R01AN3272EJ0101
Rev. 1.01
Dec.17.2018

RX23T

120-degree conducting control of permanent magnetic synchronous motor using hall sensors (Implementation)

Summary

This application note aims at explains sample programs driving a permanent magnetic synchronous motor using hall sensors in the 120-degree conducting method on the RX23T 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.

• RX23T (R5F523T5ADFM)

Target sample programs

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

RX23T_MRSSK_SPM_HALL_120_CSP_V100 (IDE : CS+)

RX23T_MRSSK_SPM_HALL_120_E2S_V100 (IDE : e2studio)

RX23T 120-degree conducting control using hall sensors sample program for RX23T 24V Motor Control Evaluation System

Reference

- RX23T Group User's Manual: Hardware (R01UH0520)
- Application note: '120-degree conducting control of permanent magnetic synchronous motor: algorithm'(R01AN2657)
- Renesas Motor Workbench User's Manual (R21UZ0004)
- Renesas Solution Starter Kit 24V Motor Control Evaluation System for RX23T User's Manual (R20UT3697)

Contents

1. Overview

This application note explains how to implement the 120-degree conducting control sample programs of permanent magnetic synchronous motor (PMSM) using hall sensors based on the RX23T microcontroller and how to use the motor control development support tool, 'Renesas Motor Workbench'. Note that these sample programs use the algorithm described in the application note '120-degree conducting control of permanent magnetic synchronous motor: algorithm'.

1.1 Development environment

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

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

Microcontroller	Evaluation board	Motor
RX23T (R5F523T5ADFM)	24V invertor board and RX23T CPU Card (Note 1)	TG-55L(24V) (Note 2)

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

Toolchain version
v2.04.01(Renesas CCRX Toolchain)

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

Notes:

1. 24V invertor board and RX23T CPU Card (RTK0EM0006S01212BJ) are products of Renesas Electronics Corporation.
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.

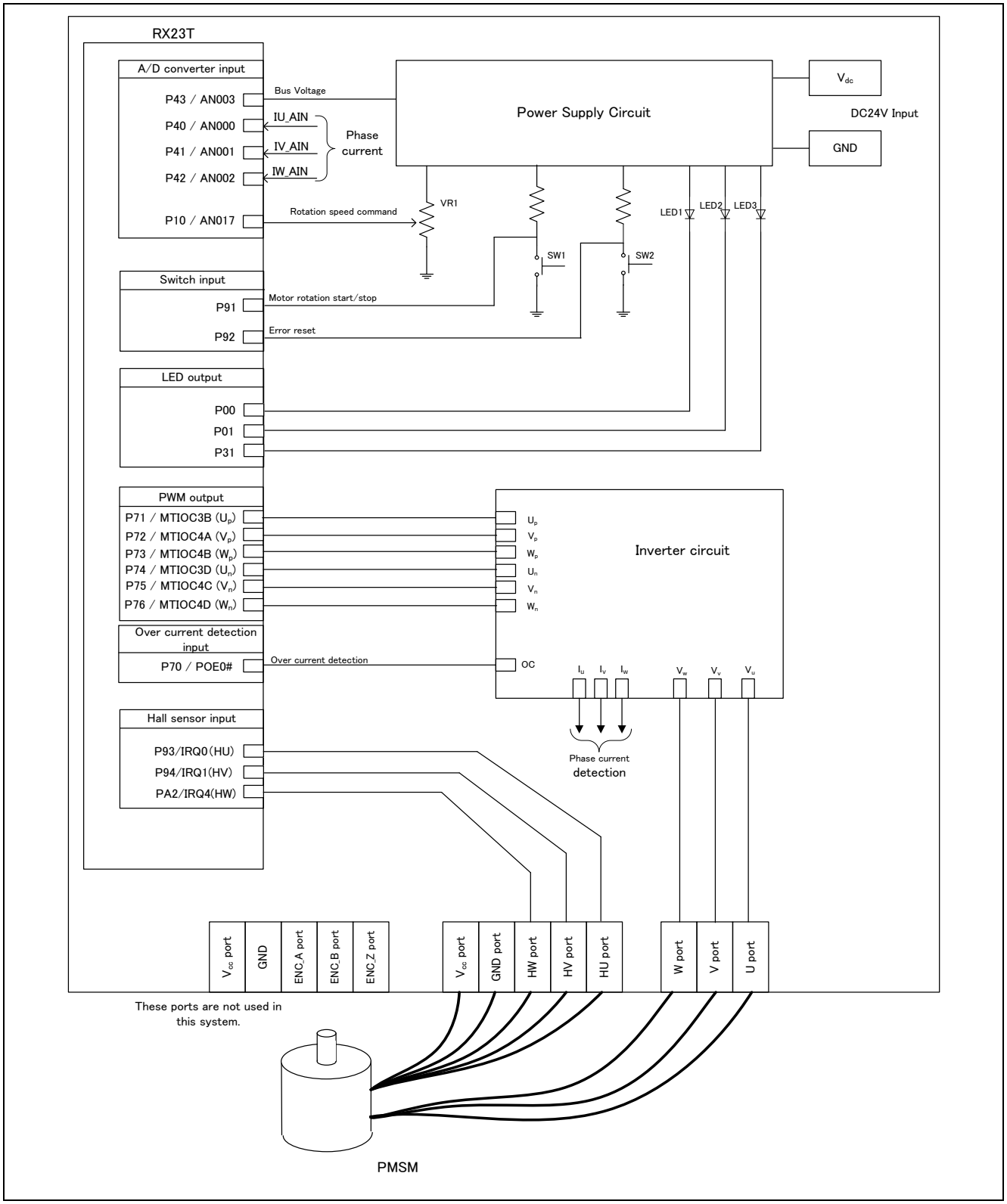


Figure 2-1 Hardware Configuration Diagram

2.2 Hardware specifications

2.2.1 User interface

Table 2-1 is a list of user interfaces of 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	Toggle switch (SW2)	Command of recovery from error status
LED1	Yellow green LED	- At the time of Motor rotation: ON - At the time of stop: OFF
LED2	Yellow green LED	- At the time of error detection: ON - At the time of normal operation: OFF
LED3	Yellow green LED	Not used in this system
RESET	Push switch (RESET1)	System reset

Table 2-2 is a list of port interfaces of RX23T microcontroller of this system.

Table 2-2 Port Interface

R5F523T5ADFM Port name	Function
P43 / AN003	Inverter bus voltage measurement
P10 / AN017	For inputting rotation speed command values (analog values)
P91	START/STOP toggle switch
P92	ERROR RESET toggle switch
P00	LED1 ON/OFF control
P01	LED2 ON/OFF control
P31	LED3 ON/OFF control (not used)
P40 / AN000	U phase current measurement
P41 / AN001	V phase current measurement
P42 / AN002	W phase current measurement
P71 / MTIOC3B	PORT output / PWM output (U_p)
P72 / MTIOC4A	PORT output / PWM output (V_p)
P73 / MTIOC4B	PORT output / PWM output (W_p)
P74 / MTIOC3D	PORT output / PWM output (U_n)
P75 / MTIOC4C	PORT output / PWM output (V_n)
P76 / MTIOC4D	PORT output / PWM output (W_n)
P70 / POE0#	PWM emergency stop input at the time of overcurrent detection
P93 / IRQ0	Hall sensor input (HU)
P94 / IRQ1	Hall sensor input (HV)
PA2 / IRQ4	Hall sensor input (HW)

2.2.2 Peripheral functions

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

Table 2-3 Peripheral Functions List

Peripheral Function	Usage
12-bit A/D converter	- Rotation speed command value input - Inverter bus voltage measurement - Current measurement of each phase U, V, and W
CMT	- 1[ms] interval timer - Free-running timer for rotation speed measurement
MTU3	Complementary PWM output
POE3	Set ports executing PWM output to high impedance state when an overcurrent is detected.
External interrupt (IRQ)	External interrupt by hall sensors' signals (both edge)

(1) 12-bit A/D converter

The rotation speed command value input, U phase current (Iu), V phase current (Iv), W phase current (Iw) and inverter bus voltage (Vdc) are measured by using the single scan mode with the sample-and- hold function (use hardware trigger).

(2) Compare match timer (CMT)

a. 1msec interval timer

The channel 0 of the compare match timer (CMT) is used as 1 millisecond 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 interrupt is 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 (high active) is performed by using the complementary PWM mode.

(4) Port output enable 3 (POE3)

The ports executing PWM output are set to high impedance state when an overcurrent is detected (when a falling edge of the POE0# port is detected) and when an output short circuit is detected.

(5) External interrupt (IRQ)

The hall sensors' signals are inputted for detection of rotor position.

Both edge mode is used. When the interrupt occurs, measurement of rotation speed, changing conduction pattern, and reading hall sensors' signals (detection of rotor position) are performed.

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 Programs

RX23T_MRSSK_SPM_HALL_120_CSP_V100 RX23T_MRSSK_SPM_HALL_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_rx23t.h	RX23T dependent processing part header
		mtr_spm_hall_120.h	120-degree conducting control using hall sensors dependent part header
		control_parameter.h	Control characteristic dependent processing part header
		motor_parameter.h	Motor characteristic dependent processing part header
		mtr_ctrl_rx23t_mrssk.h	RX23T 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 part header
	ics	ICS_RX23T.obj	Library for GUI
		ICS_RX23T.h	Header for GUI
	src	main.c	Main function, user interface control
		mtr_ctrl_mrssk.c	Board dependent processing part
		mtr_ctrl_rx23t.c	RX23T dependent processing part
		mtr_interrupt.c	Interrupt handler
		mtr_spm_hall_120.c	120-degree conducting control using hall sensors dependent part
		mtr_ctrl_rx23t_mrssk.c	RX23T 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 the module configuration of the sample programs.

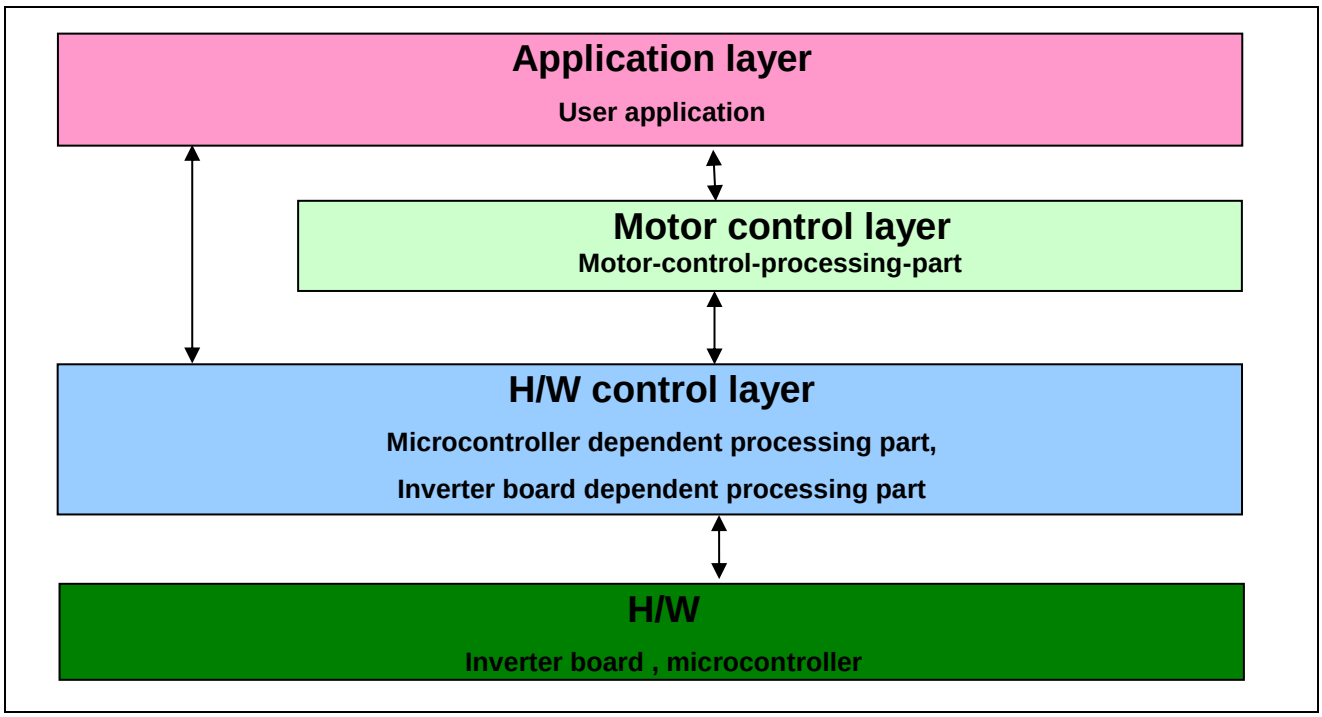


Figure 2-2 Module Configuration of the Sample Programs

Table 2-5 Module Configuration of the Sample Programs

Layers	File name
Application layer	main.c
Motor control layer	mtr_spm_hall_120.c mtr_feedback.c mtr_gmc.c mtr_filter.c mtr_driver_access.c mtr_interrupt.c(Note 1)
H/W control layer	mtr_ctrl_rx23t_mrssk.c mtr_ctrl_rx23t.c mtr_ctrl_mrssk.c mtr_interrupt.c(Note1)

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 of 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 depending on the level of SW1 (P91 ("Low": rotation start "High": stop) or input from GUI (Note 1)
Position detection of rotor magnetic pole	Position detection by signals of hall sensors (by each 60 degrees)
Input voltage	DC24[V]
Carrier frequency (PWM)	20 [kHz]
Control cycle	External interrupts by each edge of signals of hall sensors (both edge)
Rotation speed control range	Both CW and CCW: 550 [rpm] to 2650 [rpm]
Processing stop for protection	- Disables the motor control signal output (six outputs), under any of the following conditions. 1. Current of each phase exceeds 2.0 [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. When the motor rotates, the interrupt of hall sensors' signals are not detected for 200 [ms]. 6. Fault detection of 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 POE0# port is detected) and when an output short circuit is detected.

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

3. Descriptions of the control program

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

3.1 Contents of control

3.1.1 Motor start/stop

Starting and stopping of the motor are controlled by 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 “Low” level, it is determined that the start switch is being pressed. Conversely, when the level is switched to “High”, 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 550 [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 and A/D conversion of the VR1 output value (analog value). The A/D converted VR1 value is used as rotation speed command value, as shown below. When the rotation speed command value is below the minimum speed, the command value will be limited to the minimum speed value and also the value is over the maximum speed, the command value will be limited to the maximum speed.

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: 0800H to 0FFFH	AN017
	CCW	0 rpm to 2700 rpm: 07FFH to 0000H	

(2) Inverter bus voltage

Inverter bus voltage is measured as given in Table 3-2.

It is used for modulation factor calculation and over/under 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	AN003

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

The U, V, and W phase currents are measured as shown in Table 3-3 and used for determining over current (software).

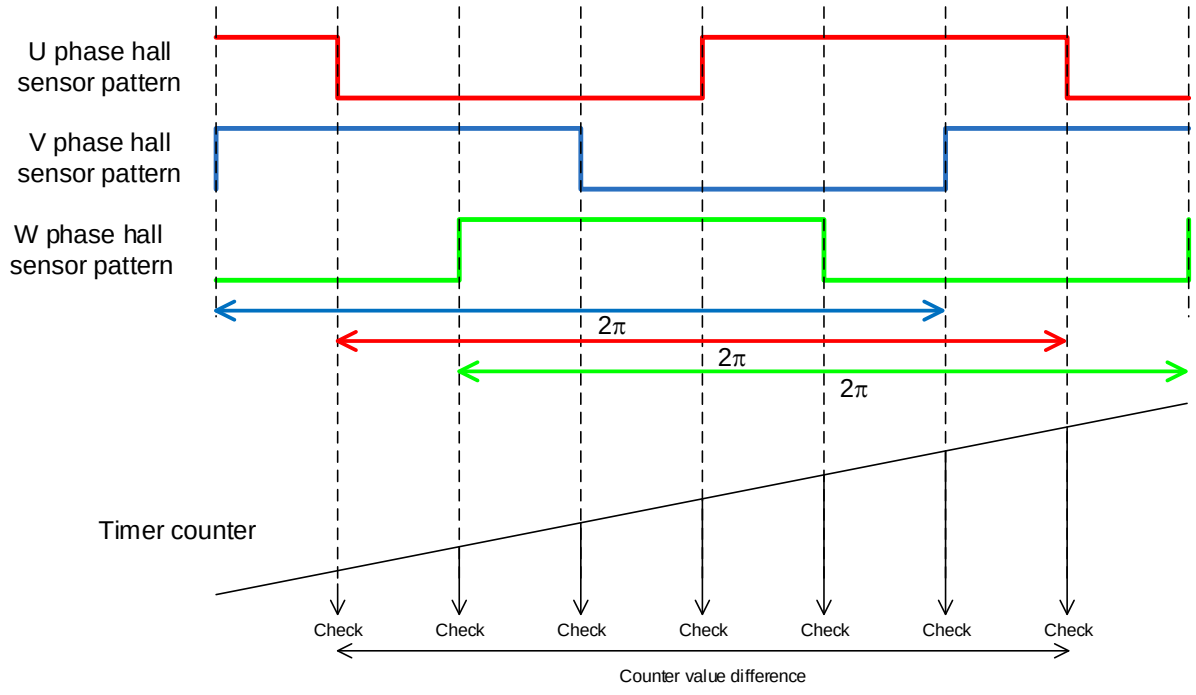
Table 3-3 Conversion Ratio of U, V, and W Phase current

Item	Conversion ratio (U, V, and W phase current : A/D conversion value)	Channel
U, V, W phase current	-10 [A] to 10 [A]: 0000H to 0FFFH (Note 1)	Iu: AN000 Iv: AN001 Iw: AN002

Note: 1. For more details of A/D conversion characteristics, refer to RX23T 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 an external interrupt due to hall sensor signals occur, while having the timer of channel 1 of compare match timer performed free running.



Motor rotation speed [rad/s] = (2π × timer count frequency) / (counter value difference)

Figure 3-1 Motor Rotation Speed Calculation Method

The target sample software of this application note use 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

$Kp\omega$: Speed PI proportional gain, $KI\omega$: 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.

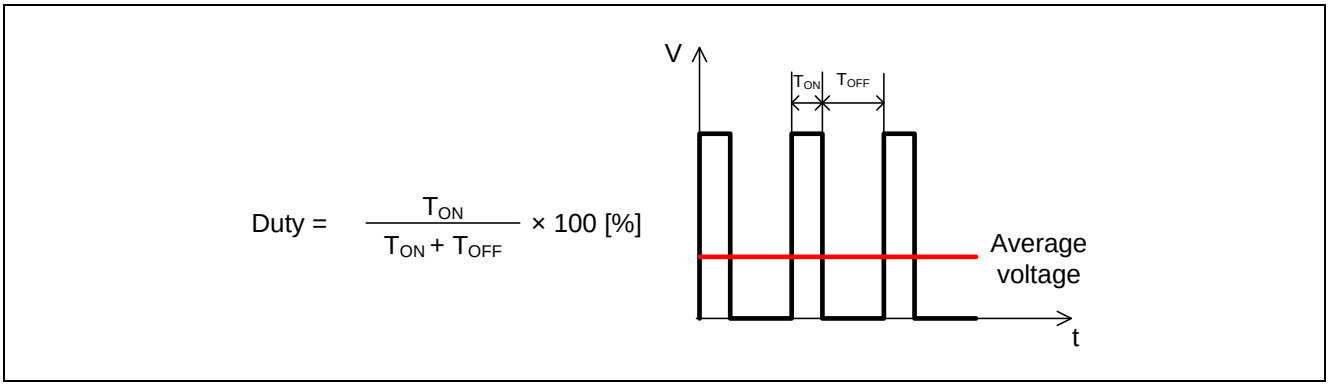


Figure 3-2 PWM Control

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.

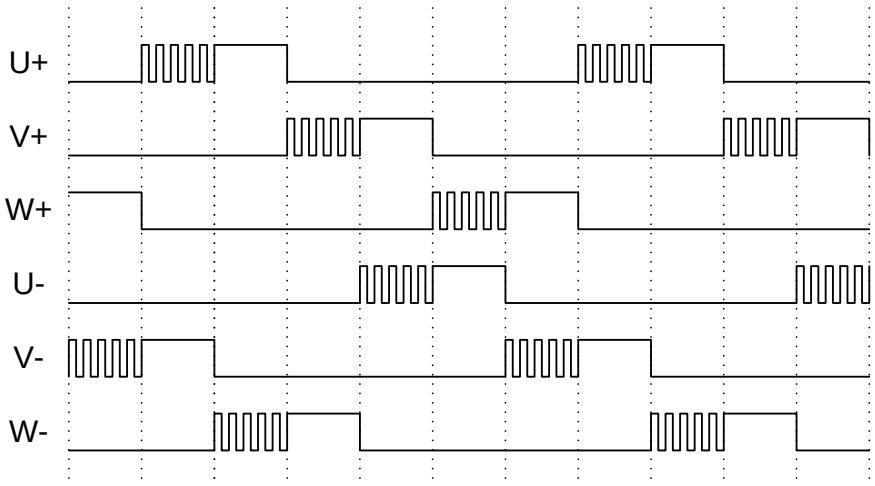


Figure 3-3 Non-complimentary first-60-degree Chopping

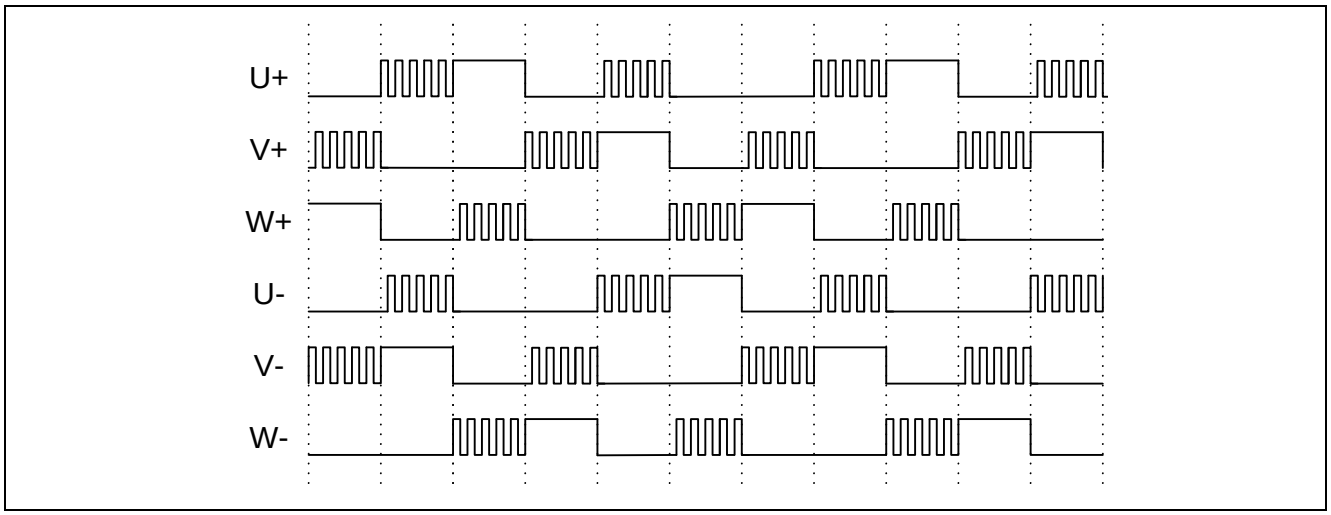
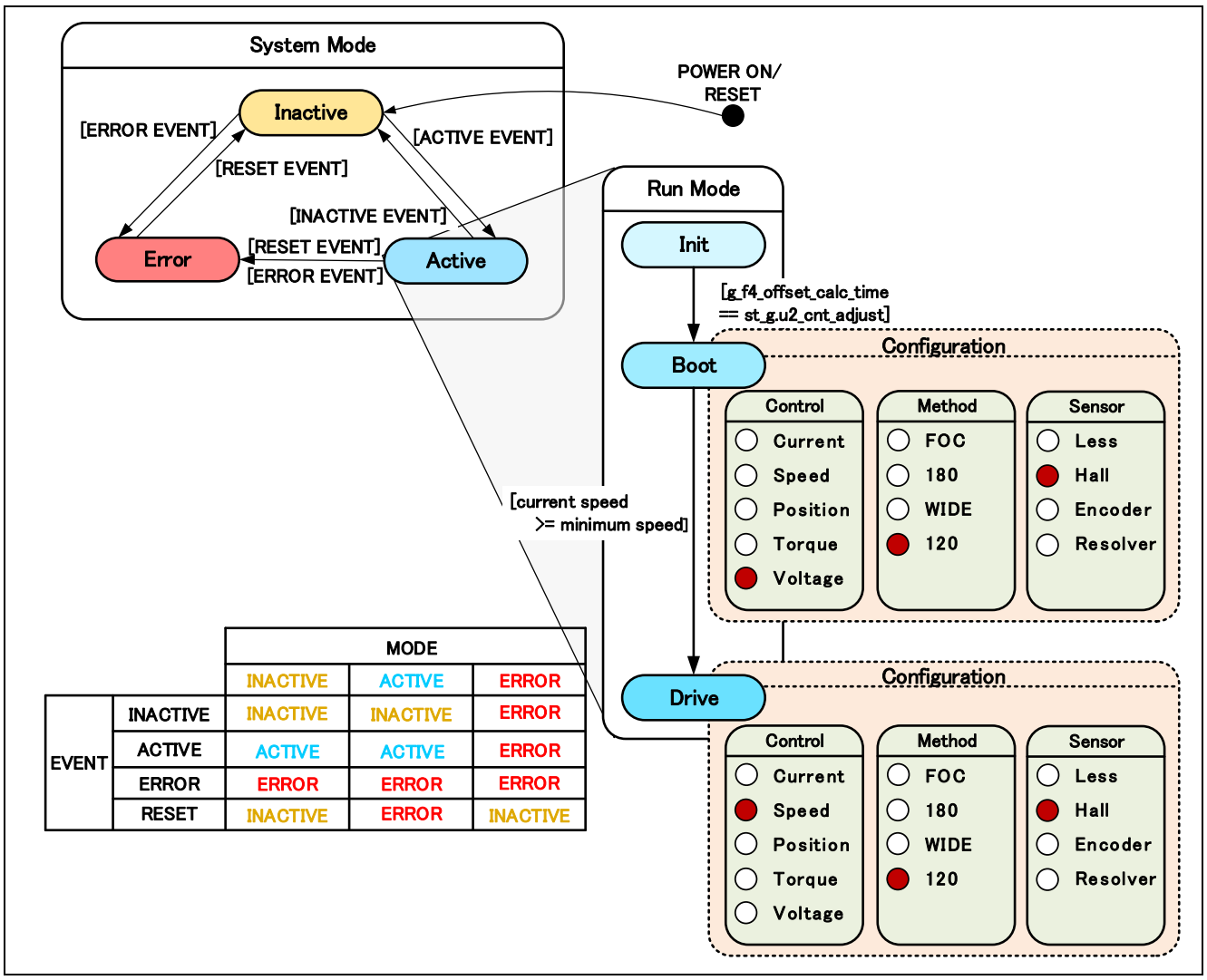


Figure 3-4 Complimentary first-60-degree Chopping

3.1.5 State transition

Figure 3-5 shows state transition diagrams of 120-degree conducting control using hall sensors 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 120-degree Conducting Control using hall sensors Software

3.1.6 Start-up method

In the case of 120-degree conducting control using hall sensors, the rotor position can be determined by hall sensors' signals. Therefore, the conduction pattern at start-up is also determined definitely.

When the control is changed to PI control, at least the motor needs to rotate one time (refer to 3.1.3). In this sample software, at start-up the motor is controlled in open loop with a constant voltage until the motor rotate one time.

Figure 3-6 shows the start-up method in this sample software. In "MTR_MODE_BOOT", open loop with a constant voltage which is set by st_g.f4_start_refv is performed. The mode changes to "MTR_MODE_DRIVE" when the current speed reaches the defined minimum speed (550rpm).

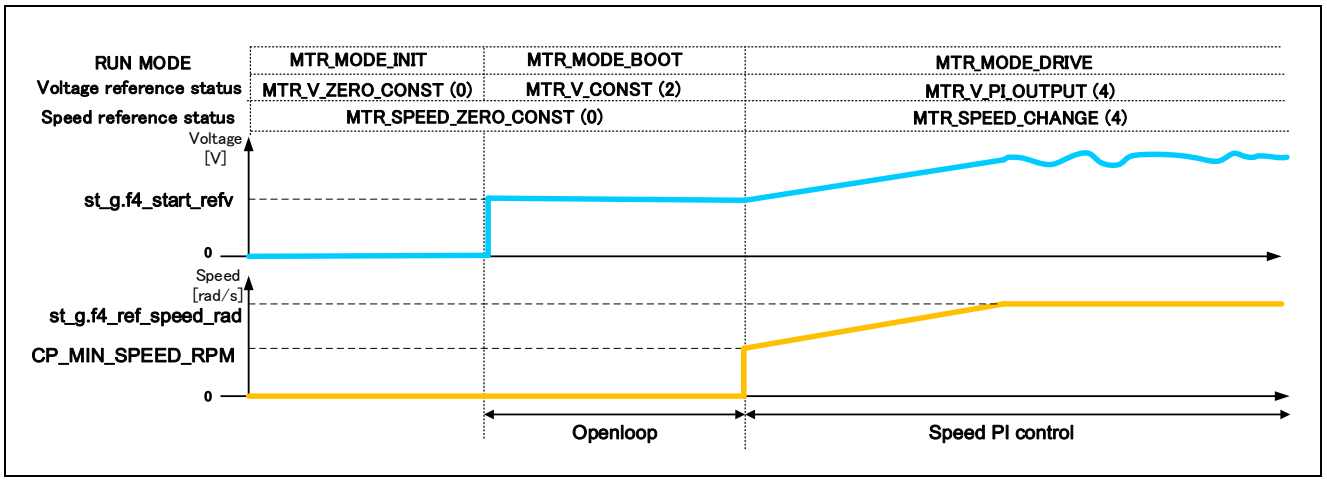


Figure 3-6 Start-up Method (Example)

3.1.7 System protection function

This system has the following types of error status and enables emergency stop functions in case of occurrence of respective error. Refer to Table 3-4 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.

- Low voltage error

The inverter bus voltage is monitored at the low voltage monitoring cycle. When a low voltage is detected (when the voltage falls below the limit value), CPU performs an emergency stop. The threshold value of the low voltage 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 hall interrupt detection

When the interrupt by hall sensors' signal doesn't occur during defined period, CPU performs an emergency stop.

- Hall sensor pattern (position information) error

When an error is detected in hall sensor patterns (position information) generated at hall interrupts, CPU performs an emergency stop.

Table 3-4 Setting Value of Each System Protection Function

Overcurrent error	Over current limit value [A]	2.0
	Monitoring cycle [μs]	50
Overvoltage error	Overvoltage limit value [V]	28
	Monitoring cycle [μs]	50
Undervoltage error	Under voltage limit value [V]	14
	Monitoring cycle [μs]	50
Rotation speed abnormality error	Speed limit value [rpm]	3000
	Monitoring cycle [μs]	50
Timeout error of hall interrupt detection	Timeout value [ms]	200

3.2 Function specifications of 120-degree conducting control using hall sensors software

Multiple control functions are used in this control program.

Table 3-5 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 ・Determination of rotation direction
	ics_ui Input: None Output: None	GUI user interface use ・Motor status change ・Determination of rotation speed command value ・Determination of rotation direction
	software_init Input: None Output: None	Initialization of variables used in the main process

Table 3-6 List of Functions “mtr_ctrl_rx23t.c”

File name	Function name	Process overview
mtr_ctrl_rx23t.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 3-7 List of Functions “mtr_ctrl_mrssk.c”

File name	Function name	Process overview
mtr_ctrl_mrssk.c	R_MTR_ChargeCapacitor Input: (uint8) u1_id / Motor ID Output: None	Wait for stability of the bus voltage
	get_vr1 Input: None Output: (uint16) A/D conversion result	VR1 status acquisition
	get_sw1 Input: None Output: (uint8) SW1 level	SW1 status acquisition
	get_sw2 Input: None Output: (uint8) 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 3-8 List of Functions “mtr_interrupt.c”

File name	Function name	Process overview
mtr_interrupt.c	mtr_hall_u_interrupt Input: None Output: None	Hall U signal interrupt function (IRQ0) ・Hall interrupt common function call
	mtr_hall_v_interrupt Input: None Output: None	Hall V signal interrupt function (IRQ1) ・Hall interrupt common function call
	mtr_hall_w_interrupt Input: None Output: None	Hall W signal interrupt function (IRQ4) ・Hall interrupt common function call
	mtr_hall_interrupt Input: None Output: None	Hall interrupt common function ・Count interrupt for start of speed measurement ・Timeout error check ・Motor stop wait process ・Drive pattern setting function call
	mtr_over_current_interrupt Input: None Output: None	Overcurrent detection process(Hard detection) ・ Event processing selection function call(Generate 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 at PWM cycle (MTU3) interrupt ・Current measurement and offset adjustment ・Inverter bus voltage capture ・Error check function call ・Motor stop detection function call ・Set GUI variables
	mtr_1ms_interrupt Input: None Output: None	Calling every 1 [ms] ・Run mode management (Calculate PI formula) ・Timeout error handling

Table 3-9 List of Functions “mtr_spm_hall_120.c” [1/2]

File name	Function name	Process overview
mtr_spm_hall_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 ・Enable hall interrupts
	mtr_act_inactive Input: (uint8)u1_state / motor status (uint8) u1_id / Motor ID Output: (uint8)u1_state / motor status	・PWM output disable ・Disable hall interrupts
	mtr_act_none Input: (uint8)u1_state / motor status (uint8) u1_id / Motor ID Output: (uint8)u1_state / motor status	No process is performed.
	mtr_act_reset Input: (uint8)u1_state / motor status (uint8) u1_id / Motor ID Output: (uint8)u1_state / motor status	Global variables 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_pattern_set Input: (MTR_ST_HALL120*)st_m / structure for Motor Output: None	・Set conduction pattern ・Call speed measurement process
	mtr_speed_calc Input: (MTR_ST_HALL120*)st_m / structure for Motor Output: None	Speed measurement calculation processing
	mtr_start_init Input: (uint8) u1_id / Motor ID Output: None	Initializing only the variables required for motor startup
	mtr_set_variables Input: None Output: None	Setting motor variables according to control layer
	R_MTR_IcsInput Input: (MTR_ICS_INPUT*)ics_input / structure for GUI Output: None	Setting GUI input value to the buffer
	mtr_watch_variables Input: None Output: None	Setting GUI output value to global variables
	mtr_error_check Input: None Output: None	Error monitoring
	mtr_wait_motorstop Input: (MTR_ST_HALL120*)st_m / structure for Motor Output: None	Check motor stop

Table 3-10 List of Functions “mtr_spm_hall_120.c” [2/2]

File name	Function overview	Processing overview
mtr_spm_hall_120.c	mtr_set_voltage_ref Input : (MTR_ST_HALL120*)st_m / structure for Motor Output : None	Set reference voltage
	mtr_set_speed_ref Input: (MTR_ST_HALL120*)st_m / structure for Motor Output :None	Set reference speed
	mtr_pattern_first60 Input : (MTR_ST_HALL120*)st_m / structure for Motor Output : None	Set voltage pattern non-complementary first 60 degree PWM
	mtr_pattern_first60_comp Input : (MTR_ST_HALL120*)st_m / structure for Motor Output : None	Set voltage pattern complementary first 60 degree PWM

Table 3-11 List of Functions "mtr_ctrl_rx23t_mrssk.c"

File name	Function name	Process overview
mtr_ctrl_rx23t_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
	mtr_init_irq Input: None Output: None	Initialization of external interrupt
	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_ current _uvw_adc Input : (float32*) f4_iu_ad / U phase current A/D conversion value (float32*) f4_iv_ad / V phase current A/D conversion value (float32*) f4_iw_ad / W phase current A/D conversion value (uint8) u1_id / Motor ID Output : None	Get u/v/w phase current A/D conversion value
	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_ctr l/ 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_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 Functions "mtr_gmc.c"

File name	Function name	Process overview
mtr_gmc.c(Note 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 PWM duty
	mtr_get_current_uvw Input : (volatile float32*) f4_iu_ad / U phase current A/D conversion value (volatile float32*) f4_iv_ad / V phase current A/D conversion value (volatile float32*) f4_iw_ad / W phase current A/D conversion value (uint8) u1_id / Motor ID Output: None	Obtaining the UVW phase current
	mtr_check_timeout_error Input : (float32) f4_cnt_timeout / counter of timeout calculation (float32) f4_timeout_limit / Timeout limit Output : (uint8) u2_temp0 / Flag of Timeout error	Checking time-out error

Note: 1. Functions which are not used in this system are undescribed

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 using hall sensors 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 valuable

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	
g_u1_hall_intr_cnt	uint8	Hall interrupt counts	Start speed measurement

3.4 List of 120-degree conducting control using hall sensors software structures

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

Table 3-17 List of structures [1/3]

	Member	Type	Content	Remarks
MTR_ST_HALL_120	u1_mode_system	uint8	State management	0x00: Inactive mode 0x01: Active mode 0x02: Error mode
	u2_run_mode	uint16	Operation mode management	0x00: Initialize mode 0x01: Boot mode 0x02: Drive mode 0x03: Analysis mode 0x04: Tune mode
	u2_error_status	uint16	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	Control 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
	f4_vdc_ad	float32	Inverter bus voltage A/D value	[V]
	f4_v_ref	float32	Voltage command value	Speed PI control output value [V]
	f4_start_ref_v	float32	Reference voltage for start-up	[V]
	s2_pwm_duty	int16	PWM duty	
	f4_ref_speed_rad	float32	Speed reference (user selected) value	Electrical angle [rad/s]
	f4_ref_speed_rad_crtl	float32	Speed command value	Electrical angle [rad/s]
	f4_speed_rad	float32	Measured speed value	Electrical angle [rad/s]
	f4_kp_speed	float32	Speed PI control proportional gain	
	f4_ki_speed	float32	Speed PI control integral gain	
	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	Increase step of speed command	[rad/s]

Table 3-18 List of structures [2/3]

	Member	Type	Content	Remarks
MTR_ST_HALL120	f4_ilim_v	float32	Limitation value for integral part of speed PI control	[V]
	u1_flg_wait_stop	uint8	Motor rotation stop waiting flag	
	u2_cnt_wait_stop	uint16	Motor rotation stop waiting counter	
	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 * 50 μ [s]
	f4_inv_offset_calc	float32	For offset calculation	Inverse number of u2_offset_calc_time
	u1_flag_offset_calc	unit8	Current offset value calculation flag	0: Start calculation 1: Finish calculation
	u2_cnt_adjust	uint16	Offset value calculation counter	
	u1_v_pattern	uint8	Conduction pattern	
	u1_flag_speed_ref	uint8	Speed state management	
	u1_flag_voltage_ref	uint8	Voltage state management	
	u1_direction	uint8	Rotation direction	0：CW 1：CCW
	u2_cnt_timeout	uint16	Timeout detection counter	
	u2_hall_timer_cnt	uint16	Free run timer count value	
	u2_pre_hall_timer_cnt	uint16	Previous free run timer count value	
	s4_timer_cnt_ave	int32	Average of speed measurement timer count	
	u2_timer_cnt_buf[6]	uint16	Speed measurement timer count buffer	
	u2_timer_cnt_num	uint16	Speed measurement timer count buffer number	
	u1_hall_signal	uint8	Hall sensors' signal capture buffer	
	u1_hall_wait_cnt	uint8	Waiting count of hall interrupts for speed measurement	
	st_speed	MTR_PI_CTRL	Structure for speed PI control	
	st_motor	MTR_PARAMETER	Motor parameter structure	
	f4_rpm_rad	float32	[rpm]→[rad/s]	2π/60*(POLE PAIRS)

Table 3-19 List of structures [3/3]

	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	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	
	u2_mtr_p	uint16	number of pole pairs	
	f4_limit_speed_change	float32	Speed command maximum increase limit	[rad/s]
	u2_offset_calc_time	uint16	Calculation time for current offset	
	f4_start_ref_v	float32	Reference voltage for start-up	
	u1_hall_wait_cnt	uint8	Hall interrupt counter for starting speed measurement	

3.5 Macro definitions of 120-degree conducting control using hall sensors software

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

Table 3-20 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]

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

File name	Macro name	Definition value	Remarks
control_parameter.h	CP_OFFSET_CALC_TIME	512	Counting number for current offset value
	CP_START_REF_V	5.8f	Voltage reference for BOOT mode
	CP_MAX_SPEED_RPM	2650	Rotation speed command maximum value (mechanical angle) [rpm]
	CP_MIN_SPEED_RPM	550	Rotation speed command minimum value(mechanical angle) [rpm]
	CP_LIMIT_SPEED_CHANGE	0.2f	Speed command maximum increase limit[rad/s]
	CP_SPEED_PI_KP	0.02f	Proportional gain
	CP_SPEED_PI_KI	0.0005f	Integral gain
	CP_SPEED_LPF_K	1.0f	Speed LPF parameter
	MTR_FIRST60	1	Non-Complementary First 60 degree PWM(default)
	MTR_FIRST60_COMP	0	Complementary First 60 degree PWM

Table 3-22 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
	START_REF_V	CP_START_REF_V	Voltage reference for BOOT mode
	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]
	LIMIT_SPEED_CHANGE	CP_ LIMIT_SPEED_CHANGE	Speed command maximum increase limit[rad/s]
	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]
	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 3-23 List of Macro definitions “mtr_ctrl_rx23t_mrssk.h”

File name	Macro name	Definition value	Remarks
mtr_ctrl_rx23t_mrrsk.h	MTR_PWM_TIMER_FREQ	40.0f	PWM timer count frequency [MHz]
	MTR_CARRIER_FREQ	20.0f	Carrier frequency [kHz]
	MTR_DEADTIME	2	Dead time [μs]
	MTR_DEADTIME_SET	(uint16)(MTR_DEADTIME MTR_PWM_TIMER_FREQ)	Dead time setting value
	MTR_AD_FREQ	40.0f	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	(uint16) ((MTR_PWM_TIMER_FREQ * 1000 /MTR_CARRIER_FREQ / 2)+MTR_DEADTIME_SET)	Carrier setting value
	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)	MTR_CARRIER_SET - MTR_DEADTIME_SET
	MTR_PORT_HALL_U	PORT9.PODR.BIT.B3	Hall sensor signal U input
	MTR_PORT_HALL_V	PORT9.PODR.BIT.B4	Hall sensor signal V input
	MTR_PORT_HALL_W	PORTA.PODR.BIT.B2	Hall sensor signal W input
	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	PORT9.PIDR.BIT.B1	SW1 input port
	MTR_PORT_SW2	PORT9.PIDR.BIT.B2	SW2 input port
	MTR_PORT_LED1	PORT0.PODR.BIT.B0	LED1 output port
	MTR_PORT_LED2	PORT0.PODR.BIT.B1	LED2 output port
	MTR_PORT_LED3	PORT3.PODR.BIT.B1	LED3 output port
	MTR_LED_ON	0	LED light 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
	MTR_CURRENT_SCALING	20.0f/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_LIMIT	2.0f	High current limit value [V]
	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 measurement

Table 3-24 List of Macro definitions "mtr_spm_hall_120.h"[1/4]

File name	Macro name	Definition value	Remarks
mtr_spm_hall_120.h	MTR_INT_DECIMATION	0	Number of interrupt decimation times
	MTR_POLE_PAIRS	MP_POLE_PAIRS	Motor Pole pairs
	MTR_TWOPI	2 * 3.14159265f	2π
	MTR_RPM_RAD	MTR_TWOPI / 60 * MTR_POLE_PAIRS	[rpm]→[rad/s]
	MTR_SPEED_LIMIT_RPM	3000	Speed limit value (mechanical angle) [rpm]
	MTR_SPEED_LIMIT	MTR_SPEED_LIMIT_RPM * MTR_RPM_RAD	Speed limit value (electrical angle) [rad/s]
	MTR_SPEED_PI_DECIMATION	0	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 * 5000000	Constant for speed measurement
	MTR_SPEED_LPF_K	CP_SPEED_LPF_K	Speed LPF parameter
	MTR_LIMIT_SPEED_CHANGE	CP_LIMIT_SPEED_CHANGE	Speed command maximum increase limit[rad/s]
	MTR_MAX_DRIVE_V	20.0f	Maximum command voltage [V]
	MTR_MIN_DRIVE_V	3.0f	Minimum command voltage [V]
	MTR_START_REFV	CP_START_REF_V	Voltage reference for BOOT mode [V]
	MTR_TIMEOUT_CNT	200	Timeout count limit [ms]
	MTR_STOP_WAIT_CNT	1000	Stop judge count (*50[μs])
	MTR_WAIT_SPEED_CALC	12	Wait speed measurement until hall interrupts become this counts

Table 3-25 List of Macro definitions "mtr_spm_hall_120.h"[2/4]

File name	Macro name	Definition value	Remarks
mtr_spm_hall_120.h	MTR_PATTERN_CW_V_U	2	CW 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	5	CCW hall sensor value
	MTR_PATTERN_CCW_V_W	1	
	MTR_PATTERN_CCW_U_W	3	
	MTR_PATTERN_CCW_U_V	2	
	MTR_PATTERN_CCW_W_V	6	
	MTR_PATTERN_CCW_W_U	4	
	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_OFFSET_CALC_TIME	CP_OFFSET_CALC_TIME	Current offset value calculation time [ms]

Table 3-26 List of Macro definitions "mtr_spm_hall_120.h"[3/4]

File name	Macro name	Definition value	Remarks
mtr_spm_hall_120.h	MTR_CW	0	Rotation direction setting value
	MTR_CCW	1	
	MTR_FLG_CLR	0	Constant for flag management
	MTR_FLG_SET	1	
	MTR_ID_A	0	Motor ID A
	MTR_ID_B	1	Motor ID B
	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 (not used)
	MTR_V_CONST	2	Voltage constant
	MTR_V_OPENLOOP	3	Open-loop voltage setting mode (not used)
	MTR_V_PI_OUTPUT	4	Speed PI output voltage setting mode
	MTR_SPEED_ZERO_CONST	0	Zero speed constant
	MTR_SPEED_OPENLOOP_1	1	Open loop MODE1 (not used)
	MTR_SPEED_OPENLOOP_2	2	Open loop MODE2 (not used)
	MTR_SPEED_OPENLOOP_3	3	Open loop MODE3 (not used)
	MTR_SPEED_CHANGE	4	Speed changing
	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	Initialize 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
	MTR_CONTROL_VOLTAGE	0x10	Voltage control

Table 3-27 List of Macro definitions “mtr_spm_hall_120.h”[4/4]

File name	Macro name	Definition value	Remarks
mtr_spm_hall_120.h	MTR_ERROR_NONE	0x00	None 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

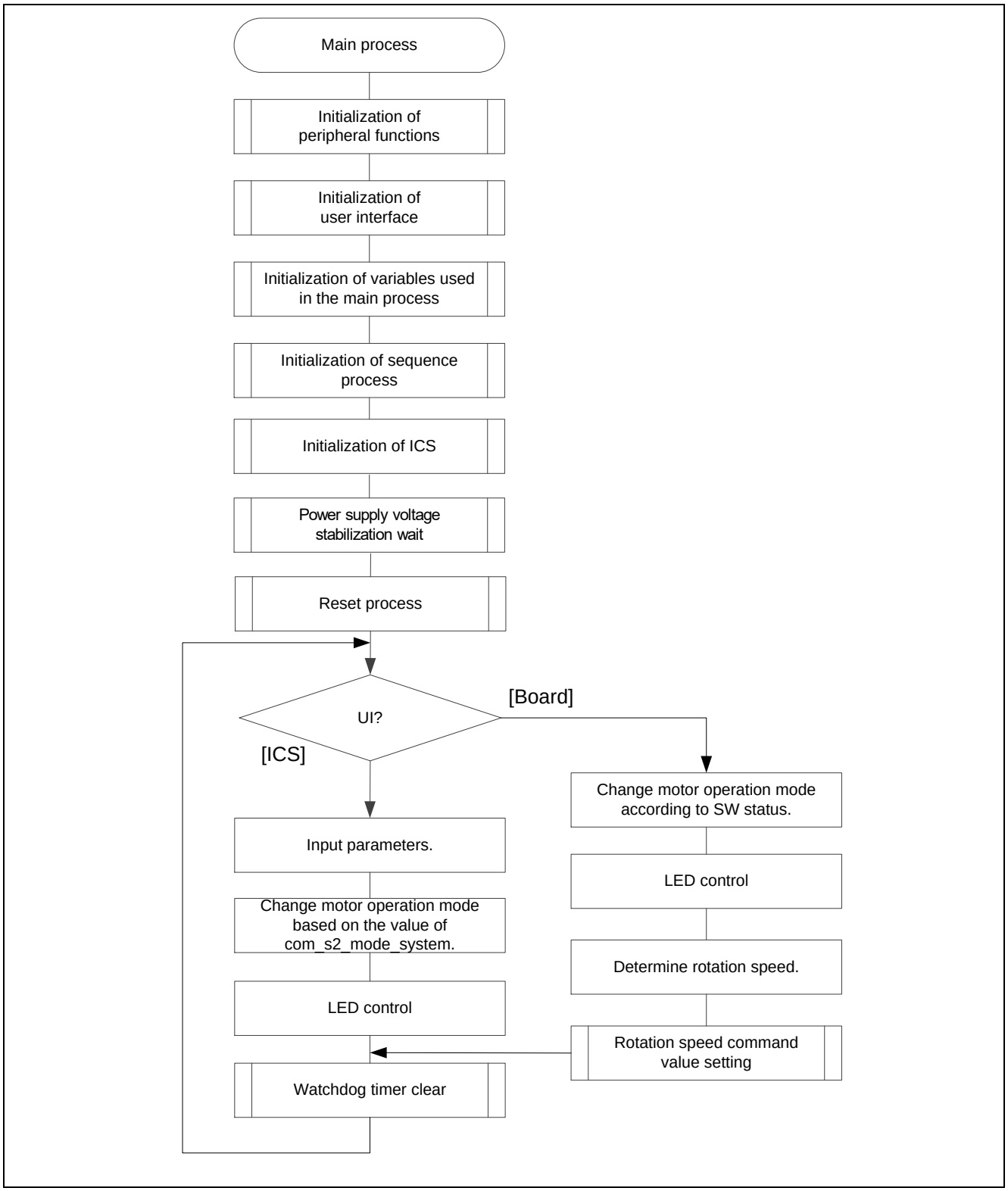


Figure 3-7 Main Process Flowchart

3.6.2 Carrier cycle interrupt handling

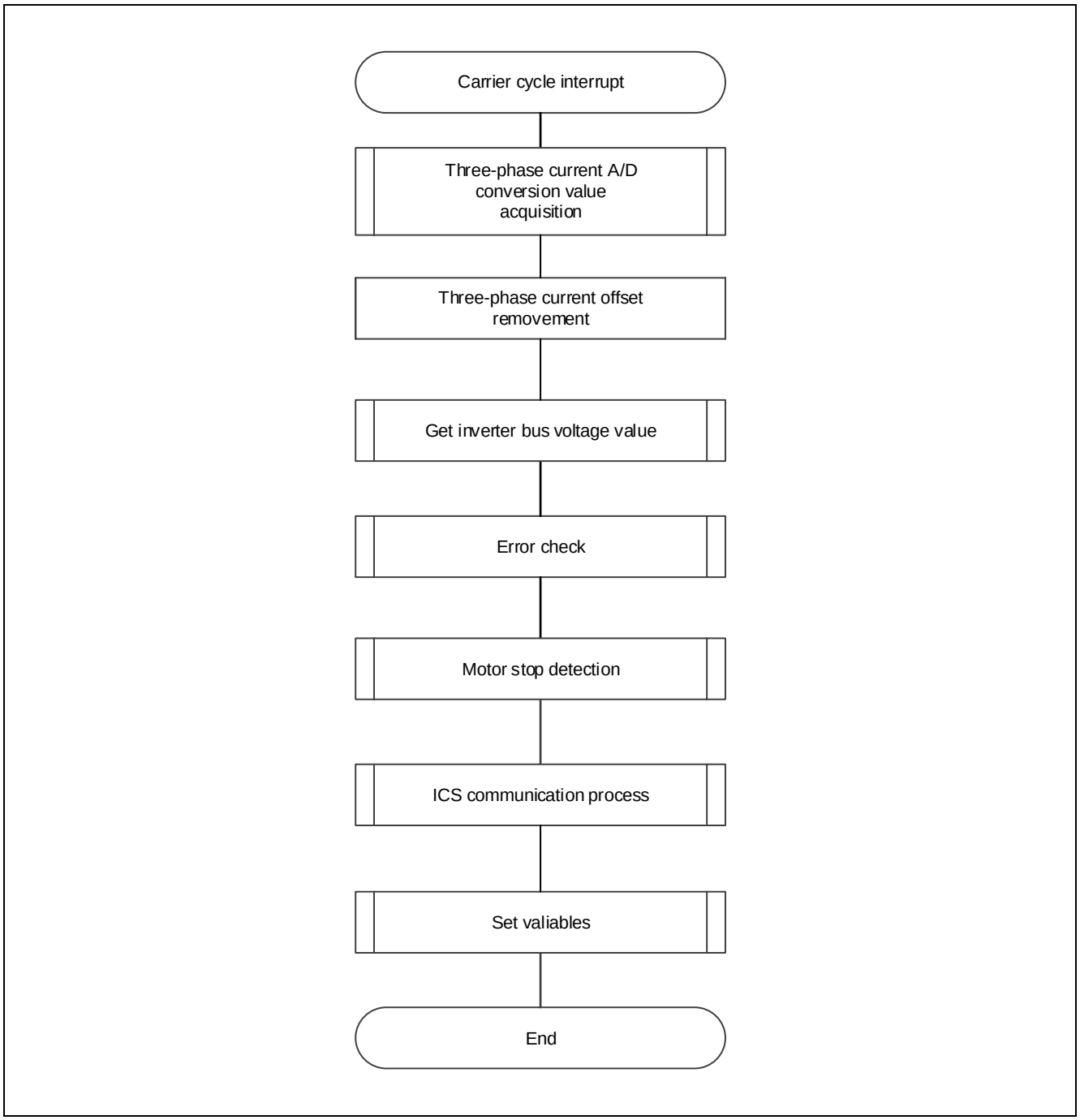


Figure 3-8 Carrier Cycle Interrupt Handling (120-degree Control using hall sensors)

3.6.3 1 msec interrupt handling

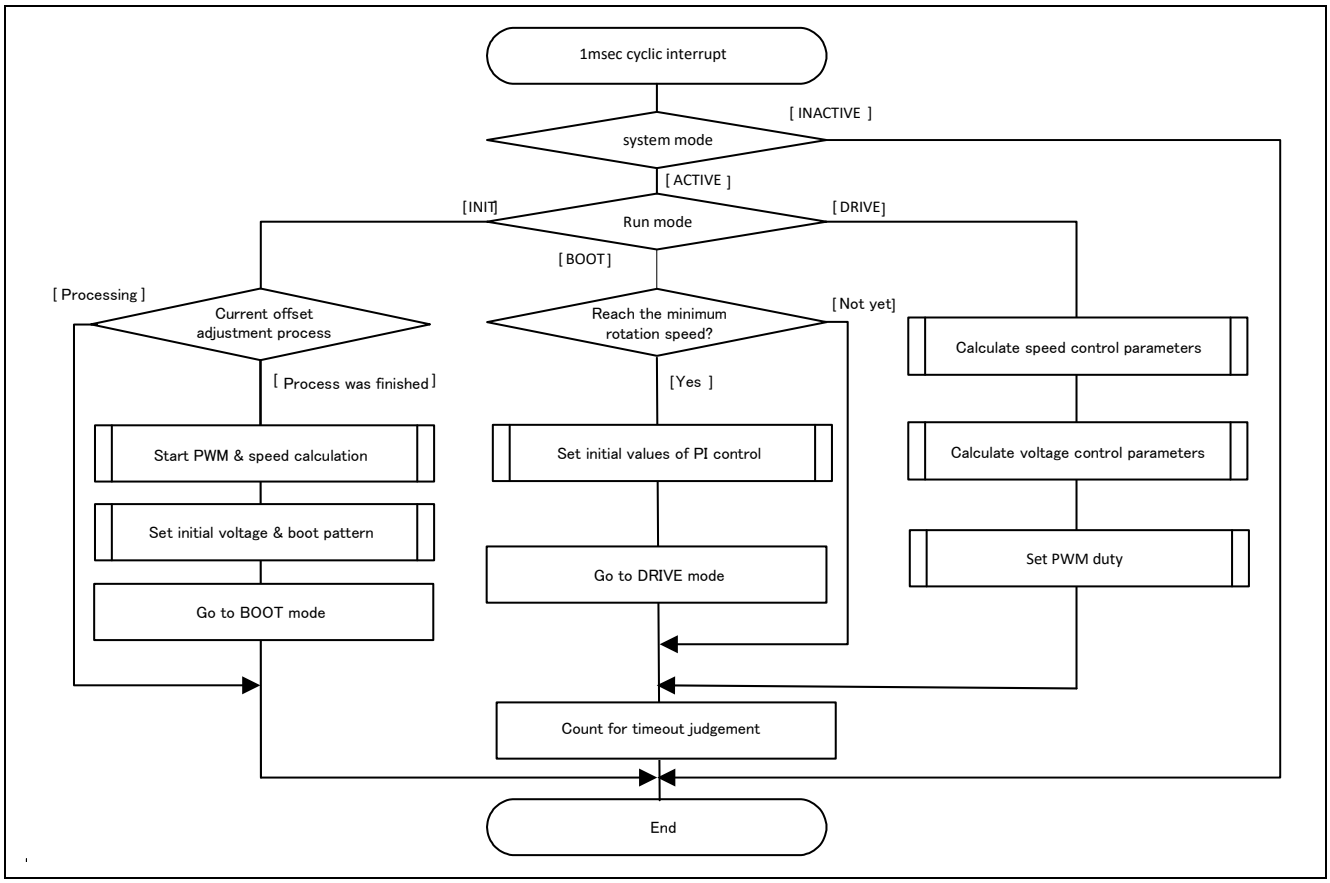


Figure 3-9 1 [ms] Interrupt Handling

3.6.4 Overcurrent interrupt handling

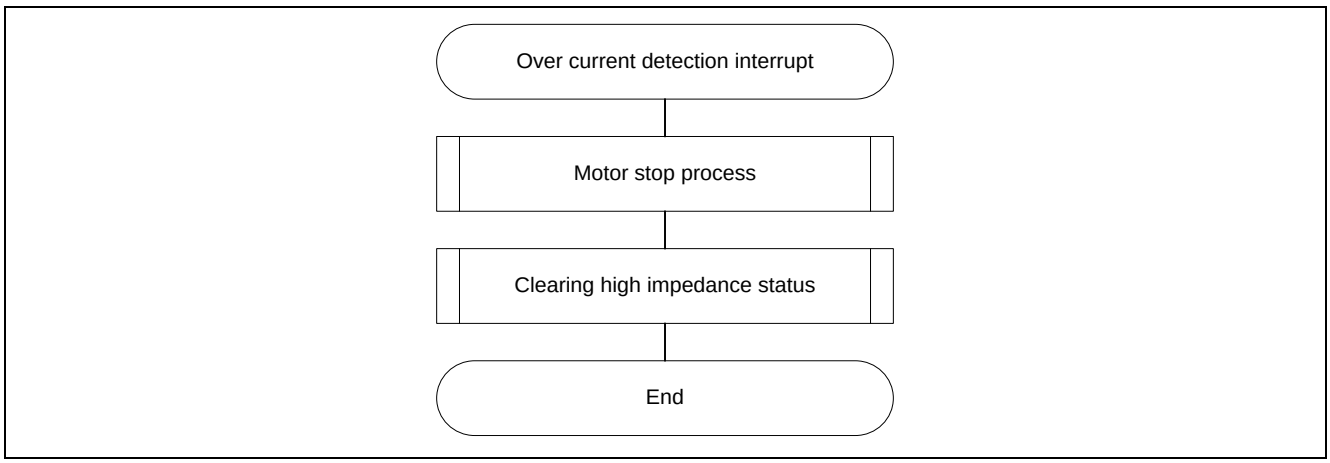


Figure 3-10 Over Current Detection Interrupt Handling

3.6.5 Hall sensors' signal interrupt handling (common process)

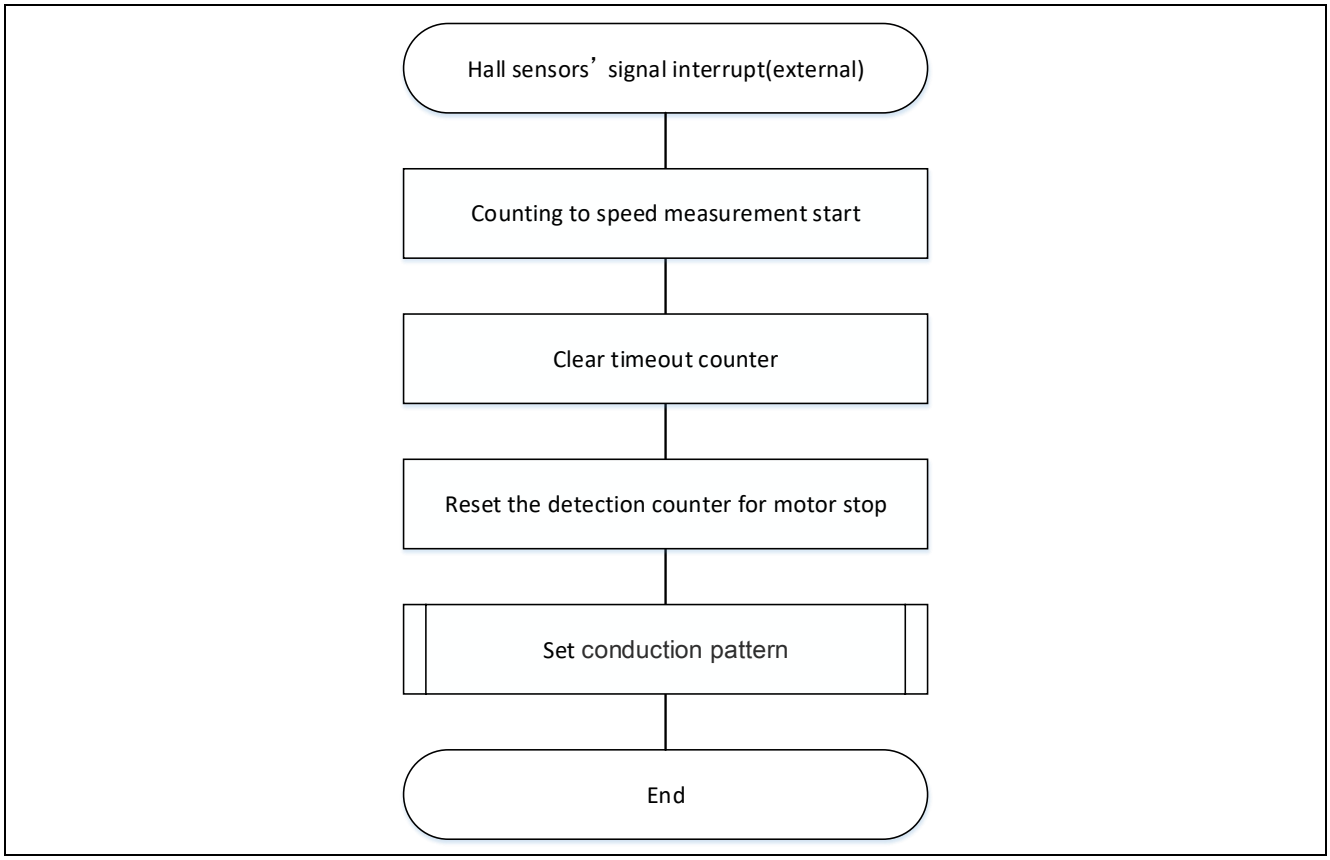


Figure 3-11 Hall sensors' signal Interrupt Handling (common process)

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.) 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.

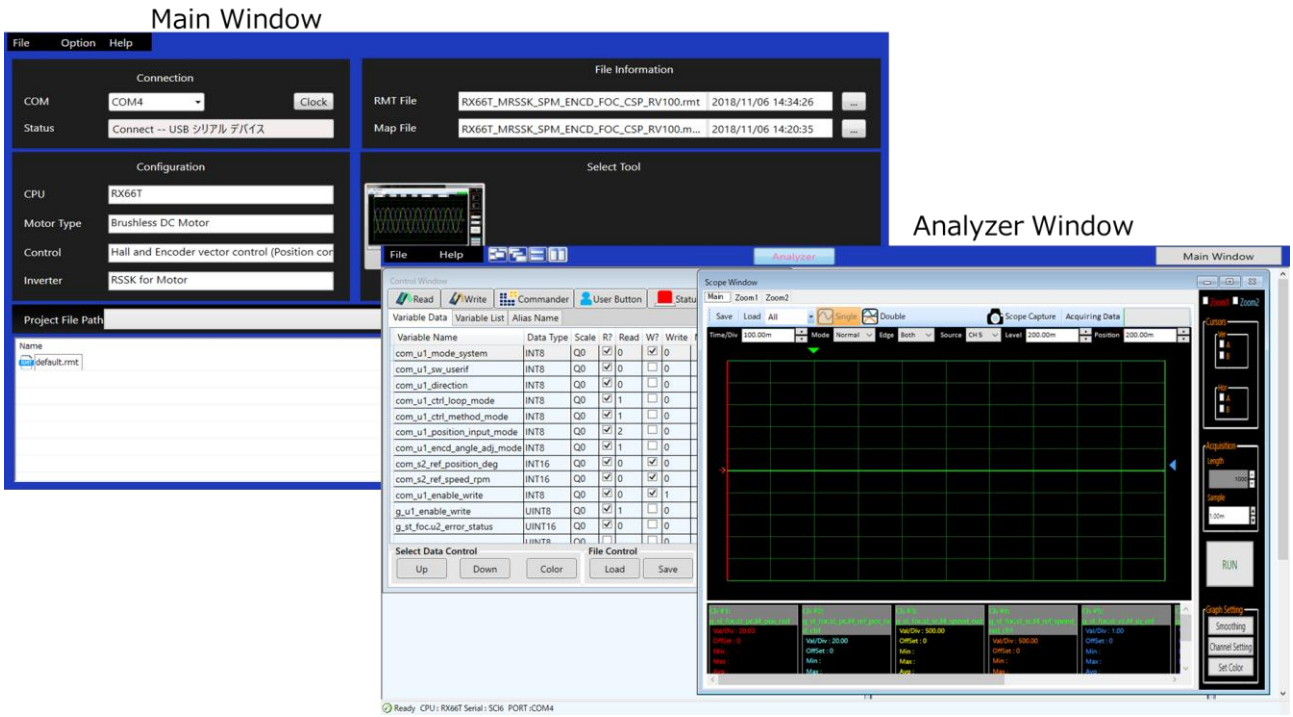


Figure 4-1 Renesas Motor Workbench – Appearance

Set up for Motor control development support tool

(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 value as g_s2_enable_write is 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	【st_g.s2_direction】
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_u2_offset_calc_time	uint16	Current offset value calculation time [ms]	【st_g.u2_offset_calc_time】
com_f4_start_ref_v	float32	Voltage command value of start-up	【st_g.f4_boot_ref_v】
com_u2_mtr_p	uint16	Number of pole pairs	【st_g.u2_mtr_p】
com_u1_hall_wait_cnt	uint8	Wait hall counts to starting speed measurement	【st_g.u1_hall_wait_cnt】
com_s2_enable_write	int16	Enable to rewriting variables	

4.3 Operation Example for Analyzer

Show an example below that motor driving operation using Analyzer. Operation is using “Control Window”. Regarding specification of ‘Control Window’, refer to ‘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_ref_speed_rpm”.
  - (6) Type a value of “1” in the [Write] box of “com_s2_mode_system”.
  - (7) Click the “Write” button.

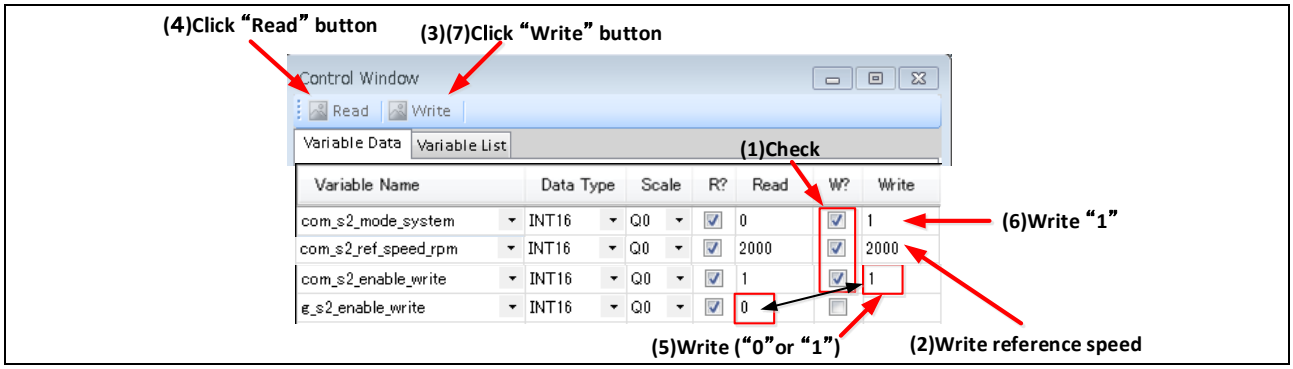


Figure 4-2 Procedure - Driving the motor

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

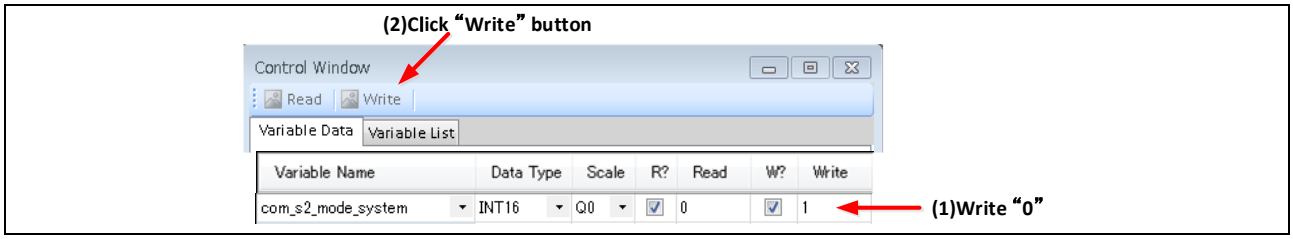


Figure 4-3 Procedure - Stop the motor

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

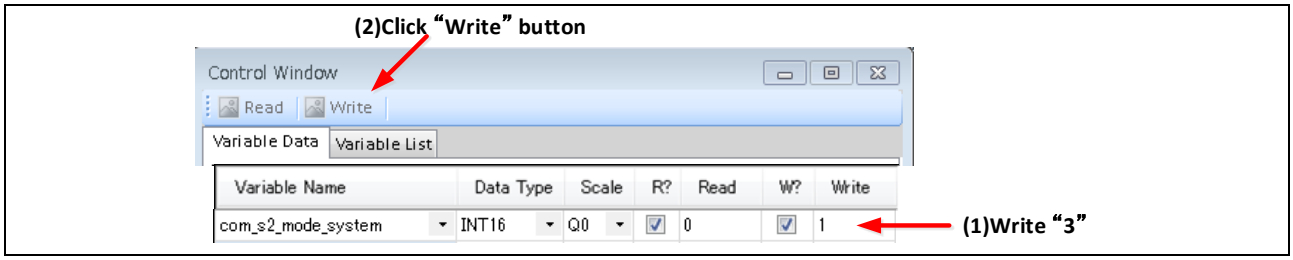


Figure 4-4 Procedure - Error cancel operation

Website and Support

Renesas Electronics Website

http://www.renesas.com/

Inquiries

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

Revision Record

Rev.	Date of issue	Descriptions	
		Page	Summary
1.00	2016.05.31	–	First edition issued
1.01	2018.12.17	–	Issued in accordance with ‘Renesas Motor Workbench 2.0’.

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. 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the 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.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.0