RX Family

Sensorless Vector Control of a Permanent Magnet Synchronous Motor - 2 motor, For MCK

Introduction

This application note is intended to explain the sample program that uses Renesas RX26T microcontroller to drive two permanent magnet synchronous motor with vector control. The current detection required for vector control is compatible with the 2-shunt method. The target software for this application note uses the Smart Configurator tool and the components required for motor control.

The target software for this application note is for reference only, and we do not guarantee the operations. Only use the target software of this application note after conducting thorough evaluation in an appropriate environment.

Target device

Operations of the target software of this application note are checked by using the following device.

- RX26T RAM64KB Version(R5F526TFCDFP)

Target software

The following shows the target software for this application note:

- RX26T_MCBA_MCILV1_2SPM_LESS_FOC_CSP_V100（IDE：CS+ edition）
- RX26T_MCBA_MCILV1_2SPM_LESS_FOC_E2S_V100（IDE：e^{2}studio edition）

Contents

1. Overview

This application note is intended to explain the method of using the sample program that uses a microcontroller manufactured by Renesas to drive two permanent magnet synchronous motor with vector control. Using the sample program together with a motor control kit (Renesas Flexible Motor Control Kit) enables motor control. This sample program supports Renesas Motor Workbench, a motor control development support tool, and therefore can be used as a user interface (UI) to check the MCU internal data and to control a motor. You can use the sample program for reference purposes or for software development by checking how MCU functions are allocated, how control is loaded by interrupts, and other information in the sample program.

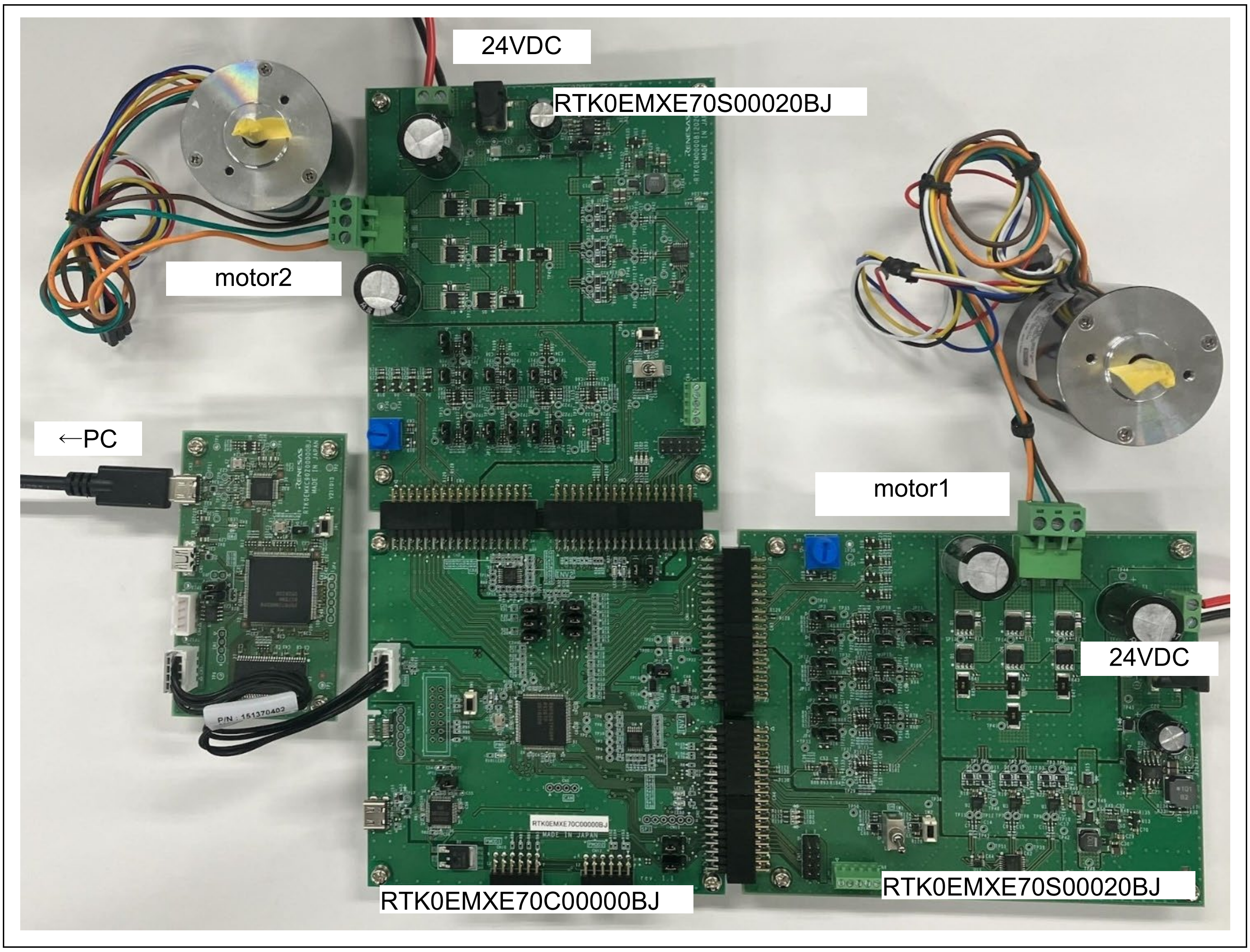


Figure1-1 Operating environment of the sample program

2. Development environments

2.1 Test environments

Table 2-1 and Table 2-2 show the development environments for the software that this application note is applicable to.

Table 2-1 Hardware development environment

Category	Product used
Microcontroller / CPU board product type	RX26T RAM64KB Version (R5F526TFCDFP) / RTK0EMXE70C00000BJ
Inverter board	Renesas MCI-LV-1 Inverter board for 48V 10A BLDC / RTK0EM0000B12020BJ
Motor	R42BLD30L3 (Manufactured by MOONS' Industries Japan Co., Ltd.)

Table 2-2 Software development environment

IDE version	RX smart configurator	Toolchain version
CS+: V8.11.00	Version 2.20.00	CC-RX: V3.06.00
e^{2}studio: 2024-01	Plug-in version of e^{2}studio	

For the purchase or technical support of this system, contact a Renesas Electronics Corporation sales representative or an authorized Renesas Electronics Corporation product distributor.

2.2 Hardware specifications

2.2.1 Hardware configuration diagram

(1) Hardware configuration diagram (Overall)

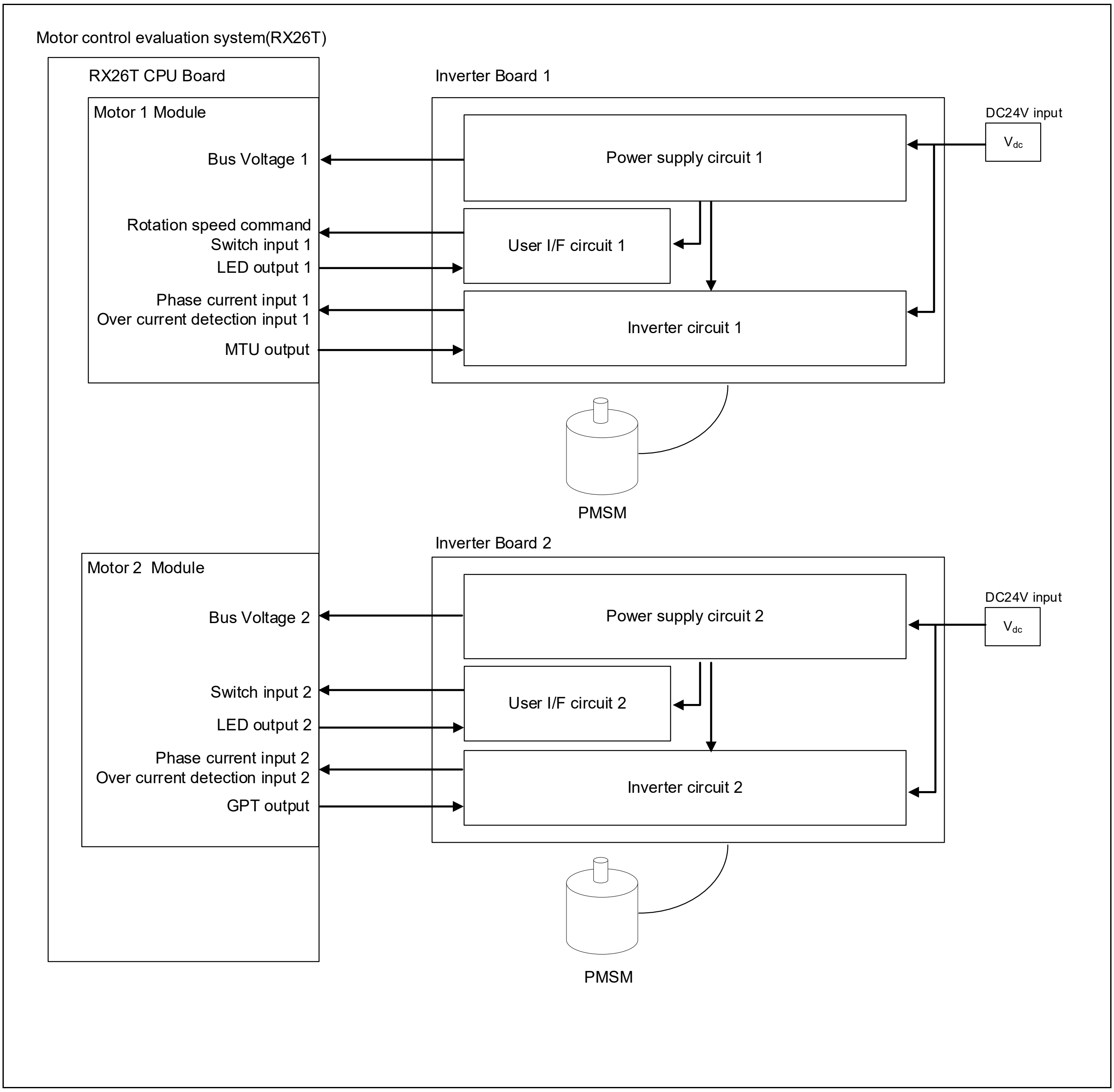


Figure 2-1 Hardware configuration diagram (Overall)

(2) Hardware configuration diagram (motor 1)

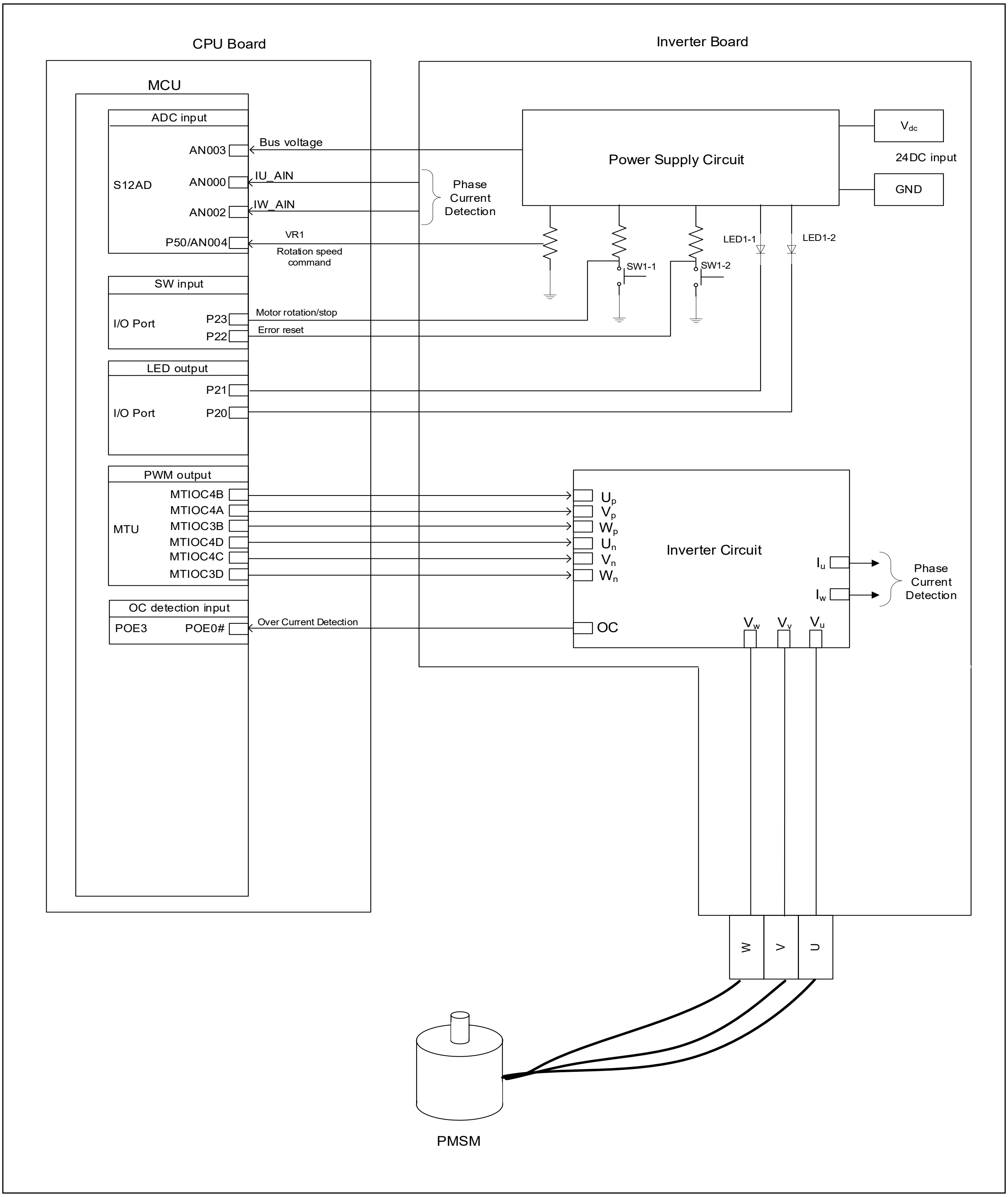


Figure2-2 Hardware configuration diagram (motor 1)

(3) Hardware configuration diagram (motor 2)

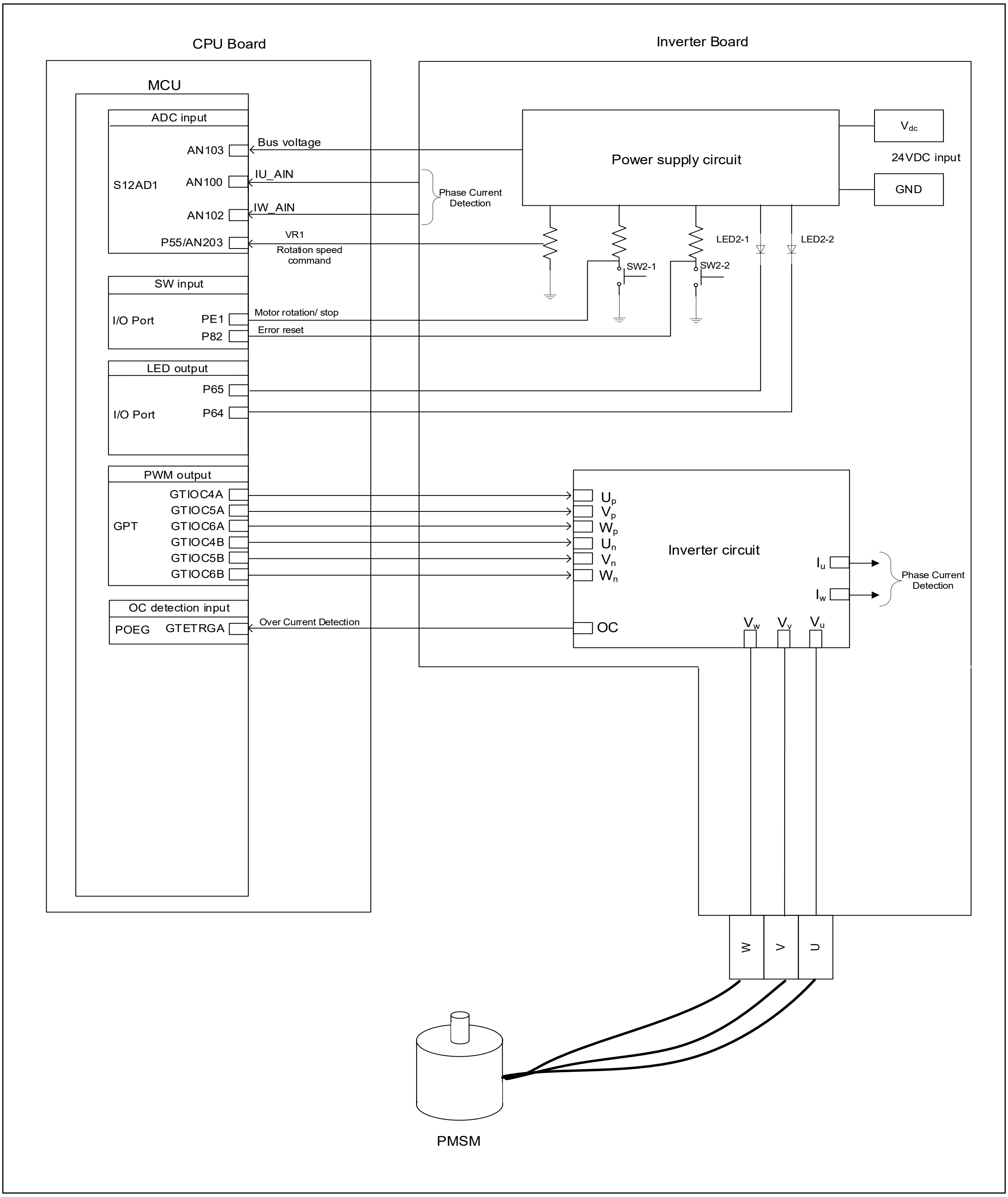


Figure2-3 Hardware configuration diagram (motor 2)

2.2.2 Board user interface

Table 2-3 lists the components of the user interface of the board for this system.

Table 2-3 Board user interface

Item	Interface component		Function
	Motor 1	Motor 2	
Rotation speed (Variable resistor)	VR1-1	VR1-2	Inputs the rotation speed command value (analog value).
START/STOP (Toggle switch)	SW1-1	SW1-2	Instructs start or stop of motor rotation.
ERROR RESET (Push switch)	SW2-1	SW2-2	Instructs recovery from an error state
LED1 (Orange LED)	LED1-1	LED1-2	On: The motor is rotating. Off: The motor is stopped.
LED2 (Orange LED)	LED2-1	LED2-2	On: An error was detected. Off: The system is operating normally.
RESET (Push button)	SW1 (On the CPU board)		System reset

2.2.3 Peripheral functions

Table 2-4 and Table 2-5 shows allocation of input/output functions to peripheral functions that are used in this system. In the sample program, Smart Configurator is used to configure the peripheral functions. For details, see 5.12.

Table 2-4 Input/output functions and peripheral functions (motor 1)

Pin	Function	I/O	Peripheral function	Port Function
57	PWM emergency stop input when an overcurrent is detected	I	POE3	POE0#
89	Measurement of the W-phase current	I	ADC(S12AD)	AN002
91	Measurement of the U-phase current	I	ADC(S12AD)	AN000
88	Measurement of the inverter bus voltage	I	ADC(S12AD)	AN003
68	Controlling whether to turn on LED1	O	I/O Port (output)	P21
69	Controlling whether to turn on LED2	O	I/O Port (output)	P20
54	PWM output (U_p) / "High" active	O	MTU	MTIOC4B
51	PWM output (U_n) / "High" active	O	MTU	MTIOC4D
55	PWM output (V_p) / "High" active	O	MTU	MTIOC4A
52	PWM output (V_n) / "High" active	O	MTU	MTIOC4C
56	PWM output (W_p) / "High" active	O	MTU	MTIOC3B
53	PWM output (W_n) / "High" active	O	MTU	MTIOC3D
83	Function for inputting the rotation speed command value (analog value)	I	ADC(S12AD2)	P50/AN204
66	START/STOP toggle switch	I	I/O Port (Input)	P23
67	ERROR RESET push switch	I	I/O Port (Input)	P22

Table 2-5 Input/output functions and peripheral functions (motor 2)

Pin	Function	I/O	Peripheral function	Port Function
7	PWM emergency stop input when an overcurrent is detected	I	POEG	GTETRGA
85	Measurement of the W-phase current	I	ADC(S12AD1)	AN102
87	Measurement of the U-phase current	I	ADC(S12AD1)	AN100
84	Measurement of the inverter bus voltage	I	ADC(S12AD1)	AN103
70	Controlling whether to turn on LED1	O	I/O Port (output)	P65
71	Controlling whether to turn on LED2	O	I/O Port (output)	P64
45	PWM output (U_p) / "High" active	O	GPT	GTIOC4A
48	PWM output (U_n) / "High" active	O	GPT	GTIOC4B
46	PWM output (V_p) / "High" active	O	GPT	GTIOC5A
49	PWM output (V_n) / "High" active	O	GPT	GTIOC5B
47	PWM output (W_p) / "High" active	O	GPT	GTIOC6A
50	PWM output (W_n) / "High" active	O	GPT	GTIOC6B
84	Measurement of the inverter bus voltage	I	ADC(S12AD1)	AN103
78	Function for inputting the rotation speed command value (analog value)	I	ADC(S12AD2)	P55/ AN203
16	START/STOP toggle switch	I	I/O Port (Input)	PE1
96	ERROR RESET push switch	I	I/O Port (Input)	P82

3. Quick start guide

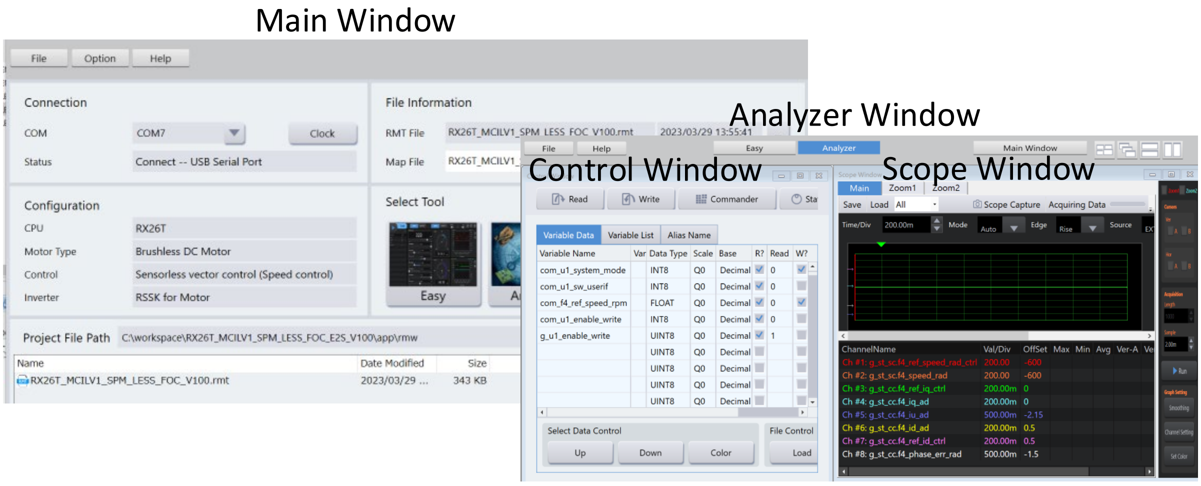
This chapter provides a quick start guide for you to drive a motor by using Renesas Flexible Motor Control Kit and the sample program. For details about the board configuration and connection procedures of Renesas Flexible Motor Control Kit, see the MCK-RX26T User's Manual (R12UZ0111). For details about how to use Renesas Motor Workbench (RMW), see the "Renesas Motor Workbench User's Manual" (R21UZ0004).

3.1 Downloading and writing the sample program

After you have downloaded the sample program from our website, use an integrated development environment (IDE) or Renesas Flash Programmer to write it to the MCU on the CPU board. For details about how to write programs, see the documentation for the IDE you use or Renesas Flash Programmer.

3.2 Analyzer startup and the RMT file

Use Renesas Motor Workbench, a motor control development support tool, as a user interface (for issuing the rotation start/stop command, rotation speed command, and other commands). Renesas Motor Workbench (RMW) can be downloaded from our website.



COM: COM7

Status: Connect -- USB Serial Port

RMT File: RX26T_MCILV1_SPM_LESS_FOC_V100.rmt 2023/03/29 13:55:41

Map File: RX26T_MCILV1_

Configuration

CPU: RX26T

Motor Type: Brushless DC Motor

Control: Sensorless vector control (Speed control)

Inverter: RSSK for Motor

Project File Path: C:\workspace\RX26T_MCILV1_SPM_LESS_FOC_E2S_V100\app\rmw

Name	Date Modified	Size
RX26T_MCILV1_SPM_LESS_FOC_V100.rmt	2023/03/29 ...	343 KB

Variable Name	Var	Data Type	Scale	Base	R?	Read	W?
com_u1_system_mode		INT8	Q0	Decimal	☑	0	☑
com_u1_sw_userif		INT8	Q0	Decimal	☑	0	□
com_f4_ref_speed_rpm		FLOAT	Q0	Decimal	☑	0	☑
com_u1_enable_write		INT8	Q0	Decimal	☑	0	□
g_u1_enable_write		UINT8	Q0	Decimal	☑	1	□
		UINT8	Q0	Decimal	□		□
		UINT8	Q0	Decimal	□		□
		UINT8	Q0	Decimal	□		□
		UINT8	Q0	Decimal	□		□

Load: All

Time/Div: 200.00m Mode: Auto Edge: Rise Source

ChannelName	Val/Div	OffSet	Max	Min	Avg	Ver-A	Ve
Ch #1: g_st_sc.f4_ref_speed_rad_ctrl	200.00	-600					
Ch #2: g_st_sc.f4_speed_rad	200.00	-600					
Ch #3: g_st_cc.f4_ref_iq_ctrl	200.00m	0					
Ch #4: g_st_cc.f4_iq_ad	200.00m	0					
Ch #5: g_st_cc.f4_iu_ad	500.00m	-2.15					
Ch #6: g_st_cc.f4_id_ad	200.00m	0.5					
Ch #7: g_st_cc.f4_ref_id_ctrl	200.00m	0.5					
Ch #8: g_st_cc.f4_phase_err_rad	500.00m	-1.5					

Figure3-1 Windows of Renesas Motor Workbench

How to use Renesas Motor Workbench (motor control development support tool)

- Click the Renesas Motor Workbench icon to start the tool.
- On the menu bar of the Main Window, select [File] > [Open RMT File].
  The RMT file in the "rmw" folder in the project folder is loaded.
- In the [Connection] area, from the [COM] drop-down list, select the COM of the connected kit.
- In the [Select Tool] area, click the [Analyzer] button to open the Analyzer Window.
- Start driving the motor as described in Using the RMW UI. (For details, see 3.4.)

What is the RMT file?

- The RMT file is a file that stores the environmental information that was manipulated or configured by using RMW.
- If the environmental information has been saved in the RMT file, the environment can be restored with the saved information by calling the RMT file.
- If the address information of a program is changed, load the map file that was generated during program building, and then save the RMT file again.

3.3 List of variables for Analyzer functions

Table 3-1 and Table 3-2 lists the data input variables that are used when the RMW UI is used. The values input to these variables are applied to the corresponding variables in the motor module and then used for controlling the motor if the value written to the com_u1_enable_write variable matches the value of g_u1_enable_write variable. Note, however, that the variables indicated by an asterisk (*) do not depend on the value of the com_u1_enable_write variable.

Table 3-1 List of main input variables for Analyzer functions (motor 1)

Analyzer function input variable name	Type	Description
com_u1_sw_userif_a (*)	uint8_t	Switching of the user interface 0: Uses the RMW UI. (Default) 1: Uses the Board UI.
com_u1_system_mode_a (*)	uint8_t	Managing the state 0: Stop mode 1: Run mode 3: Reset
com_s2_ref_speed_rpm_a (*)	int16_t	Speed command value (mechanical angle) [rpm]
com_u1_enable_write_a	uint8_t	Whether to enable rewrite of variables for user entry The input data is applied if the values of this and g_u1_enable_write_a variables are the same.

Table 3-2 List of main input variables for Analyzer functions (motor 2)

Analyzer function input variable name	Type	Description
com_u1_sw_userif_b (*)	uint8_t	Switching of the user interface 0: Uses the RMW UI. (Default) 1: Uses the Board UI.
com_u1_system_mode_b (*)	uint8_t	Managing the state 0: Stop mode 1: Run mode 3: Reset
com_s2_ref_speed_rpm_b (*)	int16_t	Speed command value (mechanical angle) [rpm]
com_u1_enable_write_b	uint8_t	Whether to enable rewrite of variables for user entry The input data is applied if the values of this and g_u1_enable_write_b variables are the same.

Table 3-3 and Table 3-4 lists main structure variables that are often observed when driving the motor under speed control was evaluated. Use this table for reference when the waveform is to be displayed or the values of variables are to be loaded with an Analyzer function. For details about the variables that are not listed in this table, see 5.1.5.

Table 3-3 List of main variables for sensorless speed control (motor 1)

Main speed control variable name	Type	Description
g_st_sensorless_vector_a.u2_error_status	uint16_t	Error status
g_st_cc_a.f4_id_ref	float	d-axis current command value [A]
g_st_cc_a.f4_id_ad	float	d-axis current detection value [A]
g_st_cc_a.f4_iq_ref	float	q-axis current command value [A]
g_st_cc_a.f4_iq_ad	float	q-axis current detection value [A]
g_st_cc_a.f4_iu_ad	float	U-phase current detection value [A]
g_st_cc_a.f4_iv_ad	float	V-phase current detection value [A]
g_st_cc_a.f4_iw_ad	float	W-phase current detection value [A]
g_st_cc_a.f4_vd_ref	float	d-axis voltage command value [V]
g_st_cc_a.f4_vq_ref	float	q-axis voltage command value [V]
g_st_cc_a.f4_refu	float	U-phase voltage command value [V]
g_st_cc_a.f4_refv	float	V-phase voltage command value [V]
g_st_cc_a.f4_refw	float	W-phase voltage command value [V]
g_st_sc_a.f4_ref_speed_rad_ctrl	float	Speed command value (mechanical angle) [rad/s]
g_st_sc_a.f4_speed_rad	float	Speed detection value (mechanical angle) [rad/s]

Table 3-4 List of main variables for sensorless speed control (motor 2)

Main speed control variable name	Type	Description
g_st_sensorless_vector_b.u2_error_status	uint16_t	Error status
g_st_cc._b.f4_id_ref	float	d-axis current command value [A]
g_st_cc._b.f4_id_ad	float	d-axis current detection value [A]
g_st_cc._b.f4_iq_ref	float	q-axis current command value [A]
g_st_cc._b.f4_iq_ad	float	q-axis current detection value [A]
g_st_cc._b.f4_iu_ad	float	U-phase current detection value [A]
g_st_cc._b.f4_iv_ad	float	V-phase current detection value [A]
g_st_cc._b.f4_iw_ad	float	W-phase current detection value [A]
g_st_cc._b.f4_vd_ref	float	d-axis voltage command value [V]
g_st_cc._b.f4_vq_ref	float	q-axis voltage command value [V]
g_st_cc._b.f4_refu	float	U-phase voltage command value [V]
g_st_cc._b.f4_refv	float	V-phase voltage command value [V]
g_st_cc._b.f4_refw	float	W-phase voltage command value [V]
g_st_sc._b.f4_ref_speed_rad_ctrl	float	Speed command value (mechanical angle) [rad/s]
g_st_sc._b.f4_speed_rad	float	Speed detection value (mechanical angle) [rad/s]

3.4 Using the RMW UI

3.4.1 Analyzer operation example

The following shows an example of using the Analyzer function to perform operations on the motor. The operations are performed from the Control Window. For details about the Control Window, see the "Renesas Motor Workbench User's Manual".

In the initial state, the control loop is set for speed control. Perform operations by referring to the procedures shown below.

Variables with "_a" in the variable name are for motor 1 control. If you want to control motor 2, use a variable with "_b" in the variable name. The following operation example is described only for the variable for motor 1 control.

(a) Start rotation of the motor

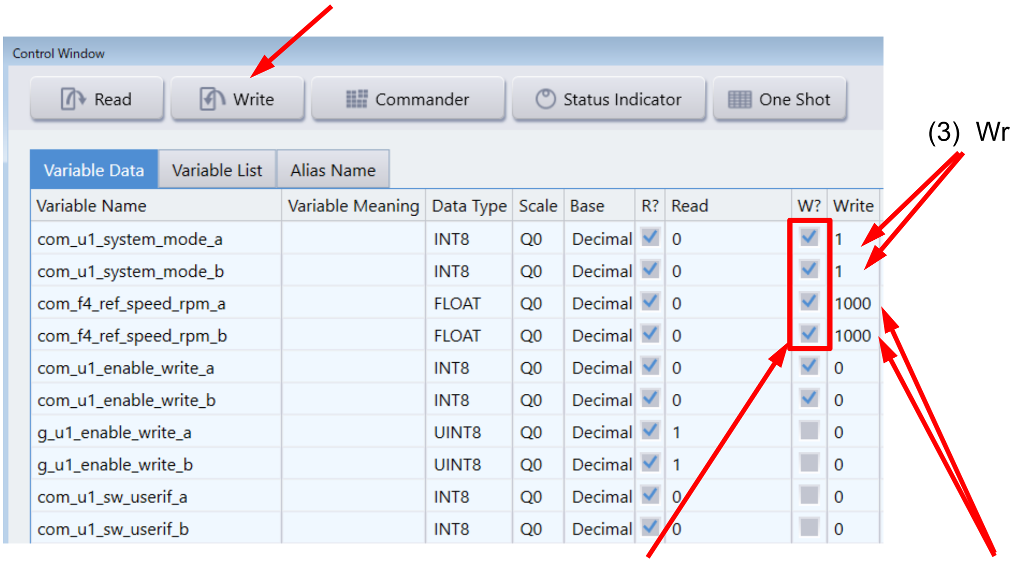
(1) Confirm that the check boxes in the [W?] column are selected on the "com_u1_system_mode_a" and "com_f4_ref_speed_rpm_a" rows.

(2) On the "com_s2_ref_speed_rpm" row, in the [Write] column, enter the command rotation speed.

(3) On the "com_u1_mode_system" row, in the [Write] column, enter "1".

(4) Click the [Write] button.

(4) Click “Write” button.



Variable Name	Variable Meaning	Data Type	Scale	Base	R?	Read	W?	Write
com_u1_system_mode_a		INT8	Q0	Decimal	☑	0	☑	1
com_u1_system_mode_b		INT8	Q0	Decimal	☑	0	☑	1
com_f4_ref_speed_rpm_a		FLOAT	Q0	Decimal	☑	0	☑	1000
com_f4_ref_speed_rpm_b		FLOAT	Q0	Decimal	☑	0	☑	1000
com_u1_enable_write_a		INT8	Q0	Decimal	☑	0	☑	0
com_u1_enable_write_b		INT8	Q0	Decimal	☑	0	☑	0
g_u1_enable_write_a		UINT8	Q0	Decimal	☑	1	□	0
g_u1_enable_write_b		UINT8	Q0	Decimal	☑	1	□	0
com_u1_sw_userif_a		INT8	Q0	Decimal	☑	0	□	0
com_u1_sw_userif_b		INT8	Q0	Decimal	☑	0	□	0

(3) Write “1”.

(1) Check.

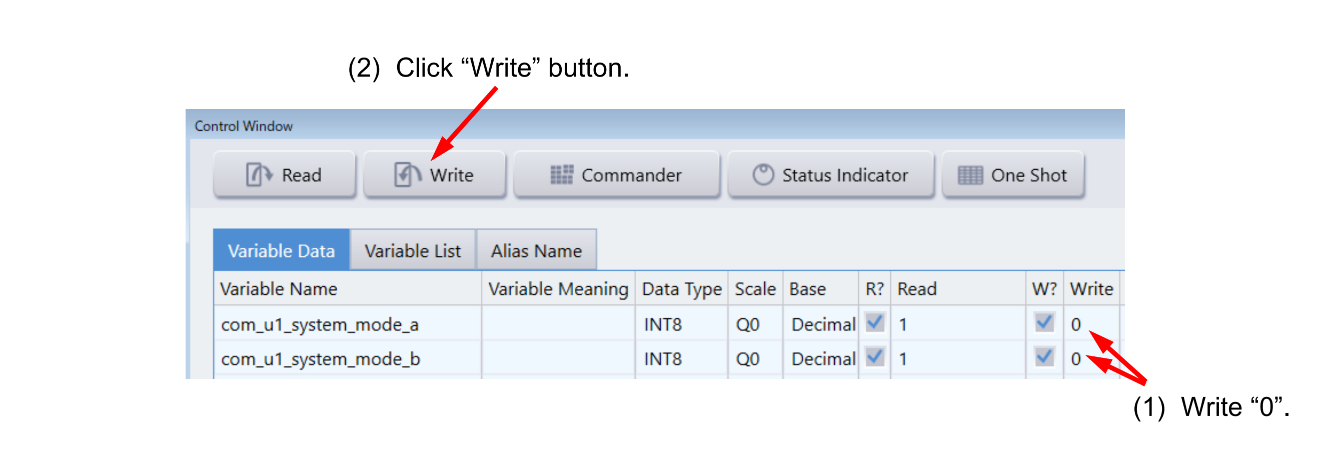
(2) Write reference speed.

Figure3-2 Procedure for starting rotation of the motor

(b) Stop the motor

(1) On the "com_u1_system_mode" row, in the [Write] column, enter "0".

(2) Click the [Write] button.



Variable Name	Variable Meaning	Data Type	Scale	Base	R?	Read	W?	Write
com_u1_system_mode_a		INT8	Q0	Decimal	☑	1	☑	0
com_u1_system_mode_b		INT8	Q0	Decimal	☑	1	☑	0

(2) Click “Write” button.

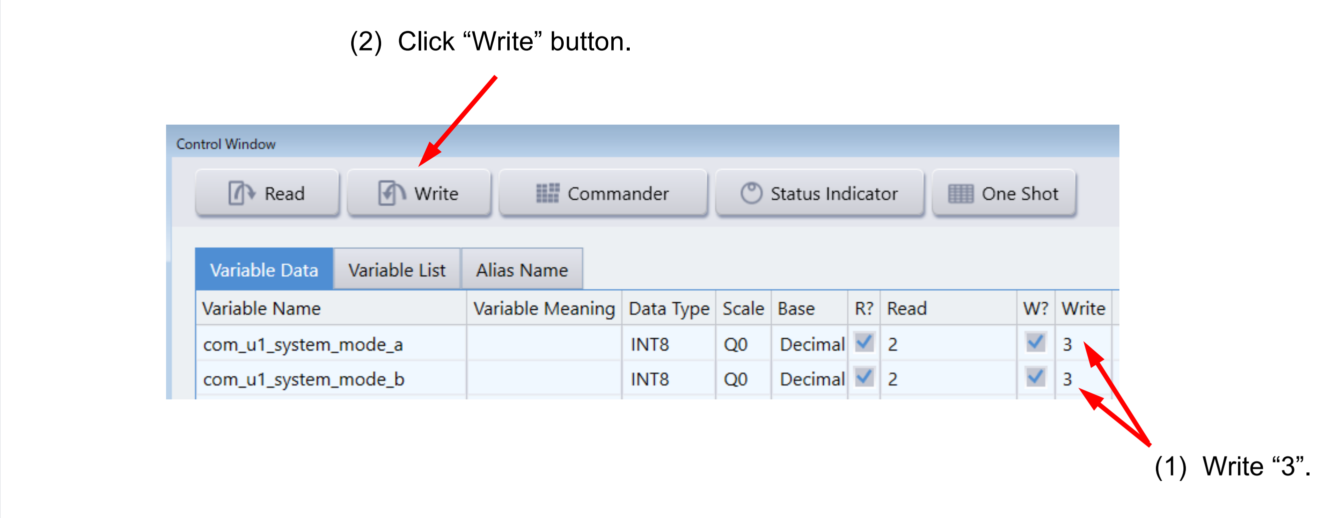
(1) Write “0”.

Figure3-3 Procedure for stopping the motor

(c) What to do in case of motor stoppage (due to an error)

(1) On the "com_u1_system_mode" row, in the [Write] column, enter "3".

(2) Click the [Write] button.



Variable Name	Variable Meaning	Data Type	Scale	Base	R?	Read	W?	Write
com_u1_system_mode_a		INT8	Q0	Decimal	☑	2	☑	3
com_u1_system_mode_b		INT8	Q0	Decimal	☑	2	☑	3

(2) Click “Write” button.

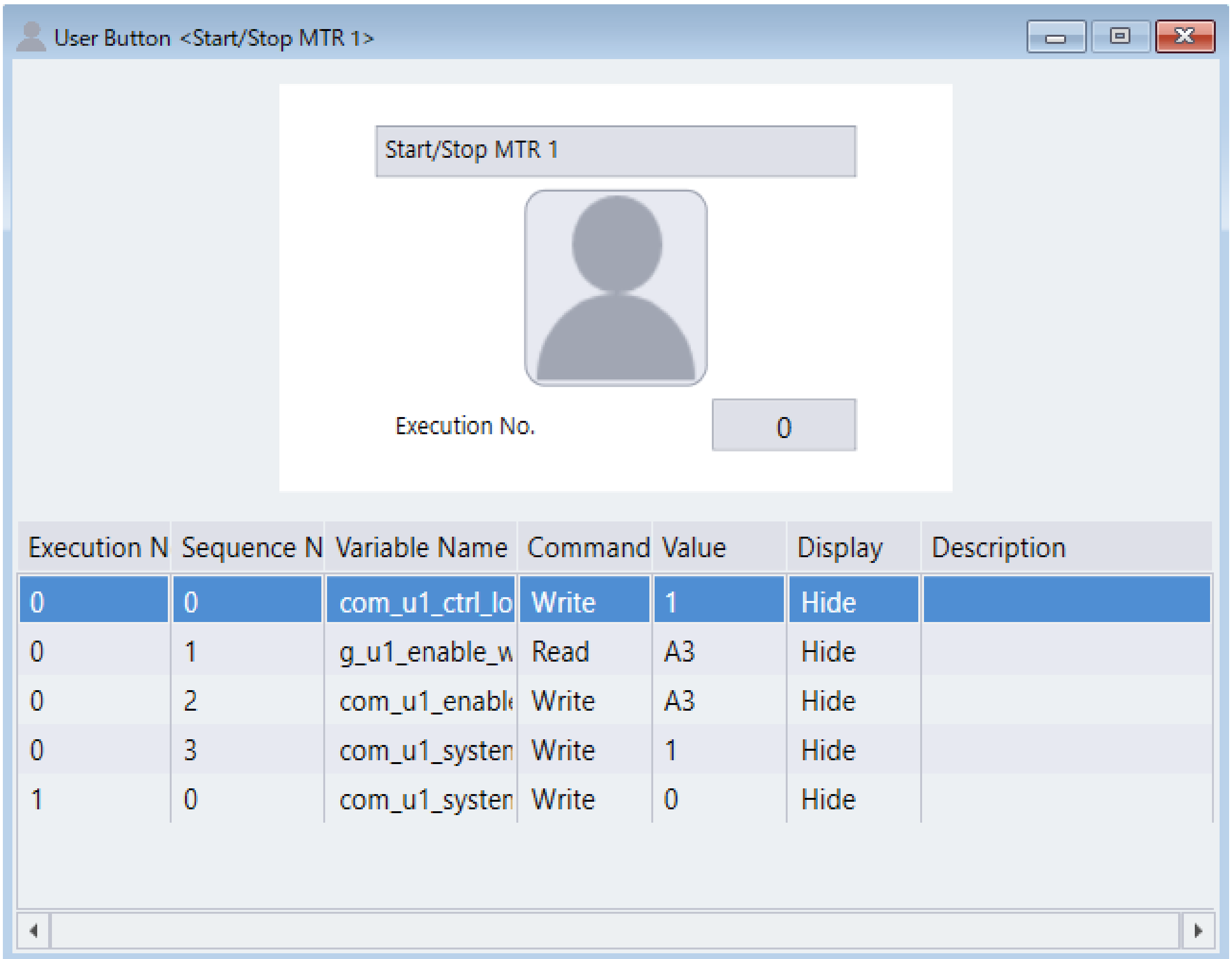
(1) Write “3”.

Figure3-4 Procedure for handling an error

3.4.2 Operation example of the User Button function

The following shows an example of using the User Button function to perform operations on the motor. The user button settings used in this example are already included in the RMT file.

- Starting or stopping motor rotation by speed control
  By specifying settings as shown in Figure3-5, each click of the button switches between starting and stopping.

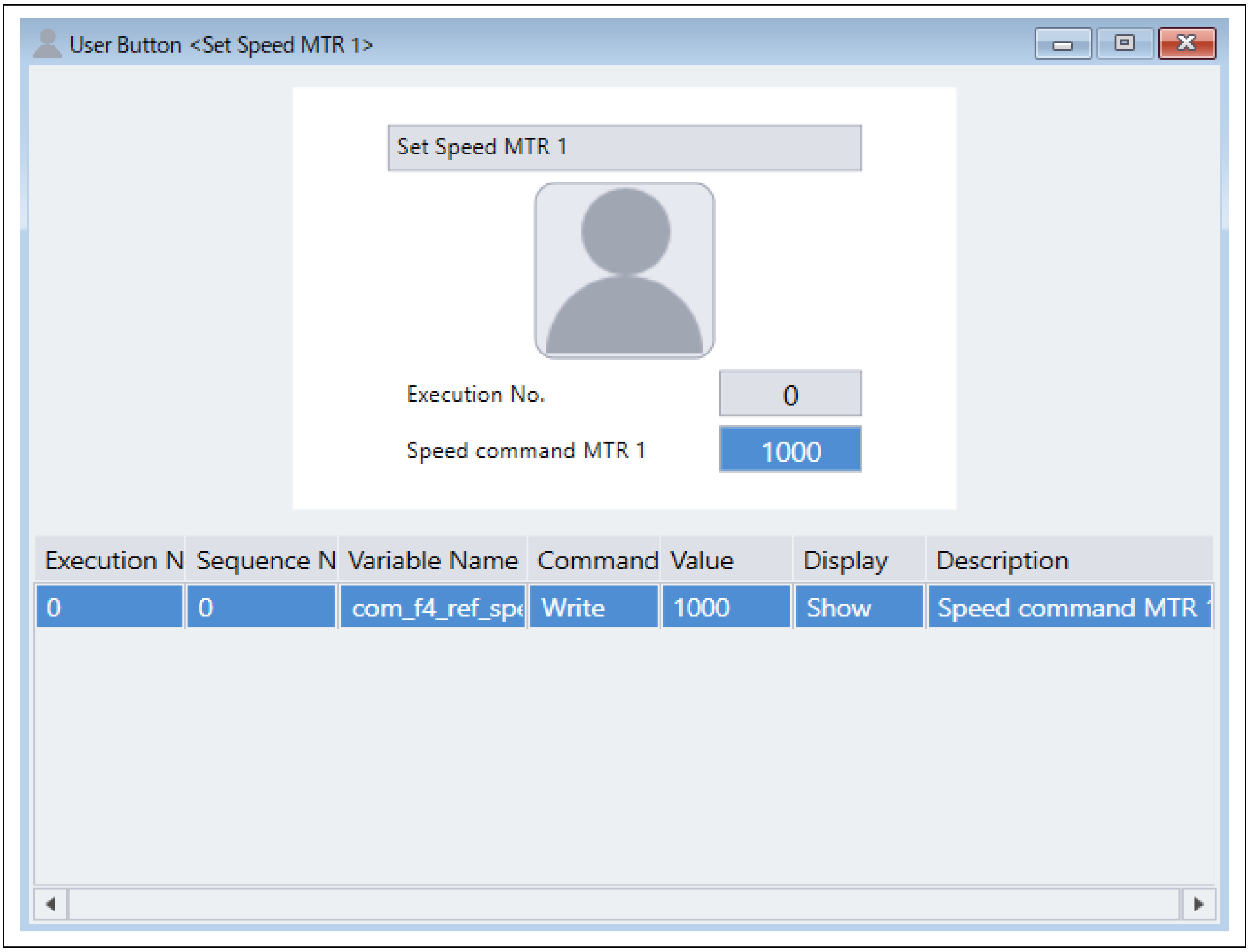


Execution No. 0

Execution N	Sequence N	Variable Name	Command	Value	Display	Description
0	0	com_u1_ctrl_lo	Write	1	Hide	
0	1	g_u1_enable_w	Read	A3	Hide	
0	2	com_u1_enabl	Write	A3	Hide	
0	3	com_u1_systen	Write	1	Hide	
1	0	com_u1_systen	Write	0	Hide	

Figure3-5 Starting or stopping motor rotation

- Changing the speed command
  By specifying settings as shown in Figure3-6, the speed command can be changed by entering the desired information and then clicking the button.



Execution No. 0

Speed command MTR 1 1000

Execution N	Sequence N	Variable Name	Command	Value	Display	Description
0	0	com_f4_ref_spe	Write	1000	Show	Speed command MTR 1

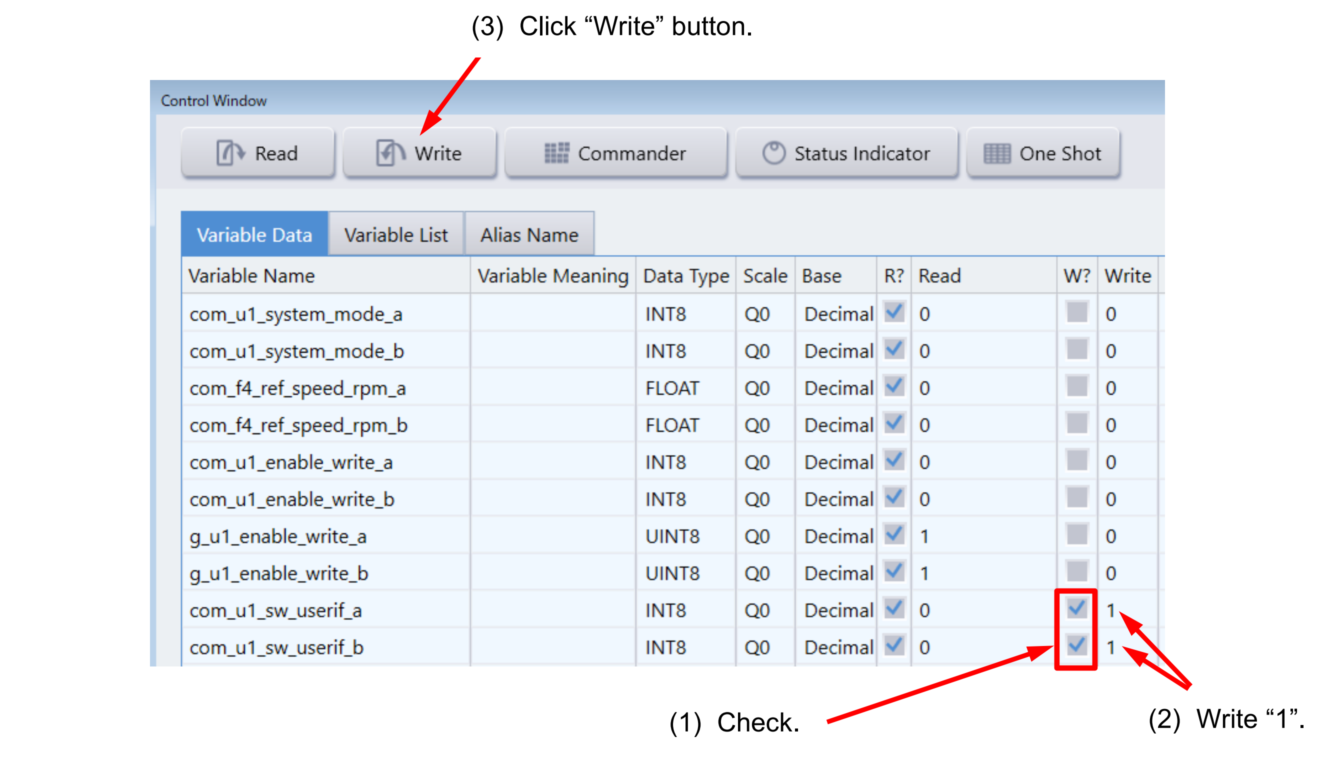
Figure3-6 Changing the speed command

3.5 Using the Board UI

3.5.1 Switching the user interface

In the sample program, the RMW UI has been set as the user interface. To change the user interface to the Board UI, perform the following procedure.

On the "com_u1_sw_userif_a”, "com_u1_sw_userif_b” row, confirm that the check mark in the [W?] column is selected, and then enter "1" in the [Write] column. Click the [Write] button.



(3) Click “Write” button.

Variable Name	Variable Meaning	Data Type	Scale	Base	R?	Read	W?	Write
com_u1_system_mode_a		INT8	Q0	Decimal	☑	0	☐	0
com_u1_system_mode_b		INT8	Q0	Decimal	☑	0	☐	0
com_f4_ref_speed_rpm_a		FLOAT	Q0	Decimal	☑	0	☐	0
com_f4_ref_speed_rpm_b		FLOAT	Q0	Decimal	☑	0	☐	0
com_u1_enable_write_a		INT8	Q0	Decimal	☑	0	☐	0
com_u1_enable_write_b		INT8	Q0	Decimal	☑	0	☐	0
g_u1_enable_write_a		UINT8	Q0	Decimal	☑	1	☐	0
g_u1_enable_write_b		UINT8	Q0	Decimal	☑	1	☐	0
com_u1_sw_userif_a		INT8	Q0	Decimal	☑	0	☑	1
com_u1_sw_userif_b		INT8	Q0	Decimal	☑	0	☑	1

(1) Check.

(2) Write “1”.

Figure3-7 Procedure for switching the UI

3.5.2 Starting or stopping the motor

If the Board UI is used, start and stop of motor rotation are controlled by the input from the SW1-1, SW1-2 switch on the inverter board (via the Board UI). A general-purpose port is assigned to the SW1-1, SW1-2. The system determines whether to start or stop motor rotation by reading a pin in the main loop. If the switch is ON, the system judges that the motor is to be started. If the switch is driven OFF, the system judges that the motor is to be stopped.

3.5.3 Motor rotation speed command value

The motor rotation speed command value is determined by performing A/D conversion for the output value (analog value) of the VR1-1, VR1-2 on the inverter board. The VR1-1, VR1-2 value after A/D conversion is used as the rotation speed command value as shown in the following table.

Table 3-1 Conversion ratio of the command value

Item	**Conversion ratio (<command-value>: <value-after-A/D-conversion>)**	
Rotation speed command value	CW	0 to 2400 [rpm]: 07FFH to 0000H
	CCW	0 to -2400 [rpm]: 0800H to 0FFFH

4. Software

4.1 Software specifications

The following shows the basic software specifications of this system.

Motor 1 and Motor 2 are identical except for Protection stops processing.

Table 4-1 Basic specifications of sensorless vector control software

<table>
<tr><th>Item</th><th colspan="2">Description</th></tr>
<tr><td>Control method</td><td colspan="2">Vector control</td></tr>
<tr><td>Starting/stopping motor control</td><td colspan="2">Determined by the level of SW1 (“ON”: start control; “OFF”: stop)
or input from RMW</td></tr>
<tr><td>Rotor magnetic pole position detection</td><td colspan="2">Sensorless</td></tr>
<tr><td>Input voltage</td><td colspan="2">DC 24V</td></tr>
<tr><td>Carrier frequency (PWM)</td><td colspan="2">20 [kHz], Carrier cycle: 50 [μs]</td></tr>
<tr><td>Dead time</td><td colspan="2">2 [μs]</td></tr>
<tr><td>Current detection</td><td colspan="2">2 shunt detection</td></tr>
<tr><td>Control cycle (current)</td><td colspan="2">50 [μs]</td></tr>
<tr><td>Control cycle (speed)</td><td colspan="2">500 [μs]</td></tr>
<tr><td>Speed command value management</td><td colspan="2">CW: 0 to 2400 [rpm]
CCW: 0 to -2400 [rpm]
When less than 600 [rpm], the motor is driven under speed open-loop control*[1]</td></tr>
<tr><td>Natural frequency for each control system</td><td colspan="2">Current control system: 300 [Hz]
Speed control system: 3 [Hz]
Inductive voltage estimation system: 1000 [Hz]
Position estimation system: 20 [Hz]</td></tr>
<tr><td rowspan="2">Compiler optimization settings</td><td>Optimization level</td><td>2 (-optimize = 2) (default)</td></tr>
<tr><td>Optimization method</td><td>Optimization focusing on the code size (-size) (default)</td></tr>
<tr><td>Protection stops processing</td><td>[Motor1]
The motor control signal output (six outputs) will be deactivated when any of the following conditions are met:
1. The currents of all phases exceed 3.54 [A] (checked at current control cycle).
2. The inverter bus voltage exceeds 60 [V] (checked at current control cycle).
3. The inverter bus voltage is less than 8 [V] (checked at current control cycle).
4. The rotation speed exceeds 4500 [rpm] (checked at current control cycle).

When the overcurrent detection signal POE3 from an external or an output short-circuit is detected, the MTU3 PWM output pin is driven to high impedance.</td><td>[Motor2]
The motor control signal output (six outputs) will be deactivated when any of the following conditions are met:
1 The currents of all phases exceed 3.54 [A] (checked at current control cycle).
2 The inverter bus voltage exceeds 60 [V] (checked at current control cycle).
3 The inverter bus voltage is less than 8 [V] (checked at current control cycle).
4 The rotation speed exceeds 4500 [rpm] (checked at current control cycle).

When the overcurrent detection signal POEG from an external or an output short-circuit is detected, the MTU3 PWM output pin is driven to high impedance.</td></tr>
</table>

Note: 1. When you spin the motor with sensorless vector control, set a rotation speed command value higher than 600 [rpm].

4.2 Software configuration

Each sample program consists of the application layer, motor module, and Smart Configurator. The motor module performs control as requested from the application layer controlled by the user. The output from the motor module is transferred by Smart Configurator to the hardware layer.

4.2.1 Overall configuration

Figure4-1 shows the overall configuration of the software.

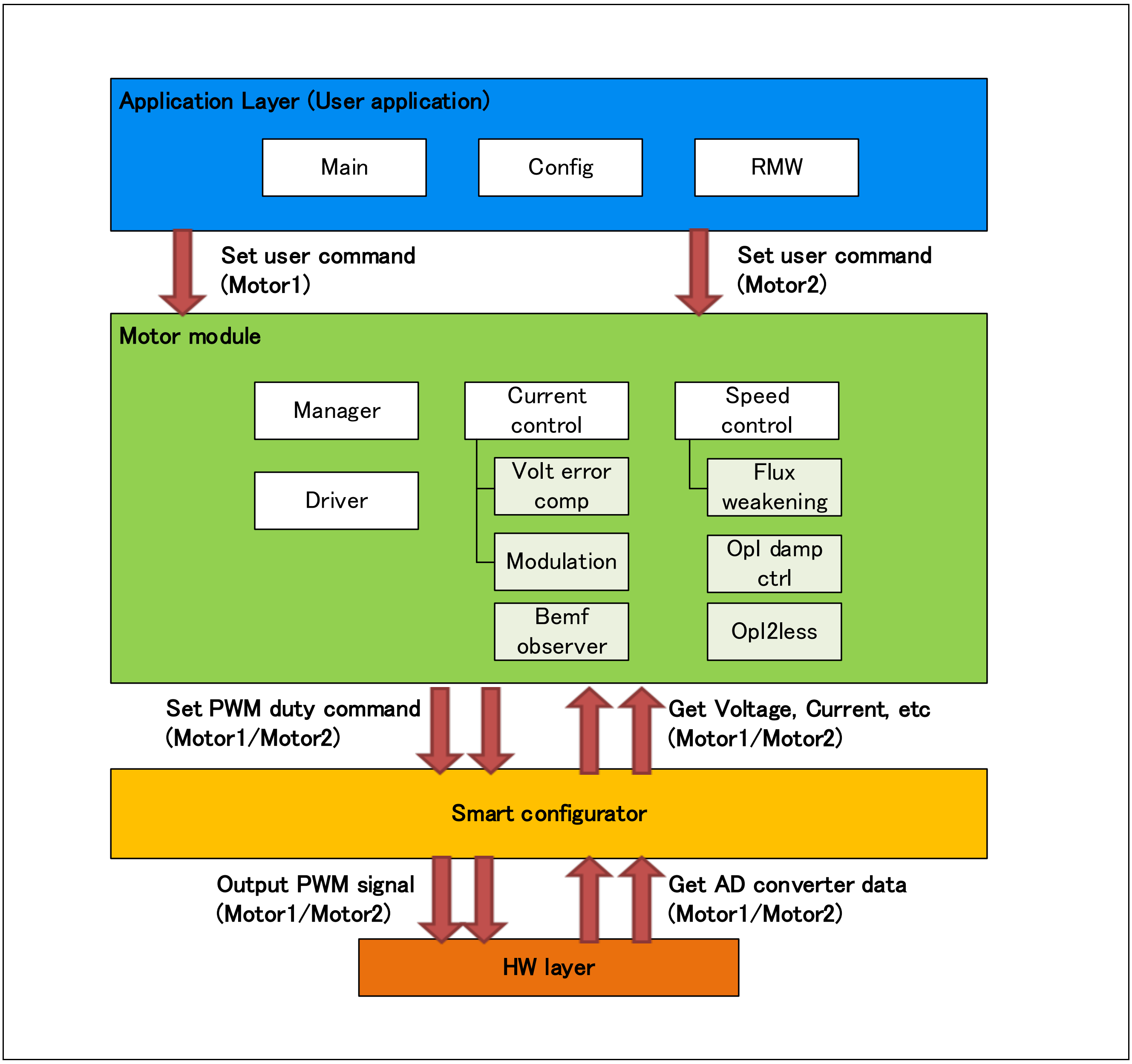

Application Layer (User application)
Main
Config
RMW
Set user command (Motor1)
Set user command (Motor2)
Motor module
Manager
Driver
Current control
Volt error comp
Modulation
Bemf observer
Speed control
Flux weakening
Opl damp ctrl
Opl2less
Set PWM duty command (Motor1/Motor2)
Get Voltage, Current, etc (Motor1/Motor2)
Smart configurator
Output PWM signal (Motor1/Motor2)
Get AD converter data (Motor1/Motor2)
HW layer

Figure4-1 shows the overall configuration of the motor control software.

4.2.2 Configuration of the motor module

Figure4-2 shows the configuration of the motor module. Table 4-2 provides a summary of each module. The manager module works as an interface between other modules and performs data acquisition and setting for the appropriate modules.

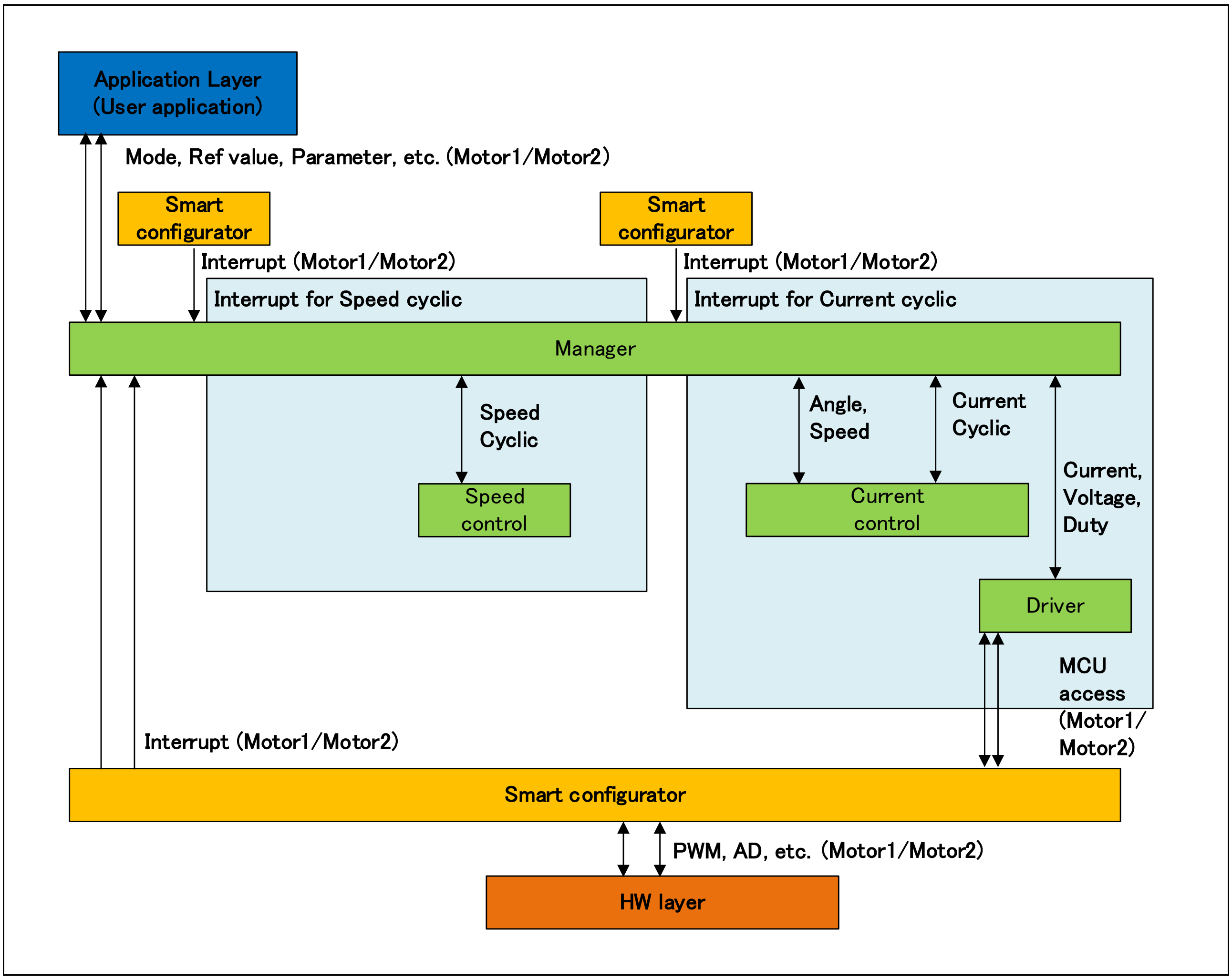

Application Layer
(User application)
Mode, Ref value, Parameter, etc. (Motor1/Motor2)
Smart configurator
Smart configurator
Interrupt (Motor1/Motor2)
Interrupt (Motor1/Motor2)
Interrupt for Speed cyclic
Interrupt for Current cyclic
Manager
Speed Cyclic
Speed control
Angle, Speed
Current Cyclic
Current control
Current, Voltage, Duty
Driver
MCU access (Motor1/ Motor2)
Interrupt (Motor1/Motor2)
Smart configurator
PWM, AD, etc. (Motor1/Motor2)
HW layer

Figure4-2 Configuration of the motor module

Table 4-2 Module summary

Module	Description	Section
Application layer	Main processing and user area	5.1
Manager module	Management of the overall sample program and interface with each module	5.2
Current control module	Module related to current control	5.3
Speed control module	Module related to speed control	5.7
Driver module	Module related to connection with Smart Configurator	5.11
Smart Configurator layer	Module related to connection with the hardware layer	5.12

4.3 File and folder configuration

Table 4-3 shows the folder and file configuration of the sample program.

Table 4-3 File and folder configuration

Folder	Subfolder	File	Remarks
app	main	r_app_main.c/h	User main function
	rmw	r_app_rmw.c/h	Definition of functions related to the RMW Analyzer UI
		r_app_rmw_interrupt.c	Definition of RMW interrupt functions
		ICS2_RX26T.lib/h	Library for RMW communication
	board_ui	r_app_board_ui.c/h	Definition of functions related to the Board UI
		r_app_board_ui_ctrl.h	Definition of MCU-dependent functions of the Board UI
		r_app_board_ui_ctrl_rx26t_mcilv1.c	Definition of MCU-dependent functions of the Board UI
	cfg	r_app_control_cfg.h	Configuration definition for the application layer
motor_module	sensorless_ vector_rx	r_motor_sensorless_vector_action.c	Definition of action functions
		r_motor_sensorless_vector_api.c/h	Definition of API functions for the manager module
		r_motor_sensorless_vector_manager.c/h	Definition of local functions for the manager module
		r_motor_sensorless_vector_protection.c/h	Definition of functions for the protection function
		r_motor_sensorless_vector_ statemachine.c/h	Definition of functions related to state transition
	current_rx	r_motor_current_api.c/h	Definition of API functions for the current control module
		r_motor_current.c/h	Definition of local functions for the current control module
		r_motor_current_modulation.c/h	Definition of functions for the modulation module
		r_motor_current_volt_err_comp.lib/h	Definition of functions for the voltage error compensation module
		r_motor_current_bemf_observer.lib/h	Definition of functions for the inductive voltage observer
		r_motor_current_pi_gain_calc.c	Definition of functions for calculating the control gain of the current control module
	speed_rx	r_motor_speed_api.c/h	Definition of API functions for the speed control module
		r_motor_speed.c/h	Definition of local functions for the speed control module
		r_motor_speed_fluxwkn.lib/h	Definition of functions for the magnetic flux weakening module
		r_motor_speed_opl_damp_ctrl.lib/h	Definition of functions for open-loop damping control

Folder	Subfolder	File	Remarks
motor_module	speed_rx	r_motor_speed_opl2less.lib/h	Definition of functions for switching-to-sensorless processing
		r_motor_speed_pi_gain_calc.c	Definition of functions for calculating the control gain of the speed control module
	driver_rx	r_motor_driver.c/h	Definition of functions for the driver module
	general	r_motor_filter.c/h	Definition of general-purpose filter functions
		r_motor_pi_control.c/h	Definition functions for PI control
		r_motor_common.h	Common definition
	cfg	r_motor_inverter_cfg.h	Configuration definition for the inverter
		r_motor_module_cfg.h	Configuration definition for the control module
		r_motor_targetmotor_cfg.h	Configuration definition for the motor
		r_mtr_control_parameter.h	Tuning Result by the Tuner functions of RMW*[1] (Control parameters definition)
		r_mtr_motor_parameter.h	Tuning Result by the Tuner functions of RMW*[1] (Motor parameters definition)
QE_Motor	-	-	Generated files by QE for Motor
src	smc_gen	See Table 4-4.	Drivers and API functions generated by Smart Configurator

Note: 1. This file is updated when tuning is performed on QE for Motor.

Smart Configurator can be used to generate peripheral drivers easily.

Smart Configurator saves the settings information about the microcontrollers, peripheral functions, pin functions, and other items that are used for the project in a project file (*.scfg), and references the information saved in the file. To check the settings of the peripheral functions for the sample program, see the following file:

RX26T_xxx_MCILV1_2SPM_LESS_FOC_yyy_Vzzz.scfg

(In the above file name, the "xxx" portion indicates the edition: MCBA indicates for RX26T RAM64KB Version and MCBC indicates for RX26T RAM48KB Version. The "yyy" portion indicates the edition: CSP indicates the CS+ edition and E2S indicates the e^2 studio edition. The "zzz" portion indicates the revision number.)

Table 4-4 shows the configuration of the folders and files generated by Smart Configurator.

Table 4-4 Smart Configurator folder and file configurations (Example for 2 shunt detection)

Folder	Subfolder	2nd subfolder	File	Remarks
src	smc_gen	Config_S12AD2	Config_S12AD2.c/h	Definition of functions related to 12-bit ADC
			Config_S12AD2_user.c	Definition of user functions related to 12-bit ADC
		Config_PORT	Config_PORT.c/h	Definition of functions related to ports
			Config_PORT_user.c	Definition of user functions related to ports
		Config_CMT0	Config_CMT0.c/h	Definition of functions related to CMT for the control interval
			Config_CMT0_user.c	Definition of user functions related to CMT for the control interval
		Config_MOTOR_A	Config_MOTOR_A.c/h	Definition of functions related to the Motor component (INV 1)
			Config_MOTOR_A_user.c	Definition of user functions related to the Motor component (INV 1)
		Config_MOTOR_B	Config_MOTOR_B.c/h	Definition of functions related to the Motor component (INV 2)
			Config_MOTOR_B_user.c	Definition of user functions related to the Motor component (INV 2)
		Config_IWDT	Config_IWDT.c/h	Definition of functions related to IWDT
			Config_IWDT_user.c	Definition of user functions related to IWDT
		Config_POE	Config_POE.c/h	Definition of functions related to POE
			Config_POE_user.c	Definition of user functions related to POE
		Config_POEG	Config_POEG.c/h	Definition of functions related to POEG
			Config_POEG_user.c	Definition of user functions related to POEG

In addition to the table above, the following four folders are automatically generated when Smart Configurator is used:

r_bsp: This folder contains various BSP (Board Support Package) files. For details, see the "readme.txt" file in the "r_bsp" folder.

general: This folder contains various files that are shared by Smart Configurator generation drivers.

r_config: This folder contains the configuration header files for the MCU package, clocks, interrupts, and driver initialization functions that have names in the "R_xxx_Open" pattern.

r_pincfg: This folder contains various files related to pin settings.

5. Functionality

Motor 1 and Motor 2 have the same control specifications.

5.1 Application layer

The application layer is used for selecting the user interface (UI), setting command values for controlling motor modules that use RMW, and updating parameters for control modules. Two user interfaces (configured and processed in the sample program) are used: the Board UI, which uses the switches and variable resistor on the inverter board to drive the motor, and the RMW UI, which uses RMW to drive the motor. These UIs are also used to control whether to drive or stop the motor and to set control command values.

5.1.1 Functions

Table 5-1 lists the functions that are configured in the application layer.

Table 5-1 Functions available in the application layer

Function	Description
Main processing	Enables or disables each user command in the system.
UI processing	Selects the Board UI or RMW UI, and switches between these UIs.
Board UI processing	Obtains and sets command values for speed control.
RMW UI processing	Acquires and sets parameters (including command values).

5.1.2 Module configuration diagram

Figure5-1 shows the module configuration.

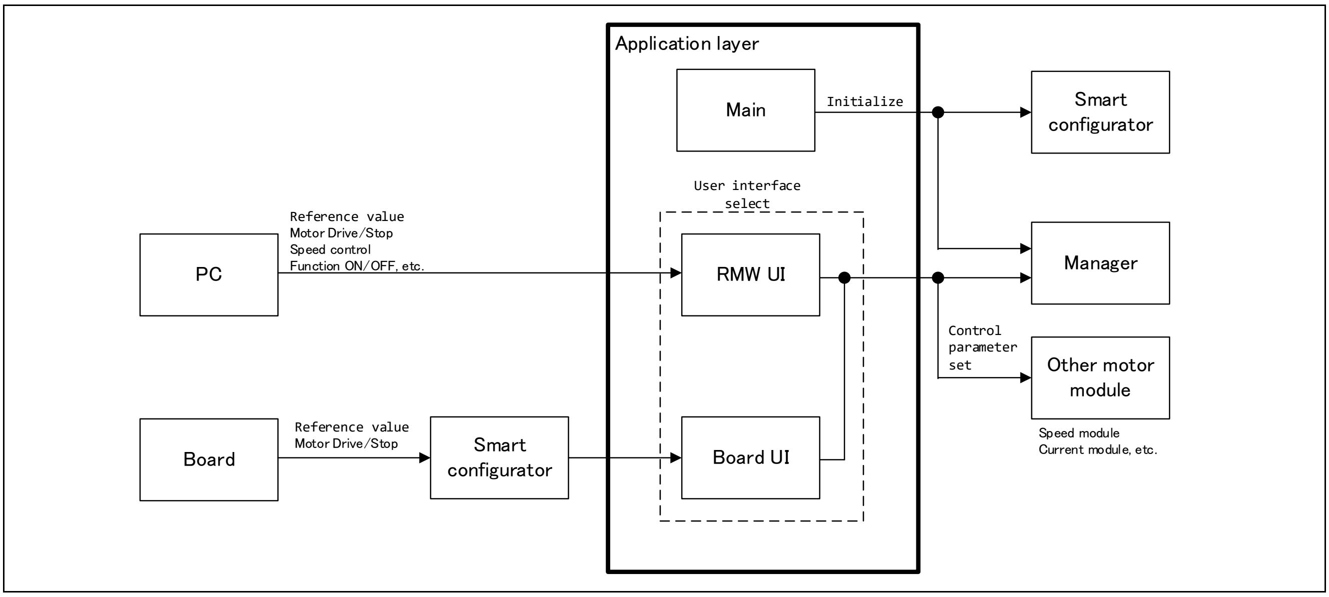


Figure5-1 Configuration of the application layer

5.1.3 Flowcharts

5.1.3.1 Main processing

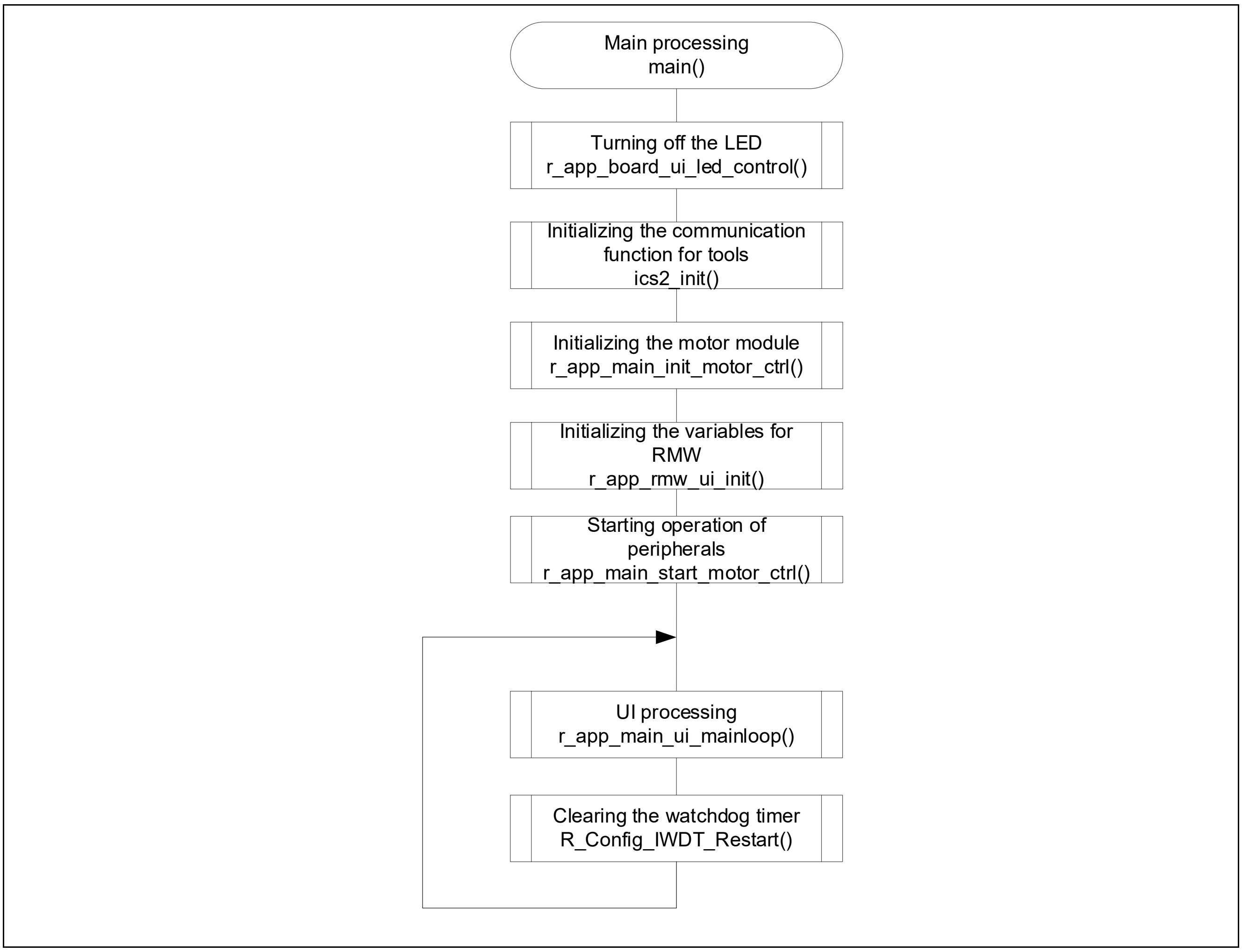

Main processing
main()
Turning off the LED
r_app_board_ui_led_control()
Initializing the communication function for tools
ics2_init()
Initializing the motor module
r_app_main_init_motor_ctrl()
Initializing the variables for RMW
r_app_rmw_ui_init()
Starting operation of peripherals
r_app_main_start_motor_ctrl()
UI processing
r_app_main_ui_mainloop()
Clearing the watchdog timer
R_Config_IWDT_Restart()

Figure5-2 Flowchart for the main processing

5.1.3.2 UI processing

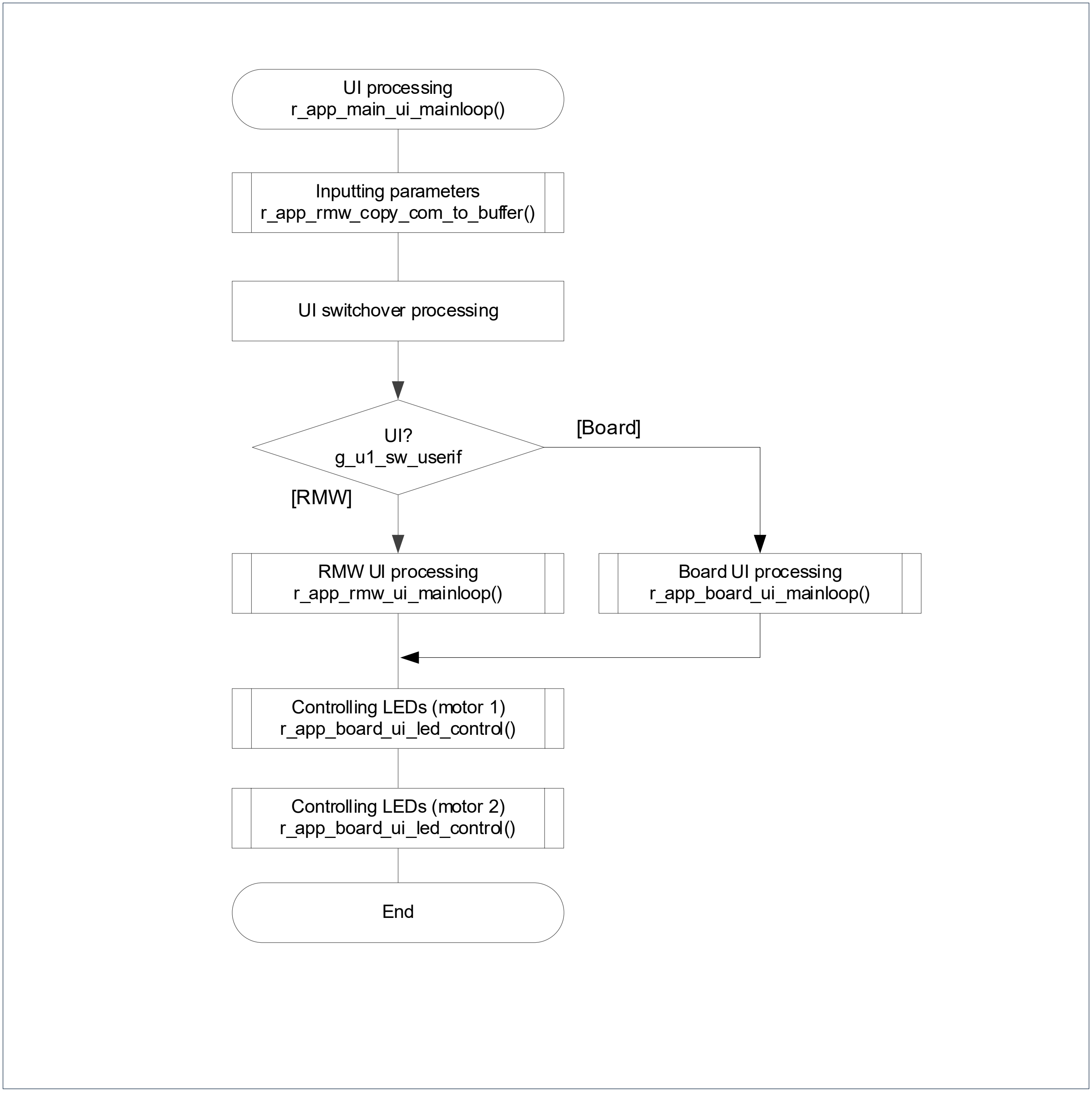


Figure5-3 Flowchart for the UI processing

5.1.3.3 Board UI processing

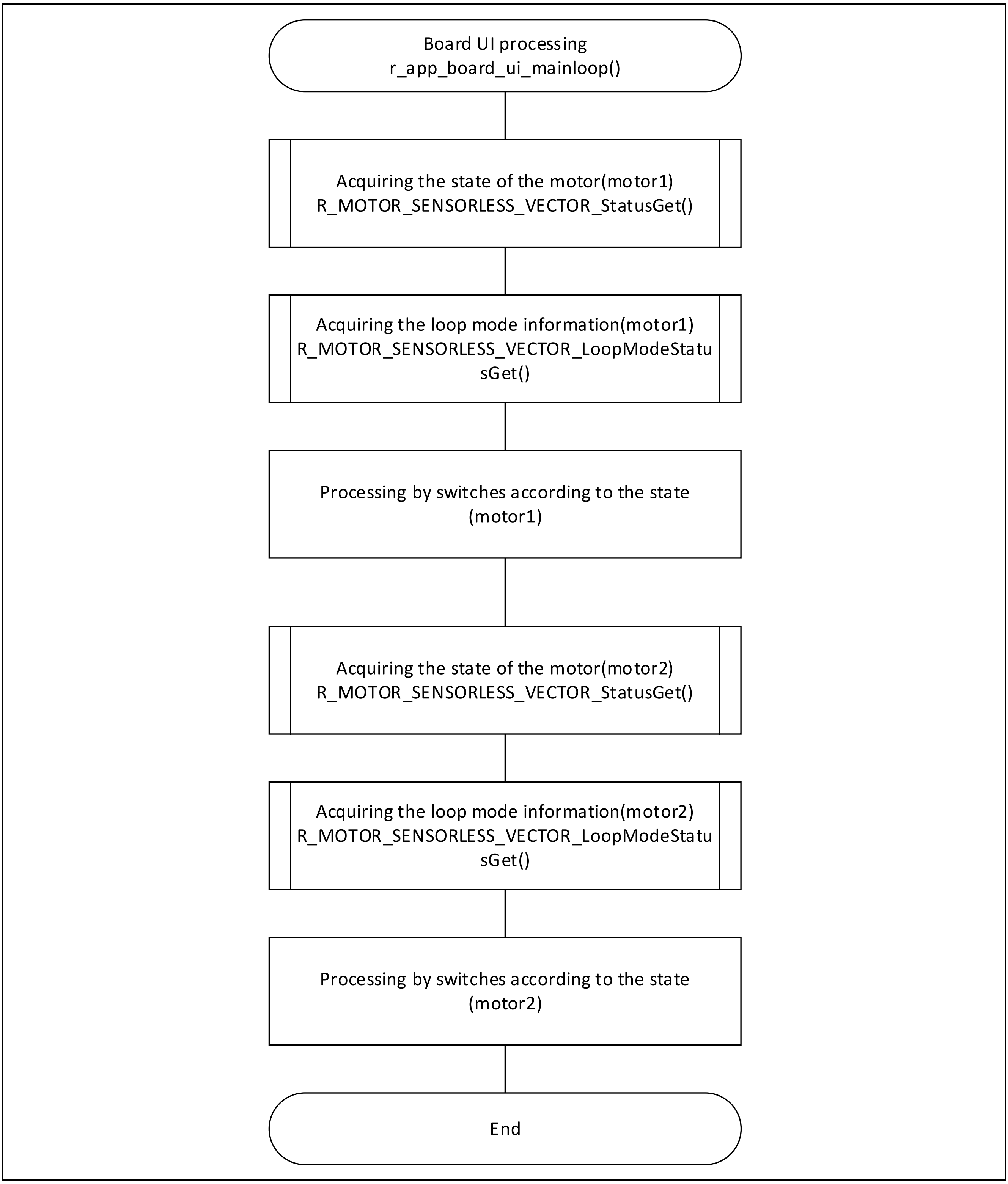

Board UI processing
r_app_board_ui_mainloop()
Acquiring the state of the motor(motor1)
R_MOTOR_SENSORLESS_VECTOR_StatusGet()
Acquiring the loop mode information(motor1)
R_MOTOR_SENSORLESS_VECTOR_LoopModeStatusGet()
Processing by switches according to the state (motor1)
Acquiring the state of the motor(motor2)
R_MOTOR_SENSORLESS_VECTOR_StatusGet()
Acquiring the loop mode information(motor2)
R_MOTOR_SENSORLESS_VECTOR_LoopModeStatusGet()
Processing by switches according to the state (motor2)
End

Figure5-4 Flowchart for the Board UI processing

5.1.3.4 RMW UI processing

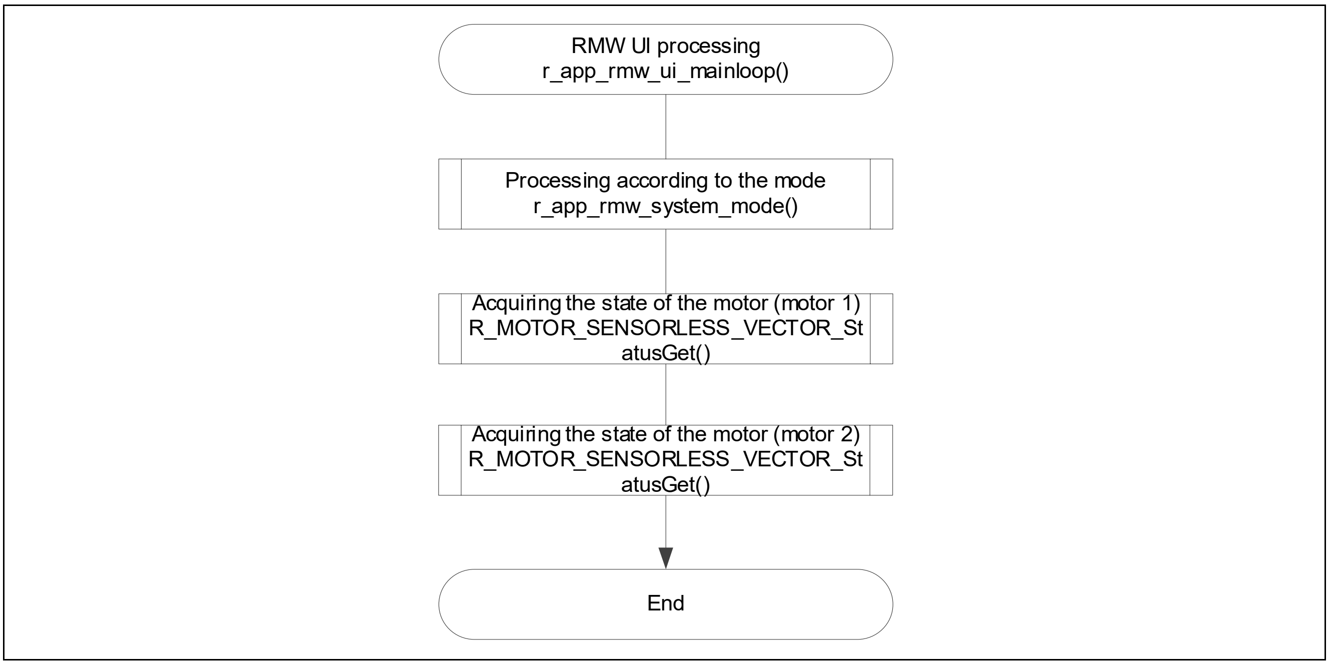

RMW UI processing
r_app_rmw_ui_mainloop()
Processing according to the mode
r_app_rmw_system_mode()
Acquiring the state of the motor (motor 1)
R_MOTOR_SENSORLESS_VECTOR_StatusGet()
Acquiring the state of the motor (motor 2)
R_MOTOR_SENSORLESS_VECTOR_StatusGet()
End

Figure5-5 Flowchart for the RMW UI processing

5.1.4 Configurations

Table 5-2, Table 5-3, and Table 5-4 shows the configurations used in the application layer.

Table 5-2 List of configurations (Motor1)

File name	Macro name	Description
r_app_control_cfg.h	APP_CFG_USE_UI_A	Initial UI setting RMW: MAIN_UI_RMW Board: MAIN_UI_BOARD
	APP_CFG_FREQ_BAND_LIMIT_A	This item sets the limit value for maintaining separation between the natural frequencies for current control, and speed control.
	APP_CFG_MAX_CURRENT_OMEGA_A	This item sets the upper limit on the natural frequency for the current control system [Hz].
	APP_CFG_MIN_OMEGA_A	This item sets the lower limit on natural frequencies [Hz].

Table 5-3 List of configurations (Motor2)

File name	Macro name	Description
r_app_control_cfg.h	APP_CFG_USE_UI_B	Initial UI setting RMW: MAIN_UI_RMW Board: MAIN_UI_BOARD
	APP_CFG_FREQ_BAND_LIMIT_B	This item sets the limit value for maintaining separation between the natural frequencies for current control, and speed control.
	APP_CFG_MAX_CURRENT_OMEGA_B	This item sets the upper limit on the natural frequency for the current control system [Hz].
	APP_CFG_MIN_OMEGA_B	This item sets the lower limit on natural frequencies [Hz].

Table 5-4 List of configurations (common)

File name	Macro name	Description
r_app_control_cfg.h	APP_CFG_SCI_CH_SELECT	This item is used to select the SCI channel for RMW.

Table 5-5 List of initial values for configurations

Macro name	Settings
APP_CFG_USE_UI_A APP_CFG_USE_UI_B	MAIN_UI_RMW
APP_CFG_FREQ_BAND_LIMIT_A APP_CFG_FREQ_BAND_LIMIT_B	3.0f
APP_CFG_MAX_CURRENT_OMEGA_A APP_CFG_MAX_CURRENT_OMEGA_B	1000.0f
APP_CFG_MIN_OMEGA_A APP_CFG_MIN_OMEGA_B	1.0f
APP_CFG_SCI_CH_SELECT	0x60 (SCI6)

5.1.5 Structure and variable information

Table 5-6, Table 5-7 lists the variables that can be used by users in the application layer. Table 5-8 lists the members of the structure provided for updating the motor module parameters by using RMW.

Table 5-6 List of variables (Motor1)

Variable	Description
g_st_rmw_input_buffer_a	Structure for updating the RMW variables (motor 1)
g_u1_update_param_flag_a	Buffer transfer completion flag (motor 1)
com_u1_system_mode_a	Variable to switch the system mode for user entry (motor 1) 0: Stopping the motor 1: Driving the motor 3: Canceling the error
g_u1_system_mode_a	System mode (motor 1) 0: Stopping the motor 1: Driving the motor 2: Error
com_u1_enable_write_a	Whether to enable rewrite of variables for user entry (motor 1)
g_u1_enable_write_a	Whether to enable rewrite of variables (motor 1)
com_u1_sw_userif_a	Variable to switch the UI for user entry (motor 1) 0: RMW UI 1: Board UI
g_u1_sw_userif_a	Variable to switch the UI (motor 1)
com_u2_offset_calc_time_a	Current offset value calculation time setting (motor 1)
com_u2_mtr_pp_a	Number of pole pairs of the motor to be driven (motor 1)
com_f4_mtr_r_a	Resistance of the motor to be driven [Ω] (motor 1)
com_f4_mtr_ld_a	d-axis inductance of the motor to be driven [H] (motor 1)
com_f4_mtr_lq_a	q-axis inductance of the motor to be driven [H] (motor 1)
com_f4_mtr_m_a	Magnetic flux of the motor to be driven [Wb] (motor 1)
com_f4_mtr_j_a	Rotor inertia of the motor to be driven [kgm^2] (motor 1)
com_f4_nominal_current_rms_a	Rated current of the motor to be driven [Arms] (motor 1)
com_f4_max_speed_rpm_a	Maximum speed (mechanical angle) of the motor to be driven [rpm] (motor 1)
com_u1_ctrl_loop_mode_a	Switching of the control loop (motor 1) 0: Position control (Not use) 1: Speed control
com_f4_ol_ref_id_a	d-axis current command value [A] (motor 1)
com_f4_id_up_time_a	Ramping-up time of the d-axis current command value setting (motor 1)
com_f4_id_down_time_a	Decreasing time of the d-axis current command value (motor 1)
com_f4_id_down_speed_rpm_a	Speed (mechanical angle) when start decreasing the d-axis current command value [rpm] (motor 1)
com_f4_id_up_speed_rpm_a	Speed (mechanical angle) when start increasing the d-axis current command value [rpm] (motor 1)

Variable	Description
com_f4_current_omega_hz_a	Natural frequency for the current control system [Hz] (motor 1)
com_f4_current_zeta_a	Attenuation coefficient for the current control system (motor 1)
com_f4_speed_omega_hz_a	Natural frequency for the speed control system [Hz] (motor 1)
com_f4_speed_zeta_a	Attenuation coefficient for the speed control system (motor 1)
com_f4_speed_lpf_hz_a	Speed LPF cut-off frequency [Hz] (motor 1)
com_f4_ref_speed_rpm_a	Speed command value (mechanical angle) [rpm] (motor 1)
com_f4_speed_rate_limit_rpm_a	Maximum increment/decrement width for the speed command [rpm/s] (used when speed control is enabled) (motor 1)
com_f4_overspeed_limit_rpm_a	Speed limit value (mechanical angle) [rpm] (motor 1)
com_u1_flag_volt_err_comp_use_a	Voltage error compensation setting (motor 1) 0: Disable 1: Enable
com_u1_flag_fluxwkn_use_a	Magnetic flux weakening control setting (motor 1) 0: Disable 1: Enable
s_u1_cnt_ics_a	ICS watchpoint skip counter (motor 1)
com_f4_e_obs_omega_hz_a	Natural frequency for the inductive voltage estimation system [Hz] (motor 1)
com_f4_e_obs_zeta_a	Attenuation coefficient for the inductive voltage estimation system (motor 1)
com_f4_pll_est_omega_hz_a	Natural frequency for the position estimation system [Hz] (motor 1)
com_f4_pll_est_zeta_a	Attenuation coefficient for the position estimation system (motor 1)
com_u1_flag_less_switch_use_a	Switching-to-sensorless processing setting (motor 1) 0: Disable 1: Enable
com_f4_switch_phase_err_deg_a	Phase error that allows switching-to-sensorless control (electrical angle) [deg] (motor 1)
com_f4_opl2less_sw_time_a	Processing time for switching-to-sensorless processing [s] (motor 1)
com_f4_phase_err_lpf_cut_freq_a	Phase error LPF cut-off frequency [Hz] (motor 1)
com_u1_flag_openloop_damping_use_a	Open-loop damping control setting (motor 1) 0: Disable 1: Enable
com_f4_ed_hpf_omega_a	d-axis inductive voltage HPF cut-off frequency [Hz] (motor 1)
com_f4_ol_damping_zeta_a	Attenuation coefficient for open-loop damping control (motor 1)
com_f4_ol_damping_fb_limit_rate_a	Feedback limit rate for open-loop damping control (motor 1)

Table 5-7 List of variables (Motor2)

Variable	Description
g_st_rmw_input_buffer_b	Structure for updating the RMW variables (motor 2)
g_u1_update_param_flag_b	Buffer transfer completion flag (motor 2)
com_u1_system_mode_b	Variable to switch the system mode for user entry (motor 2) 0: Stopping the motor 1: Driving the motor 3: Canceling the error
g_u1_system_mode_b	System mode (motor 2) 0: Stopping the motor 1: Driving the motor 2: Error
com_u1_enable_write_b	Whether to enable rewrite of variables for user entry (motor 2)
g_u1_enable_write_b	Whether to enable rewrite of variables (motor 2)
com_u1_sw_userif_b	Variable to switch the UI for user entry (motor 2) 0: RMW UI 1: Board UI
g_u1_sw_userif_b	Variable to switch the UI (motor 2)
com_u2_offset_calc_time_b	Current offset value calculation time setting (motor 2)
com_u2_mtr_pp_b	Number of pole pairs of the motor to be driven (motor 2)
com_f4_mtr_r_b	Resistance of the motor to be driven [Ω] (motor 2)
com_f4_mtr_ld_b	d-axis inductance of the motor to be driven [H] (motor 2)
com_f4_mtr_lq_b	q-axis inductance of the motor to be driven [H] (motor 2)
com_f4_mtr_m_b	Magnetic flux of the motor to be driven [Wb] (motor 2)
com_f4_mtr_j_b	Rotor inertia of the motor to be driven [kgm^2] (motor 2)
com_f4_nominal_current_rms_b	Rated current of the motor to be driven [Arms] (motor 2)
com_f4_max_speed_rpm_b	Maximum speed (mechanical angle) of the motor to be driven [rpm] (motor 2)
com_u1_ctrl_loop_mode_b	Switching of the control loop (motor 2) 0: Position control (Not use) 1: Speed control
com_f4_ol_ref_id_b	d-axis current command value [A] (motor 2)
com_f4_id_up_time_b	Ramping-up time of the d-axis current command value setting (motor 2)
com_f4_id_down_time_b	Decreasing time of the d-axis current command value (motor 2)
com_f4_id_down_speed_rpm_b	Speed (mechanical angle) when start decreasing the d-axis current command value [rpm] (motor 2)
com_f4_id_up_speed_rpm_b	Speed (mechanical angle) when start increasing the d-axis current command value [rpm] (motor 2)
com_f4_current_omega_hz_b	Natural frequency for the current control system [Hz] (motor 2)
com_f4_current_zeta_b	Attenuation coefficient for the current control system (motor 2)
com_f4_speed_omega_hz_b	Natural frequency for the speed control system [Hz] (motor 2)

Variable	Description
com_f4_speed_zeta_b	Attenuation coefficient for the speed control system (motor 2)
com_f4_speed_lpf_hz_b	Speed LPF cut-off frequency [Hz] (motor 2)
com_f4_ref_speed_rpm_b	Speed command value (mechanical angle) [rpm] (motor 2)
com_f4_speed_rate_limit_rpm_b	Maximum increment/decrement width for the speed command [rpm/s] (used when speed control is enabled) (motor 2)
com_f4_overspeed_limit_rpm_b	Speed limit value (mechanical angle) [rpm] (motor 2)
com_u1_flag_volt_err_comp_use_b	Voltage error compensation setting (motor 2) 0: Disable 1: Enable
com_u1_flag_fluxwkn_use_b	Magnetic flux weakening control setting (motor 2) 0: Disable 1: Enable
com_f4_e_obs_omega_hz_b	Natural frequency for the inductive voltage estimation system [Hz] (motor 2)
com_f4_e_obs_zeta_b	Attenuation coefficient for the inductive voltage estimation system (motor 2)
com_f4_pll_est_omega_hz_b	Natural frequency for the position estimation system [Hz] (motor 2)
com_f4_pll_est_zeta_b	Attenuation coefficient for the position estimation system (motor 2)
com_u1_flag_less_switch_use_b	Switching-to-sensorless processing setting (motor 2) 0: Disable 1: Enable
com_f4_switch_phase_err_deg_b	Phase error that allows switching-to-sensorless control (electrical angle) [deg] (motor 2)
com_f4_opl2less_sw_time_b	Processing time for switching-to-sensorless processing [s] (motor 2)
com_f4_phase_err_lpf_cut_freq_b	Phase error LPF cut-off frequency [Hz] (motor 2)
com_u1_flag_openloop_damping_use_b	Open-loop damping control setting (motor 2) 0: Disable 1: Enable
com_f4_ed_hpf_omega_b	d-axis inductive voltage HPF cut-off frequency [Hz] (motor 2)
com_f4_ol_damping_zeta_b	Attenuation coefficient for open-loop damping control (motor 2)
com_f4_ol_damping_fb_limit_rate_b	Feedback limit rate for open-loop damping control (motor 2)

Table 5-8 List of variables of the structure for RMW to update parameters

Structure	Variable	Description
st_rmw_param_buffer_t Structure for updating the RMW variables	u2_offset_calc_time	Current offset detection time set setting
	st_motor	Structure for motor parameters
	f4_max_speed_rpm	Maximum speed [rpm]
	u1_ctrl_loop_mode	Control loop mode (position control and speed control)
	f4_ol_ref_id	d-axis current command value in open loop mode [A]
	f4_id_up_time	Setting of the time required for Id increasement
	f4_id_down_time	Decreasing time of the d-axis current command value
	f4_id_down_speed_rpm	Speed (mechanical angle) when start decreasing the d-axis current command value [rpm]
	f4_id_up_speed_rpm	Speed (mechanical angle) when start increasing the d-axis current command value [rpm]
	f4_current_omega_hz	Natural frequency for the current control system [Hz]
	f4_current_zeta	Attenuation coefficient for the current control system
	f4_speed_omega_hz	Natural frequency for the speed control system [Hz]
	f4_speed_zeta	Attenuation coefficient for the speed control system
	f4_speed_lpf_hz	Speed LPF cut-off frequency [Hz]
	f4_ref_speed_rpm	Speed command value [rpm]
	f4_speed_rate_limit_rpm	Speed variation limit [rpm/s]
	f4_overspeed_limit_rpm	Speed limit value [rpm]
	u1_flag_volt_err_comp_use	Flag for whether to use voltage error compensation
	u1_flag_fluxwkn_use	Flag for whether to use magnetic flux weakening control
	f4_e_obs_omega_hz	Natural frequency for the inductive voltage estimation system [Hz]
	f4_e_obs_zeta	Attenuation coefficient for the inductive voltage estimation system
	f4_pll_est_omega_hz	Natural frequency for the position estimation system [Hz]
	f4_pll_est_zeta	Attenuation coefficient for the position estimation system
	u1_flag_less_switch_use	Flag for whether to use switching-to-sensorless processing
	f4_switch_phase_err_deg	Phase error that allows switching-to-sensorless control (electrical angle) [deg]
	f4_opl2less_sw_time	Processing time for switching-to-sensorless processing [s]
	f4_phase_err_lpf_cut_freq	Phase error LPF cut-off frequency [Hz]

Structure	Variable	Description
st_rmw_param_buffer_t Structure for updating the RMW variables	u1_flag_openloop_damping_use	Flag for whether to use open-loop damping control
	f4_ed_hpf_omega	d-axis inductive voltage HPF cut-off frequency [Hz]
	f4_ol_damping_zeta	Attenuation coefficient for open-loop damping control
	f4_ol_damping_fb_limit_rate	Feedback limit rate for open-loop damping control

5.1.6 Macro definition

Table 5-9 lists macros.

Table 5-9 List of macros

File name	Macro name	Defined value	Remarks
r_app_main.h	MAIN_UI_RMW	0	The RMW UI is used.
	MAIN_UI_BOARD	1	The Board UI is used.
	MAIN_UI_SIZE	2	The number of selectable UIs
r_app_board_ui.h	BOARD_SW1_ON	1	The switch SW1-1, SW1-2 is on.
	BOARD_SW1_OFF	0	The switch SW1-1, SW1-2 is off.
	BOARD_SW2_ON	0	The switch SW2-1, SW2-2 is on.
	BOARD_SW2_OFF	1	The switch SW2-1, SW2-2 is off.
	BOARD_CHATTERING_CNT	10	The chattering elimination counter value
	BOARD_AD12BIT_DATA	MOTOR_MCU_CFG_AD12BIT_DATA	12-bit AD value
	BOARD_VR1_SPEED_DEAD_BAND_A	80	Speed dead band for VR1-1 [rpm] (motor 1)
	BOARD_VR1_SPEED_MARGIN_A	50	Speed margin for VR1-1 [rpm] (motor 1)
	BOARD_VR1_SCALING_SPEED_A	(MOTOR_CFG_MAX_SPEED_RPM + BOARD_VR1_SPEED_MARGIN) / (BOARD_AD12BIT_DATA/2 + 1)	Speed scaling coefficient for VR1-1 (motor 1)
	BOARD_ADJUST_OFFSET_A	MOTOR_MCU_CFG_ADC_OFFSET	Offset value for VR1-1 (motor 1)
	BOARD_VR1_SPEED_DEAD_BAND_B	80	Speed dead band for VR1-2 [rpm] (motor 2)
	BOARD_VR1_SPEED_MARGIN_B	50	Speed margin for VR1-2 [rpm] (motor 2)
	BOARD_VR1_SCALING_SPEED_B	(MOTOR_CFG_MAX_SPEED_RPM + BOARD_VR1_SPEED_MARGIN) / (BOARD_AD12BIT_DATA/2 + 1)	Speed scaling coefficient for VR1-2 (motor 2)
	BOARD_ADJUST_OFFSET_B	MOTOR_MCU_CFG_ADC_OFFSET	Offset value for VR1-2 (motor 2)
r_app_rmw.h	ICS_DECIMATION	3	RMW watchpoint skip count
	ICS_INT_LEVEL	6	RMW interrupt priority
	ICS_BRR	251	Communication baud rate for RMW
	ICS_INT_MODE	1	Communication mode selection for RMW
	ICS_SCI_CH_SELECT	APP_CFG_SCI_CH_SELECT	SCI channel to be used

5.1.7 Adjustment and configuration of parameters

In the application layer, the configurations must be specified by using the r_app_control_cfg.h file. For the parameters to be set, see 5.1.4.

For the variables listed in Table 5-6, perform adjustment and configuration from RMW. For details about how to use RMW, see 3 Quick start guide and the Renesas Motor Workbench User's Guide (R21UZ0004).

5.2 Manager module

The manager module uses specific control modules to control the motor. Its processing includes system-wide management and protection for the interface with each module and for motor control.

5.2.1 Functionality

Table 5-10 lists the functions of the manager module.

Table 5-10 List of manager module functions

Function	Description
Mode management	Switches the operation mode of the system in response to the user command to control the motor.
Protection function	Handles errors by using the system protection function.
Control method management	Acquires and sets the states of speed control and current control.
Speed information acquisition	Acquires the speed information.
Control module command value setting	Selects the command values to be entered to the current control module and speed control module based on the control states.
Interrupt processing	Assigns processing to appropriate modules in response to interrupts set in Smart Configurator.

5.2.2 Module configuration diagram

Figure5-6 shows the module configuration.

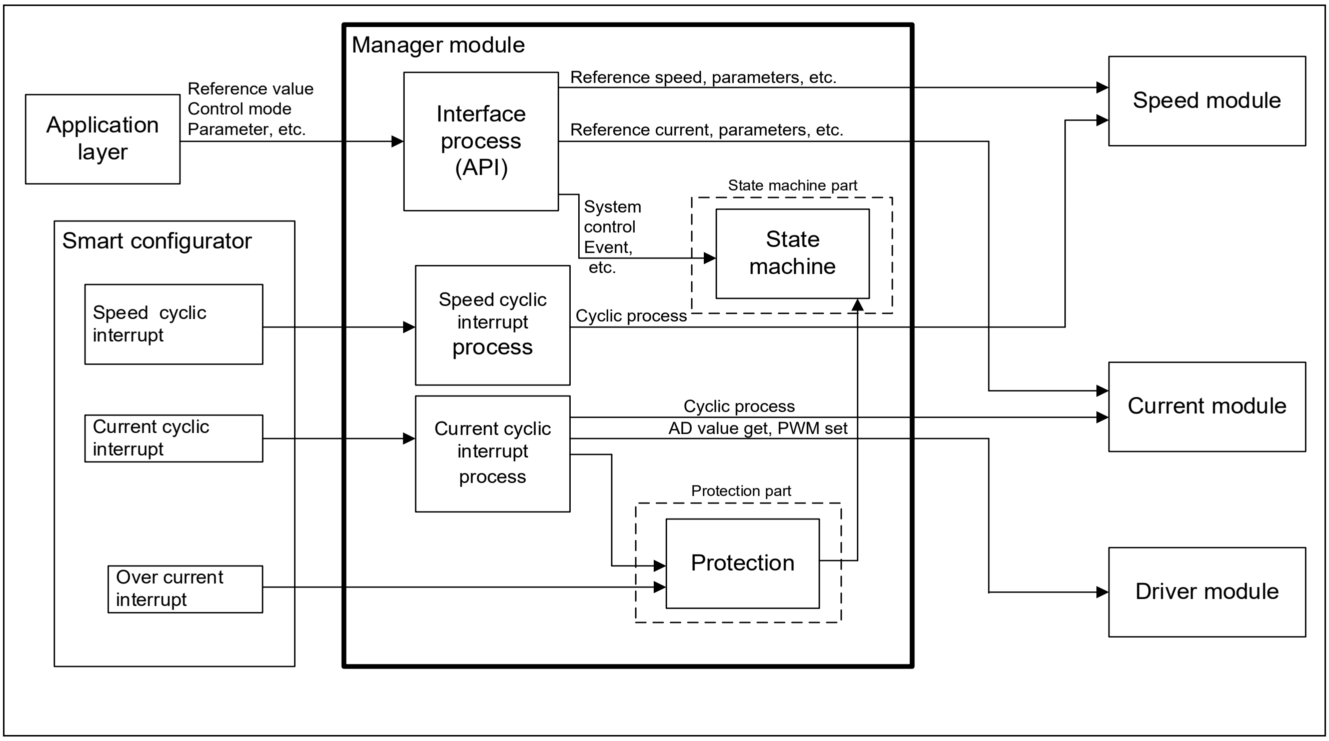
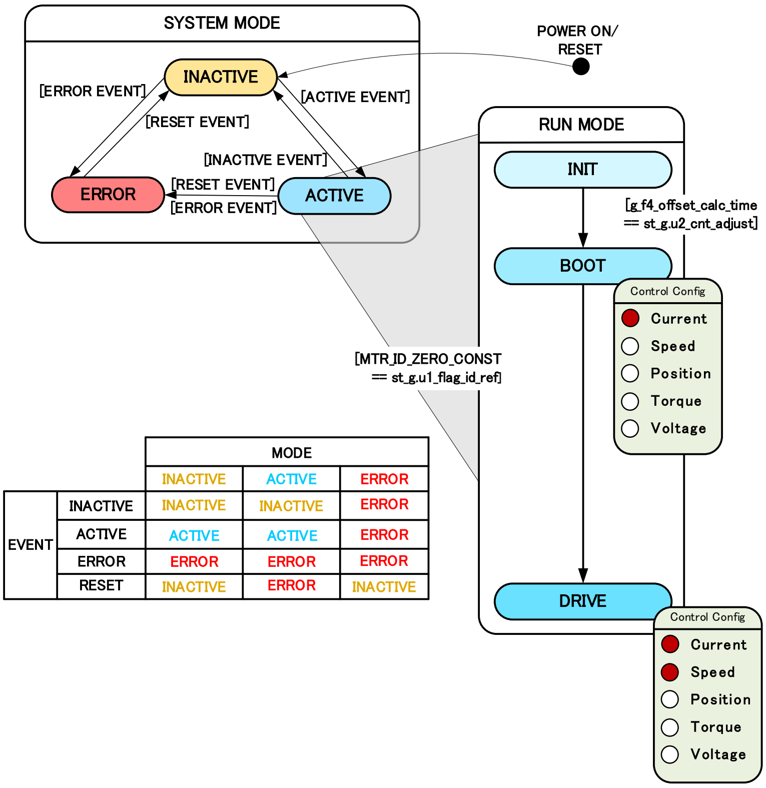


Figure5-6 Manager module configuration diagram

5.2.3 Mode management

Figure5-7 shows the state transitions of the target software of this application note. For the target software of this application note, the states are managed by using two types of modes: system modes and run modes. Control Config indicates the control systems that are currently active in the software.



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

Figure5-7 State transition diagram for the sensorless vector control software

5.2.3.1 SYSTEM MODE

These modes are used to indicate the system operation state. The state transitions as the event corresponding to a new state occurs. There are three system modes: INACTIVE (the motor is stopped), ACTIVE (the motor is running), and ERROR (an error has occurred).

5.2.3.2 RUN MODE

These modes are used to indicate the motor control state. When the system enters ACTIVE mode, the motor state transitions are shown in Figure5-7.

5.2.3.3 EVENT

The matrix table in Figure5-7 shows the system operation state transitions according to the event that occurs in each system mode. The following table shows the trigger that causes each event to occur.

Table 5-11 List of events and their triggers

Event name	Trigger
INACTIVE	Operation performed by the user
ACTIVE	Operation performed by the user
ERROR	Error detection by the system
RESET	Operation performed by the user

5.2.4 Protection function

This control program provides the following error states and implements an emergency stop function in each error state. For details about the values that can be specified for the settings of the system protection function, see Table 5-12.

- Overcurrent error

  Overcurrent errors can be detected on the hardware and in the software.

  A high-impedance output is provided to the PWM output pin in response to an emergency stop signal (overcurrent detection) from the hardware.

  This function monitors U-, V-, and W-phases at the overcurrent monitoring interval. When this function detects an overcurrent (the status in which the current is above the overcurrent limit value), it brings the program to an emergency stop (software detection). If this error occurs, the motor on the side where the error occurred will stop. The overcurrent limit value is automatically calculated from the rated current of the motor (MOTOR_CFG_NOMINAL_CURRENT_RMS).

- Overvoltage error

  This function monitors the inverter bus voltage at the overvoltage monitoring interval. When the function detects an overvoltage (that is, a voltage above the overvoltage limit value), it brings the program to an emergency stop. If this error occurs, the motor on the side where the error occurred will stop. The overvoltage limit value is preset in consideration of conditions such as an error in the resistor value of the detection circuit.

- Low-voltage error

  This function monitors the inverter bus voltage at the low-voltage monitoring interval. When the function detects a low voltage (that is, a voltage below the low-voltage limit value), it brings the program to an emergency stop. If this error occurs, the motor on the side where the error occurred will stop. The low-voltage limit value is preset in consideration of conditions such as an error in the resistor value of the detection circuit.

- Rotation speed error

  This function monitors the speed at the rotation speed monitoring interval. When the rotation speed exceeds the speed limit value, it brings the program to an emergency stop. If this error occurs, the motor on the side where the error occurred will stop.

Table 5-12 Values specified for the system protection function settings

Overcurrent error	Overcurrent limit value [A]	3.54
	Monitoring interval [μs]	Current control interval *1
Overvoltage error	Overvoltage limit value [V]	60
	Monitoring interval [μs]	Current control interval *1
Low-voltage error	Low-voltage limit value [V]	8
	Monitoring interval [μs]	Current control interval *1
Rotation speed error	Speed limit value [rpm]	4500
	Monitoring interval [μs]	Current control interval *1

Note: 1. See Table 4-1 Basic specifications of sensorless vector control software.

5.2.5 Flowcharts

The manager module performs processing in response to the occurrences of interrupts that are set in Smart Configurator by using several module API functions to control the motor. The following subsections show the flowcharts of the processing for these interrupts.

5.2.5.1 Interrupt processing for current control

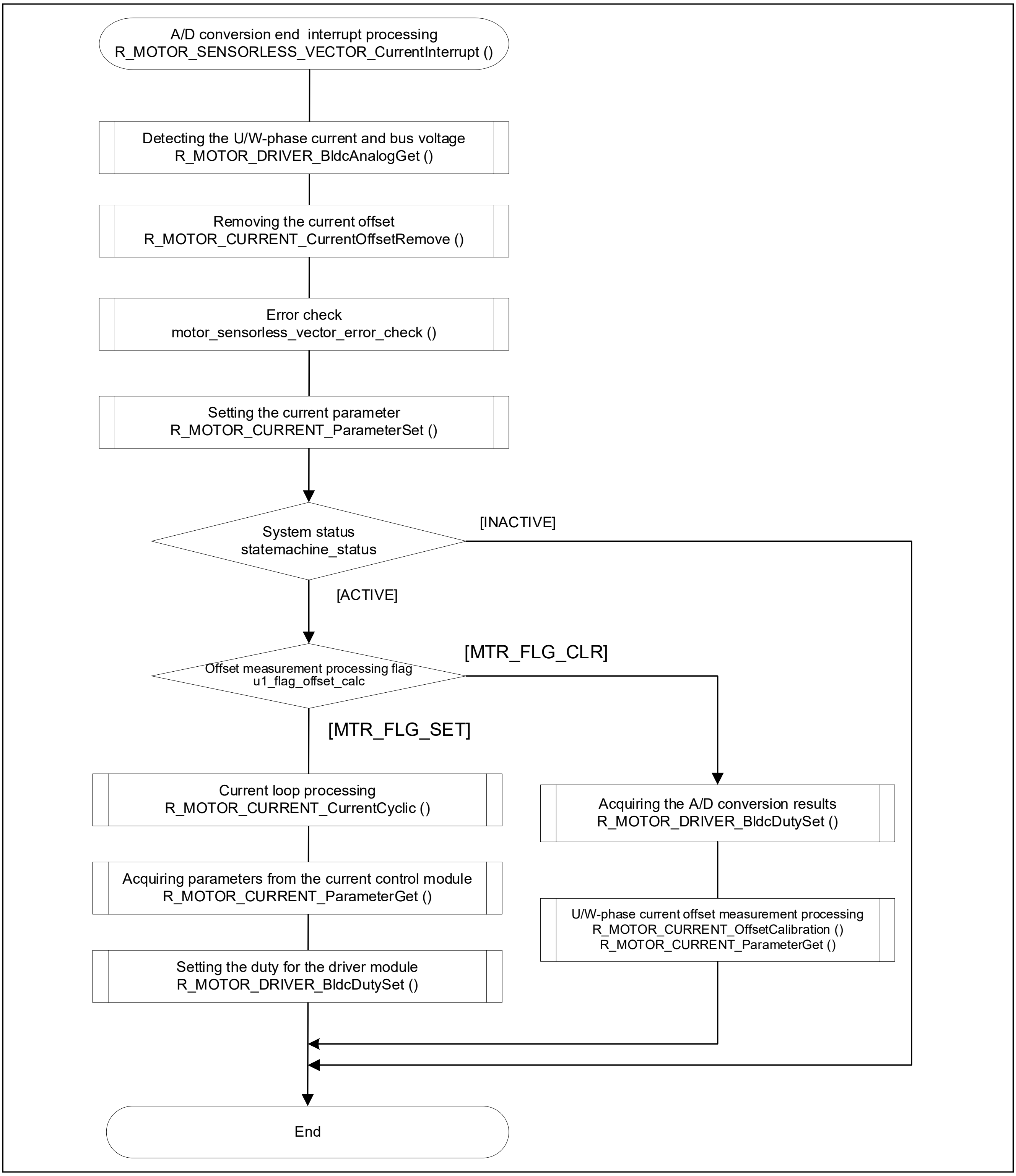

A/D conversion end interrupt processing
R_MOTOR_SENSORLESS_VECTOR_CurrentInterrupt ()
Detecting the U/W-phase current and bus voltage
R_MOTOR_DRIVER_BldcAnalogGet ()
Removing the current offset
R_MOTOR_CURRENT_CurrentOffsetRemove ()
Error check
motor_sensorless_vector_error_check ()
Setting the current parameter
R_MOTOR_CURRENT_ParameterSet ()
System status
statemachine_status
[INACTIVE]
[ACTIVE]
Offset measurement processing flag
u1_flag_offset_calc
[MTR_FLG_CLR]
[MTR_FLG_SET]
Current loop processing
R_MOTOR_CURRENT_CurrentCyclic ()
Acquiring parameters from the current control module
R_MOTOR_CURRENT_ParameterGet ()
Setting the duty for the driver module
R_MOTOR_DRIVER_BldcDutySet ()
Acquiring the A/D conversion results
R_MOTOR_DRIVER_BldcDutySet ()
U/W-phase current offset measurement processing
R_MOTOR_CURRENT_OffsetCalibration ()
R_MOTOR_CURRENT_ParameterGet ()
End

Figure5-8 Interrupt processing flowchart for current control

5.2.5.2 Interrupt processing for speed control

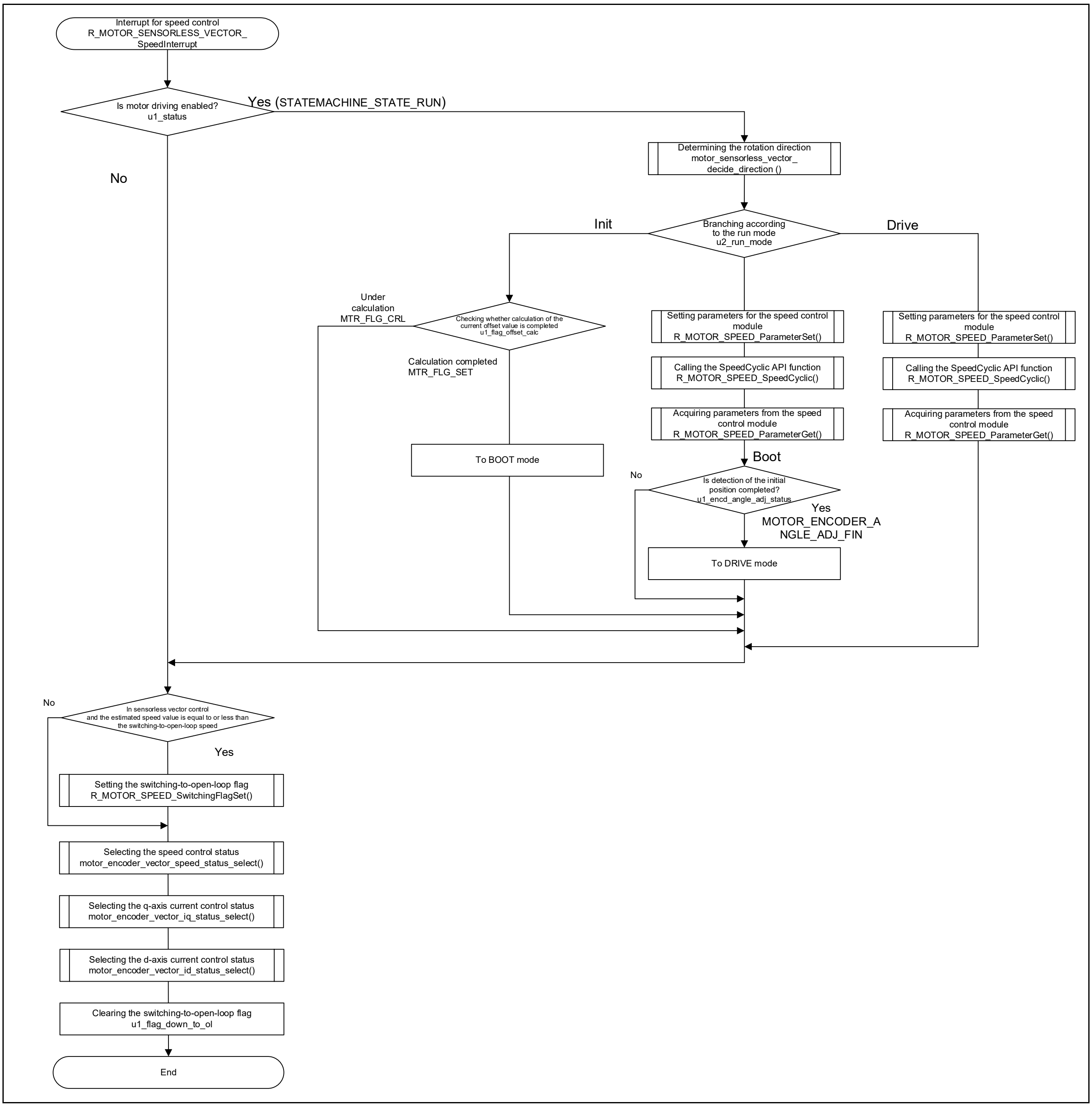

Interrupt for speed control
R_MOTOR_SENSORLESS_VECTOR_
SpeedInterrupt
Is motor driving enabled?
u1_status
Yes (STATEMACHINE_STATE_RUN)
No
Determining the rotation direction
motor_sensorless_vector_
decide_direction ()
Branching according
to the run mode
u2_run_mode
Init
Drive
Under
calculation
MTR_FLG_CRL
Checking whether calculation of the
current offset value is completed
u1_flag_offset_calc
Calculation completed
MTR_FLG_SET
To BOOT mode
Setting parameters for the speed control
module
R_MOTOR_SPEED_ParameterSet()
Calling the SpeedCyclic API function
R_MOTOR_SPEED_SpeedCyclic()
Acquiring parameters from the speed
control module
R_MOTOR_SPEED_ParameterGet()
Boot
No
Is detection of the initial
position completed?
u1_encd_angle_adj_status
Yes
MOTOR_ENCODER_A
NGLE_ADJ_FIN
To DRIVE mode
Setting parameters for the speed control
module
R_MOTOR_SPEED_ParameterSet()
Calling the SpeedCyclic API function
R_MOTOR_SPEED_SpeedCyclic()
Acquiring parameters from the speed
control module
R_MOTOR_SPEED_ParameterGet()
No
In sensorless vector control
and the estimated speed value is equal to or less than
the switching-to-open-loop speed
Yes
Setting the switching-to-open-loop flag
R_MOTOR_SPEED_SwitchingFlagSet()
Selecting the speed control status
motor_encoder_vector_speed_status_select()
Selecting the q-axis current control status
motor_encoder_vector_iq_status_select()
Selecting the d-axis current control status
motor_encoder_vector_id_status_select()
Clearing the switching-to-open-loop flag
u1_flag_down_to_ol
End

Figure5-9 Interrupt processing flowchart for speed control

5.2.5.3 Overcurrent detection interrupt processing

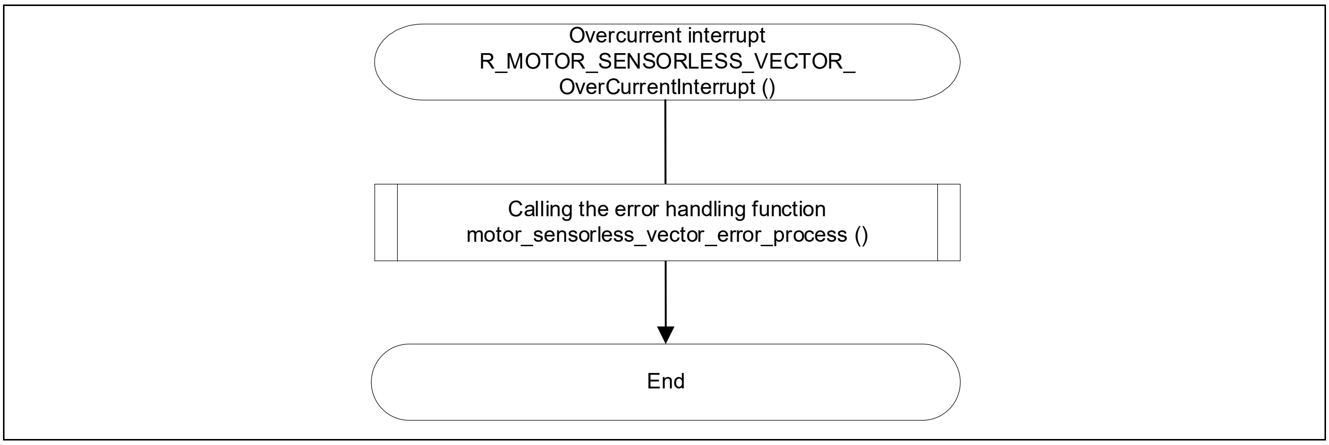

Overcurrent interrupt
R_MOTOR_SENSORLESS_VECTOR_
OverCurrentInterrupt ()
Calling the error handling function
motor_sensorless_vector_error_process ()
End

Figure5-10 Overcurrent detection interrupt processing flowchart

5.2.6 API

Table 5-13 lists the manager module API functions.

Table 5-13 List of API functions

API	Description
R_MOTOR_SENSORLESS_VECTOR_Open	Generates instances of this module and the modules to be used.
R_MOTOR_SENSORLESS_VECTOR_Close	Places this module in a reset state.
R_MOTOR_SENSORLESS_VECTOR_Reset	Initializes modules.
R_MOTOR_SENSORLESS_VECTOR_ParameterUpdate	Updates the control parameter settings of this module. This function also updates the control parameters for the related modules.
R_MOTOR_SENSORLESS_VECTOR_MotorStart	Places the motor in the running state.
R_MOTOR_SENSORLESS_VECTOR_MotorStop	Places the motor in the stop state.
R_MOTOR_SENSORLESS_VECTOR_MotorReset	Releases the system from the error state.
R_MOTOR_SENSORLESS_VECTOR_ErrorSet	Places the system in an error state.
R_MOTOR_SENSORLESS_VECTOR_SpeedSet	Sets the speed command value. This function is enabled when speed control is being performed.
R_MOTOR_SENSORLESS_VECTOR_SpeedGet	Acquires the speed information.
R_MOTOR_SENSORLESS_VECTOR_StatusGet	Acquires the status from the state machine.
R_MOTOR_SENSORLESS_VECTOR_ErrorStatusGet	Acquires the error state.
R_MOTOR_SENSORLESS_VECTOR_CtrlTypeSet	Sets the control method. To change the control method, place the motor in the stopped state. 0: Position control (Not use) 1: Speed control
R_MOTOR_SENSORLESS_VECTOR_LoopModeStatusGet	Acquires the control method. 0: Position control (Not use) 1: Speed control
R_MOTOR_SENSORLESS_VECTOR_SpeedInterrupt	Performs interrupt processing for speed control.
R_MOTOR_SENSORLESS_VECTOR_CurrentInterrupt	Performs interrupt processing for current control.
R_MOTOR_SENSORLESS_VECTOR_OverCurrentInterrupt	Performs interrupt processing when an overcurrent occurs.

Note: Since the manager module API needs to control each motor (motor 1 and motor 2), specify the structure variable pointer for the manager module (for motor 1: &g_st_sensorless_vector_a / for motor 2: &g_st_sensorless_vector_b) by arguments.

5.2.7 Configurations

Table 5-14 lists the configurations for the manager module.

Table 5-14 List of configurations

File name	Macro name	Description
r_motor_module_ cfg.h	MOTOR_MCU_CFG_PWM_TIMER_FREQ	PWM timer frequency [MHz]
	MOTOR_MCU_CFG_CARRIER_FREQ	Carrier wave frequency [kHz]
	MOTOR_MCU_CFG_INTR_DECIMATION	Skipping count for carrier wave interrupts
	MOTOR_MCU_CFG_AD_FREQ	ADC operating frequency [MHz]
	MOTOR_MCU_CFG_AD_SAMPLING_CYCLE	ADC sampling state [cycles]
	MOTOR_MCU_CFG_AD12BIT_DATA	ADC resolution
	MOTOR_MCU_CFG_ADC_OFFSET	ADC intermediate data
	MOTOR_TYPE_BLDC	Motor used (BLDC)
	MOTOR_COMMON_CFG_LOOP_MODE	Sets the default run mode.
	MOTOR_COMMON_CFG_OVERCURRENT_ MARGIN_MULT	Overcurrent limit value [A]
	MOTOR_COMMON_CFG_IA_MAX_CALC_M ULT	Coefficient for calculating the overcurrent limit value. BLDC: √3 STM: √2
	MOTOR_MCU_CFG_TFU_OPTIMIZE	Sets the processing of TFU-specific functions. MTR_ENABLE MTR_DISABLE
	SENSORLESS_VECTOR_ID_DOWN_SPEED _RPM	Speed to start decreasing id [rpm]
	SENSORLESS_VECTOR_ID_UP_SPEED_R PM	Speed to start increasing id [rpm]
	SENSORLESS_VECTOR_OPL2LESS_SWIT CH_PHASE_ERR_DEG	Phase error[deg] to decide sensor-less switch timing
	SENSORLESS_VECTOR_OPL2LESS_SWIT CH_PHASE_ERR_RAD	Phase error[rad] to decide sensor-less switch timing
	SENSORLESS_VECTOR_OPL2LESS_SWIT CH_PHASE_ERR_LPF_CUT_FREQ	Cut off frequency [Hz] of phase error LPF

Table 5-15 List of initial values for configurations

Macro name	2 shunt detection
MOTOR_MCU_CFG_PWM_TIMER_FREQ	CG_CONFIG_MOTOR_PWM_TIMER_FREQ
MOTOR_MCU_CFG_CARRIER_FREQ	CG_CONFIG_MOTOR_CARRIER_FREQ
MOTOR_MCU_CFG_INTR_DECIMATION	CG_CONFIG_MOTOR_INTR_DECIMATION
MOTOR_MCU_CFG_AD_FREQ	CG_MOTOR_MCU_CFG_AD_FREQ
MOTOR_MCU_CFG_AD_SAMPLING_CYCLE	45
MOTOR_MCU_CFG_AD12BIT_DATA	CG_MOTOR_CFG_MAX_AD_DATA
MOTOR_MCU_CFG_ADC_OFFSET	0x7FF
MOTOR_TYPE_BLDC	Defined
MOTOR_COMMON_CFG_LOOP_MODE	MOTOR_LOOP_SPEED
MOTOR_COMMON_CFG_OVERCURRENT_MARGIN_MULT	1.5
MOTOR_COMMON_CFG_IA_MAX_CALC_MULT	MTR_SQRT_3
MOTOR_MCU_CFG_TFU_OPTIMIZE	MTR_ENABLE

5.2.8 Structure and variable information

Table 5-16 lists the structures and variables for the manager module. For the manager module, the structure for the manager module (g_st_sensorless_vector) is defined by securing an instance of the module from the API.

Table 5-16 List of structures and variables

Structure	Variable	Description
st_sensorless_vector_control_t Structure for the manager module	u1_flag_less_switch_use	ON/OFF for switching-to-sensorless processing
	u1_flag_openloop_damping_use	ON/OFF for open-loop damping control
	u1_flag_down_to_ol	Switching-to-open-loop flag
	u1_state_id_ref	Status of the d-axis current command value
	u1_state_iq_ref	Status of the q-axis current command value
	u1_state_speed_ref	Status of the speed command value
	u1_direction	Rotation direction
	u1_ctrl_loop_mode	Control mode selection (Only speed control)
	u2_error_status	Error status
	u2_run_mode	Run mode
	f4_vdc_ad	Bus voltage [V]
	f4_iu_ad	U-phase current [A]
	f4_iv_ad	V-phase current [A]
	f4_iw_ad	W-phase current [A]
	f4_overcurrent_limit	Overcurrent limit value [A]
	f4_overvoltage_limit	Overvoltage limit value [V]
	f4_undervoltage_limit	Low-voltage limit value [V]
	f4_overspeed_limit_rad	Overspeed limit value [rad/s]
	f4_phase_err_rad_lpf	LPF value of phase error[rad]
	f4_switch_phase_err_rad	Phase error to decide sensor-less switch timing [rad]
	f4_id_down_speed_rad	The speed threshold[rad/s] to ramp down the d-axis current
	f4_id_up_speed_rad	The speed threshold[rad/s] to ramp up the d-axis current
	f4_damp_comp_speed	Feedback value for reference speed [rad/s]
	f4_ol_speed_rad	Speed value in open loop mode [rad/s]
	st_phase_err_lpf	Phase error LPF structure
	st_current_output	Structure for current control module output
	st_speed_output	Structure for speed control module output
	st_stm	Structure for the state machine

Structure	Variable	Description
st_sensorless_vector_control_t Structure for the manager module	st_motor	Motor parameter structure
	*p_st_driver	The driver module generation instance pointer
	*p_st_cc	The current control module generation instance pointer
	*p_st_sc	The speed control module generation instance pointer
	*p_st_sv_init	Instance for the initial value of the st_sensorless_vector_control_t
	*p_st_cfg_init	Instance for the initial value of the st_sensorless_vector_cfg_t
st_sensorless_vector_cfg_t Structure for setting the manager module control parameters	u1_flag_less_switch_use	ON/OFF for switching-to-sensorless processing
	u1_flag_openloop_damping_use	ON/OFF for open-loop damping control
	f4_overspeed_limit_rpm	Overspeed limit value [rpm]
	f4_switch_phase_err_deg	Phase error to decide sensor-less switch timing [deg]
	f4_phase_err_lpf_cut_freq	Cut off frequency of phase error LPF[Hz]
	f4_id_down_speed_rpm	The speed threshold [rpm] to ramp down the d-axis current
	f4_id_up_speed_rpm	The speed threshold [rpm] to ramp up the d-axis current
	f4_ctrl_period	Speed loop control interval [s]
	st_motor	Motor parameter structure
st_sv_cfg_t Structure for the initial value of the manager module control	st_sv_init_cfg	Initial values of the structure for setting the control parameters of the manager module

5.2.9 Macro definition

Table 5-17 lists the macros for the manager module.

Table 5-17 List of macros

File name	Macro name	Defined value	Remarks
r_motor_sensorless_vector_api.h	MOTOR_LOOP_POSITION	0	Position control mode (Not use)
	MOTOR_LOOP_SPEED	1	Speed control mode
	MOTOR_SENSORLESS_VECTOR_ERROR_NONE	(0x0000)	An error status. There is no error.
	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_CURRENT_HW	(0x0001)	An error status. A hardware overcurrent error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_VOLTAGE	(0x0002)	An error status. An overvoltage error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_SPEED	(0x0004)	An error status. An overspeed error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_HALL_PATTERN	(0x0020)	An error status. A Hall pattern error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_LOW_VOLTAGE	(0x0080)	An error status. A low-voltage error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_CURRENT_SW	(0x0100)	An error status. A software overcurrent error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_UNKNOWN	(0xffff)	An error status. An error whose error code is unknown has occurred.
r_motor_sensorless_vector_manager.h	MOTOR_MODE_INIT	(0x00)	INIT run mode
	MOTOR_MODE_BOOT	(0x01)	BOOT run mode
	MOTOR_MODE_DRIVE	(0x02)	DRIVE run mode
r_motor_sensorless_vector_api.h	MOTOR_CTRL_TYPE_POS	0	Macro for switching the control method. Position control mode. (Not use)
	MOTOR_CTRL_TYPE_SPEED	1	Macro for switching the control method. Speed control mode.
r_motor_common.h	MTR_ID_A	(0)	ID (motor 1)
	MTR_ID_B	(1)	ID (motor 2)

5.2.10 Manager Module Variable Information

Table 5-18 Manager Module Variable Information list

Structure	Variable	Description
st_sensorless_vector_control_t Structure for the Manager Module	g_st_sensorless_vector_a	Variable for motor1
	g_st_sensorless_vector_a_init	Variable for initial value of motor1
	g_st_sensorless_vector_b	Variable for motor2
	g_st_sensorless_vector_b_init	Variable for initial value of motor2
st_sensorless_vector_cfg_t Structure for setting control parameters in the manager module	st_sensorless_vector_cfg	(Local variables for function internals)
st_sv_cfg_t Structure for setting the initial value of the manager module control parameters	g_st_sensorless_vector_a_cfg	Variable for motor1
	g_st_sensorless_vector_b_cfg	Variable for motor2

5.2.11 Adjustment and configuration of parameters

When you use the sample program, you need to correctly set the inverter information and the information about the motor to be used. Table 5-19, Table 5-20, and Table 5-21 shows the values set in the sample program.

Table 5-19 Motor and inverter parameter settings (common)

File name	Macro name	Set value	Description
r_motor_inverter_cfg.h	INVERTER_CFG_SHUNT_RESIST	0.010f	Shunt resistance value [ohm]
	INVERTER_CFG_VOLTAGE_GAIN	22.2766f	Coefficient for voltage detection
	INVERTER_CFG_CURRENT_AMP_GAIN	20.0f	Gain of the amplifier for current detection
	INVERTER_CFG_CURRENT_LIMIT	21.4f	Overcurrent limit value for the inverter board [A]
	INVERTER_CFG_INPUT_V	24.0f	Input voltage [V]
	INVERTER_CFG_ADC_REF_VOLTAGE	5.0f	Analog power supply voltage for the MCU [V]

Table 5-20 Motor1 and inverter1 parameter settings

File name	Macro name	Set value	Description
r_motor_inverter_cfg.h	INVERTER_CFG_DEADTIME_A	CG_CONFIG_MOTOR_A_DEADTIME	Dead time [μs] (motor 1)
	INVERTER_CFG_OVERVOLTAGE_LIMIT_A	60.0f	Overvoltage limit [V] (motor 1)
	INVERTER_CFG_UNDERVOLTAGE_LIMIT_A	8.0f	Low-voltage limit [V] (motor 1)
	MOTOR_CFG_POLE_PAIRS_A	4	Number of pole pairs (motor 1)
	MOTOR_CFG_MAGNETIC_FLUX_A	0.01119f	Magnetic flux [wb] (motor 1)
	MOTOR_CFG_RESISTANCE_A	1.3f	Resistance [ohm] (motor 1)
	MOTOR_CFG_D_INDUCTANCE_A	0.0013f	d-axis inductance [H] (motor 1)
	MOTOR_CFG_Q_INDUCTANCE_A	0.0013f	q-axis inductance [H] (motor 1)
	MOTOR_CFG_ROTOR_INERTIA_A	0.000003666f	Rotor inertia [kg m^2] (motor 1)
	MOTOR_CFG_NOMINAL_CURRENT_RMS_A	1.67f	Rated current [A] (motor 1)
	MOTOR_CFG_MAX_SPEED_RPM_A	2400.0f	Maximum speed [rpm] (motor 1)
	INVERTER_CFG_COMP_V0_A	0.564f	Coefficient for compensation of the voltage error [V] *1 (motor 1)
	INVERTER_CFG_COMP_V1_A	0.782f	Coefficient for compensation of the voltage error [V] *1 (motor 1)
	INVERTER_CFG_COMP_V2_A	0.937f	Coefficient for compensation of the voltage error [V] *1 (motor 1)
	INVERTER_CFG_COMP_V3_A	1.027f	Coefficient for compensation of the voltage error [V] *1 (motor 1)
	INVERTER_CFG_COMP_V4_A	1.058f	Coefficient for compensation of the voltage error [V] *1 (motor 1)
	INVERTER_CFG_COMP_I0_A	0.022f	Coefficient for compensation of the voltage error [A] *1 (motor 1)
	INVERTER_CFG_COMP_I1_A	0.038f	Coefficient for compensation of the voltage error [A] *1 (motor 1)
	INVERTER_CFG_COMP_I2_A	0.088f	Coefficient for compensation of the voltage error [A] *1 (motor 1)
	INVERTER_CFG_COMP_I3_A	0.248f	Coefficient for compensation of the voltage error [A] *1 (motor 1)
	INVERTER_CFG_COMP_I4_A	0.865f	Coefficient for compensation of the voltage error [A] *1 (motor 1)

Note: 1. For details, see the chapter 5.5 on compensation of the voltage error.

Table 5-21 Motor2 and inverter2 parameter settings

File name	Macro name	Set value	Description
r_motor_inverter_cfg.h	INVERTER_CFG_DEADTIME_B	CG_CONFIG_MOTOR_B_DEADTIME	Dead time [μs] (motor 2)
	INVERTER_CFG_OVERVOLTAGE_LIMIT_B	60.0f	Overvoltage limit [V] (motor 2)
	INVERTER_CFG_UNDERVOLTAGE_LIMIT_B	8.0f	Low-voltage limit [V] (motor 2)
	MOTOR_CFG_POLE_PAIRS_B	4	Number of pole pairs (motor 2)
	MOTOR_CFG_MAGNETIC_FLUX_B	0.01119f	Magnetic flux [wb] (motor 2)
	MOTOR_CFG_RESISTANCE_B	1.3f	Resistance [ohm] (motor 2)
	MOTOR_CFG_D_INDUCTANCE_B	0.0013f	d-axis inductance [H] (motor 2)
	MOTOR_CFG_Q_INDUCTANCE_B	0.0013f	q-axis inductance [H] (motor 2)
	MOTOR_CFG_ROTOR_INERTIA_B	0.000003666f	Rotor inertia [kg m^2] (motor 2)
	MOTOR_CFG_NOMINAL_CURRENT_RMS_B	1.67f	Rated current [A] (motor 2)
	MOTOR_CFG_MAX_SPEED_RPM_B	2400.0f	Maximum speed [rpm] (motor 2)
	INVERTER_CFG_COMP_V0_B	0.564f	Coefficient for compensation of the voltage error [V] *1 (motor 2)
	INVERTER_CFG_COMP_V1_B	0.782f	Coefficient for compensation of the voltage error [V] *1 (motor 2)
	INVERTER_CFG_COMP_V2_B	0.937f	Coefficient for compensation of the voltage error [V] *1 (motor 2)
	INVERTER_CFG_COMP_V3_B	1.027f	Coefficient for compensation of the voltage error [V] *1 (motor 2)
	INVERTER_CFG_COMP_V4_B	1.058f	Coefficient for compensation of the voltage error [V] *1 (motor 2)
	INVERTER_CFG_COMP_I0_B	0.022f	Coefficient for compensation of the voltage error [A] *1 (motor 2)
	INVERTER_CFG_COMP_I1_B	0.038f	Coefficient for compensation of the voltage error [A] *1 (motor 2)
	INVERTER_CFG_COMP_I2_B	0.088f	Coefficient for compensation of the voltage error [A] *1 (motor 2)
	INVERTER_CFG_COMP_I3_B	0.248f	Coefficient for compensation of the voltage error [A] *1 (motor 2)
	INVERTER_CFG_COMP_I4_B	0.865f	Coefficient for compensation of the voltage error [A] *1 (motor 2)

Note: 1. For details, see the chapter 5.5 on compensation of the voltage error.

5.2.12 How to start the motor

The manager module controls the motor by changing the flag settings that manage the command values for the d-axis current, q-axis current, and speed according to the run mode. Also, by changing these command values appropriately, the manager module creates a starting sequence to start the motor. Figure5-11 shows startup of the motor.

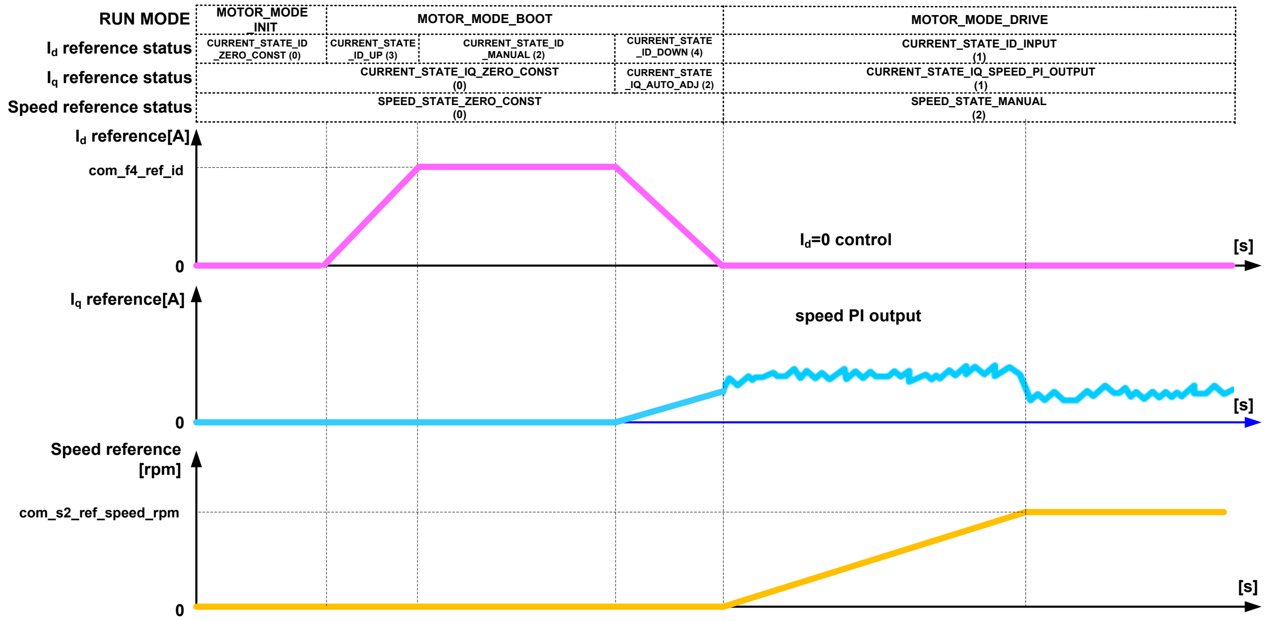


Figure5-11 Motor start control based on sensorless vector control

5.3 Current control module

The current control module uses the value of the incoming current to perform coordinate transformation and feedback control that are necessary for vector control, and then calculates the voltage of the PWM output. The module also controls submodules that perform modulation and voltage error compensation.

5.3.1 Functions

Table 5-22 lists the functions of the current control module.

Table 5-22 List of functions of the current control module

Function	Description
Current control	Performs calculation according to the current command value to set the PWM output value.
Current offset adjustment	Calculates the offset value of the current value detected by AD.
Voltage error compensation	Compensates for the effects of output voltage dead time.
Coordinate transformation and inverse transformation	Performs coordinate transformation for the current value detected to perform vector control. This function also performs inverse transformation of the coordinate for the calculation results to restore the original coordinate axis.
Modulation	Improves the efficiency by modulation to a PWM signal.
Non-interacting control	Calculates interference cancellation to prevent interference between the d and q axes.

5.3.2 Module configuration diagram

Figure5-12 shows the module configuration.

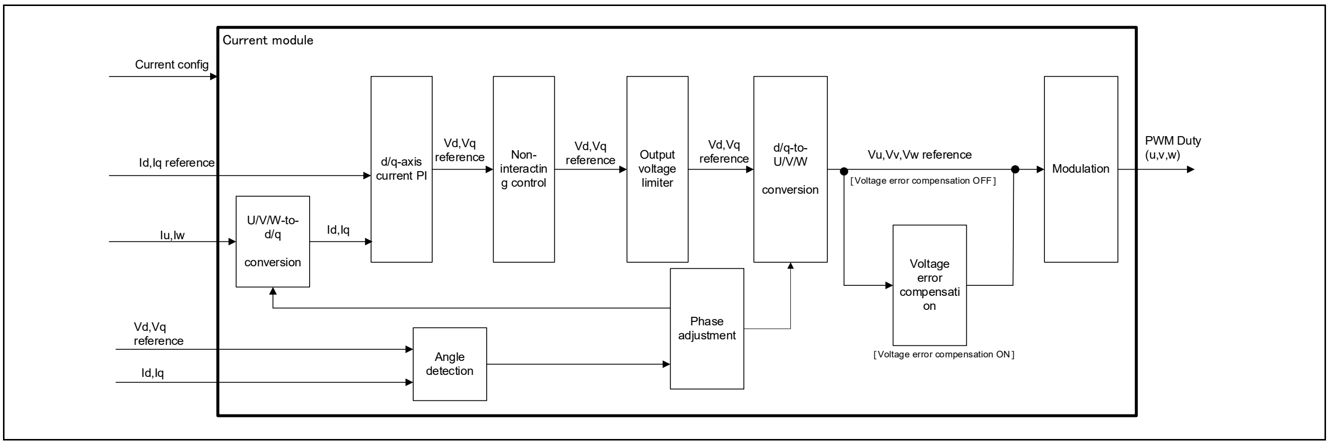


Figure5-12 Current control module configuration diagram

5.3.3 Flowcharts

Figure5-13 shows the flowchart for the loop processing of the current control module.

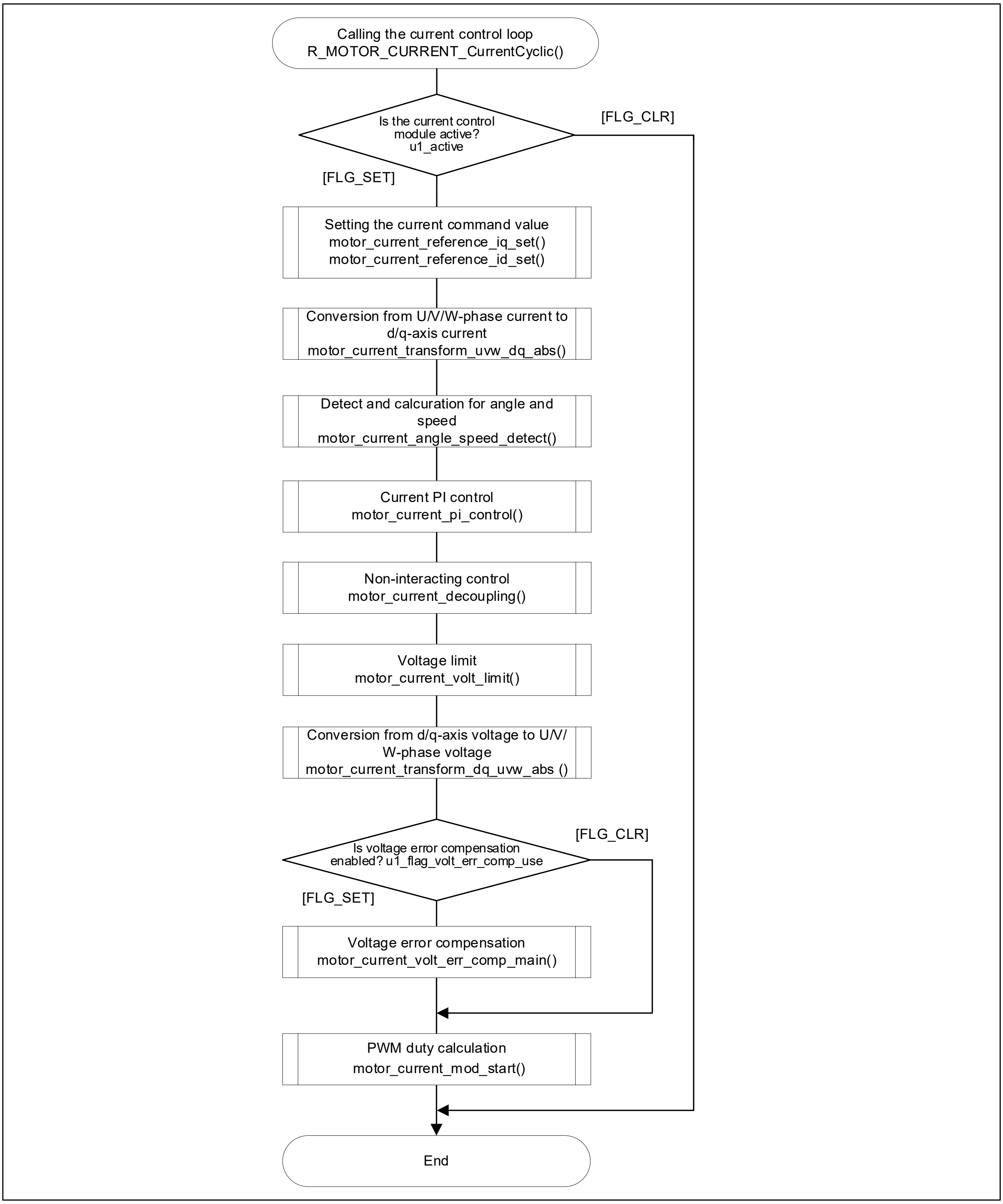

Calling the current control loop
R_MOTOR_CURRENT_CurrentCyclic()
Is the current control module active? u1_active
[FLG_CLR]
[FLG_SET]
Setting the current command value
motor_current_reference_iq_set()
motor_current_reference_id_set()
Conversion from U/V/W-phase current to d/q-axis current
motor_current_transform_uvw_dq_abs()
Detect and calcuration for angle and speed
motor_current_angle_speed_detect()
Current PI control
motor_current_pi_control()
Non-interacting control
motor_current_decoupling()
Voltage limit
motor_current_volt_limit()
Conversion from d/q-axis voltage to U/V/W-phase voltage
motor_current_transform_dq_uvw_abs ()
Is voltage error compensation enabled? u1_flag_volt_err_comp_use
[FLG_CLR]
[FLG_SET]
Voltage error compensation
motor_current_volt_err_comp_main()
PWM duty calculation
motor_current_mod_start()
End

Figure5-13 Flowchart for the loop processing of the current control module

5.3.4 API

Table 5-23 lists the API functions of the current control module.

Table 5-23 List of API functions

API	Description
R_MOTOR_CURRENT_Open	Generates an instance of the current control module.
R_MOTOR_CURRENT_Close	Places the current control module in a reset state.
R_MOTOR_CURRENT_Reset	Initializes the current control module.
R_MOTOR_CURRENT_Run	Activates the current control module.
R_MOTOR_CURRENT_ParameterSet	Inputs the variable information that is used for current control.
R_MOTOR_CURRENT_ParameterGet	Acquires the current control results that are output.
R_MOTOR_CURRENT_ParameterUpdate	Updates the control parameters of the current control module.
R_MOTOR_CURRENT_CurrentCyclic	Performs current control.
R_MOTOR_CURRENT_OffsetCalibration	Performs offset adjustment of current detection.
R_MOTOR_CURRENT_CurrentOffsetRemove	Returns the value with the current detection offset value excluded.
R_MOTOR_CURRENT_VoltErrCompParamSet	Sets the voltage error compensation parameter.
R_MOTOR_CURRENT_BEMFObserverParameterUpdate	Updates the control parameters for the inductive voltage observer.
R_MOTOR_CURRENT_HuntingSuppress	Performs a measure against estimation speed discontinuity when switching from open-loop control to sensorless vector control.

Note: Since the current control module API needs to control each motor (motor 1, motor 2), specify the structure variable pointer for the current control module (for motor 1: &g_st_cc_a / for motor 2: &g_st_cc_b) by argument.

5.3.5 Configurations

Table 5-24 shows the configurations that are used for the current control module. Set up the functions to be used and the necessary parameters. Table 5-25 shows the initial values.

Table 5-24 List of configurations

File name	Macro name	Description
r_motor_module_cfg.h	CURRENT_CFG_VOLT_ERR_COMP_A	Enables or disables the voltage error compensation function. (motor 1) Enable: MTR_ENABLE Disable: MTR_DISABLE
	CURRENT_CFG_MODULATION_METHOD	Modulation method MOD_METHOD_SPWM : Sinusoidal PWM MOD_METHOD_SVPWM : Spatial vector PWM
	CURRENT_CFG_OFFSET_CALC_TIME_A	Current offset measurement time setting (motor 1)
	CURRENT_CFG_PERIOD_MAG_VALUE_A	Coordinate transformation interval coefficient (motor 1)
	CURRENT_CFG_PI_INTEGRAL_LIMIT_VD_A	d-axis current limit [V] (motor 1) INVERTER_CFG_INPUT_V : The maximum input voltage is defined by using r_motor_inverter_cfg.h.
	CURRENT_CFG_PI_INTEGRAL_LIMIT_VQ_A	q-axis current limit [V] (motor 1)
	CURRENT_CFG_OMEGA_A	Natural frequency for the current control system [Hz] (motor 1)
	CURRENT_CFG_ZETA_A	Attenuation coefficient for the current control system (motor 1)
	CURRENT_CFG_REF_ID_OPENLOOP_A	d-axis current command value in open loop mode [A] (motor 1)
	CURRENT_CFG_ID_UP_STEP_TIME_A	Setting of additional time for the d-axis current command value (motor 1)
	CURRENT_CFG_ID_DOWN_STEP_TIME_A	Additional time for the d-axis current command value (motor 1)
	CURRENT_CFG_IQ_DOWN_STEP_TIME_INV_A	The inverse of the additional time for the d-axis current command value (motor 1)
	CURRENT_CFG_E_OBS_OMEGA_A	Natural frequency for the inductive voltage estimation system [Hz] (motor 1)
	CURRENT_CFG_E_OBS_ZETA_A	Attenuation coefficient for the inductive voltage estimation system (motor 1)

	CURRENT_CFG_PLL_EST_OMEGA_A	Natural frequency for the position estimation system [Hz] (motor 1)
	CURRENT_CFG_PLL_EST_ZETA_A	Attenuation coefficient for the position estimation system (motor 1)

File name	Macro name	Description
r_motor_module_cfg.h	CURRENT_CFG_VOLT_ERR_COMP_B	Enables or disables the voltage error compensation function. (motor 2) Enable: MTR_ENABLE Disable: MTR_DISABLE
	CURRENT_CFG_OFFSET_CALC_TIME_B	Current offset measurement time setting (motor 2)
	CURRENT_CFG_PERIOD_MAG_VALUE_B	Coordinate transformation interval coefficient (motor 2)
	CURRENT_CFG_PI_INTEGRAL_LIMIT_VD_B	d-axis current limit [V] (motor 2) INVERTER_CFG_INPUT_V : The maximum input voltage is defined by using r_motor_inverter_cfg.h.
	CURRENT_CFG_PI_INTEGRAL_LIMIT_VQ_B	q-axis current limit [V] (motor 2)
	CURRENT_CFG_OMEGA_B	Natural frequency for the current control system [Hz] (motor 2)
	CURRENT_CFG_ZETA_B	Attenuation coefficient for the current control system (motor 2)
	CURRENT_CFG_REF_ID_OPENLOOP_B	d-axis current command value in open loop mode [A] (motor 2)
	CURRENT_CFG_ID_UP_STEP_TIME_B	Setting of additional time for the d-axis current command value (motor 2)
	CURRENT_CFG_ID_DOWN_STEP_TIME_B	Additional time for the d-axis current command value (motor 2)
	CURRENT_CFG_IQ_DOWN_STEP_TIME_INV_B	The inverse of the additional time for the d-axis current command value (motor 2)
	CURRENT_CFG_E_OBS_OMEGA_B	Natural frequency for the inductive voltage estimation system [Hz] (motor 2)
	CURRENT_CFG_E_OBS_ZETA_B	Attenuation coefficient for the inductive voltage estimation system (motor 2)
	CURRENT_CFG_PLL_EST_OMEGA_B	Natural frequency for the position estimation system [Hz] (motor 2)
	CURRENT_CFG_PLL_EST_ZETA_B	Attenuation coefficient for the position estimation system (motor 2)

Table 5-25 List of initial values for configurations

Macro name	Settings
CURRENT_CFG_VOLT_ERR_COMP_A	MTR_ENABLE
CURRENT_CFG_MODULATION_METHOD	MOD_METHOD_SVPWM
CURRENT_CFG_OFFSET_CALC_TIME_A	512.0f
CURRENT_CFG_PERIOD_MAG_VALUE_A	1.0f
CURRENT_CFG_PI_INTEGRAL_LIMIT_VD_A	(INVERTER_CFG_INPUT_V * 0.5f)
CURRENT_CFG_PI_INTEGRAL_LIMIT_VQ_A	(INVERTER_CFG_INPUT_V * 0.5f)
CURRENT_CFG_OMEGA_A	300.0f
CURRENT_CFG_ZETA_A	1.0f
CURRENT_CFG_REF_ID_OPENLOOP_A	0.3f
CURRENT_CFG_ID_UP_STEP_TIME_A	2560.0f
CURRENT_CFG_ID_DOWN_STEP_TIME_A	500.0f
CURRENT_CFG_IQ_DOWN_STEP_TIME_INV_A	1.0f /CURRENT_CFG_ID_UP_STEP_TIME
CURRENT_CFG_E_OBS_OMEGA_A	1000.0f
CURRENT_CFG_E_OBS_ZETA_A	1.0f
CURRENT_CFG_PLL_EST_OMEGA_A	20.0f
CURRENT_CFG_PLL_EST_ZETA_A	1.0f
CURRENT_CFG_VOLT_ERR_COMP_B	MTR_ENABLE
CURRENT_CFG_OFFSET_CALC_TIME_B	512.0f
CURRENT_CFG_PERIOD_MAG_VALUE_B	1.0f
CURRENT_CFG_PI_INTEGRAL_LIMIT_VD_B	(INVERTER_CFG_INPUT_V * 0.5f)
CURRENT_CFG_PI_INTEGRAL_LIMIT_VQ_B	(INVERTER_CFG_INPUT_V * 0.5f)
CURRENT_CFG_OMEGA_B	300.0f
CURRENT_CFG_ZETA_B	1.0f
CURRENT_CFG_REF_ID_OPENLOOP_B	0.3f
CURRENT_CFG_ID_UP_STEP_TIME_B	2560.0f
CURRENT_CFG_ID_DOWN_STEP_TIME_B	500.0f
CURRENT_CFG_IQ_DOWN_STEP_TIME_INV_B	1.0f /CURRENT_CFG_ID_UP_STEP_TIME
CURRENT_CFG_E_OBS_OMEGA_B	1000.0f
CURRENT_CFG_E_OBS_ZETA_B	1.0f
CURRENT_CFG_PLL_EST_OMEGA_B	20.0f
CURRENT_CFG_PLL_EST_ZETA_B	1.0f

5.3.6 Structure and variable information

Table 5-26 lists the structures and variables that are used for the current control module. For the current control module, the structure for the current control module (g_st_cc) is defined by securing an instance of the module from the API.

Table 5-26 List of structures and variables

Structure	Variable	Description
st_current_control_t	u1_active	The active state of the current control module
Structure for the current control module	u1_flag_volt_err_comp_use	Enables or disables the voltage error compensation function.
	u1_state_id_ref	Status of the d axis at startup
	u1_state_iq_ref	Status of the q axis at startup
	u1_flag_offset_calc	Flag for the current offset calculation
	u2_offset_calc_time	Measurement time setting in current offset adjustment
	u2_crnt_offset_cnt	Measurement count in current offset adjustment
	f4_ctrl_period	Current control interval (period) [s]
	f4_refu	U-phase command voltage [V]
	f4_refv	V-phase command voltage [V]
	f4_refw	W-phase command voltage [V]
	f4_vd_ref	d-axis voltage command value [V]
	f4_vq_ref	q-axis voltage command value [V]
	f4_id_ref	d-axis current command value [A]
	f4_iq_ref	q-axis current command value [A]
	f4_id_ad	d-axis current value [A]
	f4_iq_ad	q-axis current value [A]
	f4_lim_iq	q-axis current limit value [A]
	f4_offset_iu	U-phase offset current value [A]
	f4_offset_iw	W-phase offset current value [A]
	f4_sum_iu_ad	U-phase total current value [A]
	f4_sum_iw_ad	W-phase total current value [A]
	f4_vdc_ad	Bus voltage value [V]
	f4_iu_ad	U-phase current value [A]
	f4_iv_ad	V-phase current value [A]
	f4_iw_ad	W-phase current value [A]
	f4_modu	U-phase duty cycle
	f4_modv	V-phase duty cycle

Structure	Variable	Description
st_current_control_t Structure for the current control module	f4_modw	W-phase duty cycle
	f4_speed_rad	Speed [rad/s]
	f4_ref_id_ctrl	d-axis current command value [A]
	f4_ref_iq_ctrl	q-axis current command value [A]
	f4_id_up_step	The amount of change in the d-axis current command value in d-axis current increase mode [A]
	f4_id_down_step	The amount of change in the d-axis current command value in d-axis current decrease mode [A]
	f4_iq_down_step	The amount of change in the q-axis current command value in q-axis current decrease mode [A]
	f4_iq_down_step_time_inv	The inverse of the change time in the q-axis current command value in q-axis current decrease mode
	f4_ol_ref_id	d-axis current command value in open loop mode [A]
	f4_va_max	Maximum voltage on the d and q axes [V]
	f4_ed	Estimated d-axis inductive voltage value
	f4_eq	Estimated q-axis inductive voltage value
	f4_phase_err_rad	Angle error
	f4_ol_speed_rad	Speed command value in open loop mode [rad/s]
	st_mod	Structure for the modulation module
	st_volt_comp	Structure for the voltage error compensation module
	st_bemf_observer	Structure for the inductive voltage observer
	st_pll_est	Structure for position and speed estimation
	st_pi_id	Structure for d-axis PI control
	st_pi_iq	Structure for q-axis PI control
	st_rotor_angle	Structure for rotor information
	st_motor	Structure for motor parameters
	*p_st_cc_init	Instance for the initial value of the st_current_control_t
	*p_st_cfg_init	Instance for the initial value of the st_cc_cfg_t
st_current_cfg_t Structure for setting the parameters for controlling the current control module	u1_flag_volt_err_comp_use	Enables or disables voltage error compensation.
	u2_offset_calc_time	Offset calculation time setting
	f4_ctrl_period	Control interval [s]
	f4_current_omega_hz	Natural frequency for the current control system [Hz]

Structure	Variable	Description
	f4_current_zeta	Attenuation coefficient for the current control system
	f4_id_up_step	Increment of the d-axis current
	f4_id_down_step	The amount of change in the d-axis current command value in d-axis current decrease mode [A]
	f4_iq_down_step_time_inv	The inverse of the change time in the q-axis current command value in q-axis current decrease mode [A]
	f4_ol_ref_id	d-axis current command value in open loop mode [A]
	st_motor	Structure for motor parameters
st_current_output_t Structure for the current control module output	u1_flag_offset_calc	Current offset flag
	f4_modu	U-phase duty cycle
	f4_modv	V-phase duty cycle
	f4_modw	W-phase duty cycle
	f4_neutral_duty	Duty cycle in offset measurement
	f4_va_max	Maximum voltage on the d and q axes [V]
st_current_output_t Structure for the current control module output	f4_ref_id_ctrl	d-axis current command value
	f4_speed_rad	Estimated speed [rad/s]
	f4_ed	Estimated d-axis inductive voltage value
	f4_eq	Estimated q-axis inductive voltage value
	f4_phase_err_rad	Angle error
st_current_input_t Structure for the current control module input	u1_state_id_ref	Status of the d axis
	u1_state_iq_ref	Status of the q axis
	f4_iu_ad	U-phase current value [A]
	f4_iv_ad	V-phase current value [A]
	f4_iw_ad	W-phase current value [A]
	f4_vdc_ad	Bus voltage value [V]
	f4_id_ref	d-axis current command value [A]
	f4_iq_ref	q-axis current command value [A]
	f4_ol_speed_rad	Speed command value in open loop mode
st_bemf_observer_cfg_t Structure for the inductive voltage observer module input	f4_e_obs_omega_hz	Natural frequency for the inductive voltage estimation system [Hz]
	f4_e_obs_zeta	Attenuation coefficient for the inductive voltage estimation system
	f4_pll_est_omega_hz	Natural frequency for the position estimation system [Hz]
	f4_pll_est_zeta	Attenuation coefficient for the position estimation system

Structure	Variable	Description
st_cc_cfg_t Structure for initial values of current control module control	f4_cfg_carrier_freq	Carrier frequency [kHz]
	f4_cfg_deadtime	Dead Time [μs]
	f4_ad_sampling_time	A/D sampling time [μs]
	f4_current_verr_comp_limit	Voltage Error Compensation Period Limiter Value
	st_cc_init_cfg	Structure for the initial value of the current control module control parameter
	st_bemf_init_cfg	Structure for Initial Value of Induced Voltage Observer Module Input

5.3.7 Macro definition

Table 5-27 lists the macros that are used for the current control module.

Table 5-27 List of macros

File name	Macro name	Defined value	Description
r_motor_current_api.h	CURRENT_STATE_ID_ZERO_CONST	0	Current status for the d axis: d-axis current always-0 mode
	CURRENT_STATE_ID_INPUT	1	Current status for the d axis: d-axis current command input mode
	CURRENT_STATE_ID_MANUAL	2	Current status for the d axis: d-axis fixed-command mode
	CURRENT_STATE_ID_UP	3	Current status for the d axis: d-axis current increase mode
	CURRENT_STATE_ID_DOWN	4	Current status for the d axis: d-axis current decrease mode
	CURRENT_STATE_IQ_ZERO_CONST	0	Current status for the q axis: q-axis current always-0 mode
	CURRENT_STATE_IQ_SPEED_PI_OUTPUT	1	Current status for the q axis: q-axis command PI input mode
	CURRENT_STATE_IQ_AUTO_ADJ	2	Current status for the q axis: q-axis current switch-to-sensorless mode
	CURRENT_STATE_IQ_DOWN	3	Current status for the q axis: q-axis current decrease mode
	CURRENT_VERR_COMP_LIMIT_A	(MOTOR_MCU_CFG_CARRIER_FREQ_A * INVERTER_CFG_DEADTIME_A / 1000.0f)	Voltage error compensation period limiter value (motor 1) For details about MOTOR_MCU_CFG_CARRIER, see r_motor_module_cfg.h. For details about INVERTER_CFG_DEADTIME, see r_motor_inverter_cfg.h.
	CURRENT_VERR_COMP_LIMIT_B	(MOTOR_MCU_CFG_CARRIER_FREQ_B * INVERTER_CFG_DEADTIME_B / 1000.0f)	Voltage error compensation period limiter value (motor 2) For details about MOTOR_MCU_CFG_CARRIER, see r_motor_module_cfg.h. For details about INVERTER_CFG_DEADTIME, see r_motor_inverter_cfg.h.

5.3.8 Current Control Module Variable Information

Table 5-28 lists the variable information for the current control module.

Table 5-28 List of current control module variables

Structure	Variable	Description
st_current_control_t Structure for current control module	g_st_cc_a	Variable for motor1
	g_st_cc_a_init	Variable for initial value of motor1
	g_st_cc_b	Variable for motor2
	g_st_cc_b_init	Variable for initial value of motor2
st_current_cfg_t Structure for setting current control module control parameters	st_cur_cfg	(Local variables for function internals)
st_current_output_t Structure for the output of the current control module	st_current_output	(Local variables for function internals)
st_current_input_t Structure for the input of the current control module	st_current_input	(Local variables for function internals)
st_bemf_observer_cfg_t Structure for Induced Voltage Observer Module Input	st_bemf_obs_cfg	(Local variables for function internals)
st_cc_cfg_t Structure for the initial value of the current control module	g_st_cc_a_cfg	Variable for motor1
	g_st_cc_b_cfg	Variable for motor2

5.3.9 Adjustment and configuration of parameters

5.3.9.1 Adjustment of the natural frequency and attenuation coefficient for the current control system

In the current control module, the control gain is adjusted by tuning the natural frequency for the current control system and the attenuation coefficient for the current control system. Set the natural frequency for the current control system in proportion to the frequency at which to perform current control. The natural frequency can be set to about 1/10 of the current control frequency. However, in many cases, a lower value may be set in consideration of noise during position detection and current detection.

For the attenuation coefficient for the current control system, a value in the range from 0.7 to 1.0 is ordinarily set. Setting a value nearer to 1.0 makes response more stable and moderate.

When you set or update the values of the natural frequency and attenuation coefficient for the current control system, use the following variables of the st_current_cfg_t structure (the structure for setting the parameters for controlling the current control module). After you have set the desired values in these variables, apply them by using R_MOTOR_CURRENT_ParameterUpdate (the API function for updating the parameters that control the current control module).

The natural frequency and attenuation coefficient for the current control system can be adjusted from RMW.

To set the natural frequency for the current control system, use f4_current_omega_hz. (See Table 5-8.)

To set the attenuation coefficient for the current control system, use f4_current_zeta. (See Table 5-8.)

5.3.9.2 Setting the parameters for current control

Because the current control module uses the control interval and motor parameters, the control parameter configuration (R_MOTOR_CURRENT_ParameterUpdate) can be used to update the parameters. For details about the items that can be set, see the description of the st_current_cfg_t structure (structure for setting the parameters for controlling the current control module).

5.3.9.3 Setting the initial values of the parameters for current control

The configurations of the current control module can be specified by using r_motor_module_cfg.h. The values set in this file are applied as initial values at system startup. For details about the items to be set, see 5.3.7 Macro definition.

5.3.10 Adjusting the AD Conversion Start Request Cycle

In the case of two-motor control, if two motors are controlled within one interrupt processing, the processing will be sequential, so there will be a difference in control between motor 1 and motor 2.

In order to eliminate the difference, interrupts due to the AD conversion start request are generated at different timings for motors 1 and motor 2.

In addition, by using the AD conversion start request decimation function and generating an interrupt once every two periods (100 µs), motor 1 and motor 2 can alternately perform interrupt processing.

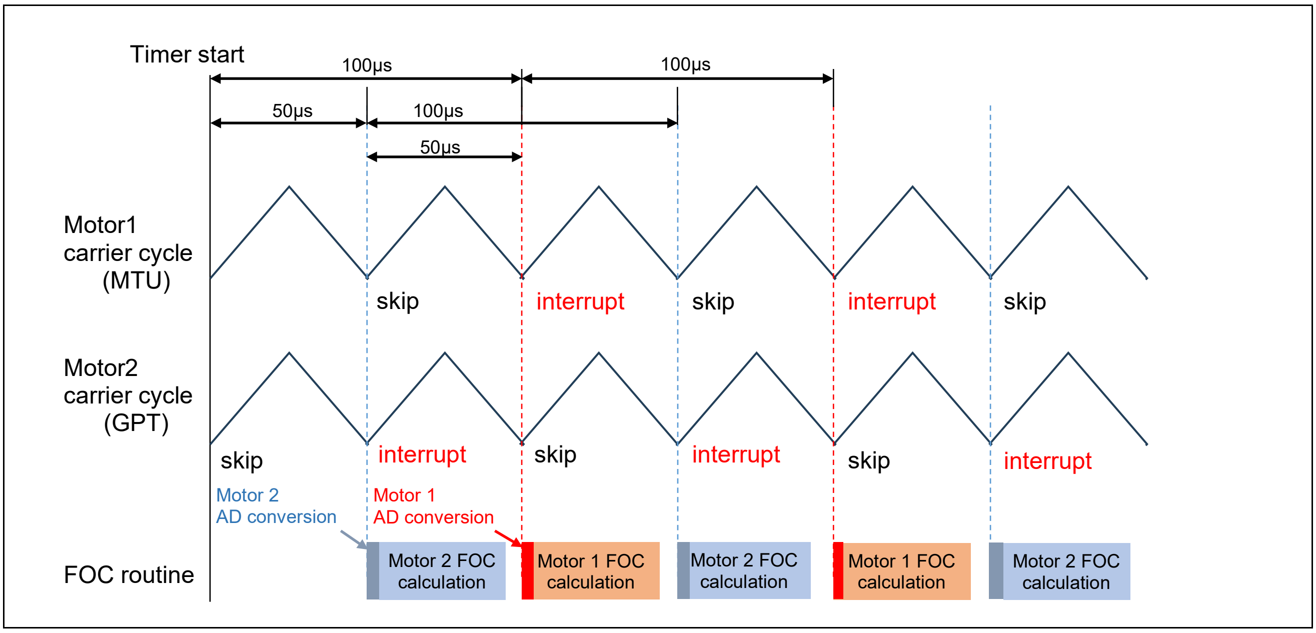


Figure5-14 AD Conversion Completion Interrupt Cycle

On the motor 2 side, the first AD conversion end interrupt occurs 50 [µs] after the timer starts.

After that, the AD conversion end interrupt period on the motor 2 side is 100 [µs], so the AD conversion end interrupt between motor 1 and motor 2 does not occur at the same time.

Note that the AD conversion end interrupts of motor 1 and motor 2 occur alternately at 50 [µs] periods. Therefore, it is necessary to keep the interrupt processing time (including the processing time when an error occurs) within 50 [µs].

5.4 Modulation (current control module)

A modulated voltage can be output to improve the efficiency of voltage usage. The modulation operation is set from the API of the current control module.

5.4.1 Description of the functionality

With this module, you can select from two types of pulse width modulation drive methods.

(a) Sine wave modulation (MOD_METHOD_SPWM)
The modulation factor m is defined as follows.

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

m : Modulation ratio V : Reference voltage E : Inverter input voltage

(b) Space Vector Modulation (MOD_METHOD_SVPWM) *
In vector control of a permanent magnet synchronous motor, generally, the desired voltage command value of each phase is generated sinusoidally. However, if the generated value is used as-is for the modulation wave for PWM generation, voltage utilization as applied to the motor (in terms of line voltage) is limited to a maximum of 86.7% with respect to inverter bus voltage. As such, as shown in the following expression, the average of the maximum and minimum values is calculated for the voltage command value of each phase, and the value obtained by subtracting the average from the voltage command value of each phase is used as the modulation wave. As a result, the maximum amplitude of the modulation wave is multiplied by $\sqrt{3}/2$, while voltage utilization becomes 100% and line voltage is unchanged.

$$\begin{pmatrix} V_u' \\ V_v' \\ V_w' \end{pmatrix} = \begin{pmatrix} V_u \\ V_v \\ V_w \end{pmatrix} + \Delta V \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\because \Delta \mathrm{V} = -\frac{V_{max}+V_{min}}{2}\ ,\ V_{max} = max\{V_u, V_v, V_w\}\ ,\ V_{min} = min\{V_u, V_v, V_w\}$$

V_u, V_v, V_w: Command values of U-, V-, and W-phases

V_u', V_v', V_w': Command values of U-, V-, and W-phases for PWM generation (modulation wave)

The modulation factor m is defined as follows.

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

m : Modulation ratio V’ : Reference phase voltage for PWM
E : Inverter input voltage

5.4.2 Configurations

Table 5-29 lists the configurations for the modulation function.

Table 5-29 List of configurations

File name	Macro name	Set value	Description
r_motor_module_cfg.h	CURRENT_CFG_MODULATION_METHOD	(MOD_METHOD_SVPWM)	Pulse-width modulation drive mode

5.4.3 Structures

Table 5-30 lists the structures that are used for the modulation function.

Table 5-30 List of variables

Structure	Variable	Description
st_mod_t	f4_vdc	Bus voltage value [V]
	f4_1_div_vdc	1/f4_vdc
	f4_voltage_error_ratio	Voltage error ratio
	f4_max_duty	Maximum PWM duty cycle
	f4_min_duty	Minimum PWM duty cycle
	f4_neutral_duty	Intermediate value of the PWM duty cycle

5.4.4 Macro definition

Table 5-31 lists the macros that are used for the modulation function.

Table 5-31 List of macros

File name	Macro name	Defined value	Description
r_motor_current_modulation.h	MOD_DEFAULT_MAX_DUTY	1.0f	Maximum PWM duty cycle
	MOD_METHOD_SPWM	0	Pulse-width modulation drive mode : Sinusoidal PWM
	MOD_METHOD_SVPWM	1	Pulse-width modulation drive mode : Spatial vector PWM
	MOD_VDC_TO_VAMAX_MULT	0.6124f	Coefficient of the conversion from the input voltage to the maximum voltage
	MOD_SVPWM_MULT	1.155f	Coefficient for spatial vector PWM

5.4.5 Adjustment and configuration of parameters

There are no parameters to be set by the user for the modulation function.

5.5 Voltage error compensation (current control module)

The voltage error compensation function corrects for the effects of output voltage dead time. It operates through the API of the current control module.

5.5.1 Description of the functionality

In the voltage PWM converter, to prevent the switching elements of the upper and lower sides from creating a short circuit, a dead time during which the two elements are simultaneously turned off is set. Therefore, an error arises between the voltage command value and the voltage actually applied to the motor, degrading the control precision. Voltage error compensation is implemented to reduce this error.

The current dependency of the voltage error depends on the current (direction and magnitude), dead time, and the switching characteristics of the power elements to be used, and has the characteristics shown below. Voltage error compensation is achieved by applying the inverse voltage pattern of the voltage error (as shown below) to the voltage command value according to the current.

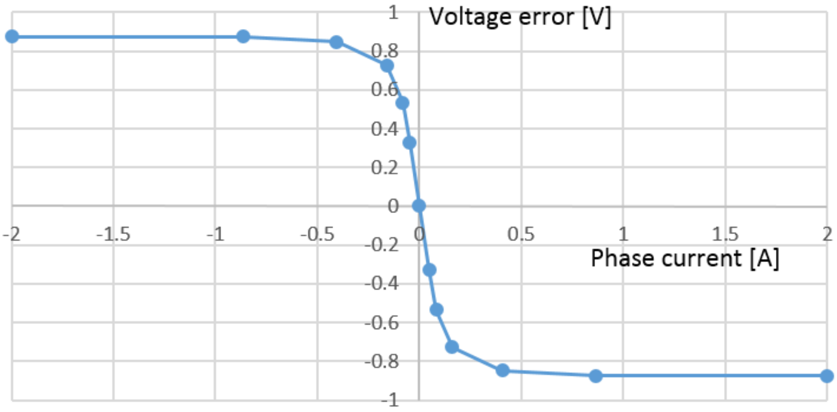


Figure5-15 Current dependency in the voltage error (example)

5.5.2 Adjustment and configuration of parameters

5.5.2.1 Setting the flag for whether to enable the voltage error compensation function

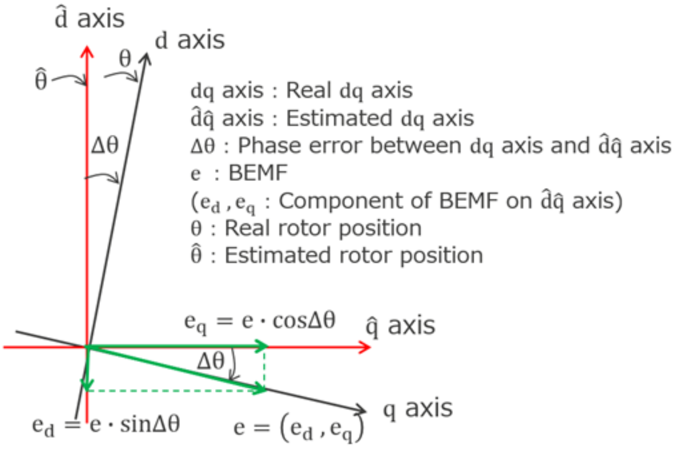
The voltage error compensation function is enabled by setting "u1_flag_volt_err_comp_use" (flag for whether to enable the voltage error compensation function) to MTR_FLG_SET when R_MOTOR_CURRENT_ParameterUpdate (setting of the control parameter for the current control module) is called. To disable the function, set this flag to MTR_FLG_CLR.

5.6 Inductive voltage observer (current control module)

To perform sensorless vector control, the position information must somehow be estimated. The sample program uses an inductive voltage observer.

5.6.1 Description of the functionality

When an inductive voltage observer is used, the observer is used to estimate the inductive voltage. Then the phase error between the estimated d/q axis and the actual d/q axis is calculated to obtain the position and speed.



d̂ axis
d axis
θ̂
θ
Δθ
dq axis : Real dq axis
d̂q̂ axis : Estimated dq axis
Δθ : Phase error between dq axis and d̂q̂ axis
e : BEMF
(e_d , e_q : Component of BEMF on d̂q̂ axis)
θ : Real rotor position
θ̂ : Estimated rotor position
e_q = e・cosΔθ
q̂ axis
Δθ
e_d = e・sinΔθ
e = (e_d , e_q)
q axis

Figure5-16 Inductive voltage on the estimated d/q-axis

From Figure5-16, the voltage equations of the estimated d and q axes can be represented as follows:

$$v_d^* = (R + sL_d)i_d - \omega^* L_q i_q + e_d$$

$$v_q^* = \left(R + sL_q\right)i_q + \omega^* L_d i_d + e_q$$

Here, $-\omega^* L_q i_q + e_d$ and $\omega^* L_d i_d + e_q$ are treated as voltage disturbance and put to $-d_d, -d_q$.

$$v_d^* = (R + sL_d)i_d - d_d$$

$$v_q^* = \left(R + sL_q\right)i_q - d_q$$

An estimate equation for the d-axis inductive voltage is derived first. Rearrange the d-axis voltage equation as follows:

$$si_d = \frac{v_d^*}{L_d} - \frac{R}{L_d} i_d + \frac{d_d}{L_d}$$

Based on the above equation, an equation of state is composed with the state variables i_d (d-axis current) and d (voltage disturbance).

$$s i_d = -\frac{R}{L_d} i_d + \frac{d}{L_d} + \frac{v_d^*}{L_d}$$

$$s d = s d_d$$

Here, if the estimated value for i_d and d is $\widehat{i_d}$ and $\hat{d}$, respectively, the estimated equation of state for the observer side can be represented as follows, by multiplying the estimated gains K_{Ed1} and K_{Ed2} to the estimated error.

$$s\widehat{i_d} = -\frac{R}{L_d}\widehat{i_d} + \frac{\hat{d}}{L_d} + \frac{v_d^*}{L_d} + K_{Ed1}(i_d - \widehat{i_d})$$

$$s\hat{d} = K_{Ed2}(i_d - \widehat{i_d})$$

From the above equation, $\widehat{i_d}$ and $\hat{d}$ can be represented as follows:

$$\widehat{i_d} = \frac{\frac{K_{Ed2}}{L_d}}{s^2 + \left(\frac{R}{L_d} + K_{Ed1}\right)s + \frac{K_{Ed2}}{L_d}}\left\{\left(1 + \frac{K_{Ed1}}{K_{Ed2}} L_d s\right) i_d + \frac{s}{K_{Ed2}} v_d^*\right\}$$

$$\hat{d} = \widehat{d_d} = \frac{\frac{K_{Ed2}}{L_d}}{s^2 + \left(\frac{R}{L_d} + K_{Ed1}\right)s + \frac{K_{Ed2}}{L_d}}\{(L_d s + R) i_d - v_d^*\}$$

Looking at the above equation, $\widehat{i_d}$ and $\widehat{d_d}$ can be represented in a quadratic form, with the inputs i_d and v_d^*. Also, the natural frequency ω_n and attenuation coefficient ζ are as follows:

$$\omega_n = \sqrt{\frac{K_{Ed2}}{L_d}}$$

$$\zeta = \frac{\frac{R}{L_d} + K_{Ed1}}{2\sqrt{\frac{K_{Ed2}}{L_d}}}$$

The estimated gains K_{Ed1} and K_{Ed2} of the d-axis inductive voltage estimation system can be represented as follows by using ω_n and ζ.

$$K_{Ed1} = 2\zeta_{EG}\omega_{EG} - \frac{R}{L_d}$$

$$K_{Ed2} = \omega_{EG}^2 L_d$$

ω_{EG}: Natural frequency for the inductive voltage estimation system

ζ_{EG}: Attenuation coefficient for the inductive voltage estimation system

The estimated equation of state is further rearranged as follows:

$$\hat{i_d} = \frac{1}{s}\left\{-\frac{R}{L_d}\hat{i_d} + \frac{\widehat{d_d}}{L_d} + \frac{v_d^*}{L_d} + K_{Ed1}(i_d - \hat{i_d})\right\}$$

$$\widehat{d_d} = \frac{1}{s}\{K_{Ed2}(i_d - \hat{i_d})\}$$

From the above equation, the block diagram of d-axis inductive voltage estimation is as shown in Figure5-17.

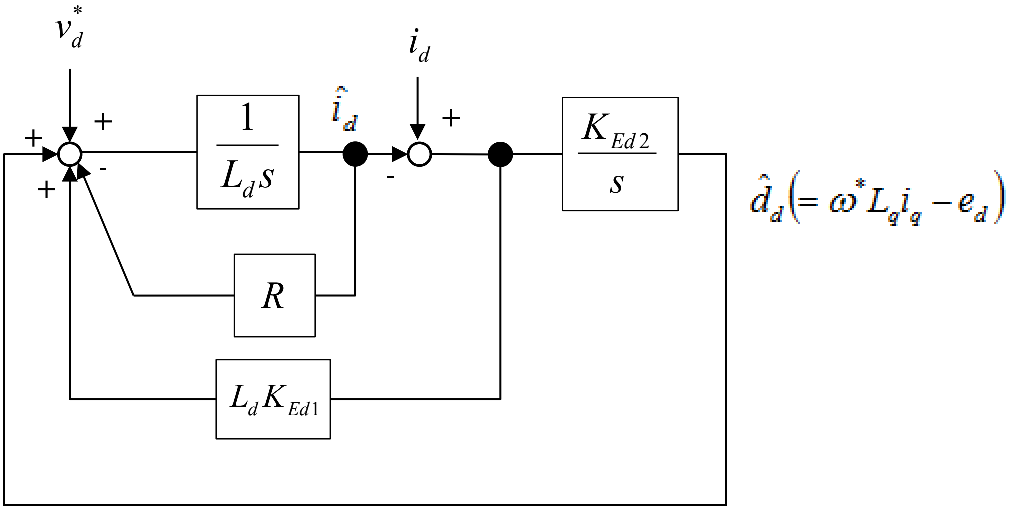


Figure5-17 Block diagram of the d-axis inductive voltage observer

The q-axis side can be calculated in the same way, and $\widehat{\imath_q}$ and $\hat{d}$ can be represented as follows:K_{Eq1} and K_{Eq2} are the estimated gains for the q axis side.

$$\widehat{\imath_q} = \frac{\frac{K_{Eq2}}{L_q}}{s^2 + \left(\frac{R}{L_q} + K_{Eq1}\right)s + \frac{K_{Eq2}}{L_q}} \left\{\left(1 + \frac{K_{Eq1}}{K_{Eq2}} L_q s\right) i_q + \frac{s}{K_{Eq2}} v_q^*\right\}$$

$$\hat{d} = \widehat{d_q} = \frac{\frac{K_{Eq2}}{L_q}}{s^2 + \left(\frac{R}{L_q} + K_{Eq1}\right)s + \frac{K_{Eq2}}{L_q}} \left\{\left(L_q s + R\right) i_q - v_q^*\right\}$$

In the same way as the d axis, looking at the above equation, $\widehat{\imath_q}$ and $\widehat{d_q}$ can be represented in a quadratic form, with the inputs i_q and v_q^*. Also, the natural frequency ω_n and attenuation coefficient ζ are as follows:

$$\omega_n = \sqrt{\frac{K_{Eq2}}{L_q}}$$

$$\zeta = \frac{\frac{R}{L_q} + K_{Eq1}}{2\sqrt{\frac{K_{Eq2}}{L_q}}}$$

Therefore, the estimated gains K_{Eq1} and K_{Eq2} of the q-axis inductive voltage estimation system can be represented as follows:

$$K_{Eq1} = 2\zeta_{EG}\omega_{EG} - \frac{R}{L_q}$$

$$K_{Eq2} = \omega_{EG}^2 L_q$$

ω_{EG}: Natural frequency for the inductive voltage estimation system

ζ_{EG}: Attenuation coefficient for the inductive voltage estimation system

In the same way as the d axis, the estimated equation of state can be represented as follows:

$$\widehat{i_q} = \frac{1}{s}\left\{-\frac{R}{L_q}\widehat{i_q} + \frac{\widehat{d_q}}{L_q} + \frac{v_q^*}{L_q} + K_{Eq1}\left(i_q - \widehat{i_q}\right)\right\}$$

$$\widehat{d_q} = \frac{1}{s}\left\{K_{Eq2}\left(i_q - \widehat{i_q}\right)\right\}$$

From the above equation, the block diagram of q-axis inductive voltage estimation is as shown in Figure5-18.

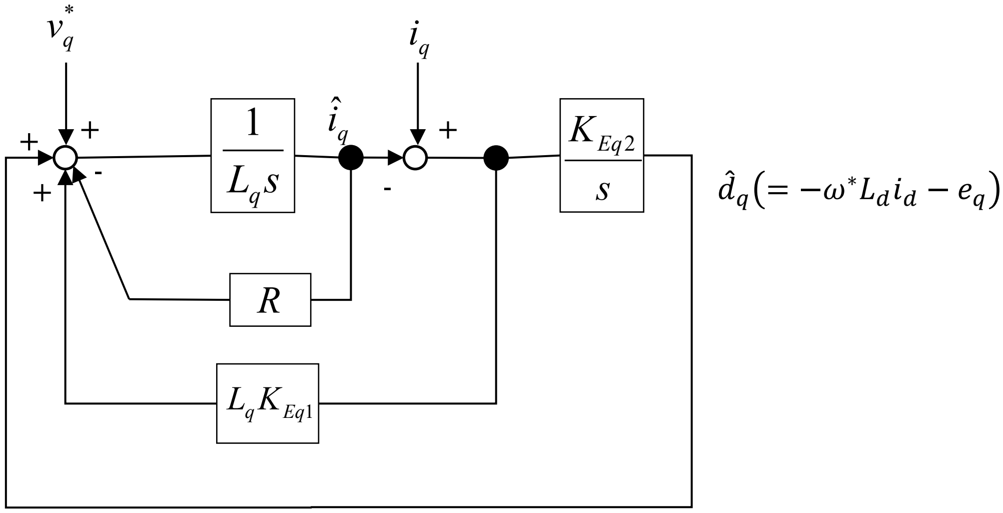


Figure5-18 Block diagram of the q-axis inductive voltage observer

Next, the inductive voltages can be calculated from voltage disturbances $\widehat{d_d}$ and $\widehat{d_q}$.

$$e_d = -\widehat{d_d} + \omega^* L_q i_q$$

$$e_q = -\widehat{d_q} - \omega^* L_d i_d$$

$$\Delta\theta = \text{atan}\left(\frac{e_d}{e_q}\right)$$

From the above, the phase error $\Delta\theta$ between the actual d/q axis and the estimated d/q axis can be obtained.

Finally, reflect the phase error $\Delta\theta$ to the estimated d/q-axis phase. The reflection is performed according to the block diagram shown in Figure5-19.

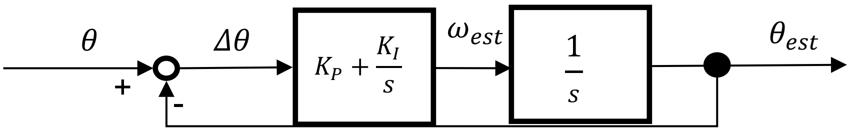


Figure5-19 Block diagram of the magnetic pole estimation system

Here, the closed-loop transfer function from the magnetic pole position θ of the motor to the estimated magnetic pole position θ_{est} can be represented as follows:

$$\frac{\theta_{est}(s)}{\theta(s)} = \frac{K_I\left(s\frac{K_P}{K_I}+1\right)}{s^2 + K_P s + K_I}$$

Therefore, the transfer function from the motor's actual magnetic pole position to the estimated magnetic pole position is in quadratic form, and the natural frequency ω_n and attenuation coefficient ζ of the magnetic pole position estimation system are determined by the following formulas.

$$\omega_n = \sqrt{K_I}$$

$$\zeta = \frac{K_P}{2\sqrt{K_I}}$$

As a result, the control gains $K_{P\_phase\_error}$ and $K_{I\_phase\_error}$ for the magnetic pole position estimation system can be represented as follows:

$$K_{P\_phase\_error} = 2\zeta_{\Delta\theta}\omega_{\Delta\theta}$$

$$K_{I\_phase\_error} = \omega_{\Delta\theta}^2$$

$\omega_{\Delta\theta}$: Natural frequency for the magnetic pole position estimation system

$\zeta_{\Delta\theta}$: Attenuation coefficient for the magnetic pole position estimation system

As above, position and speed estimation is complete.

5.6.2 Configurations

Table 5-32 lists the configurations for the inductive voltage observer.

Table 5-32 List of configurations

File name	Macro name	Set value	Description
r_motor_module_cfg.h	CURRENT_CFG_E_OBS_OMEGA_A	1000	Natural frequency for the inductive voltage observer [Hz] (motor 1)
	CURRENT_CFG_E_OBS_ZETA_A	1	Attenuation coefficient for the inductive voltage observer (motor 1)
	CURRENT_CFG_PLL_EST_OMEGA_A	20	Natural frequency for the inductive voltage observer [Hz] (motor 1)
	CURRENT_CFG_PLL_EST_ZETA_A	1	Attenuation coefficient for the inductive voltage observer (motor 1)
	CURRENT_CFG_E_OBS_OMEGA_B	1000	Natural frequency for the inductive voltage observer [Hz] (motor 2)
	CURRENT_CFG_E_OBS_ZETA_B	1	Attenuation coefficient for the inductive voltage observer (motor 2)
	CURRENT_CFG_PLL_EST_OMEGA_B	20	Natural frequency for the inductive voltage observer [Hz] (motor 2)
	CURRENT_CFG_PLL_EST_ZETA_B	1	Attenuation coefficient for the inductive voltage observer (motor 2)

5.6.3 Adjustment and configuration of parameters

5.6.3.1 Configuration of the inductive voltage observer

In sensorless vector control, the inductive voltage observed must be always enabled. Therefore, there are no parameters to be set by the user.

5.7 Speed control module

The speed control module controls the motor so that the speed follows the speed command. When receiving a speed command value, this module outputs a current command value accordingly. This module also controls the submodules, which are magnetic flux weakening control, open-loop damping control, and switching-to-sensorless control.

5.7.1 Function

Table 5-33 lists the functions of the speed control module.

Table 5-33 List of functions of the speed control module

Function	Description
Speed control	Calculates and outputs a current command value that allows the speed to follow the speed command value.
Speed command setting	Sets a speed command value in the speed control module.
Magnetic flux weakening control setting	Uses the magnetic flux weakening control to calculate and set the current command values for the d and q axes.
Open-loop damping control setting	Sets the parameters for open-loop damping control.
Switching-to-sensorless control setting	Sets the parameters for the module that switches operation from open-loop control to sensorless control.

5.7.2 Module configuration diagram

Figure5-20 shows the module configuration of the speed control module.

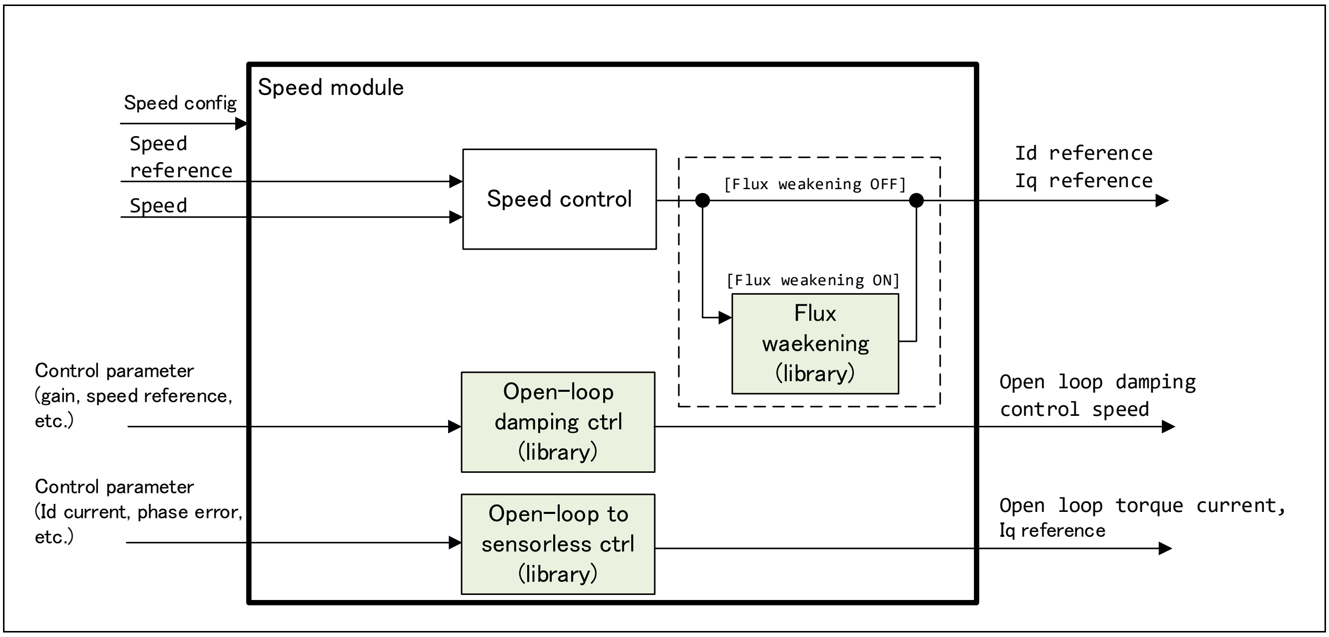

Speed module
Speed config
Speed reference
Speed
Speed control
[Flux weakening OFF]
[Flux weakening ON]
Flux waekening (library)
Id reference
Iq reference
Control parameter (gain, speed reference, etc.)
Open-loop damping ctrl (library)
Open loop damping control speed
Control parameter (Id current, phase error, etc.)
Open-loop to sensorless ctrl (library)
Open loop torque current, Iq reference

Figure5-20 Speed control module configuration diagram

For details about the submodules of the speed control module, which are magnetic flux weakening control, open-loop damping control, and switching-to-sensorless control, see 5.8 Magnetic flux weakening control (speed control module), 5.9 Open-loop damping control (speed control module), and 5.10 Switching-to-sensorless control (speed control module).

5.7.3 Flowcharts

Figure5-21 shows the flowchart for speed control.

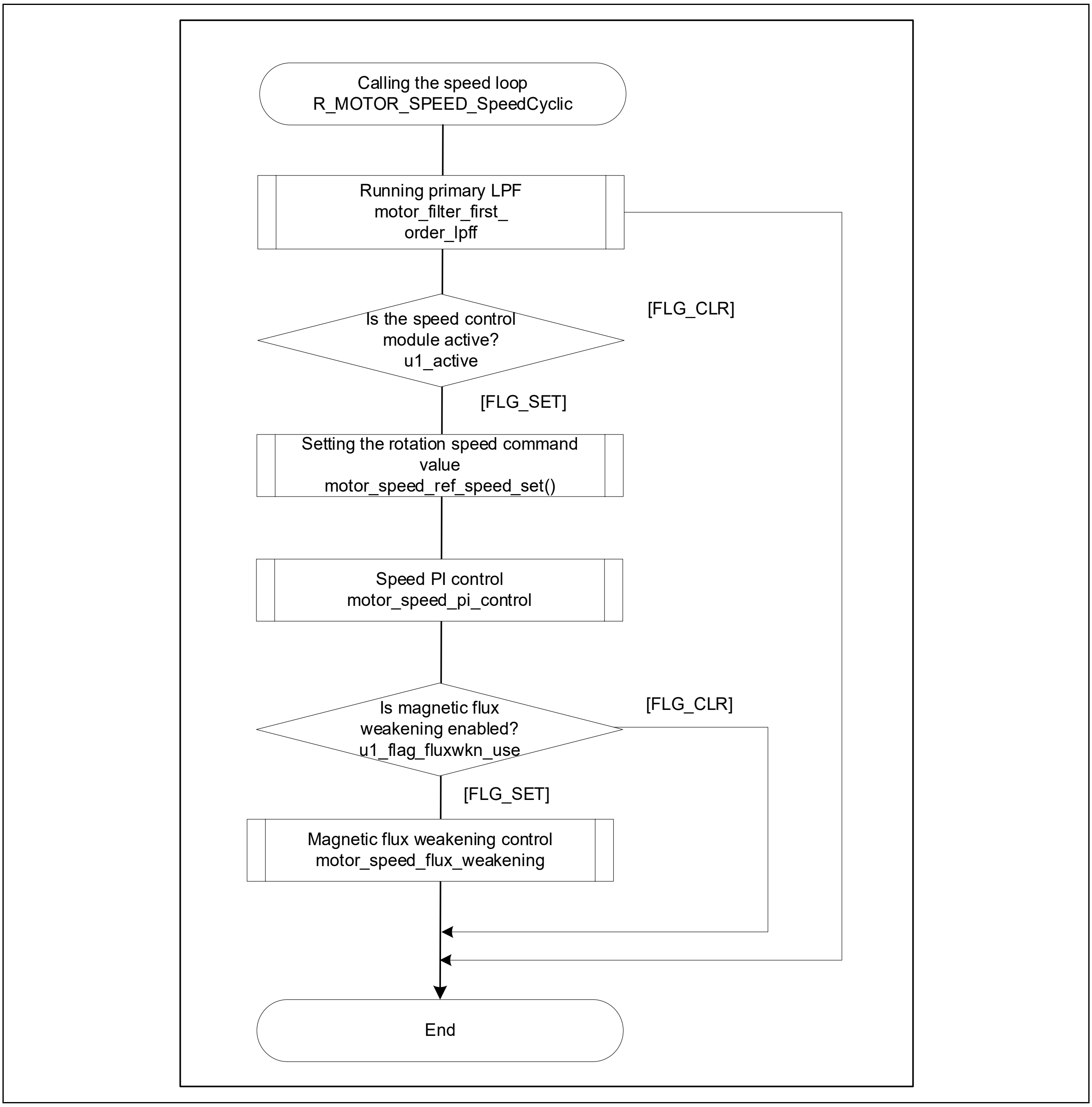


Figure5-21 Flowchart for speed control

5.7.4 API

Table 5-34 lists the API functions of the speed control module.

Table 5-34 List of API functions

API	Description
R_MOTOR_SPEED_Open	Generates an instance of the speed control module.
R_MOTOR_SPEED_Close	Places the module in a reset state.
R_MOTOR_SPEED_Reset	Initializes the module.
R_MOTOR_SPEED_Run	Activates the module.
R_MOTOR_SPEED_ParameterSet	Inputs the variable information that is used for speed control.
R_MOTOR_SPEED_ParameterGet	Acquires the speed control results that are output.
R_MOTOR_SPEED_ParameterUpdate	Updates the control parameters of the module.
R_MOTOR_SPEED_SpdRefSet	Sets the speed command value.
R_MOTOR_SPEED_SpeedCyclic	Performs speed control.
R_MOTOR_SPEED_Opl2lessReferenceIqCalc	Calculates the q-axis current command value when switching to sensorless control
R_MOTOR_SPEED_Opl2lessPreprocess	Calculate the torque current used to perform switching-to-sensorless control.
R_MOTOR_SPEED_OplDampCtrl	Starts open-loop damping control.
R_MOTOR_SPEED_OplDampReset	Resets open-loop damping control.
R_MOTOR_SPEED_HuntingSuppress	Performs a measure against estimation speed discontinuity of the q-axis current command value when switching controls.
R_MOTOR_SPEED_SwitchingFlagSet	Sets a flag that assigns an estimated speed to the speed command value when switching controls.

Note: Since the speed control module API needs to control each motor (motor 1, motor 2), specify a structure variable pointer for the speed module (for motor 1: &g_st_sc_a / for motor 2: &g_st_sc_b) by argument.

5.7.5 Configurations

Table 5-35 and Table 5-36 lists the configurations for the speed control module. Set up the functions to be used and the necessary parameters.

Table 5-37 shows the initial values.

Table 5-35 List of configurations (motor1)

File name	Macro name	Description
r_motor_module_cfg.h	SPEED_CFG_FLUX_WEAKENING_A	Magnetic flux weakening control setting (motor 1) Enable: MTR_ENABLE Disable: MTR_DISABLE
	SPEED_CFG_LESS_SWITCH_A	Switching-to-sensorless control setting (motor 1) Enable: MTR_ENABLE Disable: MTR_DISABLE
	SPEED_CFG_OPENLOOP_DAMPING_A	Open-loop damping control setting (motor 1) Enable: MTR_ENABLE Disable: MTR_DISABLE
	SPEED_CFG_CTRL_PERIOD_A	Control interval setting [s] (motor 1)
	SPEED_CFG_OMEGA_A	Natural frequency for the speed control system [Hz] (motor 1)
	SPEED_CFG_ZETA_A	Attenuation coefficient for the speed control system (motor 1)
	SPEED_CFG_LPF_OMEGA_A	LPF bandwidth for the speed control system [Hz] (motor 1)
	SPEED_CFG_SPEED_LIMIT_RPM_A	Speed limit value [rpm] (motor 1)
	SPEED_CFG_RATE_LIMIT_RPM_A	Acceleration limit [rpm/s] (motor 1)
	SPEED_OPL2LESS_SWITCH_TIME_A	Processing time required for switching-to-sensorless processing (motor 1)
	SPEED_OPL_DAMP_ED_HPF_OMEGA_A	d-axis inductive voltage estimation HPF cut-off frequency [Hz] (motor 1)
	SPEED_OPL_DAMP_ZETA_A	Default setting value of the attenuation coefficient for the open-loop damping control system (motor 1)
	SPEED_OPL_DAMP_FB_SPEED_LIMIT_RATE_A	Default setting value for the multiplier to be used to calculate the output speed command compensation value limiter. (motor 1)

Table 5-36 List of configurations (motor2)

File name	Macro name	Description
r_motor_module_cfg.h	SPEED_CFG_FLUX_WEAKENING_B	Magnetic flux weakening control setting (motor 2) Enable: MTR_ENABLE Disable: MTR_DISABLE
	SPEED_CFG_LESS_SWITCH_B	Switching-to-sensorless control setting (motor 2) Enable: MTR_ENABLE Disable: MTR_DISABLE
	SPEED_CFG_OPENLOOP_DAMPING_B	Open-loop damping control setting (motor 2) Enable: MTR_ENABLE Disable: MTR_DISABLE
	SPEED_CFG_CTRL_PERIOD_B	Control interval setting [s] (motor 2)
	SPEED_CFG_OMEGA_B	Natural frequency for the speed control system [Hz] (motor 2)
	SPEED_CFG_ZETA_B	Attenuation coefficient for the speed control system (motor 2)
	SPEED_CFG_LPF_OMEGA_B	LPF bandwidth for the speed control system [Hz] (motor 2)
	SPEED_CFG_SPEED_LIMIT_RPM_B	Speed limit value [rpm] (motor 2)
	SPEED_CFG_RATE_LIMIT_RPM_B	Acceleration limit [rpm/s] (motor 2)
	SPEED_OPL2LESS_SWITCH_TIME_B	Processing time required for switching-to-sensorless processing (motor 2)
	SPEED_OPL_DAMP_ED_HPF_OMEGA_B	d-axis inductive voltage estimation HPF cut-off frequency [Hz] (motor 2)
	SPEED_OPL_DAMP_ZETA_B	Default setting value of the attenuation coefficient for the open-loop damping control system (motor 2)
	SPEED_OPL_DAMP_FB_SPEED_LIMIT_RATE_B	Default setting value for the multiplier to be used to calculate the output speed command compensation value limiter. (motor 2)

Table 5-37 List of initial values for configurations (motor1)

Macro name	Setting
SPEED_CFG_FLUX_WEAKENING_A	MTR_DISABLE
SPEED_CFG_LESS_SWITCH_A	MTR_ENABLE
SPEED_CFG_OPENLOOP_DAMPING_A	MTR_ENABLE
SPEED_CFG_CTRL_PERIOD_A	0.0005f
SPEED_CFG_OMEGA_A	3.0f
SPEED_CFG_ZETA_A	1.0f
SPEED_CFG_LPF_OMEGA_A	25.0f
SPEED_CFG_SPEED_LIMIT_RPM_A	4500.0f
SPEED_CFG_RATE_LIMIT_RPM_A	1000.0f
SPEED_OPL2LESS_SWITCH_TIME_A	0.025f
SPEED_OPL_DAMP_ED_HPF_OMEGA_A	2.5f
SPEED_OPL_DAMP_ZETA_A	1.0f
SPEED_OPL_DAMP_FB_SPEED_LIMIT_RATE_A	0.2f

Table 5-38 List of initial values for configurations (motor2)

Macro name	Setting
SPEED_CFG_FLUX_WEAKENING_B	MTR_DISABLE
SPEED_CFG_LESS_SWITCH_B	MTR_ENABLE
SPEED_CFG_OPENLOOP_DAMPING_B	MTR_ENABLE
SPEED_CFG_CTRL_PERIOD_B	0.0005f
SPEED_CFG_OMEGA_B	3.0f
SPEED_CFG_ZETA_B	1.0f
SPEED_CFG_LPF_OMEGA_B	25.0f
SPEED_CFG_SPEED_LIMIT_RPM_B	4500.0f
SPEED_CFG_RATE_LIMIT_RPM_B	1000.0f
SPEED_OPL2LESS_SWITCH_TIME_B	0.025f
SPEED_OPL_DAMP_ED_HPF_OMEGA_B	2.5f
SPEED_OPL_DAMP_ZETA_B	1.0f
SPEED_OPL_DAMP_FB_SPEED_LIMIT_RATE_B	0.2f

5.7.6 Structure and variable information

Table 5-39 lists the structures and variables for the speed control module. For the speed control module, the structure for the speed control module (g_st_sc) is defined by securing an instance of the module from the API.

Table 5-39 List of structures and variables (1)

Structure	Variable	Description
st_speed_control_t Structure for the speed control module	u1_active	Selects whether to enable the module.
	u1_state_speed_ref	The variable for managing the states that determine the speed command value. It manages the states as shown in "Macro definition" below.
	u1_flag_fluxwkn_use	Flag for whether to use magnetic flux weakening control
	u1_flag_switching	Flag for whether to use switching-to-sensorless control
	f4_speed_ctrl_period	Speed loop control interval [s]
	f4_ref_speed_rad_ctrl	Speed command value for control [rad/s]
	f4_ref_speed_rad	Speed command value output by the position control module during position control [rad/s]
	f4_ref_speed_rad_manual	Speed command value set by the user during speed control [rad/s]
	f4_speed_rad_ctrl	Speed calculated by the speed control module [rad/s]
	f4_speed_rad	Speed that is input [rad/s]
	f4_max_speed_rad	Maximum speed [rad/s]
	f4_speed_rate_limit_rad	Speed variation limit value [rad/s]
	f4_id_ref_output	d-axis current command value [A]
	f4_iq_ref_output	q-axis current command value [A]
	f4_va_max	Maximum voltage on the d and q axes [V]
	f4_id_ad	d-axis current value [A]
	f4_iq_ad	q-axis current value [A]
	f4_opl2less_sw_time	Processing time for switching-to-sensorless processing [s]
	f4_torque_current	Torque current [A]
	st_motor	Structure for motor constants
	st_pi_speed	Structure for PI control
	st_fluxwkn	Structure for magnetic flux weakening control
	st_opl_damp	Structure for open-loop damping control
	st_slpf	Structure for LPF
	*p_st_sc_init	Instance for the initial value of the st_speed_control_t
	*p_st_cfg_init	Instance for the initial value of the st_spd_cfg_t

Table 5-40 List of structures and variables (2)

Structure	Variable	Description
st_speed_cfg_t Structure for setting the parameters for controlling the speed control module	u1_flag_fluxwkn_use	Flag for whether to use magnetic flux weakening control
	f4_max_speed_rpm	Maximum speed [rpm]
	f4_speed_ctrl_period	Speed control interval [s]
	f4_speed_rate_limit_rpm	Speed variation limit value [rpm]
	f4_speed_omega_hz	Natural frequency for the speed control system [Hz]
	f4_speed_zeta	Attenuation coefficient for the speed control system
	f4_speed_lpf_hz	LPF for speed control [Hz]
	f4_opl2less_sw_time	Processing time for switching-to-sensorless processing [s]
	f4_ed_hpf_omega	d-axis inductive voltage HPF cut-off frequency [Hz]
	f4_ol_damping_zeta	Attenuation coefficient for open-loop damping control
	f4_ol_damping_fb_limit_rate	Feedback limit rate for open-loop damping control
	f4_ol_ref_id	d-axis current command value in open loop mode [A]
	f4_id_down_speed_rpm	Speed (mechanical angle) when start decreasing the d-axis current command value [rpm]
	st_motor	Structure for motor constants
st_speed_input_t Structure for speed control module input	u1_state_speed_ref	Speed command status
	f4_speed_rad	Speed that is to be input [rad/s]
	f4_va_max	Maximum voltage in the d and q axes [V]
st_speed_output_t Structure for speed control module output	f4_id_ref	d-axis current command value [A]
	f4_iq_ref	q-axis current command value [A]
	f4_ref_speed_rad_ctrl	Speed that is used for PI control [rad/s]
	f4_speed_rad_lpf	Speed after LPF [rad/s]
st_spd_cfg_t Structure for speed module control initial value	f4_pi_integral_limit_iq	iq PI Speed Limit
	f4_lim_iq	iq Speed Limit
	st_sc_init_cfg	Structure for Initial Values of Speed Module Control Parameters

5.7.7 Macro definition

Table 5-41 lists the macros of the speed control module.

Table 5-41 List of macros

File name	Macro name	Defined value	Remarks
r_motor_speed_api.h	SPEED_STATE_ZERO_CONST	0	This macro is used to manage the state of the speed control module. The speed command value is always 0.
	SPEED_STATE_POSITION_CONTROL_OUTPUT	1	This macro is used to manage the state of the speed control module. The speed command value is used as the output of the position control module.
	SPEED_STATE_MANUAL	2	This macro is used to manage the state of the speed control module. The speed command value becomes the user-specified value.

5.7.8 Variable information of the speed control module

Table 5-42 lists the variable information for the current control module.

Table 5-42 List of variable information of the current control module

Structure name	Variable name	Remarks
st_speed_control_t Structure for speed control module	g_st_sc_a	Variable for motor1
	g_st_sc_a_init	Variable for initial value of motor1
	g_st_sc_b	Variable for motor2
	g_st_sc_b_init	Variable for initial value of motor2
st_speed_cfg_t	st_spd_cfg	(Local variables for function internals)
st_speed_input_t	st_speed_input	(Local variables for function internals)
st_speed_output_t	st_speed_output	(Local variables for function internals)
st_spd_cfg_t Structure for speed module control initial value	g_st_sc_a_cfg	Variable for motor1
	g_st_sc_b_cfg	Variable for motor2

5.7.9 Adjustment and configuration of parameters

5.7.9.1 Adjustment of the natural frequency and attenuation coefficient for the speed control system

In the speed control module, the control gain is adjusted by tuning the natural frequency for the speed control system and the attenuation coefficient for the speed control system. Increasing the natural frequency for the speed control system improves the responsiveness, expanding the following capability of the speed to the commanded speed. The maximum settable natural frequency for speed control is limited to 1/3 of the maximum settable natural frequency for current control to prevent interference with current control. For the attenuation coefficient for the speed control system, a value in the range from 0.7 to 1.0 is ordinarily set. Setting a value nearer to 1.0 makes response more stable and moderate. Make adjustment while checking the speed responsiveness.

When you set or update the values of the natural frequency and attenuation coefficient for speed control, use the following variables of the st_speed_cfg_t structure (the structure for setting the parameters for controlling the speed control module). After you have set the desired values in these variables, apply them by using R_MOTOR_SPEED_ParameterUpdate (the API function for updating the parameters that control the speed control module).

- To set the natural frequency for the speed control system, use f4_speed_omega_hz. (See Table 5-40.)
- To set the attenuation coefficient for the speed control system, use f4_speed_zeta. (See Table 5-40.)

5.7.9.2 Setting the parameters for speed control

Because the speed control module uses the control interval and motor parameters, the control parameter configuration (R_MOTOR_SPEED_ParameterUpdate) can be used to update the parameters. For details about the items that can be set, see the description of the st_speed_cfg_t structure (structure for setting the parameters for controlling the speed control module).

5.7.9.3 Setting the initial values of the parameters for speed control

The configurations of the speed control module can be specified by using r_motor_module_cfg.h. The values set in this file are applied as initial values at system startup. For details about the items to be set, see 5.7.5.

5.8 Magnetic flux weakening control (speed control module)

The magnetic flux weakening control module is a submodule of the speed control module. When a motor that uses a magnet as the rotor rotates, an inductive voltage arises in proportion to the permanent magnet magnetic flux and rotation speed of the rotor. When the rotation speed increases and the inductive voltage becomes equal to the power supply voltage (that is, the voltage saturates), higher current can no longer flow into the motor, resulting in a saturated state that restricts any further increase in motor speed. Magnetic flux weakening control is a technology that solves this problem.

5.8.1 Description of the functionality

In magnetic flux weakening control, the d-axis current is applied in the negative direction to suppress the effect of voltage saturation due to induced voltage, thus enabling higher and more stable rotational speeds to be obtained.

In practice, the d-axis current is determined and controlled according to the formula shown in Figure5-22.

$$I_d = \frac{-\psi_a + \sqrt{\left(\frac{V_{om}}{\omega}\right)^2 - \left(L_q I_q\right)^2}}{L_d}$$

$$\because V_{om} = V_{amax} - I_a R$$

V_{om} : Inductive voltage limit value [V]

V_{amax} : Maximum voltage vector value [V]

I_a : Current vector magnification [A]

Figure5-22 Formula for calculating the d-axis command value in magnetic flux weakening control

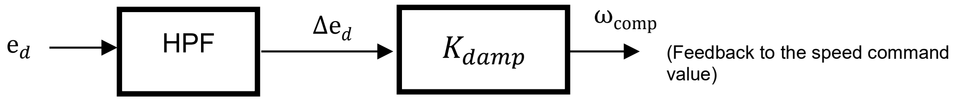
5.8.2 Adjustment and configuration of parameters

There are no parameters to be set by the user for this module. To use this module, use R_MOTOR_SPEED_ParameterUpdate (API function for updating the control parameters for the speed control module) to set "u1_flag_fluxwkn_use" (flag for whether to use magnetic flux weakening control) to 1.

5.9 Open-loop damping control (speed control module)

5.9.1 Description of the functionality

Normally, in sensorless vector control, the inductive voltage is lower in low speeds so that the accuracy of position and speed estimation does not increase. Therefore, a constant d-axis current is applied to generate a rotating magnetic field, and the motor is driven synchronously in speed open loop (open-loop control). At this point, the motor speed vibrates at the natural frequency that depends on the current and motor parameters. Therefore, implementing an open-loop damping control as shown in the block diagram in Figure5-23 can reduce the motor vibration and stabilize the motor speed even in low speeds.



e_d: Estimated d-axis inductive voltage value (output from the inductive voltage observer) [V]

$\Delta\mathrm{e}_d$: Perturbation component of the estimated d-axis inductive voltage value [V]

K_{damp}: Feedback gain to the speed command value

ω_{comp}: Feedback speed to the speed command value [rad/s (electrical angle)]

Figure5-23 Block diagram of open-loop damping control

5.9.2 Adjustment and configuration of parameters

There are no parameters to be set by the user for this module. To use this module, use R_MOTOR_SENSORLESS_VECTOR_ParameterUpdate (API function for updating the control parameters for the manager module) to set "u1_flag_openloop_damping_use" (flag for whether to use open-loop damping control) to 1.

5.10 Switching-to-sensorless control (speed control module)

5.10.1 Description of the functionality

After the motor starts up, the motor switches to sensorless control (speed closed loop control) when the speed reaches to a level sufficient to estimate the inductive voltage. However, because the motor in open-loop control is rotated by a torque generated by a phase error, hunting in the current and speed may occur when switching to sensorless control due to the high phase error, especially at higher loads. Therefore, estimate the load torque from the phase error Δθ, and then perform the switching-to-sensorless processing as shown in Figure5-24. This can reduce hunting in the current and speed when switching controls.

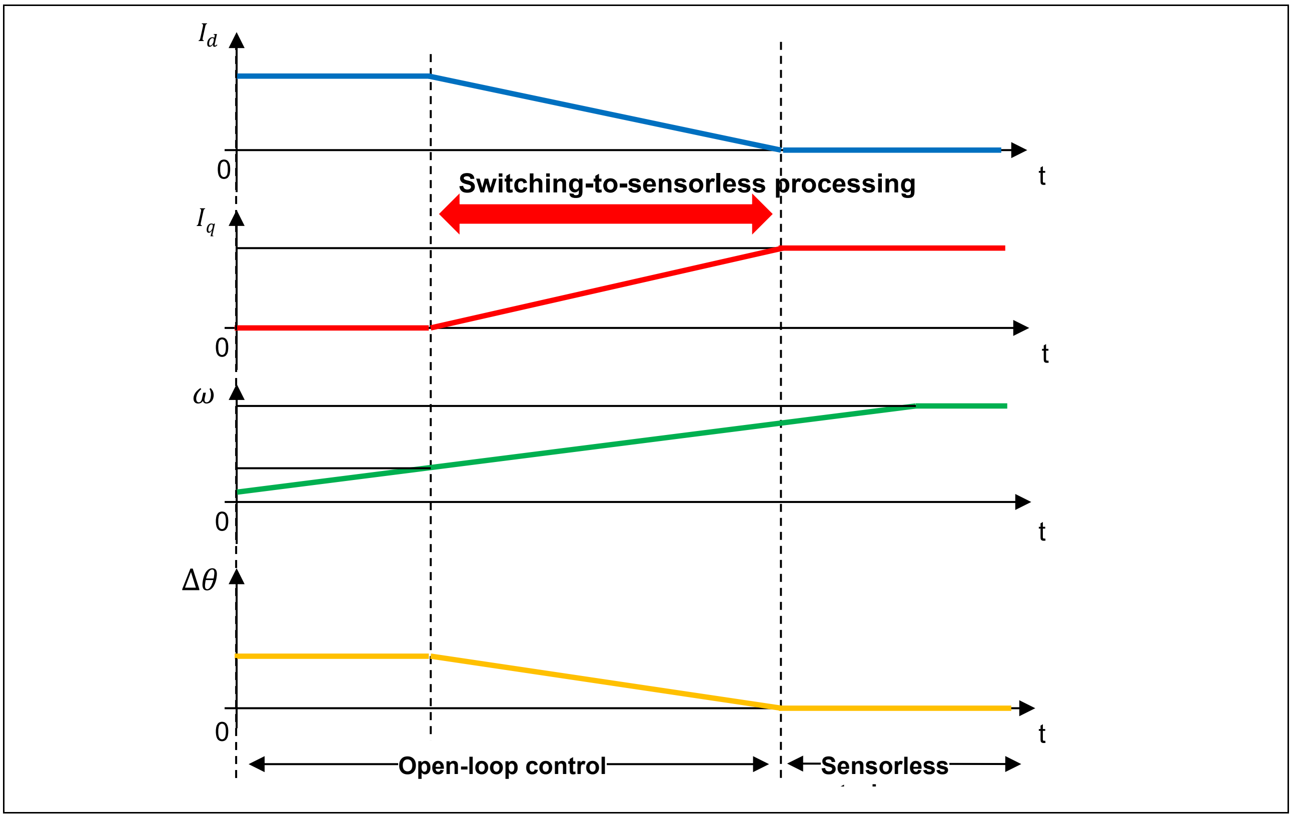


Figure5-24 Behavior diagram of each physical quantity during switching-to-sensorless processing

5.10.2 Adjustment and configuration of parameters

There are no parameters to be set by the user for this module. To use this module, use R_MOTOR_SENSORLESS_VECTOR_ParameterUpdate (API function for updating the control parameters for the manager module) to set "u1_flag_less_switch_use" (flag for whether to use switching-to-sensorless control) to 1.

5.11 Driver module

The driver module works as an interface between the manager module, which corresponds to the middleware of the sample software, and Smart Configurator, which is required to access the microcontroller peripherals. Configuring the driver module appropriately allows you to use microcontroller function allocation and the differentials of the board to be used without modifying the motor module.

5.11.1 Functionality

Table 5-43 lists the functions of the driver module.

Table 5-43 List of functions of the driver module

Function	Description
Acquisition of the A/D conversion value	Acquires AD values such as the phase current and inverter board bus voltage via a Smart Configurator function.
PWM duty setting	Sets the PWM duty value that is to be output to U-, V-, and W-phases via a Smart Configurator function.
PWM start/stop	Controls whether to start or stop PWM output via a Smart Configurator function.

5.11.2 Module configuration diagram

Figure5-25 Driver module configuration diagram shows the module configuration of the driver module.

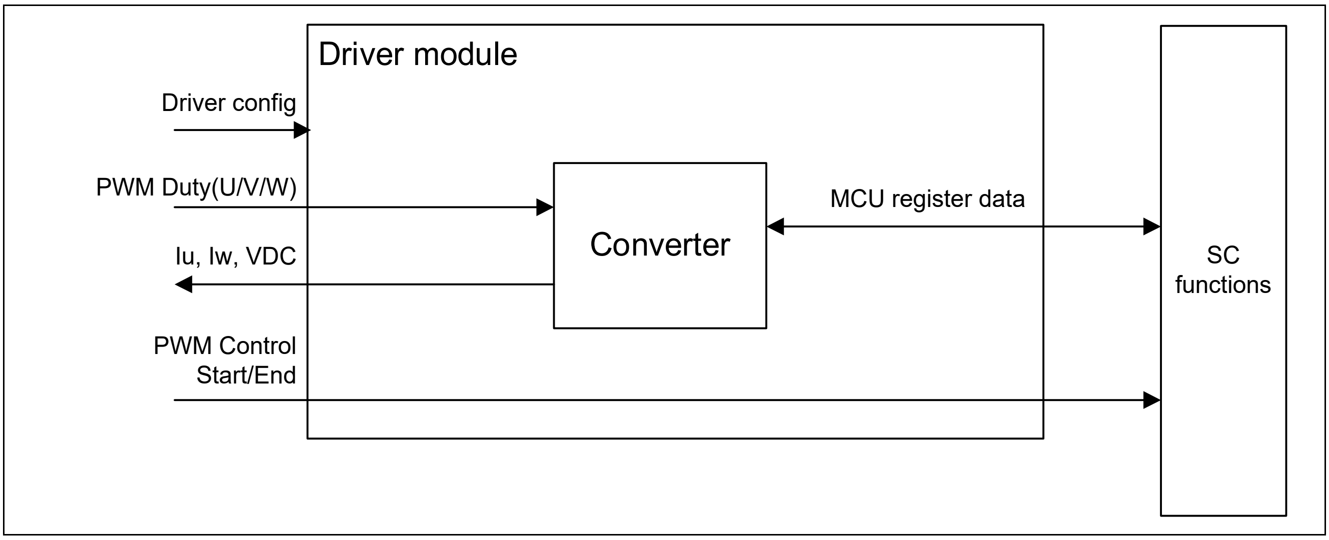


Figure5-25 Driver module configuration diagram

5.11.3 API

Table 5-44 lists and describes the API functions for the driver module.

Table 5-44 List of API functions

API	Description
R_MOTOR_DRIVER_Open	Generates an instance of the driver module.
R_MOTOR_DRIVER_Close	Places the module in a reset state.
R_MOTOR_DRIVER_ParameterUpdate	Inputs the variable information that is to be used inside the module.
R_MOTOR_DRIVER_BldcAnalogGet	Acquires the A/D conversion results.
R_MOTOR_DRIVER_BldcDutySet	Sets the PWM duty.
R_MOTOR_DRIVER_PWMControlStop	Stops PWM control.
R_MOTOR_DRIVER_PWMControlStart	Starts PWM control.

Note: Since the driver module API needs to control each motor (motor 1, motor 2), specify a structure variable pointer for the driver module (for motor 1: &g_st_driver_a / for motor 2: &g_st_driver_b) by argument.

5.11.4 Configurations

Table 5-45 lists the configurations for the driver module. Set up the functions to be used and the necessary parameters. shows the initial values.

Table 5-45 List of configurations

File name	Macro name	Description
r_motor_module_cfg.h	DRIVER_CFG_FUNC_PWM_OUTPUT_START_A	Sets the function that enables PWM output. (motor 1)
	DRIVER_CFG_FUNC_PWM_OUTPUT_STOP_A	Sets the function that disables PWM output. (motor 1)
	DRIVER_CFG_FUNC_ADC_DATA_GET_A	Sets the function that acquires the A/D conversion results. (motor 1)
	DRIVER_CFG_FUNC_DUTY_SET_A	Sets the function that sets the duty cycle. (motor 1)
	DRIVER_CFG_FUNC_PWM_OUTPUT_START_B	Sets the function that enables PWM output. (motor 2)
	DRIVER_CFG_FUNC_PWM_OUTPUT_STOP_B	Sets the function that disables PWM output. (motor 2)
	DRIVER_CFG_FUNC_ADC_DATA_GET_B	Sets the function that acquires the A/D conversion results. (motor 2)
	DRIVER_CFG_FUNC_DUTY_SET_B	Sets the function that sets the duty cycle. (motor 2)
r_motor_inverter_cfg.h	INVERTER_CFG_ADC_REF_VOLTAGE	Sets the reference voltage for A/D conversion.
r_motor_module_cfg.h	MOTOR_MCU_CFG_ADC_OFFSET	Sets the AD offset value.

Table 5-46 List of initial values for configurations

Macro name	Settings
DRIVER_CFG_FUNC_PWM_OUTPUT_START_A	R_Config_xxx_StartTimerCtrl (motor1) (Smart Configurator function) *1 *2
DRIVER_CFG_FUNC_PWM_OUTPUT_STOP_A	R_Config_xxx_StopTimerCtrl (motor1) (Smart Configurator function) *1 *2
DRIVER_CFG_FUNC_ADC_DATA_GET_A	R_Config_xxx_AdcGetConvVal (motor1) (Smart Configurator function) *1 *2
DRIVER_CFG_FUNC_DUTY_SET_A	R_Config_xxx_UpdDuty (motor1) (Smart Configurator function) *1 *2
DRIVER_CFG_FUNC_PWM_OUTPUT_START_B	R_Config_xxx_StartTimerCtrl (motor2) (Smart Configurator function) *1 *2
DRIVER_CFG_FUNC_PWM_OUTPUT_STOP_B	R_Config_xxx_StopTimerCtrl (motor2) (Smart Configurator function) *1 *2
DRIVER_CFG_FUNC_ADC_DATA_GET_B	R_Config_xxx_AdcGetConvVal (motor2) (Smart Configurator function) *1 *2
DRIVER_CFG_FUNC_DUTY_SET_B	R_Config_xxx_UpdDuty (motor2) (Smart Configurator function) *1 *2
INVERTER_CFG_ADC_REF_VOLTAGE	5.0f
MOTOR_MCU_CFG_ADC_OFFSET	0x7FF

Notes: 1. For details about the functions shown in the " Settings " column, see 5.12 Smart Configurator setting.

2. If you want to use the motor component of the Smart Configurator, "xxx" is set to "MOTOR". If you do not use a motor component, set to the module name to be used for PWM.

5.11.5 Structure and variable information

Table 5-47 lists the structures that are used for the driver module. In the driver module, the structure for the driver module (g_st_driver) is defined by securing an instance of the module from the API.

Table 5-47 List of structures and variables

Structure	Variable	Description
st_motor_driver_t Structure for the driver module	*ADCDataGet	Pointer to the Smart Configurator function (This variable sets the function that acquires the results of A/D conversion.)
	*BLDCDutySet	Pointer to the Smart Configurator function (This variable sets the function that enables PWM output.)
	*PWMOutputStop	Pointer to the Smart Configurator function (This variable sets the function that disables PWM output.)
	*PWMOutputStart	Pointer to the Smart Configurator function (This variable sets the function that sets the duty cycle.)
	f4_ad_crnt_per_digit	Scale for A/D conversion of the current
	f4_ad_vdc_per_digit	Scale for A/D conversion of the voltage
	f4_pwm_period_cnt	Count value for one interval of the PWM counter (information for the duty setting)
	f4_pwm_dead_time_cnt	Count value for the dead time (information for the duty setting)
	*p_st_driver_init	Instance for the initial value of the st_motor_driver_t
	*p_st_cfg_init	Instance for the initial value of the st_drv_cfg_t
st_motor_driver_cfg_t Structure for setting the parameters for controlling the drive module	*ADCDataGet	Pointer to the Smart Configurator function
	*BLDCDutySet	Pointer to the Smart Configurator function
	*PWMOutputStop	Pointer to the Smart Configurator function
	*PWMOutputStart	Pointer to the Smart Configurator function
	f4_shunt_ohm	Shunt resistance value [ohm] (for calculation of f4_ad_crnt_per_digit)
	f4_volt_gain	Voltage conversion gain coefficient (for calculation of f4_ad_vdc_per_digit)
	f4_crnt_amp_gain	Current conversion gain coefficient (for calculation of f4_ad_crnt_per_digit)
	f4_pwm_period_cnt	Count value for one interval of the PWM counter (information for the duty setting)
	f4_pwm_dead_time_cnt	Count value for the dead time (information for the duty setting)
st_drv_cfg_t Structure for driver module control initial value	st_drv_init_cfg	Structure for initial values of driver module control parameters

5.11.6 Macro definition

Table 5-48 lists the macros for the driver module.

Table 5-48 List of macros

File name	Macro name	Defined value	Remarks
r_motor_driver.c	MOTOR_DRIVER_PRV_ADC_REF_VOLTAGE	INVERTER_CFG_ADC_REF_VOLTAGE (See Table 5-45.)	Reference voltage [V]

5.11.7 Variable information for the driver module

Table 5-49 lists the driver module variable information.

Table 5-49 List of variable information of the current control module

Structure name	Variable name	Remarks
st_speed_control_t Structure for Driver Module	g_st_driver_a	Variable for motor1
	g_st_driver_a_init	Variable for initial value of motor1
	g_st_driver_b	Variable for motor2
	g_st_driver_b_init	Variable for initial value of motor2
st_motor_driver_cfg_t	st_driver_cfg	(Local variables for function internals)
st_drv_cfg_t Structure for initial values of driver module control parameters	g_st_driver_a_cfg	Variable for motor1
	g_st_driver_b_cfg	Variable for motor2

5.11.8 Adjustment and configuration of parameters

5.11.8.1 Setting the parameters for controlling the driver module

In the driver module, parameters that are input from the control parameter configuration (R_MOTOR_DRIVER_ParameterUpdate) are used to associate the motor module and Smart Configurator and to convert data. The parameters are input by using st_speed_cfg_t (the structure for setting the parameters for controlling the drive module). In the sample program, the information that is defined as configurations is used as the parameter settings. Table 5-50 shows the settings.

Table 5-50 Example of settings specified in the sample program (motor 1)

Variable name	Macro name	File name
*ADCDataGet	DRIVER_CFG_FUNC_ADC_DATA_GET_A	See Table 5-45.
*BLDCDutySet	DRIVER_CFG_FUNC_DUTY_SET_A	
*PWMOutputStop	DRIVER_CFG_FUNC_PWM_OUTPUT_START_A	
*PWMOutputStart	DRIVER_CFG_FUNC_PWM_OUTPUT_STOP_A	
f4_shunt_ohm	INVERTER_CFG_SHUNT_RESIST	r_motor_inverter_cfg.h
f4_volt_gain	INVERTER_CFG_VOLTAGE_GAIN	
f4_crnt_amp_gain	INVERTER_CFG_CURRENT_AMP_GAIN	
f4_pwm_period_cnt	MOTOR_COMMON_CARRIER_SET_BASE_A	r_motor_module_cfg.h
f4_pwm_dead_time_cnt	MOTOR_COMMON_DEADTIME_SET_A	

Table 5-51 Example of settings specified in the sample program (motor2)

Variable name	Macro name	File name
*ADCDataGet	DRIVER_CFG_FUNC_ADC_DATA_GET_B	See Table 5-45.
*BLDCDutySet	DRIVER_CFG_FUNC_DUTY_SET_B	
*PWMOutputStop	DRIVER_CFG_FUNC_PWM_OUTPUT_START_B	
*PWMOutputStart	DRIVER_CFG_FUNC_PWM_OUTPUT_STOP_B	
f4_shunt_ohm	INVERTER_CFG_SHUNT_RESIST	r_motor_inverter_cfg.h
f4_volt_gain	INVERTER_CFG_VOLTAGE_GAIN	
f4_crnt_amp_gain	INVERTER_CFG_CURRENT_AMP_GAIN	
f4_pwm_period_cnt	MOTOR_COMMON_CARRIER_SET_BASE_B	r_motor_module_cfg.h
f4_pwm_dead_time_cnt	MOTOR_COMMON_DEADTIME_SET_B	

5.12 Smart Configurator setting

In the sample program, Smart Configurator is used to create a project. This section describes the components used and the functions added to the user area.

5.12.1 Clock settings

Table 5-52 shows the clock settings.

Table 5-52 MCU clock settings

Clock	Frequency
Main clock	10MHz
System clock (ICLK)	120MHz
Peripheral module clock (PCLKA)	120MHz
Peripheral module clocks (PCLKB/PCLKC/PCLKD)	60MHz/120MHz/60MHz
Flash IF clock (FCLK)	60MHz
IWDTCLK	120kHz

5.12.2 Component settings

Table 5-53 and lists the components used and the functions allocated to the components.

Table 5-53 Smart Configurator components and their functions

Function	Component
3-phase PWM output, A/D conversion (current detection, inverter bus voltage detection, motor 1)	Config_MOTOR_A
3-phase PWM output, A/D conversion (current detection, inverter bus voltage detection, motor 2)	Config_MOTOR_B
A/D conversion processing (command voltage detection for the board UI)	Config_S12AD2
Setting of the port to be used	Config_PORT
Speed control interrupt timer	Config_CMT0
Independent watchdog timer	Config_IWDT
Overcurrent detection	Config_POE

5.12.3 Interrupts

Table 5-54 shows the information about the interrupts.

Table 5-54 List of interrupts

Interrupt	Peripheral	Interrupt Level *1	Description	Cycle
A/D conversion end	S12AD	12	Invoke current-controlled periodic interrupt processing.	50μs
Speed control cycle	CMT	11	Invoke speed control processing.	500μs
Hardware overcurrent	POEG	15	Invoke Hardware overcurrent abnormality processing.	-

Note 1: Processor Interrupt Priority Level: highest 15 <--> 0 lowest.

5.12.3.1 A/D conversion end interrupt

An interrupt is generated when the A/D conversion is completed, and the current-controlled interrupt process is executed. From the capture of the current value, the output PWM is calculated. In addition, error determination and communication processing with RMW are performed.

5.12.3.2 Speed control interrupt

It is created with the CMT0 component and executes processing at 500μs cycles.

Velocity PI calculation and weakening flux processing are performed.

5.12.3.3 Hardware overcurrent error interrupt

When an overcurrent abnormality occurs on the inverter board, overcurrent abnormality processing is executed by an external interrupt.

5.12.4 Details of user codes

Table 5-55 lists the functions that are created in the user code area.

Table 5-55 List of functions in the user area

Component	Function	Description
Config_PORT	R_Config_PORT_GetSW1	Acquires the status of SW1-1 or SW1-2.
	R_Config_PORT_GetSW2	Acquires the status of SW2-1 or SW2-2.
	R_Config_PORT_Led1_on	Turns on LED1-1 or LED1-2.
	R_Config_PORT_Led2_on	Turns on LED2-1 or LED2-2.
	R_Config_PORT_Led1_off	Turns off LED1-1 or LED1-2.
	R_Config_PORT_Led2_off	Turns off LED2-1 or LED2-2.

5.12.5 Pin settings

For pin interface information, refer to 2.2.3 Peripheral Functions.

5.12.6 Macro definitions

Table 5-56 and Table 5-57 lists the macros for the motor component.

Table 5-56 List of macros (motor1)

File name	Macro name	Defined value	Remarks
Config_MOTOR_A.h	CG_CONFIG_MOTOR_A_PWM_TIMER_FREQ	120.0f	PWM timer frequency [MHz] (motor 1)
	CG_CONFIG_MOTOR_A_CARRIER_FREQ	20.000f	Carrier wave actual frequency [kHz] (motor 1)
	CG_CONFIG_MOTOR_A_DEADTIME	2.000f	Deadtime actual value [μs] (motor 1)
	CG_CONFIG_MOTOR_A_INTR_DECIMATION	1.0	Interrupt skipping number (motor 1)
	CG_MOTOR_CFG_MAX_AD_DATA	4095.0f	Max A/D data (motor 1)
	CG_MOTOR_MCU_CFG_AD_FREQ	60.0f	A/D frequency [MHz] (motor 1)

Table 5-57 List of macros (motor2)

File name	Macro name	Defined value	Remarks
Config_MOTOR_B.h	CG_CONFIG_MOTOR_B_PWM_TIMER_FREQ	120.0f	PWM timer frequency [MHz] (motor 2)
	CG_CONFIG_MOTOR_B_CARRIER_FREQ	20.000f	Carrier wave actual frequency [kHz] (motor 2)
	CG_CONFIG_MOTOR_B_DEADTIME	2.000f	Deadtime actual value [μs] (motor 2)
	CG_CONFIG_MOTOR_B_INTR_DECIMATION	1.0	Interrupt skipping number (motor 2)
	CG_MOTOR_CFG_MAX_AD_DATA	4095.0f	Max A/D data (motor 2)
	CG_MOTOR_MCU_CFG_AD_FREQ	60.0f	A/D frequency [MHz] (motor 2)

6. Vector control algorithm

6.1 Analysis model of a permanent magnet synchronous motor

The voltage equation of a permanent magnet synchronous motor that has sinusoidal magnetic flux distribution as shown in Figure6-1 can be represented as below.

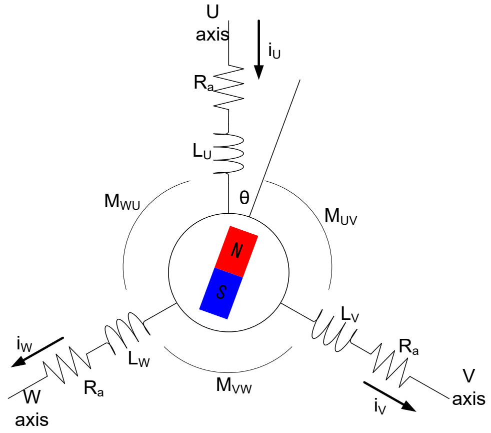


Figure6-1 Conceptual diagram for a 3-phase permanent magnet synchronous motor

$$\begin{bmatrix} v_u \\ v_v \\ v_w \end{bmatrix} = R_a \begin{bmatrix} i_u \\ i_v \\ i_w \end{bmatrix} + p \begin{bmatrix} \phi_u \\ \phi_v \\ \phi_w \end{bmatrix}$$

$$\begin{bmatrix} \phi_u \\ \phi_v \\ \phi_w \end{bmatrix} = \begin{bmatrix} L_u & M_{uv} & M_{wu} \\ M_{uv} & L_v & M_{vw} \\ M_{wu} & M_{vw} & L_w \end{bmatrix} \begin{bmatrix} i_u \\ i_v \\ i_w \end{bmatrix} + \psi \begin{bmatrix} cos\theta \\ \cos{(\theta - 2\pi/3)} \\ \cos{(\theta + 2\pi/3)} \end{bmatrix}$$

v_u, v_v, v_w: Armature voltages of each phase

i_u, i_v, i_w: Armature currents of each phase

ϕ_u, ϕ_v, ϕ_w: Armature interlinkage magnetic flux of each phase

R_a: Armature resistance of each phase

p: Differential operator

L_u, L_v, L_w: Self-inductances of each phase

M_{uv}, M_{vw}, M_{wu}: Mutual inductance among each phase

ψ: Maximum value of the armature interlinkage magnetic flux with a permanent magnet

θ: Lead angle of a permanent magnet (rotor) from the U phase

6.2 d-q axis model of a permanent magnet synchronous motor

In vector control, the AC 3-phase (u, v, and w) coordinate system is represented as the DC two-phase (d and q) coordinate system. The 3-phase winding wire of the stator is transformed into a two-phase winding wire that rotates in synchronization with the rotor of the permanent magnet. Therefore, it can be treated as two electrically independent DC circuits that are relatively stationary.

In the two-phase (d and q) coordinate system, the d axis is set in the direction of the magnetic flux (to the north pole) of the permanent magnet of the rotor, and the q axis is set at the lead angle (θ) of 90 degrees in the positive direction from the d axis. The following conversion matrix is used to obtain the voltage equation of the permanent magnet synchronous motor seen from the d-q coordinate system.

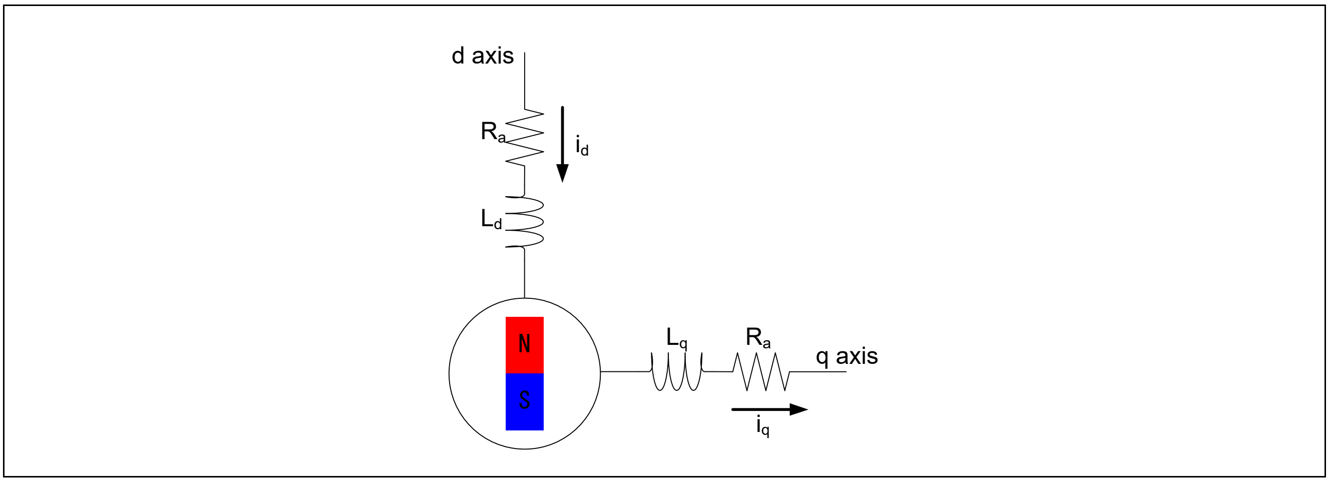


Figure6-2 Conceptual diagram for a two-phase DC motor

$$C = \sqrt{\frac{2}{3}}\begin{bmatrix} cos\theta & \cos(\theta - 2\pi/3) & \cos(\theta + 2\pi/3) \\ -sin\theta & -\sin(\theta - 2\pi/3) & -\sin(\theta + 2\pi/3) \end{bmatrix}$$

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = C \begin{bmatrix} v_u \\ v_v \\ v_w \end{bmatrix}$$

From the above shown coordinate transformation, the voltage equation in the d-q coordinate system can be represented as follows:

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R_a + pL_d & -\omega L_q \\ \omega L_d & R_a + pL_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \psi_a \end{bmatrix}$$

v_d, v_q: Armature voltages of the d and q axes

i_d, i_q: Armature currents of the d and q axes

R_a: Armature resistance of each phase

ω: Angular speed

L_d, L_q: Self-inductances of the d and q axes

$$L_d = l_a + \frac{3(L_a - L_{as})}{2}, L_q = l_a + \frac{3(L_a + L_{as})}{2}$$

ψ_a: Effective value of the armature interlinkage magnetic flux with a permanent magnet

$$\psi_\mathrm{a} = \sqrt{\frac{3}{2}}\psi$$

As above, an alternating current that flows on a stationary two-phase stator can be assumed to be a direct current that flows on a two-phase stator that rotates in synchronization with the permanent magnet of the rotor.

The magnitude of torque generated in the motor is obtained from the cross product of the current vector and armature interlinkage magnetic flux as follows. In the following expression, the first term on the right side is called "magnet torque" and the second term on the right side is called "reluctance torque".

$$T = P_n\{\psi_a i_q + (L_d - L_q) i_d i_q\}$$

T: Motor torque, P_n: Number of pole pairs

A motor having no difference in inductance between the d and q axes is called a "motor without saliency". In this case, the torque increases in proportion to the q-axis current because the reluctance torque is 0. Therefore, the q-axis current is sometimes called the "torque current". On the other hand, the d-axis current is sometimes called the excitation current due to the varying magnitude of magnetic flux of the permanent magnet according to d-axis current.

6.3 Vector control system and controller

The following shows the example of the block diagram for the overall speed control system.

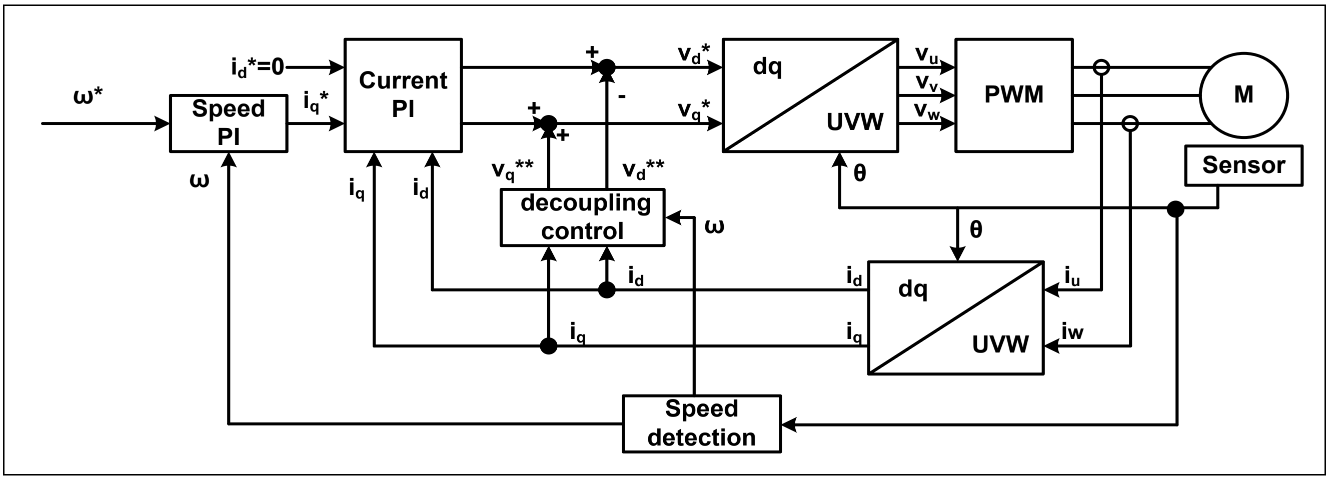


Figure6-3 System block of vector control (example)

As shown in Figure6-3, the speed control system consists of the speed control system and current control system. These systems are implemented by using a general PI controller, and the gain of each controller must be designed appropriately in order to achieve the desired control characteristics.

Also, the non-interacting control (decoupling control) in the system block feeds the inductive voltages (v_d^{**} and v_q^{**}), which are generated when the motor rotates, forward to the command voltage of each phase. This achieves a highly responsive speed control system and enables independent control of the d and q axes.

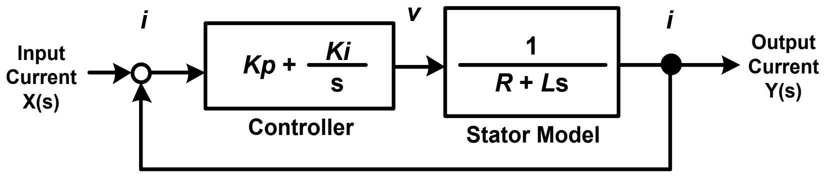
$$v_d^{**} = -\omega L_d i_q$$

$$v_q^{**} = \omega(L_d i_d + \psi_a)$$

6.3.1 Design of the current control system

In this section, you model the current control system from a viewpoint of the electrical characteristics of the motor. A stator coil can be represented with the resistance R and inductance L. Therefore, the stator model of the motor can be represented as $\frac{1}{R+Ls}$, the transfer function of a general RL series circuit.

The controller can be represented using PI control. The current control system can be represented as a feedback control system like the one shown in Figure6-4.



R: Resistance of the stator coil $[\Omega]$
L: Inductance of the stator coil $[\mathrm{H}]$
K_p: Proportional gain of current PI control
K_i: Integral gain of current PI control

Figure6-4 Model of the current control system

First, design the PI control gains of the current control system on the assumption that the values of R and L for the motor stator are known.

The closed-loop transfer function of the current control system can be obtained as follows:

$$G(s) = \frac{Y(s)}{X(s)} = \frac{\frac{K_a}{K_b}\left(1+\frac{s}{a}\right)}{s^2 + \frac{1}{K_b}(1+\frac{K_a}{a})s + \frac{K_a}{K_b}}$$

$$K_i = K_p a, \quad K_a = \frac{K_p a}{R}, \quad K_b = \frac{L}{R}$$

Also, the general expression of a second-order system having the zero point can be represented as follows:

$$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\left(1+\frac{s}{\omega_z}\right)$$

Additionally, if you perform coefficient comparison between the transfer function of the current control system and the second-order system having the zero point, the relationships shown in the following expressions can be obtained:

$$\frac{\omega_n^2\left(1+\frac{s}{\omega_z}\right)}{s^2+2\zeta\omega_n s+\omega_n^2} \Leftrightarrow \frac{\frac{K_a}{K_b}\left(1+\frac{s}{a}\right)}{s^2+\frac{1}{K_b}(1+\frac{K_a}{a})s+\frac{K_a}{K_b}}$$

$$\omega_n^2=\frac{K_a}{K_b}, \qquad 2\zeta\omega_n=\frac{1}{K_b}\left(1+\frac{K_a}{a}\right), \qquad \omega_z=a$$

From the above, the natural frequency ω_n, attenuation coefficient ζ, and zero-point frequency ω_z can be represented as follows:

$$\omega_n=\sqrt{\frac{K_a}{K_b}}, \qquad \zeta=\frac{1}{2K_b\sqrt{\frac{K_a}{K_b}}}(1+\frac{K_a}{a}), \qquad \omega_z=a=\frac{\omega_n^2 L}{2\zeta\omega_n L-R}$$

From this, the current PI control gains $K_{p\_current}$ and $K_{i\_current}$ can be represented by the following expressions:

$$K_{p\_current}=2\zeta_{CG}\omega_{CG}L-R, \qquad K_{i\_current}=K_{p\_current}a=\omega_{CG}^2 L$$

ω_{CG}: Natural frequency for the current control system

ζ_{CG}: Attenuation coefficient for the current control system

Therefore, it is found that the PI control gains of the current control system can be designed with ω_{CG} and ζ_{CG}.

6.3.2 Design of the speed control system

In this section, you model the speed control system from a viewpoint of the electrical characteristics of the motor. From the motion equation of the rotation system, the torque expression of the mechanical system can be represented by the following expression:

$$T = J\dot{\omega}_{mech}$$

J: Rotor inertia, ω_{mech}: Mechanical angular speed

On the other hand, the torque expression of the electrical system can be represented by the following expression if only the magnet torque is considered:

$$T = P_n\psi_a i_q$$

By using the above two torque expressions (of the mechanical and electrical systems), the mechanical angular speed can be represented by the following expression:

$$\omega_{mech} = \frac{P_n\psi_a}{sJ} i_q$$

ω_{mech}: Mechanical angular speed

Therefore, this is the motor model in the speed control system. Also, the controller can be represented using PI control. The speed control system can be represented as a feedback control system like the one shown in Figure6-5.

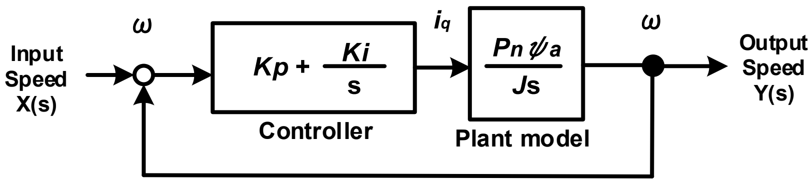


Figure6-5 Model of the speed control system

Here, you set the PI control gains of the speed control system on the assumption that the motor parameter P_n, ψ, J is known. First, obtain the transfer function of the system.

The closed-loop transfer function of the speed control system can be obtained as follows:

$$G(s) = \frac{Y(s)}{X(s)} = \frac{K_b a\left(1 + \frac{s}{a}\right)}{s^2 + K_b s + K_b a}$$

$$K_b = \frac{K_P P_n \psi}{J}, \quad K_i = K_p a$$

Also, the general expression of a second-order system having the zero point can be represented as follows:

$$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\left(1 + \frac{s}{\omega_z}\right)$$

In the same way as for the current control system, if you perform coefficient comparison between the transfer function of the speed control system and the second-order system having the zero point, the relationships shown in the following expressions can be obtained:

$$\frac{\omega_n^2(1 + s/\omega_z)}{s^2 + 2\zeta\omega_n s + \omega_n^2} \Leftrightarrow \frac{aK_b\left(1 + \frac{s}{a}\right)}{s^2 + K_b s + aK_b}$$

$$\omega_n^2 = aK_b = \frac{K_p a P_n \psi_a}{J}, \qquad 2\zeta\omega_n = K_b = \frac{K_p P_n \psi_a}{J}, \qquad \omega_z = a$$

From the above, the natural frequency ω_n, attenuation coefficient ζ, and zero-point frequency ω_z can be represented as follows:

$$\omega_n = \sqrt{\frac{K_p a P_n \psi_a}{J}}, \qquad \zeta = \frac{1}{2}\sqrt{\frac{K_p P_n \psi_a}{aJ}}, \qquad \omega_z = a = \frac{\omega_n}{2\zeta}$$

From this, the PI control gains K_{p_speed}, K_{i_speed} can be represented by the following expressions:

$$K_{p\_speed} = \frac{2\zeta_{SG}\omega_{SG}J}{P_n\psi_a}, \qquad K_{i\_speed} = K_{p\_speed} * a = \frac{\omega_{SG}^2 J}{P_n\psi_a}$$

ω_{SG}: Natural frequency for the speed control system

ζ_{SG}: Attenuation coefficient for the speed control system

Therefore, it is found that the PI control gains of the speed control system can be designed with ω_{SG} and ζ_{SG}.

6.4 Control block diagram

Figure6-6 shows the control block diagram of sensorless vector control that uses the inductive voltage observer during open-loop control.

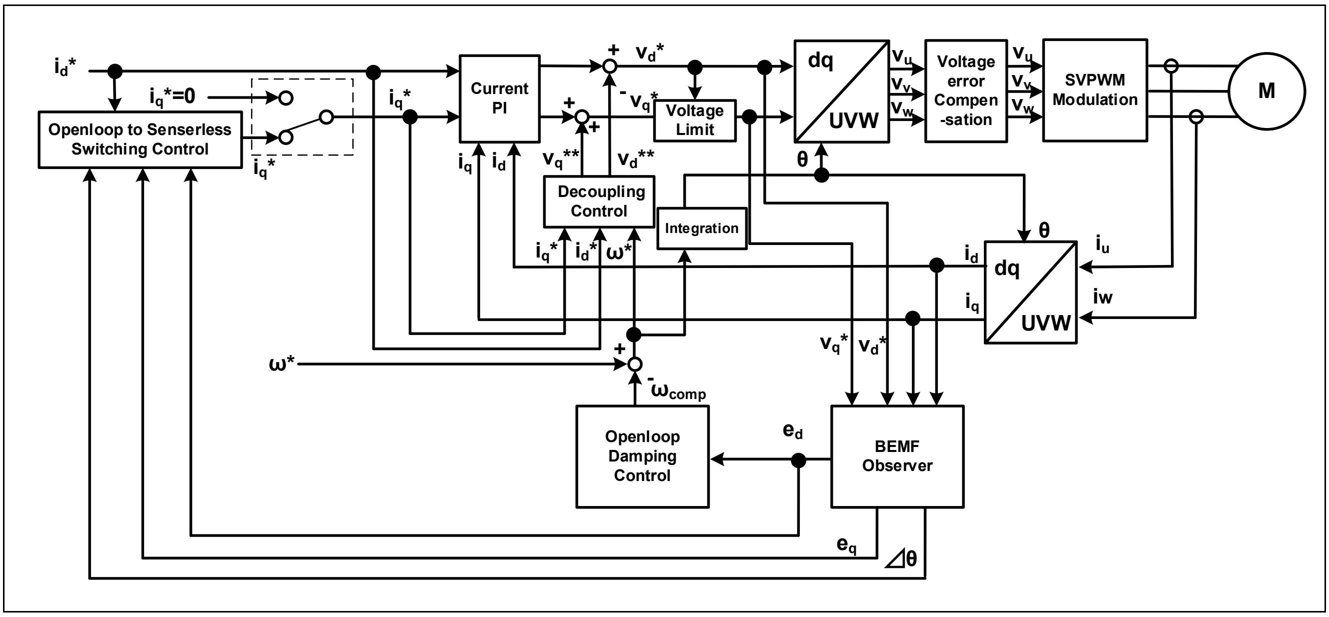


Figure6-6 Control block diagram of sensorless vector control (during open-loop control)

Figure6-7 shows the control block diagram of sensorless vector control that uses the inductive voltage observer during sensorless control (speed closed-loop control).

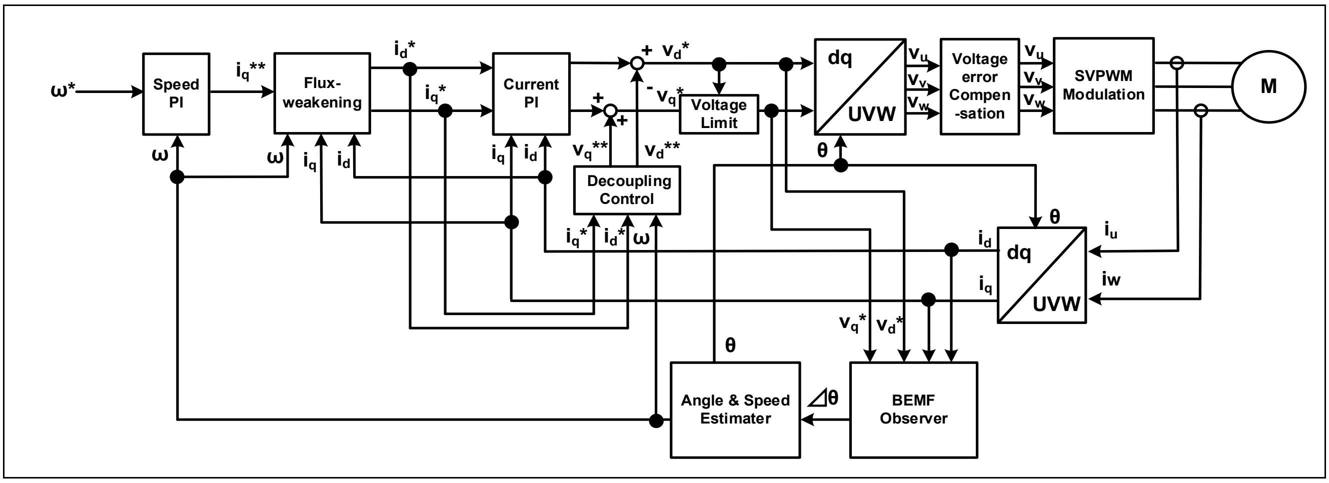


Figure6-7 Control block diagram of sensorless vector control (during sensorless control)

7. Test results

The test results shown in this chapter are reference values measured in the 2.1 Operation check environment.

(1)Program size

 and shows the size of the sample program. In the optimization settings of the compiler, the optimization level is set to 2 (-optimize = 2) and the optimization method is set to the one that is code-size oriented (-size).

Table 7-1 Program size

Memory	Size
ROM	30,035 Byte
RAM	13,627 Byte
Maximum value of stack analysis result	232 Byte
Stack size setting value	5120 Byte

(2)CPU loading rate Table 7-2 and shows the CPU processing time and loading rate for each control interval.

Table 7-2 Control loop and CPU loading rate

CPU Board	Control loop type	Control interval	Processing time	CPU loading rate
RX26T RAM64KB Version	Current control loop	50 μs (no decimation)	15.1 μs	30.2%
	Speed control loop	500 μs	19.4 μs	3.88%

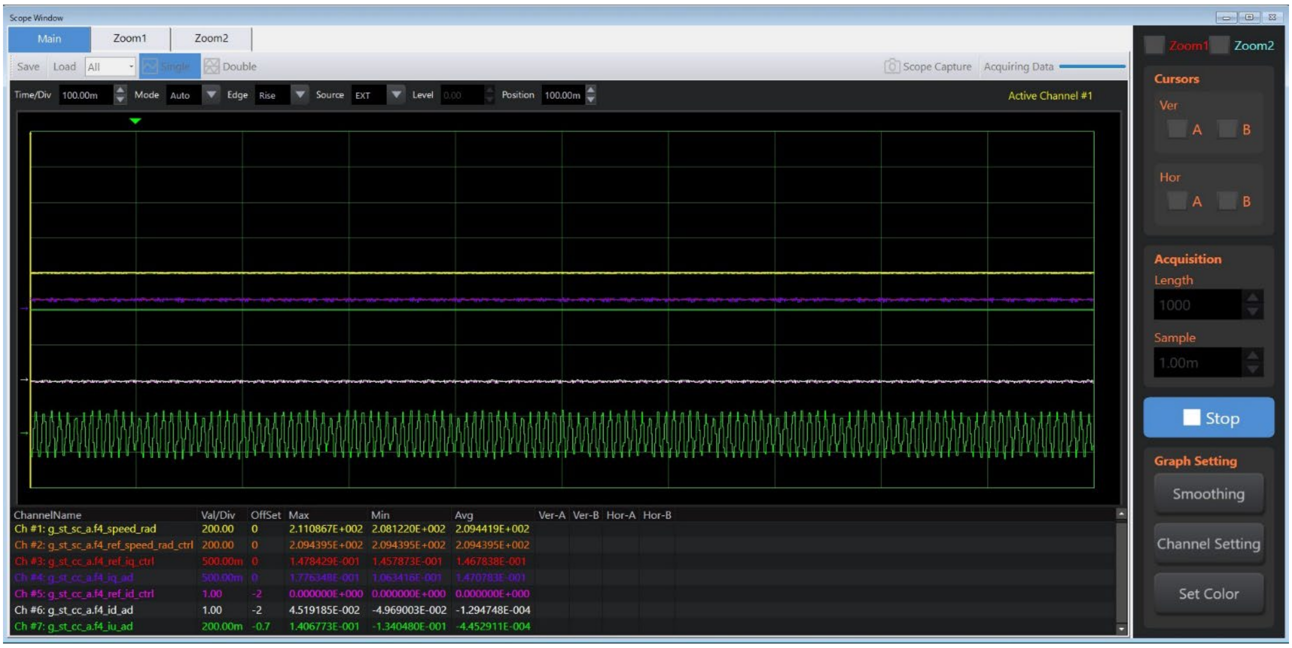
(3)Operation waveforms

For reference, a waveform observed during control using the sample program is provided from the results of tests using sensors. The waveform shown is the one seen when the motor rotates at a steady speed of 2,000 rpm during speed control.

Table 7-3 Measurement conditions

Item	Value	Remarks
Frequency for the current control system	300 [Hz]	
Attenuation coefficient for the current control system	1	
Frequency for the speed control system	3 [Hz]	
Attenuation coefficient for the speed control system	1	
Speed	2000 [rpm]	
Load	—	Conducted at no load.

Figure7-1 shows the results of testing speed control.



Time/Div 100.00m Mode Auto Edge Rise Source EXT Level 0.00 Position 100.00m Active Channel #1

Length 1000
Sample 1.00m

ChannelName	Val/Div	OffSet	Max	Min	Avg	Ver-A	Ver-B	Hor-A	Hor-B
Ch #1: g_st_sc_a.f4_speed_rad	200.00	0	2.110867E+002	2.081220E+002	2.094419E+002				
Ch #2: g_st_sc_a.f4_ref_speed_rad_ctrl	200.00	0	2.094395E+002	2.094395E+002	2.094395E+002				
Ch #3: g_st_cc_a.f4_ref_iq_ctrl	500.00m	0	1.478429E-001	1.457873E-001	1.467838E-001				
Ch #4: g_st_cc_a.f4_iq_ad	500.00m	0	1.776348E-001	1.063416E-001	1.470783E-001				
Ch #5: g_st_cc_a.f4_ref_id_ctrl	1.00	-2	0.000000E+000	0.000000E+000	0.000000E+000				
Ch #6: g_st_cc_a.f4_id_ad	1.00	-2	4.519185E-002	-4.969003E-002	-1.294748E-004				
Ch #7: g_st_cc_a.f4_iu_ad	200.00m	-0.7	1.406773E-001	-1.340480E-001	-4.452911E-004				

Figure7-1 Speed control using sensorless vector control

Drive condition:

Rotation speed: Speed command 2000 [rpm]

Waveform information:

Yellow: Detected speed [rad/s] (200 rad/s / div.), Orange: Instructed speed [rad/s] (200 rad/s / div.)
Red: q-axis current command value [A] (500 mA / div.), Purple: q-axis current value [A] (500 mA / div.)
Pink: d-axis current command value [A] (1 A / div.), White: d-axis current value [A] (1 A / div.)
Light green: U-phase current [A] (200 mA / div.)
Horizontal axis: 100 ms / div.

8. Reference materials

- Renesas Motor Workbench User's Guide (R21UZ0004)
- MCK-RX26T User's Manual (R12UZ0111)
- Smart Configurator User's Manual -- RX API Reference (R20UT4360)
- RX Smart Configurator User Guide -- CS+ (R20AN0470)
- RX Smart Configurator User Guide -- e^2 studio (R20AN0451)
- RX26T Group User's Manual: Hardware (R01UH0979)
- MCB-RX26T Type A User's Manual(R12UZ0112)

Revision History

Rev.	Date of issue	Amendments	
		Page	Point
1.00	Apr.12.24	—	First edition issued

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

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

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

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

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

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

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

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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