RA0 Series

Migration Guide from Texas Instruments MSPM0 to RA0

Overview

This application note describes development and ecosystem, core architecture, peripheral considerations, and software development kit for when migrating from Texas Instruments MSPM0 Family to RA0 Series.

The intent of this document is to highlight the differences between the two products and to leverage existing knowledge of the MSPM0 Family development environment to quickly ramp up with the RA0 Series of MCUs.

Device to Be Migrated

Texas Instruments MSPM0 Family

Target Device

RA0 Series

Reference Documents

[1] RA0E1 Group User's Manual: Hardware (R01UH1040)
[2] RA0E2 Group User's Manual: Hardware (R01UH1090)
[3] RA0E3 Group User's Manual: Hardware (R01UH1165)
[4] RA0L1 Group User's Manual: Hardware (R01UH1143)
[5] Renesas e² studio 2023 -10 or Higher Quick Start Guide User's Manual RA Family Renesas MCU (R20UT4989)
[6] Renesas Flexible Software Package (FSP) v5.9.0 User's Manual(R11UM0155)

Contents

1. RA0 Portfolio Overview

1.1 Introduction

The RA0 Series (hereafter "RA0") is equipped with an Arm® Cortex®-M23 core and delivers industry-leading low power consumption in the 32-bit general-purpose microcontroller class. It features extremely low operating current and standby power consumption, making it ideal for battery-powered devices and applications where low power consumption is crucial.

The RA0 is ideal for applications in areas such as consumer electronics, small home appliances, industrial automation, and building automation, where low power consumption and cost-efficiency are critical. With the Renesas ecosystem, a wide variety of sample codes and application notes are available, enabling fast product development.

Renesas RA0 is the competitive alternative to Texas Instruments' MSPM0 family (hereafter referred to as "MSPM0"). This application note assists in migration from MSPM0 to RA0 by comparing device features and ecosystems.

1.2 Portfolio Comparison of MSPM0 and Renesas RA0

Table 1-1 Comparison of Renesas RA0 and TI MSPM0 Family (1/2)

Item		TI[3][4] MSPM0 Lx	TI[1][2] MSPM0 Cx	Renesas RA0
Core		Arm Cortex-M0+		Arm Cortex‑M23 core
Clock frequency		32MHz	Max 24MHz	32MHz
Power supply voltage		1.62 to 3.6 V	1.62 to 3.6 V	1.6 to 5.5 V
Operating temperature range		-40 to +125°C	-40 to +125°C	-40 to +125°C
Code flash		8KB to 256KB	8KB to 32KB	16KB to 128KB
Data Flash		N/A	N/A	1KB to 2KB
RAM		2KB to 32KB	1KB to 8KB	2KB to 16KB
I/O port		Max 73	Max 29	Max 60
Analog	ADC	Max 26ch (12-bit resolution)	Max 27ch (12-bit resolution)	Max 15ch (12-bit resolution)
	DAC	8-bit	8-bit	N/A
	COMP	1 × high-speed	1 × high-speed	N/A
Serial interface (Max)	UART	Max 5	Max 3	Max 5 (SAU 3ch, UARTA 2ch)
	I²C	Max 3 (Fast)	Max 2	Max 4
	SPI	Max 2	1	Max 6
	CAN	0	0	0
	LIN	2 (UART support)	1 (UART support)	1 (UART support)
Timer		TIMA0 TIMG0‑11 TIMG12	TIMA0 TIMG0‑11 TIMG12	16-bit × 8-ch 32-bit × 1-ch (16-bit × 2-ch or 8-bit × 4-ch)
Security		CRC	CRC	CRC, Unique ID, TRNG, Flash Read Protection

Table 1-2 Comparison of Renesas RA0 and TI MSPM0 Families (2/2)

Item		TI[3][4] MSPM0 Lx	TI[1][2] MSPM0 Cx	Renesas RA0
Other key peripherals / features		2 × Op-Amp, LCD(L2228)	Smallest QFN package(2x2), 0.5/0.65 mm Pitch packages, Pin-compatible with industry	Temperature Sensor, Event Link Controller Capacitive touch (RA0L1)
Number of pins		16-80 pins	8-48 pins	16-64 pins
Low power	Normal operation	71µA /MHz	100µA /MHz	84.3µA /MHz (RA0E1)
	Standby	Standby mode 1µA	Standby mode 5µA	Sleep mode 0.82mA (RA0E1)
				Software Standby mode 0.21µA (RA0E1)

Note The RA0 software standby mode corresponds to the MSPM0 Standby mode, and the RA0 sleep mode corresponds to the MSPM0 SLEEP mode.

2. Ecosystem and Migration

The RA0 ecosystem provides a wide range of support from hardware and software to debugging tools, libraries, and a support community to assist flexible and efficient development for a variety of applications. Developers can use this ecosystem to advance their product development quickly and effectively.

2.1 Comparison of Ecosystem

Table 2-1 Comparison of Ecosystem

Item	MSPM0	RA0
Code source	MSPM0-SDK (DriverLib, Middleware, RTOS, Code example)	Flexible Software Package – FSP (Middleware, Driver, RTOS) Smart Browser (sample code, document references)
IDE	CCS IAR Embedded Workbench Arm Keil MDK IDE	e² studio IAR Embedded Workbench Arm Keil MDK IDE
Software configuration	SysConfig	FSP Configuration RA Smart Configurator (RASC)
Software programming tool	UniFlash	Renesas Flash Programmer
Hardware programming tool	MSP-GANG	PG-FP6
Debugger	XDS110 J-Link	E2 Emulator E2 Emulator Lite J-Link
Evaluation board	LP-MSPM0L1306 launchpad LP-MSPM0C1104 launchpad	FPB-RA0xx Fast Prototyping Board RSSK-RA0L1

Figure 2-1 shows the overview of the RA0 Ecosystem.

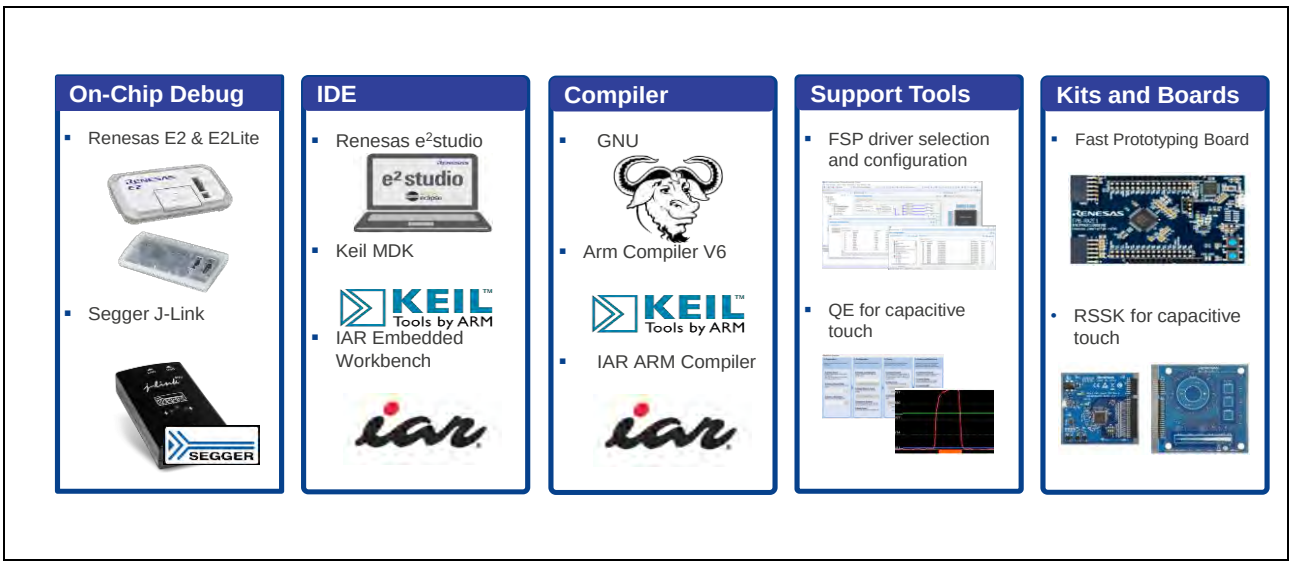


Figure 2-1 Overview of RA0 Ecosystem

2.1.1 Flexible Software Package - FSP

The Flexible Software Package includes drivers and stacks for general functions such as communications and security. The drivers and stacks are provided in middleware stacks and RTOS-independent Hardware Abstraction Layer (HAL) drivers for individual user applications.

Figure 2-2 shows the Flexible Software Package configuration.

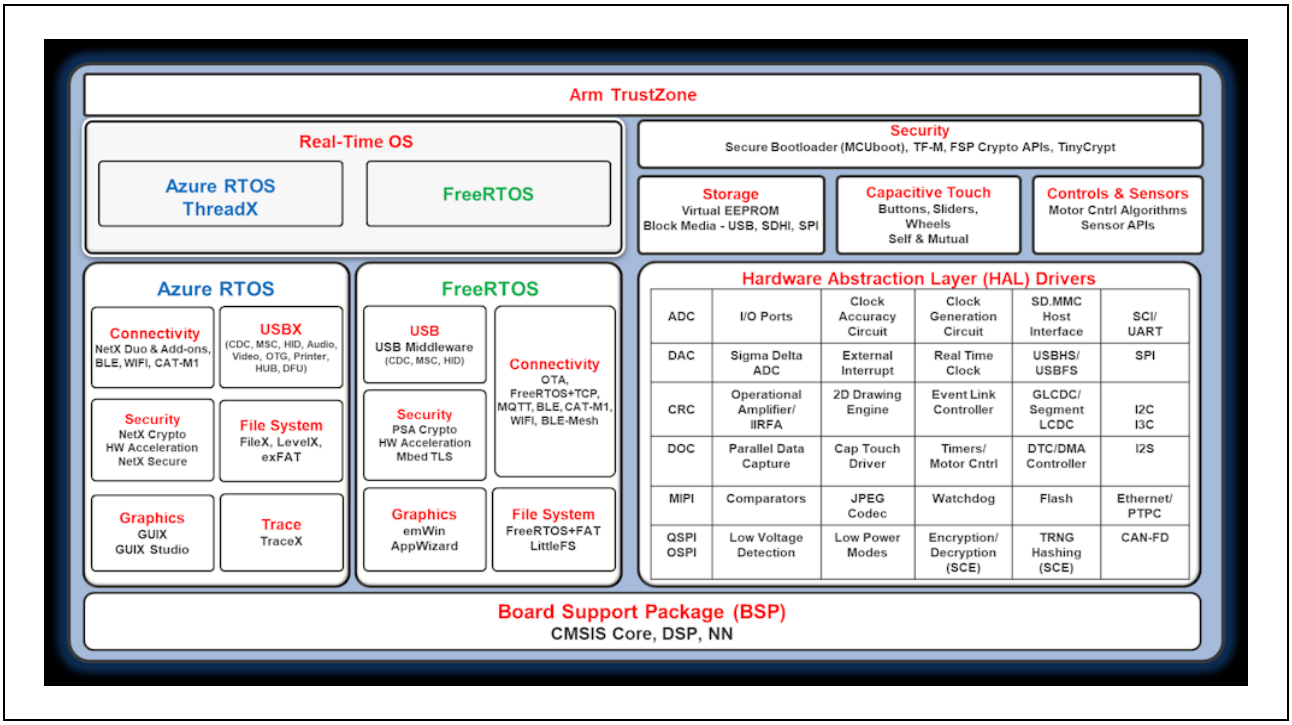


Figure 2-2 Flexible Software Package Configuration

The Renesas Board Support Package (BSP) serves as the lowest layer of the FSP. It links evaluation board-compatible programs with other FSP modules based on hardware information based on other sources provided by Renesas, such as evaluation boards.

2.1.2 IDE Supported by RA0

An Integrated Development Environment (IDE) is a software application that normally includes an editor, compiler, and debugger to help users develop software code efficiently.

The typical IDE supported by MSPM0 is CCS which consists of a set of tools used to develop and debug embedded applications, including optimizing C/C++ compiler, source code editor, project build environment, debugger, profiler, and many other functions.

The RA0 IDE is e² studio, an Eclipse-based IDE like CCS that allows users to get started easily.
e² studio also consists of a completed set of tools with many features for developing and debugging embedded applications. In addition, the e² studio is free to use and integrates automatic code generation from the FSP Configurator, integration and management of software modules with the FSP, and provision of RA0 sample code and technical documentation through the Smart Browser.

Table 2-2 Comparison of CCS and e² studio

IDE	CCS	e² studio
License	Free	Free
Compiler	TI Arm Clang / GCC	GCC / LLVM / Arm Compiler / IAR Arm Compiler
Peripheral API function support	Not supported	Supported
Display languages	English	English Japanese Chinese
Executable file output format	Hex file Binary file Motorola S-record file Ti_txt file	Hex file Binary file Motorola S-record file
Code generation GUI	SysConfig	FSP Configurator RA Smart Configurator (RASC)

In addition to e² studio, Arm Keil MDK and IAR Embedded Workbench also support the RA0.

For e² studio usage and function descriptions, refer to 2.2.2.2 How to Use e² studio.

2.1.3 Smart Browser

The Smart Browser enables the user to easily search and view the latest hardware manual, technical updates, and application notes for Renesas devices. It also allows the user to download sample codes and import sample codes that include the project.

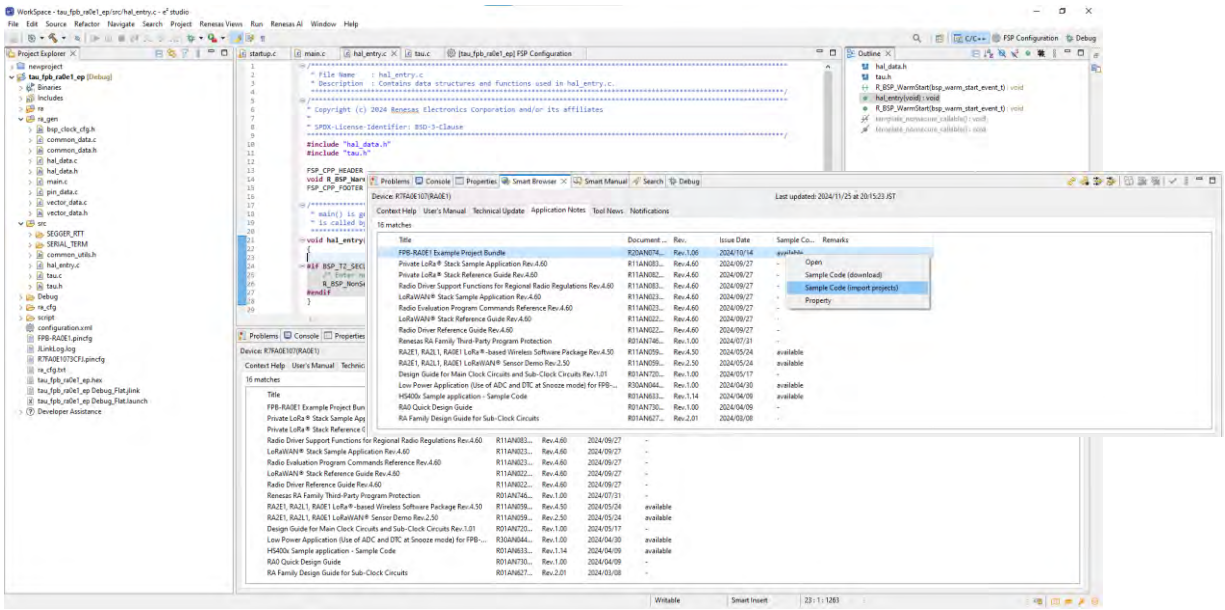


Figure 2-3 Smart Browser

The Smart Browser is included in the e² studio and can be used by installing the e² studio. The Smart Browser has multiple tabs so that developers can quickly access the information necessary for their project.

The "User's Manual" tab provides access to various development tools and official documents regarding devices.

The "Technical Update" tab provides information related to technical updates, new functions, and device updates, allowing developers to check up on the latest technical updates regarding the device and tools used for the project.

The "Application Notes" tab provides technical information useful to application development, including documents that describe how to use specific peripheral functions and software designs.

Table 2-3 Comparison of Software Ecosystem

Item	MSPM0	RA0
Driver library	Available	Available
Middleware	Available	Available
Self-programming	Available	Available
Sample program	Available	Available
Free RTOS	Available	Available

2.1.4 FSP Configurator

The FSP Configurator is integrated into e² studio. Like SysConfig, the FSP Configurator can quickly and smartly achieve suitable software combination and configurations according to the user's development needs, such as pin settings for MCU peripherals, and the embedding of middleware and drivers.
With the FSP Configurator, the user can visually resolve conflicts between pin settings or peripheral functions, while customizing settings suitable for the project itself.

The FSP Configurator is also available as a standalone version that can be run without an IDE. The standalone version can be used in combination with the IAR Embedded Workbench and Arm Keil MDK IDE.

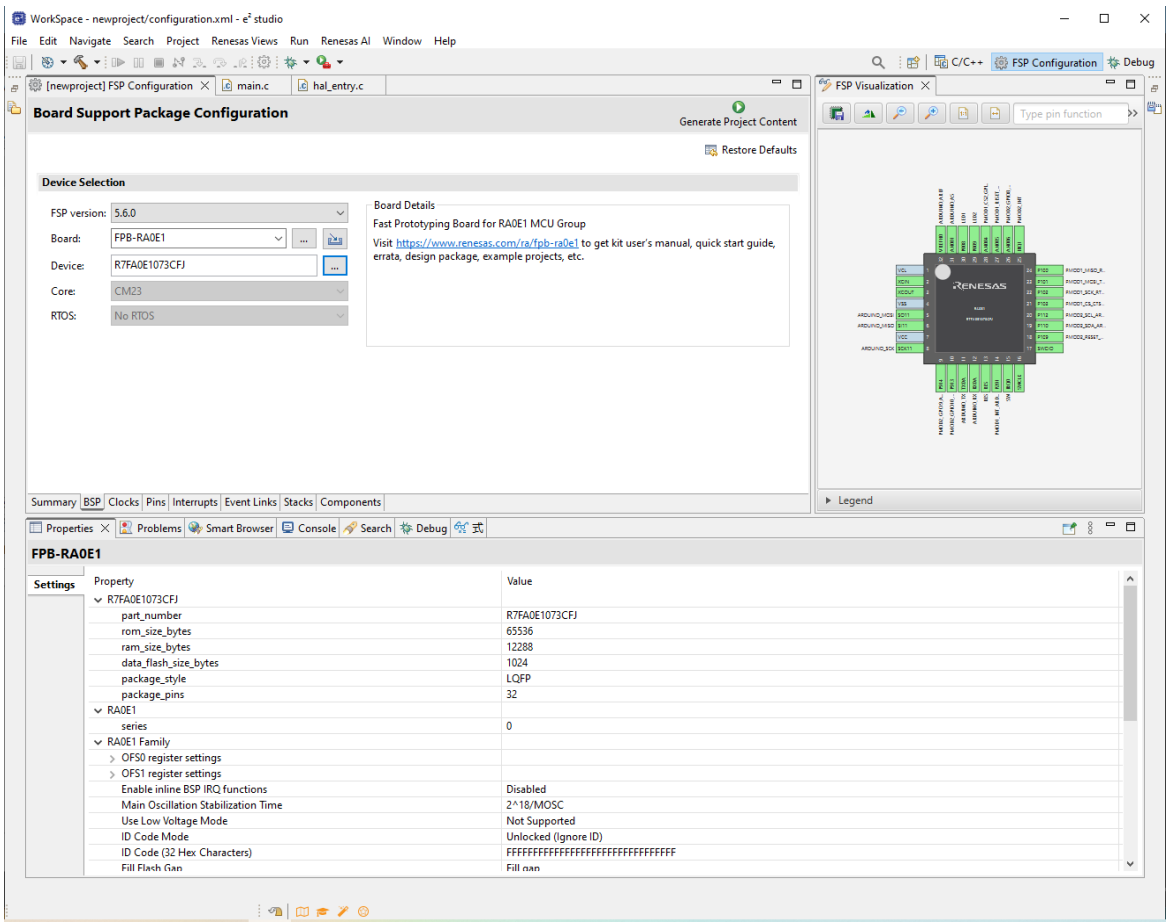


Figure 2-4 FSP Configurator

The Flexible Software Package (FSP) includes drivers and stacks for common functions such as communications and security, provided by middleware stacks and RTOS-independent Hardware Abstraction Layer (HAL) drivers for individual user applications.

Figure 2-5 compares categorized lists of peripheral functions and Figure 2-6 compares interrupt settings.

TI's SysConfig interrupt and pin settings are configured directly with specific peripheral functions, while the FSP Configurator adds HAL driver modules (called stacks in FSP) as needed and configures them individually on a per-module basis.

The FSP Configurator allows the user to focus on setting only the components implemented in the application.

After all settings are complete, the FSP Configurator will update the generated codes all at once when the user presses the "Generate Code" button.

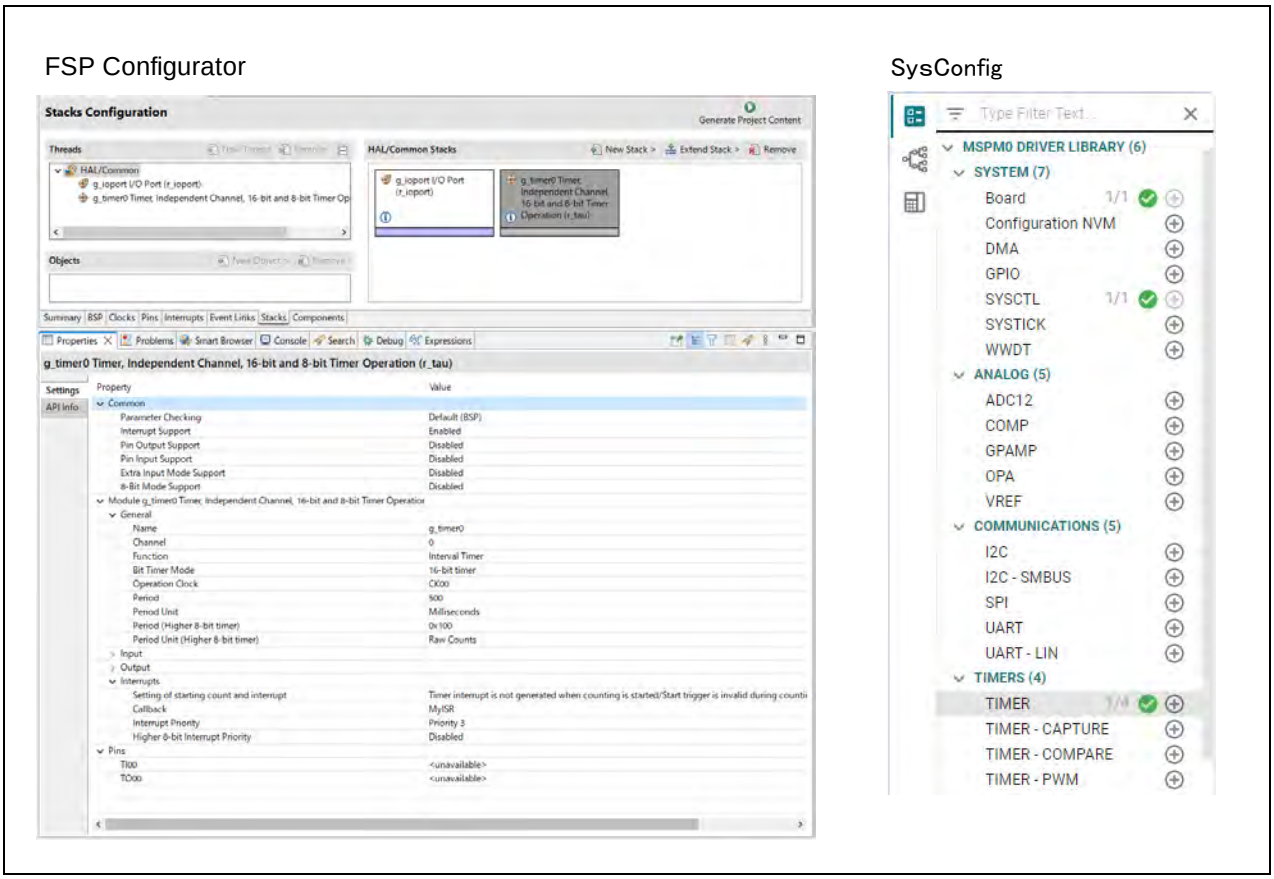
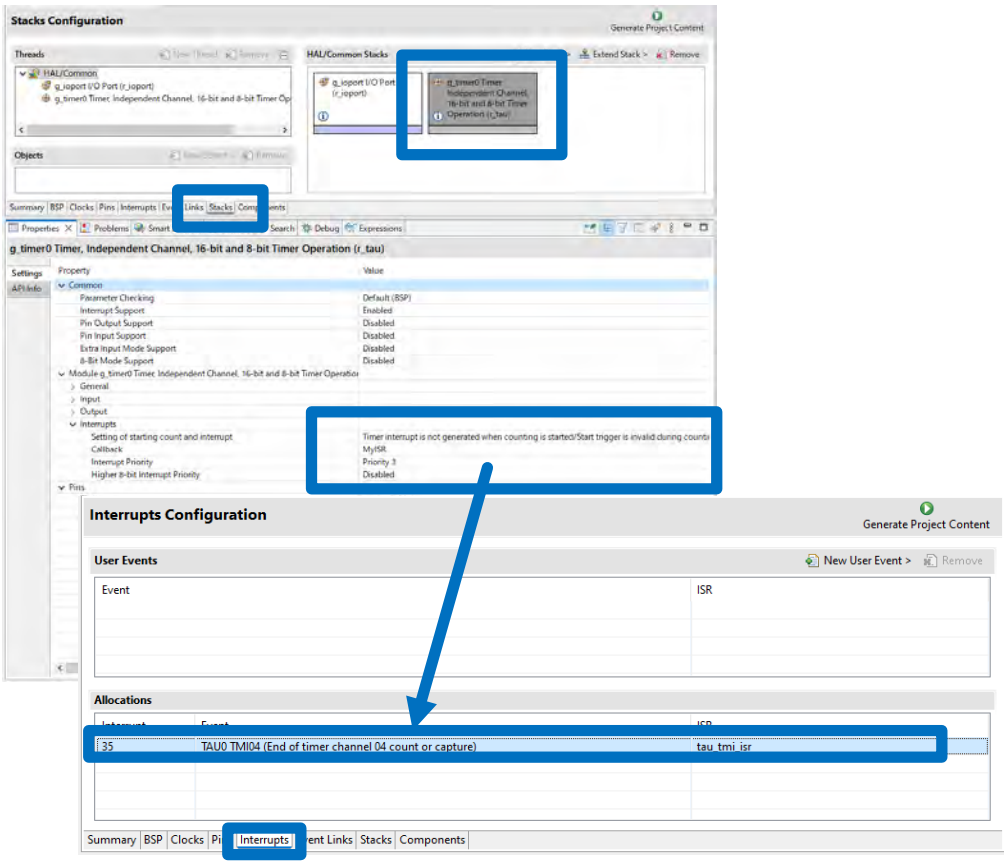


Figure 2-5 Comparison of Peripheral Function Lists

FSP Configurator



SysConfig

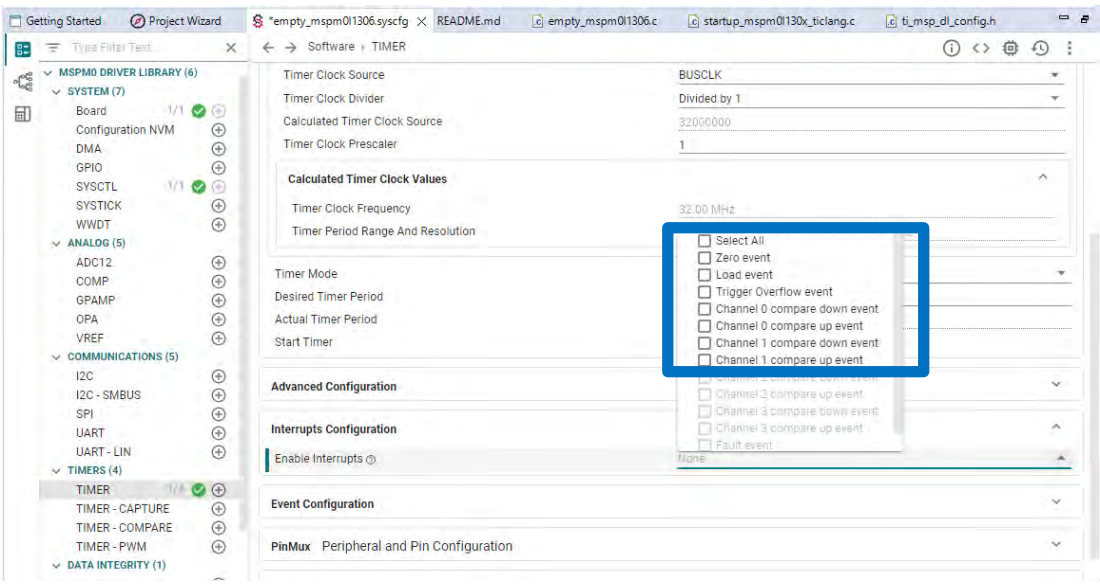


Figure 2-6 Interrupt Setting Comparison

2.1.5 Debug Tools

For MSPM0, the Debug Subsystem (DEBUGSS) connects the Serial Wire Debug (SWD) two-wire physical interface to multiple debug functions within the device. MSPM0 supports debugging of processor execution, device state, and power state (via EnergyTrace technology).

Like the MSPM0, the RA0 uses the Serial Wire Debug (SWD) interface to connect to the debugger (such as the E2 Emulator, E2 Emulator Lite, J-link, etc.). For details on the debugger connection, refer to Figure 2-7.

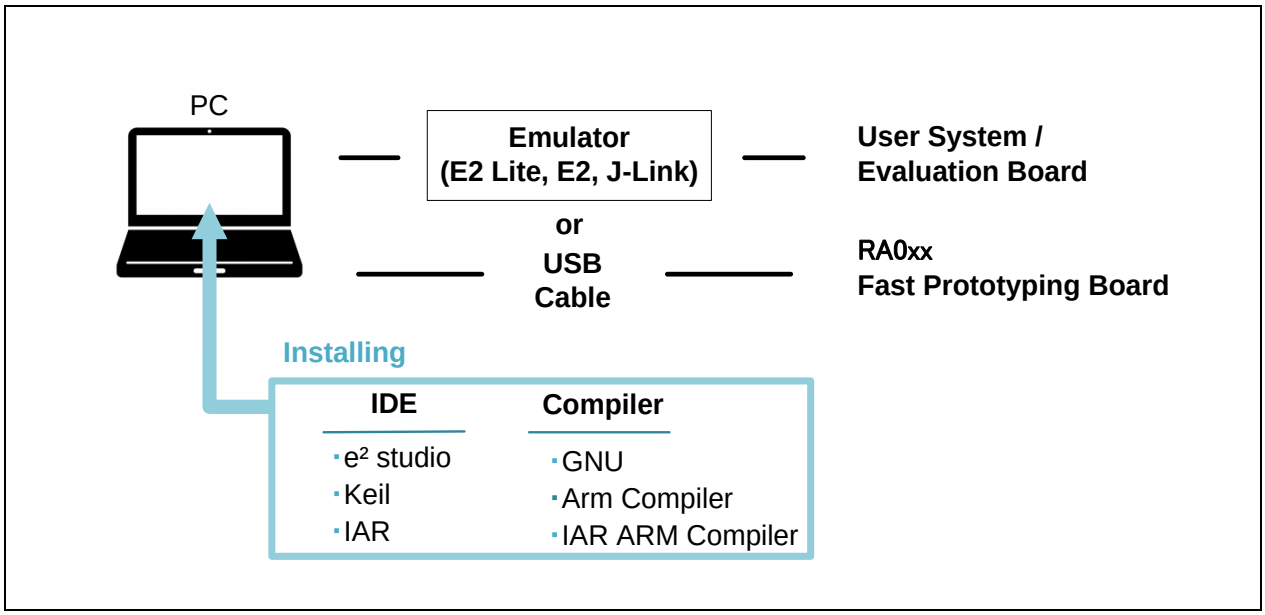


Figure 2-7 RA0 Debug

The RA0 supports on-chip debugging through the E2 emulator, the E2 emulator Lite, and J-Link via SWD connection. All these debug functions include a flash memory programming function. The RA0xx Fast Prototyping Board is equipped with an on-chip debugging function via a USB connection which enables debugging without an emulator.

The E2 Emulator and E2 Emulator Lite are versatile Renesas emulators that support a wide range of microcontrollers including the RA0. The E2 Emulator can be used for real-time debugging and breakpoint setting to analyze the operation of the program. The advanced trace function allows you to trace the executed code for precise analyzation. Note that the E2 Emulator Lite does not include the advanced trace function. However, it provides a sufficient debug function and enables development at low cost.

2.1.6 Evaluation Board

The Fast Prototyping Board allows you to easily test the operation of the microcontroller, providing strong support for evaluation and development using the RA0 MCU.

2.1.6.1 Fast Prototyping Board

The Fast Prototyping Board is an evaluation board equipped with the RA0, made specifically for prototyping and developing various applications.

The RA0xx Fast Prototyping Board embeds circuits such as user switches and LEDs. The Arduino header supports various Arduino shields and enables rapid prototyping by creating sketches with the Arduino IDE.

Figure 2-8 shows an overview of the RA0E1 Fast Prototyping Board as example.

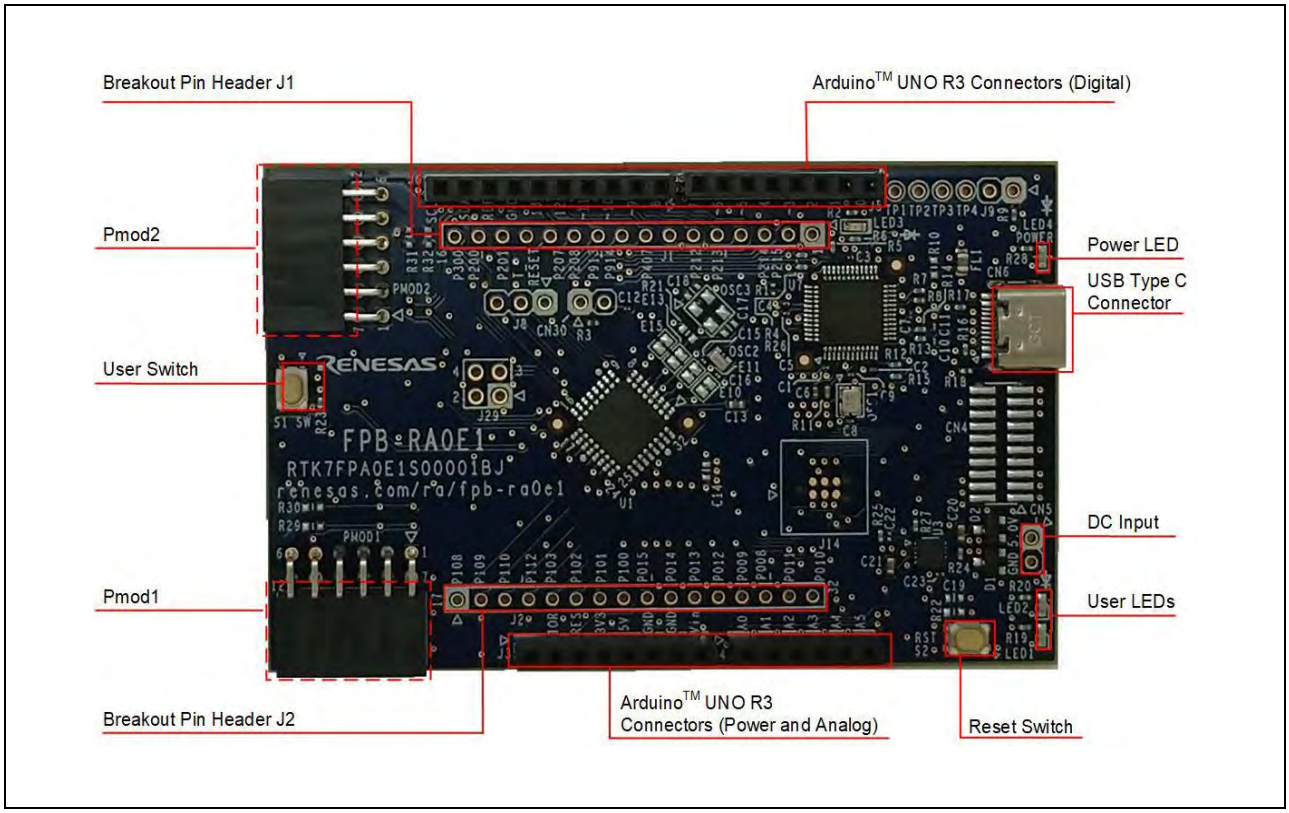


Figure 2-8 Overview of RA0E1 Fast Prototyping Board

2.2 Migration Process

The process flow for smooth migration to RA0 is shown in Figure 2-9. Each step is described in detail and examples are given in the following sections.

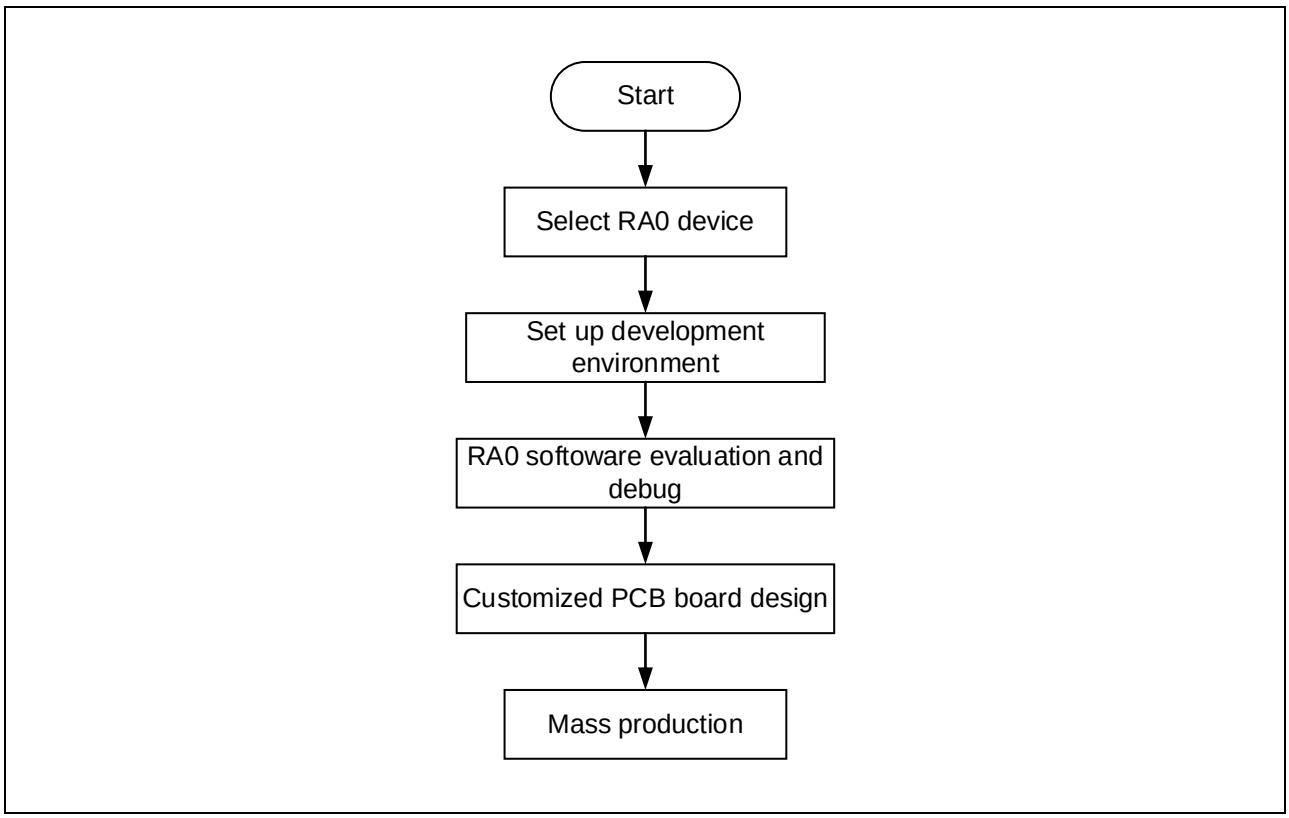


Figure 2-9 Flowchart of Migration to RA0

2.2.1 Step1: Selecting the Correct RA0

The first step of migration is to select the correct RA0 for the application. The devices are shown according to memory and package type for easy selection. The product selection flow is explained using RA0E1 as an example.
To narrow the choices down to a specific device, first access the RA0E1 Product page shown in Figure 2-10. Then select Product Selection and use the filters on the top shown in Figure 2-11 to narrow down the peripheral functions.

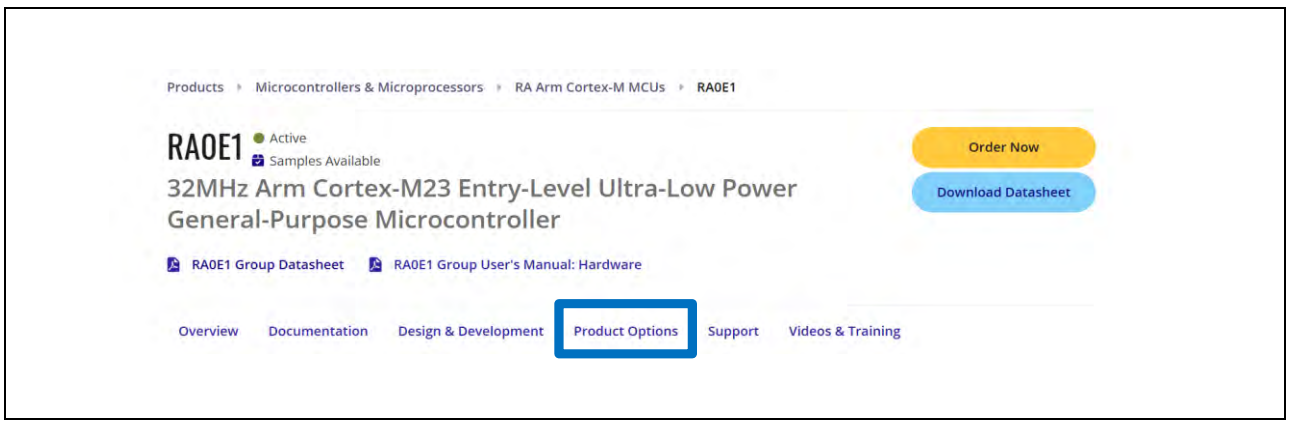


Figure 2-10 Product Page

For example, to filter out the MCUs that do not meet the number of Program memory and I/O ports required, select the settings as shown in the blue box in Figure 2-11 using the filter tool. Doing so will display the MCUs that meet the specified conditions.

You can set other filters as well to find products that meet your requirements.

When selecting from the list, the popup display shows the stock availability, price, and other information as shown in Figure 2-12

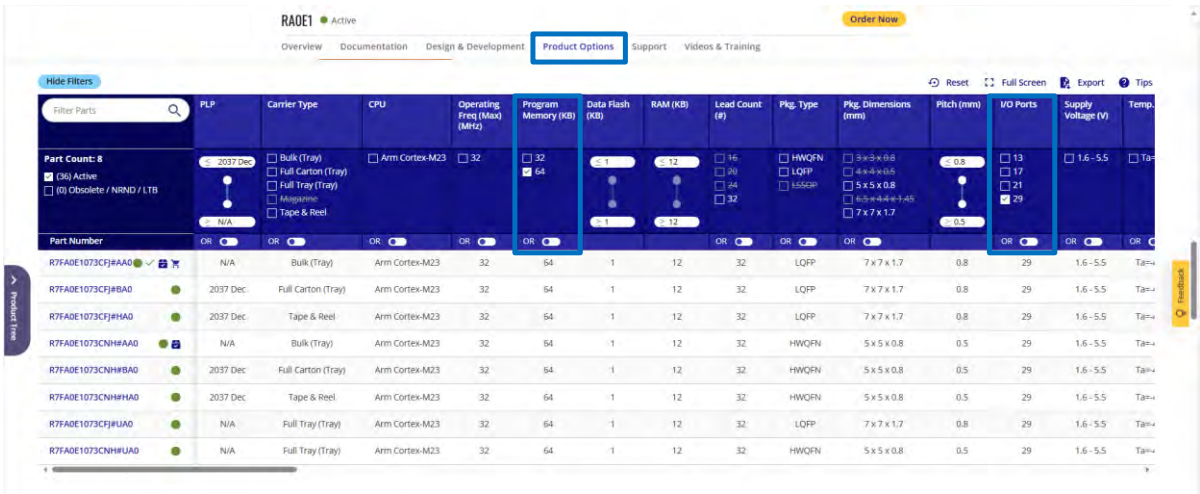


Figure 2-11 Product Selection

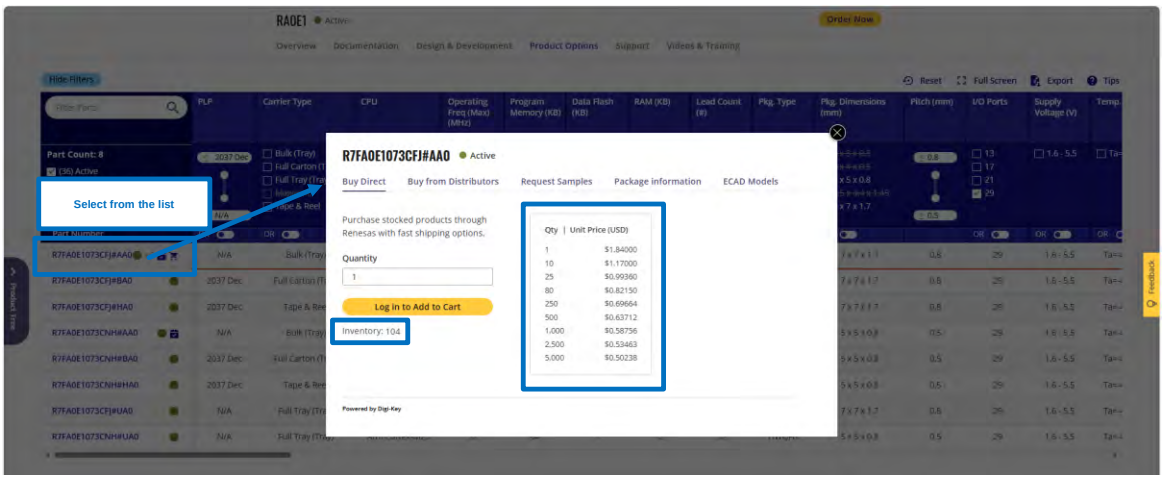


Figure 2-12 Popup Display

The Documents section of the Products page lists documents such as the datasheets, manuals, and application notes, and allows you to easily select and download necessary documents (see Figure 2-13).

The Datasheet provides a simple description of the RA0 specifications.
The Manual Hardware Version describes detailed specifications of the RA0.
The website also lists technical documents related to RA0, allowing you to find the documents necessary for development.

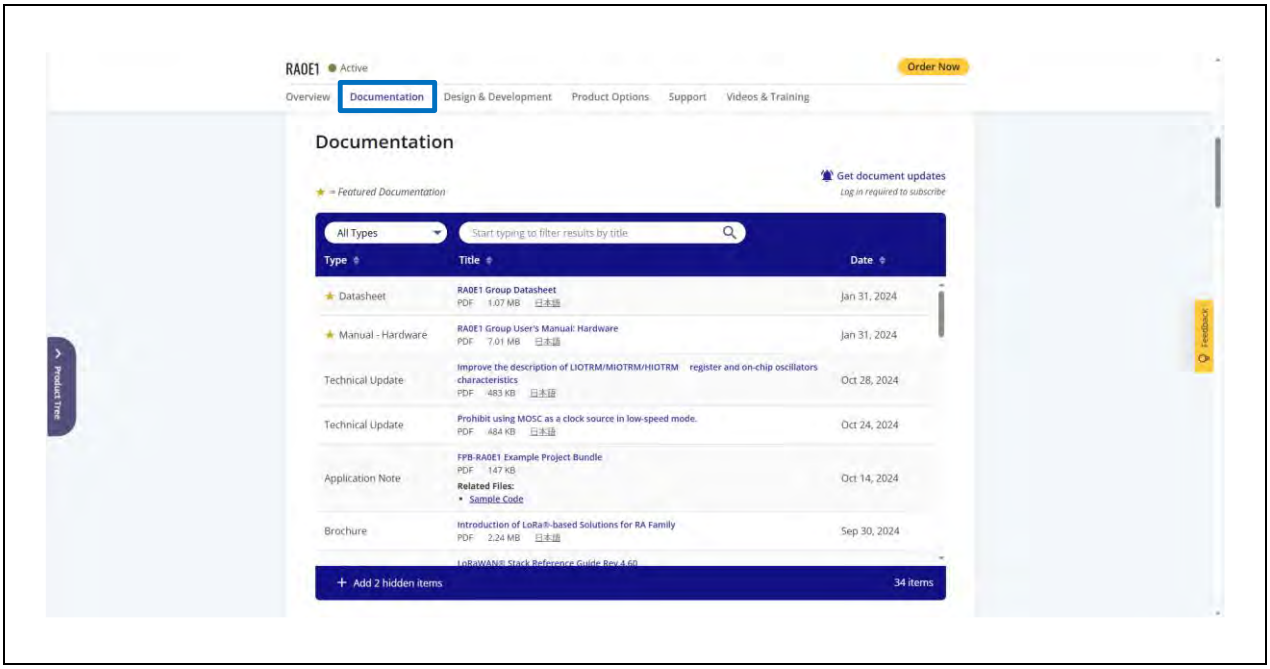


Figure 2-13 List of RA0 related Technical Documents

2.2.2 Step 2: IDE Setup and Quick Introduction of e² studio

2.2.2.1 IDE Setup

(1) Download the installer from the e² studio download link and start installation.
Select **Quick Install** for the installation type and click **[Next]**.

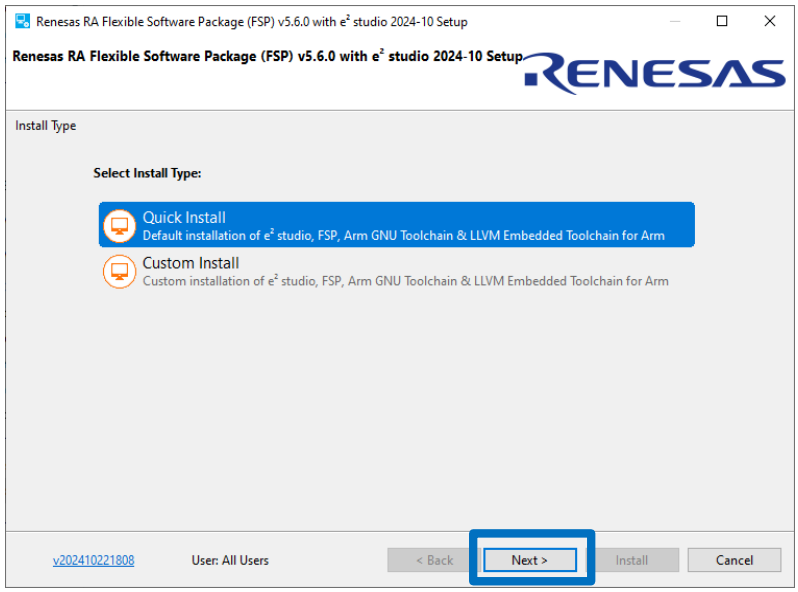


Figure 2-14 Select e² studio Installation Type

(2) If there are no issues with the installation contents, press the install button to complete the installation of e² studio.

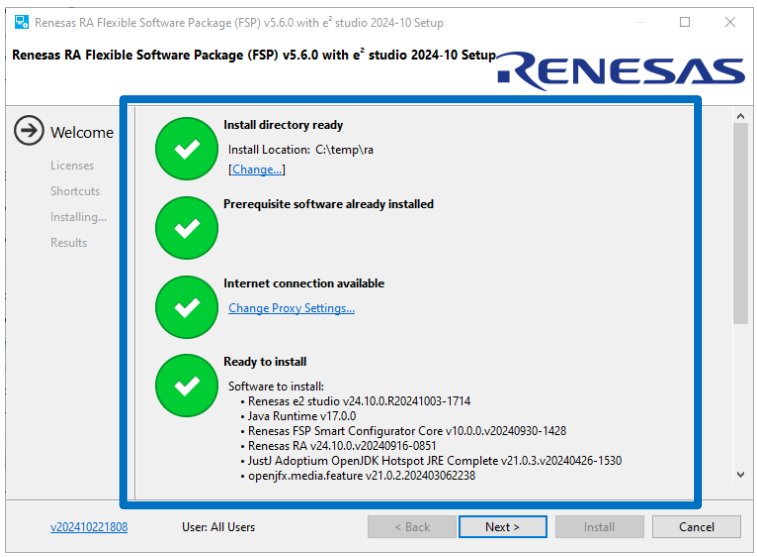


Figure 2-15 Overview of e² studio installation

2.2.2.2 How to Use e² studio

(1) Start a new workspace. "Workspace" is the path where project files and development environment settings are saved. The functions of e²studio are the same as those of CCS.

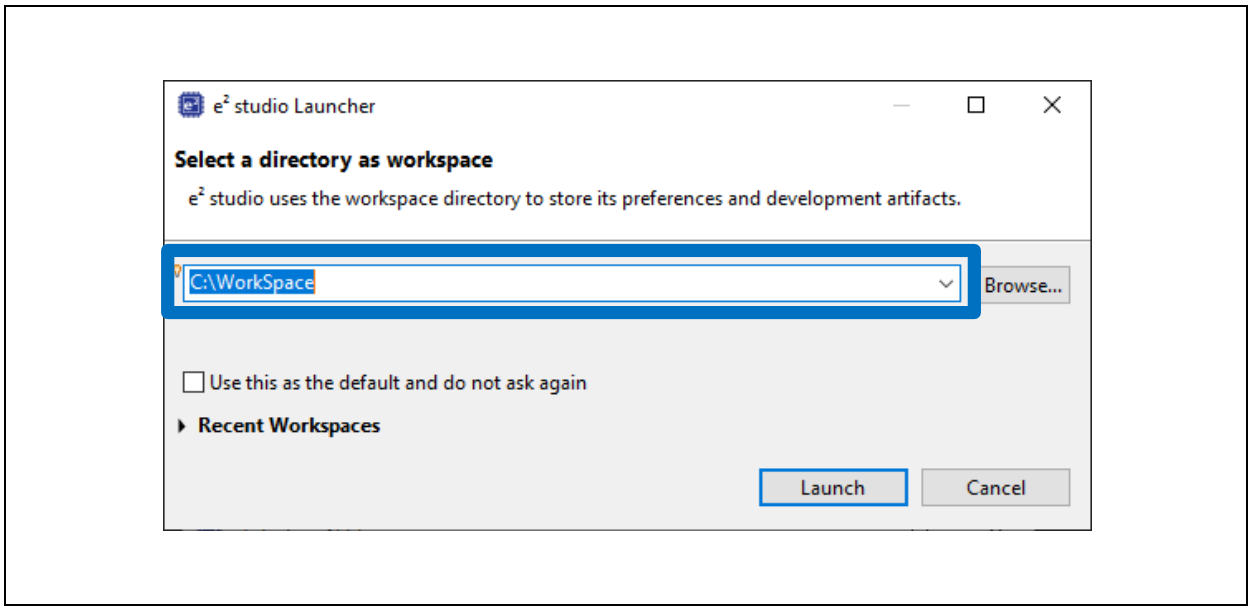


Figure 2-16 Select e² studio Workspace

(2) To create a new project, select **[File]** – **[New]** – **[Renesas C/C++ Project]** – **[Renesas RA]**. As with creating a new project in CCS, configure the device, toolchain, and project name.

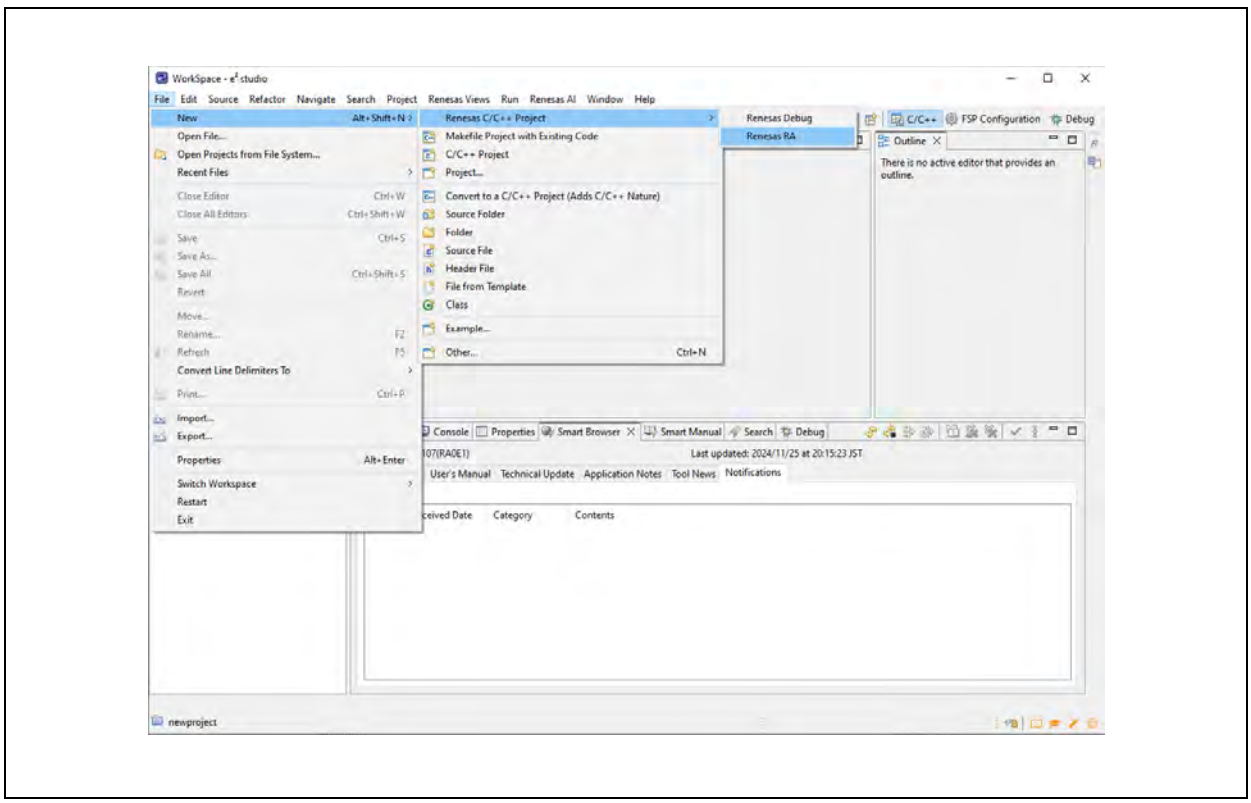


Figure 2-17 Create e² studio New Project and Select Device

(3) This is the setup screen for creating a new project. Click the **Finish** button to generate the project.

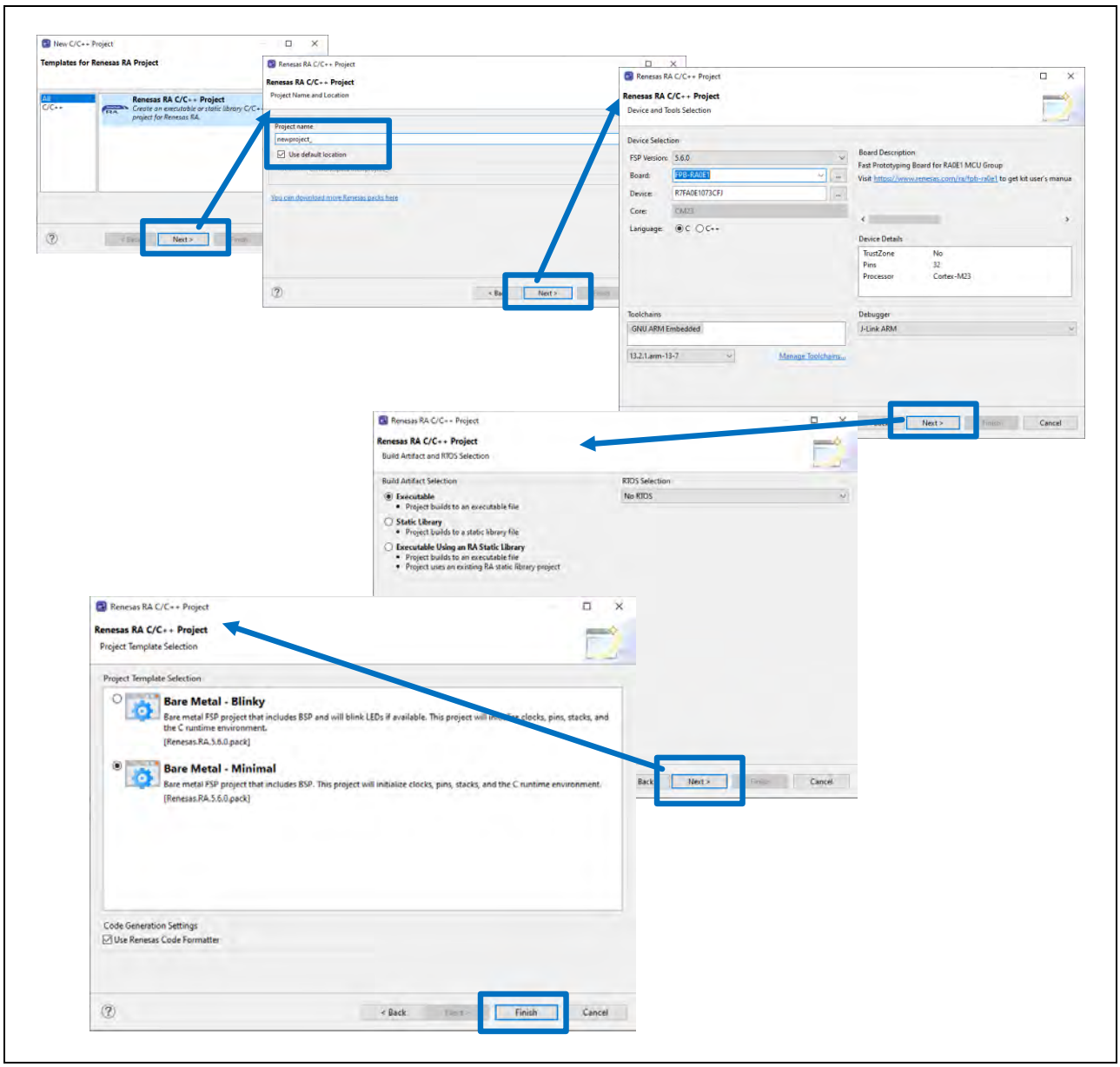


Figure 2-18 Create a New Project Overview

(4) The project is generated and opened in the workspace.

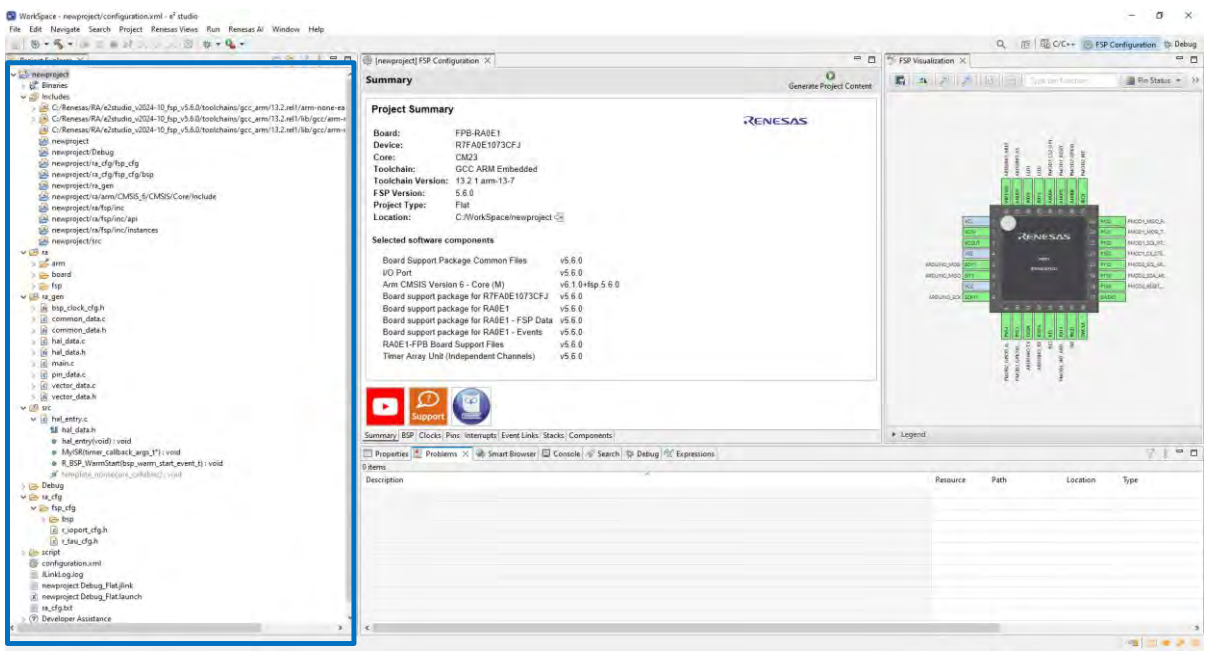


Figure 2-19 New Project Generation Completed

(5) Figure 2-20, Figure 2-21, Figure 2-22, Figure 2-23 are quick introduction of the e² studio functions.

Shortcut Key Functions:

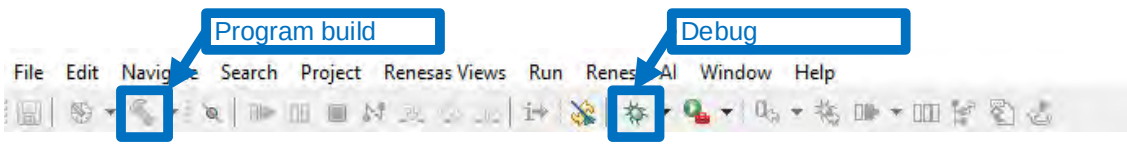


Figure 2-20 Commonly Used Functions

Debug Functions:

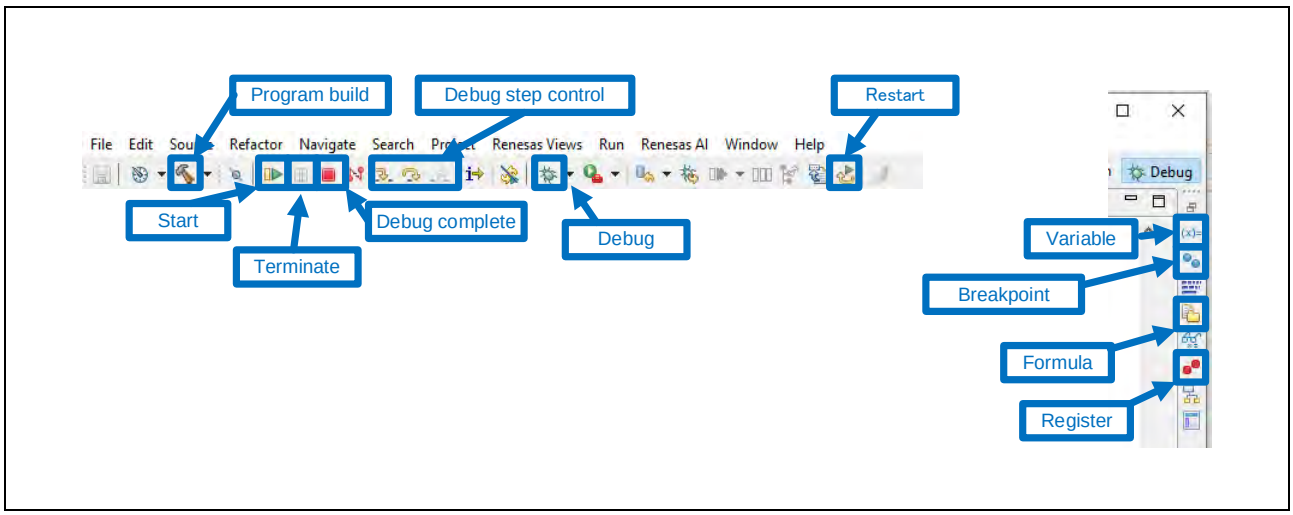


Figure 2-21 Commonly Used Debug Functions

Commonly Used Settings in the Project Properties:

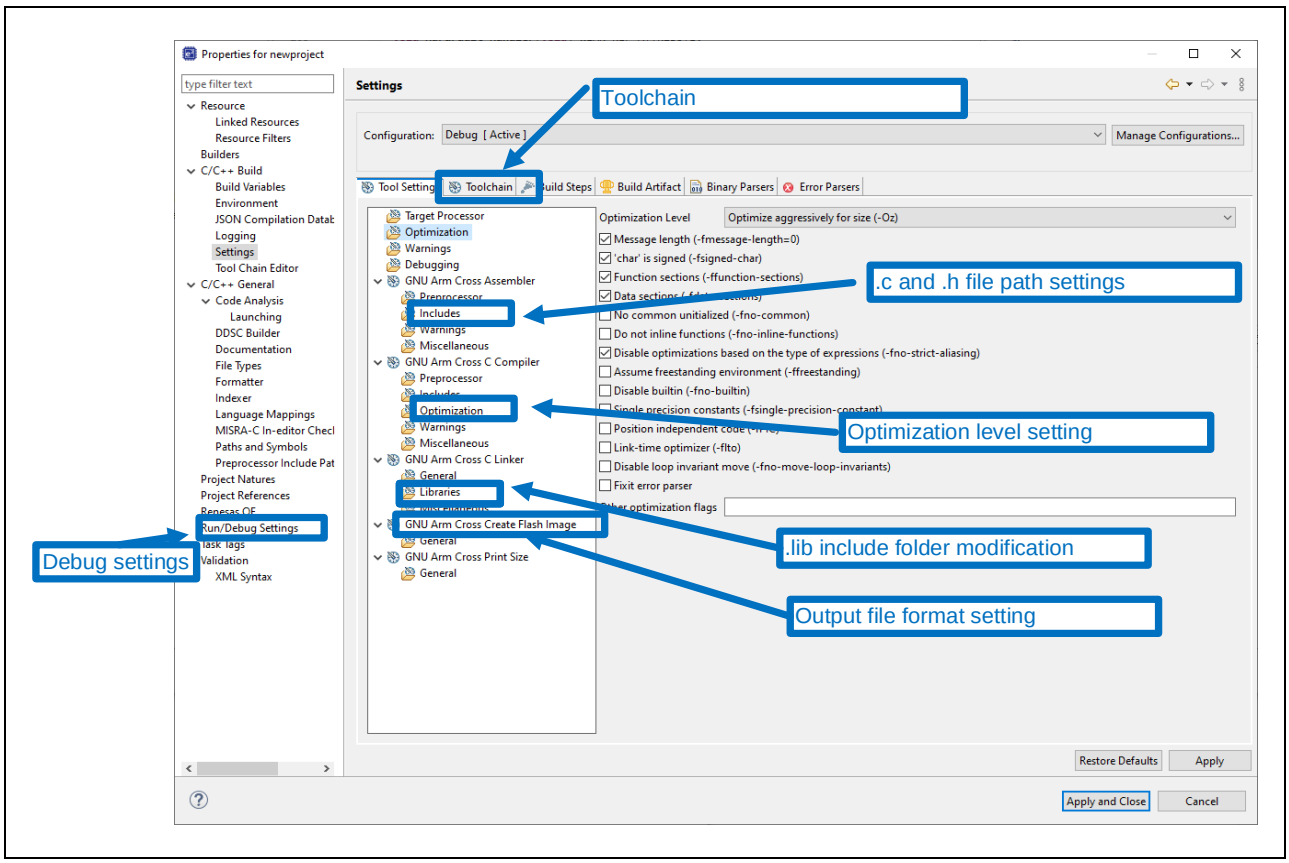


Figure 2-22 Commonly Used Project Settings (1/2)

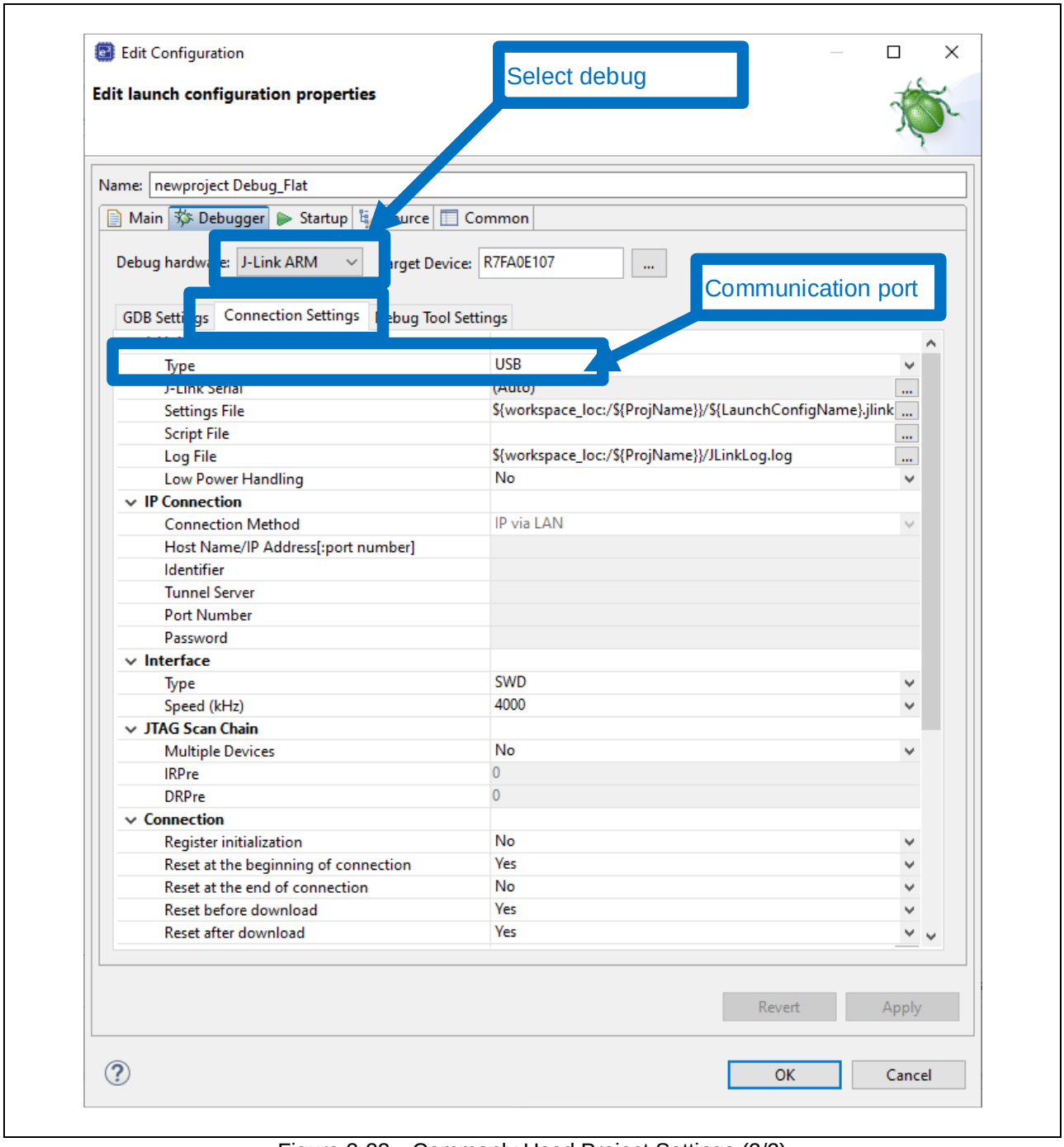


Figure 2-23 Commonly Used Project Settings (2/2)

2.2.3 Step 3: Importing the Software

This section provides is a quick guide on how to import your sample to the e² studio, using FPB-RA0E1 as an example.

(1) In this example, we will import the sample code from the Renesas website.

Select **[File]** - **[Import…]** from the menu to open the **Import** dialog box.

Since we are importing from the Renesas website to the workspace, select **[General]** - **[Sample Projects on Renesas website]** and click **[Next]** to open the **Import sample project on Renesas website** dialog box.

After selecting the **Device** (in this case, select "R7FA0E107"), a list of sample projects matching the device will be displayed.

From the list, select the sample project "FPB-RA0E1 Example Project Bundle" and press the **[Finish]** button. This "FPB-RA0E1 Example Project Bundle" consists of multiple project files, so a list of the projects contained within it will be displayed.

From this list, select the "tau_fpb_ra0e1_ep" sample project, which demonstrates the timer function, and press the **[Finish]** button to add the project to the workspace.

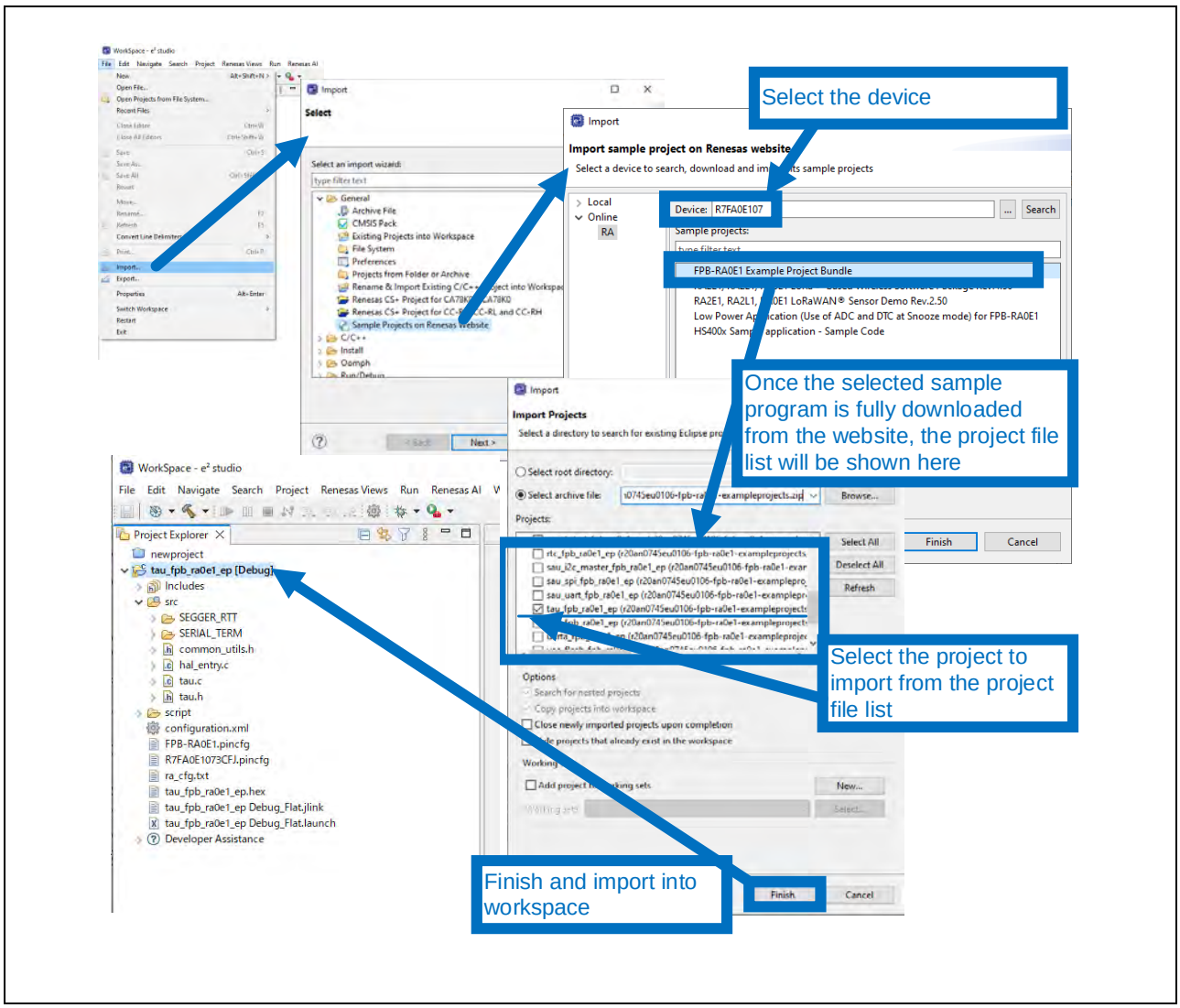


Figure 2-24 Import the Sample Program

(2) Figure 2-25 shows the most important files in the project.

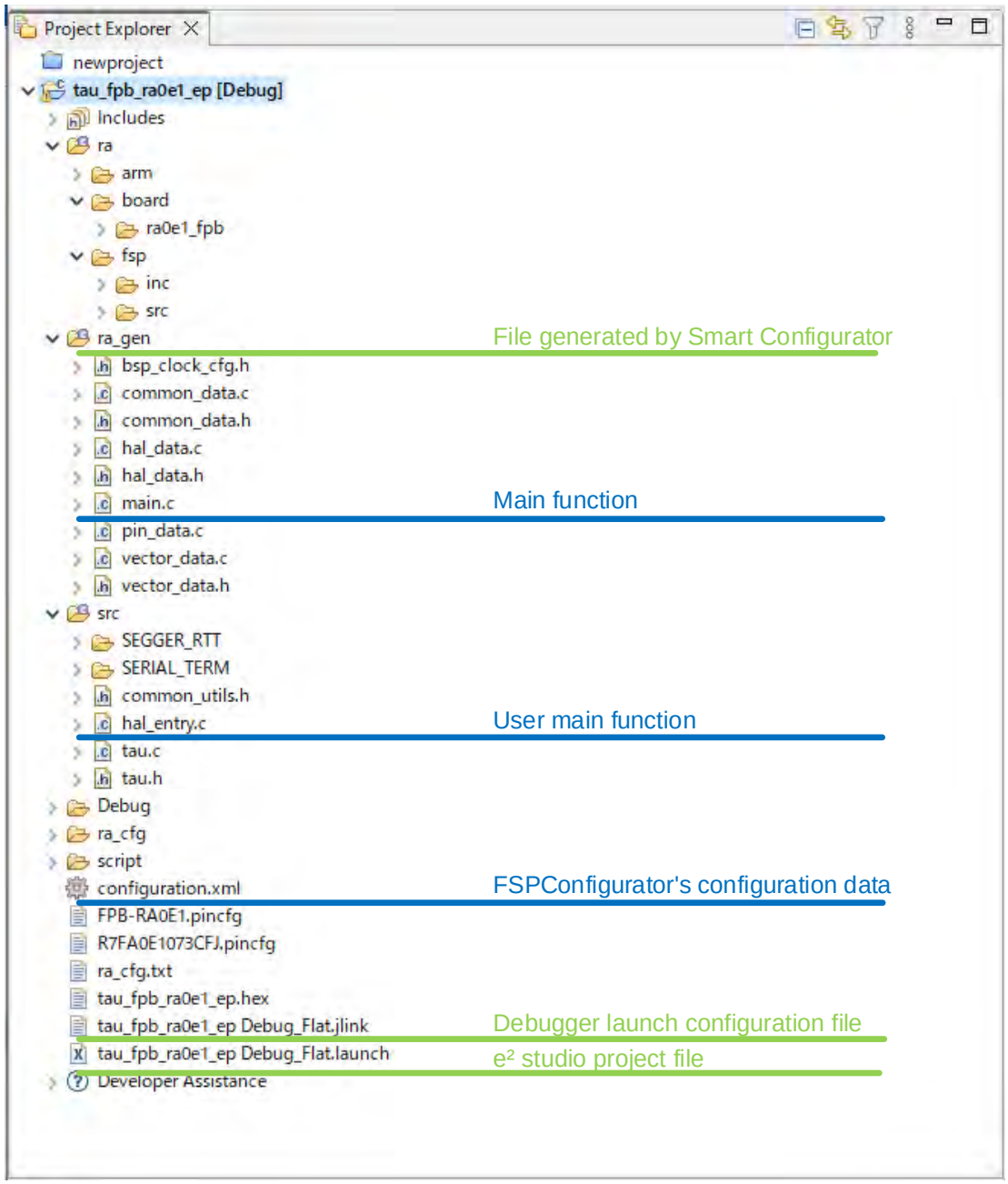


Figure 2-25 e² studio Project Files

(3) Just like double-clicking the .syscfg file to launch SysConfig, double-click the configuration.xml file to start the FSP Configurator, which allows you to configure the necessary peripheral functions through the graphical interface.

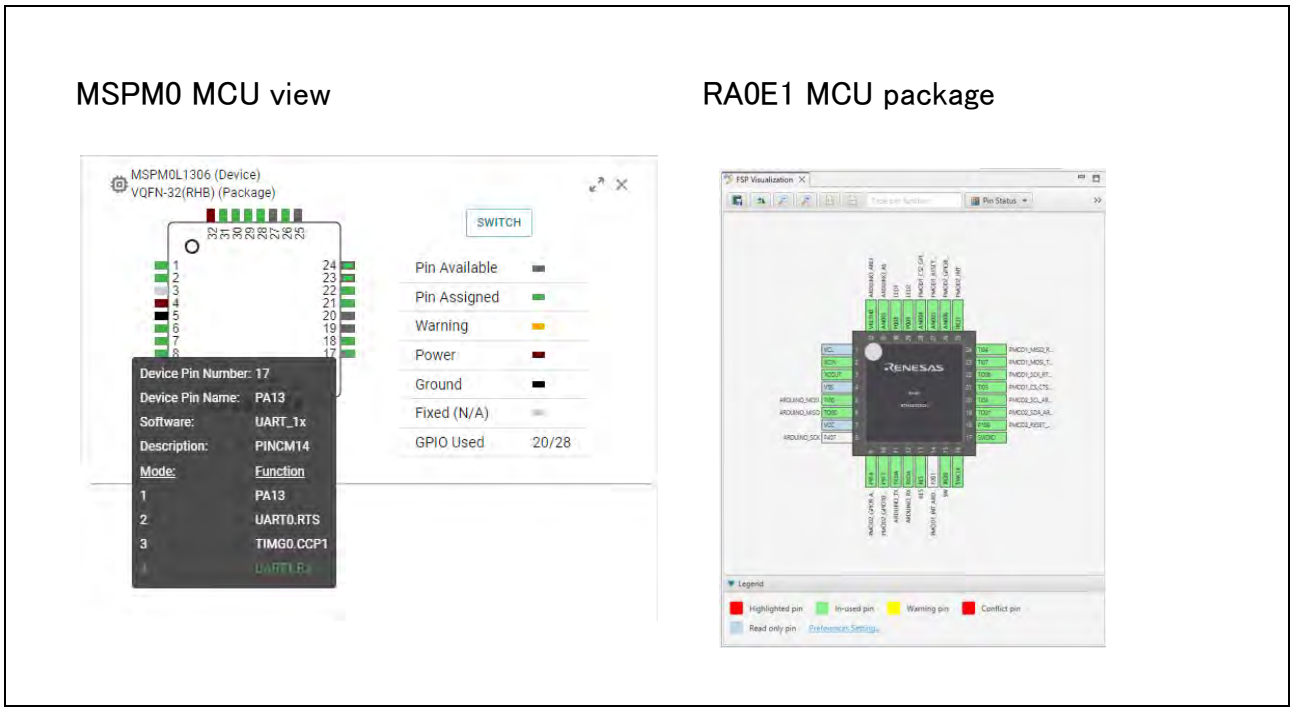


Figure 2-26 SysConfig and MCU View in FSP Configurator

(4) The user can improve and add functions to the sample program as needed; refer to the corresponding application notes published by Renesas.

(5) When the software configurations or modifications are complete, click the [Build] icon on the Main toolbar as shown in Figure 2-27.
When the building is done successfully, “Build Finished” is displayed.

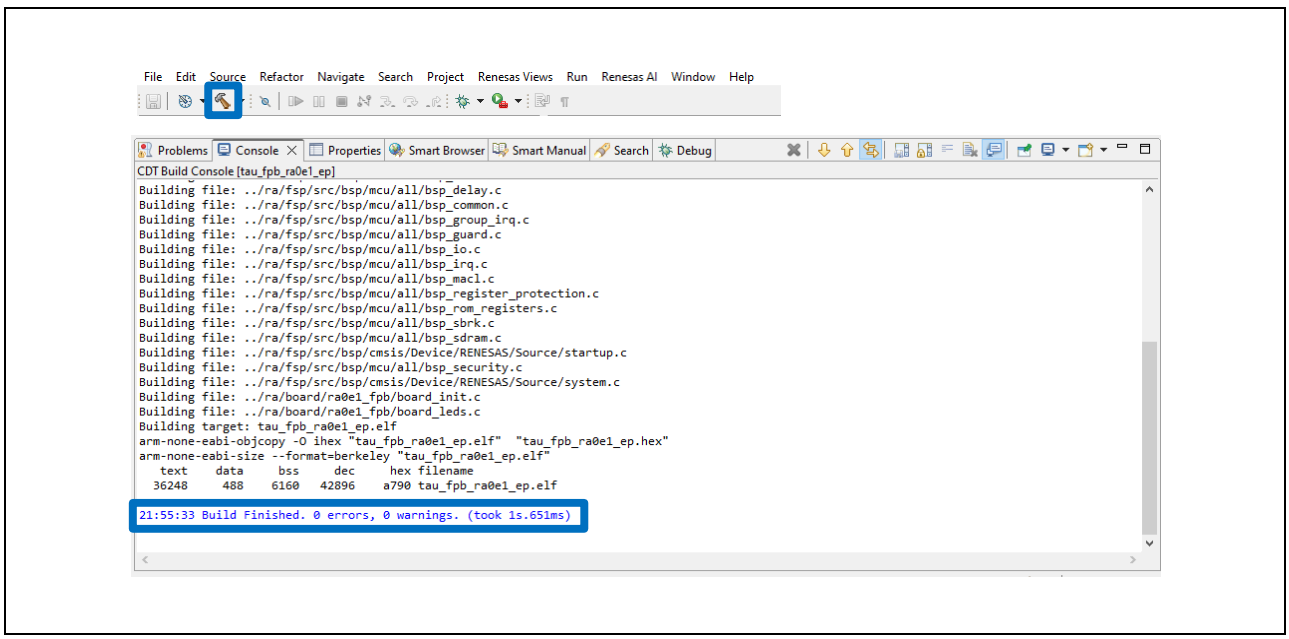


Figure 2-27 Build Finished

2.2.4 Step 4: Evaluating the Software

Use the debugging functions described in Section 2.2.2.2 How to Use e² studio, to check whether the software's performance and operation meet expectations.

2.2.5 Step 5: Mass Production

(1) Generate the load module (.hex, .mot, .bin) using e² studio.

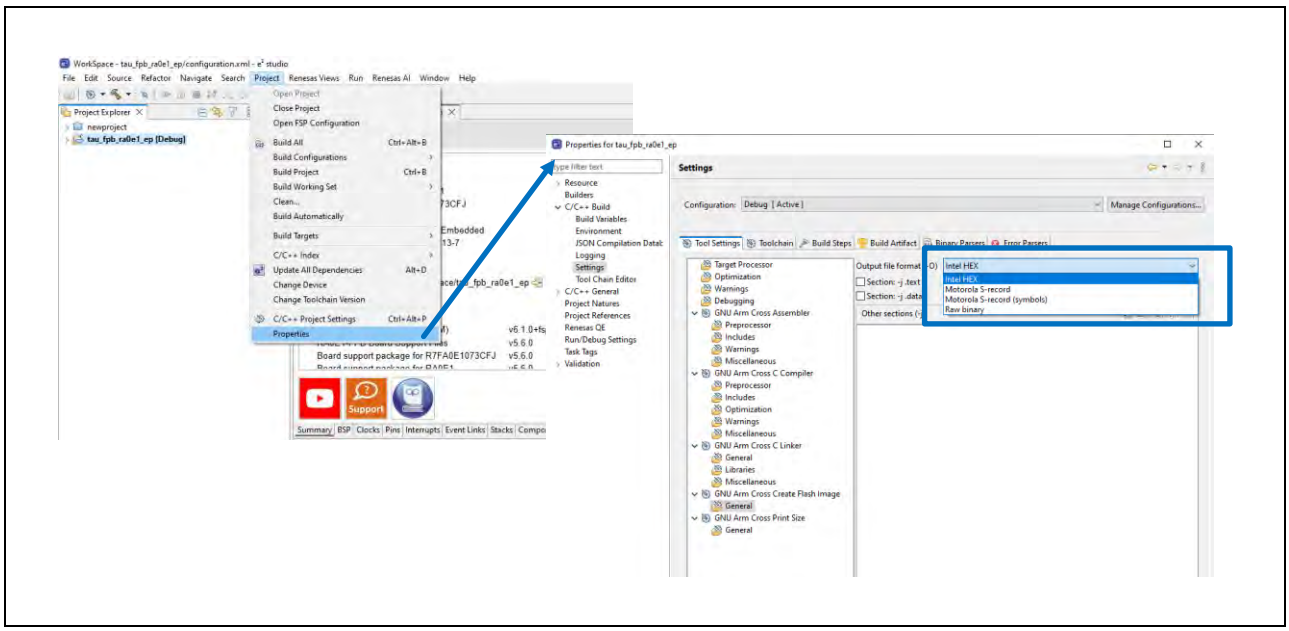


Figure 2-28 Create the Program File

(2) Select the programmer/debugger to program RA0E1

Up to 12 PG-FP6 units can be connected to one PC, allowing simultaneous programming to the target MCUs.

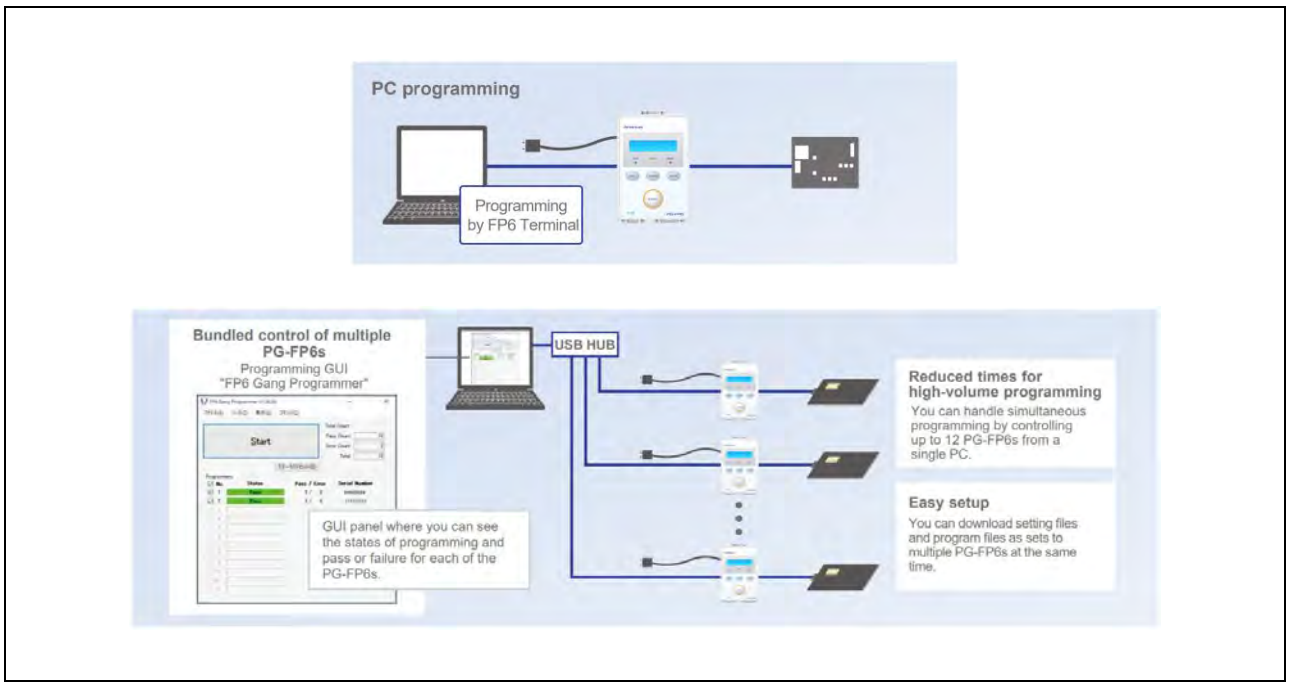


Figure 2-29 Flash Programming and Tools

For details regarding the PG-FP6 and FP6 Terminal, refer to the PG-FP6 V1.13 Flash Memory Programmer User's Manual (R20UT5518).
For details regarding the PG-FP6 Gang Programmer, refer to the FP6 Gang Programmer V1.03 User's Manual (R20UT5403).

2.3 Example of Importing a Sample Program

This section shows an example of the migration process using e² studio. This example is board migration using the program provided by Renesas. Here, the RA0E1 Fast Prototyping Board is used as an example.

(1) Select the appropriate RA0E1 and the hardware before ordering the evaluation board. This example uses RA0E1 Fast Prototyping Board.

(2) Setup e² studio. For details, refer to 2.2.2 Step 2: IDE Setup and Quick Introduction of e² studio.

(3) Import the sample code.
After you have arranged the proper environment, the code can be imported to e² studio. The example uses the timer to control the LED. First, make sure you confirm the differences between the RA0E1 and MSPM0 timer modules. Then select the applicable sample code from the Renesas website.

In this section, we will explain using the "tau_fpb_ra0e1_ep", a sample of the timer function, from the "FPB-RA0E1 Example Project Bundle".

Select **[File]** - **[Import…]** from the menu to open the **Import** dialog box.

Since we are importing from the Renesas website to the workspace, select **[General]** - **[Sample Projects on Renesas website]** and click **[Next]** to open the **Import sample project on Renesas website** dialog box.

After selecting the **Device** (in this case, select "R7FA0E107"), a list of sample projects matching the device will be displayed.

From the list, select the sample project “FPB-RA0E1 Example Project Bundle” and press the **[Finish]** button. This “FPB-RA0E1 Example Project Bundle” consists of multiple project files, so a list of the projects contained within it will be displayed.

From this list, select the "tau_fpb_ra0e1_ep" sample project, which demonstrates the timer function, and press the **[Finish]** button to add the project to the workspace.

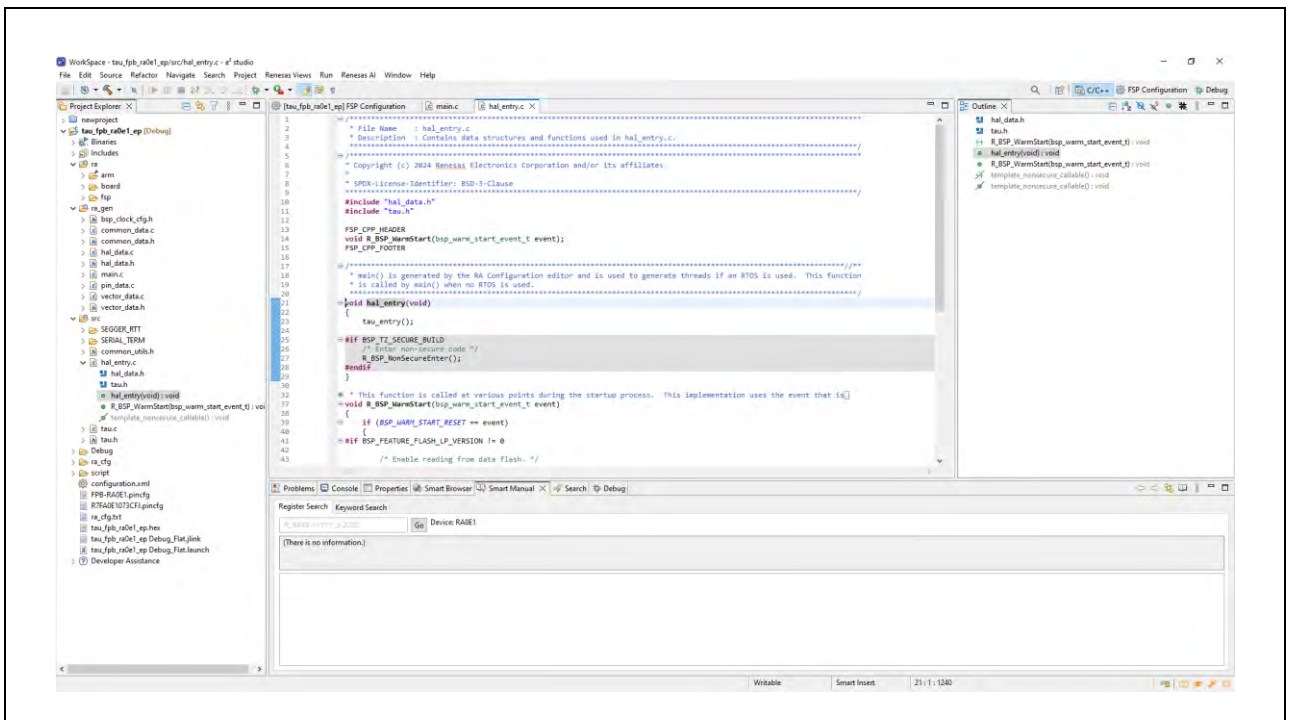


Figure 2-30 Sample Code File

(4) Confirm the Project Configuration.
To confirm the configuration in the FSP Configurator, open the configuration.xml file in the workspace.

Changes to the configuration will be necessary if the sample program differs from the hardware.

This example shows how to change the custom board to the FPB-RA0E1 board. From the FSP Configurator, select the **BSP** tab. As shown in Figure 2-31, select the dropdown list box or [...]. select the Board from the menu, then then select **FPB-RA0E1**.

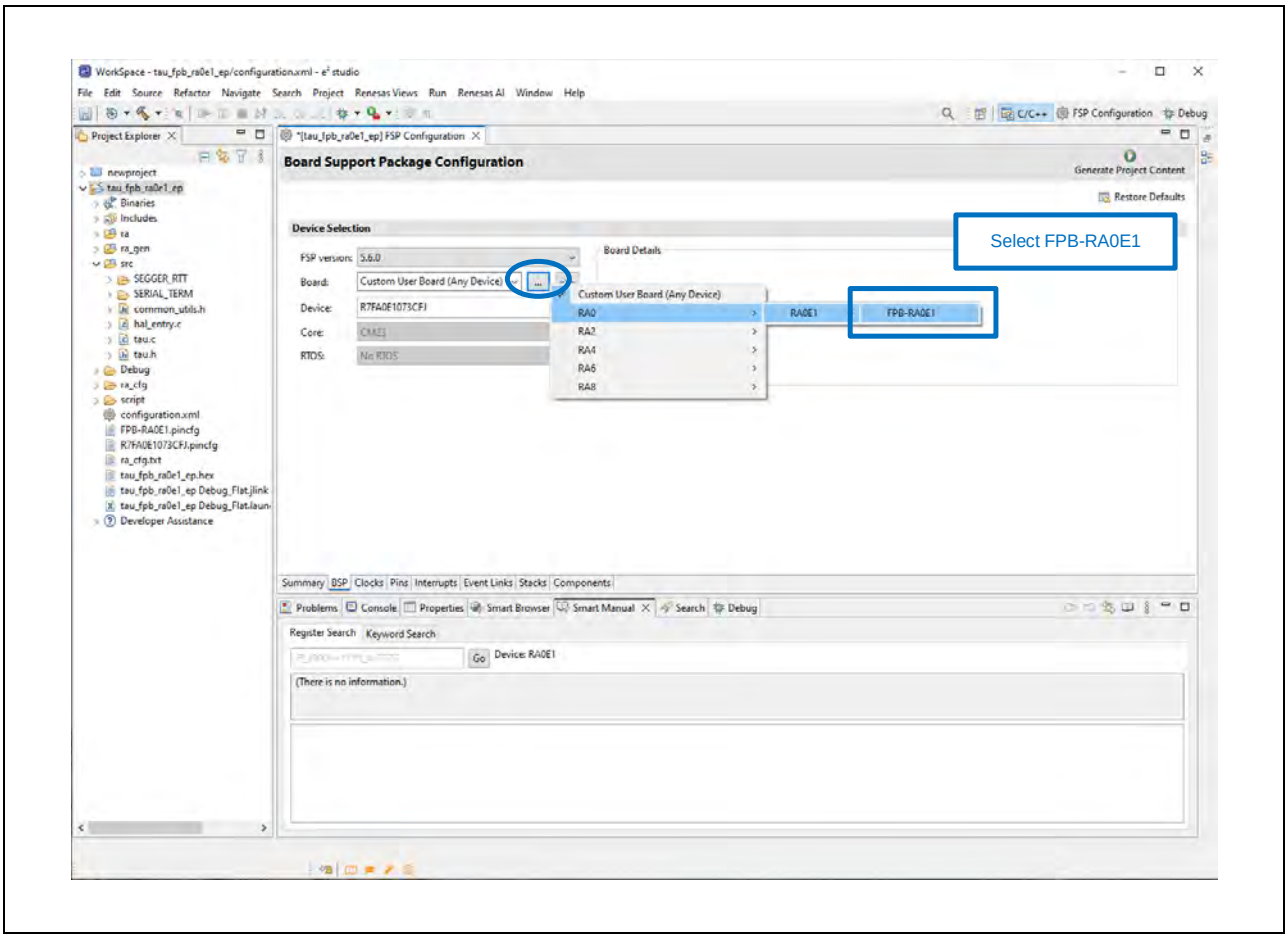


Figure 2-31 Device Selection in FSP Configurator

Confirm that the LED and port assignments are as follows: LED1 = P008 and LED2 = P009.
Select the **Pins** tab and then select the **Pin Function** tab.
Finally, select the port to confirm the port's settings.

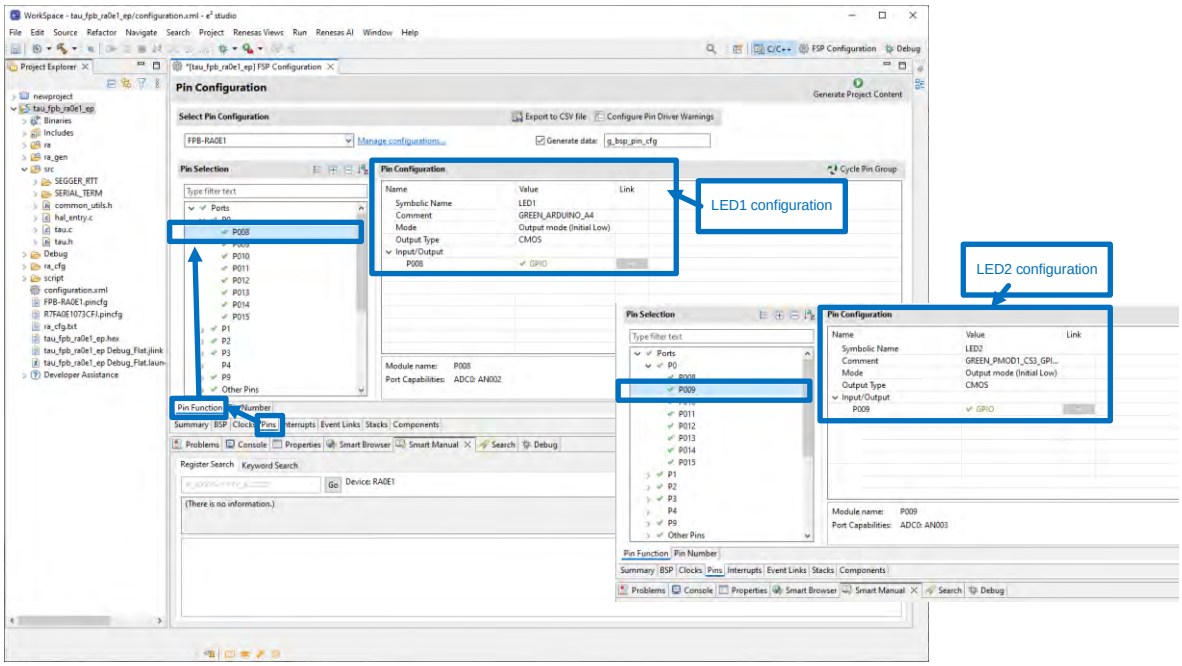


Figure 2-32 Board Setting in FSP Configurator

Next, confirm the timer settings. When you select the **Stacks** tab, the settings of the selected stack will be displayed in the Properties page, as shown in Figure 2-33.

Also confirm the timer channel and timer operation mode. In this example, the timer is set to channel 2 and the operation mode to interval timer.

By pressing the FSP Configurator **Generate Project Content** button, the source code will be generated based on the content of these settings.

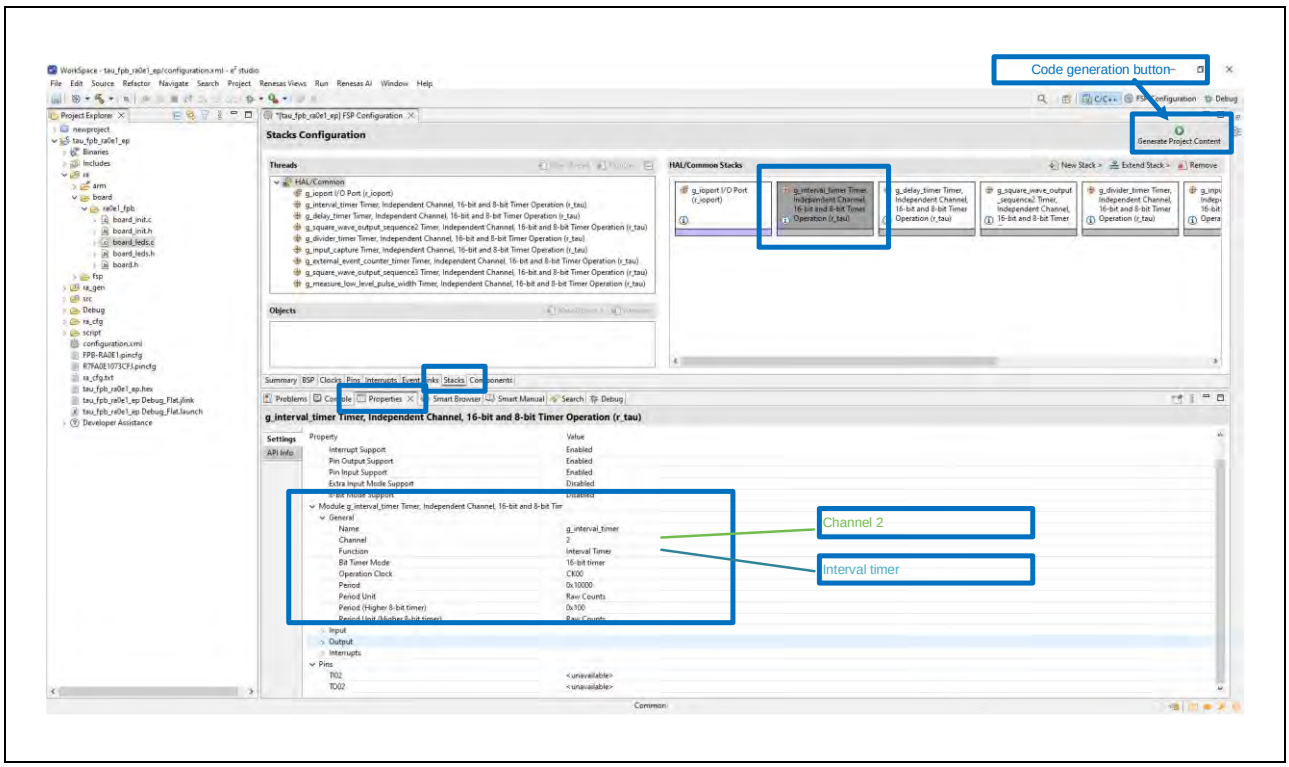


Figure 2-33 Interval Timer Settings in FSP Configurator

Developer Assistance allows you to reference the API information regarding each stack used in the project.

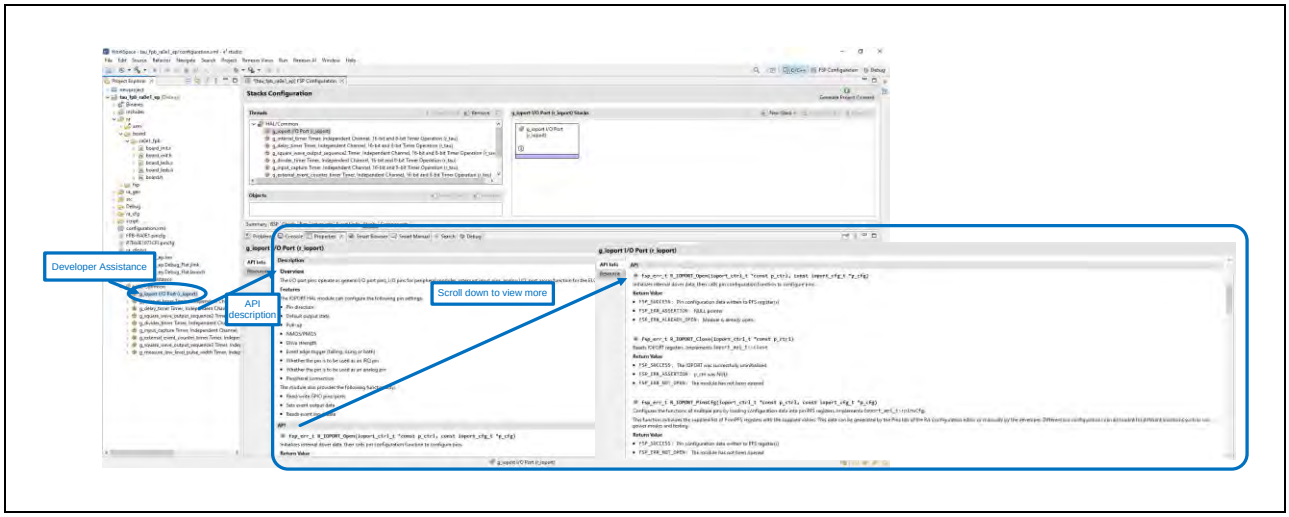


Figure 2-34 API Information Displayed in Developer Assistance

In addition, the Coding Assist feature allows you to drag and drop the API onto the code.

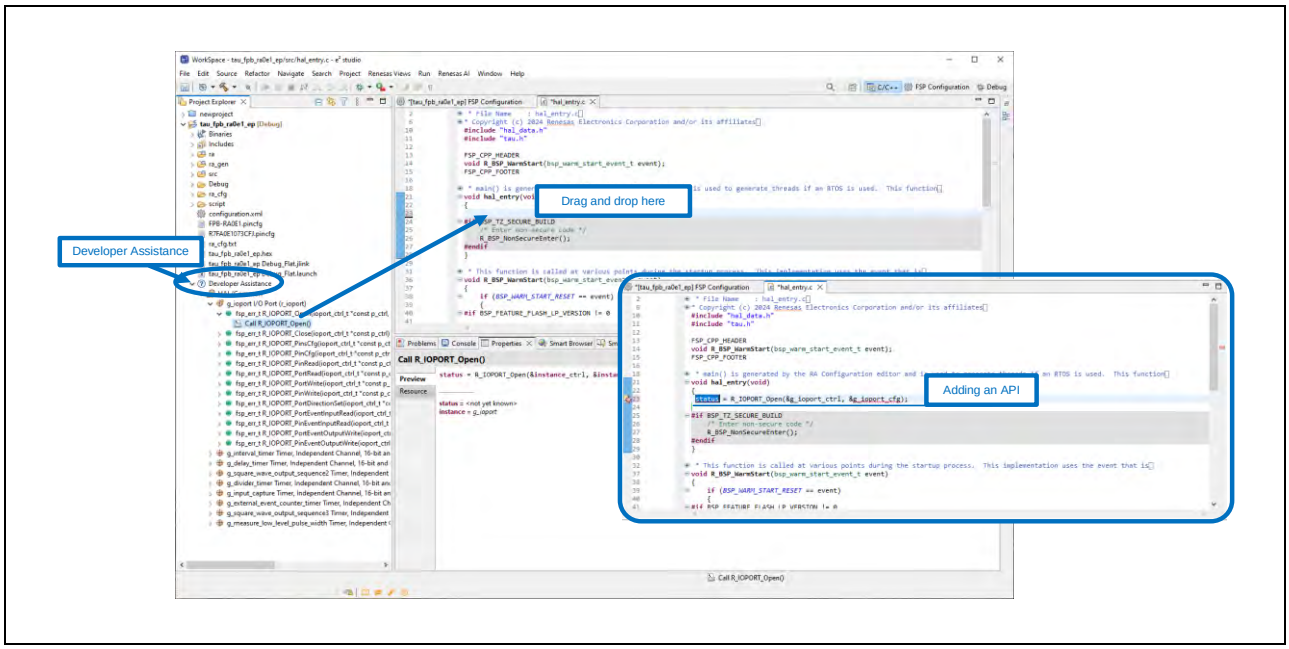


Figure 2-35 Coding Assist Function in Developer Assistance

The information icon displayed on each stack provides API information as well additional information such as module overviews and sample codes.

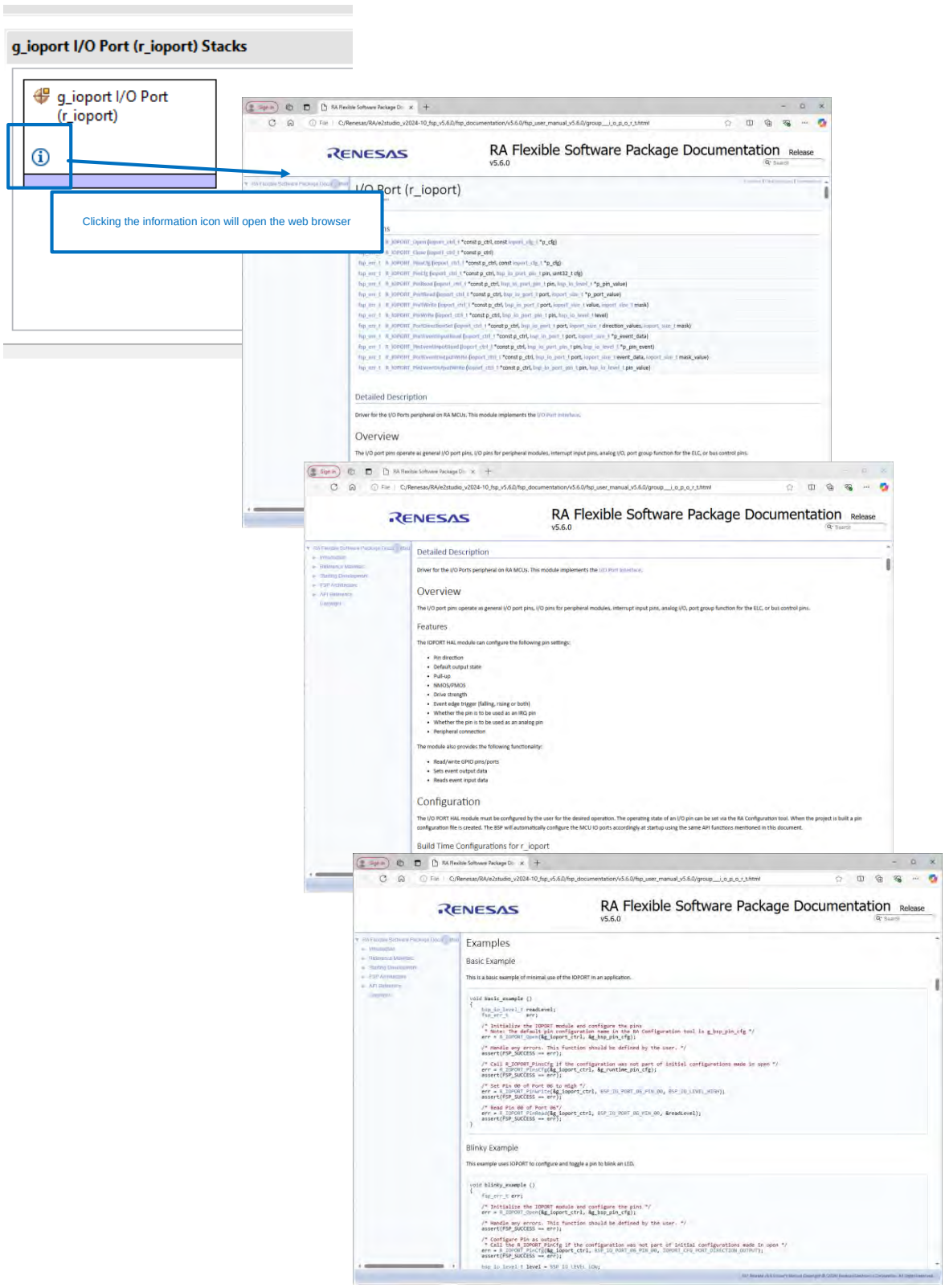


Figure 2-36 Reference Additional Info via Information Icon

(5) Build the Project.
When the software configurations or modifications are complete, click the **Build** icon in the Main toolbar as shown in Figure 2-37.

When the building is done successfully, "Build Finished" is displayed.

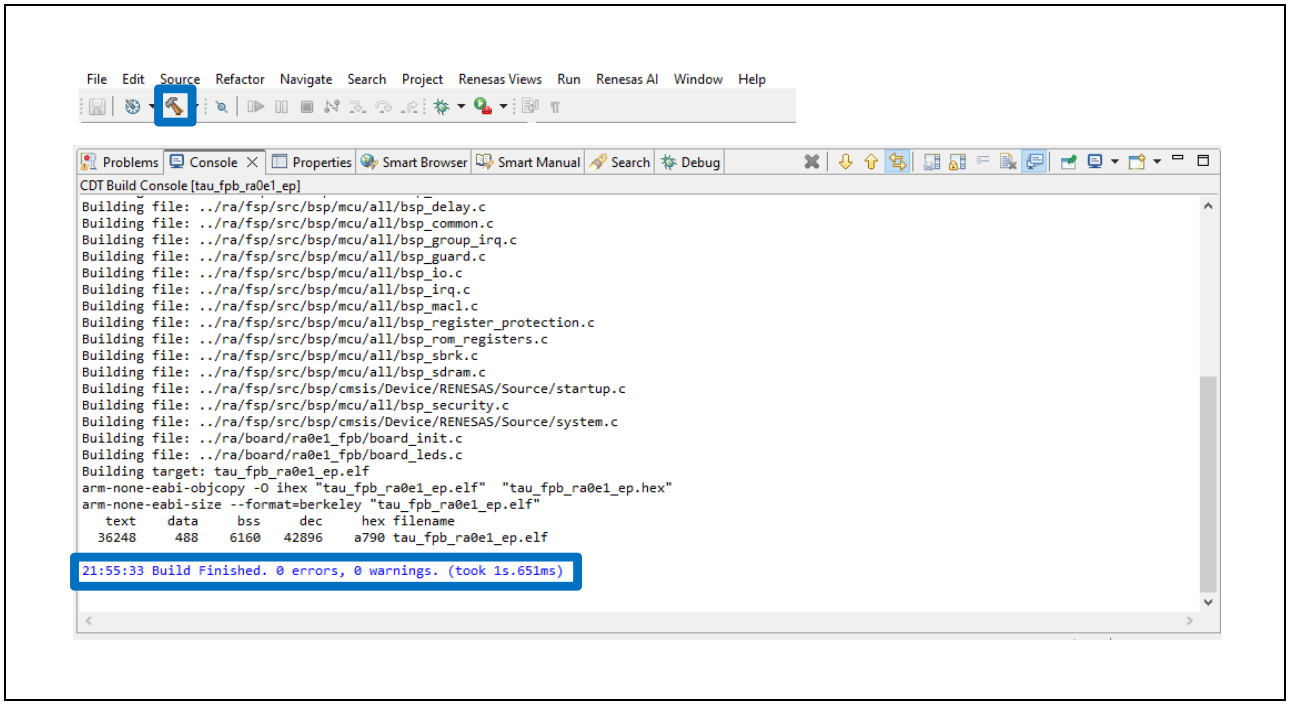


Figure 2-37 Build the Target

(6) Setup the Hardware.
Connect the FPB-RA0E1 to your PC using the USB cable.

(7) Debug to verify.
Configure the debugger according to Figure 2-23 before debugging.

Click the **Debug** icon to start debugging. You can set a breakpoint by double-clicking the space before the line number, or by adding __BKPT();, just like in MSPM0.

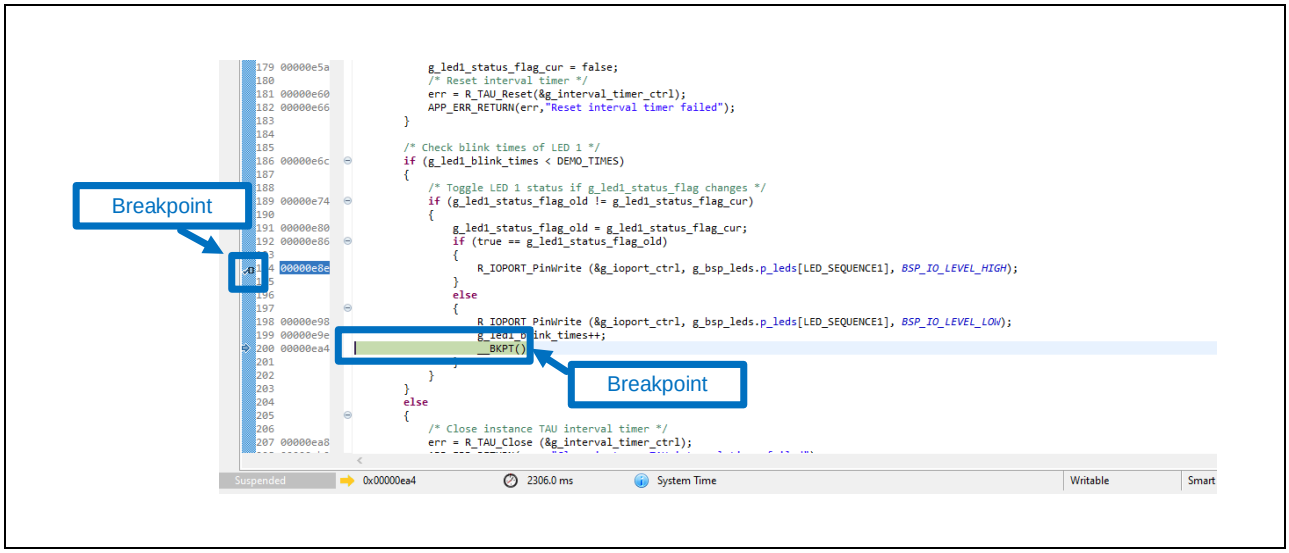


Figure 2-38 Adding Break Point Solution

Use the debugging function to check if the performance meets the requirements and if there are no errors in the program (refer to Section 2.2.2.2 How to Use e² studio for details).

3. Core Architecture Comparison

3.1 CPU

The RA0 is based on the Arm Cortex -M23 CPU core architecture and the MSPM0 Family is based on the ARM Cortex M0+ CPU core architecture. Table 3-1 shows a general functional overview, comparing the RA0 CPU to the MSPM0 CPU.

Table 3-1 CPU Comparison

Item	MSPM0	RA0
Architecture	Arm Cortex M0+	Arm Cortex M23
Instruction set	RISC	RISC
Pipeline	2-stage	2-stage
Operating clock (max.)	80/32/24 MHz	32 MHz
DMA	Available	Available (connected to DTC)
Coremark/MHz	2.46 Note	2.64 Note

Note Scores are quoted from the “Arm Cortex-M Processor Comparison Table” posted on the official ARM website.

3.2 Built-in Memory Comparison

3.2.1 Flash Memory Function

RA0 and MSPM0 MCUs feature non-volatile flash memories for storing executable program code and application data. Table 3-2 shows a comparison of flash memories.

Table 3-2 Flash Memory Function Comparison

Feature	MSPM0	RA0	
Flash memory	MSPM0Lxx: 8KB to 64KB MSPM0Cxx: 8KB or 16KB	Code flash	16KB～128KB
		Data flash	1KB～2KB
Single flash size	64 bits	Code flash	32 bits
		Data flash	8 bits
Memory organization	Sector size (512B or 1KB) Bank size (variable) Max 256KB device – 1 bank 256KB or higher device – 2 banks	Code flash	Block size: 2KB
		Data flash	Block size: 256B
Access	Single flash word (64 bits) or multiple words	8 bits	
Programming unit	Single flash word (64 bits) or multiple words	Code flash	Single flash word (32 bits)
		Data flash	Single flash word (8 bits)
Erasing unit	Sector erase Bank erase (max 256KB)	Code flash	2KB
		Data flash	256B
Data error correction/detection	ECC	N/A	
Write protection	Available (static and dynamic)	Available	
Read protection	Available	Code flash	Available
		Data flash	Available
Programming count	100k (lower 32KB) or 10k (higher 32KB)	Code flash	1 k (min.)
		Data flash	1000k (typ.)

3.2.2 Flash Memory Configuration

The RA0 flash memory is equipped with a code flash for executing programs and a data flash used as a data storage area.

The code flash and the data flash can be rewritten by the flash memory programmer, SWD programming using an external device (SWD connection), or self-programming.

3.2.2.1 Flash Memory Area

Table 3-3 shows the flash memory areas in MSPM0 and RA0.

Table 3-3 Flash Memory Area Comparison

MSPM0		RA0 (Code Flash)	
MAIN	Application and data	Program area	
NONMAIN	BCR configuration	Startup area	Small program area
			On-chip debugging ID setting
	BSL configuration		Flash serial programming security ID
			Option setting memory
FACTORY	Device ID		Unique ID
DATA Note	Data or EEPROM emulation		Vector table

Note MSPM0 devices with one bank implement the FACTORY, NONMAIN, and MAIN regions on BANK0 (the only bank present), and the DATA region is not available.
MSPM0 devices with multiple banks also implement FACTORY, NONMAIN, and MAIN regions on BANK0, but include additional banks (BANK1 through BANK4) that can implement MAIN or DATA regions

3.2.2.2 Flash Memory Control Register in RA0

The flash I/O registers are used to control the programming of the flash memories. For example, the FLMWRP register enables or disables programming of the code and data flash memories, and the DFLCTL register is used to enable or disable access to the data flash.

3.2.2.3 Self-Programming in RA0

RA0 supports the self-programming mode that can be used to rewrite the flash memory with a user program.

This mode allows the user application to rewrite the flash memory by using the RA0 microcontroller self-programming module (r_flash_lp).

3.2.3 SRAM

Both RA0 and MSPM0 are both equipped with an on-chip SRAM for storing the application data.

Table 3-4 SRAM Comparison

Type	MSPM0	RA0
SRAM memory	MSPM0Lxx 2KB to 32KB MSPM0Cxx 8KB to 32KB	2KB～16KB
Parity check	Available (MSPM0Cxx: N/A)	Available
ECC	Available (MSPM0Cxx: N/A)	N/A
Write protection (RAM guard)	Available	N/A

RA0 is equipped with a low-power SRAM that can be accessed efficiently within the CPU frequencies supported by the device. SRAM can be used to store information such as the call stack, heap, and global data. The contents of the SRAM are held regardless of changes in the low-power modes (normal operation, SLEEP, Software STANDBY, SNOOZE, etc.), but if the RAM operation mode is set to SHUTDOWN mode in the Power Save Memory Control Register (PSMCR), the contents in some areas of the SRAM will be lost.

The SRAM in RA0 can be used as a program area to store code and execute instructions, as well as a data area.

3.3 Power-On and Reset Overview and Comparison

Both MSPM0 and RA0 feature a minimum operating voltage and include a circuit that ensures proper startup by holding the device, or part of it, in the reset state. Table 3-5 shows the comparison of how this function works and which modules control the power-on process and reset between the two devices.

Table 3-5 Power-On Overview and Comparison

MSPM0		RA0	
Power-on reset (POR)	Rising edge detection: When V_{DD} > POR+, the POR state is released and bandgap reference and BOR are started. Falling edge detection: When V_{DD} < POR-, the device is held in the POR state.	POR (power-on reset circuit)	Rising edge detected: When V_{CC}>V_{POR}, POR reset is released. Falling edge detected: When V_{CC}<V_{PDR}, POR reset signal is generated.
Brown-out reset (BOR)- Level 0 Note 2	Rising edge detection: V_{DD} > BOR0+, the device continues the boot process, and PMU is started. Falling edge detection: When V_{DD} < BOR0-, the device is held in the BOR state.	LVD0 (low-voltage detector) – voltage detect0 reset mode Note 1	Rising edge detected: When V_{CC}>V_{det0}, LVD reset is released. Falling edge detected: When V_{CC}<V_{det0}, LVD reset signal is generated
Brown-out reset (BOR) -Level 1 to 3 Note 2	Falling edge detection: 1) When V_{DD} < BORx- (x = 1, 2, or 3), an interrupt request occurs and the BOR circuit automatically switches the BOR threshold level to BOR0. 2) When V_{DD} < BOR0-, the device is held in the BOR state.	LVD1 (low-voltage detector) – voltage detect1 interrupt and reset mode Note 1	Rising edge detected: When V_{CC}>V_{det1}, LVD reset is released Falling edge detected: 1) When V_{CC}<V_{det1}, an interrupt request signal is generated 2) When V_{CC}<V_{det0}, LVD reset signal is generated
Not applicable	N/A	LVD1 (low-voltage detector) – voltage detect1 interrupt and reset mode Note 1	When V_{CC}>V_{det1} or V_{CC}<V_{det1}, an interrupt request signal is generated

Note 1. The level of voltage detection can be selected in the Option-setting memory or the LVD1CR register.

Note 2. There are four selectable BOR threshold levels (BOR0-BOR3). During startup, the BOR threshold is always BOR0 (the lowest value) to enable the device to start at the specified minimum V_{DD} value. After startup, the software can optionally reconfigure the BOR circuit to use a different (higher) threshold level.

The relationship among RA0 voltage thresholds is $V_{PDR} < V_{POR} <$ lower limit of the operating voltage $< V_{det0} < V_{det1}$. The relationship among MSPM0 voltage thresholds is POR- < POR+ < BOR0- < BOR0+. BOR0+ is the smallest value of V_{DD} specified to enable the internal circuit's normal operation.

The following 6 events can generate a reset signal in RA0.

(1) External reset input with the RES pin
(2) Internal reset due to comparison of power supply voltage and detection voltage in the power-on reset circuit (POR)
(3) Internal reset due to comparison of power supply voltage and detected voltage of the voltage detection circuit (LVD0, 1)
(4) Internal reset due to SRAM parity error
(5) Internal reset due to program runaway of the independent watchdog timer
(6) Software reset

There are many events that trigger an internal reset in RA0. Use the reset control flag register (RESF) and the power-on reset status register (PORSR) to check which event triggered the reset request.

Table 3-6 Reset Flag Status when Reset is Requested (1/2)

Flag	Reset Event			
	RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset
PORSR.PORF	Hold	Clear	Hold	Hold
RESF.LVIRF	Clear	Clear	Set	Hold
RESF.IWDTRF	Clear	Clear	Hold	Set
RESF.SWRF	Clear	Clear	Hold	Hold
RESF.RPERF	Clear	Clear	Hold	Hold

Table 3-7 Reset Flag Status when Reset is Requested (2/2)

Flag	Reset Event		
	Voltage monitor 1 reset	Software reset	SRAM parity error reset
PORSR.PORF	Hold	Hold	Hold
RESF.LVIRF	Set	Hold	Hold
RESF.IWDTRF	Hold	Hold	Hold
RESF.SWRF	Hold	Set	Hold
RESF.RPERF	Hold	Hold	Set

3.4 Clock Overview and Comparison

3.4.1 Oscillator

Both RA0 and MSPM0 are equipped with many types of internal and external clock generation circuits to reduce system cost and power consumption. Table 3-8 shows the comparison of the clock circuits in RA0 and MSPM0.

Table 3-8 Oscillator Comparison

Type	MSPM0L	MSPM0C	RA0
Internal oscillator	SYSOSC: 4 MHz to 32 MHz internal oscillator	SYSOSC: 24 MHz internal oscillator	HOCO: High-speed on-chip oscillator clock with maximum frequency of 32MHz
			MOCO: Middle-speed on-chip oscillator clock with maximum frequency of 4MHz
	LFOSC: 32 kHz internal low-speed oscillator	LFOSC: 32 kHz internal oscillator	Low-speed on-chip oscillator clock with maximum frequency of 32.768 kHz
External oscillator	N/A	N/A	MOSC: Main clock oscillator with maximum frequency of 20MHz
	N/A	N/A	SOSC: Sub clock oscillator with frequency of 32.768 kHz

3.4.1.1 RA0 Oscillator

RA0 is equipped with an on-chip oscillator that eliminates the need for an external resonator, as well as many types of oscillator circuits, including a low-speed oscillator circuit that achieves low-power consumption. Table 3-9 shows RA0's clock generation circuits. Refer to the RA0 Group User's Manual: Hardware for details.

Table 3-9 RA0 MCU Oscillators

Clock Generation Circuit	Clock Signal	Description
Internal	HOCO	High-speed on-chip oscillator clock (32 MHz (max.))
	MOCO	Middle-speed on-chip oscillator clock (Max. 4 MHz (max.))
	LOCO	Low-speed on-chip oscillator clock (32.768 kHz)
External	MOSC	Main clock oscillator (20MHz (max.))
	SOSC	Sub clock oscillator (32.768 kHz)

3.4.2 Clock Signal Comparison

Different clock signals can be divided and used as input clocks for other systems or distributed to multiple peripheral devices.

Table 3-10 Clock Signal Comparison

Clock Description		MSPM0 Clock	RA0 Clock
External clock input	High-speed	HFCLK_IN	EXCLK (MOSC)
	Low-speed	LFCLK_IN	XCIN (SOSC)
High-speed external clock		HFCLK	EXCLK
Low-speed external clock		Selectable from LFCLK_IN and LFXT	FSUB Note 1
PLL circuit output clock		SYSPLLCLK0, SYSPLLCLK1, SYSPLLCLK2x Note 2	N/A
Main system clock		MCLK, ULPCLK Note 3 (BUSCLK)	FMAIN Note 4
CPU/peripheral high-speed clock		Selection of HSCLK Note 5, and SYSOSC	FMAIN
CPU/peripheral low-speed clock		LFCLK (fixed to 32 kHz)	FSUB, LOCO
Source CPU		CPUCLK	ICLK
Most peripheral hardware clocks		MCLK, ULPCLK	FMAIN, ICLK
Clock available for high-speed peripherals		MCLK	HOCO, MOCO, FMAIN, ICLK
Clock available for low-speed peripherals		ULPCLK	LOCO, FSUB, ICLK
Fixed frequency clock		MFCLK: 4 MHz, synchronized to MCLK	N/A
		MFPCLK: 4 MHz	

Note 1. FSUB is the subsystem clock supplied by a low-speed external oscillator (XCIN).
Note 2. SYSPLLCLK2x operates at twice the speed of the PLL output and can be divided as needed.
Note 3. The MSPM0 main system clock is supplied by LFCLK, HSCLK or SYSOSC. MCLK serves as the main system clock for PD1, and ULPCLK, which is derived from MCLK, acts as the main system clock for PD0.
Note 4. RA0's main system clock is supplied by MOSC, HOCO or MOCO.
Note 5. HSCLK is supplied by SYSPLL or HFCLK.

Table 3-11 Peripheral Clock Inputs

Peripheral	MSPM0	RA0
Realtime clock (RTC)	LFCLK (LFOSC, LFXT)	SOSC, LOCO
UART	BUSCLK, MFCLK, LFCLK	MOSC, SOSC, HOCO, MOCO, LOCO
SPI/CSI (Simplified SPI)	BUSCLK, MFCLK, LFCLK	MOSC, SOSC, HOCO, MOCO, LOCO
I^2C	BUSCLK, MFCLK	MOSC, SOSC, HOCO, MOCO, LOCO
ADC	ULPCLK, HFCLK, SYSOSC	MOSC, SOSC, HOCO, MOCO, LOCO
Timer	BUSCLK, MFCLK, LFCLK	MOSC, SOSC, HOCO, MOCO, LOCO

3.5 Overview and Comparison of Operation Modes

RA0 provides four main operation modes (low-power modes) to optimize device power consumption based on the application requirements. In descending order of power, the modes are normal operation, SLEEP, SNOOZE, and software STANDBY. In the normal operation mode, the CPU is active and executes code. Unmasked peripheral interrupt requests can wake up the device from SLEEP, SNOOZE, or software STANDBY mode to normal operation mode. In SLEEP mode, all peripheral devices are active while the CPU stops. In SNOOZE or software STANDBY mode, the clock oscillator circuit is stopped to reduce power consumption as much as possible.

3.5.1 Comparison of Operation Modes

Table 3-12, Table 3-13 compares operation modes in RA0 and MSPM0.

Table 3-12 Comparison of Operation Modes (1/2)

MSPM0			RA0	
Operation Mode		Description	Operation Mode	Description
RUN	0	MCLK and CPUCLK are high-speed clock sources (SYSOSC, HFCLK, or SYSPLL)	Normal Operation	CPU operates with the main system clock or the subsystem clock. All peripherals are active.
	1	MCLK and CPUCLK are run via LFCLK (32 kHz)		
	2			
SLEEP	0	CPU stops operating. SYSOSC remains enabled, the other high-speed oscillators are optional. Low-speed oscillators remain enabled. MCLK is run via the high-speed clock source.	SLEEP	CPU stops operating. The main system clock and the subsystem clock keep operating. All peripherals are active.
	1	CPU stops operating. SYSOSC remains enabled, the other high-speed oscillators are disabled. Low-speed oscillator remains enabled. MCLK is run via LFCLK.		
	2	CPU stops operating. High-speed oscillators are disabled. Low-speed oscillators remain enabled. MCLK is run via LFCLK.		

Table 3-13 Comparison of Operation Modes (2/2)

MSPM0			RA0	
Operation Mode		Description	Operation Mode	Description
STOP	0	CPU stops operating. SYSOSC remains as is. The other high-speed oscillators are disabled. Low-speed oscillators remain enabled. ULPCLK is limited to 4 MHz. PD0 is enabled and PD1 is disabled. Analog peripherals such as ADC can be active.	SNOOZE Note 1, 2	CPU stops operating. HOCO or MOCO starts operating, but main clock remains stopped. Status of subsystem clock during software STANDBY mode continues. Peripheral functions such as ADC, UART, CSI, and DTC can run without the CPU operating.
	1	Same as STOP0 except SYSOSC and ULPCLK are limited to 4 MHz.		
	2	CPU stops operating. High-speed oscillator including SYSOSC are disabled. ULPCLK operates at 32 kHz. PD0 is enabled and PD1 is disabled. ADC is not supported.		
STANDBY	0	CPU stops operating. High-speed oscillators are disabled. All PD0 peripherals receive ULPCLK and LFCLK. ADC is not supported.	Software STANDBY Note 1	CPU stops operating. HOCO and MOCO stop operating. Status of subsystem clock before going to software STANDBY mode is retained. Status of subsystem clock continues. Therefore, the peripherals that run on the subsystem clock can operate.
	1	Same as STANDBY0 except only TIMG0/1 receives ULPCLK or LFCLK.		
SHUTDOWN		Clocks are not available, and the device is shut down.	Software STANDBY	CPU stops operating. HOCO, MOCO and LOCO stop operating. Main system clock and subsystem clock stop operating, so the system also stops. Only external interrupts are enabled.

Note 1. CSI, SAU A/D converter and DTC transfers can be specified in SNOOZE mode.
Note 2. SNOOZE mode can be specified only when either the high-speed on-chip oscillator or the middle-speed on-chip oscillator has been selected for the CPU/peripheral clock (ICLK).

3.5.2 RA0 Functions in SLEEP Mode

In SLEEP mode, the clock supply to the CPU is stopped but the peripherals continue to operate.

SLEEP mode offers reduced power consumption in comparison to normal operation, and can be released automatically with an interrupt, allowing normal operations to resume immediately.

This mode is suitable for applications that require low power consumption and quick response.

3.5.3 RA0 Functions in SNOOZE Mode

As with sleep mode, the clock supply to the CPU is stopped in SNOOZE mode. However, only specific peripheral functions (UART, AD converter, etc.) can continue operating. Data reception via UART and data conversion by ADC can be completed independently of the CPU, while all other peripheral functions are completely halted.

SNOOZE mode reduces power consumption even more than SLEEP mode and is released when triggered by completion of data reception or AD conversion, allowing the CPU to start processing as required.

As normal operation can be resumed immediately when data reception or conversion is complete, this mode is suitable for applications that require real-time processing.

3.5.4 RA0 Functions in Software STANDBY Mode

STANDBY mode allows you to stop the main clock, high-speed on-chip oscillator, middle-speed oscillator, and low-speed on-chip oscillator, bringing power consumption to nearly zero.

The contents of the memories and the registers are held even when the CPU completely stops, so the processing can be continued after returning to normal mode. When a specific interrupt occurs, such as an external interrupt, independent watchdog timer (IWDT), interrupt or RTC interrupt, the CPU returns to normal operation from STANDBY mode.

For detailed specifications and the relationship between each peripheral function and STANDBY mode, refer to the corresponding chapter in the RA0 Group User's Manual: Hardware.

3.6 Comparison of Interrupts and Events

3.6.1 Interrupts and Exceptions

Both RA0 and MSPM0 register and map interrupts and exception vectors according to the available peripheral functions of the device. Table 3-14 shows an overview and comparison of interrupt vectors for each device family. An interrupt or exception with higher priority takes precedence over one with lower priority. If the processor is currently handling an interrupt, the processor will only be preempted by an interrupt with a higher programmable priority.

Table 3-14 Comparison of Interrupts

Peripheral Function	MSPM0	RA0
Interrupt types	Peripheral interrupts NVIC supports up to 32 native peripheral interrupt sources. Note 1	Maskable interrupts Internal interrupt request x 33 (max.) External interrupt request x 6 (max.)
	System exceptions Reset NMI Hard fault SVCall PendSV SysTick	Reset interrupts RES pin input, power-on reset, voltage detection, watchdog timer, SRAM parity error Software interrupts A software interrupt occurs when the SVC instruction is generated.
Priorities	Default priority NVIC number Note 2	Default priority Determined by the device Note 3
	System exceptions The priority levels for Reset, NMI, and hard fault are fixed levels, -3, -2, and -1, respectively. 4 programmable priority levels (0, 64, 128, 192) are for SVCall, PendSV, and SysTick.	Interrupt priority level Four programmable priority levels (0, 1, 2, 3) are available for maskable interrupts.
	Peripheral interrupts Four programmable priority levels (0, 64, 128, 192) are available.	
Priority setting	IPRx in NVIC The IPRx register is used to set the peripheral interrupt priority level.	NVIC IPRx Register NVIC IPRx is used to set the interrupt priority of a peripheral.
Interrupt mask	IMASK in the peripheral The IMASK register is used to configure which interrupt conditions propagate into an event.	ISER and ICER in NVIC Registers ISER and ICER are used to enable/disable peripheral interrupts.
	ISER and ICER in NVIC Registers ISER and ICER are used to enable/disable peripheral interrupts.	

Note 1. In addition to the NVIC, interrupt grouping modules (INT_GROUP0 and INT_GROUP1) can be placed on MSPM0 to allow interfacing of more than 32 peripheral interrupts to the NVIC.
Note 2. The NVIC number indicates the relative interrupt priorities if multiple NVIC interrupts have the same programmable priority.
Note 3. The default priority is used when multiple maskable interrupts with the same interrupt priority occur.

3.6.1.1 Interrupt Control in RA0

The Interrupt Controller Unit (ICU) in RA0 determines which event signals are linked to the Nested Vectored Interrupt Controller (NVIC) and the Data Transfer Controller (DTC) modules. It also controls non-maskable interrupts.

The ICU also has a function that allows the user to set interrupt priorities. This enables the user to control the priority of important processing when multiple interrupts occur simultaneously.

The ICU simplifies interrupt control and management by allowing users to enable or disable interrupts, clear interrupt flags, and more. This facilitates flexible control of interrupt functions based on specific situations, such as temporarily disabling other interrupts, while a particular function is running.

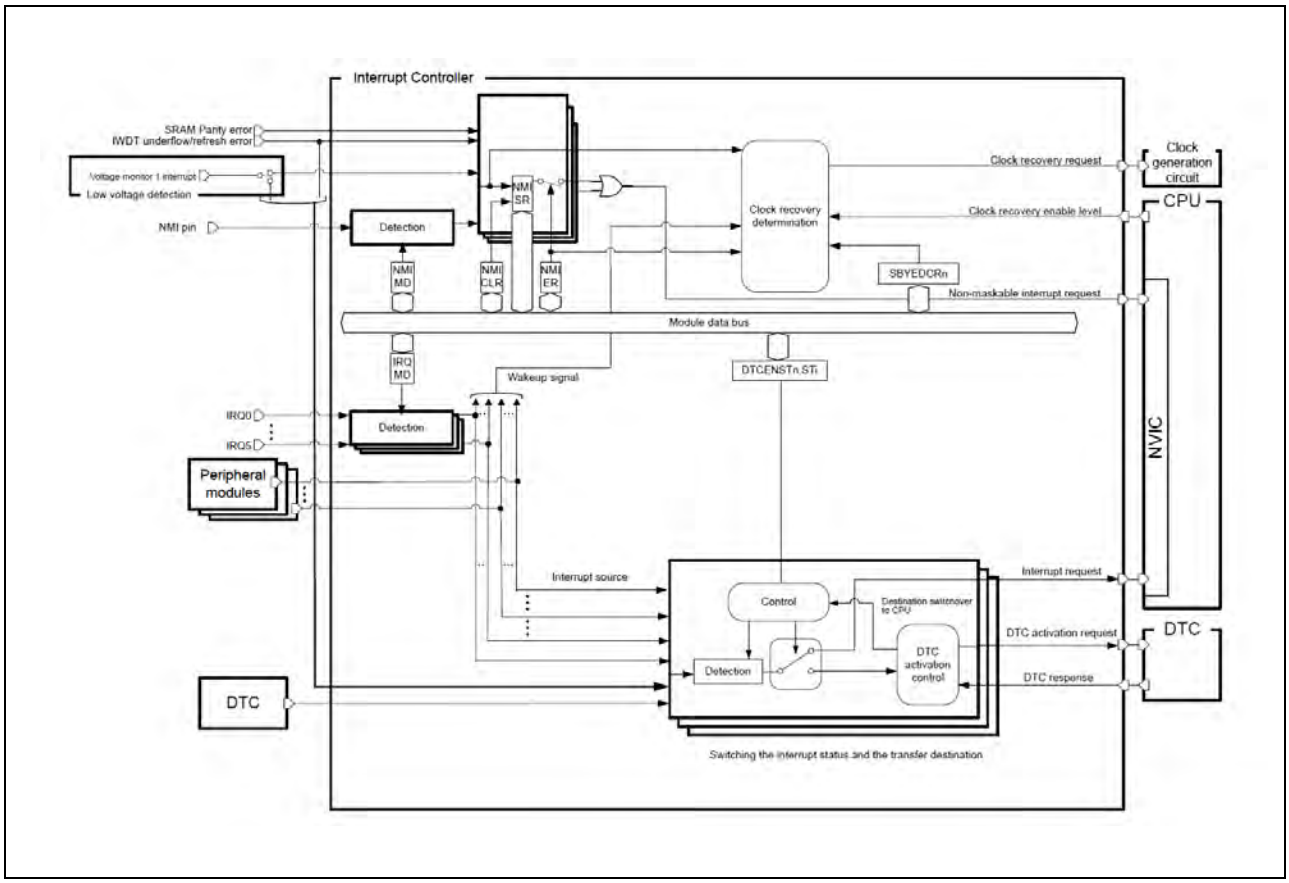


Figure 3-1 Configuration of RA0 Internal Makable Interrupts

3.6.1.2 Interrupt Control in MSPM0

As with RA0, MSPM0 sets the priority level of each peripheral interrupt source through the IPRx registers in the NVIC, and masks/unmasks the peripheral interrupt sources through the ISER and ICER registers in the NVIC. Each peripheral interrupt includes various interrupt conditions. For example, as a peripheral interrupt source, UARTx has multiple interrupt conditions, such as transmit interrupt and receive interrupt.
In addition, interrupt conditions are managed by the six standard registers on the peripheral side.

Figure 3-2 shows the peripheral interrupt hierarchy.

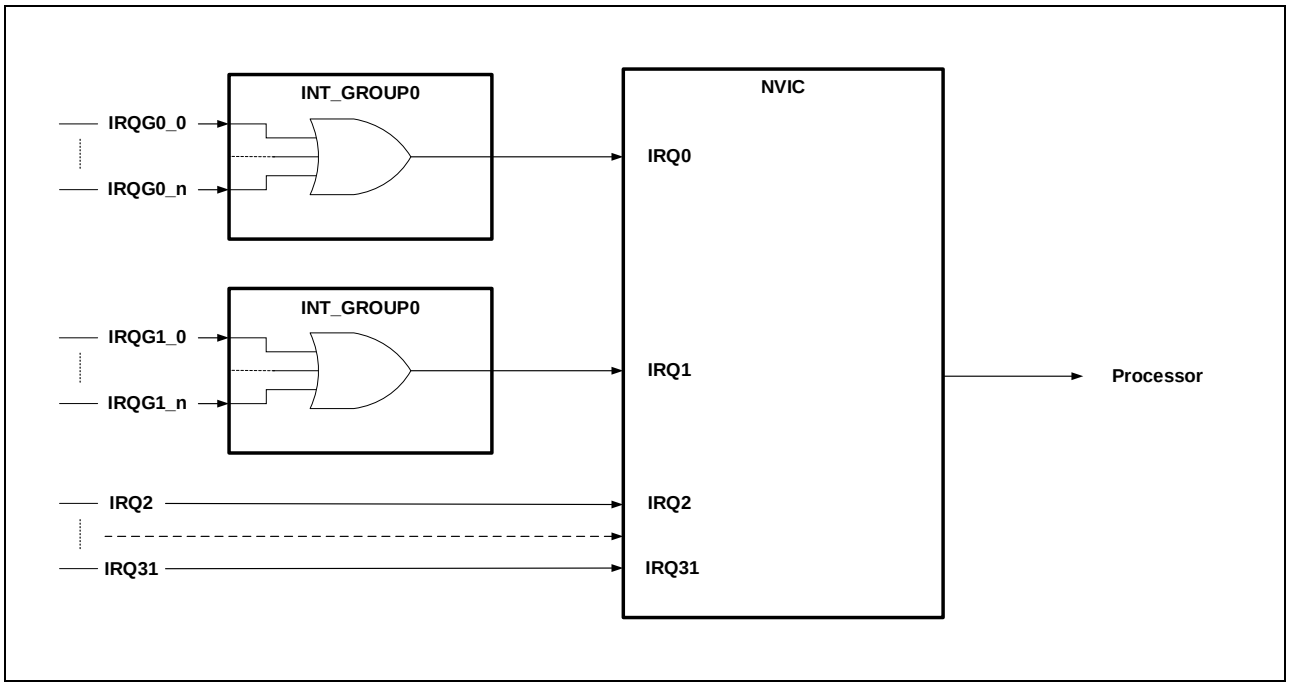


Figure 3-2 MSPM0 Peripheral Interrupt Hierarchy

3.6.2 MSPM0 Event Handler

MSPM0 MCUs are equipped with an event manager that transfers events from one entity to another. The event manager performs event transfer through the defined set of event publishers (generators) and subscribers (receivers) that are interconnected through an event fabric containing a combination of static and programmable routes. The event manager can also perform handshakes with the power management and clock unit (PMCU), to make sure that the necessary clock and power domain are present for triggered event actions to take place.

The events transferred by the event manager are as follows:

- Peripheral event transferred to the CPU as an interrupt request (IRQ)
- Peripheral event transferred to the DMA as a DMA trigger
- Peripheral event transferred to another peripheral to directly trigger an action in hardware

The event manager connects event publishers to event subscribers through an event fabric. There are three types of event fabric: CPU interrupt (fixed event route), DMA route, and generic route.

For details how to use the MSPM0 event handler, refer to the event section in the MSPM0L Technical Reference Manual, or the MSPM0C Technical Reference Manual.

3.6.3 RA0 Event Controller (ELC)

RA0 is equipped with an event link controller (ELC) that connects (links) events that are output by each peripheral function. By linking events, RA0 can adjust the operation of peripheral functions directly, without going through the CPU.

Figure 3-3 provides an ELC block diagram. The ELSEGRn register links each event signal to the operation of the event receiving peripheral function (linked peripheral) after it is received. Different ELSEGRn registers represent different event generators, and the value set in the ELSRn register determines the operation of the linked peripheral.

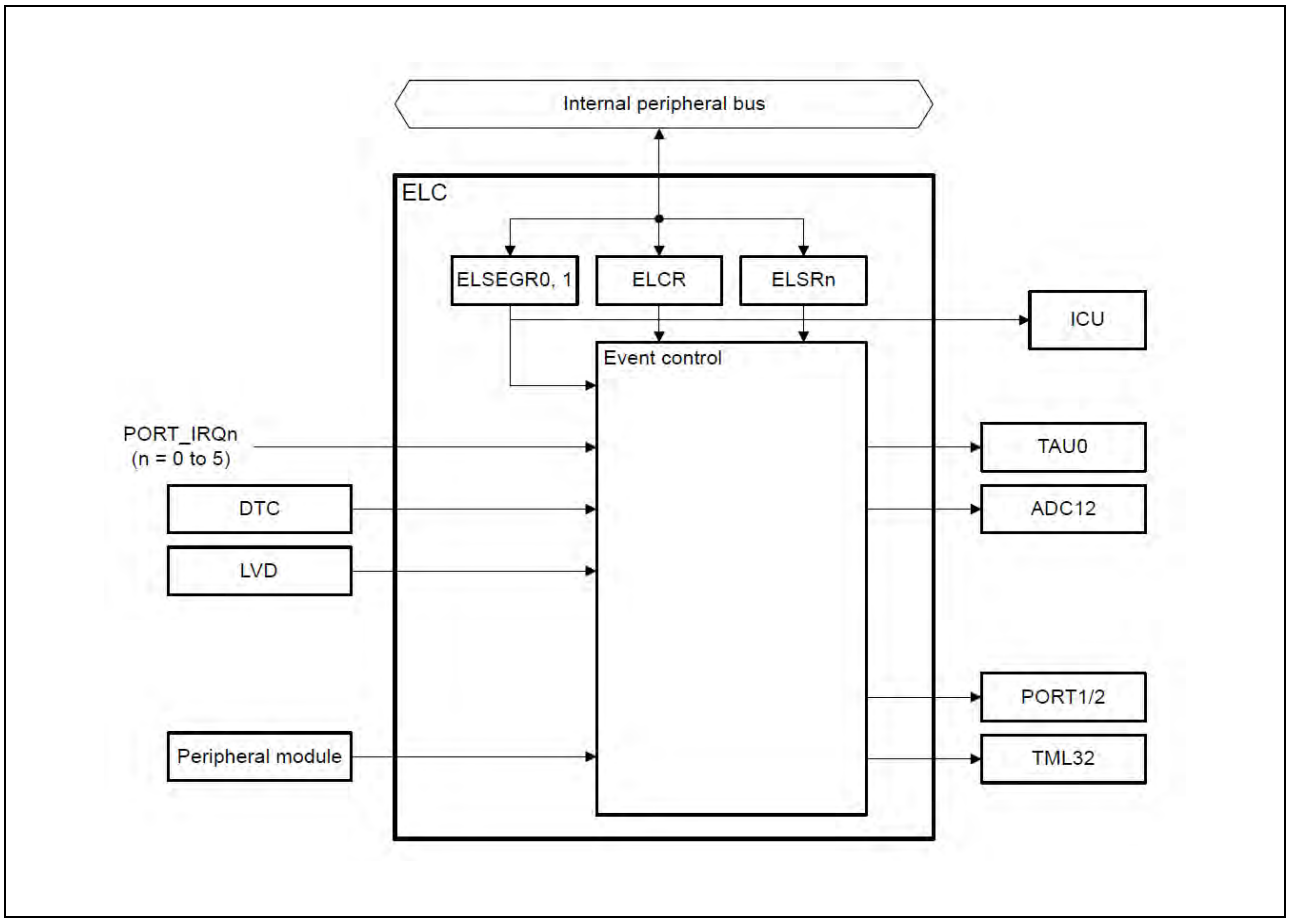


Figure 3-3 RA0 Event Link Controller

3.6.4 Comparison of Event Controls

Event control configurations and functions differ between RA0 and MSPM0. Table 3-15 shows a comparison of 3.6.2 MSPM0 Event Handler and section 3.6.3 RA0 Event Controller (ELC) .

Table 3-15 Comparison of Event Control

Event Type		MSPM0	RA0
Event source		Peripheral function	Peripheral function
Notification destination		CPU, DMA trigger, peripheral functions	CPU, DTC trigger, peripheral functions
Notification method	Peripheral function to CPU	Event manager	Interrupt control circuit
	Peripheral function to DMA		DTC controller
	Peripheral function to peripheral function		ELC
Notification route		Available routes: 1. Point-to-Point 2. Point-to-Two (splitter) Note	Available routes: 1. Point-to-Point

Note The general route channel can be configured with Point-to-Point (1:1) or Point-to-Two (1:2; also called splitter route) depending on the selected channel.

3.7 Debugging and Programming Comparison

RA0 and MSPM0 use different interfaces and functions when connecting to the device for debugging or programming. Table 3-16 shows a comparison of debugging and programming in MSPM0 and RA0.

Table 3-16 Debugging and Programming Comparison

Programming Function	MSPM0	RA0
Type	Bootstrap loader	Serial programming
Security	Secure boot options: CRC protections	Memory security and access restriction options Note Block erase, programming command, read command, security setting command, protection setting
Customizable	Configurable invoke pin and plug-in feature	N/A
Entry to the programming mode	1 pin high at BOOTRST, SW entry	After reset is released, set DFLCTL.DFLEN to 1 and FENTRYR to code flash P/E mode.
Command	Connection, unlock, erase, write, memory read back, factory reset, get info, etc.	Erase, programming, get info, security, etc. Note
Password protection	Available	Available (ID code check)
Interface	UART, I^2C, SPI (custom plug-in required)	SWD

Note Block erase, programming, and start-up area rewrite are disabled in the default settings. These commands can be enabled by issuing a Program command. Refer to the corresponding section in RA0 User's Manual: Hardware for more details.

3.7.1 Debugging Comparison

3.7.1.1 Debugging RA0

Debugging support via Serial Wire Debug (SWD) is standard on RA0. Any of the following can be used for debugging: J-Link, E2 emulator, or E2 emulator Lite. These debuggers support standard debug functions (step execution, breakpoint, memory and register inspection, etc.). They also support on-chip debugging functions. For programming, use J-Link, E2 emulator, E2 emulator Lite, or dedicated flash memory programmer PG-FP6.

3.7.1.2 Debugging MSPM0

Debugging support via JTAG and Serial Wire Debug (SWD) is standard on MSPM0. This interface is typically used during application development and production programming. The TI XDS debug probes (XDS100, XDS110, etc.) support real-time debugging and tracing.

3.7.2 Programming Mode Comparison

3.7.2.1 Serial Programming in RA0

The RA0 serial programming mode enables firmware updates using an external device (microcontroller or ASIC) connected to RA0 via SWD.

3.7.2.2 Bootstrap Loader (BSL) Programming in MSPM0

The bootstrap loader (BSL) programming interface serves as an alternative to the SWD interface.

This interface provides programming capabilities only and is typically used through a standard built-in communication interface. This allows firmware updates over existing connections to the system or other embedded devices using an external port.

The main purpose of the BSL programming interface is to update programming, however it can also be used for initial production programming.

4. Digital Peripheral Comparison

4.1 General Purpose I/O（PORT, PmnPFS_A）

RA0 is designed specifically for low-power consumption, including its I/O ports.

MSPM0 offers a highly flexible IOMUX (Input/Output Multiplexer) function. This function enables the user to assign multiple peripheral functions to one pin. For example, you can select different functions for the I/O port, such as UART, SPI, I^2C, timer output, etc.

RA0 is equipped with a pin function selection function (PmnPFS_A) that switches the ports assigned with share functions. Functions of pins with alternate functions pre-mapped by RA0 can be switched by setting the PSEL bit of the PmnPFS_A register.

Table 4-1 I/O Port Function Comparison

Item	MSPM0	RA0
Output mode	Push-pull with pull-up or pull-down Open drain with pulldown Hi-Z	CMOS Open drain with N-ch
Input mode	Floating Pull-up or pull-down	CMOS On-chip pull-up TTL input buffer
High-drive I/O port	Approx. 20 mA @ Vdd = 3.3V (High Drive IO (HDIO))	20mA (V_{CC} = 1.6 to 5.5 V)
Setting per port	Available	Available
Redirect function	Available	Uses PmnPFS_A register
Wake-up	Uses IOMUX	Can be used with external interrupt (IRQ)
I/O port controlled by DMA	Available	N/A
Input filtering	Available	Available (signals within 1us are filtered when using external interrupt (INTP))
Switching between CMOS and TTL buffer	Available	Available

For information on I/O port code examples, please download the sample project "FPB-RA0xx Example Project Bundle" from the Renesas Electronics website.

4.2 Universal Asynchronous Receiver-Transmitter (UART, SAU)

Both RA0 and MSPM0 are equipped with a peripheral function that enables asynchronous serial communication.

MSPM0 also features functions that are not included in RA0, such as IrDA hardware support, smart card mode, and hardware flow control.

RA0 is equipped with UARTA, a specialized UART communication function. The Serial Array Unit (SAU) includes the UART function and supports the general communication function as well as LIN-bus communication.

Using SAU in conjunction with SNOOZE mode enables UART communication while reducing system power consumption.

Table 4-2 UART Function Comparison

Function	MSPM0	RA0
Data direction	MSB first or LSB first	MSB first or LSB first
Continuous communication with DMA	Available	Available (can be used with DTC)
Data phase	N/A	Available
Reception in SNOOZE mode	Available, active in all low-power modes	Available
Single-wire half-duplex communication	Available Note 1	N/A
Data length	5, 6, 7, 8	5, 7, 8, 9 Note 2
LIN HW support	Available	Available
DALI HW support	Available	N/A (will be available in RL78/G24, I1A)
IrDA HW support	Available	N/A
Manchester Code HW support	Available	N/A
Smart card mode (ISO7816)	Available	N/A
Wakeup from low-power mode	Available	Available
Auto baud rate detection	N/A	Available (LIN)
Enabling external driver	Available	N/A
Hardware flow control	Available	N/A
Multiprocessor	Available	N/A
Tx/Rx FIFO Depth	4	1

Note 1. The peripherals need to be reconfigured between transmission and reception.
Note 2. 5-bit data length can be used with UARTAx and 9-bit data length can be used with SAU.

For information on UART code examples, please download the sample project "FPB-RA0xx Example Project Bundle" from the Renesas Electronics website.

4.3 Serial Peripheral Interface (SPI, SAU)

MSPM0 and RA0 both support serial peripheral interface (SPI).

RA0 is equipped with a simplified SPI in the serial array unit (SAU).

Table 4-3 compares SPI support between MSPM0 and RA0.

Table 4-3 SPI Function Comparison

Function	MSPM0	RA0
Control pins	SCLK, PICO, POCI, CSx	SCK, SI, SO
Master/Slave selection	Available	Available
Data bit width (controller mode)	4 to 16 bits	7, 8 bits
Data bit width (peripheral mode)	7 to 16 bits	
Maximum speed	MSPM0L: 16MHz	16 MHz (SPI00 only) 8 MHz (other than SPI00) 4 MHz (during slave communication)
Simplex communication	Available	Available
Hardware chip select management	Available (peripherals x 4)	N/A
I/O clock phase control	Available	Available
Data direction	MSB first or LSB first	MSB first or LSB first
SPI format support	Motorola, TI, MICROWIRE	Motorola
Hardware chip select management	N/A; MSPM0 provides SPI parity check.	N/A
Tx/Rx FIFO Depth	4	1

For information on SPI code examples, please download the sample project "FPB-RA0xx Example Project Bundle" from the Renesas Electronics website.

4.4 Inter-Integrated Circuit (IICA, SAU)

Both MSPM0 and RA0 support the I²C peripheral function.

RA0 is equipped with Simplified I²C in the Serial Array Unit (SAU) for the operation of basic functions and serial interface IICA for more advanced functions.

MSPM0 I²C comes with a single peripheral function that integrates both basic and advanced functions.

The overall I²C support in MSPM0 and RA0 is comparable, while there are some remarkable differences, as shown in Table 4-4.

Table 4-4 I²C Function Comparison

Function	MSPM0	RA0
Master/Slave selection	Available	Available (IICA)
Multi-controller function	Available	Available (IICA)
Maximum transfer speed	1 MHz	1 MHz
Addressing mode	7 bits	・7 bits ・10 bits (IICA)
Address number (target mode)	2 addresses	1 address
Event control	Available	Available (IICA)
Clock stretch	Available	Available (IICA)
Wakeup function (low-power mode)	Available	Available (IICA)
Software reset	Available	Available (IICA)
FIFO/Buffer	Transmission: 8 bytes	1 byte IICA: IICA shift register (IICAn) SAU: serial data register (SDRmn)
	Reception: 8 bytes	
DMA/DTC transfer	Available	Available (SAU only)
Programmable analog and digital noise filters	Available	Available (IICA only)

For information on I²C code examples, please download the sample project "FPB-RA0xx Example Project Bundle" from the Renesas Electronics website.

4.5 Timer (TAU, TML32)

MSPM0 and RA0 are both equipped with multi-function timers.

MSPM0 provides timers with a range of functions to support various use cases, from low-power monitoring to advanced motor control.

RA0 has a timer array unit (TAU) with 8 channels, enabling multiple timer functions to operate simultaneously.

Table 4-5 Timer Names

MSPM0		RA0	
Timer Name	Abbr.	Timer name	Abbr.
Advanced control	TIMA0	N/A	N/A
General purpose	TIMG0‑11	Timer array unit	TAU
High resolution	TIMG12	32-bit interval timer	TML32

Table 4-6 Timer Function Comparison

Function	MSPM0L	RA0
Resolution	16 bits	TAU: 8, 16 bits TML32: 8, 16, 32 bits
PWM	Available	Available (TAU)
Capture	Available	Available
Compare	Available	Available
One-shot	Available	Available (TAU)
Up/down count function	Available	N/A
Conjunctive operation with standby function	Available	Available
QEI support	Available	N/A
Programmable prescaler clock frequency	Available	Available
Temporary hold during setup	Available	Available
Event/interrupt	Available	Available
Auto reload function	Available	Available
Fault handling	Available	N/A

Table 4-7 Timer Use Case Comparison

Function	MSPM0L	RA0
PWM	TIMA, TIMG0-11	TAU
Capture	TIMA, TIMG0-11	TAU
Compare	TIMA, TIMG0-11	TAU
One-shot	TIMA, TIMG0-11	TAU
Synchronization	TIMA, TIMG0-11	TAU
16bit-Interval timer	TIMA, TIMG0-11	TAU, TML32
32bit-Interval timer	TIMG12	TML32
Prescaler	8-bit prescaler	4-bit prescaler (TAU) 3-bit prescaler (TML32)

For information on Timer code examples, please download the sample project "FPB-RA0xx Example Project Bundle" from the Renesas Electronics website.

4.6 Independent Watchdog Timer (IWDT)

Both MSPM0 and RA0 provide a watch dog timer. If the application does not restart counting within the specified time, the Independent watchdog timer (IWDT) determines this as program runaway and triggers a system reset

Table 4-8 WDT Name

MSPM0L		RA0	
Timer Name	Abbr.	Timer Name	Abbr.
Window watchdog timer	WWDT	Independent watchdog timer	IWDT

Table 4-9 WDT Function Comparison

Function	MSPM0	RA0
Window mode	Available	Available
Interval timer mode	Available	Available
Clock source	LFCLK	LOCO/2
Interrupt	Available	Available
Counter resolution	25 bits	14 bits
Clock divider	Available	N/A

For information on IWDT code examples, please download the sample project "FPB-RA0xx Example Project Bundle" from the Renesas Electronics website.

4.7 Realtime Clock (RTC)

Both MSPM0 and RA0 are equipped with a realtime clock (RTC). The RTC can set the time in binary-coded decimal format and provides a clock function for applications. The RTC also includes a fixed-cycle interrupt and an alarm function that notifies the system of events at times specified by the application.

Table 4-10 RTC Function Comparison

Function	MSPM0	RA0
Operation in all modes	Available	Available
Binary coded format	Available	N/A
BCD code format	N/A	Available
Years count up	199 years	99 years
Leap year correction	Available	Available
Number of customizable alarms	2	1
Selection of supply clock (internal/external)	Available	Available *Time count function is not available when an internal clock is selected.
Time error correction	Available	Available
Interrupts	Available	Available
1Hz clock output function	Available	Available

For information on RTC code examples, please download the sample project "FPB-RA0xx Example Project Bundle" from the Renesas Electronics website.

5. Analog Peripheral Comparison

5.1 Analog-Digital Converter (ADC12)

Both MSPM0 and RA0 are equipped with a 12-bit ADC peripheral function that converts analog signals to digital signals. Table 5-1 and Table 5-2 show a comparison of the range of ADC functions.

Table 5-1 ADC Function Comparison

Function	MSPM0L	RA0
Resolution (bits)	12, 10, 8 bits	12, 10, 8 bits
Conversion rate (Msps)	1.4	0.485
Oversampling (bits)	N/A	N/A
Hardware oversampling	N/A	N/A
FIFO	Available	N/A
ADC reference (V)	Internal: 1.48, 2.5, V_{DD}	Internal: 1.48, V_{CC}
	External: 1.4 ≤ VREF ≤ V_{DD}	External: VREFH0
Operation mode	RUN, SLEEP, STOP, STANDBY Note 1	Normal operation, SLEEP, SNOOZE Note 2
Auto power down	Available	Available
External input channel Note 3	16 (max.)	10 (max.)
Internal input channels	Temperature sensor, supply monitoring, analog signal chain	Temperature sensor, internal reference voltage
DMA support	Available (DMA)	Available (DTC)
ADC window comparator unit	Available	Available (ADUL, ADLL)
Simultaneous sampling	N/A	N/A
Number of ADC Note 4	1	1

Note 1. ADC can be triggered in STANDBY mode, which changes the operating mode.
Note 2. ADC12 can be triggered in SNOOZE mode, which changes the operating mode.
Note 3. The number of external input channels differs depending on the device.
Note 4. The number of ADCs differs depending on the device.

Table 5-2 Conversion Modes

MSPM0	RA0	Comment
Single channel single conversion	Select mode, one-shot conversion mode	ADC12 samples and converts single channel once.
Channel conversion sequence	Scan mode, one-shot conversion mode	ADC12 samples and converts a sequence of channels once.
Repeat single channel conversion	Select mode, sequential conversion mode	ADC12 samples and converts single channel continuously.
Repeat channel conversion sequence	Scan mode, sequential conversion mode	ADC12 samples and converts a sequence of channels continuously.

For information on ADC code examples, please download the sample project "FPB-RA0xx Example Project Bundle" from the Renesas Electronics website.

5.2 Reference Voltage (V_{REFP})

Both RA0 and MSPM0 have internal references that are used to supply a reference voltage to internal peripheral functions.

Table 5-3 VREF Characteristic Comparison

Function	MSPM0L	RA0
Internal reference voltage (V)	1.45, 2.5	1.48
External reference voltage (V)	External: 1.4 ≤ VREF ≤ V_{DD}	1.6 ≤ VREFH0 ≤ 5.5V
Output internal reference	N/A	N/A
Internal connection to ADC	Available	Available

For MSPM0 VREF, set to enable power bit PWREN Bit0 (ENABLE).

6. Reference

[1] MSPM0C110x、MSPM003 Data Sheet (JAJSRS4C – OCTOBER 2023 – REVISED FEBUARY 2025)
[2] MSPM0C1105、MSPM0C1106 Data Sheet (JAJSWZ8A – JULY 2025 – REVISED SEPTEMBER 2025)
[3] MSPM0L130x Data Sheet (JAJSPZ3D – OCTOBER 2022 – REVISED JANUARY 2024)
[4] MSPM0L222x、MSPM0L122x Data Sheet (JAJSUK6A – MAY 2024 – REVISED OCTOBER 2024)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Nov.30.25	-	First edition issued
1.10	Feb.24.26	p.1, p.4 p.38, p.40	Add RA0E3 content

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/.