User's Manual

RZ/T2M グループ

Encoder I/F Configuration Library ユーザーズマニュアル

ルネサスマイクロコンピュータ
RZファミリ／RZ/Tシリーズ

ルネサス エレクトロニクス
www.renesas.com

Rev.4.00　2026.02

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。

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。

このマニュアルの使い方

1.　目的と対象者

　このマニュアルはライブラリソフトウェア「Encoder I/F Configuration Library」の機能、使用方法をユーザに理解していただくためのマニュアルです。本ライブラリソフトウェアを用いた応用システムを設計するユーザを対象にしています。このマニュアルを使用するには、プログラミング言語、マイクロコンピュータに関する基本的な知識が必要です。

本マニュアルは、注意事項を十分確認の上、使用してください。注意事項は、各章の本文中、各章の最後、注意事項の章に記載しています。

改訂記録は旧版の記載内容に対して訂正または追加した主な箇所をまとめたものです。改訂内容すべてを記録したものではありません。詳細は、このマニュアルの本文でご確認ください。

目次

1. はじめに

1.1 要旨

本書は Encoder I/F Configuration Library (以下 EC-Lib)をコントロールするための API に関して説明しています。

1.2 機能

EC-Lib は RZ/T2M に搭載された Encoder I/F のコンフィグレーション、起動、および、停止を行うソフトウェアライブラリです。

1.3 ソフトウェア構成図

図 1.1 に EC-Lib のソフトウェア構成を示します。

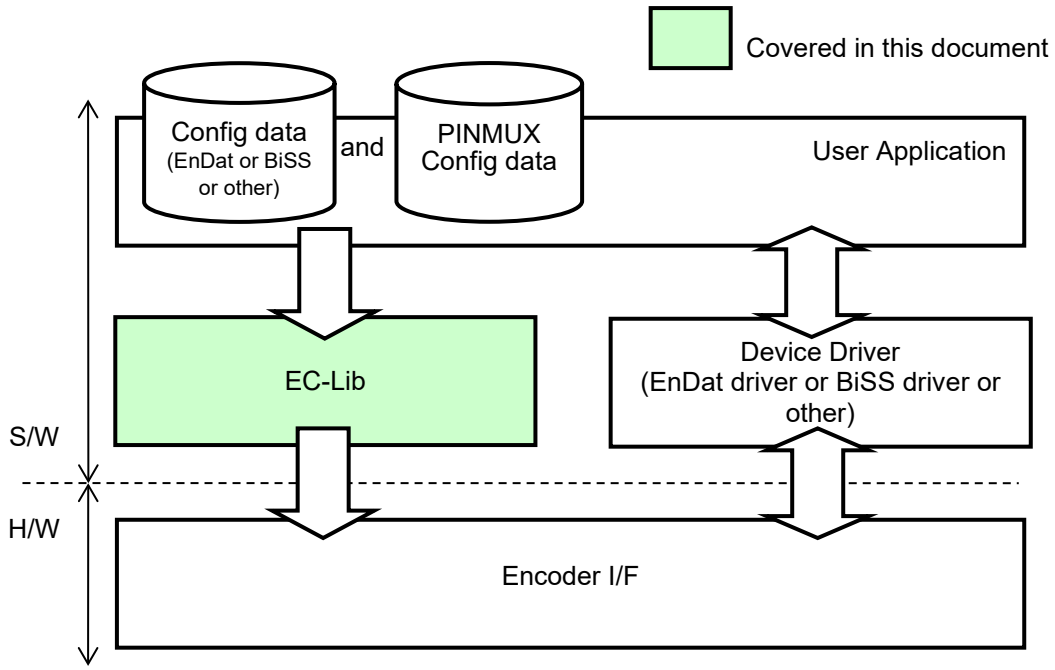


図 1.1 ソフトウェア構成図 (EC-Lib とその関連モジュール)

2. 動作条件

本書で説明するライブラリは下記の条件で動作します。

表 2.1 動作確認条件

項目	内容	
マイコン	RZ/T2M (Cortex®-R52)	
動作周波数	800 MHz	
動作電圧	1.1V (Core) / 1.8V (PLL, etc.) / 3.3V (I/O)	
開発環境	IAR 版	IAR Embedded Workbench® for Arm®
	GCC 版	RENESAS e^2 studio GNU Arm Embedded Toolchain

3. ファイル構成

図 3.1 に EC-Lib のファイル構成を示します。ヘッダファイルは IAR 版、GCC 版で共通です。ライブラリファイルは、名前で IAR 版と GCC 版とが区別されています。

```
lib
    ecl
        r_ecl_rz2_if.h          EC-Lib ヘッダファイル(IAR, GCC 共通)
        r_ecl_rzt2_iar.a        EC-Lib ライブラリファイル(IAR 版)
        r_ecl_rzt2_gcc.a        EC-Lib ライブラリファイル(GCC 版)
```

図 3.1 ファイル構成

4. EC-Lib API 仕様

4.1 API 一覧

表 4.1 に制御 API の一覧を示します。

表 4.1 EC-Lib API 一覧

API 名	内容	ページ
R_ECL_Initialize	Encoder I/F の初期化	8
R_ECL_Terminate	Encoder I/F の無効化	9
R_ECL_ConfigurePin	PINMUX のコンフィグレーション	10
R_ECL_Configure	Encoder I/F のコンフィグレーション	11
R_ECL_UnConfigure	Encoder I/F のアンコンフィグレーション	13
R_ECL_Start	Encoder I/F の起動	14
R_ECL_Stop	Encoder I/F の停止	15
R_ECL_GetVersion	EC-Lib のバージョン取得	16

4.2 使用しているデータタイプ

EC-Lib API では新たなデータタイプを定義しません。
IAR 版、GCC 版ともに、標準ライブラリで定義されている uint8_t、int32_t、uint32_t を使用します。

4.3 使用している構造体

EC-Lib API では構造体は使用しません。

4.4 各種値定義(マクロ)

EC-Lib API で使用するマクロ定義値は、それぞれ使用関数の解説内で示します。エラーコード用マクロは「4.6 エラーコード」を参照してください。

4.5 各種値定義(列挙型)

EC-Lib API では列挙型は使用しません。

4.6 エラーコード

EC-Lib APIで返されるエラーコードは、0が正常終了で負の値がエラーとなります。表 4.2にエラーコードの一覧を示します。

表 4.2 エラーコード一覧

マクロ名	値	内容
R_ECL_SUCCESS	0	正常終了
R_ECL_ERR_ARG	−1	引数エラー
R_ECL_ERR_FORMAT	−2	フォーマット不正
R_ECL_ERR_BUSY	−5	ビジー
R_ECL_ERR_STATUS	−9	ステータスエラー

5. API リファレンス

5.1 API リファレンスの読み方

API 名	分類
機能概要	同期/非同期関数

書式	API の呼出し形式を示します。#include “ヘッダファイル”で示すヘッダファイルは、この API の実行に必要な標準ヘッダファイルです。必ずインクルードしてください。 I,O は、引数がそれぞれ入力データ、出力データであることを意味します。IO の場合は入出力データであることを意味します。
戻り値	API の戻り値を示します。戻り値の後に「：」を付けて記載されているコメントは、その戻り値についての説明(リターン条件等)です。
解説	API の仕様について説明します。
注意	注意事項があればここに示します。
使用例	API の使用例があればここに示します。

5.2 R_ECL_Initialize

R_ECL_Initialize

EC-Lib API

Encoder I/F の初期化

同期関数

書式

```
#include "r_ecl_rzt2_if.h"
int32_t R_ECL_Initialize(void)
```

戻り値

R_ECL_SUCCESS	:	成功
R_ECL_ERR_STATUS	:	失敗（Encoder I/F が初期化済み）

解説

Encoder I/F の初期設定を行います。
本関数は、Encoder I/F の内部変数を初期化して、Encoder I/F を使用できる状態にします。
また、Encoder I/F を低消費電力モードから復帰してクロック供給を開始し、ハードウェアを初期化します。

注意

Encoder I/F を使用する前に必ず本関数を実行してください。

使用例

```
#include "r_ecl_rzt2_if.h"
int32_t result;

result = R_ECL_Initialize ();
if (result == R_ECL_SUCCESS)
{
    /* 通常処理 */
}
else
{
    /* エラー処理 */
}
```

5.3 R_ECL_Terminate

R_ECL_Terminate

EC-Lib API

Encoder I/F の無効化

同期関数

書式

```
#include "r_ecl_rzt2_if.h"
int32_t R_ECL_Terminate (void)
```

戻り値

R_ECL_SUCCESS	:	成功
R_ECL_ERR_STATUS	:	失敗（Encoder I/F が無効化状態、Encoder I/F にコンフィグレーションデータがロードされている）

解説

本関数は、Encoder I/F へのクロック供給を停止し、Encoder I/F を低消費電力モードへ移行します。
本関数実行後は、再度、R_ECL_Initialize 関数をコールするまで、Encoder I/F は使用できません。

注意

Encoder I/F にコンフィグレーションデータをロードした場合、本関数を使用する前に、
R_ECL_UnConfigure 関数でコンフィグレーションデータをアンロードして下さい。

使用例

```
#include "r_ecl_rzt2_if.h"
int32_t result;

result = R_ECL_Terminate ();
if (result == R_ECL_SUCCESS)
{
    /* 通常処理 */
}
else
{
    /* エラー処理 */
}
```

5.4 R_ECL_ConfigurePin

R_ECL_ConfigurePin

EC-Lib API

PINMUX のコンフィグレーション

同期関数

書式

```
#include "r_ecl_rzt2_if.h"
int32_t R_ECL_ConfigurePin(const void *const pconfig1)
```

pconfig1	I	PINMUX コンフィグレーションデータの先頭アドレス データは 8 バイト境界に配置してください。

戻り値

R_ECL_SUCCESS	:	成功
R_ECL_ERR_ARG	:	失敗（引数 pconfig1 が NULL、）
R_ECL_ERR_FORMAT	:	失敗（コンフィグレーションデータのフォーマットが不正）
R_ECL_ERR_BUSY	:	失敗（指定したチャネルが Encoder I/F のコンフィグレーション済み）
R_ECL_ERR_STATUS	:	失敗（Encoder I/F が無効化状態）

解説

本関数は、Encoder I/F の PINMUX コンフィグレーションデータをロードします。
PINMUX を 1 度コンフィグレーションした後で、本関数をもう一度使用すると PINMUX の設定が新しく指定したコンフィグレーションデータで上書きされます。

注意

本関数は、Encoder I/F がコンフィグレーションされていると使用できません。Encoder I/F を既にコンフィグレーションしている場合、本関数を使用する前に、R_ECL_UnConfigure 関数で Encoder I/F からコンフィグレーションデータをアンロードして下さい。

戻り値が R_ECL_FORMAT となった場合は、引数 pconfig1 で指定したアドレスが、正しいコンフィグレーションデータのアドレスとなっているか確認して下さい。

引数 pconfig1 で指定したコンフィグレーションデータが LSI のキャッシュ上などに存在し、物理メモリの内容がコンフィグレーションデータと一致しない状態になっていると正常にロードできません。本 API 関数コール前にキャッシュのクリーンを行う。または、コンフィグレーションデータを非キャッシュ領域に配置するなどの対処を行う必要があります。

使用例

```
#include "r_ecl_rzt2_if.h"
extern const uint32_t pinmux_config[];
int32_t result;

result = R_ECL_Configure(pinmux_config);
if (result == R_ECL_SUCCESS)
{
    /* 通常処理 */
}
else
{
    /* エラー処理 */
}
```

5.5 R_ECL_Configure

R_ECL_Configure

EC-Lib API

Encoder I/F のコンフィグレーション

同期関数

書式

```
#include "r_ecl_rzt2_if.h"
int32_t R_ECL_Configure(const uint8_t id, const void *const pconfig2)
```

id	I	エンコーダチャネル番号 チャネル 0 の場合は R_ECL_CH_0(r_ecl_rzt2_if.h で定義) チャネル 1 の場合は R_ECL_CH_1(r_ecl_rzt2_if.h で定義) チャネルを 2 つ使用するコンフィグレーションデータをロードする場合は、R_ECL_CH_0 を設定して下さい。 チャネルの複数選択はできません。
pconfig2	I	コンフィグレーションデータの先頭アドレス データは 32 バイト境界に配置してください。

戻り値

R_ECL_SUCCESS	:	成功
R_ECL_ERR_ARG	:	失敗（引数 id が無効な値、引数 id で複数のチャネルが設定、引数 pconfig2 が NULL、チャネルを 2 つ使用するコンフィグレーションデータのロードで引数 id に R_ECL_CH_1 を指定）
R_ECL_ERR_FORMAT	:	失敗（コンフィグレーションデータのフォーマットが不正）
R_ECL_ERR_BUSY	:	失敗（指定したチャネルがコンフィグレーション済み）
R_ECL_ERR_STATUS	:	失敗（Encoder I/F が無効化状態）

解説

本関数は、Encoder I/F へコンフィグレーションデータをロードする関数です。
本関数では、引数 id にチャネル 0 かチャネル 1 のどちらか一方のみを設定して下さい。
本関数は、引数 pconfig で指定したコンフィグレーションデータによって Encoder I/F を設定します。

注意

本関数実行後は Encoder I/F は停止状態となります。本関数実行後、Encoder I/F を使用する前に、R_ECL_Start 関数で Encoder I/F を起動してください。

本関数の実行前に必ず R_ECL_Initialize 関数、R_ECL_ConfigurePin 関数の順番で Encoder I/F の初期化を行ってください。

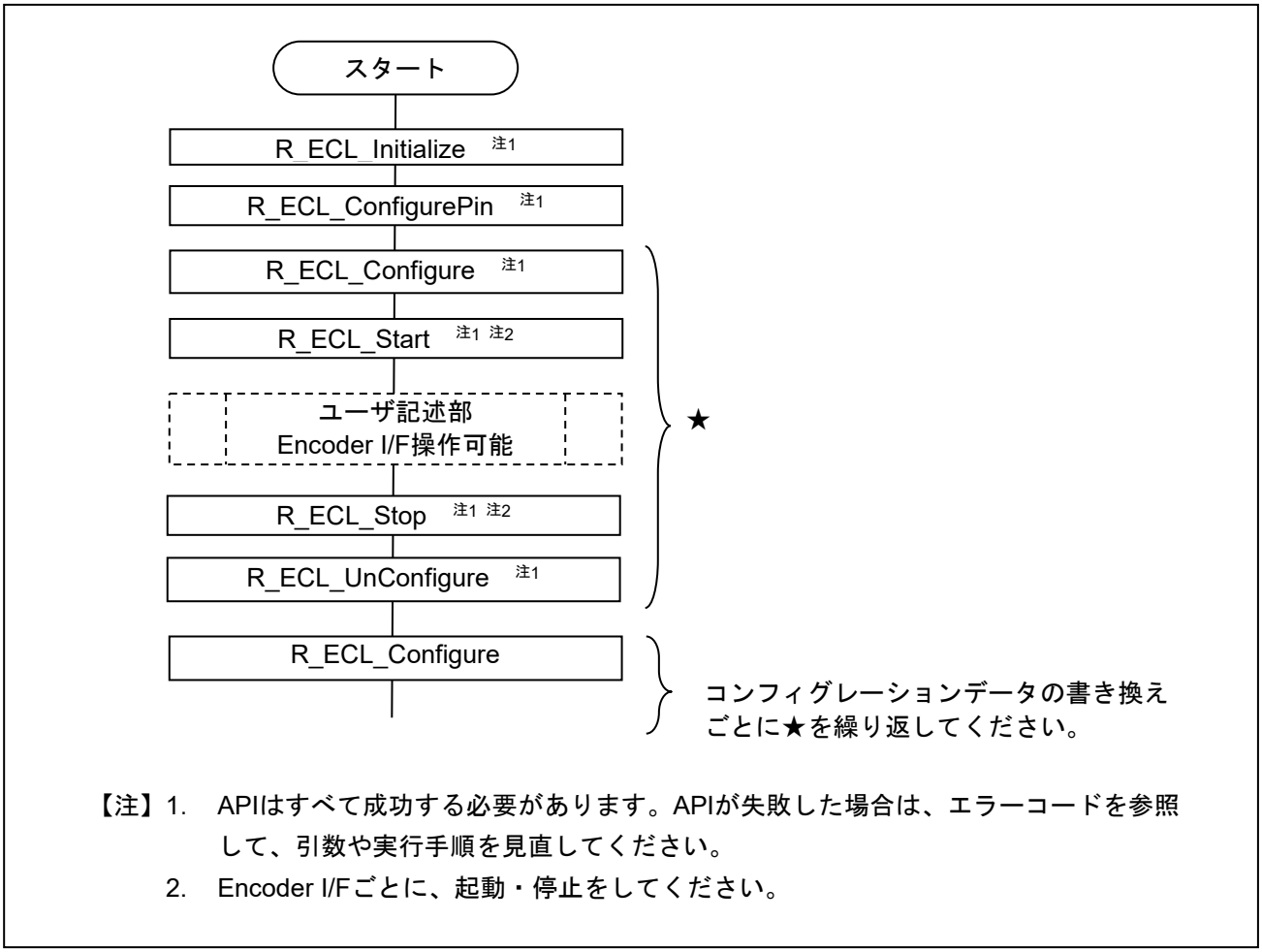
戻り値が R_ECL_FORMAT となった場合は、引数 pconfig2 で指定したアドレスが、正しいコンフィグレーションデータのアドレスとなっているか確認して下さい。

引数 pconfig2 で指定したコンフィグレーションデータが LSI のキャッシュ上などに存在し、物理メモリの内容がコンフィグレーションデータと一致しない状態になっていると正常にロードできません。本 API 関数コール前にキャッシュのクリーンを行う。または、コンフィグレーションデータを非キャッシュ領域に配置するなどの対処を行う必要があります。

使用例

```
#include "r_ecl_rzt2_if.h"
extern const uint32_t config_data[];
int32_t result;

result = R_ECL_Configure(R_ECL_CH_0,config_data);
if (result == R_ECL_SUCCESS)
{
    /* 通常処理 */
}
else
{
    /* エラー処理 */
}
```



【注】1. APIはすべて成功する必要があります。APIが失敗した場合は、エラーコードを参照して、引数や実行手順を見直してください。
2. Encoder I/Fごとに、起動・停止をしてください。

図 5.1　コンフィグレーションデータの書き換え手順

5.6 R_ECL_UnConfigure

R_ECL_UnConfigure

EC-Lib API

Encoder I/F のアンロード

同期関数

書式

```
#include "r_ecl_rzt2_if.h"
int32_t R_ECL_UnConfigure(const uint8_t id)
```

id	I	エンコーダチャネル番号。 チャネル 0 の場合は R_ECL_CH_0(r_ecl_rzt2_if.h で定義) チャネル 1 の場合は R_ECL_CH_1(r_ecl_rzt2_if.h で定義) チャネルを 2 つ使用するコンフィグレーションデータの場合は、R_ECL_CH_0 を設定して下さい。 チャネルの複数選択が可能です。チャネル 1,2 を同時にアンロードする場合は、R_ECL_CH_0 と R_ECL_CH_1 を論理和で指定して下さい。

戻り値

R_ECL_SUCCESS	:	成功
R_ECL_ERR_ARG	:	失敗（引数 id が無効な値、チャネルを 2 つ使用するコンフィグレーションデータのアンロードで引数 id に R_ECL_CH_1 を指定）
R_ECL_ERR_STATUS	:	失敗（指定したチャネルにコンフィグレーションデータがロードされていない）

解説

本関数は、Encoder I/F の各チャネルのコンフィグレーションデータをアンロードする関数です。
Encoder I/F が動作していた場合、Encoder I/F を停止してからアンロードを行います。

注意

チャネル 0,1 のどちらか一方、または両方のチャネルからコンフィグレーションデータが既にアンロードされている状態で、引数 id にチャネル 0 と 1 を指定して本関数を使用した場合、
R_ECL_ERR_STATUS エラーを返します。

使用例

```
#include "r_ecl_rzt2_if.h"
int32_t result;

result = R_ECL_UnConfigure(R_ECL_CH_0);
if (result == R_ECL_SUCCESS)
{
    /* 通常処理 */
}
else
{
    /* エラー処理 */
}
```

5.7 R_ECL_Start

R_ECL_Start

EC-Lib API

Encoder I/F の起動

同期関数

書式

```
#include "r_ecl_rzt2_if.h"
int32_t R_ECL_Start(const uint8_t id, const uint32_t freq)
```

id	I	エンコーダチャネル番号。 チャネル 0 の場合は R_ECL_CH_0(r_ecl_rzt2_if.h で定義) チャネル 1 の場合は R_ECL_CH_1(r_ecl_rzt2_if.h で定義) チャネルを 2 つ使用するコンフィグレーションデータの場合は、R_ECL_CH_0 を設定して下さい。 チャネルの複数選択が可能です。チャネル 1,2 を同時に起動する場合は、R_ECL_CH_0 と R_ECL_CH_1 を論理和で指定して下さい。
freq	I	動作周波数を kHz 単位で設定して下さい。10MHz の場合は、10000 となります。関数で設定できる動作周波数の範囲は、1562(1.562MHz) ～50000(50MHz)です。引数にはこの範囲を超えない値かつコンフィグレーションデータの最大動作周波数を超えない値を設定して下さい。

戻り値

R_ECL_SUCCESS	:	成功
R_ECL_ERR_ARG	:	失敗（引数 id が無効な値、チャネルを 2 つ使用するコンフィグレーションデータをロードした Encoder I/F の起動で引数 id に R_ECL_CH_1 を指定、引数 freq が不正）
R_ECL_ERR_STATUS	:	失敗（引数 id で指定したチャネルにコンフィグレーションデータがロードされていない、指定したチャネルが起動済み）

解説

本関数は、Encoder I/F の各チャネルの起動を行う関数です。

注意

Encoder I/F を使用する前に必ず本関数を実行してください。

本関数の実行前に必ず R_ECL_Initialize 関数、R_ECL_ConfigurePin 関数、R_ECL_Configure 関数の順番で Encoder I/F の初期化を行ってください。

チャネル 0,1 のどちらか一方、または両方のチャネルが既に起動している状態で、引数 id にチャネル 0 と 1 を指定して本関数を使用した場合、R_ECL_ERR_STATUS エラーを返します。

使用例

```
#include "r_ecl_rzt2_if.h"
#include "r_endat_rzt2_dat.h"

int32_t result;

result = R_ECL_Start(R_ECL_CH_0, R_ENDAT_FREQ);
if (result == R_ECL_SUCCESS)
{
    /* 通常処理 */
}
else
{
    /* エラー処理 */
}
```

5.8 R_ECL_Stop

R_ECL_Stop

EC-Lib API

Encoder I/F の停止

同期関数

書式

```
#include "r_ecl_rzt2_if.h"
int32_t R_ECL_Stop(const int32_t id)
```

id	I	エンコーダチャネル番号。 チャネル 0 の場合は R_ECL_CH_0(r_ecl_rzt2_if.h で定義) チャネル 1 の場合は R_ECL_CH_1(r_ecl_rzt2_if.h で定義) チャネルを 2 つ使用するコンフィグレーションデータの場合は、R_ECL_CH_0 を設定して下さい。 チャネルの複数選択が可能です。チャネル 1,2 を同時に停止する場合は、R_ECL_CH_0 と R_ECL_CH_1 を論理和で指定して下さい。

戻り値

R_ECL_SUCCESS	:	成功
R_ECL_ERR_ARG	:	失敗（引数 id が無効な値、チャネルを 2 つ使用するコンフィグデータをロードした Encoder I/F の停止で引数 id に R_ECL_CH_1 を指定）
R_ECL_ERR_STATUS	:	失敗（引数 id で指定したチャネルにコンフィグレーションデータがロードされていない、指定したチャネルが停止状態）

解説

本関数は、Encoder I/F の動作を停止する関数です。

注意

チャネル 0,1 のどちらか一方、または両方のチャネルが既に停止している状態で、引数 id にチャネル 0 と 1 を指定して本関数を使用した場合、R_ECL_ERR_STATUS エラーを返します。

使用例

```
#include "r_ecl_rzt2_if.h"
int32_t result;

result = R_ECL_Stop(R_ECL_CH_0);
if (result == R_ECL_SUCCESS)
{
    /* 通常処理 */
}
else
{
    /* エラー処理 */
}
```

5.9 R_ECL_GetVersion

R_ECL_GetVersion

EC-Lib API

EC-Lib のバージョン取得

同期関数

書式

```
#include "r_ecl_rzt2_if.h"
uint32_t R_ECL_GetVersion(void)
```

戻り値

バージョン　：

b31	b24	b23	b16	b15	b8	b7	b0
予約領域		メジャーバージョン		マイナーバージョン		ビルド番号	

ビット 7～0：ビルド番号が格納されます。

ビット 15～8：マイナーバージョンが格納されます。

ビット 23～16：メジャーバージョンが格納されます。

ビット 31～24：予約領域です。0 が格納されます。

例）
戻り値が 0x00010002 の場合、Ver.1.02
戻り値が 0x00020305 の場合、Ver.2.35

解説

Encoder I/F Configuration Library のバージョンを取得します。

使用例

```
uint32_t ver;

uint8_t major_ver;
uint8_t minor_ver;
uint8_t build_no;

ver = R_ECL_GetVersion();

major_ver = (uint8_t)((ver >> 16) & 0xFF);
minor_ver = (uint8_t)((ver >> 8) & 0xFF);
build_no = (uint8_t)(ver & 0xFF);

printf("EC-Lib Ver.%d.%d.%d.", major_ver, minor_ver, build_no);
```

6. リソース

6.1 H/W

EC-Lib が占有する H/W リソースはありません。

6.2 OS

EC-Lib は OS を利用しません。

6.3 メモリ

表 6.1にEC-Libが使用するメモリのセクション名とサイズの概算を示します。正確なメモリサイズはリリースノートを参照してください。

表 6.1 メモリリソース一覧

カテゴリ	セクション名	サイズ(概算)
Code	.text 注1	5.5k バイト注2
RO Data	.rodata 注1	0.1k バイト以下
RW Data	.data 注1	0.3k バイト注2
ZI Data	.bss 注1	0.1k バイト以下
R_ECL_Initialize 関数スタック	―	0.1k バイト以下
R_ECL_ConfigurePin 関数スタック	―	0.1k バイト以下
R_ECL_Configure 関数スタック	―	0.2k バイト
R_ECL_UnConfigure 関数スタック	―	0.1k バイト
R_ECL_Start 関数スタック	―	0.1k バイト
R_ECL_Stop 関数スタック	―	0.1k バイト以下
R_ECL_Terminate 関数スタック	―	0.1k バイト以下
R_ECL_GetVersion 関数スタック	―	0

【注】 1. 固有のセクション名を付加しません。
2. バージョンによって異なります。

7.　フローチャート

図 7.1 に EC-Lib の制御例としてフローチャートを示します。

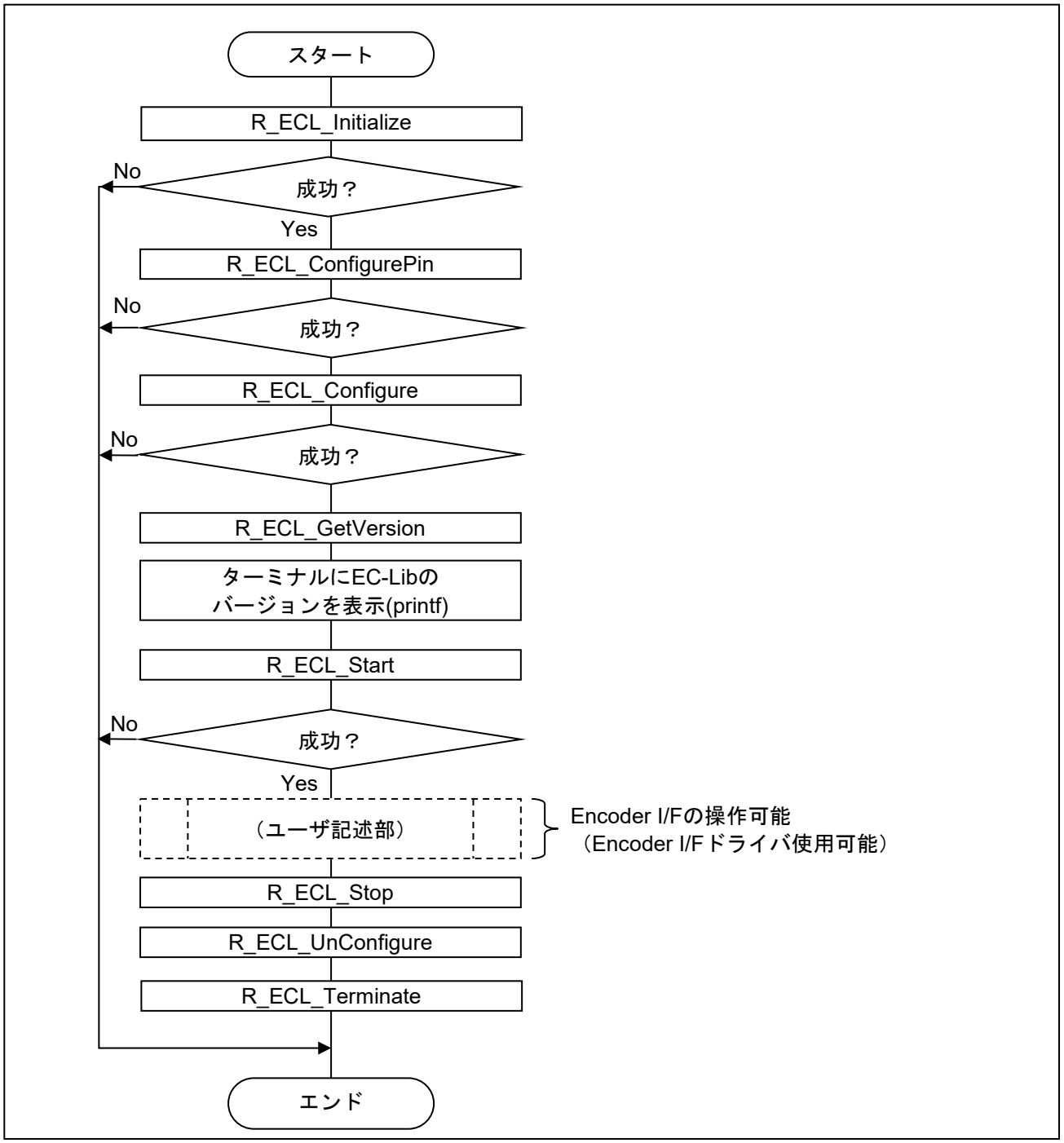


図 7.1　EC-Lib 制御例フローチャート

改訂記録	RZ/T2M グループ Encoder I/F Configuration Library ユーザーズマニュアル

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Apr.08.22	ー	初版発行
2.00	Jun 07.24	3	動作確認条件の記載内容を更新。
3.00	Oct 10.25	3	動作確認条件および商標の表記を更新。
		5	固定幅整数の定義場所に関する記載を更新。
		10, 11	データの 4 バイト境界配置に関する記載を追加。
		15	R_ECL_Stop 関数の解説の表記を更新。
4.00	Feb 27.26	10, 11	データの配置に関する記載を更新。
		17	固有のセクション名を使用しないように変更。サイズ情報を更新。

RZ/T2Mグループ
Encoder I/F Configuration Library ユーザーズマニュアル

発行年月日　2026年02月27日　　Rev.4.00

発行　　ルネサス エレクトロニクス株式会社
　　〒135-0061　東京都江東区豊洲3-2-24（豊洲フォレシア）

RZ/T2M グループ

ルネサス エレクトロニクス株式会社

R01US0563JJ0400