RL78 ファミリ

FFT ライブラリ: 導入ガイド

要旨

本資料は、FFT ライブラリを導入するための情報を記します。FFT (Fast Fourier Transform) とは離散フーリエ変換(Discrete Fourier Transform) を高速に実行するためのアルゴリズムです。Cooley 氏、Tukey 氏が 1965 年に開発した実装方法が FFT として一般的によく知られており、デジタル信号処理アプリケーションの飛躍的な発展に貢献しています。

FFT ライブラリはルネサスマイコン向けにより効率的に処理できるように、アセンブラチューニング版を用意しています。

動作確認デバイス

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

目次

1. 製品構成

本製品は、以下のものから構成されています。

1. FFT ライブラリ V.1.03 Release00 および RL78/G24 FAA FFT ライブラリ V.1.01
2.上記ライブラリ 導入ガイド（r20an0150jj0109_rl78_fft.pdf）
本製品の型名 : R0M7800LF0010RRC

本製品は、以下の表 1-1 のファイルが含まれます。

表 1-1 FFT ライブラリ の製品構成

	内容
r20an0150jj0109_rl78_fft.pdf	導入ガイド(本書)
ワークスペース (workspace)	
ドキュメント (doc)	
英語(en)	
r20uw0099ej0102_fft.pdf	ユーザーズマニュアル
r20an0150ej0109_rl78_fft.pdf	導入ガイド
日本語(ja)	
r20uw0099jj0102_fft.pdf	ユーザーズマニュアル
r20an0150jj0109_rl78_fft.pdf	導入ガイド (本書)
CS+,e²studio for CC 用 (CS+,e2studio for CC)	
FFT ライブラリ(sample¥<サンプルプログラムフォルダ>¥lib)	
libfft_rl78g13.lib	RL78/G13 用 FFT ライブラリ（アセンブラ版）version 1.01
libfft_rl78g14.lib	RL78/G14, RL78/G23, RL78/G24 256 ポイント用 FFT ライブラリ（アセンブラ版）version 1.01
libfft_rl78_S2_NOMDA.lib	RL78/G15 用 FFT ライブラリ（アセンブラ版）version 1.03
r_fft_int16.h	FFT ライブラリヘッダファイル
r_stdint.h	型定義ヘッダファイル
サンプルプログラム(sample)	
rl78g14_fft_ccrl_CS+	サンプル CS+ for CC プロジェクト(RL78/G14, RL78/G23)
rl78g14_fft_ccrl_e2studio	サンプル e²studio for CC プロジェクト(RL78/G14, RL78/G23)
rl78g15_fft_ccrl_CS+	サンプル CS+ for CC プロジェクト(RL78/G15)
rl78g15_fft_ccrl_e2studio	サンプル e²studio for CC プロジェクト(RL78/G15)
CS+,e²studio for CC 用 (CS+,e2studio for CC (RL78G24 FAA))	
FFT ライブラリ(sample¥<サンプルプログラムフォルダ>¥lib)	
libfft_rl78g14.lib	RL78/G14, RL78/G23, RL78/G24 256 ポイント用 FFT ライブラリ（アセンブラ版）version 1.01
r_fft_int16.h	FFT ライブラリヘッダファイル
FAA 用 FFT ライブラリ(sample¥<サンプルプログラムフォルダ>¥smc_gen)	
Config_FAA	FAA 用 FFT ライブラリ version 1.00（64 および 128 ポイント用）
サンプルプログラム(sample)	
rl78g24_fft_ccrl_CS+	サンプル CS+ for CC プロジェクト(RL78/G24)
rl78g24_fft_ccrl_e2studio	サンプル e²studio for CC プロジェクト(RL78/G24)
IAR Embedded Workbench 用 (IAR)	
FFT ライブラリ(sample¥<サンプルプログラムフォルダ>¥lib)	

libfft_rl78g14.a	RL78/G14, RL78/G23 用 FFT ライブラリ（アセンブラ版） version 1.01
libfft_rl78_S2_NOMDA.a	RL78/G15 FFT ライブラリ（アセンブラ版）version 1.03
r_fft_int16.h	FFT ライブラリヘッダファイル
r_stdint.h	型定義ヘッダファイル
サンプルプログラム(sample)	
rl78g14_fft_iar	サンプル IAR Embedded Workbench プロジェクト (RL78/G14, RL78/G23)
rl78g15_fft_iar	サンプル IAR Embedded Workbench プロジェクト (RL78/G15)
IAR Embedded Workbench 用 (IAR (RL78G24-FAA))	
FFT ライブラリ(sample¥<サンプルプログラムフォルダ>¥lib)	
libfft_rl78g14.a	RL78/G14, RL78/G23, RL78/G24 256 ポイント用 FFT ライブラリ（アセンブラ版）version 1.01
r_fft_int16.h	FFT ライブラリヘッダファイル
FAA 用 FFT ライブラリ(sample¥<サンプルプログラムフォルダ>¥smc_gen)	
Config_FAA	FAA 用 FFT ライブラリ version 1.01（64 および 128 ポイント用）
サンプルプログラム(sample)	
rl78g24_fft_iar	サンプル IAR Embedded Workbench プロジェクト (RL78/G24)
e²studio for LLVM 用（e2studio for LLVM）	
FFT ライブラリ(sample¥<サンプルプログラムフォルダ>¥lib)	
libfft_rl78.a	RL78/G23 用 FFT ライブラリ（アセンブラ版） version 1.01
libfft_rl78_S2_NOMDA.a	RL78/G15 用 FFT ライブラリ（アセンブラ版） version 1.03
r_fft_int16.h	FFT ライブラリヘッダファイル
r_stdint.h	型定義ヘッダファイル
サンプルプログラム(sample)	
rl78g23_fft_llvm_e2studio	サンプル e²studio for LLVM 用プロジェクト(RL78/G23)
rl78g15_fft_llvm_e2studio	サンプル e²studio for LLVM 用プロジェクト(RL78/G15)

2. ライブラリ関数

FFT ライブラリは、以下のライブラリ関数（API）をサポートしています。

表 2-1 FFT ライブラリ関数

API	説明
R_rfft64_int16	16-bit 固定小数点 実数 FFT（64 ポイント）
R_rfft128_int16	16-bit 固定小数点 実数 FFT（128 ポイント）
R_rfft256_int16	16-bit 固定小数点 実数 FFT（256 ポイント）

3. CS+, e^{2}studio for CC 用

3.1 制限事項

RL78/G13 用 FFT ライブラリは内蔵の乗除積和算器を積和演算のために使用しています。そのため、ユーザの割り込み関数の内部で以下のレジスタ値を変更しないでください。なお、乗除積和算器および関連レジスタについては「RL78/G13 ユーザーズマニュアル ハードウェア編」の第 14 章「第 14 章 乗除積和算器」を参照ください。

レジスタ

- 乗除算デ−タ・レジスタ A（L）（MDAL）
- 乗除算デ−タ・レジスタ A（H）（MDAH）
- 乗除算データ・レジスタ B（L）（MDBL）
- 乗除算データ・レジスタ B（H）（MDBH）
- 乗除算データ・レジスタ C（L）（MDCL）
- 乗除算データ・レジスタ C（H）（MDCH）

制御レジスタ

- 乗除算コントロール・レジスタ（MDUC）

3.2 コンパイラオプション

本ライブラリは、以下のコンパイルオプションにてライブラリを生成しています。

【コンパイルオプション】

RL78/G13, RL78/G14, RL78/G23, RL78/G24 256 ポイント用 FFT ライブラリ：

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

RL78/G15 用 FFT ライブラリ：

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

3.3 開発環境

開発環境を以下に示します。

ユーザアプリケーション開発時は以下のバージョンより新しいものをご使用下さい。

[ソフトウェアツール]

RL78/G13, RL78/G14, RL78/G23, RL78/G24 256 ポイント用 FFT ライブラリ：

-統合開発環境

CS+ for CC V8.06.00

-C コンパイラ

CC-RL V1.10

-デバッガ

RL78 シミュレータ

RL78/G15 用 FFT ライブラリ：

-統合開発環境

CS+ for CC V8.08.00

e²studio Version: 2022-10(22.10.0)

-C コンパイラ

CCRL V1.11.00

-デバッガ

E2 エミュレータ Lite

3.4 ROM / RAM /スタックサイズ

各 FFT ライブラリの ROM / RAM /スタックのサイズは次の表 3-1、表 3-2 表 3-3 とおりです（単位はバイト数）。

表 3-1 ROM / RAM /スタックのサイズ (CS+,e²studio for CC (RL78/G13 用))

API	ROM	RAM	スタック
R_rfft64_int16	1260	0	68
R_rfft128_int16	1512	0	68
R_rfft256_int16	2018	0	68

表 3-2 ROM / RAM /スタックのサイズ (CS+,e²studio for CC (RL78/G14, RL78/G23, RL78/G24 256 ポイント用))

API	ROM	RAM	スタック
R_rfft64_int16	1224	0	68
R_rfft128_int16	1476	0	68
R_rfft256_int16	1982	0	68

表 3-3 ROM / RAM /スタックのサイズ (CS+,e²studio for CC (RL78/G15 用))

API	ROM	RAM	スタック
R_rfft64_int16	1358	0	82
R_rfft128_int16	1610	0	82

3.5 セクション情報

各 FFT ライブラリは、次の表 3-4、表 3-5 に示すセクション（セグメント）を使用します。

表 3-4 セクション情報 (CS+,e²studio for CC
(RL78/G13, RL78/G14, RL78/G23, RL78/G24 256 ポイント用))

セクション名	内容	セクション属性
.textf	プログラム	.CSEG TEXTF
.const	定数データ	.CSEG CONST

表 3-5 セクション情報 (CS+,e²studio for CC (RL78/G15 用))

セクション名	内容	セクション属性
.text	プログラム	SECTION=.text
.const	定数データ	SECTION=.const

3.6 ライブラリ性能

以下の表 3-6、表 3-7、表 3-8 に本ライブラリのライブラリ関数（API）呼び出し毎の処理時間を示します。

表 3-6 処理時間 (CS+,e²studio for CC (RL78/G13 用))

API	Time (system clock = 32MHz)
R_rfft64_int16	約 0.4ms
R_rfft128_int16	約 0.9ms
R_rfft256_int16	約 1.9ms

表 3-7 処理時間 (CS+,e²studio for CC (RL78/G14, RL78/G23, RL78/G24 256 ポイント用))

API	Time (system clock = 32MHz)
R_rfft64_int16	約 0.3ms
R_rfft128_int16	約 0.7ms
R_rfft256_int16	約 1.6ms

【注】 統合開発環境(CS+)の実行時間計測機能を使用して計測

表 3-8 処理時間 (CS+,e²studio for CC (RL78/G15 用))

API	Time (system clock = 16MHz)
R_rfft64_int16	約 29.3ms
R_rfft128_int16	約 73.1ms

【注】 統合開発環境(e²studio)の実行時間計測機能を使用して計測

3.7 バージョン情報

本ライブラリは、r_fft_a_version 変数に文字列でバージョン情報を格納しています。以下の extern 宣言によりこの変数にアクセスすることが出来ます。

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

また、本製品のライブラリに格納されているデータは以下の通りです。

RL78/G13 用 FFT ライブラリ：

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

RL78/G14, RL78/G23, RL78/G24 256 ポイント用 FFT ライブラリ：

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

RL78/G15 用 FFT ライブラリ：

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

4. IAR Embedded Workbench 用

4.1 コンパイラオプション

本ライブラリは、以下のコンパイルオプションにてライブラリを生成しています。

【コンパイルオプション】

RL78/G14, RL78/G23, RL78/G24 256 ポイント用 FFT ライブラリ：

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

RL78/G15 用 FFT ライブラリ：

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

4.2 開発環境

開発環境を以下に示します。

ユーザアプリケーション開発時は以下のバージョンより新しいものをご使用下さい。

[ソフトウェアツール]-統合開発環境

IAR Embedded Workbench for Renesas RL78 4.21.1

-C コンパイラ

IAR C/C++ Compiler for Renesas RL78 4.21.1.2409

-デバッガ

IAR C-SPY Debugger Kernel 8.5.2.7561

4.3 ROM / RAM /スタックサイズ

各 FFT ライブラリの ROM / RAM /スタックのサイズは次の表 4-1、表 4-2 とおりです（単位はバイト数）。

表 4-1 ROM / RAM /スタックのサイズ (IAR (RL78/G14, G23, G24 256 ポイント用))

API	ROM	RAM	スタック
R_rfft64_int16	1226	0	68
R_rfft128_int16	1478	0	68
R_rfft256_int16	1984	0	68

表 4-2 ROM / RAM /スタックのサイズ (IAR (RL78/G15 用))

API	ROM	RAM	スタック
R_rfft64_int16	1350	0	82
R_rfft128_int16	1602	0	82

4.4 セクション情報

各 FFT ライブラリは、次の表 4-3、表 4-4 に示すセクション（セグメント）を使用します。

表 4-3 セクション情報 (IAR (RL78/G14, RL78/G23, RL78/G24 256 ポイント用))

セクション名	内容
.textf	プログラム
.const	定数データ

表 4-4 セクション情報 (IAR (RL78/G15 用))

セクション名	内容
.text	プログラム
.const	定数データ

4.5 ライブラリ性能

以下の表 4-5、表 4-6 に本ライブラリのライブラリ関数（API）呼び出し毎の処理時間を示します。

表 4-5 処理時間 (IAR (RL78/G14, RL78/G23, RL78/G24 256 ポイント用))

API	Time (system clock = 32MHz)
R_rfft64_int16	約 0.2ms
R_rfft128_int16	約 0.6ms
R_rfft256_int16	約 1.5ms

【注】 統合開発環境(IAR Embedded Workbench for Renesas RL78)の実行時間計測機能を使用して計測

表 4-6 処理時間 (IAR (RL78/G15 用))

API	Time (system clock = 16MHz)
R_rfft64_int16	約 18.3ms
R_rfft128_int16	約 44.4ms

【注】 統合開発環境(IAR Embedded Workbench for Renesas RL78)の実行時間計測機能を使用して計測

4.6 バージョン情報

本ライブラリは、r_fft_a_version 変数に文字列でバージョン情報を格納しています。以下の extern 宣言によりこの変数にアクセスすることが出来ます。

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

また、本製品のライブラリに格納されているデータは以下の通りです。

RL78/G14, RL78/G23 用 FFT ライブラリ：

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

RL78/G15 用 FFT ライブラリ：

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

5. e²studio for LLVM 用

5.1 コンパイラオプション

本ライブラリは、以下のコンパイルオプションにてライブラリを生成しています。

【コンパイルオプション】

RL78/G23 用 FFT ライブラリ：

```
CPU Type: S3-core
Optimizition: None(-O0)
```

RL78/G15 用 FFT ライブラリ：

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

5.2 開発環境

開発環境を以下に示します。

ユーザアプリケーション開発時は以下のバージョンより新しいものをご使用下さい。

[ソフトウェアツール]

RL78/G23 用 FFT ライブラリ：

-統合開発環境

e²studio Version: 2022-04(22.4.0)

-C コンパイラ

LLVM V10.0.0.202203

-デバッガ

E2 エミュレータ Lite

RL78/G15 用 FFT ライブラリ：

-統合開発環境

e²studio Version: 2022-10(22.10.0)

-C コンパイラ

LLVM V10.0.0.202207

-デバッガ

E2 エミュレータ Lite

5.3 ROM / RAM /スタックサイズ

各 FFT ライブラリの ROM / RAM /スタックのサイズは次の表 5-1、表 5-2 とおりです（単位はバイト数）。

表 5-1 ROM / RAM /スタックのサイズ (e²studio for LLVM (RL78/G23 用))

API	ROM	RAM	スタック
R_rfft64_int16	1224	0	68
R_rfft128_int16	1476	0	68
R_rfft256_int16	1982	0	68

表 5-2 ROM / RAM /スタックのサイズ (e²studio for LLVM (RL78/G15 用))

API	ROM	RAM	スタック
R_rfft64_int16	1346	0	82
R_rfft128_int16	1854	0	82

5.4 セクション情報

各 FFT ライブラリは、次の表に示すセクション（セグメント）を使用します。

表 5-3 セクション情報 (e²studio for LLVM (RL78/G15, RL78/G23 用))

セクション名	内容
.text	プログラム
.rodata	定数データ

5.5 ライブラリ性能

以下の表 5-4、表 5-5 に本ライブラリのライブラリ関数（API）呼び出し毎の処理時間を示します。

表 5-4 処理時間 (e²studio for LLVM (RL78/G23 用))

API	Time (system clock = 32MHz)
R_rfft64_int16	約 0.3ms
R_rfft128_int16	約 0.7ms
R_rfft256_int16	約 1.6ms

表 5-5 処理時間 (e²studio for LLVM(RL78/G15 用))

API	Time (system clock = 16MHz)
R_rfft64_int16	約 29.2ms
R_rfft128_int16	約 73.2ms

【注】 統合開発環境(e²studio)の実行時間計測機能を使用して計測

5.6 バージョン情報

本ライブラリは、r_fft_a_version 変数に文字列でバージョン情報を格納しています。以下の extern 宣言によりこの変数にアクセスすることが出来ます。

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

また、本製品のライブラリに格納されているデータは以下の通りです。

RL78/G23 用 FFT ライブラリ：

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

RL78/G15 用 FFT ライブラリ：

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

6. RL78/G24 FAA 用

6.1 CS+, e2studio for CC

6.1.1 開発環境

開発環境を以下に示します。

ユーザアプリケーション開発時は以下のバージョンより新しいものをご使用下さい。

[ソフトウェアツール]

RL78/G24 FAA 用 FFT ライブラリ：

-統合開発環境

CS+ for CC V8.09.00

e2studio Version: 2023-07(23.7.0)

-C コンパイラ

CC-RL V1.12

-デバッガ

E2 エミュレータ Lite

6.1.2 RL78/G24 FAA 用 FFT ライブラリ

64 ポイント,128 ポイントの API を使用する場合は下記に示す方法で、スマートコンフィグレータを使用して FAA 用 FFT ライブラリを生成してください。

スマートコンフィグレータの基本的な操作方法については下記ユーザーガイドを参照してください。

- RL78 スマート・コンフィグレータユーザーガイド：CS+編 (R20AN0580)
- RL78 スマート・コンフィグレータユーザーガイド: e2 studio 編 (R20AN0579)

RL78/G24 に搭載の FAA のプログラムのビルドおよびデバッガ操作について下記ユーザーガイドを参照してください。

- RL78/G24 FAA ツールガイド CS+編 (R01AN7094)
- RL78/G24 FAA ツールガイド e2 studio 編 (R01AN7094)

FAA 用 FFT ライブラリは 256 ポイントに非対応のため、256 ポイントの API を使用する場合は「**3 CS+, e2studio for CC 用**」に記載のライブラリ libfft_rl78g14.lib (RL78/G14, RL78/G23, RL78/G24 256 ポイント用)をリンクして使用してください。

6.1.2.1 ライブラリ生成方法

RL78/G24 FAA 用 FFT ライブラリはスマートコンフィグレータを使用して生成します。

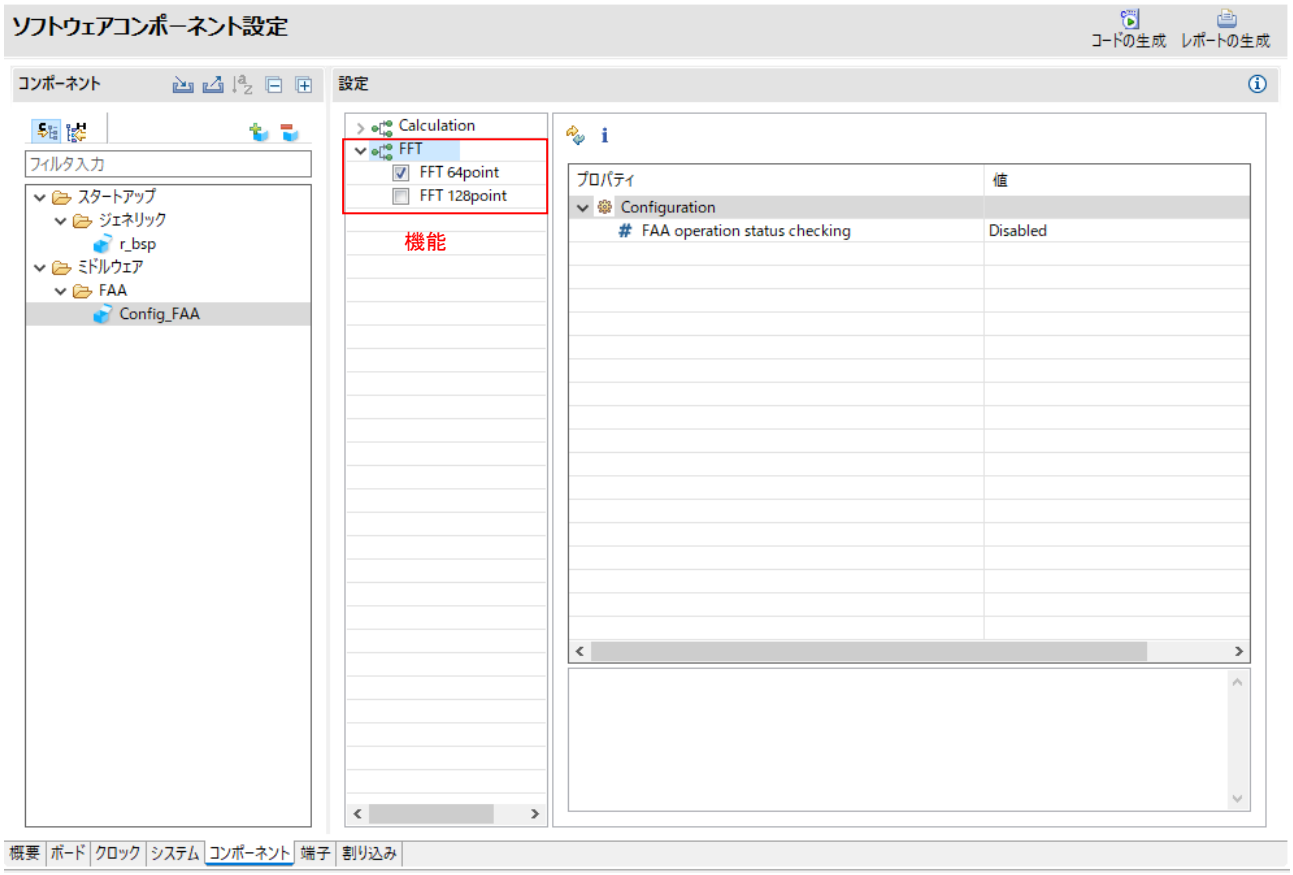

図 6-1 FAA モジュールのコンフィグレーション

1. スマートコンフィグレータの機能から使用する FFT ポイント数を選択してください。
   ※複数選択できますが一つだけ選択してください
2. 機能を選択するとプロパティの設定ができます。
3. ¥src¥smc_gen¥Config_FAA にコードが生成されます。

設定するプロパティを以下に示します。

表 6-1 FAA 動作状態確認の設定するプロパティ

プロパティ	説明
FAA operation status checking	FAA 動作状態の確認 ・Enabled：API を呼び出す際に FAA 動作状態を確認し、他の関数により FAA が動作中の場合はエラーコードを返します。（R_FFT_ERR_FAA_ALREADY_RUNNING） ・Disabled：FAA 動作状態の確認を行いません。

6.1.2.2 プリプロセッサ・マクロ定義

RL78/G24 FAA 用 FFT ライブラリ使用時は、プロジェクトのプリプロセッサ・マクロ定義に以下の定義を追加してください。

追加する定義は「R_FFT_FAA」です。

6.1.2.3 API に指定する work 領域

RL78/G24 FAA 用 FFT ライブラリ使用時は、API の第 4 引数に与える work 領域に FAA 側で確保している work 領域を指定してください。
指定する work 領域名は「r_fft_int16.h」に記載している「V_rfft_work」です。

6.1.2.4 API の戻り値

RL78/G24 FAA 用 FFT ライブラリでは API へ新規に戻り値を追加しています。
以下に追加した定義を示します。

表 6-2 戻り値

戻り値	説明
R_FFT_STATUS_OK	正常終了
R_FFT_ERR_INPUT_NULL	入力パラメータ input が NULL です。
R_FFT_ERR_OUTPUT_NULL	出力パラメータ output が NULL です。
R_FFT_ERR_WINDOW_NULL	入力パラメータ window が NULL です。
R_FFT_ERR_WORK_NULL	入力パラメータ work が NULL です。
R_FFT_ERR_FAA_ALREADY_RUNNING	FAA が動作中です。

6.1.3 ROM / RAM /スタックサイズ

本ライブラリの ROM / RAM /スタックのサイズは次のとおりです（単位はバイト数）。

表 6-3 ROM / RAM /スタックサイズ (CS+,e²studio for CC (RL78/G24 FAA 用))

API	ROM	RAM	スタック	FAACODE	FAADATA	FAA スタック
R_rfft64_int16	575	0	36	688	1432	8
R_rfft128_int16	704	0	36	688	1808	8

6.1.4 セクション情報

本ライブラリは、次の表 6-4 に示すセクション（セグメント）を使用します。

表 6-4 セクション情報 (CS+,e²studio for CC (RL78/G24 FAA 用))

セクション名	内容
.textf	プログラム
.const	定数データ
FAACODE	FAA コード領域
FAADATA	FAA データ領域

6.1.5 ライブラリ性能

以下に本ライブラリのライブラリ関数（API）呼び出し毎の処理時間を示します。

表 6-5 処理時間 (CS+,e²studio for CC (RL78/G24 FAA 用))

API	Time (system clock = 48MHz)
R_rfft64_int16	約 0.2ms
R_rfft128_int16	約 0.5ms

【注】 統合開発環境(e²studio)の実行時間計測機能を使用して計測

6.1.6 バージョン情報

本ライブラリはバージョン情報を示す r_fft_a_version 変数はサポートしていません。ソースのヘッダ情報を参照してください。

6.2 IAR Embedded Workbench

6.2.1 開発環境

開発環境を以下に示します。

ユーザアプリケーション開発時は以下のバージョンより新しいものをご使用下さい。

[ソフトウェアツール]

-統合開発環境

IAR Embedded Workbench for Renesas RL78 V5.10.3

-C コンパイラ

IAR C/C++ Compiler for Renesas RL78 5.10.3.2716 (5.10.3.2716)

-コンフィグレータ(SC)

Renesas Smart Configurator for RL78 V1.12.0

-デバッガ

E2 エミュレータ Lite

【コンパイルオプション】

RL78/G24 用 FFT ライブラリ：

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

6.2.2 RL78/G24 FAA 用 FFT ライブラリ

64 ポイント,128 ポイントの API を使用する場合は下記に示す方法で、スマートコンフィグレータを使用して FAA 用 FFT ライブラリを生成してください。

スマートコンフィグレータの基本的な操作方法については下記ユーザーガイドを参照してください。

- RL78 スマートコンフィグレータ ユーザーガイド：IAR 編 (R20AN0581)

FAA 用 FFT ライブラリは 256 ポイントに非対応のため、256 ポイントの API を使用する場合は「**4 IAR Embedded Workbench 用**」に記載のライブラリ libfft_rl78g14.a(RL78/G14, RL78/G23, RL78/G24 256 ポイント用)をリンクして使用してください。

6.2.2.1 ライブラリ作成方法

1. スマートコンフィグレータの起動
   Windows スタートメニューから「Renesas Electronics Smart Configurator」→「Smart Configurator for RL78 Vx.x.x」を選択します。選択後、スマートコンフィグレータのメインウィンドウが起動します。
   【注】Vx.x.x はご使用のバージョンに読み替えてください。

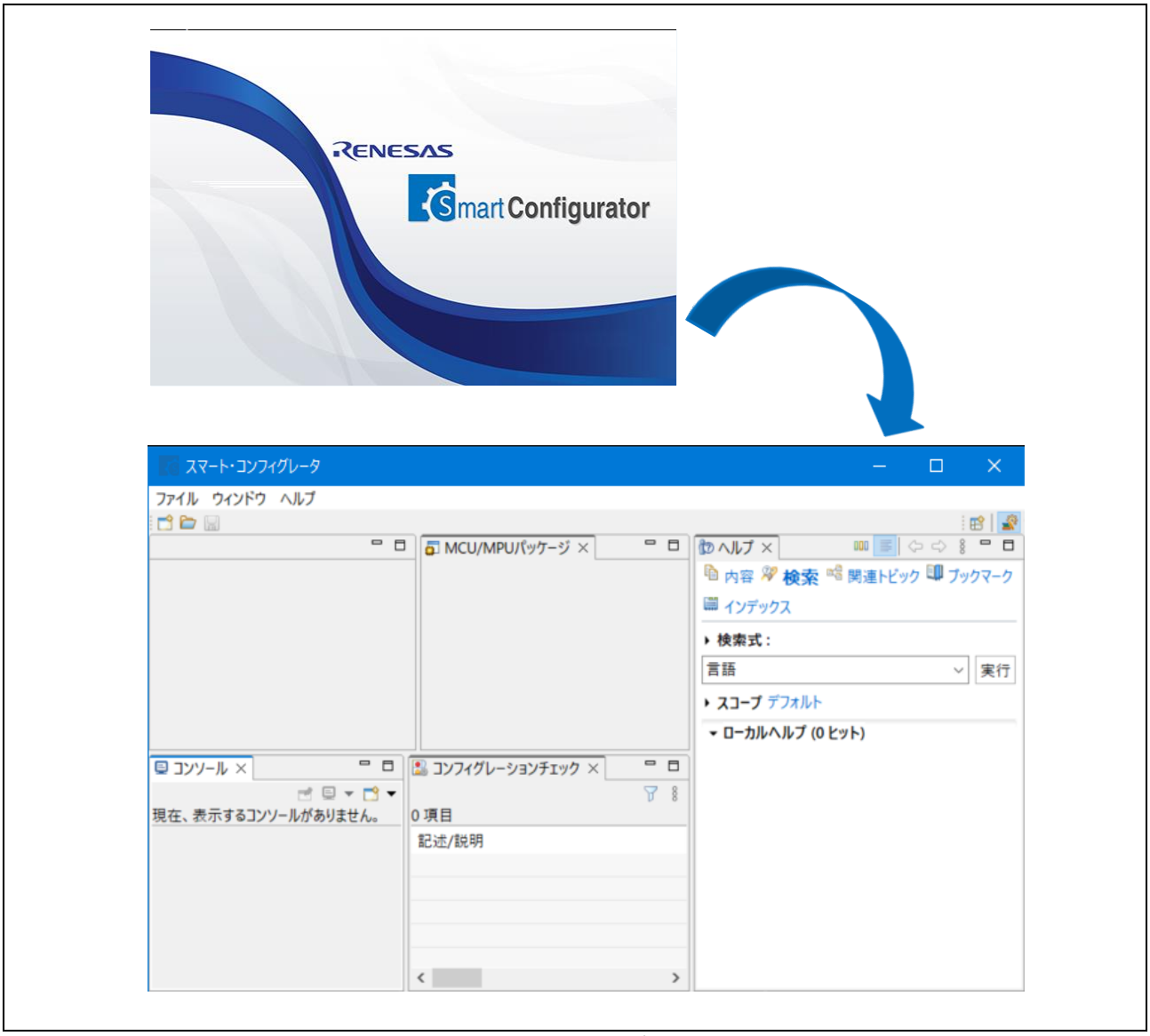

図 6-2 スマートコンフィグレータの起動

2. 新規作成

メインツールバーの 「新規コンフィグレーションファイル」ボタンをクリックするとダイアログが表示されます。

(1). ［プラットフォーム:］で、デバイスを選択します。

(2). ［ツールチェーン:］で、「IAR RL78 Toolchain」を選択します。

(3). ［ファイル名:］に、ファイル名を入力します。

(4). ［ロケーション:］を確認します。変更したい場合は、「参照」をクリックして保存先を選択してください。

【注】 「コード生成」 ボタンをクリックすると、*.eww、*.ewp、*.ewd、main.c、および buildinfo.ipcf ファイルがこの場所に生成されます。

(5). 「終了」をクリックして、コンフィグレーションファイルを作成します。

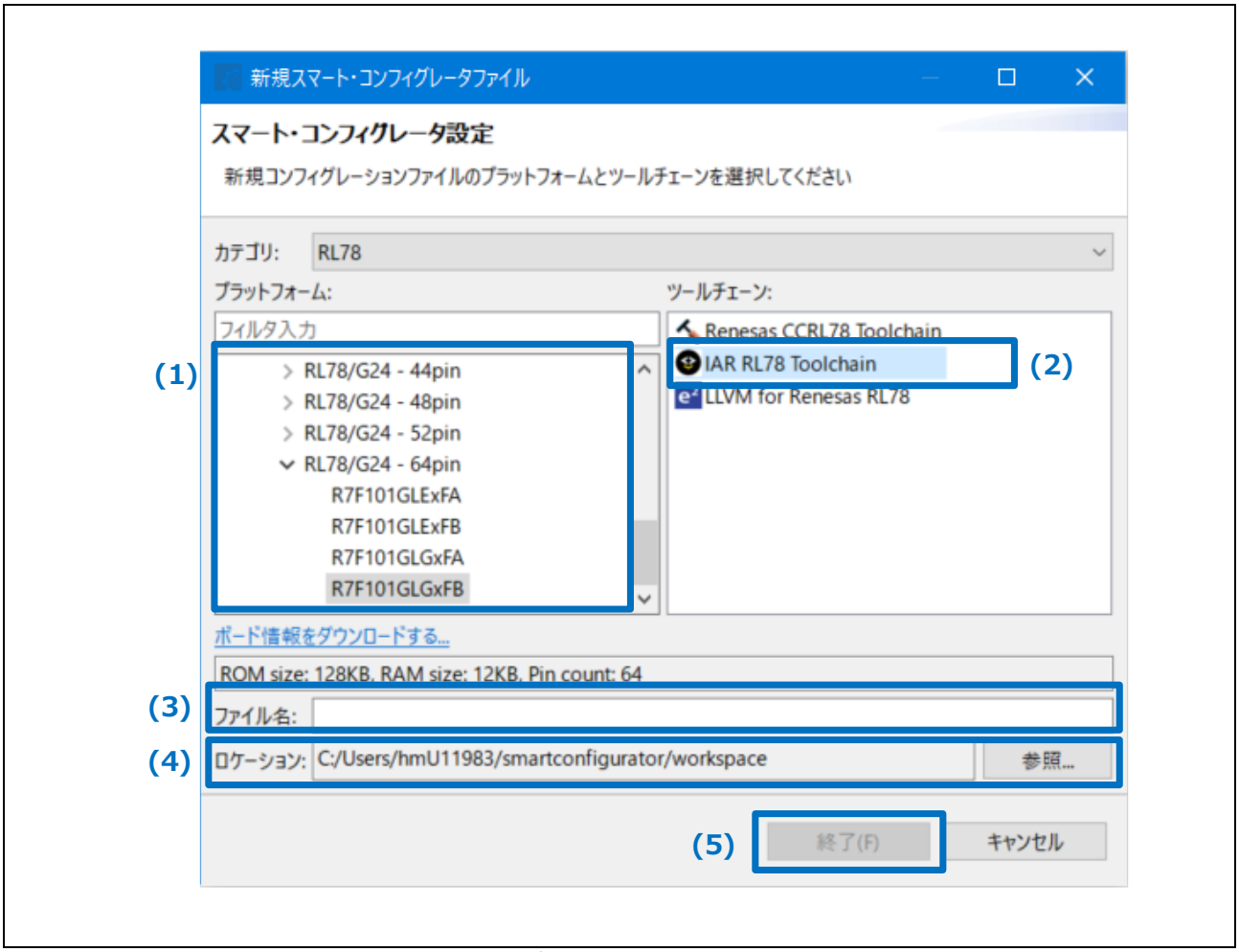

図 6-3 コンフィグレーションファイルの新規作成

(6). 任意のコンポーネントを追加し設定したあと、コードを生成し、プロジェクトを保存します。

【注】*.eww、*.ewp、*.ewd、および main.c ファイルは初回のコード生成でのみ生成されますが、buildinfo.ipcf ファイルはコード生成のたびに生成されます。

3. FAA コンポーネントの追加

(1). 「スマート・コンフィグレータビュー」の「コンポーネント」ページを選択し、「コンポーネントの追加」ボタンを押下してください。

(2). 次に「ソフトウェアコンポーネントの選択」画面より「フレキシブル・アプリケーション・アクセラレータ」コンポーネントを追加してください。

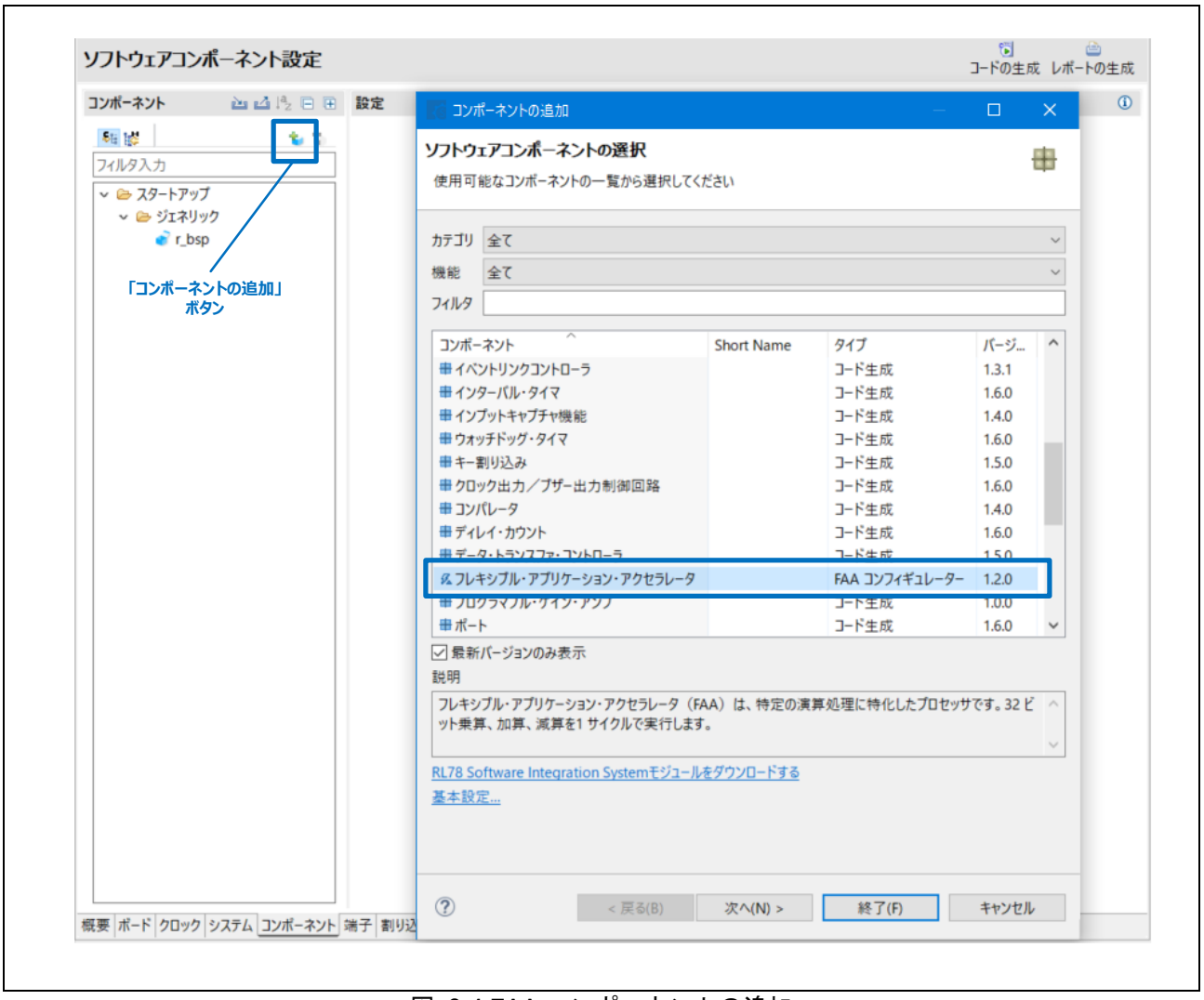

図 6-4 FAA コンポーネントの追加

4. FAA モジュールのダウンロード

画面上に表示されている「Please download FAA data」をクリックするとダウンロード可能な FAA モジュールが表示されます。「FFT Library」を選択してダウンロードしてください。

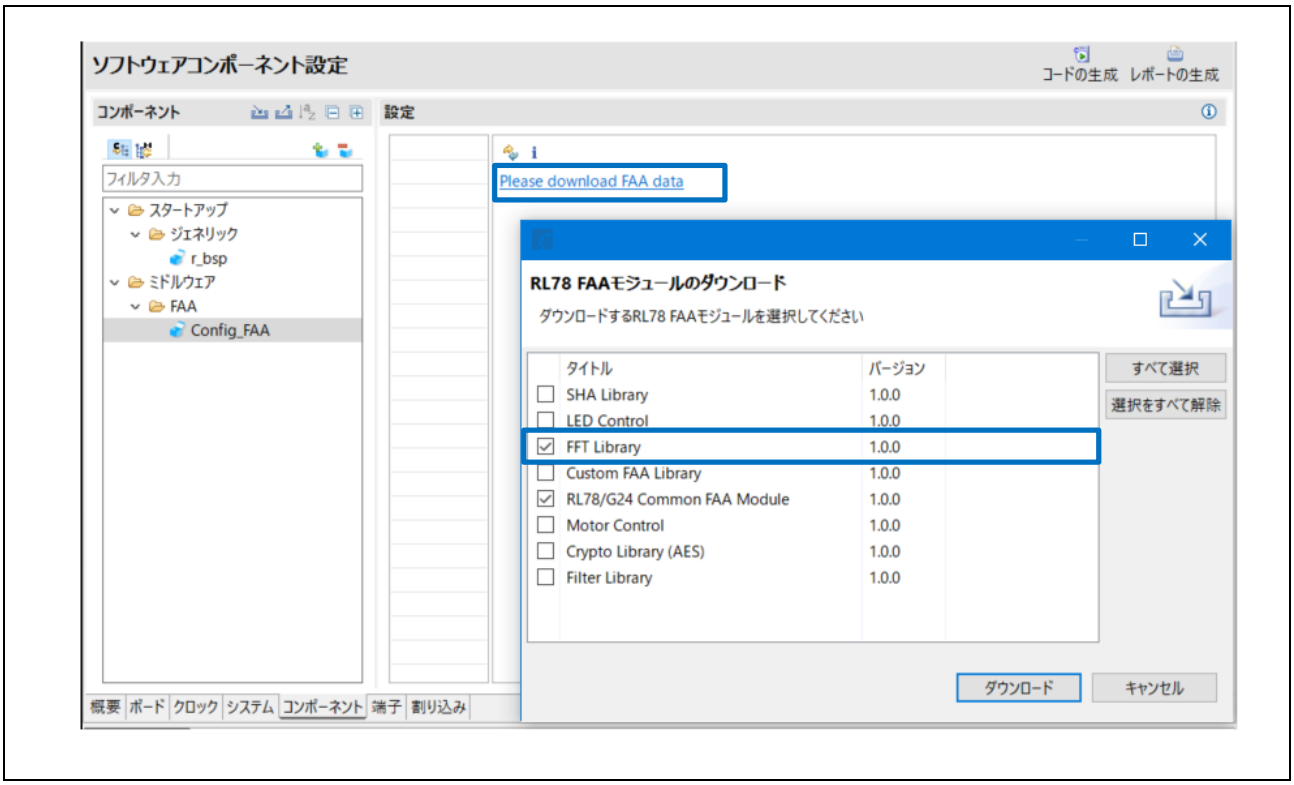

図 6-5 FAA モジュールのダウンロード

5. FAA モジュールのコンフィグレーション
ダウンロードされた FAA モジュールの一覧より「FFT」モジュールを選択すると、コンフィグレーション画面が表示されます。ユーザ環境に応じてコンフィグレーション設定を行ってください。

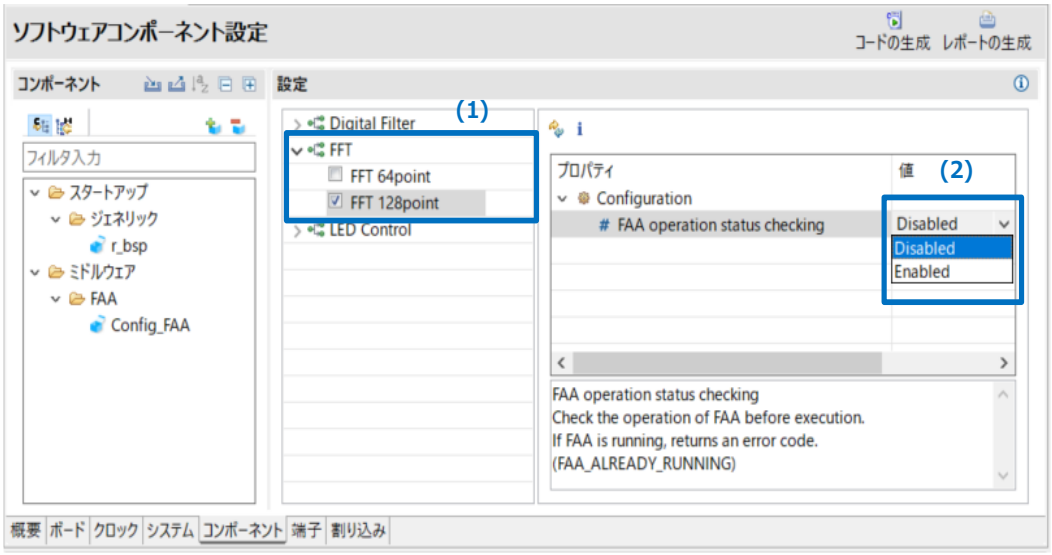

図 6-6 FAA モジュールのコンフィグレーション

(1). 使用する FFT 機能（ポイント数）を選択してください。
【注】複数選択できますが一つだけ選択してください。
(2). FAA の動作状態を確認するためのプロパティを設定できます。
設定するプロパティは以下の通りです。

表 6-6 FAA 動作状態確認のプロパティ

プロパティ	値	説明
FAA operation status checking	Enabled	API を呼び出す際に FAA 動作状態を確認し、他の関数により FAA が動作中の場合はエラーコードを返します。
	Disabled	FAA 動作状態の確認を行いません。

6. コード生成

スマート・コンフィグレータビューの 「コードの生成」 ボタンをクリックすると、設定した内容に応じたソースファイルを出力します。

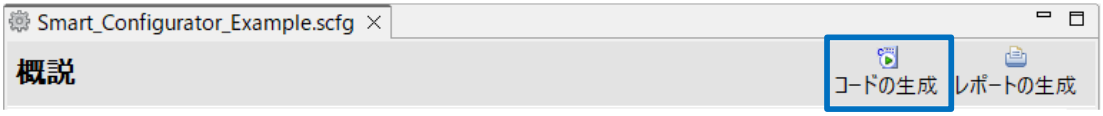

図 6-7 ソースファイルの生成

スマートコンフィグレータは、¥<ProjectDir>¥src¥smc_gen にソースファイルを生成し、IAR 関連ファイルをコンフィグファイルの保存場所「2 新規作成」に生成します。

7. IAR Embedded Workbench への読み込み

スマートコンフィグレータは、使用するコンパイラに IAR 環境を選択したとき、ソースファイルと共に IAR Embedded Workbench 関連ファイル (*.eww, *.ewp. *.ewd, main.c) を出力します。IAR Embedded Workbench でプロジェクト ファイルを作成する必要はありません。

下記の手順で使用してください。

(1). IAR Embedded Workbench の「ファイル」メニューから 「ワークスペースを開く」を選択します。

(2). 「ワークスペースを開く」ダイアログボックスで、プロジェクトファイルが保存されているフォルダを参照し、プロジェクトファイル (*.eww) を選択して 「開く」ボタンをクリックします。

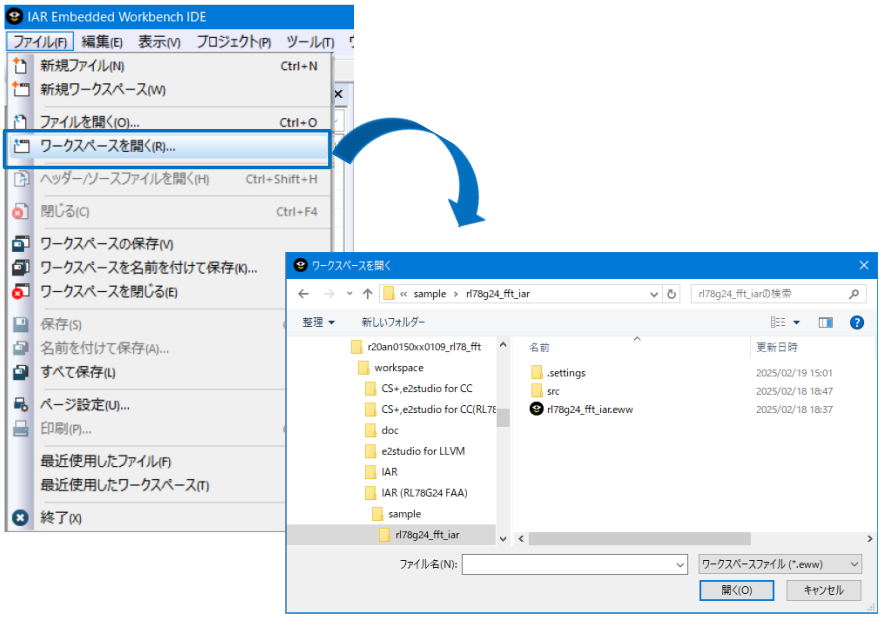

図 6-8 *.eww ファイルの読み込み

(3). スマートコンフィグレータによって出力したソースファイルは、IAR ワークスペース／プロジェクトに追加されます。

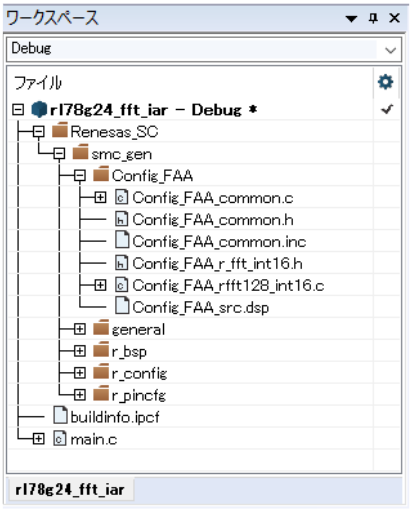

図 6-9 IAR ワークスペースの追加

(4). IAR Embedded Workbench の 「プロジェクト」 メニューから 「オプション」 を選択します。
(5). 「ノード “ProjectName” のオプション」 ダイアログボックスで、「ターゲット」タブのデバイスを対象デバイスに変更します。

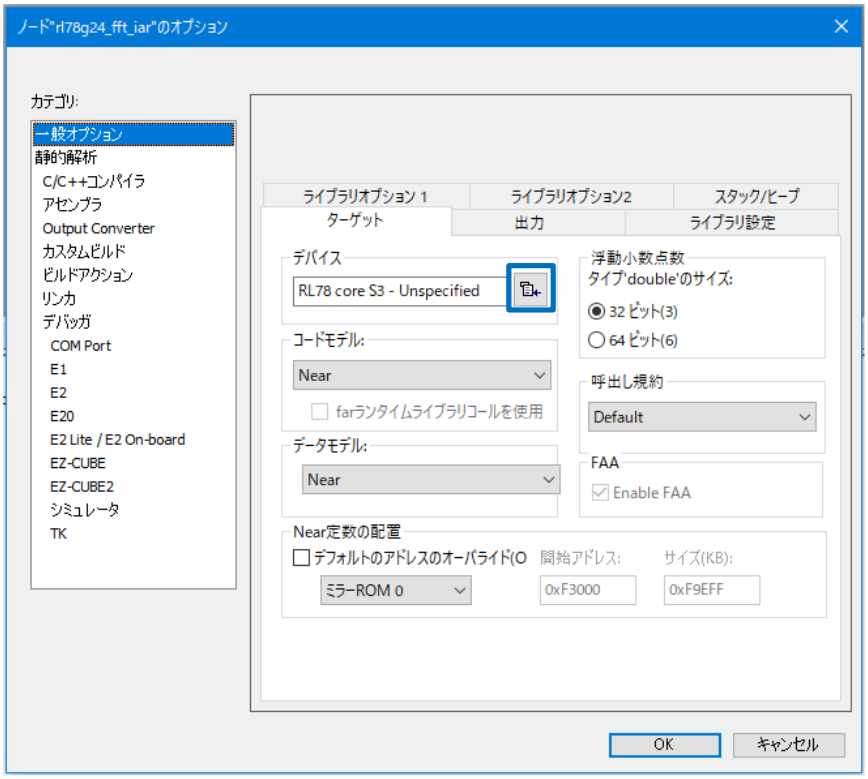

図 6-10 ターゲットデバイスの変更

6.2.2.2 プリプロセッサ・マクロ定義

RL78/G24 FAA 用 FFT ライブラリ使用時は、プロジェクトのプリプロセッサ・マクロ定義に「R_FFT_FAA」定義を追加してください。

[プロジェクト]をマウス右クリックして [オプション]を選択、表示された画面内の[C/C++コンパイラ]を選択して、 [プリプロセッサ]タブの[シンボル定義(D) : (1 行に 1 シンボル)] 欄に以下のマクロを定義して、「OK」ボタンをクリックしてください。

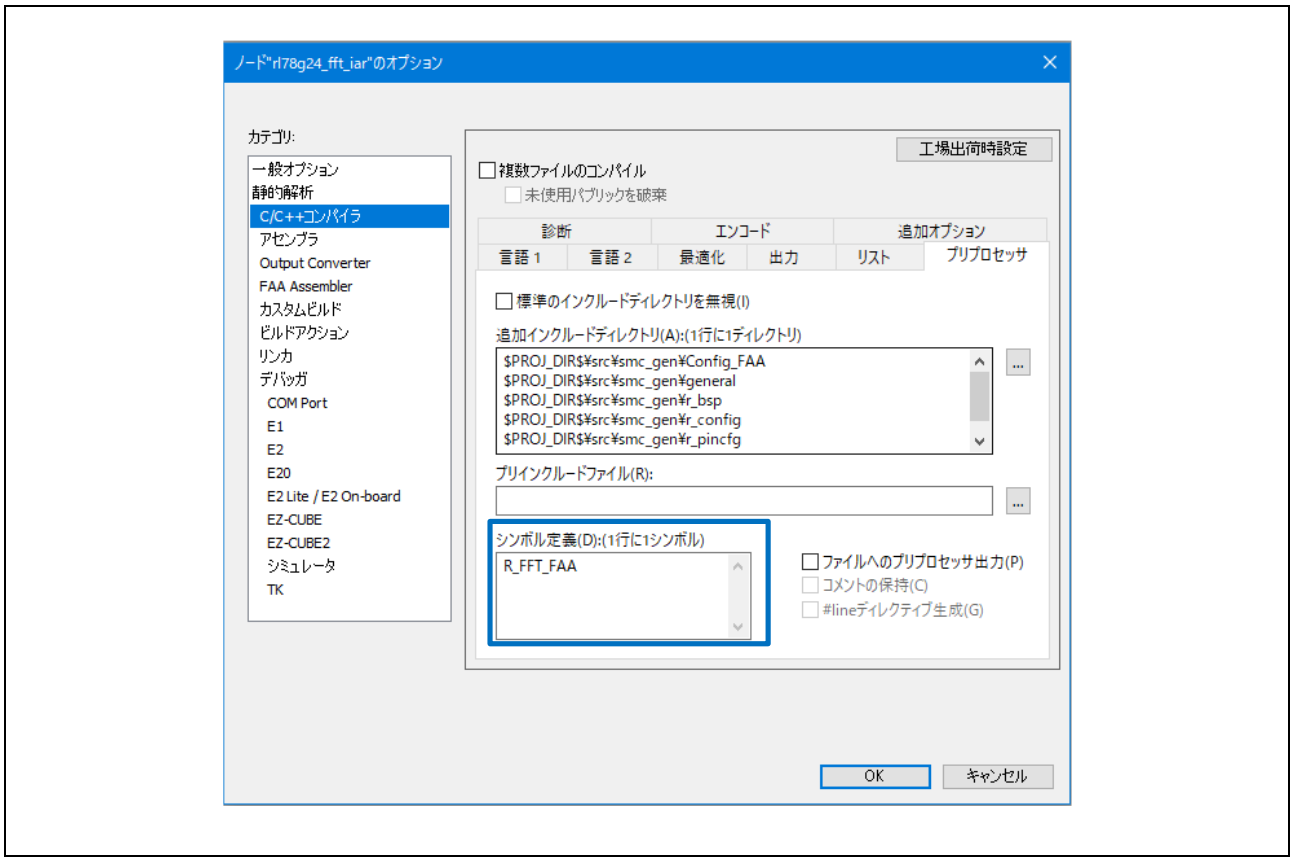

図 6-11 プリプロセッサ・マクロ定義

6.2.2.3 API に指定する work 領域

詳細については **6.1.2.3 API に指定する work 領域**をご覧ください。

6.2.2.4 API の戻り値

詳細については **6.1.2.4 API の戻り値**をご覧ください。

6.2.3 ROM / RAM / スタックサイズ

本ライブラリの ROM / RAM /スタックのサイズは次の表 6-7 とおりです（単位はバイト数）。

表 6-7 ROM / RAM /スタックのサイズ (IAR (RL78/G24 FAA 用))

API	ROM	RAM	スタック	FAACODE	FAADATA	FAA スタック
R_rfft64_int16	579	0	36	688	1432	10
R_rfft128_int16	682	0	36	688	1808	12

6.2.4 セクション情報

本ライブラリは、次の表 6-8 に示すセクション（セグメント）を使用します。

表 6-8 セクション情報 (IAR (RL78/G24 FAA 用))

セクション名	内容
.textf	プログラム
.const	定数データ
FAACODE	FAA コード領域
FAADATA	FAA データ領域

6.2.5 ライブラリ性能

以下の表 6-9 に本ライブラリのライブラリ関数（API）呼び出し毎の処理時間を示します。

表 6-9 処理時間 (IAR (RL78/G24 FAA 用))

API	Time (system clock = 48MHz)
R_rfft64_int16	約 0.2ms
R_rfft128_int16	約 0.5ms

【注】 統合開発環境(IAR Embedded Workbench for Renesas RL78)の実行時間計測機能を使用して計測

6.2.6 バージョン情報

本ライブラリはバージョン情報を示す r_fft_a_version 変数はサポートしていません。ソースのヘッダ情報を参照してください。

改訂履歴

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2012.03.31	—	初版発行
1.01	2014.04.01	—	パッケージバージョン V.1.00 Release 01 に合わせて製品構成を更新。 IAR Embedded Workbench 対応を追加。
1.02	2015.04.01	P2	パッケージバージョン V.1.00 Release 02 に合わせて製品構成を更新。
1.03	2015.10.01	—	CubeSuite+から CS+ for CA,CX に変更 CS+ for CC 対応
1.04	2021.04.13	—	CS+ for CC に RL78/G23 を追加 IAR を削除
1.05	2021.10.25	P6 P8 – P10	CS+ for CA, CX を削除 RL78/G14, RL78/G23 用 FFT ライブラリの処理時間を更新 IAR を追加
1.06	2022.06.27	P1 P22-P13	表 1　FFT ライブラリ の製品構成に「e²studio for LLVM 用」を追加 「5. e²studio for LLVM 用」を追加
1.07	2022.09.26	—	RL78/G15 用 FFT ライブラリの対応を追加
1.08	2023.04.19	—	RL78/G24 FAA 用 FFT ライブラリの対応を追加
1.09	2025.03.19	—	RL78/G24 FAA 用 FFT ライブラリに IAR 用の対応を追加。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。
3. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。
5. クロックについて
   リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。
8. 製品間の相違について
   型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

   標準水準：　コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1　2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。