

RZ/T1 グループ

ENCOUT application package

R01AN3806JJ0100
Rev.1.00
2017.8.4

要旨

本書は、RZ/T1 ENCOUT のアプリケーション・パッケージについて記載します。

本アプリケーション・パッケージを使用する時は、「RZ/T1 Encoder I/F Configuration Library」のリリースパッケージを入手してください。

動作確認デバイス

RZ/T1 CPU ボード (RTK7910018C00000BE)

過去バージョンからの変更履歴

Ver.	Date	内容	備考
1.0	2017.8	RZ/T1 ENCOUT サンプルプログラムを Ver.1.0 に更新	
		(1) R_ENCOUT_Control(R_ENCOUT_CMD_INIT)関数に指定可能なキャリア周期を 3276999ns までに変更 (2) R_ENCOUT_GetVersion 関数の戻り値を Ver.1.0 に更新 (3) 誤記修正等	
		RZ/T1 ENCOUT Configuration Data を Ver.1.0 に更新	
		(1) ABZ 相出力端子を 1 系統追加 (2) VER レジスタの値を Ver.1.0 に更新	
		RZ/T1 グループ エンコーダ分周出力 (ENCOUT) ユーザーズマニュアルを Rev.1.00 に更新	
0.5	2017.4	新規	

目次

1. パッケージ内容	3
1.1 ソフトウェア	3
・ソースコード	3
・コンフィグレーション・データ	3
1.2 ドキュメント	3
2. ファイル構成	4
3. ENCOUT サンプルプログラムについて	5
3.1 動作環境	5
3.2 ターゲットボード	6
3.3 サンプルプログラム実行前の準備	7
3.4 各開発環境における動作手順	8
3.4.1 EWARM : IAR システムズ社製	8
3.4.2 DS-5 : ARM 社製	9
3.4.3 e2 studio : RENESAS 社製	12
3.4.4 実行結果	14
3.5 仕様	15
3.5.1 メモリサイズ	15
3.5.2 ENCOUT ドライバ API	16
3.5.3 処理概要	21
3.5.4 設定値の変更方法	23
3.6 Encoder I/F と ENCOUT の組み合わせ方法	24
4. 制限事項	25
5. 注意事項	25
5.1 処理時間	25

1. パッケージ内容

本パッケージには次のものが含まれています。

1.1 ソフトウェア

・ソースコード

No.	名称	版数
1	RZ/T1 ENCOUT サンプルプログラムコード一式	1.0

・コンフィグレーション・データ

No.	名称	版数
1	RZ/T1 ENCOUT Configuration Data	1.0

1.2 ドキュメント

No.	文書名	版数	ファイル名
1	RZ/T1 グループ ENCOUT application package リリースノート	1.00	(日)r01an3806jj0100-rzt1.pdf (本書)
2	RZ/T1 グループ エンコーダ分周出力 (ENCOUT) ユーザーズマニュアル	1.00	(日) r01uh0701jj0100_rzt1_encout.pdf (英) r01uh0701ej0100_rzt1_encout.pdf

2. ファイル構成

本パッケージのファイル構成と内容物の詳細を以下に示します。

```

Top
├──r01an3806jj0100-rzt1.pdf
└──workspace
    ├──Documentation
    │   ├──r01uh0701ej0100_rzt1_encout.pdf
    │   └──r01uh0701jj0100_rzt1_encout.pdf
    └──Software
        ├──armcc
        │   └──RZ_T1_encout.zip : RZ/T1 ENCOUT サンプルプログラムコード一式(DS-5)
        ├──iccarms
        │   └──RZ_T1_encout.zip : RZ/T1 ENCOUT サンプルプログラムコード一式(EWARM)
        └──kpitgcc
            └──RZ_T1_encout.zip : RZ/T1 ENCOUT サンプルプログラムコード一式(e2 studio)
  
```

RZ_T1_encout.zip のファイル構成を以下に示します。

Top folder		
inc		
iodefne.h		RZ/T1 レジスタ定義ファイル
iodefne_encout.h		ENCOUT レジスタ定義ファイル
r_encout_rzt1_dat.h		r_encout_rzt1.dat 用ヘッダファイル
r_encout_rzt1_if.h		ENCOUT ドライバヘッダファイル
初期設定などの共通ヘッダファイル		
lib		
ecl		
r_encout_rzt1.dat		RZ/T1 ENCOUT Configuration Data
src		
common		
初期設定などの共通ソース		
drv		
encout		
r_encout_rzt1_config.h		ENCOUT ドライバファイル
r_encout_rzt1.c		ENCOUT ドライバファイル
scifa_uart		
SCIFA ドライバファイル		
sample		
main.c		メインプログラムファイル
encout_dat.s		Configuration data 用リンク設定ファイル ※1
nestintr_wrap.s		初期設定用サンプルプログラム (DS-5 のみ)
siorw.c		SCIFA サンプルプログラム
siochar.c		SCIFA サンプルプログラム
retarget.c		SCIFA サンプルプログラム(DS-5 のみ)

※1 : DS-5/e2 studio 用ファイル
 DS-5 : encout_dat.s
 e2 studio : encout_dat.asm

3. ENCOUT サンプルプログラムについて

ENCOUT サンプルプログラムを使用するために必要な情報を記載します。

3.1 動作環境

ENCOUT サンプルプログラムは、下記の環境を想定しています。

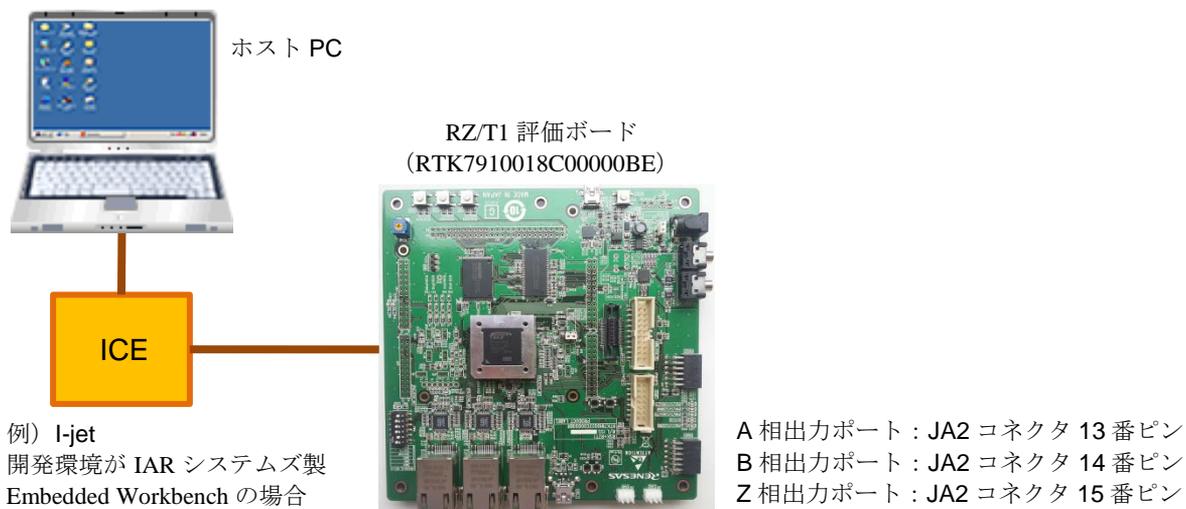
項目	内容
使用マイコン	RZ/T1 グループ
動作周波数	CPUCLK = 450MHz
動作電圧	3.3V
統合開発環境	IAR システムズ社製 Embedded Workbench® for ARM Version 7.80.2 (ICE : I-jet) ARM 社製 ARM Development Studio 5 (DS-5™) Version 5.25.0 ARM Compiler 5.06 update 3 (ICE : ULINK2) RENESAS 社製 RENESAS e2 studio 5.2.0.020 KPIT GNUARM-NONE-EABI Toolchain v16.01 (ICE : J-Link BASE)
動作モード	SPI ブートモード 16 ビットバスブートモード
使用ボード	RZ/T1 評価ボード (RTK7910018C00000BE)
使用デバイス (ボード上で使用する機能)	シリアルインタフェース (USB-Mini B コネクタ J8) NOR フラッシュメモリ (CS0、CS1 空間に接続) メーカー名 : Macronix International Co. Ltd. 型名 : MX29GL512FLT2I-10Q SDRAM (CS2、CS3 空間に接続) メーカー名 : Integrated Silicon Solution Inc. 型名 : IS42S16320D-7TL シリアルフラッシュメモリ メーカー名 : Macronix International Co. Ltd. 型名 : MX25L51245G
ベース OS	OS レス

使用する RZ/T1 の周辺機能は下記です。

周辺機能	用途
エンコーダ分周出力機能 (ENCOUT)	ABZ 相信号出力
コンペアマッチタイマ (CMT) ユニット 0 チャンネル 1	キャリア周期の生成
イベントリンクコントローラ (ELC)	キャリア周期を ENCOUT に入力
FIFO 内蔵シリアルコミュニケーション インタフェース (SCIFA)	デバッグ情報の出力

3.2 ターゲットボード

ホスト PC とターゲットボード「RZ/T1 評価ボード (RTK7910018C00000BE)」との接続は以下のようになります。



ターゲットボード「RZ/T1 評価ボード (RTK7910018C00000BE)」の設定は以下の通り。

SW4-1: ON

SW4-2: シリアルフラッシュを使用する場合は ON、NOR を使用する場合は OFF

SW4-3: ON

SW4-4: ON

SW4-5: ON

SW4-6: OFF

JP2: 2-3 ショート

JP7: 1-2 ショート

3.3 サンプルプログラム実行前の準備

本サンプルプログラムでは、ホスト PC と通信動作を行います。サンプルプログラムの実行前に、USB シリアルドライバのインストールと、ターミナルソフトの設定を行ってください。

ルネサスエレクトロニクスホームページから「RZ/T1 グループ FIFO 内蔵シリアルコミュニケーションインターフェース (SCIF) サンプルプログラム」をダウンロードして、同梱されている USB シリアルドライバをインストールしてください。

ターミナルソフトの設定は以下の通り。

ボーレート : 115200kbps

データ : 8 ビット

パリティ : なし

ストップビット : 1 ビット

フローコントロール : なし

3.4 各開発環境における動作手順

3.4.1 EWARM : IAR システムズ社製

- サンプルプログラムのビルド手順

サンプルプログラムのビルド手順は以下の通り。

1. 展開したソースファイルを任意の場所にコピー
2. “RZ/T1 Encoder I/F Configuration Library” (IAR EWARM 版)の以下のファイルを各フォルダへコピー

lib\ecl\r_ecl_rzt1.a

inc\r_ecl_rzt1_if.h

3. EWARM を起動
4. [ファイル]メニュー→[開く]→[ワークスペース]を選択
5. 展開したソースファイルの RZ_T1_encout_boot\RZ_T1_encout_****_boot.eww を開く

Nor/Serial	プロジェクトファイル
Nor 版	RZ_T1_encout_nor_boot.eww
Serial Flash 版	RZ_T1_encout_serial_boot.eww

6. [プロジェクト]メニュー→[すべてを再ビルド]を選択

次のファイルが生成される。

RZ_T1_encout_boot\Debug\Exe\RZ_T1_encout_****_boot.out

Nor/Serial	プロジェクトファイル
Nor 版	RZ_T1_encout_nor_boot.out
Serial Flash 版	RZ_T1_encout_serial_boot.out

- サンプルプログラムの実行手順

「ビルド手順」を実行後、ターゲットボード、デバッガを正しく接続し、以下の操作を行う。

1. [プロジェクト]メニュー→[ダウンロードしてデバッグ]を選択
2. [デバッグ]メニュー→[実行]を選択

3.4.2 DS-5 : ARM 社製

- サンプルプログラムのビルド手順

サンプルプログラムのビルド手順は以下の通り。

1. 展開したソースファイルを任意の場所にコピー
2. “RZ/T1 Encoder I/F Configuration Library” (ARM DS-5 版)の以下のファイルを、各フォルダへコピー

```
lib\ecl\r_ecl_rzt1.a  
inc\r_ecl_rzt1_if.h
```
3. DS-5 を起動
4. [ウィンドウ]メニュー→[ビューの表示] →[プロジェクト・エクスプローラー]を選択
5. [プロジェクト・エクスプローラー]ビューを右クリックし、ポップアップメニューの[インポート]を選択
6. 「インポート」ダイアログの「一般」 → 「既存プロジェクトをワークスペースへ」 を選択し、「次へ」 ボタンをクリック
7. 「インポートダイアログ」 の「参照」 をクリック
8. 「フォルダの参照」 ダイアログで、ソースファイルのコピー先のトップを選択し、「OK」 をクリック
9. 「インポート」 ダイアログの「プロジェクトをワークスペースにコピー」 のチェックを付ける
10. 「インポート」 ダイアログの「終了」 をクリック
11. [プロジェクト]メニュー→[すべてビルド]を選択

次のファイルが生成される。

Debug\RZ_T_nor_sample.axf

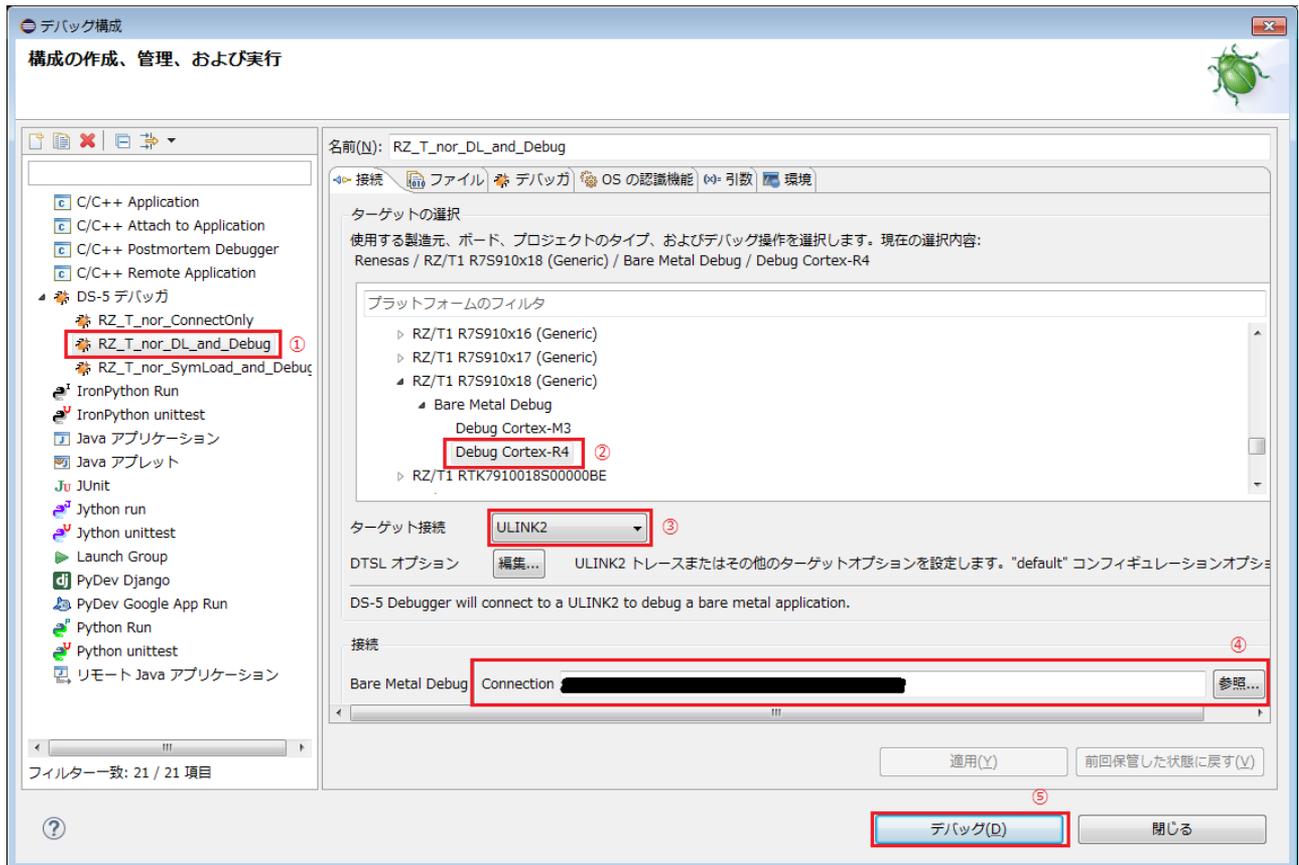
(シリアルフラッシュの場合は、「RZ_T_sflash_sample.axf」)

- サンプルプログラムの実行手順

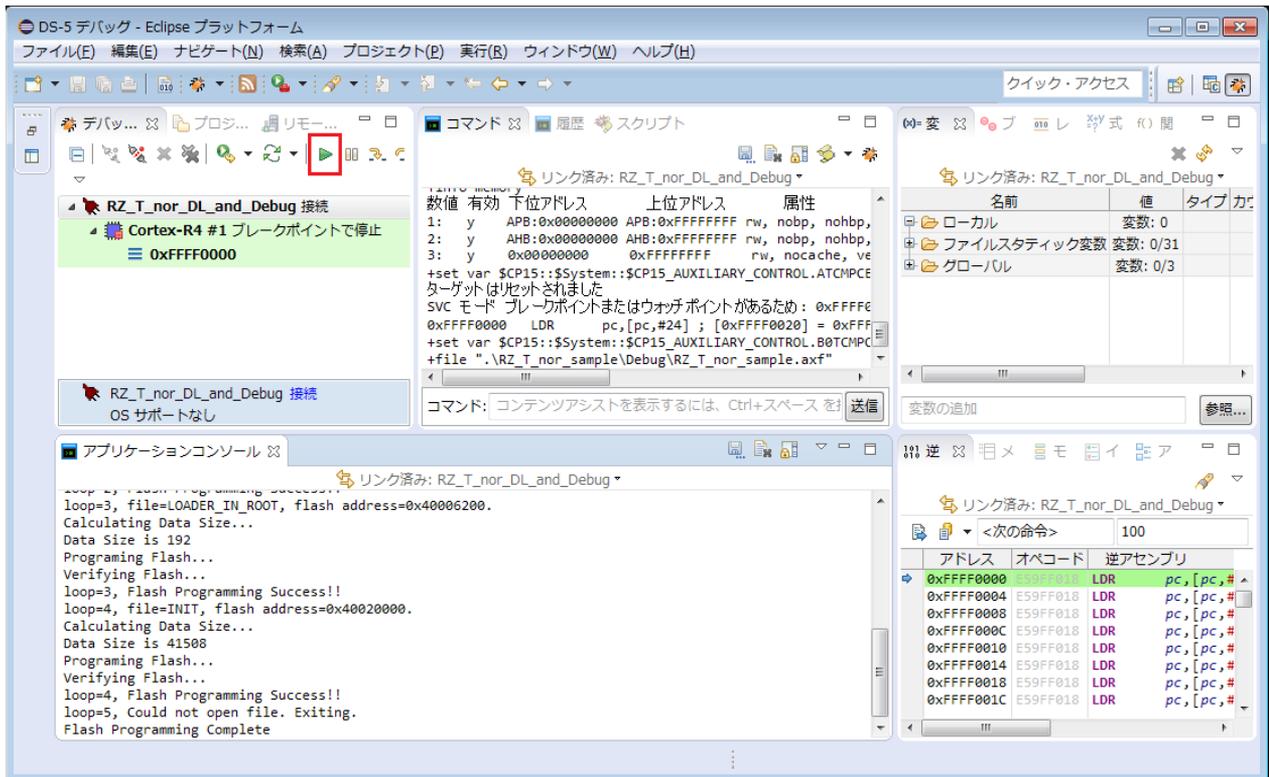
「ビルド手順」を実行後、ターゲットボード、デバッガを正しく接続し、以下の操作を行う。

1. [実行]→[デバッグの構成]でデバッグ構成を開き、“RZ_T_nor_DL_and_Debug”の接続設定を選択します。（シリアルフラッシュの場合は、“RZ_T_sflash_DL_and_Debug”）
ターゲットの選択で “RZ/T1 R7S910x18 (Generic)” の Debug Cortex-R4 を選択します。

接続タブよりターゲット接続の ULINK2 を選択し、[参照] を選択し、検出されたターゲット接続を選択後、[デバッグ] を選択しデバッグを開始します。



2. フラッシュ書き込みが完了すると、アプリケーションコンソール画面に「Flash Programming Complete」が表示され、デバッグを開始できます。



3.4.3 e2 studio : RENESAS 社製

- サンプルプログラムのビルド手順

サンプルプログラムのビルド手順は以下の通り。

1. 展開したソースファイルを任意の場所にコピー
2. “RZ/T1 Encoder I/F Configuration Library” (KPIT GCC 版)の以下のファイルを、各フォルダへコピー

```
lib\ecl\r_ecl_rzt1.a  
inc\r_ecl_rzt1_if.h
```
3. e2studio を起動
4. [ウィンドウ]メニュー→[ビューの表示] →[プロジェクト・エクスプローラー]を選択
5. [プロジェクト・エクスプローラー]ビューを右クリックし、ポップアップメニューの[インポート]を選択
6. 「インポート」ダイアログの「一般」 → 「既存プロジェクトをワークスペースへ」 を選択し、「次へ」 ボタンをクリック
7. 「インポートダイアログ」 の「参照」 をクリック
8. 「フォルダの参照」 ダイアログで、ソースファイルのコピー先のトップを選択し、「OK」 をクリック
9. 「インポート」 ダイアログの「プロジェクトをワークスペースにコピー」 のチェックを付ける
10. 「インポート」 ダイアログの「終了」 をクリック
11. [プロジェクト]メニュー→[すべてビルド]を選択

次のファイルが生成される。

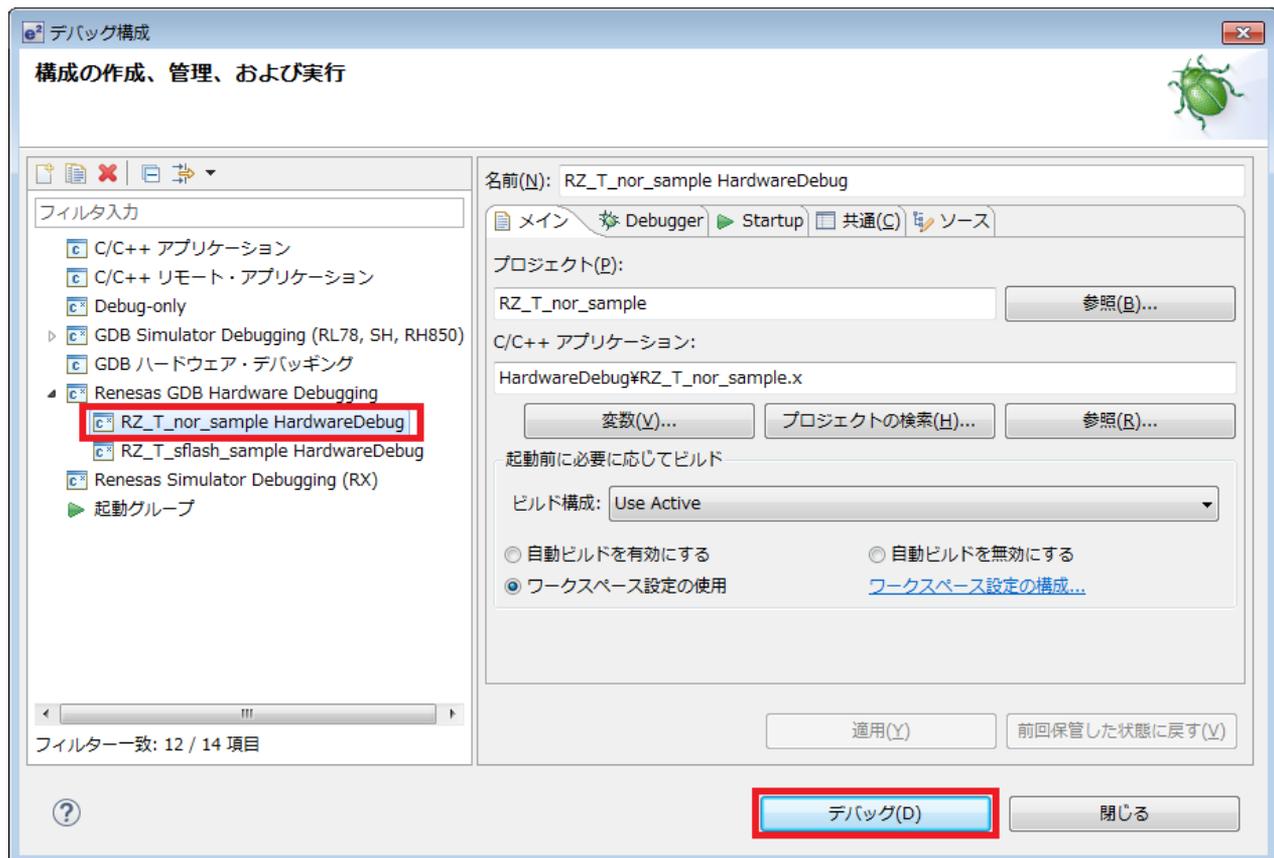
HardwareDebug\RZ_T_nor_sample.x

(シリアルフラッシュの場合は、「RZ_T_sflash_sample.x」)

- サンプルプログラムの実行手順

「ビルド手順」を実行後、ターゲットボード、デバッガを正しく接続し、以下の操作を行う。

1. [プロジェクト] → [すべてビルド] を実行後、[実行] → [デバッグの構成] を選択。
2. 以下の画面の [RZ_T_nor_sample HardwareDebug] を選択、[デバッグ]をクリックするとフラッシュメモリへダウンロードを開始。
(シリアルフラッシュの場合は、「RZ_T_sflash_sample HardwareDebug」)



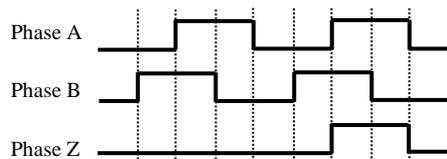
3. [実行] → [再開]をクリックするとサンプルプログラムの実行を開始。

3.4.4 実行結果

サンプルプログラムを実行するとターミナルソフトに下記のように表示されます。

```
ENCOUT sample program start
  EC-Lib Ver.1.2
  ENCOUT Ver.1.0
  ENCOUT driver Ver.1.0
```

また、「3.2 ターゲットボード」に記載された出力ポートに下記のような信号が出力されます。



3.5 仕様

3.5.1 メモリサイズ

メモリサイズの概算値を以下に記載します。

項目		サイズ[bytes]			
		EWARM	DS-5	e2 studio	
ENCOUT ドライバ	コード領域	700	1428	2052	
	データ領域 (初期値あり)	0	0	0	
	データ領域 (初期値なし)	18	18	18	
	定数領域	48	48	48	
	スタック サイズ	R_ENCOUT_Open 関数	8	16	48
		R_ENCOUT_Close 関数	16	24	48
R_ENCOUT_Control 関数		36	96	104	
R_ENCOUT_GetVersion 関数		0	0	4	
RZ/T1 ENCOUT Configuration Data	定数領域	8276			
メインプログラム	コード領域	1084	1752	1836	
	データ領域 (初期値あり)	20	0	20	
	データ領域 (初期値なし)	4	24	4	
	定数領域	328	0	328	

3.5.2 ENCOUT ドライバ API

(1) R_ENCOUT_Open

R_ENCOUT_Open	
概要	ENCOUT ドライバの初期化
ヘッダ	r_encout_rzt1_if.h r_encout_rzt1_dat.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 : R_ENCOUT0_ID 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)

(2) R_ENCOUT_Close

R_ENCOUT_Close	
概要	ENCOUT ドライバの終了
ヘッダ	r_encout_rzt1_if.h r_encout_rzt1_dat.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 : R_ENCOUT0_ID 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)

(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_rzt1_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) R_ENCOUT_Control

R_ENCOUT_Control	
概 要	ENCOUT の操作
ヘッダ	r_encout_rzt1_if.h r_encout_rzt1_dat.h
宣 言	r_encout_err_t R_ENCOUT_Control(const int32_t id, const r_encout_cmd_t cmd, void *const pbuf);
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「(a) R_ENCOUT_CMD_INIT」、「(b) R_ENCOUT_CMD_START」、「(c) R_ENCOUT_CMD_STOP」、「(d) R_ENCOUT_CMD_SET」を参照してください。
引 数	id : R_ENCOUT0_ID を指定してください cmd : 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 pbuf : cmd に依存
リターン値	「(a) R_ENCOUT_CMD_INIT」、「(b) R_ENCOUT_CMD_START」、「(c) R_ENCOUT_CMD_STOP」、「(d) R_ENCOUT_CMD_SET」を参照してください。

(a) R_ENCOUT_CMD_INIT

R_ENCOUT_CMD_INIT

概要	ENCOUT の初期化
ヘッダ	「(4) R_ENCOUT_Control」参照
宣言	「(4) R_ENCOUT_Control」参照
説明	ENCOUT の初期化を行います。 「RZ/T1 グループ エンコーダ分周出力 (ENCOUT) ユーザーズマニュアル」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/T1 グループ エンコーダ分周出力 (ENCOUT) ユーザーズマニュアル」を参照してください。
引数	<p>id : 「(4) R_ENCOUT_Control」参照</p> <p>cmd : R_ENCOUT_CMD_INIT を指定してください</p> <p>pbuf : 設定値を記述した r_encout_init_t 構造体へのポインタを指定します。r_encout_init_t 構造体のメンバ変数は以下です。</p> <p>bool reverse_b : CTL レジスタ POL ビットに設定する値を指定します。false を指定すると 0、true を指定すると 1 を設定します。</p> <p>uint16_t position_max : POSMAX レジスタに設定する値を指定します。詳細は「RZ/T1 グループ エンコーダ分周出力 (ENCOUT) ユーザーズマニュアル」を参照してください。</p> <p>uint16_t encoder_count : エンコーダの初期位置値を 0~ENCODER_RESOLUTION-1 の範囲で設定します。 マクロ「ENCODER_RESOLUTION」については、「3.5.4 設定値の変更方法」参照。</p> <p>uint32_t carrier_period : キャリア周期を ns 単位で指定します。50000~3276999 の範囲で指定可能です。</p>
リターン値	<p>R_ENCOUT_SUCCESS : 正常終了</p> <p>R_ENCOUT_ERR_INVALID_ARG : 異常終了 (引数 id、cmd、encoder_count、carrier_period が規定されていない値、引数 position_max が設定禁止の値)</p> <p>R_ENCOUT_ERR_ACCESS : 異常終了 (ENCOUT ドライバが初期化されていません)</p> <p>R_ENCOUT_ERR_BUSY : 異常終了 (ENCOUT が動作中です)</p>

(b) R_ENCOUT_CMD_START

R_ENCOUT_CMD_START	
概要	ENCOUT の起動
ヘッダ	「(4) R_ENCOUT_Control」参照
宣言	「(4) R_ENCOUT_Control」参照
説明	ENCOUT の起動を行います。 「RZ/T1 グループ エンコーダ分周出力 (ENCOUT) ユーザーズマニュアル」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/T1 グループ エンコーダ分周出力 (ENCOUT) ユーザーズマニュアル」を参照してください。
引数	id : 「(4) R_ENCOUT_Control」参照 cmd : R_ENCOUT_CMD_START を指定してください pbuf : 使用しません
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)

(c) R_ENCOUT_CMD_STOP

R_ENCOUT_CMD_STOP	
概要	ENCOUT の停止
ヘッダ	「(4) R_ENCOUT_Control」参照
宣言	「(4) R_ENCOUT_Control」参照
説明	ENCOUT の停止を行います。
引数	id : 「(4) R_ENCOUT_Control」参照 cmd : R_ENCOUT_CMD_STOP を指定してください pbuf : 使用しません
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)

(d) R_ENCOUT_CMD_SET

R_ENCOUT_CMD_SET

概要	ENCOUT の設定
ヘッダ	「(4) R_ENCOUT_Control」参照
宣言	「(4) R_ENCOUT_Control」参照
説明	ENCOUT 動作中に OUTCNT レジスタの設定を行います。 「RZ/T1 グループ エンコーダ分周出力 (ENCOUT) ユーザーズマニュアル」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/T1 グループ エンコーダ分周出力 (ENCOUT) ユーザーズマニュアル」を参照してください。
引数	id : 「(4) R_ENCOUT_Control」参照 cmd : R_ENCOUT_CMD_SET を指定してください pbuf : 設定値を記述した r_encout_set_t 構造体へのポインタを指定します。 r_encout_set_t 構造体のメンバ変数は以下です。 uint32_t encoder_count : エンコーダの位置値を 0~ ENCODER_RESOLUTION-1 の範囲で設定します。 マクロ「ENCODER_RESOLUTION」については、「3.5.4 設定値の変更方法」参照。
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、encoder_count が規定されていない値) R_ENCOUT_ERR_ACCESS : 異常終了 (ENCOUT が動作していません)

3.5.3 処理概要

「RZ/T1 グループ エンコーダ分周出力 (ENCOUT) ユーザーズマニュアル」の「4.1 初期化」の手順を main.c の encout_start 関数に、「4.2 メイン処理」の手順を main.c の encout_main 関数に実装しています。ただし、「4.2 メイン処理」に記載されている「1 位置情報の取得」の処理については、位置情報の取得の代わりに配列 encoder_data の値を順番に参照しています。また、「2 制御処理 (システム依存)」の処理については、実装されていません。

詳細は「RZ/T1 グループ エンコーダ分周出力 (ENCOUT) ユーザーズマニュアル」を参照してください。

下記にサンプルプログラムの初期化処理を行う main 関数とキャリア周期毎に定期的に起動してメイン処理を行う timer_isr 関数のフローチャートを記載します。



図 3.1 main 関数のフローチャート

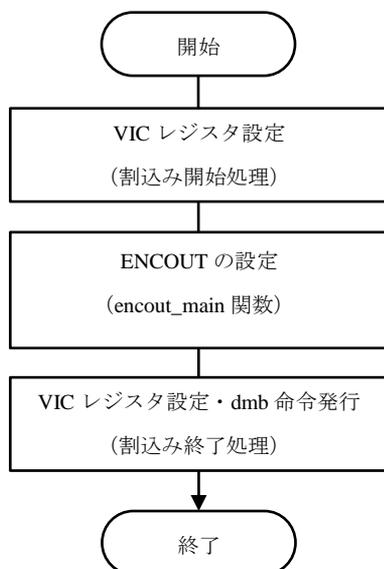


図 3.2 timer_isr 関数のフローチャート

3.5.4 設定値の変更方法

ENCOUT サンプルプログラムの設定値は以下のように変更できます。

設定値	ファイル	変更方法
エンコーダの解像度	r_encout_rzt1_config.h	ENCOUT ドライバが位置の計算に用いるエンコーダの解像度を設定できます。 マクロ「ENCODER_RESOLUTION」にエンコーダの解像度を設定してください。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。
キャリア周期	main.c	ENCOUT が出力する ABZ 相信号のキャリア周期を設定できます。 マクロ「CARRIER_PERIOD」にキャリア周期を ns 単位で指定してください。設定範囲は 50000 ~ 3276999 となります。デフォルト値は 100000ns (100 μ s) です。
B 相の極性	main.c	ENCOUT が出力する ABZ 相信号の B 相の極性を設定できます。 マクロ「REVERSE_B」を false に設定すると、B 相が正相、true に設定すると、B 相が逆相になります。デフォルト値は false です。
最大位置	main.c	ENCOUT が出力する ABZ 相信号の最大位置を設定できます。 マクロ「POSITION_MAX」に最大位置を設定してください。この値が ENCOUT の POSMAX レジスタに設定されます。設定値の範囲は、「RZ/T1 グループ エンコーダ分周出力 (ENCOUT) ユーザーズマニュアル」の「2.3 最大位置レジスタ (POSMAX)」を参照してください。デフォルト値は 99 です。
ABZ 相出力端子	main.c	サンプルプログラムが使用する ABZ 相出力端子を変更することができます。 port_reset 関数と port_set 関数を変更してください。設定可能な端子と詳しい設定方法は、「RZ/T1 グループ エンコーダ分周出力 (ENCOUT) ユーザーズマニュアル」の「1.2 入出力端子」、「1.3 入出力端子と I/O ポートの対応」、「4.1 初期化」を参照してください。デフォルトは POUTA0、POUTB0、POUTZ0 を使用する設定になっています。

3.6 Encoder I/F と ENCOUT の組み合わせ方法

Encoder I/F と ENCOUT のサンプルプログラムを組み合わせる方法を記載します。

1. Encoder I/F のサンプルプログラムに対して、ENCOUT のサンプルプログラムの下記のファイルをコピーしてください。

<ul style="list-style-type: none"> Top folder <ul style="list-style-type: none"> inc <ul style="list-style-type: none"> iodefine_encout.h ENCOUT レジスタ定義ファイル r_encout_rzt1_dat.h r_encout_rzt1.dat 用ヘッダファイル r_encout_rzt1_if.h ENCOUT ドライバヘッダファイル lib <ul style="list-style-type: none"> ecl <ul style="list-style-type: none"> r_encout_rzt1.dat RZ/T1 ENCOUT Configuration Data src <ul style="list-style-type: none"> drv <ul style="list-style-type: none"> encout <ul style="list-style-type: none"> r_encout_rzt1_config.h ENCOUT ドライバファイル r_encout_rzt1.c ENCOUT ドライバファイル sample <ul style="list-style-type: none"> encout_dat.s Configuration data 用リンカ設定ファイル ※1 	
---	--

※1 : DS-5/e2 studio 用ファイル
 DS-5 : encout_dat.s
 e2 studio : encout_dat.asm

IAR システムズ社製 EWARM を使用する場合は、r_encout_rzt1.c をプロジェクトに追加してください。「drv」グループの下に「encout」グループを作成して「encout」グループに追加することを推奨します。メニューの「プロジェクト」→「オプション」→「リンカ」カテゴリ→「追加オプション」タブの「コマンドラインオプションの使用」にチェックを入れ、「コマンドラインオプション」に「--image_input \$PROJ_DIR\$¥lib¥ecl¥r_encout_rzt1.dat,g_encout_conf,ENCOUT_CONF_SEC,4」と入力してください。

2. main.c の R_ECL_Configure 関数呼び出しを以下のように変更してください。

変更前	変更後
<pre>... extern const uint32_t g_xxx_config[]; ... ret_code = R_ECL_Configure(g_xxx_config); ...</pre>	<pre>... extern const uint32_t g_xxx_config[]; extern const uint32_t g_encout_config[]; ... const void *conf_array[2] = { g_xxx_config, g_encout_config }; ret_code = R_ECL_ConfigureMulti(conf_array, 2); ...</pre>

※ 「xxx」の部分はエンコーダの種類によって異なります。

3. main.c を「3.5.2 ENCOUT ドライバ API」、「3.5.3 処理概要」、「3.5.4 設定値の変更方法」を参考に修正してください。
4. Encoder I/F のビルド及び実行手順に従って、プログラムを実行してください。

注意 : Encoder I/F と ENCOUT を組み合わせるためには、「RZ/T1 Encoder I/F Configuration Library」の Ver.2.0 Preliminary 以降が必要になります。

4. 制限事項

特になし。

5. 注意事項

5.1 処理時間

ENCOUT サンプルプログラムでは、制御ループにおけるユーザーが使用可能な時間は以下のようになっています。ご使用の環境で問題がないか十分評価を行ってください。

キャリア周期を 62.5us とした場合の例を以下に示します。

62.5us のうち、ENCOUT サンプルプログラムの使用時間は約 1us (1.6%) となり、ユーザーが使用可能な時間は約 61.5us (98.4%) となります。

処理		時間	占有率
ENCOUT サンプルプログラムの処理 ※	OUTCNT レジスタ設定時間	約 1us	1.6%
	ユーザー使用可能時間	約 61.5us	98.4%

※：初期設定時間は除きます。