Sensorless 120-degree conducting control for permanent magnetic synchronous motor (Implementation)

RX13T with “Evaluation System for BLDC Motor”

Summary

This application note aims at explaining the sample programs to drive a permanent magnetic synchronous motor in the 120-degree conducting method using the RX13T microcontroller and how to use the motor control development support tool, ‘Renesas Motor Workbench’.

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 programs have been checked by using the following device.

- RX13T (R5F513T5ADFL)

Target sample programs

The target sample programs of this application note are as follows.

- RX13T_MRSSK2_SPM_LESS_120_CSP_RV100 (IDE: CS+)
- RX13T_MRSSK2_SPM_LESS_120_E2S_RV100 (IDE: e^2 studio)
- RX13T Sensorless 120-degree conducting control sample program for Evaluation System for BLDC Motor and RX13T CPU Card

Reference

- RX13T Group User’s Manual: Hardware (R01UH0822)
- Application note: ‘120-degree conducting control of permanent magnetic synchronous motor: algorithm’ (R01AN2657)
- Renesas Motor Workbench User’s Manual (R21UZ0004)
- Evaluation System for BLDC Motor User’s Manual (R12UZ0062)
- RX13T CPU CARD User’s Manual (R12UZ0051)

Contents

1. Overview

This application note explains how to implement the 120-degree conducting control sample programs of permanent magnetic synchronous motor (PMSM) *1 using the RX13T microcontroller and how to use the motor control development support tool, 'Renesas Motor Workbench'. Note that these sample programs are based on the algorithm described in the application note '120-degree conducting control of permanent magnetic synchronous motor: algorithm'.

Note: 1. PMSM is also known as brushless DC motor (BLDC).

1.1 Development environment

Table 1-1 and Table 1-2 show development environment for the sample programs explained in this application note.

Table 1-1 Development Environment of the Sample Programs (H/W)

Microcontroller	Evaluation board	Motor
RX13T (R5F513T5ADFL)	48V 5A Inverter Board for BLDC Motor and RX13T CPU Card*1	TG-55L*2

Table 1-2 Development Environment of the Sample Programs (S/W)

CS+ version	e²studio version	Toolchain version
V8.04.00	2020-10	CC-RX: V3.02.00

For purchase and technical support contact, Sales representatives and dealers of Renesas Electronics Corporation.

Notes: 1. 48V 5A Inverter Board for BLDC Motor (RTK0EM0000B10020BJ) and RX13T CPU Card (RTK0EMXA10C00000BJ) are products of Renesas Electronics Corporation. 48V 5A Inverter Board for BLDC Motor is included in Evaluation System for BLDC Motor (RTK0EMX270S00020BJ).
2. TG-55L is a product of TSUKASA ELECTRIC.
TSUKASA ELECTRIC. (http://www.tsukasa-d.co.jp/)

2. System overview

Overview of this system is explained below.

2.1 Hardware configuration

The hardware configuration is shown below.

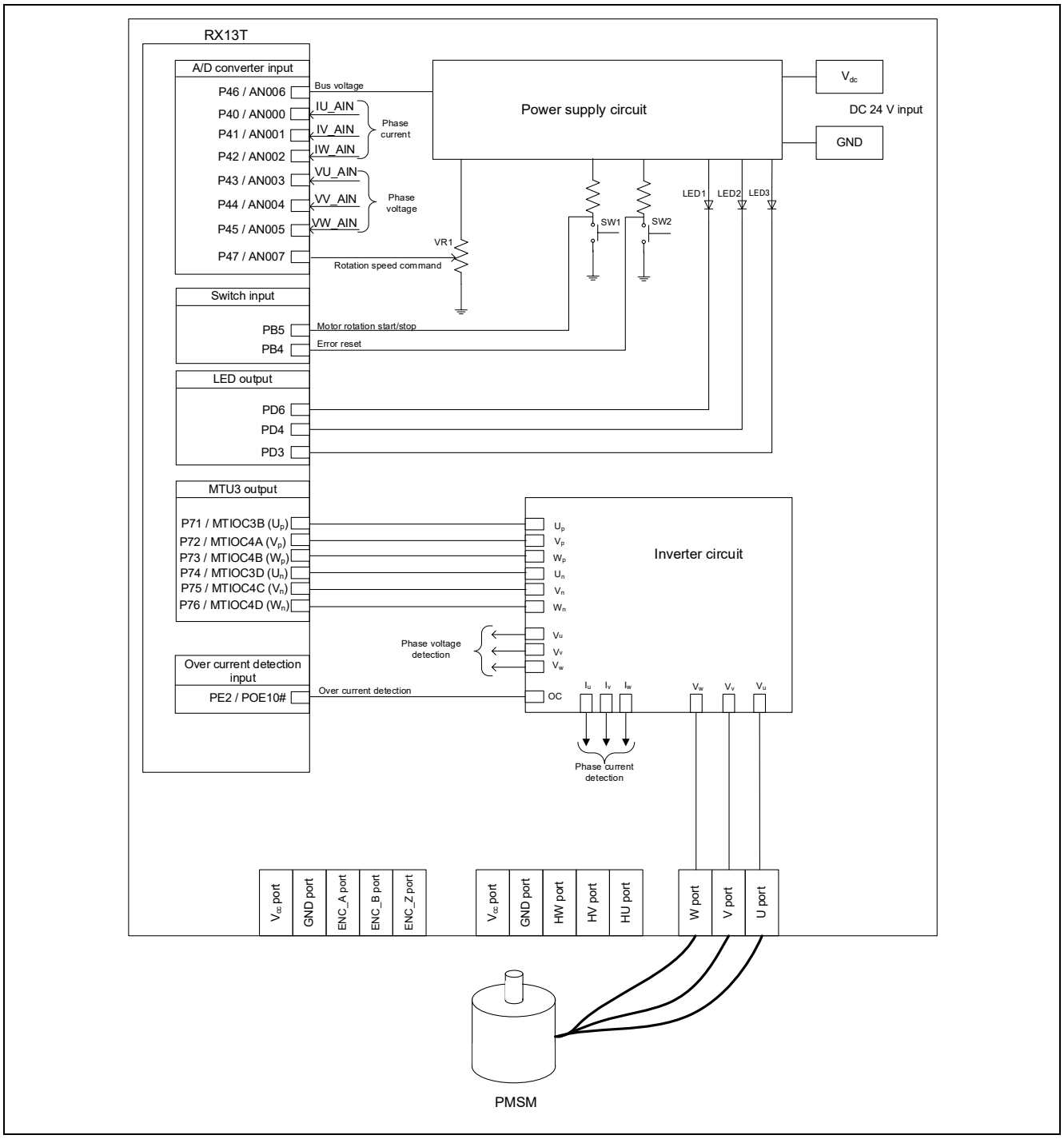


Figure 2-1 Hardware Configuration Diagram

2.2 Hardware specifications

2.2.1 User interface

Table 2-1 is a list of user interfaces for this system.

Table 2-1 User Interface

Item	Interface component	Function
Rotation speed	Variable resistance (VR1)	Rotation speed command value input (analog values)
START/STOP	Toggle switch (SW1)	Motor rotation start/stop command
ERROR RESET	Push switch (SW2)	Command of recovery from error status
LED1	Orange LED	• At the time of Motor rotation: ON • At the time of stop: OFF
LED2	Orange LED	• At the time of error detection: ON • At the time of normal operation: OFF
LED3	Orange LED	Not used
RESET	Push switch (RESET1)	System reset

Table 2-2 is a list of port interfaces in RX13T microcontroller for this system.

Table 2-2 Port Interface

R5F513T5ADFL Port name	Function
P46 / AN006	Inverter bus voltage measurement
P47 / AN007	For inputting rotation speed command values (analog values)
PB5	START/STOP toggle switch
PB4	ERROR RESET toggle switch
PD6	LED1 ON/OFF control
PD4	LED2 ON/OFF control
PD3	LED3 ON/OFF control (not used)
P40 / AN000	U phase current measurement
P41 / AN001	V phase current measurement
P42 / AN002	W phase current measurement
P43 / AN003	U phase voltage measurement
P44 / AN004	V phase voltage measurement
P45 / AN005	W phase voltage measurement
P71 / MTIOC3B	PORT output / PWM output (U_p) / Low Active
P72 / MTIOC4A	PORT output / PWM output (V_p) / Low Active
P73 / MTIOC4B	PORT output / PWM output (W_p) / Low Active
P74 / MTIOC3D	PORT output / PWM output (U_n) / High Active
P75 / MTIOC4C	PORT output / PWM output (V_n) / High Active
P76 / MTIOC4D	PORT output / PWM output (W_n) / High Active
PE2 / POE10#	PWM emergency stop input at the time of overcurrent detection

2.2.2 Peripheral functions

Table 2-3 is a list of peripheral functions used in this system.

Table 2-3 Peripheral Functions List

12-bit A/D	CMT	MTU3	POE3
• Rotation speed command value input • Inverter bus voltage measurement • Current of each phase U, V, and W • Voltage of each phase U, V, and W	• 1-ms interval timer • Free-running timer for rotation speed measurement	Complementary PWM output	Sets ports executing PWM output to high impedance state when an overcurrent is detected.

(1) 12-bit A/D converter

The rotation speed command value input, U phase voltage (V_u), V phase voltage (V_v), W phase voltage (V_w), and inverter bus voltage (V_{dc}) are measured by using the '12-bit A/D converter'.
The operation modes must be set to the 'Single scan mode' (use software trigger).
U phase current (I_u), V phase current (I_v), W phase current (I_w) are measured by using the sample-and-hold function (use hardware trigger).

(2) Compare match timer (CMT)

(a) One-ms interval timer
The channel 0 of the compare match timer (CMT) is used as 1 ms interval timer.

(b) Free-running timer for measuring speed
The channel 1 of the compare match timer is used as free-running timer for speed measurement. Note that interrupts are not used.

(3) Multi-function timer pulse unit 3 (MTU3)

The operation mode varies depending on channels. On the channels 3 and 4, output with dead time (p-side is low active, n-side is high active) is performed by using the complementary PWM mode.

(4) Port output enable (POE3)

The ports executing PWM output are set to high impedance state when an overcurrent is detected (when a falling edge of the POE10# port is detected).

2.3 Software structure

2.3.1 Software file structure

The folder and file configurations of the sample programs are given below.

Table 2-4 Folder and File Configuration of the Sample Program

RX13T_MRSSK2_SPM_LESS_120_CSP_V100 RX13T_MRSSK2_SPM_LESS_120_E2S_V100	Inc	main.h	Main function, user interface control header
		mtr_common.h	Common definition header
		mtr_ctrl_mrssk.h	Board dependent processing part header
		mtr_ctrl_RX13T.h	RX13T dependent processing part header
		mtr_spm_less_120.h	Sensorless 120-degree conducting control dependent part header
		control_parameter.h	Control characteristic dependent processing part header
		motor_parameter.h	Motor characteristic dependent processing part header
		mtr_ctrl_RX13T_mrssk.h	RX13T and board dependent processing part header
		mtr_feedback.h	Feedback control processing part header
		mtr_filter.h	Filters processing part header
		mtr_gmc.h	General motor control function part header
		mtr_driver_access.h	Driver access function on part header
	ics	ICS2_RX13T.lib	Library for GUI
		ICS2_RX13T.h	Header for GUI
	src	main.c	Main function, user interface control
		mtr_ctrl_mrssk.c	Board dependent processing part
		mtr_ctrl_RX13T.c	RX13T dependent processing part
		mtr_interrupt.c	Interrupt handler
		mtr_spm_less_120.c	Sensorless 120-degree conducting control dependent part
		mtr_ctrl_RX13T_mrssk.c	RX13T and board dependent processing part
		mtr_feedback.c	Feedback control processing
		mtr_filter.c	Filters processing
		mtr_gmc.c	General motor control function
		mtr_driver_access.c	Driver access function

2.3.2 Module configuration

Figure 2-2 and Table 2-5 show module configuration of the sample programs.

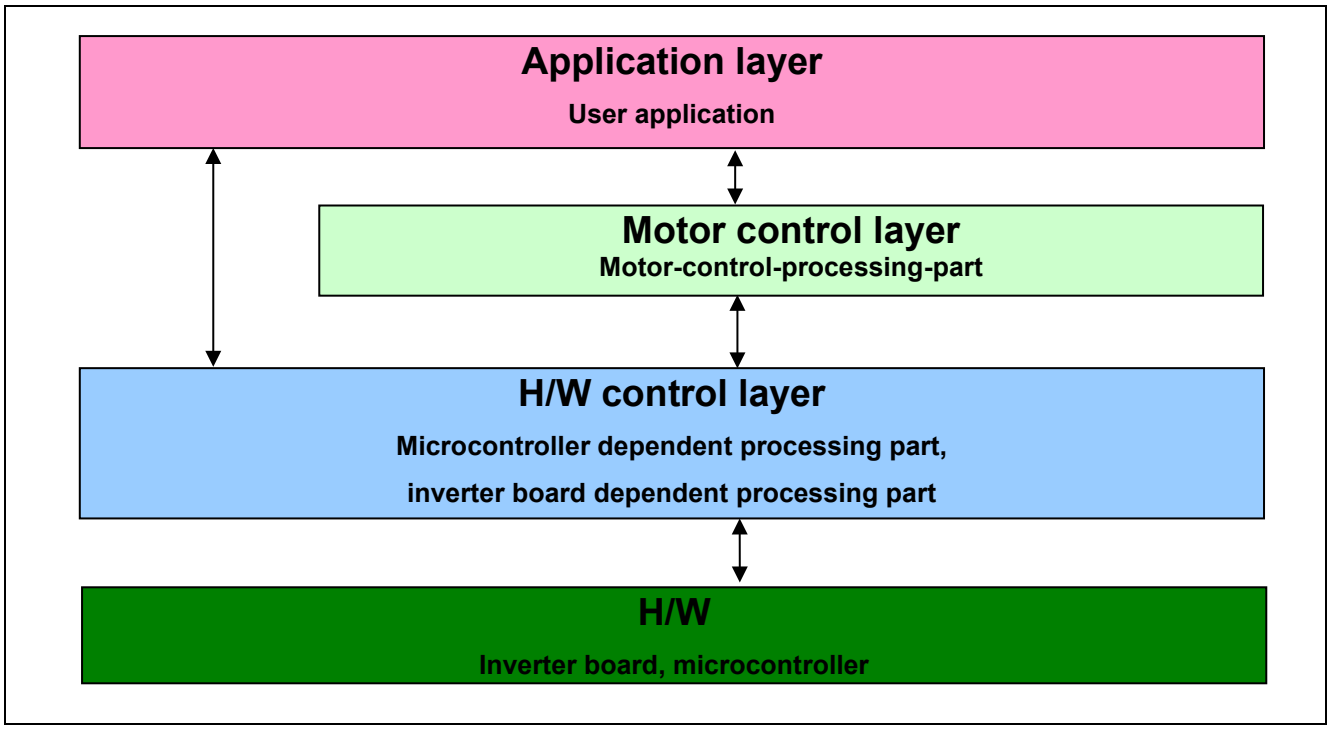


Figure 2-2 Module Configuration of the Sample Programs

Table 2-5 Module Configuration of the Sample Programs

Layers	File name
Application layer	main.c
Motor control layer	mtr_spm_less_120.c mtr_feedback.c mtr_gmc.c mtr_filter.c mtr_driver_access.c mtr_interrupt.c *1
H/W control layer	mtr_ctrl_RX13T_mrssk.c mtr_ctrl_RX13T.c mtr_ctrl_mrssk.c mtr_interrupt.c *1

Note: 1. “mtr_interrupt.c” is belong to the motor control layer and H/W control layer.

2.4 Software specifications

Table 2-6 shows the basic specifications of target software described in this application note. For details of 120-degree conducting control, refer to the application note '120-degree conducting control of permanent magnetic synchronous motor: algorithm'.

Table 2-6 Basic Specifications of Software

Item	Content
Control method	120-degree conducting method (chopping at the first 60 degrees)
Motor rotation start/stop	Determined by the level of SW1(PB5) "OFF": rotation stop "ON": rotation start or input from GUI *1
Position detection of rotor magnetic pole	Position detection by inductive voltage (by 60 degrees)
Input voltage	DC24 [V]
Carrier frequency (PWM)	20 [kHz]
Control cycle	• Zero-crossing determination is performed from inductive voltage, for each carrier cycle • PWM duty settings and conduction pattern are determined when switching patterns • Speed PI control: every 2 ms
Rotation speed control range	Both CW and CCW: 1000 [rpm] to 2650 [rpm]
SW routine for protection	• Disables the motor control signal output (six outputs), under any of the following conditions. 1. Current of each phase exceeds 0.89 [A] (monitored every 50 [µs]) 2. Inverter bus voltage exceeds 28 V (monitored per 50 [µs]) 3. Inverter bus voltage is less than 14 V (monitored per 50 [µs]) 4. Rotation speed exceeds 3000 rpm (monitored per 50 [µs]) 5. At the time of sensorless drive, zero-crossing is not detected for 50 [ms]. 6. Fault detection of virtual hall sensor pattern (position information) • The ports executing PWM output are set to high impedance state when an overcurrent is detected (when a falling edge of the POE10# port is detected).

Note: 1. For more details, refer to 4. Motor Control Development Support Tool, 'Renesas Motor Workbench'.

3. Descriptions of the control program

Following are some details regarding the SW implementation of Sensorless 120-degree conduction control.

3.1 Contents of control

3.1.1 Motor start/stop

Start and stop of the motor are controlled by two methods: input from GUI or SW1.

A general-purpose port is assigned to SW1. The port is read within the main loop. When the port is at a “ON” level, it is determined that the start switch is being pressed. Conversely, when the level is switched to “OFF”, the program determines that the motor should be stopped.

Also, an analog input port is assigned to VR1. The input is A/D converted within the main loop to generate a rotation speed command value. When the command value is less than 1000 [rpm], the program determines that the motor should be stopped.

3.1.2 A/D Converter

(1) Motor rotation speed command value

The motor rotation speed command value can be set by GUI input or potentiometer [VR1 an analog input]. The A/D converts VR1 value and use as rotation speed command value, as shown below.

Table 3-1 Conversion Ratio of the Rotation Speed Command Value

Item	Conversion ratio (Command value: A/D conversion value)		Channel
Rotation speed command value	CW	0 [rpm] to 2700 [rpm]: 07FFH to 0000H	AN007
	CCW	0 [rpm] to 2700 [rpm]: 0800H to 0FFFH	

(2) Inverter bus voltage

It is used for modulation factor calculation and over voltage detection. (When an abnormality is detected, PWM is stopped.)

Table 3-2 Inverter Bus Voltage Conversion Ratio

Item	Conversion ratio (Inverter bus voltage: A/D conversion value)	Channel
Inverter bus voltage	0 [V] to 111 [V]: 0000H to 0FFFH	AN006

(3) U phase, V phase, and W phase voltage

The U, V and W phase voltages are measured as shown in Table 3-3 and used to determine zero-crossing.

Table 3-3 Conversion Ratio of U, V, and W Phase Voltage

Item	Conversion ratio (U, V, and W phase voltage: A/D conversion value)	Channel
U, V, W phase voltage	0 [V] to 111 [V]: 0000H to 0FFFH	AN003, AN004, AN005

(4) U, V, W phase current

The U, V and W phase currents are measured as shown in Table 3-4 and used for checking the current limits in the software.

Table 3-4 Conversion Ratio of U, V and W Phase Current

Item	Conversion ratio (U, V, and W phase current: A/D conversion value)	Channel
U, V, W phase current	-12.5 [A] to 12.5 [A]: 0000H to 0FFFH *1	Iu: AN000 Iv: AN001 Iw: AN002

Note: 1. For more details of A/D conversion characteristics, refer to RX13T Group User's Manual: Hardware.

3.1.3 Speed control

In this system, the motor rotation speed is calculated from a difference of the current timer value and the timer value 2π [rad] before. The timer values are obtained when patterns are switched after zero-crossing detection, while having the timer of channel 1 of compare match timer performed free running.

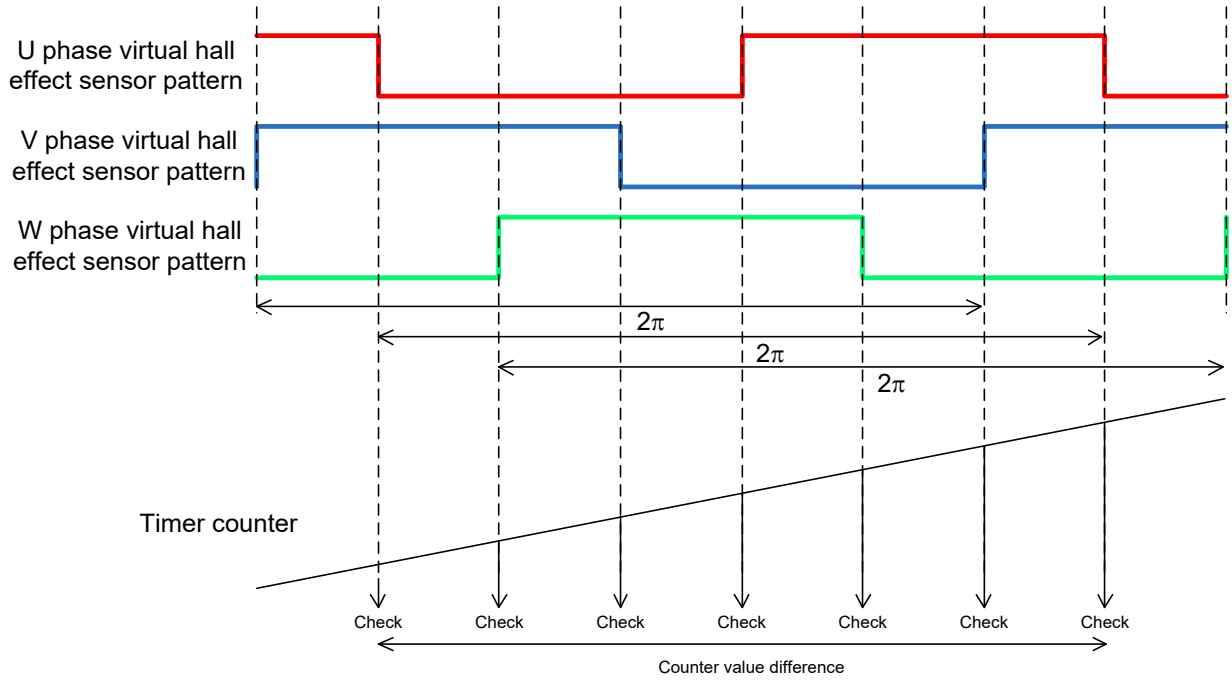


Figure 3-1 Motor Rotation Speed Calculation Method

The target sample software [explained in this application note] uses PI control for speed control. A voltage command value is calculated by the following formula of speed PI control.

$$v^* = (K_{P\omega} + \frac{K_{I\omega}}{s})(\omega^* - \omega)$$

v^*: Voltage command value, ω^*: Speed command value, ω: Rotation speed

$K_{P\omega}$: Speed PI proportional gain, $K_{I\omega}$: Speed PI integral gain, s: Laplace operator

For more details of PI control, please refer to specialized books.

3.1.4 Voltage control by PWM

PWM control is used for controlling output voltage. The PWM control is a control method that continuously adjusts the average voltage by varying the duty of pulse, as shown in Figure 3-2.

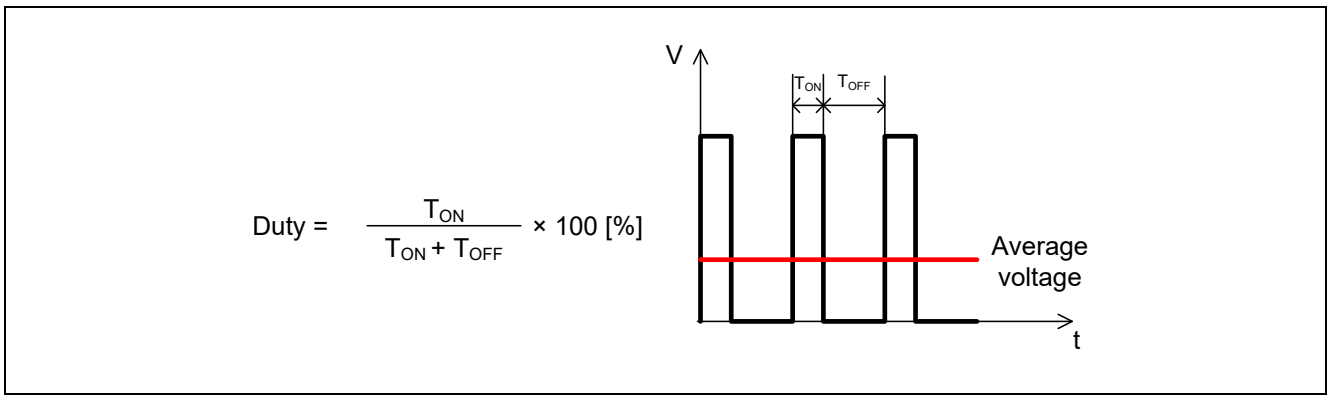


Figure 3-2 PWM Control

Here, modulation factor m is defined as follows.

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

m : Modulation factor V : Command value voltage E : Inverter bus voltage

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

In the target software of this application note, first-60-degree chopping is used to control the output voltage and speed. Figure 3-3 shows an example of motor control signal output waveforms at Non-complimentary first-60-degree Chopping. Figure 3-4 shows an example of motor control signal output waveforms at Complimentary first-60-degree Chopping.

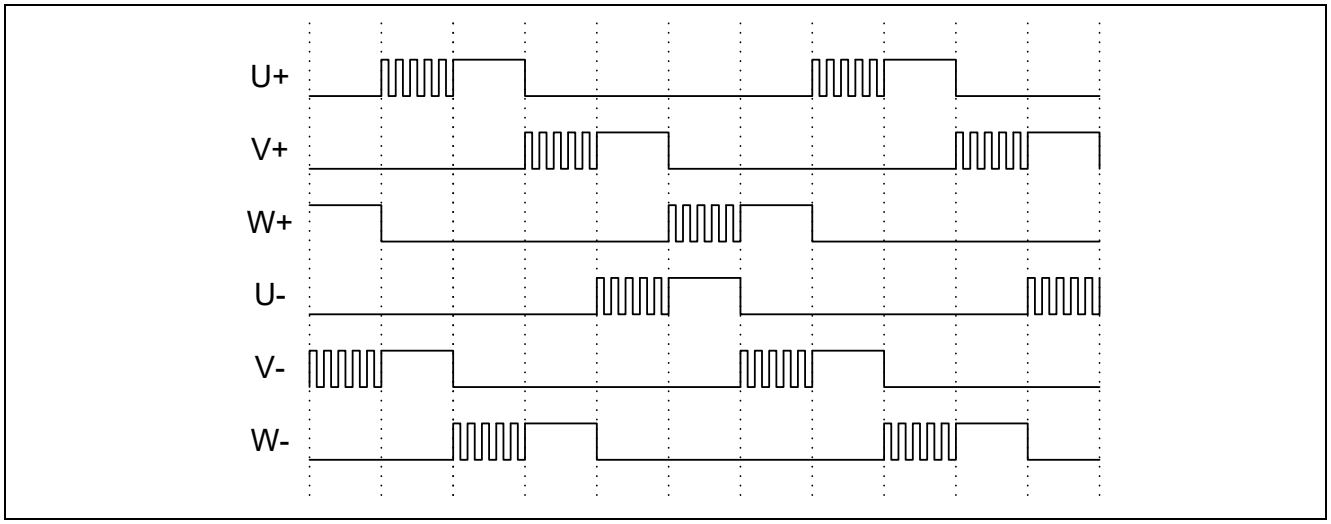


Figure 3-3 Non-complimentary first-60-degree Chopping

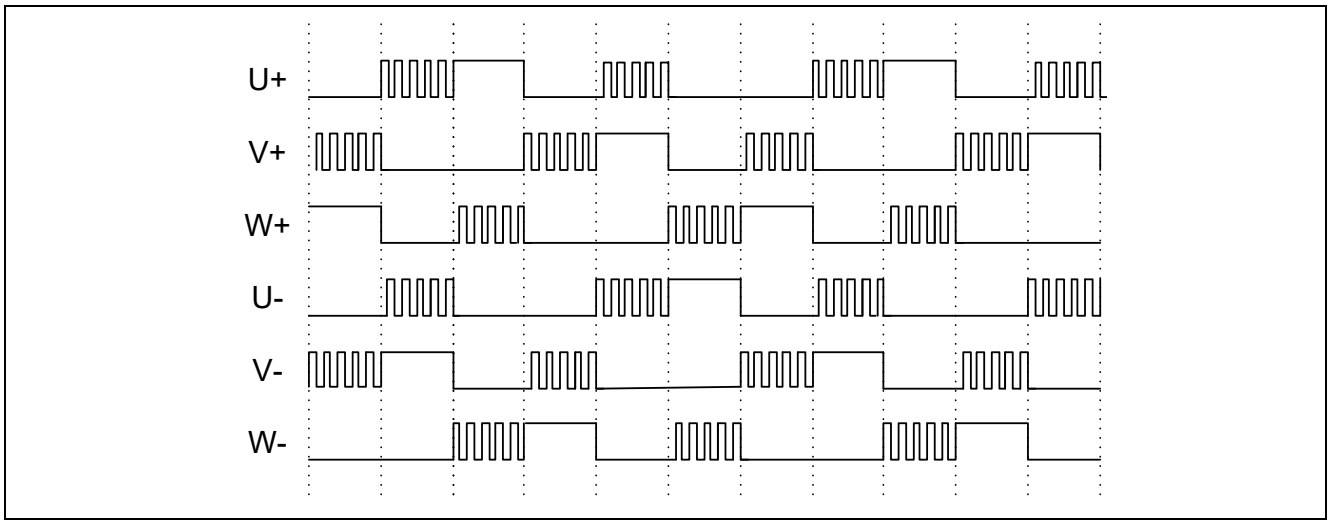
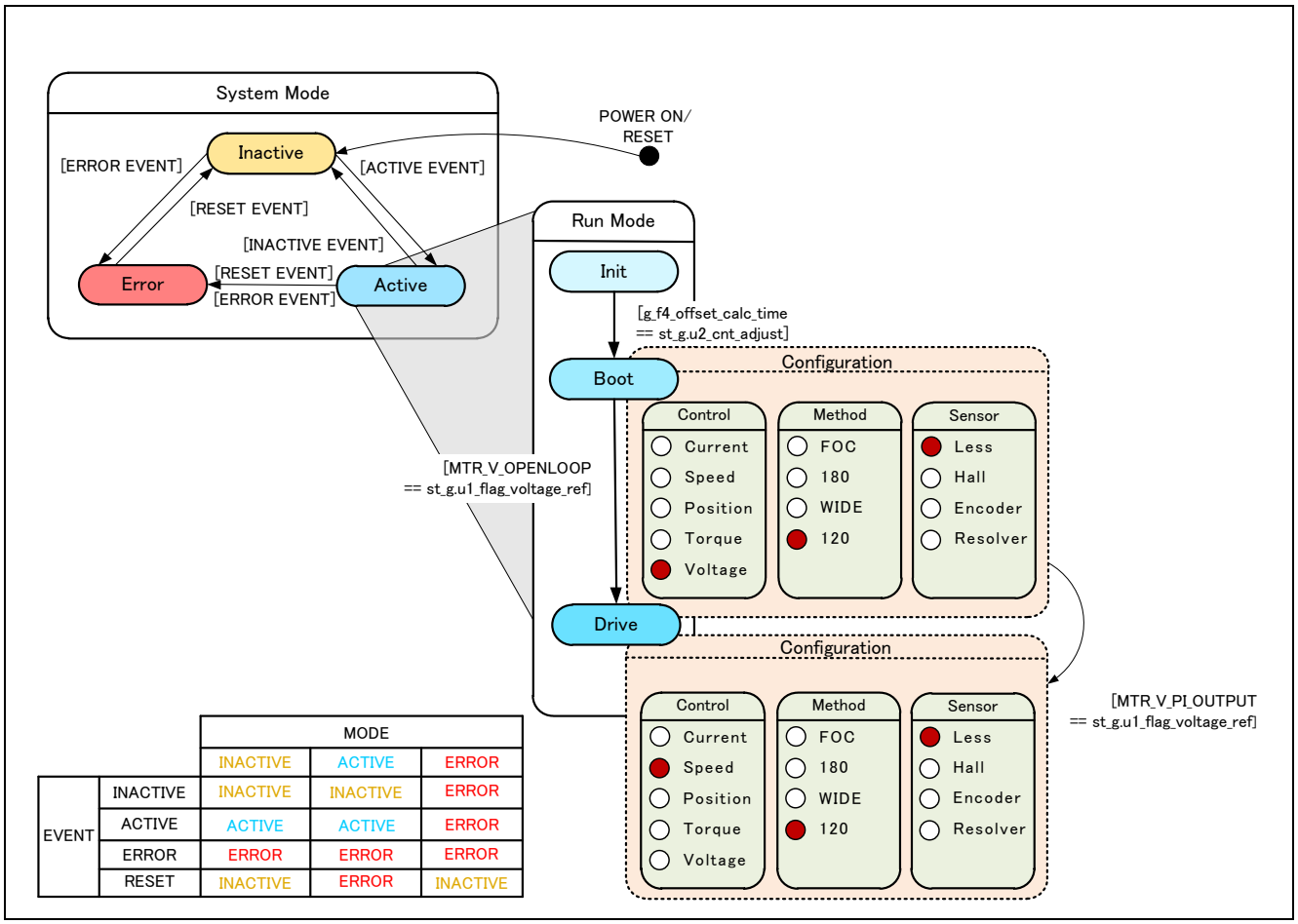


Figure 3-4 Complimentary first-60-degree Chopping

3.1.5 State transition

Figure 3-5 show state transition diagrams of 120-degree conducting control software.



		MODE		
		INACTIVE	ACTIVE	ERROR
EVENT	INACTIVE	INACTIVE	INACTIVE	ERROR
	ACTIVE	ACTIVE	ACTIVE	ERROR
	ERROR	ERROR	ERROR	ERROR
	RESET	INACTIVE	ERROR	INACTIVE

Figure 3-5 State Transition Diagram of Sensorless 120-degree Conducting Control Software

3.1.6 Start-up method in sensorless control

Sensorless 120-degree conducting control, estimates the position of the magnetic poles at every 60 degrees from the induced voltage [back EMF]. Change in the induced voltage is caused by the change in the magnetic flux of the permanent magnet (rotor).

Therefore, as a start-up method, there is a method to lead the synchronous speed by generating a rotating magnetic field by forcibly switching conduction patterns regardless of position of the permanent magnet.

Figure 3-6, shows the start-up method in the sample software. In “MTR_MODE_BOOT”, it is draw in of the magnet position. It prevents overcurrent of start-up.

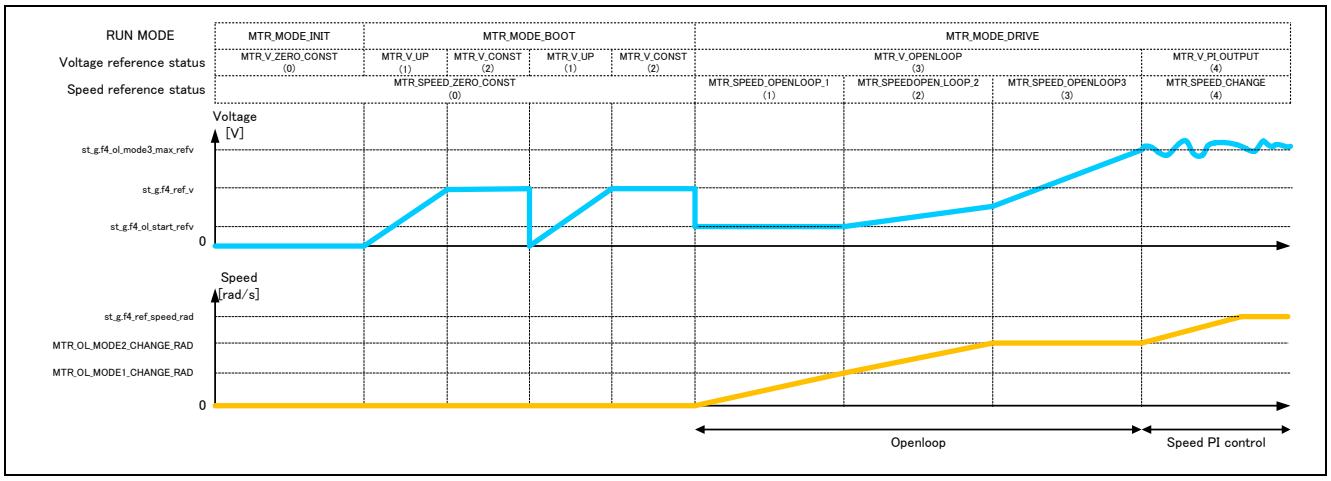


Figure 3-6 Start-up Method (Example)

3.1.7 System protection function

This system has the following six types of error status and enables emergency stop functions in case of occurrence of respective error. Refer to Table 3-5 for settings.

- Overcurrent error
  High impedance output is made to the PWM output port in response to an emergency stop signal (over current detection) from the hardware. In addition, U, V, and W phase currents are monitored in over current monitoring cycle. When an over current (when the current exceeds the over current limit value) is detected, the CPU executes emergency stop (software detection).
- Overvoltage error
  The inverter bus voltage is monitored at the overvoltage monitoring cycle. When an over voltage is detected (when the voltage exceeds the limit value), CPU performs an emergency stop. The threshold value of the overvoltage is set in consideration of the error of resistance value of the detection circuit.
- Undervoltage error
  The inverter bus voltage is monitored at the under-voltage monitoring cycle. When an under voltage is detected (when the voltage lowers the limit value), CPU performs an emergency stop. The threshold value of the overvoltage is set in consideration of the error of resistance value of the detection circuit.
- Rotation speed abnormality error
  The rotation speed is monitored at the rotation speed monitoring cycle. When the speed exceeds the limit value, CPU performs an emergency stop.
- Timeout error of zero-cross detection
  When no pattern switching by detecting a timeout zero-crossing occurs, CPU performs an emergency stop.
- Virtual hall sensor pattern (estimated from back EMF) error
  When an error is detected in virtual hall sensor patterns (estimated from back EMF) generated from each of U, V, and W phase voltage, CPU performs an emergency stop.

Table 3-5 Setting Value of Each System Protection Function

Overcurrent error	Over current limit value [A]	0.89
	Monitoring cycle [μs]	50
Overvoltage error	Overvoltage limit value [V]	28
	Monitoring cycle [us]	50
Undervoltage error	Under voltage limit value [V]	14
	Monitoring cycle [us]	50
Rotation speed abnormality error	Speed limit value [rpm]	3000
	Monitoring cycle [us]	50
Timeout error of zero-cross detection	Timeout value [ms]	50

3.2 Function specifications of 120-degree conducting control software

Multiple control functions are used in this control program.

Table 3-6 List of Functions “main.c”

File name	Function name	Process overview
main.c	main Input: None Output: None	• Hardware initialization function call • User interface initialization function call • Initialization function call of the variable used in the main process • Status transition and event execution function call • Main process ⇒ user interface call ⇒ Watchdog timer clear function call
	board_ui Input: None Output: None	Board user interface use • Motor status change • Determination of rotation speed command value
	ics_ui Input: None Output: None	GUI user interface use • Motor status change • Determination of rotation speed command value
	software_init Input: None Output: None	Initialization of variables used in the main process

Table 3-7 List of Functions “mtr_ctrl_RX13T.c”

File name	Function name	Process overview
mtr_ctrl_RX13T.c	R_MTR_InitHardware Input: None Output: None	Initialization of the clock and peripheral functions
	mtr_init_clock Input: None Output: None	Initialization of clock
	init_wdt Input: None Output: None	Initialization of the watchdog timer (WDT)
	mtr_init_cmt Input: None Output: None	Initialization of compare match timer (CMT)
	mtr_init_poe3 Input: None Output: None	Initialization of port output enable 3 (POE3)
	clear_wdt Input: None Output: None	Clearing the watchdog timer (WDT)
	mtr_clear_oc_flag Input: None Output: None	Clearing the high impedance state

Table 3-8 List of Functions “mtr_ctrl_mrssk.c”

File name	Function name	Process overview
mtr_ctrl_mrssk.c	R_MTR_ChargeCapacitor Input: None Output: None	Wait for Stability of the bus voltage
	get_vr1 Input: None Output: (uint16) ad_data / A/D conversion result	VR1 status acquisition
	get_sw1 Input: None Output: (uint8) tmp_port / SW1 level	SW1 status acquisition
	get_sw2 Input: None Output: (uint8) tmp_port / SW2 level	SW2 status acquisition
	led1_on Input: None Output: None	Turning LED1 ON
	led2_on Input: None Output: None	Turning LED2 ON
	led3_on Input: None Output: None	Turning LED3 ON
	led1_off Input: None Output: None	Turning LED1 OFF
	led2_off Input: None Output: None	Turning LED2 OFF
	led3_off Input: None Output: None	Turning LED3 OFF

Table 3-9 List of Functions “mtr_interrupt.c”

File name	Function name	Process overview
mtr_interrupt.c	mtr_over_current_interrupt Input: None Output: None	Overcurrent detection process (Hard detection) • Event processing selection function call (Generation error event) • Changing the motor status (to error mode) • High impedance state clearing function call (to PWM output disable process)
	mtr_carrier_interrupt Input: None Output: None	Calling every 50 [µs] • Current and voltage offset adjustment • Error check function call • Start control • Detection of zero-cross function call • Velocity PI control • Calculate velocity function call • Set GUI variables
	mtr_1ms_interrupt Input: None Output: None	Calling every 1 [ms] • Set reference velocity • Set reference voltage

Table 3-10 List of Functions “mtr_spm_less_120.c” (1/2)

File name	Function name	Process overview
mtr_spm_less_120.c	R_MTR_InitSequence Input: (uint8) u1_id / Motor ID Output: None	Initialization of the sequence process
	R_MTR_ExecEvent Input: (uint8) u1_event / occurred event (uint8) u1_id / Motor ID Output: None	• Changing the status • Calling an appropriate process execution function for the occurred event
	mtr_act_active Input: (uint8) u1_state / motor status (uint8) u1_id / Motor ID Output: (uint8) u1_state / motor status	PWM output enable
	mtr_act_inactive Input: (uint8) u1_state / motor status (uint8) u1_id / Motor ID Output: (uint8) u1_state / motor status	PWM output disable
	mtr_act_none Input: (uint8) u1_state / motor status (uint8) u1_id / Motor ID Output: (uint8) u1_state / motor status	No processing is performed.
	mtr_act_reset Input: (uint8) u1_state / motor status (uint8) u1_id / Motor ID Output: (uint8) u1_state / motor status	Global variable initialization
	mtr_act_error Input: (uint8) u1_state / motor status (uint8) u1_id / Motor ID Output: (uint8) u1_state / motor status	Motor control stop function call
	mtr_ol_signal_set Input: (MTR_ST_LESS_120*) st_m / structure for Motor Output: None	Set conduction pattern when openloop mode
	mtr_pattern_set Input: (MTR_ST_LESS_120*) st_m / structure for Motor (uint8) u1_signal /conduction pattern Output: None	Set conduction pattern
	mtr_speed_calc Input: (MTR_ST_LESS_120*) st_m / structure for Motor Output: None	Speed measurement calculation processing
	mtr_start_init Input: (MTR_ST_LESS_120*) st_m / structure for Motor Output: None	Initializing only the variables required for motor startup
	mtr_set_variables Input: None Output: None	Setting motor variables for control layer
	R_MTR_IcsInput Input: (MTR_ICS_INPUT*) ics_input / structure for GUI Output: None	Setting GUI input value for the buffer
	mtr_watch_variables Input: None Output: None	Setting GUI output value for the buffer

Table 3-10 List of Functions “mtr_spm_less_120.c” (2/2)

File name	Function overview	Processing overview
mtr_spm_less_120.c	mtr_error_check Input: None Output: None	Error monitoring
	mtr_wait_motorstop Input: (MTR_ST_LESS_120*) st_m / structure for Motor Output: None	Check motor stop
	mtr_set_voltage_ref Input: (MTR_ST_LESS_120*) st_m / structure for Motor Output: None	Set reference voltage
	mtr_set_speed_ref Input: (MTR_ST_LESS_120*) st_m / structure for Motor Output: None	Set reference speed
	mtr_start_openloop Input: (MTR_ST_LESS_120*) st_m / structure for Motor Output: None	Start open-loop control
	mtr_set_angle_shift Input: (MTR_ST_LESS_120*) st_m / structure for Motor Output: None	Calculate phase shift count
	mtr_check_pattern Input: (MTR_ST_LESS_120*) st_m / structure for Motor Output: None	Check zero-cross
	mtr_shift_angle Input: (MTR_ST_LESS_120*) st_m / structure for Motor Output: None	Phase shift • Set conduction pattern function-call • To Reflect the setting of conduction pattern PWM
	mtr_pattern_first60 Input: (MTR_ST_LESS_120*) st_m / structure for Motor Output: None	Set voltage pattern first 60-degree PWM
	mtr_pattern_first60_comp Input: (MTR_ST_LESS_120*) st_m / structure for Motor Output: None	Set voltage pattern complementary first 60-degree PWM

Table 3-11 List of Functions “mtr_ctrl_RX13T_mrssk.c”

File name	Function name	Process overview
mtr_ctrl_RX13T_mrssk.c	mtr_init_mtu Input: None Output: None	Initial setting of MTU3
	mtr_init_ad_converter Input: None Output: None	Initial setting of the A/D converter
	init_ui Input: None Output: None	Initialization of user interface
	mtr_ctrl_start Input: (uint8) u1_id / Motor ID Output: None	Motor startup processing
	mtr_ctrl_stop Input: (uint8) u1_id / Motor ID Output: None	Motor stop processing
	mtr_get_vdc_adc Input: (uint8) u1_id / Motor ID Output: (float32*) f4_vdc_ad / Vdc A/D conversion value	A/D conversion of inverter bus voltage
	mtr_get_vr1_adc Input: None Output: (unit16) u2_temp / VR1 A/D conversion value	Get VR1 A/D conversion value
	mtr_get_v_uvw_adc Input: (float32*) vu_ad / U phase A/D conversion value (float32*) vv_ad / V phase A/D conversion value (float32*) vw_ad / W phase A/D conversion value (uint8) u1_id / Motor ID Output: None	Ger u/v/w phase voltage
	mtr_get_current_uvw_adc Input: (float32*) iu_ad / U phase A/D conversion value (float32*) iv_ad / V phase A/D conversion value (float32*) iw_ad / W phase A/D conversion value (uint8) u1_id / Motor ID Output: None	Ger u/v/w phase current
	mtr_change_pattern Input: (uint8) pattern / Conduction pattern Output: None	Change conduction pattern

Table 3-12 List of Functions “mtr_feedback.c”

File name	Function name	Process overview
mtr_feedback.c	mtr_pi_ctrl Input: (MTR_PI_CTRL*) pi_ctrl / PI control structure Output: (float32) f4_ref / PI control output value	PI control

Table 3-13 List of Functions “mtr_filter.c”

File name	Function name	Process overview
mtr_filter.c	R_MTR_Lpff Input: (float32) f4_lpf_input / LPF input value (float32) f4_pre_lpf_output / LPF output value from the last time (float32) f4_lpf_k / LPF gain Output: (float32) f4_temp / LPF output value	LPF processing (float32)
	R_MTR_Lpf Input: (int16) s2_lpf_input / LPF input value (int16) s2_pre_lpf_output / LPF output value from the last time (int16) s2_lpf_k / LPF gain Output: (int16) s2_temp / LPF output value	LPF processing (int16)
	R_MTR_Limitf Input: (float32) f4_value / input value (float32) f4_max / maximum value (float32) f4_min / minimum value Output: (float32) f4_temp / output value	Upper and Lower limit processing (float32)
	R_MTR_Limit Input: (int16) s2_value / input value (int16) s2_max / maximum value (int16) s2_min / minimum value Output: (int16) s2_temp / output value	Upper and Lower limit processing (int16)
	R_MTR_Limitf_h Input: (float32) f4_value / input value (float32) f4_max / maximum value Output: (float32) f4_temp / output value	Upper limit processing (float32)
	R_MTR_Limit_h Input: (int16) s2_value / input value (int16) s2_max / maximum value Output: (int16) s2_temp / output value	Upper limit processing (int16)
	R_MTR_Limitf_l Input: (float32) f4_value / input value (float32) f4_min / minimum value Output: (float32) f4_temp / output value	Lower limit processing (float32)
	R_MTR_Limit_l Input: (int16) s2_value / input value (int16) s2_min / minimum value Output: (int16) s2_temp / output value	Lower limit processing (int16)
	R_MTR_Limitf_abs Input: (float32) f4_value / input value (float32) f4_limit_value / limit value Output: (float32) f4_temp / output value	absolute limit processing (float32)
	R_MTR_Limit_abs Input: (int16) s2_value / input value (int16) s2_limit_value / limit value Output: (int16) s2_temp / output value	absolute limit processing (int16)

Table 3-14 List of "mtr_gmc.c"Functions

File name	Function name	Process overview
mtr_gmc.c *1	mtr_get_vdc Input: (uint8) u1_id / Motor ID Output: (float32) f4_temp_vdc / Vdc value	Obtaining the bus voltage
	mtr_check_over_voltage_error Input: (float32) f4_vdc / vdc value (float32) f4_overvoltage_limit / over voltage limit value Output: (uint16) u2_temp0 / over voltage error flag	Over voltage error check
	mtr_check_under_voltage_error Input: (float32) f4_vdc/ Vdc value (float32) f4_undervoltage_limit / under voltage limit value Output: (uint16) u2_temp0 / under voltage error flag	Under voltage error check
	mtr_check_over_speed_error Input: (float32) f4_speed_rad / motor angle (float32) f4_speed_limit_rad / speed limit value Output: (uint16) u2_temp0 / over speed error flag	Over speed error check
	mtr_check_over_current_error Input: (float32) f4_iu / U phase current value (float32) f4_iv / V phase current value (float32) f4_iw / W phase current value (float32) f4_overcurrent_limit / over current limit value Output: (uint16) u2_temp0 / over current error flag	Over current error check
	mtr_get_duty Input: (float32) f4_v_ref / Reference voltage (float32) f4_vdc_ad / Bus voltage A/D conversion Value Output: (int16) s2_temp / Rate of PWM duty	Calculate
	mtr_get_v_uvw Input: (volatile float32*) vu_ad / U phase voltage A/D conversion value (volatile float32*) vv_ad / V phase voltage A/D conversion value (volatile float32*) vw_ad / W phase voltage A/D conversion value (uint8) u1_id / Motor ID Output: None	Obtaining the UVW phase voltage
	mtr_get_current_uvw Input: (volatile float32*) iu_ad / U phase current A/D conversion value (volatile float32*) iv_ad / V phase current A/D conversion value (volatile float32*) iw_ad / W phase current A/D conversion value (uint8) u1_id / Motor ID Output: None	Obtaining the UVW phase current
	mtr_generate_pattern Input: (float32) vu_ad / U phase voltage A/D conversion value (float32) vv_ad / V phase voltage A/D conversion value (float32) vw_ad / W phase voltage A/D conversion value (float32) vn_ad / 3 phase average A/D conversion value Output: (uint8) u1_temp / Virtual hall sensor value	Generate virtual hall sensor value
	mtr_check_timeout_error Input: (float32) f4_cnt_timeout / counter of timeout calculation (float32) f4_timeout_limit / Timeout limit Output: (uint8) u1_temp0 / Flag of Timeout error	Checking time-out error

Note: 1. Undescribed function is not used in this system.

Table 3-15 List of Functions “mtr_driver_access.c”

File name	Function name	Process overview
mtr_driver_access.c	R_MTR_SetSpeed Input: (int16) ref_speed / speed command value (uint8) u1_id / Motor ID Output: None	Setting the speed command value
	R_MTR_SetDir Input: (uint8) dir/ rotation direction (uint8) u1_id / Motor ID Output: None	Setting the rotation direction
	R_MTR_GetSpeed Input: (uint8) u1_id / Motor ID Output: (int16) s2_speed_rpm / speed	Obtaining the speed calculation value
	R_MTR_GetDir Input: (uint8) u1_id / Motor ID Output: (uint8) u1_direction / rotation direction	Obtaining the rotation direction
	R_MTR_GetStatus Input: (uint8) u1_id / Motor ID Output: (uint8) u1_mode_system / motor status	Obtaining the motor status

3.3 List of variables of 120-degree conducting control software

Lists of variables used in this control program are given below. However, note that the local variables are not mentioned.

Table 3-16 List of variables

Variable name	Type	Content	Remarks
g_s2_max_speed	int16	Rotation speed command maximum value	Mechanical angle [rpm]
g_s2_min_speed	int16	Rotation speed command minimum value	Mechanical angle [rpm]
g_s2_margin_min_speed	int16	Rotation speed command minimum value for motor stop	Mechanical angle [rpm]
g_s2_ref_speed	int16	User setting rotation speed	Mechanical angle [rpm]
g_u1_rot_dir	uint8	User setting rotation direction	0: CW 1: CCW
g_u1_motor_status	uint8	User motor status management	0: Stop 1: Rotating 2: Error
g_u1_reset_req	uint8	Reset request flag	0: Turning SW2 ON in error status 1: Turning SW2 OFF in error status
g_u1_sw1_cnt	uint8	SW1 determination counter	Chattering removal
g_u1_sw2_cnt	uint8	SW2 determination counter	Chattering removal
g_u1_stop_req	uint8	VR1 stop command flag	
g_s2_sw_ui	int16	User interface switch	0: GUI user interface use (default) 1: Board user interface use
g_s2_mode_system	int16	System mode	
g_s2_enable_write	int16	GUI write enable flag	
st_ics_input	MTR_ICS_INPUT	GUI input structure	
g_u1_cnt_ics	uint8	GUI decimation counter	
g_u1_enable_write	uint8	Variable for GUI	
st_ics_input_buff	MTR_ICS_INPUT	Buffer of GUI input structure	
st_g	MTR_ST_LESS_120	Less120-degree control structure	

3.4 List of sensorless 120-degree conducting control software structures

Lists of structures used in this control program are given below.

Table 3-17 List of variables (1/3)

	Variable name	Type	Content	Remarks
MTR_ST_LESS_120	u1_mode_system	uint8	State management	0x00: Inactive mode 0x01: Active mode 0x02: Error mode
	u2_run_mode	unit16	Operation mode management	0x00: Init mode 0x01: Boot mode 0x02: Drive mode 0x03: Analysis mode 0x04: Tune mode
	u1_error_status	unit8	Error status management	0x00: None error 0x01: Over current error 0x02: Over voltage error 0x04: Rotation speed error 0x08: Hall time out error 0x10: BEMF time out error 0x20: Hall pattern error 0x40: BEMFpattern error 0x80: Under voltage error 0xFF: Undefined error
	u2_sensor_conf	uint16	Sensor configuration management	0x01: Sensorless 0x02: Hall sensor 0x04: Encoder 0x08: Resolver
	u2_method_conf	uint16	Method configuration management	0x00: FOC (Fields Oriented Control) 0x01: 180 degree control 0x02: Wide angle electricity control 0x03: 120 degree control
	u2_ctrl_conf	uint16	Control configuration management	0x01: Current control 0x02: Speed control 0x04: Position control 0x08: Torque control 0x10: Voltage control
	st_motor	MTR_ PARAMETER	Motor parameter structure	
	f4_rpm_rad	float32	[rpm]→[rad/s]	2π/60*(POLE PAIRS)
	f4_vdc_ad	float32	Inverter bus voltage A/D value	[V]
	f4_v_ref	float32	Voltage command value	Speed PI control output value [V]
	s2_pwm_duty	int16	PWM duty	
	f4_ref_speed_rad	float32	Speed command value	Electrical angle [rad/s]
	f4_ref_speed_rad_crtl	float32	Speed command value	Electrical angle [rad/s]
	f4_speed_rad	float32	Speed calculation value	Electrical angle [rad/s]
	f4_kp_speed	float32	Speed PI control proportional gain	
	f4_ki_speed	float32	Speed PI control integral gain	

Table 3-17 List of variables (2/3)

	Variable name	Type	Content	Remarks
MTR_ST_LESS_120	u1_cnt_speed_pi	uint8	Speed PI control function call interval counter	
	f4_speed_lpf_k	float32	Speed LPF parameter	
	f4_limit_speed_change	float32	Speed command maximum increase limit	[rad/s]
	u1_flg_wait_stop	uint8	Motor rotation stop waiting flag	
	f4_ilim_v	float32	Speed PI control integral limit value	[V]
	f4_vu_ad	float32	U phase voltage A/D value	[V]
	f4_vv_ad	float32	V phase voltage A/D value	[V]
	f4_vw_ad	float32	W phase voltage A/D value	[V]
	f4_vn_ad	float32	Three-phase voltage average A/D value	[V]
	f4_offset_vu	float32	U phase voltage offset value	[V]
	f4_offset_vv	float32	V phase voltage offset value	[V]
	f4_offset_vw	float32	W phase voltage offset value	[V]
	f4_offset_off_vu	float32	U phase voltage offset value for all phase output disable	[V]
	f4_offset_off_vv	float32	V phase voltage offset value for all phase output disable	[V]
	f4_offset_off_vw	float32	W phase voltage offset value for all phase output disable	[V]
	f4_sum_vu_ad	float32	U phase voltage sum of value	[V]
	f4_sum_vv_ad	float32	V phase voltage sum of value	[V]
	f4_sum_vw_ad	float32	W phase voltage sum of value	[V]
	f4_iu_ad	float32	U phase current A/D value	[A]
	f4_iv_ad	float32	V phase current A/D value	[A]
	f4_iw_ad	float32	W phase current A/D value	[A]
	f4_offset_iu	float32	U phase current offset value	[A]
	f4_offset_iv	float32	V phase current offset value	[A]
	f4_offset_iw	float32	W phase current offset value	[A]
	f4_sum_iu_ad	float32	U phase current sum of value	[A]
	f4_sum_iv_ad	float32	V phase current sum of value	[A]
	f4_sum_iw_ad	float32	W phase current sum of value	[A]
	u2_offset_calc_time	uint16	Calculation time for current offset	Setting parameter * 50μ [s]
	u1_flag_offset_calc	unit8	Current offset value calculation flag	0: Calculation in transition to the boot mode 1: Calculation in transition to the boot mode (first time only)
	u2_cnt_adjust	uint16	offset value calculation flag	
	f4_boot_ref_v	float32	voltage command value	[V]
	u2_v_up_time	uint16	voltage command value addition time	
	f4_v_up_step	float32	voltage command value addition value	
	u2_v_const_time	uint16	voltage command value constant value	
	u2_cnt_adj_v	uint16	Counter to calculate constant voltage	
	u1_flag_draw_in	uint8	Draw in flag	
	u1_v_pattern	uint8	Conduction pattern	
	u1_v_pattern_num	uint8	Conduction pattern command number	
	u1_bemf_signal	uint8	Pattern created from inductive voltage	

Table 3-17 List of variables (3/3)

	Variable name	Type	Content	Remark
MTR_ST_LESS_120	u1_pre_bemf_signal	uint8	Pattern created from the previous inductive voltage	
	u1_flag_pattern_change	uint8	Zero-cross detection flag	
	u1_flag_speed_ref	uint8	Speed state management flag	0: Zero Speed 1: Open-loop 1 2: Open-loop 2 3: Open-loop 3 4: Variable speed
	u1_flag_voltage_ref	uint8	Voltage state management	0: Zero voltage 1: Increase voltage 2: voltage constant 3: Open loop 4: speed PI output
	u1_direction	uint8	Rotation direction	0: CW 1: CCW
	s2_ol_start_rad	int16	Open loop starting rpm	[rad]
	s2_ol_mode1_change_rad	int16	Open loop mode1 change speed	[rad]
	s2_ol_mode2_change_rad	int16	Open loop mode2 change speed	[rad]
	f4_ol_start_refv	float32	Open loop start reference voltage	[V]
	f4_ol_mode1_rate_rad	float32	Open loop mode1 rate of reference speed	[rad/control period]
	f4_ol_mode2_rate_refv	float32	Open loop mode2 rate of reference voltage	[V/control period]
	f4_ol_mode2_rate_rad	float32	Open loop mode2 rate of reference speed	[rad/control period]
	f4_ol_mode3_rate_refv	float32	Open loop mode3 rate of reference voltage	[V/control period]
	f4_ol_mode3_max_refv	float32	Open loop mode3 rate of reference voltage	[V]
	u1_v_pattern_open	uint8	Conduction pattern for open loop	
	u1_ol_signal	uint8	Open loop conduction pattern	
	u2_ol_pattern_set	uint16	Open loop cycle	
	u2_cnt_ol_pattern_set	uint16	Open loop conduction pattern switch counter	
	u2_cnt_timeout	uint16	Stop determination time measurement counter	Cleared when the conduction pattern is switched.
	u2_bemf_timer_cnt	uint16	Free run timer count value	
	u2_pre_bemf_timer_cnt	uint16	Previous free run timer count value	
	s4_timer_cnt_ave	int32	Average of speed measurement timer count	
	u2_timer_cnt_buf	uint16	Speed measurement timer count buffer	
	u2_timer_cnt_num	uint16	Speed measurement timer count buffer number	
	u2_cnt_carrier	uint16	Carrier cycle interruption counter	
	u2_pre_cnt_carrier	uint16	Previous carrier interruption counter value	
	u2_angle_shift_cnt	uint16	Pattern switching timing command value	
	s2_angle_shift_adjust	int16	Pattern switching timing adjustment value	
	st_speed	MTR_PI_CTRL	Structure for speed PI control	

Table 3-18 List of structures

	Member	Type	Content	Remarks
MTR_PARAMETER	u2_mtr_p	uint16	Number of pole pairs	
	f4_mtr_r	float32	Resistance	[Ω]
	f4_mtr_ld	float32	d-axis inductance	[H]
	f4_mtr_lq	float32	q-axis inductance	[H]
	f4_mtr_m	float32	Permanent magnetic flux	[Wb]
MTR_PI_CTRL	f4_err	float32	Error	
	f4_kp	float32	PI control proportional gain	
	f4_ki	float32	PI control integral gain	
	f4_refi	float32	Integral output value	
	f4_ilimit	float32	Integral output limit value	
MTR_ICS_INPUT	u2_mtr_p	uint16	Number of pole pairs	
	s2_ref_speed	int16	Reference speed	Mechanical angle [rpm]
	s2_direction	int16	Rotation direction	0: CW 1: CCW
	f4_kp_speed	float32	Speed PI control proportional gain	
	f4_ki_speed	float32	Speed PI control Integral gain	
	f4_speed_lpf_k	float32	Speed LPF parameter	
	f4_limit_speed_change	float32	Speed command maximum increase limit	[rad/s]
	s2_ol_start_rpm	int16	Open loop starting rpm	[rpm]
	s2_ol_mode1_change_rpm	int16	Open loop mode1 change speed	[rpm]
	s2_ol_mode2_change_rpm	int16	Open loop mode2 change speed	[rpm]
	f4_ol_start_refv	float32	Open loop start reference voltage	[V]
	f4_ol_mode1_rate_rpm	float32	Open loop mode1 rate of reference speed	[rpm/control period]
	f4_ol_mode2_rate_refv	float32	Open loop mode2 rate of reference voltage	[V/control period]
	f4_ol_mode2_rate_rpm	float32	Open loop mode2 rate of reference speed	[rpm/control period]
	f4_ol_mode3_rate_refv	float32	Open loop mode3 rate of reference voltage	[V/control period]
	f4_ol_mode3_max_refv	float32	Open loop mode3 rate of reference voltage	[V]
	u2_offset_calc_time	uint16	Calculation time for current offset	
	f4_boot_ref_v	float32	Voltage command value	
	u2_v_up_time	uint16	Voltage command value addition time	
	u2_v_const_time	uint16	Voltage command value constant time	

3.5 Macro definitions of 120-degree conducting control software

Lists of macro definitions used in this control program are given below.

Table 3-19 List of Macro definitions “motor_parameter.h”

File name	Macro name	Definition value	Remarks
motor_parameter.h	MP_POLE_PAIRS	2	Number of pole pairs
	MP_MAGNETIC_FLUX	0.02159f	Flux [Wb]
	MP_RESISTANCE	6.447f	Resistance [Ω]
	MP_D_INDUCTANCE	0.0045f	d-axis Inductance [H]
	MP_Q_INDUCTANCE	0.0045f	q-axis Inductance [H]
	MP_NOMINAL_CURRENT_RMS	0.42f	Nominal current [Arms]

Table 3-20 List of Macro definitions “control_parameter.h”

File name	Macro name	Definition value	Remarks
control_parameter.h	CP_OFFSET_CALC_TIME	20000	LPF parameter of current offset value
	CP_REF_V	3.0f	voltage command value
	CP_V_UP_TIME	128	voltage command value addition time
	CP_V_CONST_TIME	64	voltage command value constant time
	CP_MAX_SPEED_RPM	2650	Rotation speed command maximum value (mechanical angle) [rpm]
	CP_MIN_SPEED_RPM	1000	Rotation speed command minimum value (mechanical angle) [rpm]
	CP_LIMIT_SPEED_CHANGE	0.2f	Speed command maximum increase limit [rad/s]
	CP_OL_START_RPM	150	Open loop starting rpm [rpm]
	CP_OL_MODE1_CHANGE_RPM	185	Open loop mode1 change speed [rpm]
	CP_OL_MODE2_CHANGE_RPM	1000	Open loop mode2 change speed [rpm]
	CP_OL_START_REFV	3.0f	Open loop start reference voltage [V]
	CP_OL_MODE1_RATE_RPM	0.25f	Open loop mode1 rate of reference speed [rpm/control period]
	CP_OL_MODE2_RATE_REFV	0.00285f	Open loop mode2 rate of reference voltage [V/control period]
	CP_OL_MODE2_RATE_RPM	0.71f	Open loop mode2 rate of reference speed [rpm/control period]
	CP_OL_MODE3_RATE_REFV	0.002f	Open loop mode3 rate of reference voltage [V/control period]
	CP_OL_MODE3_MAX_REFV	6.5f	Open loop mode3 rate of reference voltage [V/control period]
	CP_SPEED_PI_KP	0.02f	Proportional gain
	CP_SPEED_PI_KI	0.004f	Integral gain
	CP_SPEED_LPF_K	1.0f	Speed LPF parameter
	MTR_FIRST60	1	Non-Complementary First 60-degree PWM
	MTR_FIRST60_COMP	0	Complementary First 60-degree PWM

Table 3-21 List of Macro definitions “main.h”

File name	Macro name	Definition value	Remarks
main.h	ICS_UI	0	GUI user interface use
	BOARD_UI	1	Board user interface use
	M_CW	0	Rotation direction
	M_CCW	1	
	OFFSET_CALC_TIME	CP_OFFSET_CALC_TIME	Calculation time for current offset
	BOOT_REF_V	CP_BOOT_REF_V	voltage command value
	V_UP_TIME	CP_V_UP_TIME	voltage command value addition time
	V_CONST_TIME	CP_V_CONST_TIME	voltage command value constant time
	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 [rpm]
	MARGIN_SPEED	50.0f	Rotation speed command minimum value creation constants for stop [rpm]
	MARGIN_MIN_SPEED	MIN_SPEED - MARGIN_SPEED	Rotation speed command minimum value for motor stop [rpm]
	OL_START_RPM	CP_OL_START_RPM	Open loop start reference voltage
	OL_MODE1_CHANGE_RPM	CP_OL_MODE1_CHANGE_RPM	Open loop mode1 change speed
	OL_MODE2_CHANGE_RPM	CP_OL_MODE2_CHANGE_RPM	Open loop mode2 change speed
	OL_START_REFV	CP_OL_START_REFV	Open loop start reference voltage
	OL_MODE1_RATE_RPM	CP_OL_MODE1_RATE_RPM	Open loop mode1 rate of reference speed
	OL_MODE2_RATE_REFV	CP_OL_MODE2_RATE_REFV	Open loop mode2 rate of reference voltage
	OL_MODE2_RATE_RPM	CP_OL_MODE2_RATE_RPM	Open loop mode2 rate of reference speed
	OL_MODE3_RATE_REFV	CP_OL_MODE3_RATE_REFV	Open loop mode3 rate of reference voltage
	OL_MODE3_MAX_REFV	CP_OL_MODE3_MAX_REFV	Open loop mode3 rate of reference voltage
	LIMIT_SPEED_CHANGE	CP_LIMIT_SPEED_CHANGE	Speed command maximum increase limit [rad/s]
	SPEED_PI_KP	CP_SPEED_PI_KP	Speed proportional gain
	SPEED_PI_KI	CP_SPEED_PI_KI	Speed Integral gain
	SPEED_LPF_K	CP_SPEED_LPF_K	Speed LPF parameter
	SW_ON	0	Active in case of “Low”
	SW_OFF	1	
	CHATTERING_CNT	10	Chattering removal
	VR1_SCALING	(MAX_SPEED + 50.0f) / 2048	Speed command value creation constant
	ADJUST_OFFSET	0x7FF	Speed command value offset adjustment constant
	POLE_PAIR	MP_POLE_PAIRS	Pole pairs
	REQ_CLR	0	VR1 stop command flag clearing
	REQ_SET	1	VR1 stop command flag setting
	ICS_INT_LEVEL	6	Interrupt priority level for GUI
	ICS_BRR	4	Bit late register select for GUI
	ICS_INT_MODE	1	Transfer mode select for GUI

Table 3-22 List of Macro definitions “mtr_ctrl_RX13T_mrssk.h” (1/2)

File name	Macro name	Definition value	Remarks
mtr_ctrl_RX13T_mrssk.h	MTR_PWM_TIMER_FREQ	32.0f	PWM timer count frequency [MHz]
	MTR_CARRIER_FREQ	20.0f	Carrier frequency [kHz]
	MTR_INT_DECIMATION	0	Interrupt skip number
	MTR_CTRL_PERIOD	((MTR_INT_DECIMATION + 1) / (MTR_CARRIER_FREQ * 1000))	Control Period
	MTR_CTRL_PERIOD_INV	(1.0f / (float)MTR_CTRL_PERIOD)	1/ Control Period
	MTR_CONTROL_FREQ	((MTR_CARRIER_FREQ * 1000) / (MTR_INT_DECIMATION + 1))	Control Frequency [Hz]
	MTR_DEADTIME	2	Dead time [μs]
	MTR_DEADTIME_SET	(uint16) (MTR_DEADTIME MTR_PWM_TIMER_FREQ)	Dead time setting value
	MTR_AD_FREQ	32.0f	Frequency of A/D conversion clock
	MTR_AD_SAMPLING_CYCLE	47.0f	A/D sampling time [Cycle]
	MTR_AD_SAMPLING_TIME	MTR_AD_SAMPLING_CYCLE / MTR_AD_FREQ	A/D sampling time [μs]
	MTR_AD_TIME_SET	(uint16) (MTR_PWM_TIMER_FREQ * MTR_AD_SAMPLING_TIME)	A/D sampling time count value
	MTR_CARRIER_SET_BASE	(uint16) ((MTR_PWM_TIMER_FREQ * 1000 /MTR_CARRIER_FREQ / 2))	Carrier period setting value
	MTR_CARRIER_SET	((uint16) MTR_CARRIER_SET_BASE + MTR_DEADTIME_SET))	Max setting value of carrier period
	MTR_HALF_CARRIER_SET	(uint16) (MTR_CARRIER_SET / 2)	Half of “MTR_CARRIER_SET”
	MTR_NDT_CARRIER_SET	(uint16) (MTR_CARRIER_SET - MTR_DEADTIME_SET)	no dead time of MTR_CARRIER_SET
	MTR_PORT_UP	PORT7.PODR.BIT.B1	U phase (positive phase) output port
	MTR_PORT_UN	PORT7.PODR.BIT.B4	U phase (negative phase) output port
	MTR_PORT_VP	PORT7.PODR.BIT.B2	V phase (positive phase) output port
	MTR_PORT_VN	PORT7.PODR.BIT.B5	V phase (negative phase) output port
	MTR_PORT_WP	PORT7.PODR.BIT.B3	W phase (positive phase) output port
	MTR_PORT_WN	PORT7.PODR.BIT.B6	W phase (negative phase) output port
	MTR_PORT_SW1	PORTB.PIDR.BIT.B5	SW1 input port
	MTR_PORT_SW2	PORTB.PIDR.BIT.B4	SW2 input port
	MTR_PORT_LED1	PORTD.PODR.BIT.B6	LED1 output port
	MTR_PORT_LED2	PORTD.PODR.BIT.B4	LED2 output port
	MTR_PORT_LED3	PORTD.PODR.BIT.B3	LED3 output port
	MTR_LED_ON	0	Active in case of “Low”
	MTR_LED_OFF	1	
	MTR_INPUT_V	24	Power supply voltage [V]
	MTR_MCU_ON_V	MTR_INPUT_V * 0.8f	MCU power on voltage [V]
	MTR_ADC_OFFSET	0x7FF	A/D offset

Table 3-22 List of Macro definitions “mtr_ctrl_RX13T_mrssk.h” (2/2)

File name	Macro name	Definition value	Remarks
mtr_ctrl_RX13T_mrssk.h	MTR_CURRENT_SCALING	25.0f/4095.0f	Inverter three phase current A/D conversion value resolution
	MTR_VDC_SCALING	111.0f/4095.0f	Inverter bus voltage A/D conversion value resolution
	MTR_OVERCURRENT_MARGIN_MULT	1.5f	Multiplier for over-current limit
	MTR_OVERCURRENT_LIMIT	MP_NOMINAL_CURRENT_RMS * MTR_SQRT_2 * MTR_OVERCURRENT_MARGIN_MULT	High current limit value [A]
	MTR_OVERVOLTAGE_LIMIT	28.0f	High voltage limit value [V]
	MTR_UNDERVOLTAGE_LIMIT	14.0f	Low voltage limit value [V]
	MTR_SPEED_TCNT	CMT1.CMCNT	Timer counter for speed calculation
	_80_MTU_TOLBR_T	(0X80U)	Bit select to define the data transfer time from TOLBRA to TOCR2A
	_20_MTU_OLS3N_LH	(0x20U)	OLS3N negative phase: initial output "L", active level "H”
	_00_MTU_OLS3P_HL	(0x00U)	OLS3P positive phase: initial output "H", active level "L"
	_08_MTU_OLS2N_LH	(0x08U)	OLS2N negative phase: initial output "L", active level "H"
	_00_MTU_OLS2P_HL	(0x00U)	OLS2P positive phase: initial output "H", active level "L"
	_02_MTU_OLS1N_LH	(0x02U)	OLS1N negative phase: initial output "L", active level "H"
	_00_MTU_OLS1P_HL	(0x00U)	OLS1P negative phase: initial output "H", active level "L"

Table 3-23 List of Macro definitions “mtr_spm_less_120.h” (1/4)

File name	Macro name	Definition value	Remarks
mtr_spm_less_120.h	MTR_POLE_PAIRS	MP_POLE_PAIRS	Motor Pole pairs
	MTR_TWOPI	2 * 3.14159265f	2π
	MTR_SQRT_2	1.41421356	Sqrt (2)
	MTR_RPM_RAD	MTR_TWOPI / 60 * MTR_POLE_PAIRS	2π/60
	MTR_SPEED_LIMIT_RPM	3000	Speed limit value (mechanical angle) [rpm]
	MTR_SPEED_LIMIT	MTR_SPEED_LIMIT_RPM * MTR_POLE_PAIRS * MTR_TWOPI / 60	Speed limit value (electrical angle) [rad/s]
	MTR_SPEED_PI_DECIMATION	1	Number of interrupt decimation times for speed PI control
	MTR_SPEED_PI_KP	CP_SPEED_PI_KP	Speed PI proportional gain
	MTR_SPEED_PI_KI	CP_SPEED_PI_KI	Speed PI Integral gain
	MTR_SPEED_PI_I_LIMIT_V	24.0f	Voltage PI control output limit value [V]
	MTR_SPEED_CALC_BASE	MTR_TWOPI * 4000000	Constant for speed measurement
	MTR_SPEED_LPF_K	CP_SPEED_LPF_K	Speed LPF parameter
	MTR_LIMIT_SPEED_CHANGE	CP_LIMIT_SPEED_CHANGE * MTR_RPM_RAD	Speed command maximum increase limit [rad/s]
	MTR_MAX_DRIVE_V	20.0f	Maximum command voltage [V]
	MTR_MIN_DRIVE_V	5.0f	Minimum command voltage [V]
	MTR_MAX_BOOT_V	8.0f	Maximum command voltage for BOOT MODE [V]
	MTR_TIMEOUT_CNT	2000	Timeout count limit
	MTR_SHIFT_ADJUST	0	Value of angle shift adjusting
	MTR_STOP_BEMF	0.5	value of stop motor BEMF [ms]
	MTR_OL_START_RAD	CP_OL_START_RPM * MTR_RPM_RAD	Open loop start speed [rpm]
	MTR_OL_MODE1_CHANGE_RAD	CP_OL_MODE1_CHANGE_RPM * MTR_RPM_RAD	Open loop mode 1 changing speed [rpm]
	MTR_OL_MODE2_CHANGE_RAD	CP_OL_MODE2_CHANGE_RPM * MTR_RPM_RAD	Mode changing speed [rpm]
	MTR_OL_START_REFV	CP_OL_START_REFV	Command voltage at startup [V]
	MTR_OL_MODE1_RATE_RAD	CP_OL_MODE1_RATE_RPM * MTR_RPM* MTR_RPM_RAD	Command speed adding value [rad]
	MTR_OL_MODE2_RATE_REFV	CP_OL_MODE2_RATE_REFV	Command voltage adding value [V]
	MTR_OL_MODE2_RATE_RAD	CP_OL_MODE2_RATE_RPM * MTR_RPM * MTR_RPM_RAD	Command speed adding value [rad]
	MTR_OL_MODE3_RATE_REFV	CP_OL_MODE3_RATE_REFV	Command voltage adding value [V]
	MTR_OL_MODE3_MAX_REFV	CP_OL_MODE3_MAX_REFV	Maximum command voltage in open loop mode [V]
	MTR_OL_PATTERN_SET_CALC	MTR_CARRIER_FREQ*1000*MTR_TWOPI/6	Pattern change cycle in open loop mode

Table 3-23 List of Macro definitions “mtr_spm_less_120.h” (2/4)

File name	Macro name	Definition value	Remarks
mtr_spm_less_120.h	MTR_PATTERN_CW_V_U	2	CW virtual hall sensor value
	MTR_PATTERN_CW_W_U	3	
	MTR_PATTERN_CW_W_V	1	
	MTR_PATTERN_CW_U_V	5	
	MTR_PATTERN_CW_U_W	4	
	MTR_PATTERN_CW_V_W	6	
	MTR_PATTERN_CCW_V_U	3	CCW virtual hall sensor value
	MTR_PATTERN_CCW_V_W	2	
	MTR_PATTERN_CCW_U_W	6	
	MTR_PATTERN_CCW_U_V	4	
	MTR_PATTERN_CCW_W_V	5	
	MTR_PATTERN_CCW_W_U	1	
	MTR_PATTERN_ERROR	0	Conduction pattern
	MTR_UP_PWM_VN_ON	1	
	MTR_UP_PWM_WN_ON	2	
	MTR_VP_PWM_UN_ON	3	
	MTR_VP_PWM_WN_ON	4	
	MTR_WP_PWM_UN_ON	5	
	MTR_WP_PWM_VN_ON	6	
	MTR_UP_ON_VN_PWM	7	
	MTR_UP_ON_WN_PWM	8	
	MTR_VP_ON_UN_PWM	9	
	MTR_VP_ON_WN_PWM	10	
	MTR_WP_ON_UN_PWM	11	
	MTR_WP_ON_VN_PWM	12	
	MTR_U_PWM_VN_ON	13	
	MTR_U_PWM_WN_ON	14	
	MTR_V_PWM_UN_ON	15	
	MTR_V_PWM_WN_ON	16	
	MTR_W_PWM_UN_ON	17	
	MTR_W_PWM_VN_ON	18	
	MTR_UP_ON_V_PWM	19	
	MTR_UP_ON_W_PWM	20	
	MTR_VP_ON_U_PWM	21	
	MTR_VP_ON_W_PWM	22	
	MTR_WP_ON_U_PWM	23	
	MTR_WP_ON_V_PWM	24	
	MTR_DRAW_IN_1ST_TIME	1	Draw in for first time
	MTR_DRAW_IN_2ND_TIME	2	Draw in for second time
	MTR_OFFSET_CALC_TIME	CP_OFFSET_CALC_TIME	Current offset value calculation time [ms]

Table 3-23 List of Macro definitions “mtr_spm_less_120.h” (3/4)

File name	Macro name	Definition value	Remarks
mtr_spm_less_120.h	MTR_BOOT_REF_V	CP_BOOT_REF_V	voltage command value
	MTR_V_UP_TIME	CP_V_UP_TIME	voltage command value addition time
	MTR_V_UP_STEP	MTR_BOOT_REF_V/MTR_V_UP_TIME	voltage command value addition step
	MTR_V_CONST_TIME	CP_V_CONST_TIME	voltage command value constant time
	MTR_CW	0	Rotation direction setting value
	MTR_CCW	1	
	MTR_FLG_CLR	0	Constant for flag management
	MTR_FLG_SET	1	
	MTR_ICS_DECIMATION	4	Number of function call decimation times for GUI
	MTR_V_ZERO_CONST	0	zero voltage constant
	MTR_V_UP	1	increase of voltage
	MTR_V_CONST	2	voltage constant
	MTR_V_OPENLOOP	3	Open-loop voltage setting mode
	MTR_V_PI_OUTPUT	4	Speed PI output voltage setting mode
	MTR_SPEED_ZERO_CONST	0	Speed 0
	MTR_SPEED_OPENLOOP_1	1	Open loop MODE1
	MTR_SPEED_OPENLOOP_2	2	Open loop MODE2
	MTR_SPEED_OPENLOOP_3	3	Open loop MODE3
	MTR_SPEED_CHANGE	4	Speed changing
	MTR_ID_A	0	Motor ID A
	MTR_ID_B	1	Motor ID B
	MTR_MODE_INACTIVE	0x00	Inactive mode
	MTR_MODE_ACTIVE	0x01	Active mode
	MTR_MODE_ERROR	0x02	Error mode
	MTR_SIZE_STATE	3	State size
	MTR_EVENT_INACTIVE	0x00	Inactive event
	MTR_EVENT_ACTIVE	0x01	Active event
	MTR_EVENT_ERROR	0x02	Error event
	MTR_EVENT_RESET	0x03	Reset event
	MTR_SIZE_EVENT	4	Event size
	MTR_MODE_INIT	0x00	Init mode
	MTR_MODE_BOOT	0x01	Boot mode
	MTR_MODE_DRIVE	0x02	Drive mode
	MTR_MODE_ANALYSIS	0x03	Analysis Mode
	MTR_MODE_TUNE	0x04	Tune mode
	MTR_SENSOR_LESS	0x01	Sensorless
	MTR_SENSOR_HALL	0x02	Hall sensor
	MTR_SENSOR_ENCD	0x04	Encoder
	MTR_SENSOR_RESO	0x08	Resolver
	MTR_METHOD_FOC	0x00	Fields Oriented Control
	MTR_METHOD_180	0x01	180-degree control
	MTR_METHOD_WIDE	0x02	Wide angle electricity control
	MTR_METHOD_120	0x03	120-degree control
	MTR_CONTROL_CURRENT	0x01	Current control
	MTR_CONTROL_SPEED	0x02	Speed control

Table 3-23 List of Macro definitions “mtr_spm_less_120.h” (4/4)

File name	Macro name	Definition value	Remarks
mtr_spm_less_120.h	MTR_CONTROL_POSITION	0x04	Position control
	MTR_CONTROL_TORQUE	0x08	Torque control
	MTR_CONTROL_VOLTAGE	0x10	Voltage control
	MTR_ERROR_NONE	0x00	No error
	MTR_ERROR_OVER_CURRENT	0x01	Over current error
	MTR_ERROR_OVER_VOLTAGE	0x02	Over voltage error
	MTR_ERROR_OVER_SPEED	0x04	Over speed error
	MTR_ERROR_HALL_TIMEOUT	0x08	Hall timeout error
	MTR_ERROR_BEMF_TIMEOUT	0x10	BEMF timeout error
	MTR_ERROR_HALL_PATTERN	0x20	Hall pattern error
	MTR_ERROR_BEMF_PATTERN	0x40	BEMF pattern error
	MTR_ERROR_UNDER_VOLTAGE	0x80	Under voltage error
	MTR_ERROR_UNKNOWN	0xff	Unknown error

3.6 Control flows (flow charts)

3.6.1 Main process

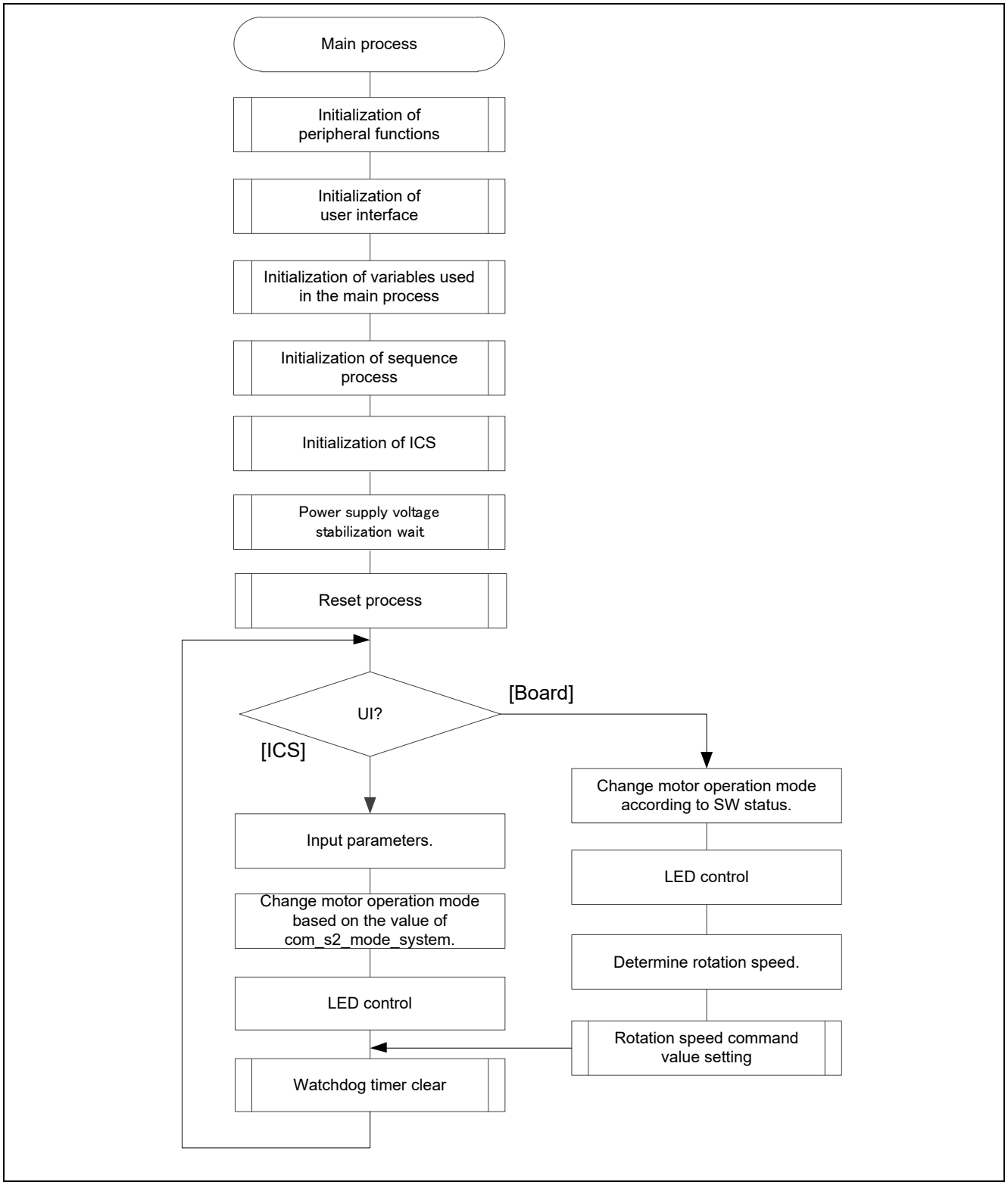


Figure 3-7 Main Process Flowchart

3.6.2 Carrier cycle interrupt handling

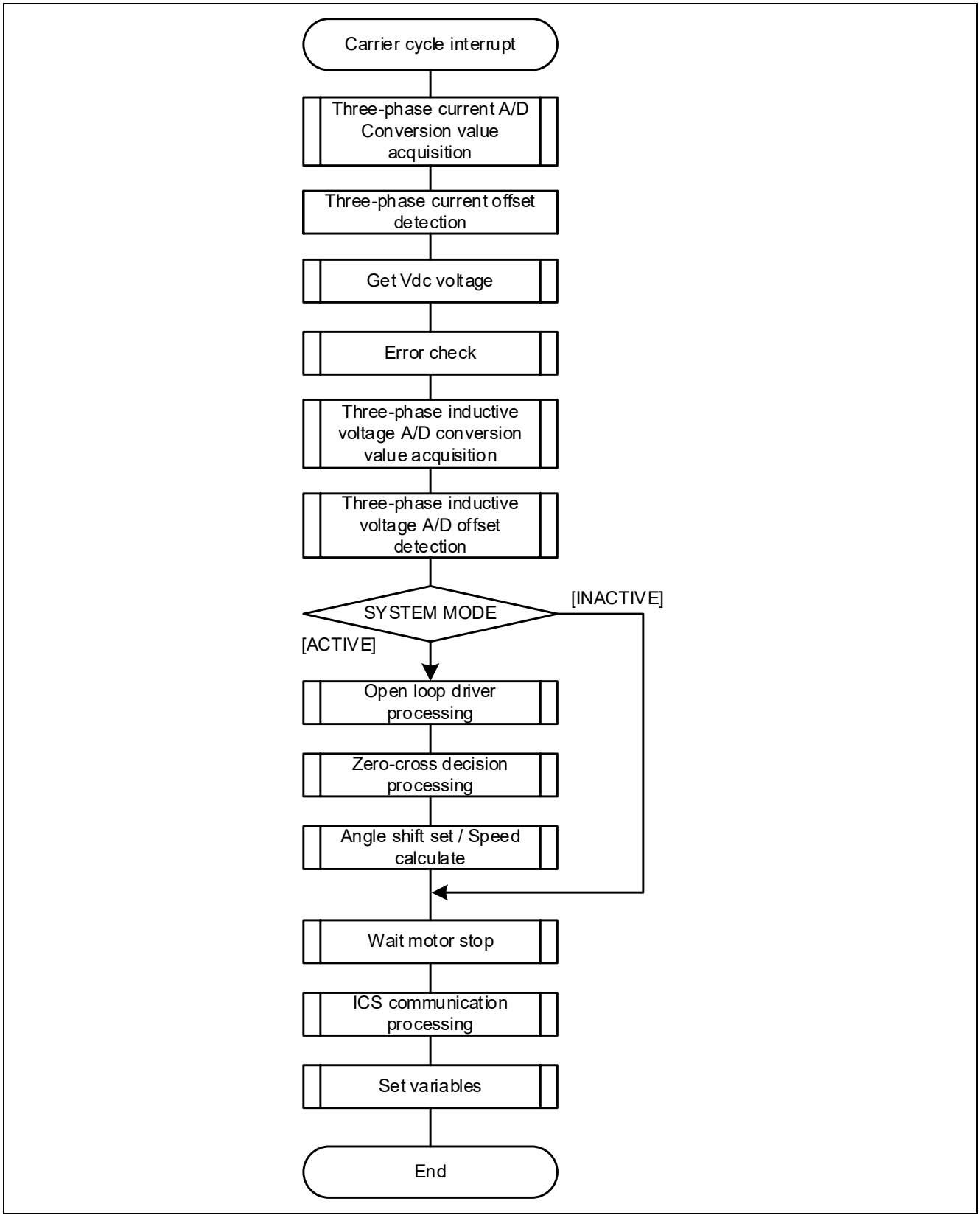


Figure 3-8 50 [µs] Cycle Interrupt Handling (Sensorless 120-degree Control)

3.6.3 1 ms interrupt handling

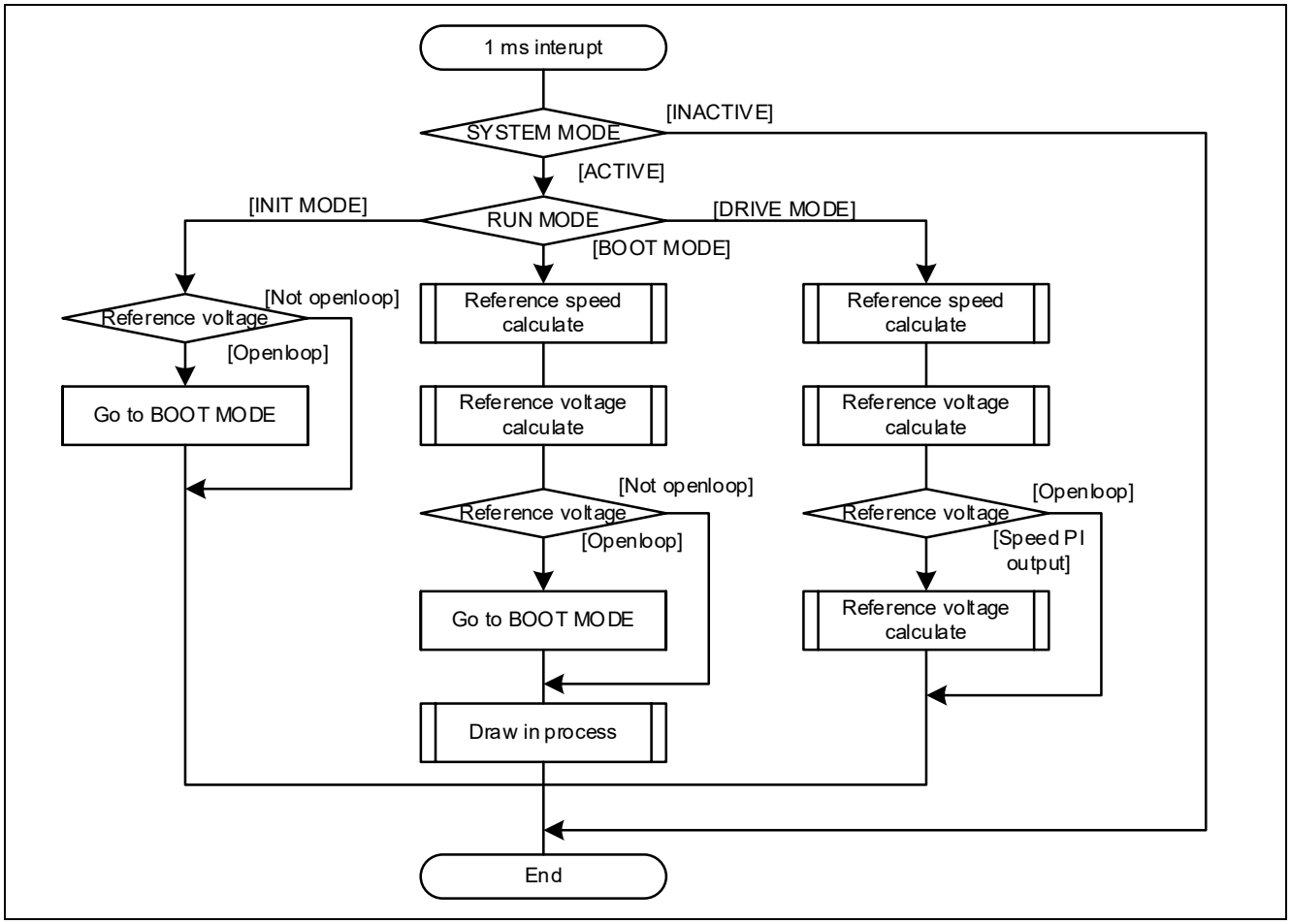


Figure 3-9 1 [ms] Interrupt Handling

3.6.4 Overcurrent interrupt handling

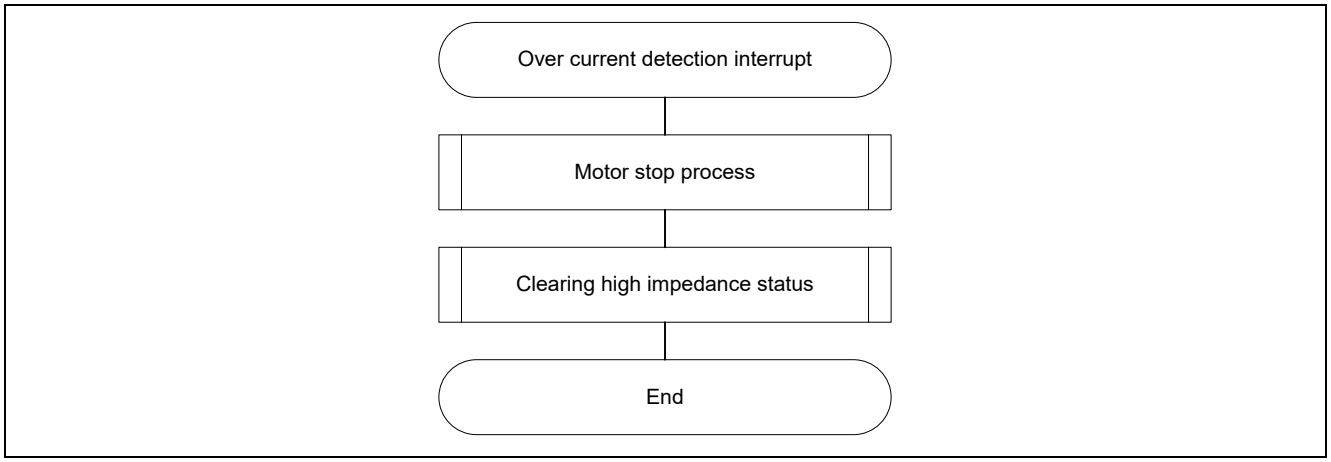


Figure 3-10 Over Current Detection Interrupt Handling

4. Motor Control Development Support Tool, ‘Renesas Motor Workbench’

4.1 Overview

In the target sample programs described in this application note, user interfaces (rotating/stop command, rotation speed command, etc.) are based on the motor control development support tool, ‘Renesas Motor Workbench’ can be used. Please refer to ‘Renesas Motor Workbench User’s Manual’ for usage and more details.

You can find ‘Renesas Motor Workbench’ on Renesas Electronics Corporation website.

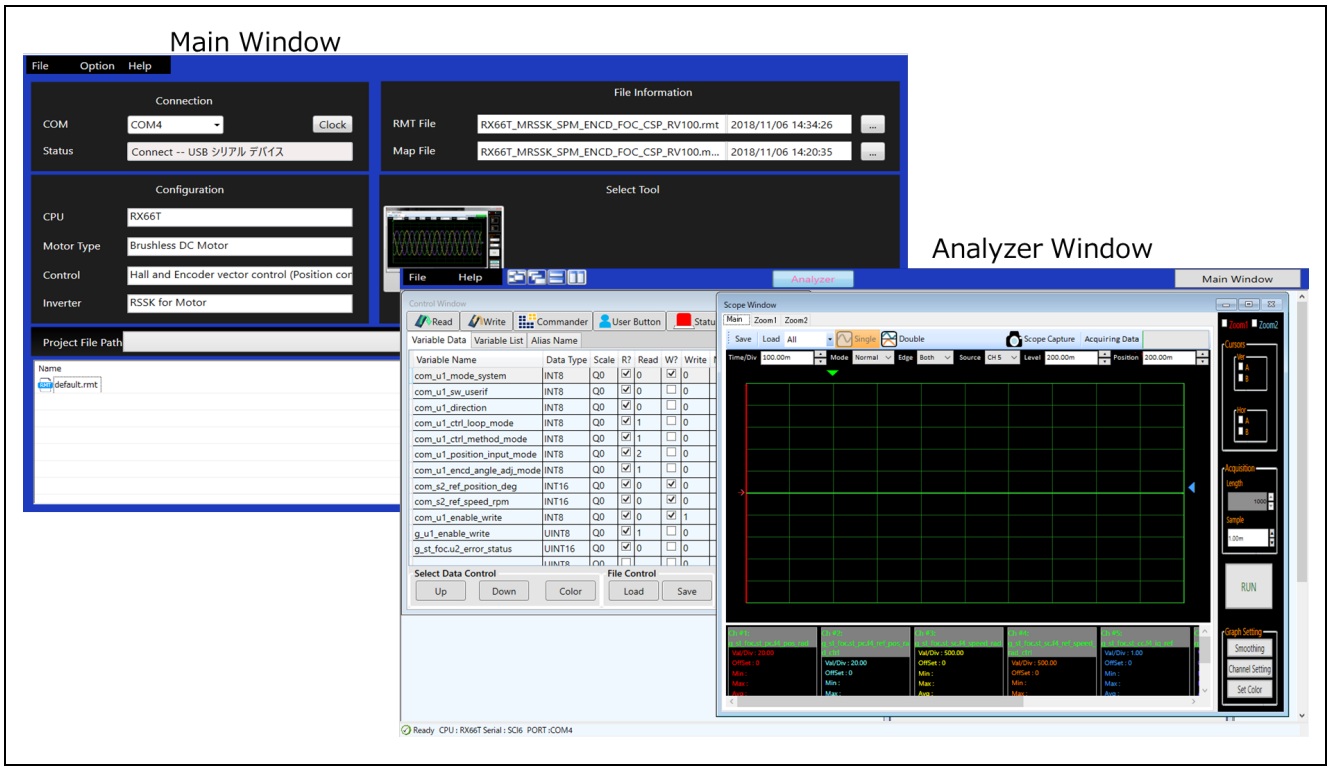


Figure 4-1 Renesas Motor Workbench – Appearance

Set up for Renesas Motor Workbench

(1) Start ‘Motor RSSK Support Tool’ by clicking this icon.
(2) Drop down menu [RMT File] → [Open RMT File(O)].
And select RMT file in ‘[Project Folder]/ics/’.
(3) Use the ‘Connection’ COM select menu to choose the COM port for Motor RSSK.
(4) Click on the ‘Analyzer’ icon of Select Tool panel to open Analyzer function window.
(5) Please refer to ‘4.3 Operation Example for Analyzer’ for motor driving operation.

4.2 List of variables for Analyzer function

Table 4-1 is a list of variables for Analyzer. These variable values are reflected to the protect variables when the same values as g_s2_enable_write are written to com_s2_enable_write. However, note that variables with (*) do not depend on com_s2_enable_write.

The display variable “ics_*” is corresponding to the structure variable.

Table 4-1 List of Variables for Analyzer

Variable name	Type	Content	Remarks ([ ]: reflection variable name)
com_s2_sw_ui (*)	int16	User interface switch 0: GUI user interface use (default) 1: Board user interface use	[g_s2_sw_ui]
com_s2_mode_system (*)	int16	State management 0: Stop mode 1: Run mode 3: Reset	[g_s2_mode_system]
com_s2_direction	int16	Rotation direction 0: CW 1: CCW	[s2_direction]
com_u2_mtr_p	uint16	Number of pole pairs	[st_g.u2_mtr_p]
com_s2_ref_speed_rpm	int16	Speed command value (mechanical angle) [rpm]	[st_g.f4_ref_speed_rad]
com_f4_kp_speed	float32	Speed PI control proportional gain	[st_g.f4_kp_speed]
com_f4_ki_speed	float32	Speed PI control integral gain	[st_g.f4_ki_speed]
com_f4_speed_lpf_k	float32	Speed LPF parameter	[st_g.f4_speed_lpf_k]
com_f4_limit_speed_change	float32	Command speed changing limit [rad/s]	[st_g.f4_limit_speed_change]
com_s2_ol_start_rpm	int16	Speed at startup	[st_g.s2_ol_start_rad]
com_s2_ol_mode1_change_rpm	int16	Mode changing speed	[st_g.s2_ol_mode1_change_rad]
com_s2_ol_mode2_change_rpm	int16	Mode changing speed	[st_g.s2_ol_mode2_change_rad]
com_f4_ol_start_refv	float32	Command voltage at startup	[st_g.f4_ol_start_refv]
com_f4_ol_mode1_rate_rpm	float32	Command speed adding value	[st_g.f4_ol_mode1_rate_rad]
com_f4_ol_mode2_rate_refv	float32	Command voltage adding value	[st_g.f4_ol_mode2_rate_refv]
com_f4_ol_mode2_rate_rpm	float32	Command speed adding value	[st_g.f4_ol_mode2_rate_rad]
com_f4_ol_mode3_rate_refv	float32	Command voltage adding value	[st_g.f4_ol_mode3_rate_refv]
com_f4_ol_mode3_max_refv	float32	Maximum command voltage in open loop mode	[st_g.f4_ol_mode3_max_refv]
com_u2_offset_calc_time	uint16	Current offset value calculation time [ms]	[st_g.u2_offset_calc_time]
com_f4_boot_ref_v	float32	Voltage command value	[st_g.f4_boot_ref_v]
com_u2_v_up_time	uint16	Voltage command value addition time	[st_g.u2_v_up_time]
com_u2_v_const_time	uint16	Voltage command value constant time	[st_g.u2_v_const_time]
com_s2_enable_write	int16	Enable to rewriting variables	

4.3 Operation Example for Analyzer

Following is an example demonstrating the motor driving operation using Analyzer. Operation is using “Control Window”. Refer to ‘In Circuit Scope manual’ for ‘Renesas Motor Workbench User’s Manual’.

- Driving the motor
  (1) The [W?] check boxes contain checkmarks for “com_s2_mode_system”, “com_s2_ref_speed_rpm”, “com_s2_enable_write”
  (2) Type a reference speed value in the [Write] box of “com_s2_ref_speed_rpm”.
  (3) Click the “Write” button.
  (4) Click the “Read” button. Confirm the [Read] box of “com_s2_ref_speed_rpm”, “g_s2_enable_write”.
  (5) Type a same value of “g_s2_enable_write” in the [Write] box of “com_s2_enable_write”.
  (6) Type a value of “1” in the [Write] box of “com_s2_mode_system”.
  (7) Click the “Write” button.

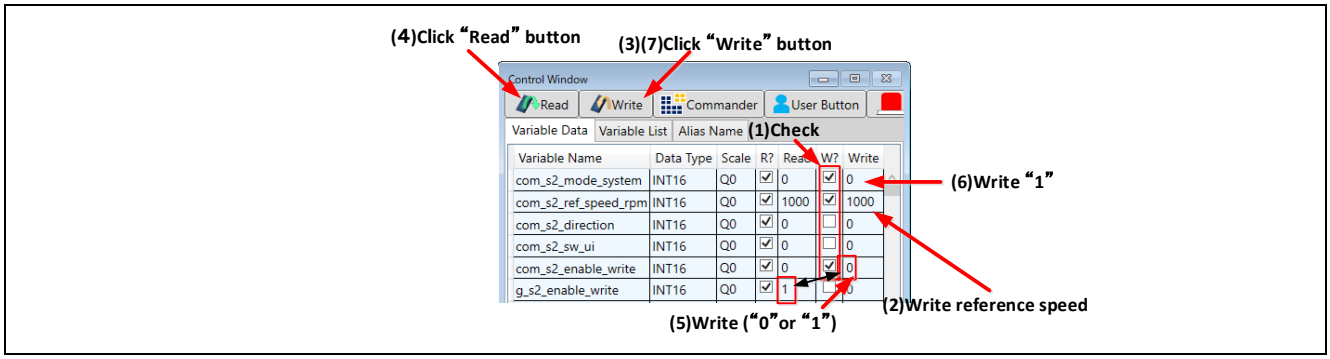


Figure 4-2 Procedure – Driving the motor

- Stop the motor
  (1) Type a value of “0” in the [Write] box of “com_s2_mode_system”
  (2) Click the “Write” button.

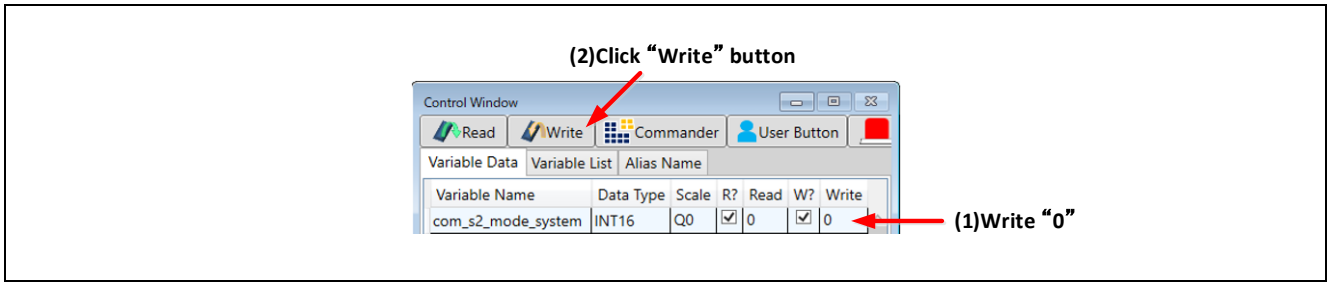


Figure 4-3 Procedure – Stop the motor

- Error cancel operation
  (1) Type a value of “3” in the [Write] box of “com_s2_mode_system”
  (2) Click the “Write” button.

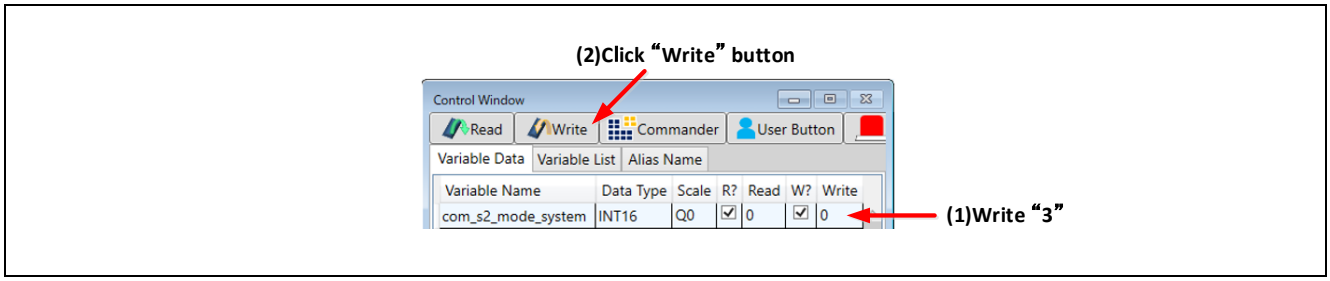


Figure 4-4 Procedure – Error cancel operation

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec.11.20	—	First edition issued

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

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

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

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

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

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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