RX Family

Sample Program for Open Source FAT File System Library M3S-TFAT-Tiny Module with Serial NOR Flash Memory

Abstract

This application note describes how to use the sample program for the open source FAT file system library M3S-TFAT-Tiny module with serial NOR flash memory. Hereinafter, the "M3S-TFAT-Tiny module" is referred to as the "TFAT library".

Target device

RX72N Group

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. Outline 3
1.1 About This Application Note 3
1.2 Environment in Which Operation Was Verified 3
1.3 Configuration of This Application Note 3
1.4 Modules 4
2. Hardware Description 5
2.1 Hardware Configuration 5
3. Executing the Sample Program 5
3.1 Launching the Workspace 5
3.2 Executing the Sample Program and Verifying Its Operation 9
4. Sample Program 14
4.1 Overview 14
4.2 Flowchart 14
4.3 Module Initialization 15
4.3.1 Flowchart 15
4.3.2 System Timer Initialization 15
4.3.3 Device Driver Initialization 15
4.3.4 Middleware Initialization 15
4.4 File Manipulation 16
4.4.1 Flowchart 16
4.4.2 Formatting Process 16
4.4.3 File Manipulation Processes 17
4.5 Memory Usage 17

5. Board-Dependent Settings 18
5.1 Modifying the Settings 18
5.2 Changes to the Initial Settings of Smart Configurator 20
5.2.1 BSP Module 20
5.2.2 Memory Access Driver Interface Module 20
5.2.3 Clock Synchronous Control Module for Serial NOR Flash Memory Access 21
6. Migration to CS+ 23
7. Notes 24
7.1 About the Formatting Process 24
7.2 Timer Function for Serial NOR Flash Memory 24
7.3 About the Stack and Heap Sizes 24
8. Reference Documents 25
Revision History 26

1. Outline

1.1 About This Application Note

This application note describes how to use the TFAT library with serial NOR flash memory. In this application note, you create a FAT file system work area in serial NOR flash memory by using the TFAT library, and then write files to, and read files from, the area.

This application note is bundled with a sample program that is compatible with the RX72N Envision Kit. If you want to use another device, refer to "5. Board-Dependent Settings".

1.2 Environment in Which Operation Was Verified

Operation of the sample program bundled with this application note was verified under the conditions shown below.

Table 1.1 Operating Environment

Item	Description
MCU used	R5F572NNHDFB (RX72N Group)
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics e^2 studio 2024-01
C compiler	Renesas Electronics C/C++ Compiler for RX Family V3.06.00
Compiler option	The "-lang=c99" and “-optimize=0” options were added with the default settings specified in the integrated development environment.
Endianness	Little endian
Operating mode	Single-chip mode
Sample program version	Rev.1.00
Board used	RX72N Envision Kit (model: RTK5RX72N0C00000BJ)
Serial NOR flash memory used	Macronix MX25L3233F (32 megabits)

1.3 Configuration of This Application Note

This application note consists of the items listed below.

Table 1.2 Configuration of This Application Note

Item		Description
r01an7187jj0100_rx.pdf		Application note document (Japanese version)
r01an7187ej0100_rx.pdf		Application note document (English version)
workspace		
	TFAT_sample_RX72N_with_Flash	Sample program for the RX72N (compatible with e^2 studio)[*1]

Note: 1. The e^2 studio project can be converted to a CS+ project.

1.4 Modules

Table 1.3 lists the FIT modules used in the sample program that is bundled with this application note. These FIT modules are provided by RX Driver Package v1.42.

Table 1.3 List of FIT Modules Used

Category	Application Note (Document Number)	FIT Module	Rev.
BSP	RX Family Board Support Package Module Using Firmware Integration Technology (R01AN1685)	r_bsp	7.42
Device driver	RX Family CMT Module Using Firmware Integration Technology (R01AN1856)	r_cmt_rx	5.60
	RX Family QSPI Clock Synchronous Single Master Control Module Using Firmware Integration Technology (R01AN1940)	r_qspi_smstr_rx	1.21
	RX Family Clock Synchronous Control Module for Serial NOR Flash Memory Access Firmware Integration Technology (R01AN2662)	r_flash_spi	3.30
Middleware	RX Family System Timer Module Firmware Integration Technology (R20AN0431)	r_sys_time_rx	1.01
	RX Family Memory Access Driver Interface Module Using Firmware Integration Technology (R01AN4548)	r_memdrv_rx	1.20
	RX Family M3S-TFAT-Tiny Memory Driver Interface Module Firmware Integration Technology (R20AN0335)	r_tfat_driver_rx	2.30
	RX Family Open Source FAT File System M3S-TFAT-Tiny Module Firmware Integration Technology (R20AN0038)	r_tfat_rx	4.10

2. Hardware Description

2.1 Hardware Configuration

The target board of this application note is equipped with Macronix serial NOR flash memory. Table 2.1 shows the pin connections between the MCU on the target board and the serial NOR flash memory.

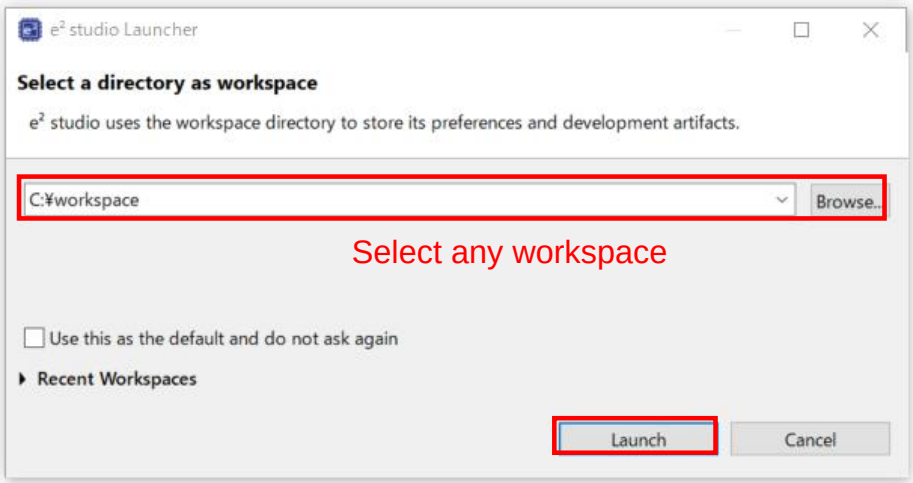
Table 2.1 Pin Connections

Pin Name/Signal	Input/Output	Description
PD4/QSSL-B	Output	Chip Select
PD5/QSPCLK-B	Output	Clock
PD6/QIO0-B	Input/output	Bidirectional Data 0
PD7/QIO1-B	Input/output	Bidirectional Data 1
PD2/QIO2-B	Input/output	Bidirectional Data 2
PD3/QIO3-B	Input/output	Bidirectional Data 3

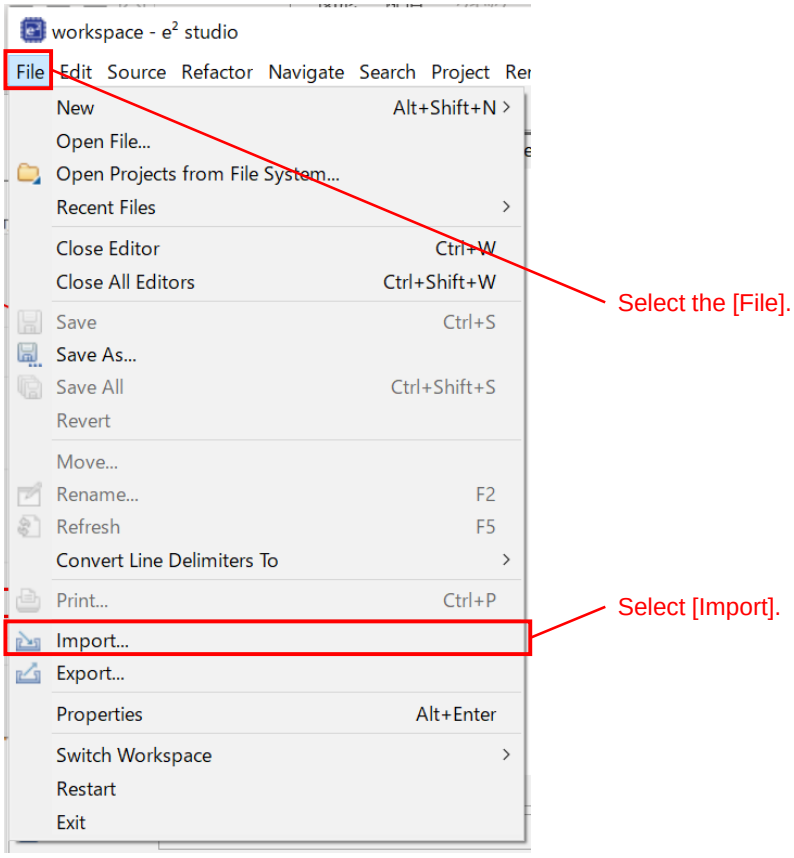
3. Executing the Sample Program

3.1 Launching the Workspace

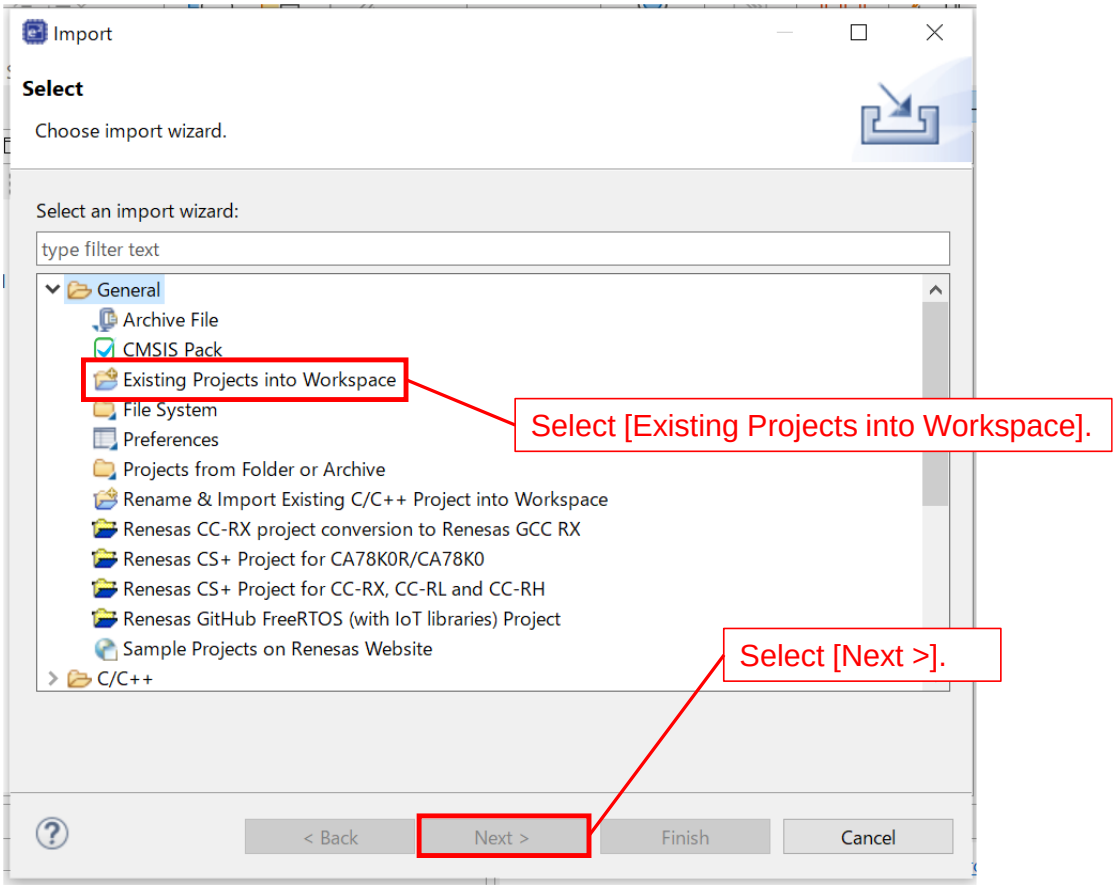
1. Start e² studio. The dialog box for selecting the workspace opens. If the dialog box does not open, from the [File] menu, select [Switch Workspace] and then [Others].
2. Select any workspace of your choice, and then click [Launch].



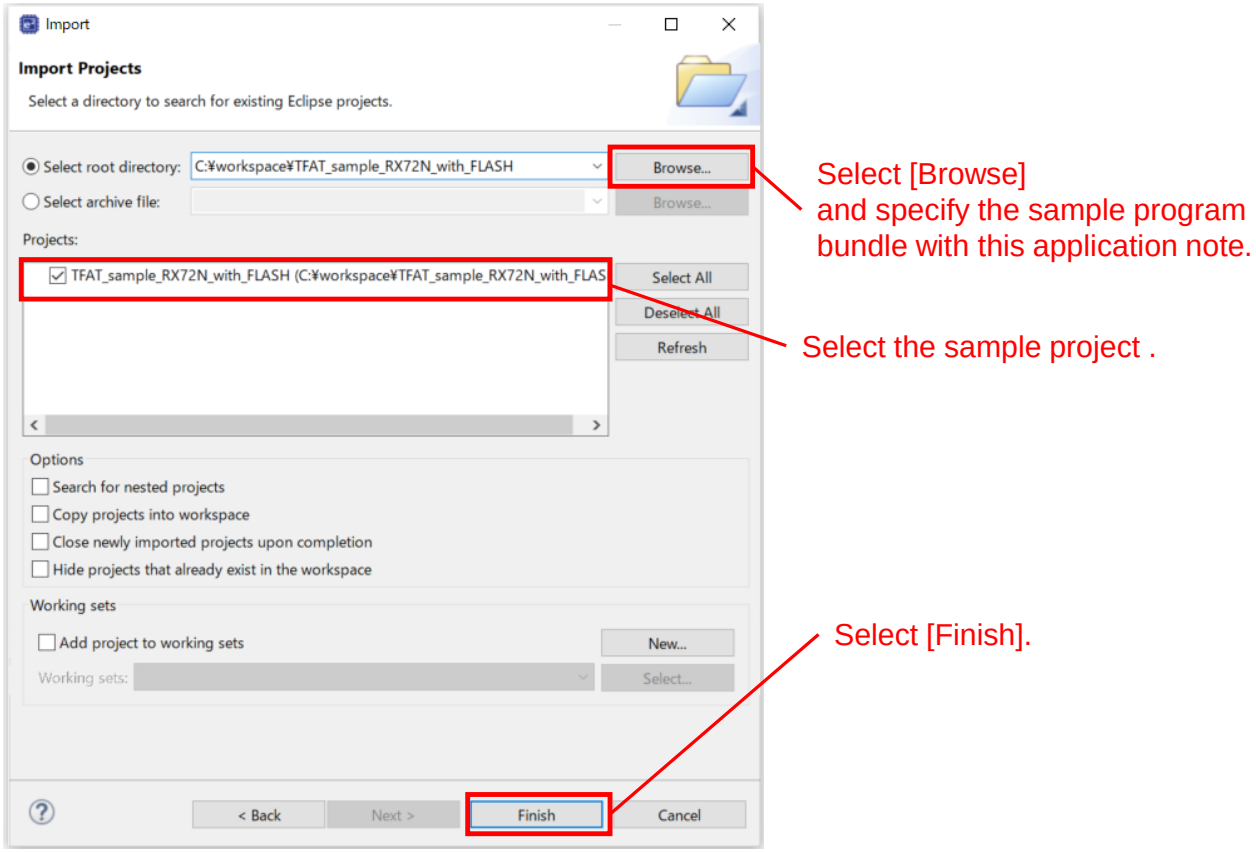
3. From the [File] menu, select [Import].



4. Under the [General] node, select [Existing Projects into Workspace], and then select [Next].



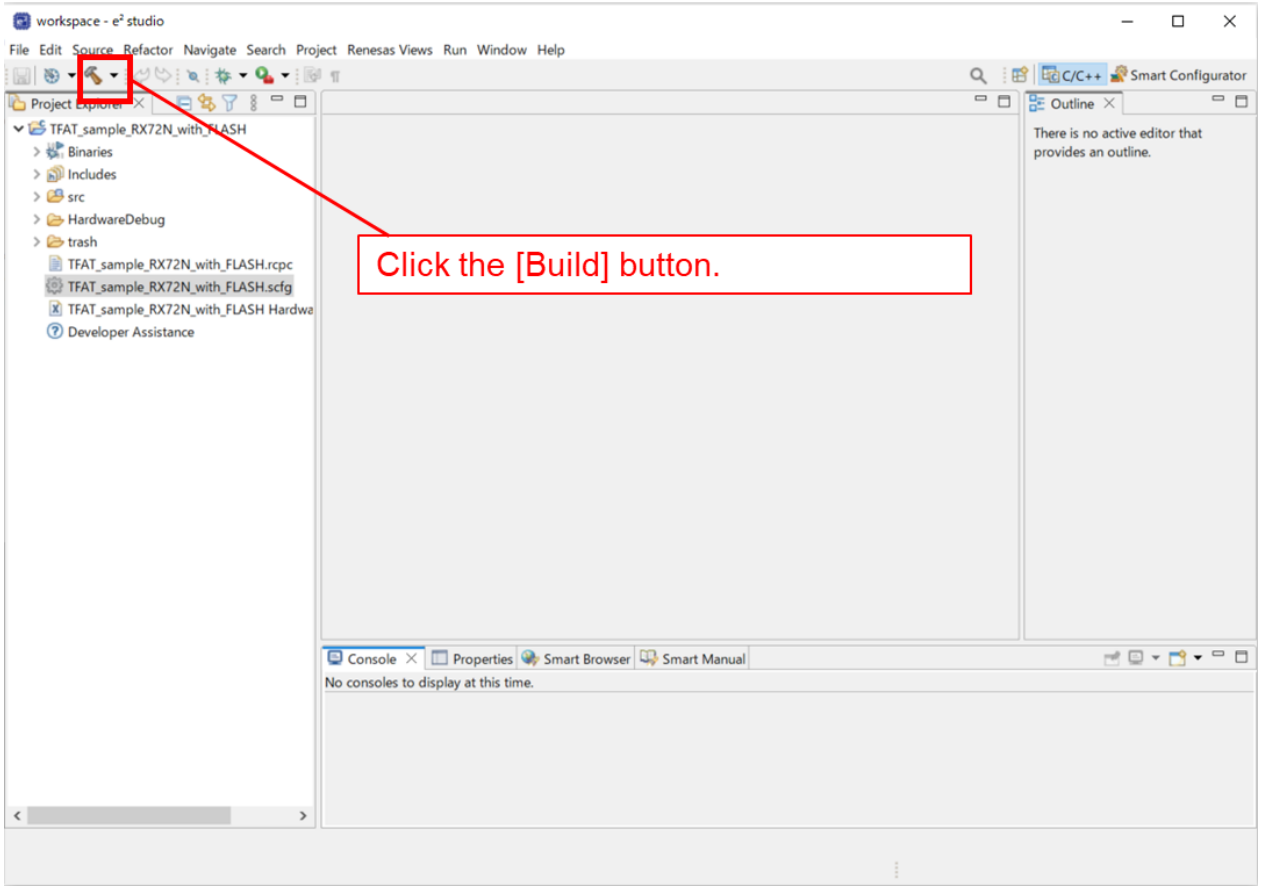
5. Select [Browse] and specify the sample program bundled with this application note.
6. Select [Finish].



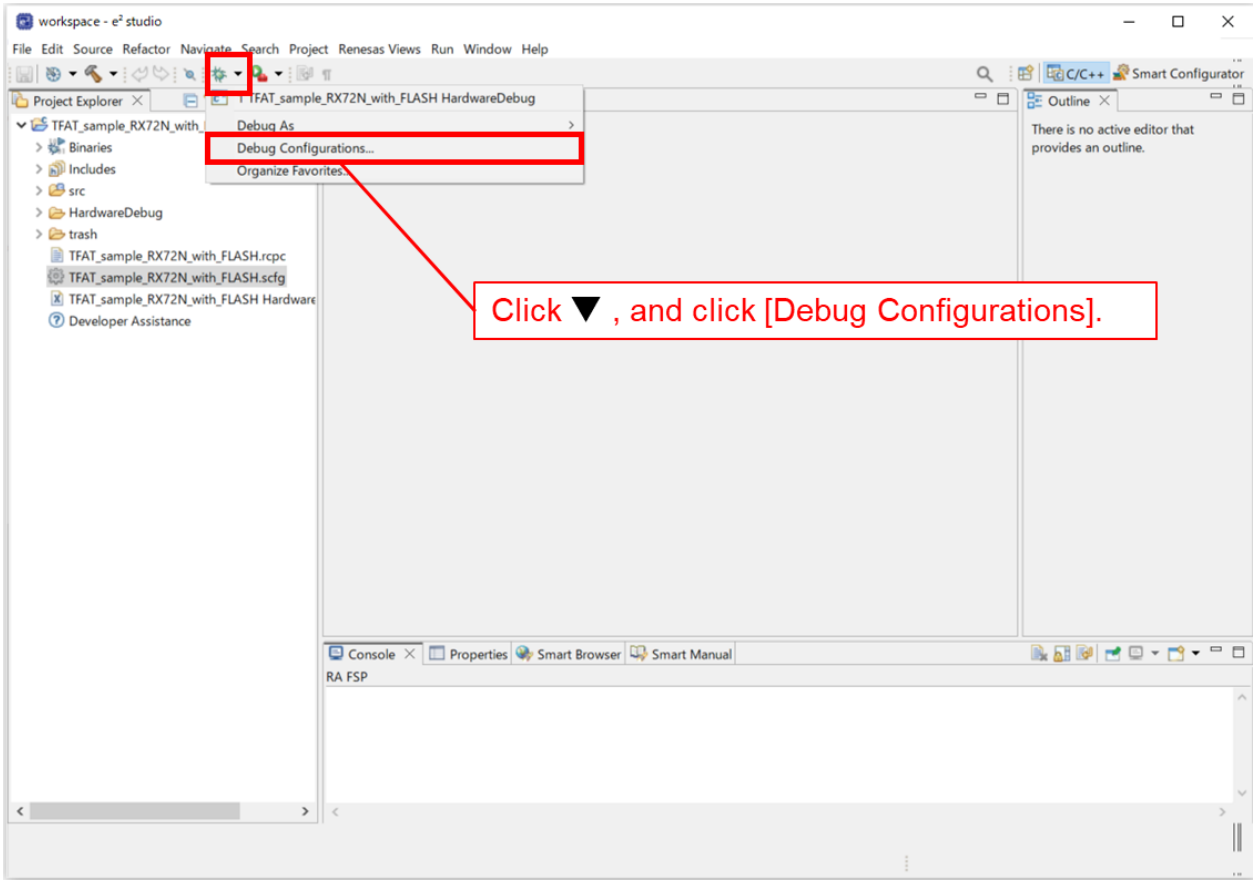
3.2 Executing the Sample Program and Verifying Its Operation

This section describes how to execute the sample program that you imported and how to verify its operation.

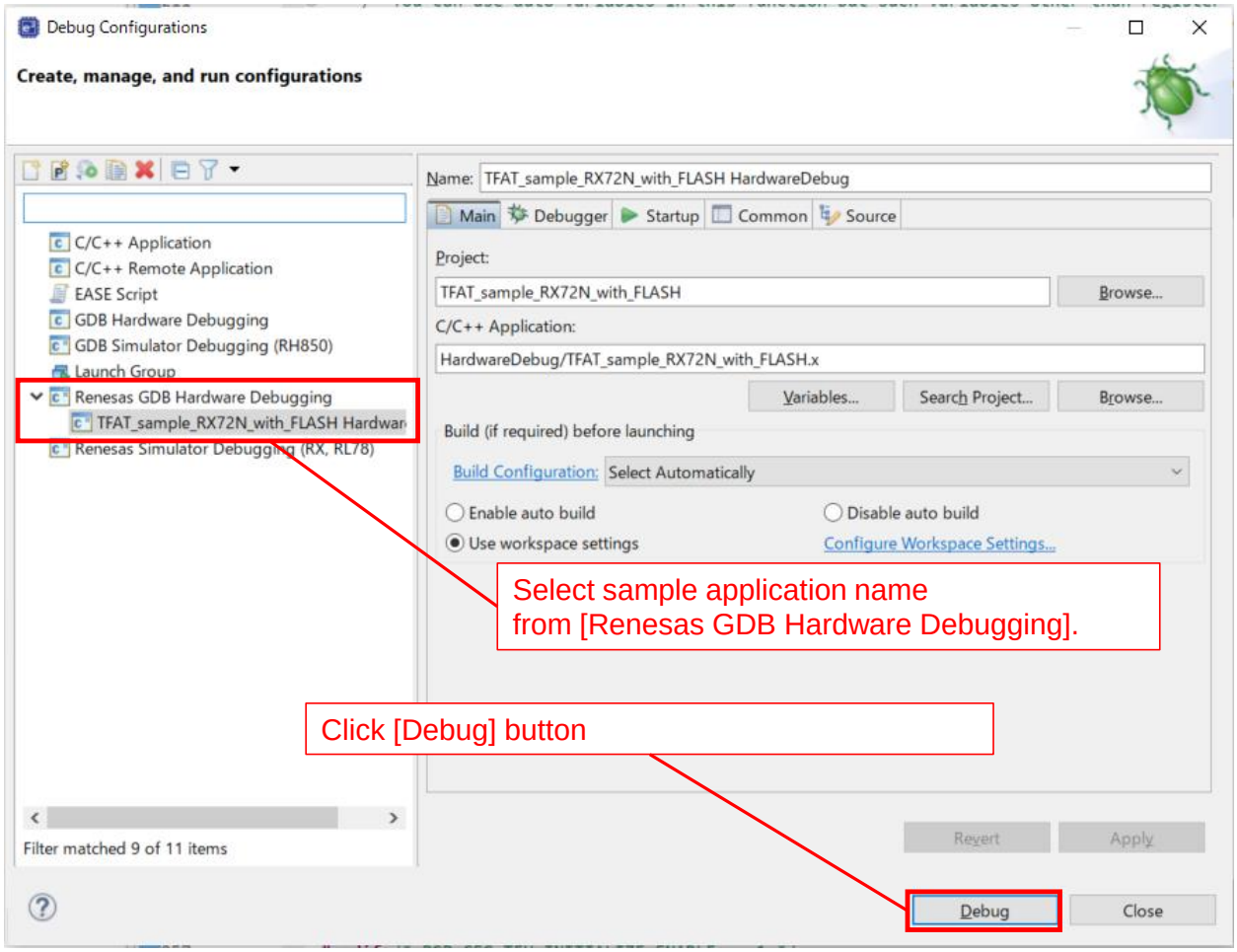
1. Select the Build icon to build the imported sample program.



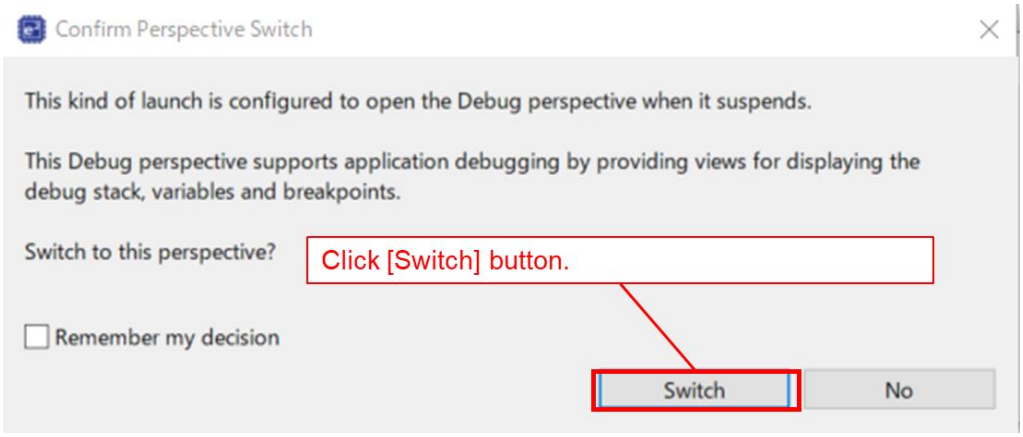
2. When the building is complete, perform debugging to verify operation. To perform debugging, open [Debug Configurations].



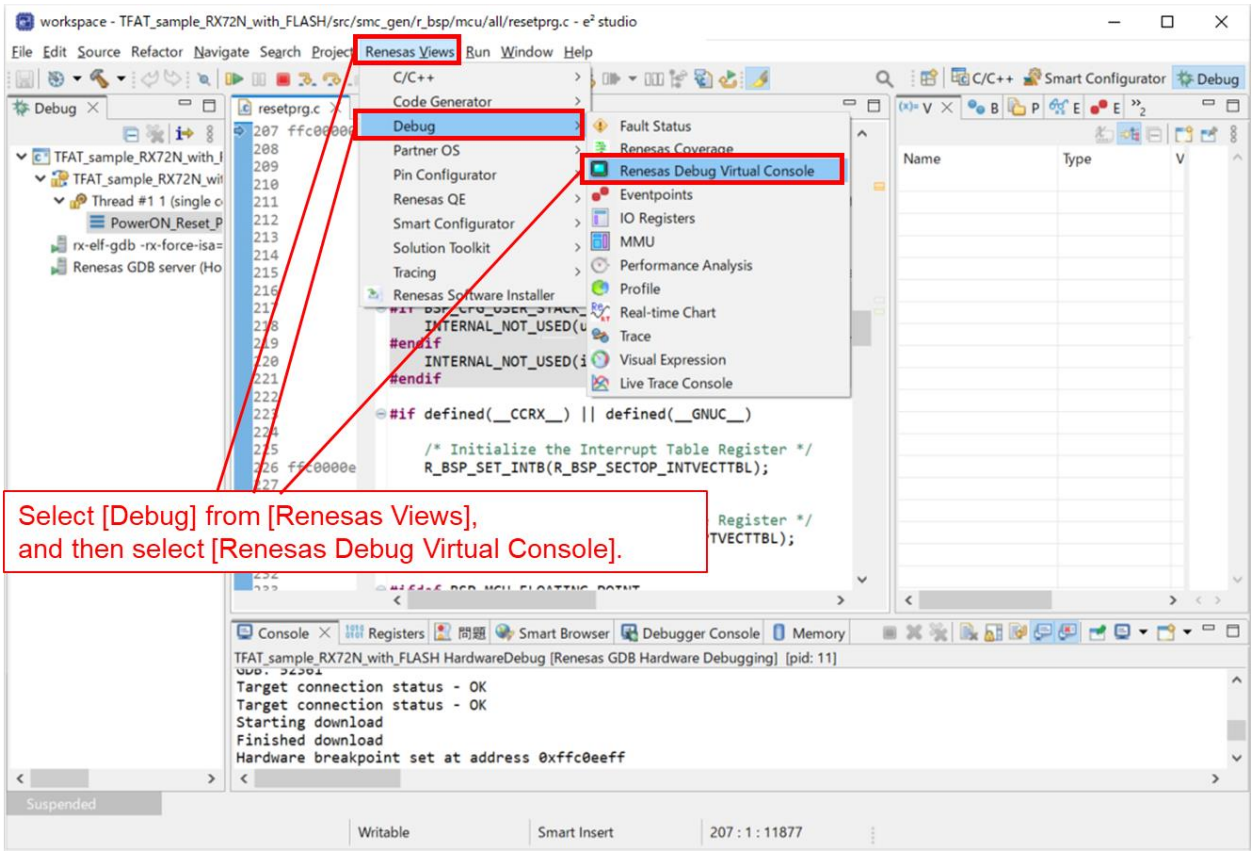
3. Open the debug configurations for the sample program that you built, and then start debugging it.



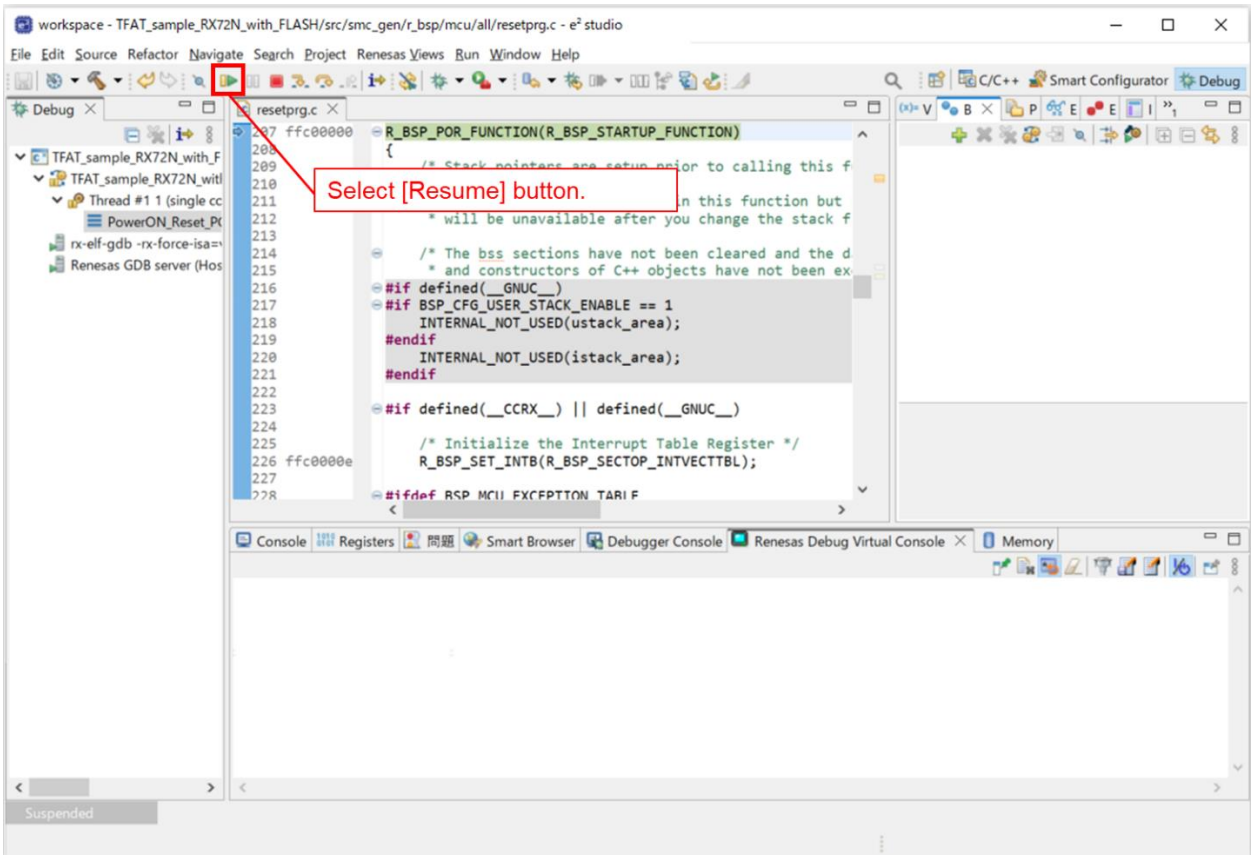
4. At this time, e^2 studio displays a pop-up window that confirms whether to switch the perspective. Switch it if necessary.



5. Open Renesas Debug Virtual Console so that you can view the operation results.



6. Execute the sample program.



7. If the program execution was successful, the following messages are displayed in Renesas Debug Virtual Console:

```
Start TFAT sample.
Start FAT format.
*** Scan (Tree Mode) ***
0:
|-FLD/
| |-TEXT.TXT
| |-
|-
*** End of Scan ***
Finished TFAT sample.
```

If the following message is displayed, you must format the serial NOR flash memory:
`Error (FR_NO_FILESYSTEM) Drive mount`
Format the serial NOR flash memory by referring to "4.4.2 Formatting Process" and "7.1 About the Formatting Process".

4. Sample Program

4.1 Overview

The sample program is an e^2 studio project that is compatible with the target board shown in Table 1.1 Operating Environment.

4.2 Flowchart

Figure 4.1 shows the flowchart of the "main" function of the sample program.

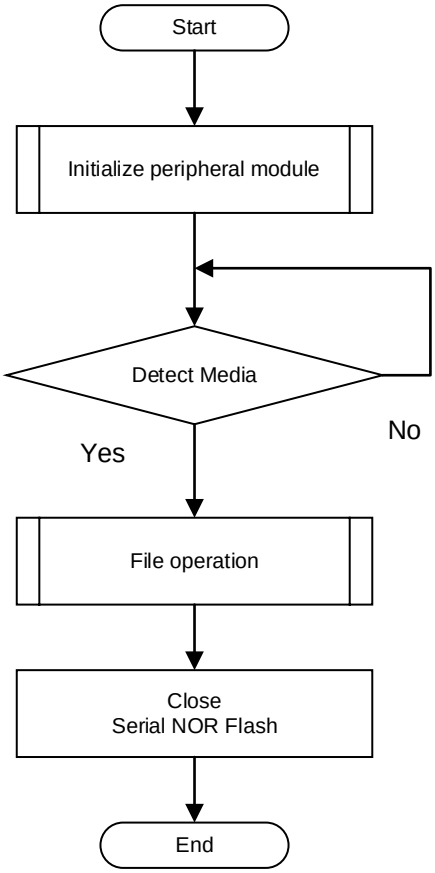


Figure 4.1 Flowchart of the "main" Function

4.3 Module Initialization

Before you can use serial NOR flash memory with the TFAT library, you must initialize the following modules:
• r_sys_time_rx
• r_flash_spi
• r_tfat_driver_rx

You must also initialize the settings of the pins that are connected to the serial NOR flash memory.

4.3.1 Flowchart

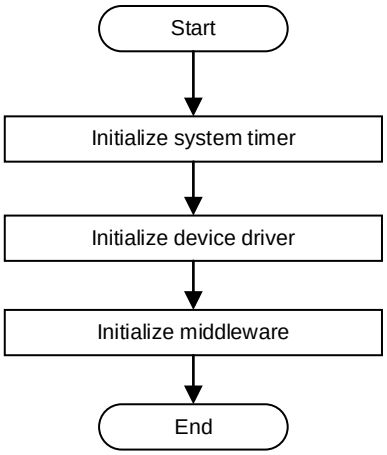


Figure 4.2 Flowchart of the Module Initialization Processes

4.3.2 System Timer Initialization

The system timer initialization process initializes the r_sys_time_rx module. This module is used to set the file update time.

4.3.3 Device Driver Initialization

The device driver initialization process initializes the r_flash_spi module. This module configures the communication with the serial NOR flash memory. It also registers the callback function that checks whether the flash memory is busy as a periodic timer callback for the r_sys_time_rx module at an interval of 10 ms.

4.3.4 Middleware Initialization

The middleware initialization process initializes the r_tfat_driver_rx module. This module also registers its own callback function that checks whether the flash memory is busy as a periodic timer callback for the r_cmt_rx module at an interval of 1 ms. For more detailed information, refer to "7.2 Timer Function for Serial NOR Flash Memory".

4.4 File Manipulation

4.4.1 Flowchart

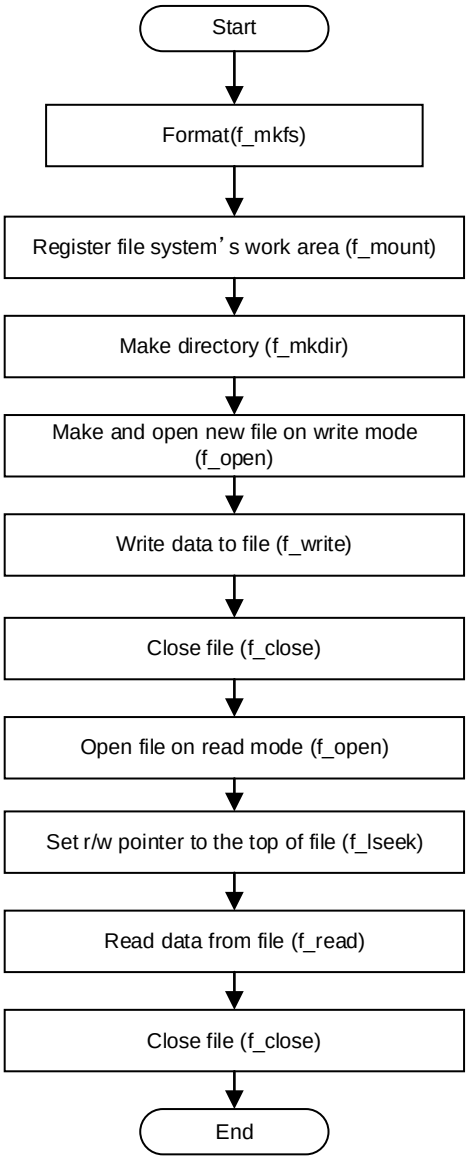


Figure 4.3 Flowchart of File Manipulation Processes

4.4.2 Formatting Process

The open source FAT file system FatFs provides several optional settings that are defined in "ffconf.h". You can change the values of these settings to customize the API functions and FatFs features that are available to you. You cannot change these settings by using Smart Configurator.

These FatFs settings include the FF_USE_MKFS definition shown in Table 4.1 Formatting Option. This definition is an option to enable the f_mkfs function that performs the formatting process. If this option is set to 1, the TFAT library gives the users access to the f_mkfs function. In the sample program, when the formatting process is started, the f_mkfs function is called by changing the value of this option to 1.

Note that the formatting process erases the data existing on the serial NOR flash memory. Before the process starts, confirm that no important data is contained in the memory. For more detailed information about the formatting process, refer to "7.1 About the Formatting Process".

Table 4.1 Formatting Option

Definition Name	Possible Value	Value Meaning	Default
FF_USE_MKFS	“1”	Enables the f_mkfs function.	“0”
	“0”	Disables the f_mkfs function.	

4.4.3 File Manipulation Processes

After the formatting process (f_mkfs) is executed, file manipulation processes are sequentially executed as shown in Figure 4.3 Flowchart of File Manipulation Processes.

In these file manipulation processes, the following operations are performed:

1. Register the work area in the FAT file system (f_mount)
2. Create a folder named "FLD" in the root directory (f_mkdir)
3. Create a file named "TEXT.TXT" in the "FLD" folder and open the file in write mode (f_open)
4. Write text data of 2,048 bytes to the "TEXT.TXT" file (f_write)
5. Close the "TEXT.TXT" file (f_close)
6. Reopen the "TEXT.TXT" file in read mode (f_open)
7. Move the file pointer to the start address of the "TEXT.TXT" file (f_lseek)
8. Read the data in the file in units of 512 bytes from the file beginning and check whether each unit of data is identical to the written data (f_read)
9. Repeat step 8 to check whether all 2,048 bytes are identical to the 2,048-byte data that was written in step 4
10. Close the "TEXT.TXT" file (f_close)

The data to be written to the file is defined in "r_data_file.c". By default, repetitions of the string "Renesas," are written. You can change the data to be written and the relevant macro FILESIZE corresponding to the data size in “r_data_file.h”, if necessary.

4.5 Memory Usage

This section shows the ROM size, RAM size, and the maximum stack size used by the sample program of this application note. "Table 4.2 Confirmation Conditions" shows the conditions under which memory usage was confirmed. "1.4 Modules" shows the modules that were used for confirmation and their revision numbers. "Table 4.3 Memory Usage" shows the confirmation results.

Table 4.2 Confirmation Conditions

Condition	Description
Integrated development environment	Renesas Electronics e^2 studio 2023-10
C compiler	Renesas Electronics C/C++ Compiler for RX Family V3.06.00
Endianness	Little endian
Compilation option	The "-lang=c99" and “-optimize=0” options were added with the default settings specified in the integrated development environment.

Table 4.3 Memory Usage

Item	Memory Usage
ROM	69195 byte
RAM	48570 byte
Stack	5276 byte

5. Board-Dependent Settings

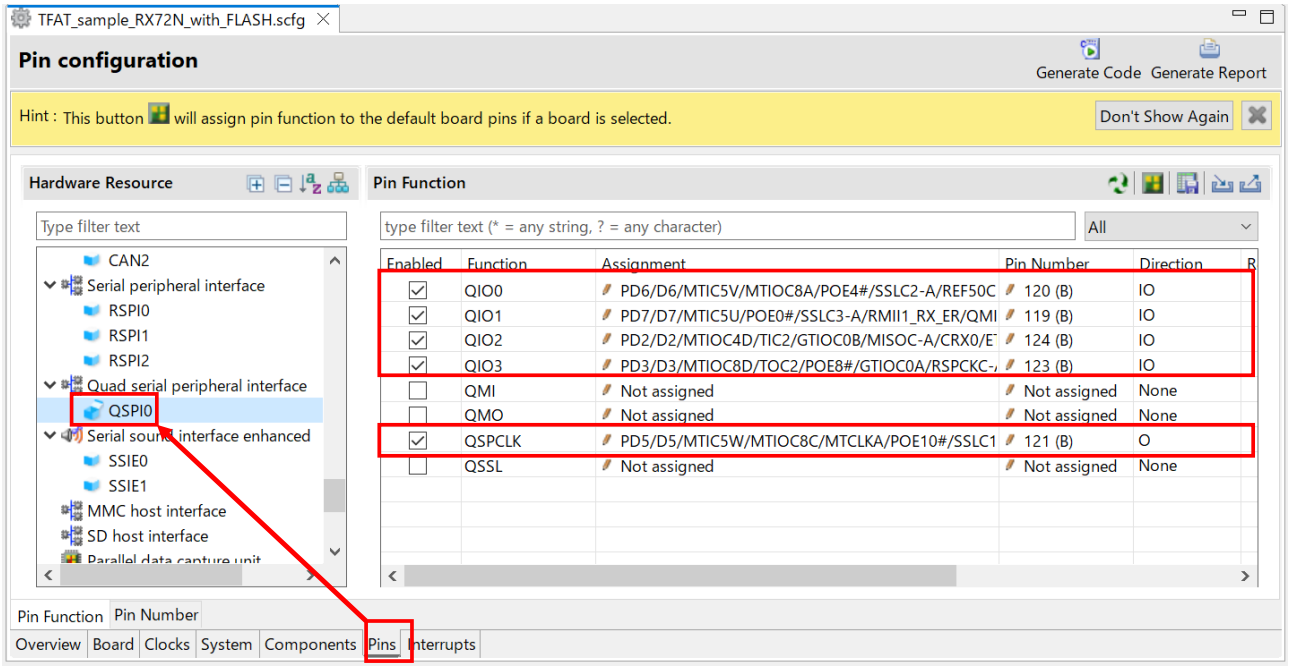
5.1 Modifying the Settings

This section describes how to modify the sample program of this application note if you use the sample program for a board other than the target board (RX72N Envision Kit).

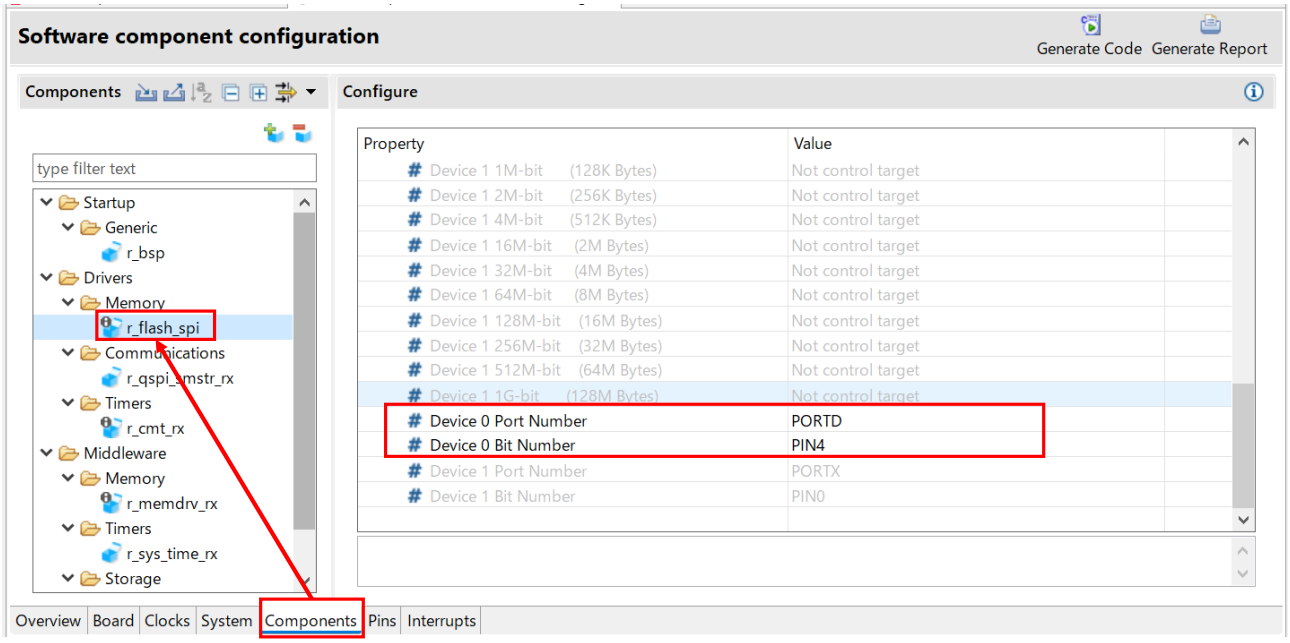
In the sample program, the sections that are dependent on the target board are the settings of pins that are used for connections between the Macronix serial NOR flash memory and the MCU mounted on the board. For details about the pins that are used for connection between the serial NOR flash memory and MCU on the target board, refer to "2.1 Hardware Configuration".

The settings that are dependent on the target board and need to be modified are as follows:

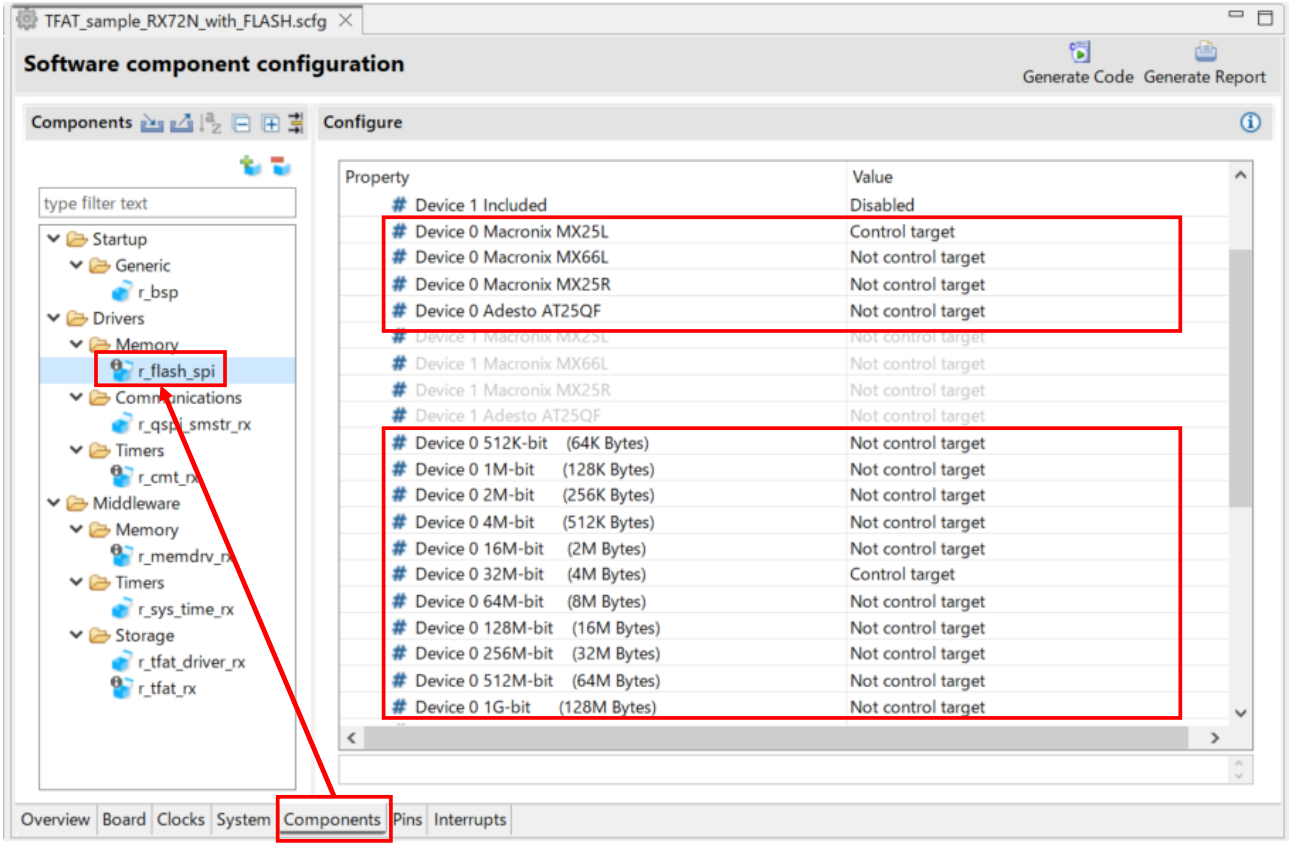
1. The pin settings of the quad serial peripheral interface (QSPI). Change the pin settings according to the board that you want to use. In Smart Configurator, open the [Pins] page, and then select [QSPI]. You can then access the relevant QSPI pin settings.



2. The SS# pin settings of the clock synchronous control module for serial NOR flash memory access (r_flash_spi). Change the settings to the MCU pins (on the board that you want to use) to which the SS# pins of the serial NOR flash memory are connected. In Smart Configurator, open the [Components] page, and then select [r_flash_spi]. In the properties list, [Device 0 Port Number] and [Device 0 Bit Number] are the relevant settings.



3. The control-target serial NOR Flash memory settings of the clock synchronous control module for serial NOR flash memory access (r_flash_spi). Set the serial NOR flash memory device that you use as the control target. In Smart Configurator, open the [Components] page, and then select [r_flash_spi]. In the properties list, find the model name and memory size of the Macronix serial NOR flash memory device that you use. Make sure that "Control target" is displayed for only the model name and memory size of the flash memory device that you use.



5.2 Changes to the Initial Settings of Smart Configurator

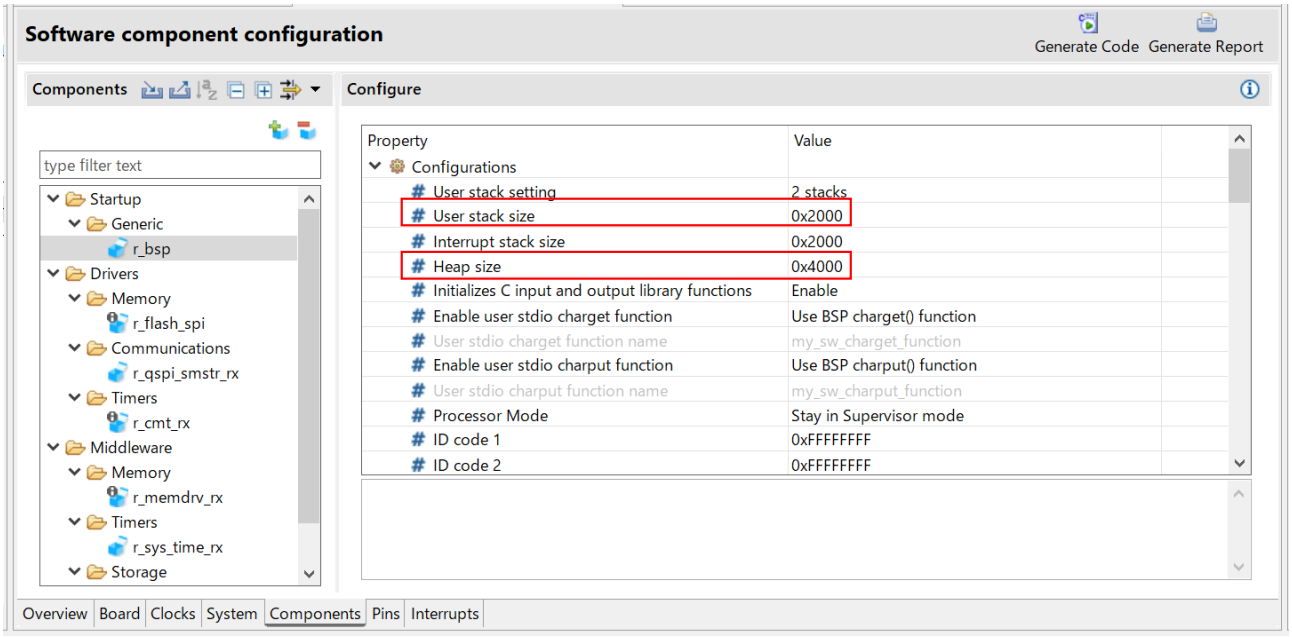
This section shows the Smart Configurator settings that have been changed from their initial values for use with the sample program of this application note.

5.2.1 BSP Module

Table 5.1 shows the relevant settings of the BSP module and their values that are currently set. For details about the "User stack size" and "Heap size" settings, refer to "7.3 About the Stack and Heap Sizes".

Table 5.1 Changes to the "User stack size" and "Heap size" Settings

Item	Current Value
User stack size	0x2000
Heap size	0x1000

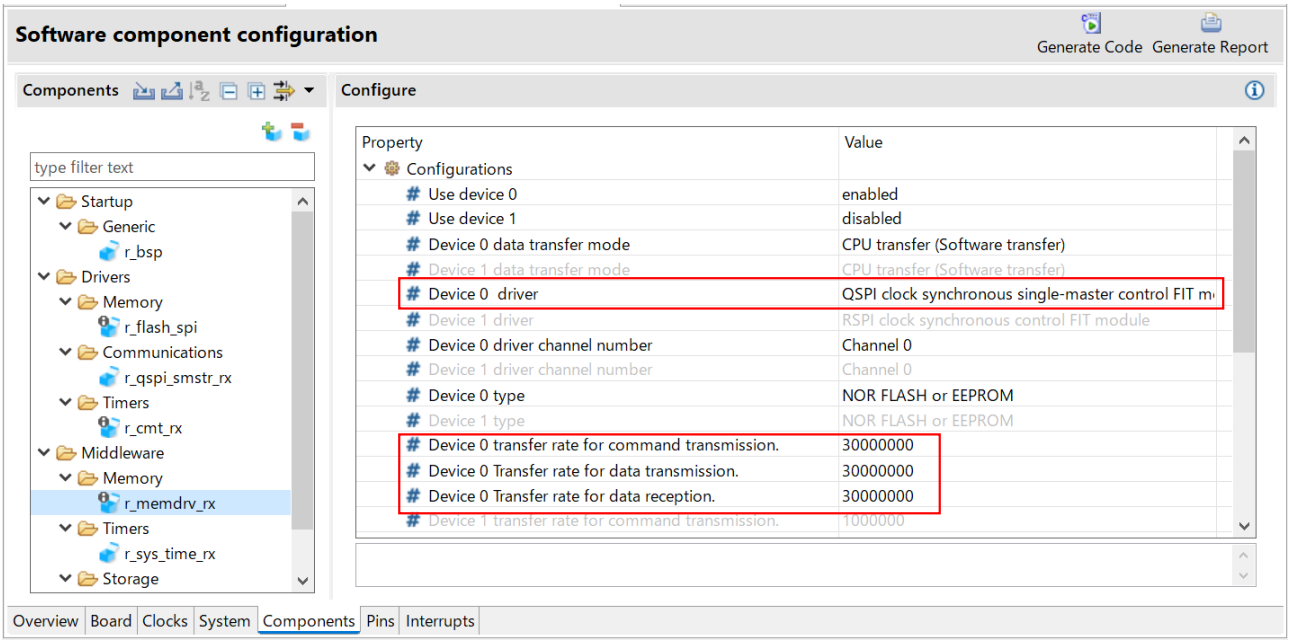


5.2.2 Memory Access Driver Interface Module

Table 5.2 shows the relevant settings of the memory access driver interface module and their currently set values. These settings pertain to the communication mode and speed.

Table 5.2 Changes to the Settings of the Memory Access Driver Interface Module

Item	Current Value
Device 0 driver	QSPI clock synchronous single master control FIT module
Device 0 transfer rate for command transmission.	30000000 (30 Mbps)
Device 0 Transfer rate for date transmission.	30000000 (30 Mbps)
Device 0 Transfer rate for data reception.	30000000 (30 Mbps)

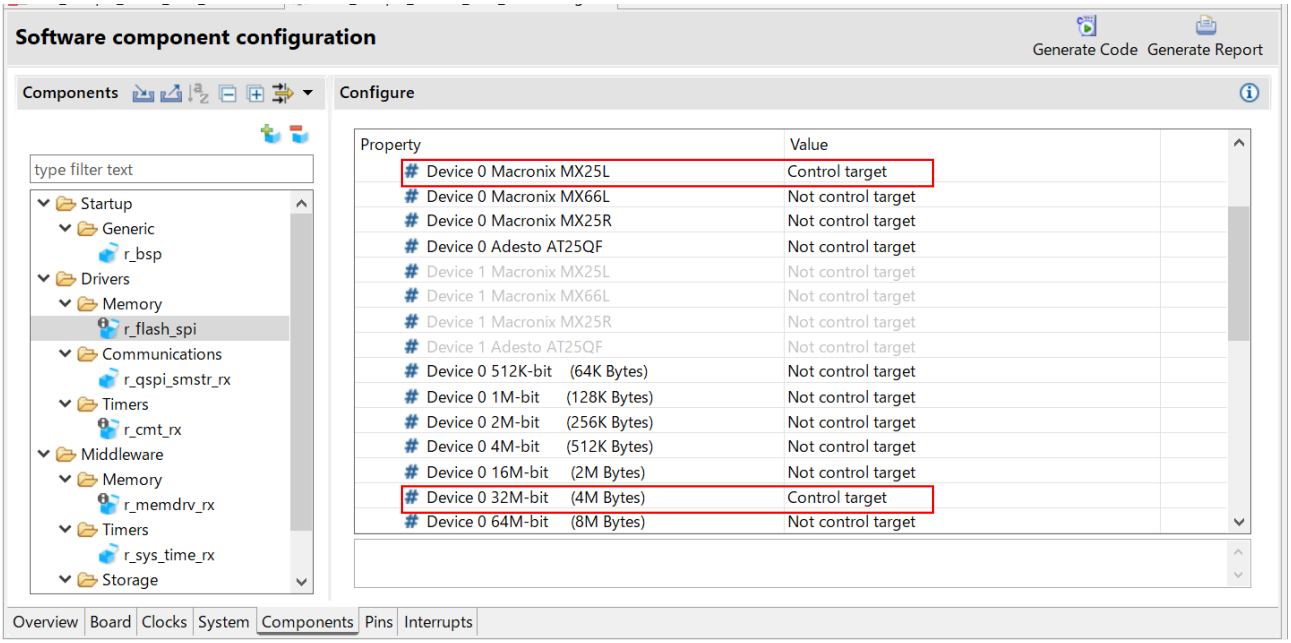


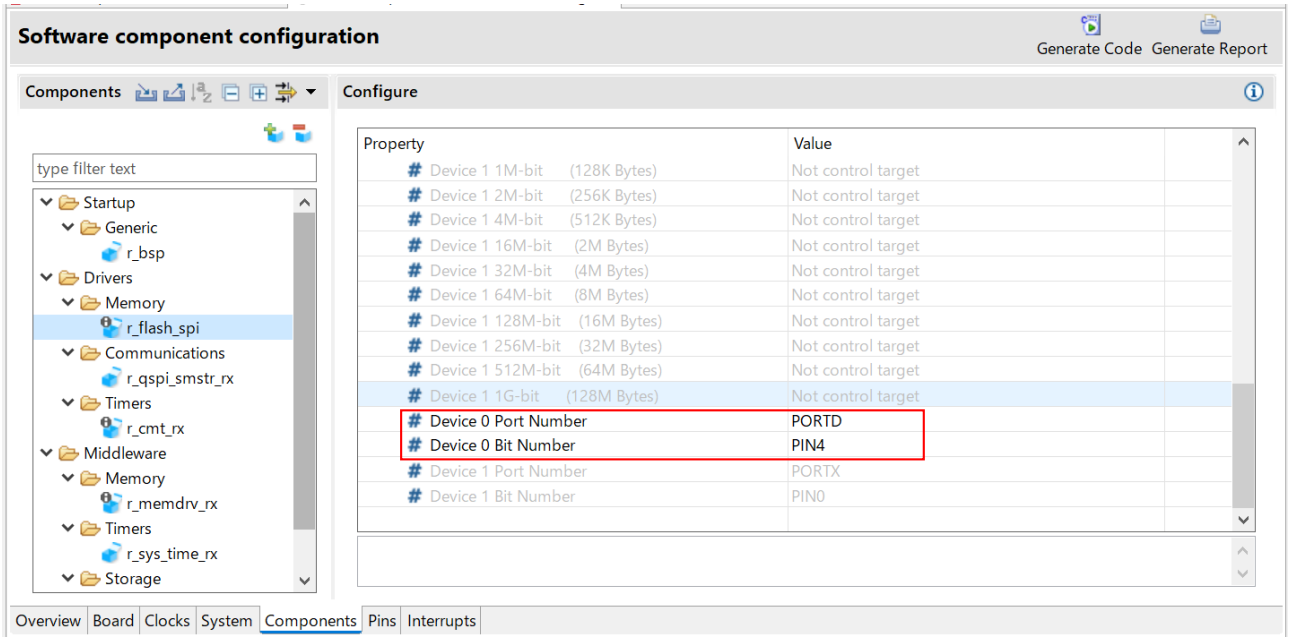
5.2.3 Clock Synchronous Control Module for Serial NOR Flash Memory Access

Table 5.3 shows the relevant settings of the clock synchronous control module for serial NOR flash memory access. For this module, the settings for selection of the control-target serial NOR flash memory device and the pin that is connected to the Chip Select pin of the target memory device have been changed.

Table 5.3 Changes to the Settings of the Clock Synchronous Control Module for Serial NOR Flash Memory Access

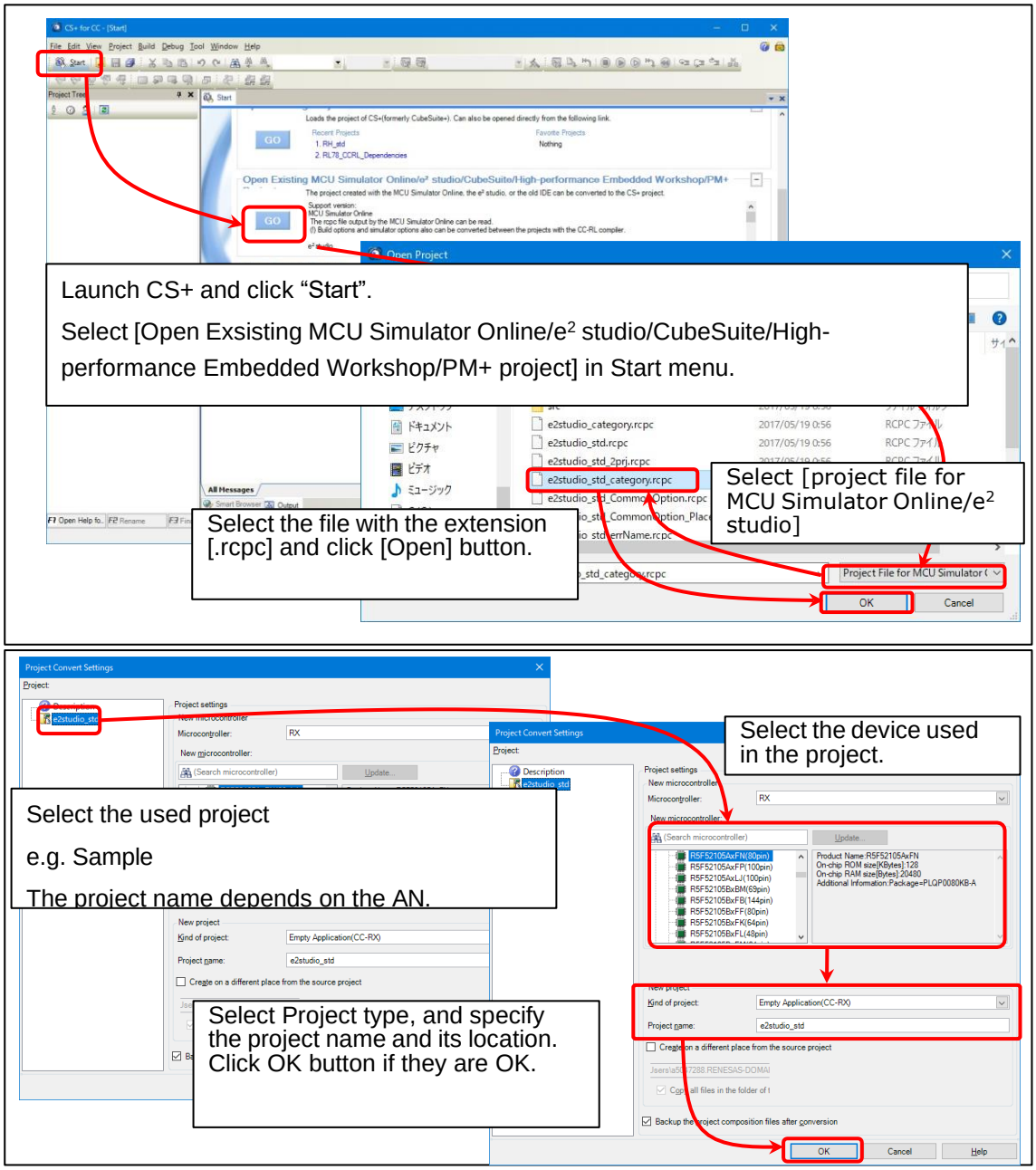
Item	Current Value
Device 0 Macronix MX25L	Control target
Device 0 32M-bit (4M Bytes)	Contral target
Device 0 Port Number	PORTD
Device 0 Bit Number	PIN4


Software component configuration
Generate Code
Generate Report
Components
Configure
type filter text
Startup
Generic
r_bsp
Drivers
Memory
r_flash_spi
Communications
r_qspi_smstr_rx
Timers
r_cmt_rx
Middleware
Memory
r_memdrv_rx
Timers
r_sys_time_rx
Storage
Property
Value
Device 0 Macronix MX25L
Control target
Device 0 Macronix MX66L
Not control target
Device 0 Macronix MX25R
Not control target
Device 0 Adesto AT25QF
Not control target
Device 1 Macronix MX25L
Not control target
Device 1 Macronix MX66L
Not control target
Device 1 Macronix MX25R
Not control target
Device 1 Adesto AT25QF
Not control target
Device 0 512K-bit (64K Bytes)
Not control target
Device 0 1M-bit (128K Bytes)
Not control target
Device 0 2M-bit (256K Bytes)
Not control target
Device 0 4M-bit (512K Bytes)
Not control target
Device 0 16M-bit (2M Bytes)
Not control target
Device 0 32M-bit (4M Bytes)
Control target
Device 0 64M-bit (8M Bytes)
Not control target
Overview
Board
Clocks
System
Components
Pins
Interrupts


Software component configuration
Generate Code
Generate Report
Components
Configure
type filter text
Startup
Generic
r_bsp
Drivers
Memory
r_flash_spi
Communications
r_qspi_smstr_rx
Timers
r_cmt_rx
Middleware
Memory
r_memdrv_rx
Timers
r_sys_time_rx
Storage
Property
Value
Device 1 1M-bit (128K Bytes)
Not control target
Device 1 2M-bit (256K Bytes)
Not control target
Device 1 4M-bit (512K Bytes)
Not control target
Device 1 16M-bit (2M Bytes)
Not control target
Device 1 32M-bit (4M Bytes)
Not control target
Device 1 64M-bit (8M Bytes)
Not control target
Device 1 128M-bit (16M Bytes)
Not control target
Device 1 256M-bit (32M Bytes)
Not control target
Device 1 512M-bit (64M Bytes)
Not control target
Device 1 1G-bit (128M Bytes)
Not control target
Device 0 Port Number
PORTD
Device 0 Bit Number
PIN4
Device 1 Port Number
PORTX
Device 1 Bit Number
PIN0
Overview
Board
Clocks
System
Components
Pins
Interrupts

6. Migration to CS+

Only an e² studio project sample program is bundled with this application note. If you use the sample program with CS+, load the e² studio project by using the following procedure.



7. Notes

7.1 About the Formatting Process

Make sure that the memory device (such as serial NOR flash memory) to be used with the TFAT library is formatted as a FAT file system (in FAT12, FAT16, or FAT32 format). If the serial NOR flash memory to be used for executing the sample program of this application note is not formatted (or formatted otherwise), appropriately format it by referring to "4.4.2 Formatting Process".

If you execute the sample program with a serial NOR flash memory device that is not formatted as a FAT file system (in FAT12, FAT16, or FAT32 format), the f_mount API function does not run correctly. As a result, the program fails to mount a file system and the following message appears in Renesas Debug Virtual Console:

```
Error (FR_NO_FILESYSTEM) Drive mount
```

7.2 Timer Function for Serial NOR Flash Memory

To use the TFAT library with serial NOR flash memory, the disk_1ms_interval callback function included in the TFAT memory driver FIT module must be registered as a periodic timer callback at an interval of 1 ms. In the sample program, to perform this processing, the R_CMT_CreatePeriodic API function of the CMT module is used as follows:

```
R_CMT_CreatePeriodic((uint32_t)(1000), disk_1ms_interval,
&gs_cmt_channel);
```

7.3 About the Stack and Heap Sizes

To use the TFAT library with serial NOR flash memory, the sector size must be changed to 4,096. Set "4096" for FF_MIX_SS and FF_MAX_SS in ffconf.h.

Note that this change increases the sizes of local and global variables used in the TFAT library and therefore may cause a stack overflow or heap overflow. Change the stack and heap sizes if necessary. For details about how to change the stack and heap sizes and the stack and heap sizes that are set in the sample program of this application note, refer to "5.2.1 BSP Module".

8. Reference Documents

User's Manual: Hardware
RX Family: RX72N Group User's Manual: Hardware (R01UH0824)

User's Manual: Evaluation Board
RX72N Envision Kit User's Manual (R20UT4788)
RX72N Envision Kit - Schematics (R20UT4789)

User's Guide: Development Environment
RX Smart Configurator User's Guide (R20AN0451)
RX Family: CC-RX Compiler User's Manual (R20UT3248)

The latest versions can be downloaded from the Renesas Electronics website. (www.renesas.com)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	May.01.24	—	Initial release

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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. 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.
6. 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. 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.5.0-1 October 2020)

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/.