RZ/T1, EC-1 Groups

Adding ETG.5003.1 and ETG.5003.2 Functionality

R01AN3781EJ0200
Rev.2.00
Sep 30, 2020

Outline

This manual explains the sample program which allows easy addition of firmware updating functionality and the object dictionary of the semiconductor device profile to utilize file-access over EtherCAT® (FoE) services in the EtherCAT Slave Stack Code (SSC) environment provided by Beckhoff Automation GmbH for RZ/T1 and EC-1 devices.

Target Devices

RZ/T1 Group

EC-1

Contents

1. Overview

Renesas provides a sample program which allows easy addition of firmware updating functionality (ETG.5003.2, ver. 0.9.13) and the object dictionary of the common device profile (ETG.5003.1, ver. 1.0.0) to utilize file-access over EtherCAT (FoE) services in the EtherCAT Slave Stack Code (SSC) environment provided by Beckhoff Automation GmbH for RZ/T1 and EC-1 devices.

Supported Targets	Description
Applicable RZT1 and EC-1 boards	RZ/T1 evaluation board (RTK7910022C00000BR) EC-1 remote I/O board (TS-EC-1)
Supported FoE service	Writing files Note: Reading files is not supported.
Supported flash memory	Serial flash ROM
Capacity to run the updating program	Tightly coupled memory: ATCM 512 KB/BTCM 32 KB
Master having confirmed operation	TwinCAT
Development environment having confirmed operation	IAR Embedded Workbench for ARM, ver. 7.7 and later

Table 1.1 Supported Targets

Function Overview

The firmware updater conforms with the ETG.5003.2 specification and includes the following features.

1) Compliant with the FoE
2) Writing to the serial flash ROM while EtherCAT is operating
3) Self-booting as a slave
4) Rewriting of the EEPROM

For details of the common device profile, see section 12, Semiconductor Device Profile.

2. Updating the Firmware

The sample program can be used to update the firmware of the slave as described below.

A slave vendor is able to provide an updating firmware file and password to a user, while the user is able to download the firmware to a slave by using the FoE from a master such as TwinCAT.

The file for updating the firmware has a checksum, which allows checking the validity of received data.

The updated firmware is written to a different area from that for the factory-default firmware in the serial flash ROM. After the update, the user application program in the form of the updated firmware is loaded to the ATCM to run through the sequence of booting.

If updating fails, restoration of the firmware written at the time of shipment is possible.

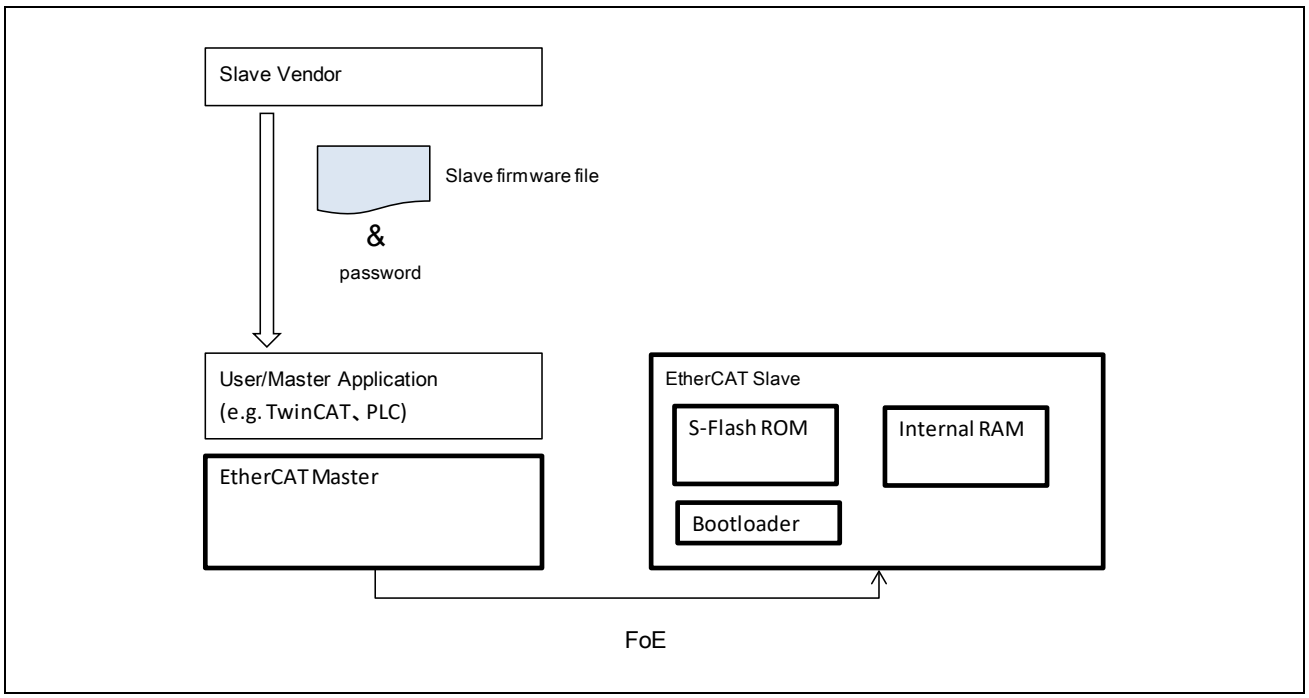

Figure 2.1 Example of System Configuration

3. Configuration of the Sample Program

The documents related to descriptions in this manual are listed below. Consult the following documents along with this manual.

- RZ/T1 Group User's Manual: Hardware (R01UH0483EJ)
- RZ/T1 Group Initial Settings Application Note (R01AN2554EJ)
- RZ/T1 Group Serial Flash Sample Program (SPIBSC) (R01AN3010EJ)

The sample program adds the firmware updater and the object dictionary of the common device profile as an application of the boot loader and FoE to the serial synchronous controller (SSC) of RZ/T1 and EC-1 devices.

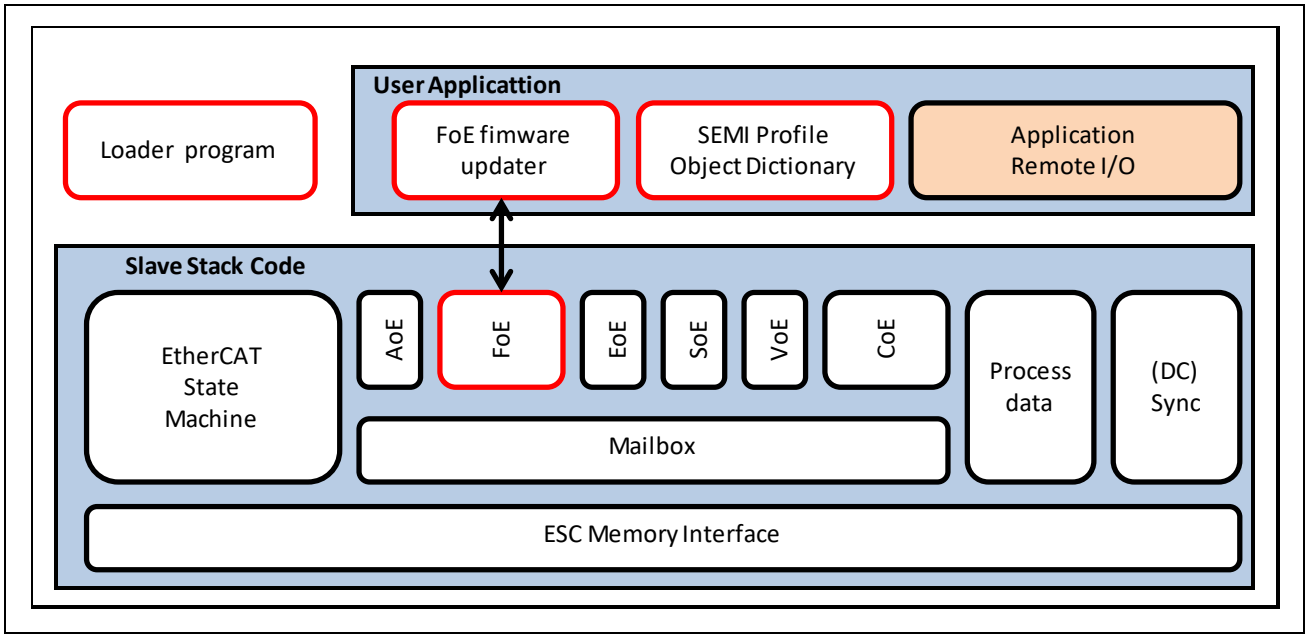

Figure 3.1 Configuration of the Sample Program

4. Procedure for Updating the Firmware

This section describes the procedure for updating the firmware for a slave and operations of the EtherCAT master and slave during the procedure. “Function” in the table shows which program is used to implement the corresponding slave operation.

No	Master/User	Slave	Function		
			SSC	FW boot loader	FW updater
1	requset BOOT	confirm BOOT	○		
2	download new slave FW	download new slave FW			
		(1)check filename			○
		(2)check password			○
		(3)write file data to S-Flash			○
		(4)check checksum of S-Flash			○
3		update SII			○
4	request INIT	reboot			
		(1) download new firmware to Internal RAM		○	
		(2)start new FW		○	
5	request PREOP	check if SII and firmware match	○		
6		confirm PREOP	○		
7	user:Check firmware version				
8	request SAFEOP	confirm SAFEOP	○		
9	request OP	confirm OP	○		

Table 4.1 Procedure for Updating the Firmware

1. Request the BOOT state.

Make the transition to the BOOT state for execution of the FoE.

2. Download new slave firmware.

Download the new updating firmware from the master.

The slave checks if (1) the filename and (2) the password are correct. If they are correct, it (3) writes data to the serial flash ROM. After the reception of all data, it (4) checks whether the checksum is correct.

3. Update the EEPROM.

Write the revision number of the new firmware to the EEPROM.

4. Request the INIT state.

After the transition from the BOOT to the INIT state, the slave is rebooted, and (1) downloads the program code from the serial flash ROM to the internal ROM then (2) operates with the new firmware.

5. Request the PREOP state.

Check if the revision number in the EEPROM matches that of the firmware.

6. Confirm the PREOP state.

Confirm the transition to the PREOP state.

7. User: Check firmware version.

The user can check whether the firmware has been updated to the new version by reading the value at 0x100A through the CoE object. The user can check the revision number by reading the value at 0x1018:03.

8. Request the SAFEOP state.

Make the transition to the SAFEOP state.

9. Request the OP state.

Make the transition to the OP state.

5. Hardware Configuration of the Sample Program

5.1 Booting from the Serial Flash ROM

To start the boot loader in the serial flash ROM, booting must be set to select SPI boot mode (booting up from serial flash memory).

5.2 Memory Map of Serial Flash ROM

Usage of the serial flash ROM is divided into three different areas.

Address Range	Name (Size)	Description
3000_0000H to 3000_FFFFH	Boot loader parameter area (64 KB)	Parameter area for the boot loader to be referred to by the boot function of the RZ/T1 or EC-1 Note: Bank 0 and bank 1 are referred to at the time of shipment and firmware updating, respectively.
3001_0000H to 3009_FFFFH	BANK0 area (64 KB + 512 KB = 576 KB)	Factory-default firmware area to be written by a serial flash ROM writer, ICE, etc.
3010_0000H to 3018_FFFFH	BANK1 area (64 KB + 512 KB = 576 KB)	Updating firmware area to be written by the FoE

Table 5.1 Classification of the Serial Flash ROM Areas

Figure 5.1 Memory Map, shows the memory map of the serial flash ROM.

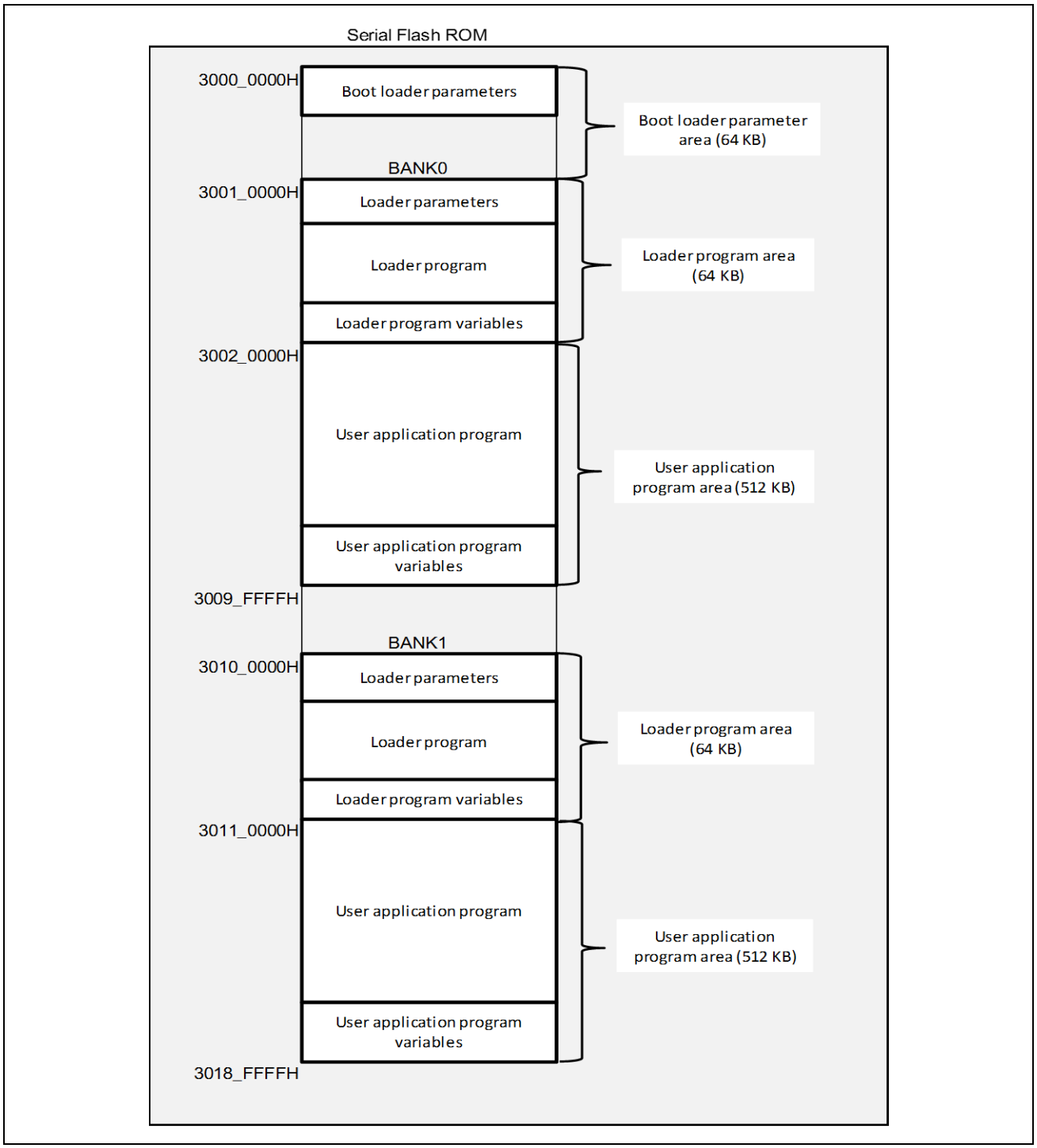

Figure 5.1 Memory Map

5.3 Overview of Bank 0 Boot Operations

This following describes operation for booting the factory-default firmware written to bank 0 through the procedure illustrated in Figure 5.2 BANK 0 Boot Operations.

The boot function of the RZ/T1 or EC-1

1) refers to the values in the boot loader parameter area,

2) transfers the loader program in bank 0 to the BTCM, and then

3) hands processing to the loader program.

After initializing the various stack pointers, the loader program

4) transfers the loader program variables to the BTCM and makes settings for peripheral modules, etc.

It also refers to the values in the boot loader parameter area,

5) transfers the user application program to the ATCM, and then

6) hands processing to the user application program.

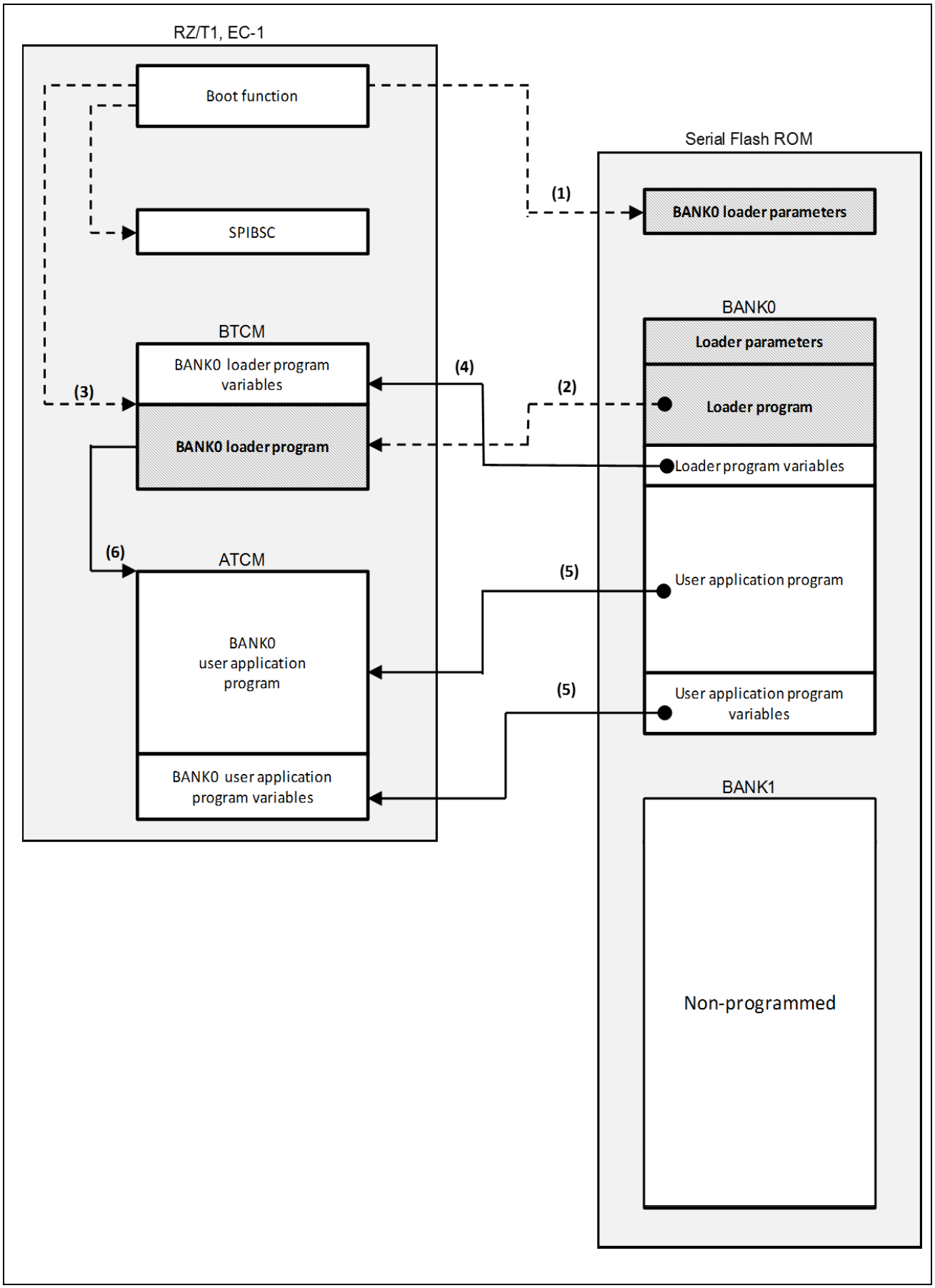

Figure 5.2 BANK 0 Boot Operations

5.4 Overview of Operation for Updating Firmware in Bank 1

The following describes operation for updating the firmware by using the FoE while a user application program is running from the ATCM, with the procedure illustrated in Figure 5.3, Operation for Updating the Firmware in Bank 1.

1. The master sends the filename and password of the binary file for the updating firmware at the opening of the FoE, so check if the prefix of the filename and the password are correct. If they are correct, reception of binary data starts.
2. Erase the boot loader parameter area at the beginning of the serial flash ROM.
3. If the firmware update is interrupted for any reason, copy the BANK0 loader parameters to the boot loader parameter area so that the factory firmware can be started.
4. One sector (64 KB) is erased from the address where bank 1 starts. During erasure, the busy status indicator is returned so that the master does not reach the timeout time.
3. Data are received on completion of the erasure. The data are stored in the reception buffer allocated for the storage of user application program variables in the ATCM. The data in the reception buffer are written to the serial flash ROM every time two pages (512 bytes) of data are accumulated. An ACK packet is returned to the master. On completion of writing one sector of data, erase the next sector ((1) in the figure).
4. Repeat step 3 until writing reaches the address where bank 1 ends.
5. On completion of writing to the whole area of bank 1, check if the checksums match.
6. If they match, erase the boot loader parameter area at the start of the serial flash ROM.
7. Copy the bank 1 loader parameters to the boot loader parameter area ((2) in the figure).
8. Update the revision number in the SII memory to the revision number of the updating firmware.

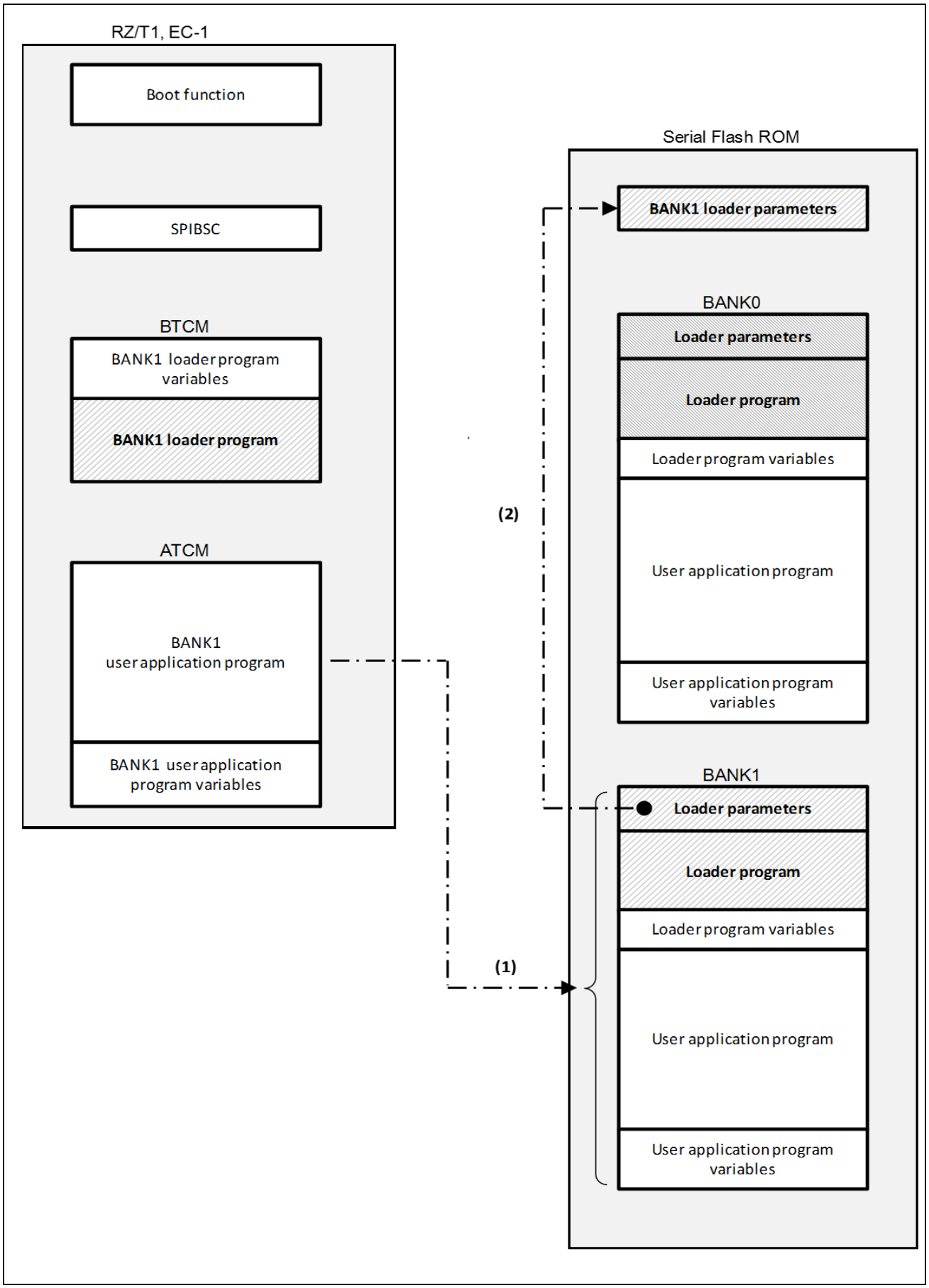

Figure 5.3 Operation for Updating Firmware in Bank 1

5.5 Overview of Operations to Reboot from Bank 1

The following describes operation from writing of the updating firmware to bank 1 until rebooting, with the procedure illustrated in Figure 5.4, Operations to Reboot from Bank 1.

After updating of the firmware in bank 1 has been completed normally,

(1) the reloader program copies the loader program to the BTCM with reference to the bank 1 loader parameters in the boot loader parameter area, and then

(2) runs the loader program by jumping to the address where the bank 1 loader program starts.

After that, processing is the same as in step 4 and subsequent steps in section 5.6, Overview of Bank 1 Boot Operations, until the application program is run.

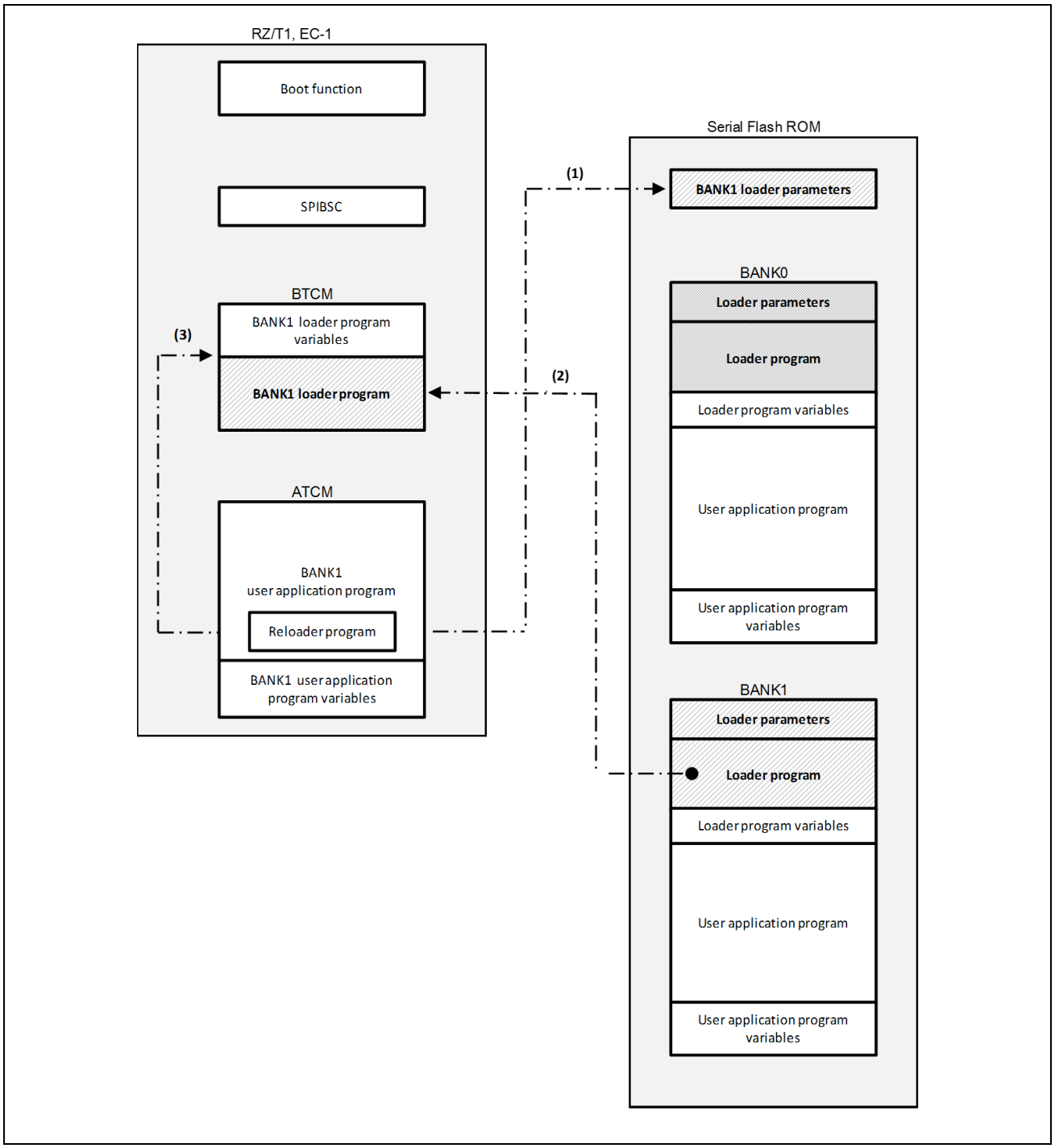

Figure 5.4 Operations to Reboot from Bank 1

5.6 Overview of Bank 1 Boot Operations

The following describes operation for booting from bank 1 when power is supplied after updating the firmware, with the procedure illustrated in Figure 5.5

(1) The boot function refers to the values in the boot loader parameter area,

(2) transfers the loader program in bank 1 to the BTCM, and then

(3) hands processing to the loader program.

After initializing the various stack pointers, the loader program

(4) transfers the loader program variables to the BTCM and makes settings for peripheral modules, etc.

It also refers to the values in the boot loader parameter area,

(5) transfers the user application program to the ATCM, and then

(6) hands processing to the user application program.

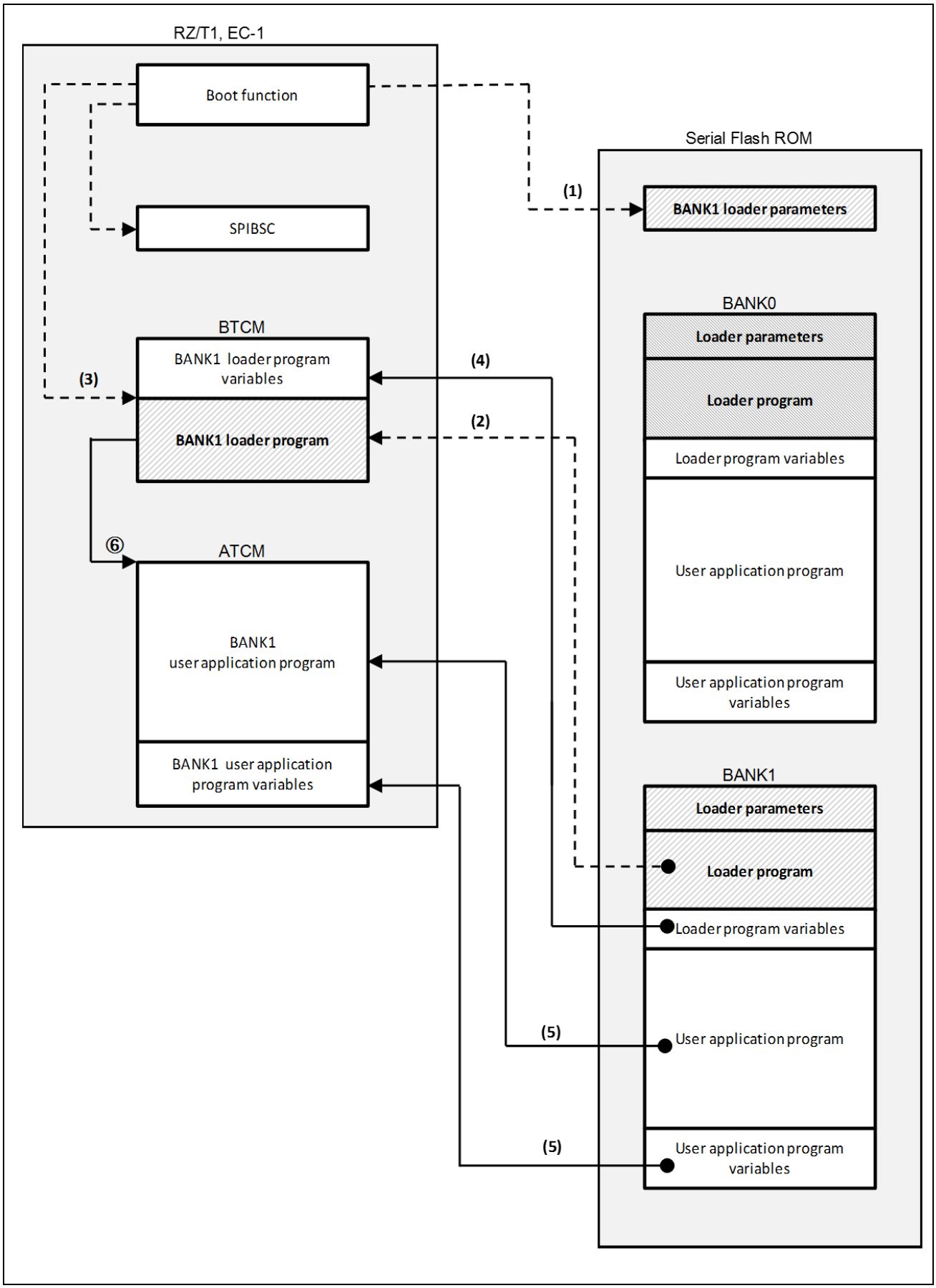

Figure 5.5 Bank 1 Boot Operations

5.7 Loader Parameters

In the sample program, parameters for downloading the user application program to the ATCM have been added as DUMMY1 to DUMMY3, which are not used by default. The loader parameter information for the sample program is listed in Table 5.2, Loader Parameter Information.

Figure 5.6, Reference to the Loader Parameters, shows the relationship between the parameters and addresses in the serial flash ROM.

Parameter Name	Offset Address	Description
CACHE_FLG	0000_0000H	Selects whether to enable the I1 cache and D1 cache of the Cortex-R4 in boot processing (for speeding up operations).
SSLDR_V	0000_0004H	Setting of the SSL delay register (SSLDR)
SPBCR_V	0000_0008H	Setting of the bit-rate configuration register (SPBCR)
DRCR_V	0000_000CH	Setting of the data read control register (DRCR)
SPIBSC_FLG	0000_0010H	Selects whether to change the SPIBSC setting back to the initial value after boot processing finishes.
LDR_ADDR_NML	0000_0014H	Sets the address where the loader program starts.
LDR_SIZE_NML	0000_0018H	Sets the size of the loader program.
DEST_ADDR_NML	0000_001CH	Specifies the address in the BTCM where the area for use as the destination for extraction of the loader program starts.
VECTOR_RBLK	0000_0020H	Specifies the address where the vector table of the user application program starts.
USR_P_RBLK	0000_0024H	Specifies the address where the user application program starts.
USR_D_RBLK	0000_0028H	Specifies the address where the user application program variable starts.
DUMMY4-10	0000_002CH	Not used
	0000_0030H	Not used
	0000_0034H	Not used
	0000_0038H	Not used
	0000_003CH	Not used
	0000_0040H	Not used
	0000_0044H	Not used
CECJ_SUM	0000_0048H	Checksum value of the loader parameters

Table 5.2 Loader Parameter Information

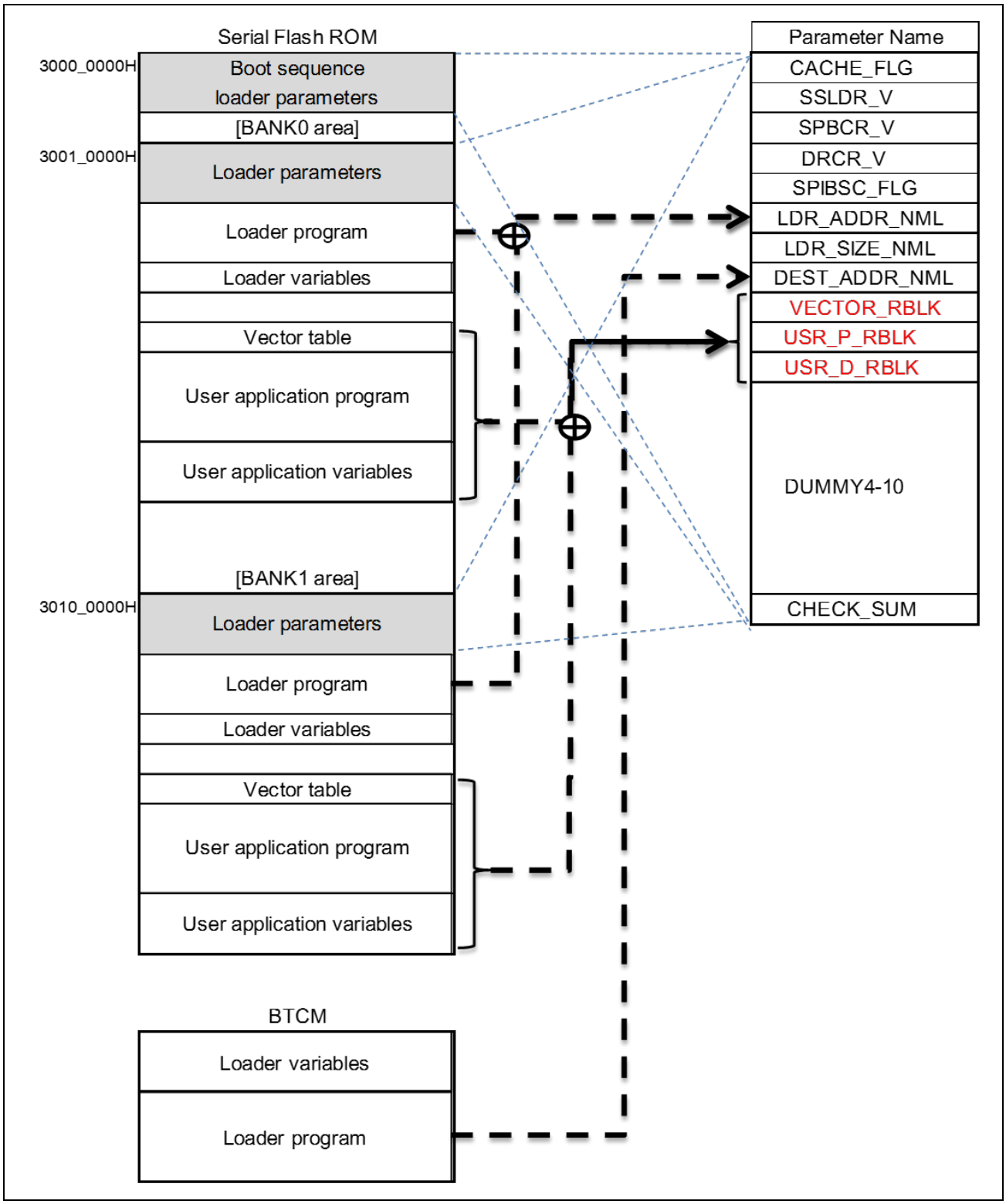

Figure 5.6Reference to the Loader Parameters

6. Assignment of the Sample Program to Sections

Figure 6.1, Assignment of the Sample Program to Sections, shows the assignment of the sample program to sections.

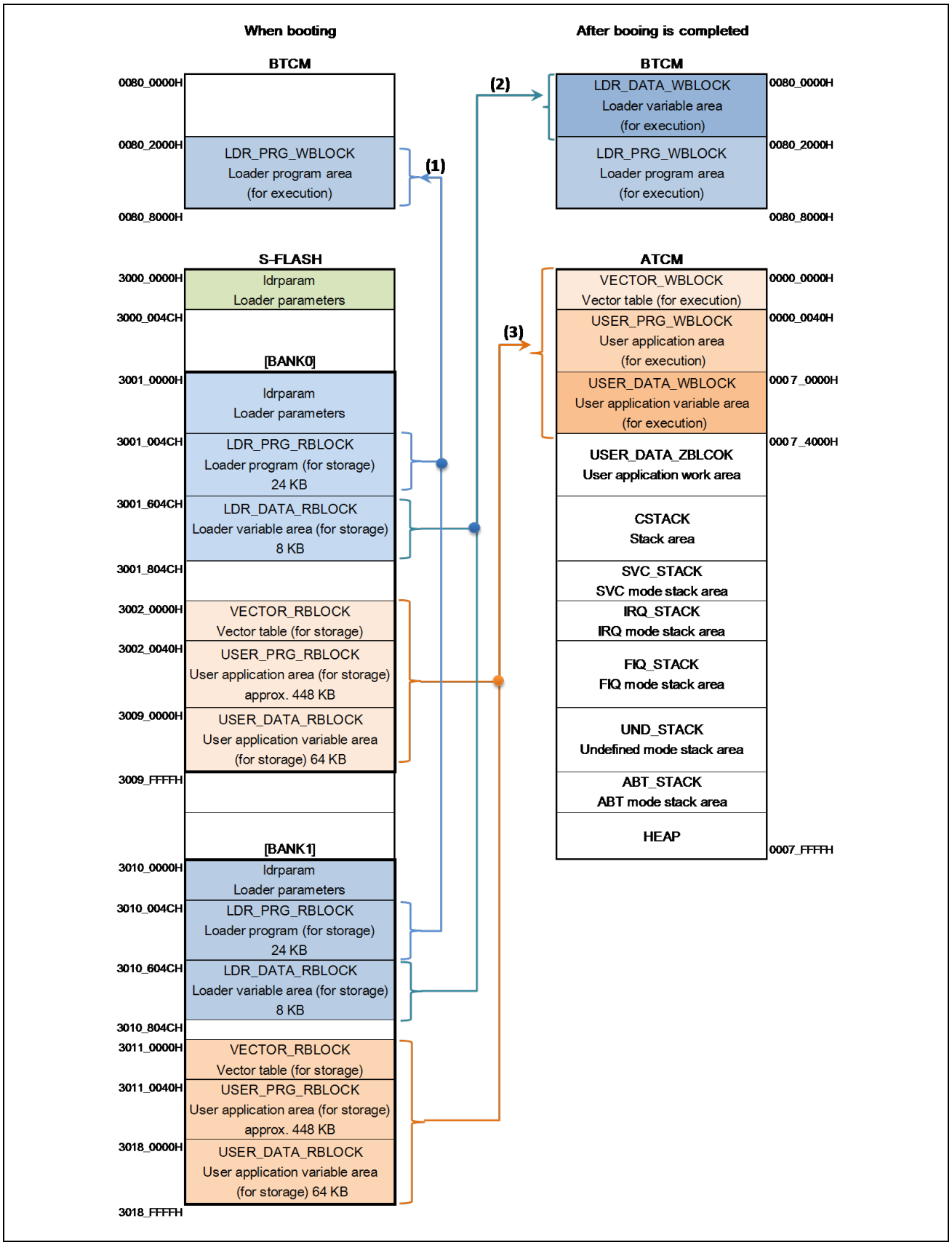

Figure 6.1 Assignment of the Sample Program to Sections

7. Build Configuration of the Sample Program

Renesas provides project files for writing the factory-default firmware to bank 0 or bank 1 of the serial flash ROM and for creating the binary file for downloading the updating firmware.

Figure 7.1, Build Configuration of the Sample Program, shows the relationships between the project files, build configuration, icf files, and sections to be linked. Select Debug_BANK1 for the build configuration when debugging the updating firmware, and select Release_BANK1 for the build configuration when creating the binary file.

With Debug_BANK1, the bank 1 loader parameters are written to the boot loader parameter area.

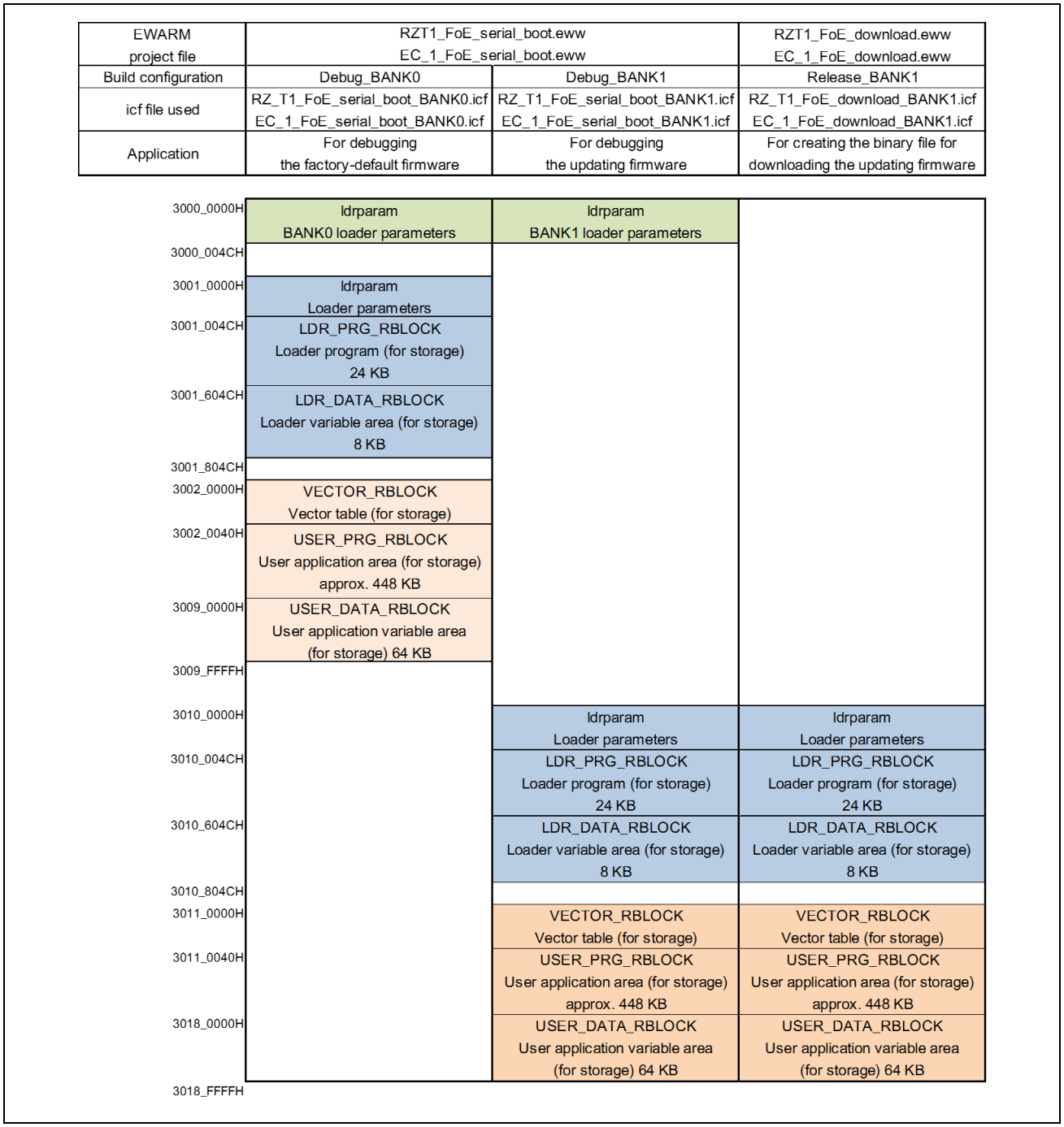

Figure 7.1 Build Configuration of the Sample Program

8. Constants

Table 8.1 Constants Used in the Sample Program (1)

Constant Name	Setting	Description
SPIBSC_LDR_ADDR	(0x10000014)	Address where “LDR_ADDR_NML” of the loader parameters is stored
SPIBSC_LDR_SIZE	(0x10000018)	Address where “LDR_SIZE_NML” of the loader parameters is stored
SPIBSC_DEST_ADDR	(0x1000001C)	Address where “DEST_ADDR_NML” of the loader parameters is stored
SPIBSC_VCTR_ADDR	(0x10000020)	Address where “VECTOR_RBLK” of the loader parameters is stored
SPIBSC_USRP_ADDR	(0x10000024)	Address where “USR_P_RBLK” of the loader parameters is stored
SPIBSC_USRD_ADDR	(0x10000028)	Address where “USR_D_RBLK” of the loader parameters is stored

Table 8.2 Constants Used in the Sample Program (2)

Constant Name	Setting	Description
SF_PAGE_SIZE	(256)	Page size of the serial flash ROM
SF_SECTOR_SIZE	(65536)	Sector size of the serial flash ROM (64 KB)
SF_NUM_OF_SECTOR	(1024)	Total number of sectors of the serial flash ROM
SF_FOE_BANK0_ADDR	(0x10010000)	Address where bank 0 starts
SF_FOE_BANK1_ADDR	(0x10100000)	Address where bank 1 starts
SF_FOE_APPLI_SIZE	(0x00090000)	Bank size (576 KB)

9. Functions

The tables below list the functions related to the boot loader and updating of the FoE firmware.

Table 9.1 List of Boot Loader Related Functions

Function Name	Outline
copy_to_atcm	This function runs from the BTCM and handles processing to deploy the user application program from the serial flash ROM to the ATCM at the time of booting.
copy_to_btcm	This function runs from the ATCM and handles processing to deploy the loader program to the BTCM at the time of rebooting.

Table 9.2 List of Functions Related to Updating of the FoE Firmware

Function Name	Outline
BL_Start	Handles processing to start the transition from the INIT to the BOOT state.
BL_StartDownload	Handles processing to start downloading the FoE file data.
BL_Data	Handles processing to receive the FoE file data.
BL_CheckSum	Handles processing to check the checksum of the updating firmware area.
BL_Data_write	Handles processing to write the file data to the serial flash ROM.
BL_SetRebootFlag	Sets the reboot flag.
BL_CheckRebootFlag	Checks the reboot flag.
BL_Reboot	Reboot processing (BOOT -> INIT)
BL_Copy_1Page	Copies one page of data to the serial flash ROM.

10. Creating the Sample Program Source Files

10.1 Installing the SSC Tool

You will need to obtain the SSC Tool under license from the EtherCAT Technology Group before you can install it.

The SSC Tool filename assumed by this sample program is "*SSC_V5i11.zip*", version 5.11.

10.2 Extracting the Sample Program Files

Extract the sample program files.

10.3 Creating the SSC Source Files

(1) Run the SSC Tool project file (*.esp) included with the sample program to start the SSC Tool.

• RZ/T1

¥workspace¥iccarm¥EtherCAT_SSC_FoE¥src¥sample¥src¥ssc_project¥RZT1-R EtherCAT [FoE] s.esp

• EC-1

¥Source¥Project¥EtherCAT_RemoteIO¥SSC¥EC-1 [FoE].esp

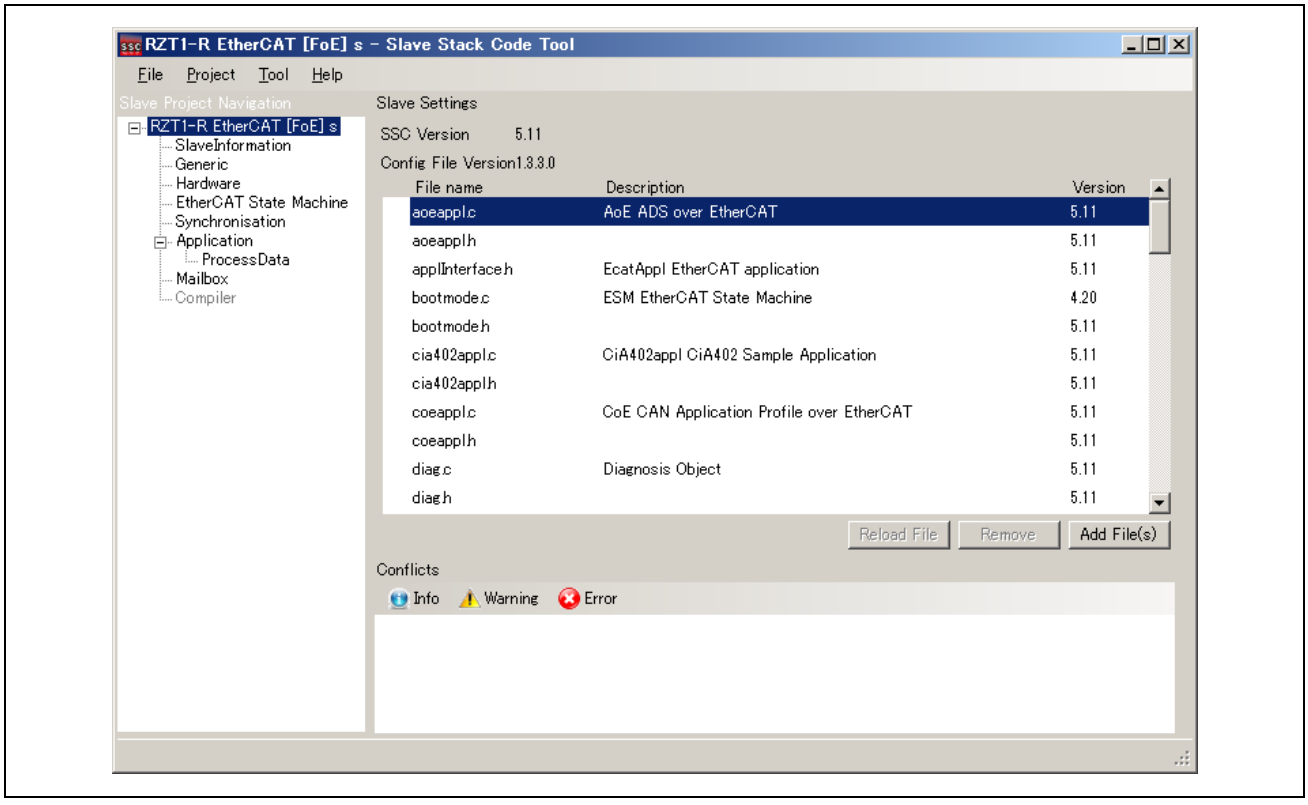

Figure10.1 RZ/T1 SSC Tool Startup Window

(2) Create the source files.

Menu [Project] -> [Create new Slave Files] -> [Start] -> [OK]

The above step leads to the creation of the *"¥Src"* directory and source files.

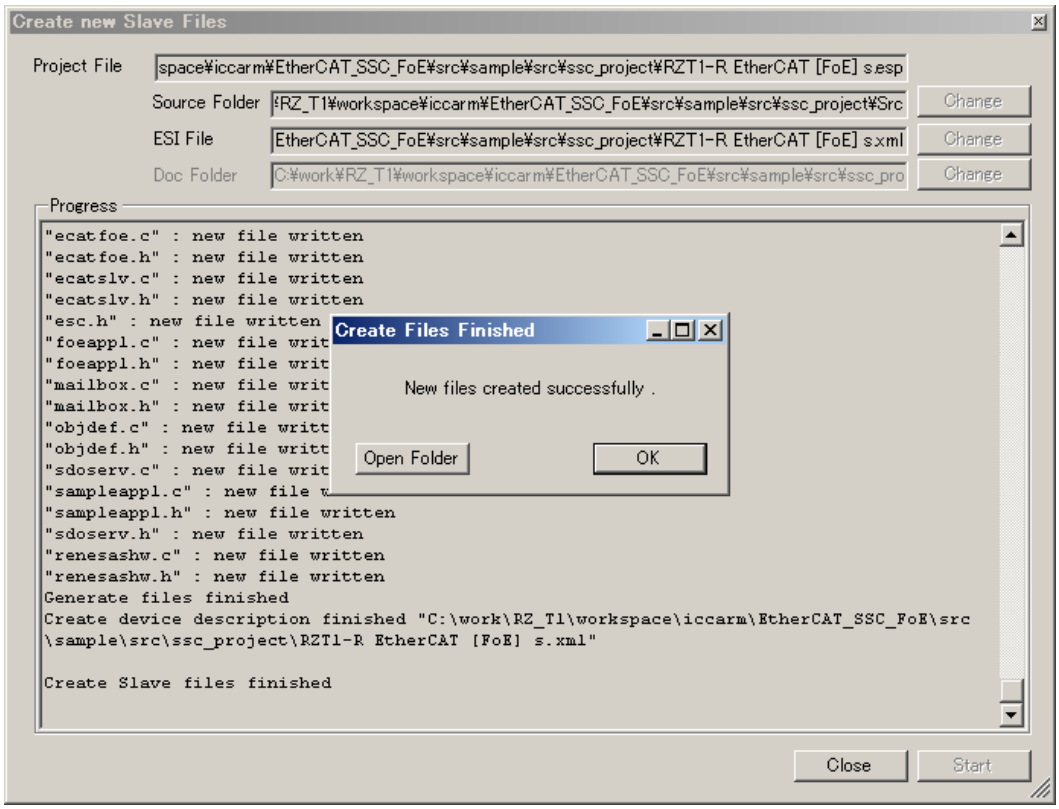

Figure10.2 RZ/T1 SSC Tool Source File Creation Window

■ CAUTION

When creating source files, do not check the following setting.

"Tool" -> "Options" -> "Add comments if obsolete code was skipped" in the "Create Files" tabbed page

If the required patch command (ver. 2.5.9 or a later version of GNU patch) is not installed on your PC, you will need to install it.

Go to the following link to download the patch command (currently ver. 2.5.9) and store "patch.exe" in the path to the directory.

http://gnuwin32.sourceforge.net/packages/patch.htm

10.4 Running the Batch File

The batch file (.bat) is used to apply the patch file to add the boot loader, FoE firmware update functionality, etc. to the SSC source file.

Run “apply_patch.bat” included with the sample program. This applies the patch file.

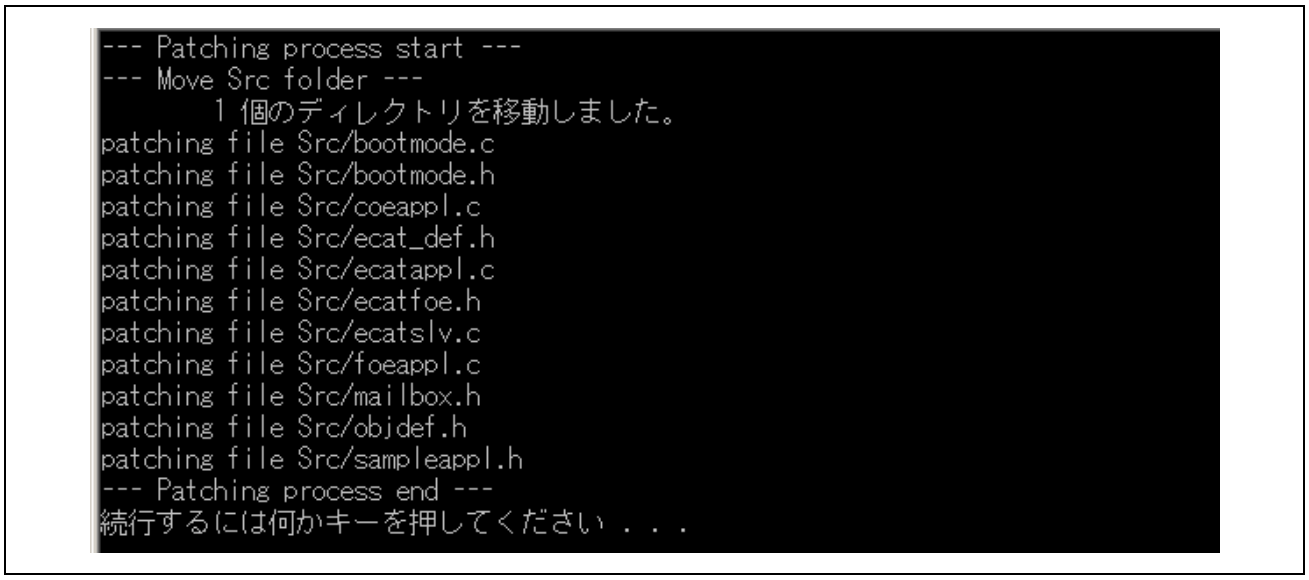

Figure10.3 “apply_patch.bat” Execution Display

■ CAUTION

When the patch command cannot be executed in Windows 7

Right-click on the command prompt icon or shortcut and click on “Run as Administrator” to start command prompt.

11. Checking Operations

11.1 Debugger Start (IAR EWARM)

Double-click on the included IAR project file to start IAR Embedded Workbench for ARM.

• RZ/T1

¥workspace¥iccarm¥EtherCAT_SSC_FoE¥RZ_T1_FoE.eww

• EC-1

¥Source¥Project¥EtherCAT_ComB_FoE¥IAR¥EC_1_FoE.eww

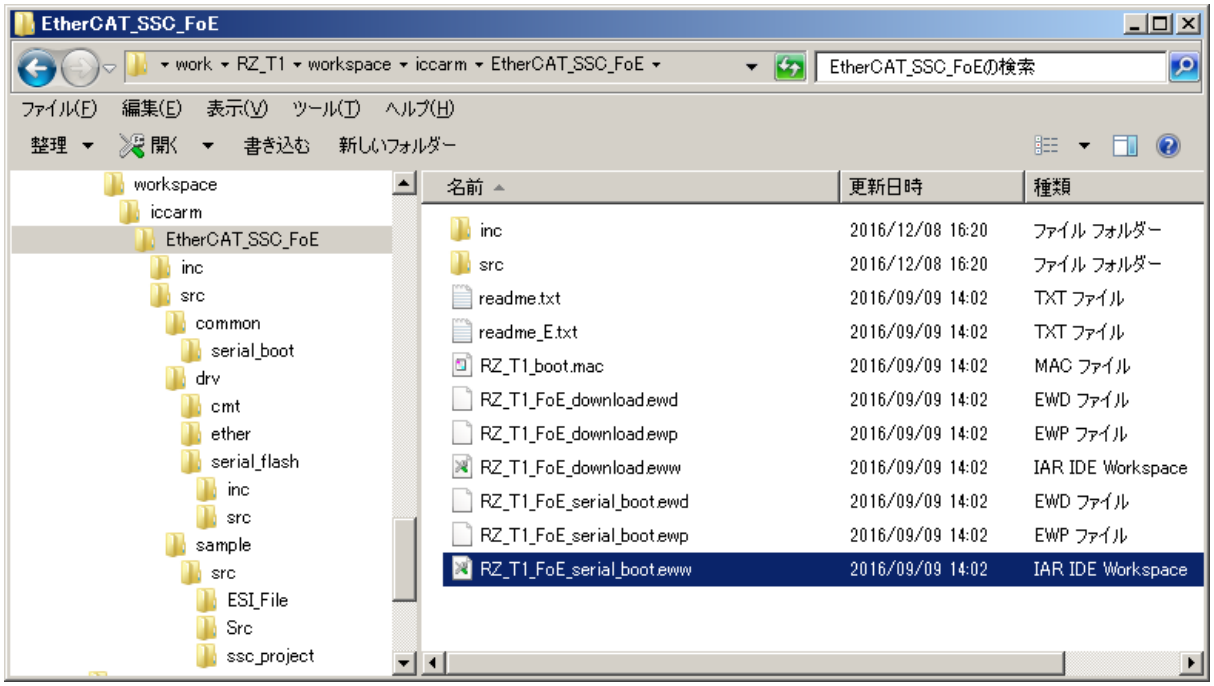

Figure 11.1 RZ/T1 Factory-Default Firmware IAR Project File Directory Window

Connect ICE to the evaluation board.

11.1.1 BANK0 Build and Debug

(1) Select "BANK0" project for the factory-default firmware.

(2) Set “Debug_BANK0” mode and build with “Project”-> “Rebuild All”.

(3) Double-click "Download and Debug" to write the factory-default firmware code to BANK0 on the serial flash ROM.

If there is no error display and the debugger screen is displayed, it is successful.

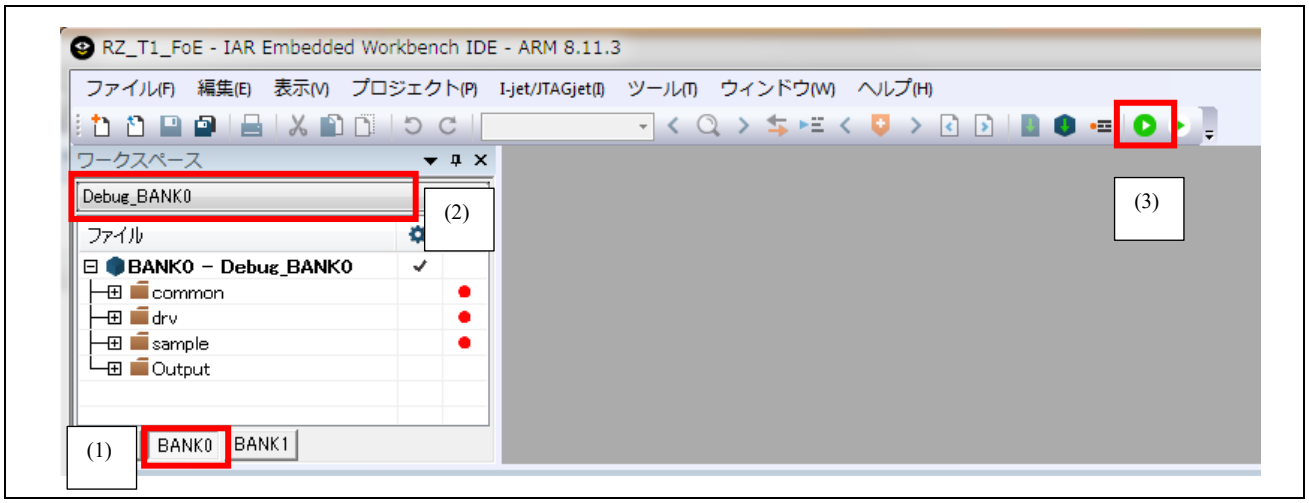

Figure 11.2 RZ/T1 Debug_BANK0 build Window

11.1.2 BANK1 Build and Debug

(1) Select "BANK1" project for the update firmware.

(2) Set “Debug_BANK1” mode and build with “Project”-> “Rebuild All”.

(3) Double-click "Download and Debug" to write the update firmware code to BANK1 on the serial flash ROM.

If there is no error display and the debugger screen is displayed, it is successful.

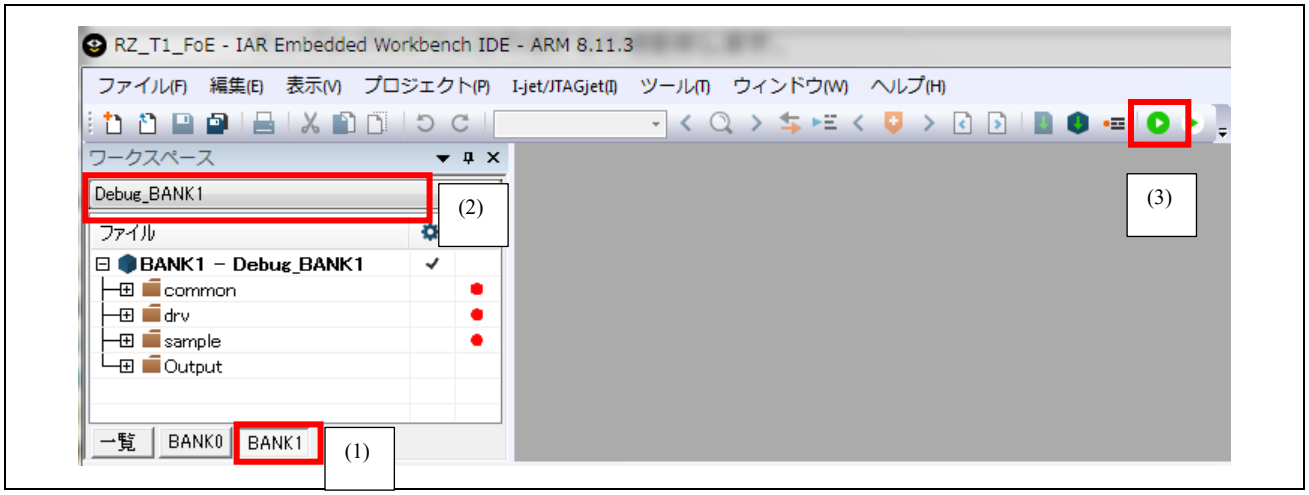

Figure 11.3 RZ/T1 Debug_BANK1 build Window

11.1.3 BANK1 Download File Creation

Create a download file when debugging the update firmware is complete.

(1) Select "BANK1" project for factory firmware.

(2) Set “Release_BANK1” mode and build with “Project”-> “Rebuild All”

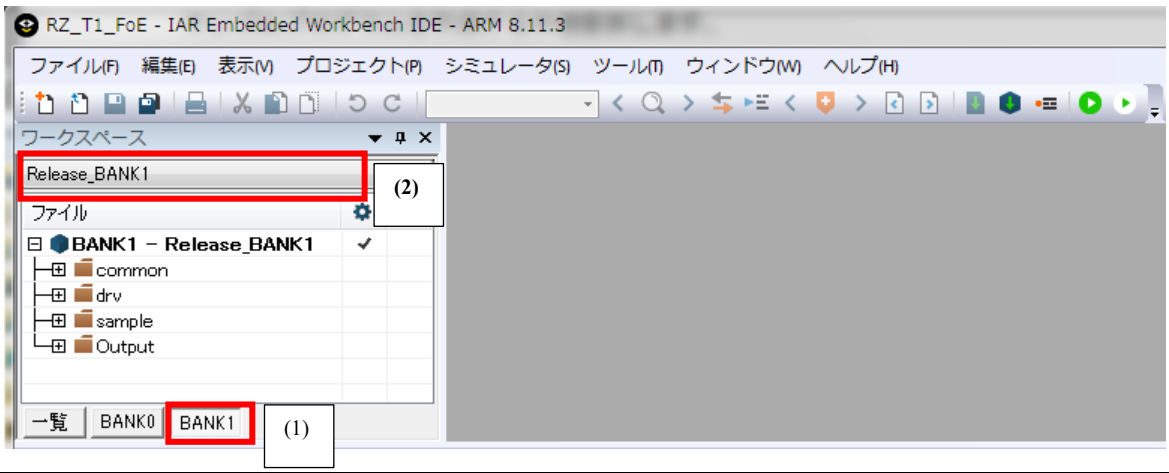

Figure 11.3 Release_BANK1 build Window

When the build is complete, the update firmware download file will be created.

・RZ/T1

¥workspace¥iccarm¥EtherCAT_SSC_FoE¥Release_BANK1¥Exe¥ ECATFW_B1_FoE.efw

・EC-1

¥Source¥Project¥EtherCAT_RemoteIO¥IAR¥ Release_BANK1¥Exe¥ ECATFW_B1_FoE.efw

The following items from among the parameters for the updated firmware file can be changed in the source code.

Table 11.1 Updated Firmware File Parameters

Parameter	Outline	Corresponding Sections in the Source Files
Prefix of the filename	String: “ECATFW_B1”	Function aFirmwareDownloadHeader in foeapppl.c
Password of the file	Eight digit numbers: 00000000	Function aFilePassword in foeappl.c
Firmware version	RZ/T1 String: “5.12” EC-1 String: “1.01”	Function DEVCE_SW_VERSION in ecat_def.h

11.2 Starting TwinCAT®

11.2.1 Preparing the ESI File

Copy the included EtherCAT Slave Information (ESI) file to the directory “C:¥TwinCAT¥Io¥EtherCAT”.

• RZ/T1

¥workspace¥iccarm¥EtherCAT_SSC_FoE¥src¥sample¥src¥ESI_File¥RZT1-R EtherCAT [FoE] s.xml

• EC-1

¥Source¥Project¥EtherCAT_RemoteIO¥SSC¥ESI_File¥EC-1 [FoE].xml

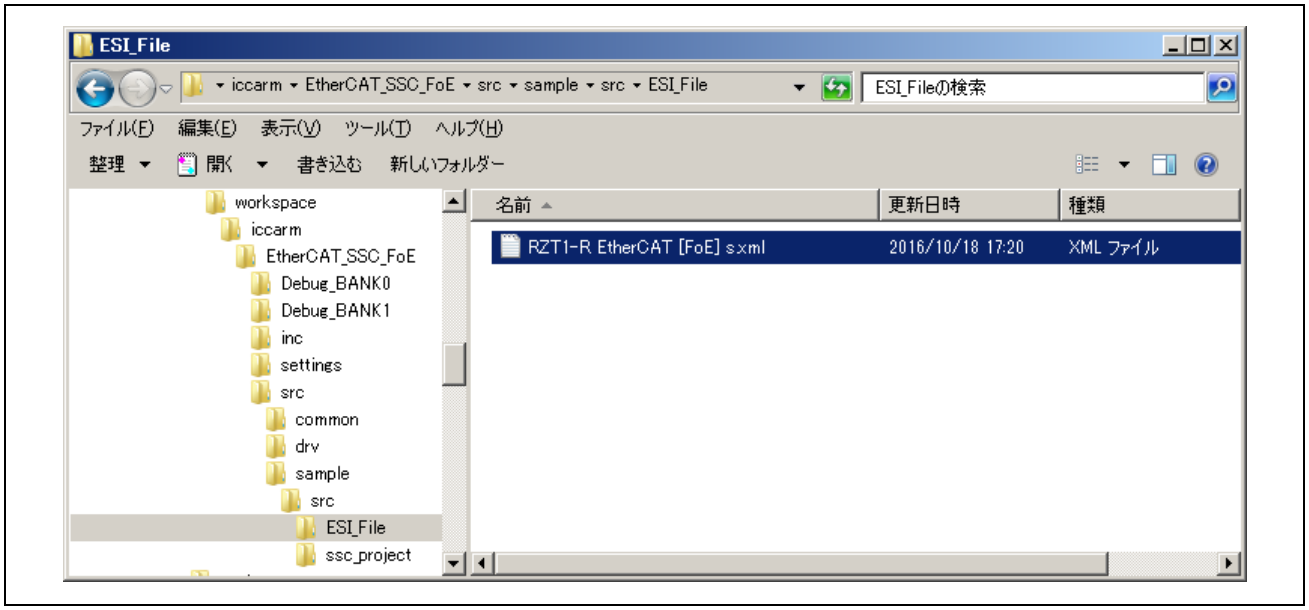

Figure 11.5 ESI File for the RZ/T1

11.2.2 Starting TwinCAT®

To start the TwinCAT System Manager, right-click on “I/O Device” and select “Scan Devices…”.

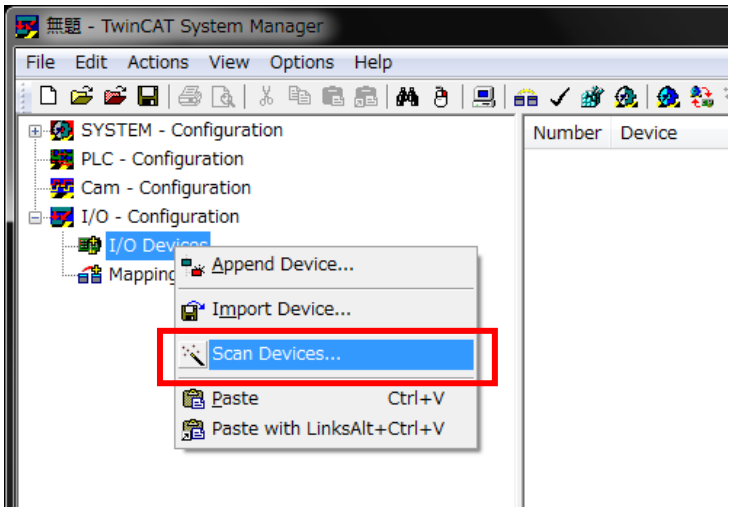

Figure 11.6 Device Search: Step 1

Select “OK”.

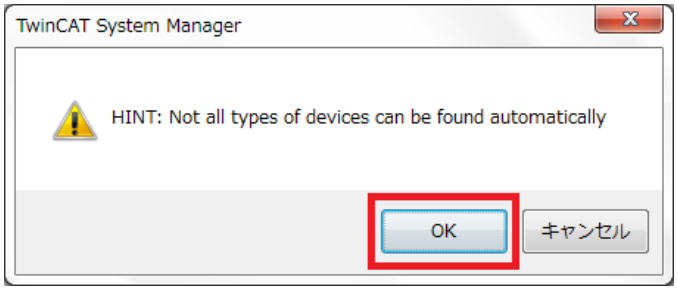

Figure 11.7 Device Search: Step 2

Check only “EtherCAT” and select “OK”.

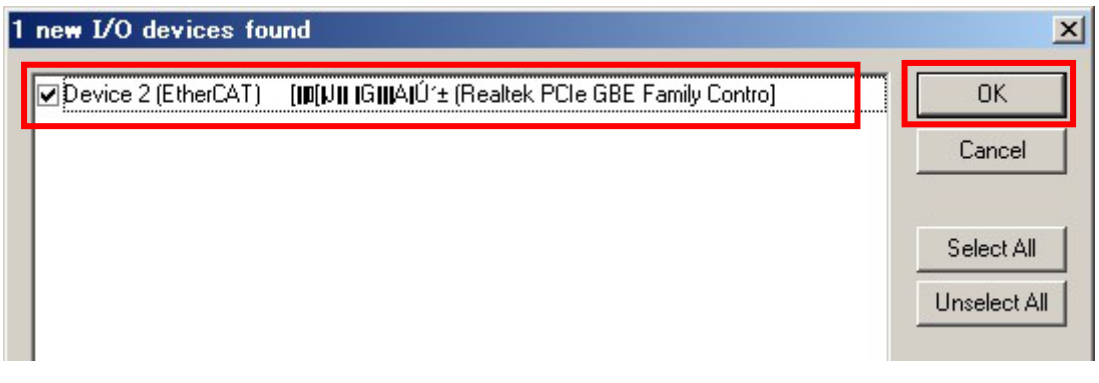

Figure 11.8 Device Search: Step 3

Click on the “OK” button.

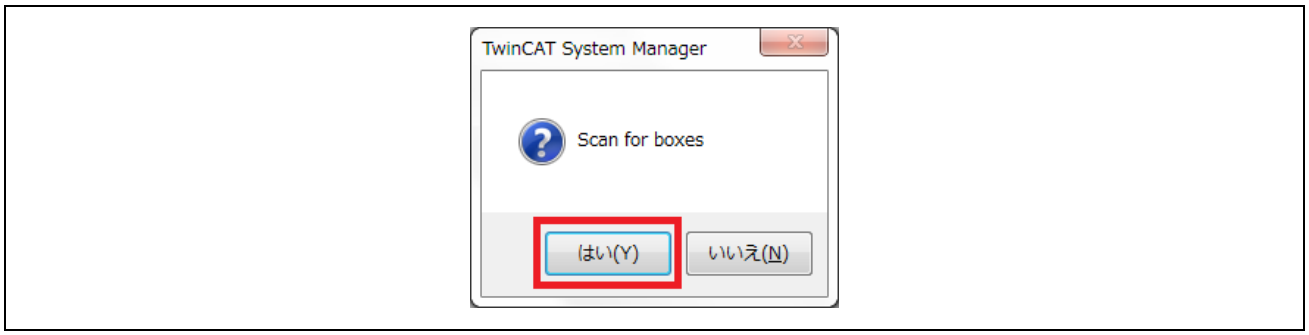

Figure 11.9 Device Search: Step 4

Click on the “OK” button to activate Free Run.

Figure 11.4 Device Search: Step 5

11.2.3 Writing the ESI File

If either of the following device names is displayed against the box name, the ESI file has already been written.

In this case, do not proceed with the following steps, but follow the instructions from section 11.4. If a different device name is displayed, follow the procedure below.

Device name

- RZ/T1: “RZ/T1-R EtherCAT FoE”
- EC-1: ”EC-1 FoE”

Select “Box 1”, and then select the “EtherCAT” tab, then click on the “Advanced Settings…” button.

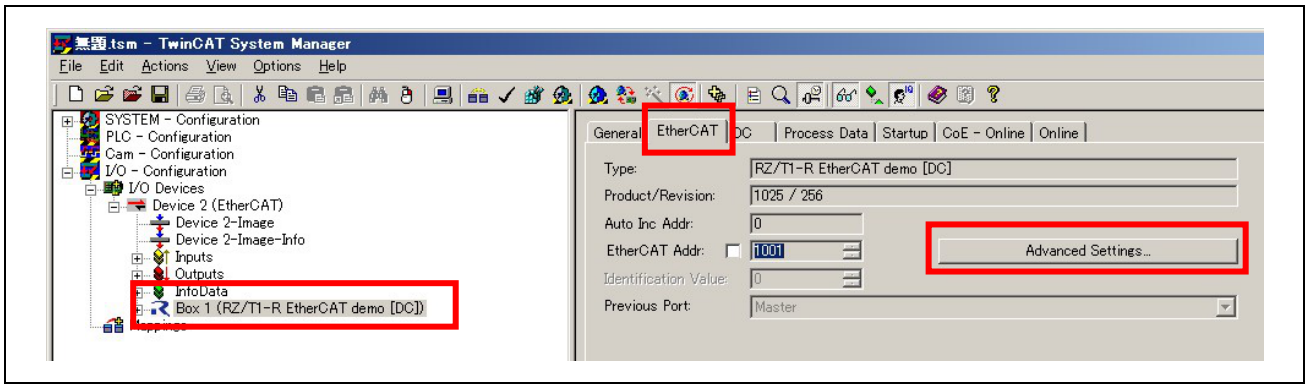

Figure 11.5 Writing the ESI File for the RZ/T1: Step 1

Select “Hex Editor”, then click on the “Download from List…” button.

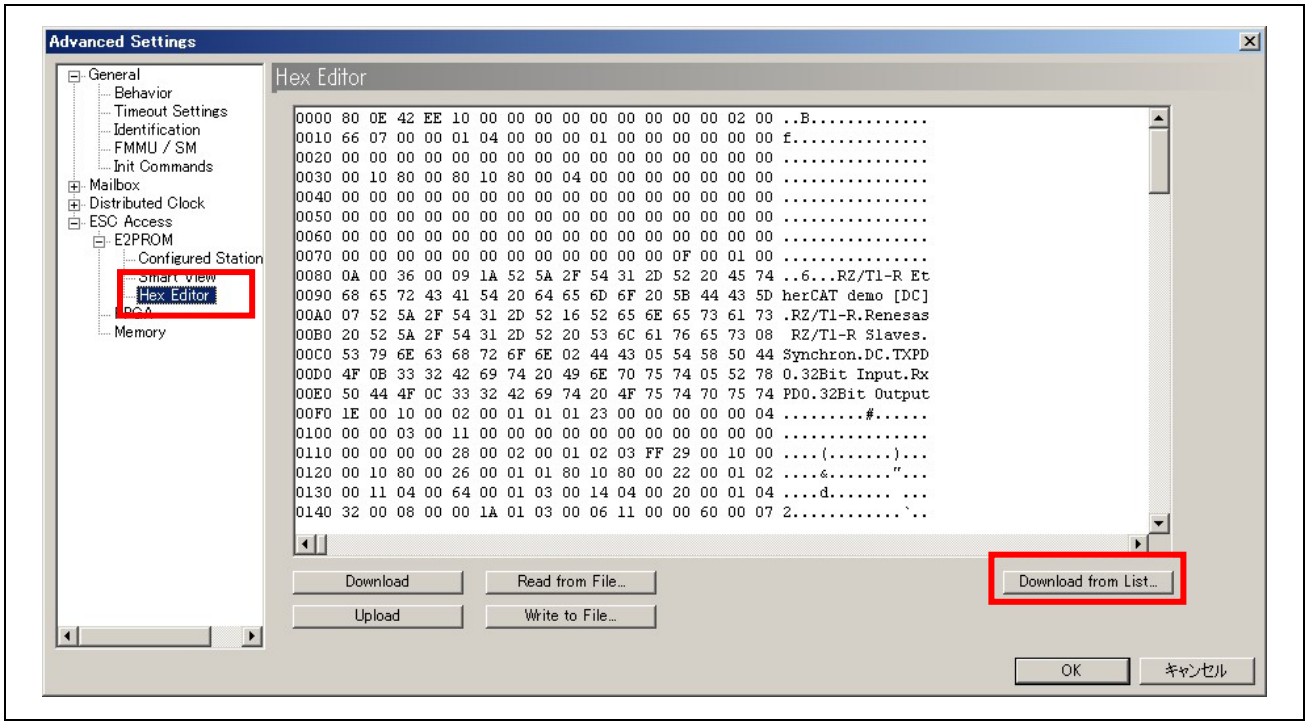

Figure 11.12 Writing the ESI File for the RZ/T1: Step 2

Select the device name for the ESI file to be written then click on the “OK” button.

The respective device names for the RZ/T1 and EC-1 are given below.

Note: This procedure takes some time since the file is written to the EEPROM.

Device name

- RZ/T1: “RZ/T1-R EtherCAT FoE”
- EC-1: “EC-1 FoE”

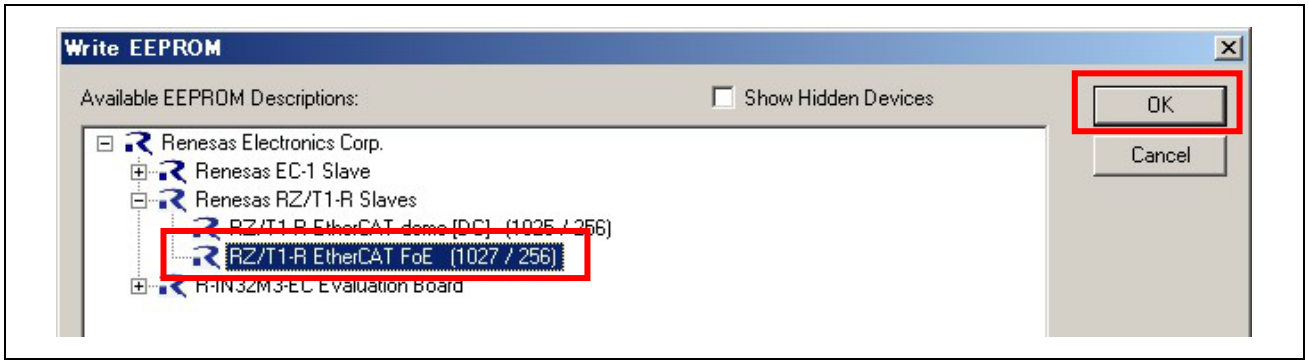

Figure 11.13 Selecting the ESI File for the RZ/T1

Click on the “OK” button. Writing the contents of the ESI file to the EEPROM is complete.

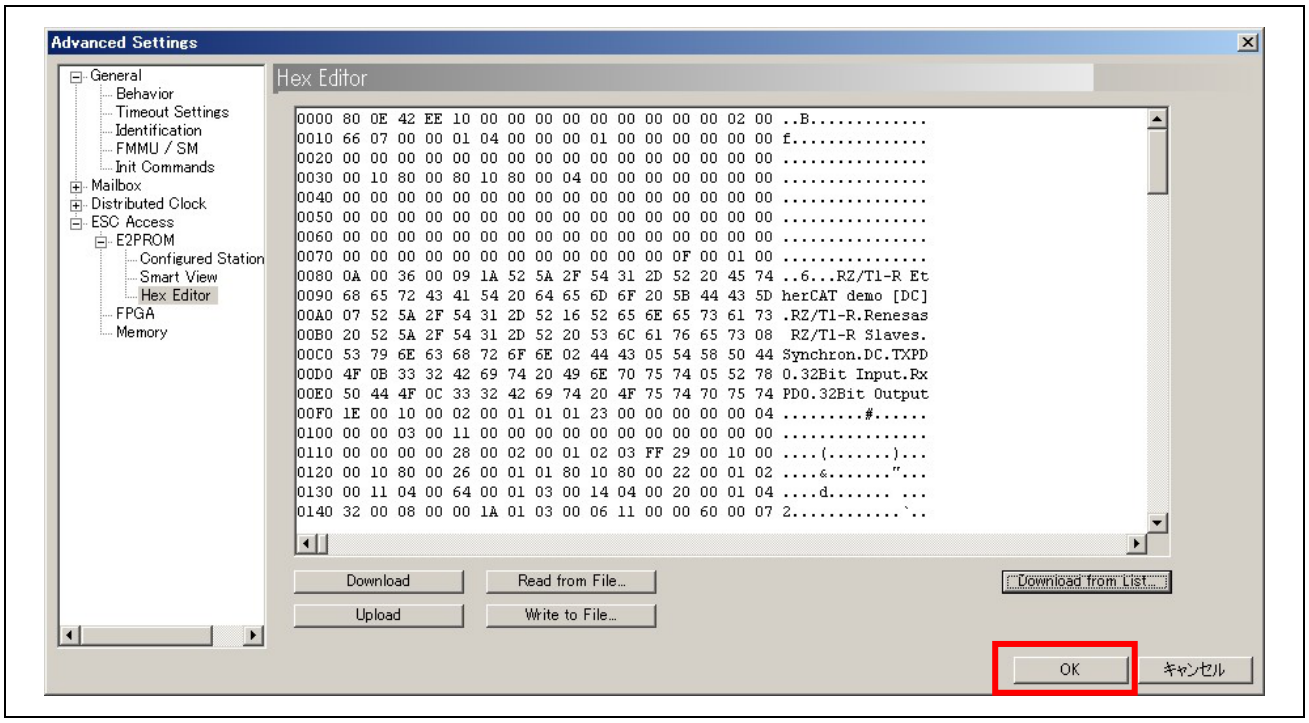

Figure 11.14 Completion of Writing the ESI File for the RZ/T1

To detect the device again after writing the ESI file, right-click on the device name under “I/O Devices” and select “Delete Device”.

After you have deleted the device, start again from the stage of searching for the device described in section 11.3.2.

If the device name corresponding to the written ESI is displayed against the box name, follow the procedure from section 11.4.

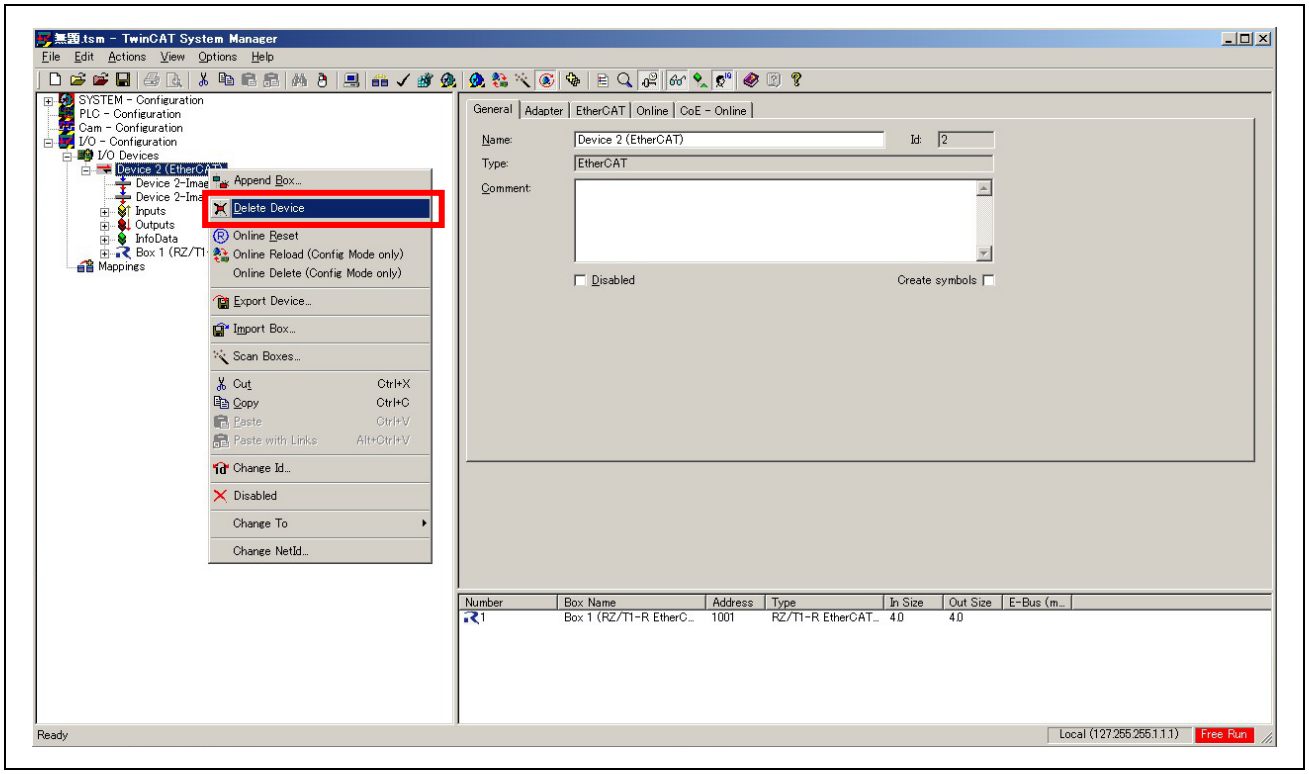

Figure 11.15 Deleting the RZ/T1 Device

11.3 Updating the Firmware by TwinCAT®

Select “Box 1 (RZ/T1-R EtherCAT FoE or EC-1 FoE)”, then click on the “Online” tab.

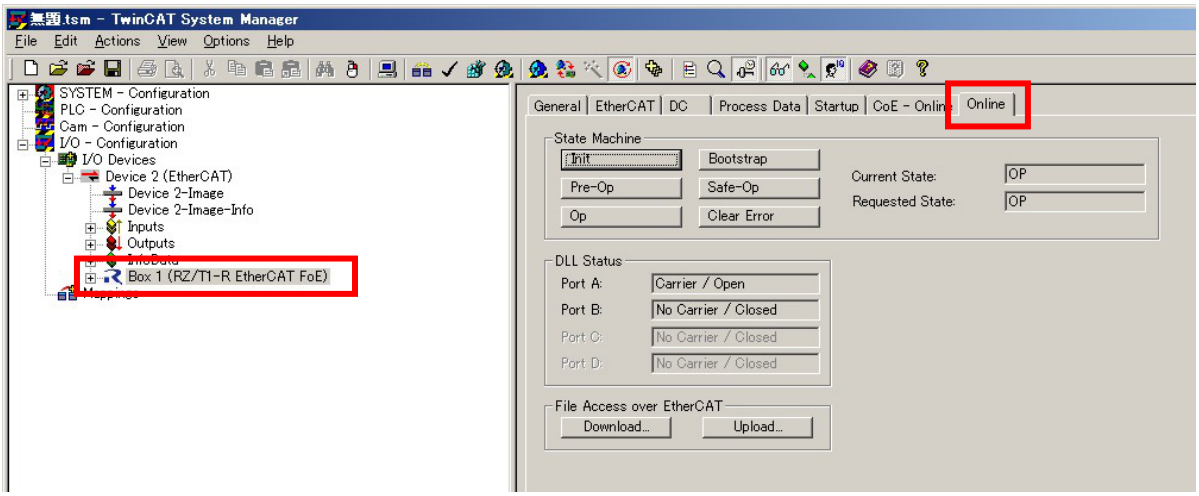

Figure 11.6 RZ/T1 “Online” Tabbed Page 1

Press button (1) “Init” then (2) “Bootstrap” in that order, and confirm that “Current State” has changed to (3) “BOOT”.

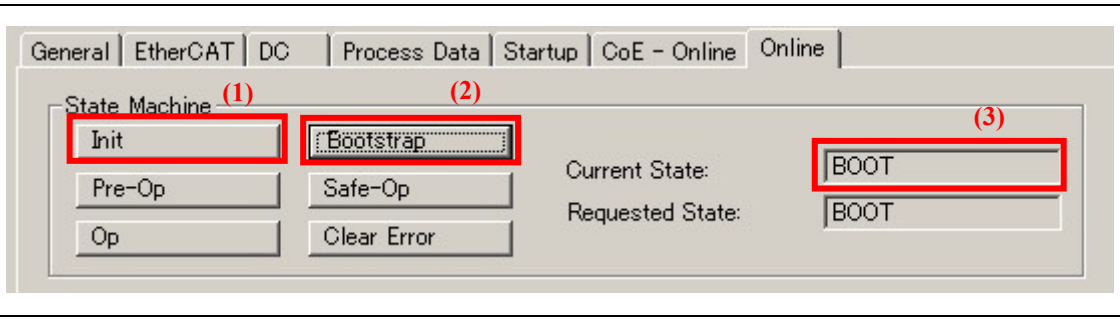

Figure 11.7 RZ/T1 “Online” Tabbed Page 2

Next, press the “Download” button in “File Access over EtherCAT”. This opens the window to select the file for downloading. Select the updating firmware file, then press “Open”.

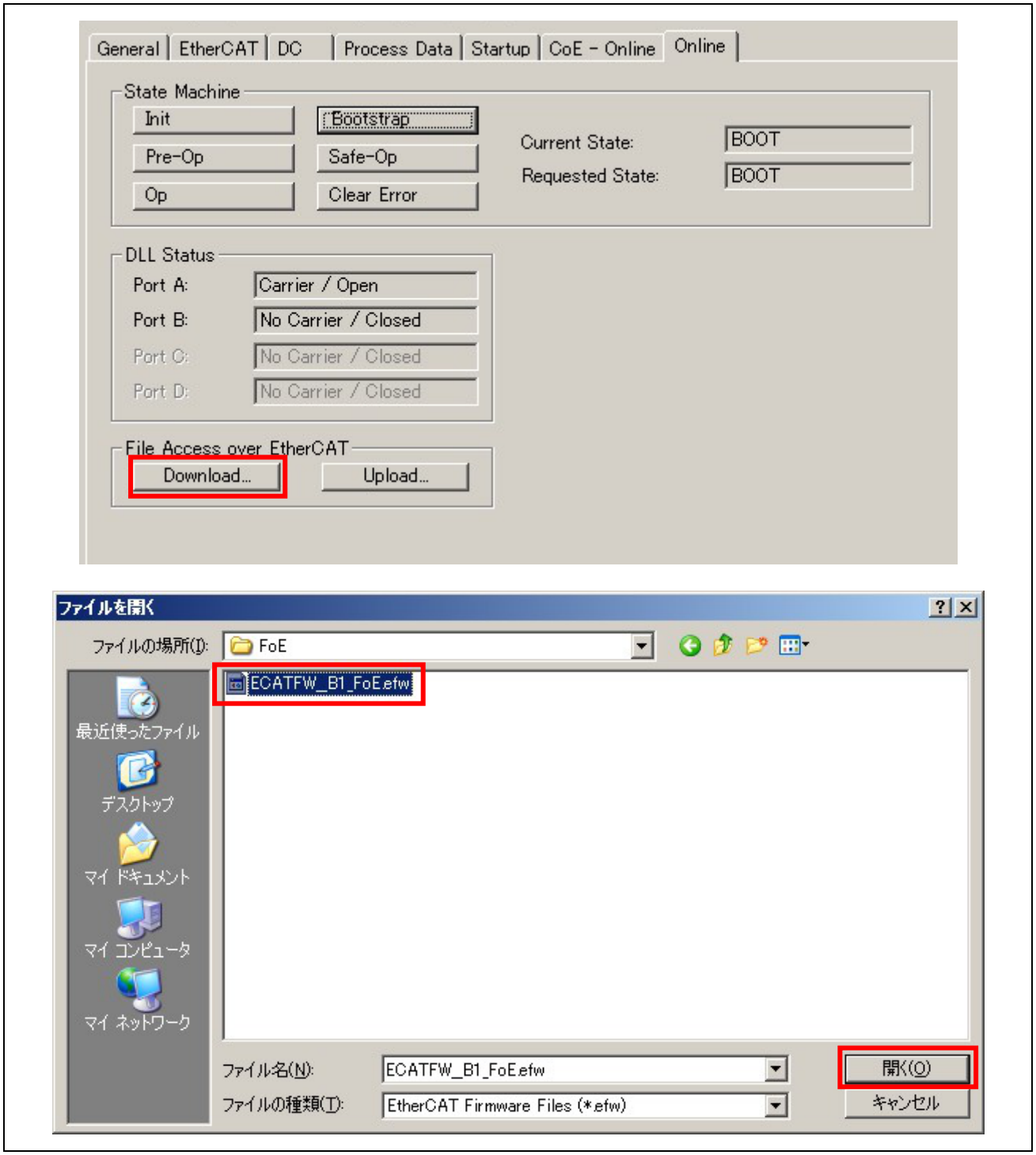

Figure 11.18 RZ/T1 Updating Firmware File Selection Window

The filename editing window opens.

Press “OK” with the password displayed as “00000000”

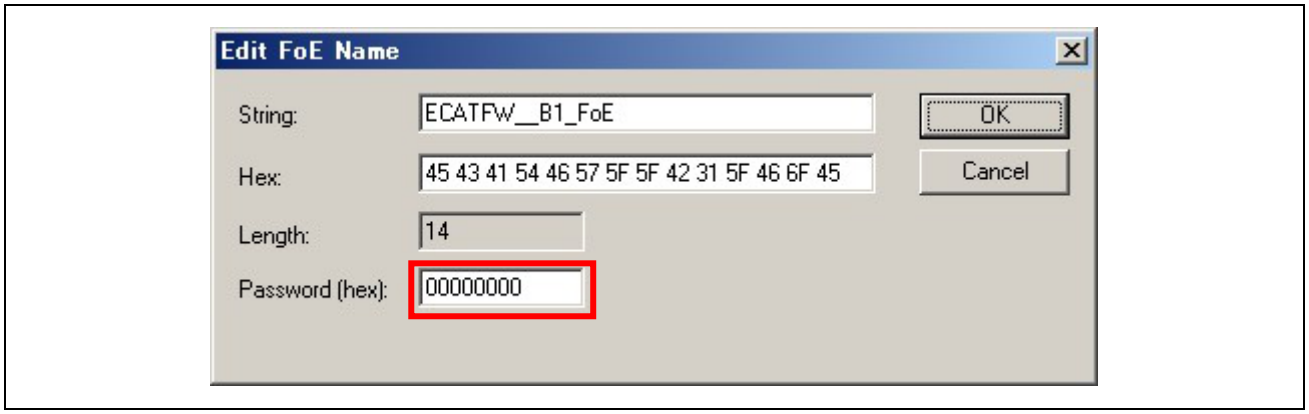

Figure 11.19 RZ/T1 Updating Firmware File Name Editing Window

The progress of downloading is displayed along with the “Downloading” message in the bottom-left part of the TwinCAT System Manager window. If no error message is displayed and the above window (figure 11.21) disappears and is replaced by “Ready”, updating of the firmware was successful.

Pressing button (1) “Init” on the “Online” tabbed page restarts the firmware with the updated version.
Pressing button (2) “Preop” then (3) “Op” changes “Current State” to “OP” and you can check the operation.

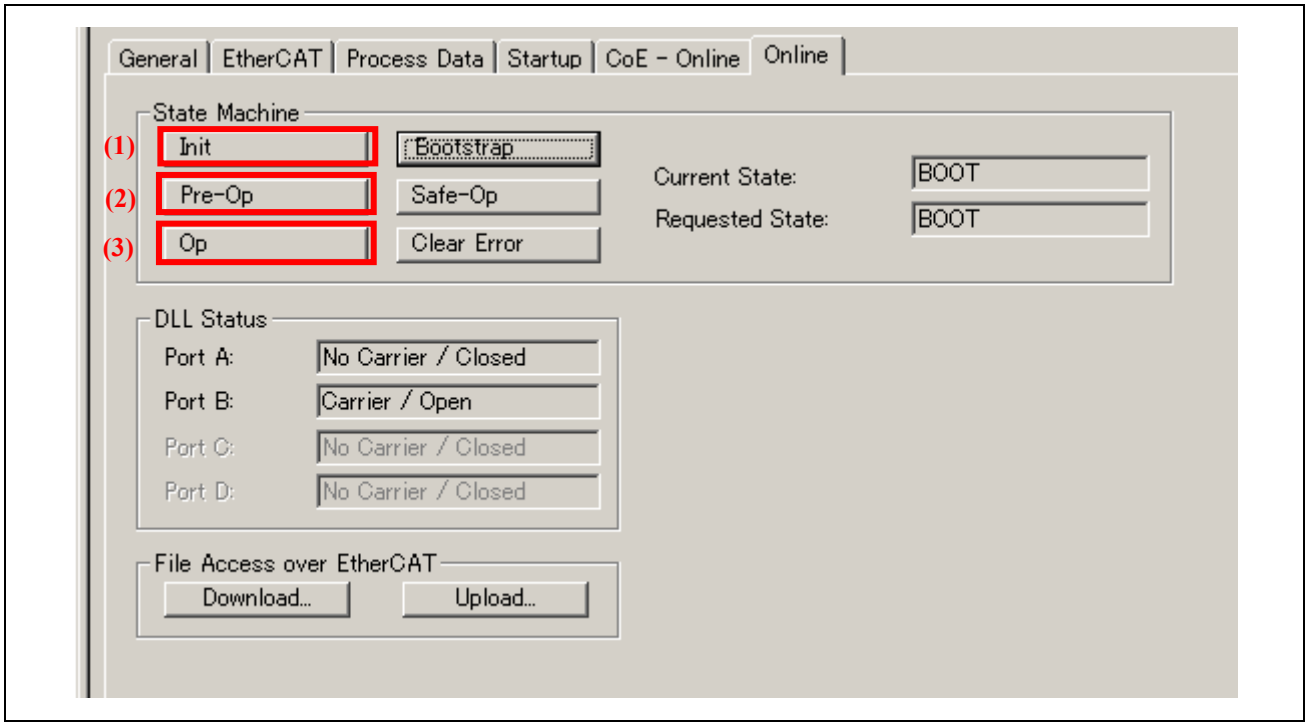

Figure 11.20 RZ/T1 “Online” Tabbed Page 3

You can check the version number of the firmware at (1) 0x100A on the “CoE - Online” page and the revision number at (2) 0x1018:03.

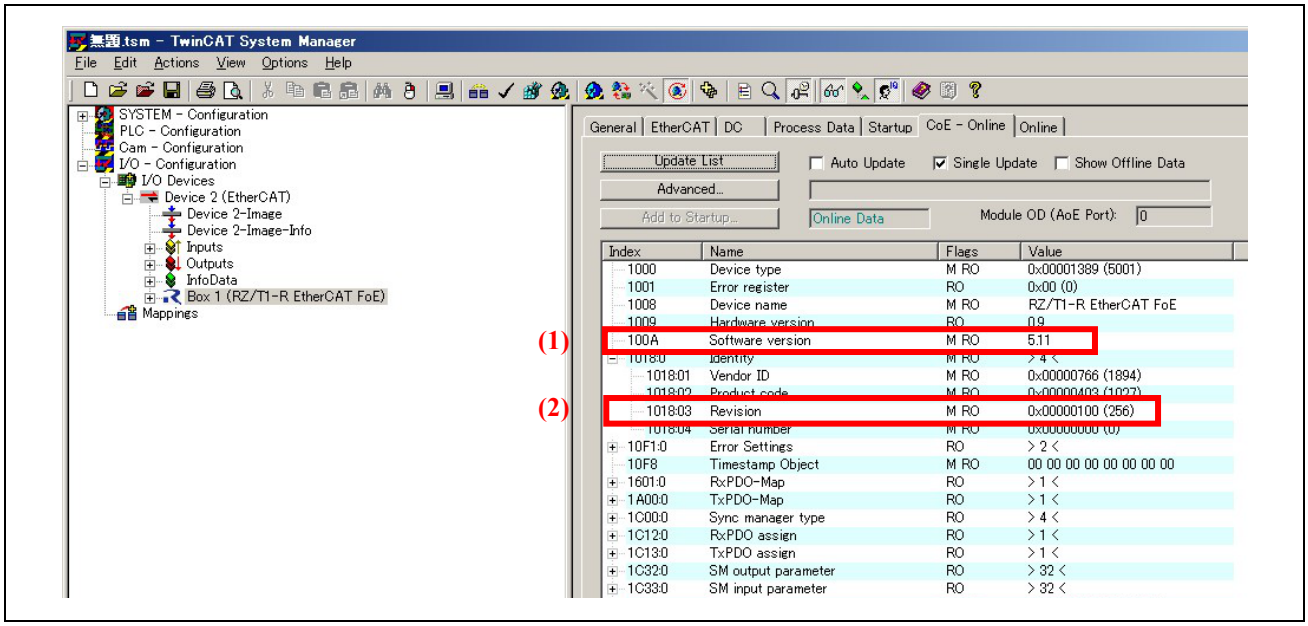

Figure 11.21 RZ/T1 “CoE” Tabbed Page

11.4 Uploading the Firmware by TwinCAT®

Select “Box 1 (RZ/T1-R EtherCAT FoE)”, then click on the “Online” tab.

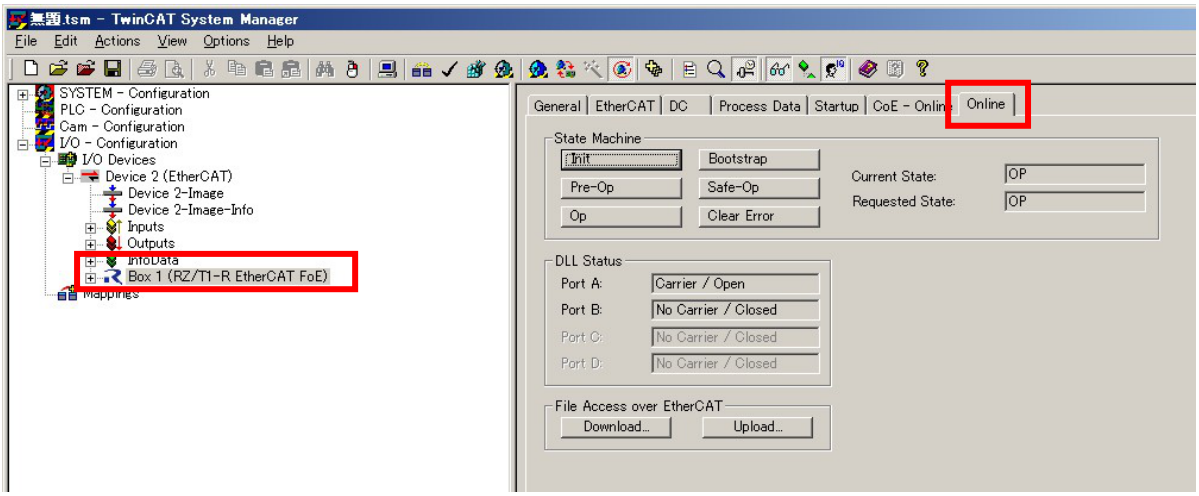

Figure 11.22 RZ/T1 “Online” Tabbed Page 1

Press button (1) “Init” then (2) “Bootstrap” in that order and confirm that “Current State” has changed to (3) “BOOT”.

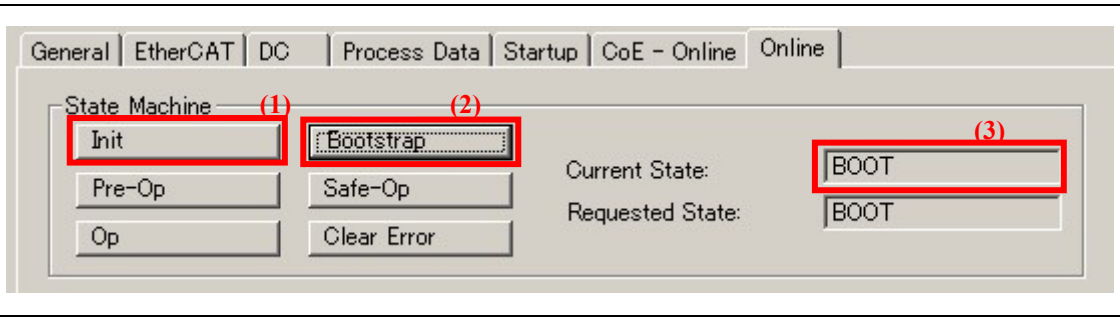

Figure 11.23 RZ/T1 “Online” Tabbed Page 2

Next, press the “Upload” button in “File Access over EtherCAT”. This opens the window to select the file for downloading. Select the updating firmware file, then press “Save”.

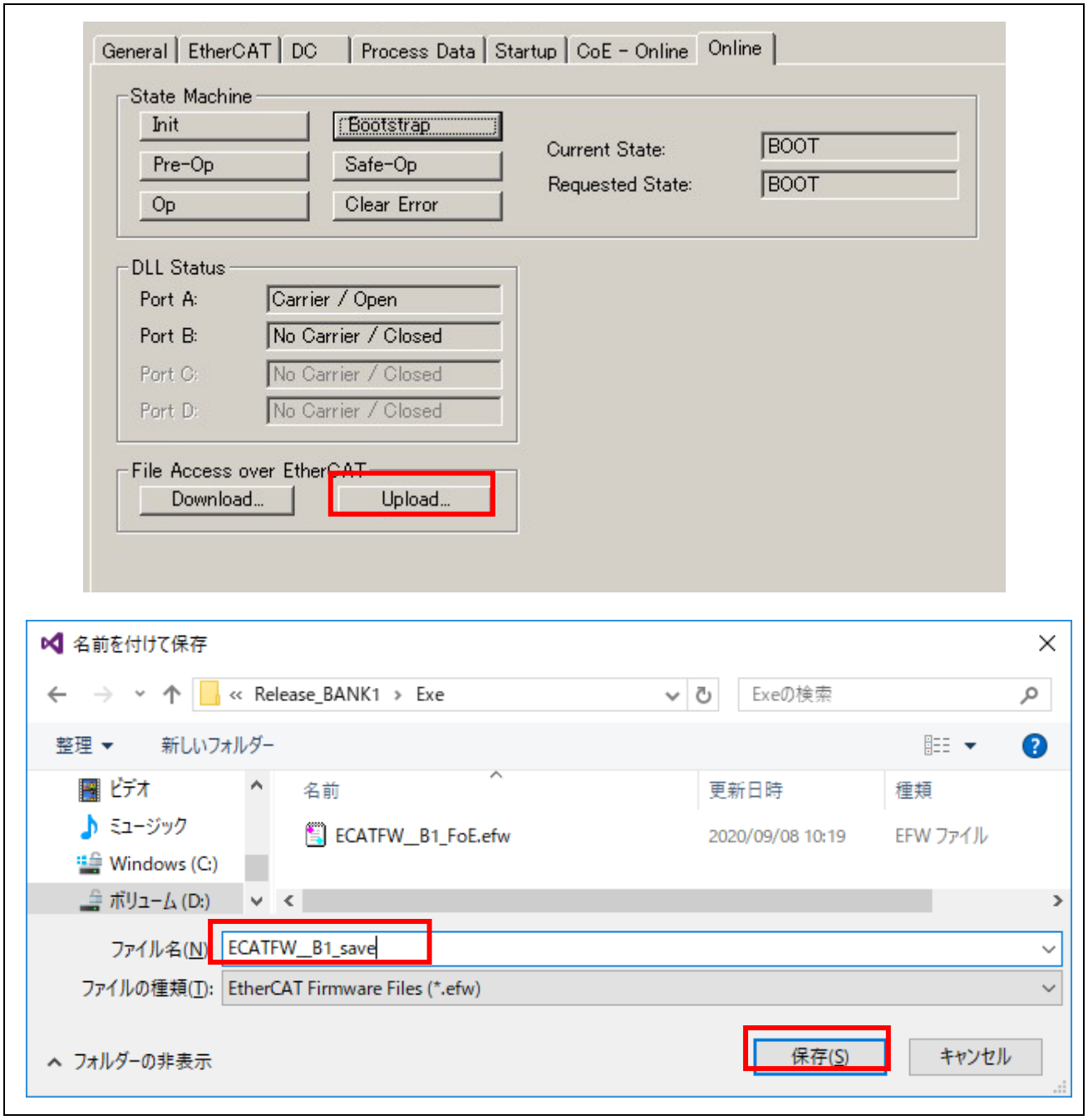

Figure 11.24 RZ/T1 Uploading Firmware File Selection Window

The filename editing window opens.

Press “OK” with the password displayed as “00000000”

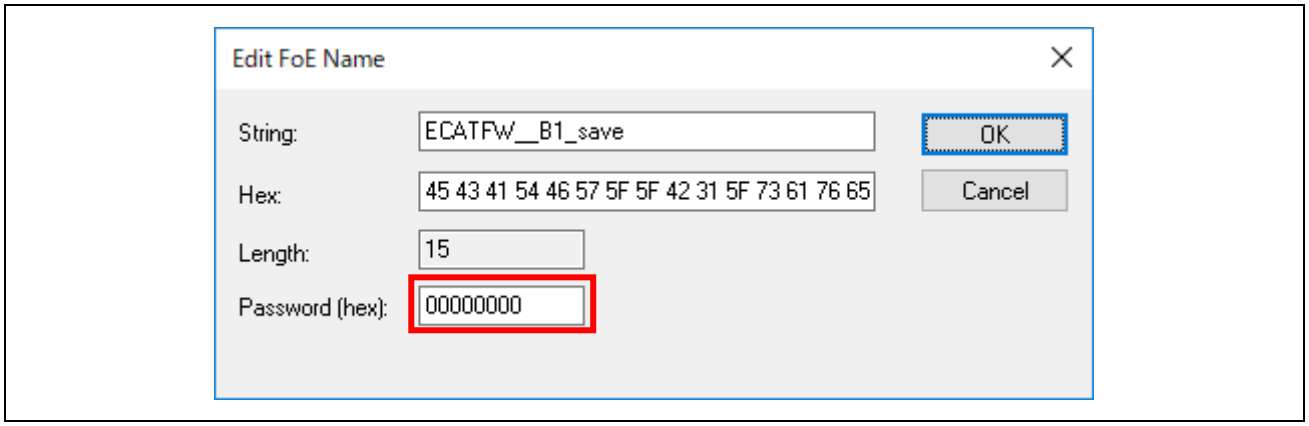

Figure 11.25 RZ/T1 Uploading Firmware File Name Editing Window

The progress of uploading is displayed along with the “Uploading” message in the bottom-left part of the TwinCAT System Manager window. If no error message is displayed and the above window (figure 11.27) disappears and is replaced by “Ready”, uploading of the firmware was successful.

Binary compare the uploading file (ECATFW_B1_save.efw) with the downloading file (ECATFW_B1_FoE.efw). You can see that they match.

12. Semiconductor Device Profiles

For a semiconductor device to be handled through EtherCAT, it must support the device profiles defined in the specifications of ETG.5003.

The configuration of ETG.5003 is as listed below.

1. Common Device Profile (CDP) [ETG.5003.1]
2. Firmware update functionality [ETG.5003.2]
3. Specific Device Profile (SDP) [ETG.5003.2xxx]

The Common Device Profile (CDP) specifies the requirements that are applicable to all semiconductor devices.
The sample program provides a framework of definitions to allow easy addition of the object dictionary of the CDP [ETG.5003.1, ver. 1.0.0]. Since we only provide the framework of definitions with this sample software, you will need to implement the necessary processing and make settings.

The framework of CDP definitions for use in adding the dictionary is as listed below.

File Name	Addition and Change
coeappl.c	For adding CDP definitions to GenObjDic[] For adding and changing address definitions and settings in the CDP
sampleappl.h	For adding CDP definitions to ApplicationObjDic[] For adding and changing address definitions and settings in the CDP
objdef.h	For changing TSYNCMANPAR definitions
ecat_def.h	For adding provisional values at 0xF9F3 and 0xF9F4
• RZ/T1 RZT1-R EtherCAT [FoE] s.xml • EC-1 EC-1 [FoE].xml	For adding and changing Datatype definitions and object settings in the CDP

Table 12.1 Files to be Changed in the Common Device Profile

Note: 0x8nn0 (0x8000) and 0xFBF0 to 0xFBF4 are provided in commented-out form.

13. Website and Support

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

Inquiries
http://www.renesas.com/inquiry

Revision History

		Description	
Rev.	Date	Page	Summary
1.00	May. 22, 2017	-	First Edition issued
2.00	Sep. 30, 2020	13	5.4 Added a procedure to copy the parameters for the BANK0 loader.
		20	Deleted the address USER_DATA_WBLOCK/USER_DATA_RBLOCK
		30,41	Fixed file and symbol names in the source file support area.
		4	表 1-1 サポートする FoE サービスにファイル読み出しを追加
		52-54	11.5 TwinCAT による更新ファームウェア読み出しを追加
		27-30	11.1 Debugger launch changed to Debugger launch (IAR EWARM)
		55	12. Changed the description of Common Device Profile (ETG5003.1)
		13	5.4 Added procedure to copy parameters for BANK0 loader

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.