RZ/T2M Group

ENCOUT sample program

Summary

This document describes the RZ/T2M ENCOUT sample program package. To use this sample program, please obtain “RZ/T2M Group Encoder I/F Configuration Library” release package (Rev.2.00 or later).

Functionality Checked Device

RSK+RZT2M Board (RTK9RZT2M0C00000BE)

Contents

1. Package Contents

This package contains the following contents.

1.1 Software

- Source Code

No.	Name	Version number
1	RZ/T2M ENCOUT sample program	2.0

- Configuration Data

No.	Name	Version number
1	RZ/T2M ENCOUT Configuration Data	1.0
2	RZ/T2M ENCOUT PINMUX Data	1.0

1.2 Document

No.	Document name	Version	File name
1	RZ/T2M ENCOUT sample program Release Note	2.0	(j) r01an6416jj0200-rzt2m.pdf (e) r01an6416ej0200-rzt2m.pdf (this document)
2	RZ/T2M Group Encoder Divided-Output Module (ENCOUT) Application Note	1.0	(j) r01an6394jj0100-rzt2m_encout.pdf (e) r01an6394ej0100-rzt2m_encout.pdf

2. File Structure

The file structure and contents of this package are detailed below.

```
Top
├──r01an6416jj08200-rzt2m.pdf
├──r01an6416ej0200-rzt2m.pdf
└──workspace
    ├──Documentation
    │   ├──r01an6394jj0100-rzt2m_encout.pdf
    │   └──r01an6394ej0100-rzt2m_encout.pdf
    └──Software
        ├──iccarm
        │   └──RZ_T2_encout.zip : RZ/T2M ENCOUT sample program set (IAR)
        └──gcc
            └──RZ_T2_encout.zip : RZ/T2M ENCOUT sample program set (e2 studio)
```

The file structure of the RZ_T2_encout.zip is shown below.

```
Top folder
├── configuration.xml                    : FSP Configuration Data
├── ( Build Tool Dependent Environment File )
├── lib
│   └── ecl
│       ├── Config_Encout_v1.0_original.dat  : RZ/T2M ENCOUT Configuration Data
│       └── RZT2_pinmux_v1.0.bin             : RZ/T2M ENCOUT Pin Configuration Data
└── src
    ├── hal_entry.c                      : ENCOUT sample program
    ├── encout_main.c                    : ENCOUT sample program
    ├── encout_dat.asm                   : ENCOUT Linker setup file (only e2 studio)
    ├── siochar.c                        : SCI_UART sample program
    ├── siorw.c                          : SCI_UART sample program
    ├── sio_char.h                       : SCI_UART sample program
    └── drv
        ├── src
        │   └── r_encout_rzt2.c          : ENCOUT driver file
        └── inc
            ├── iodefine_encout.h        : ENCOUT register definition file
            ├── r_encout_rzt2_config.h   : Configuration header file of ENCOUT driver file
            └── r_encout_rzt2_if.h       : APIs of ENCOUT driver file
```

3. About ENCOUT Sample Program

This section contains information necessary to use the complete set of ENCOUT sample program.

3.1 Operating Environment

The sample code covered in this application note is for the environment below.

Table 3.1 Operating Environment

Item	Description
Microcomputer	RZ/T2M Group
Operating frequency	CPU Core0 : 800MHz(Arm® Cortex®-R52)
Operating voltage	1.1 V (Core) / 1.8 V (PLL, etc.) / 3.3 V (I/O)
Integrated development environment	IAR Systems Embedded Workbench® for ARM RENESAS e² studio
Board	RSK+RZT2M (RTK9RZT2M0C00000BE)
Devices (Function to be used on the board)	None

Table 3.2 shows peripheral Functions and Applications.

Table 3.2 Peripheral Functions and Applications

Peripheral Function	Application
Encoder Divided-Output Module （ENCOUT)	Generates Phase A, B and Z outputs corresponding to the positional values.
Interrupt Controller (ICU)	Controls the ENCOUT interrupts.
General PWM Timer (GPT) Channel 0	Generates events at fixed intervals for input to the ELC, and timer interrupts.
Event Link Controller (ELC)	Inputs GPT channel 0 event to the ENCOUT module.
Serial Communications Interface (SCI) UART	Makes COM port communications by using USB interface.

The connection between the host PC and the target board is as follows.

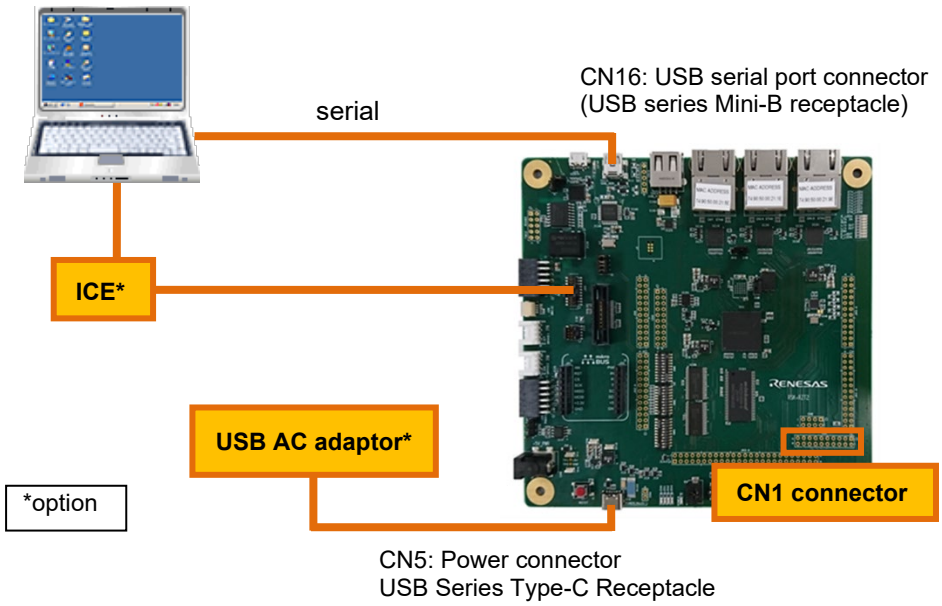


Figure 3.1 Operating Environment (EWARM)

3.2 Software Information

3.2.1 Base OS

This sample program is OS-independent.

3.2.2 Memory Size

Memory size used by this sample program, ENCOUT driver, and configuration data is shown in following table. This table does not include memory size used by Encoder I/F Configuration Library, Flexible Software Package, or C language libraries of the compiler.

Items		Memory Size	
		EWARM [kBytes]	e² studio [kBytes]
ENCOUT driver	Code	0.7	0.7
	Data (with initial value)	0.0	0.0
	Data (without initial value)	0.0	0.0
	Constant Data	0.0	0.1
ENCOUT configuration data	Constant Data	8.4	8.4
Sample program	Code	1.8	1.9
	Data (with initial value)	0.0	0.0
	Data (without initial value)	0.0	0.0
	Constant Data	0.6	0.6

3.3 Hardware Information

3.3.1 Device

RZ/T2M

3.3.2 Target Board

(1) Board Name

RSK+RZT2M (RTK9RZT2M0C00000BE)

(2) Setting of the Target Board

The target board configuration is as follows.

SW4-1: ON

SW4-2: OFF

SW4-3: ON

SW4-4: ON

SW4-5: OFF

SW6-1: OFF

(3) Used Pins of the Target Board

Table 3.3 shows the pins used as the phase A, B and Z output ports, and their functions.

Table 3.3 Used pins and their function

Channel	Signal Name (Pin Name)	Pin Header	Input/Output	Description
ENCOUT0	POUTA0 (ENCIF2)	CN1 #3	Output	Phase A output pin
	POUTB0 (ENCIF3)	CN1 #4	Output	Phase B output pin
	POUTZ0 (ENCIF4)	CN1 #6	Output	Phase Z output pin
ENCOUT1	POUTA1 (ENCIF7)	CN1 #9	Output	Phase A output pin
	POUTB1 (ENCIF8)	CN1 #11	Output	Phase B output pin
	POUTZ1 (ENCIF9)	CN1 #12	Output	Phase Z output pin

3.4 Procedures on Development Environments

3.4.1 Preparation before Executing the Sample Program

In this sample program, communicates with the host PC. The USB connection terminal on the target board is CN16. The terminal software of the host PC is set as shown in the following table.

Function	Setting
Communication method	Asynchronous serial transmit / receive
Sending / receiving order	LSB first
Transfer rate	19200 bps
Character length	8 bits
Stop bit length	1 bit
Parity function	None
Hardware flow control	None

3.4.2 EWARM from IAR Systems

(1) Build Environment

IAR Embedded Workbench for ARM v9.50.1

RENESAS RZ/T2 Flexible Software Package v2.0.0

(2) Execution Environment ICE

IAR I-jet

(3) Build Procedure for Sample Programs

The build procedure for the sample program is as follows.

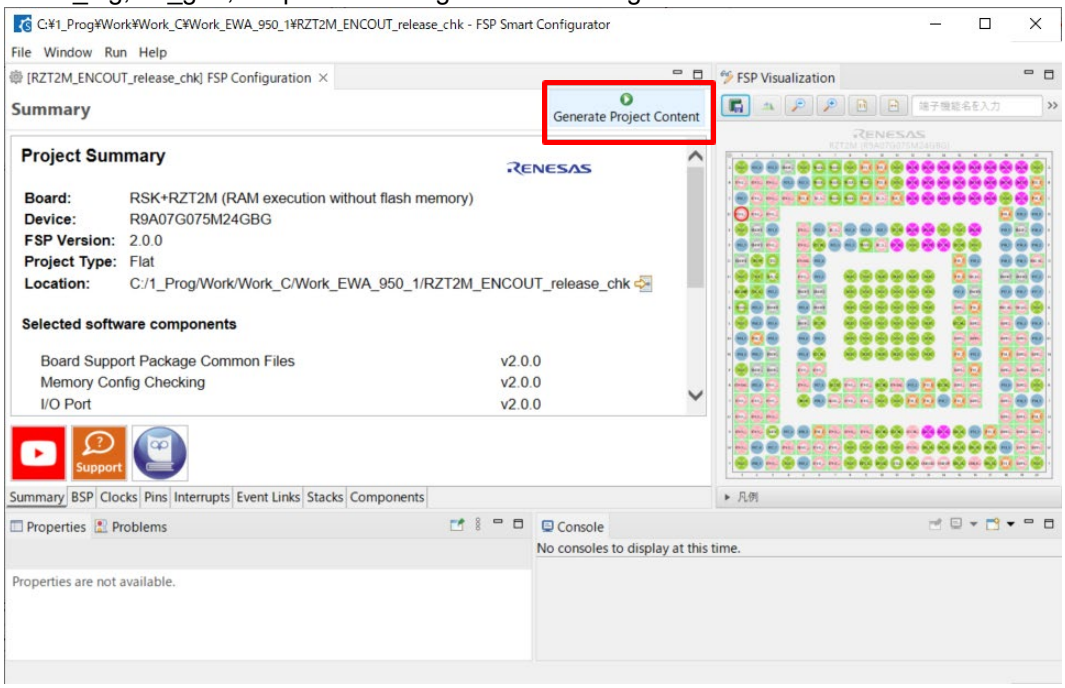
1 Copy the extracted source files to the desired location.
2 Copy the following files from “RZ/T2M Group Encoder I/F Configuration Library” under lib¥ecl in the same folder as the source files. (If the “lib¥ecl” folder already exists, overwrite it.) *1
lib¥ecl¥r_ecl_rzt2_iar.a
lib¥ecl¥r_ecl_rzt2_if.h
3 Activate EWARM.
4 Select [File] menu -> [Open Workspace].
5 Open the extracted source file RZ_T2_encout.eww.
6 Start the FSP Smart Configurator from the [Tools] menu of the EWARM IDE. *2

Note: 1. Please use Encoder I/F Configuration Library revision 2.00 or later.
2. The following procedure adds the activation of the FSP Smart Configurator to the [Tools] menu of the EWARM IDE. Select [Tools] menu -> [Tool Configuration] in the EWARM IDE. Select the [New] button, specify a table string in each field, and press [OK].

Field	String
Menu text	RZ Smart Configurator
Command	$RASC_EXE_PATH$
Argument	--compiler IAR configuration.xml
Initial directory	$PROJ_DIR$

String for the command is variable holding the path of the Smart Configurator execution file, rasc.exe.
You can also start the FSP Smart Configurator directly from the command prompt by specifying the folder where it is installed.

7 In the FSP Configuration pane of the Smart Configurator, click Generate Project Content. The rzt, rzt_cfg, rzt_gen, script and .setting folders will be generated.



8 When project generation is complete, close the Smart Configurator.

9 Select [Rebuild ALL] from the [Project] menu of EWARM.
The file Debug\Exe\RZ_T2_encout.out is generated.

(4) Sample Program Execution Procedure

After executing the “build procedure”, connect the target board and debugger correctly, and perform the following operations.

1 Select [Project] menu -> [Download and Debug].

2 Select [Debug] menu -> [Execute].

3.4.3 e² studio from RENESAS

(1) Build Environment

RENESAS e² studio 2024-01.1

GNU ARM Embedded Toolchain 12.2.1.arm-12-24

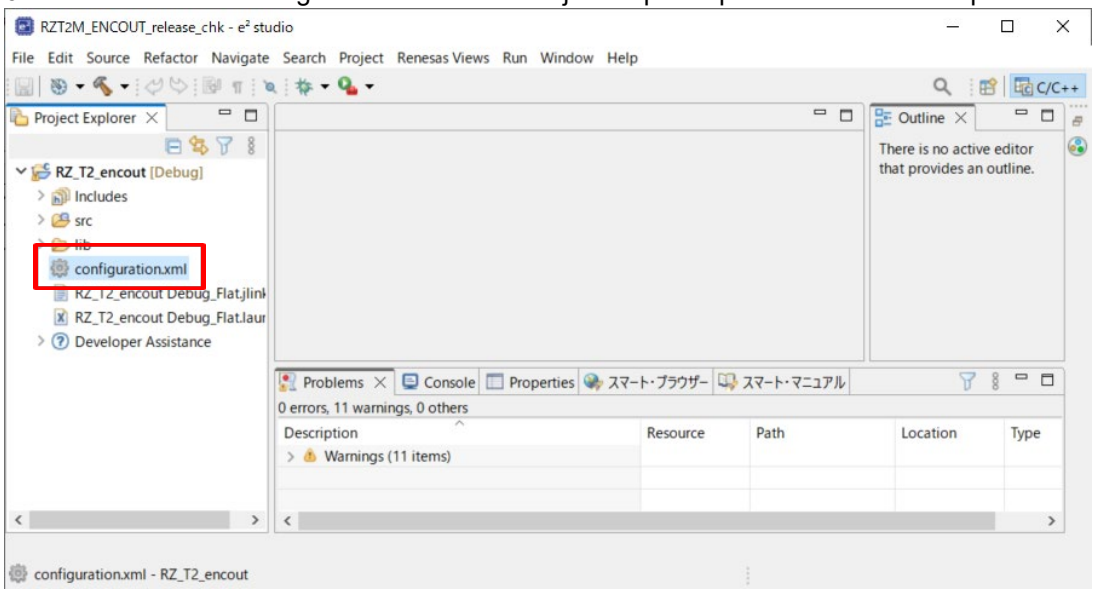
RENESAS RZ/T2 Flexible Software Package v2.0.0

(2) Execution Environment ICE

SEGGER J-Link v7.94h

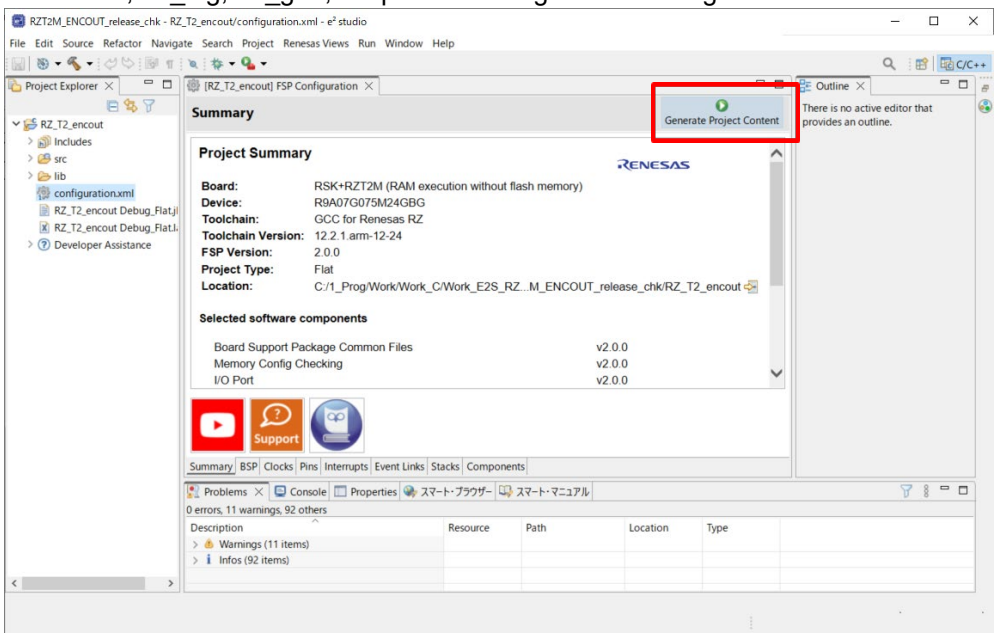
(3) Build Procedure for Sample Programs

1 The build procedure for the sample program is as follows.
2 Copy the following files in the "RZ/T2M Group Encoder I/F Configuration Library" to lib¥ecl in the same folder as the source file. (If the lib¥ecl folder already exists, overwrite it.) *1
lib¥ecl¥r_ecl_rzt2_gcc.a
lib¥ecl¥r_ecl_rzt2_if.h
3 After launching e² studio and moving to the workspace, click the [File] menu -> [Import] and select Existing project to workspace and click [Next].
4 On the project import screen, select the folder where the sample program was expanded as the root directory.
5 Select a project, check Copy Project to Workspace, and click [Finish].
6 Double-click the configuration.xml in the Project Explorer pane of e² studio to open it.



Note 1. Please use Encoder I/F Configuration Library revision 2.00 or later.

7 Click Generate Project Content in the FSP Configuration pane of e² studio.
The rzt, rzt_cfg, rzt_gen, script and .settings folders are generated.



8 Select [Project] menu -> [Build All].
The Debug\RZ_T2_encout.elf file is generated.

(4) Execution Procedure of the Sample Program

After executing the "build procedure", connect the target board and debugger correctly, and perform the following operations.

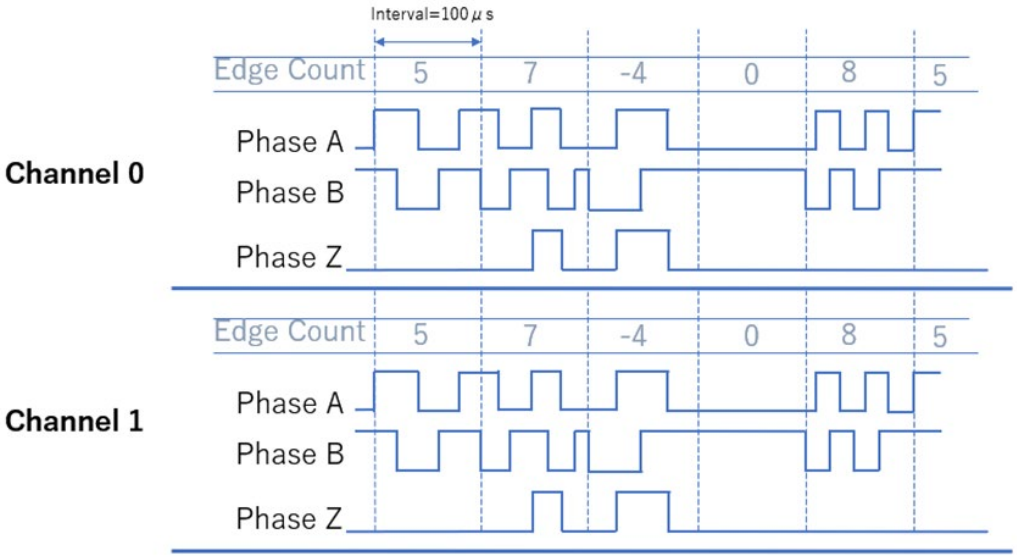
1 Select [Run] menu → [Debug As] → [Renesas GDB Hardware Debugging].
2 Click [Debug] to start downloading to internal RAM.
3 Click [Run] menu → [Resume] to run the sample program.

3.4.4 Execution Result

After executing the sample program, the following will be displayed on the terminal software.

```
ENCOUT sample program start
EC-Lib      Ver.2.0.0
ENCOUT drv  Ver.2.0
ENCOUT0 dat Ver.1.0
ENCOUT1 dat Ver.1.0
```

The following signals are output to the output port described in 3.3.2(3) Used Pins.



Note: The channel 1 starts 50 μs (carrier cycle * 1/2) earlier than the channel 0.

Figure 3.2 ABZ phase output signal

3.5 Sample Program Specifications

3.5.1 ENCOUT Driver

Table 3.4 lists the ENCOUT Driver functions.

Table 3.4 ENCOUT Driver Function List

Layer / Block	Function Name	Chapter
ENCOUT Driver	R_ENCOUT_Open	3.5.2.1
	R_ENCOUT_Close	3.5.2.2
	R_ENCOUT_GetVersion	3.5.2.3
	R_ENCOUT_Control	3.5.2.4

3.5.2 ENCOUT Driver API

3.5.2.1 R_ENCOUT_Open

R_ENCOUT_Open		
Synopsis	ENCOUT driver initialization	
Header	r_encout_rzt2_if.h	
Declaration	r_encout_err_t R_ENCOUT_Open(const int32_t id);	
Description	Initialize the ENCOUT driver. Be sure to call this function before using the ENCOUT driver.	
Argument	id	: Specify R_ENCOUT_ID_0 or R_ENCOUT_ID_1
Return value	R_ENCOUT_SUCCESS	: Normal termination
	R_ENCOUT_ERR_INVALID_ARG	: Abnormal termination (value for which the argument id is not specified)
	R_ENCOUT_ERR_ACCESS	: Abnormal termination (ENCOUT driver has already been initialized)

3.5.2.2 R_ENCOUT_Close

R_ENCOUT_Close		
Synopsis	Close ENCOUT driver	
Header	r_encout_rzt2_if.h	
Declaration	r_encout_err_t R_ENCOUT_Close(const int32_t id);	
Description	Close ENCOUT driver If this function is called while ENCOUT is operating, the end processing is executed after the ENCOUT stop processing.	
Argument	id	: Specify R_ENCOUT_ID_0 or R_ENCOUT_ID_1
Return value	R_ENCOUT_SUCCESS	: Normal termination
	R_ENCOUT_ERR_INVALID_ARG	: Abnormal termination (value for which the argument id is not specified)

3.5.2.3 R_ENCOUT_GetVersion

R_ENCOUT_GetVersion	
Synopsis	Acquire the version number of the ENCOUT driver.
Header	r_encout_rzt2_if.h
Declaration	uint32_t R_ENCOUT_GetVersion(void);
Description	This function acquires the version number of theENCOUT driver.
Argument	None
Return value	Version information : The upper 16 bits store the major version and the lower 16 bits store the minor version. eg:) If the return value is 0x00010002, Ver.1.2

3.5.2.4 R_ENCOUT_Control

R_ENCOUT_Control	
Synopsis	Controlling operation of the ENCOUT
Header	r_encout_rzt2_if.h
Declaration	r_encout_err_t R_ENCOUT_Control(const int32_t id, const r_encout_cmd_t cmd, void *const pbuf);
Description	Operate ENCOUT. This function controls operations of the ENCOUT by using the cmd argument. See (1) R_ENCOUT_CMD_INIT, (2) R_ENCOUT_CMD_START, (3) R_ENCOUT_CMD_STOP and (4) R_ENCOUT_CMD_SET.
Argument	id : Specify R_ENCOUT_ID_0 or R_ENCOUT_ID_1 cmd : Specify one of R_ENCOUT_CMD_INIT, R_ENCOUT_CMD_START, R_ENCOUT_CMD_STOP, R_ENCOUT_CMD_SET. pbuf : Depends on cmd
Return value	See (1) R_ENCOUT_CMD_INIT, (2) R_ENCOUT_CMD_START, (3) R_ENCOUT_CMD_STOP and (4) R_ENCOUT_CMD_SET.

(1) R_ENCOUT_CMD_INIT

R_ENCOUT_CMD_INIT		
Synopsis	ENCOUT initialization	
Header	See 3.5.2.4 R_ENCOUT_Control	
Declaration	See 3.5.2.4 R_ENCOUT_Control	
Description	Initialize ENCOUT. Perform "2. Making initial settings for the ENCOUT" and "3. Setting the initial value in POSCNT " of the initialization procedure described in "4.1 Initialization" of "RZ / T2M Group Encoder Divided-Output Module (ENCOUT) Application Note". See the RZ / T2M Group Encoder Divided-Output Module (ENCOUT) Application Note for more information.	
Argument	id	: See 3.5.2.4 R_ENCOUT_Control
	cmd	: Specify R_ENCOUT_CMD_INIT
	pbuf	: Specify a pointer to the r_encout_init_t structure that describes the setting value. The member variables of the r_encout_init_t structure are:
		uint8_t control_param : CTL register Specifies the value to be set in the POL bit, ZW bit, and ZS bit. Specify using the ENCOUT_CTL_ * macro.
		uint16_t position_max : Specifies the value to set in the POSMAX register. See the RZ / T2M Group Encoder Divided-Output Module (ENCOUT) Application Note for more information.
		uint16_t encoder_count : Set the initial position value of the encoder in the range of 0 to ENCODER_RESOLUTION-1. (ENCODER_RESOLUTION is the encoder resolution of each channel) For the macro "ENCODER_RESOLUTION", refer to "3.5.4 Modification Method of Setting Value".
		uint32_t carrier_period : Specify the carrier cycle in ns. It can be specified in the range of 50000 to 3276750. Specify in multiples of 50.
Return value	R_ENCOUT_SUCCESS	: Normal termination
	R_ENCOUT_ERR_INVALID_ARG	: Abnormal termination (values where the arguments id, cmd, encoder_count, carrier_period are not specified, and the argument position_max is a value whose setting is prohibited)
	R_ENCOUT_ERR_ACCESS	: Abnormal termination (ENCOUT driver has not been initialized)
	R_ENCOUT_ERR_BUSY	: Abnormal termination (ENCOUT is operating)

(2) R_ENCOUT_CMD_START

R_ENCOUT_CMD_START		
Synopsis	Start ENCOUT	
Header	See 3.5.2.4 R_ENCOUT_Control	
Declaration	See 3.5.2.4 R_ENCOUT_Control	
Description	Start ENCOUT. Perform "5. Starting the AB-phase and Z output" of the initialization procedure described in "4.1 Initialization" of " RZ / T2M Group Encoder Divided-Output Module (ENCOUT) Application Note ". See the RZ / T2M Group Encoder Divided-Output Module (ENCOUT) Application Note for more information.	
Argument	id : See 3.5.2.4 R_ENCOUT_Control	
	cmd : Specify R_ENCOUT_CMD_START	
	pbuf : No use	
Return value	R_ENCOUT_SUCCESS	: Normal termination
	R_ENCOUT_ERR_INVALID_ARG	: Abnormal termination (value for which arguments id and cmd are not specified)
	R_ENCOUT_ERR_ACCESS	: Abnormal termination (ENCOUT driver has not been initialized)
	R_ENCOUT_ERR_BUSY	: Abnormal termination (ENCOUT is operating)

(3) R_ENCOUT_CMD_STOP

R_ENCOUT_CMD_STOP		
Synopsis	Stop ENCOUT	
Header	See 3.5.2.4 R_ENCOUT_Control	
Declaration	See 3.5.2.4 R_ENCOUT_Control	
Description	Stop ENCOUT	
Argument	id : See 3.5.2.4 R_ENCOUT_Control	
	cmd : Specify R_ENCOUT_CMD_STOP	
	pbuf : No use	
Return value	R_ENCOUT_SUCCESS	: Normal termination
	R_ENCOUT_ERR_INVALID_ARG	: Abnormal termination (value for which arguments id and cmd are not specified)
	R_ENCOUT_ERR_ACCESS	: Abnormal termination (ENCOUT driver has not been initialized)

(4) R_ENCOUT_CMD_SET

R_ENCOUT_CMD_SET		
Synopsis	Setting ENCOUT	
Header	See 3.5.2.4 R_ENCOUT_Control	
Declaration	See 3.5.2.4 R_ENCOUT_Control	
Description	Set the OUTCNT register during ENCOUT operation. Perform "3. Calculating the value in the OUTCNT register" and "4. Setting the OUTCNT register " in the procedure described in "4.2 Main processing" of " RZ / T2M Group Encoder Divided-Output Module (ENCOUT) Application Note ". See the RZ / T2M Group Encoder Divided-Output Module (ENCOUT) Application Note for more information.	
Argument	id	: See 3.5.2.4 R_ENCOUT_Control
	cmd	: Specify R_ENCOUT_CMD_SET
	pbuf	: Specify a pointer to the r_encout_set_t structure that describes the set value. The member variables of the r_encout_set_t structure are:
	uint32_t encoder_count	: Set the initial position value of the encoder in the range of 0 to ENCODER_RESOLUTION-1. (ENCODER_RESOLUTION is the encoder resolution of each channel) For the macro "ENCODER_RESOLUTION", refer to "3.5.4 Modification Method of Setting Value".
Return value	R_ENCOUT_SUCCESS	: Normal termination
	R_ENCOUT_ERR_INVALID_ARG	: Abnormal termination (value without arguments id, cmd, encoder_count specified)
	R_ENCOUT_ERR_ACCESS	: Abnormal termination (ENCOUT is not working)

3.5.3 Processing Overview

The procedure of "4.1 Initialization" in " RZ/T2M Group Encoder Divided-Output Module (ENCOUT) Application Note" is implemented in the enc_main function of encout_main.c. The procedure of "4.2 Main processing" is implemented in the encout_main function of encout_main.c. However, for the process of "1 Acquiring positional information" described in "4.2 Main processing", the value of the array encoder_data is referenced in order instead of the acquisition of location information. Also, the processing of "2 Control processing (system dependent)" is not implemented.

See the RZ/T2M Group Encoder Divided-Output Module (ENCOUT) Application Note for more information.

Figure 3.3 shows the flowchart of the enc_main function that initializes the sample program.

Figure 3.4 shows the flowcharts of the int_gpt0_ovf function (for channel 0) and the int_gpt0_ccmpa function (for channel 1) that are started periodically for each carrier cycle and perform the main process.

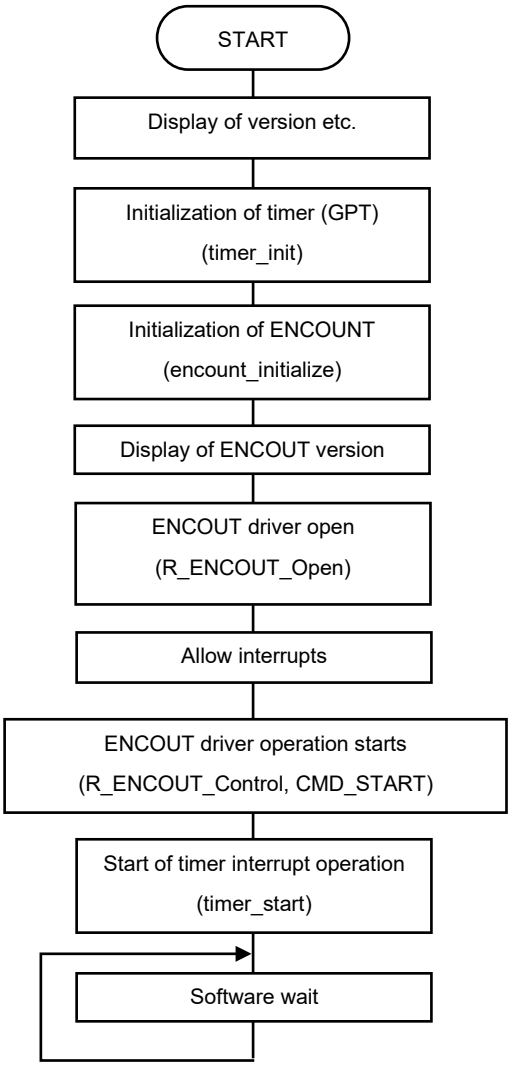


Figure 3.3 Flowchart of enc_main function

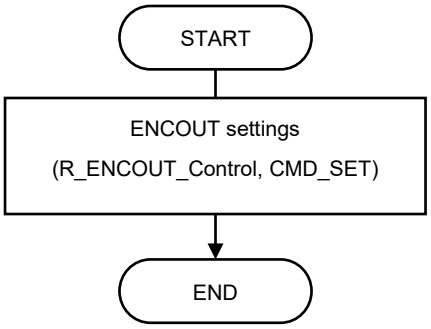


Figure 3.4 Flowchart of int_gpt0_ovf and int_gpt0_ccmpa functions

3.5.4 Modification Method of Setting Value

The setting value of the ENCOUT sample program can be changed as follows.

Setting value	file	Modification method
Encoder resolution ENCODER_RESOLUTION0 ENCODER_RESOLUTION1	r_encout_rzt2_config.h	You can set the resolution of the encoder that the ENCOUT driver uses to calculate the position. Set for each channel. Set the encoder resolution in the macro "ENCODER_RESOLUTIONn" (where n is the channel number). You can set a 32-bit value other than 0. For example, if the encoder resolution is 20 bits (position values are 0 to 1048575), specify 1048576. The default value is 1048576.
Carrier cycle CARRIER_PERIOD	encout_main.c	You can set the carrier period of the ABZ phase signal output by ENCOUT. Specify the carrier cycle in ns in the macro "CARRIER_PERIOD". The setting range is 50000 to 3276750. The default value is 100000ns (100µs). Set it as an integral multiple of 50.
ENCOUT control ENCOUT_CTL	encout_main.c	You can set the operation content of the ABZ phase signal output by ENCOUT. Set the operation content in the macro "ENCOUT_CTL". This value is set in the ENCOUT CTL register. It is possible to specify the direction of the B phase (positive phase, negative phase) and the phase to be synchronized with the Z pulse width (0,1,2,3,4) (A phase, B phase). For details, refer to "2.1 Control Register (CTL)" in " RZ / T2M Group Encoder Divided-Output Module (ENCOUT) Application Note ". The default settings are positive phase for phase B and 2 pulse output for phase Z.
Maximum position POSITION_MAX	encout_main.c	You can set the maximum position count for one revolution. (Number of edges – 1) Set the maximum position in the macro "POSITION_MAX". This value is set to the POSMAX bit of the POSMAX_PERIOD register in the ENCOUT. Number of edges for one revolution are multiple of 4 (4N), and thus setting value of maximum position is (4N – 1). Available settings are 15, 19, 23, …, 65531, 65535. Default value is 99.
Specifying the output channel ENCOUT_ID	encout_main.c	You can set the ENCOUT channel used by the sample program. You can specify R_ENCOUT_ID_0, R_ENCOUT_ID_1, or both channels (R_ENCOUT_ID_0 | R_ENCOUT_ID_1). The default setting is to use both channels.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	May.31.2022	-	First Edition issued
1.10	Jun. 30.2022	3	Chapter 1.1 Modify Version number of Source code
		3	Chapter 1.2 Modify File name
		4	Chapter 2 Modify the file structure and contents
		5	Table 3.1: Supports Flexible Software Package Version 1.0.0
		7	Modify 3.3.1 Chapter name
		7	Modify 3.3.2 Chapter name
		8	Modify Table 3.4
		-	Deleted 3.7.1 Memory size
1.20	May 31.2023	2, 5	Appended section 3.1.1 for memory size information.
		3, 4	Updated the release note version number.
		-	Remove unused folders of the zip sample program set.
1.30	Oct.6.2023	1, 7, 9	Append notes for the Encoder I/F Configuration Library revision.
		1, 4, 6	Updated the target board name.
		2, 3	Updated the source code, documents revisions and the document filenames. Revised to control GPT, ELC from FSP API. Updated the R_ENCOUT_GetVersion return value to “1.2”.
		3	Updated file structure. (Zip file name is changed. Removed RZ/T2M Pin Configuration data from zip file. Updated ENCOUT configuration data filenames.)
		5	Updated memory size information.
		6	Updated description of the CPU board setting.
		7 to 10	Updated build environment for FSP v1.3.0. Descriptions for development procedures.
		17, 18	Flowchart of the enc_main and timer interrupt function names are revised according to the code update.
		19	Description for the POSITION_MAX modification method is updated.
2.00	Jun 28.2024	1, 4, 6	Updated description of the board name.
		2, 3	Updated the source code, documents revisions and the document filenames. Updated the R_ENCOUT_GetVersion return value to “2.0”.
		5	Updated memory size information.
		7 to 10	Updated build environment for FSP v2.0.0. Figures are replaced.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.