

## RZ/N2H Group

### ENCOUT sample program

---

#### Introduction

This application note describes a sample program for generating Phase A, Phase B, and Phase Z (index) output signals corresponding to the specified positional values by using the encoder divided-output (ENCOUT) module of the RZ/N2H.

The major features of the program are listed below.

- Generates Phase A, Phase B, and Phase Z output signals by periodically updating positional values.
- Timer module events are entered, and edge of the output signals are adjusted by the events.

#### Target Device

RZ/N2H

---

**Table of Contents**

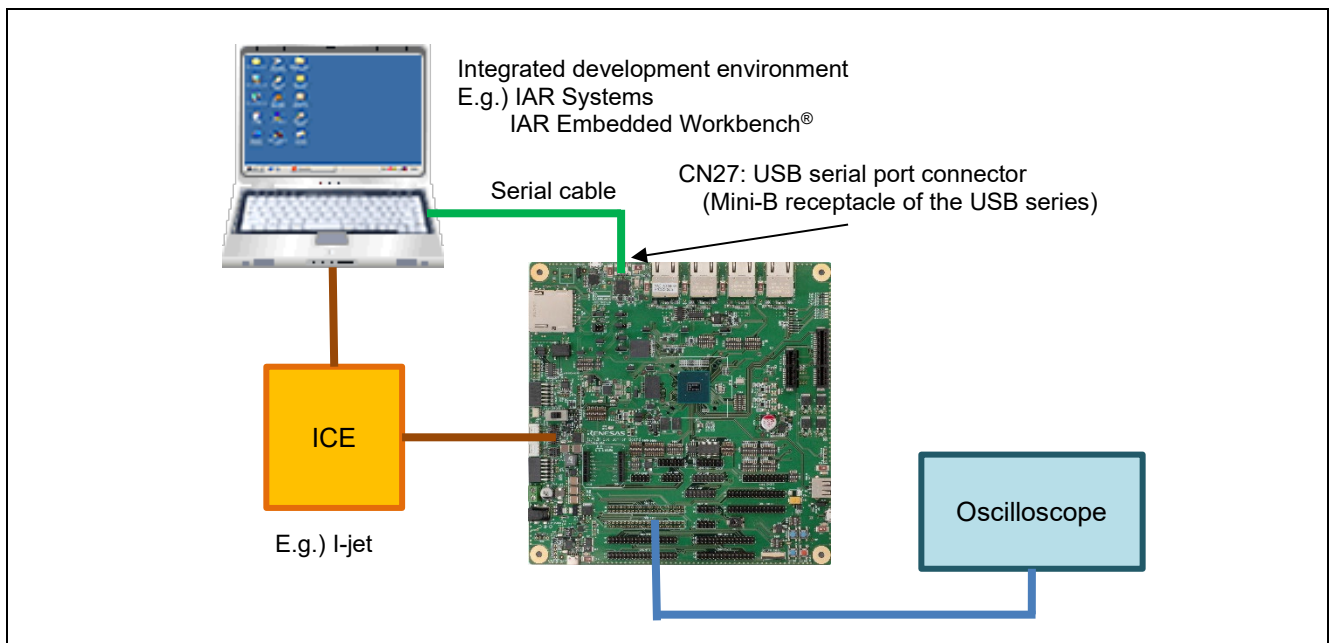
1. Specifications .....	3
2. Operating Environment.....	4
3. Peripheral Functions.....	5
3.1 Pins.....	5
4. Software .....	6
4.1 ENCOUT Driver Function .....	6
4.2 File Structure .....	6
4.3 Functions .....	6
4.4 Specifications of API Functions.....	7
4.4.1 R_ENCOUT_Open.....	7
4.4.2 R_ENCOUT_Close .....	7
4.4.3 R_ENCOUT_GetVersion.....	7
4.4.4 R_ENCOUT_Control.....	8
4.4.5 Control Commands .....	8
4.5 Constants / Error Codes .....	10
4.6 Fixed-width Integer.....	12
4.7 Structures / Unions / Enumerated Types .....	13
4.7.1 Structures .....	13
4.7.2 Unions .....	13
4.7.3 Enumerated Types .....	13
4.8 Sample Program.....	14
4.8.1 Operation Overview.....	14
4.8.2 Functions Used in the Sample Program .....	16
4.8.3 Specifications of Sample Program Functions .....	16
4.8.4 Variables Used in the Sample Program .....	18
4.8.5 Constants Used in the Sample Program.....	18
4.8.6 Flowchart of Main Process .....	19
4.8.7 Operation Sequence .....	21
4.8.8 Console Commands.....	23
4.8.9 Execution Result of the Sample Program .....	23
5. Sample Code.....	24
Revision History.....	25

## 1. Specifications

Table 1.1 lists the peripheral functions to be used and their applications. Figure 1.1 shows the operating environment when the sample code is being executed.

**Table 1.1 Peripheral Functions and Applications**

Peripheral Module	Application
Encoder Divided-Output (ENCOUT) module	Generates Phase A, Phase B, and Phase Z outputs corresponding to the positional values.
Interrupt controller (ICU)	Controls interrupts from the GPT unit 0 channel 0.
General PWM Timer (GPT) Unit 0 Channel 0	Generates events at fixed intervals for input to the ELC, and timer interrupts.
Event link controller (ELC)	Makes the link between events output from unit 0 channel 0 of the GPT and the ENCOUT module.
Serial communication interface (SCI) UART	Asynchronous communications of the SCI are used for COM port communications by using USB interface. It is used for console interface of the sample program.



**Figure 1.1 Operating Environment**

## 2. Operating Environment

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

**Table 2.1 Operating Environment**

Item		Description
MCU		RZ/N2H Group
Operating frequency *1	CR52 ver.	CPUCLK = 1000 MHz (Cortex®-R52 CPU0)
	CA55 ver.	CPUCLK = 1200 MHz (Cortex®-A55 Core0)
Operating voltage		0.8 V (Core) / 1.1 V (DDR) / 1.8 V (PLL, etc.) / 3.3 V (I/O)
Integrated development environment *2		IAR Systems: IAR Embedded Workbench® for Arm® RENESAS: e <sup>2</sup> studio
Board		RZ/N2H Evaluation Board (RTK9RZN2Hxxxxxxxx)
Devices (function to be used on the board)		None

- Note: 1. This sample program has a CR52 version that runs on the CPU core Cortex®-R52 and a CA55 version that runs on the CPU core Cortex®-A55. CR52 ver. and CA55 ver. are descriptions of the respective versions.
2. Refer to the RZ/N2H Group ENCOUT sample program Release Note to check the version number of the integrated development environment.

### 3. Peripheral Functions

The basics of the peripheral modules, operating modes, and registers are described in the “RZ/N2H Group User’s Manual: Hardware”

#### 3.1 Pins

The pins used and their functions are listed in the table below.

**Table 3.1 Pins Used and their Functions**

Channel	Pin Name	I/O Port	Input/Output	Voltage Domain	Description
Channel 0	POUTA	P02_5	Output	VDD33	Phase A output pin
	POUTB	P02_6	Output	VDD33	Phase B output pin
	POUTZ	P02_7	Output	VDD33	Phase Z output pin

## 4. Software

### 4.1 ENCOUT Driver Function

The functions of the ENCOUT driver are listed below.

- 1 Initial settings
- 2 Sets positional information
- 3 Outputs Phase A, Phase B, and Phase Z signals

### 4.2 File Structure

For the file structure, refer to the release note for the RZ/N2H Group ENCOUT program.

### 4.3 Functions

The functions to be used are listed in the table below.

**Table 4.1 List of Functions**

Category	Function Name	Page
ENCOUT driver API functions	R_ENCOUT_Open	7
	R_ENCOUT_Close	7
	R_ENCOUT_GetVersion	7
	R_ENCOUT_Control	8

## 4.4 Specifications of API Functions

### 4.4.1 R\_ENCOUT\_Open

---

<b>R_ENCOUT_Open</b>	
Synopsis	Initializes ENCOUT driver
Header	r_encout_rzt2_if.h
Declaration	r_encout_err_t R_ENCOUT_Open(const int32_t id);
Description	This function initializes ENCOUT driver. Call this function before using ENCOUT driver.
Argument	id : Designate ENCOUT_ID_0.
Return value	R_ENCOUT_SUCCESS : Normal termination R_ENCOUT_ERR_INVALID_ARG : Abnormal termination (The id is not specified.) R_ENCOUT_ERR_ACCESS : Abnormal termination (ENCOUT driver is already opened.)

### 4.4.2 R\_ENCOUT\_Close

---

<b>R_ENCOUT_Close</b>	
Synopsis	Terminates ENCOUT driver
Header	r_encout_rzt2_if.h
Declaration	r_encout_err_t R_ENCOUT_Close(const int32_t id);
Description	This function terminates ENCOUT driver.
Argument	id : Designate ENCOUT_ID_0.
Return Value	R_ENCOUT_SUCCESS : Normal termination R_ENCOUT_ERR_INVALID_ARG : Abnormal termination (The id is not specified.)

### 4.4.3 R\_ENCOUT\_GetVersion

---

<b>R_ENCOUT_GetVersion</b>	
Synopsis	Acquires 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 the ENCOUT driver.
Argument	None
Return value	A major part of the version number is stored in the sixteen MSBs and the minor part of the version number is stored in the sixteen LSBs. Ex.) The value returned is 0x00010002 for Ver.1.2.

#### 4.4.4 R\_ENCOUT\_Control

---

<b>R_ENCOUT_Control</b>	
Synopsis	Controls 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 p_buf);
Description	This function controls the operation of the ENCOUT. The operation of this function depends on the argument cmd. See “4.4.5 Control Commands” for the operation of each command.
Argument	id : Designate ENCOUT_ID_0. cmd : Control command Designate one of R_ENCOUT_CMD_INIT, R_ENCOUT_CMD_START, R_ENCOUT_CMD_STOP, or R_ENCOUT_CMD_SET. p_buf : Argument depending on cmd. See “4.4.5 Control Commands” for details.
Return value	It depends on cmd. See “4.4.5 Control Commands” for details.

#### 4.4.5 Control Commands

##### (1) R\_ENCOUT\_CMD\_INIT

---

<b>R_ENCOUT_CMD_INIT</b>	
Synopsis	Initializes ENCOUT
Header	r_encout_rzt2_if.h
Declaration	e_encout_err_t R_ENCOUT_Control(const int32_t id, const r_encout_cmd_t cmd, void *const p_buf);
Description	This function initializes ENCOUT. Control Register (CTL), Maximum Position and Carrier Period Register (POSMAX_PERIOD), Output Count Register (OUTCNT), and Position Count Register (POSCNT) are initialized using the initializing information of the argument.
Argument	id : Designate ENCOUT_ID_0. cmd : Designate R_ENCOUT_CMD_INIT. p_buf : Initialization information Designate the pointer to the initialization information structure r_encout_init_t. See “4.7.1(1) r_encout_init_t” for details.
Return value	R_ENCOUT_SUCCESS : Normal termination R_ENCOUT_ERR_INVALID_ARG : Abnormal termination (The id or the cmd is not specified, the p_buf is NULL, or the p_buf member is not specified or invalid value.) R_ENCOUT_ERR_ACCESS : Abnormal termination (The ENCOUT driver is not initialized.) R_ENCOUT_ERR_BUSY : Abnormal termination (The ENCOUT is already running.)

**(2) R\_ENCOUT\_CMD\_START****R\_ENCOUT\_CMD\_START**


---

Synopsis	Starts 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 p_buf);
Description	This function starts ENCOUT operation.
Arguments	id : Designate ENCOUT_ID_0. cmd : Designate R_ENCOUT_CMD_START. p_buf : Not used. (Designate NULL.)
Return value	R_ENCOUT_SUCCESS : Normal termination R_ENCOUT_ERR_INVALID_ARG : Abnormal termination (The id or the cmd is not specified.) R_ENCOUT_ERR_ACCESS : Abnormal termination (The ENCOUT driver is not initialized.) R_ENCOUT_ERR_BUSY : Abnormal termination (The ENCOUT is already running.)

**(3) R\_ENCOUT\_CMD\_STOP****R\_ENCOUT\_CMD\_STOP**


---

Synopsis	Stops 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 p_buf);
Description	This function stops ENCOUT operation.
Arguments	id : Designate ENCOUT_ID_0. cmd : Designate R_ENCOUT_CMD_STOP. p_buf : Not used. (Designate NULL.)
Return value	R_ENCOUT_SUCCESS : Normal termination. R_ENCOUT_ERR_INVALID_ARG : Abnormal termination (The id or the cmd is not specified.) R_ENCOUT_ERR_ACCESS : Abnormal termination (The ENCOUT driver is not initialized.)

**(4) R\_ENCOUT\_CMD\_SET**

<b>R_ENCOUT_CMD_SET</b>	
Synopsis	Sets ENCOUT positional information
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 p_buf);
Description	This function updates positional information while ENCOUT is running.
Argument	id : Designate ENCOUT_ID_0. cmd : Designate R_ENCOUT_CMD_SET p_buf : Positional information Designate the pointer to the positional information structure r_encout_set_t. See "4.7.1(2) r_encout_set_t" for details.
Return value	R_ENCOUT_SUCCESS : Normal termination R_ENCOUT_ERR_INVALID_ARG : Abnormal termination (The id or the cmd is not specified, the p_buf is NULL, or the p_buf member is invalid value.) R_ENCOUT_ERR_ACCESS : Abnormal termination (The ENCOUT is not running.)

**4.5 Constants / Error Codes**

See the individual tables below for settings and descriptions related to the constants and error codes.

**Table 4.2 List of Tables for Definitions of Constants and Error Codes**

Table number	Contents
Table 4.3	User-Defined Constants to be Used in the ENCOUT Driver (r_encout_rzt2_config.h)
Table 4.4	Control Commands for the R_ENCOUT_Control Function (r_encout_rzt2_if.h)
Table 4.5	ENCOUT Driver ID (r_encout_rzt2_if.h)
Table 4.6	Output Polarity of Phase B (r_encout_rzt2_if.h)
Table 4.7	Output Pattern of Phase Z (r_encout_rzt2_if.h)
Table 4.8	Error Codes (r_encout_rzt2_if.h)

**Table 4.3 User-Defined Constants to be Used in the ENCOUT Driver (r\_encout\_rzt2\_config.h)**

Constant Name	Setting	Description
ENCODER_RESOLUTION0	1048576	Encoder resolution which is used by the ENCOUT driver to calculate position count. 32-bit value is available except for 0. For example, 1048576 is defined in case that the encoder resolution is 20 bits (positional value is 0 to 1048575). The default value is 1048576.

**Table 4.4 Control Commands for the R\_ENCOUT\_Control Function (r\_encout\_rzt2\_if.h)**

Constant Name	Setting	Description
R_ENCOUT_CMD_INIT	0	Initializes ENCOUT
R_ENCOUT_CMD_START	1	Starts ENCOUT
R_ENCOUT_CMD_STOP	2	Stops ENCOUT
R_ENCOUT_CMD_SET	3	Sets ENCOUT positional information

Table 4.5 ENCOUT Driver ID (r\_encout\_rzt2\_if.h)

Constant Name	Setting	Description
ENCOUT_ID_NUM	1	Number of ENCOUT driver channels
ENCOUT_ID_0	0x01	ENCOUT driver channel 0 ID Nothing but channel 0 is available. Always use ENCOUT_ID_0 as the driver argument id.

Table 4.6 Output Polarity of Phase B (r\_encout\_rzt2\_if.h)

Constant Name	Setting	Description
ENCOUT_CTL_PHASE_NORMAL	0x00	Phase B is positive phase
ENCOUT_CTL_PHASE_REVERSE	0x01	Phase B is negative phase

Table 4.7 Output Pattern of Phase Z (r\_encout\_rzt2\_if.h)

Constant Name	Setting	Description
ENCOUT_CTL_Z_NONE	0x00	Disable Phase Z output
ENCOUT_CTL_Z_1P	0x02	Pulse output when current position, POSCNT, is 0.
ENCOUT_CTL_Z_2P_A	0x04	Pulse output when current position, POSCNT, is 0 or maximum value. (Synchronizes with Phase A.)
ENCOUT_CTL_Z_2P_B	0x14	Pulse output when current position, POSCNT, is 1 or below. (Synchronizes with Phase B.)
ENCOUT_CTL_Z_3P	0x06	Pulse output when current position, POSCNT, is 1 or below, or equals to maximum value.
ENCOUT_CTL_Z_4P_A	0x08	Pulse output when current position, POSCNT, is 2 or below, or equals to maximum value. (Synchronizes with Phase A.)
ENCOUT_CTL_Z_4P_B	0x18	Pulse output when current position, POSCNT, is 1 or below, or (maximum value – 1) or above. (Synchronizes with Phase B.)

Table 4.8 Error Codes (r\_encout\_rzt2\_if.h)

Constant Name	Setting	Description
R_ENCOUT_SUCCESS	0	Normal termination
R_ENCOUT_ERR_INVALID_ARG	-1	Invalid argument
R_ENCOUT_ERR_BUSY	-2	API cannot be executed
R_ENCOUT_ERR_ACCESS	-3	API execution sequence error

## 4.6 Fixed-width Integer

Table below lists the fixed-width integers used in the sample code. These fixed-width integers are defined in the standard libraries.

**Table 4.9 Fixed-width Integers for the Sample Code**

Symbol	Description
int8_t	8-bit signed integer
int16_t	16-bit signed integer
int32_t	32-bit signed integer
int64_t	64-bit signed integer
uint8_t	8-bit unsigned integer
uint16_t	16-bit unsigned integer
uint32_t	32-bit unsigned integer
uint64_t	64-bit unsigned integer

## 4.7 Structures / Unions / Enumerated Types

The major structures, unions, and enumerated types are listed below.

### 4.7.1 Structures

#### (1) r\_encout\_init\_t

Initialization information of the ENCOUT control unit,

```
typedef struct
{
    uint8_t    control_param;  Designates POL bit, ZW bit, and ZS bit value of the Control register
                               (CTL). Set value using macros shown in "Table 4.6 Output Polarity
                               of Phase B (r_encout_rzt2_if.h)" and "Table 4.7 Output Pattern of
                               Phase Z (r_encout_rzt2_if.h)".

    uint16_t   position_max;   Designates maximum position count for one revolution (number of
                               edges – 1).
                               This setting is reflected in the POSMAX bit of the
                               POSMAX_PERIOD register.

    uint32_t   carrier_period; Designates carrier interval in nanoseconds. Set value multiple of 50
                               within the range of 50000 to 3276750.
                               This setting is reflected in the PERIOD bit of the POSMAX_PERIOD
                               register.

    uint32_t   encoder_count;  Designates initial position of the encoder within the range of 0 to
                               (ENCODER_RESOLUTION0 – 1). (ENCODER_RESOLUTION0 is
                               the resolution of the encoder.)
                               This setting is reflected in the POSCNT register.
} r_encout_init_t
```

#### (2) r\_encout\_set\_t

Positional information to be set to the ENCOUT.

```
typedef struct
{
    uint32_t   encoder_count;  Designates position of the encoder within the range of 0 to
                               (ENCODER_RESOLUTION0 – 1). (ENCODER_RESOLUTION0 is
                               the resolution of the encoder.)
} r_encout_set_t
```

### 4.7.2 Unions

Unions are not used in this sample program.

### 4.7.3 Enumerated Types

Enumerate types are not used in this sample program.

## 4.8 Sample Program

### 4.8.1 Operation Overview

This sample program handles the following processes.

- 1) Designates carrier interval, maximum position count for one revolution, and output pattern of Phase A, Phase B, Phase Z signals.
- 2) Set up GPT to generate carrier interval, and input to the ENCOUT as the carrier interval event signal via ELC.
- 3) Outputs Phase A, Phase B, and Phase Z signals with updating edge count of the ENCOUT by the carrier interval interrupt from the GPT.

#### (1) System Block Diagram

A system block diagram is shown below.

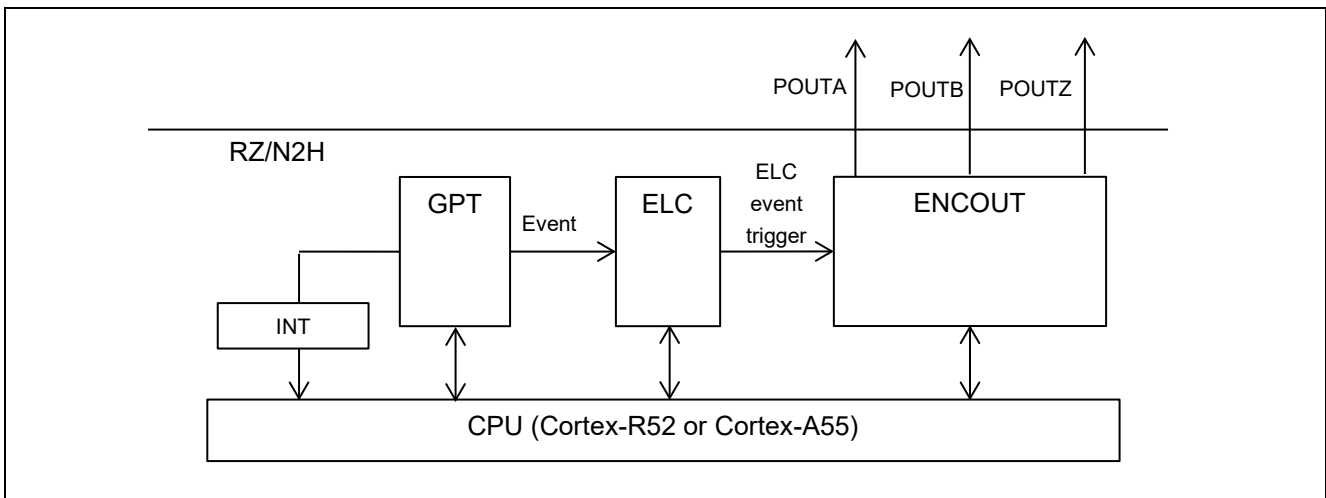


Figure 4.1 System Block Diagram

## (2) Software Structure

Figure 4.2 shows a software structure.

The ENCOUT driver has the opening process part configured by the function R\_ENCOUT\_Open, the closing process part configured by the function R\_ENCOUT\_Close, the control part configured by the function R\_ENCOUT\_Control.

The sample program has the ENCOUT driver initialization part which controls ENCOUT driver, launches and starts it, and the update positional information part which updates positional information by the timer interrupt.

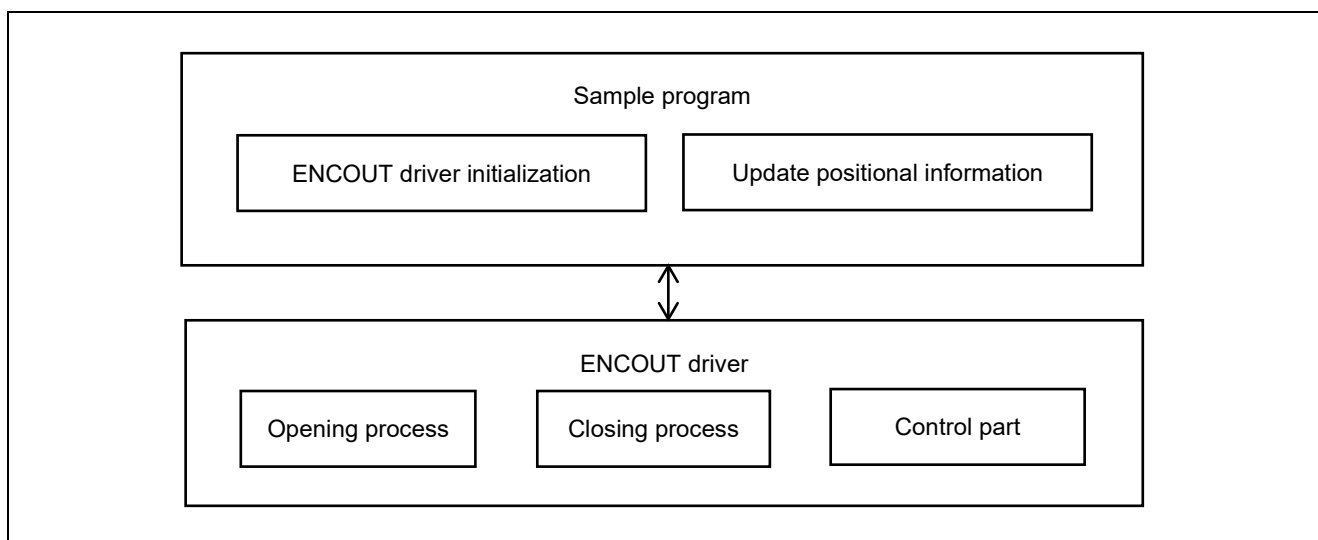


Figure 4.2 Software Structure

## 4.8.2 Functions Used in the Sample Program

The major functions used in the sample program are listed below.

**Table 4.10 Major Functions Used in the Sample Program**

Function Name	Page
hal_entry	16
enc_main	16
show_encout_drv_version	16
timer_init	17
timer_start	17
int_gpt0_ovf	17
r_g_timer0_callback	17

## 4.8.3 Specifications of Sample Program Functions

### (1) hal\_entry

---

<b>hal_entry</b>	
Synopsis	Entry function of the ENCOUT sample program
Header	-
Declaration	void hal_entry(void);
Description	This is the entry function of the ENCOUT sample program. The function enc_main() is called from here.
Argument	None
Return value	None

### (2) enc\_main

---

<b>enc_main</b>	
Synopsis	Main function of the ENCOUT sample program.
Header	-
Declaration	int32_t enc_main(uint8_t ch);
Description	This is the main function of the ENCOUT sample program. See “4.8.6(1) Flowchart of enc_main” for details.
Argument	ch                   ENCOUT channel number to use 0: use channel 0 (Only channel 0 is available.)
Return value	0: Normal termination Others: Abnormal termination (an error code of the ENCOUT driver)

### (3) show\_encout\_drv\_version

---

<b>show_encout_drv_version</b>	
Synopsis	ENCOUT driver version output function
Header	-
Declaration	static void show_encout_drv_version(void);
Description	This function outputs ENCOUT driver version to console.
Argument	None
Return value	None

**(4) timer\_init****timer\_init**


---

Synopsis	Timer initialization function
Header	-
Declaration	static void timer_init(void);
Description	This function sets interval and interrupts- of the timer GPT unit 0 channel 0.
Argument	None
Return value	None

---

**(5) timer\_start****timer\_start**


---

Synopsis	Timer start function
Header	-
Declaration	static void timer_start(void);
Description	This function starts timer GPT unit 0 channel 0.
Argument	None
Return value	None

---

**(6) int\_gpt0\_ovf****int\_gpt0\_ovf**


---

Synopsis	Timer GPT0_OVF interrupt function
Header	-
Declaration	void int_gpt0_ovf(void);
Description	This function is called at the interval of CARRIER_PERIOD defined in "Table 4.12 Major Constants Used in the Sample Program". This function updates positional information of the ENCOUT by the values of encoder_data[] array shown in "Table 4.11 Static Variables Used in the Sample Program".
Arguments	None
Return value	None

---

**(7) r\_g\_timer0\_callback****r\_g\_timer0\_callback**


---

Synopsis	Timer GPT0_INT4 interrupt callback function
Header	-
Declaration	void r_g_timer0_callback(timer_callback_args_t *p_args);
Description	This function is called by the timer INT4 interrupt. The argument is callback function parameter data including timer event. If the timer event is interrupt of the maximum count value, this function calls the timer GPT0_OVF interrupt function.
Arguments	p_args      Pointer to the callback function parameter data
Return value	None

---

#### 4.8.4 Variables Used in the Sample Program

The major static variables used in the sample program are listed below.

**Table 4.11 Static Variables Used in the Sample Program**

Type	Variable Name	Description
uint32_t	encoder_data_index[]	<p>Index of the encoder positional information array</p> <p>It is index of the encoder positional information array to extract element for sending to ENCOUT. It is incremented at the timer intervals.</p> <p>Index itself is defined as an array on the assumption that the plural channels exist. However, number of ENCOUT channel supported by RZ/N2H is one. The array has just one element. Initial value is {0}.</p>
uint32_t	encoder_data[]	<p>Encoder positional information array</p> <p>It stores positional information in an array to send its elements to ENCOUT at the timer intervals. Default value is { 943719, 996148, 20972, 1027605, 1038090 }.</p> <p>Each element of the array corresponds to position count { 90, 95, 2, 98, 98 } respectively, in case that the encoder resolution is 1048576 and maximum position count for one revolution is 99.</p> <p>Instead of using elements of this array, if positional information read from encoder at the timer intervals are used, Phase ABZ signals are generated corresponding to the encoder rotation.</p>

#### 4.8.5 Constants Used in the Sample Program

The major constants used in the sample program are listed below.

**Table 4.12 Major Constants Used in the Sample Program**

Constant	Setting	Description
CARRIER_PERIOD	100000	<p>Carrier period of the Phase ABZ signals generated by ENCOUT</p> <p>It designates carrier period in nanoseconds. Available range is 50000 to 3276750. Default value is 100000 [ns] (100 [us]). Use multiple of 50.</p>
POSITION_MAX	99	<p>Maximum position count for one revolution (Number of edges – 1)</p> <p>It designates maximum position count. This value is set to the POSMAX bit of the POSMAX_PERID register in the ENCOUT module. 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.</p>
ENCOUT_CTL	0x04	<p>Output pattern of Phase ABZ signals generated by ENCOUT</p> <p>It designates Phase B polarity (positive or negative phase), pulse width of Phase Z signal (0, 1, 2, 3, 4), and synchronization phase of Phase Z (Phase A or Phase B). Select pattern with macro shown in Table 4.6 and Table 4.7. Default setting is Phase B with positive phase, 2-pulse Phase Z signal synchronized with Phase A.</p>

#### 4.8.6 Flowchart of Main Process

The flowcharts for the major processes are given below.

##### (1) Flowchart of enc\_main

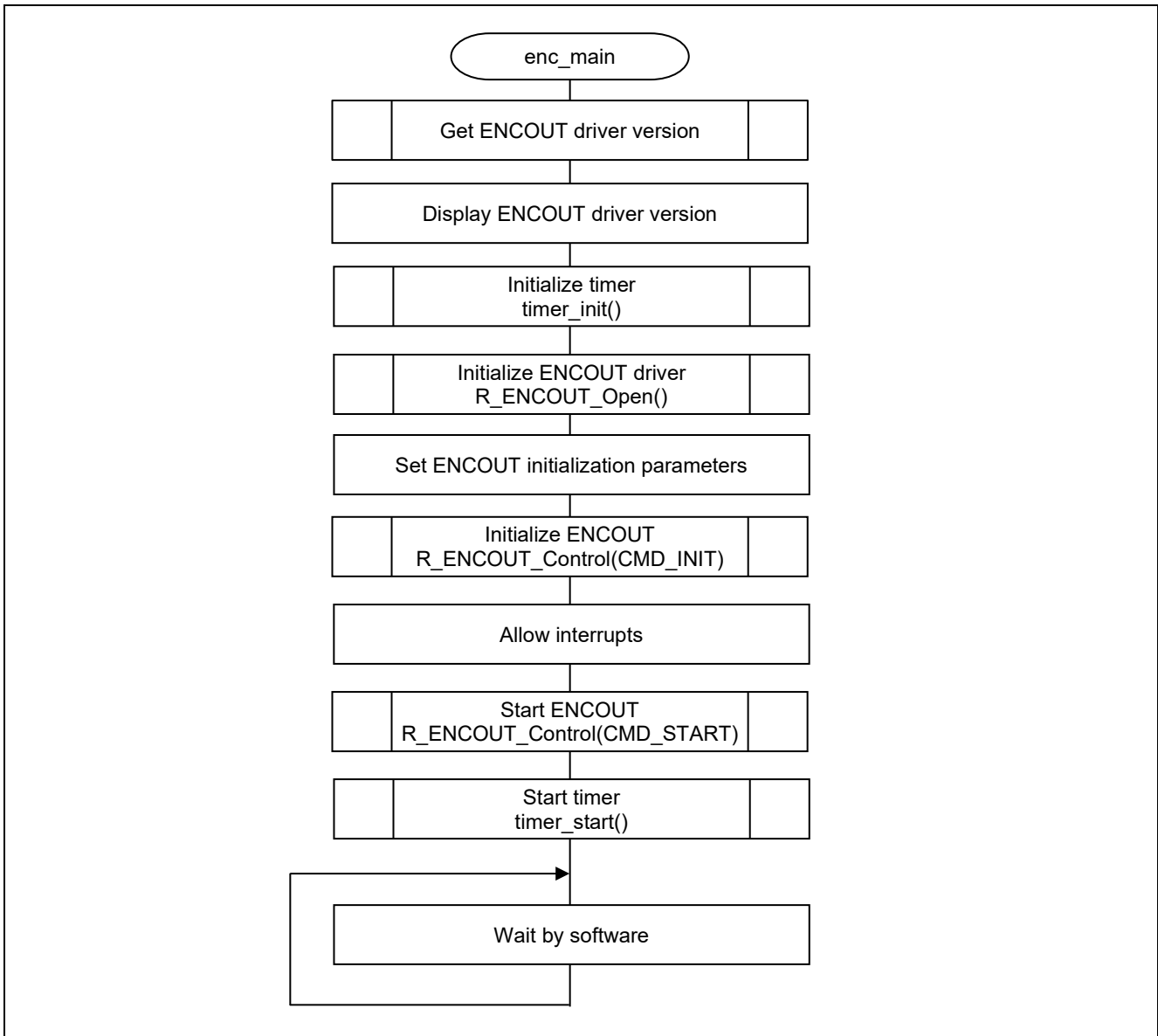


Figure 4.3 Flowchart of enc\_main function

(2) Flowchart of int\_gpt0\_ovf

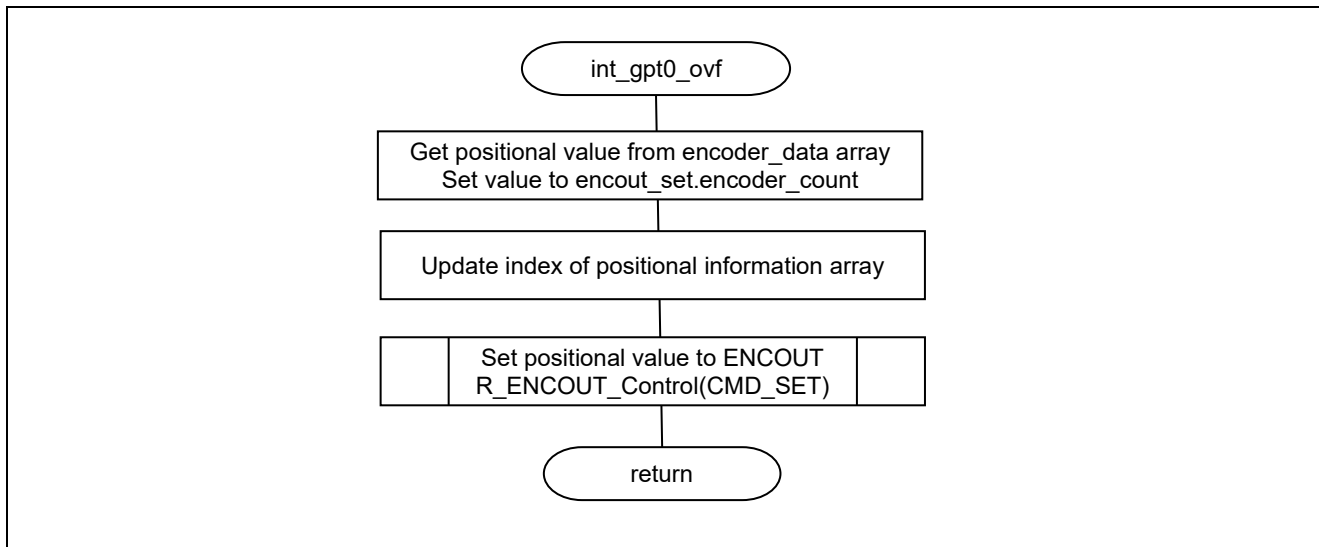


Figure 4.4 Flowchart of int\_gpt0\_ovf function

### 4.8.7 Operation Sequence

#### (1) Startup Sequence

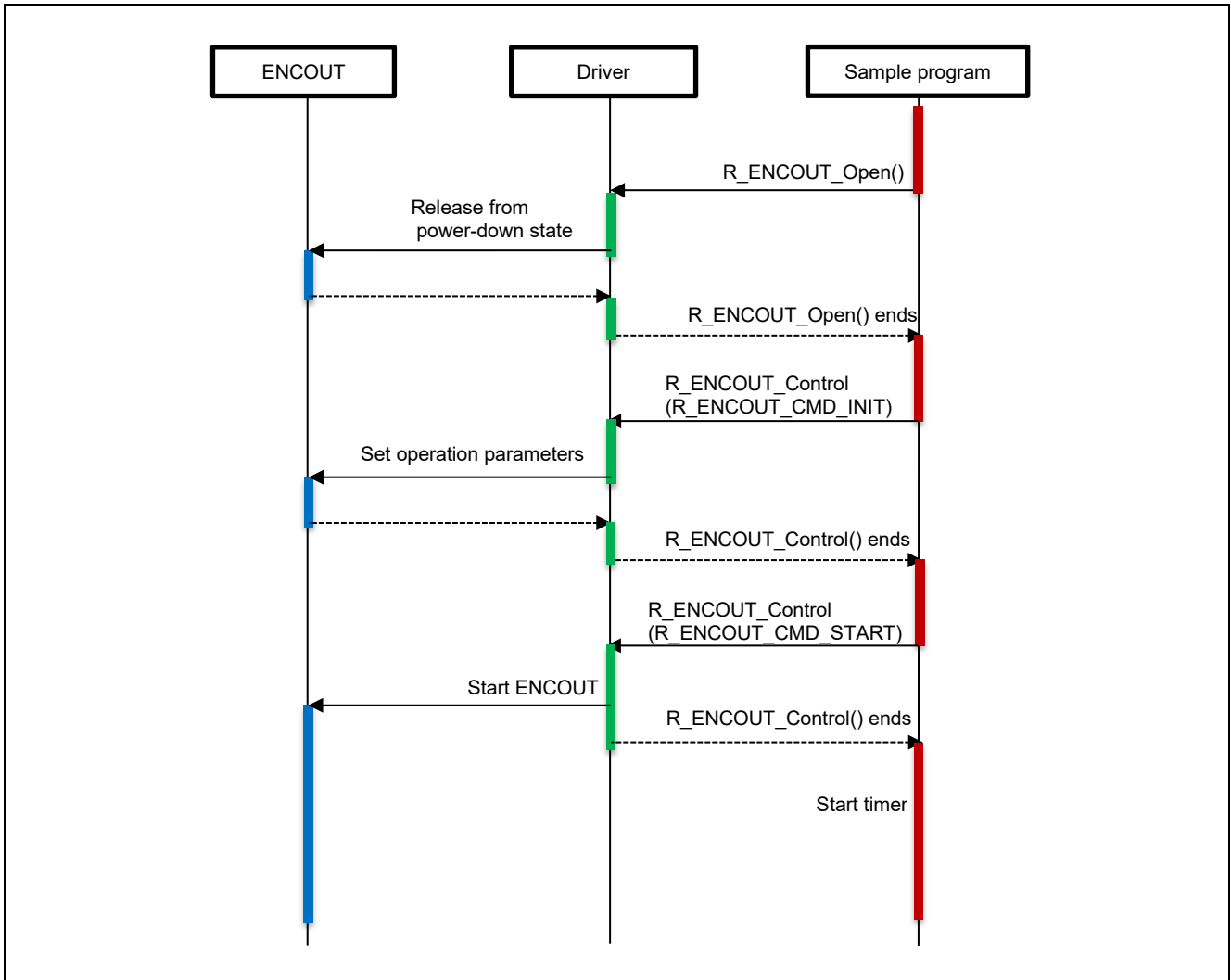


Figure 4.5 Startup Sequence Diagram

(2) Positional Information Update Sequence

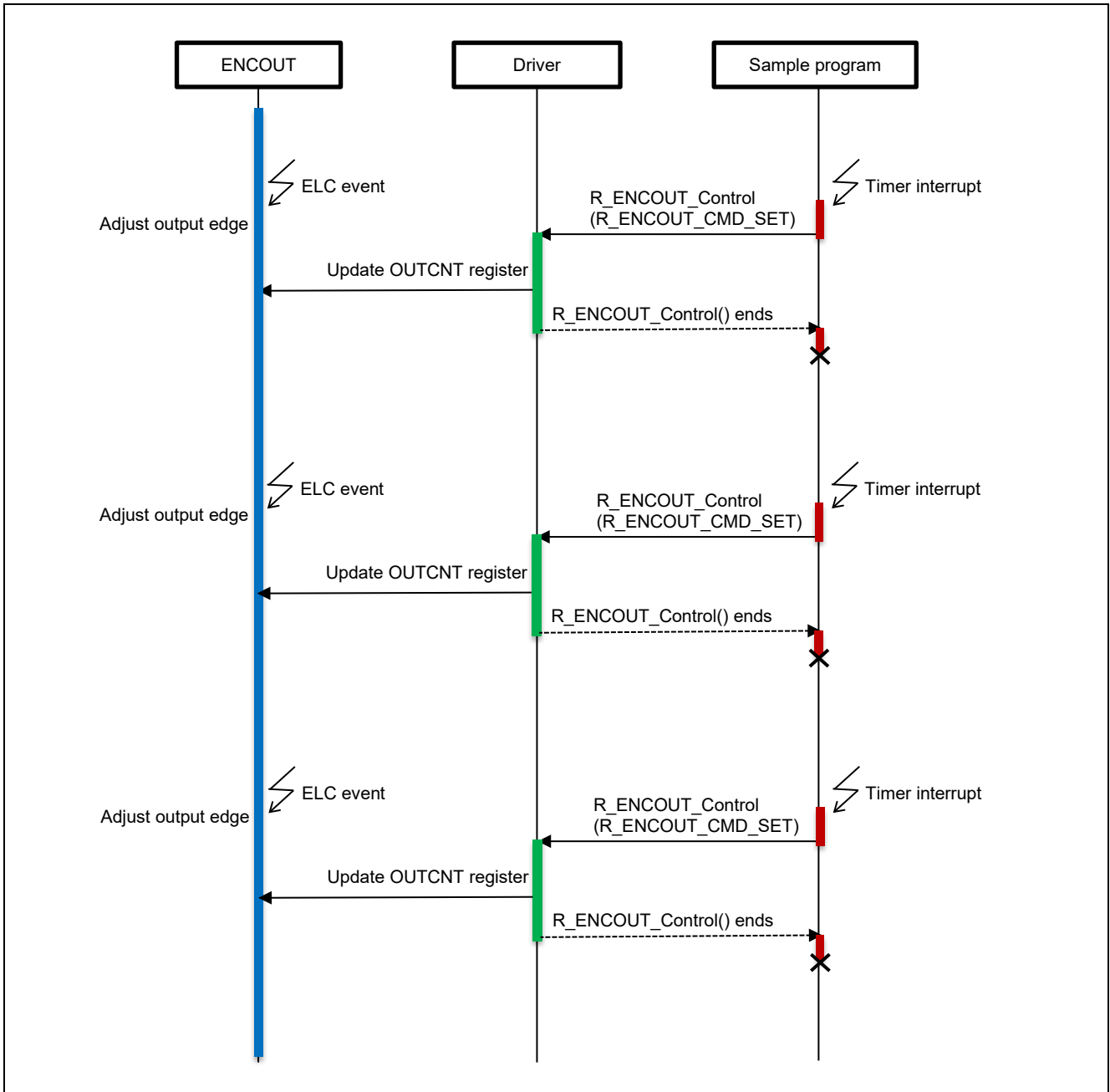


Figure 4.6 Positional Information Update Sequence Diagram

### 4.8.8 Console Commands

This sample program has no commands to input from console.

### 4.8.9 Execution Result of the Sample Program

The following signals are generated to the output ports shown in “3.1 Pins” by the sample program.

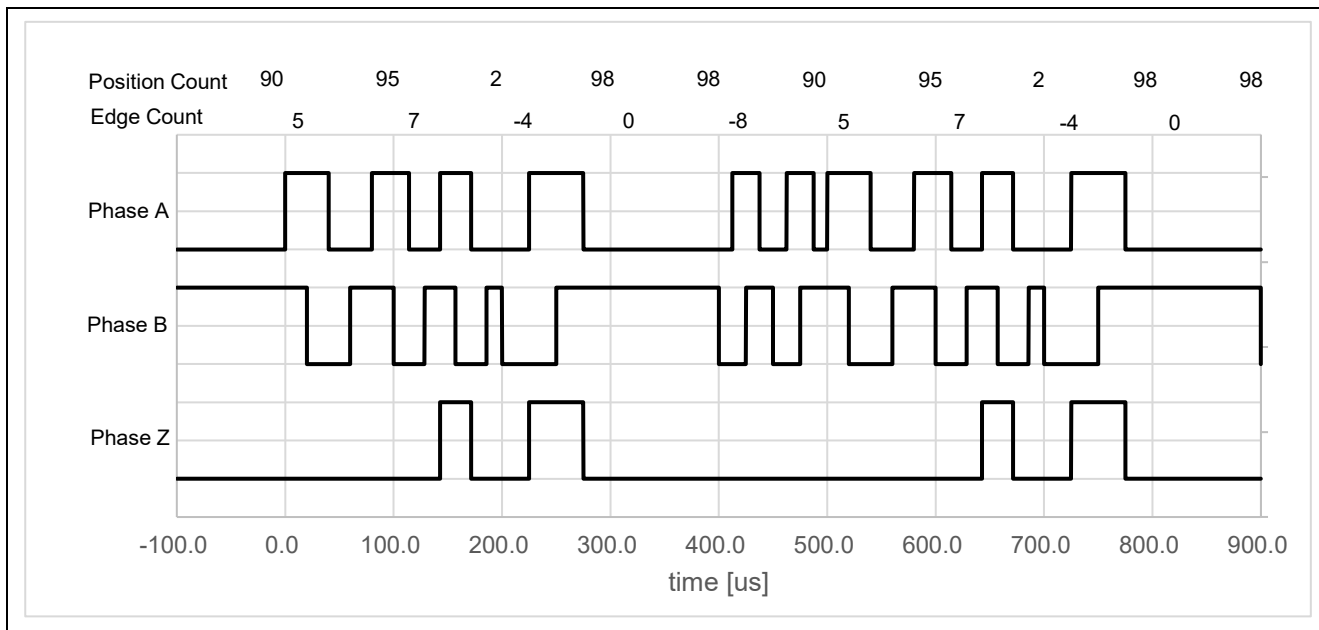


Figure 4.7 Phase ABZ Output Signals

## 5. Sample Code

The sample program is available on the Renesas Electronics website.

**Revision History**

Rev.	Date	Description	
		Page	Summary
2.00	Dec 23.24	-	First Edition issued.
3.00	Nov 21.25	1, 3, 4	Change description for trademarks.
4.00	May 15.26	4 8 to 10	Change frequency of the Cortex-A55 Core0 to 1200MHz. Change prefix of pointer variables to "p_".

# 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.

- 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.

## 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](http://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/](http://www.renesas.com/contact/).