RL78 Family

FFT Library: Deployment Guide

Introduction

This document provides information for deploying FFT Library. Fast Fourier transform (FFT) is an algorithm that executes the discrete Fourier transform at high speed. The implementation developed in 1965 by James Cooley and John Tukey, now widely known as FFT, has contributed to the rapid advancement of digital signal processing applications.

The FFT library is provided in a version that has been tuned at the assembly language level to enable efficient processing on Renesas MCUs.

Target Devices

RL78/G13, RL78/G14, RL78/G23, RL78/G15, RL78/G24

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

Contents

1. Product Configuration

The product comprises the following items.

1. FFT Library V. 1.03 Release 00 and FFT Library for RL78/G24 FAA V.1.01
2. Libraries above: Deployment Guide (r20an0150ej0109_rl78_fft.pdf)
   Product No.: R0M7800LF0010RRC

The product comprises the files listed in Table 1-1.

Table 1-1 FFT Library Configuration

	Description
r20an0150ej0109_rl78_fft.pdf	Deployment guide (this document)
Workspace (workspace)	
Documents (doc)	
English (en)	
r20uw0099ej0102_fft.pdf	User's Manual
r20an0150ej0109_rl78_fft.pdf	Deployment guide (this document)
Japanese (ja)	
r20uw0099jj0102_fft.pdf	User's Manual
r20an0150jj0109_rl78_fft.pdf	Deployment guide
CS+, e^{2}studio for CC (CS+, e2studio for CC)	
FFT library (sample/<sample program folder>lib)	
libfft_rl78g13.lib	FFT library for RL78/G13 (assembler version), version 1.01
libfft_rl78g14.lib	FFT library for RL78/G14, RL78/G23 and RL78/G24 256 points (assembler version), version 1.01
libfft_rl78_S2_NOMDA.lib	FFT library for RL78/G15 (assembler version), version 1.03
r_fft_int16.h	FFT library header file
r_stdint.h	Type definition header file
Sample program (sample)	
rl78g14_fft_ccrl_CS+	Sample CS+ for CC project (RL78/G14 and RL78/G23)
rl78g14_fft_ccrl_e2studio	Sample e^{2}studio for CC project (RL78/G14 and RL78/G23)
rl78g15_fft_ccrl_CS+	Sample CS+ for CC project (RL78/G15)
rl78g14_fft_ccrl_e2studio	Sample e^{2}studio for CC project (RL78/G15)
CS+, e^{2}studio for CC (CS+, e2studio for CC (RL78G24 FAA))	
FFT library (sample/<sample program folder>lib)	
libfft_rl78g14.lib	FFT library for RL78/G14, RL78/G23 and RL78/G24 256 points (assembler version), version 1.01
r_fft_int16.h	FFT library header file
FFT library for FAA (sample/<sample program folder>smc_gen)	
Config_FAA	FFT library for FAA, version 1.00(For 64 and 128 points)
Sample program (sample)	
rl78g24_fft_ccrl_CS+	Sample CS+ for CC project (RL78/G24)
rl78g24_fft_ccrl_e2studio	Sample e^{2}studio for CC project (RL78/G24)
IAR Embedded Workbench (IAR)	
FFT library (sample/<sample program folder>lib)	
libfft_rl78g14.a	FFT library for RL78/G14 and RL78/G23 (assembler version), version 1.01
libfft_rl78_S2_NOMDA.a	FFT library for RL78/G15 (assembler version), version 1.03
r_fft_int16.h	FFT library header file
r_stdint.h	Type definition header file
Sample program (sample)	
rl78g14_fft_iar	Sample IAR Embedded Workbench project (RL78/G14 and RL78/G23)
rl78g15_fft_iar	Sample IAR Embedded Workbench project (RL78/G15)
IAR Embedded Workbench (IAR (RL78G24-FAA))	
FFT library (sample/<sample program folder>lib)	
libfft_rl78g14.a	FFT library for RL78/G14, RL78/G23 and RL78/G24 256 points (assembler version), version 1.01
r_fft_int16.h	FFT library header file
FFT library for FAA (sample/<sample program folder>smc_gen)	
rl78g24_fft_iar	Sample IAR Embedded Workbench project (RL78/G24)

			Description
e²studio for LLVM (e2studio for LLVM)			
	FFT library (sample\<sample-program-folder>\lib)		
		libfft_rl78.a	FFT library for RL78/G23 (assembler version) version 1.01
		libfft_rl78_S2_NOMDA.a	FFT library for RL78/G15 (assembler version) version 1.03
		r_fft_int16.h	FFT library header file
		r_stdint.h	Type definition header file
	Sample program (sample)		
		rl78g14_fft_llvm_e2tudio	Sample e²studio for LLVM project (RL78/G23)
		rl78g15_fft_llvm_e2tudio	Sample e²studio for LLVM project (RL78/G15)

2. Library Functions

The FFT library supports the following library functions (APIs).

Table 2-1 FFT Library Functions

API	Description
R_rfft64_int16	16-bit fixed-point real-number FFT (64 points)
R_rfft128_int16	16-bit fixed-point real-number FFT (128 points)
R_rfft256_int16	16-bit fixed-point real-number FFT (256 points)

3. CS+, e²studio for CC

3.1 Limitations

The FFT library for the RL78/G13 uses the MCU's on-chip multiplier and divider/multiply-accumulator for multiply-accumulate operations. Therefore, it is necessary to ensure that the register values listed below are not changed within interrupt handlers implemented by the user. For information on the multiplier and divider/multiply-accumulator and related registers, refer to chapter 14, Multiplier and Divider/Multiply-Accumulator, in RL78/G13 User's Manual: Hardware.

Registers

- Multiplication/division data register A (L) (MDAL)
- Multiplication/division data register A (H) (MDAH)
- Multiplication/division data register B (L) (MDBL)
- Multiplication/division data register B (H) (MDBH)
- Multiplication/division data register C (L) (MDCL)
- Multiplication/division data register C (H) (MDCH)

Control register

- Multiplication/division control register (MDUC)

3.2 Compiler Options

The library files are generated using the following compile options.

[Compile options]

FFT library for RL78/G13, RL78/G14, RL78/G23 and RL78/G24 256 points:

```
-asmopt=-mirror_source=common -memory_model=medium
```

FFT library for RL78/G15:

```
-asmopt=-mirror_source=0 -memory_model=medium
-cpu=S2 -Odefault
```

3.3 Development Environment

The Renesas development environment consists of the items listed below.

Make sure to use the latest versions available when developing user applications.

[Software tools]

FFT library for RL78/G13, RL78/G14 and RL78/G23 and RL78/G24 256 points:

Integrated development environment
CS+ for CC V8.06.00
C compiler
CC-RL V1.10
Debugger
RL78 simulator

FFT library for RL78/G15:

Integrated development environment
CS+ for CC V8.08.00
e²studio Version: 2022-10(22.10.0)
C compiler
CC-RL V1.11.00
Debugger
E2 Lite emulator

3.4 ROM, RAM, and Stack Sizes

The ROM, RAM, and stack sizes of the various FFT library APIs are listed below (unit: bytes).

Table 3-1 ROM/ RAM/ and Stack Sizes (CS+, e²studio for CC (for RL78/G13))

API	ROM	RAM	Stack
R_rfft64_int16	1,260	0	68
R_rfft128_int16	1,512	0	68
R_rfft256_int16	2,018	0	68

Table 3-2 ROM/ RAM/ and Stack Sizes (CS+, e²studio for CC (for RL78/G14, RL78/G23 and RL78/G24 256 points))

API	ROM	RAM	Stack
R_rfft64_int16	1,224	0	68
R_rfft128_int16	1,476	0	68
R_rfft256_int16	1,982	0	68

Table 3-3 ROM/ RAM/ and Stack Sizes (CS+, e²studio for CC (RL78/G15))

API	ROM	RAM	Stack
R_rfft64_int16	1,358	0	82
R_rfft128_int16	1,610	0	82

3.5 Section Information

The sections (segments) used by the various FFT library APIs are listed in the table below.

Table 3-4 Section Information (CS+, e²studio for CC (for RL78/G13, RL78/G14, RL78/G23 and RL78/G24 256 points))

Section Name	Description	Section Attribute
.textf	Program	.CSEG TEXTF
.const	Constant data	.CSEG CONST

Table 3-5 Section Information (CS+, e²studio for CC (for RL78/G15))

Section Name	Description	Section Attribute
.text	Program	SECTION=.text
.const	Constant data	SECTION=.const

3.6 Library Performance

The processing times when calling the various library functions (APIs) are listed below.

Table 3-6 Processing times (CS+, e²studio for CC (for RL78/G13))

API	Time (System Clock = 32 MHz)
R_rfft64_int16	Approx. 0.4 ms
R_rfft128_int16	Approx. 0.9 ms
R_rfft256_int16	Approx. 1.9 ms

Table 3-7 Processing times (CS+, e²studio for CC (for RL78/G14, RL78/G23 and RL78/G24 256 points))

API	Time (System Clock = 32 MHz)
R_rfft64_int16	Approx. 0.3 ms
R_rfft128_int16	Approx. 0.7 ms
R_rfft256_int16	Approx. 1.6 ms

Note: Measured using the execution time measurement function of the integrated development environment (CS+)

Table 3-8 Processing times (CS+, e²studio for CC (for RL78/G15))

API	Time (System Clock = 16 MHz)
R_rfft64_int16	Approx. 29.3 ms
R_rfft128_int16	Approx. 73.1 ms

Note: Measured using the execution time measurement function of the integrated development environment (e²studio)

3.7 Version Information

The version information for the library is stored as a character string in the r_fft_a_version variable. This variable can be accessed by means of the following extern declaration.

```
extern const char r_fft_a_version[];
```

The data stored in the libraries comprising the current product is shown below.

FFT library for RL78/G13:

```
const char r_fft_a_version[] =
“FFT Library version 1.01 for RL78 Family (RL78G13) (Dec  7 2015, 17:30:04)”;
```

FFT library for RL78/G14, RL78/G23, and RL78/G24 256 points:

```
const char r_fft_a_version[] =
“FFT Library version 1.01 for RL78 Family (RL78G14) (Dec  7 2015, 17:29:42)”;
```

FFT library for RL78/G15:

```
const char r_fft_a_version[] =
“FFT Library version 1.03 for RL78 Family”;
```

4. IAR Embedded Workbench

4.1 Compiler Options

The library files are generated using the following compile options.

[Compile options]

FFT library for RL78/G14 and RL78/G23:

```
__FAR_MODEL__  __NEAR_DATA_MODEL__
NDEBUG  __RL78__  __TARGET__=RL78G14
```

FFT library for RL78/G15:

```
__NEAR_MODEL__  __NEAR_DATA_MODEL__
NDEBUG  __RL78__  __TARGET__=RL78G15
```

4.2 Development Environment

The Renesas development environment consists of the items listed below.

Make sure to use the latest versions available when developing user applications.

[Software tools]

Integrated development environment
    IAR Embedded Workbench for Renesas RL78 4.21.1
C compiler
    IAR C/C++ Compiler for Renesas RL78 4.21.1.2409
Debugger
    IAR C-SPY Debugger Kernel 8.5.2.7561

4.3 ROM, RAM, and Stack Sizes

The ROM, RAM, and stack sizes of the various FFT library APIs are listed below (unit: bytes).

Table 4-1 ROM, RAM, and Stack Sizes (IAR (for RL78/G14, RL78/G23 and RL78/G24 256 points))

API	ROM	RAM	Stack
R_rfft64_int16	1,226	0	68
R_rfft128_int16	1,478	0	68
R_rfft256_int16	1,984	0	68

Table 4-2 ROM, RAM, and Stack Sizes (IAR (for RL78/G15))

API	ROM	RAM	Stack
R_rfft64_int16	1,350	0	82
R_rfft128_int16	1,602	0	82

4.4 Section Information

The sections (segments) used by the various FFT library APIs are listed in the table below.

Table 4-3 Section Information (IAR (for RL78/G14, RL78/G23 and RL78/G24 256 points))

Section Name	Description
.textf	Program
.const	Constant data

4.5 Library Performance

The processing times when calling the various library functions (APIs) are listed below.

Table 4-4 Processing times (IAR (for RL78/G14, RL78/G23 and RL78/G24 256 points))

API	Time (System Clock = 32 MHz)
R_rfft64_int16	Approx. 0.2 ms
R_rfft128_int16	Approx. 0.6 ms
R_rfft256_int16	Approx. 1.5 ms

Note: Measured using the execution time measurement function of the integrated development environment (IAR Embedded Workbench for Renesas RL78))

Table 4-5 Processing times (IAR (for RL78/G15))

API	Time (System Clock = 16 MHz)
R_rfft64_int16	Approx. 18.3 ms
R_rfft128_int16	Approx. 44.4 ms

Note: Measured using the execution time measurement function of the integrated development environment(IAR Embedded Workbench for Renesas RL78))

4.6 Version Information

The version information for the library is stored as a character string in the r_fft_a_version variable. This variable can be accessed by means of the following extern declaration.

```
extern const char r_fft_a_version[];
```

The data stored in the libraries comprising the current product is shown below.

FFT library for RL78/G14 and RL78/G23:

```
const char r_fft_a_version[] =
“FFT Library version 1.01 for RL78 Family (RL78G14) (Sep  7 2021, 13:40:39)”;
```

FFT library for RL78/G15:

```
const char r_fft_a_version[] =
“FFT Library version 1.03 for RL78 Family”;
```

5. e²studio for LLVM

5.1 Compiler Options

The library files were generated by using the following compile options.

[Compile options]

```
FFT library for RL78/G23: CPU Type: S3-core
Optimization: None(-O0)
```

FFT library for RL78/G15:

```
CPU Type: S2-core
Optimization: None(-O0)
```

5.2 Development Environment

The Renesas development environment consists of the items listed below.

Make sure to use the latest versions available when developing user applications.

[Software tools]

FFT library for RL78/G23:

Integrated development environment
    e²studio (version 2022-04 (22.4.0))
C compiler
    LLVM V10.0.0.202203
Debugger
    E2 Lite emulator

FFT library for RL78/G15:

Integrated development environment
    e²studio (version 2022-10 (22.10.0))
C compiler
    LLVM V10.0.0.202207
Debugger
    E2 Lite emulator

5.3 ROM, RAM, and Stack Sizes

The ROM, RAM, and stack sizes of the various FFT APIs are listed below (unit: bytes).

Table 5-1 ROM, RAM, and Stack Sizes (e²studio for LLVM (for RL78/G23))

API	ROM	RAM	Stack
R_rfft64_int16	1224	0	68
R_rfft128_int16	1476	0	68
R_rfft256_int16	1982	0	68

Table 5-2 ROM, RAM, and Stack Sizes (e²studio for LLVM (for RL78/G15))

API	ROM	RAM	Stack
R_rfft64_int16	1346	0	82
R_rfft128_int16	1854	0	82

5.4 Section Information

The sections (segments) used by the various FFT library APIs are listed in the table below.

Table 5-3 Section Information (e²studio for LLVM)

Section Name	Description
.text	Program
.rodata	Constant data

5.5 Library Performance

The processing times when calling the various library functions (APIs) are listed below.

Table 5-4 Processing times (e²tusio for LLVM (for RL78/G23))

API	Time (System Clock = 32 MHz)
R_rfft64_int16	Approx. 0.3 ms
R_rfft128_int16	Approx. 0.7 ms
R_rfft256_int16	Approx. 1.6 ms

Table 5-5 Processing times (e²studio for LLVM (for RL78/G15))

API	Time (System Clock = 16 MHz)
R_rfft64_int16	Approx. 29.2 ms
R_rfft128_int16	Approx. 73.2 ms

Note: Measured using the execution time measurement function of the integrated development environment (e²studio)

5.6 Version Information

The version information for the library is stored as a character string in the r_fft_a_version variable. This variable can be accessed by means of the following extern declaration.

```
extern const char r_fft_a_version[];
```

The data stored in the libraries comprising the current product is shown below.

FFT library for RL78/G23:

```
const char r_fft_a_version[] =
“FFT Library version 1.01 for RL78 Family”;
```

FFT library for RL78/G15:

```
const char r_fft_a_version[] =
“FFT Library version 1.03 for RL78 Family”;
```

6. RL78/G24 FAA

6.1 CS+, e²studio for CC

6.1.1 Development Environment

The Renesas development environment consists of the items listed below.

Make sure to use the latest versions available when developing user applications.

[Software tools]

FFT library for RL78/G24 FAA:

Integrated development environment
    CS+ for CC V8.09.00
    e²studio Version: 2023-07(23.7.0)
C compiler
    CC-RL V1.20
Debugger
    E2 Lite emulator

6.1.2 FFT library for RL78/G24 FAA

If you want to use FFT 64 points or 128 points API, generate the FFT library for RL78/G24 FAA by using smart configurator.

For basic operation of the Smart Configurator, refer to the User's Guide for basic Smart Configurator operations.

- RL78 Smart Configurator User's Guide: e² studio (R20AN0579)
- RL78 Smart Configurator User's Guide: CS+ (R20AN0580)

For the build process and debugger of the flexible application accelerator (FAA) contained in RL78/G24, please refer to the following user guides:

- RL78/G24 FAA Tool Guide for CS+ (R01AN7094)
- RL78/G24 FAA Tool Guide for e² studio (R01AN70945)

This library does not support 256 points. So link and use “libfft_rl78g14.lib” (RL78/G14, RL78/G23, RL78/G24 for FFT 256 points) as described in section **3 CS+, e2studio for CC** when using FFT 256 points API.

6.1.2.1 How to generate the FFT library for FAA

The code of the FFT library for RL78/G24 FAA is generated by smart configurator.

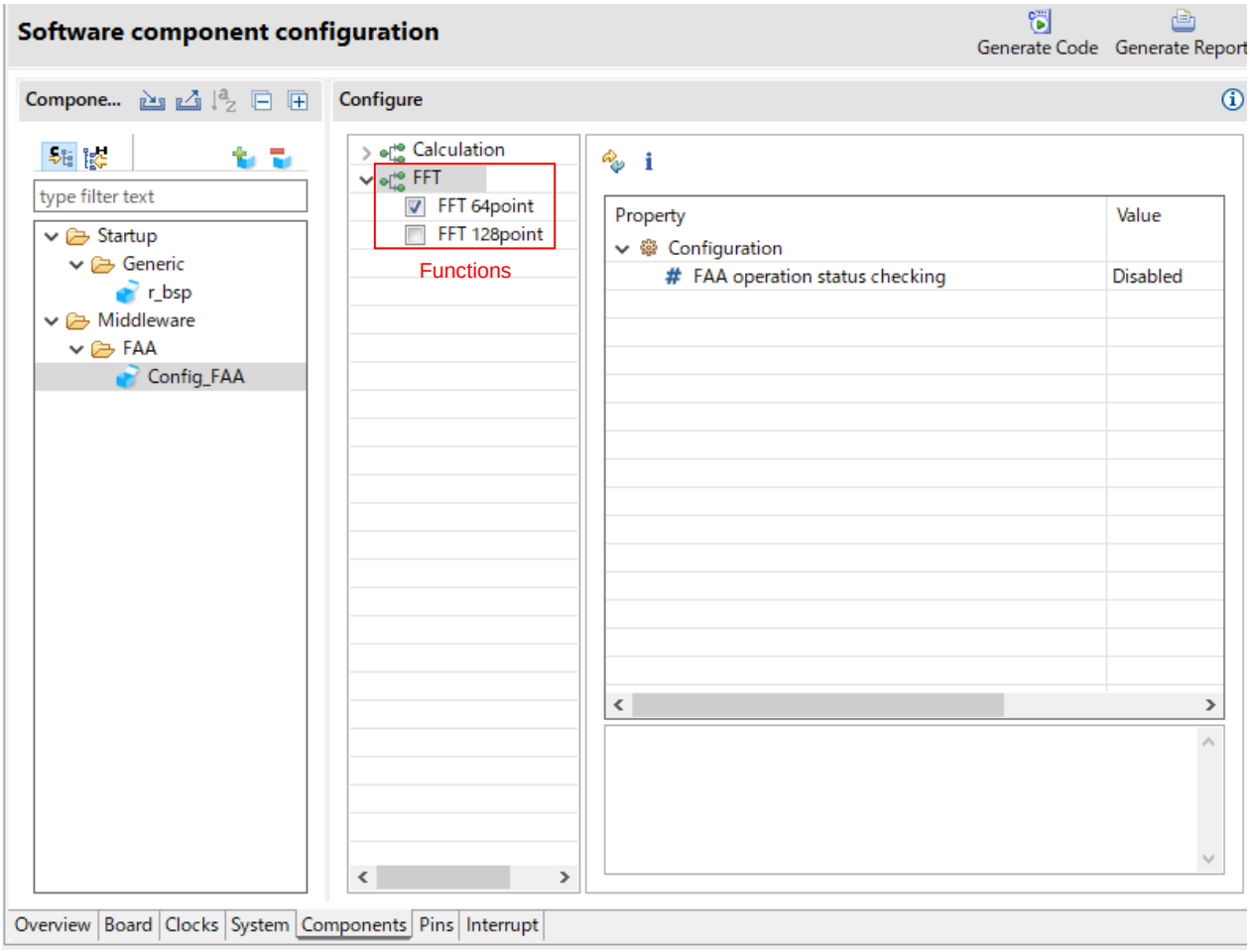

Figure 6-1 Configuration of the FAA module

1. Select FFT points to be used in the Functions of smart configurator.

Note: You can select more than one but select only one FFT points.

2. After selecting a function, set its property.
3. The code is generated in /src/smc_gen/Config_FAA

Selectable Functions are shown below.

Table 6-1 Properties for configuring the FAA operation status check

Property	Description
FAA operation status checking	・Enabled : The operating state of FAA is checked when calling the API, and return an error code if the FAA is in operation by another function. (R_DSCL_ERR_FAA_ALREADY_RUNNING) ・Disabled : The operating state of FAA is not checked.

6.1.2.2 Preprocessor Macro Definition

When using the FFT library for RL78/G24 FAA, add the following definition to Preprocessor Macro Definition of the project.
The definition to be added is “R_FFT_FAA”

6.1.2.3 Work area to be specified in API

When using the FFT library for RL78/G24 FAA, specify the work area allocated on the FAA for the work area given as the fourth argument of the API.
The work area name to be specified is “V_rfft_work” described in “r_fft_int16.h”.

6.1.2.4 Return Values of API

FFT library for RL78/G24 FAA adds new return values to the API.

The return values are listed below.

Table 6-2 Return Values

Return Value	Description
R_FFT_STATUS_OK	Normal termination.
R_FFT_ERR_INPUT_NULL	The input parameter “input” is NULL.
R_FFT_ERR_OUTPUT_NULL	The output parameter “output” is NULL.
R_FFT_ERR_WINDOW_NULL	The input parameter “window” is NULL.
R_FFT_ERR_WORK_NULL	The input parameter “work” is NULL.
R_FFT_ERR_FAA_ALREADY_RUNNING	FAA is running.

6.1.3 ROM, RAM, and Stack Sizes

The ROM, RAM, and stack sizes of the various FFT library APIs are listed below (unit: bytes)

Table 6-3 ROM / RAM and Stack Size (for CS+, e²studio for CC (RL78/G24 FAA))

API	ROM	RAM	stack	FAACODE	FAADATA	FAAStack
R_rfft64_int16	575	0	36	688	1432	8
R_rfft128_int16	704	0	36	688	1808	8

6.1.4 Section Information

The sections (segments) used by the various FFT library APIs are listed in the table below.

Table 6-4 Section Information (for CS+, e²studio for CC (RL78/G24 FAA))

Section Name	Description
.textf	Program
.const	Constant data
FAACODE	FAA code area
FAADATA	FAA data area

6.1.5 Library Performance

The processing times when calling the various library functions (APIs) are listed below.

Table 6-5 Processing times (for CS+, e²studio for CC (RL78/G24 FAA))

API	Time (System Clock = 48 MHz)
R_rfft64_int16	Approx.0.2ms
R_rfft128_int16	Approx.0.5ms

Note: Measured using the execution time measurement function of the integrated development environment (e²studio)

6.1.6 Version Information

This library does not support the r_fft_a_version variable to indicate version information. Refer to the header information in the source.

6.2 IAR Embedded Workbench

6.2.1 Development Environment

The Renesas development environment consists of the items listed below.

Make sure to use the latest versions available when developing user applications.

[Software tools]

- Integrated development environment
  IAR Embedded Workbench for Renesas RL78 V5.10.3
- C compiler
  IAR C/C++ Compiler for Renesas RL78 5.10.3.2716 (5.10.3.2716)

Smart Configurator
  Renesas Smart Configurator for RL78 V1.12.0

- Debugger
  E2 Lite emulator

The library files are generated using the following compile options.

[Compile options]

FFT library for RL78/G24:

```
__core         = s3
__code_model = far
__data_model = near
```

6.2.2 FFT library for RL78/G24 FAA

If you want to use FFT 64 points or 128 points API, generate the FFT library for RL78/G24 FAA by using smart configurator.

For basic operation of the Smart Configurator, refer to the User's Guide for basic Smart Configurator operations.

- RL78 Smart Configurator User's Guide: IAR (R20AN0581)

This library does not support 256 points. Therefore, link and use “libfft_rl78g14.a” (for RL78/G14, RL78/G23, RL78/G24 for FFT 256 points) when using the FFT 256 points API, as described in section **4 IAR Embedded Workbench** when using FFT 256 points API.

6.2.2.1 How to generate the FFT library for FAA

1. Starting the Smart Configurator
   Select [Smart Configurator for RL78 Vx.x.x] of [Renesas Electronics Smart Configurator] from the Windows start menu. The main window of the Smart Configurator will be starting.

Note: Please replace Vx.x.x with the user's version.

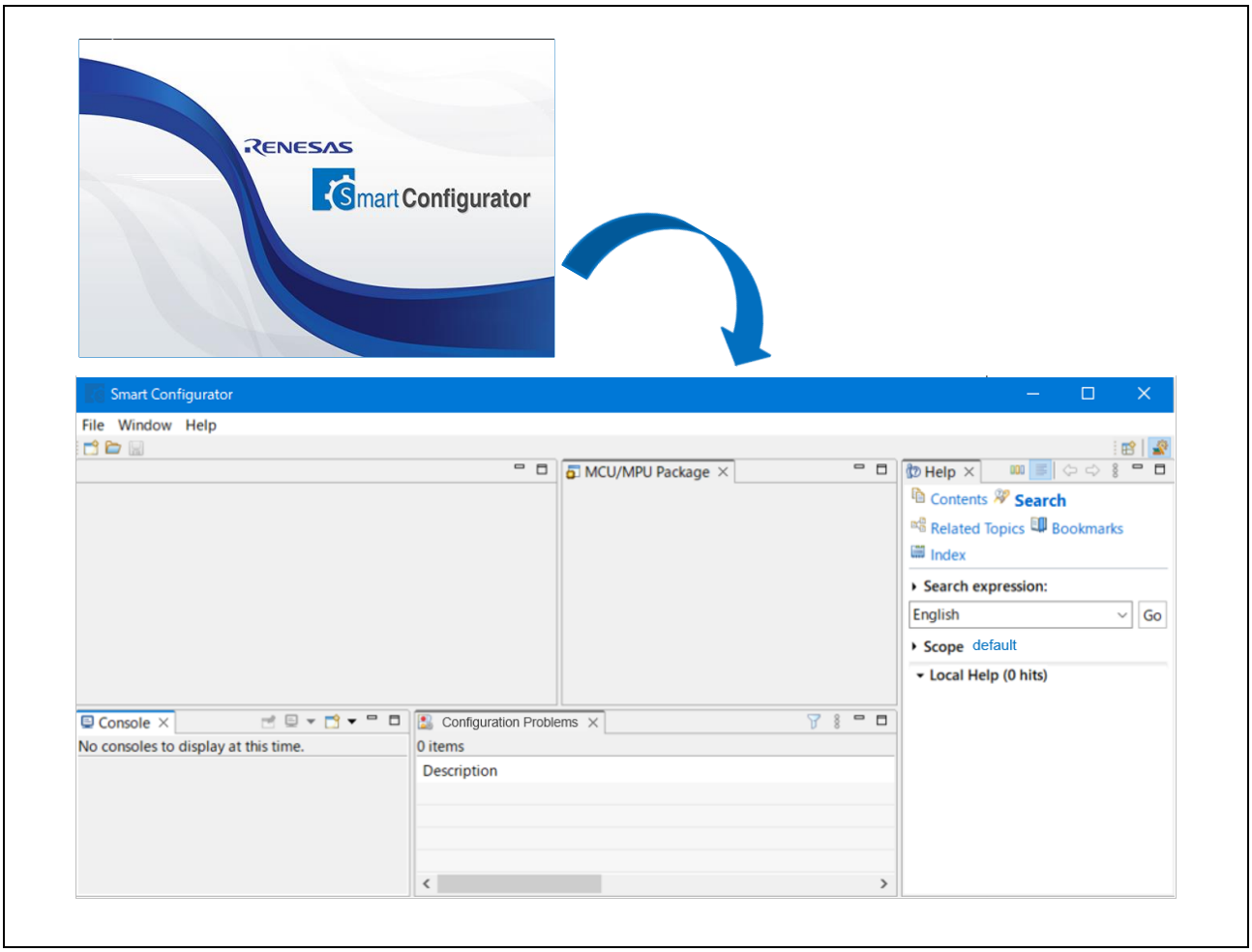

Figure 6-2 Starting of Smart Configurator

2. Creating a New Smart Configurator Configuration File
   On the main window, click the [New Configuration File] button to display the [New Smart Configuration File] dialog box.
   (1). In [Platform:], panel, select the device.
   (2). In [Toolchain:], select [IAR RL78 Toolchain].
   (3). In [File name:], enter the file name.
   (4). Confirm [Location:], To change the location, please click [Browse] and select the save destination.

Note: The *.eww, *.ewp, *.ewd, main.c and buildinfo.ipcf files will be generated to this location after clicking "Generate Code" button.

   (5). Click [Finish] to create the configuration file.

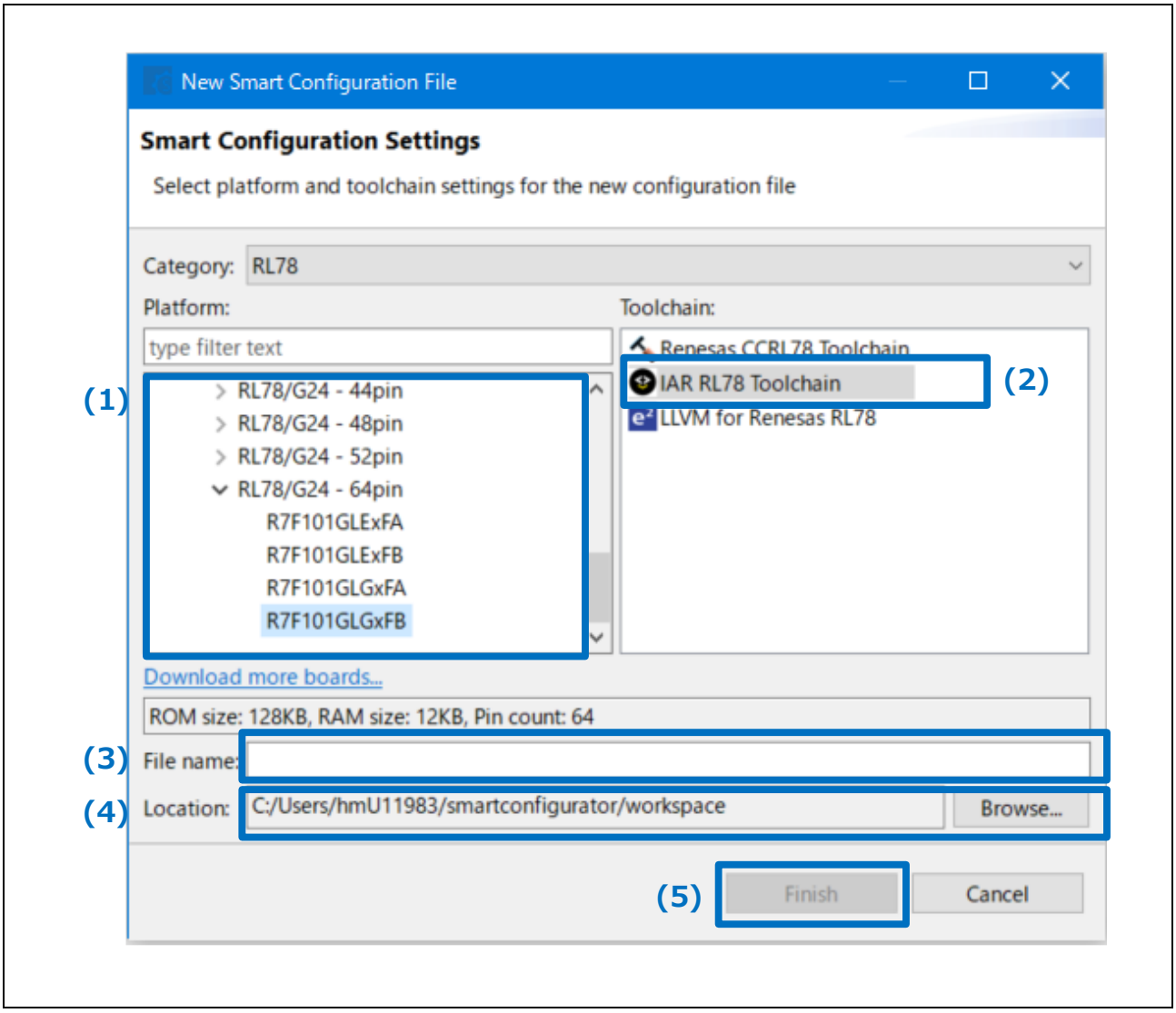

Figure 6-3 Create a Configuration File

   (6). Add driver component, configure the setting, generate code, and save the project.

Note: The *.eww, *.ewp, *.ewd and main.c files will be generated only for the first-time code generation, while the buildinfo.ipcf file will be generated always for each code generation.

3. Adding FAA Components
   (1). Select the "Components" page of the "Smart Configurator View" and click the "Add component" button.
   (2). Next, add the "Flexible Application Accelerator" component from the "Software Component Selection" page.

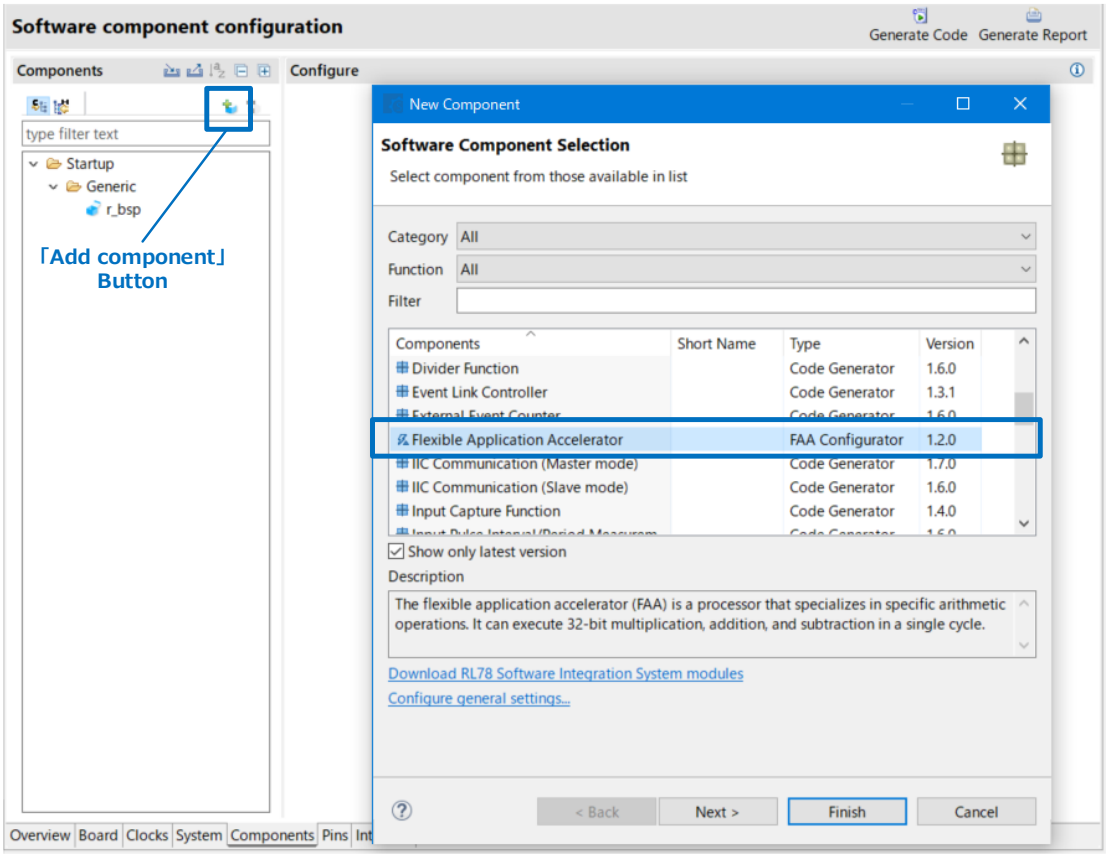

Figure 6-4 Adding FAA Components

4. Downloading FAA Modules
   Click on "Please download FAA data" displayed on the screen to see the FAA modules available for download. Select "FFT Library" to download.

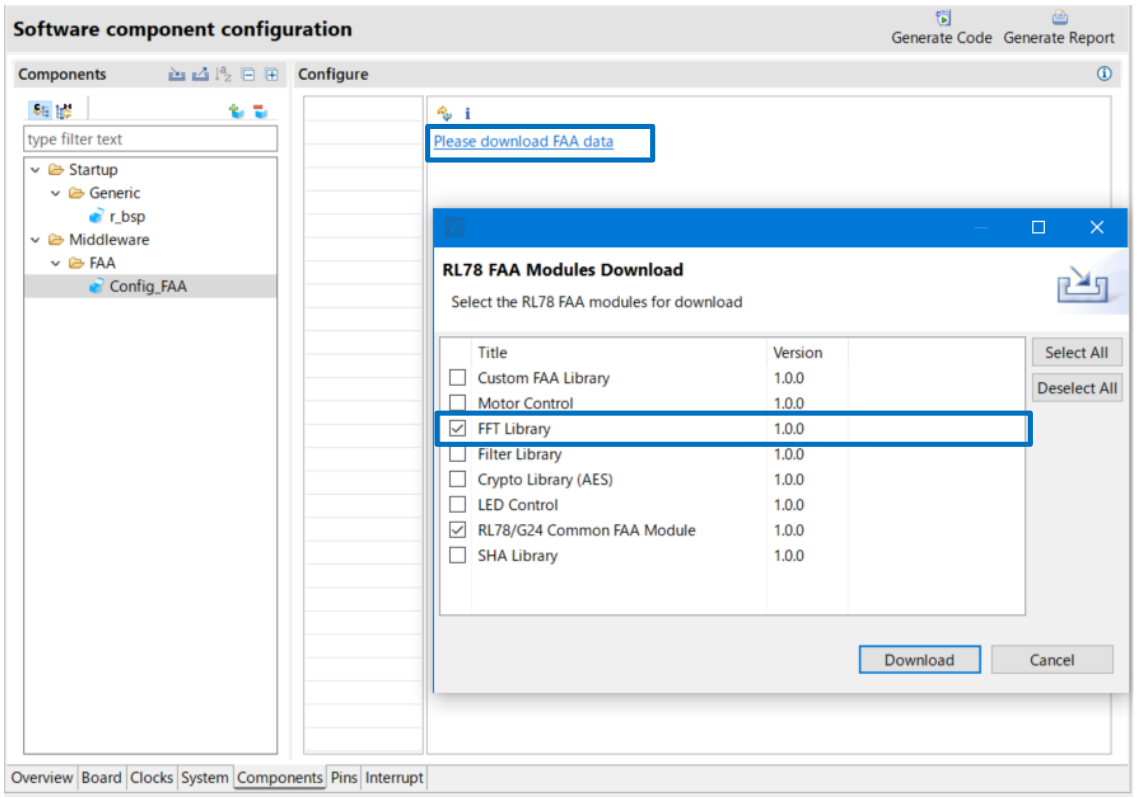

Figure 6-5 Downloading FAA Modules

5. FAA Module Configuration
   Select the "FFT" module from the list of downloaded FAA modules to display the configuration screen.
   Configure the configuration settings according to the user environment.

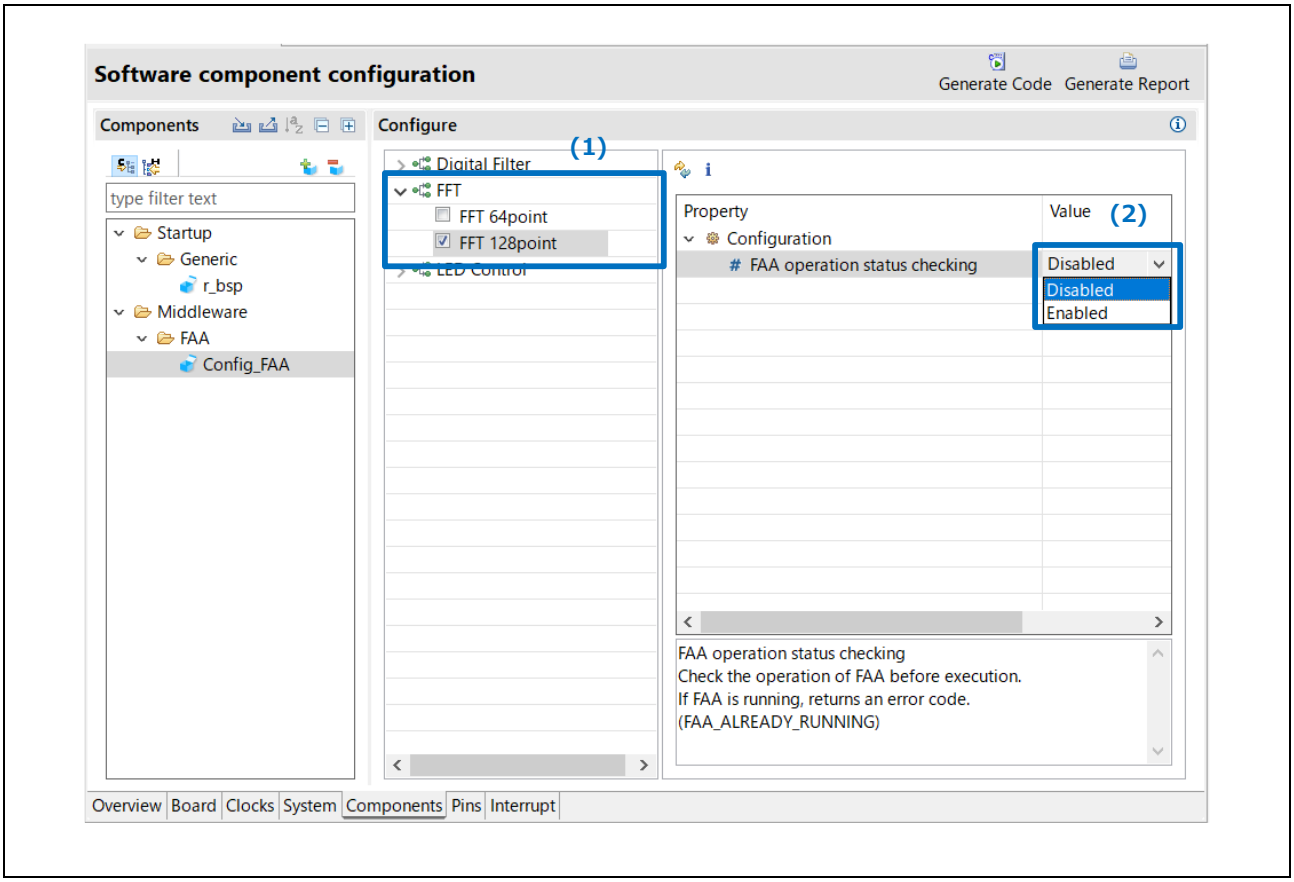

Figure 6-6 FAA Module Configuration

(1). Please select the FFT function (number of points) to use.

Note: You can select multiple options, but please choose only one.

(2). You can configure properties to check the operational status of the FAA.
The properties to be configured are as follows Table 6-6.

Table 6-6 Properties for Checking FAA Operational Status

Property	Value	Description
FAA operation status checking	Enabled	The operating state of FAA is checked when calling the API and return an error code if the FAA is in operation by another function. (R_DSCL_ERR_FAA_ALREADY_RUNNING)
	Disabled	The operating state of FAA is not checked.

6. Code Generation
   Output a source file for the configured details by clicking on the “Generate Code” button in the Smart Configurator view.

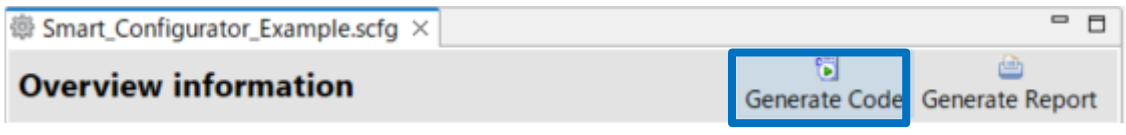

Figure 6-7 Generating a Source File

The Smart Configurator generates a source file in ¥<ProjectDir>¥src¥smc_gen and IAR related files in save location (refer to 2 Creating a New Smart Configurator Configuration File)

7. Loading in IAR Embedded Workbench
   When IAR environment is selected for the compiler to be used, Smart Configurator outputs the related files (*.eww / *.ewp / *.ewd / main.c) together with the source file. It is not necessary for the user to create project files in IAR Embedded Workbench.
   The usage procedure is as follows.
   (1). Select “Open Workspace…” from the “File” menu of IAR Embedded Workbench.
   (2). In the “Open Workspace” dialog box, browse to the folder where the project file is saved, select the project file (.eww), and click the “Open” button.

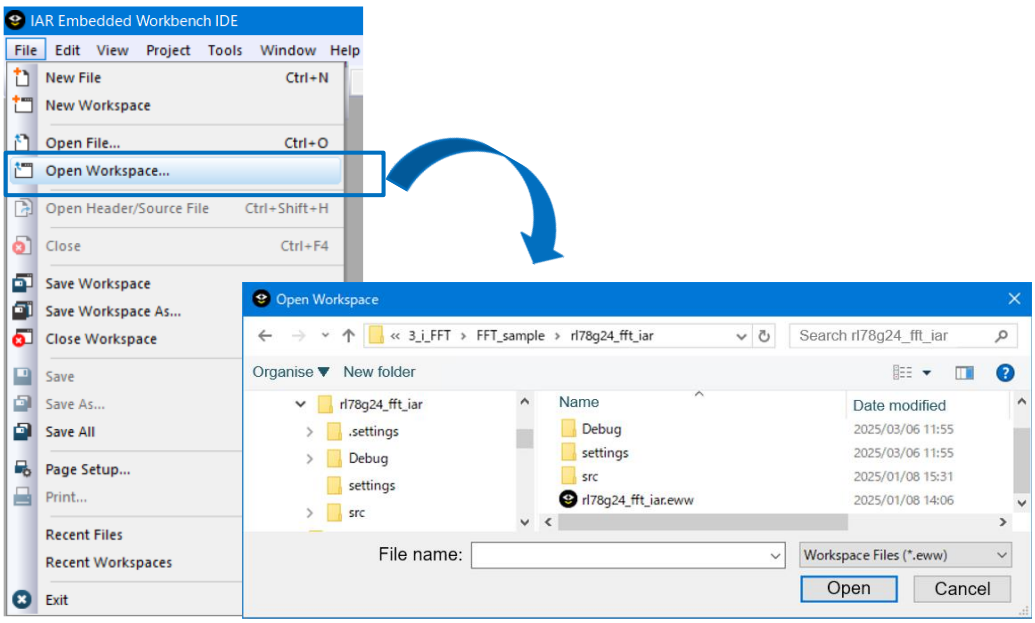

Figure 6-8 Load a *.eww File

(3). The source file output by the Smart Configurator is added to the IAR C project workspace.

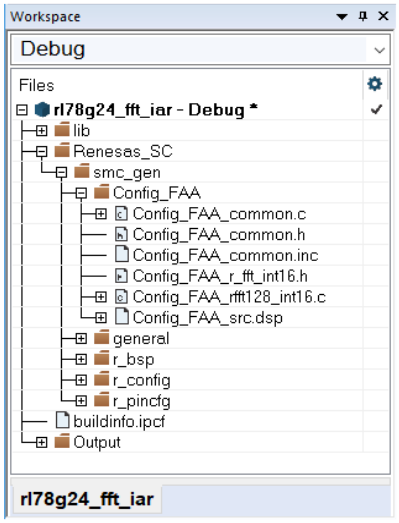

Figure 6-9 New Files Added to IAR Workspace

(4). Select “Options…” from the “Project” menu of IAR Embedded Workbench.
(5). In the “Options for node “Project Name” dialog box, change the target device to match with the target device selected when creating Smart Configurator’s configuration file.

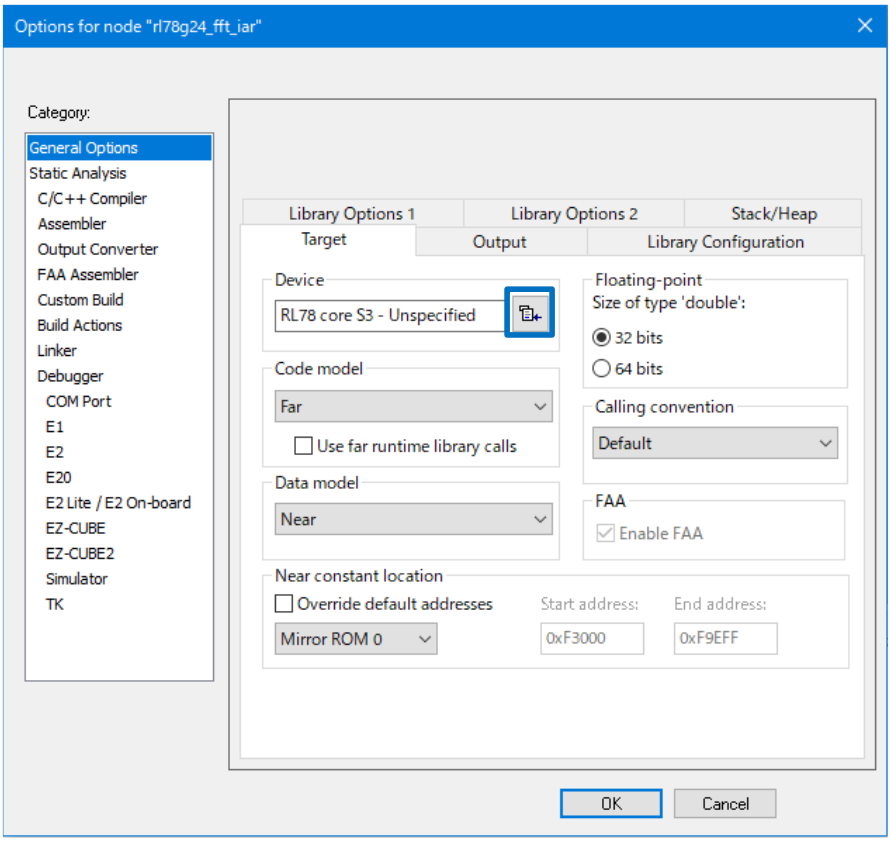

Figure 6-10 Change the target device

6.2.2.2 Preprocessor Macro Definition

When using the RL78/G24 FAA FFT library, please add the "R_FFT_FAA" definition to the project's preprocessor macro definitions.

Right-click on [Project Name], select [Options], choose [C/C++ Compiler] on the displayed screen, and in the [Preprocessor] tab, define the following macro in the [Symbol Definitions: (one per line)] field, then click the "OK" button.

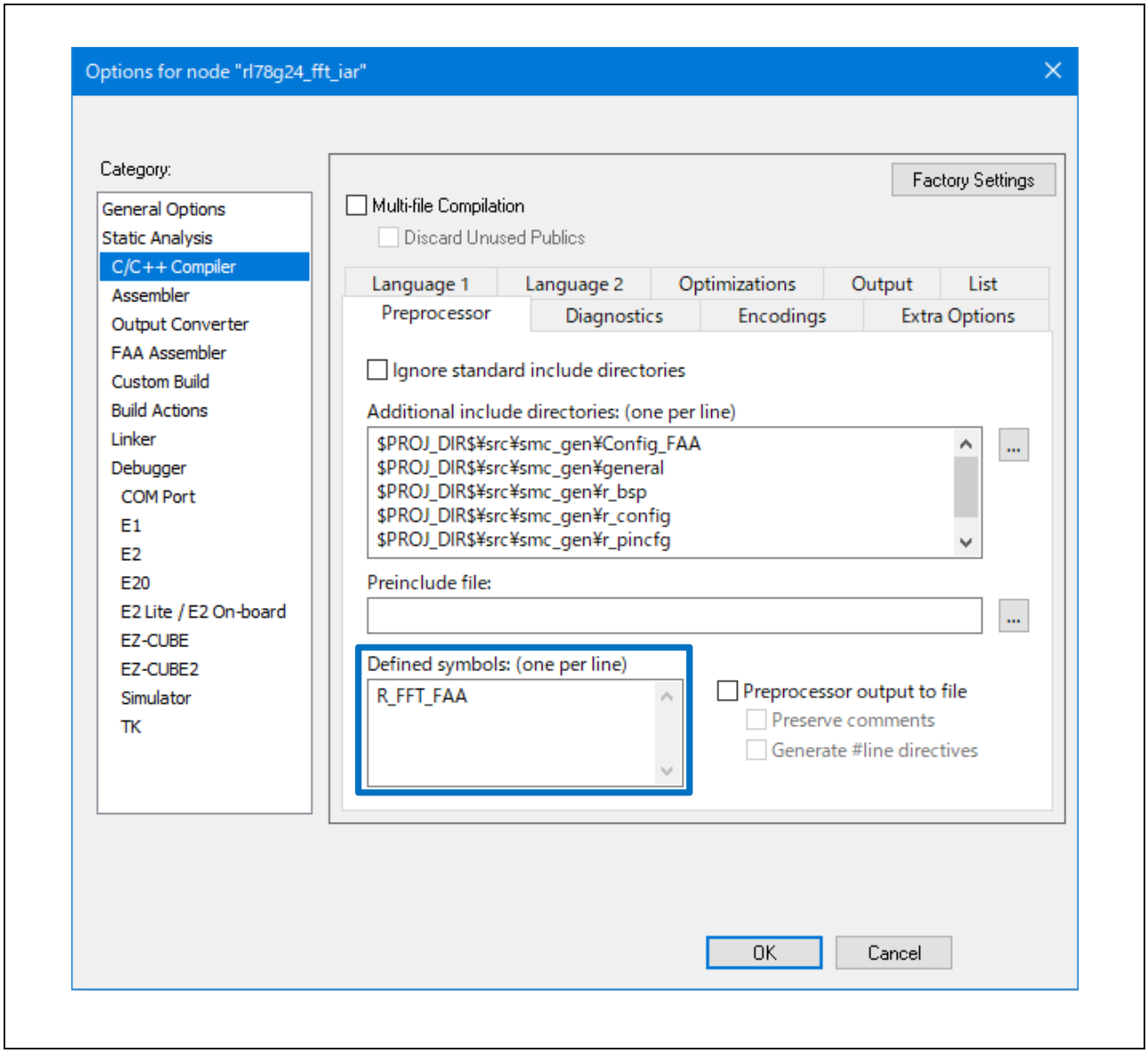

Figure 6-11 Preprocessor macro definition

6.2.2.3 Work area to be specified in API

For details, please refer to section **6.1.2.3 Work area to be specified in API.**

6.2.2.4 Return Values of API

For details, please refer to section **6.1.2.4 Return Values of API.**

6.2.3 ROM, RAM, and Stack Sizes

The ROM, RAM, and stack sizes of the various FFT library APIs are listed below (unit: bytes)

Table 6-7 ROM / RAM and Stack Size (IAR (RL78/G24 FAA))

API	ROM	RAM	stack	FAACODE	FAADATA	FAAStack
R_rfft64_int16	579	0	36	688	1432	10
R_rfft128_int16	682	0	36	688	1808	12

6.2.4 Section Information

The sections (segments) used by the various FFT library APIs are listed in the table below.

Table 6-8 Section Information (IAR (RL78/G24 FAA))

Section Name	Description
.textf	Program
.const	Constant data
FAACODE	FAA code area
FAADATA	FAA data area

6.2.5 Library Performance

The processing times when calling the various library functions (APIs) are listed below.

Table 6-9 Processing times (IAR (RL78/G24 FAA))

API	Time (System Clock = 48 MHz)
R_rfft64_int16	Approx.0.2ms
R_rfft128_int16	Approx.0.5ms

Note: Measured using the execution time measurement function of the integrated development environment (IAR Embedded Workbench for Renesas RL78)

6.2.6 Version Information

This library does not support the r_fft_a_version variable to indicate version information. Refer to the header information in the source.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar. 31, 2012	—	First edition issued
1.01	Apr.1, 2014	—	Updated product configuration to match package version V. 1.00 Release 01. Added support for IAR Embedded Workbench.
1.02	Apr.1, 2015	2	Updated product configuration to match package version V. 1.00 Release 02.
1.03	Oct. 1, 2015	—	Changed CubeSuite+ to CS+ for CA and CX. Added support for CS+ for CC.
1.04	Apr. 13, 2021	—	Added RL78/G23 to CS+ for CC. Deleted IAR.
1.05	Oct. 25, 2021	P6 P8-P10	Remove CS + for CA, CX Updated the processing time of FFT library for RL78/G14 and RL78/G23 Add IAR
1.06	Jun. 27, 2022	P3 P11-P13	Added "e^{2}studio for LLVM" in "Table 1.1 FFT Library Configuration". Added chapter "5. e^{2}studio for LLVM".
1.07	Sep. 26, 2022	—	Added RL78/G15
1.08	Apr. 19, 2023	—	Added RL78/G24 FAA
1.09	Mar. 19, 2025	—	Added RL78/G24 FAA for IAR

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/.