RL78/G24

FAA AES ライブラリ 導入ガイド

要旨

本資料は、RL78/G24 FAA AES ライブラリ(以下 AES ライブラリ)を導入するための情報を記します。AES ライブラリは AES 暗号処理を RL78 マイコンで実現するためのソフトウェアライブラリです。AES ライブラリは RL78 マイコンを用いて効率よく処理が出来るように設計されています。

FAA(フレキシブル・アプリケーション・アクセラレータ)は、ハーバード・アーキテクチャを採用したルネサスエレクトロニクス株式会社のオリジナルのアプリケーションアクセラレータです。暗号処理に FAA を用いることで、AES ライブラリの処理速度が向上します。本ライブラリは、スマート・コンフィグレータで、暗号化と復号のいずれかを選択してコード生成を行います。詳細は「2.2 AES ライブラリ導入手順」を参照してください。

FAA の詳細は「RL78/G24 ユーザーズマニュアル ハードウェア編（R01UH0961）」 第 4 章を参照してください。

API 関数の詳細については、「RL78/G24 FAA AES ライブラリ: ユーザーズマニュアル(R20UW0176）」を参照してください。

動作確認デバイス

RL78/G24

本ソフトウェアの使用メモリ量に関しては、本資料に記載の ROM/RAM/スタックサイズの情報を参照してください。

目次

1. 製品構成

本製品は、以下の表 1-1 のファイルが含まれます。

表 1-1 AES ライブラリの製品構成

構成	内容
サンプルプログラム(r20an0691xx0101-rl78g24-aes-faa) <DIR>	
workspace <DIR>	
doc （ドキュメント） <DIR>	
en （英語版） <DIR>	
r20uw0176ej0100-rl78g24-aes-faa.pdf	ユーザーズマニュアル
r20an0691ej0101-rl78g24-aes-faa.pdf	導入ガイド
ja （日本語版） <DIR>	
r20uw0176jj0101-rl78g24-aes-faa.pdf	ユーザーズマニュアル
r20an0691jj0101-rl78g24-aes-faa.pdf	導入ガイド (本書)
CS+ <DIR>	CS+用プロジェクトフォルダ
sample_aes_rl78_faa <DIR>	G24 用サンプルプロジェクト格納用フォルダ
src <DIR>	プログラム格納用フォルダ
main.c	サンプルプログラムのソースコード
main.h	
r_sample_aes128.c	
r_sample_aes256.c	
r_test_data.c	
r_test_data.h	
smc_gen <DIR>	スマート・コンフィグレータ自動生成フォルダ
Config_FAA	FAA 関連ソースファイル格納フォルダ
General	共通ヘッダファイル・ソースファイル格納フォルダ
r_bsp	初期化コード・レジスタ定義などの格納フォルダ
r_config	ドライバ初期化コンフィグヘッダ格納フォルダ
r_pincfg	ポートのシンボリック名設定ヘッダ格納フォルダ
e^2 studio <DIR>	e^2 studio 用プロジェクトフォルダ
sample_aes_rl78_faa <DIR> 以下省略	G24 用サンプルプロジェクト格納用フォルダ 以下省略
IAR <DIR>	IAR 用プロジェクトフォルダ
sample_aes_rl78_faa <DIR> 以下省略	G24 用サンプルプロジェクト格納用フォルダ 以下省略

2. 製品仕様

2.1 API 関数

AES ライブラリは、以下の API 関数をサポートしています。

API 関数の詳細については、「RL78/G24 FAA AES ライブラリ: ユーザーズマニュアル(R20UW0176）」を参照してください。

表 2-1 AES ライブラリの API 関数

API 関数名[注 1]	内容
R_”XXX”_Aes_128_Keysch	AES 128-bit 拡大鍵生成関数
R_”XXX”_Aes_128_Ecbenc	AES 128-bit 暗号化関数 (ECB モード)
R_”XXX”_Aes_128_Ecbdec	AES 128-bit 復号関数 (ECB モード)
R_”XXX”_Aes_128_Cbcenc	AES 128-bit 暗号化関数 (CBC モード)
R_”XXX”_Aes_128_Cbcdec	AES 128-bit 復号関数 (CBC モード)
R_”XXX”_Aes_256_Keysch	AES 256-bit 拡大鍵生成関数
R_”XXX”_Aes_256_Ecbenc	AES 256-bit 暗号化関数 (ECB モード)
R_”XXX”_Aes_256_Ecbdec	AES 256-bit 復号関数 (ECB モード)
R_”XXX”_Aes_256_Cbcenc	AES 256-bit 暗号化関数 (CBC モード)
R_”XXX”_Aes_256_Cbcdec	AES 256-bit 復号関数 (CBC モード)

【注】 1. “XXX”にはコンフィグレーション名が入ります。コンフィグレーション名はスマート・コンフィグレータで FAA コンポーネントを追加する際に設定します。詳細は「2.2.1 コード生成手順」を参照してください。

2.2 AES ライブラリ導入手順

AES ライブラリは、スマート・コンフィグレータを使用してコードを生成します。コード生成手順、コード生成に関する設定について説明します。

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、以下ドキュメント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: e² studio 編 (R20AN0579)
- 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 : IAR 編 (R20AN0581)

FAA のプログラムのビルドおよびデバッガ操作については、以下ドキュメントを参照してください。

- RL78/G24 フレキシブル・アプリケーション・アクセラレータ(FAA)ツールガイド CS+編 (R01AN7094)
- RL78/G24 フレキシブル・アプリケーション・アクセラレータ(FAA) ツールガイド e2 studio 編 (R01AN7095)

2.2.1 コード生成手順

(1) スマート・コンフィグレータで[コンポーネント]ページを開き、[コンポーネントの追加]アイコンをクリックします。

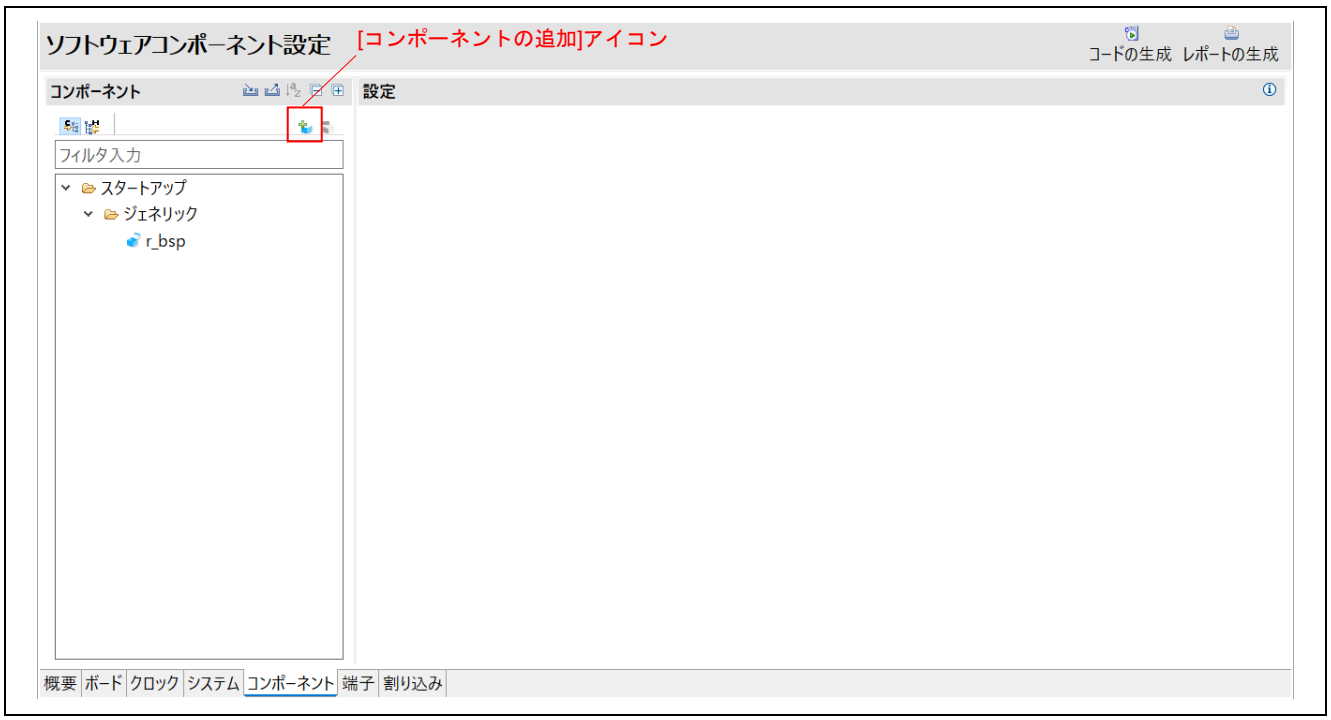

図 2-1 [コンポーネントの追加]アイコン

(2) [コンポーネントの追加]ダイアログが表示されるので、コンポーネント[フレキシブル・アプリケーション・アクセラレータ]を選択して、[次へ]をクリックします。

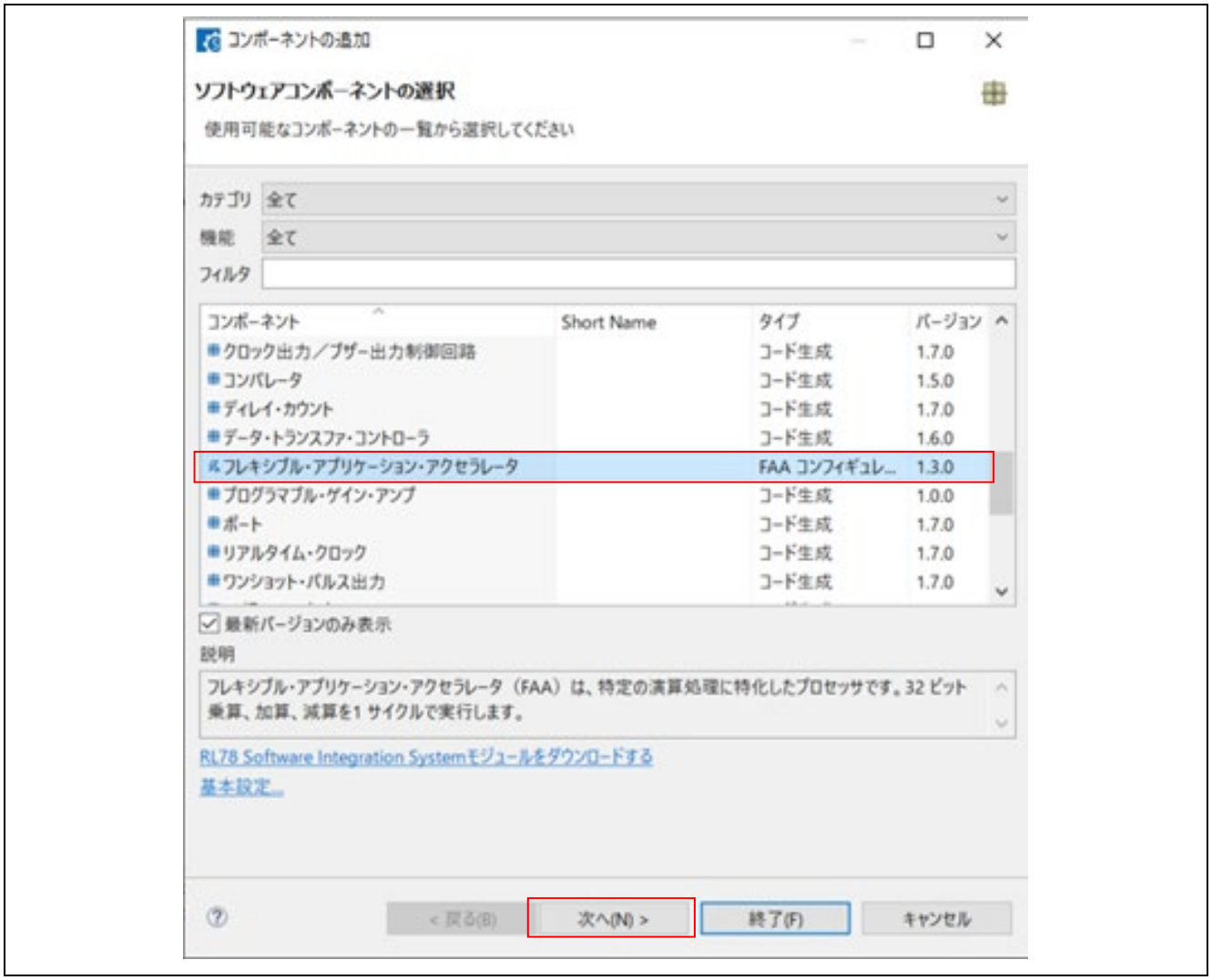

図 2-2 [コンポーネントの追加]ダイアログ

(3) [コンフィグレーション名]を入力して（または初期値を使用）[終了]をクリックします。
コンフィグレーション名の初期値は”Config_FAA”です。
設定した[コンフィグレーション名]は生成コードのファイル名および API 関数名の一部に反映されます。
生成コードのファイル名の詳細は「2.2.4 生成コード詳細」、API 関数名の詳細は「2.1 API 関数」を参照してください。

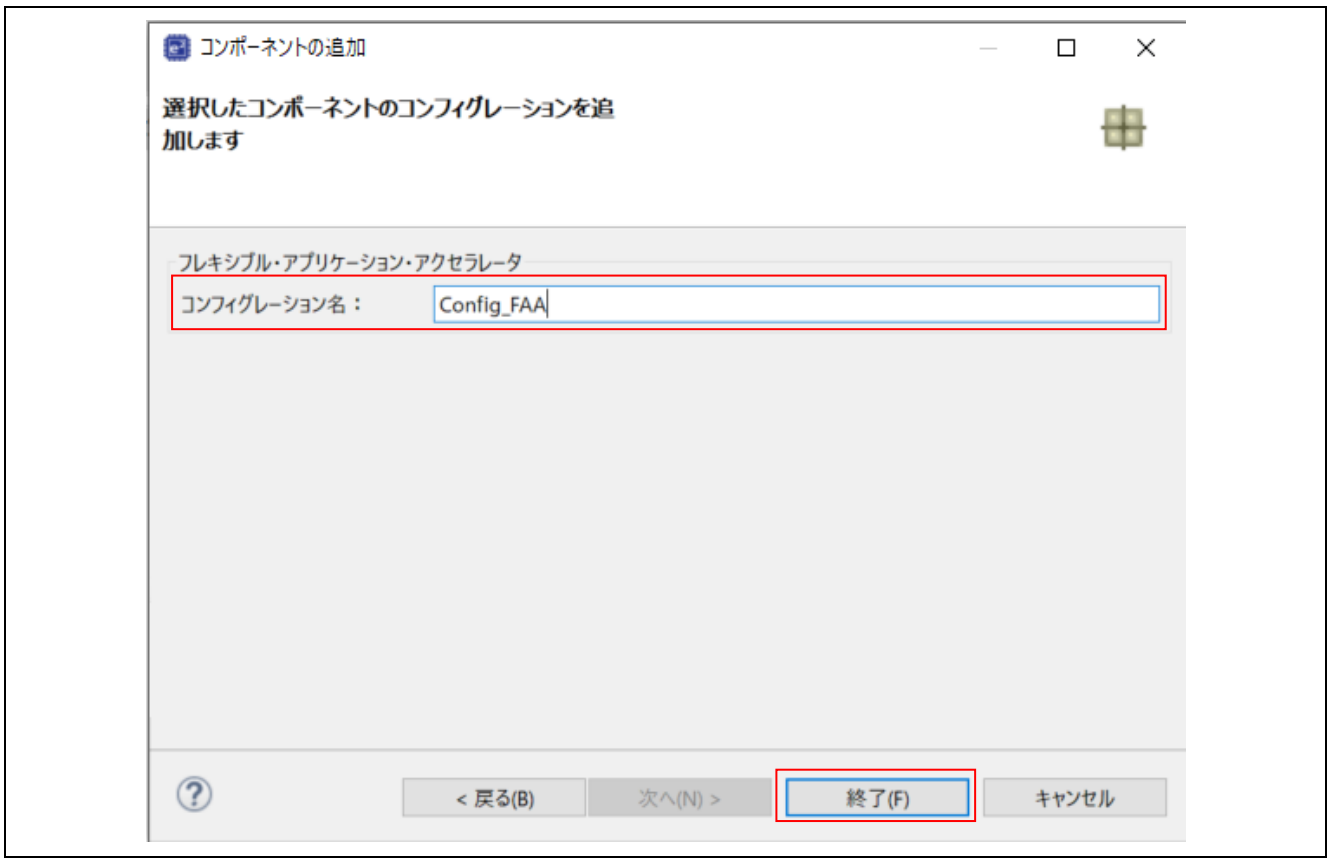

図 2-3 [コンフィグレーション名]

(4) [FAA モジュールを更新]アイコンまたは[Please download FAA data]をクリックします。

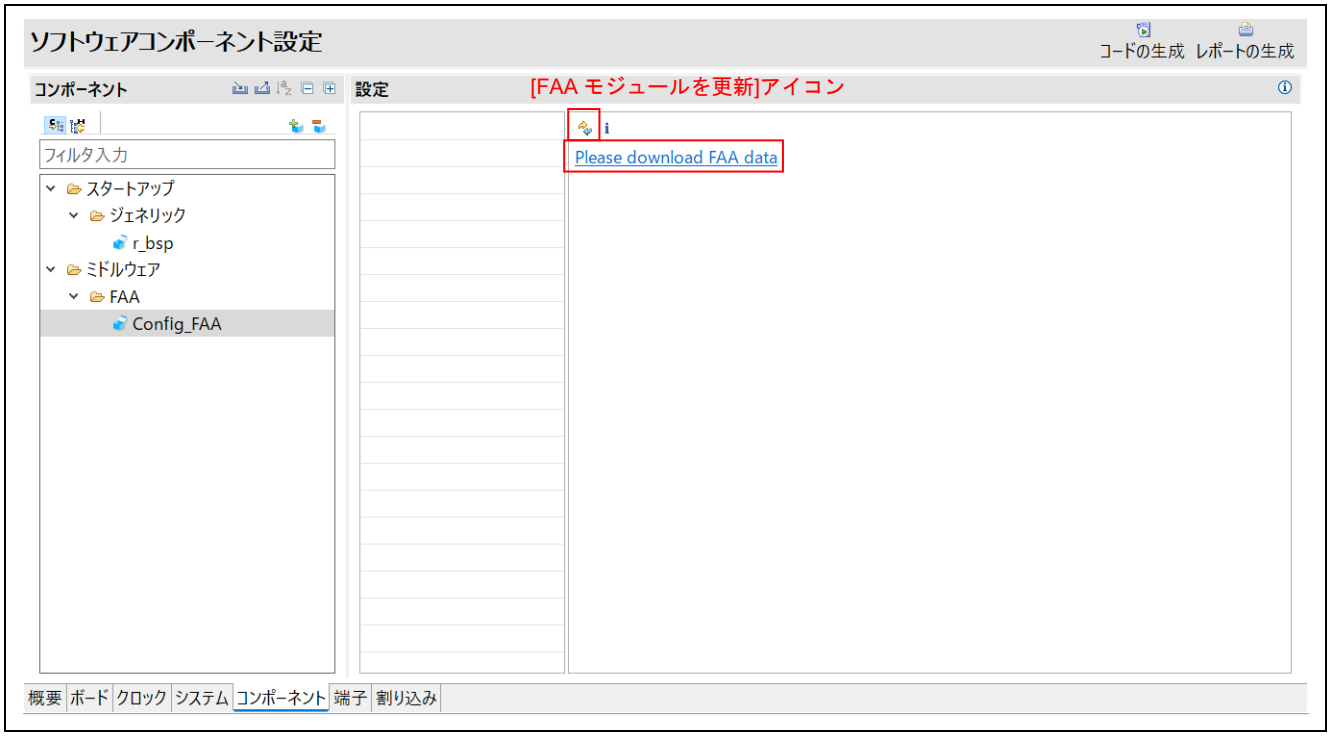

図 2-4 [FAA モジュールを更新]アイコン

(5) [Crypto Library (AES)]を選択して[ダウンロード]をクリックします。

図 2-5 FAA モジュールのダウンロード

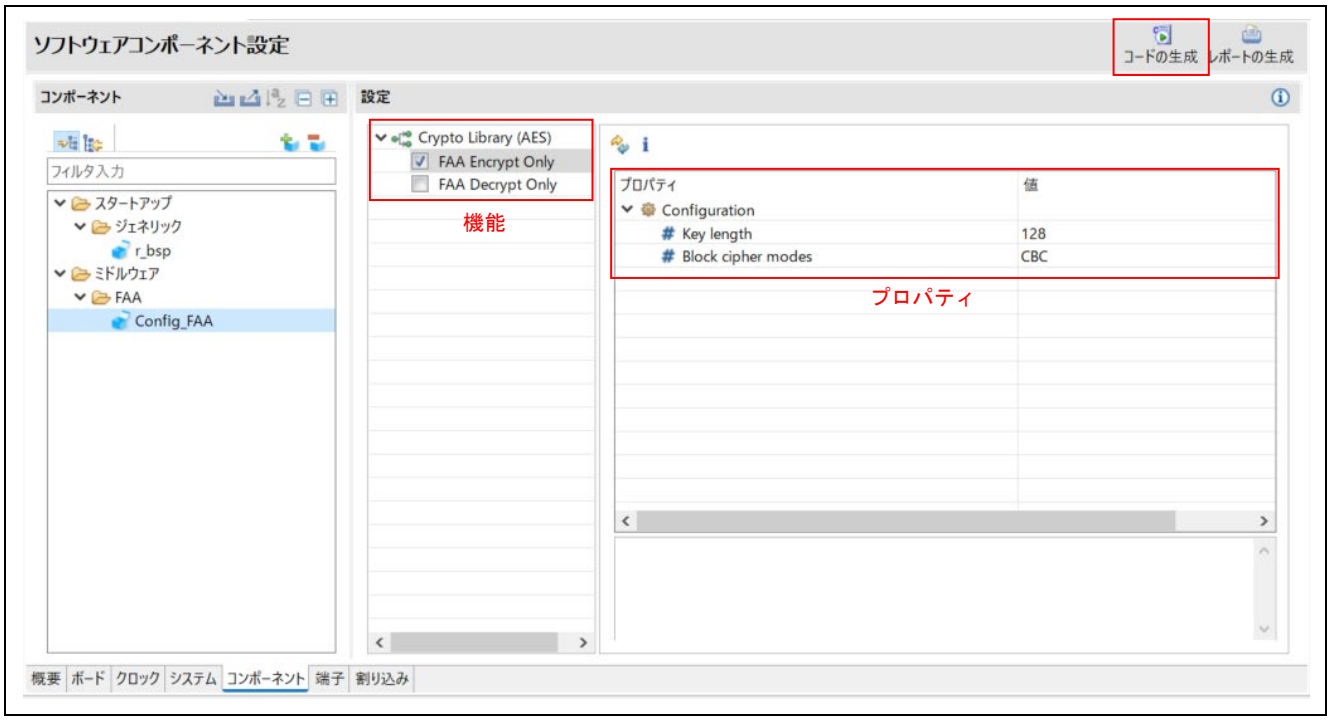

図 2-6 [コンポーネント]ページ

(6) 機能を選択します。
いずれか一つを選択してください。両方選択してコード生成した場合、FAA のデータ容量不足となり、ライブラリを使用できません。機能の詳細は「2.2.2 機能」を参照してください。

(7) プロパティで Key length、Block cipher modes を設定します。プロパティの詳細は「2.2.3 プロパティ」を参照してください。

(8) [コードの生成]ボタンをクリックして、コードを生成します。
¥src¥smc_gen¥<コンフィグレーション名> に選択した機能、プロパティに応じたコードが生成されます。
生成されるコードの詳細については「2.2.4 生成コード詳細」を参照してください。

2.2.2 機能

機能の説明を以下に示します。

表 2-2 機能

機能[注 1]	説明
FAA Encrypt Only	FAA による AES 暗号化
FAA Decrypt Only	FAA による AES 復号

【注】 1. FAA のデータ容量不足となるため、両方の機能を同時に選択することはできません。

2.2.3 プロパティ

プロパティの説明を以下に示します。

表 2-3 プロパティ

プロパティ	説明
Key length	暗号に使用する鍵長の選択(128 / 256 ビット)
Block cipher modes	ブロック暗号モードの選択(ECB / CBC / ECB+CBC)

2.2.4 生成コード詳細

表 2-4、表 2-5 はスマート・コンフィグレータで選択した機能ごとに、生成されるコードの詳細を示します。

表 2-4 機能が FAA Encrypt Only の場合に生成されるコード

ファイル名[注1]	説明
“XXX”_AesEnc.c	FAA 暗号化 C ソースファイル
“XXX”_AesEnc.h	FAA 暗号化ヘッダファイル
“XXX”_common.c	FAA 共通関数 C ソースファイル
“XXX”_common.h	FAA 共通関数ヘッダファイル
“XXX”_common.inc	FAA 用 iodefine ヘッダファイル
“XXX”_r_aes_version.c	AES ライブラリのバージョン情報定義
“XXX”_r_aeskey.c	AES 鍵生成 C ソースファイル
“XXX”_r_mw_version.h	バージョン情報ヘッダファイル
“XXX”_r_stdint.h	型定義ヘッダファイル
“XXX”_src.dsp	FAA 暗号化アセンブラファイル

【注】 1. “XXX”にはコンフィグレーション名が入ります。コンフィグレーション名はスマート・コンフィグレータで FAA コンポーネントを追加する際に設定します。詳細は「2.2.1 コード生成手順」を参照してください。

表 2-5 機能が FAA Decrypt Only の場合に生成されるコード

ファイル名[注1]	説明
“XXX”_AesDec.c	FAA 復号 C ソースファイル
“XXX”_AesDec.h	FAA 復号ヘッダファイル
“XXX”_common.c	FAA 共通関数 C ソースファイル
“XXX”_common.h	FAA 共通関数ヘッダファイル
“XXX”_common.inc	FAA 用 iodefine ヘッダファイル
“XXX”_r_aes_version.c	AES ライブラリのバージョン情報定義
“XXX”_r_aeskey.c	AES 鍵生成 C ソースファイル
“XXX”_r_mw_version.h	バージョン情報ヘッダファイル
“XXX”_r_stdint.h	型定義ヘッダファイル
“XXX”_src.dsp	FAA 復号アセンブラファイル

【注】 1. “XXX”にはコンフィグレーション名が入ります。コンフィグレーション名はスマート・コンフィグレータで FAA コンポーネントを追加する際に設定します。詳細は「2.2.1 コード生成手順」を参照してください。

2.2.5 スマート・コンフィグレータの設定ごとに使用可能な API 関数

表 2-6、表 2-7 はスマート・コンフィグレータで選択した機能、鍵長、ブロック暗号モードごとに、使用可能な API 関数を示します。

表 2-6 機能が FAA Encrypt Only の場合に使用可能な API 関数

API 関数名[注 1]	鍵長					
	128 ビット			256 ビット		
	ブロック暗号モード					
	ECB	CBC	ECB+CBC	ECB	CBC	ECB+CBC
R_”XXX”_Aes_128_Keysch	○	○	○	×	×	×
R_”XXX”_Aes_128_Ecbenc	○	×	○	×	×	×
R_”XXX”_Aes_128_Ecbdec	×	×	×	×	×	×
R_”XXX”_Aes_128_Cbcenc	×	○	○	×	×	×
R_”XXX”_Aes_128_Cbcdec	×	×	×	×	×	×
R_”XXX”_Aes_256_Keysch	×	×	×	○	○	○
R_”XXX”_Aes_256_Ecbenc	×	×	×	○	×	○
R_”XXX”_Aes_256_Ecbdec	×	×	×	×	×	×
R_”XXX”_Aes_256_Cbcenc	×	×	×	×	○	○
R_”XXX”_Aes_256_Cbcdec	×	×	×	×	×	×

○=使用可能
×=使用不可

【注】 1. “XXX”にはコンフィグレーション名が入ります。コンフィグレーション名はスマート・コンフィグレータで FAA コンポーネントを追加する際に設定します。詳細は「2.2.1 コード生成手順」を参照してください。

表 2-7 機能が FAA Decrypt Only の場合に使用可能な API 関数

API 関数名[注 1]	鍵長					
	128 ビット			256 ビット		
	ブロック暗号モード					
	ECB	CBC	ECB+CBC	ECB	CBC	ECB+CBC
R_”XXX”_Aes_128_Keysch	○	○	○	×	×	×
R_”XXX”_Aes_128_Ecbenc	×	×	×	×	×	×
R_”XXX”_Aes_128_Ecbdec	○	×	○	×	×	×
R_”XXX”_Aes_128_Cbcenc	×	×	×	×	×	×
R_”XXX”_Aes_128_Cbcdec	×	○	○	×	×	×
R_”XXX”_Aes_256_Keysch	×	×	×	○	○	○
R_”XXX”_Aes_256_Ecbenc	×	×	×	×	×	×
R_”XXX”_Aes_256_Ecbdec	×	×	×	○	×	○
R_”XXX”_Aes_256_Cbcenc	×	×	×	×	×	×
R_”XXX”_Aes_256_Cbcdec	×	×	×	×	○	○

○=使用可能
×=使用不可

【注】 1. “XXX”にはコンフィグレーション名が入ります。コンフィグレーション名はスマート・コンフィグレータで FAA コンポーネントを追加する際に設定します。詳細は「2.2.1 コード生成手順」を参照してください。

2.3 サンプルプログラム

サンプルプログラムは暗号化もしくは復号の処理を行い、期待値と一致するかを確認します。

以下のスマート・コンフィグレータの設定でコードを生成しています。

機能：FAA Encrypt Only

鍵長：128 ビット

ブロック暗号モード：ECB+CBC

機能、鍵長、ブロック暗号モードを変更したい場合は、スマート・コンフィグレータの設定を変更してコードを生成してください。コードの生成方法については、「2.2 AES ライブラリ導入手順」を参照してください。

サンプルプログラムを参考に API 関数を使用する部分のプログラムを作成してください。

3. CC-RL (C コンパイラ)

3.1 開発環境

本製品の開発で使用したツールのバージョンは以下になります。

ユーザアプリケーション開発時は最新のバージョンをご使用下さい。

- 統合開発環境
  e^2 studio 2023-07
  CS+ for CC V8.10.00

- C コンパイラ
  CC-RL V1.12.01

- DSP アセンブラ
  FAA Assembler V1.04.02

3.2 ROM / RAM / Stack サイズおよび処理時間

以下のオプションを使用してビルドした際の API 関数の各種サイズや処理時間を参考として記します。

- コンパイラオプション
  -cpu=S3 -memory_model=medium –Odefault

- リンクオプション
  -NOOPtimize

コード生成時のスマート・コンフィグレータの設定により、使用可能な API 関数は異なります。

詳細は「2.2.5 スマート・コンフィグレータの設定ごとに使用可能な API 関数」を参照してください。

表 3-1 は API 関数の各種サイズを示します。

各種サイズとは、CPU 上の ROM / RAM / Stack と FAA 上の FAACODE / FAADATA を指します。

表 3-1　API 関数の ROM / RAM / Stack / FAACODE / FAADATA サイズ [byte]

API 関数名[注1]	ROM[注2]	ROM[注3]	RAM	Stack	FAACODE[注2]	FAACODE[注3]	FAADATA
R_”XXX”_Aes_128_Keysch	2,880		0	26	—[注4]		
R_”XXX”_Aes_128_Ecbenc	143	317		30	1,056	1,164	1,972
R_”XXX”_Aes_128_Cbcenc	174			36	1,100		
R_”XXX”_Aes_128_Ecbdec	143	317		58	1,120	1,228	1,972
R_”XXX”_Aes_128_Cbcdec	174			64	1,160		
R_”XXX”_Aes_256_Keysch	3,325			28	—[注4]		
R_”XXX”_Aes_256_Ecbenc	143	317		30	1,104	1,212	1,972
R_”XXX”_Aes_256_Cbcenc	174			36	1,148		
R_”XXX”_Aes_256_Ecbdec	143	317		58	1,168	1,276	1,972
R_”XXX”_Aes_256_Cbcdec	174			64	1,208		

【注】 1. “XXX”にはコンフィグレーション名が入ります。コンフィグレーション名はスマート・コンフィグレータで FAA コンポーネントを追加する際に設定します。詳細は「2.2.1 コード生成手順」を参照してください。
2. ブロック暗号モードが ECB または CBC の場合の値です。
3. ブロック暗号モードが ECB＋CBC の場合の値です。
4. 拡大鍵生成関数は FAA を使用しません。

表 3-2 は API 関数の処理時間を示します。

表 3-2　API 関数の処理時間

API 関数名[注1]	time [us] @ system clock = 32MHz	
	1 block	3 blocks
R_”XXX”_Aes_128_Keysch	80	
R_”XXX”_Aes_128_Ecbenc	170	380
R_”XXX”_Aes_128_Ecbdec	340	590
R_”XXX”_Aes_128_Cbcenc	180	400
R_”XXX”_Aes_128_Cbcdec	350	600
R_”XXX”_Aes_256_Keysch	90	
R_”XXX”_Aes_256_Ecbenc	220	510
R_”XXX”_Aes_256_Ecbdec	460	790
R_”XXX”_Aes_256_Cbcenc	230	530
R_”XXX”_Aes_256_Cbcdec	470	810

【注】 1. “XXX”にはコンフィグレーション名が入ります。コンフィグレーション名はスマート・コンフィグレータで FAA コンポーネントを追加する際に設定します。詳細は「2.2.1 コード生成手順」を参照してください。

4. IAR C/C++ Compiler for Renesas RL78 (C コンパイラ)

4.1 開発環境

本製品の開発で使用したツールのバージョンは以下になります。

ユーザアプリケーション開発時は最新のバージョンをご使用下さい。

- 統合開発環境
  IAR Embedded Workbench for Renesas RL78 version 5.20.1

- C コンパイラ
  IAR C/C++ Compiler for Renesas RL78 version 5.20.1.2826

- DSP アセンブラ
  FAA/GREEN_DSP Structured Assembler version 1.05.00.01 (1.5.0.1)

4.2 ROM / RAM / Stack サイズおよび処理時間

以下のオプションを使用してビルドした際の API 関数の各種サイズや処理時間を参考として記します。

- コンパイラオプション
  --core=S3 --code_model=far --data_model=near --near_const_location=rom0 -e -Oh --calling_convention=v2

コード生成時のスマート・コンフィグレータの設定により、使用可能な API 関数は異なります。

詳細は「2.2.5 スマート・コンフィグレータの設定ごとに使用可能な API 関数」を参照してください。

表 4-1 は API 関数の各種サイズを示します。

各種サイズとは、CPU 上の ROM / RAM / Stack と FAA 上の FAACODE / FAADATA を指します。

表 4-1　API 関数の ROM / RAM / Stack / FAACODE / FAADATA サイズ [byte]

API 関数名[注1]	ROM[注2]	ROM[注3]	RAM	Stack	FAACODE[注2]	FAACODE[注3]	FAADATA
R_”XXX”_Aes_128_Keysch	2,925		0	48	—[注4]		
R_”XXX”_Aes_128_Ecbenc	147	317		22	1,056	1,164	1,972
R_”XXX”_Aes_128_Cbcenc	170			24	1,100		
R_”XXX”_Aes_128_Ecbdec	147	317		72	1,120	1,228	1,972
R_”XXX”_Aes_128_Cbcdec	170			74	1,160		
R_”XXX”_Aes_256_Keysch	3,307			48	—[注4]		
R_”XXX”_Aes_256_Ecbenc	147	317		22	1,104	1,212	1,972
R_”XXX”_Aes_256_Cbcenc	170			24	1,148		
R_”XXX”_Aes_256_Ecbdec	147	317		72	1,168	1,276	1,972
R_”XXX”_Aes_256_Cbcdec	170			74	1,208		

【注】 1. “XXX”にはコンフィグレーション名が入ります。コンフィグレーション名はスマート・コンフィグレータで FAA コンポーネントを追加する際に設定します。詳細は「2.2.1 コード生成手順」を参照してください。
2. ブロック暗号モードが ECB または CBC の場合の値です。
3. ブロック暗号モードが ECB＋CBC の場合の値です。
4. 拡大鍵生成関数は FAA を使用しません。

表 4-2 は API 関数の処理時間を示します。

表 4-2　API 関数の処理時間

API 関数名[注1]	time [us] @ system clock = 32MHz	
	1 block	3 blocks
R_”XXX”_Aes_128_Keysch	300	
R_”XXX”_Aes_128_Ecbenc	150	370
R_”XXX”_Aes_128_Ecbdec	1,270	1,520
R_”XXX”_Aes_128_Cbcenc	160	380
R_”XXX”_Aes_128_Cbcdec	1,280	1,530
R_”XXX”_Aes_256_Keysch	380	
R_”XXX”_Aes_256_Ecbenc	200	490
R_”XXX”_Aes_256_Ecbdec	1,820	2,150
R_”XXX”_Aes_256_Cbcenc	210	510
R_”XXX”_Aes_256_Cbcdec	1,830	2,160

【注】 1. “XXX”にはコンフィグレーション名が入ります。コンフィグレーション名はスマート・コンフィグレータで FAA コンポーネントを追加する際に設定します。詳細は「2.2.1 コード生成手順」を参照してください。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2023.08.01	—	初版発行
1.01	2025.08.01		IAR Embedded Workbench に対応しました。
		5	2.2 手順を修正しました。
		16	4 章を追加しました。

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。