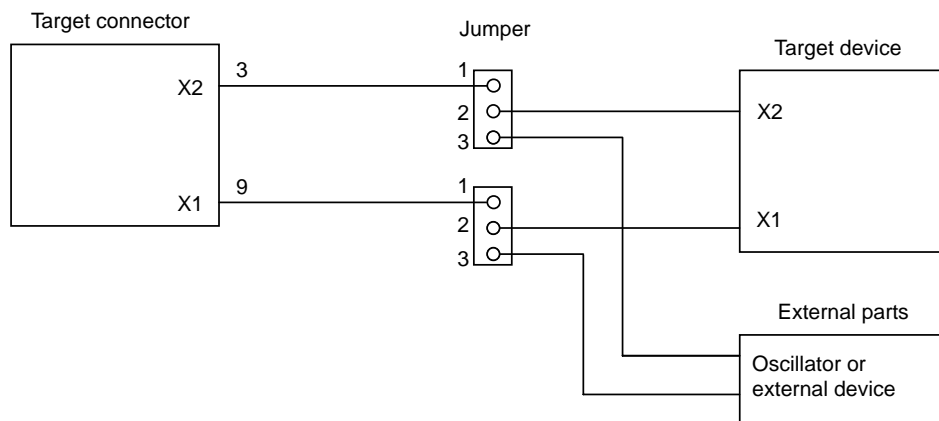
* Jumper setting
When debugger is started for the first time
(downloading the monitor program) or when
programming is performed with QB-Programmer:　1-2 shorted
Other than above:　　　　　　　　　　　　　　　　2-3 shorted

Remark　Refer to the pin connection in Figure 5-10 when the X1 or X2 pin is not used.

5.1.6 Mounting connector onto target system

The connector must be mounted on the target system for connecting MINICUBE2 with the target system. A 2.54 mm pitch 16-pin general-purpose connector can be used. The following products are recommended.

- HIF3FC-16PA-2.54DS (right-angle type, made by Hirose Electric Co., Ltd.)
- HIF3FC-16PA-2.54DSA (straight type, made by Hirose Electric Co., Ltd.)
- 7616-5002PL (right-angle type, made by Sumitomo 3M, Ltd.)
- 7616-6002PL (straight type, made by Sumitomo 3M, Ltd.)

Compact connectors and loose-wire cables are also available as options. For more information on these products, access the MINICUBE2 information site.

URL: http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html

Figure 5-12. 2.54 mm Pitch 16-Pin General-Purpose Connector Pin Layout
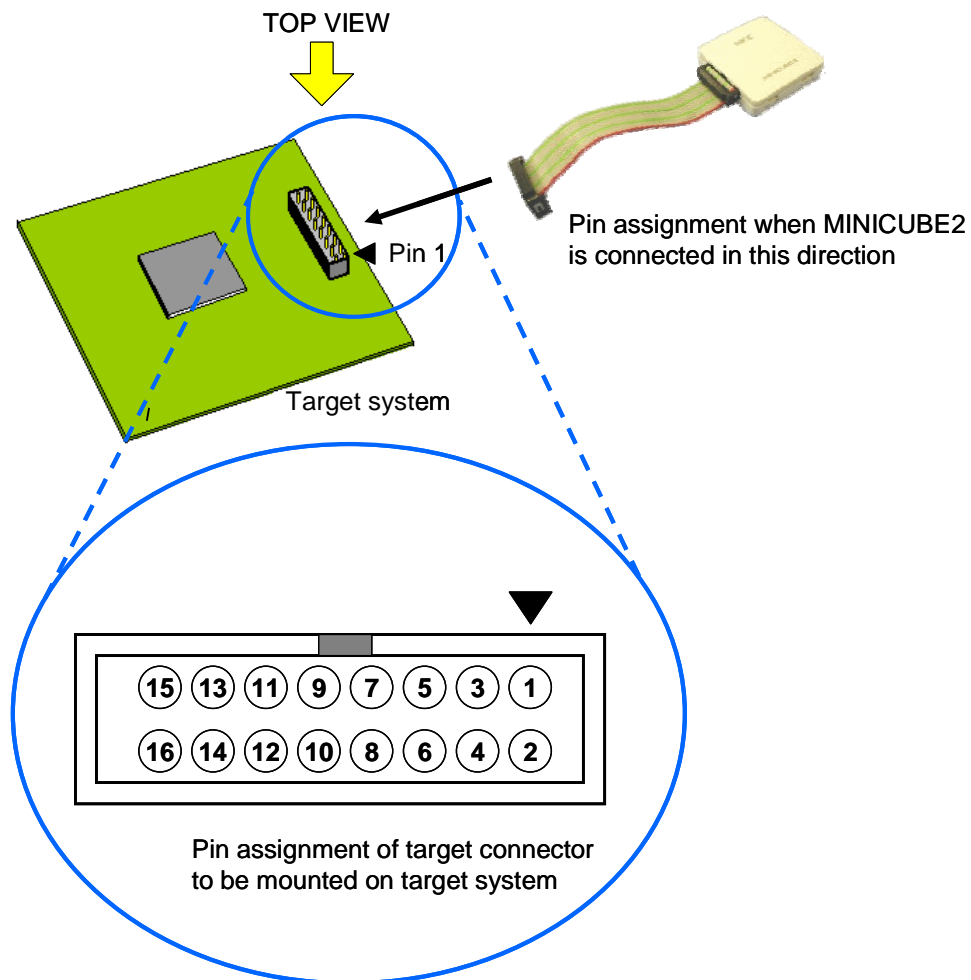
TOP VIEW

Pin 1

Pin assignment when MINICUBE2 is connected in this direction

Target system

15 13 11 9 7 5 3 1
16 14 12 10 8 6 4 2

Pin assignment of target connector to be mounted on target system

5.2 On-Chip Debugging

This section describes the system configuration, startup/shutdown procedure and cautions for debugging when on-chip debugging is performed with MINICUBE2.

5.2.1 Debug functions

Table 5-3 lists the debug functions when a 78K0S microcontroller is the target device and the ID78K0S-QB is used. If a debugger other than the ID78K0S-QB (made by partner manufacturer) is used, the available functions may differ, so check the specifications of the debugger used.

Table 5-3. Debug Functions

Functions	Specifications
Security ID	Unavailable
Download	Available
Execution	Go & Go, Start from Here, Come Here, Restart, step execution
Hardware break	Unavailable
Software break	2,000 points
Forced break	Available except while interrupts are disabled
RAM monitoring	Unavailable
DMM (writing to memory during RUN)	Unavailable
Pin masking	Available (external reset pin only)
Time measurement (from execution start to break)	Measurement resolution: 100 μs Max. measurement time: Approximately 100 hours
User spaces used for debugging	Internal ROM: 308 bytes Internal RAM: 5 bytes (used as stack)
Function pins used for debugging	INTP[Note], X1[Note], X2[Note], $\overline{\text{RESET}}$

Note The INTP pins to be used are described in the document ″QB-MINI2 Operating Precautions″.

The X1 and X2 pins are used only when the debugger is started for the first time (when downloading the monitor program).

5.2.2 System configuration

Figure 5-13 illustrates the system configuration for on-chip debugging.

Figure 5-13. System Configuration for On-Chip Debugging
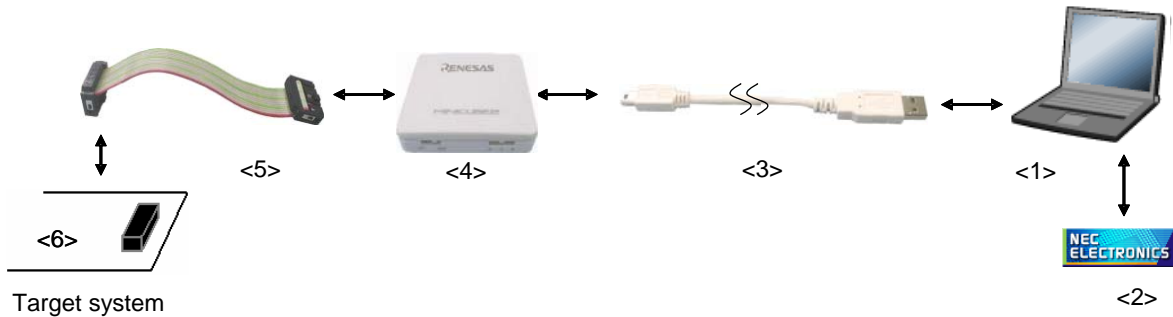
Target system

<1> Host machine

Products with USB ports

<2> Software

Includes the debugger, USB driver, device files, and so on.

Download these Renesas Electronics software from the following website (ODS).

URL: http://www2.renesas.com/micro/en/ods/index.html

<3> USB cable (accessory)

<4> MINICUBE2 (this product)

<5> 16-pin target cable (accessory)

<6> Target connector (sold separately)

5.2.3 System startup procedure

This section describes the system startup procedure. Observe the following order.

(1) Preparation and installation of software

The following software is required to perform on-chip debugging. For details on preparation and installation of software, refer to the setup manual supplied with MINICUBE2.

- Debugger
- USB driver
- Device file (may be unnecessary when using a partner manufacture•fs debugger)

(2) Switch setting

Set the mode select switch to ″M1″. Set the power select switch by referring to Table 5-4.

Caution   Do not change the switch setting while the USB cable is connected.

Table 5-4. Setting of Power Select Switch

Setting	Description
3	3 V is supplied from MINICUBE2 to the target system[Note].
5	5 V is supplied from MINICUBE2 to the target system[Note].
T	Power supply of the target system is used.
	MINICUBE2 only detects the power for the target system.

Note The maximum rating of the current is 100 mA, so do not use MINICUBE2 with the target system with the higher current rating. The power is always supplied after MINICUBE2 is connected to the host machine.

(3) Connecting the target system

Connect MINICUBE2 to the target system as shown in Figure 5-14, **before the power to the target system is turned on**.

Figure 5-14. Connecting MINICUBE2 to Target System

16-pin target cable

MINICUBE2
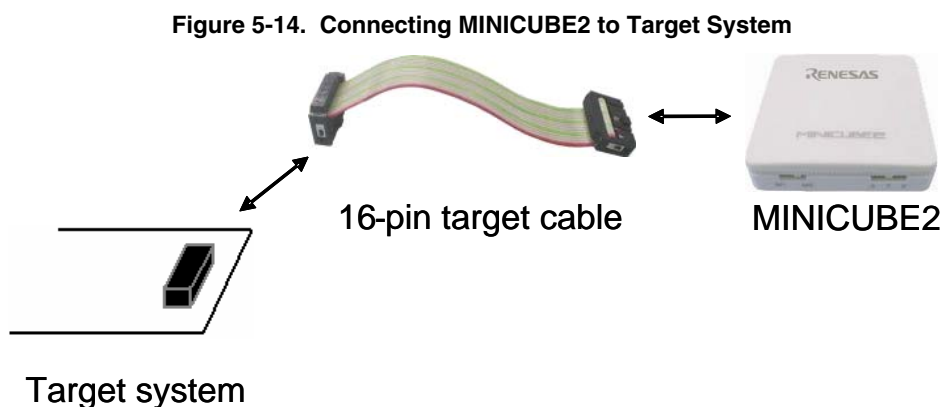
Target system

(4) Connecting the USB cable

Connect MINICUBE2 to the host machine as shown in Figure 5-15, **before the power to the target system is turned on**.

When the power select switch is set to ″T″, the mode LED **flashes** white after connection.

When the power select switch is set to ″3″ or ″5″, the mode LED **glows** white after connection.

Figure 5-15. Connecting MINICUBE2 to Host Machine
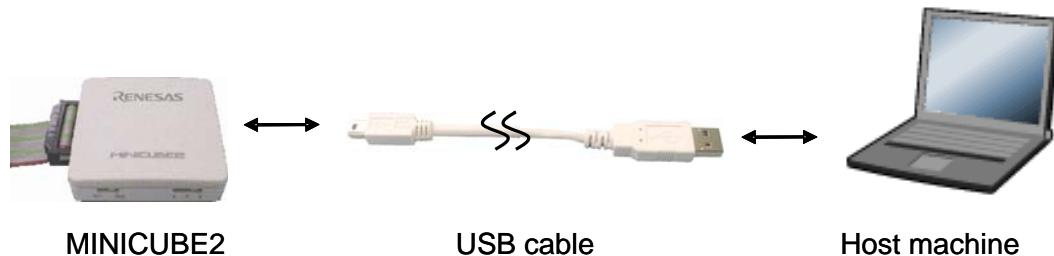
MINICUBE2   USB cable   Host machine

(5) Applying power to target system

Turn on power to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″. The mode LED glows white after power application.

(6) Debugger startup

Start the debugger. The mode LED glows green after debugger startup.

For the operation after this step, refer to the user′s manual for the debugger.

If the debugger does not start normally or the operation is unstable, the possible causes may be the following.

• Communication error between MINICUBE2 and target system

   Whether communication is performed normally can be checked using the OCD Checker. For details, refer to the user′s manual for the OCD Checker.

• The user resource has not been secured

   To perform debugging with MINICUBE2, the debug monitor area must be secured. For details, refer to **5.2.5 Securing of user resources**.

• Unsupported software (debugger, device file, or firmware) is used

   The software used may not support debugging of the target device. Refer to the document ″QB-MINI2 Operating Precautions″ and check the supported version. When using software provided by a Renesas Electronics partner, refer to the documents prepared by the partner company.

• Defect of MINICUBE2

   MINICUBE2 may have a defect. Refer to **CHAPTER 7 SELF-TESTING**.

5.2.4 System shutdown procedure

Terminate debugging and shutdown the system in the following order.

If the following order is not observed, the target system or MINICUBE2 may be damaged.

(1) Debugger termination

Terminate the debugger.

(2) Target system power shutdown

Shutdown the power supplied to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″.

(3) Unplugging the USB cable

Unplug the USB cable from MINICUBE2 or the host machine.

(4) Unplugging the target cable

Unplug the target cable from MINICUBE2 or the target system.

5.2.5 Securing of user resources

The user must prepare the following to perform communication between MINICUBE2 and the target device and implement each debug function. Refer to the descriptions on the following pages and set these items in the user program or using the compiler options.

When C-SPY manufactured by IAR Systems is used, read also the following material.
- IAR C-SPY Hardware Debugger Systems User Guide issued by IAR Systems

- Securement of memory space
  The shaded portions in Figure 5-16 are the areas reserved for placing the debug monitor program, so user programs cannot be allocated in these spaces.

Figure 5-16. Memory Spaces Where Debug Monitor Programs Are Allocated

Internal ROM space

304 bytes	Internal ROM end address
2 bytes	0x7E (for software break)
2 bytes	0x0A or 0x18 (INTP interrupt vector)

Internal RAM space

	Internal RAM end address
5 bytes	Stack area for debugging

☐: Debug monitor program area
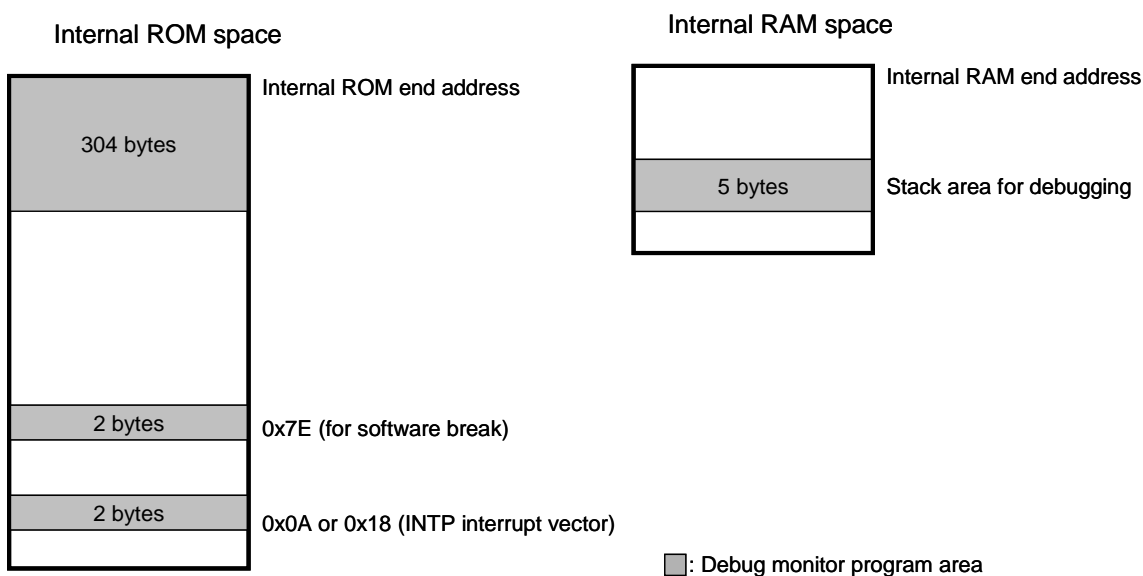
- Securement of serial interface for communication
  The register settings, concerning the INTP pin used for communication between MINICUBE2 and the target device, performed by the debug monitor program must not be changed.

(1) Securement of area for debug monitor program

In the internal ROM space of the target device, the shaded portions in Figure 5-16 are the areas where the debug monitor program is allocated. The monitor program performs initialization processing for debug communication interface and RUN or break processing for the CPU. The internal ROM area must be filled with 0xFF. This area must not be rewritten by the user program.

[How to secure areas]

It is not necessarily required to secure this area if the user program does not use this area.

To avoid problems that may occur during the debugger startup, however, it is recommended to secure this area in advance, using the compiler.

The following shows examples for securing the area, using the Renesas Electronics assembler RA78K0S. Add the assemble source code, as shown below.

• Assemble source example that securse INTP interrupt vector

```
SSS     CSEG    AT      0018h;  ″SSS″ is an arbitrary symbol name (eight characters or less)
        DB      0ffh,   0ffh
```

Remark Describe the INTP interrupt vector address to the portion of ″0018h″. The following shows examples of the 78K0S/Kx1+.

     When using 78K0S/KU1+ or 78K0S/KY1+: INTP1 (000ah)

     When using 78K0S/KA1+ or 78K0S/KB1+: INTP3 (0018h)

• Assemble source example that secures an area for software break

```
SSS     CSEG    AT      007eh;  ″SSS″ is an arbitrary symbol name (eight characters or less)
        DB      0ffh,   0ffh
```

• Assemble source example that secures monitor program area lower than internal ROM end address

```
SSS     CSEG    AT      0ed0h;  ″SSS″ is an arbitrary symbol name (eight characters or less)
        DB      0ffh,   0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh
        DB      0ffh,   0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh
        :                                                                    19 lines
        :
        DB      0ffh,   0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh
```

Remark Describe the address to the portion of ″0ed0h″ in accordance with the flash memory capacity of the target
device (see the following table).

Flash Memory Capacity of Target Device	Value Described in Assembler Source
1 KB	0x2D0
2 KB	0x6D0
4 KB	0xED0
8 KB	0x1ED0

(2) Securement of stack area for debugging

In the internal RAM space of the target device, the shaded portion in Figure 5-16 are the area where the debug
monitor program uses as the stack area (5 bytes). Since this area is allocated immediately before the stack area
that is used by the user program, the address of this area varies depending on the stack increase and decrease.
Therefore, the stack pointer must be set by making allowances for the stack size (5 bytes) used for debugging.

The following figure illustrates the case where the stack area is increased when the internal high-speed RAM starts
from 0xFE00.
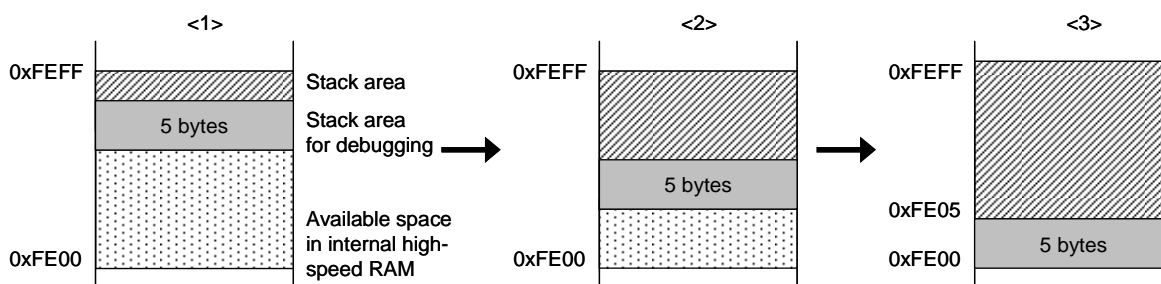
[How to secure areas]

Refer to the address range shown below and set the stack pointer.

Example When internal high-speed RAM starts from 0xFE00

Within the range 0xFE05 to 0xFEFF

(3) Securement of communication serial interface

The INTP pin is used for communication between MINICUBE2 and the target system. The settings related to the INTP pin are performed by the debug monitor program, but if the setting is changed by the user program, a communication error may occur.

To prevent such a problem from occurring, communication serial interface must be secured in the user program.

[How to secure communication serial interface]

Create the user program observing the following points.

- Interrupt mask flag register
  Do not disable interrupts using the interrupt mask flag register corresponding to the INTP pin used.

  Example   Setting other than below is prohibited when the target device is the 78K0S/KB1+ and INTP3 is used.

	7	6	5	4	3	2	1	0
MK1	1	x	x	x	x	0	x	1

  x: Any

- Port mode registers of alternate-function ports
  Do not set to the output mode the alternate-function port corresponding to the INTP pin used.

  Example   Setting other than below is prohibited when the target device is the 78K0S/KB1+ and INTP3 is used.

	7	6	5	4	3	2	1	0
PM4	x	x	x	x	x	x	1	x

  x: Any

- External interrupt mode registers
  Do not set to other than the rising edge the external interrupt mode register corresponding to the INTP pin used.

  Example   Setting other than below is prohibited when the target device is the 78K0S/KB1+ and INTP3 is used.

	7	6	5	4	3	2	1	0
INTM1	x	x	x	x	x	x	0	1

  x: Any

5.2.6 Debugging of alternate-functions of X1 and X2 pins

The X1 and X2 pins are used when the debugger is started for the first time (when downloading the monitor program) and when programming is performed with the QB-Programmer.

It is not necessary to use the X1 and X2 pins if the monitor program has already been downloaded to the target device, so the X1 and X2 pins and their alternate functions can be debugged by using the following procedures. To implement debugging of the X1 and X2 pins, the circuit must be designed appropriately. For details, refer to **5.1.5 Connection of X1 and X2 pins**.

(1) Disconnect the external parts connected to the X1 and X2 pins in the target device. (If the circuit is designed as shown in Figure 5-11, short jumper pins 1 and 2.)

(2) Turn on power to the target system and MINICUBE2 in accordance with **5.2.3 System startup procedure**.

(3) Start the debugger. At this time, do not download the load module[Note]. (The monitor program will be downloaded automatically immediately after startup.)

(4) Terminate the debugger.

(5) Turn off power to the target system and MINICUBE2 in accordance with **5.2.4 System shutdown procedure**.

(6) Connect the external parts to the X1 and X2 pins in the target device. (If the circuit is designed as shown in Figure 5-11, short jumper pins 2 and 3.)

(7) Turn on power to the target system and MINICUBE2 in accordance with **5.2.3 System startup procedure**.

(8) Start the debugger.

Note　If a project has already been created in the project manager (PM+), the load module will automatically be downloaded. In this case, terminate PM+ and then start the debugger.

Remark　The next time the debugger is started, perform the procedure from step (7). However, if the program is written via the QB-Programmer and the monitor program is erased, the procedure must be performed from step (1) again.

5.2.7 Cautions on debugging

This section describes cautions on performing on-chip debugging for a 78K0S microcontroller.

Be sure to read the following to use MINICUBE2 properly.

(1) Handling of device that was used for debugging

Do not mount a device that was used for debugging on a mass-produced product, because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed.

(2) Cases where debugger operation speed is degraded

The debugger operation speed may be degraded when too many contents of the memory or registers are displayed in a debugger window, particularly when using a host machine with a USB 1.1 interface.

The debugger response is also slowed while the flash memory is being overwritten.

This phenomenon may be improved to some extent by raising the CPU operation clock frequency, by setting the PCC or PPCC register.

(3) Reset processing

If the RESET pin of the target device alternately functions as a port, enable the RESET pin by the user program settings, such as option byte setting.

(4) When breaks cannot be executed

Forced breaks cannot be executed if one of the following conditions is satisfied.

- Interrupts are disabled (DI)
- Interrupts issued for the INTP pin, which is used for communication between MINICUBE2 and the target device, are masked
- The alternate-function port corresponding to the INTP pin is not set to the input mode
- The external interrupt edge corresponding to the INTP pin is not set to the rising edge
- Standby mode is entered while standby release by a maskable interrupt is prohibited

(5) Flash self programming

If a space where the debug monitor program is allocated is rewritten by flash self programming, the debugger can no longer operate normally.

(6) Operation after reset

After an external pin reset or internal reset, the monitor program performs debug initialization processing. Consequently, the time from reset occurrence until user program execution differs from that in the actual device operation.

(7) Emulation of POC function

The POC function of the target device cannot be emulated. Make sure that the power to the target system is not shut down during debugging.

(8) Debugging with real machine running without using MINICUBE2

If debugging is performed with a real machine running, without using MINICUBE2, write the user program using the QB-Programmer. Programs downloaded by the debugger include the monitor program, and such a program malfunctions if it is not controlled via MINICUBE2.

(9) Caution on protect byte

Set the protect byte so as to permit programming to the entire block and erasing of the entire block.

5.3 Flash Programming

This section describes the system configuration and startup/shutdown procedure when flash programming is performed for a 78K0S microcontroller, using MINICUBE2.

#### 5.3.1 Specifications of programming function

Table 5-5. Specifications of Programming Function

Functions	Specifications
Host interface	USB 2.0 (compatible with 1.1)
Target interface	UART
Target system voltage	1.65 to 5.5 V (depends on the target device)
Clock supply	8 MHz clock can be supplied
Power supply	3 or 5 V (maximum current rating: 100 mA)
Acquisition of device-specific information	Parameter file for MINICUBE2 made by Renesas Electronics is used
Security flag setting	Available
Standalone operation	Unavailable (must be connected to host machine)

\<R\> (Target system voltage)
\<R\> (Acquisition of device-specific information)

5.3.2 System configuration

Figure 5-17 illustrates the system configuration for flash programming.

Figure 5-17. System Configuration for Flash Programming
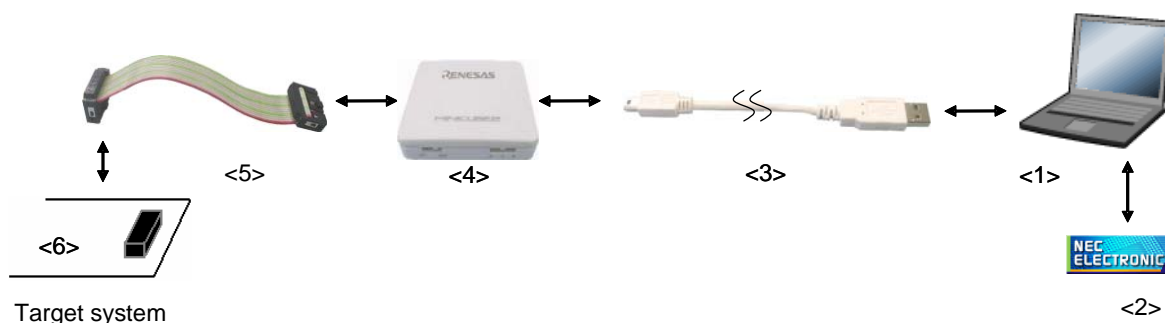
Target system   <2>

<1> Host machine
   Products with USB ports
<2> Software
   Includes the QB-Programmer USB driver, parameter files, and so on.
   Download these Renesas Electronics software from the following website (ODS).
      URL: http://www2.renesas.com/micro/en/ods/index.html
<3> USB cable (accessory)
<4> MINICUBE2 (this product)
<5> 16-pin target cable (accessory)
<6> Target connector (sold separately)

5.3.3 System startup procedure

This section describes the system startup procedure. Observe the following order.

(1) Preparation and installation of software

The following software is required to perform flash programming. For details on preparation and installation of software, refer to the setup manual supplied with MINICUBE2.

- QB-Programmer
- USB driver
- Parameter file

(2) Switch setting

Set the mode select switch to ″M1″. Set the power select switch by referring to the following table.

Caution   Do not change the switch setting while the USB cable is connected.

Table 5-6. Setting of Power Select Switch

Setting	Description
3	3 V is supplied from MINICUBE2 to the target system[Note].
5	5 V is supplied from MINICUBE2 to the target system[Note].
T	Power supply of the target system is used.
	MINICUBE2 detects the power for the target system.
	The detected power is also used as the power supply for the communication interface.

Note The maximum rating of the current is 100 mA, so do not use MINICUBE2 with the target system with the higher current rating. The power is always supplied after MINICUBE2 is connected to the host machine.

(3) Connecting the target system

Connect MINICUBE2 to the target system as shown in Figure 5-18, **before the power to the target system is turned on**. When the power select switch is set to ″T″, the step of **(4) Connecting the USB cable** can be performed before this step.

Figure 5-18. Connecting MINICUBE2 to Target System

16-pin target cable   MINICUBE2
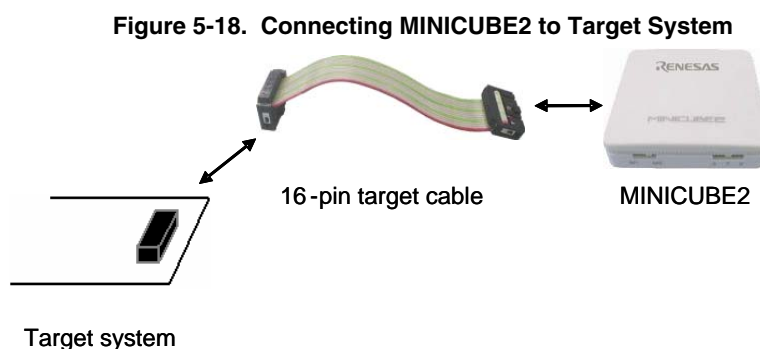
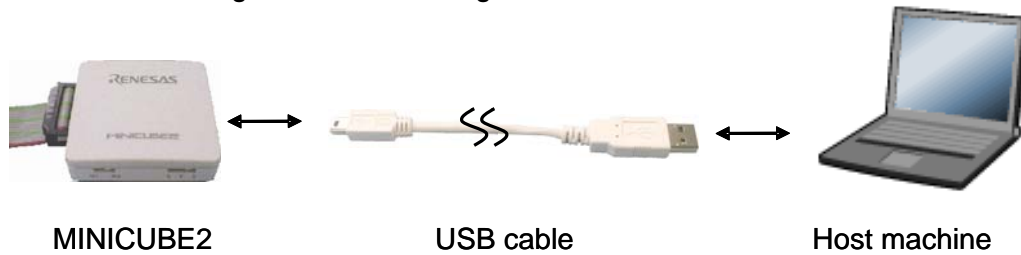Target system

(4) Connecting the USB cable

Connect MINICUBE2 to the host machine as shown in Figure 5-19, **before the power to the target system is turned on**.

When the power select switch is set to ″T″, the mode LED **flashes** white after connection.

When the power select switch is set to ″3″ or ″5″, the mode LED **glows** white after connection. The power is always supplied after MINICUBE2 is connected to the host machine with the USB cable.

Caution   Do not change the switch setting while the USB cable is connected.

Figure 5-19. Connecting MINICUBE2 to Host Machine

MINICUBE2   USB cable   Host machine

(5) Applying power to target system

Turn on power to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″. The mode LED glows white after power application.

(6) QB-Programmer startup

Start the QB-Programmer from the Start menu of Windows. The mode LED glows green after QB-Programmer startup.

A series of basic operations after this step is described in the following section. For detailed usage of the QB-Programmer, refer to the **QB-Programmer Programming GUI Operation User's Manual**.

5.3.4 Usage examples

　　This section describes a series of basic operations using the QB-Programmer, using the μPD78F9234 as an example. The operations from execution of the [Autoprocedure (EPV)] command to programming for the target device are described. For other commands and their applications, refer to the **QB-Programmer Programming GUI Operation User's Manual**.

The conditions used for a series of operations in this section are as follows.

\<Target system\>

Target device:	μPD78F9234
Power supply voltage:	5 V (supplied from MINICUBE2)
Communication channel:	UART, 115,200 bps

\<MINICUBE2\>

Switches:	Mode select switch: M1
	Power select switch: 5

\<QB-Programmer\>

|	
Parameter file:	78F9234.PRM
Clock setting:	8 MHz (supplied from MINICUBE2)
Operation mode:	chip
Program file:	sample.hex
Command options:	[Blank check before Erase] selected
	[Checksum flag after Program] selected

(1) Setting of programming environment

Set the programming environment, following the steps <1> to <6> described below.

<1> Click the [Device] menu on the menu bar and then click [Setup...].
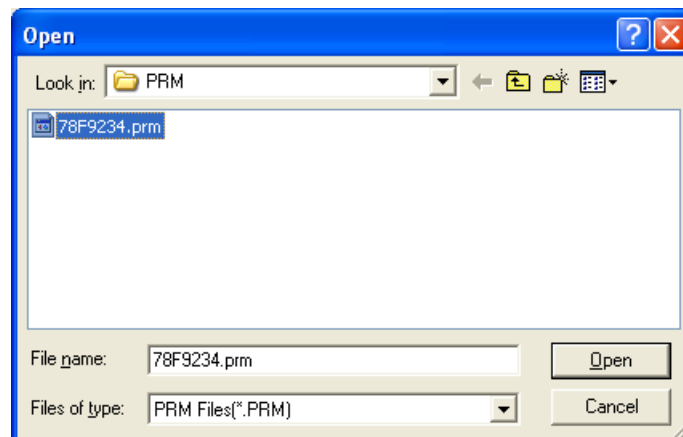
<2> The Device Setup dialog box appears, in which the [Standard] tab is active.

Figure 5-20. [Standard] Tab of Device Setup Dialog Box
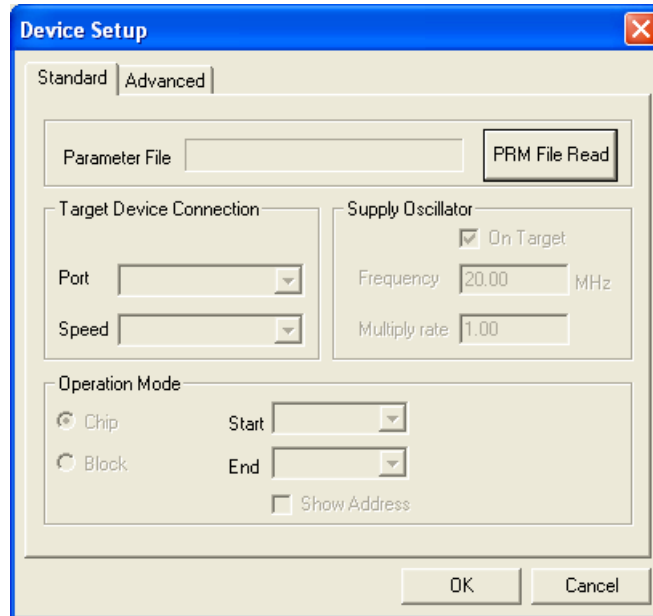
<3> Click the PRM File Read button to open the Parameter File Select dialog box. Select the parameter file for the target device and then click the Open button.

Figure 5-21. Parameter File Select Dialog Box

<4> Set the items in the [Target Device Connection], [Operation Mode] and [Supply Oscillator] areas, in accordance with your programming environment. The following figure shows an example.

Figure 5-22. [Standard] Tab of Device Setup Dialog Box After Setting
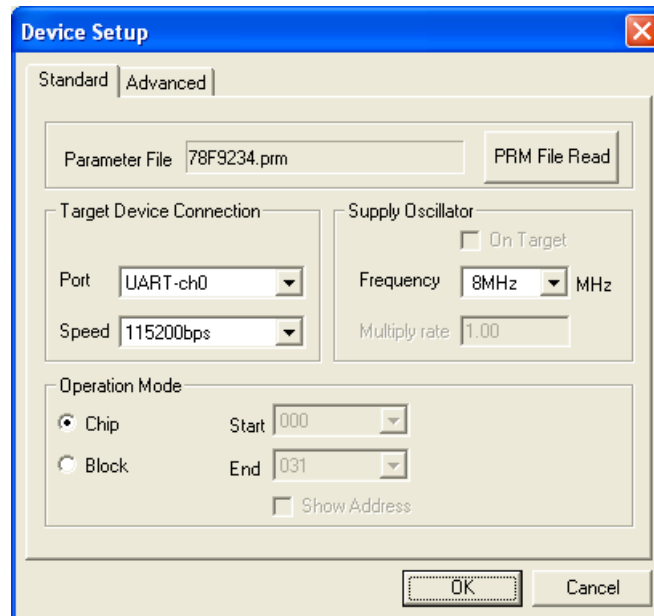
<5> Next, set the items in the [Command options] and [Security flag settings] areas, in accordance with your programming environment. The following figure shows an example.

Figure 5-23. [Advanced] Tab of Device Setup Dialog Box
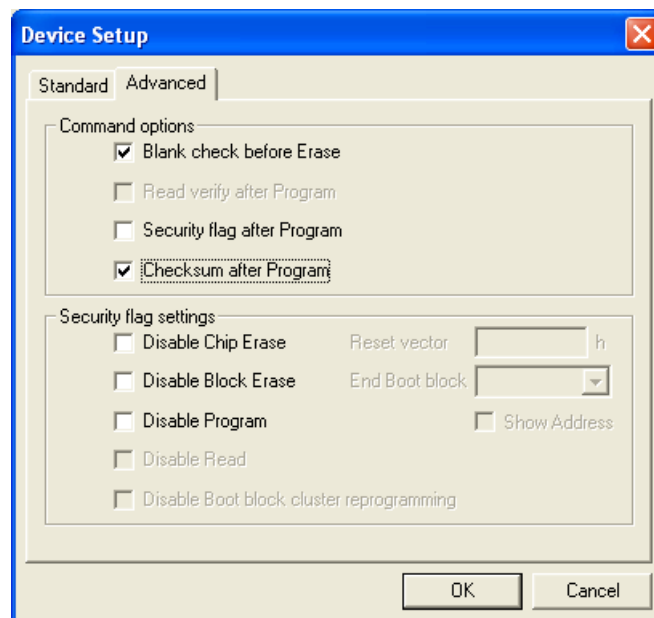
<6> By clicking the OK button, the programming environment is set and the Device Setup dialog box is closed. The main window appears as follows.

Figure 5-24. Completion of Programming Environment Setting
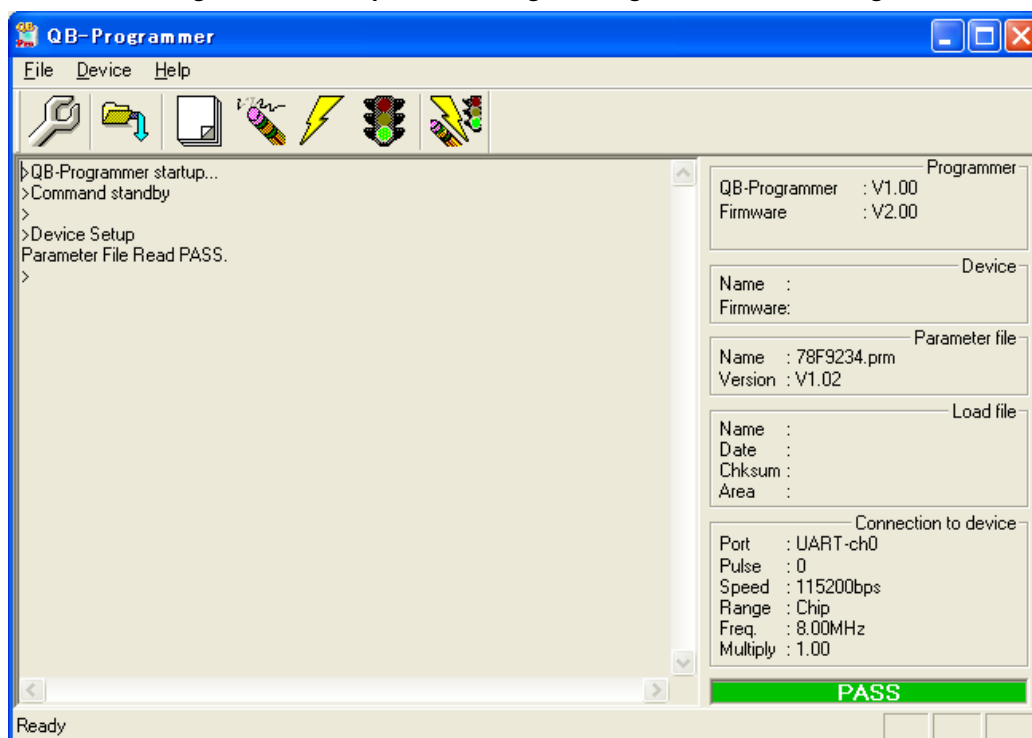
(2) Selection of program file

Select the program file, following the steps <1> to <3> described below.

<1> Click the [File] menu on the menu bar and then click [Load...].

<2> The program file select dialog box appears.

Figure 5-25. Program File Select Dialog Box
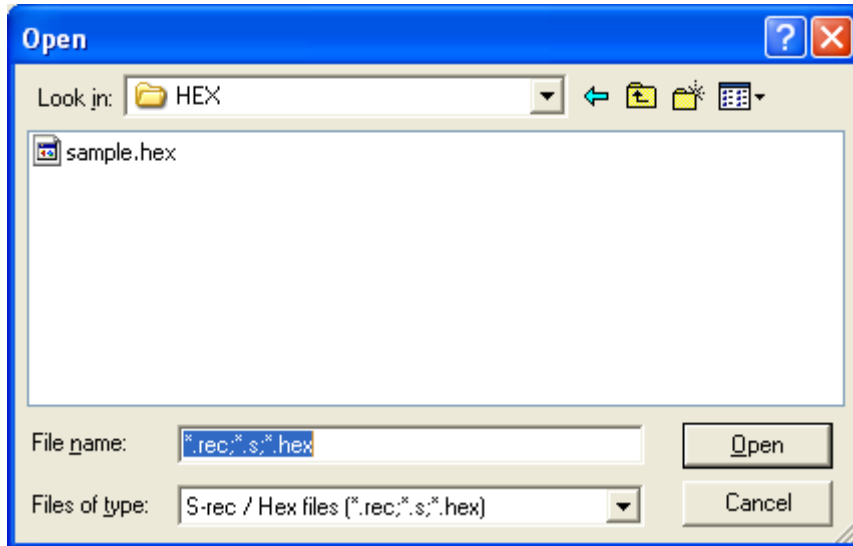
<3> Select the program file for the target device and then click the Open button.

Figure 5-26. Completion of Program File Selection
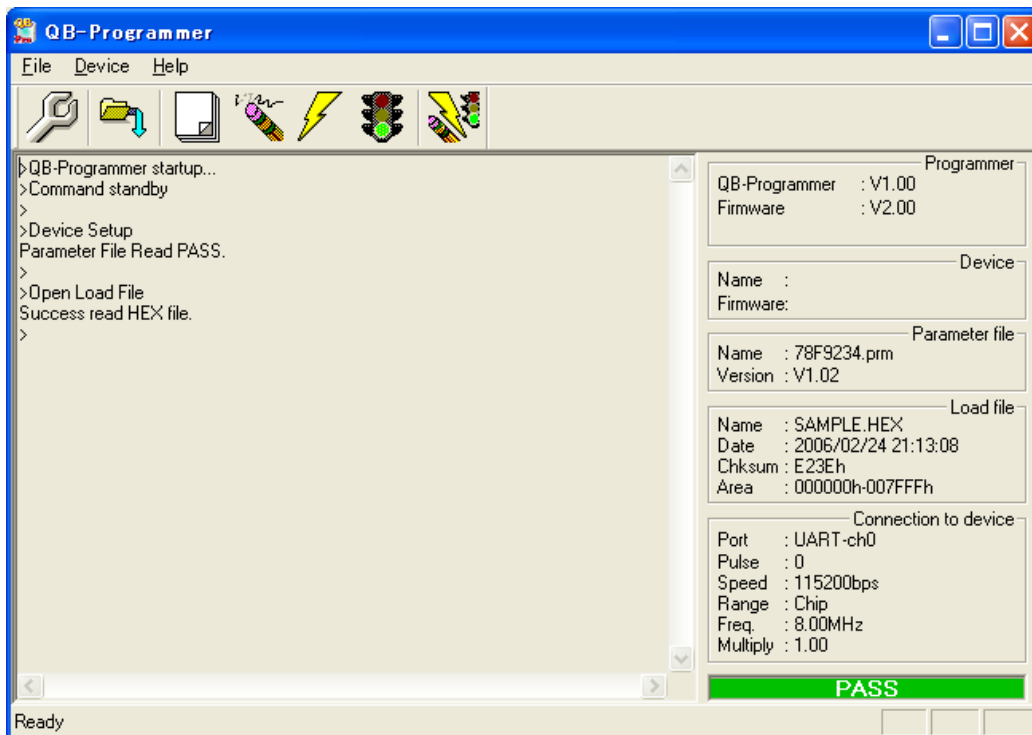
(3) Execution of [Autoprocedure (EPV)] command

Click the [Device] menu on the menu bar and then click [Autoprocedure (EPV)].

<R>   The [Blank Check], [Erase] (if the flash memory in the target device is not blank), [Program], [Checksum] commands are executed for the target device sequentially. The mode LED on MINICUBE2 flashes yellow during execution. When the command execution is completed normally, the mode LED glows green and the following window appears.

Figure 5-27. Completion of [Autoprocedure (EPV)] Command Execution

5.3.5 System shutdown procedure

Terminate flash programming and shutdown the system in the following order.

If the following order is not observed, the target system or MINICUBE2 may be damaged.

(1) Terminating the QB-Programmer

Terminate the QB-Programmer if you are not going to perform programming to other devices.

Remark   When the QB-Programmer is terminated, the setting information is saved in the *qbp.ini* file. Due to this file, the same setting is maintained when the QB-Programmer is started the next time.

(2) Target system power shutdown

Shutdown the power supplied to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″.

(3) Unplugging the USB cable

Unplug the USB cable from MINICUBE2 or the host machine.

When the power select switch is set to ″T″, the step of **(4) Unplugging the target cable** can be performed before this step.

(4) Unplugging the target cable

Unplug the target cable from MINICUBE2 or the target system.

5.3.6 Cautions on flash programming

This section describes the cautions for flash programming. Be sure to read the following for the proper use of MINICUBE2.

- To improve the writing quality, fully understand, verify, and evaluate the following items before using MINICUBE2.
    - Circuits are designed as described in the user's manuals for the device and MINICUBE2.
    - The device, QB-Programmer and MINICUBE2 are used as described in each user's manual.
    - The power supplied to the target system is stable.

CHAPTER 6　HOW TO USE MINICUBE2 WITH 78K0R MICROCONTROLLER

　　　This chapter describes how to use MINICUBE2 when performing on-chip debugging and flash programming for a 78K0R microcontroller.

　　　On-chip debugging is a method to debug a microcontroller mounted on the target system, using a debug function implemented in the device. Since debugging is performed with the target device operating on the board, this method is suitable for field debugging.

　　　Flash programming is a method to write a program to the flash memory embedded in a device. Erasing, writing and verifying the program can be performed on-board with the device.

　　　Read the following chapters if you are using MINICUBE2 for the first time with a 78K0R microcontroller as the target device.

- 6.1 Target System Design

  For communication between MINICUBE2 and the target system, communication circuits must be mounted on the target system. This section describes the circuit design and mounting of connectors.

- 6.2 On-Chip Debugging

  This section describes the system configuration and startup method to perform on-chip debugging with MINICUBE2.


- 6.3 Flash Programming

  This section describes the system configuration and startup method to perform flash programming with MINICUBE2.

6.1 Target System Design

This section describes the target system circuit design required for on-chip debugging and flash programming.

Figure 6-1 presents an overview of the MINICUBE2 communication interface. As shown on the left side of the figure, MINICUBE2 performs serial communication with the target device on the target system. For this communication, communication circuits must be mounted on the target system. Refer to this section to design circuits appropriately.

Figure 6-1. Outline of Communication Interface
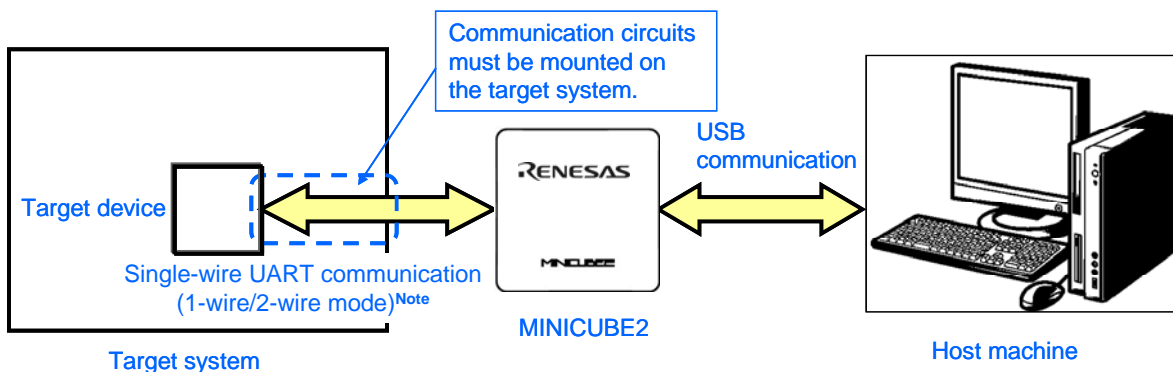
Note   1-wire mode: Single-wire UART communication using TOOL0 pin

          2-wire mode: Single-wire UART communication using TOOL0 and TOOL1 pins

For serial communication, 1-wire mode (single-wire UART communication) using the TOOL0 pin, or 2-wire mode using the TOOL0 and TOOL1 pins is used. Use 1-wire mode when performing flash programming. Use 1-wire mode or 2-wire mode when performing on-chip debugging. Differences between 1-wire mode and 2-wire mode are as shown below. There are no functional differences.

Table 6-1. Differences Between 1-Wire Mode and 2-Wire Mode

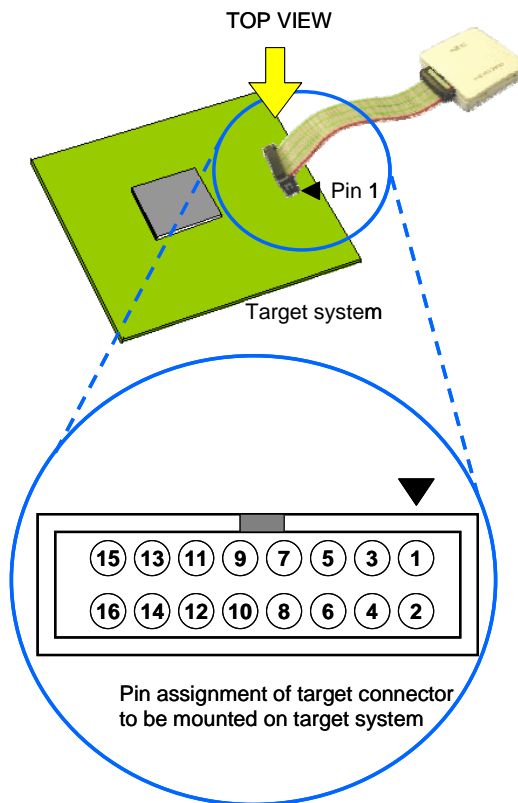| Communication Mode	During Flash Programming	During Debugging
1-wire mode	No differences	• User resources secured for debugging Internal ROM: 1036 bytes Internal RAM: 6 bytes (stack)
2-wire mode		• User resources secured for debugging Internal ROM: 100 bytes (RAM monitor used) / 1036 bytes (RAM monitor not used) Internal RAM: 6 bytes (stack)

<R>

6.1.1 Pin assignment

This section describes the interface signals used between MINICUBE2 and the target system. Table 6-2 lists the pin assignment. Table 6-3 describes the functions of each pin. The pin assignment varies between 1-wire and 2-wire modes, so design the circuit appropriately according to the circuit connection examples described on the following sections.

Table 6-2. Pin Assignment

Pin No.	Pin Name[Note]	
	For 1-Wire Mode	For 2-Wire Mode
1	GND	
2	RESET_OUT	
3	RxD	
4	VDD	
5	TxD	
6	R.F.U.	
7	R.F.U.	
8	R.F.U.	
9	R.F.U.	
10	R.F.U.	
11	R.F.U.	
12	R.F.U.	
13	R.F.U.	
14	FLMD0	
15	RESET_IN	
16	R.F.U.	CLK_IN

Note   Signal names in MINICUBE2

TOP VIEW

Pin 1

Target system

```
15 13 11  9  7  5  3  1
16 14 12 10  8  6  4  2
```

Pin assignment of target connector to be mounted on target system

Table 6-3. Pin Functions

Pin Name	IN/OUT[Note 1]	Description
RESET_IN	IN	Pin used to input reset signal from the target system
RESET_OUT	OUT	Pin used to output reset signal to the target device
FLMD0	OUT	Pin used to set the target device to debug mode or programming mode
RxD[Note 2]	IN/OUT	Pin used to transmit/receive command/data between the target device
TxD[Note 2]	IN/OUT	Pin used to transmit/receive command/data between the target device
CLK_IN	IN	Pin used to input handshake signal from the target device
R.F.U.	-	This pin is not used, so leave it open.

Notes 1. As seen from MINICUBE2

2. The RxD and TxD pins are internally shorted by setting the mode select switch to M1.

6.1.2 Circuit connection example

Refer to Figure 6-2 and design an appropriate circuit.

Caution 1. The constants described in the circuit connection example are reference values. If you perform flash programming aiming at mass production, thoroughly evaluate whether the specifications of the target device are satisfied.

<R>
2. The following figure shows the recommended circuit. Be sure to design the circuit, considering specifications of the target devices, noise characteristic of the circuit, etc.

Figure 6-2. Recommended Circuit Connection
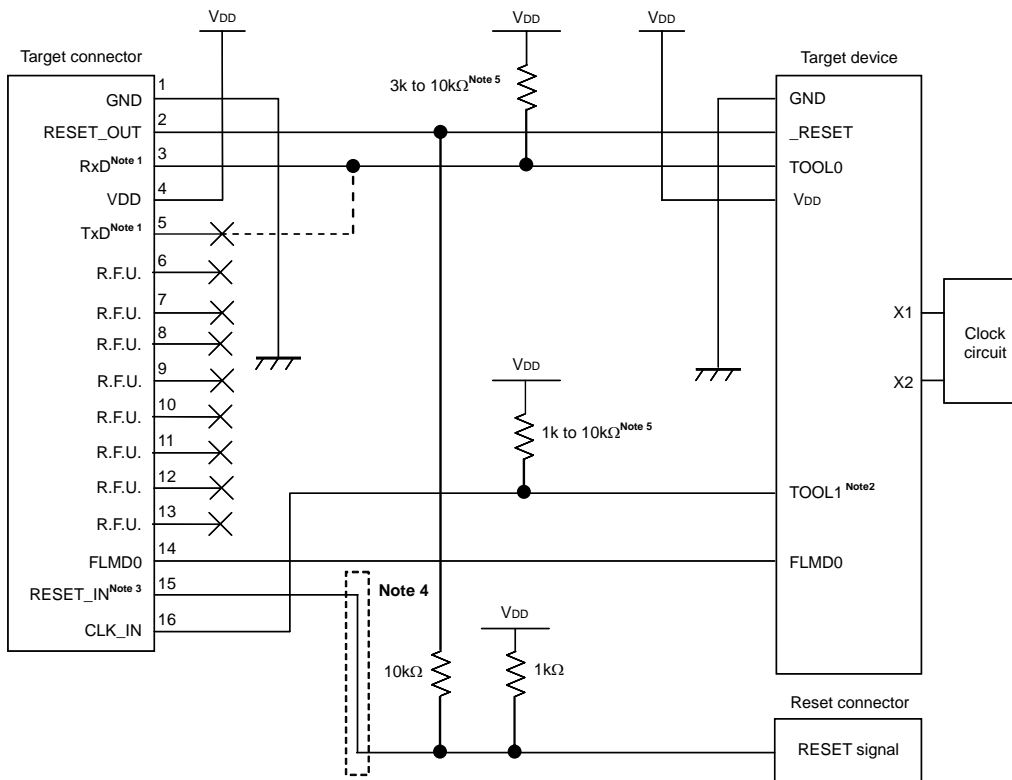
Notes 1. With MINICUBE2, the circuit enclosed by a dashed line is not necessary because the RxD and TxD pins are internally shorted. If any other flash programmer is used, these pins must be shorted on the target system because these pins may not be able to be shorted in the programmer.

2. This connection is required for 2-wire communication, but not for 1-wire communication. This pin is left open when MINICUBE2 is not connected, so connect a pull-up or pull-down resistor to this pin before using.

The TOOL1 pin becomes the pin that supplies a clock from the target device to MINICUBE2. Digital noise that is synchronized with this clock may interfere with the GND power supply. Take general countermeasures against noise, such as inserting a dumping resistor, as required. Take a general

<R> countermeasure for noise, such as inserting a dumping resistor or performing shielding by connecting to GND, as required.

3. This connection is designed assuming that the RESET signal is output from the N-ch open-drain buffer (output resistance: 100Ω or less). For details, refer to **6.1.3 Connection of reset pin**.

4. The circuit enclosed by a dashed line is not required when only flash programming is performed.

5. This is for pin processing when not used as a device.

To change this value to another resistance value, see **APPENDIX A EQUIVALENT CIRCUIT**.

6.1.3 Connection of reset pin

This section describes the connection of the reset pin, for which special attention must be paid, in the circuit connection example shown in the previous section.

During on-chip debugging, a reset signal from the target system is input to MINICUBE2, masked, and then output to the target device. Therefore, the reset signal connection varies depending on whether MINICUBE2 is connected.

For flash programming, the circuit must be designed so that the reset signals of the target system and MINICUBE2 do not conflict.

Select one of the following methods and connect the reset signal in the circuit. The details of each method are described on the following pages.

(1) Automatically switching the reset signal via series resistor (recommended; described in recommended circuit connection in the previous section)

(2) Manually switching the reset signal with jumper

(3) Resetting the target device by power-on clear (POC) only

(1) Automatically switching the reset signal via series resistor

Figure 6-3 illustrates the reset pin connection described in **6.1.2 Circuit connection example**.
This connection is designed assuming that the reset circuit on the target system contains an N-ch open-drain
buffer (output resistance: 100Ω or less). The VDD or GND level may be unstable when the logic of
RESET_IN/OUT of MINICUBE2 is inverted, so observe the conditions described below in **Remark**.

Figure 6-3. Circuit Connection with Reset Circuit That Contains Buffer
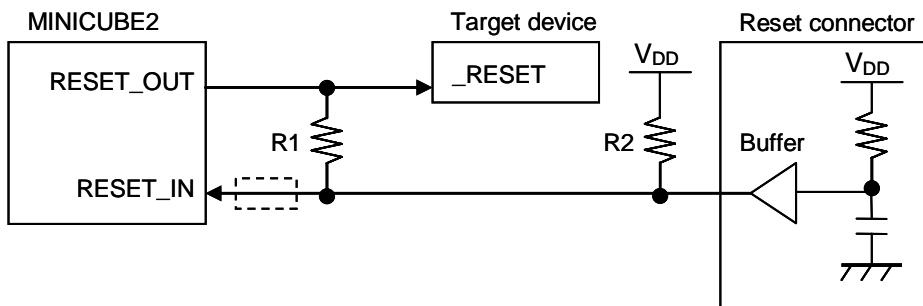
Remark   Make the resistance of at least R1 ten times that of R2, R1 being 10 kΩ or more.
Pull-up resistor R2 is not required if the buffer of the reset circuit consists of CMOS output.
The circuit enclosed by a dashed line is not required when only flash programming is performed.

Figure 6-4 illustrates the circuit connection for the case where the reset circuit on the target system contains no
buffers and the reset signal is only generated via resistors or capacitors. Design the circuit, observing the
conditions described below in **Remark**.

Figure 6-4. Circuit Connection with Reset Circuit That Contains No Buffers
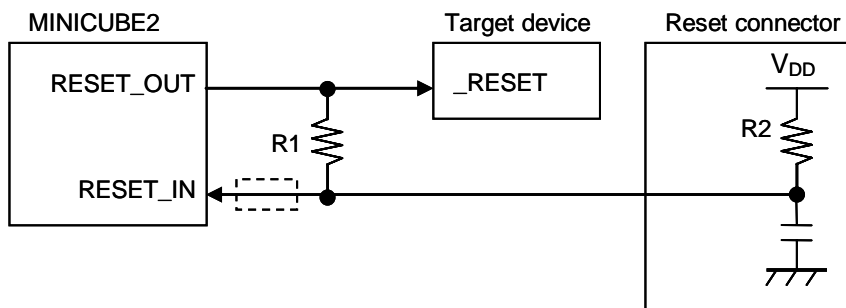

Remark   Make the resistance of at least R1 ten times that of R2, R1 being 10 kΩ or more.
The circuit enclosed by a dashed line is not required when only flash programming is performed.

(2) Manually switching the reset signal with jumper

Figure 6-5 illustrates the circuit connection for the case where the reset signal is switched using the jumper, with or without MINICUBE2 connected. This connection is simple, but the jumper must be set manually.

Figure 6-5. Circuit Connection for Switching Reset Signal with Jumper
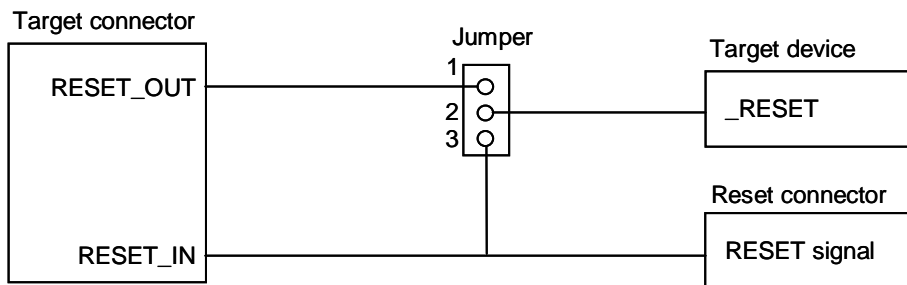
* Jumper setting

When MINICUBE2 is connected: 1-2 shorted

When MINICUBE2 is not connected: 2-3 shorted

(3) Resetting the target device by power-on clear (POC) only

Figure 6-6 illustrates the circuit connection for the case where the target device is only reset via POC without using the reset pin. RESET_OUT is valid only when the debugger is running or during flash programming.

Figure 6-6. Circuit Connection for the Case Where Target Device Is Only Reset via POC

6.1.4 Mounting connector onto target system

The connector must be mounted on the target system for connecting MINICUBE2 with the target system. A 2.54 mm pitch 16-pin general-purpose connector can be used. The following products are recommended.

- HIF3FC-16PA-2.54DS (right-angle type, made by Hirose Electric Co., Ltd.)
- HIF3FC-16PA-2.54DSA (straight type, made by Hirose Electric Co., Ltd.)
- 7616-5002PL (right-angle type, made by Sumitomo 3M, Ltd.)
- 7616-6002PL (straight type, made by Sumitomo 3M, Ltd.)

Compact connectors and loose-wire cables are also available as options. For more information on these products, access the MINICUBE2 information site.

URL: http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html

Figure 6-7. 2.54 mm Pitch 16-Pin General-Purpose Connector Pin Layout
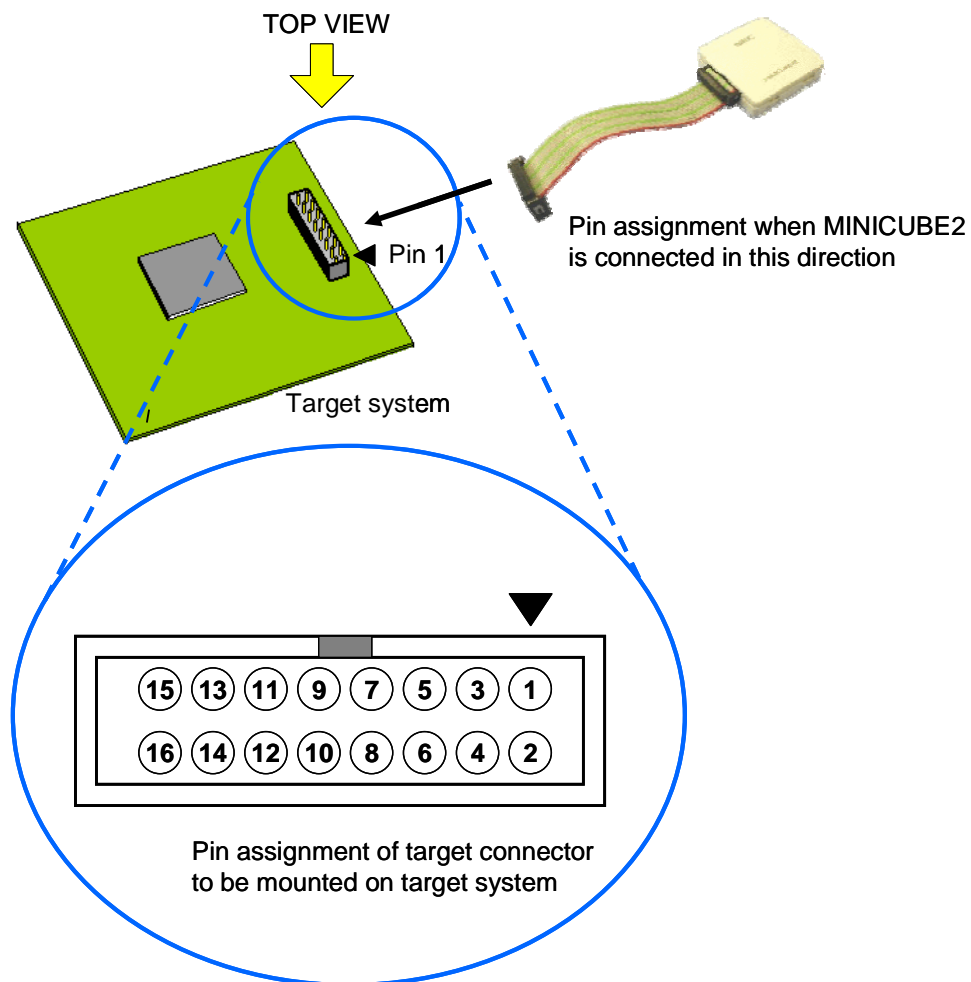
6.2 On-Chip Debugging

This section describes the system configuration, startup/shutdown procedure and cautions for debugging when on-chip debugging is performed with MINICUBE2.

6.2.1 Debug functions

Table 6-4 lists the debug functions when a 78K0R microcontroller is the target device and the ID78K0R-QB is used.

Table 6-4. Debug Functions

	Functions	Specifications
<R>	Target interface	1-wire mode (TOOL0) or 2-wire mode (TOOL0 + TOOL1)
	Security	10 byte ID code authentication
	Download	Available
	Execution	Go & Go, Start from Here, Come Here, Restart, step execution
	Hardware break	1 point (commonly used by execution and access)
	Software break	2,000 points
	Forced break	Available
<R>	Pseudo real-time RAM monitoring RRM)	Available
<R>	DMM (writing to memory during RUN)	Available
	Reset masking	Available (internal/external reset)
	Time measurement (from execution start to break)	Measurement resolution: 100 μs Max. measurement time: Approximately 100 hours
<R>	User spaces used for debugging	In 1-wire mode: Internal ROM: 1036 bytes, internal RAM: 6 bytes[Note] In 2-wire mode: Internal ROM: 100 bytes, internal RAM: 6 bytes[Note]

Note   For details, refer to **6.2.5 Securing of user resources and setting of security ID and on-chip debug option byte**.

6.2.2 System configuration

Figure 6-8 illustrates the system configuration for on-chip debugging.

Figure 6-8. System Configuration for On-Chip Debugging
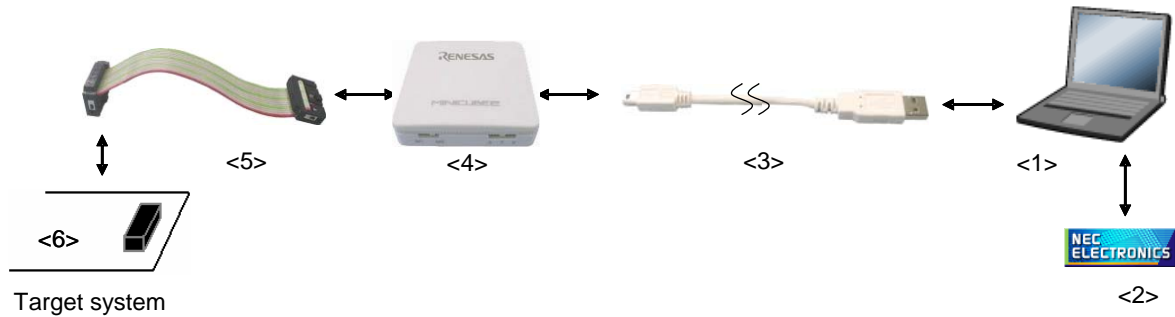
<5>　　　　　　　　　<4>　　　　　　　　　<3>　　　　　　　　　<1>

<6>

Target system　　　　　　　　　　　　　　　　　　　　　　　　　　　<2>

<1> Host machine
　　Products with USB ports
<2> Software
　　Includes the debugger, USB driver, device files, and so on.
　　Download these Renesas Electronics software from the following website (ODS).
　　　　URL: http://www2.renesas.com/micro/en/ods/index.html
<3> USB cable (accessory)
<4> MINICUBE2 (this product)
<5> 16-pin target cable (accessory)
<6> Target connector (sold separately)

6.2.3 System startup procedure

This section describes the system startup procedure. Observe the following order.

(1) Preparation and installation of software

The following software is required to perform on-chip debugging. For details on preparation and installation of software, refer to the setup manual supplied with MINICUBE2.

- Debugger
- USB driver
- Device file (may be unnecessary when using a partner manufacture's debugger)

(2) Switch setting

Set the mode select switch to ″M1″. Set the power select switch by referring to Table 6-5.

Caution   Do not change the switch setting while the USB cable is connected.

Table 6-5. Setting of Power Select Switch

Setting	Description
3	3 V is supplied from MINICUBE2 to the target system[Note].
5	5 V is supplied from MINICUBE2 to the target system[Note].
T	Power supply of the target system is used.
	MINICUBE2 only detects the power for the target system.

Note The maximum rating of the current is 100 mA, so do not use MINICUBE2 with the target system with the higher current rating. The power is always supplied after MINICUBE2 is connected to the host machine.

(3) Connecting the target system

Connect MINICUBE2 to the target system as shown in Figure 6-9, **before the power to the target system is turned on**.

Figure 6-9. Connecting MINICUBE2 to Target System

16-pin target cable   MINICUBE2
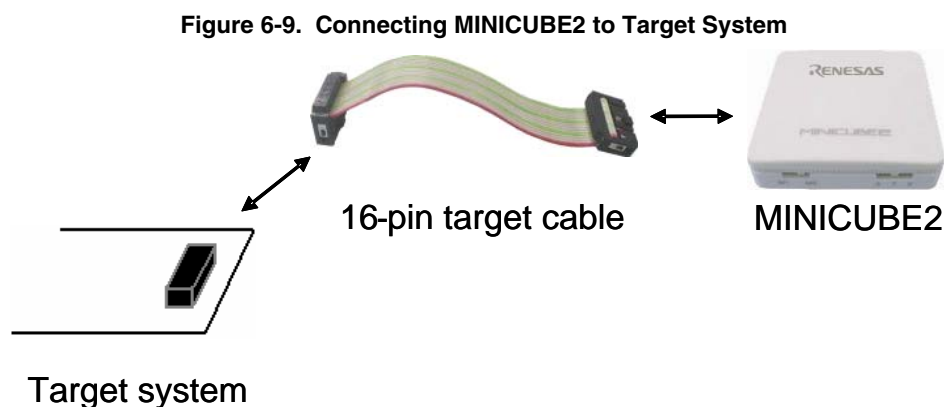
Target system

(4) Connecting the USB cable

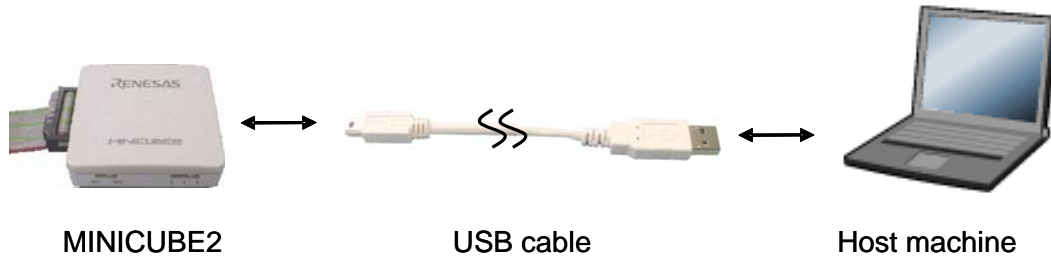Connect MINICUBE2 to the host machine as shown in Figure 6-10, **before the power to the target system is turned on**.

When the power select switch is set to ″T″, the mode LED **flashes** white after connection.

When the power select switch is set to ″3″ or ″5″, the mode LED **glows** white after connection.

Figure 6-10. Connecting MINICUBE2 to Host Machine

MINICUBE2　　　　　　　　　USB cable　　　　　　　　Host machine

(5) Applying power to target system

Turn on power to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″. The mode LED glows white after power application.

(6) Debugger startup

Start the debugger. The mode LED glows yellow after debugger startup.

For the operation after this step, refer to the user's manual for the debugger.

If the debugger does not start normally or the operation is unstable, the possible causes may be the following.

• Communication error between MINICUBE2 and target system

Whether communication is performed normally can be checked using the OCD Checker. For details, refer to the users manual for the OCD Checker.

• The user resource has not been secured or the security ID and on-chip debug option byte have not been set

To perform debugging with MINICUBE2, the debug monitor area must be secured and the security ID and on-chip debug option byte must be set. For details, refer to **6.2.5 Securing of user resources and setting of security ID and on-chip debug option byte**.

• Unsupported software (debugger, device file, or firmware) is used

The software used may not support debugging of the target device. Refer to the document ″QB-MINI2 Operating Precautions″ and check the supported version. When using software provided by a Renesas Electronics partner, refer to the documents prepared by the partner company.

• Defect of MINICUBE2

MINICUBE2 may have a defect. Refer to **CHAPTER 7 SELF-TESTING**.

6.2.4 System shutdown procedure

Terminate debugging and shutdown the system in the following order.

If the following order is not observed, the target system or MINICUBE2 may be damaged.

(1) Debugger termination

Terminate the debugger.

(2) Target system power shutdown

Shutdown the power supplied to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″.

(3) Unplugging the USB cable

Unplug the USB cable from MINICUBE2 or the host machine.

(4) Unplugging the target cable

Unplug the target cable from MINICUBE2 or the target system.

6.2.5 Securing of user resources and setting of security ID and on-chip debug option byte

The user must prepare the following to perform communication between MINICUBE2 and the target device and implement each debug function. If Renesas Electronics assembler RA78K0R or compiler CC78K0R is used, the items can be set by using linker options. Refer to the descriptions on the following pages and set these items.
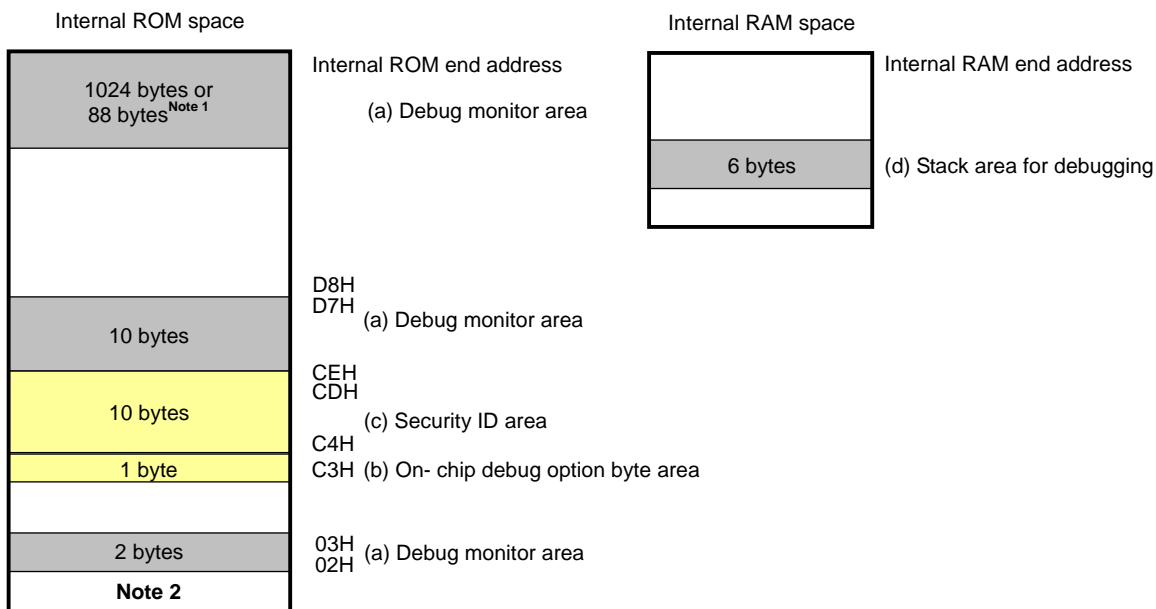
When C-SPY manufactured by IAR Systems is used, read also the following material.
- IAR C-SPY Hardware Debugger Systems User Guide issued by IAR Systems

Securement of memory space

The shaded portions in Figure 6-11 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. These spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Figure 6-11. Memory Spaces Where Debug Monitor Programs Are Allocated

Internal ROM space

1024 bytes or 88 bytes[Note 1]	Internal ROM end address
	(a) Debug monitor area
	D8H
10 bytes	D7H (a) Debug monitor area
	CEH
10 bytes	CDH (c) Security ID area
	C4H
1 byte	C3H (b) On- chip debug option byte area
2 bytes	03H (a) Debug monitor area
Note 2	02H

Internal RAM space

	Internal RAM end address
6 bytes	(d) Stack area for debugging

▭ : Area used for on- chip debugging

Note 1. When the pseudo RRM function is not used during 2-wire mode, it will be 88 bytes.
  2. In debugging, reset vector is rewritten to address allocated to a monitor program.

- Setting of security ID and on-chip debug option byte

  The security ID or on-chip debug option byte must be embedded in the area filled with yellow (0xC3 and 0xC4 to 0xCD) in Figure 6-11, to prevent the memory from being read by an unauthorized person.

(a) Securement of debug monitor area

This is the area to which the debug monitor program is to be allocated. The monitor program performs initialization processing for debug communication interface and RUN or break processing for the CPU. This user programs or data must not be placed in an area of 22 bytes near the on-chip debug option byte, and an area of 1,024 bytes[Note] before the internal ROM end address. In addition, reset vector is rewritten to address allocated to a monitor program.

Note It is an area of 88 bytes when the pseudo RRM function is not used during debugging in 2-wire mode. If the internal ROM end address is 0x3FFFF, a monitor program of 88 bytes is allocated to the area from 0x3FFA8 to 0x3FFFF.

[How to secure areas]

The following shows an example for securing the area when using Renesas Electronics assembler RA78K0R or compiler CC78K0R. Select the ″On-Chip Debug Option Bytes [-go]″ check box as shown in the figure below. An area of 22 bytes is thus secured near the on-chip debug option byte. An area of 1,024 bytes before the internal ROM end address is also secured by setting the start address and size.

Example Internal ROM end address is 0x3FFFF

Figure 6-12. Securement of Debug Monitor Area
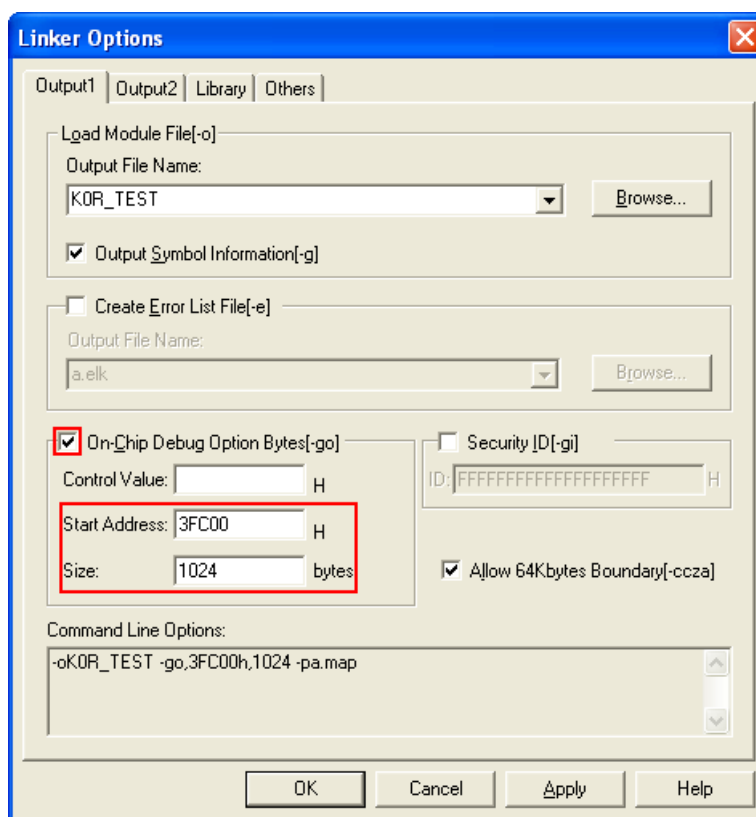
(b) On-chip debug option byte area

This is the area for the security setting to prevent the flash memory from being read by an unauthorized person. The debugger manipulates the target device in accordance with the set value, as shown below.

Table 6-6. On-Chip Debug Option Byte Setting and Operation

Set Value	Description	Remark
0x04	Debugging is disabled even if an on-chip debug emulator (such as MINICUBE2) is connected.	This setting is available only for flash programming and self programming.
0x85	The on-chip flash memory is not erased no matter how many times the security ID code authentication fails.	-
0x84	All on-chip flash memory areas are erased if the security ID code authentication fails.	-
Other than above	Setting prohibited	-

Caution   **Values can be set only to bits 7 and 0 (OCDENSET and OCDERSD). Be sure to write 000010B to bits 6 to 1.**

Remark   Values set to bits 3 to 1 are changed during on-chip debugging, so values of these bits are undefined after setting. However, be sure to set these bits to initial values (0, 1, 0).

[How to set]

The following shows a setting example when using Renesas Electronics assembler RA78K0R or compiler CC78K0R. Set a value for the Control Value text box under the ″On-Chip Debug Option Bytes [-go]″ check box as shown in the figure below.

Example Setting 0x85 for control value

Figure 6-13. On-Chip Debug Option Byte Setting Example
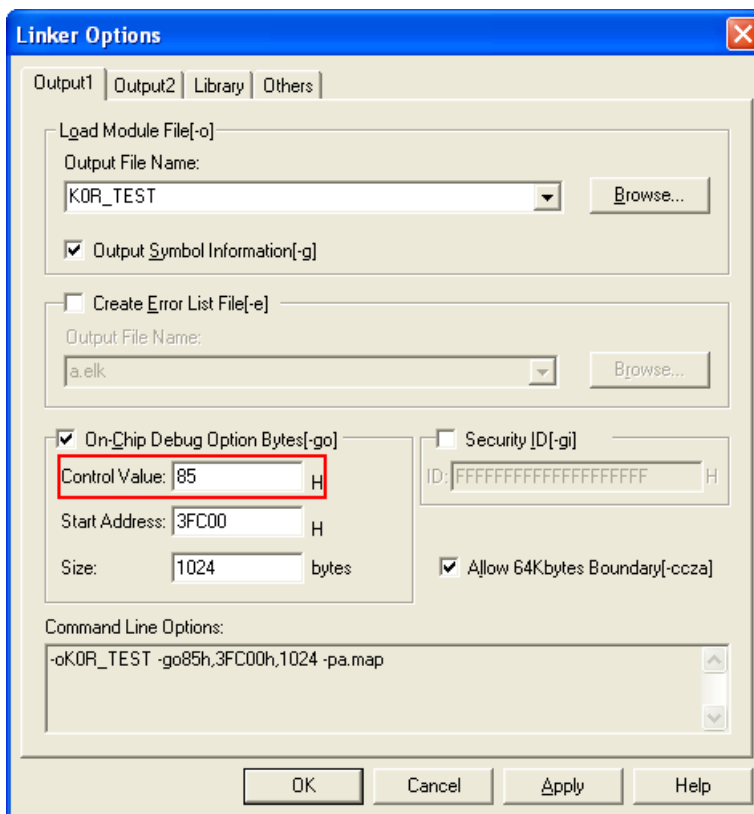
Caution   **If bit 7 (OCDENSET) of the on-chip debug option byte area (address 0xC3) is set to ″0″ by flash programming or self programming, debugging is disabled and the debugger can no longer start unconditionally. To enable debugging, erase the flash memory by flash programming.**
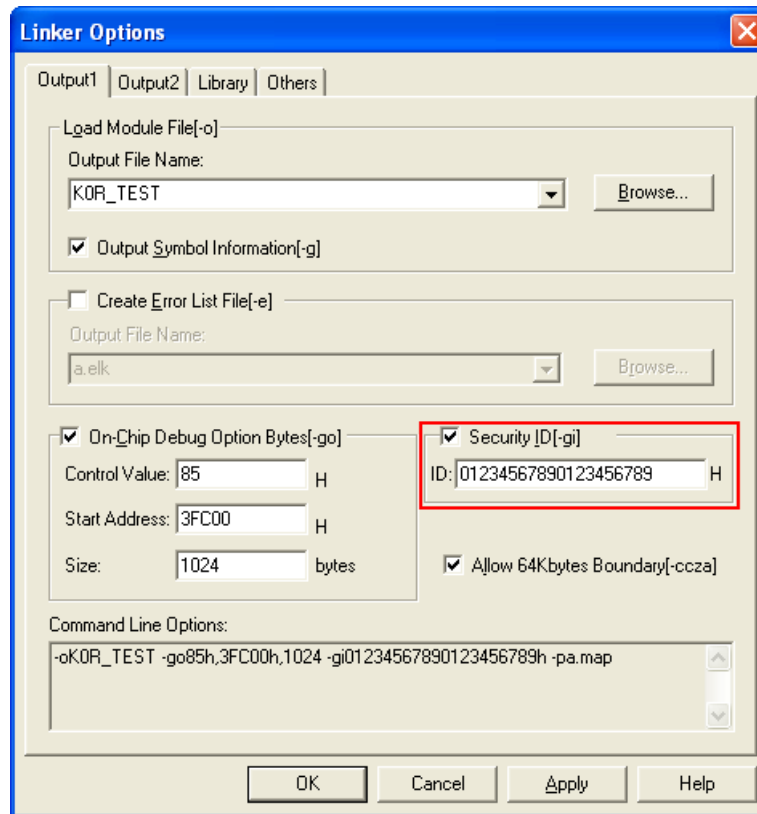
(c) Security ID area

This is the area for the security setting to prevent the flash memory from being read by an unauthorized person. The security ID functions as a password for starting the debugger. The debugger starts only when the security ID that is input during debugger startup and the security ID embedded in this area match. If the ID codes do not match, the debugger manipulates the target device in accordance with the value set to the on-chip debug option byte area (refer to **Table 6-6**).

[How to set]

An ID code is generated in the security ID area by setting the security ID when using Renesas Electronics assembler RA78K0R or compiler CC78K0R. Select the ″Security ID [-gi]″ check box and set an arbitrary ID code (10 bytes) as shown in the figure below.

Figure 6-14. Security ID Setting Example

Caution   If you have forgotten the security ID, erase the flash memory by flash programming or self programming and then set the security ID again.

(d) Securement of stack area for debugging

This area requires 6 bytes as the stack area for debugging. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 6 extra bytes are consumed for the stack area used.

Figure 6-15 illustrates the case where the stack area is increased when the internal high-speed RAM starts from 0xFCF00.

Figure 6-15. Variation of Address of Stack Area for Debugging
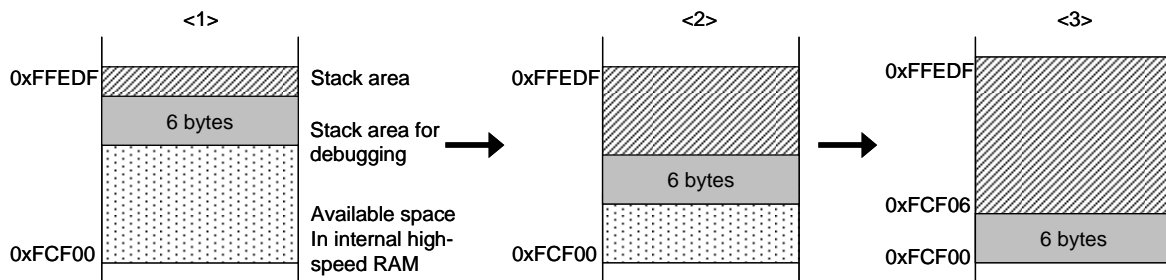
[How to secure areas]

Set the stack pointer by estimating the stack area consumed by the user program + 6 bytes. Make sure that the stack pointer does not extend beyond the internal high-speed RAM start address.

Remark   Refer to the self programming manual for how to secure the stack area for self programming.

6.2.6 Cautions on debugging

This section describes cautions on performing on-chip debugging for a 78K0R microcontroller.

Be sure to read the following to use MINICUBE2 properly.

(1) Handling of device that was used for debugging

Do not mount a device that was used for debugging on a mass-produced product, because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed. Moreover, do not embed the debug monitor program into mass-produced products.

(2) Flash self programming

If a space where the debug monitor program is allocated is rewritten by flash self programming, the debugger can no longer operate normally. This caution also applies to boot swapping for such an area.

(3) Operation after reset

After an external pin reset or internal reset, the monitor program performs debug initialization processing. Consequently, the time from reset occurrence until user program execution differs from that in the actual device operation. If Not Permit is selected in the Flash Programming area in the Configuration dialog box of the debugger, the time until the user program is executed compared with the time when Permit is selected is delayed several 100 ms.

(4) Debugging with real machine running without using MINICUBE2

If debugging is performed with a real machine running, without using MINICUBE2, write the user program using the QB-Programmer. Programs downloaded by the debugger include the monitor program, and such a program malfunctions if it includes processing to make the TOOL0 pin low level.

(5) Operation when debugger starts

When the debugger is started, if the Target Device Connection setting in the Configuration dialog box of the debugger is different from the setting for the previous debugging, the internal flash memory is erased.

(6) Debugging after program is written by flash programming

If a program is written to the internal flash memory using the QB-Programmer or PG-FP4, on-chip debugging is disabled even if it is enabled in the on-chip debugging option byte setting. To perform debugging of the target device after that, erase the internal flash memory using the QB-Programmer or PG-FP4 and then download the program using the debugger.

<R>   When ID78K0R V3.61 or later, or CubeSuite V1.31 or later is used, erase internal flash ROM memory automatically and download the program to the memory area by the debugger.

For the method of the other debuggers made by partner manufacturers, see the document related to each product.

(7) LVI default start function setting (address C1H)

During debugging, the debug monitor program stops the LVI default start function at address C1H. Consequently, the LVI default start function is kept stopped even after debugging is completed, unless the setting to address C1H is changed through flash programming.

(8) On-chip debugging option byte setting (address C3H)

The on-chip debugging option byte setting is rewritten arbitrarily by the debugger.

(9) FLMD0 pin output status while debugger is running

In accordance with the setting in the Flash Programming area in the Configuration dialog box of the debugger, the FLMD0 pin output status while the debugger is running changes as follows. Rewriting by flash self-programming is not possible when the output status is low level.

- When Permit is selected: High level (low level for about 100 μs after reset release)
- When Not Permit is selected: Low level

(10) Operation at voltage with which flash memory cannot be written

If any of the following debugger operations <1> to <7>, which involve flash memory rewriting, is performed while flash memory cannot be rewritten, the debugger automatically changes the register setting so as to enable flash memory rewriting, and restores the register setting after the operation is completed. If any of the following operations <1> to <7> is performed while flash memory rewriting has been disabled or operation is performed at a voltage with which flash memory cannot be rewritten, however, the debugger outputs an error and the operation is ignored.

To prevent the flash memory from being rewritten, select "Not Permit" in the Flash Programming area in the ID78K0R-QB Configuration dialog box. To prevent the frequency from being switched automatically, select "User" in the Monitor Clock area in the Flash Programming area in the ID78K0R-QB Configuration dialog box.

<1> Writing to internal flash memory

<2> Setting or canceling of software breakpoint

<3> Starting execution at the set software breakpoint position

<4> Step execution at the set software breakpoint position

<5> Step-over execution, Return Out execution

<6> Come Here

<7> If Permit is selected in the Flash Programming area in the Configuration dialog box, the following operations cannot be performed.

     a) Setting, changing, or canceling of hardware breaks

     b) Masking/unmasking of internal reset

     c) Switching of peripheral breaks

(11) Debugging in 1-wire mode

In the condition that debugging is performed in 1-wire mode (selected by choosing TOOL0 in the Target Device Connection area in the Configuration dialog box of the debugger), when the internal high-speed oscillator is used for the CPU operating clock, breaks may not occur normally if the frequency variation between debugger startup and break occurrence (except for when changing the register) is too large. This situation may occur when the variation of operating voltage or temperature is too large.

(12) Pseudo real-time monitor function

Note the following points when using the pseudo real-time monitor function.

<1> Standby mode (HALT or STOP) may be cancelled during monitoring.

<2> The pseudo real-time monitor function does not operate while the CPU operating clock is stopped.

<R>   <3> If the targets to be monitored are too numerous, the operability of the debugger may be affected because the monitoring speed is slow when using the pseudo RRM function in 1-wire mode. When using the ID78K0R-QB, therefore, monitoring by using the Watch window, rather than the Memory window, is recommended.

6.3 Flash Programming

This section describes the system configuration and startup/shutdown procedure when flash programming is performed for a 78K0R microcontroller, using MINICUBE2.

6.3.1 Specifications of programming function

Table 6-7. Specifications of Programming Function

Functions	Specifications
Host interface	USB 2.0 (compatible with 1.1)
Target interface	UART (1-wire mode)
Target system voltage	1.65 to 5.5 V (depends on the target device)
Clock supply	Internal high-speed oscillation clock is used
Power supply	3 or 5 V (maximum current rating: 100 mA)
Acquisition of device-specific information	Parameter file for MINICUBE2 made by Renesas Electronics is used
Security flag setting	Available
Standalone operation	Unavailable (must be connected to host machine)

\<R\> (rows: Target system voltage; Acquisition of device-specific information)

6.3.2 System configuration

Figure 6-16 illustrates the system configuration for flash programming.

Figure 6-16. System Configuration for Flash Programming
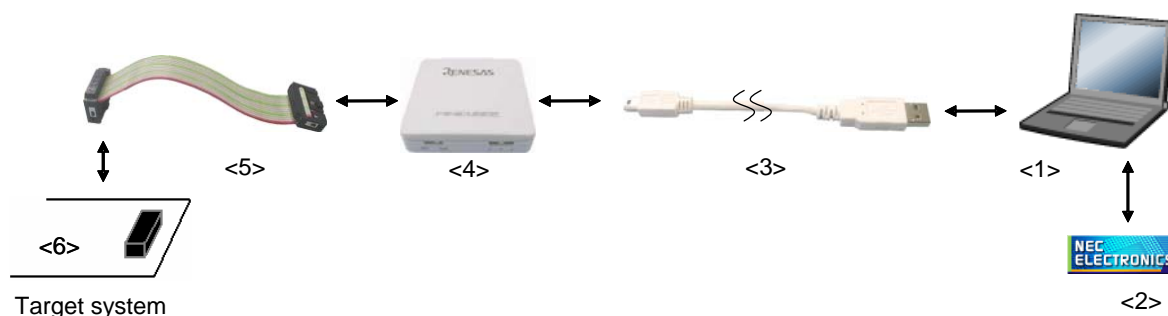
Target system

<1> Host machine
    Products with USB ports
<2> Software
    Includes the QB-Programmer USB driver, parameter files, and so on.
    Download these Renesas Electronics software from the following website (ODS).
        URL: http://www2.renesas.com/micro/en/ods/index.html
<3> USB cable (accessory)
<4> MINICUBE2 (this product)
<5> 16-pin target cable (accessory)
<6> Target connector (sold separately)

6.3.3 System startup procedure

This section describes the system startup procedure. Observe the following order.

(1) Preparation and installation of software

The following software is required to perform flash programming. For details on preparation and installation of software, refer to the setup manual supplied with MINICUBE2.

- QB-Programmer
- USB driver
- Parameter file

(2) Switch setting

Set the mode select switch to ″M1″. Set the power select switch by referring to the following table.

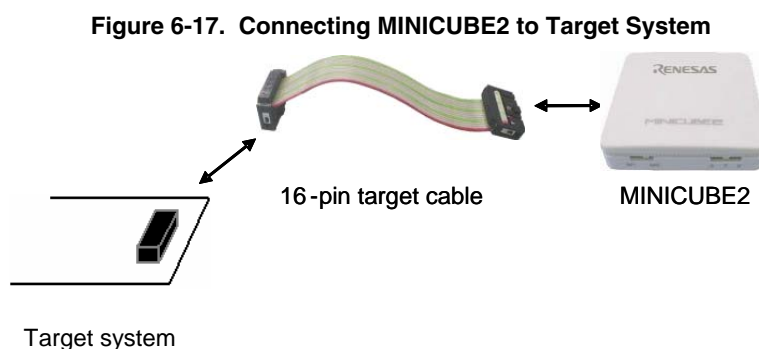**Caution   Do not change the switch setting while the USB cable is connected.**

Table 6-8. Setting of Power Select Switch

Setting	Description
3	3 V is supplied from MINICUBE2 to the target system[Note].
5	5 V is supplied from MINICUBE2 to the target system[Note].
T	Power supply of the target system is used. MINICUBE2 detects the power for the target system. The detected power is also used as the power supply for the communication interface.

Note The maximum rating of the current is 100 mA, so do not use MINICUBE2 with the target system with the higher current rating. The power is always supplied after MINICUBE2 is connected to the host machine.

(3) Connecting the target system

Connect MINICUBE2 to the target system as shown in Figure 6-17, **before the power to the target system is turned on**. When the power select switch is set to ″T″, the step of **(4) Connecting the USB cable** can be performed before this step.

Figure 6-17. Connecting MINICUBE2 to Target System

16 -pin target cable   MINICUBE2

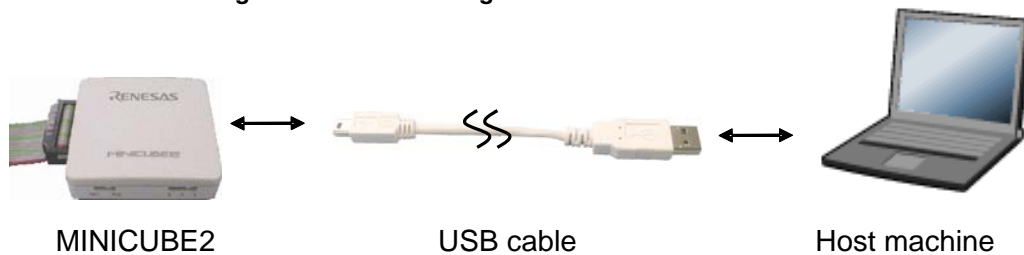Target system

(4) Connecting the USB cable

Connect MINICUBE2 to the host machine as shown in Figure 6-18, **before the power to the target system is turned on**.

When the power select switch is set to ″T″, the mode LED **flashes** white after connection.

When the power select switch is set to ″3″ or ″5″, the mode LED **glows** white after connection. The power is always supplied after MINICUBE2 is connected to the host machine with the USB cable.

Caution   Do not change the switch setting while the USB cable is connected.

Figure 6-18. Connecting MINICUBE2 to Host Machine

MINICUBE2   USB cable   Host machine

(5) Applying power to target system

Turn on power to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″. The mode LED glows white after power application.

(6) QB-Programmer startup

Start the QB-Programmer from the Start menu of Windows. The mode LED glows green after QB-Programmer startup.

A series of basic operations after this step is described in the following section. For detailed usage of the QB-Programmer, refer to the **QB-Programmer Programming GUI Operation User's Manual**.

6.3.4 Usage examples

This section describes a series of basic operations using the QB-Programmer, using the μPD78F1166 as an example. The operations from execution of the [Autoprocedure (EPV)] command to programming for the target device are described.   For other commands and their applications, refer to the **QB-Programmer Programming GUI Operation User's Manual**.

The conditions used for a series of operations in this section are as follows.

<Target system>
Target device:	μPD78F1166
Power supply voltage:	5 V (supplied from MINICUBE2)
Communication channel:	UART, 1 Mbps

<MINICUBE2>
Switches:	Mode select switch: M1
	Power select switch: 5

<QB-Programmer>
|	
Parameter file:	78F1166.PRM
Clock setting:	None (internal high-speed oscillation clock is used)
Operation mode:	chip
Program file:	sample.hex
Command options:	[Blank check before Erase] selected
	[Read verify after Program] selected

(1) Setting of programming environment

Set the programming environment, following the steps <1> to <6> described below.

<1> Click the [Device] menu on the menu bar and then click [Setup...].
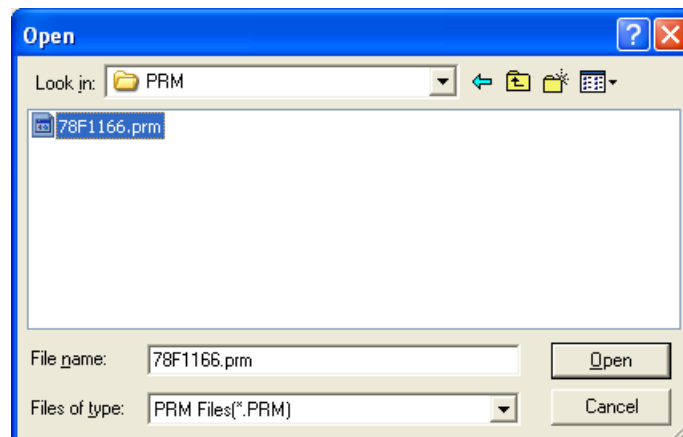
<2> The Device Setup dialog box appears, in which the [Standard] tab is active.

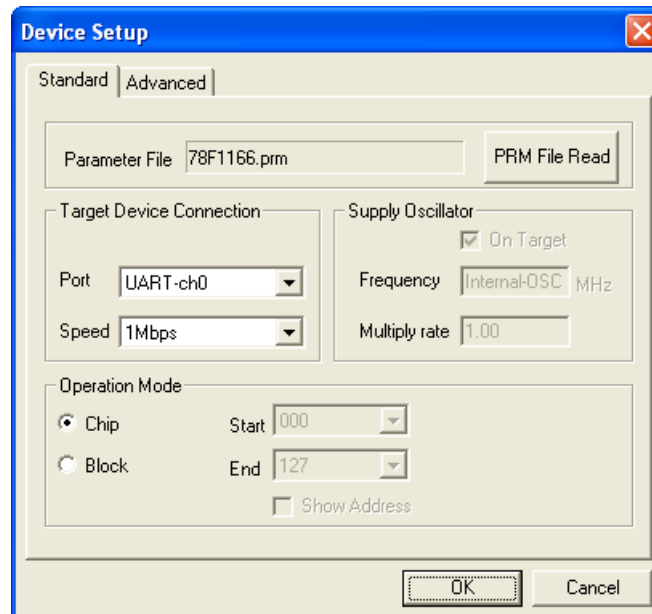**Figure 6-19.　[Standard] Tab of Device Setup Dialog Box**
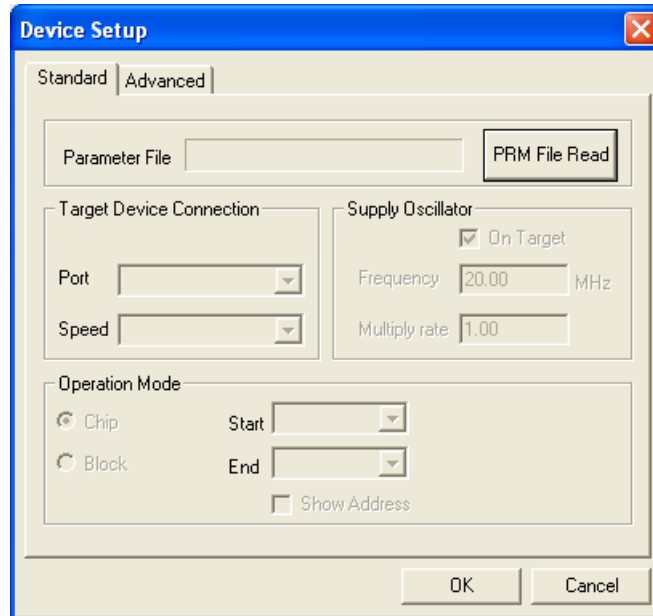
<3> Click the PRM File Read button to open the Parameter File Select dialog box.　Select the parameter file for the target device and then click the Open button.

Figure 6-20.　Parameter File Select Dialog Box

<4> Set the items in the [Target Device Connection], [Operation Mode] and [Supply Oscillator] areas, in accordance with your programming environment. The following figure shows an example.

**Figure 6-21. [Standard] Tab of Device Setup Dialog Box After Setting**

<5> Next, set the items in the [Command options] and [Security flag settings] areas, in accordance with your programming environment. The following figure shows an example.

Figure 6-22. [Advanced] Tab of Device Setup Dialog Box
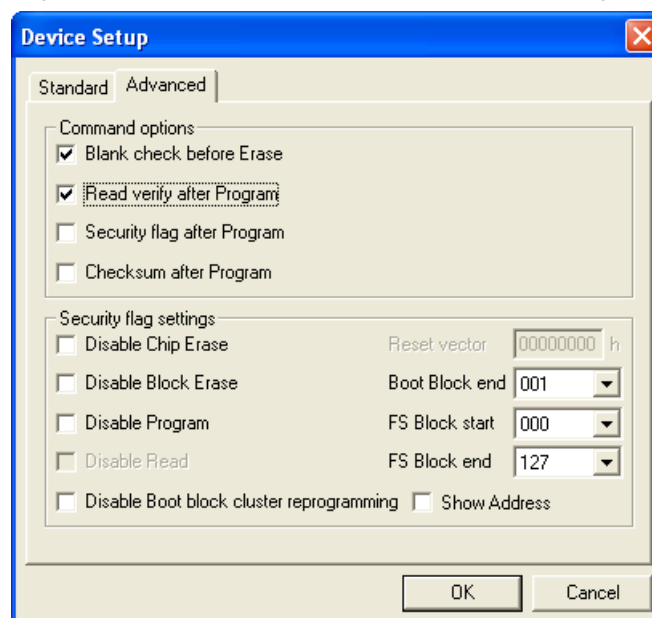
<6> By clicking the OK button, the programming environment is set and the Device Setup dialog box is closed. The main window appears as follows.

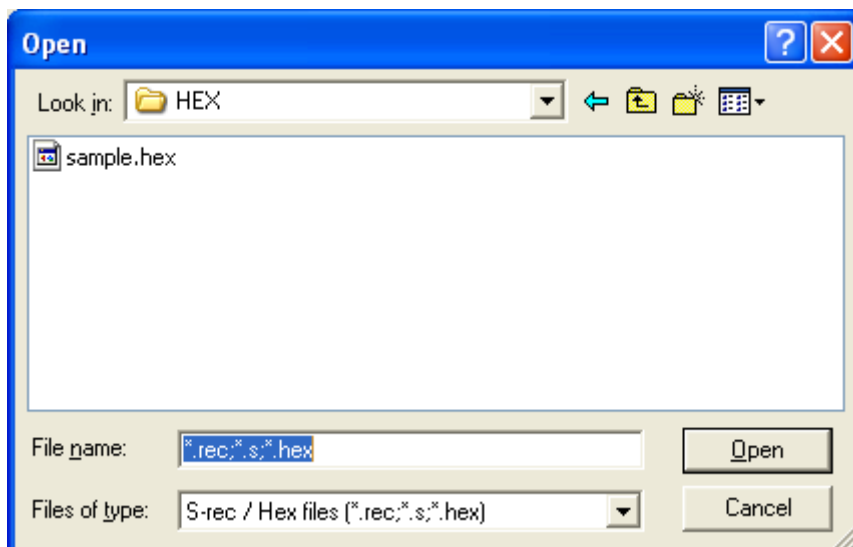**Figure 6-23. Completion of Programming Environment Setting**
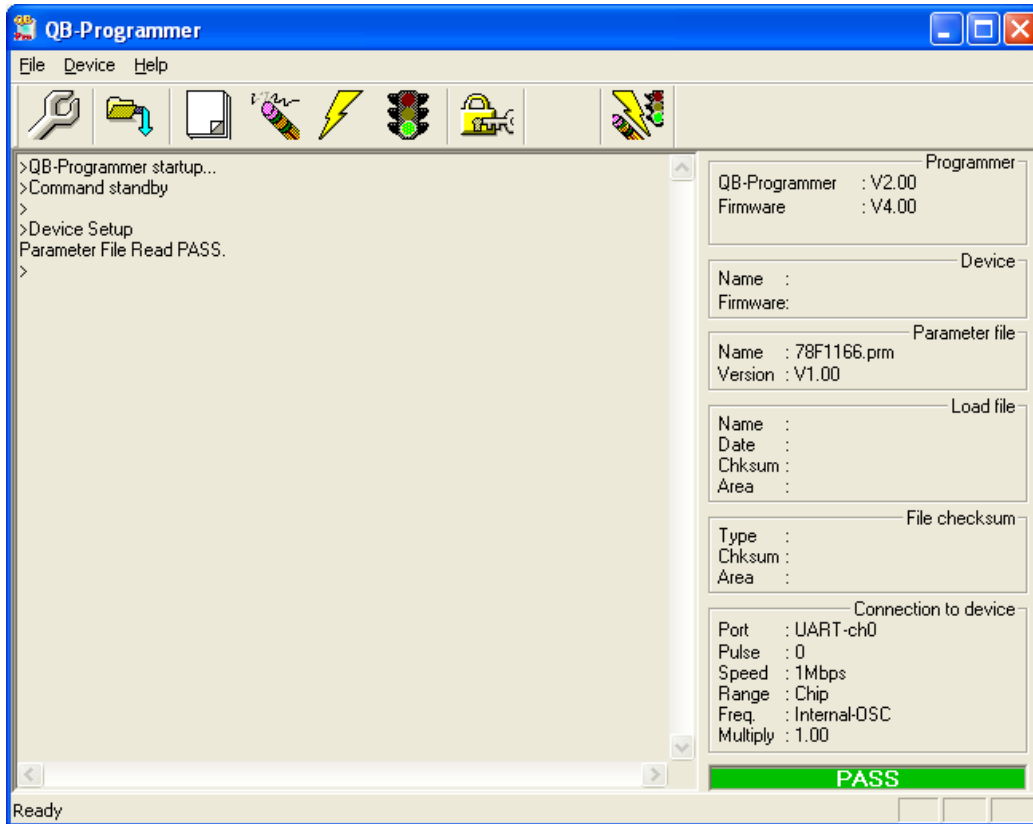
(2) Selection of program file

Select the program file, following the steps <1> to <3> described below.

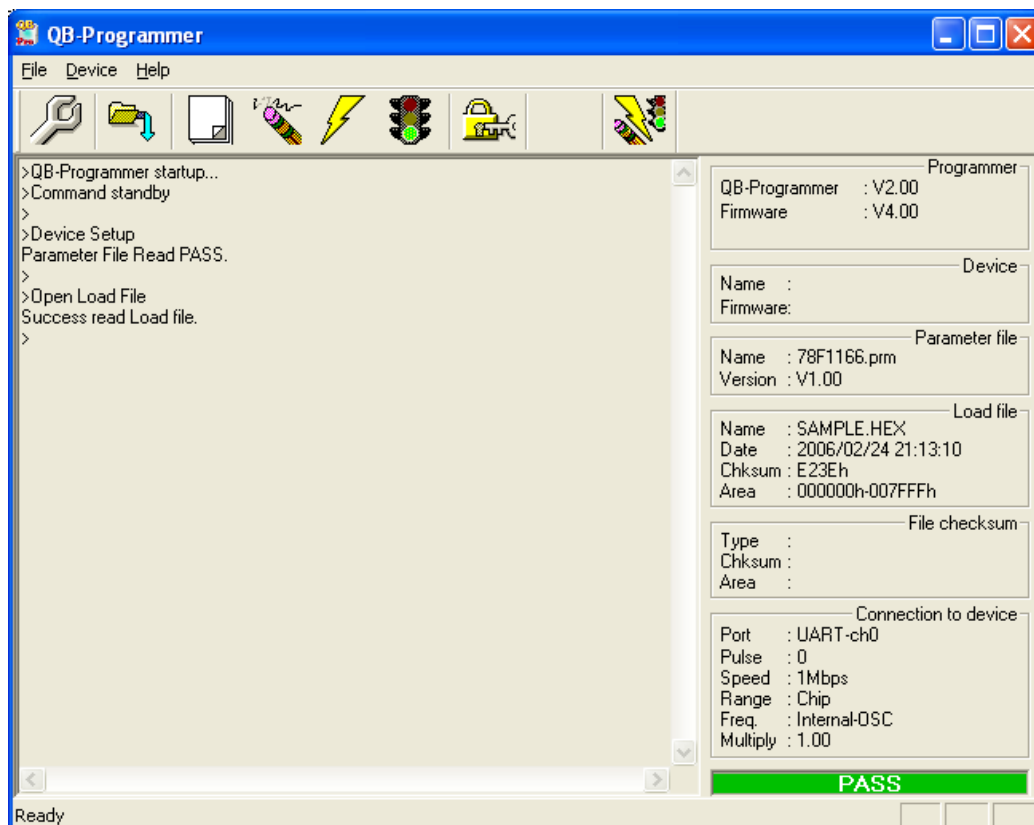<1> Click the [File] menu on the menu bar and then click [Load...].

<2> The program file select dialog box appears.

Figure 6-24. Program File Select Dialog Box

<3> Select the program file for the target device and then click the Open button.

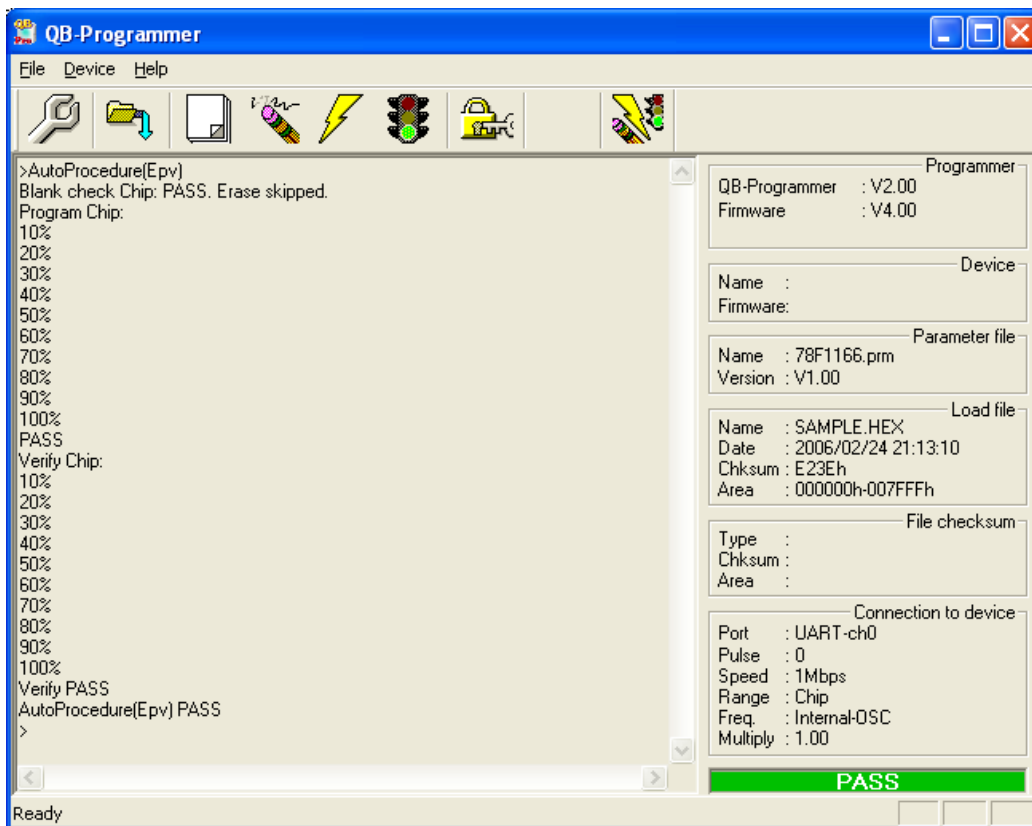**Figure 6-25. Completion of Program File Selection**

(3) Execution of [Autoprocedure (EPV)] command

Click the [Device] menu on the menu bar and then click [Autoprocedure (EPV)].

The [Blank Check], [Erase] (if the flash memory in the target device is not blank), [Program], and [Verify] commands are executed for the target device sequentially. The mode LED on MINICUBE2 flashes yellow during execution. When the command execution is completed normally, the mode LED glows green and the following window appears.

Figure 6-26. Completion of [Autoprocedure (EPV)] Command Execution

6.3.5 System shutdown procedure

Terminate flash programming and shutdown the system in the following order.

If the following order is not observed, the target system or MINICUBE2 may be damaged.

(1) Terminating the QB-Programmer

Terminate the QB-Programmer if you are not going to perform programming to other devices.

Remark   When the QB-Programmer is terminated, the setting information is saved in the *qbp.ini* file. Due to this file, the same setting is maintained when the QB-Programmer is started the next time.

(2) Target system power shutdown

Shutdown the power supplied to the target system. This step is not necessary if the power select switch is set to ″3″ or ″5″.

(3) Unplugging the USB cable

Unplug the USB cable from MINICUBE2 or the host machine.

When the power select switch is set to ″T″, the step of **(4) Unplugging the target cable** can be performed before this step.

(4) Unplugging the target cable

Unplug the target cable from MINICUBE2 or the target system.

6.3.6 Cautions on flash programming

This section describes the cautions for flash programming. Be sure to read the following for the proper use of MINICUBE2.

- To improve the writing quality, fully understand, verify, and evaluate the following items before using MINICUBE2.
    - Circuits are designed as described in the user's manuals for the device and MINICUBE2.
    - The device, QB-Programmer and MINICUBE2 are used as described in each user's manual.
   - The power supplied to the target system is stable.

CHAPTER 7   SELF-TESTING

This section describes the method to perform self-testing with MINICUBE2.

The MINICUBE2 diagnostic tool can be used to find if the reason that the debugger does not operate normally derives from a MINICUBE2 defect or from other hardware.

7.1 System Configuration

Figure 7-1 illustrates the system configuration for self-testing.
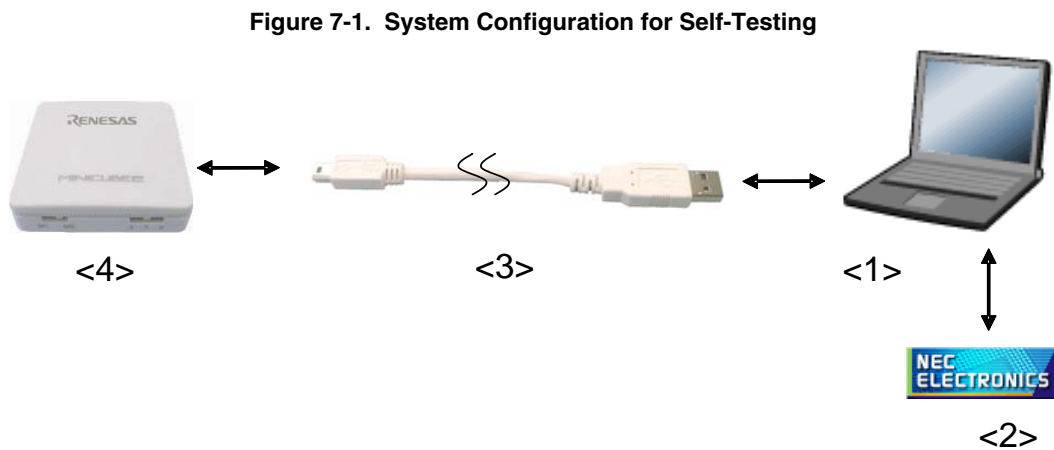
Figure 7-1. System Configuration for Self-Testing

<1>   Host machine
      Products with USB ports
<2>   Software
      Includes the USB driver and MINICUBE2 diagnostic tool.
      These software can be installed when a Renesas Electronics debugger or QB-Programmer is installed.
      Download these Renesas Electronics software from the following website (ODS).
            URL: http://www2.renesas.com/micro/en/ods/index.html
<3>   USB cable (accessory)
<4>   MINICUBE2 (this product)

7.2 Self-Testing Procedure

(1) Preparation and installation of software

The following software is required to perform self-testing. For details on preparation and installation of software, refer to the setup manual supplied with MINICUBE2.

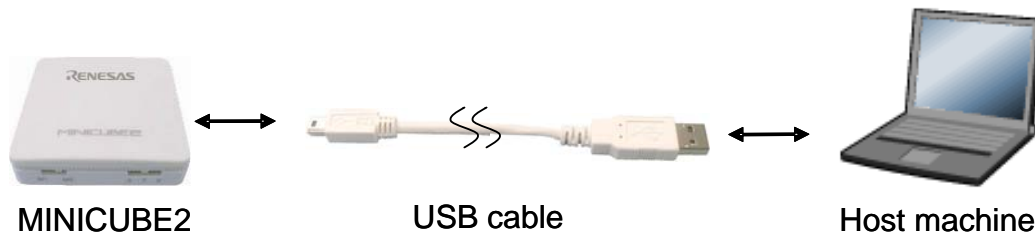- MINICUBE2 diagnostic tool
- USB driver

(2) Switch setting

The mode select switch can be set to ″M1″ or ″M2″.

Set the power select switch to ″3″ or ″5″.

(3) Connecting the USB cable

Connect MINICUBE2 to the host machine. **Do not connect MINICUBE2 to the target system**. The mode LED glows white after connection.

Figure 7-2. Connecting MINICUBE2 to Host Machine

MINICUBE2   USB cable   Host machine

(4) Startup of MINICUBE2 diagnostic tool

Start the MINICUBE2 diagnostic tool.

If the MINICUBE2 diagnostic tool was installed when Renesas Electronics software was installed, the diagnostic tool is registered on the Start menu as follows.

- ″NEC Electronics″→″MINICUBE Utilities″→″Vx.xx (version)″

For the operation after this step, refer to the user's manual for the MINICUBE2 diagnostic tool.

CHAPTER 8   FIRMWARE UPDATE

Firmware is a program embedded in the device for controlling MINICUBE2. The following can be implemented by updating firmware.

- Addition of supported devices
- Correction of restrictions

This chapter describes the method for checking firmware version, system configuration, and firmware update procedure. Note that MINICUBE2 may no longer operate if firmware is updated incorrectly.

8.1 System Configuration

Figure 8-1 illustrates the system configuration for updating firmware.
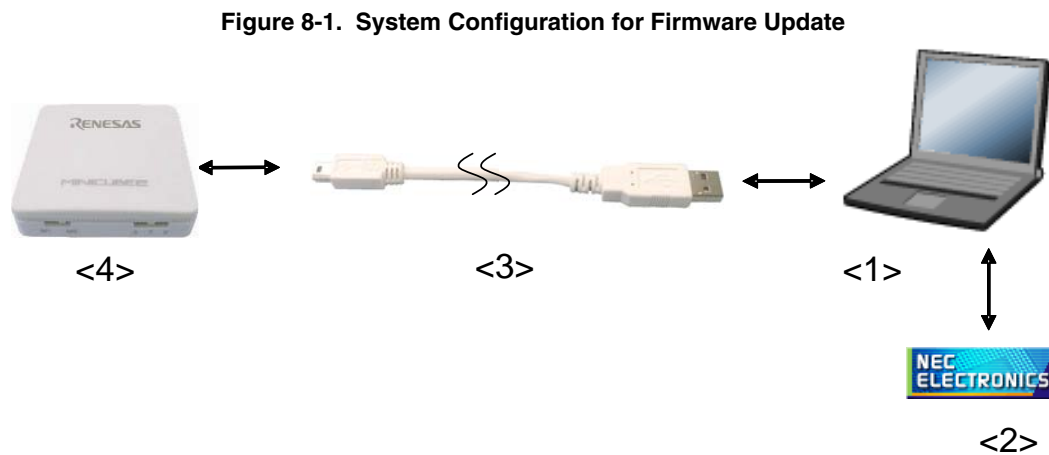
Figure 8-1. System Configuration for Firmware Update

<1> Host machine
     Products with USB ports
<2> Software
     Includes the USB driver and MINICUBE2 diagnostic tool.
     These software can be installed when a Renesas Electronics debugger or QB-Programmer is installed.
     Download these Renesas Electronics software from the following website (ODS).
          URL: http://www2.renesas.com/micro/en/ods/index.html
<3> USB cable (accessory)
<4> MINICUBE2 (this product)

8.2 Firmware Update Procedure

(1) Preparation and installation of software

The following software is required to update firmware. For details on preparation and installation of software, refer to the setup manual supplied with MINICUBE2.

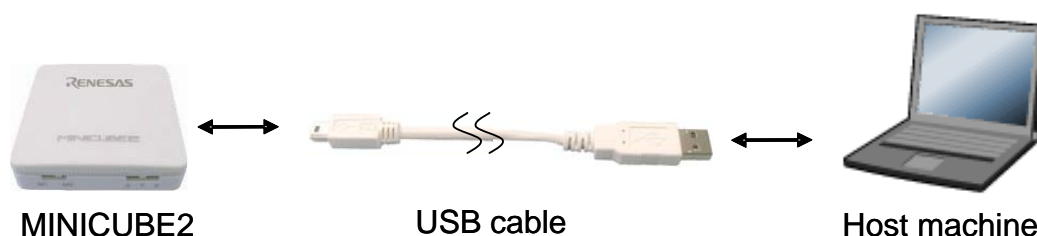- MINICUBE2 diagnostic tool
- USB driver

(2) Switch setting

The mode select switch and power select switch may be set to any position.

(3) Connecting the USB cable

Connect MINICUBE2 to the host machine. **Do not connect MINICUBE2 to the target system**. The mode LED glows white after connection.

Figure 8-2. Connecting MINICUBE2 to Host Machine

MINICUBE2   USB cable   Host machine

(4) Startup of MINICUBE2 diagnostic tool

Start the MINICUBE2 diagnostic tool.

If the MINICUBE2 diagnostic tool was installed when Renesas Electronics software was installed, the diagnostic tool is registered on the Start menu as follows.

• ″NEC Electronics″→″MINICUBE Utilities″→″Vx.xx (version)″

For the operation after this step, refer to the user's manual for the MINICUBE2 diagnostic tool.

APPENDIX A EQUIVALENT CIRCUIT

The equivalent circuit in MINICUBE2, which is related to the communication interface between MINICUBE2 and the target system, is shown below. Although an example of the target system circuit connections is presented in this document, refer to it when determining the parameters during substrate designing.

Figure A-1. MINICUBE2 Equivalent Circuit
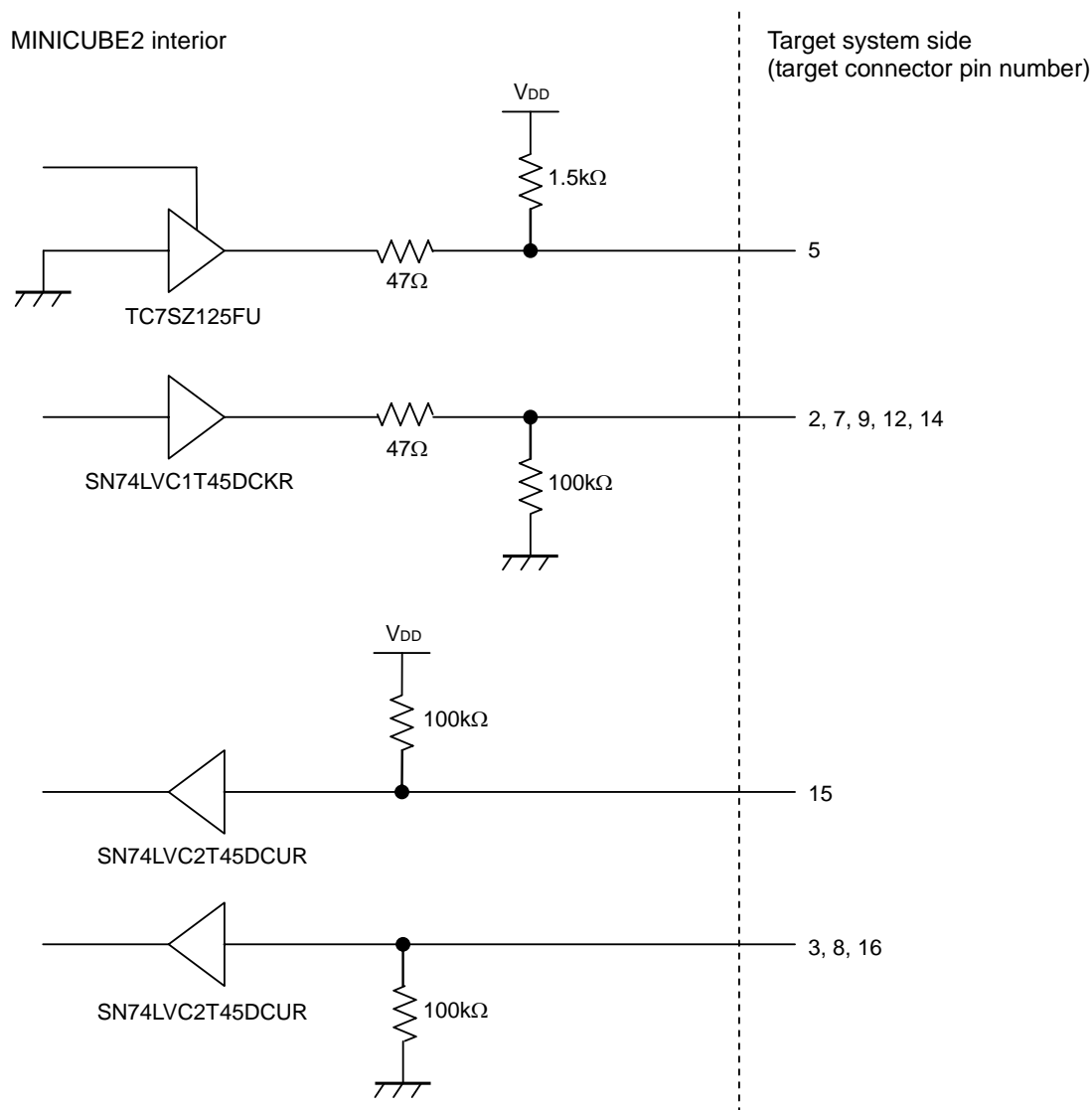
Remark With 78K0S and 78K0R microcontrollers, pins 3 and 5 are shorted out. During on-chip debugging of 78K0 microcontrollers, the following routing is performed when connecting the 78K0-OCD board.
- Pins 3 and 5 are shorted out and assigned to the target system side as DATA pins (pins 13).
- Pins 12 and 14 are shorted out and assigned to the target system side as FLMD pins.

<R>

REVISION HISTORY

Revisions up to the previous edition are shown below. The "Applied to" column indicates the chapter in each edition to which the revision was applied.

Edition	Description	Applied to
6th edition	Change of **URL**	**GENERAL**
	Change of **Table 1-2. Hardware Specifications**	**CHAPTER 1 GENERAL**
	Change of **Table 3-2. Pin Functions** Addition of Caution in **Figure 3-2. Recommended Circuit Connection When UART Is Used for Communication Interface** Addition of Caution in **Figure 3-3. Recommended Circuit Connection When CSI-H/S Is Used for Communication Interface** Change of **Table 3-6. Specifications of Programming Function**	**CHAPTER 3 HOW TO USE MINICUBE2 WITH V850 MICROCONTROLLER**
	Change of **Table 4-1. Pin Assignment When Using 16-Pin Target Cable** Change of **Table 4-2. Pin Functions** Change of **Table 4-4 Pin Functions** Change of **Figure 4-32. Reserved Area Used by MINICUBE2** Change of **4.2.6 (a) Debug monitor area (this area must be secured), (f) Pseudo real-time RAM monitor (RRM) area** Change of **4.2.7 (15) Cautions when debugging with only the real device (without using MINICUBE2), (16) Cautions when reading the location and the reserved area where SW/HW breaks are set**	**CHAPTER 4 HOW TO USE MINICUBE2 WITH 78K0 MICROCONTROLLER**
	Change of **Table 5-5. Specifications of Programming Function**	**CHAPTER 5 HOW TO USE MINICUBE2 WITH V850 MICROCONTROLLER**
	Change of **Table 6-1. Differences Between 1-Wire Mode and 2-Wire Mode** Change of **6.1.2 Circuit connection example** Change of **6.2.6 Cautions on debugging** Change of **Table 6-7. Specifications of Programming Function**	**CHAPTER 6 HOW TO USE MINICUBE2 WITH 78K0R MICROCONTROLLER**
	Change of **Remark**	**APPENDIX A EQUIVALENT CIRCUIT**
5th edition	Change of **Figure 3-2. Recommended Circuit Connection When UART Is Used for Communication Interface** Change of **Figure 3-3. Recommended Circuit Connection When CSI-H/S Is Used for Communication Interface**	**CHAPTER 3 HOW TO USE MINICUBE2 WITH V850 MICROCONTROLLER**

Edition	Description	Applied to
5th edition	Addition of **Caution** Change of **4.1 Target System Design** Addition of **Note** in **Figure 4-1. Outline of Communication Interface for On-Chip Debugging** Change of **Note** in **Figure 4-2. Outline of Communication Interface for Flash Programming** Change of **Table 4-1. Pin Assignment When Using 16-Pin Target Cable** Change of **Table 4-2. Pin Functions** Change of **Table 4-3. Pin Assignment When Using 10-Pin Target Cable** Change of **Table 4-4. Pin Functions** Change of **Table 4-5. Corresponding Circuit Connection Diagrams** Addition of **Figure 4-3. Flow Chart for Selecting the Circuit** Change of **Figure 4-12. When Only Debugging Is Performed (with OCD1A/OCD1B communication)** to **Figure 4-4. When Both Debugging and Programming Are Performed (Not communicating by using the OCD0A and OCD0B pins or not using the X1 oscillator)** Addition of **Caution** in **4.1.3 Connection of reset pin** Addition of **Figure 4-14. Cautions for using the RESET-OUT pin** Change of **4.1.3 (2) Automatically switching the reset signal via selector logic** Change of Description in **4.1.3 (4) Resetting the target device by power-on clear (POC) only** Change of Description in **4.1.4 Cautions on target system design** Change of **Table 4-6. Debug Functions** Addition of **Note** in **Figure 4-24. System Configuration for On-Chip Debugging** Addition of **Caution** in **Figure 4-25. Connecting MINICUBE2 to Target System** Change of **4.2.5 Clock setting** Change of **Figure 4-32. Reserved Area Used by MINICUBE2** Change of **4.2.6 (a) Debug monitor area (this area must be secured)** Change of **Table 4-8. Option Byte (84H) Settings and Descriptions** Addition of **Table 4-9. Option Byte (84H) Settings and Descriptions** Change of **4.2.6 (d) Stack area for debugging (this area must be secured)** Change of Description in **4.2.6 (f) Pseudo real-time RAM monitor (RRM) area** Change of Description in **4.2.7 (2) Overwriting flash memory during on-chip debugging** Change of **Caution** in **4.2.7 (9) Cautions when stepping into code**	**CHAPTER 4 HOW TO USE MINICUBE2 WITH 78K0 MICROCONTROLLER**

Edition	Description	Applied to
5th edition	Change of **Caution** in **4.2.7 (10) Cautions when using pseudo real-time RAM monitor function** Addition of Description in **4.2.7 (12) Emulation of POC function** Change of **Table 4-11. Specifications of Programming Function** Addition of **Remark** in **4.3.2 System configuration** Deletion of [Security] command in **4.3.4 (3) Execution of [Autoprocedure (EPV)] command**	**CHAPTER 4 HOW TO USE MINICUBE2 WITH 78K0 MICROCONTROLLER**
	Addition of **Note** in **Figure 5-2. Recommended Circuit Connection** Change of the [Security], [Verify] command in **5.3.4 (3) Execution of [Autoprocedure (EPV)] command**, addition of [Checksum] command	**CHAPTER 5 HOW TO USE MINICUBE2 WITH 78K0S MICROCONTROLLER**
	Change of **Table 6-1. Differences Between 1-Wire Mode and 2-Wire Mode** Change of **Note 2** in **Figure 6-2. Recommended Circuit Connection** Change of **Table 6-4. Debug Functions** Addition of Description of <3> in **6.2.6 (12) Pseudo real-time monitor function**	**CHAPTER 6 HOW TO USE MINICUBE2 WITH 78K0R MICROCONTROLLER**
4th edition	Change of **Figure 6-2** in **6.1.2 Circuit connection example**	**CHAPTER 6 HOW TO USE MINICUBE2 WITH 78K0R MICROCONTROLLER**
	Change of **Figure A-1** in **APPENDIX A EQUIVALENT CIRCUIT**	**APPENDIX A EQUIVALENT CIRCUIT**
3rd edition	Change of **Figures 3-2** and **3-3** in **3.1.2 Circuit connection example**	**CHAPTER 3 HOW TO USE MINICUBE2 WITH V850 MICROCONTROLLER**
	Change of **Figure 5-2** in **5.1.2 Circuit connection example**	**CHAPTER 5 HOW TO USE MINICUBE2 WITH 78K0S MICROCONTROLLER**
	Change of **Figure 6-2** in **6.1.2 Circuit connection example**	**CHAPTER 6 HOW TO USE MINICUBE2 WITH 78K0R MICROCONTROLLER**
2nd edition	Change of **Related Documents** in **INTRODUCTION**	**INTRODUCTION**
	1.3 Supported Devices • Deletion of **Remark** in **MINICUBE2 information site** • Addition of **Remark** in **Document**	**CHAPTER 1 OVERVIEW**
	Note 3 of **Figures 3-2** and **3-3** in **3.1.2 Circuit connection example** corrected, **Note 7** added	**CHAPTER 3 HOW TO USE MINICUBE2 WITH V850 MICROCONTROLLER**
	3.2.5 Securing of user resources and setting of security ID • Description of material when partner tool is used added • Addition of **Note** in **Figure 3-12** • Change of Assemble source in **(b) Securement of area for debug monitor program** • Change of **Note** in **(c) Securement of communication serial interface**	
	Addition of (12) Cautions unique to devices and (13) Processing of monitoring program for debugging in **3.2.6 Cautions on debugging**	

Edition	Description	Applied to
2nd edition	Change of **Figure 4-2** in **4.1 Target System Design** **4.1.2 Circuit connection examples** • Addition of **Note 2** in **Table 4-5** • Addition of **Note 7** in **Figure 4-3** and **4-4** • Addition of **Note 6** in **Figure 4-5** to **4-9** • Addition of **Note 5** in **Figure 4-10** and **4-11**	**CHAPTER 4 HOW TO USE MINICUBE2 WITH 78K0 MICROCONTROLLER**
	Change of **4.1.3 (4) Resetting the target device by power-on clear (POC) only**	
	Description added to **4.2.6 Securing of user resources and setting of security ID**	
	4.2.7 Cautions on debugging • Change of **(13) Emulation of POC function** • Addition of **(14) Device with/without on-chip debug function**	
	Note 2 of **Figures 5-2** in **5.1.2 Circuit connection example** corrected, **Note 6** added	**CHAPTER 5 HOW TO USE MINICUBE2 WITH 78K0S MICROCONTROLLER**
	Description added to **5.2.5 Securing of user resources**	
	Change of **Table 6-1** in **6.1 Target System Design**	**CHAPTER 6 HOW TO USE MINICUBE2 WITH 78K0R MICROCONTROLLER**
	Note 2 of **Figures 6-2** in **6.1.2 Circuit connection example** corrected, **Note 5** added	
	6.2.5 Securing of user resources and setting of security ID and on-chip debug option byte • Description of material added • Addition of **Note 1** in **Figure 6-11** • Addition of **Note** in **(a) Securement of debug monitor area**	
	6.2.6 Cautions on debugging • Deletion of **(5) Step execution** • Change of **(7) LVI default start function setting (address C1H)** • Change of **(10) Operation at voltage with which flash memory cannot be written** • Change of **(11) Debugging in 1-wire mode** • Change of **(12) Pseudo real-time monitor function**	
	Addition of **APPENDIX A EQUIVALENT CIRCUIT**	**APPENDIX A EQUIVALENT CIRCUIT**

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

QB-MINI2

RENESAS