Vector control with encoder

RAJ306000 implementation guide

Summary

This application note explains a sample program to support the drive by the vector control with encoder of permanent magnetic synchronous motor by RAJ306000, and the method using the library of development support tool "In Circuit Scope".

These sample programs are only to be used as reference and Renesas Electronics Corporation does not guarantee the operations. Please use them after carrying out a thorough evaluation in a suitable environment.

Operation checking device

Operations of the sample program are checked by using the following device.

・RAJ306000

Target of sample program

A sample program that this application note is intended shown below.

・RAJ306000_ENCD_FOC_OPEN_V101

Vector control sample program with encoder for RAJ306000
(Complementary PWM Mode)

Reference materials

・RL78/G1F User's Manual: Hardware (R01UH0516EJ0110)

・RAJ306000 Series User's Manual: Hardware (R18UZ0066EJ0100)

・Vector Control for Permanent Magnet Synchronous Motor with Encoder: Algorithm (R01AN3789EJ0101)

・In Circuit Scope Manual

Downloadable from: http://www.desktoplab.co.jp/download.html

Contents

1. Overview

This application note explains a sample program to support the drive by the vector control with encoder of permanent magnetic synchronous motor by RAJ306000, and the method using the library of development support tool "In Circuit Scope".(Note 1).

1.1 Development environment

Development environment of the sample programs are showed in Table 1-1 and Table 1-2.

Table 1-1 Software development environment

Integrated Development Environment	CS+ for CA, CX V3.02.00 [15 Mar 2016]
Compiler	CA78K0R V1.72

Table 1-2 Hardware development environment

On-chip Debugging Emulator	RENESAS E1 Emulator (R0E000010KCE00)
Operation Checking Device	RAJ306000 (Note 2)
RAJ306000 Series Evaluation Board	RTK0EML2A0D00010BJ

Note:

1. The development support tool In Circuit Scope (ICS) is a product of Desk Top Laboratories Inc.

   Desk Top Laboratories Inc. (http://www.desktoplab.co.jp/)
2. The configuration of RAJ306000 which is a SIP product containing MCU(RL78/G1F) and PreDriver is shown in Figure 1-1.

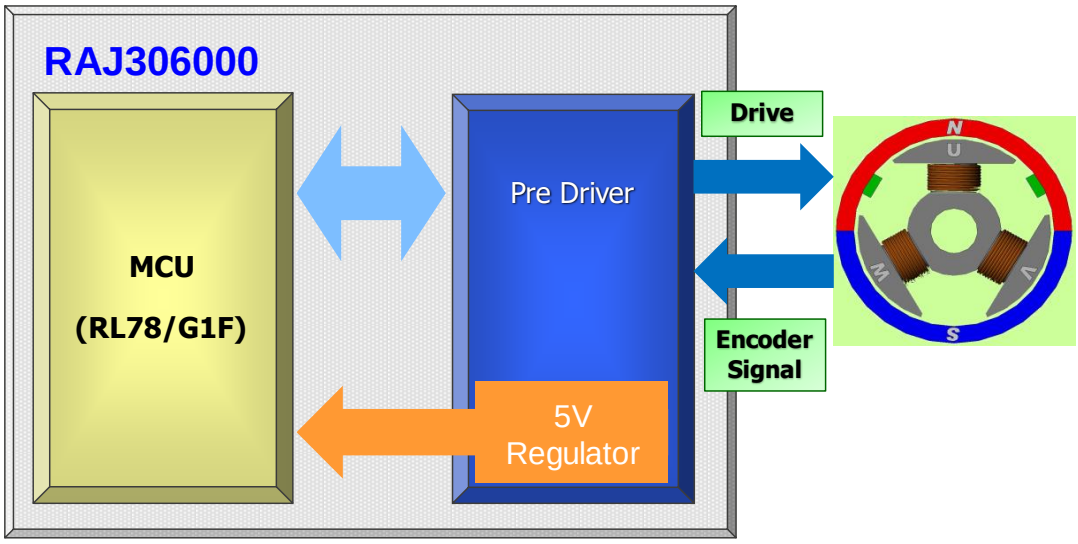

Figure 1-1 RAJ306000

2. System overview

Overview of RAJ306000 system is shown in Figure 2-1.

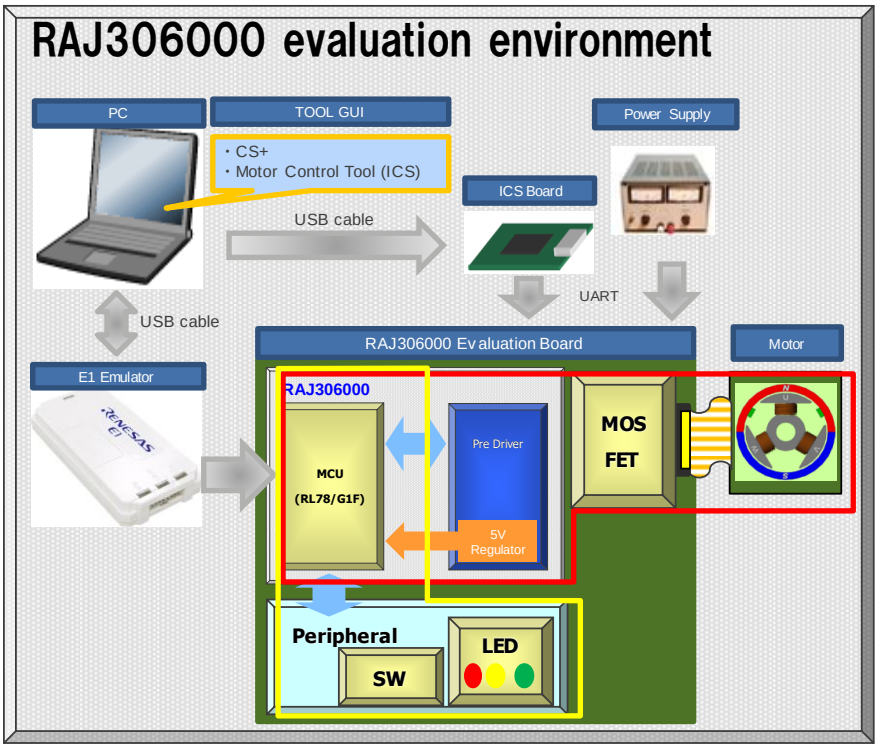

Figure 2-1 System configuration

2.1 Hardware configuration

As a hardware configuration, hardware connection of between RL78/G1F and PreDriver are shown in Figure 2-2(Figure 2-1 red line area), hardware connection of between RL78/G1F and Peripheral are shown in Figure 2-3(Figure 2-1 yellow line area).

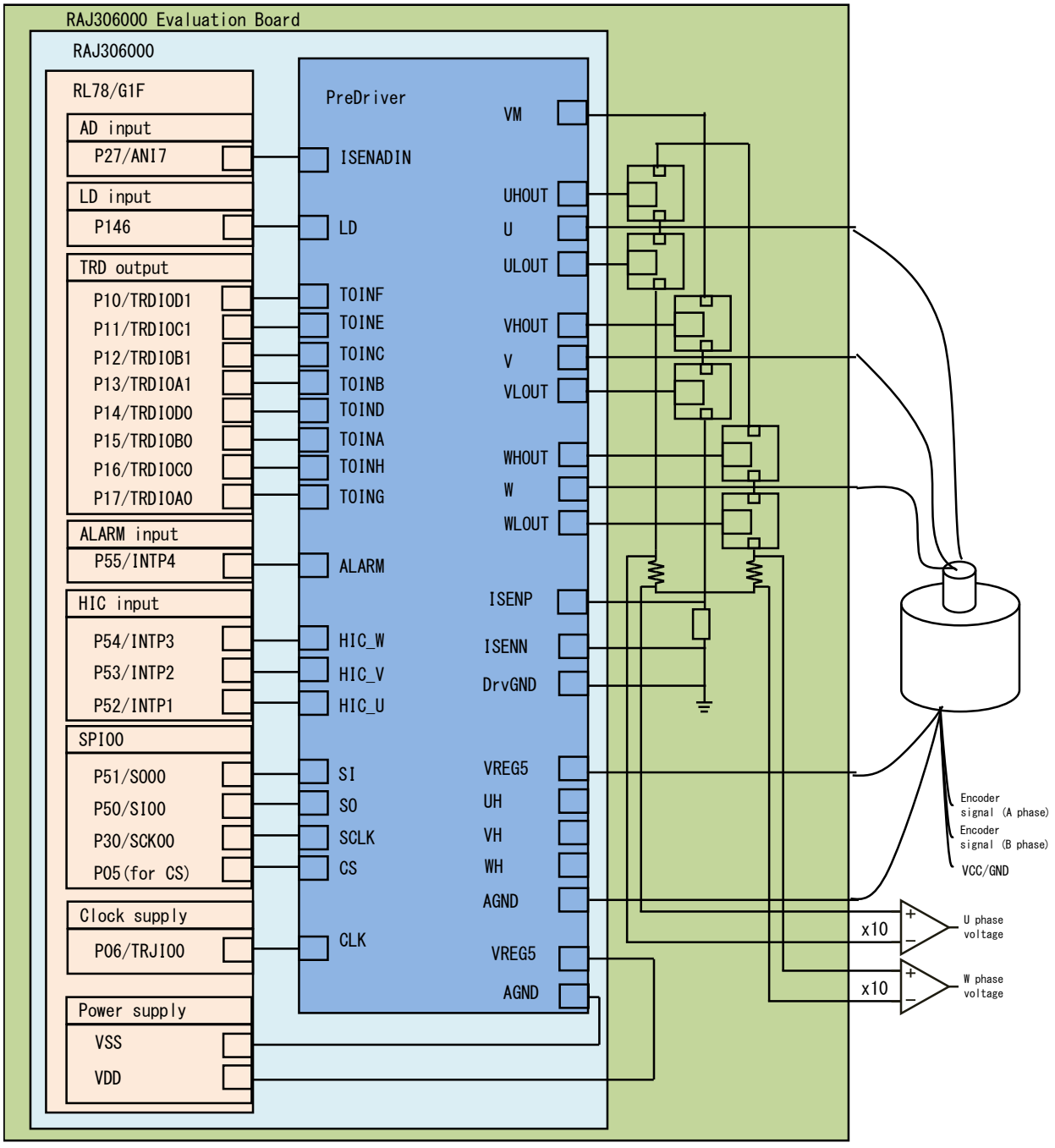

Figure 2-2 Hardware Configuration Diagram (RL78/G1F, PreDriver)

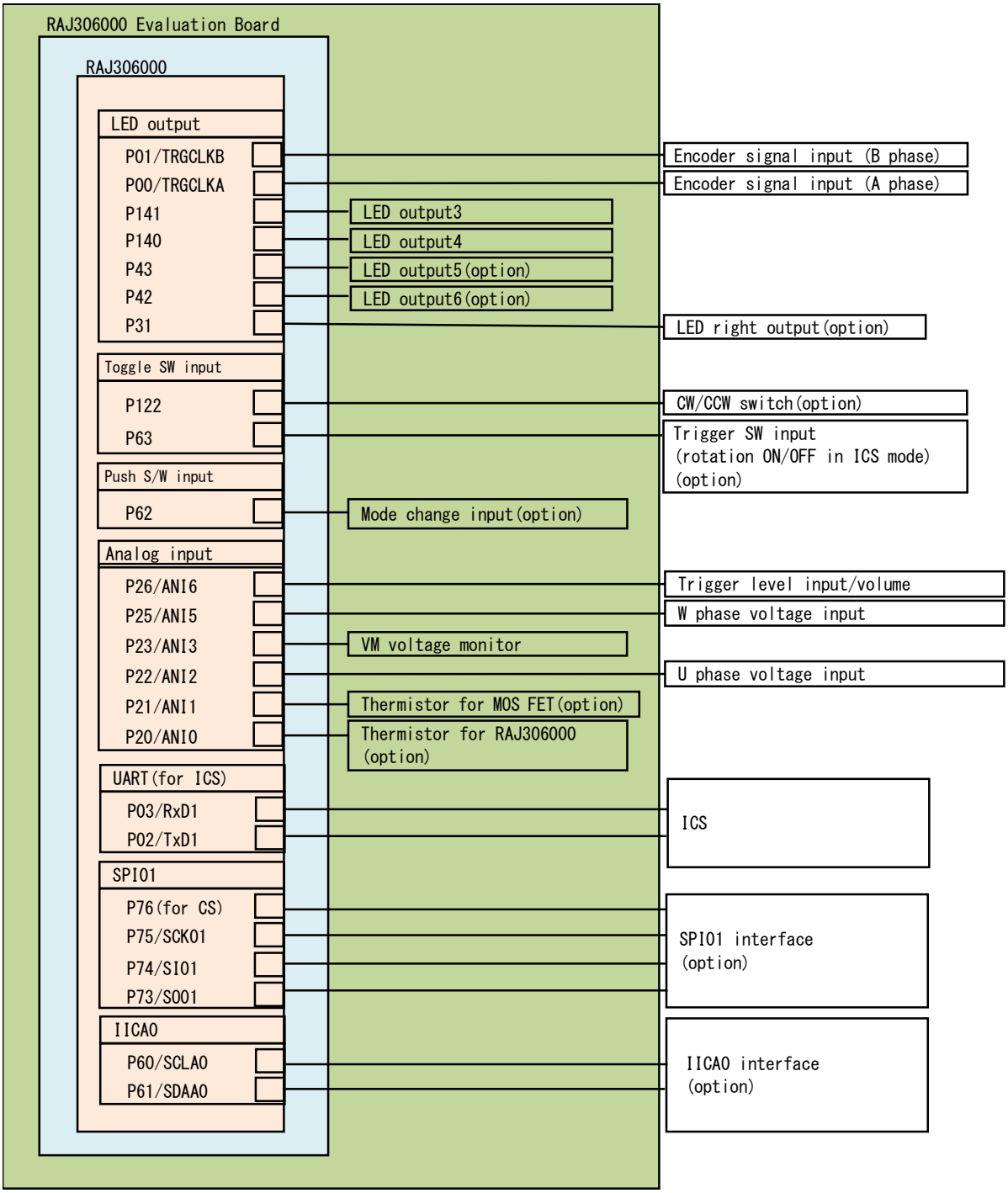

Figure 2-3 Hardware Configuration Diagram (RL78/G1F, Peripheral)

2.2 Hardware specifications

2.2.1 User interface

List of user interfaces of this system is shown in Table 2-1.

Table 2-1 User Interface

Item	Interface component	Function
Position	Input of trigger level/volume (VR1) or ICS	Position command value input (analog value)
Rotation direction	Selector switch of CW/CCW (SW1)	Input of rotation direction (CW Only)
Rotation speed	Input of trigger level/volume (VR1) or ICS	Rotation speed command value input (analog value)
START/STOP	Input of trigger level/volume (VR1) or ICS	Motor servo start/stop command
RED LED	LED output3	・Rotation speed under 250[rpm]: OFF ・Rotation speed over 250[rpm]: ON
	LED output4	・Rotation speed under 500[rpm]: OFF ・Rotation speed over 500[rpm]: ON
	LED output5	・At the time of normal operation: OFF ・At the time of error detection: ON
	LED output6	・At the time of normal operation: OFF ・At the time of error detection: ON

List of interfaces of RL78/G1F micro controller of this system is shown in Table 2-2.

Table 2-2 Port Interface (RL78/G1F)

Terminal name	Function
P27/ANI7	PreDriver voltage measurement (input)
P10/TRDIOD1	PWM output (W_n)
P11/TRDIOC1	PWM output (V_n)
P12/TRDIOB1	PWM output (W_p)
P13/TRDIOA1	PWM output (V_p)
P14/TRDIOD0	PWM output (U_n)
P15/TRDIOB0	PWM output (U_p)
P55/INTP4	ALARM signal input
P51/SO00	SPI data output for PreDriver control
P50/SI00	SPI data input for PreDriver control
P30/SCK00	SPI clock output for PreDriver control
P05(CS)	SPI chip selection for PreDriver control
P06/TRJIO0	System clock output for PreDriver
VSS	Ground voltage
VDD	Positive power supply
P146, P16/TRDIOC0, P17/TRDIOA0 P54/INTP3, P53/INTP2, P52/INTP1	Unused terminal
P01/TRGCLKB	Encoder signal input (B phase)
P00/TRGCLKA	Encoder signal input (A phase)
P141	LED output3 ON/OFF control
P140	LED output4 ON/OFF control
P43	LED output5 ON/OFF control
P42	LED output6 ON/OFF control
P122	For rotation direction command value input (CW Only)
P26/ANI6	Position command value input (analog value)
	For rotation speed command value input (analog value)
	Motor servo start/stop command
P25/ANI5	W phase shunt resistance voltage measurement (input)
P23/ANI3	VM voltage measurement (input)
P22/ANI2	U phase shunt resistance voltage measurement (input)
P03/RxD1	UART input for ICS
P02/TxD1	UART output for ICS
P31, P63, P62, P21/ANI1, P20/ANI0 P76(CS), P75/SCK01, P74/SI01, P73/SO01 P60/SCLA0, P61/SDLA0	Unused terminal

List of interfaces of PreDriver of this system is shown in Table 2-3.

Table 2-3 Port Interface (PreDriver)

Terminal name	Function
ISENADIN	PreDriver voltage output
TOINF	Motor control signal input (W_n)
TOINE	Motor control signal input (V_n)
TOINC	Motor control signal input (W_p)
TOINB	Motor control signal input (V_p)
TOIND	Motor control signal input (U_n)
TOINA	Motor control signal input (U_p)
ALARM	ALARM signal output
SI	Data input for SPI control
SO	Data output for SPI control
SCLK	Clock input for SPI control
CS	Chip select input for SPI control
CLK	System clock input
LD, TOINH, TOING HIC_W, HIC_V, HIC_U	Unused terminal
VM	Power Supply
UHOUT	U phase High-Side Driver (Nch) driving output
U	For U phase detection
ULOUT	U phase Low-Side Driver (Nch) driving output
VHOUT	V phase High-Side Driver (Nch) driving output
V	For V phase detection
VLOUT	V phase Low-Side Driver (Nch) driving output
WHOUT	W phase High-Side Driver (Nch) driving output
W	For W phase detection
WLOUT	W phase Low-Side Driver (Nch) driving output
ISENP	Shunt resistance Plus side connection
ISENN	Shunt resistance Minus side connection
DrvGND	PreDriver output stage circuit GND
VREG5	Regulator Output (5V)
AGND	PreDriver analog circuit GND
UH, VH, WH	Unused terminal

2.2.2 Peripheral functions

List of peripheral functions used in this system is shown in Table 2-4.

Table 2-4 Peripheral Functions List

Peripheral function	Usage
A/D converter	Position command value input (analog value)
	Rotation speed command value input (analog value)
	Voltage measurement (PreDriver voltage measurement/VM voltage measurement/U phase and W phase shunt resistance voltage measurement)
	Option: temperature measurement
General-purpose port	For rotation direction command value input (CW Only)
	LED output ON/OFF control
	Option: LED right output, toggle switch input/push switch input
Timer Array Unit	1[ms] interval timer
	25[us] interval timer
Timer RJ	System clock output for PreDriver
Timer RG	Encoder signal input (position detection)
Timer RD	Motor control signal output: PWM output using complimentary PWM mode (six outputs)
External interruption	ALARM signal detection
Communication interface	SPI00(for PreDriver control)
	UART1(for ICS)
	option: SPI01, IICA0

(1) A/D converter

The position command value input (analog value), rotation speed command value input (analog value) and voltage are measured by using "A/D converter"

A/D conversion is set channel selection mode to "Select mode" and conversion operation mode to "One shot conversion mode" (use software trigger).

Conversion speed of the A/D converter is 2.375[us] per channel and the smallest unit of conversion input value is shown in Table 2-5.

Table 2-5 A/D converter

Item	Control value for A/D converter 1 bit	Channel
Position command value input (analog value)	1.0[LSB] step (position range is 40[LSB] to 721[LSB])	ANI6
Rotation speed command value input (analog value)	1.066[rpm] step (rotation speed range is CW: 50[rpm] to 750[rpm])	
Voltage measurement	VM voltage measurement: 45.9[V] / 1024 = 0.045[V]	ANI3
	PreDriver voltage (Note 3) measurement: 48.0[V] / 1024 = 0.047[V]	ANI7
	U phase shunt resistance voltage measurement: 5.0[V] / 1024 = 0.005[V]	ANI5
	W phase shunt resistance voltage measurement: 5.0[V] / 1024 = 0.005[V]	ANI2

Note:

3. The PreDriver voltage switches a signal converting A/D by setting of register (ADC_SEL) for ADC selectors of the PreDriver side and can measure it.
   The control value of when ADC_SEL_0 (VM voltage detection) was set to ADC_SEL is reflected.
   Please refer to "RAJ306000 Series User's Manual: Hardware (R18UZ0066EJ0100)" about the details.

(2) Timer Array Unit

1[ms] interval timer uses "Interval timer function" of Timer Array Unit. In this system, channel 0 is used.

25[us] interval timer uses "Interval timer function" of Timer Array Unit. In this system, channel 2 is used.

Also, in this system, channel 1and channel 3 are not used.

(3) Timer RJ

Using the pulse output mode, it outputs a 4 MHz square wave and supplies it as System clock for PreDriver.

(4) Timer RG

Using the phase counting mode, it counts input pulses from the encoder.

The combination of timer input terminal and encoder signal input are shown in Table 2-6.

Table 2-6 Timer input terminal and encoder signal input

Terminal name	Encoder signal
P01/TRGCLKB	B phase
P00/TRGCLKA	A phase

(5) Timer RD

Using the Complementary PWM mode, it output (6-wire) a three-phase PWM with a triangle wave modulation and a short circuit preventive time.

In this system, support the PWM output of High active. (PWM frequency is 50[us]) In case of detect the ALARM (At the time of Input of Low signal to INTP4 port), PreDriver output signal will be change to Hi-Z (Output terminal value for Motor control signal becomes set to Low)

The combination of timer output terminal and motor control signal output are shown in Table 2-7.

Table 2-7 Timer output terminal and motor control signal output

Terminal name	Motor control signal
P10/TRDIOD1	W_n
P11/TRDIOC1	V_n
P12/TRDIOB1	W_p
P13/TRDIOA1	V_p
P14/TRDIOD0	U_n
P15/TRDIOB0	U_p

(6) Interruption

List of interruptions in this system is shown in Table 2-8.

Table 2-8 Interruption

Interruption name	Interruption source
P55/INTP4	ALARM signal detection
INTTM00	1[ms] interval timer
INTTM02	25[us] interval timer
INTTRD0	Carrier frequency (PWM)
INTCSI00	Complete of SPI00 communication for PreDriver control

2.3 Software structure

2.3.1 Software file structure

Folders and files structure of the sample program is shown in Table 2-9 and Table 2-10.

Table 2-9 Folder and Files Structure of Sample Program (1)

RAJ306000_ENCD_FOC_CLOSED_V101		
Inc	control_parameter.h	Header for control characteristic dependent processing part
	mtr_main.h	Main function, Header for user interface control
	motor_parameter.h	Header for motor characteristic dependent processing part
	mtr_calc_encd_foc.h	Header for encoder characteristic dependent processing part
	mtr_common.h	Header for Common definition
	mtr_ctrl_rl78g1f.h	Header for RL78/G1F dependent processing part
	mtr_ctrl_rl78g1f_t2001.h	Header for RL78/G1F & Board dependent processing part
	mtr_ctrl_t2001.h	Header for Board dependent processing part
	mtr_ssns_encd_foc.h	Header for vector control using encoder dependent part
	r_dsp.h	Header for operation library
	r_stdint.h	Header for operation library
	version.h	Header for software revision
ics	ics_RL78G1F_Lx.h	Header for ICS
	RL78G1F_vector.c	File for ICS_UART part
	ICS_RL78G1F_Lx.lib	Library for ICS
src	mtr_calc_encd_foc.c	Encoder characteristic dependent processing part
	mtr_ctrl_rl78g1f.c	RL78/G1F dependent processing part
	mtr_ctrl_rl78g1f_t2001.c	RL78/G1F & Board dependent processing part
	mtr_ctrl_t2001.c	Board dependent processing part
	mtr_interrupt.c	Interrupt handler
	mtr_main.c	Main function, user interface control
	mtr_ssns_encd_foc.c	Vector control using encoder dependent part

Table 2-10 Folder and Files Structure of Sample Program (2)

RAJ306000_ENCD_FOC_CLOSED_V101		
cg_src	r_cg_adc.c	RL78/G1F ADC processing
	r_cg_adc.h	RL78/G1F header of ADC processing
	r_cg_adc_user.c	RL78/G1F ADC processing (for User)
	r_cg_cgc.c	RL78/G1F clock output processing
	r_cg_cgc.h	Header for RL78/G1F clock output processing
	r_cg_cgc_user.c	RL78/G1F clock output processing (for User)
	r_cg_intp.c	RL78/G1F interrupt function processing
	r_cg_intp.h	Header for RL78/G1F interrupt function processing
	r_cg_intp_user.c	RL78/G1F interrupt function processing (for User)
	r_cg_macrodriver.h	Header for RL78/G1F Error definition
	r_cg_main.c	RL78/G1F main processing
	r_cg_main.h	Header for RL78/G1F main processing
	r_cg_port.c	RL78/G1F port function processing
	r_cg_port.h	Header for RL78/G1F port function processing
	r_cg_port_user.c	RL78/G1F port function processing (for User)
	r_cg_predrv.c	PreDriver processing
	r_cg_predrv.h	Header for PreDriver processing
	r_cg_predrv_prm.h	Header for PreDriver register parameter definition
	r_cg_predrv_reg.h	Header for PreDriver register address definition
	r_cg_predrv_user.c	PreDriver processing (for User)
	r_cg_sau.c	RL78/G1F Serial array unit processing
	r_cg_sau.h	Header for RL78/G1F Serial array unit processing
	r_cg_sau_user.c	RL78/G1F serial array unit processing (for User)
	r_cg_systeminit.c	RL78/G1F initial processing
	r_cg_tau.c	RL78/G1F timer array unit processing
	r_cg_tau.h	Header for RL78/G1F timer array unit processing
	r_cg_tau_user.c	RL78/G1F timer array unit processing (for User)
	r_cg_tmrd.c	RL78/G1F timer RD processing
	r_cg_tmrd.h	Header for RL78/G1F timer RD processing
	r_cg_tmrd_user.c	RL78/G1F timer RD processing (for User)
	r_cg_tmrg.c	RL78/G1F timer RG processing
	r_cg_tmrg.h	Header for RL78/G1F timer RG processing
	r_cg_tmrg_user.c	RL78/G1F timer RG processing (for User)
	r_cg_tmrj.c	RL78/G1F timer RJ processing
	r_cg_tmrj.h	Header for RL78/G1F timer RJ processing
	r_cg_tmrj_user.c	RL78/G1F timer RJ processing (for User)
	r_cg_userdefine.h	Header for RL78/G1F user definition
	r_cg_wdt.c	RL78/G1F watch dog timer processing
	r_cg_wdt.h	Header for RL78/G1F watch dog timer processing
	r_cg_wdt_user.c	RL78/G1F watch dog timer processing (for User)

2.3.2 Module structure

Module structure of the sample program is described on Figure 2-4.

The relationship between module and file are shown in Table 2-11.

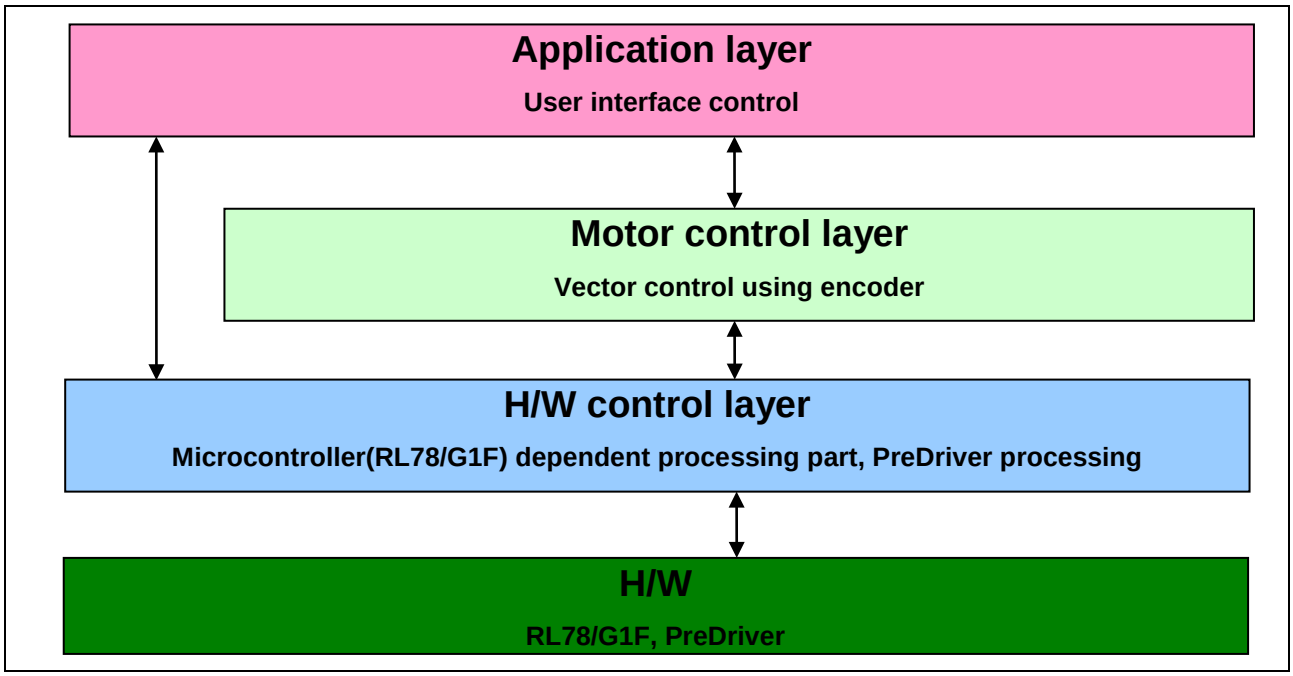

Figure 2-4 Hierarchical Structure of Sample Program

Table 2-11 Hierarchical structure of Sample Program

Application layer	mtr_main.c
Motor control layer	mtr_calc_encd_foc.c, mtr_interrupt.c, mtr_ssns_encd_foc.c
H/W control layer	mtr_ctrl_rl78g1f.c, mtr_ctrl_rl78g1f_t2001.c, mtr_ctrl_t2001.c, r_cg_adc.c, r_cg_adc_user.c, r_cg_cgc.c, r_cg_cgc_user.c, r_cg_intp.c, r_cg_intp_user.c, r_cg_main.c, r_cg_port.c, r_cg_port_user.c, r_cg_predrv.c, r_cg_predrv_user.c, r_cg_sau.c, r_cg_sau_user.c, r_cg_systeminit.c, r_cg_tau.c, r_cg_tau_user.c, r_cg_tmrd.c, r_cg_tmrd_user.c, r_cg_tmrg.c, r_cg_tmrg_user.c, r_cg_tmrj.c, r_cg_tmrj_user.c, r_cg_wdt.c, r_cg_wdt_user.c

2.4 Software specifications

Basic specifications of software of this application note are shown in Table 2-12.

Table 2-12 Software Basic Specifications

Item	Content
Control method	Vector control
Motor servo start/stop	Motor servo start/stop command is determined depending on the level of VR1(AIN6 terminal). Input from ICS (Note 4)
Position control	Position command value is determined from input voltage of VR1 (ANI6 terminal). Input from ICS (Note 4)
Rotation direction control	Rotation direction command value (CW Only) control is determined depending on the level of SW1(P122 terminal).
Rotation speed control	Rotation speed command value is determined from input voltage of VR1 (ANI6 terminal). Input from ICS (Note 4)
Rotation speed control range	CW: 50[rpm] to 750[rpm]
Position detection of rotor magnetic pole	Encoder
Carrier frequency (PWM)	20[KHz]
Control cycle	1[ms]
Processing stop for protection	Output terminal of Motor control signal is set to Low state at the time of detect the below errors. ・ALARM error ・Rotation speed abnormal error ・Timeout error

Note:

4. Please refer to the "4 Development support tool In Circuit Scope" about details.

3. Descriptions of control program

The target sample programs of this application note are explained here.

3.1 Contents of control

3.1.1 Motor servo start / stop

Starting and stopping of the motor servo are controlled by input from VR1 and SW1. An analog input port (ANI6) is assigned to VR1. The input is A/D converted within the main loop to calculate position command value and rotation speed command value. Program is judged that motor servo was started at the time of the position command value is more than 80[LSB]. and Program is judged the motor servo was stopped at the time of the position command value is less than 40[LSB]. Additional, Program is judged that motor servo was started at the time of the rotation speed command value is more than 100[rpm]. and Program is judged the motor servo was stopped at the time of the rotation speed command value is less than 50[rpm].

General-purpose port (P122 terminal) is assigned to SW1 and, in Maine loop, acquires a High/Low state of the P122 terminal and assumes it a rotation direction command value (CW Only). The rotation direction is judged from a rotation direction command value.

3.1.2 Rotation direction command value, Rotation speed command value, VM voltage.

(1) Position command value

Position command value can be set by A/D conversion of the VR1 output value (analog value) or input information from ICS.
VR1 value that A/D converted is used to position command value as shown below (Table 3-1).

Table 3-1 Conversion Ratio of the Position Command Value

Item	Conversion ratio (Command value: A/D conversion value)	Channel
Position command value	40[LSB] to 721[rpm]: 03FFH to 0000H	ANI6

(2) Rotation direction command value

Rotation direction command value (CW Only) can be set by high/low state of SW1.

(3) Rotation speed command value

Rotation speed command value can be set by A/D conversion of the VR1 output value (analog value) or input information from ICS.
VR1 value that A/D converted is used to Rotation speed command value as shown below (Table 3-2).

Table 3-2 Conversion Ratio of the Rotation Speed Command Value

Item	Conversion ratio (Command value: A/D conversion value)	Channel
Rotation speed command value	50[rpm] to 750[rpm]: 03FFH to 0000H	ANI6

(4) VM voltage

Conversion ratio of VM voltage is shown in Table 3-3.

Table 3-3 Conversion Ratio of VM Voltage

Item	Conversion ratio (VM voltage: A/D conversion value)	Channel
VM voltage	0.0[V] to 45.9[V]: 0000H to 03FFH	ANI3

(5) U phase and W phase shunt resistance voltage

Conversion ratio of U phase and W phase shunt resistance voltage is shown in Table 3-4.

Table 3-4 Conversion Ratio of U Phase and W Phase Shunt Resistance Voltage

Item	Conversion ratio (U phase and W phase shunt resistance voltage: A/D conversion value)	Channel
U phase shunt resistance voltage	0.0[V] to 5.0[V]: 0000H to 03FFH	ANI5
W phase shunt resistance voltage		ANI2

3.1.3 Vector control

Block diagram of vector control of this application note is shown in Figure 3-1.
Please refer to "Vector Control for Permanent Magnet Synchronous Motor with Encoder: Algorithm (R01AN3789EJ0101)" about the details of vector control.

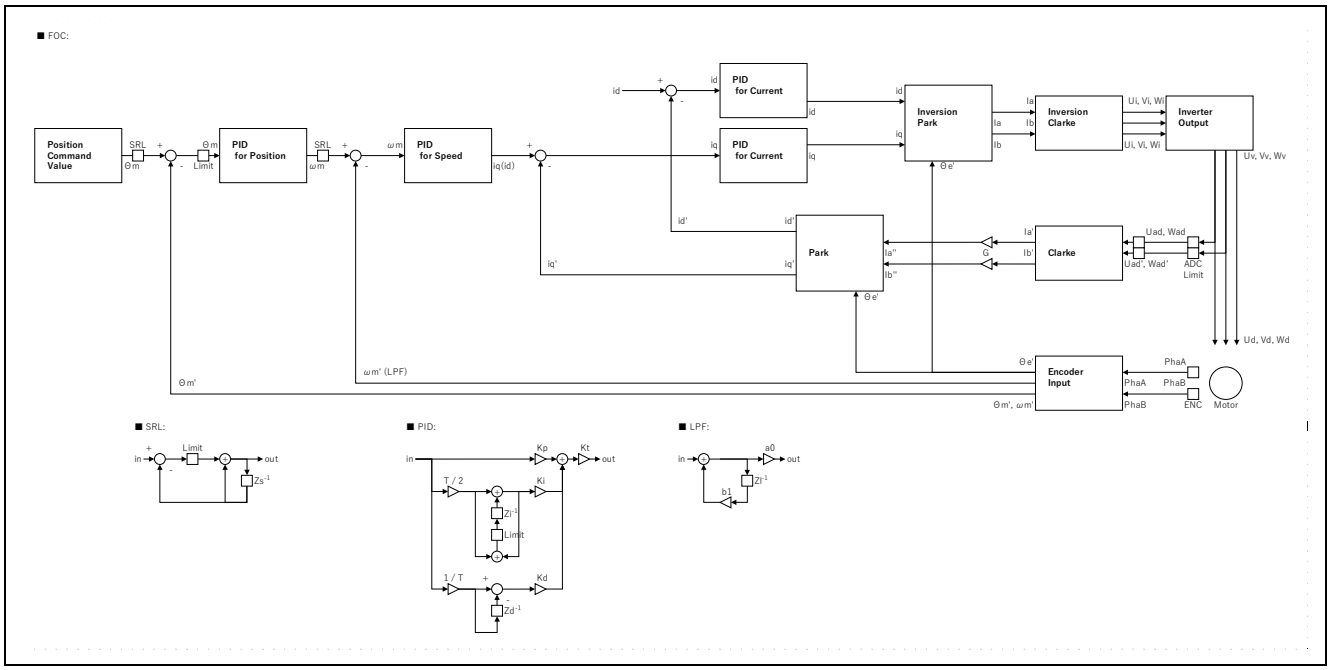

Figure 3-1 Vector control

3.1.4 Voltage control by PWM

PWM control is used for the output voltage control. The PWM control is a control method that continually adjusts the average voltage by varying the duty of pulse. Conception diagram of the PWM control is shown Figure 3-2.

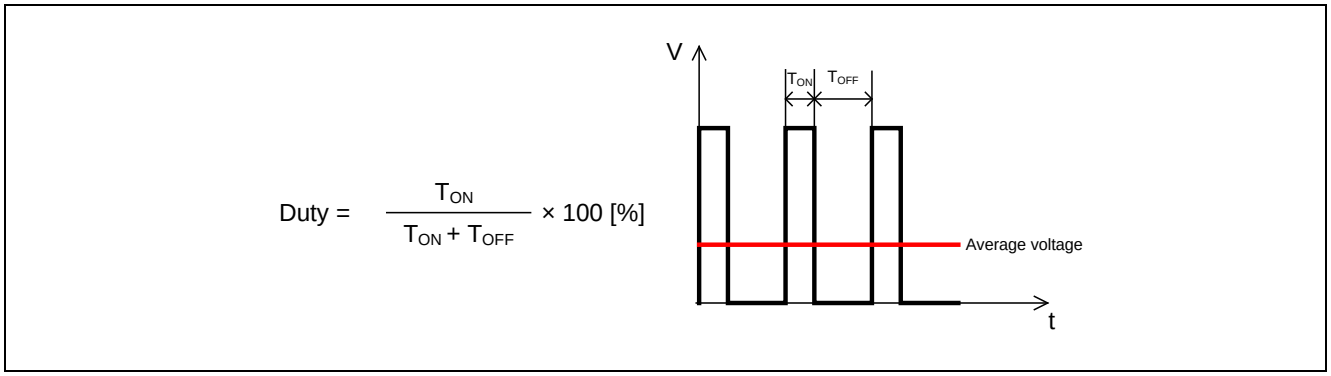

Figure 3-2 PWM control

Here, modulation factor "m" is defined as follows.

This modulation factor is reflected to the setting value of the register that determines the PWM duty.

$$m = \frac{V}{E}$$

m : Modulation factor V : Voltage command value E : VM Voltage

3.1.5 State transition

State transition diagram of the sample programs are shown in Figure 3-3.

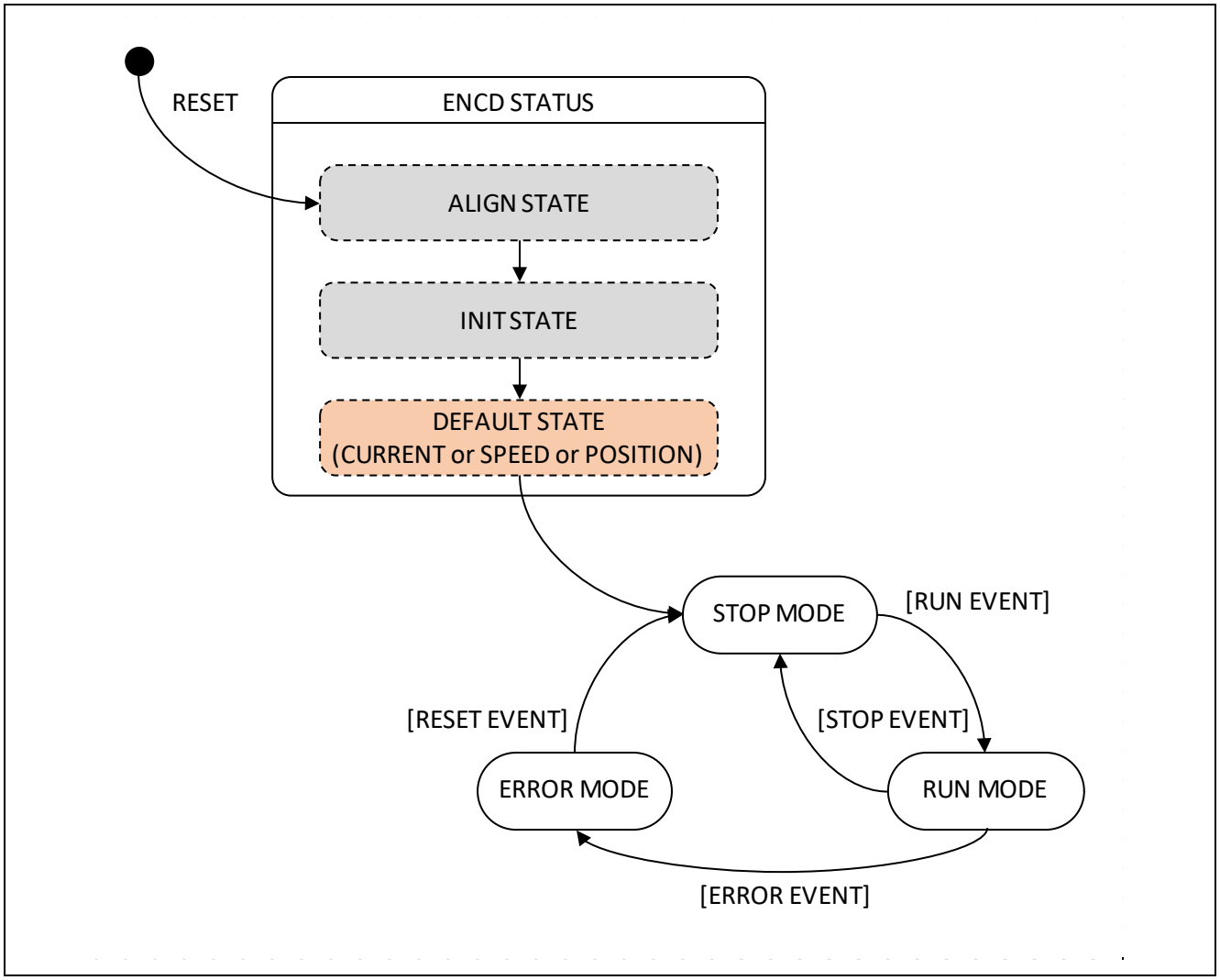

Figure 3-3 State Transition Diagram

3.1.6 System protection function

This system has below error condition. Emergency stop function is operate as each condition of the following. Each set value related to the system protection function is shown in Table 3-5.

・ALARM error

Emergency stop is performed by setting the output of PreDriver to the high impedance state (Output terminal signal for motor control is Low state) by the emergency stop signal (ALARM detection) from PreDriver.

・Rotation speed abnormality error

When the rotation speed exceeded the limit value on a cycle of the rotation speed detect operation, System is performed an emergency stop.

・Timeout error

When a motor control signal output of an interval by control cycle is not change for over Timeout limit value in Timeout error detection, System is performed an emergency stop.

Table 3-5 Setting value for Protect function of each system

Error Condition		Setting value
Rotation speed abnormality error	Rotation speed limit value	1000[rpm]
	Monitoring interval	50[us]
Timeout error	Timeout setting	20[ms]

3.1.7 System protect function (PreDriver safety function)

The PreDriver safety function can be enabled / disabled with the ALARM operation setting register (ALMOPE).

Please refer to the data sheet about details.

3.2 Function specifications

Lists of control functions are shown in Table 3-6 and Table 3-7.

Table 3-6 List of Control Functions (1)

File name	Function overview	Processing overview
mtr_main.c	main() input: none output: none	・Hardware initialization function call ・User interface initialization function call ・ Main function use variable initialization function call ・Status transition and event execution function call ・Main function - Main processing execution function call - Watchdog timer clear function call
	ctrl_ui() input: none output: none	・Change Motor status ・Determination of position command value and rotation speed command value
	ics_ui() input: none output: none	・Change Motor status ・Determination of position command value and rotation speed command value
	ctrl_led() input: none output: none	Control the output pattern of ON/OFF for LED
	ics_predrv_reg_ctrl() input: none output: none	Control for PreDriver register read/write from ICS
	mcu_sw_init() input: none output: none	Initialization of F/W ・Initialization of F/W variables ・Initialization of ICS ・Initialization of sequence processing ・Execution of RESET event ・Initialization of encoder input calculation
	software_init() input: none output: none	Initialization of variables used in the main function
mtr_ctrl_rl78g1f.c	clear_wdt() input: none output: none	Clear Flag for the watchdog timer
	mtr_clear_oc_flag() input: none output: none	Clear Flag for the pulse output forced shutdown
	mtr_clear_trd0_imfa() input: none output: none	Clear Flag for the TRD0 compare match (IMFA)

mtr_ctrl_rl78g1f_t2001.c	mtr_ctrl_start() input: none output: none	Motor startup processing
	mtr_ctrl_stop() input: none output: none	Motor stop processing
	mtr_change_pattern() input: none output: none	Change the motor control signal output
	mtr_get_adc() input: A/D channel output: A/D conversion result	Processing execution of the A/D convert
mtr_ctrl_t2001.c	get_vr1() input: none output: A/D conversion result of VR1	Obtain of the A/D conversion value of the trigger level
	led_on() input: LED channel number output: none	Turning LED ON
	led_off() input: LED channel number output: none	Turning LED OFF
mtr_interrupt.c	mtr_alarm_interrupt() input: none output: none	ALARM interrupt processing ・Change motor status ・Function call for selection of an event processing ・Function call for clear of flag of a forced interception of the pulse output
	mtr_tau0_interrupt() input: none output: none	1[ms] interrupt processing ・Vector control using encoder
	mtr_tau2_interrupt() input: none output: none	25[us] interrupt processing ・Obtain of the A/D conversion value of U phase and W phase shunt resistance voltage
	mtr_carrier_interrupt() input: none output: none	Carrier frequency interrupt processing ・Waiting for motor servo stop ・Error check function call

mtr_ssns_encd_foc.c	R_MTR_InitSequence() input: none output: none	Initialization of sequence processing
	R_MTR_ExecEvent() input: occurred event output: none	・Execute to change the status. ・Call execution function of suitable processing for the event.
	mtr_act_run() input: motor status output: motor status	・Variable initialization function call upon motor startup ・Motor control startup function call
	mtr_act_stop() input: motor status output: motor status	Motor control stop function call
	mtr_act_none() input: motor status output: motor status	No processing is performed.
	mtr_act_reset() input: motor status output: motor status	Initialization of Global variable for return from Error state.
	mtr_act_error() input: motor status output: motor status	Motor control stop function call at the time of Error occur.
	mtr_pattern_set() input: none output: none	Motor control signal output change function call
	mtr_start_init() input: none output: none	Initializing only the variables required at the time of motor startup
	mtr_set_variables() input: none output: none	Set Input data at ICS to Protecting variable.
	R_MTR_IcsInput() input: structure of ICS variables output: none	Obtaining of variable that inputted from the ICS.
	R_MTR_SetSpeed() input: rotation speed command value output: none	Rotation speed setting
	R_MTR_SetDir() input: rotation direction command value output: none	Rotation direction setting
	R_MTR_GetDir() input: none output: rotation direction information	Obtaining the rotation direction.
	R_MTR_GetStatus() input: none output: motor status	Obtaining the motor status
	mtr_error_check() input: none output: none	Monitoring and Detection of Error

	R_MTR_InitEncoderVector() input: none output: none	Initialization of variables of vector control using encoder
	R_MTR_SetFlagCtrlDuty() input: Flag for PWM duty control output: none	Flag for PWM duty control setting
mtr_calc_encd_foc.c	mtr_calc_LPF() input: LPF input output: Calculation result of LPF	Processing of SRL calculation
	mtr_calc_InitVariables() input: none output: none	Initialization of variables of vector control using encoder
	mtr_calc_PID() input: PID input PID parameter output: Calculation result of PID	Processing of PID calculation
	mtr_calc_SRL() input: SLR input Delay element Limit value output: Calculation result of SLR Delay element	Processing of SRL calculation
	mtr_calc_EncoderInput() input: none output: Calculation result of position (electrical angle) Calculation result of position (mechanical angle) Calculation result of rotation speed (mechanical angle)	Initialization processing of encoder input calculation

Table 3-7 List of Control functions (2)

File name	Function overview	Processing overview
r_cg_adc.c	R_ADC_Create() input: none output: none	Initialization of A/D converter
r_cg_adc_user.c	r_adc_interrupt() input: none output: none	SPI communication ADC mode SPI start judgement
r_cg_cgc.c	R_CGC_Create() input: none output: none	Initialization of clock frequency(CGC)
r_cg_intp.c	R_INTP_Create() input: none output: none	Initialization of external interrupt (INTP)
r_cg_main.c	R_MAIN_UserInit() input: none output: none	PreDriver startup processing
r_cg_port.c	R_PORT_Create() input: none output: none	Initialization of I/O port setting
r_cg_predrv.c	predriver_hw_init() input: none output: none	PreDriver initialization setting
	R_PREDRV_TRIM_Create() input: none output: SPI status	PreDriver trimming data setting
	R_PREDRV_InitSequence() input: none output: none	PreDriver initialization processing
	R_PREDRV_ ErrorRecoverySequence() input: ALARM status output: none	ALARM recovery processing
r_cg_predrv_user.c	R_PreDrvReg_Read() input: read address output: SPI status, read data	Read processing to PreDriver register
	R_PreDrvReg_Write() input: write address, write data output: SPI status	Write processing to PreDriver register

r_cg_sau.c	R_SAU0_Create() input: none output: none	Initialization of serial array unit(SAU)
	R_UART1_Create() input: none output: none	Initialization of UART1
	R_CSI00_Create() input: none output: none	Initialization of SPI communication (for PreDriver communication)
	R_CSI00_Start() input: none output: none	Startup SPI communication (for PreDriver communication)
	R_CSI00_ Send_Receive_SPI_mode() input: tx buffer buffer size rx buffer SPI mode output: SPI status	SPI communication processing
r_cg_sau_user.c	r_csi00_interrupt() input: none output: none	SPI interrupt processing (for PreDriver communication)
r_cg_systeminit.c	hdwinit() input: none output: none	Initial setting of H/W
r_cg_tau.c	R_TAU0_Create() input: none output: none	Initialization of TAU
	R_TAU0_Channel0_Start() input: none output: none	Start counting the 1[ms] interval timer
	R_TAU0_Channel2_Start() input: none output: none	Start counting the 25[us] interval timer
	R_TAU0_Channel2_Stop() input: none output: none	Stop counting the 25[us] interval timer
r_cg_tmrd.c	R_TMRD0_Create() input: none output: none	Initialization of Timer RD(TRD)
	R_TMRD0_Start() input: none output: none	PWM output start
r_cg_tmrg.c	R_TMRG0_Create() input: none output: none	Initialization of Timer RG(TRG)
	R_TMRG0_Start() input: none output: none	Start counting of the encoder

r_cg_tmrj.c	R_TMRJ0_Create() input: none output: none	Initialization of Timer RJ(TRJ)
	R_TMRJ0_Start() input: none output: none	Start supply a clock for PreDriver
r_cg_wdt.c	R_WDT_Create() input: none output: none	Initialization of Watch dog timer

3.3 Specification of variables

Lists of variables for the sample program are shown in Table 3-8.

Table 3-8 List of Variables

Variable name	Type	Content	Remarks
g_s2_min_angle	int16_t	Position command maximum value	A/D conversion value of the trigger level
g_s2_min_angle	int16_t	Position command minimum value	A/D conversion value of the trigger level
g_s2_margin_min_angle	int16_t	Position command minimum value for motor stop	A/D conversion value of the trigger level
g_s2_max_speed	int16_t	Rotation speed command maximum value	Mechanical angle [rpm]
g_s2_min_speed	int16_t	Rotation speed command minimum value	Mechanical angle [rpm]
g_s2_margin_min_speed	int16_t	Rotation speed command minimum value for motor stop	Mechanical angle [rpm]
g_s2_ref_speed	int16_t	Setting of rotation speed by user	Mechanical angle [rpm]
g_u1_rot_dir	uint8_t	Setting of rotation direction by user	0: CW 1: CCW
g_u1_motor_status	uint8_t	Management of motor status by user	0: Stop 1: Rotating 2: Error
g_u1_stop_req	uint8_t	Motor stop command flag	Stop is determined when the position command value is less than 40[LSB] or rotation speed command value is less than 50[rpm]
g_u1_pdrv_status	uint8_t	Register Read/Write for PreDriver Error status	-
g_u1_err_recovery_req	uint8_t	ALARM recovery processing request flag	0: Disable 1: Enable
g_u1_get_alarm_sts1	uint8_t	PreDriver register ALMSTS1 acquired value	-
g_u1_store_alarm_sts1	uint8_t	PreDriver register ALMSTS1 stored value	-
g_u1_get_alarm_sts2	uint8_t	PreDriver register ALMSTS2 acquired value	-
g_u1_store_alarm_sts2	uint8_t	PreDriver register ALMSTS2 stored value	-
g_u2_fw_revision	uint16_t	F/W Revision information	F/W Version information
g_s2_sw_userif	int16_t	Flag for switch of HW UI	0: not use HW UI 1: use HW UI
g_s2_mode_system	int16_t	Flag for system mode	0: Stop 1: Motor startup 2: Error 3: Reset
g_s2_enable_write	int16_t	Flag for Write enable of ICS	Toggle operate
ics_input	MTR_ICS_INPUT	Structure for input of ICS	-
g_s2_ref_speed_rpm_vr1	int16_t	Rotation speed command value	Mechanical angle [rpm]
g_u1_alarm_sts1	uint8_t	PreDriver register ALMSTS1 stored value	For ICS display
g_u1_alarm_sts2	uint8_t	PreDriver register ALMSTS2 stored value	For ICS display
g_u2_cnt_wait_stop	uint16_t	Motor servo stop waiting counter	10[ms] after motor stop processing is counted. (Note that the count is reset when a hall IC signal interrupt is detected.)

g_u1_flg_wait_stop	uint8_t	Flag for waiting time of Motor servo stop	The flag set upon motor stop command. When no hall IC signal interrupt is detected for 10[ms] after motor stop processing, the flag is cleared.
g_u1_enable_write	uint8_t	Flag for Write enable of Structure for ICS input	0: Disable 1: Enable
g_s2_vdc_ad	int16_t	A/D conversion value of VM voltage	[V]
g_s2_pdrv_ad	int16_t	A/D conversion value of PreDriver voltage	[V]
g_u1_cnt_ics	uint8_t	Counter for interval of ICS function call	-
g_u1_error_status	uint8_t	Error status management	0x01: ALARM error 0x04: Rotation speed abnormality error 0x08: Timeout error (0x80: Undefined error)
g_u1_mode_system	uint8_t	Management of system mode	0: Stop, 1: Run, 2: Error
g_s2_foc_encd_status	int16_t	Status management of vector control using encoder	0: Initialization processing (Align the position) 1: initialization processing (move to the initial position) 2: Current control 3: Rotation speed control 4: Position control
g_s2_foc_mech_count	int16_t	Encoder count	Mechanical angle
g_s2_foc_elec_count	int16_t	Encoder count	Electrical angle
g_s2_foc_ref_theta	int16_t	Position command value	Mechanical angle (Normalized: 0 to 4095) [rad]
g_s2_foc_ref_omega	int16_t	Rotation speed command value	Mechanical angle (Scale: Q4) [rad]
g_s2_foc_theta	int16_t	Calculated value of position	Mechanical angle (Normalized: 0 to 4095) [rad]
g_s2_foc_theta_elec	int16_t	Encoder position	Electrical angle (Normalized: 0 to 4095) [rad]
g_s2_foc_theta_mech	int16_t	Encoder position	Mechanical angle (Normalized: 0 to 4095) [rad]
g_s2_foc_theta_zs	int16_t	Delay element of position SRL	-
g_f4_foc_theta_zi	float32_t	Delay element of position PID control (integration)	-
g_f4_foc_theta_zd	float32_t	Delay element of position PID control (differentiation)	-
pid_foc_theta	MTR_PID_PRM	PID parameter for position control	-
g_s2_foc_omega	int16_t	Calculated value of rotation speed	Mechanical angle (Scale: Q4) [rad/s]
g_s2_foc_omega_mech	int16_t	Encoder rotation speed	Mechanical angle (Scale: Q4) [rad/s]
g_s2_foc_omega_zs	int16_t	Delay element of speed SRL	-
g_f4_foc_omega_zi	float32_t	Delay element of speed PID control (integration)	-
g_f4_foc_omega_zd	float32_t	Delay element of speed PID control (differentiation)	-
pid_foc_omega	MTR_PID_PRM	PID parameter for rotation speed control	-
g_s2_foc_id	int16_t	Calculated value of d-axis current	Input value of inverse park transformation (Scale: Q12)
g_f4_foc_id_zi	float32_t	Delay element of d-axis current SRL	-
g_f4_foc_id_zd	float32_t	Delay element of d-axis current PID control (integration)	-
pid_foc_id	MTR_PID_PRM	Delay element of d-axis current PID control (differentiation)	-
g_s2_foc_iq	int16_t	Calculated value of q-axis current	Input value of inverse park transformation (Scale: Q12)
g_f4_foc_iq_zi	float32_t	Delay element of q-axis current SRL	-

g_f4_foc_iq_zd	float32_t	Delay element of q-axis current PID control (integration)	-
pid_foc_iq	MTR_PID_PRM	Delay element of q-axis current PID control (differentiation)	-
g_s2_foc_ia	int16_t	Alpha-axis current value	Output value of inverse park transformation and Input value of inverse clarke transformation (Scale: Q12)
g_s2_foc_ib	int16_t	Beta-axis current value	Output value of inverse park transformation and Input value of inverse clarke transformation (Scale: Q12)
g_s2_foc_ui	int16_t	U phase current value	Output value of inverse clarke transformation (Scale: Q12)
g_s2_foc_vi	int16_t	V phase current value	Output value of inverse clarke c transformation (Scale: Q12)
g_s2_foc_wi	int16_t	W phase current value	Output value of inverse clarke transformation (Scale: Q12)
g_f4_foc_coef_k	float32_t	Coefficient for A/V conversion	-
g_s2_foc_uv	int16_t	U phase voltage value (PWM Duty)	[%]
g_s2_foc_vv	int16_t	V phase voltage value (PWM Duty)	[%]
g_s2_foc_wv	int16_t	W phase voltage value (PWM Duty)	[%]
g_u2_foc_ud	uint16_t	Setting value of U phase timer RD compare register	-
g_u2_foc_vd	uint16_t	Setting value of V phase timer RD compare register	-
g_u2_foc_wd	uint16_t	Setting value of W phase timer RD compare register	-
g_s2_foc_uad_offset	int16_t	Offset value of U phase shunt resistance voltage	A/D conversion value of shunt resistance voltage value
g_s2_foc_wad_offset	int16_t	Offset value of W phase shunt resistance voltage	A/D conversion value of shunt resistance voltage value
g_s2_foc_uad	int16_t	U phase shunt resistance voltage value	A/D conversion value of shunt resistance voltage value
g_s2_foc_wad	int16_t	W phase shunt resistance voltage value	A/D conversion value of shunt resistance voltage value
g_s2_foc_adc_toggle	int16_t	Toggle switch for A/D conversion	2: Measurement of U phase shunt resistance voltage 5: Measurement of W phase shunt resistance voltage
g_s2_foc_uadd	int16_t	U phase AMP output voltage value	Input value of clarke transformation (Scale: Q12)
g_s2_foc_wadd	int16_t	W phase AMP output voltage value	Input value of clarke transformation (Scale: Q12)
g_s2_foc_iad	int16_t	Alpha-axis current value	Output value of clarke transformation (Scale: Q12)
g_s2_foc_ibd	int16_t	Beta-axis current value	Output value of clarke transformation (Scale: Q12)
g_s2_foc_iadd	int16_t	Alpha-axis current value	Input value of park transformation (Scale: Q12)
g_s2_foc_ibdd	int16_t	Beta-axis current value	Input value of park transformation (Scale: Q12)
g_s2_foc_idd	int16_t	d-axis current value	Output value of park transformation (Scale: Q12)
g_s2_foc_iqd	int16_t	q-axis current value	Output value of park transformation (Scale: Q12)
g_u2_foc_flg_ctrl_duty	uint16_t	Flag for PWM duty control	0: Disable 1: Enable

g_u2_cnt_timeout	uint16_t	Stop determination time measurement counter	Cleared when the motor control signal output change function is called
g_u1_direction	uint8_t	Rotation direction management	0: CW 1: CCW
ics_input_buff	MTR_ICS_INPUT	ICS input variable structure	-
g_u2_foc_encd_count	uint16_t	Calculated value of encoder count	-
g_s2_foc_calc_count	int16_t	Previous value of encoder count	-
g_f4_foc_lpf_zl	float32_t	Delay element of speed LPF	-
g_u1_PreDriver_error	uint8_t	PreDriver sequence error status	・PreDriver initial sequence ・ALARM recovery sequence
g_spi00_comend_flag	uint8_t	SPI communication condition flag	TURE: communication end FALSE: connecting
g_spi00_adcend_flag	uint8_t	SPI communication ADC End flag	TURE: ADC end FALSE: ADC executing
g_spi00_commode	uint8_t	SPI communication mode	-
gp_csi00_rx_address	uint8_t	SPI communication receives data address	Obtain of PreDriver register value
g_csi00_rx_length	uint16_t	SPI communication receives data length	-
g_csi00_rx_count	uint16_t	SPI communication receives counter	-
gp_csi00_tx_address	uint8_t	SPI communication transmission data address	Designation of PreDriver register address
g_csi00_send_length	uint16_t	SPI communication transmission data length	-
g_csi00_tx_count	uint16_t	SPI communication transmission counter	-

3.4 Specification of Macro definition

Lists of macro definitions used in this sample program are shown in Table 3-9

Table 3-9 List of Macro Definitions

File name	Macro name	Definition value	Remarks
control_parameter.h	CP_MAX_SPEED_RPM	750	Rotation speed command maximum value (Mechanical angle) [rpm]
	CP_MIN_SPEED_RPM	100	Rotation speed command minimum value (Mechanical angle) [rpm]
mtr_main.h	ICS_UI	0	Set UI to ICS
	BOARD_UI	1	Set UI to Board
	M_CW	0	User setting rotation direction: CW
	M_CCW	1	User setting rotation direction: CCW
	MAX_ANGLE	721	Position command maximum value (A/D conversion value of the trigger level)
	MIN_ANGLE	80	Position command minimum value (A/D conversion value of the trigger level)
	MARGIN_ANGLE	40	Position command minimum value creation constants for motor stop (A/D conversion value of the trigger level)
	MARGIN_MIN_ANGLE	(MIN_ANGLE - MARGIN_ANGLE)	Position command minimum value for motor stop (A/D conversion value of the trigger level)
	MAX_SPEED	CP_MAX_SPEED_RPM	Rotation speed command maximum value (mechanical angle) [rpm]
	MIN_SPEED	CP_MIN_SPEED_RPM	Rotation speed command minimum value (mechanical angle) [rpm]
	MARGIN_SPEED	50	Rotation speed command minimum value creation constants for motor stop (mechanical angle) [rpm]
	MARGIN_MIN_SPEED	(MIN_SPEED - MARGIN_SPEED)	Rotation speed command minimum value for motor stop (mechanical angle) [rpm]
	REQ_CLR	0	Clear Flag for stop command
	REQ_SET	1	Set Flag for stop command
	LED_ON_1ST_SPEED	250	rotation speed LED3 ON
	LED_ON_2ND_SPEED	500	rotation speed LED4 ON
	REQ_ROT_CCW	0	CCW: Acquisition value of Rotation direction port
	REQ_ROT_CW	1	CW: Acquisition value of Rotation direction port
motor_parameter.h	MP_POLE_PAIRS	7	Constant for correcting number of pole pairs

mtr_ctrl_rl78g1f_t2001.h	MTR_PWM_TIMER_FREQ	64.0f	PWM timer count frequency [MHz]
	MTR_CARRIER_FREQ	20.0f	Carrier frequency [KHz]
	MTR_DEADTIME	0	Dead Time [ns]
	MTR_DEADTIME_SET	((MTR_DEADTIME * MTR_PWM_TIMER_FREQ) / 1000)	Dead Time setting value
	MTR_CARRIER_SET	((((MTR_PWM_TIMER_FREQ * 1000) / MTR_CARRIER_FREQ) / 2) + MTR_DEADTIME_SET - 2)	Carrier setting value
	MTR_START_CARRIER_SET	((MTR_CARRIER_SET * 50) / 100)	Carrier setting value (initial value)
	MTR_VR1_ADC_MAX	802	A/D conversion maximum value of the trigger level
	MTR_THETA_CALC_COEF	(4095.0f / MTR_VR1_ADC_MAX)	Coefficient for calculating position command value
	MTR_OMEGA_CALC_COEF	(MTR_TWOPI / 60.0f * 16.0f)	Coefficient for calculating rotation speed command value
	MTR_RPM_CALC_COEF	(1221.7f / MTR_VR1_ADC_MAX)	Coefficient for calculating target rotation speed
	MTR_GET_ROT_DIR_REQ	P12.2	Rotation direction detection port
	MTR_PORT_LED3	P14.1	LED3 output port
	MTR_PORT_LED4	P14.0	LED4 output port
	MTR_PORT_LED5	P4.3	LED5 output port
	MTR_PORT_LED6	P4.2	LED6 output port
	MTR_LED_ON	0	Active in case of Low
	MTR_LED_OFF	1	
	MTR_VDC_SCALING	1471	Resolution of A/D conversion value of VM voltage
	MTR_ADCCH_RAJ306000_TEMP	0	A/D converter channel for RAJ306000 temperature measurement
	MTR_ADCCH_MOS_TEMP	1	A/D converter channel for MOS FET temperature measurement
	MTR_ADCCH_IU	2	A/D converter channel for U phase shunt resistance voltage measurement
	MTR_ADCCH_VM	3	A/D converter channel for VM voltage measurement
	MTR_ADCCH_IW	5	A/D converter channel for W phase shunt resistance voltage measurement
	MTR_ADCCH_VR1	6	A/D converter channel for trigger level
	MTR_ADCCH_PDRV	7	A/D converter channel for PreDriver voltage measurement
mtr_ctrl_t2001.h	MTR_LED3	3	LED pattern
	MTR_LED4	4	
	MTR_LED5	5	
	MTR_LED6	6	
mtr_ssns_encd_foc.h	MTR_TWOPI	(2 * 3.14159265f)	2 Pi
	MTR_SPEED_LIMIT_RPM	1000	Limit value of Rotation speed (Mechanical angle) [rpm]

	MTR_SPEED_LIMIT	((MTR_SPEED_LIMIT_RPM / 60.0f) * MTR_TWOPI * 16.0f)	Limit value of Rotation speed (Mechanical angle) [rad/s]
	MTR_TIMEOUT_CNT	400	Waiting time for judgement of motor stop (Count value x 50[us])
	MTR_CW	0	Rotation direction setting value: CW
	MTR_CCW	1	Rotation direction setting value: CCW
	MTR_FLG_CLR	0	Constant for flag clear
	MTR_FLG_SET	1	Constant for flag setting
	MTR_STOP_WAIT_CNT	200	Period to wait for motor stop (Count value x 50[us])
	MTR_ICS_DECIMATION	4	Number of function call decimation times for ICS (Count value x 50[us])
	MTR_ALARM_ERROR	0x01	ALARM error
	MTR_OVER_SPEED_ERROR	0x04	Rotation speed abnormality error
	MTR_TIMEOUT_ERROR	0x08	Timeout error
	MTR_UNKNOWN_ERROR	0x80	Undefined error
	MTR_MODE_STOP	0x00	Stop status
	MTR_MODE_RUN	0x01	Rotating status
	MTR_MODE_ERROR	0x02	Error status
	MTR_SIZE_STATE	3	Status count
	MTR_EVENT_STOP	0x00	Motor stop event
	MTR_EVENT_RUN	0x01	Motor startup event
	MTR_EVENT_ERROR	0x02	Motor error event
	MTR_EVENT_RESET	0x03	Motor reset event
	MTR_SIZE_EVENT	4	Events count
mtr_calc_encd_foc.h	MTR_ENCD_CPR_MECH	1200	Crop value of encoder count
	MTR_ENCD_POLE_PAIRS	MP_POLE_PAIRS	Constant for pole pairs correction
	MTR_ENCD_STATE_ALIGN	0	Initialization processing (Align the position)
	MTR_ENCD_STATE_INIT	1	initialization processing (move to the initial position)
	MTR_ENCD_STATE_CURRENT	2	Current control
	MTR_ENCD_STATE_SPEED	3	Rotation speed control
	MTR_ENCD_STATE_POSITION	4	Position control
	MTR_ENCD_STATE_DEFAULT	MTR_ENCD_STATE_POSITION	Default State
	MTR_COEF_ENCD_THETA	3.413333333f	Coefficient for convert encoder count to position
	MTR_COEF_ENCD_OMEGA	5.235987756f	Coefficient for convert encoder count (delta) to rotation speed
	MTR_COEF_LPF1	0.7f	Coefficient 1 for speed LPF
	MTR_COEF_LPF2	0.3f	Coefficient 2 for speed LPF
	MTR_COEF_AMP	20.01955034f	Coefficient for convert shunt resistance voltage to AMP output voltage
	MTR_COEF_G	0.1f	Current gain of stationary coordinates
	MTR_COEF_K	2216.684724f	Coefficient for A/V conversion
	MTR_CENTER_AMP	10240	Center value of AMP output voltage
	MTR_CENTER_ADC	512	Center value of A/D conversion
	MTR_CENTER_PWM_DUTY	50	Center value of PWM Duty

	MTR_FIXED_THETA	585	Fixed value of position
	MTR_FIXED_OMEGA	0	Fixed value of rotation speed
	MTR_FIXED_ID	20480	Fixed value of d-axis current
	MTR_FIXED_IQ	0	Fixed value of q-axis current
	MTR_LIMIT_ADC	1023U	limit value of A/D conversion (shunt resistance voltage)
	MTR_LIMIT_THETA	4095U	limit value of position
	MTR_SRL_THETA	10U	Value of position SRL
	MTR_SRL_OMEGA	10U	Value of speed SRL
	MTR_PID_LIMIT_ZI	4000	Limit value of delay element of position PID
	MTR_PID_THETA_TH	100	Threshold for switching PID parameter for position control
	MTR_PID_THETA_KP	100.0f	Gain of position PID control (proportion)
	MTR_PID_THETA_TI	0.0005f	Sampling gain of position PID control (integration)
	MTR_PID_THETA_KI	0.3f	Gain of position PID control (integration)
	MTR_PID_THETA_TD	1000.0f	Sampling gain of position PID control (differentiation)
	MTR_PID_THETA_KD	0.001f	Gain of position PID control (differentiation)
	MTR_PID_THETA_KT	0.004f	Output gain of position PID control
	MTR_PID_THETA_KP_H	160.0f	Gain of position PID control (proportion) (High gain setting for stationary)
	MTR_PID_THETA_KT_H	0.005f	Output gain of position PID control (High gain setting for stationary)
	MTR_PID_OMEGA_KP	1.0f	Gain of speed PID control (proportion)
	MTR_PID_OMEGA_TI	0.0005f	Sampling gain of speed PID control (integration)
	MTR_PID_OMEGA_KI	0.01f	Gain of speed PID control (integration)
	MTR_PID_OMEGA_TD	1000.0f	Sampling gain of speed PID control (differentiation)
	MTR_PID_OMEGA_KD	0.001f	Gain of speed PID control (differentiation)
	MTR_PID_OMEGA_KT	0.075f	Output gain of speed PID control
	MTR_PID_OMEGA_KT_H	0.1f	Output gain of speed PID control (High gain setting for stationary)
	MTR_PID_ID_KP	0.5f	Gain of d-axis current PID control (proportion)
	MTR_PID_ID_TI	0.0005f	Sampling gain of d-axis current PID control (integration)
	MTR_PID_ID_KI	0.0f	Gain of d-axis current PID control (integration)
	MTR_PID_ID_TD	1000.0f	Sampling gain of d-axis current PID control (differentiation)
	MTR_PID_ID_KD	0.0f	Gain of d-axis current PID control (differentiation)
	MTR_PID_ID_KT	0.26f	Output gain of d-axis current PID control

	MTR_PID_IQ_KP	1000.0f	Gain of q-axis current PID control (proportion)
	MTR_PID_IQ_TI	0.0005f	Sampling gain of q-axis current PID control (integration)
	MTR_PID_IQ_KI	0.1f	Gain of q-axis current PID control (integration)
	MTR_PID_IQ_TD	1000.0f	Sampling gain of q-axis current PID control (differentiation)
	MTR_PID_IQ_KD	0.0f	Gain of q-axis current PID control (differentiation)
	MTR_PID_IQ_KT	0.0013f	Output gain of q-axis current PID control
version.h	FW_REVISION	-	F/W Revision information

File Name	Macro Name	Content	Remark
r_cg_userdefine.h	SPI00_CS_H	(P0 = P0 | 0x20)	SPI communication Chip Select signal = H
	SPI00_CS_L	(P0 = P0 & 0xDF)	SPI communication Chip Select signal = L
	SPI_WAIT_MODE	0x01	SPI communication Wait mode
	SPI_INTR_MODE	0x02	SPI communication Interrupt mode
	SPI_ADC_MODE	0x03	SPI communication ADC mode

File Name	Macro Name	Content	Remarks
r_cg_predrv.h	REG_BUFF_SIZE	2	PreDriver register buffer size
	SPI_CHK_MAX	100	PreDriver SPI communication check count
	PREDRV_NORMAL	0	PreDriver sequence none
	PREDRV_SPI_ERROR	1	PreDriver sequence SPI communication error
	PREDRV_ALARM_ERROR	2	PreDriver sequence ALARM error
	PREDRV_REGRW_ERROR	4	PreDriver sequence Register R/W error
	MOTOR_LOCK	0	Motor lock status
	MOTOR_UNLOCK	1	Motor unlock status
	PREDRV_SPI_ACCESS_OK	0x6A	PreDriver SPI communication judgement
	PREDRV_ALMRAW1_OK	0xEF	PreDriver ALMRAW1 judgement
	ALMSTS1_TSD_N	0x01	ALARM Status1 judgement
	ALMSTS1_OCP_N	0x02	
	ALMSTS1_VGB_UVP_N	0x04	
	ALMSTS1_VGB_OVP_N	0x08	
	ALMSTS1_VGT_UVP_N	0x10	
	ALMSTS1_VGT_OVP2_N	0x20	
	ALMSTS1_VGT_OVP1_N	0x40	
	ALMSTS1_VREG5_OVP_N	0x80	
	ALMSTS1_NO_ERROR	0xEF	
	ALMSTS1_VGT_UVP_MASK	0xEF	
	ALMSTS2_VM_UVP_N	0x01	ALARM Status2 judgement
	ALMSTS2_DI_SEL_W_CMP_N	0x20	
	ALMSTS2_DI_SEL_V_CMP_N	0x40	
	ALMSTS2_DI_SEL_U_CMP_N	0x80	
	ALMSTS2_NO_ERROR	0xFF	
	INIT_PS_ALL	0x01	PS_ALL initial value
	INIT_PS_1ST	0x3A	PS initial value 1st
	INIT_PS_2ND	0x3B	PS initial value 2nd
	INIT_PS_3RD	0xBB	PS initial value 3rd
	INIT_SELSIG_U	0x03	SELSIG_U initial value
	INIT_SELSIG_V	0x14	SELSIG_V initial value
	INIT_SELSIG_W	0x25	SELSIG_W initial value
	INIT_HALL_SIG	0x00	HALL_SIG initial value
	INIT_ALMOPE1	0x10	ALMOPE1 initial value
	INIT_ALMOUT1	0x10	ALMOUT1 initial value

	INIT_CS_SET2	0x60	CS_SET2 initial value
	INIT_ERROR_WAIT	0x00	ERROR_WAIT initial value
	INIT_CS_SET1	0x08	CS_SET1 initial value
	INIT_HAIC_TH	0x00	HAIC_TH initial value
	INIT_LD_WAIT	0x00	LD_WAIT initial value
	INIT_DRIVE_SET	0x01	DRIVE_SET initial value
	INIT_IDRCNT_H	0x00	IDRCNT_H initial value
	INIT_IDRCNT_L	0x00	IDRCNT_L initial value
	INIT_TRCNT_P	0x00	TRCNT_P initial value
	INIT_CPSET1	0x01	CPSET1 initial value
	INIT_CPSET2	0x02	CPSET2 initial value
	INIT_CP_TRIM	0x00	CP_TRIM initial value
	INIT_VREG5_TRIM	0x20	VREG5_TRIM initial value
	INIT_CSAMP_TRIM	0x20	CSAMP_TRIM initial value
	INIT_TRIM_PT	0x00	TRIM_PT initial value protected
	INIT_TRIM_PT_UP	0x95	TRIM_PT initial value unprotected
	INIT_TRIM_EN	0x00	TRIM_EN initial value
	INIT_TRIM_EN_EFWD	0x01	TRIM_EN initial value valid trimming data
	INIT_BGR_TRIM	0x00	BGR_TRIM initial value
	INIT_BFAMP_TRIM	0x00	BFAMP_TRIM initial value
	ERRRCV_PS_1ST	0x38	PS ALARM recovery value1st
	ERRRCV_PS_2ND	0x3A	PS ALARM recovery value 2nd
	ERRRCV_PS_3RD	0x3B	PS ALARM recovery value 3rd
	ERRRCV_PS_4TH	0xBB	PS ALARM recovery value 4th
	ERRRCV_MOT_EN_CLR	0x00	DRIVE_SET ALARM recovery value Prohibition of motor rotation
	ERRRCV_MOT_EN_SET	0x01	DRIVE_SET ALARM recovery value Permission of motor rotation
	ERRRCV_ALM_LATCH_CLR	0x40	DRIVE_SET ALARM recovery value Clear of ALARM latch
	WAITTIME_1_MS	0x11F8	1[ms] wait
	WAITTIME_3_MS	0x35E8	3[ms] wait
	INIT_ICS_PS_ALL	INIT_PS_ALL	PS_ALL ICS variable initial value

	INIT_ICS_PS	INIT_PS_3RD	PS ICS variable initial value
	INIT_ICS_SW_RESET	0x00	SW_RESET ICS variable initial value
	INIT_ICS_ADC_SEL	0x00	ADC_SEL ICS variable initial value
	INIT_ICS_SELSIG_U	0x03	SELSIG_U ICS variable initial value
	INIT_ICS_SELSIG_V	0x14	SELSIG_V ICS variable initial value
	INIT_ICS_SELSIG_W	0x25	SELSIG_W ICS variable initial value
	INIT_ICS_HALL_SIG	0x00	HALL_SIG ICS variable initial value
	INIT_ICS_ALMSTS1	0xFF	ALMSTS1 ICS variable initial value
	INIT_ICS_ALMOPE1	INIT_ALMOPE1	ALMOPE1 ICS variable initial value
	INIT_ICS_ALMOUT1	INIT_ALMOUT1	ALMOUT1 ICS variable initial value
	INIT_ICS_ALMSTS2	0xFF	ALMSTS2 ICS variable initial value
	INIT_ICS_CS_SET2	INIT_CS_SET2	CS_SET2 ICS variable initial value
	INIT_ICS_ALMOUT2	0x00	ALMOUT2 ICS variable initial value
	INIT_ICS_ERROR_WAIT	0x00	ERROR_WAIT ICS variable initial value
	INIT_ICS_CS_SET1	INIT_CS_SET1	CS_SET1 ICS variable initial value
	INIT_ICS_HAIC_TH	0x00	HAIC_TH ICS variable initial value
	INIT_ICS_PDDSTS	0xF0	PDDSTS ICS variable initial value
	INIT_ICS_LD_WAIT	0x00	LD_WAIT ICS variable initial value
	INIT_ICS_DRIVE_SET	INIT_DRIVE_SET	DRIVE_SET ICS variable initial value
	INIT_ICS_DI_TIME	0x00	DI_TIME ICS variable initial value
	INIT_ICS_IDRCNT_H	0x00	IDRCNT_H ICS variable initial value
	INIT_ICS_IDRCNT_L	0x00	IDRCNT_L ICS variable initial value
	INIT_ICS_TRCNT_P	0x00	TRCNT_P ICS variable initial value
	INIT_ICS_CPSET1	0x01	CPSET1 ICS variable initial value
	INIT_ICS_CPSET2	0x02	CPSET2 ICS variable initial value
	INIT_ICS_CP_TRIM	INIT_CP_TRIM	CP_TRIM ICS variable initial value
	INIT_ICS_VREG5_TRIM	INIT_VREG5_TRIM	VREG5_TRIM ICS variable initial value

	INIT_ICS_CSAMP_TRIM	INIT_CSAMP_TRIM	CSAMP_TRIM ICS variable initial value
	INIT_ICS_ALMRAW1	0xFF	ALMRAW1 ICS variable initial value
	INIT_ICS_TOIN_MONI	0x00	TOIN_MONI ICS variable initial value
	INIT_ICS_WHO_AM_I	0x6A	WHO_AM_I ICS variable initial value
	INIT_ICS_TRIM_PT	INIT_TRIM_PT	TRIM_PT ICS variable initial value
	INIT_ICS_TRIM_EN	INIT_TRIM_EN	TRIM_EN ICS variable initial value
	INIT_ICS_BGR_TRIM	INIT_BGR_TRIM	BGR_TRIM ICS variable initial value
	INIT_ICS_BFAMP_TRIM	INIT_BFAMP_TRIM	BFAMP_TRIM ICS variable initial value
	SEQ_INIT	0	PreDriver initial sequence definition
	SEQ_CHK_SPI	1	
	SEQ_CHK_TSD_N	2	
	SEQ_SET_5VTRIM	3	
	SEQ_SET_ALMOPE1_PRM	4	
	SEQ_SET_ALMOUT1_PRM	5	
	SEQ_SET_CS_SET2_PRM	6	
	SEQ_SET_CS_SET1_PRM	7	
	SEQ_SET_SEQINIT_PRM	8	
	SEQ_SET_PS_ALL_PRM	9	
	SEQ_SET_PS_1ST_PRM	10	
	SEQ_SET_PS_2ND_PRM	11	
	SEQ_CHK_ALMRAW1	12	
	SEQ_SET_PS_3RD_PRM	13	
	SEQ_CHK_ALMSTS	14	
	SEQ_SET_MOT_EN	15	
	SEQ_END	16	
	SEQ_NUM_MAX	17	
	ERR_RCV_SEQ_INIT	0	PreDriver ALARM recovery sequence definition
	ERR_RCV_SEQ_CHK_STS	1	
	ERR_RCV_SEQ_CLR_MOT_EN	2	
	ERR_RCV_SEQ_SET_PS_1ST	3	
	ERR_RCV_SEQ_CHK_ALMSTS_1ST	4	
	ERR_RCV_SEQ_SET_ALM_LATCH_CLR	5	
	ERR_RCV_SEQ_SET_PS_2ND	6	
	ERR_RCV_SEQ_SET_PS_3RD	7	
	ERR_RCV_SEQ_CHK_ALMRAW1	8	
	ERR_RCV_SEQ_SET_PS_4TH	9	
	ERR_RCV_SEQ_CHK_ALMSTS_2ND	10	
	ERR_RCV_SEQ_SET_MOT_EN	11	
	ERR_RCV_SEQ_END	12	
	ERR_RCV_SEQ_NUM_MAX	13	

3.5 Flow chart

Figure 3-4 shows the whole flow chart and the flow chart of initialization function.

Also, flow chart of main processing in sample program are shown in Figure 3-4 to Figure 3-11.

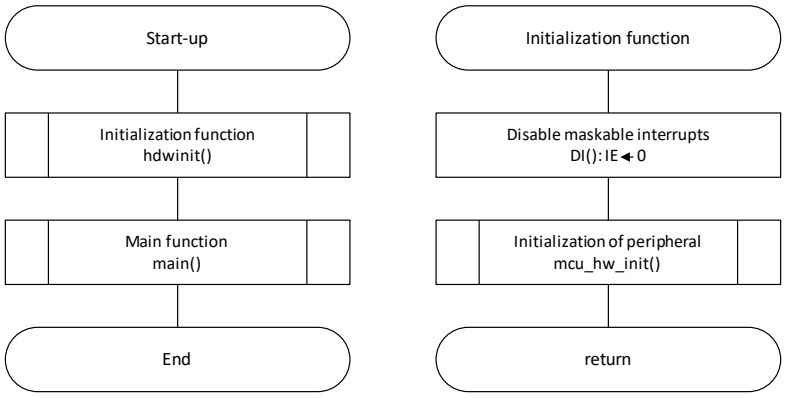

Figure 3-4 Flow chart (Overall and Initialization function)

3.5.1 **Main function**

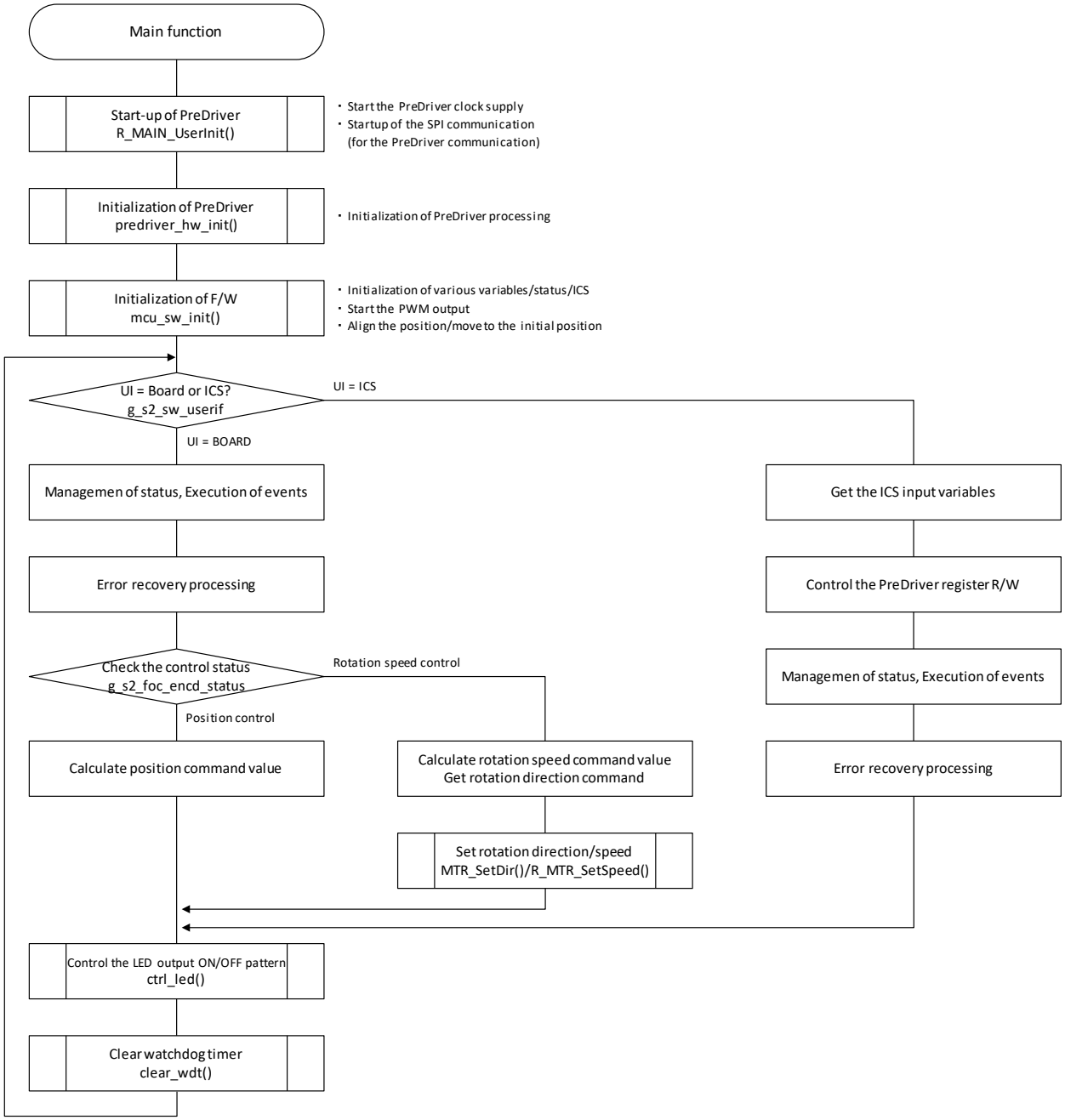

Figure 3-5 Flow chart (Main function)

3.5.2 Initialization of PreDriver processing

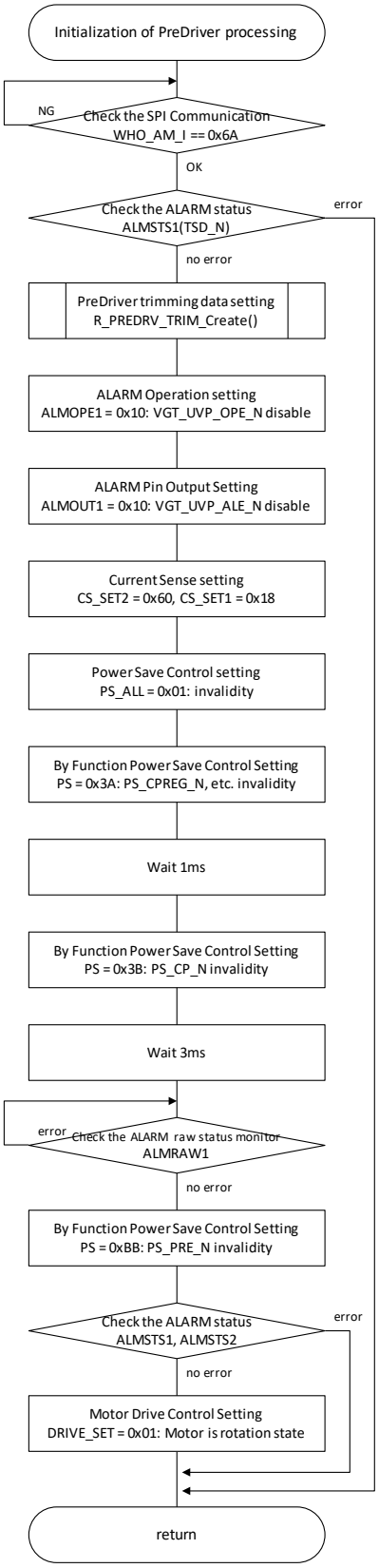

Figure 3-6 Flow chart (Initialization of PreDriver processing)

3.5.3 Carrier frequency interruption processing

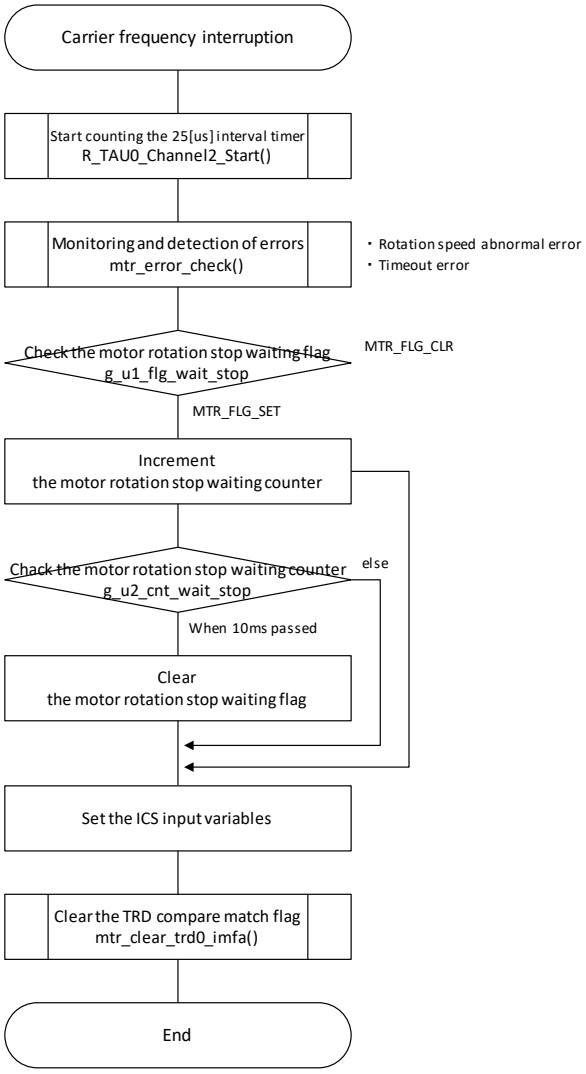

Figure 3-7 Flow chart (Carrier frequency interruption processing)

3.5.4 1[ms] interruption processing

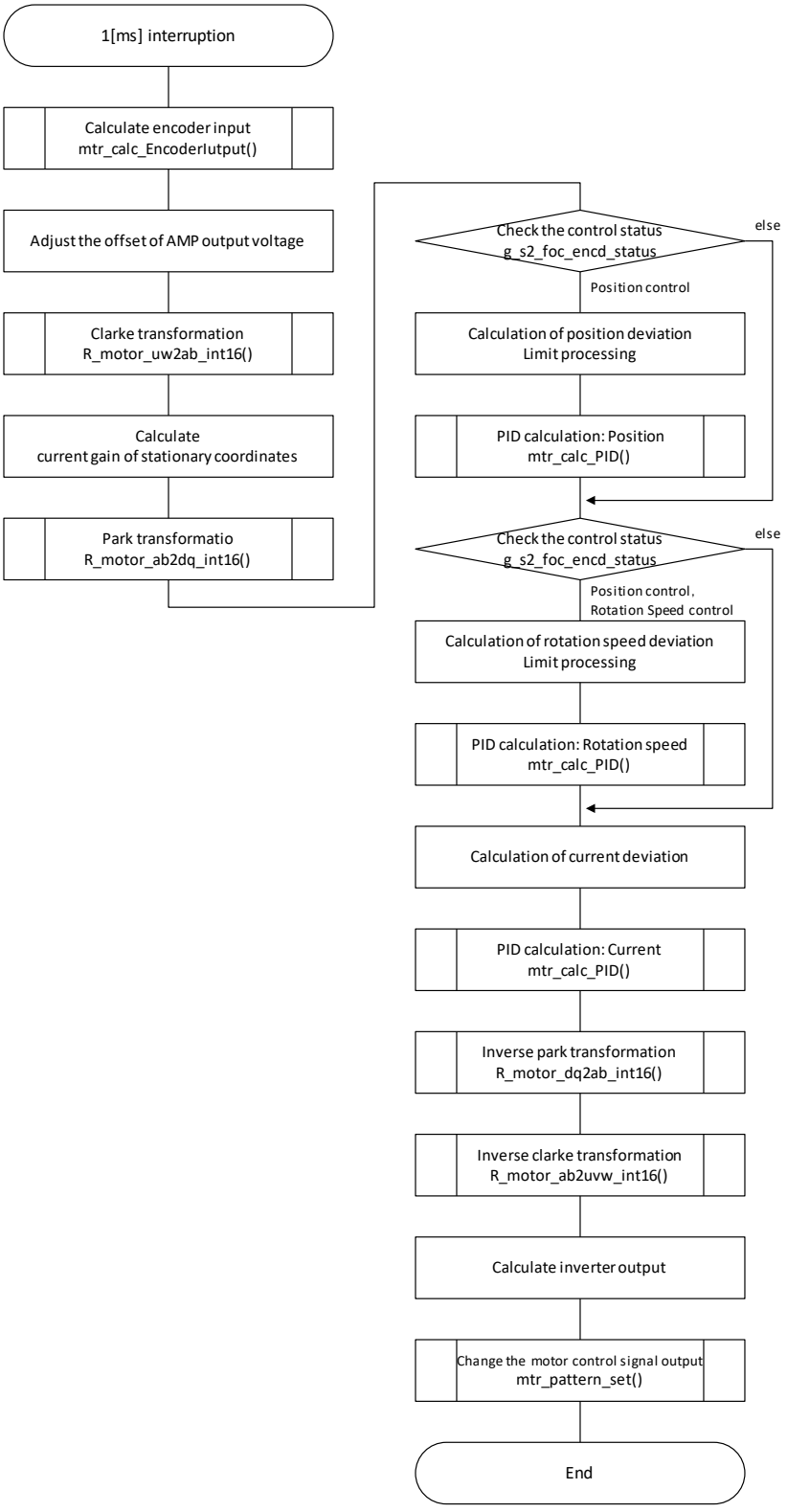

Figure 3-8 Flow chart (1[ms] interruption processing)

3.5.5 **25[us] interruption processing**

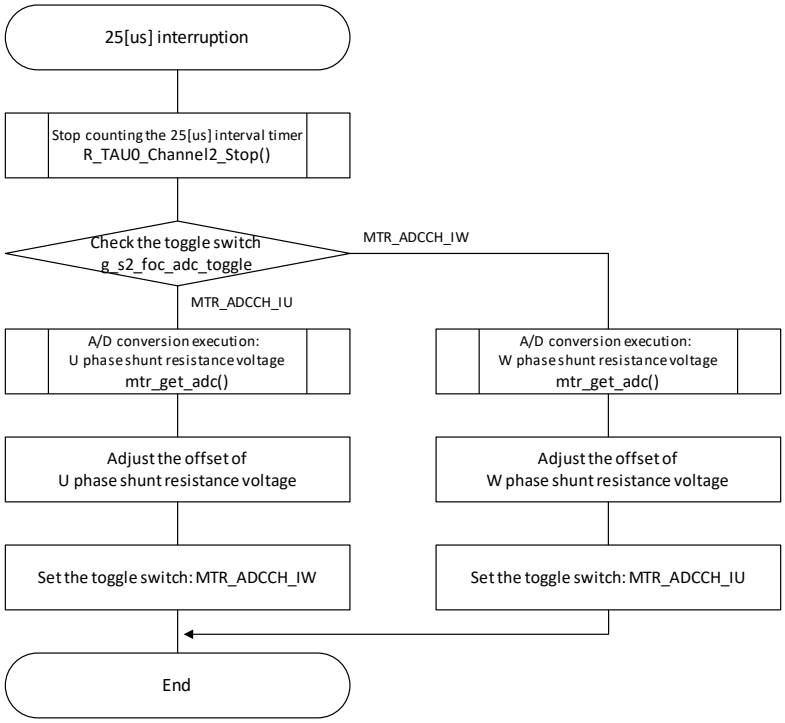

Figure 3-9 Flow chart (25[us] interruption processing)

3.5.6 ALARM interruption processing

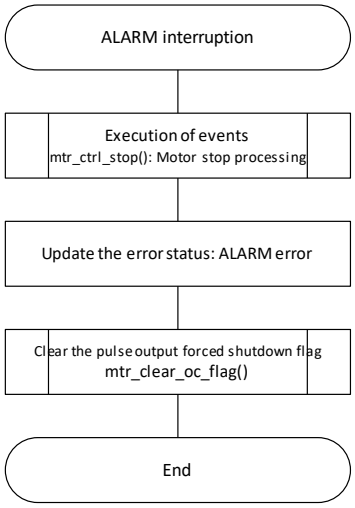

Figure 3-10 Flow chart (ALARM interruption processing)

3.5.7 ALARM recovery processing

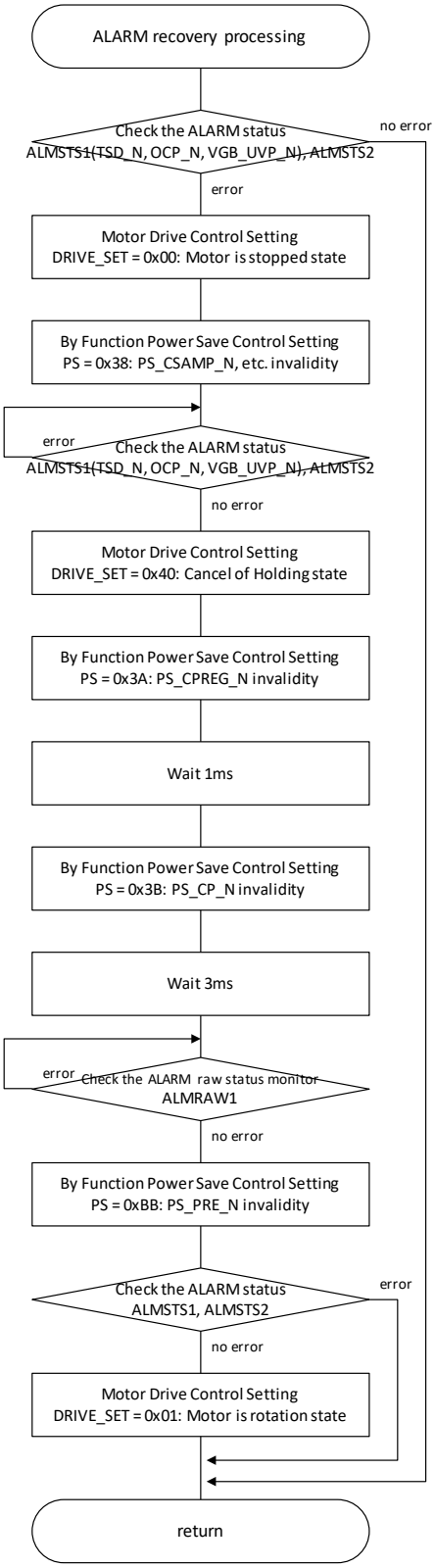

Figure 3-11 Flow chart (ALARM recovery processing)

4. Development support tool In Circuit Scope

4.1 Overview

In the target sample programs described in this application note, user interfaces (position command, rotation speed command, etc.) based on the development support tool "In Circuit Scope" (ICS) can be used. ICS is a tool which displays on PC real-time waveforms of global variables of the program being executed on the target system. Refer to "In Circuit Scope manual" for usage and more details.

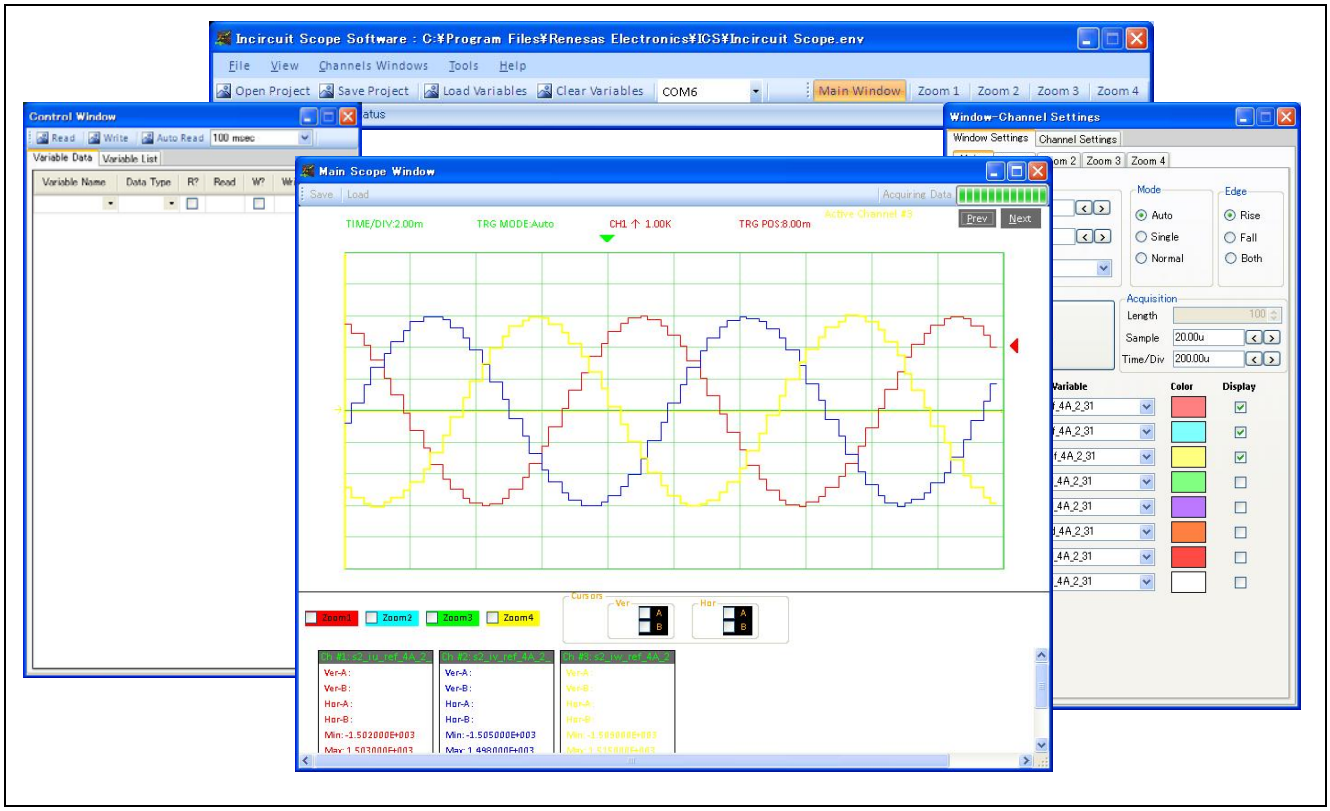

Figure 4-1 In Circuit Scope - Appearance

4.2 How to use library

In order to use ICS for the low-voltage version, it is necessary to call functions related to ICS. The ICS-related functions have been set by conditional compilation (#ifdef--#endif). To use ICS, set as follows.
[File name] mtr_common.h
[Point to change] Add the following declaration.
#define ICS_USE

4.3 List of variables for ICS

Table 4-1 and Table 4-2 are list of variables for ICS. Table 4-1 variable values are reflected to the protect variables when the same values as g_s2_enable_write are written to com_s2_enable_write.
Table 4-2 variable values do not depend on com_s2_enable_write.

Table 4-1 List of Variables for ICS

Variable bame	Type	Content	Remarks ([ ]: protect variable name)
com_s2_enable_write	int16_t	Enable to rewriting variables	-
com_s2_foc_ref_theta	int16_t	Position command value (Normalized mechanical angle) [rad]	[g_s2_foc_ref_theta]
com_s2_foc_ref_omega	int16_t	Rotation speed command value (Mechanical angle) [rad/s]	[g_s2_foc_ref_omega]

Table 4-2 List of Variables for ICS

Variable name	Type	Content	Remarks
com_s2_sw_userif	int16_t	User interface switch 0: ICS user interface use 1: Board user interface use	-
com_s2_mode_system	int16_t	State management 0: Stop mode 1: Run mode 3: Reset	-
com_u1_pdrvreg_ctrl	uint8_t	PreDriver register R/W control flag	0: R/W disable 1: R/W enable
com_u1_pdrvreg_ps_all_pre	uint8_t	PreDriver register PS_ALL previous value	Read value
com_u1_pdrvreg_ps_all_now	uint8_t	PreDriver register PS_ALL current value	Write value
com_u1_pdrvreg_ps_pre	uint8_t	PreDriver register PS previous value	Read value
com_u1_pdrvreg_ps_now	uint8_t	PreDriver register PS current value	Write value
com_u1_pdrvreg_sw_reset_pre	uint8_t	PreDriver register SW_RESET previous value	Read value
com_u1_pdrvreg_sw_reset_now	uint8_t	PreDriver register SW_RESET current value	Write value
com_u1_pdrvreg_adc_sel_pre	uint8_t	PreDriver register ADC_SEL previous value	Read value
com_u1_pdrvreg_adc_sel_now	uint8_t	PreDriver register ADC_SEL current value	Write value
com_u1_pdrvreg_selsig_u_pre	uint8_t	PreDriver register SELSIG_U previous value	Read value
com_u1_pdrvreg_selsig_u_now	uint8_t	PreDriver register SELSIG_U current value	Write value
com_u1_pdrvreg_selsig_v_pre	uint8_t	PreDriver register SELSIG_V previous value	Read value
com_u1_pdrvreg_selsig_v_now	uint8_t	PreDriver register SELSIG_V current value	Write value
com_u1_pdrvreg_selsig_w_pre	uint8_t	PreDriver register SELSIG_W previous value	Read value
com_u1_pdrvreg_selsig_w_now	uint8_t	PreDriver register SELSIG_W current value	Write value
com_u1_pdrvreg_hall_sig_pre	uint8_t	PreDriver register HALL_SIG previous value	Read value
com_u1_pdrvreg_hall_sig_now	uint8_t	PreDriver register HALL_SIG current value	Write value
com_u1_pdrvreg_almsts1_pre	uint8_t	PreDriver register ALMSTS1 previous value	Read value (ALMSTS1 Read Only)
com_u1_pdrvreg_almope1_pre	uint8_t	PreDriver register ALMOPE1 previous value	Read value

com_u1_pdrvreg_almope1_now	uint8_t	PreDriver register ALMOPE1 current value	Write value
com_u1_pdrvreg_almout1_pre	uint8_t	PreDriver register ALMOUT1 previous value	Read value
com_u1_pdrvreg_almout1_now	uint8_t	PreDriver register ALMOUT1 current value	Write value
com_u1_pdrvreg_almsts2_pre	uint8_t	PreDriver register ALMSTS2 previous value	Read value (ALMSTS2 Read Only)
com_u1_pdrvreg_cs_set2_pre	uint8_t	PreDriver register CS_SET2 previous value	Read value
com_u1_pdrvreg_cs_set2_now	uint8_t	PreDriver register CS_SET2 current value	Write value
com_u1_pdrvreg_almout2_pre	uint8_t	PreDriver register ALMOUT2 previous value	Read value
com_u1_pdrvreg_almout2_now	uint8_t	PreDriver register ALMOUT2 current value	Write value
com_u1_pdrvreg_error_wait_pre	uint8_t	PreDriver register ERROR_WAIT previous value	Read value
com_u1_pdrvreg_error_wait_now	uint8_t	PreDriver register ERROR_WAIT current value	Write value
com_u1_pdrvreg_cs_set1_pre	uint8_t	PreDriver register CS_SET1 previous value	Read value
com_u1_pdrvreg_cs_set1_now	uint8_t	PreDriver register CS_SET1 current value	Write value
com_u1_pdrvreg_haic_th_pre	uint8_t	PreDriver register HAIC_TH previous value	Read value
com_u1_pdrvreg_haic_th_now	uint8_t	PreDriver register HAIC_TH current value	Write value
com_u1_pdrvreg_pddsts_pre	uint8_t	PreDriver register PDDSTS previous value	Read value (PDDSTS Read Only)
com_u1_pdrvreg_ld_wait_pre	uint8_t	PreDriver register LD_WAIT previous value	Read value
com_u1_pdrvreg_ld_wait_now	uint8_t	PreDriver register LD_WAIT current value	Write value
com_u1_pdrvreg_drive_set_pre	uint8_t	PreDriver register DRIVE_SET previous value	Read value
com_u1_pdrvreg_drive_set_now	uint8_t	PreDriver register DRIVE_SET current value	Write value
com_u1_pdrvreg_di_time_pre	uint8_t	PreDriver register DI_TIME previous value	Read value
com_u1_pdrvreg_di_time_now	uint8_t	PreDriver register DI_TIME current value	Write value
com_u1_pdrvreg_idrcnt_h_pre	uint8_t	PreDriver register IDRCNT_H previous value	Read value
com_u1_pdrvreg_idrcnt_h_now	uint8_t	PreDriver register IDRCNT_H current value	Write value
com_u1_pdrvreg_idrcnt_l_pre	uint8_t	PreDriver register IDRCNT_L previous value	Read value
com_u1_pdrvreg_idrcnt_l_now	uint8_t	PreDriver register IDRCNT_L current value	Write value
com_u1_pdrvreg_trcnt_p_pre	uint8_t	PreDriver register TRCNT_P previous value	Read value
com_u1_pdrvreg_trcnt_p_now	uint8_t	PreDriver register TRCNT_P current value	Write value
com_u1_pdrvreg_cpset1_pre	uint8_t	PreDriver register CPSET1 previous value	Read value
com_u1_pdrvreg_cpset1_now	uint8_t	PreDriver register CPSET1 current value	Write value
com_u1_pdrvreg_cpset2_pre	uint8_t	PreDriver register CPSET2 previous value	Read value
com_u1_pdrvreg_cpset2_now	uint8_t	PreDriver register CPSET2 current value	Write value
com_u1_pdrvreg_cp_trim_pre	uint8_t	PreDriver register CP_TRIM previous value	Read value
com_u1_pdrvreg_cp_trim_now	uint8_t	PreDriver register CP_TRIM current value	Write value
com_u1_pdrvreg_vreg5_trim_pre	uint8_t	PreDriver register VREG5_TRIM previous value	Read value
com_u1_pdrvreg_vreg5_trim_now	uint8_t	PreDriver register VREG5_TRIM current value	Write value
com_u1_pdrvreg_csamp_trim_pre	uint8_t	PreDriver register CSAMP_TRIM previous value	Read value
com_u1_pdrvreg_csamp_trim_now	uint8_t	PreDriver register CSAMP_TRIM current value	Write value
com_u1_pdrvreg_almraw1_pre	uint8_t	PreDriver register ALMRAW1 previous value	Read value (ALMRAW1 Read Only)
com_u1_pdrvreg_toin_moni_pre	uint8_t	PreDriver register TOIN_MONI previous value	Read value (TOIN_MONI Read Only)
com_u1_pdrvreg_who_am_i_pre	uint8_t	PreDriver register WHO_AM_I previous value	Read value (WHO_AM_I Read Only)
com_u1_pdrvreg_test_pt_pre	uint8_t	PreDriver register TEST_PT previous value	Read value
com_u1_pdrvreg_test_pt_now	uint8_t	PreDriver register TEST_PT current value	Write value
com_u1_pdrvreg_efdsel_pre	uint8_t	PreDriver register EFDSEL previous value	Read value
com_u1_pdrvreg_efdsel_now	uint8_t	PreDriver register EFDSEL current value	Write value
com_u1_pdrvreg_efwdata1_pre	uint8_t	PreDriver register EFWDATA1 previous value	Read value
com_u1_pdrvreg_efwdata1_now	uint8_t	PreDriver register EFWDATA1 current value	Write value
com_u1_pdrvreg_efwdata2_pre	uint8_t	PreDriver register EFWDATA2 previous value	Read value
com_u1_pdrvreg_efwdata2_now	uint8_t	PreDriver register EFWDATA2 current value	Write value

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Aug.29.18	–	First edition issued.
1.01	Mar.15.19	–	RAJ306000_ENCD_FOC_CLOSED_V101 Correction of errors.

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 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. 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.4.0-1 November 2017)

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/.