R01AN4028EJ0100
Rev.1.00
Oct.2.2017

RL78/G14

Speed control of 120-degree conducting controlled 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 using the RL78/G14 microcontroller.

These control 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 control programs have been checked by using the following device.

- RL78/G14 (R5F104LEAFB)

Target control programs

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

(1) RL78G14_MRSSK_SPM_HALL_120_CSP_CA_V100 (IDE: CS+ for CA, CX)

(2) RL78G14_MRSSK_SPM_HALL_120_CSP_CC_V100 (IDE: CS+ for CC)

(3) RL78G14_MRSSK_SPM_HALL_120_E2S_CC_V100 (IDE: e^2studio)

RL78/G14 120-degree conducting control using Hall sensors sample program for

RL78/G14 24V Motor Control Evaluation System

Reference

- RL78/G14 Group User's Manual: Hardware (R01UH0186EJ0330)
- Application note: '120-degree conducting control of permanent magnetic synchronous motor: algorithm' (R01AN2657EJ0120)
- Renesas Motor Workbench V.1.00 User's Manual (R21UZ0004EJ0100)
- Renesas Solution Starter Kit 24V Motor Control Evaluation System for RX23T User's Manual (R20UT3697EJ0110)
- RL78/G14 CPU card User's Manual (R12UZ0023EJ0100)

Contents

1. Overview

This application note explains how to implement the 120-degree conducting control programs of permanent magnetic synchronous motor (PMSM) using Hall sensors based on the RL78/G14 microcontroller.

Note that these control 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 environments of the control programs explained in this application note.

Table 1-1 Development Environment (H/W)

Microcontroller	Evaluation board	Motor
RL78/G14 (R5F104LEAFB)	24V inverter board (Note 1) RL78/G14 CPU Card (Note 2)	TG-55L-KA(Note 3)

Table 1-2 Development Environment (S/W)

CS+ version	Build tool version
V4.00.00	CA78K0R V5.00.00.03
V6.00.00	CC-RL V1.04.00.00

e^2studio version	Build tool version
5.4.0.018	CC-RL V1.04.00.00

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

Notes: 1. 24V inverter board (RTK0EM0001B0012BJ) is a product of Renesas Electronics Corporation.

2. RL78/G14 CPU Card (RTK0EML130C06000BJ) is a product of Renesas Electronics Corporation.

3. TG-55L-KA is a product of TSUKASA ELECTRIC.
TSUKASA ELECTRIC. (https://www.tsukasa-d.co.jp/en/)

2. System overview

Overview of this system is explained below.

2.1 Hardware configuration

The hardware configuration is shown below.

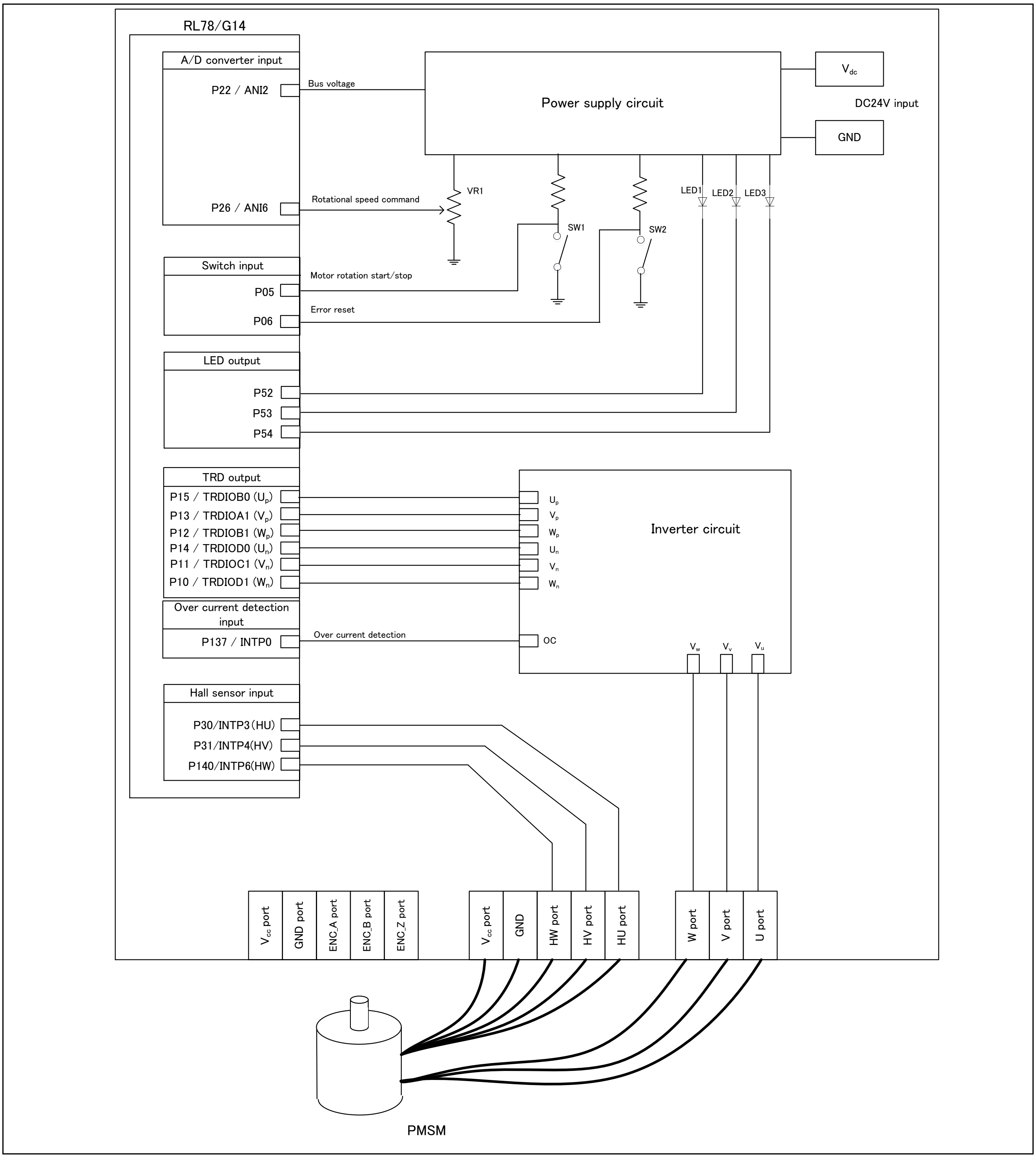


Figure 2-1 Hardware Configuration Diagram

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
Rotational speed	Variable resistance (VR1)	Rotational 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
RESET	Push switch (RESET1)	System reset

Table 2-2 is a list of port interfaces of RL78/G14 microcontroller of this system.

Table 2-2 Port Interface

R5F104LEAFB Port name	Function
P22 / ANI2	Inverter bus voltage measurement
P26 / ANI6	For inputting rotational speed command values (analog values)
P05	START / STOP toggle switch
P06	ERROR RESET toggle switch
P52	LED1 ON / OFF control
P53	LED2 ON / OFF control
P54	LED3 ON / OFF control
P30 / INTP3	Hall sensor input (HU)
P31 / INTP4	Hall sensor input (HV)
P140 / INTP6	Hall sensor input (HW)
P10 / TRDIOD1	PORT output / PWM output (W_n)
P11 / TRDIOC1	PORT output / PWM output (V_n)
P12 / TRDIOB1	PORT output / PWM output (W_p)
P13 / TRDIOA1	PORT output / PWM output (V_p)
P14 / TRDIOD0	PORT output / PWM output (U_n)
P15 / TRDIOB0	PORT output / PWM output (U_p)
P137 / INTP0	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

Peripheral Function	Purpose
A/D converter	- Rotational speed command value input - Inverter bus voltage measurement
Timer Array Unit (TAU)	- 1 [ms] interval timer - Free-running timer for rotational speed measurement
Timer RD (TRD)	Complementary PWM output
External interrup (INTP3, INTP4, INTP6)	- Hall sensors' signal input (detection of rotor position) - Edge detection for speed measurement and change of control signal (both edge)
External interrupt (INTP0)	Overcurrent detection

(1) A/D converter

The rotational speed command value input, and inverter bus voltage (Vdc) are measured by using the 'A/D converter'.

The operation mode is set as below.

The channel selection mode: the select-mode.
The conversion operation mode: the one-shot conversion mode.

And software trigger is used.

(2) Timer Array Unit (TAU)

a. 1 [ms] interval timer

The channel 0 of Timer Array Unit (TAU) is used as 1 millisecond interval timer.

b. Free-running timer for measuring speed

The channel 1 of Timer Array Unit (TAU) is used as free-running timer for speed measurement.
Note that interrupt is not used.

(3) Timer RD (TRD)

Three-phase PWM output of chopping at the first 60 degrees with dead time (complementary) or without dead time (non-complementary) is performed using the Complementary PWM Mode. When detecting an overcurrent, the PWM output ports are set to high impedance output using the pulse output forced cutoff function.

(4) External interrupt (INTP3, INTP4, INTP6)

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

(5) External interrupt (INTP0)

An overcurrent is detected by an external circuit.

2.3 Software structure

2.3.1 Software file structure

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

Table 2-4 Folder and File Configuration

Project	Folder	File	Content
(1) RL78G14_MRSSK_SPM_HALL_120_CSP_CA_V100 (2) RL78G14_MRSSK_SPM_HALL_120_CSP_CC_V100 (3) RL78G14_MRSSK_SPM_HALL_120_E2S_CC_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_rl78g14.h	RL78/G14 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_rl78g14_mrssk.h	RL78/G14 and board dependent processing part header
		mtr_feedback.h	Feedback control processing part header
		mtr_gmc.h	General motor control function part header
		mtr_driver_access.h	Driver access function part header
		mtr_filter.h	Filters processing part header (not used)
	ics	Ics2_RL78G14_LE.lib	Library for GUI
		Ics2_RL78G14_Lx.h	Header for GUI
		RL78_vector.c	Interrupt processing part for GUI interface.
	prj	RL78G14_MRSSK_SPM_HALL_120_CSP_CA_V100.dr	Link directive file (Note1)
	lib	R_dsp_rl78_CA.lib	Digital signal controller library for CA tool-chain (Note2)
		R_dsp_rl78_CC.lib	Digital signal controller library for CC-RL tool-chain (Note2)
	src	main.c	Main function, user interface control
		mtr_ctrl_mrssk.c	Board dependent processing part
		mtr_ctrl_rl78g14.c	RL78/G14 dependent processing part
		mtr_interrupt.c	Interrupt handler
		mtr_spm_hall_120.c	120-degree conducting control using Hall sensors dependent part
		mtr_ctrl_rl78g14_mrssk.c	RL78/G14 and board dependent processing part
		mtr_feedback.c	Feedback control processing
		mtr_gmc.c	General motor control function
		mtr_driver_access.c	Driver access function
		mtr_filter.c	Filters processing (not used)

Notes: 1. Link directive file is included only in RL78G14_MRSSK_SPM_HALL_120_CSP_CA_V100.
2. R_dsp_rl78_CA.lib is included only in RL78G14_MRSSK_SPM_HALL_120_CSP_CA_V100.
R_dsp_rl78_CC.lib is included in RL78G14_MRSSK_SPM_HALL_120_CSP_CC_V100 and RL78G14_MRSSK_SPM_HALL_120_E2S_CC_V100.

2.3.2 Module configuration

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

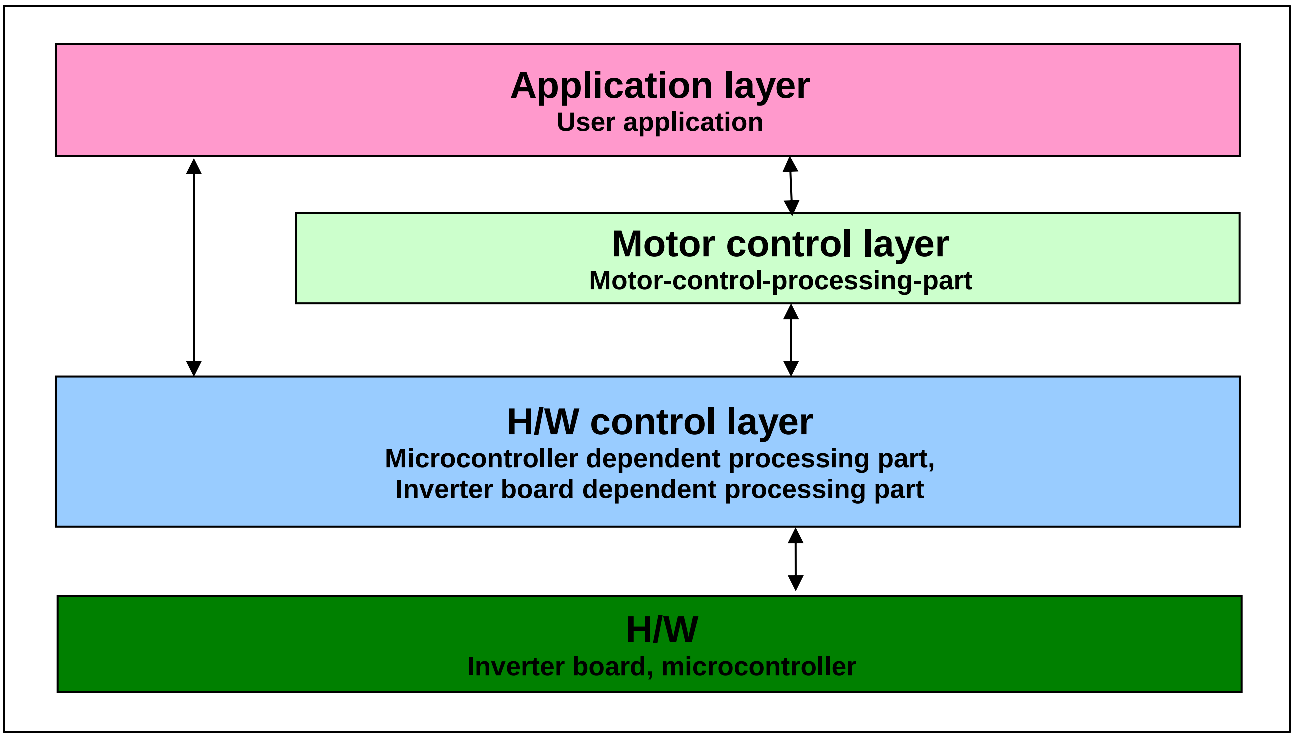


Figure 2-2 Module Configuration

Table 2-5 Module Configuration

Layers	**File name**
Application layer	main.c
Motor control layer	mtr_spm_hall_120.c mtr_feedback.c mtr_gmc.c mtr_driver_access.c mtr_interrupt.c (Note) mtr_filter.c
H/W control layer	mtr_ctrl_rl78g14_mrssk.c mtr_ctrl_rl78g14.c mtr_ctrl_mrssk.c mtr_interrupt.c (Note)

Note: "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) (Complementary / Non-Complementary)	
Motor rotation start / stop	Determined depending on the level of SW1 (P05) ("Low": rotation start "High": stop) or input from Motor Control Development Support Tool. (Note)	
Position detection of rotor magnetic pole	Position detection by signals of Hall sensors (by each 60 degrees)	
Input voltage	DC24 [V]	
Main clock frequency	CPU clock: f_{CLK} 32 [MHz] TRD clock: f_{HOCO} 64 [MHz]	
Carrier frequency (PWM)	20 [kHz]	
Dead time	2 [μs]	
Control cycle	- The conduction pattern is changed at a Hall signal interrupt timing. - A duty of PWM and a conduction pattern is determined at a pattern change. - Speed PI control is performed every 1 [ms].	
Rotational speed control range	550 [rpm] to 2650 [rpm] Both CW and CCW are supported	
Optimization	CA	Standard optimization
	CC-RL	Perform Default optimization
ROM / RAM size	CA	ROM : 9.09 KB / RAM : 0.51 KB
	CC-RL	ROM : 8.20 KB / RAM : 0.52 KB
Processing stop for protection	- Disables the motor control signal output (six outputs), under any of the following conditions. 1. Inverter bus voltage exceeds 28 [V] (monitored per 1 [ms]) 2. Inverter bus voltage is less than 15 [V] (monitored per 1 [ms]) 3. Rotational speed exceeds 3500 [rpm] (monitored per 1 [ms]) 4. When the motor rotates, the interrupt of Hall sensors' signals are not detected for 500 [ms]. 5. Fault detection of Hall sensor pattern (position information) - The ports executing PWM output are set to high impedance state when an overcurrent is detected by external circuit (low level edge input occurs in INTP0 port).	

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

3. Descriptions of the control program

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

3.1 Contents of control

3.1.1 Motor start / stop

The start and stop of the motor are controlled by input from Motor Control Development Support Tool 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 rotational 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 rotational speed command value

The motor rotational speed command value can be set by A/D conversion of the VR1 output value (analog value). The A/D converted VR1 value is used as rotational speed command value, as shown below.

The maximum of the command value is set as the value from which maximum rotational speed is generated by the resolution of the A/D converter.

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

Item	Conversion ratio (Command value: A/D conversion value)		Channel
Rotational speed command value	CW	0 [rpm] ~ 3072 [rpm]：0200H～03FFH	ANI6
	CCW	-3072 [rpm] ~ 0 [rpm]：0000H～01FFH	

(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 error 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] ~ 111[V]: 0000H ~ 03FFH	ANI2

For more details of A/D conversion characteristics, refer to RL78/G14 User’s Manual: Hardware.

3.1.3 Speed control

In this system, the motor rotational 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 Timer Array Unit (TAU) channel 1 performed free running.

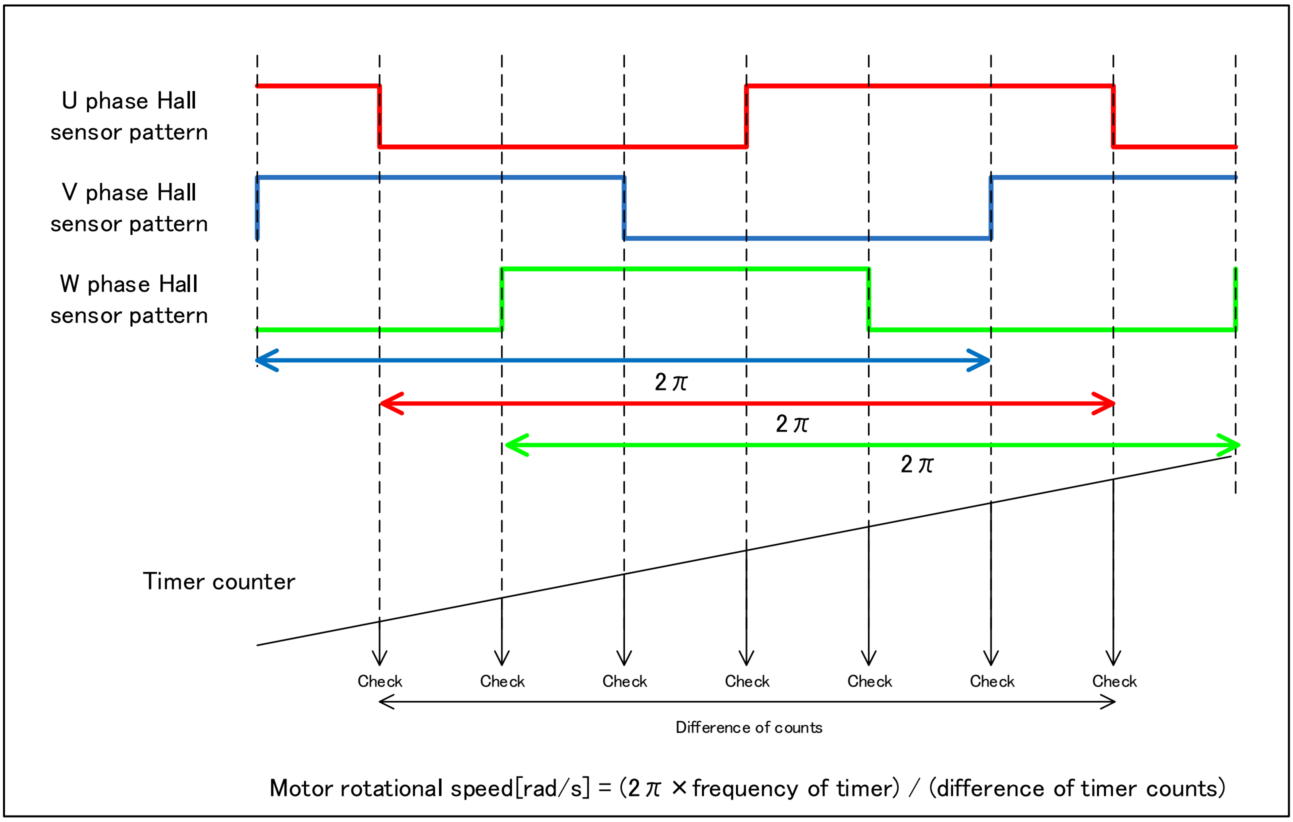


Figure 3-1 Method of Calculation for the Rotational Speed

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: Rotational 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.

$$\text{Duty} = \frac{T_{ON}}{T_{ON} + T_{OFF}} \times 100\ [\%]$$

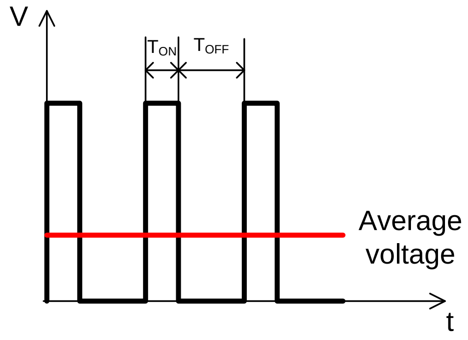


Figure 3-2 PWM Control

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.

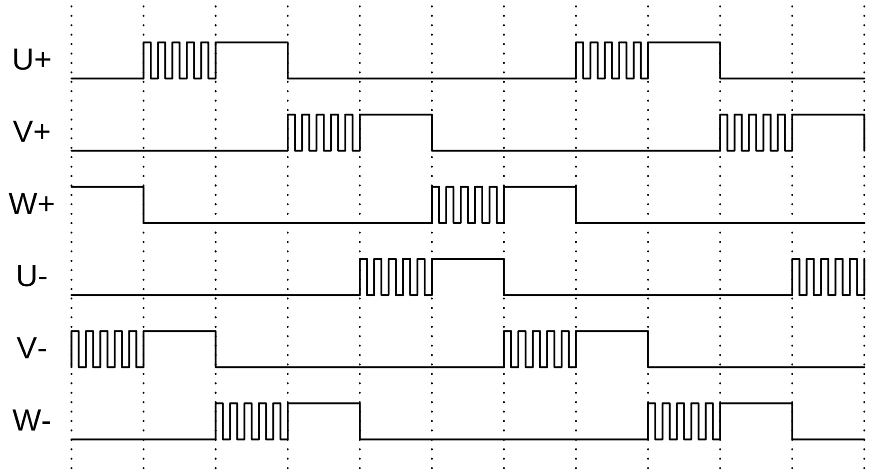


Figure 3-3 Non-complimentary first-60-degree Chopping

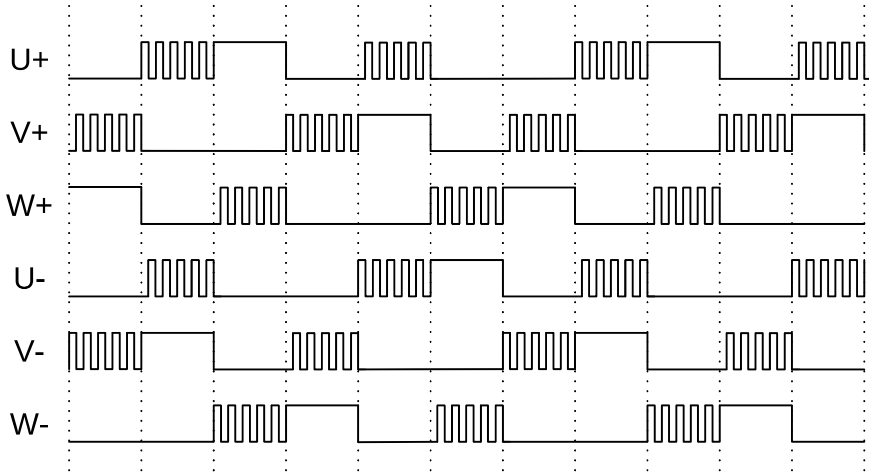
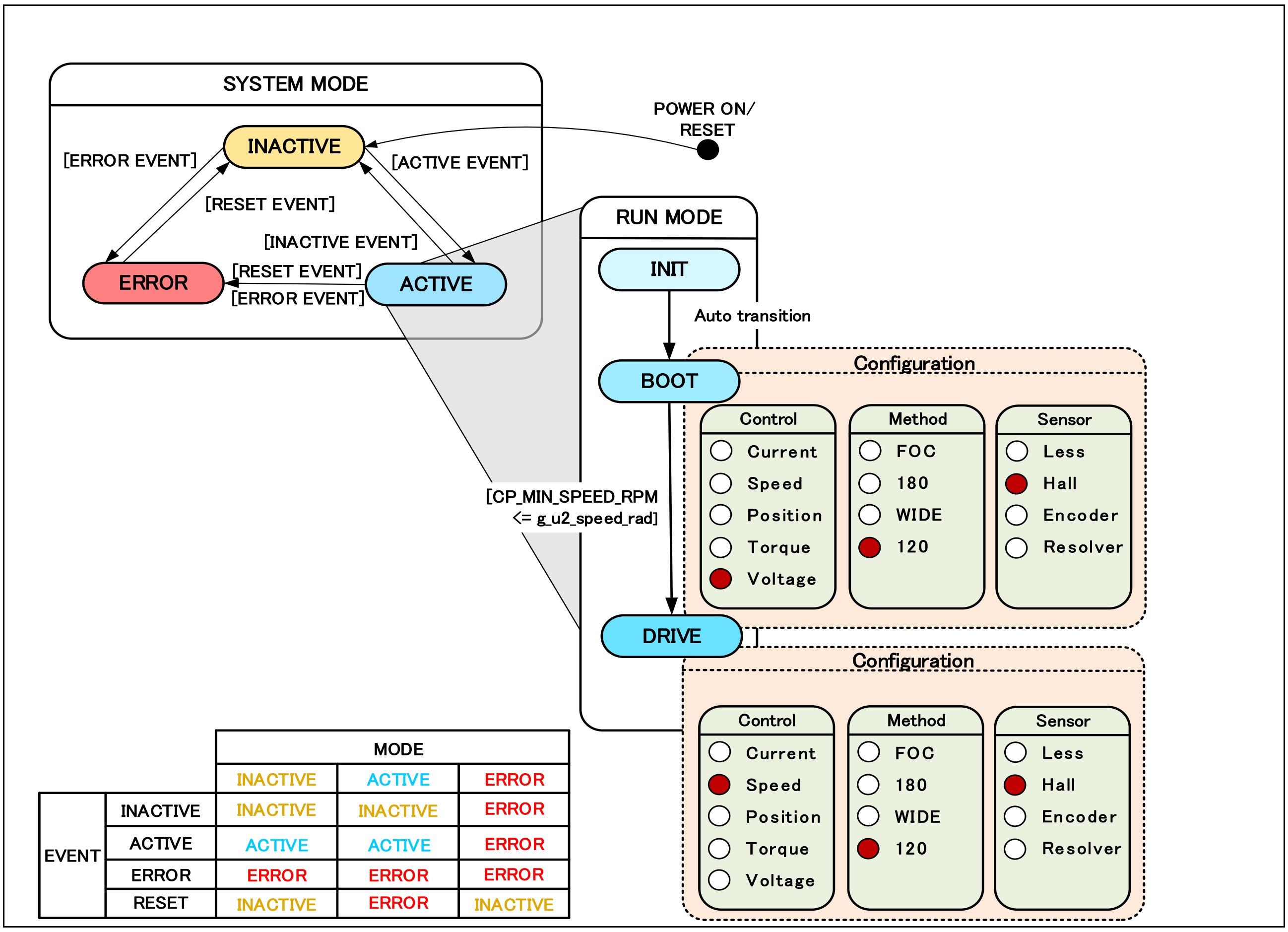


Figure 3-4 Complimentary first-60-degree Chopping

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

(1) SYSTEM MODE

"SYSTEM MODE" indicates the operating states of the system. The state transits on occurrence of each event (EVENT). "SYSTEM MODE" has 3 states that are motor drive stop (INACTIVE), motor drive (ACTIVE), and abnormal condition (ERROR).

(2) RUN MODE

"RUN MODE" indicates the condition of the motor control. "RUN MODE" transits sequentially as shown in Figure 3-5 when "SYSTEM MODE" is "ACTIVE".

(3) EVENT

When "EVENT" occurs in each "SYSTEM MODE", "SYSTEM MODE" changes as shown table in Figure 3-5 , per that "EVENT".

Table 3-3 List of EVENT

EVENT name	Occurrence factor
INACTIVE	by user operation
ACTIVE	by user operation
ERROR	when the system detects an error
RESET	by user operation

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.

When the control will be changed to PI control, at least the speed data is necessary to reach 2π (refer to 3.1.3). In the sample softwares, at the start-up of rotation the motor is controlled in open loop with a constant voltage until the speed data reach 2π.

Figure 3-6 shows the start-up method in the sample softwares. In "MTR_MODE_BOOT", open loop with a constant voltage which is set by g_s2_start_ref_v is performed. The mode changes to "MTR_MODE_DRIVE" when the current speed reaches the defined minimum speed.

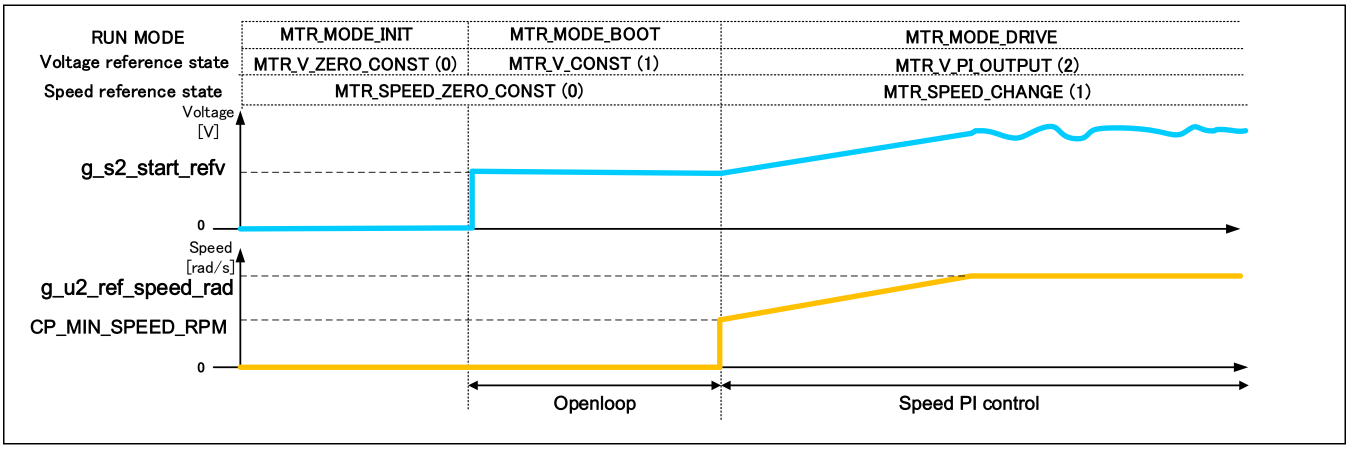


Figure 3-6 Start-up Method (Example)

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 エラー! 参照元が見つかりません。 for settings.

- Overcurrent error

High impedance output is made to the PWM output port in response to an emergency stop signal (overcurrent detection) from the hardware.

- 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), 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 lowers the limit), 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.

- Rotational speed error

The rotational speed is monitored at the rotational speed monitoring cycle. When the speed exceeds the limit, CPU performs an emergency stop.

- Timeout error of Hall signal 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 signal interrupts, CPU performs an emergency stop.

Table 3-4 Setting Value of Each System Protection Function

Error name	Threshold	
Overvoltage error	Overvoltage limit [V]	28
	Monitoring cycle [ms]	1
Low voltage error	Low voltage limit [V]	15
	Monitoring cycle [ms]	1
Rotational speed error	Speed limit [rpm]	3500
	Monitoring cycle [ms]	1
Timeout error of Hall signal interrupt detection	Timeout value [ms]	500

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

Multiple control functions are used in this control program. However, functions which are not used in this system are undescribed.

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 variables used in the main process - Waiting for stability of the bus voltage function call - Status transition and event execution function call - Main process ⇒ User interface call ⇒ Watchdog timer clear function call
	board_ui Input: None Output: None	- Motor status change - Determination of rotational speed command value - Determination of rotation direction
	ics_ui Input: None Output: None	"Motor RSSK Support Tool" use - Motor status change - Determination of rotational 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_mrssk.c”

File name	Function name	Process overview
mtr_ctrl_mrssk.c	R_MTR_ChargeCapacitor Input: None Output: None	Waiting for stability of the bus voltage
	get_vr1 Input: None Output: (uint16) u2_ad_data / 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 3-7 List of Functions “mtr_ctrl_rl78g14.c”

File name	Function name	Process overview
mtr_ctrl_rl78g14.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
	mtr_init_tau Input: None Output: None	Initialization of the Timer Array Unit (TAU)
	mtr_init_intp Input: None Output: None	Initialization of external interrupt
	clear_wdt Input: None Output: None	Clearing the watchdog timer (WDT)
	mtr_clear_oc_flag Input: None Output: None	Cancelation the forced cutoff of the pulse output
	mtr_clear_trd0_imfa Input: None Output: None	Clearing the Compare Match Timer A (IMFA)
	mtr_disable_hall_intr Input: None Output: None	Disable Hall interrupts

Table 3-8 List of Functions "mtr_interrupt.c"

File name	Function name	Process overview
mtr_interrupt.c	mtr_oc_intp0_interrupt Input: None Output: None	Overcurrent detection process (Hardware detection) - Disable INTP0 interrupt servicing - Event processing selection function call (Generate error event) - Changing the motor status (Set the flag of error about overcurrent)
	mtr_hall_u_interrupt Input: None Output: None	Hall U signal interrupt function (INTP3) - Call common function of Hall signal interrupt
	mtr_hall_v_interrupt Input: None Output: None	Hall V signal interrupt function (INTP4) - Call common function of Hall signal interrupt
	mtr_hall_w_interrupt Input: None Output: None	Hall W signal interrupt function (INTP6) - Call common function of Hall signal interrupt
	mtr_1ms_interrupt Input: None Output: None	Calling every 1 [ms] (INTTM00) - Run mode management ⇒Setting speed reference ⇒Setting voltage reference ⇒Setting PWM duty - Detection of Hall signals' timeout - Error check function call - Motor stop detection function call
	mtr_carrier_interrupt Input: None Output: None	Calling every 50 [μs] (INTTRD0) - Measure inverter bus voltage - Clear compare match flag A function call

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: None Output: None	Initialization of the sequence process
	R_MTR_ExecEvent Input: (uint8) u1_event / occurred event Output: None	- Changing the status - Calling an appropriate process execution function for the occurred event
	mtr_act_active Input: (uint8) u1_state / motor status Output: (uint8) u1_state / motor status	- Calling for motor startup function - Enable Hall signal interrupts - Error check
	mtr_act_inactive Input: (uint8) u1_state / motor status Output: (uint8) u1_state / motor status	- Calling Timer RD(TRD) and PWM sotp function - Reset mode and configurations
	mtr_act_none Input: (uint8) u1_state / motor status Output: (uint8) u1_state / motor status	No process is performed.
	mtr_act_reset Input: (uint8) u1_state / motor status Output: (uint8) u1_state / motor status	- Global variables initialization - Cancelation of the forced cutoff function call
	mtr_act_error Input: (uint8) u1_state / motor status Output: (uint8) u1_state / motor status	Calling for motor control stop function
	mtr_pattern_set Input: None Output: None	- Call speed measurement process - Detection of Hall pattern - Voltage pattern set function call - Judge error status - Pattern change function call
	mtr_speed_calc Input: None Output: None	Speed measurement & calculation processing
	mtr_start_init Input: None 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_error_check Input: None Output: None	Error monitoring
	mtr_wait_motorstop Input: None Output: None	- Check motor stop - Disable Hall interrupts
	mtr_set_voltage_ref Input: None Output: None	Set reference of voltage
	mtr_set_speed_ref Input: None Output: None	Set reference of speed

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

File name	Function name	Process overview
mtr_spm_hall_120.c	mtr_hall_signal_process Input: None Output: None	Hall signal interrupt common function - Count interrupts for start of speed measurement - Clear counter for Hall timeout error - Reset motor stop wait conter - Conduction pattern set function call
	mtr_pattern_first60 Input: (uint8) u1_pattern / Conduction pattern Output: None	Set voltage pattern non-complementary first 60 degree PWM
	mtr_pattern_first60_comp Input: (uint8) u1_pattern / Conduction pattern Output: None	Set voltage pattern complementary first 60 degree PWM

Table 3-11 List of Functions “mtr_ctrl_rl78g14_mrssk”

File name	Function name	Process overview
mtr_ctrl_rl78g14_mrssk	mtr_init_trd Input: None Output: None	Initial setting of TRD
	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: None Output: None	Motor start processing - Enable Hall signal interrupts
	mtr_ctrl_stop Input: None Output: None	Motor stop processing - Stop Timer RD (TRD) - Stop PWM
	mtr_change_pattern Input: (uint8) u1_pattern / Conduction pattern Output: None	- Change conduction pattern - Error check
	mtr_get_adc Input: (uint8) u1_ad_ch / A/D conversion channel Output: (int16) s2_temp / A/D conversion value	Get A/D conversion value

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: (int16) s2_ref / PI control output value	PI control

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

File name	Function name	Process overview
mtr_gmc.c	mtr_get_vdc Input: None Output: (int16) s2_temp / vdc value	Obtaining the bus voltage
	mtr_check_over_voltage_error Input: (int16) s2_vdc / vdc value (int16) s2_limit_voltage / over voltage limit Output: (uint16) u2_temp / over voltage error flag	Checking over voltage error
	mtr_check_under_voltage_error Input: (int16) s2_vdc / vdc value (int16) s2_limit_voltage / under voltage limit Output: (uint16) u2_temp / under voltage error flag	Checking under voltage error
	mtr_check_over_speed_error Input: (uint16) u2_speed_rad / motor speed (uint16) u2_speed_limit / speed limit Output: (uint16) u2_temp / over speed error flag	Checking over speed error
	mtr_get_duty Input: (volatile int16) s2_v_ref / reference voltage (volatile int16) s2_vdc_ad / bus voltage A/D conversion value Output: (int16) s2_temp / rate of PWM duty	Calculate PWM duty
	mtr_check_timeout_error Input: (uint16) u2_cnt_timeout / counter of timeout (uint16) u2_timeout_limit / timeout limit Output: (uint16) u2_temp / flag of timeout error	Checking time-out error

Table 3-14 List of Functions “mtr_driver_access.c”

File name	Function name	Process overview
mtr_driver_access.c	R_MTR_SetSpeed Input: (int16) s2_ref_speed / speed command value Output: None	Setting the speed command value
	R_MTR_SetDir Input: (uint8) u1_dir / rotation direction Output: None	Setting the rotation direction
	R_MTR_GetSpeed Input: None Output: (uint16) u2_speed_rpm / speed	Obtaining the calculated speed value
	R_MTR_GetDir Input: None Output: (uint8) g_u1_direction / rotation direction	Obtaining the rotation direction
	R_MTR_GetStatus Input: None Output: (uint8) g_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 the control programs are given below. However, note that the local variables are not mentioned.

In the control programs in this application note use fixed-point calculation. Therefore, some variables are already established with fixed-point calculation. Bits number in fractional part of fixed-point number is expressed in the Q format. For example, a "Q3" number has 3 fractional bits. "Qn" number is indicated on "Scale" column in below table.

Table 3-15 List of variables [1/3]

Variable name	Type	Scale	Content	Remarks
g_u2_max_speed_rpm	uint16	-	Rotational speed command maximum value	Mechanical angle [rpm]
g_u2_min_speed_rpm	uint16	-	Rotational speed command minimum value	Mechanical angle [rpm]
g_u2_margin_min_speed_rpm	uint16	-	Rotational speed command minimum value for motor stop	Mechanical angle [rpm]
g_s2_ref_speed_rpm	int16	-	User setting rotational speed	Mechanical angle [rpm]
g_u2_speed_rpm	uint16	-	Measured speed value	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_userif	int16	-	User interface switch	0: GUI use (default) 1: Board user interface use
g_s2_mode_system	int16	-	System mode	
g_s2_enable_write	int16	-	Control flag to reflect input data	
st_ics_input	MTR_ICS_INPUT	-	GUI input structure	
g_u1_cnt_ics	uint8	-	GUI decimation counter	
g_u2_trig_enable_write	uint16	-	Enable flag to reflect input data to internal data	
st_ics_input_buff	MTR_ICS_INPUT	-	Buffer of GUI input structure	
g_u1_mode_system	uint8	-	State management	0x00: Inactive mode 0x01: Active mode 0x02: Error mode
g_u2_run_mode	uint16	-	Operation mode management	0x00: Initialize mode 0x01: Boot mode 0x02: Drive mode 0x03: Analysis mode 0x04: Tune mode

Table 3-16 List of variables [2/3]

Variable name	Type	Scale	Content	Remarks
g_u2_error_status	uint16	-	Error status management	0x00: None error 0x01: Over current error 0x02: Over voltage error 0x04: Over speed error 0x08: Hall signal time out error 0x10: BEMF time out error 0x20: Hall pattern error 0x40: BEMFpattern error 0x80: Under voltage error 0xFF: Undefined error
g_u2_state_voltage_ref	uint16	-	Voltage state	0: INIT mode (reference voltage 0) 1: BOOT mode 2: PI Control mode
g_u2_state_speed_ref	uint16	-	Speed state	0: BOOT mode (reference speed 0) 1: Speed Control mode
g_u2_sensor_conf	uint16	-	Sensor configuration management	0x01: Sensorless 0x02: Hall sensor 0x04: Encoder 0x08: Resolver
g_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
g_u2_ctrl_conf	uint16	-	Control configuration management	0x01: Current control 0x02: Speed control 0x04: Position control 0x08: Torque control 0x10: Voltage control
g_u2_motor_pp	uint16	-	number of pole pairs	
g_s2_vdc_ad	int16	Q7	Inverter bus voltage A/D value	[V]
g_s2_v_ref	int16	Q7	Voltage command value	Speed PI control output value [V]
g_s2_start_ref_v	int16	Q7	Start voltage command value	[V]
g_u2_pwm_duty	uint16		PWM duty	
g_u2_ref_speed_rad	uint16	Q3	Speed reference (user selected) value	Electrical angle [rad/s]
g_u2_ref_speed_rad_crtl	uint16	Q3	Speed command value	Electrical angle [rad/s]
g_u2_speed_rad	uint16	Q3	Measured speed value	Electrical angle [rad/s]
g_s2_kp_speed	int16	Q16	Speed PI control proportional gain	
g_s2_ki_speed	int16	Q22	Speed PI control integral gain	

Table 3-17 List of variables [3/3]

Variable name	Type	Scale	Content	Remarks
g_u1_cnt_speed_pi	uint8	-	Speed PI control function call interval counter	
g_s2_speed_lpf_k	int16	Q14	Speed LPF parameter	
g_s2_limit_speed_change	int16	Q3	Step of speed reference	Electrical angle [rad/s]
g_s2_lim_v	int16	Q7	Limit of speed PI control	[V]
g_s4_ilim_v	int32	Q26	Limit for integral part of speed PI control	[V]
g_u1_flag_charge_cap	uint8	-	Flag for capacitor charging completed	
g_u2_speed_calc_base	uint16	―	Base parameter to calculate speed	
g_u1_flg_wait_stop	uint8	-	Motor rotation stop waiting flag	
g_u2_cnt_wait_stop	uint16	-	Motor rotation stop waiting counter	
g_u1_v_pattern	uint8	-	Conduction pattern	
g_u1_direction	uint8	-	Rotation direction	0：CW 1：CCW
g_u2_cnt_timeout	uint16	-	Timeout detection counter	
g_u2_hall_timer_cnt	uint16	-	Free run timer count value	
g_u2_pre_hall_timer_cnt	uint16	-	Previous free run timer count value	
g_u2_timer_cnt_sum	uint16	-	Speed measurement timer count value of 2 pi (erectrical angle)	
g_u2_timer_cnt_buf[6]	uint16	-	Speed measurement timer count buffer	
g_u1_timer_cnt_num	uint8	-	Speed measurement timer count buffer number	
g_u1_hall_signal	uint8	-	Hall signal capture buffer	
g_u1_hall_intr_cnt	uint8	-	Waiting counter of Hall signal interrupts for speed measurement	
g_u1_hall_wait_cnt	uint8	-	Waiting counts of Hall signal interrupts for speed measurement	
st_pi_speed	MTR_PI_CTRL	-	Structure for speed PI control	

3.4 List of structure of 120-degree conducting control using Hall sensors software

List of structure used in this control program are given below.

Table 3-18 List of structures

Structures	Member	Type	Scale	Content	Remarks
MTR_PI_CTRL	s2_err	int16	Q3	Error	Electrical angle [rad/s]
	s2_kp	int16	Q16	PI control proportional gain	
	s2_ki	int16	Q22	PI control integral gain	
	s2_limit	int16	Q7	Integral output limit	
	s4_refi	int32	Q7	Integral output value	
	s4_ilimit	int32	Q26	Integral output limit	
MTR_ICS_INPUT	s2_direction	int16	-	Rotational direction	0：CW 1：CCW
	u2_ref_speed	uint16	-	Reference speed	Mechanical angle [rpm]
	u2_motor_pp	uint16	-	Number of pole pairs	
	s2_kp_speed	int16	Q16	Speed PI control proportional gain	
	s2_ki_speed	int16	Q22	Speed PI control Integral gain	
	s2_speed_lpf_k	int16	Q14	Speed LPF parameter	
	s2_limit_speed_change	int16	Q3	Speed command maximum increase limit	Electrical angle [rad/s]
	s2_start_refv	int16	Q7	Reference voltage for start-up	[V]
	u1_hall_wait_cnt	uint8	-	Hall signal 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 the control program are given below.

Table 3-19 List of Macro definitions "main.h"

File name	Macro name	Definition value	Remarks
main.h	ICS_UI	0	Analyzer use mode
	BOARD_UI	1	Board user interface use mode
	M_CW	0	Rotation direction: CW
	M_CCW	1	Rotation direction: CCW
	MAX_SPEED	CP_MAX_SPEED_RPM	Rotational speed command maximum value (mechanical angle) [rpm]
	MIN_SPEED	CP_MIN_SPEED_RPM	Rotational speed command minimum value (mechanical angle) [rpm]
	MARGIN_SPEED	50	Rotational speed command minimum value creation constants for stop (mechanical angle) [rpm]
	MARGIN_MIN_SPEED	MIN_SPEED - MARGIN_SPEED	Rotational speed command minimum value for motor stop (mechanical angle) [rpm]
	LIMIT_SPEED_CHANGE	CP_ LIMIT_SPEED_CHANGE	Speed command maximum increase limit (electrical angle) [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
	START_REF_V	CP_START_REF_V	Voltage command value at start-up[V] (scale: Q7)
	SW_ON	0	Active in case of "Low"
	SW_OFF	1	Inctive in case of "High"
	CHATTERING_CNT	10	Chattering removal
	VR1_SCALING	(MAX_SPEED + 422) / 0x0200	Speed command value creation constant
	ADJUST_OFFSET	0x01FF	Speed command value offset adjustment constant
	POLE_PAIRS	MP_POLE_PAIRS	Number of pole pairs
	REQ_CLR	0	Flag clearing
	REQ_SET	1	Flag setting
	ICS_ADDR	0xFE00	Address data for ICS
	ICS_INT_LEVEL	2	Interrupt priority level for ICS
	ICS_NUM	CA : 0x50 CC-RL : 0x40	Size of data on ICS interface
	ICS_BRR	15	Selection of bitrate register on ICS (only CC-RL)
	ICS_INT_MODE	0	Setting of ICS interrupt mode (only CC-RL)

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

File name	Macro name	Definition value	Remarks
motor_parameter.h	MP_POLE_PAIRS	2	Motor pole pairs
	MP_RESISTANCE	6.447f	Resistance [Ω] (not used)
	MP_D_INDUCTANCE	0.0045f	d-axis Inductance [H] (not used)
	MP_Q_INDUCTANCE	0.0045f	q-axis Inductance [H] (not used)
	MP_MAGNETIC_FLUX	0.02159f	Magnetic flux [Wb] (not used)

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

File name	Macro name	Definition value	Remarks
control_parameter.h	CP_MAX_SPEED_RPM	2650	Maximum rotational speed (mechanical angle) [rpm]
	CP_MIN_SPEED_RPM	550	Minimum rotational speed (mechanical angle) [rpm]
	CP_LIMIT_SPEED_CHANGE	0.20f * 0x08	Step to increase speed reference (electrical angle) [rad/s] (scale: Q3)
	CP_START_REF_V	5.8f * 0x80	Start voltage command value [V] (scale: Q7)
	CP_SPEED_PI_KP	0.0150f * 0x10000	Speed PI proportional gain (scale: Q16)
	CP_SPEED_PI_KI	0.0003f * 0x400000	Speed PI integral gain (scale: Q22)
	CP_SPEED_LPF_K	1.0f * 0x4000	Speed LPF parameter
	MTR_FIRST60	0	Non-Complementary First 60 degree PWM
	MTR_FIRST60_COMP	1	Complementary First 60 degree PWM (default)

Table 3-22 List of Macro definitions “mtr_ctrl_rl78g14_mrssk.h” [1/2]

File name	Macro name	Definition value				Remarks
mtr_ctrl_rl78g14_mrssk.h	MTR_PWM_TIMER_FREQ	64.0f				PWM timer count frequency [MHz]
	MTR_CARRIER_FREQ	20.0f				Carrier frequency [kHz]
	MTR_DEADTIME	2000				Dead time [ns]
	MTR_DEADTIME_SET	(int16)(MTR_DEADTIME*MTR_PWM_TIMER_FREQ/1000)				Dead time setting value
	MTR_CARRIER_SET	(MTR_PWM_TIMER_FREQ*1000/MTR_CARRIER_FREQ/2)-2				Carrier setting value for non-complementary (selectable by a compile option)
		(MTR_PWM_TIMER_FREQ*1000/MTR_CARRIER_FREQ/2)+MTR_DEADTIME_SET-2				Carrier setting value for complementary (selectable by a compile option)
	MTR_HALF_CARRIER_SET	MTR_CARRIER_SET / 2				Half of “MTR_CARRIER_SET”
	MTR_NDT_CARRIER_SET	MTR_CARRIER_SET - MTR_DEADTIME_SET				
	MTR_PORT_HALL_U	CA	P3.0	CC-RL	P3_bit.no0	Hall signal U input
	MTR_PORT_HALL_V		P3.1		P3_bit.no1	Hall signal V input
	MTR_PORT_HALL_W		P14.0		P14_bit.no0	Hall signal W input
	MTR_PORT_UP		P1.5		P1_bit.no5	U phase (positive phase) output port
	MTR_PORT_UN		P1.4		P1_bit.no4	U phase (negative phase) output port
	MTR_PORT_VP		P1.3		P1_bit.no3	V phase (positive phase) output port
	MTR_PORT_VN		P1.1		P1_bit.no1	V phase (negative phase) output port
	MTR_PORT_WP		P1.2		P1_bit.no2	W phase (positive phase) output port
	MTR_PORT_WN		P1.0		P1_bit.no0	W phase (negative phase) output port
	MTR_PORT_SW1		P0.5		P0_bit.no5	SW1 input port
	MTR_PORT_SW2		P0.6		P0_bit.no6	SW2 input port
	MTR_PORT_LED1		P5.2		P5_bit.no2	LED1 output port
	MTR_PORT_LED2		P5.3		P5_bit.no3	LED2 output port
	MTR_PORT_LED3		P5.4		P5_bit.no4	LED3 output port

Table 3-23 List of Macro definitions "mtr_ctrl_rl78g14_mrssk.h" [2/2]

File name	Macro name	Definition value	Remarks
mtr_ctrl_rl78g14_mrssk.h	MTR_LED_ON	0	LED active in case of "Low"
	MTR_LED_OFF	1	
	MTR_INPUT_V	(int16) (24*0x80)	Input DC voltage [V] (scale: Q7)
	MTR_MCU_ON_V	(int16) (MTR_INPUT_V*0.8)	MCU power on voltage (scale: Q7)
	MTR_VDC_SCALING	3555	Inverter bus voltage A/D conversion value resolution
	MTR_RECIVDC_SCALING	64	Reciprocal value of MTR_VDC_SCALING
	MTR_OVERVOLTAGE_LIMIT	(int16) (28*0x80)	High voltage limit [V] (scale: Q7)
	MTR_UNDERVOLTAGE_LIMIT	(int16) (15*0x80)	Low voltage limit [V] (scale: Q7)
	MTR_TAU1_CNT	TCR01	Register of timer counter for speed measurement
	MTR_ADCCH_VR1	2	A/D Converter channel of VR1
	MTR_ADCCH_VDC	6	A/D Converter channel of VDC
	MTR_OC_HW_FLG	TRDSHUTS	Forced cutoff flag
	MTR_OC_INTR_MASK	PMK0	INTP0 interrupt mask flag
	MTR_DISABLE_OC_INTR	1	Disable INTP0 interrupt service

Table 3-24 List of Macro definitions “mtr_spm_hall_120.h” [1/3]

File name	Macro name	Definition value	Remarks
mtr_spm_hall_120.h	MTR_POLE_PAIRS	MP_POLE_PAIRS	Motor Pole pairs
	MTR_TWOPI	2 * 3.14159265f	2π
	MTR_RPM_RAD	13726	[rpm]→[rad/s]
	MTR_RAD_RPM	4889	[rad/s]→[rpm]
	MTR_SPEED_LIMIT_RPM	3000	Speed limit (mechanical angle) [rpm]
	MTR_SPEED_LIMIT	MTR_SPEED_LIMIT_RPM * (MTR_TWOPI / 60)	Speed limit (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 * 0x80	Voltage PI control output limit [V] (scale: Q7)
	MTR_SPEED_PI_I_LIMIT_V	24 * 0x80 * 0x40000	Voltage PI control output limit [V] Integral part (for calculation) (scale: Q25)
	MTR_SPEED_CALC_BASE	767	Calculation factor to translate the timer counter to rotational speed
	MTR_SPEED_LPF_K	CP_SPEED_LPF_K	Speed LPF parameter (scale: Q14)
	MTR_LIMIT_SPEED_CHANGE	CP_LIMIT_SPEED_CHANGE	Speed command maximum increase limit (electrical angle) [rad/s] (scale: Q3)
	MTR_MIN_SPEED_RAD	CP_MIN_SPEED_RPM * (MTR_TWOPI / 60) * 0x08	Rotational speed command minimum value (electrical angle) [rad/s] (scale: Q3)
	MTR_MAX_DRIVE_V	(int16)22 * 0x80	Maximum command voltage [V] (scale: Q7)
	MTR_MIN_DRIVE_V	(int16)0.1f * 0x80	Minimum command voltage [V] (scale: Q7)
	MTR_START_REF_V	CP_START_REF_V	Voltage reference for BOOT mode[V] (scale: Q7)
	MTR_TIMEOUT_CNT	800	Timeout count limit [ms]
	MTR_STOP_WAIT_CNT	300	Stop judge count [ms]
	MTR_WAIT_SPEED_CALC	48	Wait speed measurement still Hall signal interrupts become this counts
	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	

Table 3-25 List of Macro definitions “mtr_spm_hall_120.h” [2/3]

File name	Macro name	Definition value	Remarks
mtr_spm_hall_120.h	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_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 interrupt decimation times for GUI function call
	MTR_V_ZERO_CONST	0	Zero voltage mode (for Inactive state)
	MTR_V_CONST	1	Start voltage mode
	MTR_V_PI_OUTPUT	2	PI control mode

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

File name	Macro name	Definition value	Remarks
mtr_spm_hall_120.h	MTR_SPEED_ZERO_CONST	0	Init speed mode
	MTR_SPEED_CHANGE	1	Speed control mode
	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_STOP	0x00	Stop event
	MTR_EVENT_RUN	0x01	Run event
	MTR_EVENT_ERROR	0x02	Error event
	MTR_EVENT_RESET	0x03	Reset event
	MTR_SIZE_EVENT	4	Event size
	MTR_MODE_INIT	0x00	Initial 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	Sensor less
	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_POSITION	0x04	Position control
	MTR_CONTROL_TORQUE	0x08	Torque control
	MTR_CONTROL_VOLTAGE	0x10	Voltage control
	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 Set Control flows (flow charts)

3.6.1 Main process

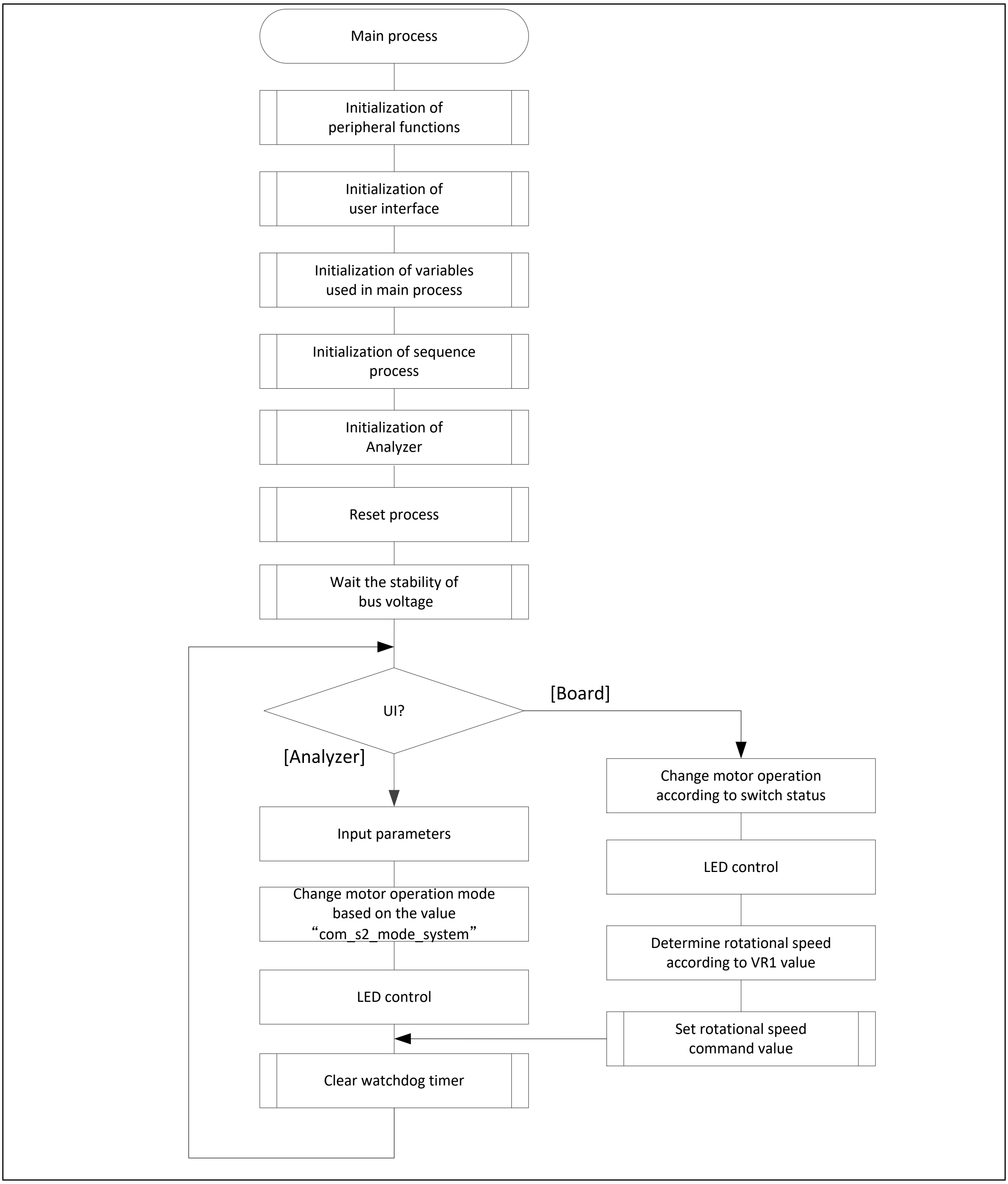


Figure 3-7 Main Process Flowchart

3.6.2 Carrier cycle interrupt handling

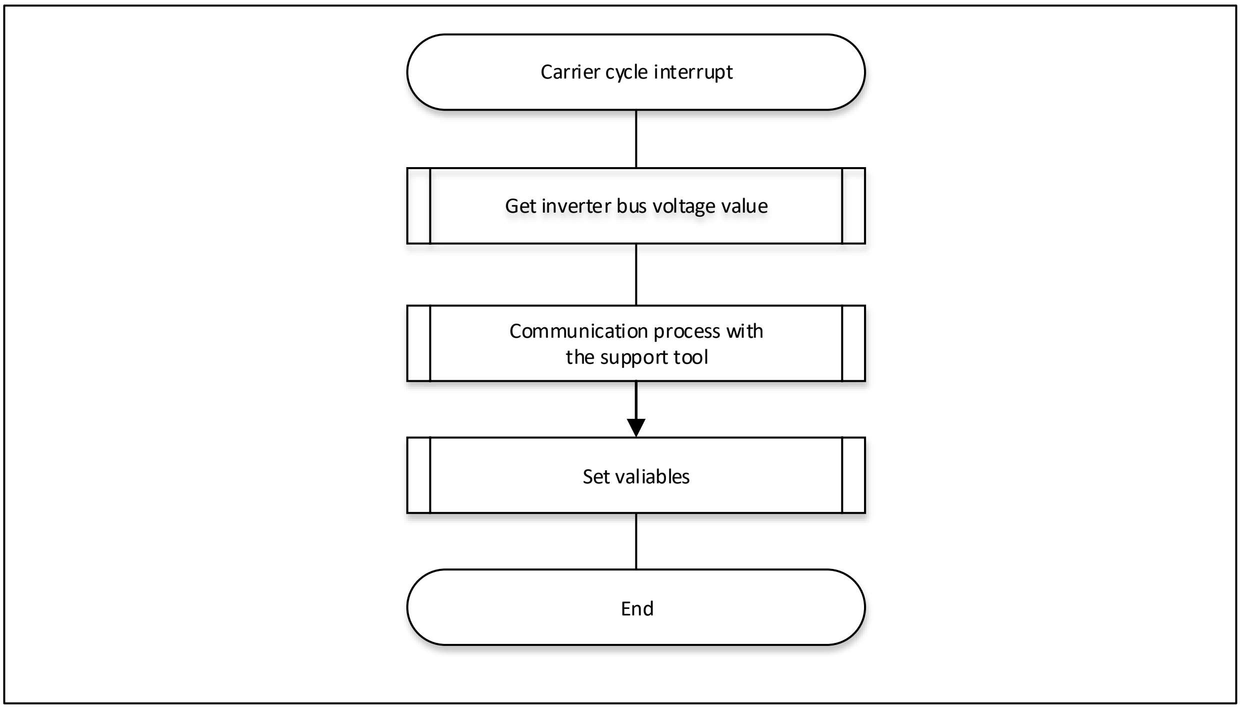


Figure 3-8 50[μs] Cycle Interrupt Handling (120-degree Control using Hall sensors)

3.6.3 1 [ms] interrupt handling

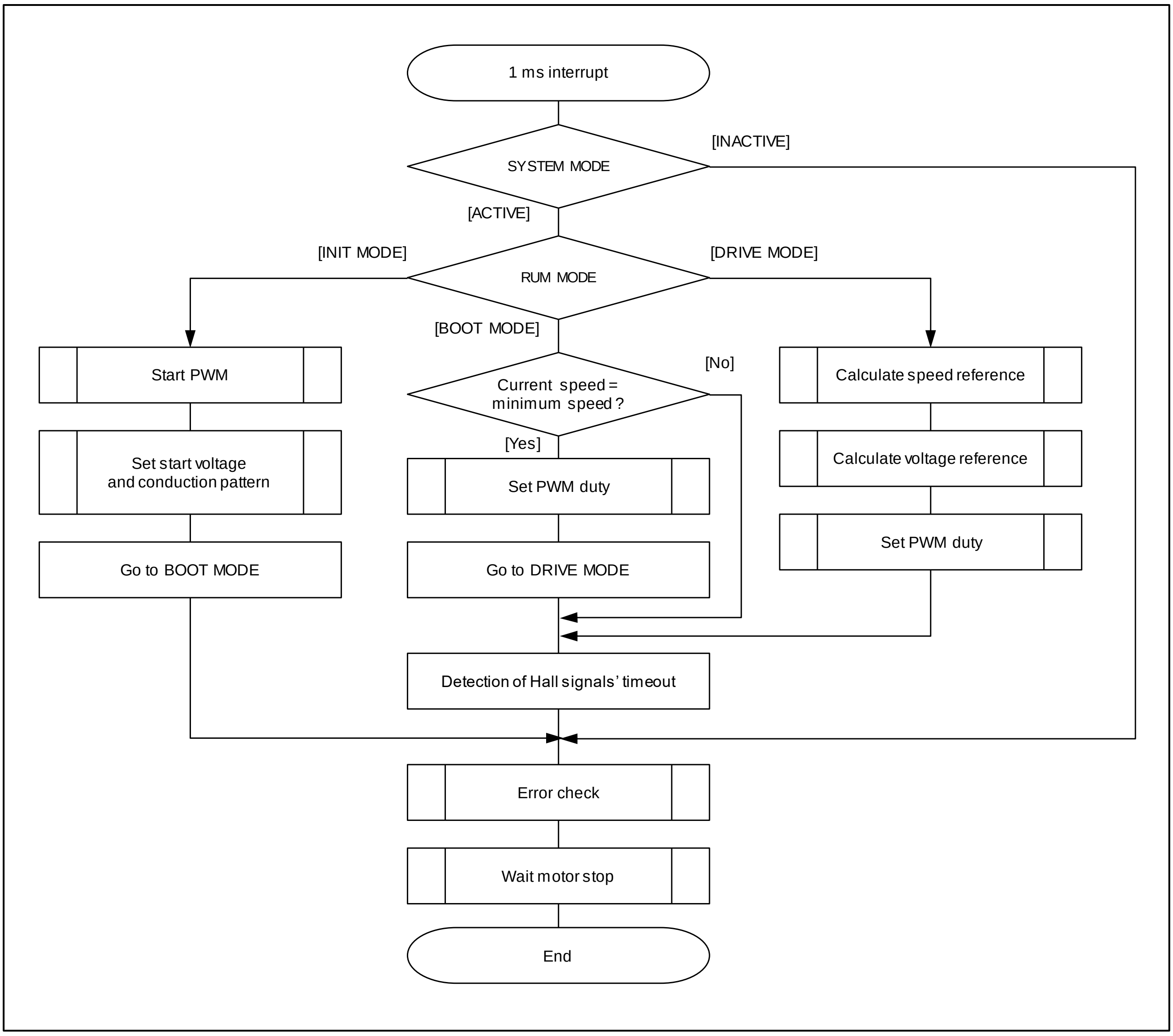


Figure 3-9 1 [ms] Interrupt Handling

3.6.4 Overcurrent interrupt handling

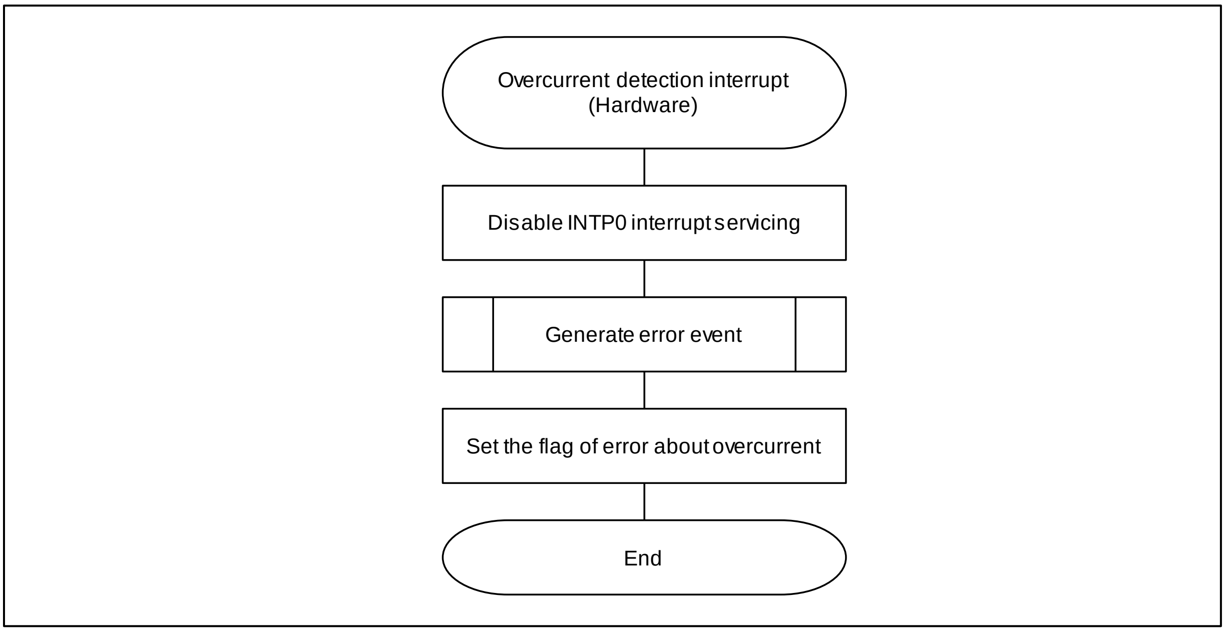


Figure 3-10 Over Current Detection Interrupt Handling (INTP0)

3.6.5 Hall signal interrupt handling (common process)

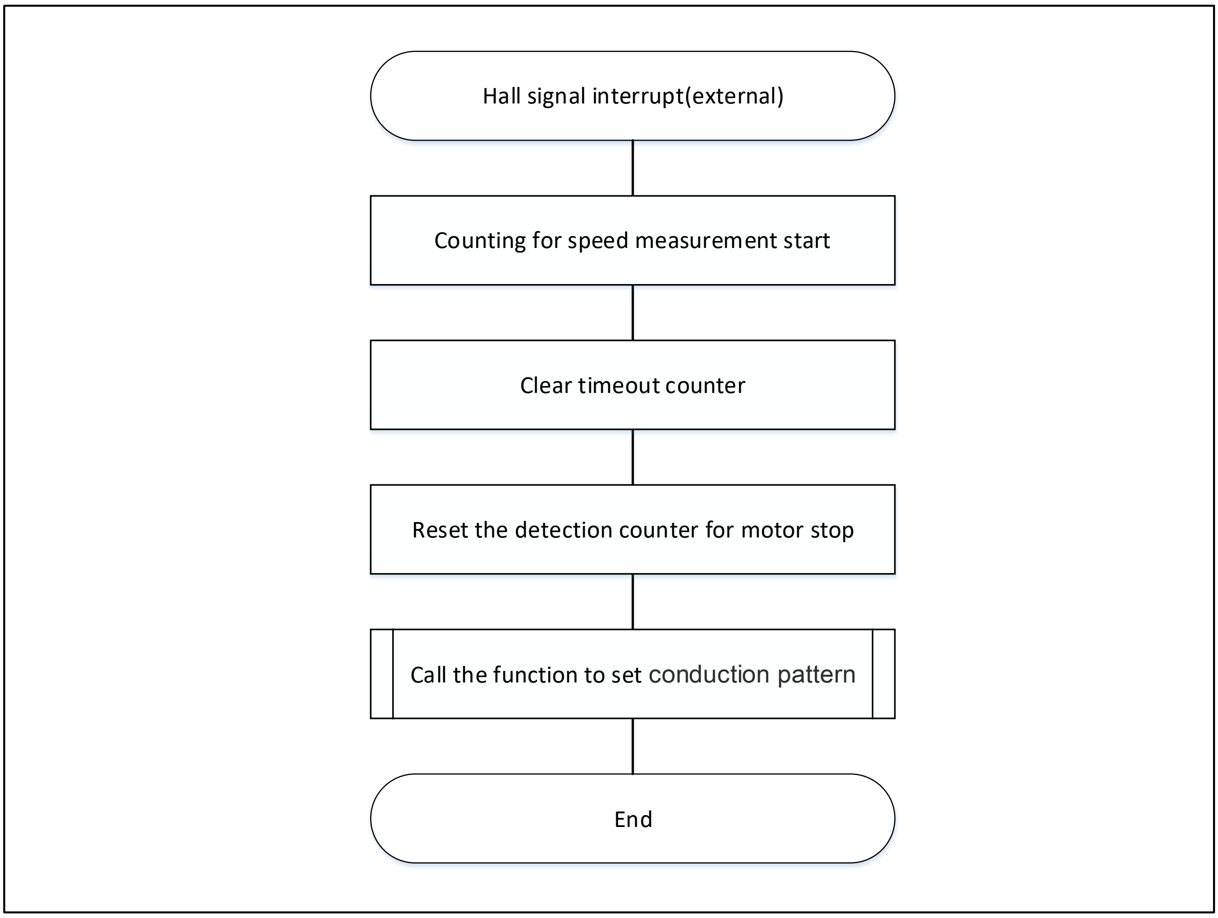


Figure 3-11 Hall signal Interrupt Handling (common process)

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, rotational speed command, etc.) based on the motor control development support tool, ' Renesas Motor Workbench' can be used. Please refer to 'Renesas Motor Workbench V.1.00 User's Manual' for usage and more details. You can find ' Renesas Motor Workbench' on Renesas Electronics Corporation website.

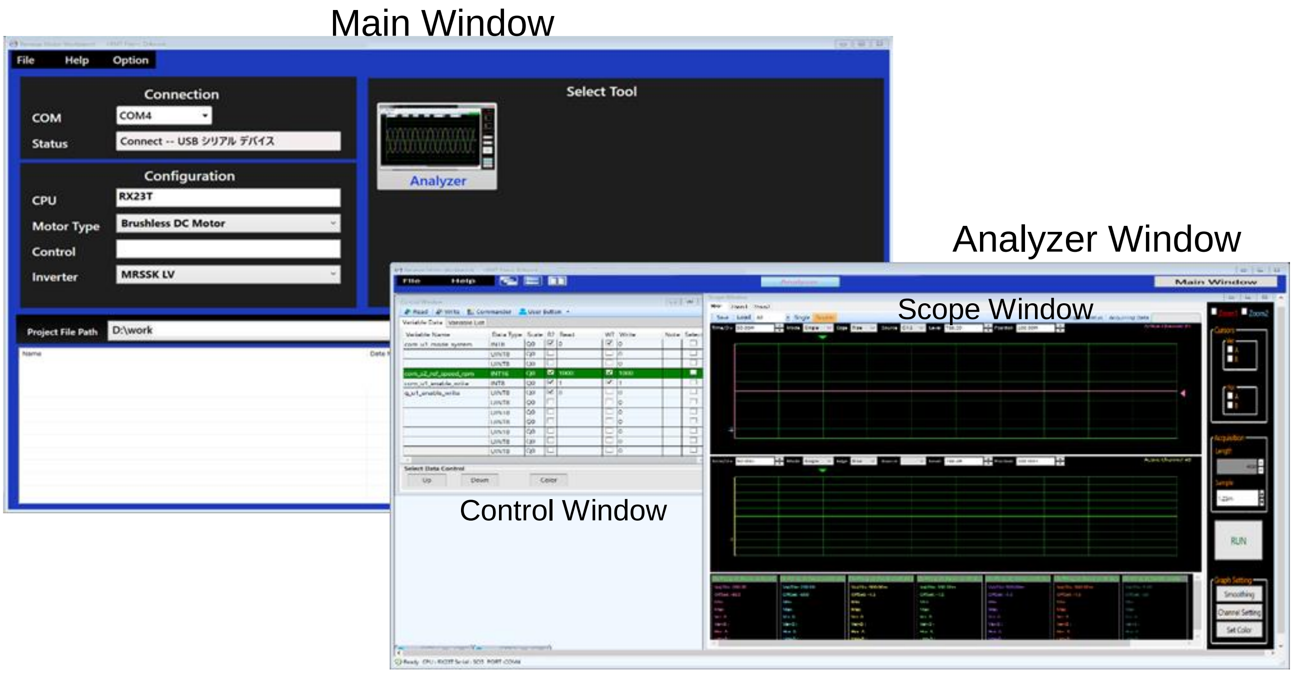
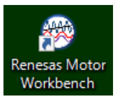


Figure 4-1 Renesas Motor Workbench – Appearance

<u>Set up for Motor control development support tool</u>

(1) Start 'Renesas Motor Workbench' by clicking this icon. 

(2) Drop down menu [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 the 'Analyzer' icon in right side of Main Window. (Then, "Analyzer Window" will be displayed.)

(5) Please refer to '4.3 Operation Example for Analyzer' for motor driving operation.

4.2 List of variables for Analyzer

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.

In the sample programs in this application note use fixed-point calculation. Therefore some variables are already established with fixed-point calculation. Bits number in fractional part of fixed-point number is expressed in the Q format. For example, a "Q3" number has 3 fractional bits. "Qn" number is indicated on "Scale" column in below table.

When referring to variables with fixed-point number, it is possible to display the value without scaling by choosing same "Qn" in "Control Window".

Table 4-1 List of Variables for Analyzer

Variable name	Type	Scale	Content	Remarks ([ ]: reflection variable name)
com_s2_sw_userif (*)	int16	-	User interface switch 0: Analyzer use (default) 1: Board user interface use	[g_s2_sw_userif]
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	[g_u1_direction]
com_u2_ref_speed_rpm	uint16	-	Speed command value (mechanical angle) [rpm]	[g_u2_ref_speed_rad]
com_s2_kp_speed	int16	Q16	Speed PI control proportional gain	[g_s2_kp_speed]
com_s2_ki_speed	int16	Q22	Speed PI control integral gain	[g_s2_ki_speed]
com_s2_speed_lpf_k	int16	Q14	Speed LPF parameter	[g_s2_speed_lpf_k]
com_s2_limit_speed_change	int16	Q3	Command speed changing limit (electrical angle) [rad/s]	[g_s2_limit_speed_change]
com_s2_start_ref_v	int16	Q7	Soltage command value	[g_s2_start_ref_v]
com_u2_motor_pp	uint16	-	Number of pole pairs	[g_u2_motor_pp]
com_u1_hall_wait_cnt	uint8	-	Wait Hall counts for start speed measurement	[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”. Refer to ‘Renesas Motor Workbench V.1.00 User’s Manual’ for “Control Window”.

- 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.

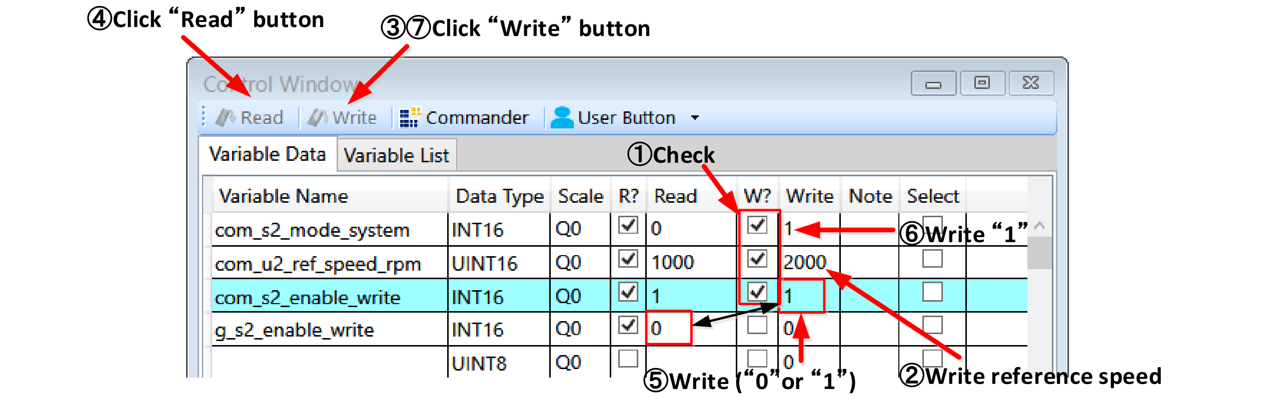


Figure 4-2 Procedure - Driving the motor

- Stop the motor

① Type a value of “0” in the [Write] box of “com_s2_mode_system”

② Click the “Write” button.

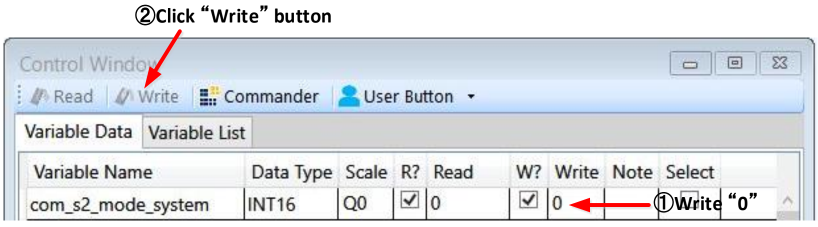


Figure 4-3 Procedure - Stop the motor

- Error cancel operation

① Type a value of “3” in the [Write] box of “com_s2_mode_system”

② Click the “Write” button.

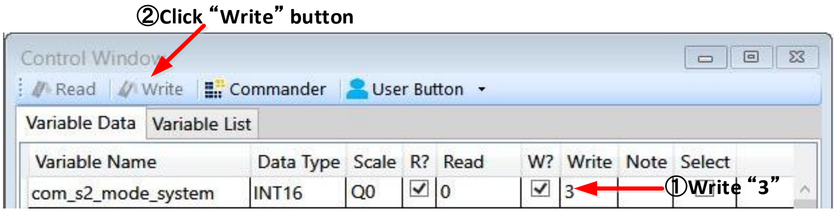


Figure 4-4 Procedure - Error cancel operation

Website and Support

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

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Oct.02.2017	─	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. 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 Microprocessing unit or Microcontroller unit products 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.

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.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

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: +44-1628-585-100, Fax: +44-1628-585-900

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, P. R. China 200333
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, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141