

# RA0 Series

## Migration Guide from Microchip PIC®16(L)F1 to RA0

---

### Overview

This application note describes development and tool ecosystem, core architecture, peripheral considerations, and software development kit for when migrating from Microchip PIC16(L)F1 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 PIC16(L)F1 Family development environment to quickly ramp up with the RA0 series Group of MCUs.

All trademarks and registered trademarks, including company names and product names mentioned in this document belong to their respective owners. Note that the TM and ® marks have been omitted in the main text.

### Device to Be Migrated

Microchip Technology.inc PIC16(L)F1 family

### Target Device

RA0 Group

### 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 e<sup>2</sup> studio 2023 -10 or Higher Quick Start Guide User's Manual RA Family Renesas MCU (R20UT5210)
- [6] Renesas Flexible Software Package (FSP) v6.1.0 User's Manual(R11UM0155)

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### Contents

1. RA0 Portfolio Overview.....	4
1.1 Introduction.....	4
1.2 Portfolio Comparison of PIC16(L)F1 and Renesas.....	4
2. Ecosystem and Migration.....	6
2.1 Comparison of Ecosystem.....	6
2.1.1 Smart Browser.....	8
2.1.2 IDE Supported by RA0.....	9
2.1.3 FSP Configurator.....	10
2.1.4 Debug Tools.....	13
2.1.5 Evaluation Board.....	14
2.1.5.1 Fast Prototyping Board.....	14
2.2 Migration Process.....	15
2.2.1 Step1: Selecting the Correct RA0.....	15
2.2.2 Step 2: IDE Setup and Quick Introduction of e <sup>2</sup> studio.....	18
2.2.2.1 IDE Setup.....	18
2.2.2.2 How to Use e <sup>2</sup> studio.....	20
2.2.3 Step 3: Importing Software.....	25
2.2.4 Step 4: Evaluating the Software.....	29
2.2.5 Step 5: Mass Production.....	29
2.3 Example of importing a sample program.....	31
3. Core Architecture Comparison.....	38
3.1 CPU.....	38
3.2 Built-in Memory Comparison.....	39
3.2.1 Flash Memory Function.....	39
3.2.2 Flash Memory Configuration.....	40
3.2.2.1 Flash Memory Area.....	40
3.2.2.2 Flash I/O register in RA0.....	40
3.2.2.3 Self-Programming in RA0.....	40
3.2.3 SRAM.....	41
3.3 Power-On and Reset Overview and Comparison.....	42
3.4 Clock Overview and Comparison.....	44
3.4.1 Oscillator.....	44
3.4.1.1 RA0 Oscillator.....	44
3.4.2 Clock Signal Comparison.....	45
3.5 Overview and Comparison of Operation Modes.....	46
3.5.1 Comparison of Operation Modes.....	46
3.5.2 RA0 Functions in SLEEP Mode.....	47

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

3.5.3	RA0 Functions in SNOOZE Mode.....	47
3.5.4	RA0 Functions in Software STANDBY Mode.....	47
3.6	Comparison of Interrupts and Events.....	48
3.6.1	Interrupts and Exceptions.....	48
3.6.1.1	Interrupt Control in RA0.....	49
3.6.1.2	Interrupt Control in PIC16(L)F183xx .....	50
3.7	Debugging and Programming Comparison .....	51
3.7.1	Debugging Comparison.....	51
3.7.1.1	Debugging RA0 .....	51
3.7.1.2	Debugging PIC16(L)F183xx.....	51
3.7.2	Programming Mode Comparison .....	52
3.7.2.1	Serial Programming in RA0.....	52
3.7.2.2	Serial Programming PIC16(L)F183xx .....	52
4.	Digital Peripheral Comparison .....	53
4.1	General Purpose I/O (PORT, PmnPFS_A).....	53
4.2	Universal Asynchronous Receiver-Transmitter (UART, SAU) .....	54
4.3	Serial Peripheral Interface (SPI, SAU) .....	55
4.4	Inter-Integrated Circuit (IICA, SAU).....	56
4.5	Timer (TAU,TML32) .....	57
4.6	Independent Watchdog Timer (IWDT) .....	58
5.	Analog Peripheral Comparison .....	59
5.1	Analog-Digital Converter (ADC12) .....	59
5.2	Reference Voltage (V <sub>BGR</sub> ).....	59
6.	Reference.....	60

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

## 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 Microchip Technology, Inc.'s PIC16(L)F1 family (hereafter referred to as "PIC16(L)F1"). This application note assists in migration from PIC16(L)F1 to RA0 by comparing device features and ecosystems.

### 1.2 Portfolio Comparison of PIC16(L)F1 and Renesas

Table 1-1 Comparison of Microchip PIC16(L)F1 and Renesas (1/2)

Item		Microchip <sup>[1]</sup> PIC16F180xx	Microchip <sup>[2]</sup> PIC16(L)F183xx	Microchip <sup>[3]</sup> PIC16(L)F184xx	Renesas RA0
Core		Enhanced Mid-Range CPU			Arm Cortex-M23
Clock frequency		Max 32MHz	Max 32MHz	Max 32MHz	Max 32MHz
Minimum instruction execution time		0.125µs	0.125µs	0.125µs	0.03125µs
Power supply voltage		1.8~5.5V	1.8~3.6V 2.3~5.5V	1.8~3.6V 2.3~5.5V	1.6~5.5V
Operating temperature range		-40~+85°C -40~+125°C	-40~+85°C -40~+125°C	-40~+85°C -40~+125°C	-40~+125°C
Code flash		3.5~28KB	3.5~28KB	7~28KB	16~128KB
Data Flash		-	-	-	1KB
EEPROM		128~256B	256B	256B	-
RAM		0.25~2KB	0.25~2KB	0.5~2KB	2~16KB
Interrupt	Internal	Max 30	Max 33	Max 45	Max 33
	External	1+IOC Max25	1+IOC Max18	1+IOC Max25	Max 6
I/O port		Max 35	Max 18	Max 26	Max 60
Analog	ADC	10bit ADC (Max 35ch)	10bit ADC (Max 17ch)	12bit ADC (Max 24ch)	12bit ADC (Max 15ch)
	DAC	8bit DAC (1ch)	5bit DAC (1ch)	5bit DAC (1ch)	-
	COMP	1ch	Max 2ch	Max 2ch	-
Serial interface (Max)	UART	Max 2ch	1ch	Max 2ch	Max 5ch
	I2C	Max 2ch	Max 2ch	Max 2ch	Max 4ch
	SPI	Max 2ch	Max 2ch	Max 2ch	Max 6ch
Timer		8bit : 4ch 16bit : 3ch	8bit : Max 4ch 16bit : Max 4ch	8bit : 4ch 16bit : 4ch	16bit : 8ch 32bit : 1ch (or 16bit : 2ch or 8bit : 4ch)

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

Table 1-2 Comparison of Microchip PIC16(L)F1 and Renesas (2/2)

Item		Microchip PIC16F180xx	Microchip PIC16(L)F183xx	Microchip PIC16(L)F184xx	Renesas RA0
Other key peripherals / features		ZCD Temp. Indicator HCVD PPS	Temp. Indicator PPS PMD	ZCD Temp. Indicator HCVD PPS PMD	Temperature Sensor, Event Link Controller Capacitive touch (RA0L1)
Number of pins		8~44	8~20	14~28	16~64
Low power <sup>Note</sup>	Normal operation	4.5mA@32MHz, 3V, typical	2.2mA@32MHz, 3V, typical	3.9mA@32MHz, 3V, typical	2.7mA@32MHz, 3.3V, typical
	Standby	-	DOZE mode 863uA@16MHz, 3V, typical	DOZE mode 1.5mA@16MHz, 3V, typical	Sleep mode 820uA@32MHz, 3.3V, typical
			IDLE mode 804uA@16MHz, 3V, typical	IDLE mode 1.5mA@16MHz, 3V, typical	
		SLEEP mode 900nA@3V, typical	SLEEP mode 950nA@3V, typical	SLEEP mode 880nA@3V, typical	Software Standby mode 200nA@3.3V, typical

Note The RA0 software standby mode corresponds to the PIC16(L)F1's SLEEP mode, and the sleep mode corresponds to either DOZE mode or IDLE mode.

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

## 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	PIC16(L)F1	RA0
Code source	Driver Library Middleware RTOS Code example	Flexible Software Package – FSP (Middleware, Driver, RTOS) Smart Browser (sample code, document references)
IDE	MPLAB X IDE MPLAB Xpress Cloud-Based IDE	e <sup>2</sup> studio IAR Embedded Workbench Arm Keil MDK IDE
Software configuration	MPLAB Code Configurator (MCC)	FSP Configuration RA Smart Configurator (RASC)
Software programming tool	MPLAB X IDE MPLAB Xpress Cloud-Based IDE	Renesas Flash Programmer
Hardware programming tool	MPLAB PICKit 5 MPLAB ICD 5 MPLAB ICE 4	PG-FP6
Debugger	MPLAB PICKit 5 MPLAB ICD 5 MPLAB ICE 4	E2 Emulator E2 Emulator Lite J-Link ULINKplus
Evaluation Board	MPLAB Xpress Evaluation Board PIC16F18446 Curiosity Nano PIC16F18076 Curiosity Nano Evaluation Kit	RA0xx Fast Prototyping Board

# RA0 Series

## Migration Guide from Microchip PIC®16(L)F1 to RA0

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

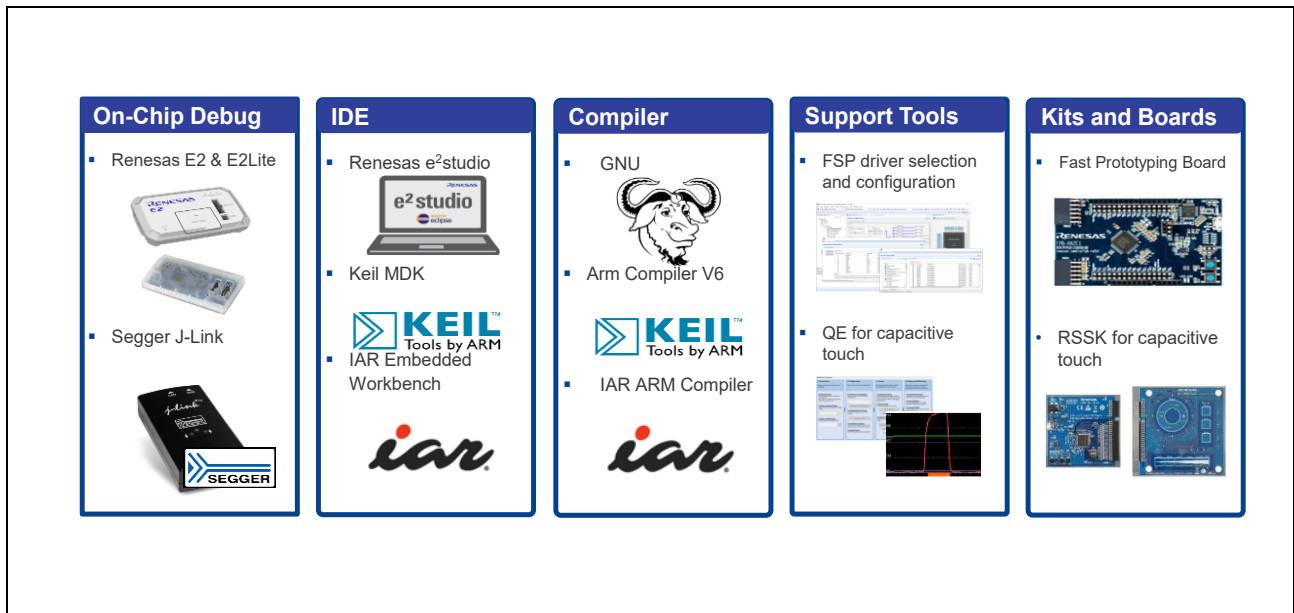


Figure 2-1 Overview of RA0 Ecosystem

# RA0 Series

## Migration Guide from Microchip PIC®16(L)F1 to RA0

### 2.1.1 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.

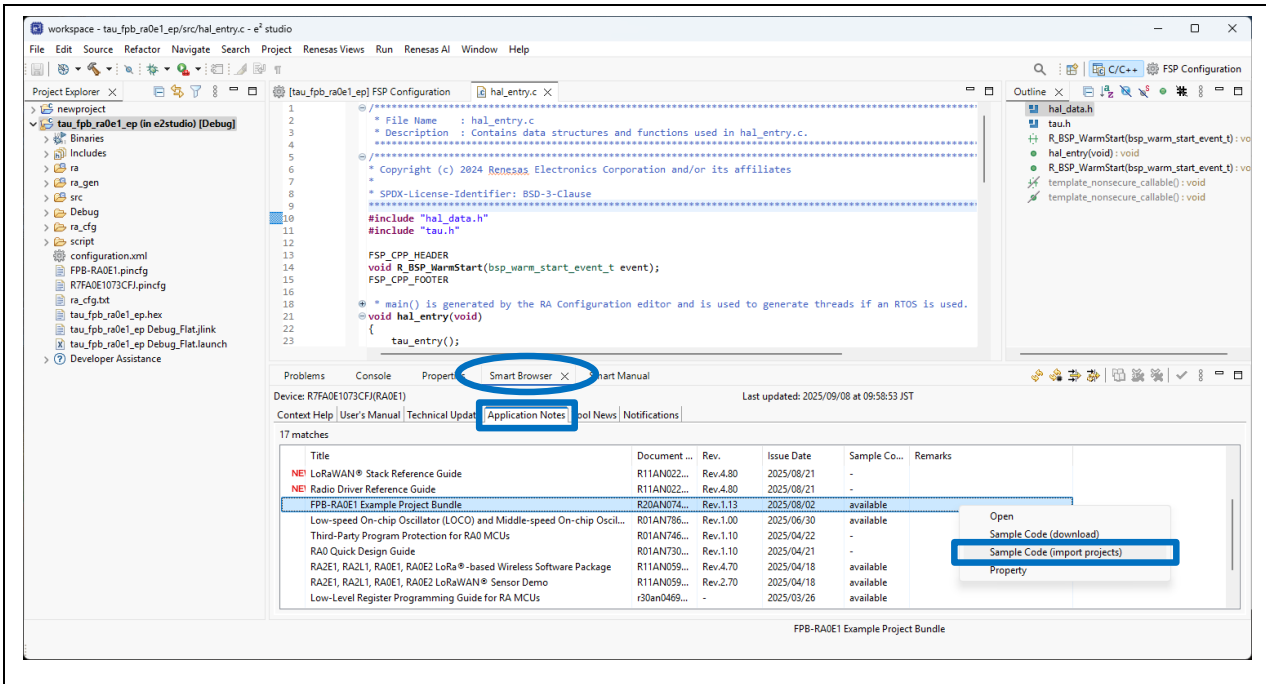


Figure 2-2 Smart Browser

The Smart Browser is included in the e<sup>2</sup> studio and can be used by installing the e<sup>2</sup> 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 2-2 Comparison of Software Ecosystem

Item	PIC16(L)F1	RA0
Driver library	Available	Available
Middleware	Available	Available
Self-programming	Available	Available
Sample program	Available	Available
Free RTOS	Available	Available

The RA0 sample programs are designed to allow easy initialization and configuration of common hardware functions. For example, it includes all necessary code and initialization procedures for major peripheral functions such as PORT (I/O ports), timers, UART, I<sup>2</sup>C, and SPI. Most of the RA0 sample programs support the FSP Configurator for accelerating software developments.

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### 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 PIC16(L)F1 is MPLAB X IDE 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.

RA0's IDE, e<sup>2</sup> studio, also consists of a completed set of tools with many features for developing and debugging embedded applications. In addition, the e<sup>2</sup> 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 2-3 Comparison of MPLAB X IDE and e<sup>2</sup> studio**

IDE	MPLAB X IDE	e <sup>2</sup> studio
License	Free	Free
Compiler	MPLAB XC Compilers/GCC	GCC / LLVM / Arm Compiler / IAR Arm Compiler
Peripheral API function support	Not supported	Supported
Display languages	English (default) Japanese Chinese German French Spanish Italian Korean	English Japanese Chinese
Executable file output format	Hex file Binary file Executable and Linkable Format Common Object File Format	Hex file Binary file Motorola S-record file
Code generation GUI	MPLAB Code Configurator	FSP Configurator RA Smart Configurator (RASC)

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

In addition to e<sup>2</sup> studio, RA0 also supports the industry-standard IDEs listed in Table 2-4.

**Table 2-4 Overview of RA0 supported IDEs**

IDE	e <sup>2</sup> studio	Arm Keil MDK IDE	IAR Embedded Workbench
License	Free	Paid	Paid
Compiler	GCC / LLVM / Arm Compiler / IAR Arm Compiler	Arm Compiler	IAR C/C++ Compiler for Renesas Arm
Disk Size	2G or mode	2G or mode	6.33G(Arm 8.50.4)
E2 Emulator	Supported	Not supported	Supported
E2 Emulator Lite	Supported	Not supported	Supported
J-Link	Supported	Supported	Supported
ULINKplus	Not supported	Supported	Not supported
Renesas QE low power consumption debugging	Supported	Not supported	Not supported
MISRA-C	Not supported	Not supported	Supported

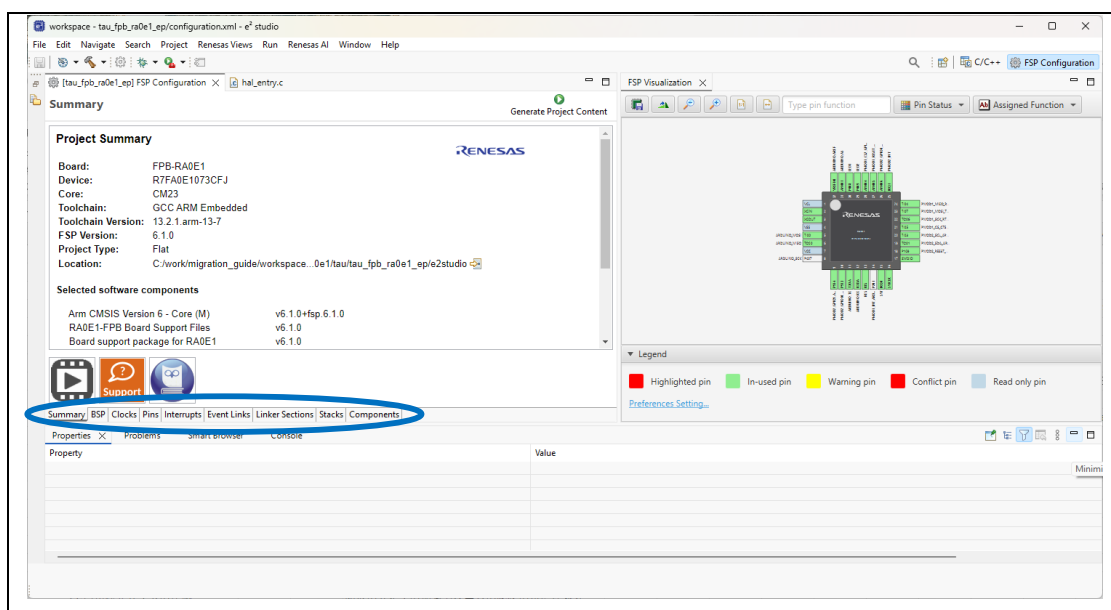
For e<sup>2</sup> studio usage and function descriptions, refer to 2.2.2.2 How to Use e<sup>2</sup> studio.

### 2.1.3 FSP Configurator

The FSP Configurator is integrated into e<sup>2</sup> studio. Like MPLAB Code Configurator, 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 RA Smart Configurator (RASC) 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 2-3 FSP Configurator**

# RA0 Series

## Migration Guide from Microchip PIC®16(L)F1 to RA0

The FSP Configurator, like MPLAB Code Configurator, provides the hardware circuit diagram display, detailed descriptions of each peripheral function, and detailed functional descriptions of all GUI interface configurations.

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 2-4 compares categorized lists of peripheral functions and Figure 2-5 compares interrupt settings.

Interrupt and pin settings in MPLAB Code Configurator are configured individually for each component used, 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.

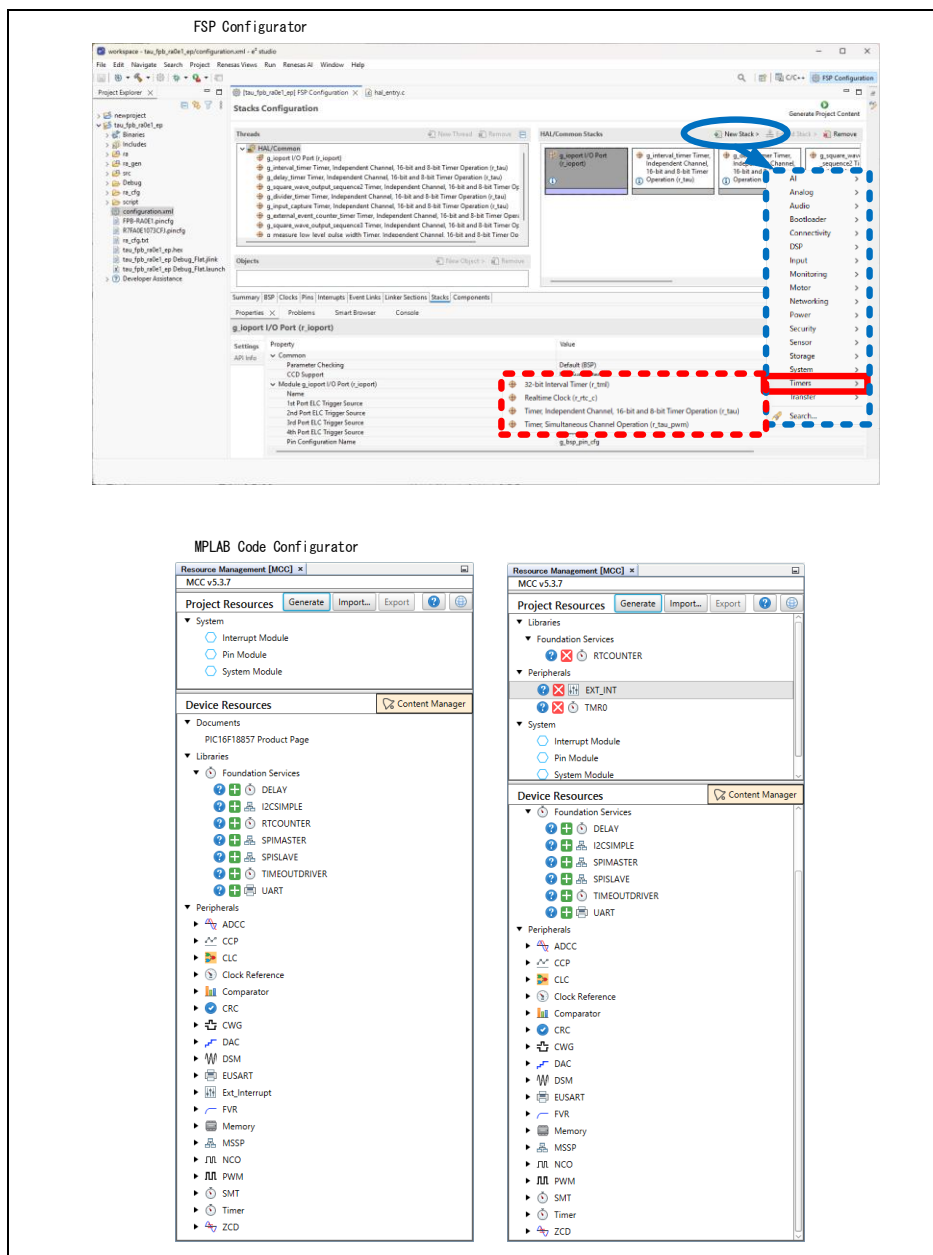


Figure 2-4 Comparison of Peripheral Function Lists

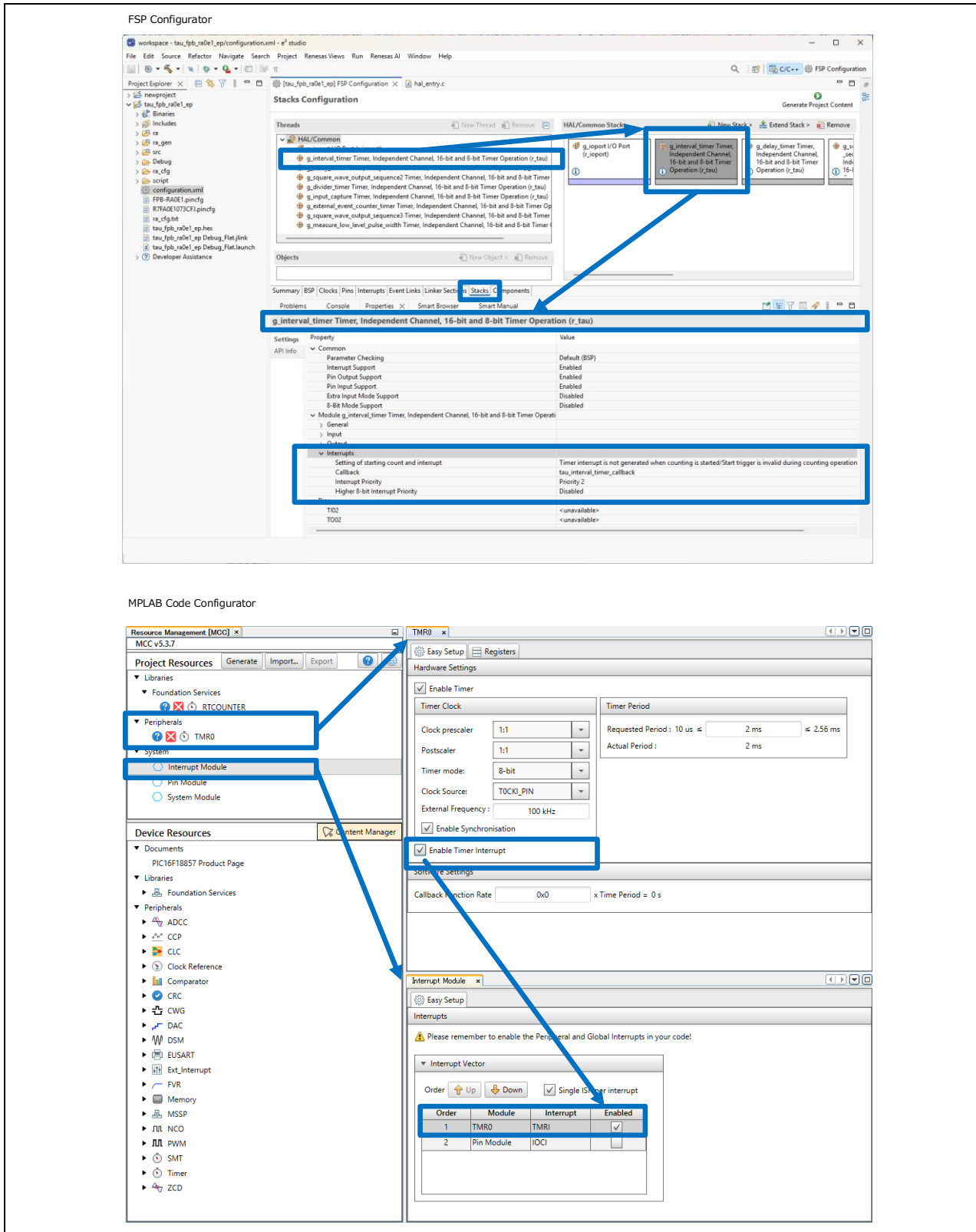


Figure 2-5 Interrupt Setting Comparison

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

#### 2.1.4 Debug Tools

For the PIC16(L)F1, the MPLAB PICkit 5 tool connects to the PIC16(L)F1 via the PGD (Program Data) and PGC (Program Clock) pins using the In-Circuit Serial Programming (ICSP) and In-Circuit Debugging (ICD) interfaces. It supports real-time monitoring of device status during program execution and debugging in low-power modes.

For the RA0, connect the Serial Wire Debug (SWD) interface as the interface between the debugger (such as the E2 Emulator or E2 Emulator Lite) and the RA0. The standard debugger tool for RA0 is the E2 Emulator, which supports current consumption measurement, point monitoring, and external trigger input/output configuration. For details on debugger connection, refer to Figure 2-6.

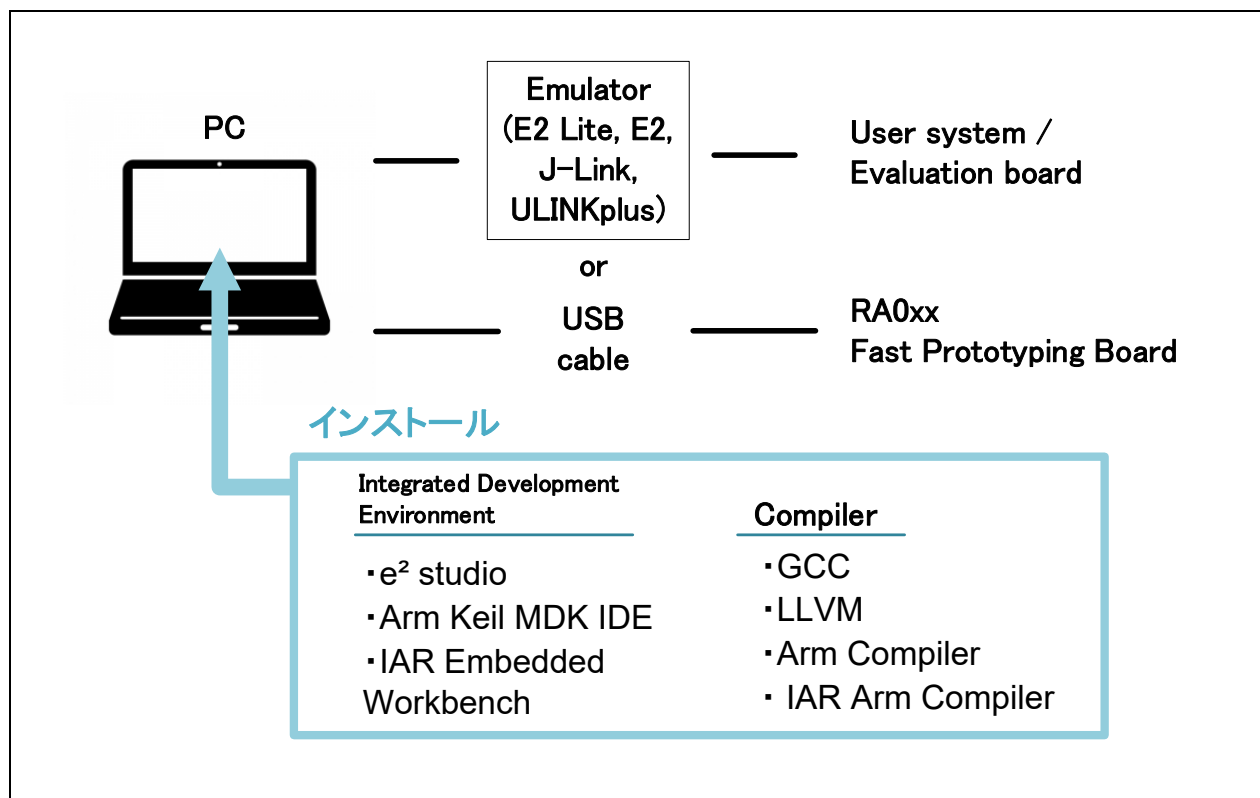


Figure 2-6 RA0 Debug

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.

Table 2-5 provides a function overview of the USB connection, the E2 Emulator, E2 Emulator Lite, and J-Link all of which support RA0 on-chip debugging.

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

Table 2-5 Comparison of RA0 Debuggers

Item	On board debugger	E2 emulator	E2 emulator Lite	J-Link	ULINKplus
BSL <sup>Note</sup> tool	Available	Available	Available	Available	Available
Backchannel UART	Available	Available	Available	Available	Available
Supply voltage	3.3 V or 5.0 V	1.8 V ~ 5.0 V	3.3 V	5.0 V	1.2 V ~ 5.5 V
IDE: e <sup>2</sup> studio	Supported	Supported	Supported	Supported	Not Supported
IDE: 3rd party	IAR	IAR	IAR	IAR, Keil	Keil

Note BSL indicates bootstrap loader.

#### 2.1.5 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.

##### 2.1.5.1 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 2-7 shows an overview of the Fast Prototyping Board, using the RA0E1 Fast Prototyping Board as an example.

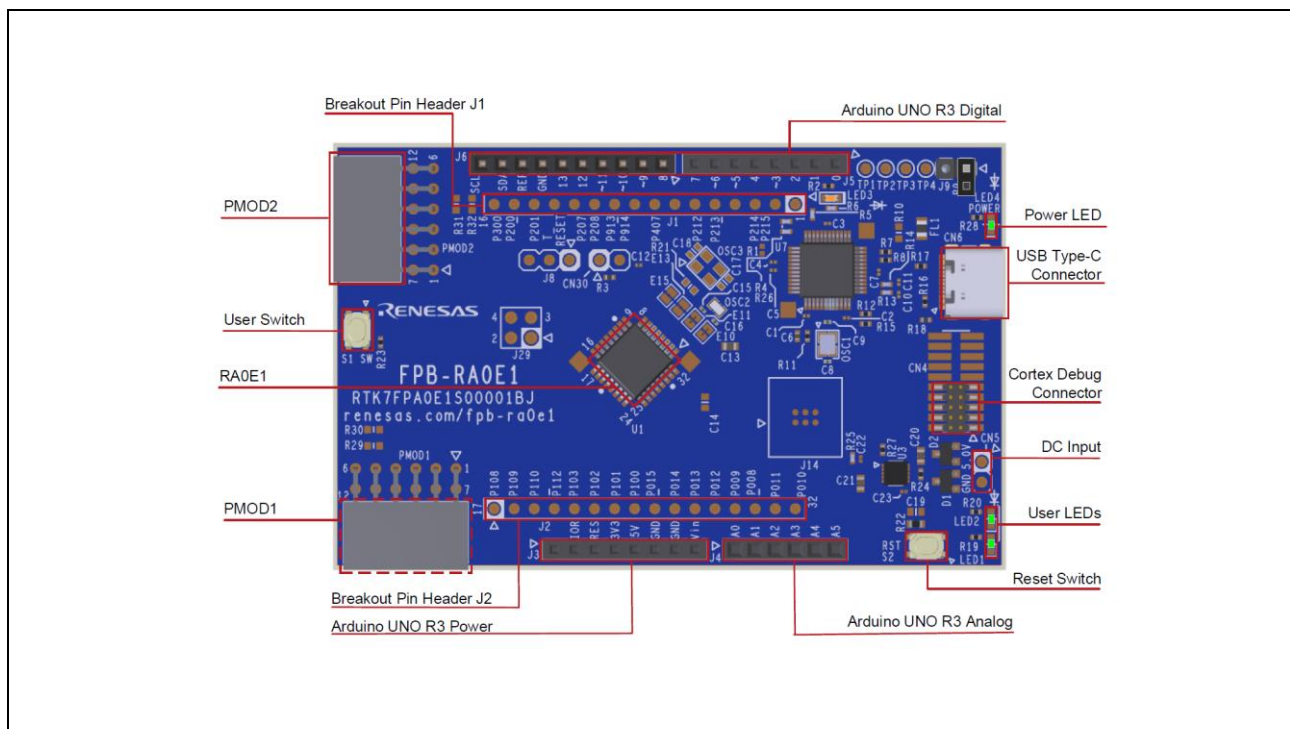


Figure 2-7 Overview of RA0E1 Fast Prototyping Board

The FPB supports on-chip debugging via USB connection. The Cortex Debug Connector supports the SWD (Serial Wire Debug) interface, enabling debugging of the target RA0 MCU using external debug tools.

### 2.2 Migration Process

The process flow for smooth migration to RA0 is shown in Figure 2-8. Each step is described in detail and examples are given in the following sections.

This chapter assumes the use of the GCC compiler.

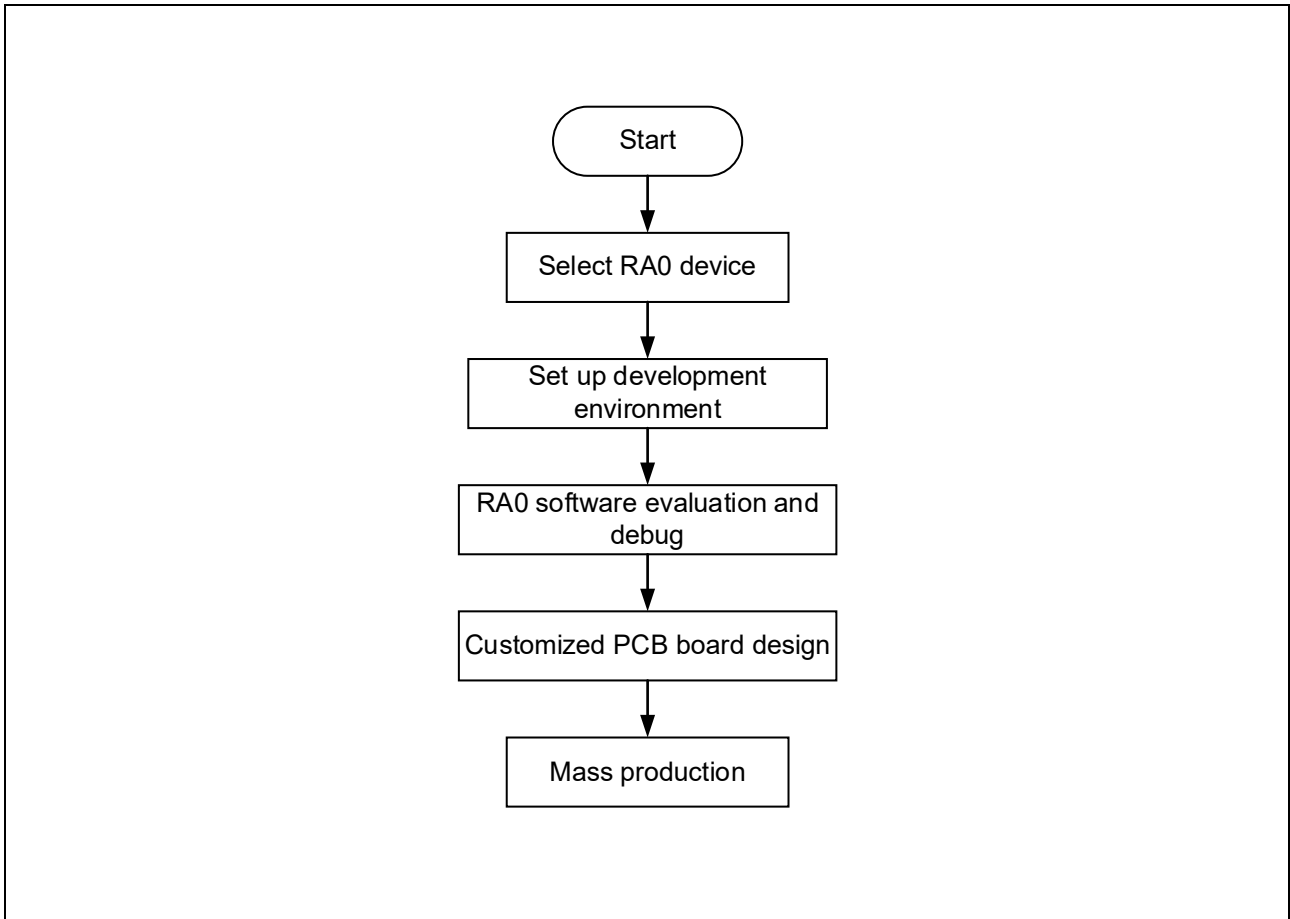


Figure 2-8 Flowchart of Migration to RA0

#### 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 RAOE1 as an example.

To narrow the choices down to a specific device, first access the RA0 Product page shown in Figure 2-9.

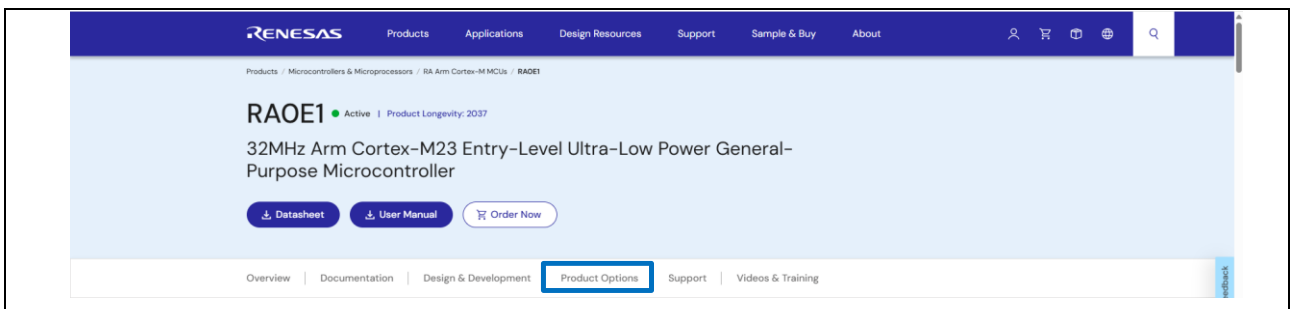


Figure 2-9 Product Page

# RA0 Series

## Migration Guide from Microchip PIC®16(L)F1 to RA0

Then select Product Selection and use the filters on the top shown in Figure 2-10 to narrow down the peripheral functions.

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 2-10 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 2-11.

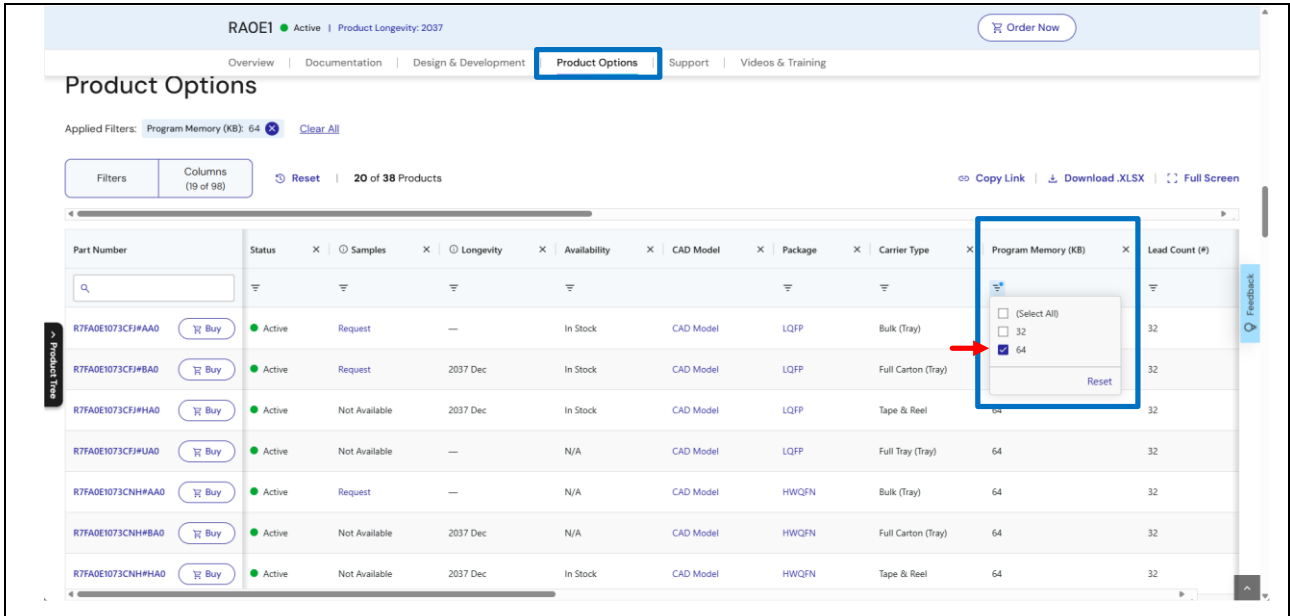


Figure 2-10 Product Selection

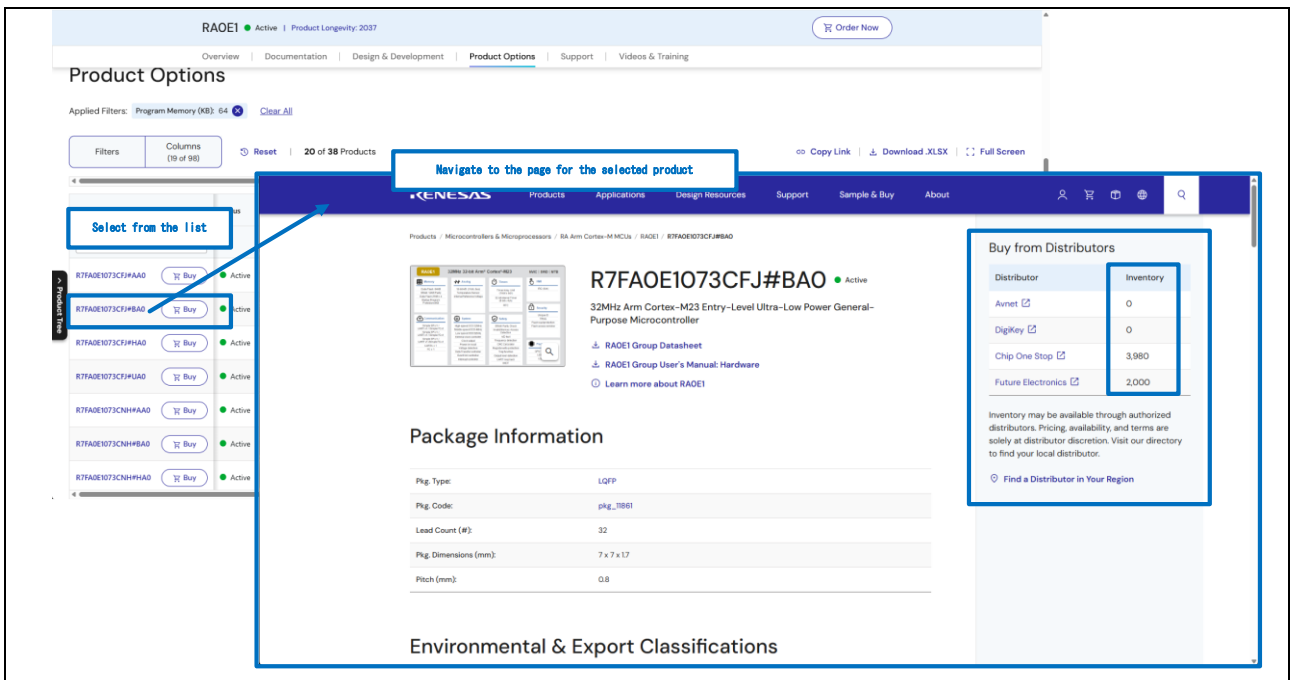


Figure 2-11 Product Selection

## RA0 Series

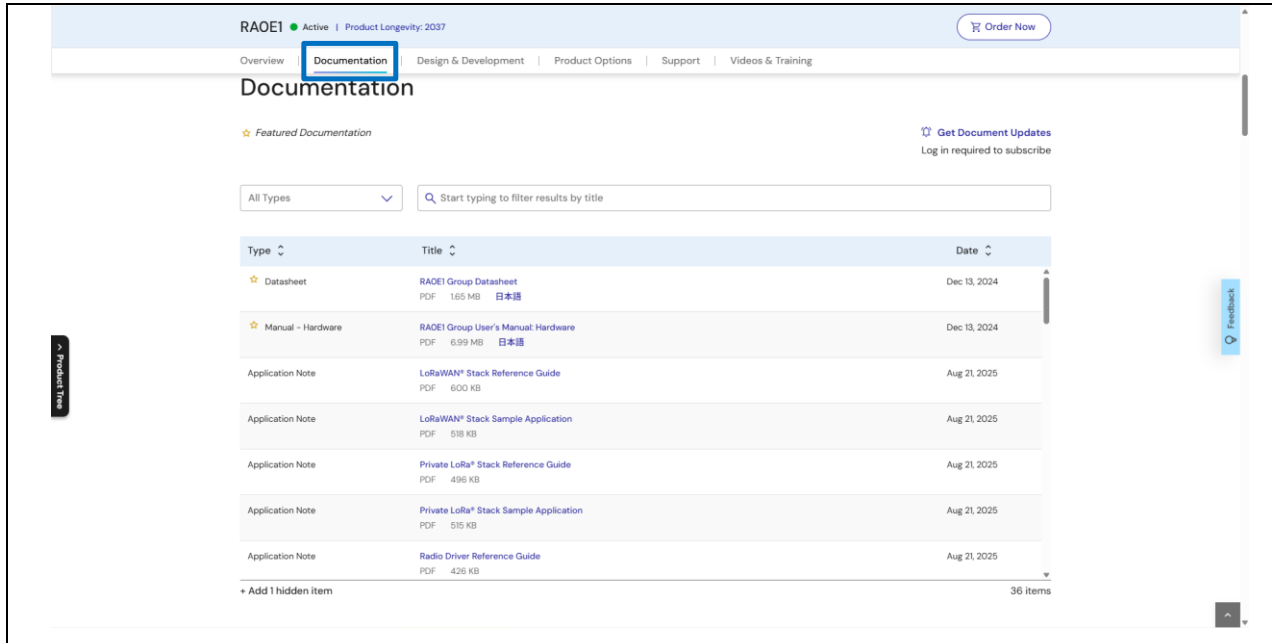
### Migration Guide from Microchip PIC®16(L)F1 to RA0

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 2-12).

The Datasheet provides a simple description of each product specifications.

The Manual Hardware Version describes detailed specifications of each product.

Each product page also lists technical documents, allowing you to find the documents necessary for development.



The screenshot shows the 'Documentation' section of a product page for the RA0E1 series. The page is titled 'Documentation' and includes a search bar and a list of documents. The 'Documentation' tab is highlighted in the navigation menu. The list of documents is as follows:

Type	Title	Date
★ Datasheet	RA0E1 Group Datasheet PDF 1.65 MB 日本語	Dec 13, 2024
★ Manual - Hardware	RA0E1 Group User's Manual: Hardware PDF 6.99 MB 日本語	Dec 13, 2024
Application Note	LoRaWAN® Stack Reference Guide PDF 600 KB	Aug 21, 2025
Application Note	LoRaWAN® Stack Sample Application PDF 518 KB	Aug 21, 2025
Application Note	Private LoRa® Stack Reference Guide PDF 496 KB	Aug 21, 2025
Application Note	Private LoRa® Stack Sample Application PDF 515 KB	Aug 21, 2025
Application Note	Radio Driver Reference Guide PDF 426 KB	Aug 21, 2025

At the bottom of the list, it says '+ Add 1 hidden item' and '36 items'.

Figure 2-12 Related Technical Documents

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

#### 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 **Lite Install** for the installation type and click [Next].

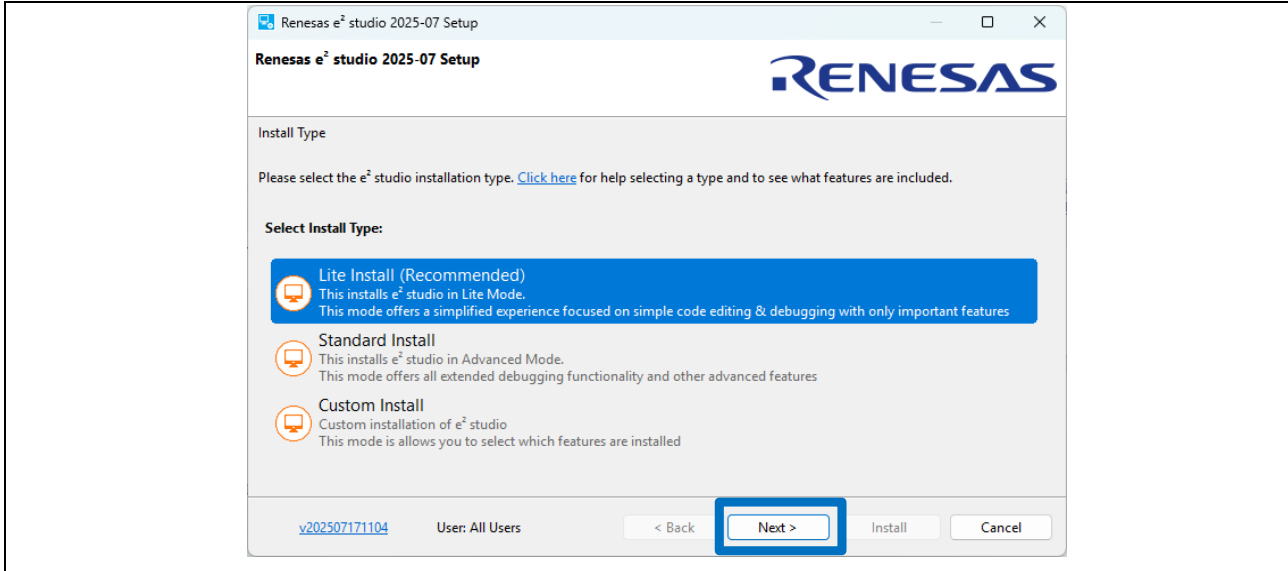


Figure 2-13 Select e² studio Installation Type

- (2) Select the RA family for the device. Press “Next” to select the compiler and other required software.

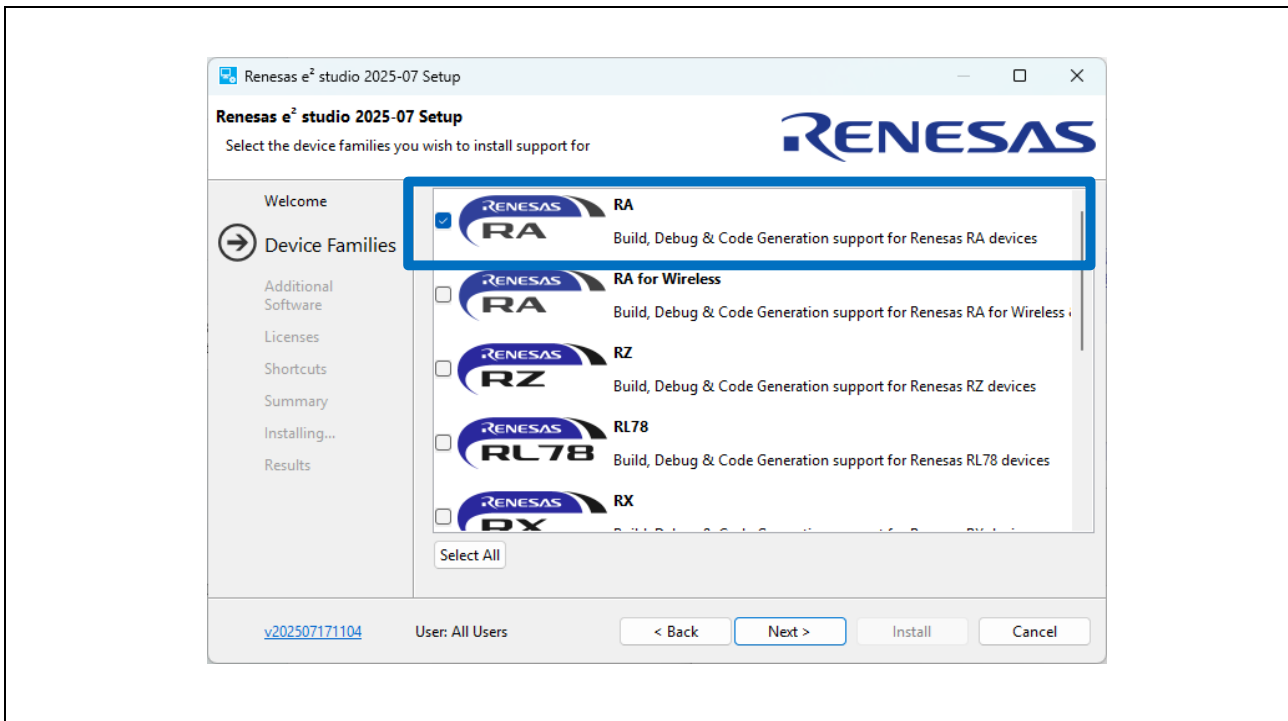


Figure 2-14 Overview of e² studio installation

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

(3) If there are no issues with the installation contents, press the install button to complete the installation of e<sup>2</sup> studio.

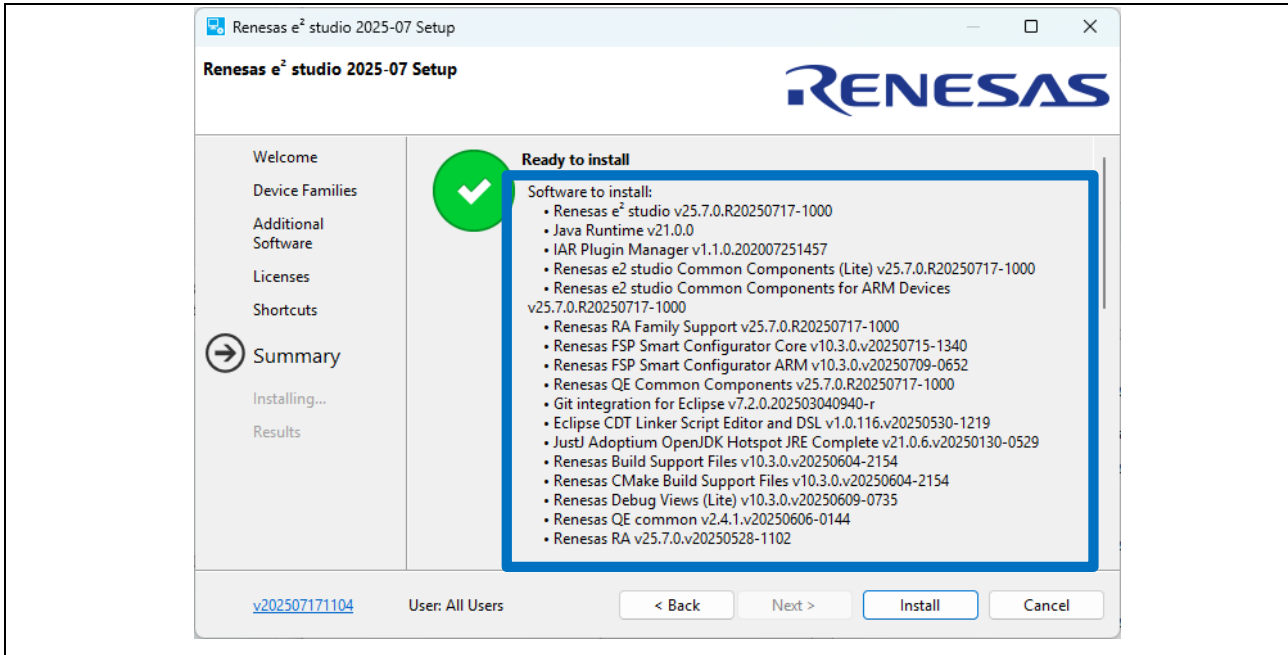


Figure 2-15 Overview of e<sup>2</sup> studio installation

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

#### 2.2.2.2 How to Use e<sup>2</sup> studio

- (1) Start a new workspace. "Workspace" is the path where project files and development environment settings are saved. The functions of e<sup>2</sup>studio are the same as those of MPLAB X IDE.

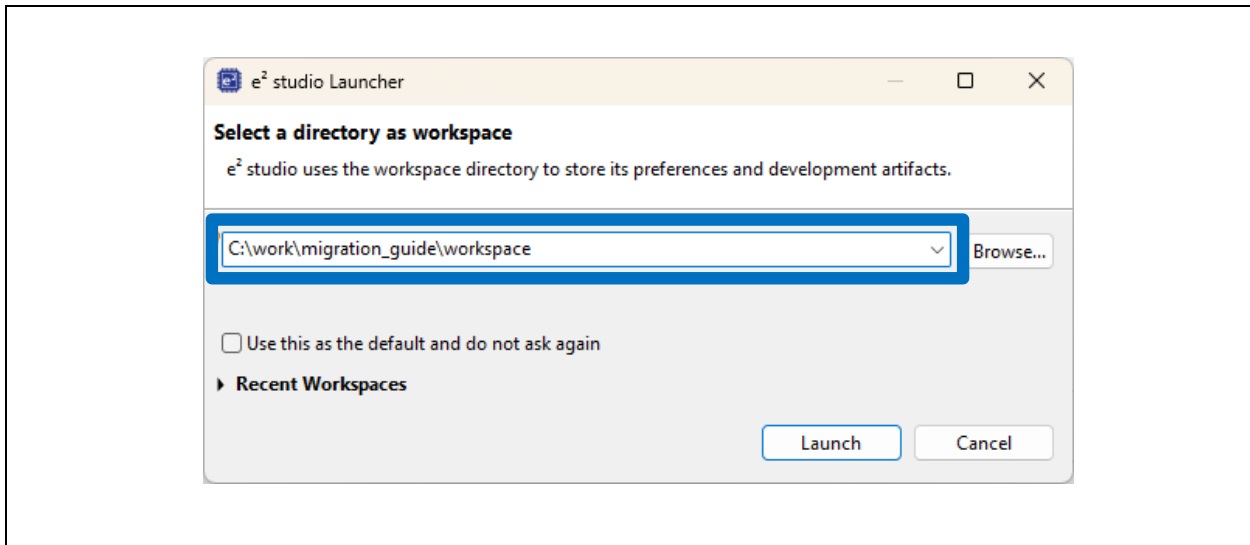


Figure 2-16 Select e<sup>2</sup> studio Workspace

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

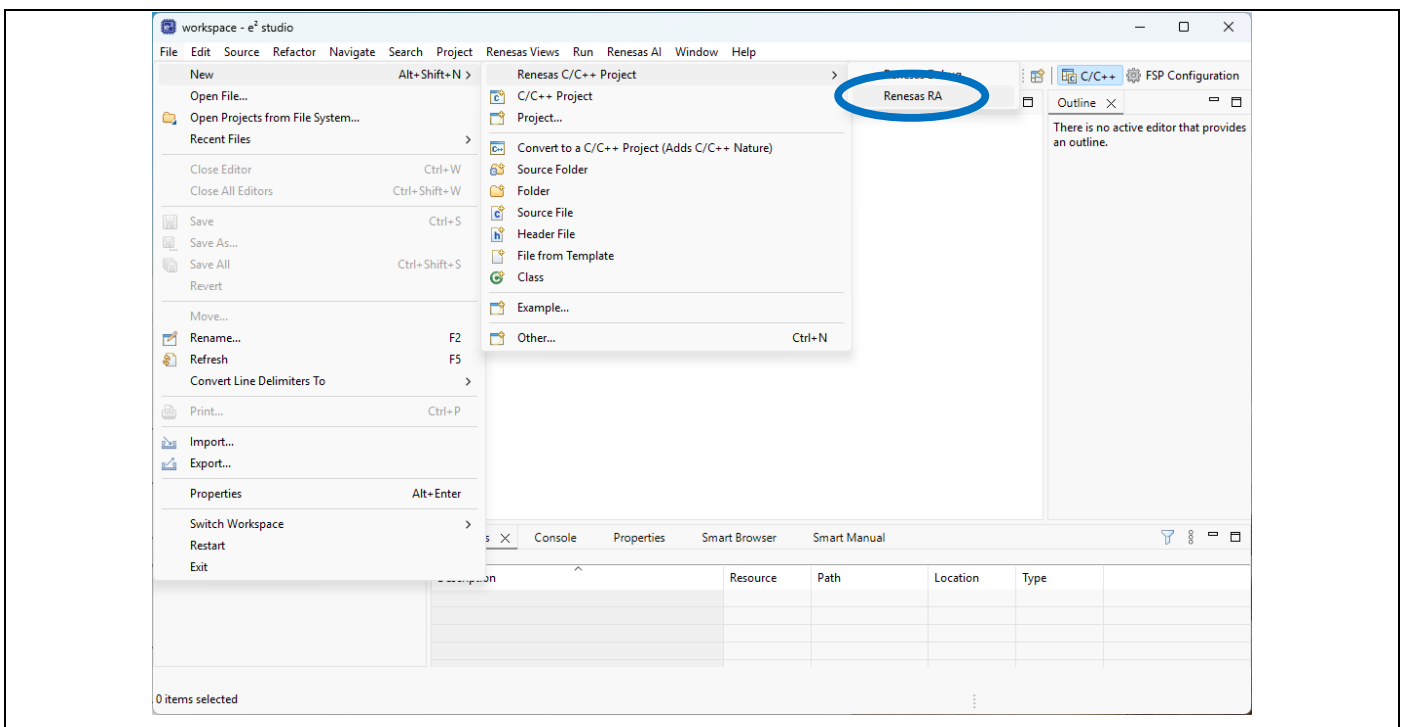


Figure 2-17 Create e<sup>2</sup> studio New Project and Select Device

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

(3) Following the setup instructions will lead you to the confirmation screen for creating a new project. Click the **Finish** button to generate the project.

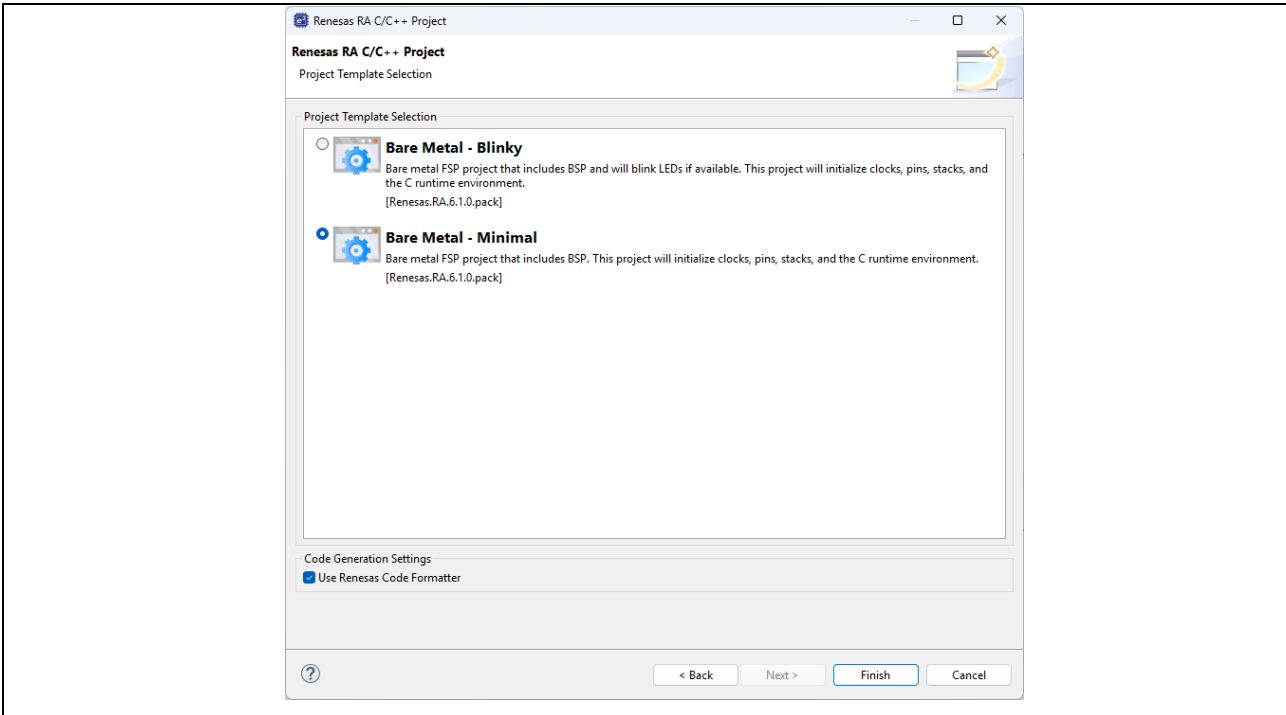


Figure 2-18 Create a New Project Overview

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

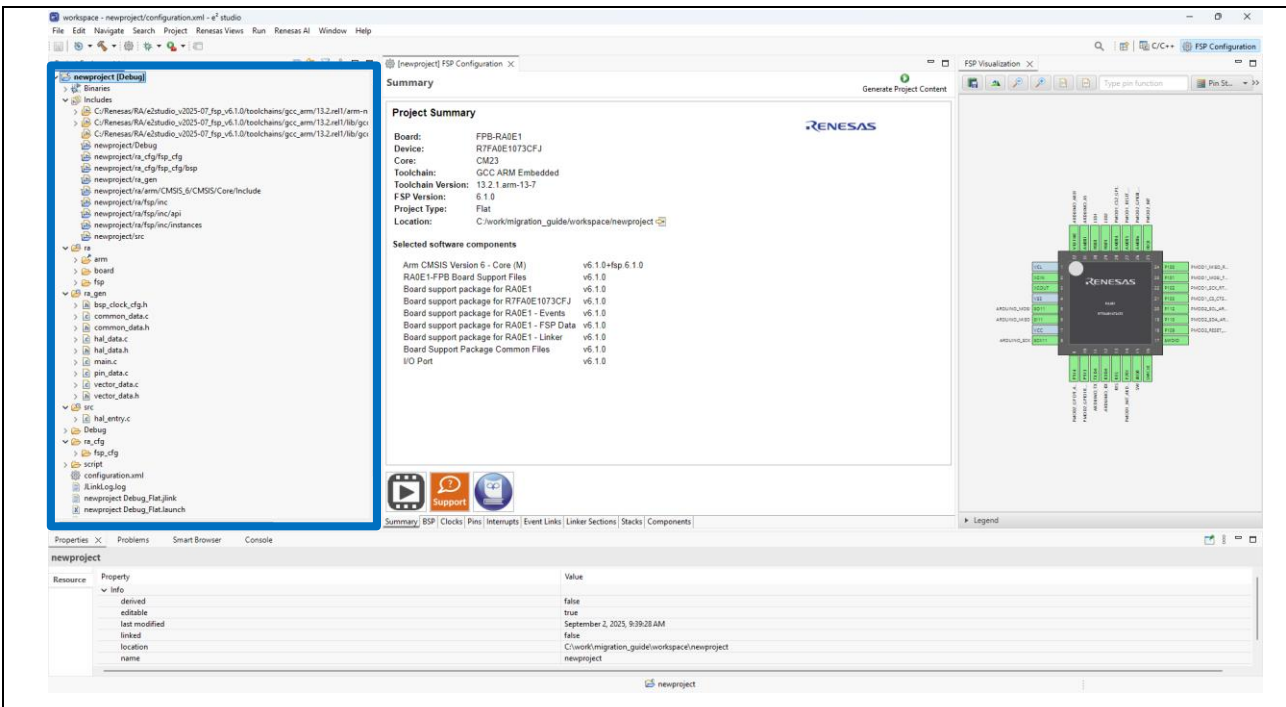


Figure 2-19 New Project Generation Completed

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

(5) Figure 2-20, Figure 2-21, Figure 2-22, Figure 2-23 are quick introduction of the e<sup>2</sup> studio functions.

Shortcut Key Functions:

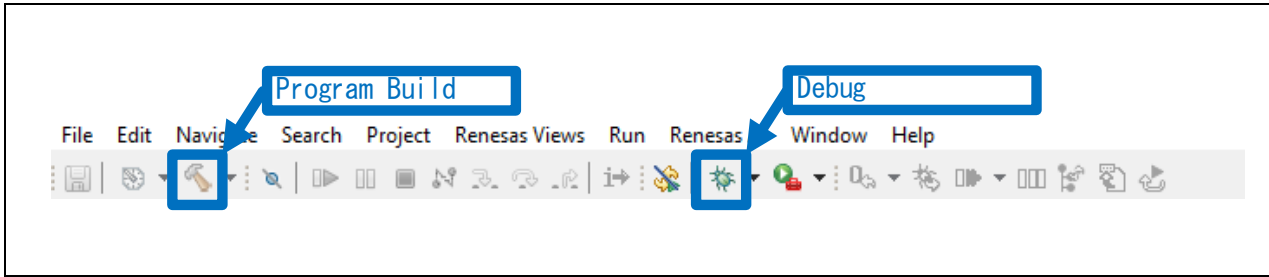


Figure 2-20 Commonly Used Functions

Debug Functions:

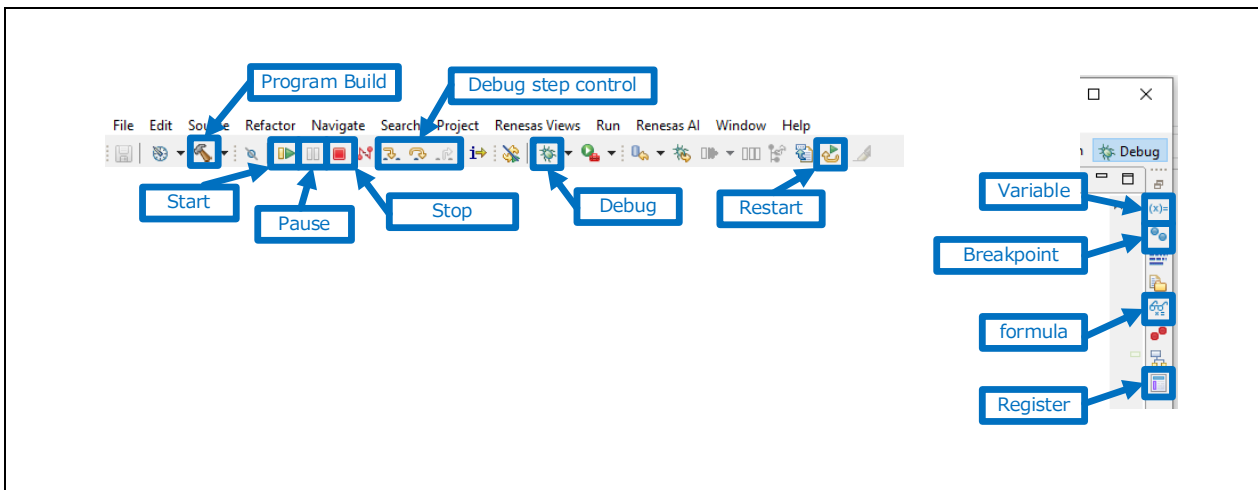


Figure 2-21 Commonly Used Debug Functions

## Migration Guide from Microchip PIC®16(L)F1 to RA0

Commonly Used Settings in the Project Properties:

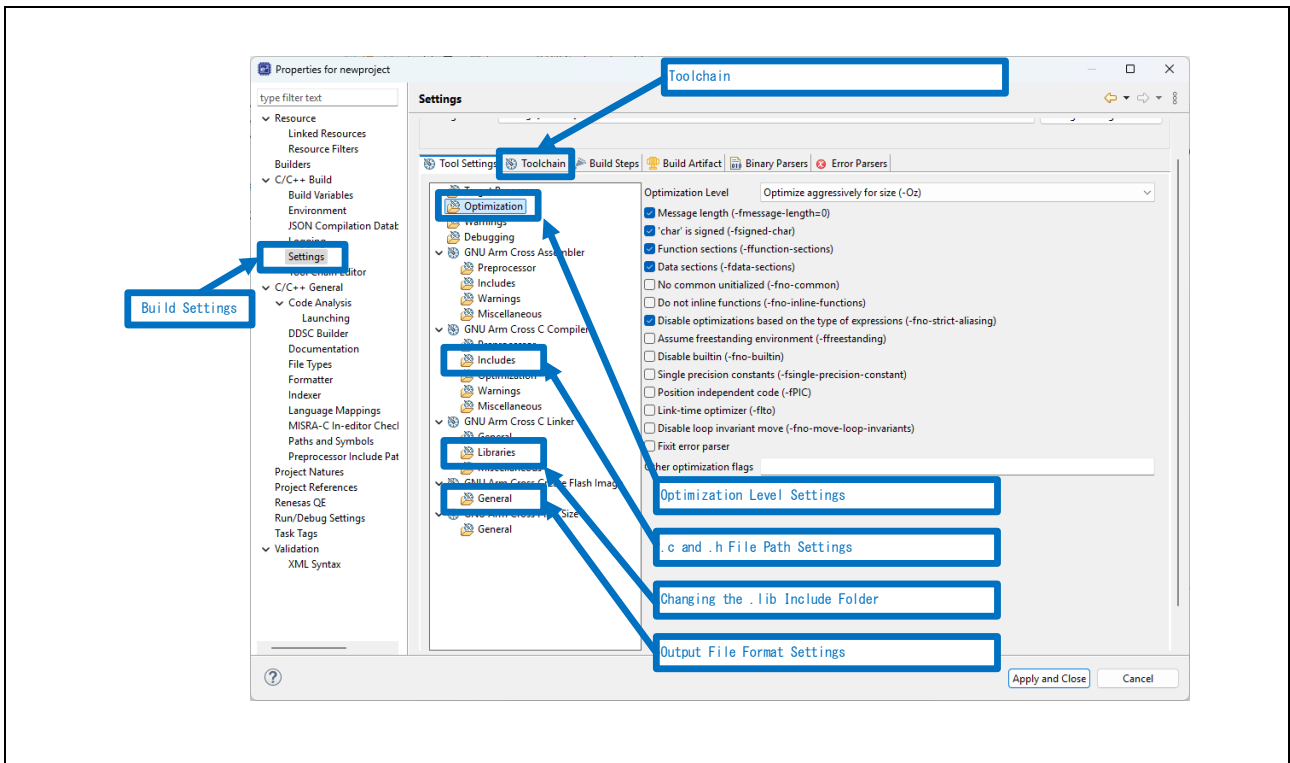


Figure 2-22 Commonly Used Project Settings (1/2)

# RA0 Series

## Migration Guide from Microchip PIC®16(L)F1 to RA0

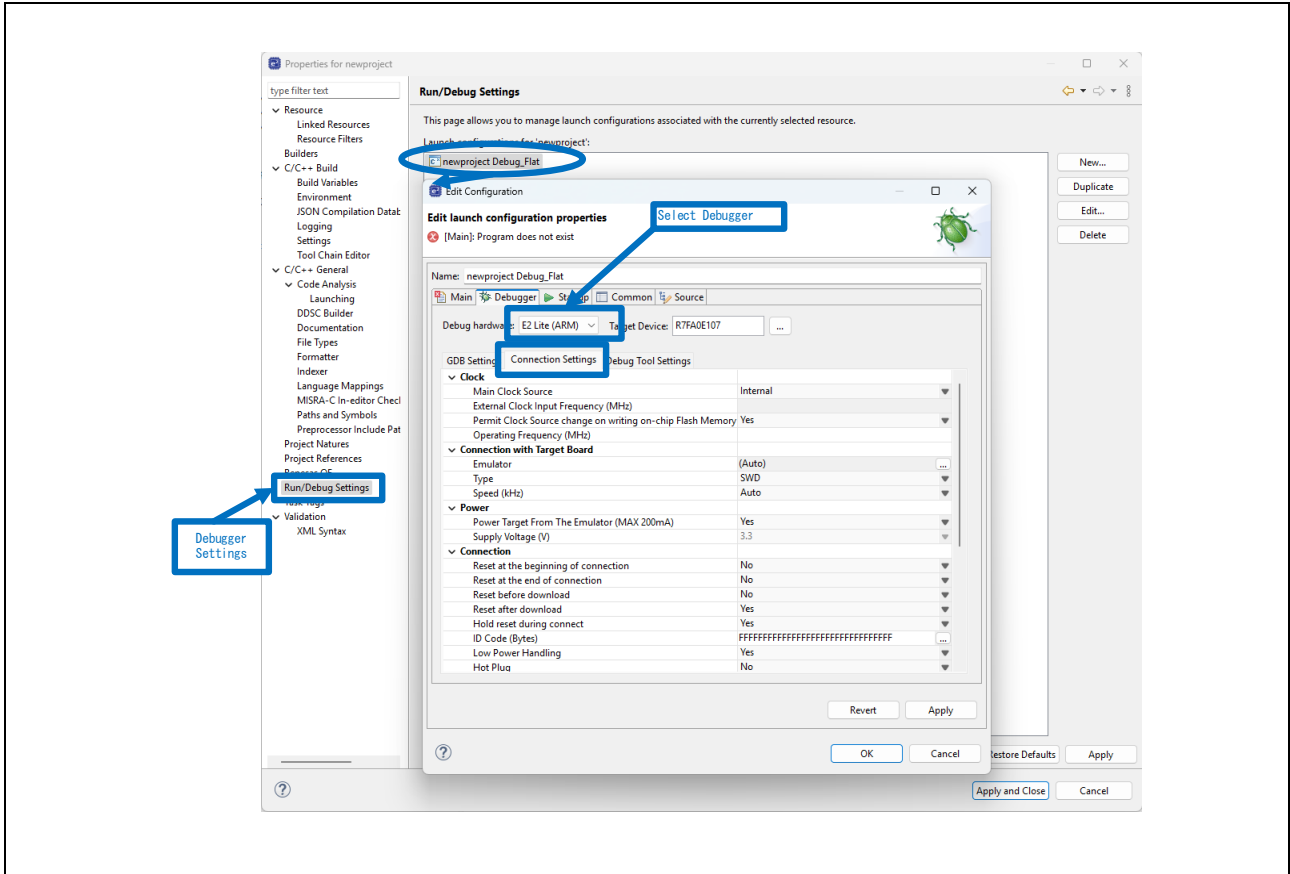


Figure 2-23 Commonly Used Project Settings (2/2)

For details, please refer to the Renesas Electronics website.

# RA0 Series

## Migration Guide from Microchip PIC®16(L)F1 to RA0

### 2.2.3 Step 3: Importing Software

This section provides a quick guide on how to import your sample to the e<sup>2</sup> studio, using FPB-RA0E1 as an example.

#### (1) Import the sample project downloaded from the menu.

Here, we will import the sample project downloaded from the Renesas website. The sample project is “tau\_fpb\_ra0e1\_ep” located within the “FPB-RA0E1 Example Project Bundle”.

Select [File] -> [Import...] to open the import dialog.

To import from the downloaded zip file into the workspace, select “Project from folder or archive” to open the import dialog.

Click the “Archive” button, select the project, then click the “Finish” button to add it to the workspace.

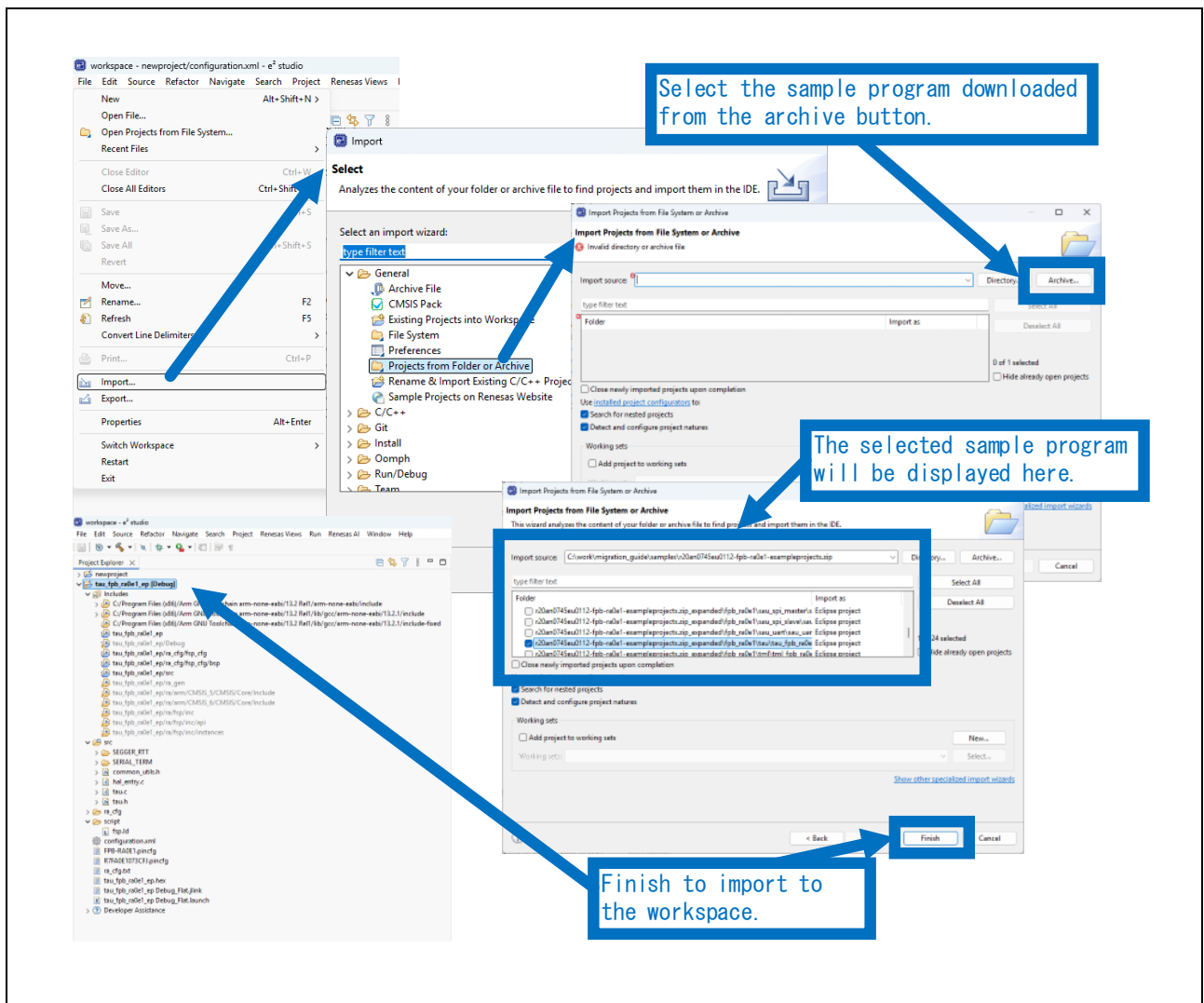


Figure 2-24 Import the Sample Program

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

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

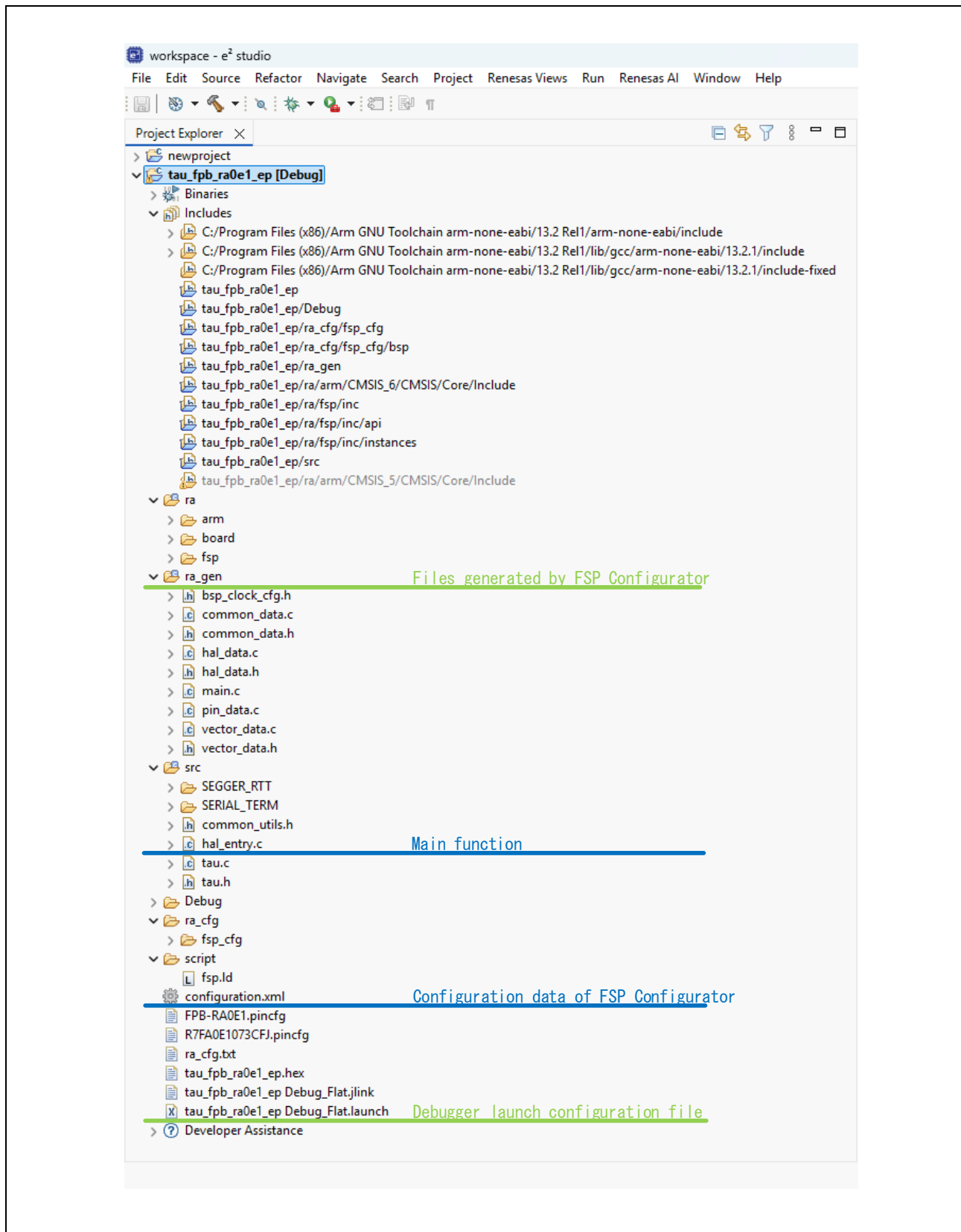


Figure 2-25 e2 studio Project Files

# RA0 Series

## Migration Guide from Microchip PIC®16(L)F1 to RA0

- (3) Double-click the configuration.xml file to start the FSP Configurator, which allows you to configure the necessary peripheral functions through the graphical interface.

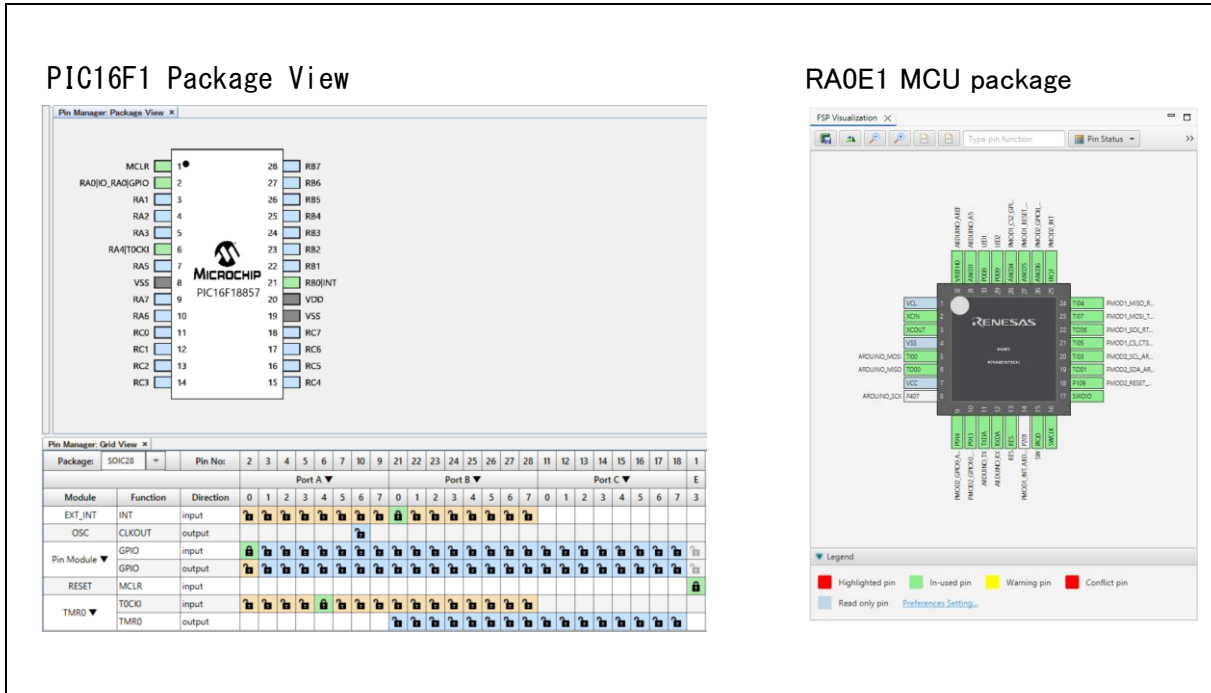


Figure 2-26 MPLAB Code Configurator and MCU View in FSP Configurator

- (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.

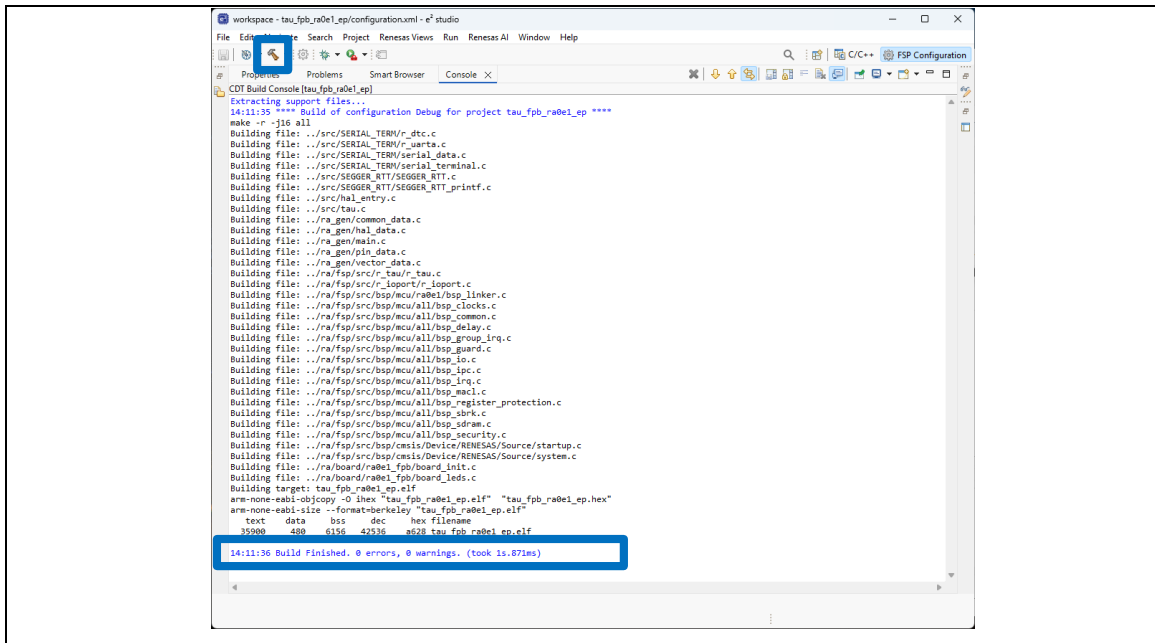


Figure 2-27 Build Finished

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

When building the sample project, toolchain-related errors may occur. If errors occur, select the project, open the [Properties] dialog via [Project] → [Properties], navigate to [C/C++ Build] → [Settings], select the [Toolchain] tab, and modify the settings.

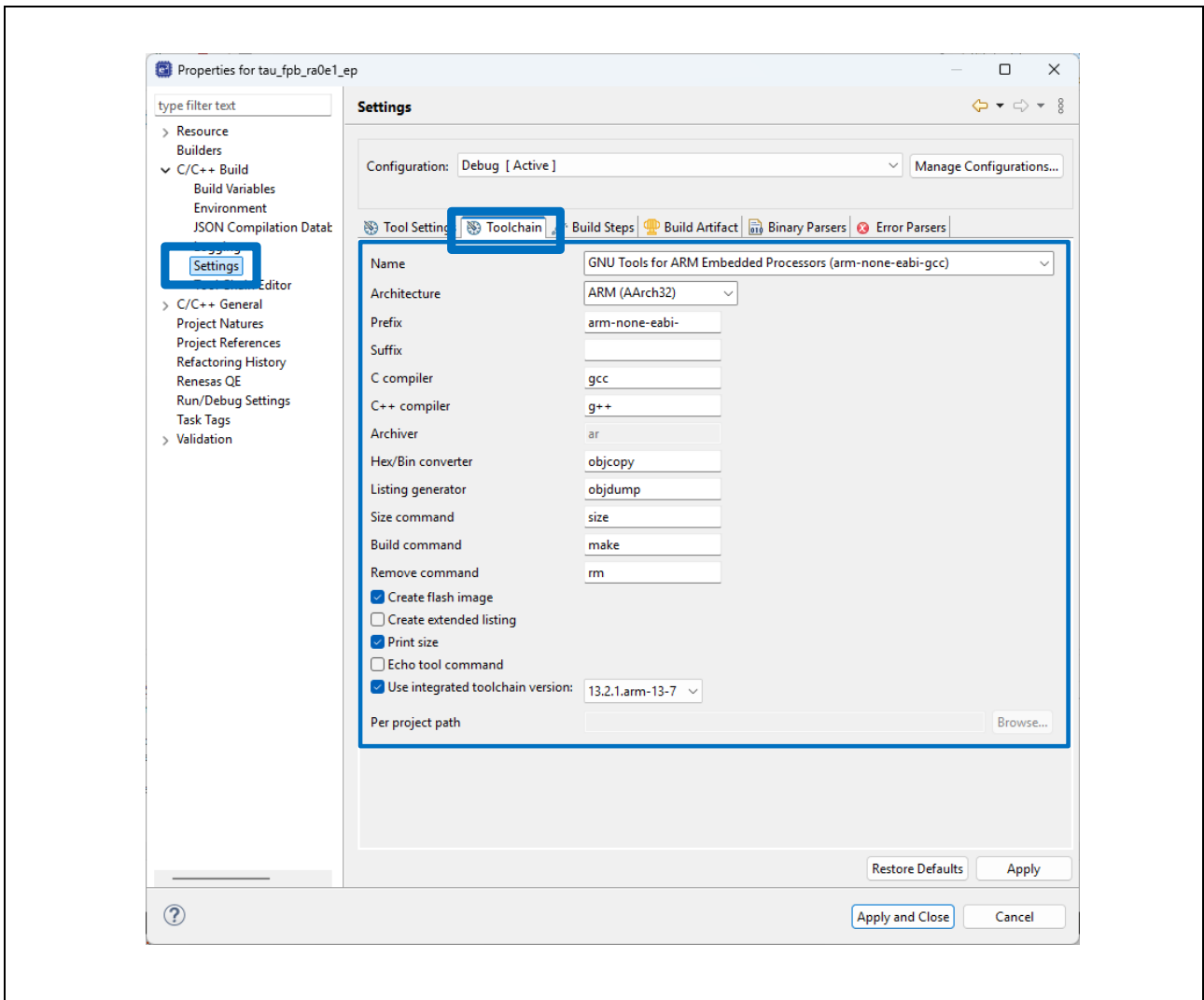


Figure 2-28 Toolchain setting

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

#### 2.2.4 Step 4: Evaluating the Software

Use the debugging functions described in Section 2.2.2.2 How to Use e<sup>2</sup> 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 e<sup>2</sup> studio.

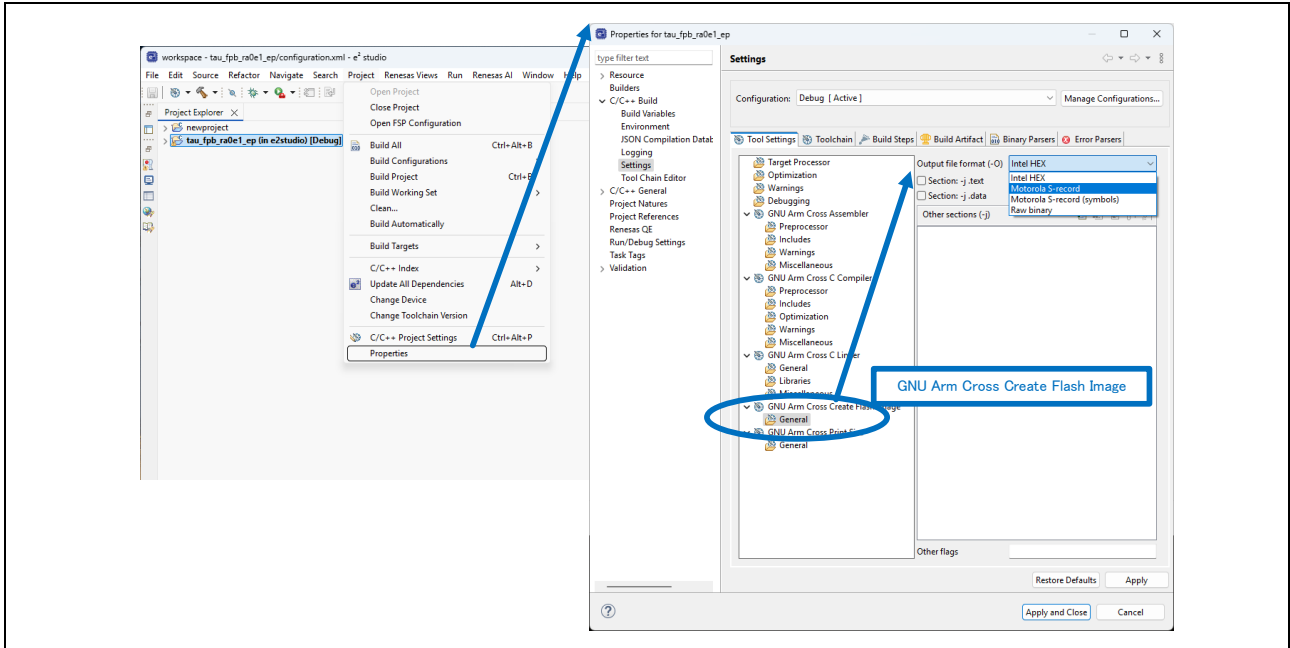


Figure 2-29 Create the Program File

(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.

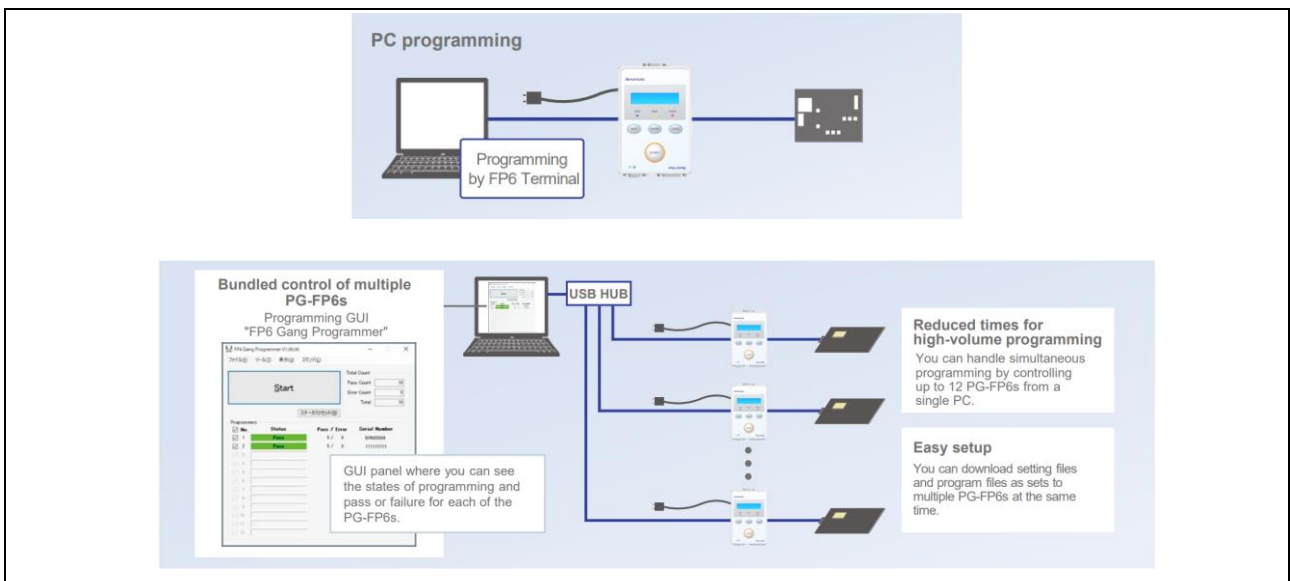


Figure 2-30 Flash Programming and Tools

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

For details regarding the PG-FP6 and FP6 Terminal, refer to the PG-FP6 V1.17 Flash Memory Programmer User's Manual (R20UT5654).

For details regarding the PG-FP6 Gang Programmer,, refer to the FP6 Gang Programmer V1.03 User's Manual (R20UT5403).

## RA0 Series

# Migration Guide from Microchip PIC®16(L)F1 to RA0

### 2.3 Example of importing a sample program

This section shows an example of the migration process using e<sup>2</sup> 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.
- (2) Setup e<sup>2</sup> studio. For details, refer to 2.2.2 Step 2: IDE Setup and Quick Introduction of e<sup>2</sup> studio.
- (3) Import the sample code.

After you have arranged the proper environment, the code can be imported to e<sup>2</sup> studio. The example uses the timer to control the LED. First, make sure you confirm the differences between the RA0E1 and PIC16(L)F18xxx timer modules. Then select the applicable sample code from the Renesas website.

In this section, we will explain using the [FPB-RA0E1 Example Project Bundle] (R20AN0745).

Download the sample project from the homepage, open e<sup>2</sup> studio, and select **[General] - [Project from folder or archive]** from the “Import - Select” dialog opened via **[File] - [Import...]**. In the “Import Project” dialog that opens, select the import source. Here, choose **[Archive...]** to import the downloaded sample project.

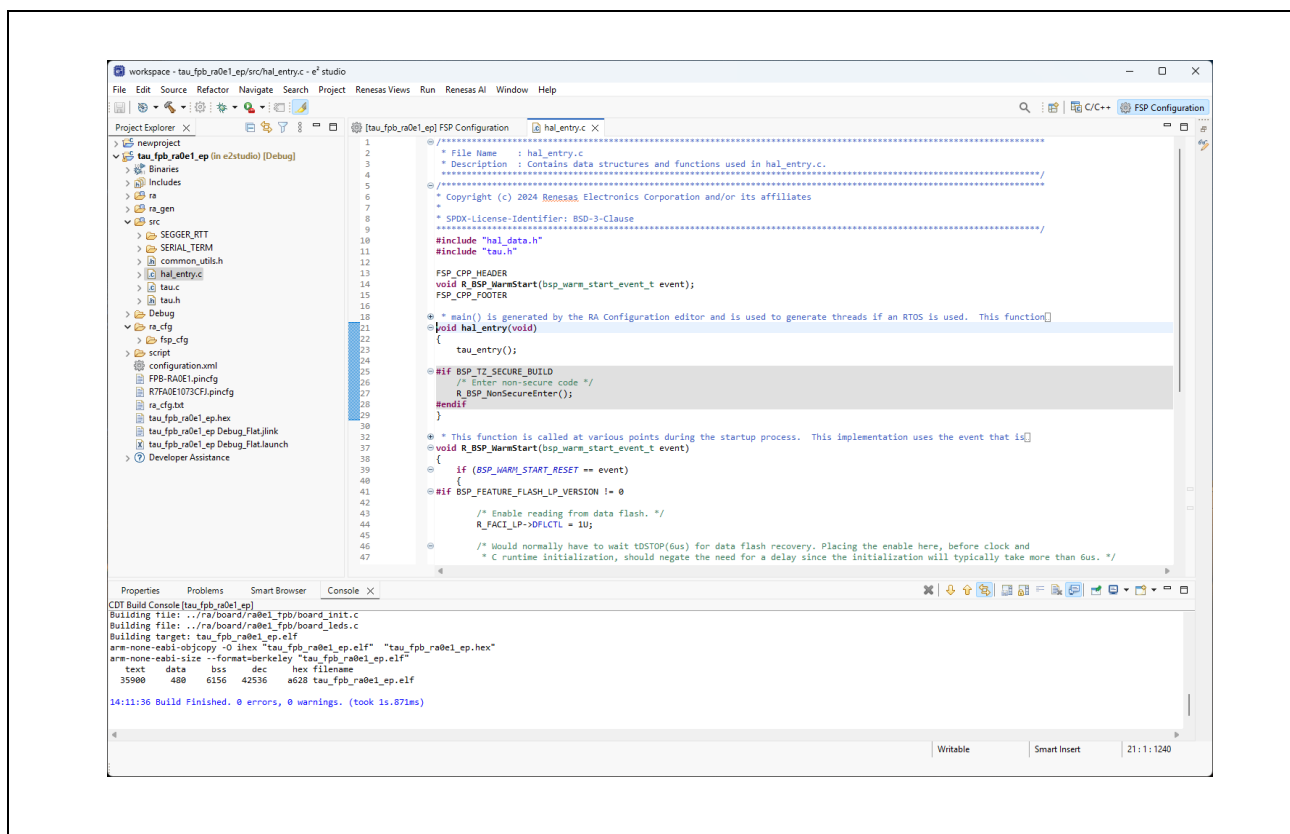


Figure 2-31 Sample Code File

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

#### (4) Change the project

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. Select the BSP tab from the FSP Configurator to change target board. To select the board as shown in Figure 2-32, choose [...], then navigate from the opened list to select the Target Board via “RA0” → ‘RA0E1’. In this example, “FPB-RA0E1” is selected.

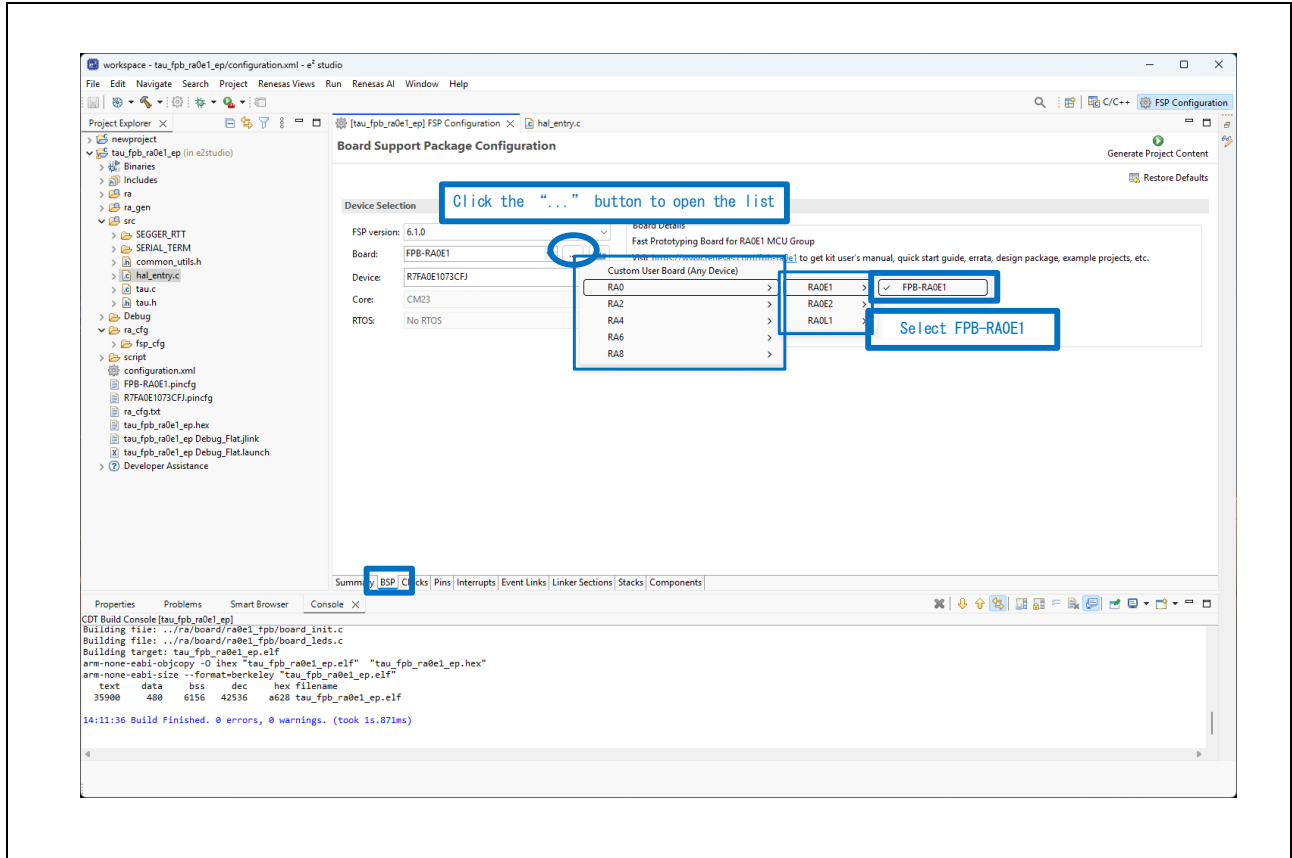


Figure 2-32 Device Selection in FSP Configurator

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

In the sample project, LED1 is assigned to I/O port P008 and LED2 to I/O port P009, which can be confirmed using the FSP Configurator. To verify LED1's port settings, as shown in Figure 2-33: Select the “Pins” tab, open Pin Configuration, choose the “Pin Function” tab, then navigate to [Ports] - [P0] - [P008].

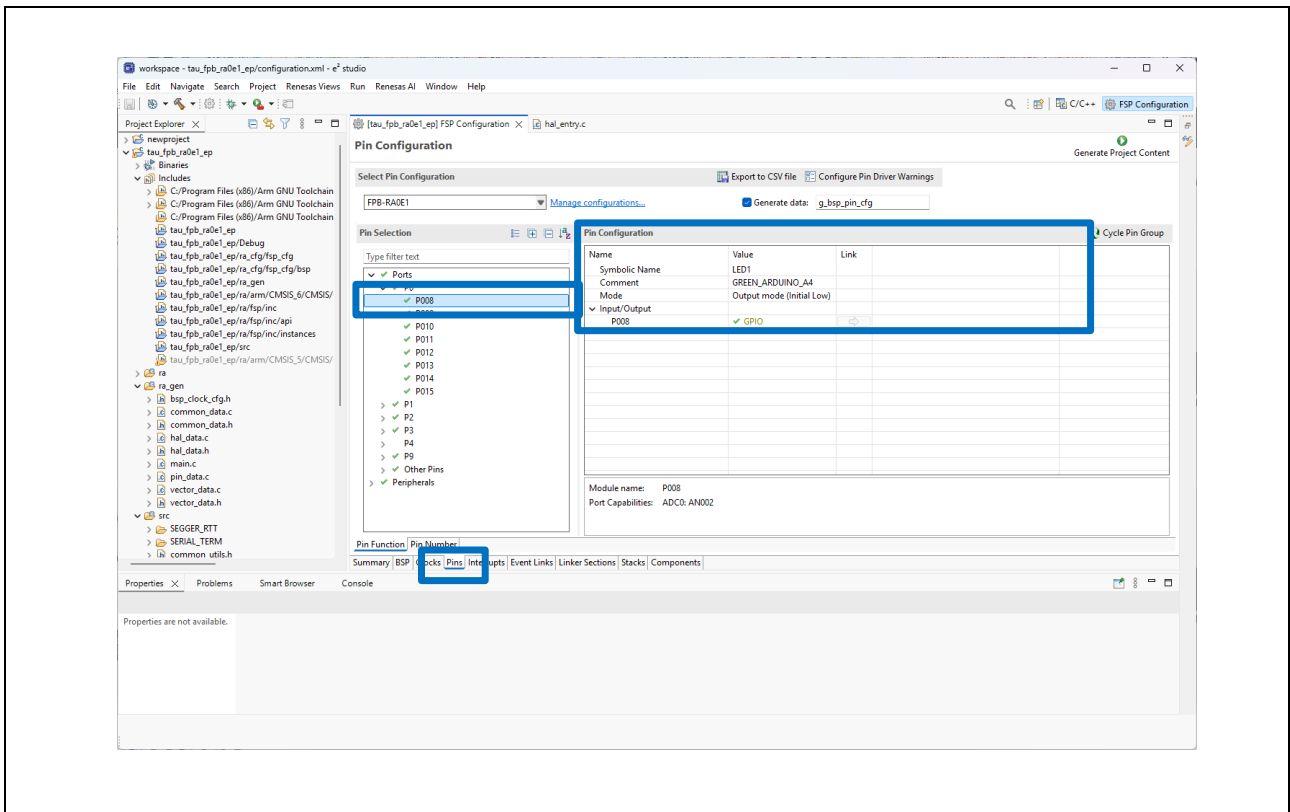


Figure 2-33 Board Setting in FSP Configurator

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

Ports are operated within a function called from the main function.

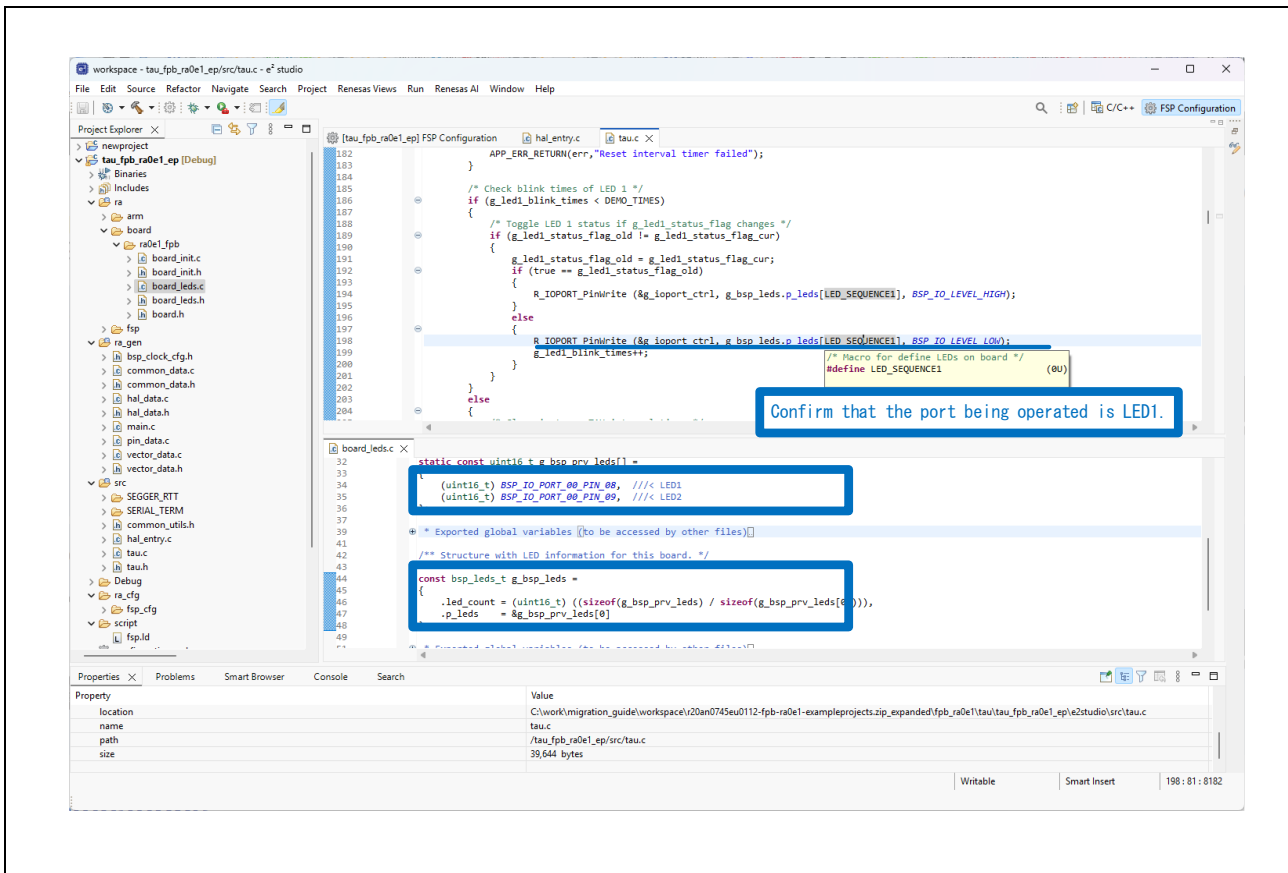


Figure 2-34 Port Operation Processing

# RA0 Series

## Migration Guide from Microchip PIC®16(L)F1 to RA0

Next, confirm the timer settings. When you select the **Stacks** tab, the list of stacks will be displayed, as shown in Figure 2-35.

Select a timer stack from the list and click the [Properties] tab at the bottom. In the left-hand tabs, [Settings] displays the timer's configuration values, while [API Info] shows API information. This stack has the interval timer function selected.

By clicking the FSP Configurator **Generate Project Content** button, the source code will be generated.

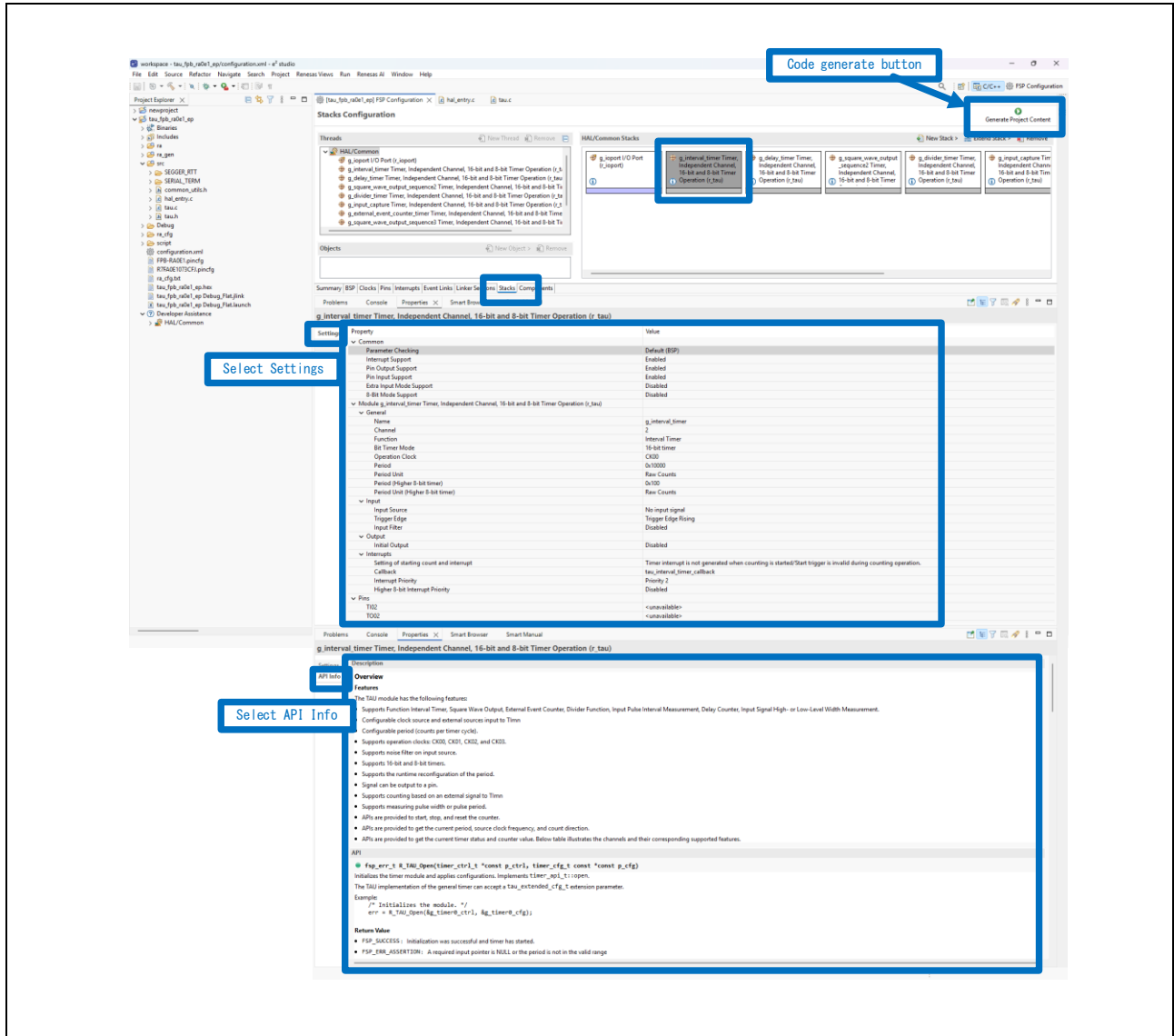


Figure 2-35 Interval Timer Settings in FSP Configurator

# RA0 Series

## Migration Guide from Microchip PIC®16(L)F1 to RA0

Developer Assistance allows you to reference the API information each component.

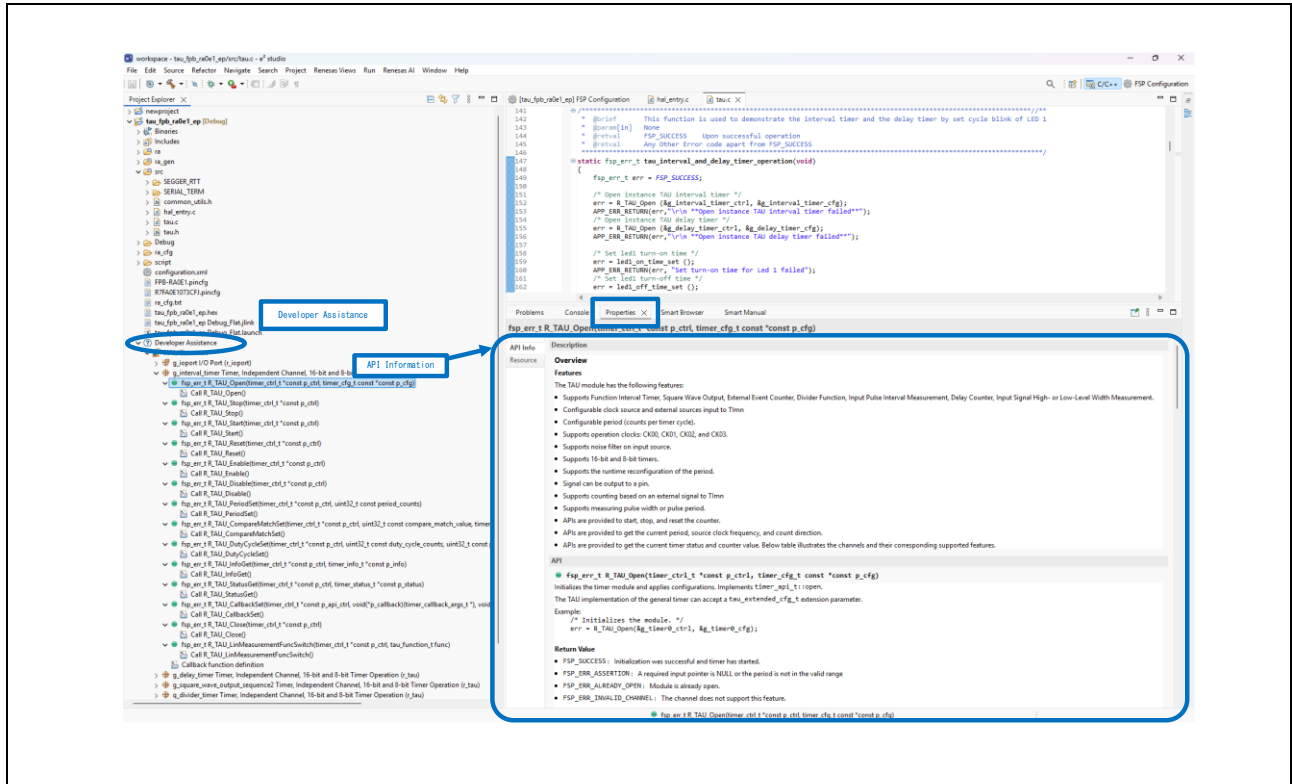


Figure 2-36 API Information Displayed in Developer Assistance

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

#### (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.

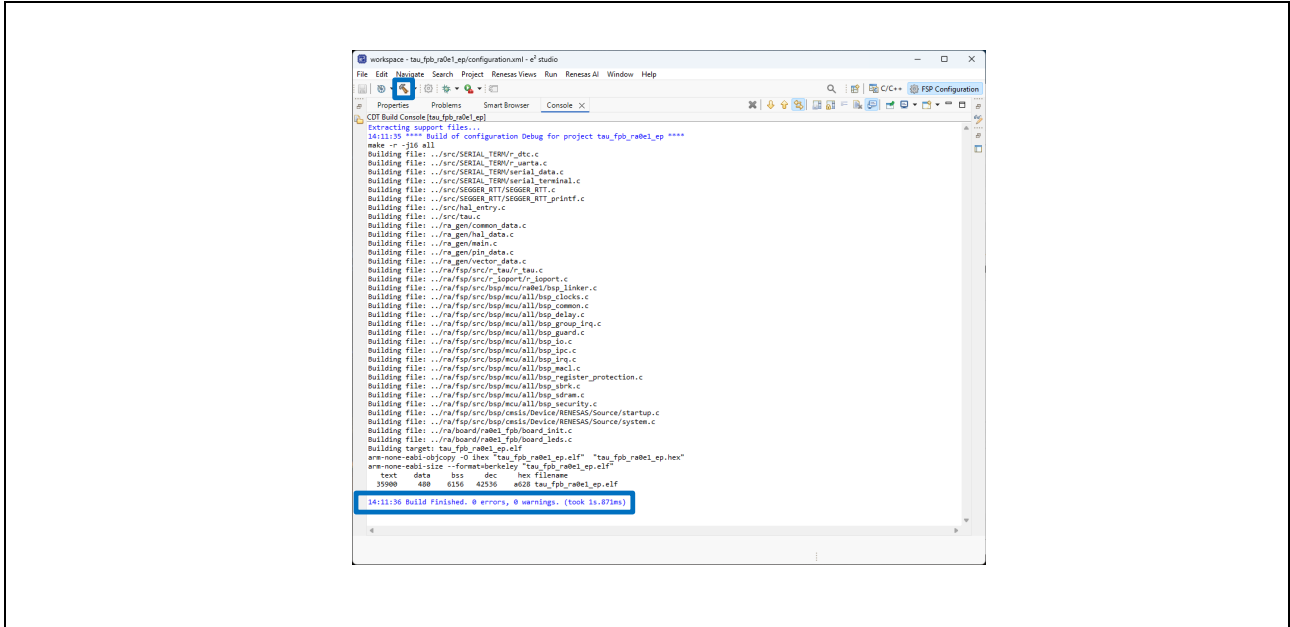


Figure 2-37 Build the Target

#### (6) Setup the Hardware.

#### (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();`.



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 e<sup>2</sup> studio for details).

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### 3. Core Architecture Comparison

##### 3.1 CPU

The RA0 is based on the Arm Cortex -M23 CPU core architecture and the PIC16(L)F1 is based on the enhanced Mid-Range CPU core architecture. Table 3-1 shows a general functional overview, comparing the RA0 CPU to the PIC16(L)F1 CPU.

**Table 3-1 CPU Comparison**

Item	PIC16(L)F1	RA0
Architecture	Enhanced Mid-Range CPU	Arm Cortex-M23
Instruction set	RISC	RISC
Pipeline	2-stage	2-stage
Operating clock (max.)	32MHz	32MHz
Minimum instruction execution time	0.125µs@32MHz	0.03125µs@32MHz
Address space	32KB	4GB
Number of instructions	49	84
Multiplication instruction	-	32-bit × 32-bit = 32-bit

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

## 3.2 Built-in Memory Comparison

### 3.2.1 Flash Memory Function

RA0 and PIC16(L)F1 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	PIC16(L)F183xx	RA0	
Flash memory	3.5~28KB	Code flash	16~128KB
		Data flash	1~2KB
Memory organization	Word size: 14 bits line size : 32 words In case of memory size is 28KB, 16384 words (512lines)	Code flash	Block size: 2KB
		Data flash	Block size: 256byte
Access	14bit (1word)	8 bits	
Programming unit	1line (32words)	Code8 flash	32 bits
		Data flash	8 bits
Erasing unit	1line (32words)	Code flash	2KB
		Data flash	256B
Data error correction/detection	N/A	N/A	
Specific Sequence	Unlock Sequence	N/A	
Security	Code Protection Write Protection	Protection against illicit tampering with or reading of data in flash memory	
Programming method	In-Circuit Serial Programming Self-Programming	Serial Wire Debug (SWD) programming Self-programming	
Programming count	10k (min.)	Code flash	1k (min.)
		Data flash	1000k (typ.)

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

#### 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 or SWD programming via an external device (SWD connection), self-programming.

##### 3.2.2.1 Flash Memory Area

Table 3-3 shows the flash memory areas in PIC16(L)F183xx and RA0.

**Table 3-3 Flash Memory Area Comparison**

PIC16(L)F183xx		RA0 (Code Flash)	
Program area	Reset vector	Program area	Application
	Interrupt vector	Startup area	Small program area
	Application		On-chip debugging ID setting
Startup area	User ID		Option setting memory
	Revision ID		Unique ID
	Device ID		Vector table
	Configuration Words		
	Calibration Words		

Additionally, the RA0 and PIC16(L)F1 feature the following non-volatile memory as data storage areas.

Table 3-4 shows the data storage areas for the PIC16(L)F183xx and RA0.

**Table 3-4 Comparison of Data Storage Areas**

Feature	PIC16(L)F183xx	RA0
Data storage area	EEPROM	Data flash
Access size	8 bits	8 bits
Write unit	8 bits	8 bits
Number of flash rewrites	100k (min.)	1000k (typ.)

##### 3.2.2.2 Flash I/O 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 FCR register is used to control command selection and the start/stop of command processing.

##### 3.2.2.3 Self-Programming in RA0

RA0 supports self-programming mode, enabling the flash memory to be rewritten from the user application.

The user application can rewrite the flash memory in this mode by using the Flash HAL driver API.

[Self-programming code example]

For information regarding self-programming code examples, please refer to the Renesas Electronics website.

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### 3.2.3 SRAM

Both RA0 and PIC16(L)F1 are both equipped with on-chip SRAM for storing the application data.

**Table 3-5 SRAM Comparison**

Type	PIC16(L)F183xx	RA0
SRAM memory	PIC16F180xx : 0.25~2KB PIC16(L)F183xx : 0.25~2KB PIC16(L)F184xx : 0.5~2KB	2~16KB
Parity check	N/A	Available
ECC	N/A	N/A
Write protection (RAM guard)	N/A	N/A
Instruction execution	Disable	Available

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

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.

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

#### 3.3 Power-On and Reset Overview and Comparison

Both PIC16(L)F1 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 3-6 shows the comparison of how this function works and which modules control the power-on process and reset between the two devices.

**Table 3-6 Power-On Overview and Comparison**

PIC16(L)F183xx		RA0	
Power-on reset (POR)	$V_{DD} > V_{POR}$ : POR reset is released. $V_{DD} < V_{PORR}$ : Remains in POR state.	POR (power-on reset circuit)	$V_{CC} > V_{POR}$ : POR reset is released. $V_{CC} < V_{PDR}$ : POR reset signal is generated.
Brown-out reset (BOR) - Level 0 <sup>Note 1</sup>	$V_{DD} > V_{BOR}$ : BOR is released. $V_{DD} < V_{BOR}$ : Remains in BOR state.	LVD0 (low-voltage detector) – reset mode <sup>Note 2</sup>	$V_{CC} > V_{det0}$ : LVD reset is released. $V_{CC} < V_{det0}$ : LVD reset signal is generated.
Brown-out reset (BOR) - Level 1 to 3	$V_{DD} > V_{LPBOR}$ : LPBOR is released. $V_{DD} < V_{LPBOR}$ : Remains in LPBOR state.	N/A	
N/A		LVD1 (low-voltage detector) – interrupt and reset mode <sup>Note 2</sup>	$V_{CC} \geq V_{det1}$ : LVD reset is released. $V_{CC} < V_{det1}$ : LVD reset signal is generated and released after the specified time
N/A		LVD1 (low-voltage detector) – interrupt mode <sup>Note 2</sup>	interrupt request signal is generated when $V_{CC}$ crosses $V_{det1}$ .

Note 1. The level of voltage detection can be selected in configuration words.

Note 2. The level of voltage detection can be selected in the Option-setting memory or the LVD1CR register.

The relationship among RA0 voltage thresholds is  $V_{PDR} < V_{POR} < \text{lower limit of the operating voltage} < V_{det0} < V_{det1}$ . The relationship among PIC16(L)F183xx voltage thresholds is  $V_{PORR} < V_{POR} < V_{BOR} < V_{LPBOR}$ .  $V_{LPBOR}$  is the smallest value of  $V_{DD}$  specified to enable the internal circuit's normal operation.

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

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 status flag register (RESF) and the power-on reset status register (PORSR) to check which event triggered the reset request.

**Table 3-7 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 3-8 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

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

#### 3.4 Clock Overview and Comparison

##### 3.4.1 Oscillator

Both RA0 and PIC16(L)F1 are equipped with many types of internal and external clock generation circuits to reduce system cost and power consumption. Table 3-9 shows the comparison of the clock circuits in RA0 and PIC16(L)F1.

**Table 3-9 Oscillator Comparison**

Type	PIC16(L)F183xx	RA0
Internal oscillator	$F_{HFOSC}$ : High-Frequency Internal Oscillator of 1/2/4/8/12/16/32MHz When using a PLL ( $\times 2$ ), a maximum of 32 MHz is possible.	HOCO: High-speed on-chip oscillator clock with maximum frequency of 32MHz MOCO: Middle-speed on-chip oscillator clock with maximum frequency of 4MHz
	$F_{LFOSC}$ : Low-Frequency Internal Oscillator of 31kHz.	Low-speed on-chip oscillator clock with maximum frequency of 32.768 kHz
External oscillator	External Oscillator LP mode: $\sim 100$ kHz ( $F_{LP}$ ) XT mode: $\sim 4$ MHz ( $F_{XT}$ ) HS mode: $\sim 20$ MHz ( $F_{HS}$ ) When using a PLL ( $\times 4$ ), a maximum of 32 MHz is possible.	MOSC: Main clock oscillator with maximum frequency of 20MHz
	$F_{T1}$ : Secondary oscillator of 32.768kHz	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 3-10 shows RA0's clock generation circuits. Refer to the each RA0 Series User's Manual: Hardware for details.

**Table 3-10 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)

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### 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 3-11 Clock Signal Comparison**

Clock Description		PIC16(L)F183xx Clock	RA0 Clock
External clock input	High-speed	OSC1	EXCLK(MOSC)
	Low-speed	SOSCI	XCIN(SOSC)
High-speed external clock		Output of EXTOSC	EXCLK
Low-speed external clock		Output of SOSC	FSUB <sup>Note 1</sup>
Main system clock		Clock set by OSCCON2 register	FMAIN <sup>Note 2</sup>
CPU/peripheral clock		System Clock Peripheral Clock	FMAIN, FSUB, LOCO

Note 1. FSUB is the subsystem clock supplied by a low-speed external oscillator (XCIN).

Note 2. RA0's main system clock is supplied by MOSC, HOCO or MOCO.

### 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 Main clock oscillator and High-speed on-chip oscillator are stopped, causing the entire system to halt. It can reduce CPU power consumption as much as possible.

#### 3.5.1 Comparison of Operation Modes

Table 3-12 compares operation modes in RA0 and PIC16(L)F183xx.

Table 3-12 Comparison of Operation Modes

PIC16(L)F183xx		RA0	
Operation Mode	Description	Operation Mode	Description
RUN	The CPU operates using the clock signals output by internal and external oscillators. All peripherals are active.	Normal Operation	CPU operates with the main system clock or the subsystem clock. All peripherals are active.
DOZE	Reduces CPU activity and program memory access without affecting peripheral function operation (by inserting a wait time for each instruction processed). Operation of the internal or external oscillator continues.	SLEEP	CPU stops operating. The main system clock and the subsystem clock keep operating. All peripherals are active.
IDLE	The CPU stops operating, but peripheral functions continue to operate. While peripheral devices operate, the CPU clock stops completely.	SNOOZE	CPU stops operating. HOCO or MOCO operation begins, but the main system clock remains stopped. The state of the subsystem clock during standby mode is maintained. Peripheral functions such as ADC, UART, CSI, or DTC transfers can operate without activating the CPU.
SLEEP	CPU stops operating. Peripheral function using the internal oscillator and secondary oscillator as clock sources remain operational.	Software STANDBY	CPU stops operating. HOCO and MOCO stop. The subsystem clock retains its state prior to the standby mode setting. The system stops, except for peripheral functions operating on the subsystem clock.

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### 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 and middle-speed 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.

Returning to normal operation takes longer than waking from sleep mode because it involves the main clock and on-chip oscillator oscillation, but it is suitable for applications requiring minimal power consumption.

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

3.6 Comparison of Interrupts and Events

3.6.1 Interrupts and Exceptions

Both RA0 and PIC16(L)F1 register and map interrupts and exception vectors according to the available peripheral functions of the device. Table 3-13 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, it will accept a higher-priority interrupt as a multiplexed interrupt when it occurs.

Table 3-13 Comparison of Interrupts

Peripheral Function	PIC16(L)F183xx	RA0
Interrupt types	Peripheral interrupts Internal interrupt request x 33 (max.) External interrupt request x 2 (max.)	Maskable interrupts Internal interrupt request x 33 (max.) External interrupt request x 6 (max.)
	Reset interrupt MCLR pin input, watchdog timer, Power-On Reset, Brown-Out Reset, Low-Power Brown-Out Reset, RESET instruction, Stack Overflow/underflow, Programming mode exit	[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	No priority. The software identifies interrupt requests and controls the priority of interrupt handling.	Default priority The default priority is used when multiple maskable interrupts with the same interrupt priority occur.
		Interrupt priority level Four programmable priority levels (0, 1, 2, 3) are available for maskable interrupts.
Priority setting	No configuration registers. Priority is controlled via software.	NVIC IPRx Register NVIC IPRx is used to set the interrupt priority of a peripheral.
Interrupt mask	PIEx register Registers PIEx are used to enable/disable peripheral interrupts.	ISER and ICER in NVIC Registers ISER and ICER are used to enable/disable peripheral interrupts.

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.

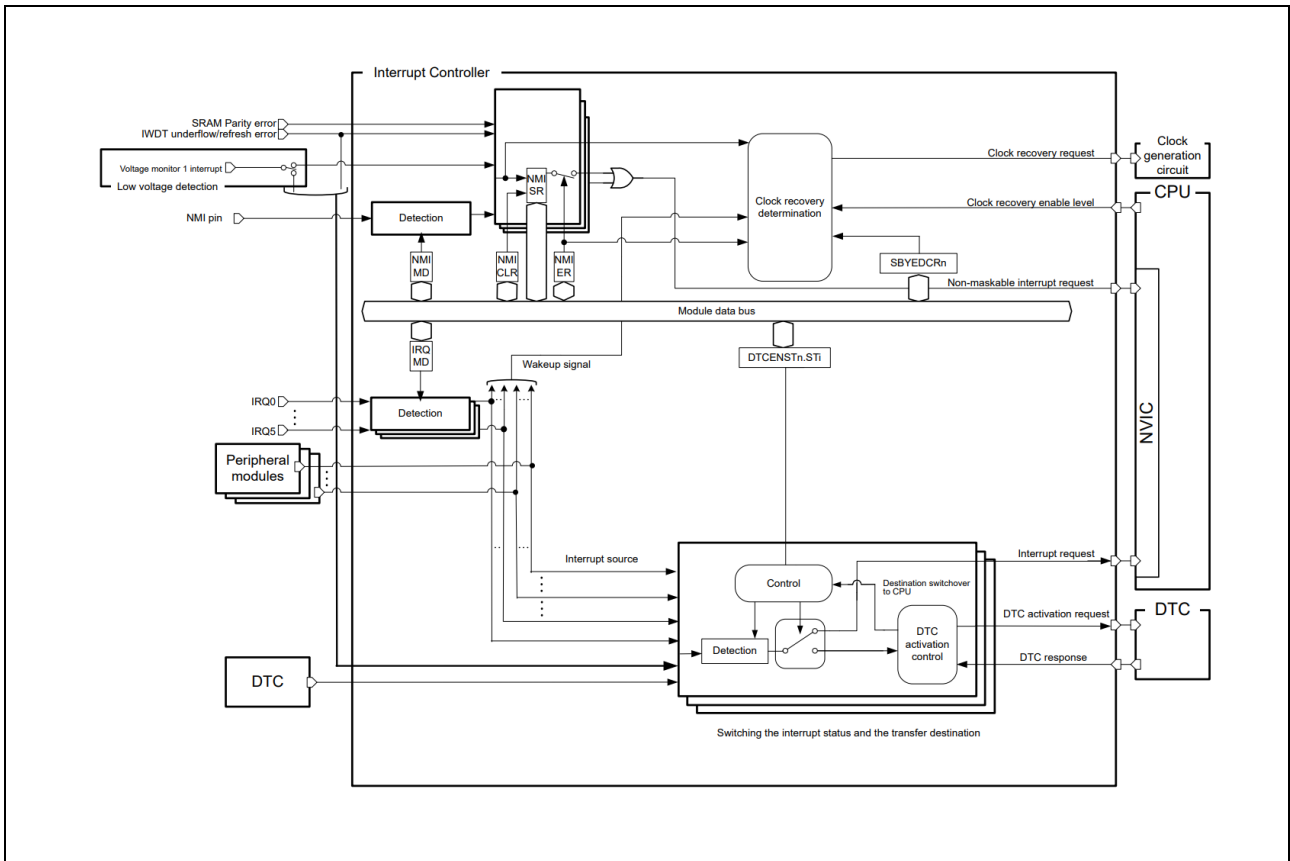


Figure 3-1 Configuration of RA0 Internal Maskable Interrupts

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

#### 3.6.1.2 Interrupt Control in PIC16(L)F183xx

Figure 3-2 shows the peripheral interrupt configuration for the PIC16(L)F183xx.

The PIC16(L)F183xx enables or disables interrupt conditions via the PIE<sub>x</sub> register. Before any interrupt request flag is set, the GIE bit in the INTCON register must be set to 1 to enable interrupts.

Unlike RA0, priority is not handled by the device but is controlled by the software handling the interrupt.

The interrupt-handling software must poll the interrupt request flags to identify the interrupt source. Furthermore, to prevent the interrupt handler from looping, the interrupt request flag must be cleared before the interrupt handler terminates.

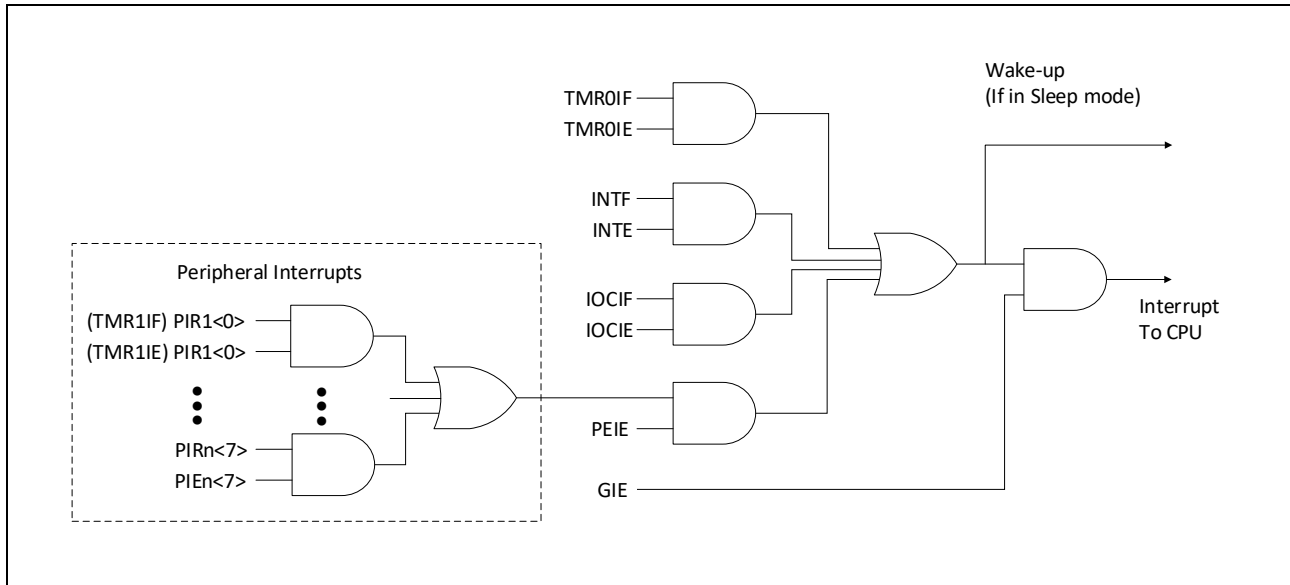


Figure 3-2 PIC16(L)F183xx Peripheral Interrupt Hierarchy

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### 3.7 Debugging and Programming Comparison

RA0 and PIC16(L)F1 use different interfaces and functions when connecting to the device for debugging or programming. Table 3-14 shows a comparison of debugging and programming in PIC16(L)F1 and RA0.

**Table 3-14 Debugging and Programming Comparison**

Programming Function	PIC16(L)F183xx	RA0
Type	Serial programming	Serial programming
Security	Code Protection Write Protection	Memory security and access restriction options <sup>Note</sup>
Entry to the programming mode	High-Voltage Programming Entry Mode Low-Voltage Programming Entry Mode	After reset is released, set DFLCTL.DFLEN to 1 and FENTRYR to code flash P/E mode.
Command	Configuration, address setting, erase write, read	Erase, programming, get info, security, etc. <sup>Note</sup>
Interface	ICSP, LVDS	SWD

**Note** By default, block erase, write commands, and boot cluster rewriting are disabled. Enabling block erase, enabling writes, and enabling boot cluster rewriting can be performed by manipulating the registers according to the correct procedure. Refer to the appropriate chapter in the each RA0 Serise User's Manual: Hardware.

#### 3.7.1 Debugging Comparison

##### 3.7.1.1 Debugging RA0

For debugging RA0, use J-Link, E2 Emulator, or E2 Emulator Lite. These debuggers support standard debugging functions (step execution, breakpoints, memory and register inspection, etc.). For programming, use J-Link, E2 Emulator, E2 Emulator Lite, or the dedicated flash memory programmer PG-FP6.

Additionally, it supports on-chip debugging, enabling debugging via the SWD (Serial Wire Debug) serial interface.

##### 3.7.1.2 Debugging PIC16(L)F183xx

For debugging the PIC16(L)F183xx, use the MPLAB PICkit 5, MPLAB ICD 5, or MPLAB ICE 4. Debugging and programming the PIC16(L)F183xx is possible through the graphical user interface of the MPLAB IDE. These debugger systems connect to a PC via a USB interface and connect to the target using a connector compatible with the debugger system. The MPLAB ICE 4 features full-speed emulation, runtime variable watch, trace analysis, and complex breakpoints.

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### 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 Serial Programming PIC16(L)F183xx

Non-volatile memory (NVM) programming data is supplied via the in-circuit serial programming (ICSP) interface and can be programmed into flash memory or EEPROM.

ICSP programming uses the following five pins: ICSPCLK

- ICSPDAT
- MCLR/VPP
- VDD
- VSS

In program/verify mode, the program memory, EEPROM, user ID, and configuration words are programmed via serial communication. The ICSPDAT pin is a bidirectional I/O used for serial data transfer, and the ICSPCLK pin is the clock input.

Connection to the target device is typically made via an ICSP header. A connector commonly found on development tools is the RJ-11 in a 6P6C (6-pin, 6-connector) configuration.

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### 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.

Additionally, the RA0 incorporates a pin function selection feature that allows switching the port assigned to a multifunction pin. The peripheral function is selected by changing the setting value of the peripheral function select bit (PSEL) in the pin function selection register (PmnPFS\_A) for pins that the RA0 has pre-mapped to multifunction pins.

The PIC16(L)F1 provides a PPS (Peripheral Pin Select) function. This allows both one-time configuration of port assignments and dynamic pin assignment changes during code execution.

**Table 4-1 I/O Port Function Comparison**

Item	PIC16(L)F183xx	RA0
Output mode	CMOS Open drain with N-ch	CMOS Open drain with N-ch
Input mode	CMOS or TTL Internal pull-up	CMOS On-chip pull-up TTL input buffer
Select slew rate	Set by SLRCONx register	N/A
Setting per port	Available	Available
Redirect function	PPS function	Uses PmnPFS_A register
Wake-up	Can be used in combination with external interrupts (INT pin or IOC pin)	Can be used with external interrupt (IRQ)
Input filtering	N/A	Available (signals within 1µs are filtered when using external interrupt (IRQx))
User-controllable input hysteresis	Control by INLVLx register (CMOS or TTL)	Available

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

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### 4.2 Universal Asynchronous Receiver-Transmitter (UART, SAU)

Both RA0 and PIC16(L)F1 are equipped with a peripheral function that enables asynchronous serial communication.

RA0 is equipped with UARTA, a specialized UART communication function. Additionally, the SAU (Serial Array Unit) incorporates UART functionality, enabling asynchronous communication via two lines: serial data transmission (TXD) and serial data reception (RXD).

Using these two communication lines, data is transmitted and received asynchronously with the communication partner (using an internal baud rate) for each data frame consisting of a start bit, data, parity bit, and stop bit. Using two channels—one dedicated to transmission (even-numbered channel) and one dedicated to reception (odd-numbered channel)—enables full duplex UART communication. Furthermore, combining the timer array unit with an external interrupt (IRQ0) allows support for the LIN bus.

**Table 4-2 UART Function Comparison**

Function	PIC16(L)F183xx	RA0
Data direction	LSB first	MSB first or LSB first
Data phase	Available	Available
Single-wire half-duplex communication	Available (Synchronous)	Available
Data length	8, 9 bits	5, 7, 8, 9 bits <sup>Note</sup>
LIN HW support	Available	Available
Wakeup from Stmd	Available	Available
Auto baud rate detection	Available	Available (LIN)
Tx/Rx FIFO Depth	1	1

Note 5-bit data length can be used with UARTAx and 9-bit data length can be used with UARTx.

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

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### 4.3 Serial Peripheral Interface (SPI, SAU)

PIC16(L)F1 and RA0 both support serial peripheral interface (SPI).

The RA0 features a simplified SPI function within the SAU (Serial Array Unit), enabling data transmission and reception synchronized to the serial clock (SCK) output from the master. This clock-synchronous communication function utilizes a total of three communication lines: one serial clock (SCK) line and two serial data lines (SO, SI) for transmission and reception.

**Table 4-3 SPI Function Comparison**

Function	PIC16(L)F183xx	RA0
Control pins	SCK, SDO, SDI, SS	SCK, SI, SO
Master/Slave selection	Available	Available
Data bit width	8 bits	7, 8 bits
Maximum speed	[Master] 8MHz ( $f_{CLK}/4$ ) [Slave] 4MHz ( $f_{OSC}/8$ )	[Master] SPI00 only : 16MHz (PCLKB/2) other than SPI00 : 8MHz (PCLKB/4) [Slave] 5.33MHz ( $f_{MCK}/6$ )
Simplex communication	Available	Available
Hardware chip select management	Available	N/A
I/O clock phase control	Available	Available
Data direction	MSB first	MSB first or LSB first
SPI format support	Motorola	Motorola
Tx/Rx FIFO Depth	1	1

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

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### 4.4 Inter-Integrated Circuit (IICA, SAU)

Both PIC16(L)F1 and RA0 support the I<sup>2</sup>C peripheral function.

The RA0 is equipped with a Serial Array Unit (SAU) featuring simplified I<sup>2</sup>C functionality and an IICA with more advanced capabilities. While it enables clock-synchronous communication with multiple devices via two lines—Serial Clock (SCL) and Serial Data (SDA)—the simplified I<sup>2</sup>C is designed for single-device communication with components like flash memory and A/D converters, functioning solely as a master.

**Table 4-4 I<sup>2</sup>C Function Comparison**

Function	PIC16(L)F183xx	RA0
Master/Slave selection	Available	Available (IICA)
Multi-controller function	Available	Available (IICA)
Maximum transfer speed	400kHz	1MHz
Addressing mode	7, 10 bits	7 bits 10 bits (IICA)
Clock stretch	Available	Available (IICA)
Wakeup function (low-power mode)	Available	Available (IICA)
Software reset	Available	Available (IICA)
FIFO/Buffer	1 byte SSPxSR	1 byte IICAn (IICA) SDRmn (SAU)
Programmable analog and digital noise filters	N/A	Available (IICA)

For information on I<sup>2</sup>C code examples, please download the sample project "FPB-RA0xx Example Project Bundle" from the Renesas Electronics website.

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

#### 4.5 Timer (TAU,TML32)

PIC16(L)F1 and RA0 are both equipped with multi-function timers.

The RA0 is equipped with a timer array unit (TAU) consisting of eight 16-bit timers. Each timer can be used independently, and multiple timers can be combined to provide advanced timer functions.

**Table 4-5 Timer Names**

PIC16(L)F183xx		RA0	
Timer Name	Abbr.	Timer Name	Abbr.
Timer0 Module	Timer0	Timer array unit	TAU
Timer1/3/5 Module with Gate Control	Timer1/3/5	32-bit interval timer	TML32
Timer2/4/6 Module	Timer2/4/6		

**Table 4-6 Timer Function Comparison**

Function	PIC16(L)F183xx	RA0
Resolution	8, 16 bits	TAU: 8, 16 bits TML32: 8, 16, 32 bits
PWM	Available <sup>Note</sup>	Available (TAU)
Capture	Available <sup>Note</sup>	Available
Compare	Available <sup>Note</sup>	Available
One-shot	Available <sup>Note</sup>	Available (TAU)
Up/down count function	N/A	N/A
Conjunctive operation with standby function	Available	Available
Programmable prescaler clock frequency	Available	Available
Event/interrupt	Available	Available
Auto reload function	Available	Available

Note : Operates in conjunction with the CCP module

**Table 4-7 Timer Use Case Comparison**

Function	PIC16(L)F183xx	RA0
PWM	Timer2/ 4/ 6 <sup>Note</sup>	TAU
Capture	Timer1/ 3/ 5 <sup>Note</sup>	TAU
Compare	Timer1/ 3/ 5 <sup>Note</sup>	TAU
One-shot	Timer1/ 3/ 5 <sup>Note</sup>	TAU
Synchronization	-	TAU
Interval timer	Timer0 Timer1/ 3/ 5 <sup>Note</sup>	TAU TML32
Prescaler	4-bit Prescaler	4-bit Prescaler (TAU) 3-bit Prescaler (TML32)

Note : Operates in conjunction with the CCP module

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

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### 4.6 Independent Watchdog Timer (IWDT)

PIC16(L)F1 and RA0 are both equipped with watch dog timers.

The RA0 Independent Watchdog Timer (IWDT) features the ability to reset the MCU and generate non-maskable interrupts or underflow interrupts. Since this timer operates with LOCO, it is particularly useful as a fail-safe mechanism during system runaway, restoring the MCU to a known state. The IWDT can be automatically triggered by a register reset, underflow, refresh error, or count value refresh.

**Table 4-8 WDT Name**

PIC16(L)F183xx		RA0	
Timer Name	Abbr.	Timer Name	Abbr.
watchdog timer	WDT	Independent watchdog timer	IWDT

**Table 4-9 Function Comparison**

Function	PIC16(L)F183xx	RA0
Window mode	N/A	Available
Interval timer mode	Available	Available
Clock source	LFINTOSC	LOCO/2
Interrupt	N/A	Available
Counter resolution	23 bits	14 bits
Clock divider	N/A	N/A

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

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

## 5. Analog Peripheral Comparison

### 5.1 Analog-Digital Converter (ADC12)

Both PIC16(L)F1 and RA0 are equipped with a ADC that converts analog signals to digital signals.

The RA0 is configured to control up to 10 analog input channels. It also allows selection of the internal reference voltage and the temperature sensor output voltage. The A/D converter resolution can be selected from 12-bit, 10-bit, or 8-bit resolution.

**Table 5-1 ADC Function Comparison**

Function	PIC16(L)F183xx	RA0
Resolution (bits)	10 bits	12, 10, 8 bits
Conversion rate (Msps)	16	0.485
FIFO	N/A	N/A
ADC reference (V)	internal : $V_{DD}$ , FVR <sup>Note 2</sup>	internal : $V_{CC}$
	external : VREF+	external : VREFH0
Operation mode	Normal operation, DOZE, IDLE, SLEEP	Normal operation, SNOOZE
Auto power down	N/A	Available
External input channel <sup>Note 1</sup>	Max 17	Max 10
Internal input channels	Temperature sensor, DAC output voltage, FVR <sup>Note 2</sup>	Temperature sensor, internal reference voltage
Number of ADC	1	1

Note 1. The number of external input channels differs depending on the device.

Note 2. Fixed Voltage Reference

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 ( $V_{BGR}$ )

Both the RA0 and PIC16(L)F1 feature an internal reference voltage that can supply reference voltage to internal peripherals.

**Table 5-2 BGR Characteristic Comparison**

Function	PIC16(L)F183xx	RA0
Internal reference voltage (V)	1.024, 2.048, 4.096	1.48
Output internal reference	N/A	N/A
Internal connection to ADC	Available	Available

**6. Reference**

- [1] Microchip, PIC16F18056/76 datasheet, DS40002325C
- [2] Microchip, PIC16(F)F18326/18346 datasheet, DS40001839E
- [3] Microchip, PIC16(L)F18426/46 datasheet, DS40001985C

## RA0 Series

### Migration Guide from Microchip PIC®16(L)F1 to RA0

---

#### Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2025.11.30	-	First edition issued
1.10	2026.2.24	p.1, p.4 p.39, p.41	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.
7. 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](http://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/](http://www.renesas.com/contact/).