
RZ/T2M グループ

ENCOUT サンプルプログラム

要旨

本アプリケーションノートでは、RZ/T2M の Encoder I/F Configuration Library (以下 EC-Lib) を使用し、指定した位置情報をもとに、A 相, B 相, および Z 相 (インデックス) 出力信号を発生するサンプルプログラムについて説明します。

プログラムの特徴を以下に示します。

- ・ 位置情報を周期的に更新することにより、A 相, B 相, および Z 相の出力を生成
- ・ タイマモジュールイベントを入力し、イベントによって出力エッジを補正

動作確認デバイス

RZ/T2M

目次

1. 仕様	3
2. 動作環境	4
3. 周辺機能説明	5
3.1 使用端子一覧	5
4. ソフトウェア説明	6
4.1 ENCOUT ドライバ機能	6
4.2 ファイル構成	6
4.3 関数一覧	6
4.4 API 関数仕様	7
4.4.1 R_ENCOUT_Open	7
4.4.2 R_ENCOUT_Close	7
4.4.3 R_ENCOUT_GetVersion	7
4.4.4 R_ENCOUT_Control	8
4.4.5 制御コマンド	8
4.5 定数/エラーコード一覧	10
4.6 固定幅整数一覧	12
4.7 構造体/共用体/列挙型一覧	13
4.7.1 構造体	13
4.7.2 共用体	13
4.7.3 列挙型	13
4.8 サンプルプログラムの説明	14
4.8.1 動作概要	14
4.8.2 サンプルプログラム関数一覧	16
4.8.3 サンプルプログラム関数仕様	16
4.8.4 サンプルプログラムの変数一覧	19
4.8.5 サンプルプログラムの定数一覧	19
4.8.6 メイン処理のフローチャート	20
4.8.7 動作シーケンス	22
4.8.8 コンソールコマンド	24
4.8.9 サンプルプログラム実行結果	24
5. サンプルコード	25
改訂記録	26

1. 仕様

表 1-1 に使用する周辺機能と用途を、図 1-1 にサンプルコード実行時の動作環境を示します。

表 1-1 使用する周辺機能と用途

周辺機能	用途
エンコーダ分周出力(ENCOUT)	位置情報に応じて A 相, B 相, Z 相信号を出力
割り込みコントローラ(ICU)	GPT チャンネル 0 の割り込み制御
汎用 PWM タイマ(GPT) チャンネル 0	ELC に入力する周期的タイマイベントの生成と、タイマ割り込みの生成
イベントリンクコントローラ (ELC)	GPT チャンネル 0 が出力するイベントと ENCOUT をリンク
シリアル通信インターフェース(SCI) UART	SCI の調歩同期式 I/F を使用し、USB インターフェースによる COM ポート通信に使用 サンプルプログラムのコンソールインターフェース用

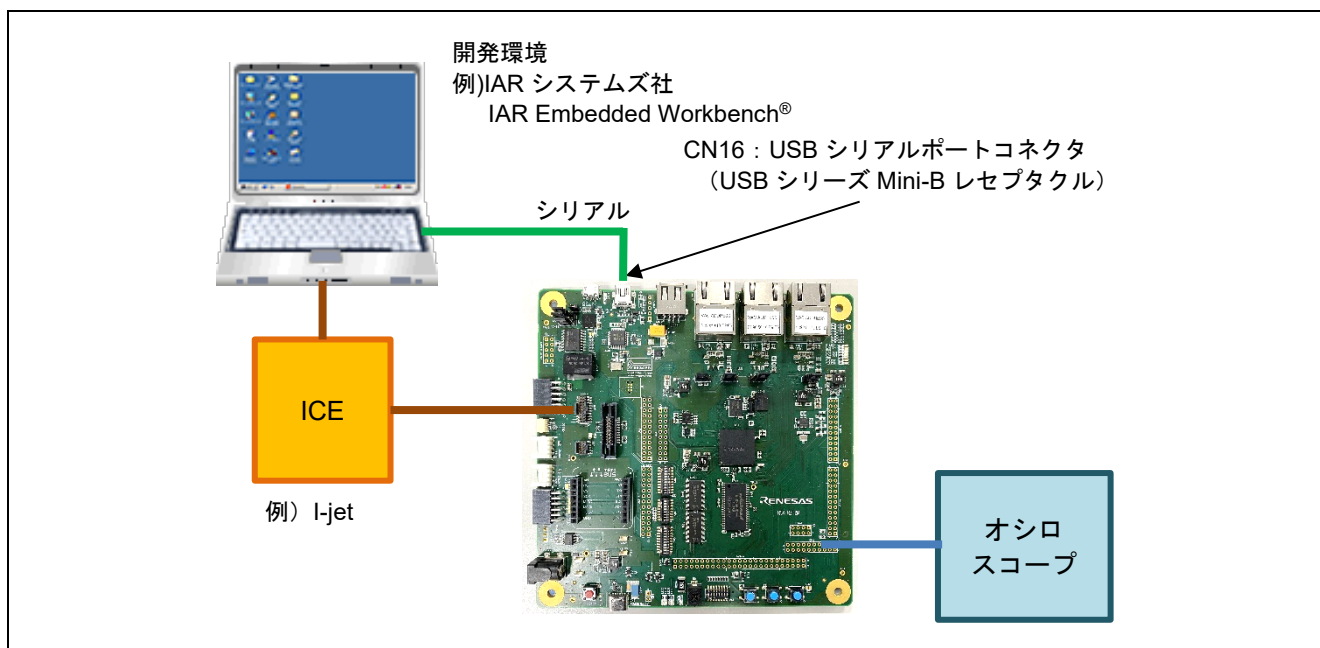


図 1-1 動作環境

2. 動作環境

本アプリケーションノートのサンプルコードは、下記の環境を想定しています。

表 2-1 動作環境

項目	内容
使用マイコン	RZ/T2M グループ
動作周波数	CPUCLK = 800MHz
動作電圧	1.1V (Core) / 1.8V (PLL, etc.) / 3.3V (I/O)
統合開発環境 ^{注1}	IAR システムズ製 IAR Embedded Workbench® for Arm® RENESAS 製 e² studio
使用ボード	RSK+RZT2M (RTK9RZT2M0C00000BE)
使用デバイス (ボード上で使用する機能)	なし

【注】 1. 統合開発環境のバージョンは、RZ/T2M グループ ENCOUT sample program リリースノートを参照してください。

3. 周辺機能説明

周辺機能、動作モード、レジスタについての基本的な内容は、RZ/T2M グループ・ユーザズマニュアルハードウェア編に記載しています。

3.1 使用端子一覧

表 3-1 に使用端子と機能を示します。

表 3-1 使用端子と機能

チャンネル	端子名 (機能ピン名)	I/O ポート	入出力	内容
ENCOUT0	POUTA0 (ENCIF2)	P02_0	出力	A 相出力端子
	POUTB0 (ENCIF3)	P02_2	出力	B 相出力端子
	POUTZ0 (ENCIF4)	P02_3	出力	Z 相出力端子
ENCOUT1	POUTA1 (ENCIF7)	P17_5	出力	A 相出力端子
	POUTB1 (ENCIF8)	P03_0	出力	B 相出力端子
	POUTZ1 (ENCIF9)	P03_3	出力	Z 相出力端子

4. ソフトウェア説明

4.1 ENCOUT ドライバ機能

ENCOUT ドライバの機能は以下です。

1. 初期設定
2. 位置情報の設定
3. A相, B相, Z相信号の出力

4.2 ファイル構成

ファイル構成は、RZ/T2M グループ ENCOUT sample program リリースノートを参照してください。

4.3 関数一覧

表 4-1 に関数を示します。

表 4-1 関数一覧

カテゴリ	関数名	ページ番号
ENCOUT ドライバAPI 関数	R_ENCOUT_Open	7
	R_ENCOUT_Close	7
	R_ENCOUT_GetVersion	7
	R_ENCOUT_Control	8

4.4 API 関数仕様

4.4.1 R_ENCOUT_Open

R_ENCOUT_Open	
概要	ENCOUT ドライバの初期化
ヘッダ	r_encout_rzt2_if.h
宣言	r_encout_err_t R_ENCOUT_Open(const int32_t id);
説明	ENCOUT ドライバの初期化を行います。 ENCOUT ドライバを使用する前に必ず本関数をコールしてください。
引数	id : ENCOUT_ID_0 あるいは ENCOUT_ID_1 を指定してください
リターン値	R_ENCOUT_SUCCESS : 正常終了 R_ENCOUT_ERR_INVALID_ARG : 異常終了 (引数 id が規定されていない値) R_ENCOUT_ERR_ACCESS : 異常終了 (ENCOUT ドライバが既に初期化されています)
注意	本関数実行前に、必ず EC-Lib を用いて Multi-Protocol Encoder I/F のコンフィギュレーションと起動を行ってください。

4.4.2 R_ENCOUT_Close

R_ENCOUT_Close	
概要	ENCOUT ドライバの終了
ヘッダ	r_encout_rzt2_if.h
宣言	r_encout_err_t R_ENCOUT_Close(const int32_t id);
説明	ENCOUT ドライバを終了します。 ENCOUT 動作中に、本関数をコールした場合、ENCOUT 停止処理の後、終了処理を実施します。
引数	id : ENCOUT_ID_0 あるいは ENCOUT_ID_1 を指定してください
リターン値	R_ENCOUT_SUCCESS : 正常終了 R_ENCOUT_ERR_INVALID_ARG : 異常終了 (引数 id が規定されていない値)

4.4.3 R_ENCOUT_GetVersion

R_ENCOUT_GetVersion	
概要	ENCOUT ドライバのバージョン取得
ヘッダ	r_encout_rzt2_if.h
宣言	uint32_t R_ENCOUT_GetVersion(void);
説明	ENCOUT ドライバのバージョンを取得します。
引数	なし
リターン値	バージョン情報 : 上位 16 ビットにメジャーバージョン、下位 16 ビットにマイナーバージョンが格納されます。 例) 戻り値が 0x00010002 の場合、Ver.1.2

4.4.4 R_ENCOUT_Control

R_ENCOUT_Control	
概要	ENCOUT の操作
ヘッダ	r_encout_rzt2_if.h
宣言	r_encout_err_t R_ENCOUT_Control(const int32_t id, const r_encout_cmd_t cmd, void *const p_buf);
説明	ENCOUT の操作を行います。 本関数は引数 cmd の値で動作が異なります。それぞれの動作は「4.4.5 制御コマンド」を参照してください。
引数	id : ENCOUT_ID_0 あるいは ENCOUT_ID_1 を指定してください cmd : 制御コマンド R_ENCOUT_CMD_INIT, R_ENCOUT_CMD_START, R_ENCOUT_CMD_STOP, R_ENCOUT_CMD_SET の何れかを指定してください
リターン値	p_buf : cmd に依存します。詳細は「4.4.5 制御コマンド」を参照してください。 cmd に依存します。詳細は「4.4.5 制御コマンド」を参照してください。

4.4.5 制御コマンド

(1) R_ENCOUT_CMD_INIT

R_ENCOUT_CMD_INIT	
概要	ENCOUT の初期化
ヘッダ	r_encout_rzt2_if.h
宣言	r_encout_err_t R_ENCOUT_Control(const int32_t id, const r_encout_cmd_t cmd, void *const p_buf);
説明	ENCOUT の初期化を行います。 「RZ/T2M グループ エンコーダ分周出力 (ENCOUT) アプリケーションノート」の「4.1 初期化」に記載されている初期化手順の「2. ENCOUT の初期設定」と「3. POSCNT の初期値の設定」を実施します。詳細は「RZ/T2M グループ エンコーダ分周出力 (ENCOUT) アプリケーションノート」を参照してください。
引数	id : ENCOUT_ID_0 あるいは ENCOUT_ID_1 を指定してください cmd : R_ENCOUT_CMD_INIT を指定してください p_buf : 初期化情報 設定値を記述した初期化情報構造体 r_encout_init_t へのポインタを指定します。構造体の詳細は「4.7.1(1) r_encout_init_t」を参照してください。
リターン値	R_ENCOUT_SUCCESS : 正常終了 R_ENCOUT_ERR_INVALID_ARG : 異常終了 (引数 id、cmd が規定されていない値、p_buf が NULL、p_buf に指定された r_encout_init_t 構造体のメンバ変数が規定されていない値または設定禁止の値) R_ENCOUT_ERR_ACCESS : 異常終了 (ENCOUT ドライバが初期化されていません) R_ENCOUT_ERR_BUSY : 異常終了 (ENCOUT が動作中です)

(2) R_ENCOUT_CMD_START

R_ENCOUT_CMD_START	
概要	ENCOUT の起動
ヘッダ	r_encout_rzt2_if.h
宣言	r_encout_err_t R_ENCOUT_Control(const int32_t id, const r_encout_cmd_t cmd, void *const p_buf);
説明	ENCOUT の起動を行います。 「RZ/T2M グループ エンコーダ分周出力 (ENCOUT) アプリケーションノート」の「4.1 初期化」に記載されている初期化手順の「5. ABZ 相出力開始」を実施します。詳細は「RZ/T2M グループ エンコーダ分周出力 (ENCOUT) アプリケーションノート」を参照してください。
引数	id : ENCOUT_ID_0 あるいは ENCOUT_ID_1 を指定してください cmd : R_ENCOUT_CMD_START を指定してください p_buf : 使用しません (NULL を指定してください)
リターン値	R_ENCOUT_SUCCESS : 正常終了 R_ENCOUT_ERR_INVALID_ARG : 異常終了 (引数 id、cmd が規定されていない値) R_ENCOUT_ERR_ACCESS : 異常終了 (ENCOUT ドライバが初期化されていません) R_ENCOUT_ERR_BUSY : 異常終了 (ENCOUT が動作中です)

(3) R_ENCOUT_CMD_STOP

R_ENCOUT_CMD_STOP	
概要	ENCOUT の停止
ヘッダ	r_encout_rzt2_if.h
宣言	r_encout_err_t R_ENCOUT_Control(const int32_t id, const r_encout_cmd_t cmd, void *const p_buf);
説明	ENCOUT の停止を行います。
引数	id : ENCOUT_ID_0 あるいは ENCOUT_ID_1 を指定してください cmd : R_ENCOUT_CMD_STOP を指定してください p_buf : 使用しません (NULL を指定してください)
リターン値	R_ENCOUT_SUCCESS : 正常終了 R_ENCOUT_ERR_INVALID_ARG : 異常終了 (引数 id、cmd が規定されていない値) R_ENCOUT_ERR_ACCESS : 異常終了 (ENCOUT ドライバが初期化されていません)

(4) R_ENCOUT_CMD_SET

R_ENCOUT_CMD_SET	
概要	ENCOUT の位置値設定
ヘッダ	r_encout_rzt2_if.h
宣言	r_encout_err_t R_ENCOUT_Control(const int32_t id, const r_encout_cmd_t cmd, void *const p_buf);
説明	ENCOUT 動作中に OUTCNT レジスタの設定を行います。 「RZ/T2M グループ エンコーダ分周出力 (ENCOUT) アプリケーションノート」の「4.2 メイン処理」に記載されている手順の「3. OUTCNT レジスタ設定値の計算」と「4. OUTCNT レジスタの設定」を実施します。詳細は「RZ/T2M グループ エンコーダ分周出力 (ENCOUT) アプリケーションノート」を参照してください。
引数	id : ENCOUT_ID_0 あるいは ENCOUT_ID_1 を指定してください cmd : R_ENCOUT_CMD_SET を指定してください p_buf : 位置情報 設定値を記述した位置情報構造体 r_encout_set_t へのポインタを指定します。構造体の詳細は「4.7.1(2) r_encout_set_t」を参照してください。
リターン値	R_ENCOUT_SUCCESS : 正常終了 R_ENCOUT_ERR_INVALID_ARG : 異常終了 (引数 id、cmd が規定されていない値、p_buf が NULL、p_buf に指定された r_encout_set_t 構造体のメンバ変数が設定禁止の値) R_ENCOUT_ERR_ACCESS : 異常終了 (ENCOUT が動作していません)

4.5 定数/エラーコード一覧

表 4-2 に定数/エラーコード定義表の一覧を示します。各定義については、それぞれの表を参照してください。

表 4-2 定数/エラーコード定義表の一覧

表番号	内容
表 4-3	ENCOUT ドライバで使用するユーザー定義の定数 (r_encout_rzt2_config.h)
表 4-4	R_ENCOUT_Control 関数の制御コマンド (r_encout_rzt2_if.h)
表 4-5	ENCOUT ドライバの ID (r_encout_rzt2_if.h)
表 4-6	B 相の出力極性 (r_encout_rzt2_if.h)
表 4-7	Z 相の出力パターン (r_encout_rzt2_if.h)
表 4-8	エラーコード (r_encout_rzt2_if.h)

表 4-3 ENCOUT ドライバで使用するユーザー定義の定数 (r_encout_rzt2_config.h)

定数名	設定値	内容
ENCODER_RESOLUTION0	1048576	ENCOUT ドライバが位置の計算に用いる ENCOUT0 チャンネルのエンコーダの解像度 0 以外の 32 ビット値が設定できます。例えば、エンコーダの解像度が 20 ビット (位置値が 0~1048575) の場合は、1048576 を指定します。デフォルト値は 1048576 です。
ENCODER_RESOLUTION0	1048576	ENCOUT ドライバが位置の計算に用いる ENCOUT1 チャンネルのエンコーダの解像度 0 以外の 32 ビット値が設定できます。例えば、エンコーダの解像度が 20 ビット (位置値が 0~1048575) の場合は、1048576 を指定します。デフォルト値は 1048576 です。

表 4-4 R_ENCOUT_Control 関数の制御コマンド (r_encout_rzt2_if.h)

定数名	設定値	内容
R_ENCOUT_CMD_INIT	0	ENCOUT の初期化
R_ENCOUT_CMD_START	1	ENCOUT の起動
R_ENCOUT_CMD_STOP	2	ENCOUT の停止
R_ENCOUT_CMD_SET	3	ENCOUT の設定

表 4-5 ENCOUT ドライバの ID (r_encout_rzt2_if.h)

定数名	設定値	内容
ENCOUT_ID_NUM	2	ENCOUT ドライバのチャンネル数
ENCOUT_ID_0	0x01	ENCOUT ドライバ チャンネル 0 の ID
ENCOUT_ID_1	0x02	ENCOUT ドライバ チャンネル 1 の ID

表 4-6 B 相の出力極性 (r_encout_rzt2_if.h)

定数名	設定値	内容
ENCOUT_CTL_PHASE_NORMAL	0x00	B 相出力が正相
ENCOUT_CTL_PHASE_REVERSE	0x01	B 相出力が逆相

表 4-7 Z 相の出力パターン (r_encout_rzt2_if.h)

定数名	設定値	内容
ENCOUT_CTL_Z_NONE	0x00	Z 相出力を無効にする
ENCOUT_CTL_Z_1P	0x02	現在位置 POSCNT が 0 のときパルス出力
ENCOUT_CTL_Z_2P_A	0x04	現在位置 POSCNT が 0 または最大値のときパルス出力 (A 相と同期)
ENCOUT_CTL_Z_2P_B	0x14	現在位置 POSCNT が 1 以下のときパルス出力 (B 相と同期)
ENCOUT_CTL_Z_3P	0x06	現在位置 POSCNT が 1 以下または最大値のときパルス出力
ENCOUT_CTL_Z_4P_A	0x08	現在位置 POSCNT が 2 以下または最大値のときパルス出力 (A 相と同期)
ENCOUT_CTL_Z_4P_B	0x18	現在位置 POSCNT が 1 以下または最大値-1 以上のときパルス出力 (B 相と同期)

表 4-8 エラーコード (r_encout_rzt2_if.h)

定数名	設定値	内容
R_ENCOUT_SUCCESS	0	正常終了
R_ENCOUT_ERR_INVALID_ARG	-1	引数異常
R_ENCOUT_ERR_BUSY	-2	API を実行できない状態
R_ENCOUT_ERR_ACCESS	-3	API の実行順序エラー

4.6 固定幅整数一覧

表 4-9 にサンプルコードで使用する固定幅整数を示します。サンプルコードで使用する固定幅整数は、標準ライブラリで定義されています。

表 4-9 サンプルコードで使用する固定幅整数

シンボル	内容
int8_t	8 ビット整数、符号あり
int16_t	16 ビット整数、符号あり
int32_t	32 ビット整数、符号あり
int64_t	64 ビット整数、符号あり
uint8_t	8 ビット整数、符号なし
uint16_t	16 ビット整数、符号なし
uint32_t	32 ビット整数、符号なし
uint64_t	64 ビット整数、符号なし

4.7 構造体/共用体/列挙型一覧

主要な構造体/共用体/列挙型の一覧を記載します。

4.7.1 構造体

(1) r_encout_init_t

ENCOUT 制御部の初期化情報。

```
typedef struct
{
    uint8_t    control_param;   コントロールレジスタ(CTL)の POL ビット, ZW ビット, ZS ビット
                               に設定する値を指定します。「表 4-6 B 相の出力極性
                               (r_encout_rzt2_if.h)」、「表 4-7 Z 相の出力パターン
                               (r_encout_rzt2_if.h)」に示したマクロを用いて指定してください。
    uint16_t   position_max;    1 回転の最大位置 (エッジの数 - 1)を指定します。
                               本設定は POSMAX レジスタに反映されます。
    uint32_t   carrier_period;  キャリア周期を ns 単位で指定します。50000~3276750 の範囲の
                               50 の倍数を指定してください。
                               本設定は PERIOD レジスタに反映されます。
    uint32_t   encoder_count;   エンコーダの初期位置値を 0~(ENCODER_RESOLUTIONn - 1)
                               の範囲で設定します。(ENCODER_RESOLUTIONn は各エンコーダ
                               の解像度)
                               本設定は POSCNT レジスタに反映されます。
} r_encout_init_t
```

(2) r_encout_set_t

ENCOUT に設定する位置情報。

```
typedef struct
{
    uint32_t   encoder_count;   エンコーダの位置値を 0~(ENCODER_RESOLUTIONn - 1) の範
                               囲で設定します。(ENCODER_RESOLUTIONn は各エンコーダの解
                               像度)
} r_encout_set_t
```

4.7.2 共用体

使用しません。

4.7.3 列挙型

使用しません。

4.8 サンプルプログラムの説明

4.8.1 動作概要

本サンプルプログラムは以下の処理を行います。

- 1) ENCOUT モジュールの動作パラメータとして、キャリア周期や 1 回転の最大位置、ABZ 相信号の出力パターンを指定
- 2) GPT をキャリア周期に設定し、ELC を介して ENCOUT のキャリア周期イベント信号として入力
- 3) GPT からの割り込みを使用して、キャリア周期で ENCOUT のエッジカウント設定を更新しながら、A 相, B 相, Z 相信号を出力

(1) システムブロック図

図 4-1 にシステムブロック図を示します。

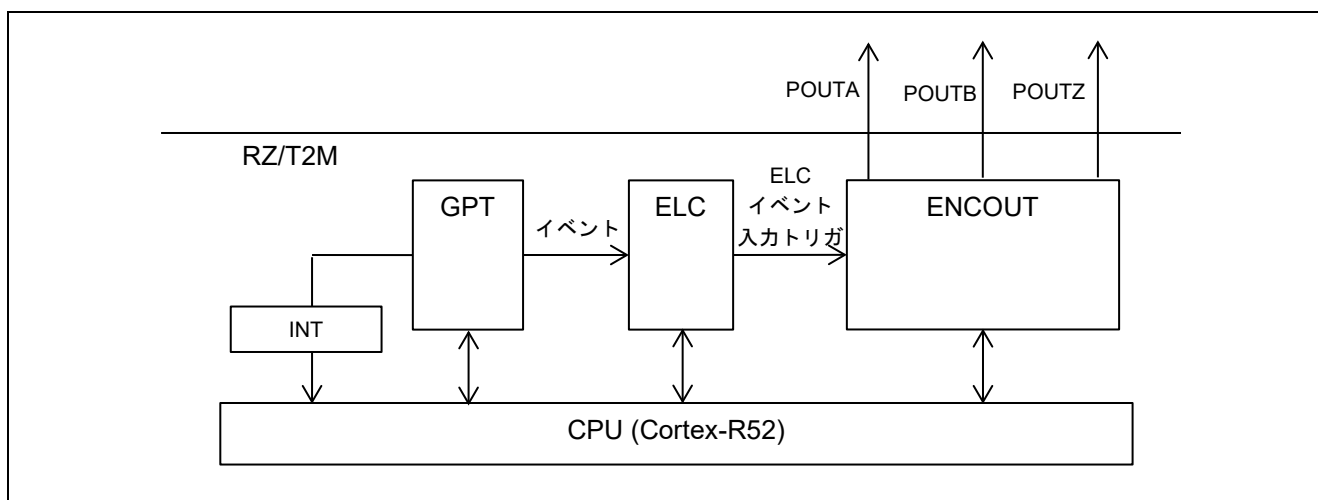


図 4-1 システムブロック図

(2) ソフトウェア構成図

図 4-2 にソフトウェア構成図を示します。

ENCOUT ドライバには、R_ENCOUT_Open 関数で構成される開始処理部、R_ENCOUT_Close 関数で構成される終了処理部、R_ENCOUT_Control 関数で構成される制御部があります。

サンプルプログラムには、ENCOUT ドライバを制御し、起動・初期化を行う初期化部分、タイマ割り込みに応じて位置情報の更新を行う位置情報更新部分があります。

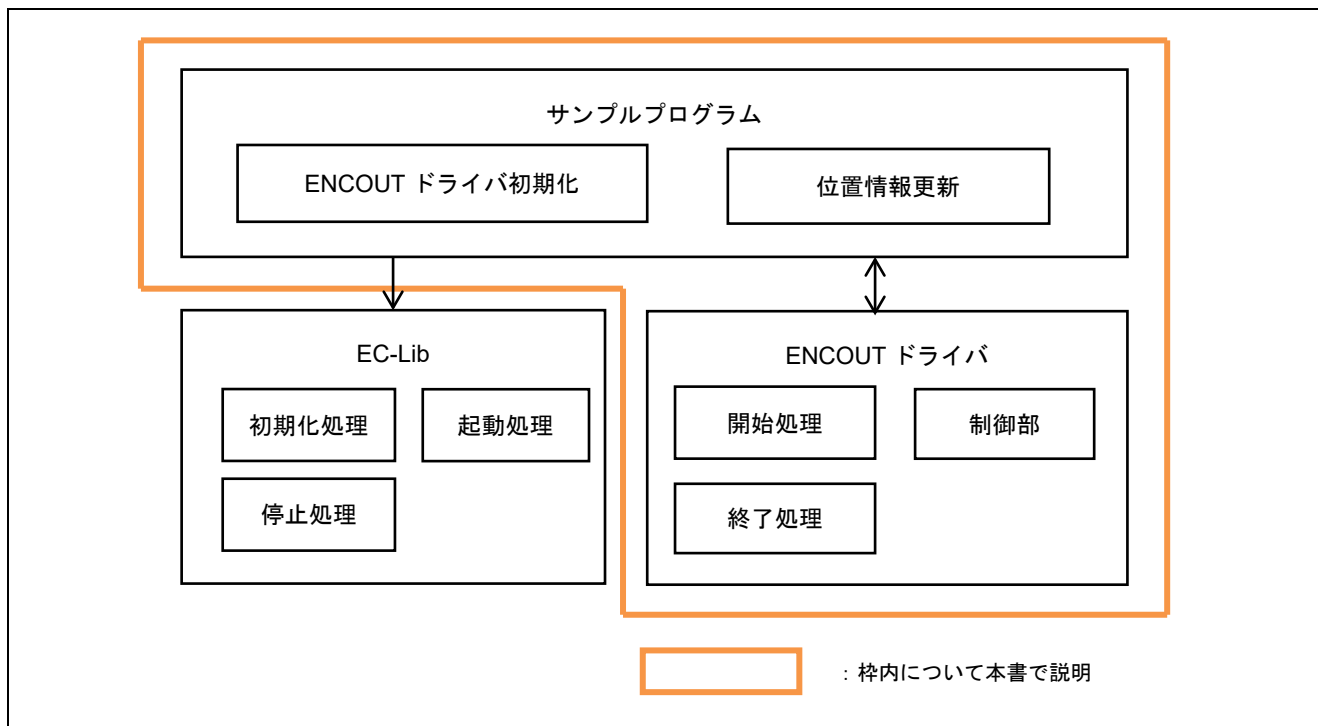


図 4-2 ソフトウェア構成図

4.8.2 サンプルプログラム関数一覧

表 4-10 に主要なサンプルプログラム関数一覧を示します

表 4-10 主要なサンプルプログラム関数一覧

関数名	ページ番号
hal_entry	16
enc_main	16
show_encout_drv_version	17
timer_init	17
encout_initialize	17
show_encout_dat_version	17
timer_start	18
int_gpt0_ovf	18
int_gpt0_ccmpa	18

4.8.3 サンプルプログラム関数仕様

(1) hal_entry

hal_entry	
概要	ENCOUT サンプルプログラムのエントリー関数
ヘッダ	-
宣言	void hal_entry(void);
説明	ENCOUT サンプルプログラムのエントリー関数です。ここから、関数 enc_main()が呼び出されます。
引数	なし
リターン値	なし

(2) enc_main

enc_main	
概要	ENCOUT サンプルプログラムのメイン関数
ヘッダ	-
宣言	int32_t enc_main(uint8_t ch);
説明	ENCOUT サンプルプログラムのメイン関数です。詳細は「4.8.6(1) enc_main フローチャート」を参照してください。
引数	ch : ENCOUT チャンネル番号 0: ch0 を使用する 1: ch1 を使用する 2: ch0 と ch1 の両チャンネルを使用する
リターン値	0 : 正常終了 0 以外 : 異常終了 (ENCOUT ドライバのエラーコード)

(3) show_encout_drv_version

show_encout_drv_version	
概要	ENCOUT ドライババージョン表示関数
ヘッダ	-
宣言	static void show_encout_drv_version(void);
説明	コンソールに ENCOUT ドライバのバージョンを表示します。
引数	なし
リターン値	なし

(4) timer_init

timer_init	
概要	タイマ初期化関数
ヘッダ	-
宣言	static void timer_init(void);
説明	タイマ GPT チャンネル 0 の周期設定、割り込み設定を行います。 周期を「表 4-12 主要な定数」の CARRIER_PERIOD で指定した値に設定します。
引数	なし
リターン値	なし

(5) encout_initialize

encout_initialize	
概要	ENCOUT コンフィグレーション / 初期化
ヘッダ	-
宣言	static void encout_initialize(uint8_t id);
説明	EC-Lib を用いて、Encoder I/F のコンフィグレーションと初期化を行います。
引数	id : ENCOUT ドライバ ID ENCOUT_ID_0, ENCOUT_ID_1, あるいは両方のチャンネルを示す (ENCOUT_ID_0 ENCOUT_ID_1) を指定してください。
リターン値	なし

(6) show_encout_dat_version

show_encout_dat_version	
概要	ENCOUT コンフィグレーションデータバージョン表示関数
ヘッダ	-
宣言	static void show_encout_dat_version(uint8_t id);
説明	コンソールに ENCOUT コンフィグレーションデータのバージョンを表示します。
引数	id : ENCOUT ドライバ ID ENCOUT_ID_0, ENCOUT_ID_1, あるいは両方のチャンネルを示す (ENCOUT_ID_0 ENCOUT_ID_1) を指定してください。
リターン値	なし

(7) timer_start

timer_start	
概要	タイマ起動関数
ヘッダ	-
宣言	static void timer_start(void);
説明	タイマ GPT チャンネル 0 を起動します。
引数	なし
リターン値	なし

(8) int_gpt0_ovf

int_gpt0_ovf	
概要	タイマ GPT0_OVF 割り込みハンドラ
ヘッダ	-
宣言	void int_gpt0_ovf(void);
説明	タイマ起動後、「表 4-12 主要な定数」の CARRIER_PERIOD で指定した周期で呼び出され、ENCOUT0 チャンネルの位置値を「表 4-11 主要な static 型変数」の encoder_data[] 配列で設定した値で逐次更新します。
引数	なし
リターン値	なし

(9) int_gpt0_ccmpa

int_gpt0_ccmpa	
概要	タイマ GPT0_CCMPA 割り込みハンドラ
ヘッダ	-
宣言	void int_gpt0_ccmpa(void);
説明	タイマ起動後、「表 4-12 主要な定数」の CARRIER_PERIOD で指定した周期で呼び出され、ENCOUT1 チャンネルの位置値を「表 4-11 主要な static 型変数」の encoder_data[] 配列で設定した値で逐次更新します。 GPT0_CCMPA 割り込みは、GPT0_OVF 割り込みより CARRIER_PERIOD の 1/2 だけ進んで発生します。
引数	なし
リターン値	なし

4.8.4 サンプルプログラムの変数一覧

表 4-11 に主要な static 型変数を示します。

表 4-11 主要な static 型変数

型	変数名	内容
uint32_t	encoder_data_index[2]	エンコーダ位置情報配列のインデックス 位置情報配列から ENCOUT に与える要素を取り出すためのインデックスで、タイマ周期でインクリメントします。 初期値は {0, 0} です。
uint32_t	encoder_data[]	エンコーダ位置情報配列 タイマ周期で ENCOUT に与える位置情報を配列として格納しています。デフォルト値は {943719, 996148, 20972, 1027605, 1038090} です。 解像度が 1048576、1 回転の最大位置が 99 のとき、位置情報デフォルト値の各要素は、それぞれ位置カウント {90, 95, 2, 98, 98} に対応します。 この配列の要素の代わりに、タイマ周期でエンコーダから読みだした位置情報を使うと、エンコーダの回転に応じた ABZ 相信号が出力されます。

4.8.5 サンプルプログラムの定数一覧

表 4-12 にサンプルプログラムで使用する主要な定数を示します。

表 4-12 主要な定数

定数名	設定値	内容
CARRIER_PERIOD	100000	ENCOUT が出力する ABZ 相信号のキャリア周期 キャリア周期を ns 単位で指定してください。設定範囲は 50000～3276750 となります。デフォルト値は 100000 [ns] (100 [us]) です。50 の整数倍で設定して下さい。
POSITION_MAX	99	1 回転の最大位置 (エッジの数 - 1) 最大位置を指定してください。この値が ENCOUT モジュールの POSMAX_PERIOD レジスタ POSMAX ビットに設定されます。1 回転あたりのエッジの数は 4 の倍数 (4N) で、設定値は (4N - 1) としてください。設定可能な値は 15, 19, 23, ..., 65531, 65535 です。デフォルト値は 99 です。
ENCOUT_CTL	0x04	ENCOUT が出力する ABZ 相信号の出力パターン B 相の極性 (正相, 逆相)、Z 相のパルス幅 (0, 1, 2, 3, 4) と、同期する相の指定 (A 相, B 相) を行います。表 4-6, 表 4-7 に示したマクロを用いて指定してください。デフォルト設定は、B 相が正相、Z 相が A 相と同期した 2 パルス出力です。

4.8.6 メイン処理のフローチャート

以下に主要な処理を行う関数のフローチャートを記載します。

(1) enc_main フローチャート

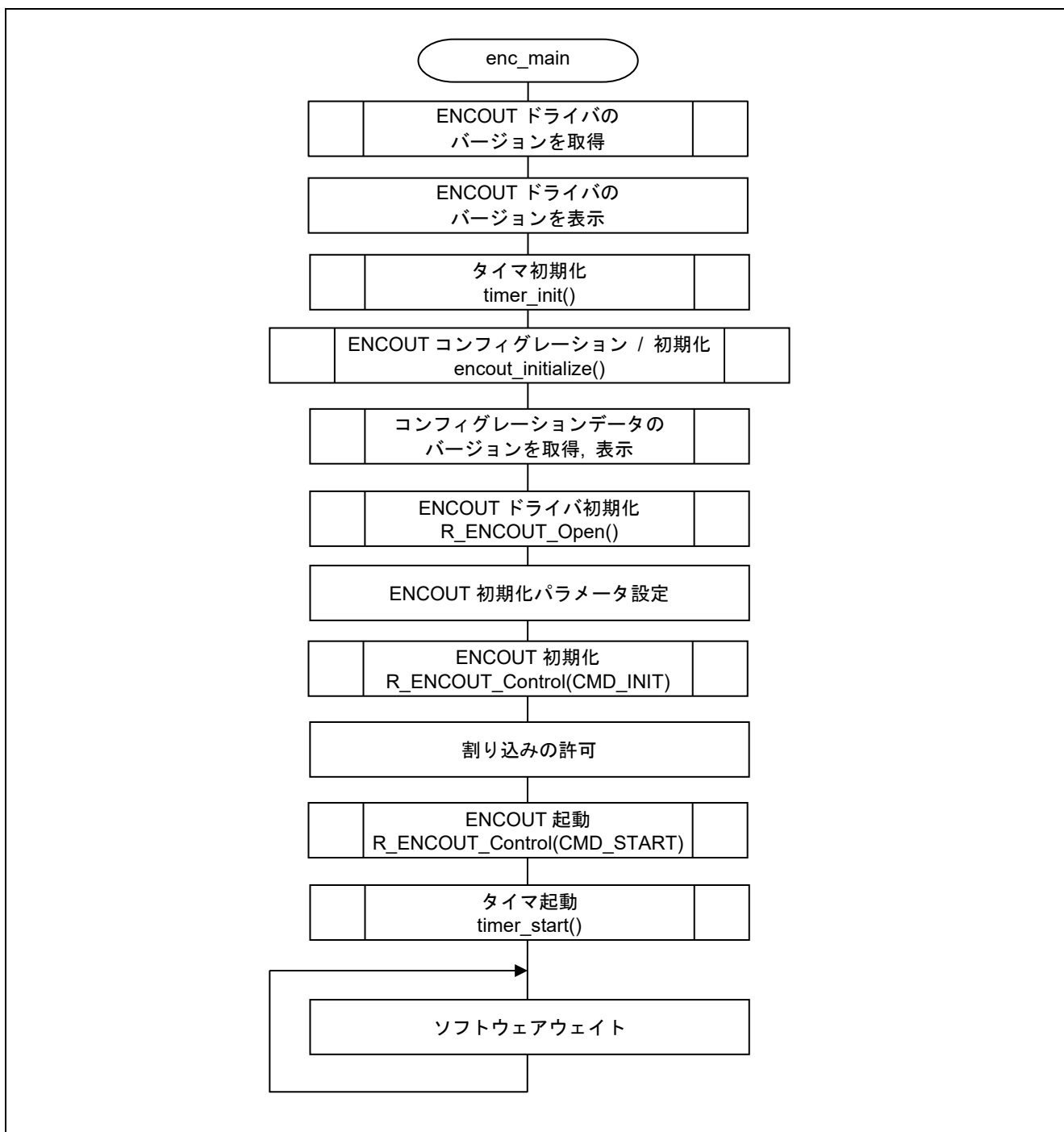


図 4-3 enc_main 関数のフローチャート

(2) int_gpt0_ovf, int_gpt0_ccmpa フローチャート

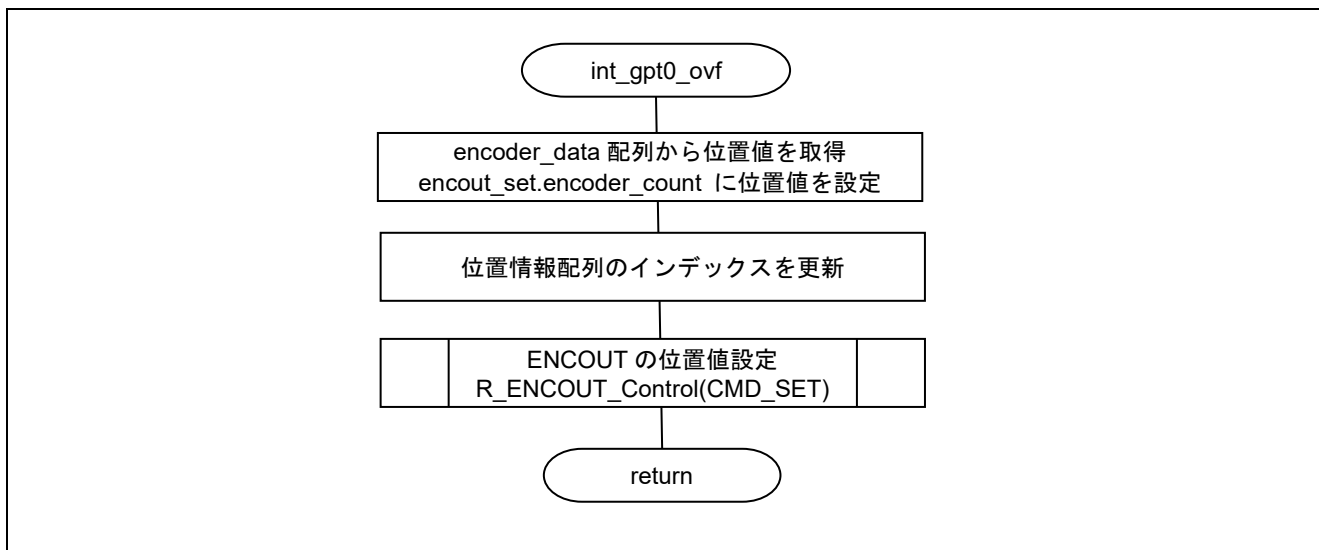


図 4-4 int_gpt0_ovf, int_gpt0_ccmpa 関数のフローチャート

4.8.7 動作シーケンス

(1) 開始シーケンス

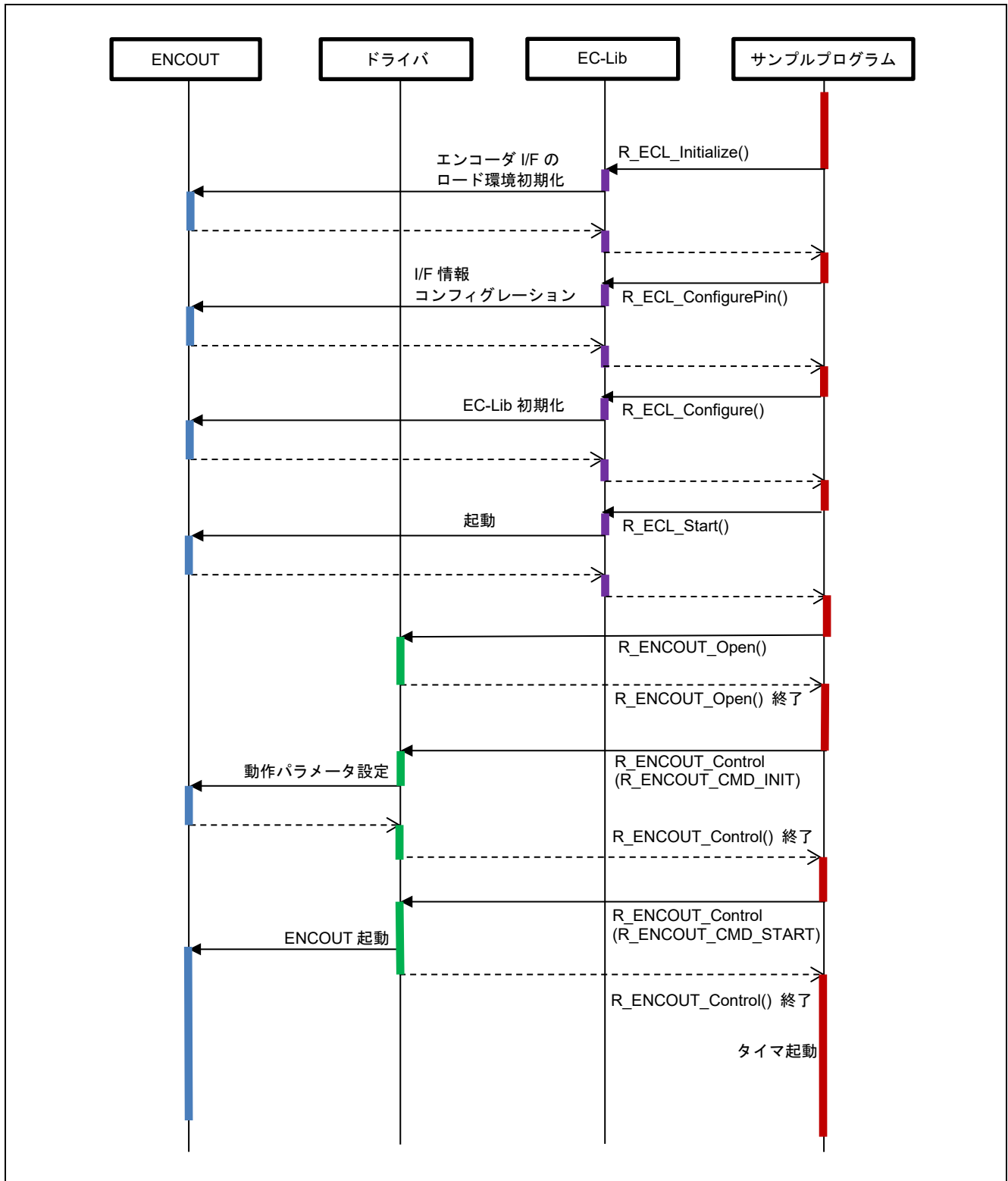


図 4-5 開始シーケンス図

(2) 位置情報更新シーケンス

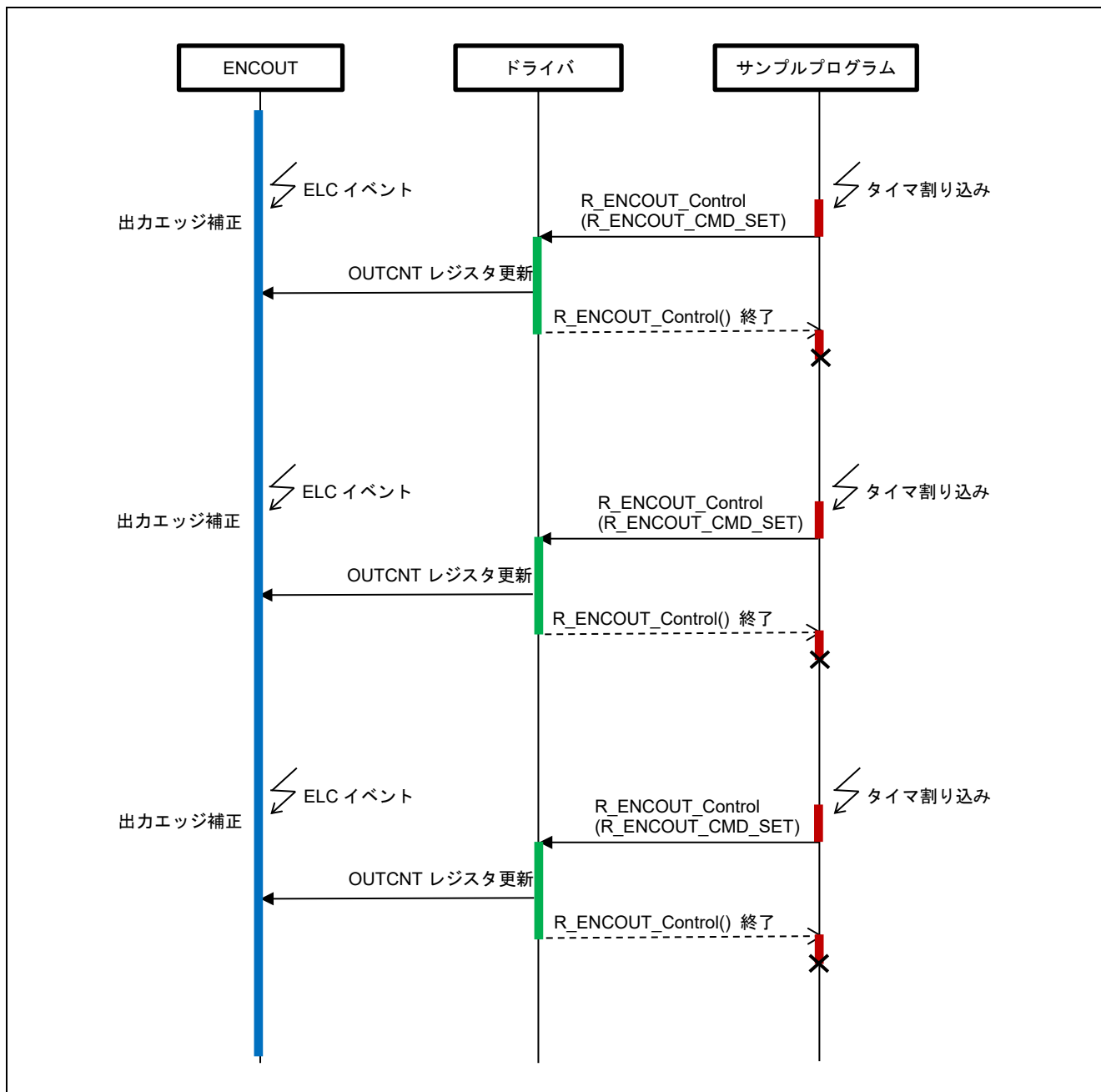


図 4-6 位置情報更新シーケンス図

4.8.8 コンソールコマンド

本サンプルプログラムには、コンソールからの入力コマンドはありません。

4.8.9 サンプルプログラム実行結果

本サンプルプログラムを実行すると、「3.1 使用端子一覧」に記載した出力ポートに、図 4-7 のような信号が出力されます。ENCOUT1 チャンネルからは、ENCOUT0 チャンネルと同様の波形が、ENCOUT0 チャンネルより 50 us (キャリア周期の 1/2) 早く出力されます。

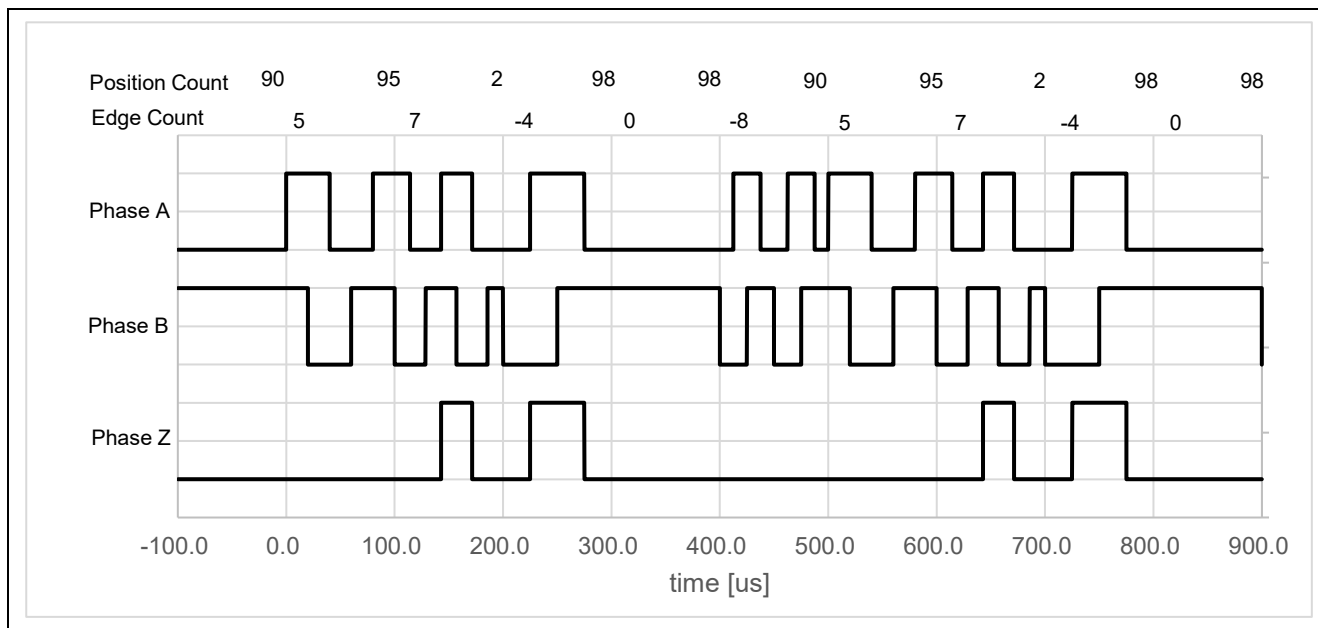


図 4-7 ABZ 相出力信号

5. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
3.00	Oct 17.25	-	初版発行 (リリースノートから分離)
4.00	Apr 03.26	8 - 10	ポインタ変数のプレフィクスを” p_” に変更

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しないでください。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えば、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、ノイズ輻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。

- IAR Embedded Workbench is a registered trademark of IAR Systems.
- Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.
- Additionally all product names and service names in this document are a trademark or a registered trademark which belongs to the respective owners.

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まれます。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、変更、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、変更、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）、交通管制（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、これに限られません。）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た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。