Hall sensor vector control of PM motors

RX Family

Introduction

This sample program provides a vector control (Hall Field Oriented Control, FOC) function that uses a three-phase permanent magnet synchronous motor (PM motor) equipped with a Hall sensor as a position, speed and rotary direction detector over the whole speed range from standstill to low, middle, and high speed. Hall sensor can be viewed as a low-resolution encoder.

Sensor vector control is realized by generating six pulses per one rotation of the electrical angle from the three signals output from the Hall sensor and performing linear interpolation using calculus to obtain the magnetic pole position. In addition, since the integral term of the speed regulator is susceptible to saturation, which is a phenomenon peculiar to low-resolution encoders in the speed control mode, friction compensation can be introduced to improve characteristics such as the characteristics of starting and low-speed range.

In the conventional sample program, the magnetic pole position of the Hall sensor is detected only in a standstill, and current draw control by open loop is provided in the low-speed range, and sensorless vector control by the BEMF observer is provided in the middle to high-speed range. In this sample program, additional functions have been implemented to improve torque output performance and the starting characteristics from a standstill.

This document explains a sample program for 24 VDC inverters and PM motors using a RX26T microcontroller.

Figure 1-1 shows the waveform of characteristics of acceleration/deceleration and forward/reverse rotation when operated with this sample program

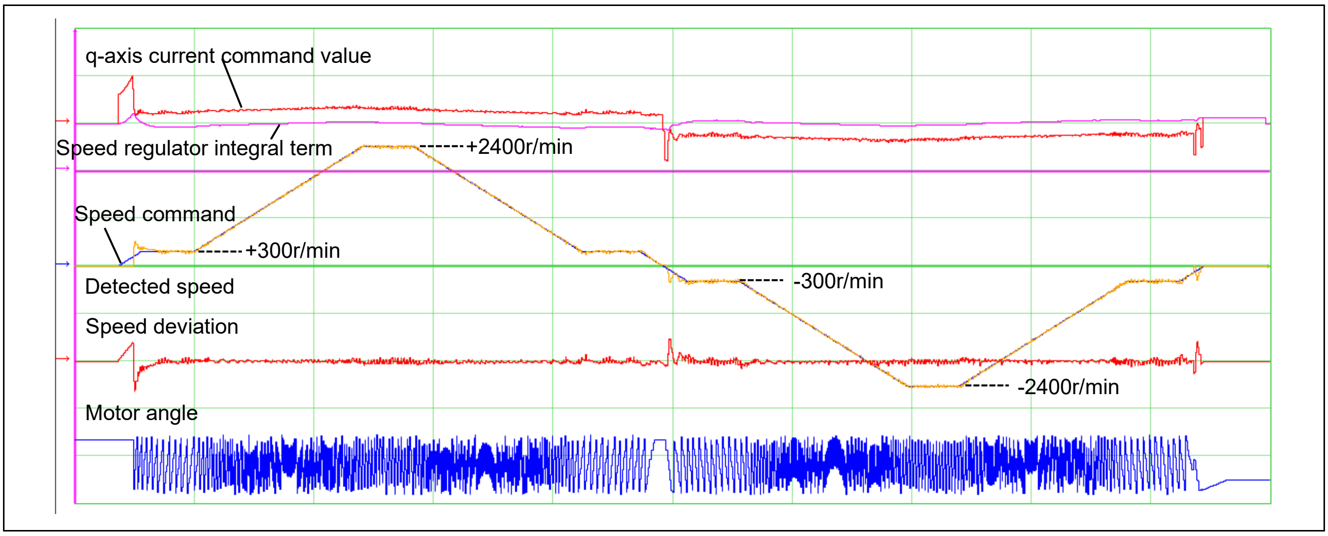


Figure 1-1 Acceleration/deceleration and forward/reverse rotation characteristics

Target device

Operations of the target software of this application were checked by using the following devices.

- RX26T RAM64KB Version (R5F526TFCDFP)

Table of contents

1. Overview

This application note is intended to explain the method used by the sample program that uses RX26T a microcontroller (MCU) manufactured by Renesas, to drive a permanent magnet synchronous motor by vector control using Hall sensors over the whole speed range, including from a standstill and low speeds.

The sample program uses Renesas' motor control kit MCK-RX26T to control the MOONS' R42BLD30L3 (24 VDC PM motor) by vector control using Hall sensors. In addition, this sample program supports Renesas Motor Workbench, a motor control development support tool, which can be used as a user interface (UI) for checking the MCU internal data and controlling motors. You can use the sample program for reference purposes when selecting the MCU to be used or developing software by checking how MCU functions are allocated, how control is loaded by interrupts, and other information in the sample program.

The sample program described in this application note was developed and evaluated in the PM motor and inverter environment as described in this document and is not guaranteed to work in your PM motor or inverter environment. Control performance may be limited by factors such as mounting errors of the current sensor or Hall sensor; PCB design pattern of the signal path; specifications of sampling/resolution/filter; magnetic saturation characteristics of the motor and variations between individual motors. Under the responsibility of the user, algorithm refinement and parameter tuning are required.

Note that the tools and devices described in this application note may not be available due to discontinuation or modification by the respective manufacturers.

Main hardware and devices used for evaluation
- Motor control kit: MCK-RX26T (Type-A) from Renesas
- Motor: R42BLD30L3 from MOON’S

Target software
The following shows the target software for this application:
- RX26T_MCBA_MCILV1_SPM_HALL_FOC_CSP_V100 (IDE: CS+)
- RX26T_MCBA_MCILV1_SPM_HALL_FOC_E2S_V100 (IDE: e^2 studio)

Reference materials
- RX26T Group User's Manual: Hardware (R01UH0979)
- Renesas Motor Workbench User's Manual (R21UZ0004)
- 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)

The following shows a summary of the items that you should check carefully when using this application note and the corresponding chapter for each.

Table 1-1 List of contents to be checked and corresponding chapters

Items to be checked	Corresponding chapter
Identify and select necessary devices	3
Select a power supply	4.2
Select a motor	4.3
Select an Inverter	4.4
Verify the wiring	4.9
Prepare a software development environment	5
Write the sample program to MCU	6.4
Install software for operating the motor on the PC	6.5
Modify the sample program, and then reflect the changes in RMW	6.6
Review the inverter’s internal information on the PC	6.7
Drive the motor	6.8
Stop the motor	6.9
Examine motor control algorithms	7
Examine the structure of the sample program	8
Verify and change the inverter parameters	9.7、9.4, 10.3, 10.4, 10.7, 10.8
Verify and change the motor parameters	9.8, 9.4
Tune the control performance of Hall sensor vector control	9.12
Change the PWM carrier frequency	9.5
Change the Hall sensor settings	9.12
Change the MCU settings	9.2,10
Frequently Asked Questions Check the troubleshooting tips	12

2. Glossary

The following lists the main terms used in this document and their explanations.

Table 2-1 Glossary

Term	Description
E2OB	E2 On-Board. The emulator function of E2Lite on the CPU board.
IDE	Refers to an Integrated Development Environment, for example, e^2 studio and CS+.
MC-COM	Refers to a set of communication jigs and tools for displaying waveforms. For details, refer to the following URL. https://www.renesas.com/en/products/microcontrollers-microprocessors/rx-32-bit-performance-efficiency-mcus/rtk0emxc90s00000bj-mc-com-renesas-flexible-motor-control-communication-board
RMW	Refers to Renesas Motor Workbench, which is the operation software specifically designed for motor control.
SPM motor	Also called SPMSM. This type of motors is used for servo motors that require smooth motion even at low speeds.
Inverter bus voltage	Refers to the DC voltage fed to the inverter circuit. Also called DC intermediate voltage.
Emulator	A device used to program an MCU. Also called ICE.
OpenLoop	Refers to a motor control technique that does not need position feedback signals to control the voltage.
Component	Refers to a driver module generated by Smart Configurator to facilitate the use of MCU peripheral functions.
Sensor-less	In this manual, this refers to "the absence of magnetic pole position sensors and speed sensors." Position and speed sensors have drawbacks in terms of cost and environmental resistance, so omitting the sensors is considered an advantage.
Magnetic saturation	Phenomenon in which the motor is magnetically saturated, and the magnetic flux is no longer intensified because a current above a certain level is applied. It causes the parameters to change, thus affecting motor control by the inverter.
Surface permanent magnet synchronous motor	See SPM motor.
Electrical angle	Phase angle of the output current flowing in the motor. It can be converted to a mechanical angle by dividing it by the number of pole pairs of the motor.
Mechanical angle	Refers to the rotation angle of the motor axis. One rotation of the axis per minute is 1 r/min.

3. Equipment and Software Used

3.1 List of Hardware Used

The following lists the hardware devices used for evaluating this sample program.

Table 3-1 List of hardware devices used

Hardware	Manufacturer	Part name
MCK-RX26T Renesas Flexible Motor Control Kit for RX26T MCU Group	Renesas	RTK0EMXE70S00020BJ
CPU board MCB-RX26T Type A (included in the motor control kit)	Renesas	RX26T RAM64KB Version (R5F526TFCDFP) / RTK0EMXE70C00000BJ
Inverter board MCI-LV-1 (included in the motor control kit)	Renesas	RTK0EM0000B12020BJ It is rated at 10 amps, but if it exceeds 5 amps, it is highly recommended to have a heat-dissipation-fin on the inverter
MC-COM (included in the motor control kit)	Renesas	Renesas Flexible Motor Control Communication Board RTK0EMXC90Z00000BJ
Motor	MOONS’	R42BLD30L3
DC power supply	COSEL	24V,11A
Power meter	Yokogawa Test & Measurement	WT500

3.2 List of Software Used

The following lists the software and its version used for evaluating this sample program. This sample program can be used within limitations of our development environment CS+ or e^2 studio evaluation edition.

Table 3-2 List of software used

Manufacturer	Software	Version	Remarks
Renesas	CS+	V8.12.00	Evaluation edition
Renesas	e^2 studio	2024-10	Free-of-charge edition
Renesas	RX Smart Configurator	V2.23.0	
Renesas	CC-RX	V3.06.00	Evaluation edition
Renesas	Renesas Motor Workbench	3.1.2	

4. Building a Hardware Environment

4.1 Hardware Environment Overview

This section describes a hardware environment in which a PM motor is operated using this sample program. Figure 4-1 shows an example of hardware configuration. In the sections that follow, the power supply (4.2), the motor and load system (4.3), the inverter (4.4), and the RX26T CPU board, and its monitoring/writing mechanisms (4.6) are described in detail.

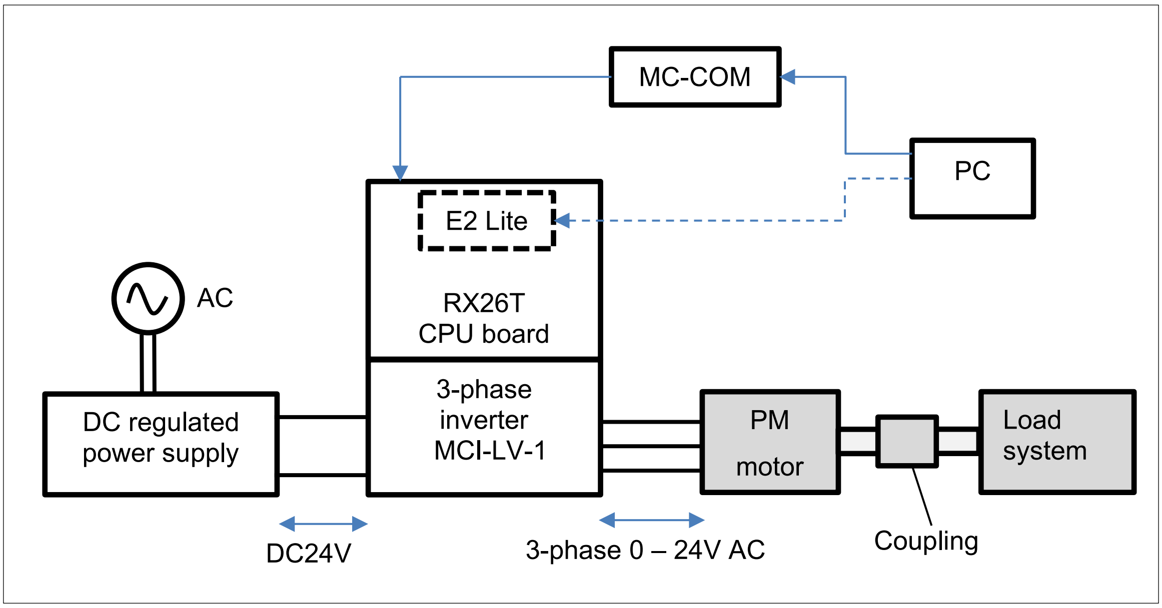


Figure 4-1 Hardware configuration example

4.2 Preparing the Power Supply

In this sample program, a voltage of 24 VDC is supplied from single-phase 100 VAC to the 3-phase inverter MCI-LV-1 by using a DC regulated power supply, AC adapter, or general-purpose power supply (capable of output 24V, 10A or more). USB power supply is not recommended because in this sample program, a higher rated power supply capacity for low speed high torque is required.

The voltage supplied to the inverter varies depending on the inductive voltage, rating conditions, and maximum load conditions of the motor to be used. Select an appropriate type of power supply based on your experimental environment and restrictions and conditions of the power supply to be used. The inverter introduced here has an output current of 10A max, and if the motor is changed, the rated operation may not be possible depending on the rating conditions of the motor. Please note.

4.3 Preparation of the Motor and Load System

Before connecting the inverter to the motor, obtain the parameters and constraints of the PM motor (required to drive the motor with vector control) using an LCR meter or other measuring instrument. If necessary, contact the manufacturer of the PM motor to obtain the parameter information.

If the motor parameters are changed, the following parameters for the current regulator, speed regulator, and Hall sensor vector control must be changed accordingly.

- Rated values (current, voltage, speed, number of poles)
- Ld, Lq, and resistance values
- Inductive voltage and magnetic flux linkage
- Inertia (moment of inertia) of the motor and the load system connected to the motor shaft

Table 4-1 shows the parameters of the R42BLD30L3 motor from MOONS', which we investigated. The parameters are based on our own measurements and may vary between individual motors and dependent on the measurement conditions. The accuracy or performance of these parameters is not guaranteed. Note that the magnetic saturation caused by the load current may change the motor parameter values during operation, thus affecting position estimation accuracy or operational performance.

When you perform a load test on the motor, a load tester (load system) or an actual load system capable of performing a load test is required in addition to the PM motor to verify the control performance. An appropriate load system must be selected based on the required load capacity of the motor and provided by the user.

Table 4-1 MOONS' R42BLD30L3 motor parameters

Primary resistance R	1.3 Ω
d-axis inductance	1.3 mH
q-axis inductance	1.3 mH
Moment of inertia	0.000003666 kgm2
Magnetic flux linkage Ψ	0.01119 Wb(rms)
Number of poles	8 (4 pairs)
Rated speed	4000 r/min
Maximum speed	4500 r/min
Rated torque	0.080 Nm
Rated frequency	266.67Hz(Electric angle) , 66.67Hz(Mechanical angle)
Rated voltage	36 V
Rated current	1.67Arms
Rated output	30 W

4.4 Preparation of the Inverter

When preparing your inverter, note the following information. This sample program is configured for MCI-LV-1 and must be changed if you use another inverter.

The control performance of vector control depends on the current detection value input from the current sensor and the magnetic pole position obtained from the Hall sensor. In other words, it is greatly influenced by the performance of the sensor itself and the variation and accuracy of the circuit that serves as a path of the signal output from the sensor. When selecting an inverter, careful consideration must be given to the design of the inverter.

- Rated capacity (VA)
- Dead time value (us)
- Types, characteristics, and signal specifications of the current sensor
- Characteristic data of the current sensor, including gain and offset values, the relationship between the current and voltage, and linearity of the signals
- Characteristic data of the voltage sensor, including gain and offset values and linearity of the signals

In addition, MCI-LV-1 has a board user interface (board UI) that allows the user to operate motor control commands. Table 4-2 shows the list of components and functions of the board UI.

Table 4-2 List of board user interface components and functions

Item	Interface components	Function
Rotational position/speed	Volume (VR1)	Rotation speed command value input (analog value)
START/STOP	Toggle switch (SW1)	Motor rotation start/stop command
ERROR RESET	Push Switch (SW2)	Command to recover from an error status (when an error occurs)
Current Offset Adjust	Push Switch (SW2)	Adjustment of current offset (when motor is stopped)
LED1	Orange LED	・At the time of motor rotation: ON ・At the time of motor stop: OFF ・During automatic starting of current offset adjustment 0.5s upon power on: ON ・After completion of automatic starting of current offset adjustment 0.5s upon power on : OFF
LED2	Orange LED	・At the time of error detection: ON ・At the time of normal operation: OFF
RESET	Push switch (RESET1)	System reset

4.5 Connecting Hall Sensor Wires

In this sample program, connecting the Hall sensor signal lines to MCI-LV-1 is required to use a Hall sensor. The correspondence table between MCI-LV-1 and MOONS's R42BLD30L3 is shown below.

For R42BLD30L3 connectors, MOLEX 43025-0600 is used. Mating parts are 43020-0600 and 43045-0600. The wire is AWG24 UL1061. If it is difficult to prepare the terminals, disconnect the connector from the Hall sensor wire, strip the coating, insert and secure it to the terminal block of MCI-LV-1.

Table 4-3 Correspondence between Hall sensor and inverter sensor input terminals

Signal line specifications	MCI-LV-1	MOON’S R42BLD30L3
DGND	CN6-1	5-BLACK
+5V	CN6-2	1-RED
HALL_W	CN6-3	4-BLUE
HALL_V	CN6-4	3-YELLOW/WHITE
HALL_U	CN6-5	6-ORANGE/YELLOW

4.6 Setting up RX26T CPU board

This section describes how to install RX26T CPU board (RTK0EMXE70C00000BJ), which can be plugged in to MCI-LV-1. You can plug RX26T CPU board to the side of MCI-LV-1 board. A connector for writing the sample program, a connector for MC-COM, and a PG connector for an external encoder are also provided.

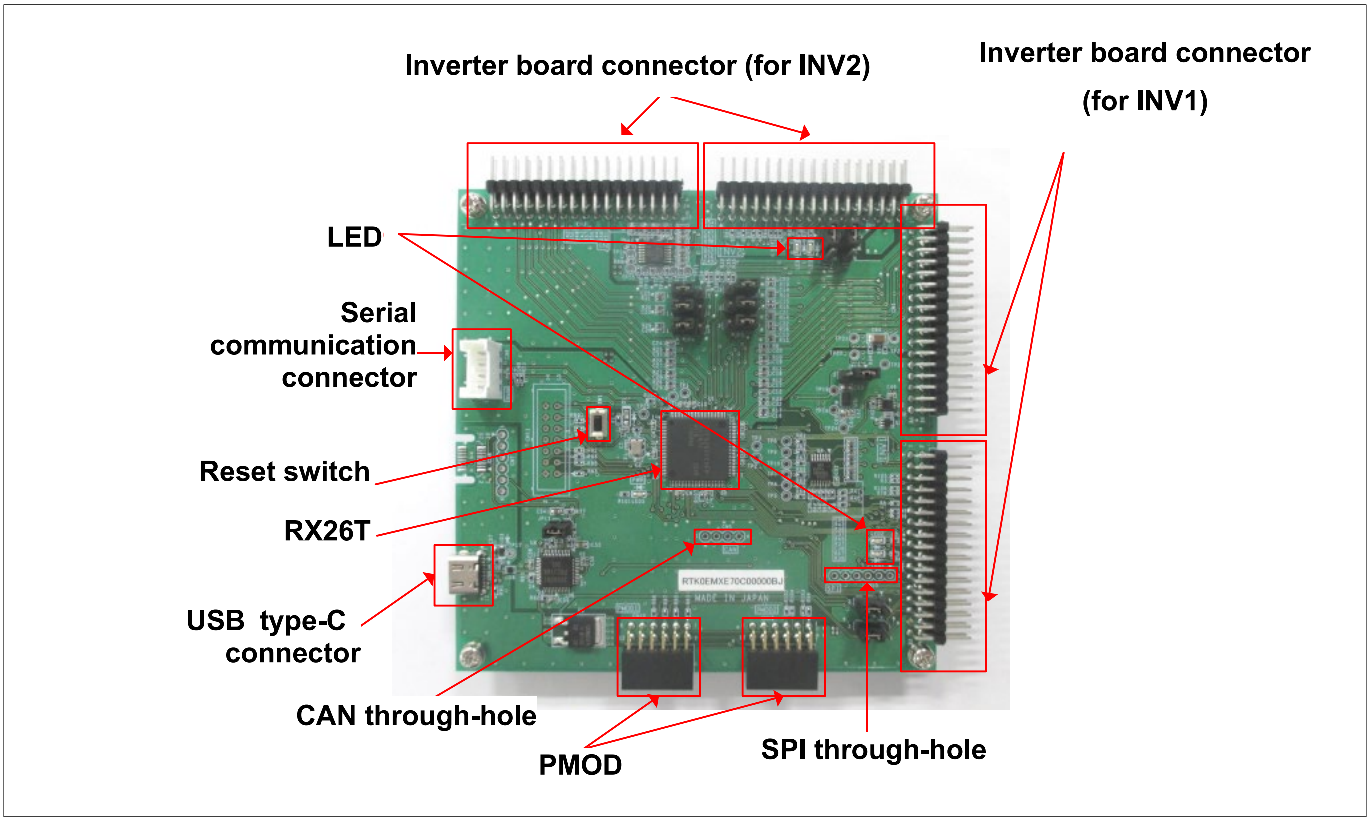


Figure 4-2 RX26T CPU board and its interface

4.7 Connection Example of Kit (MCK-RX26T)

Figure 4-3 shows an example of connection of a CPU board in combination with an inverter board kit (MCI-LV-1, model name: RTK0EM0000B12020BJ) and a communication board kit (MC-COM, model name: RTK0EMXC90Z00000BJ).

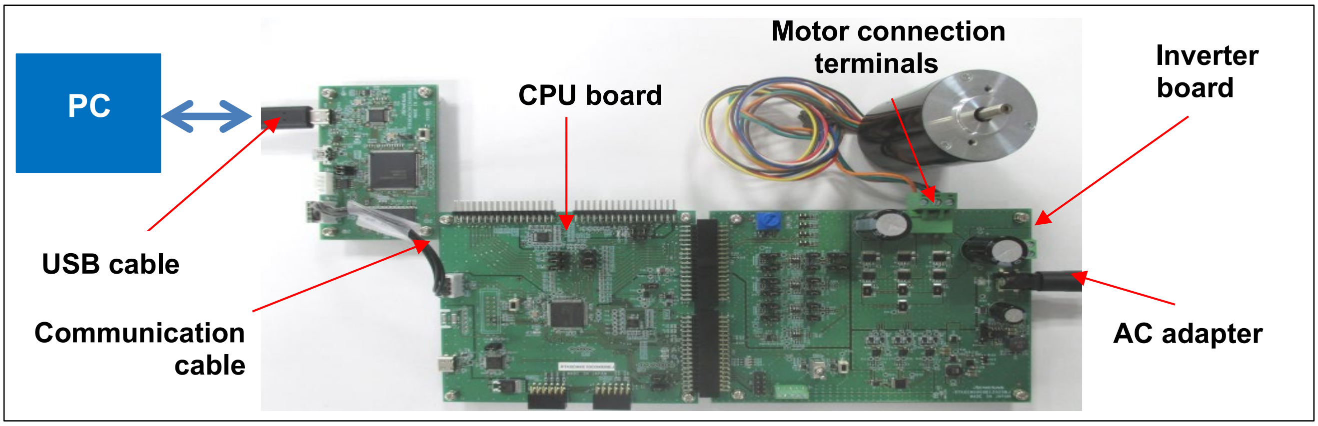


Figure 4-3 Connection example of kit (MCK-RX26T)

4.8 On-board Debugger

This product is equipped with an on-board debugger circuit E2 On-Board (hereinafter referred to as E2OB), and the RX26T program is rewritten using E2OB. When rewriting the program, open jumper JP11 and connect the CPU board to PC with a USB cable. E2OB acts as a debugger equivalent to E2 emulator Lite. When connecting from an integrated development environment (e.g., e^{2}studio) or a flash programming tool (e.g., Renesas Flash Programmer), set the debugger (tool) type to "E2 emulator Lite” at the time of configuration.

After rewriting the program, short out JP11 to enable the CPU board.

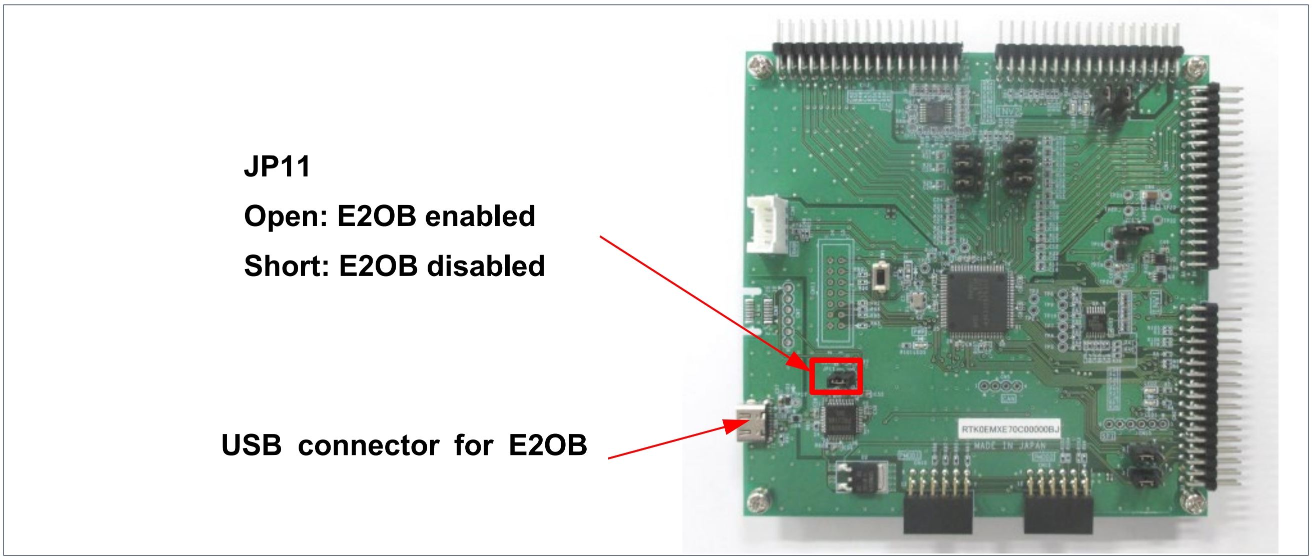


Figure 4-4 On-board debugger

4.9 Wiring

This section describes how to wire between the power supply, inverter, and motor. Terminal names vary depending on the device used, so be sure to refer to the instruction manuals of the device to verify the contents and specifications before wiring.

Figure 4-5 shows an example of wiring between the power supply and the inverter. Here, the output terminals of the regulated DC power supply are connected to the P and GND terminals of the inverter. Be careful not to connect with the wrong polarity. Figure 4-6 shows an example of wiring between the inverter and the motor. In addition to the three-phase motor wires, Hall sensor wires for position detection are connected to the inverter.

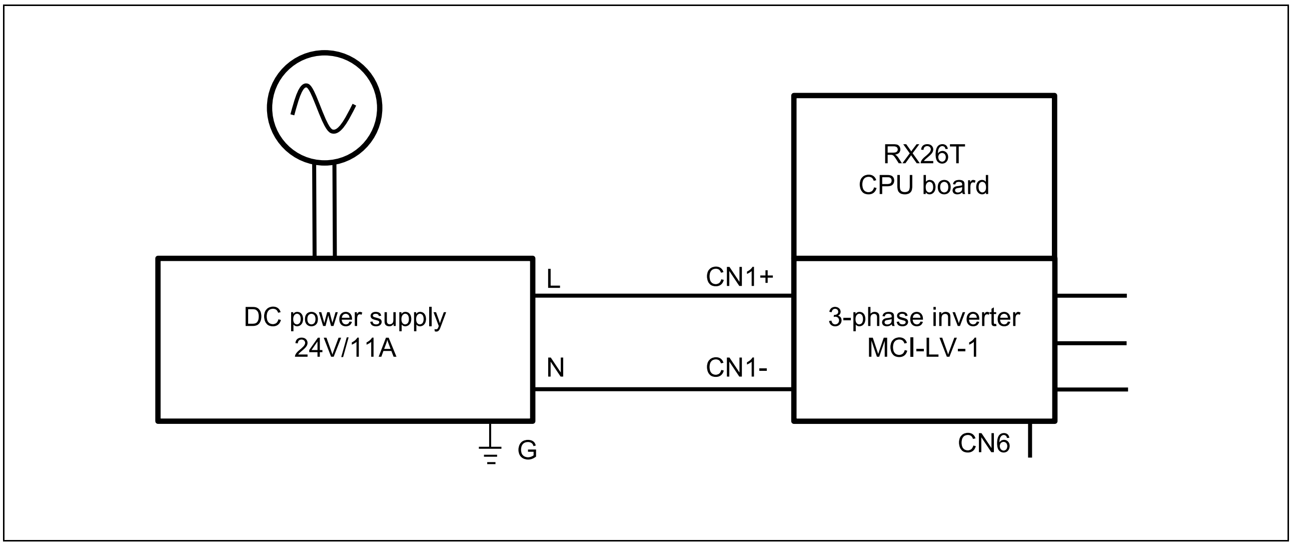


Figure 4-5 Wiring between power supply and inverter

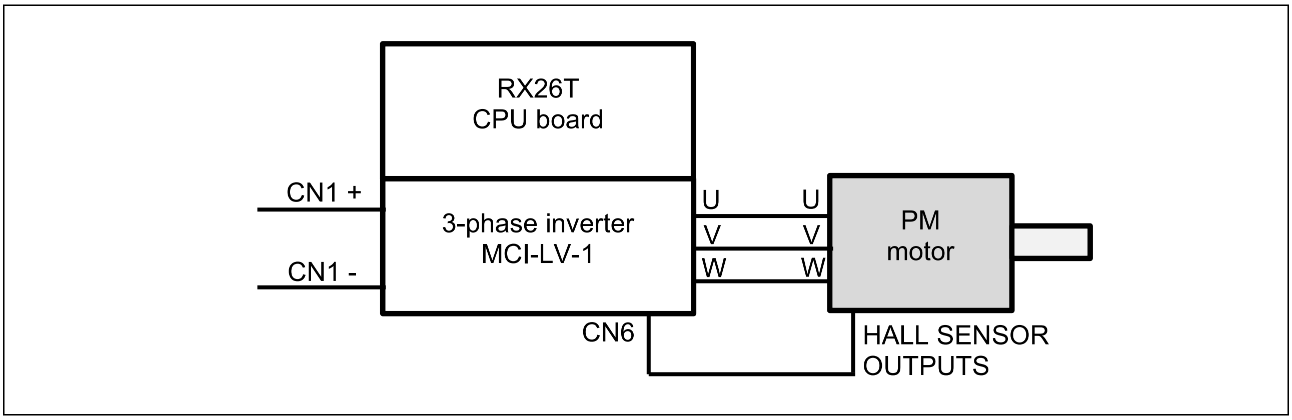


Figure 4-6 Wiring between inverter and motor

4.10 Using the Measurement Instruments

When evaluating control performance of the PM motor, a power meter, digital multimeter, torque meter, or external encoder can be provided for detailed control analysis of the inverter and motor. Consider which measurement instruments are required according to the user environment, required measurement accuracy, and the target performance specifications.

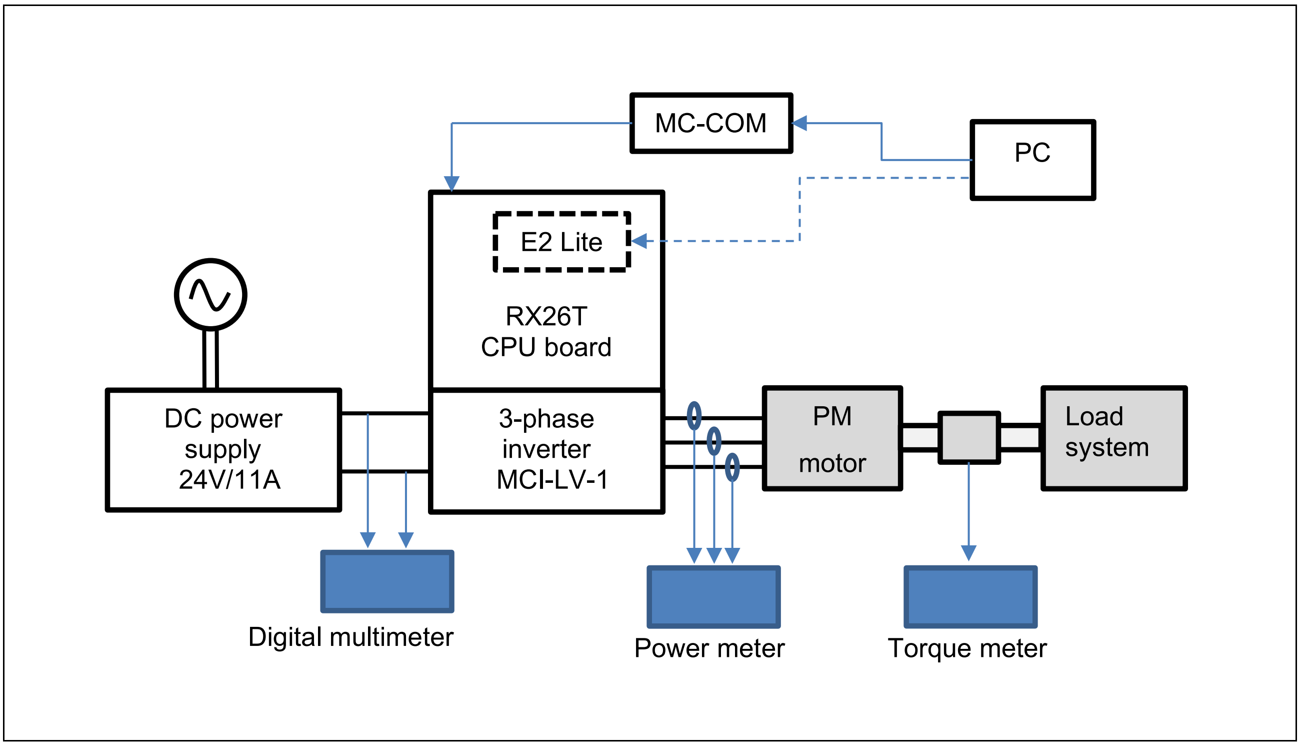


Figure 4-7 Examples of adding measurement instruments

5. Building a Software Environment

5.1 Using CS+

Download CS+ from the following URL:

https://www.renesas.com//en/software-tool/cs

CC-RX and Smart Configurator must be installed separately. For installation instructions, refer to the PDF manual supplied with CS+ or the video "CS+ Quick Start Guide (1/4) – Installation" posted on the above URL.

For more information on how to use it, refer to the PDF manual that you can download from the above URL or the videos.

5.2 Using e^2 Studio

Download e^2 studio from the following URL:

https://www.renesas.com//en/software-tool/e-studio

CC-RX and Smart Configurator must be installed separately. For installation instructions, refer to the PDF manual supplied with e^2 studio or the video "e^2 studio Tutorial Video for RX Family (1/4) - Installation" posted on the above URL.

For more information on how to use it, refer to the PDF manual that you can download from the above URL or the videos.

6. Driving the Motor

6.1 Precautions before Driving the Motor

When running the motor, note the following points. Improper use may cause electric shock or failure of the devices.

- Do not control the motor under conditions where the tracing and breakpoints are set while using E2OB. A sudden stop may cause the inverter to operate abnormally. Use RMW and debug under the conditions where the safety functions work properly.
- MC-COM can be safely used during operation because the signals are isolated. When a similar device is used, the GND of the PC and the inverter may be common, which could cause an electric shock hazard via the GND.
- Design the evaluation and experiment environment with the consideration that the motor can be stopped in an emergency.
- Beware of disconnection of the Hall sensor wire. If the Hall sensor wire is broken or has poor connection, the motor cannot be controlled normally.
- Because the motor shaft rotates, be sure to install the cover as a guard. Especially, parts such as couplings may scatter outside if damaged during rotation.
- When the inverter is stopped but the PM motor is still rotating, the PM motor generates an inductive voltage, thus applying voltage to the U/V/W three-phase wiring. Touching exposed conductive parts may cause electric shock.

6.2 Operation Preparation Procedure

Preparation procedures for operation are as follows:

Table 6-1 Operation preparation procedure

Step	Description	Related Chapters
1	Insert the CPU board to the inverter in advance.	-
2	Install the sample program, development environment (CS+, e^{2} studio, etc.), and emulator (E2OB, etc.) to the PC used.	5
3	Connect the PC to the CPU board via the in-circuit-emulator and supply a 5 V power to the CPU board.	-
4	Build the sample program in the development environment.	6.4
5	Write the built sample program to the CPU board.	
6	Remove the cable connecting between the emulator and the CPU board.	4.7
7	Connect MC-COM to the CPU board.	4.7
8	Supply a 24 VDC power to the inverter.	4.2
9	Use RMW installed on the PC to connect to the CPU board via MC-COM and verify that it can be connected properly.	6.5
10	In RMW, verify that the variables of this sample program and sensor information are properly displayed.	6.7
11	Use RMW to operate the motor.	6.8
	Use the board UI to operate the motor	6.9
12	Stop and shut down the motor	6.10

6.3 Connecting

Note that the device to be used between the CPU board and the PC is different when writing and when operating. Wiring methods for ① writing and ② motor operation are described below.

① For writing

E2OB is used to write the program to the CPU board.

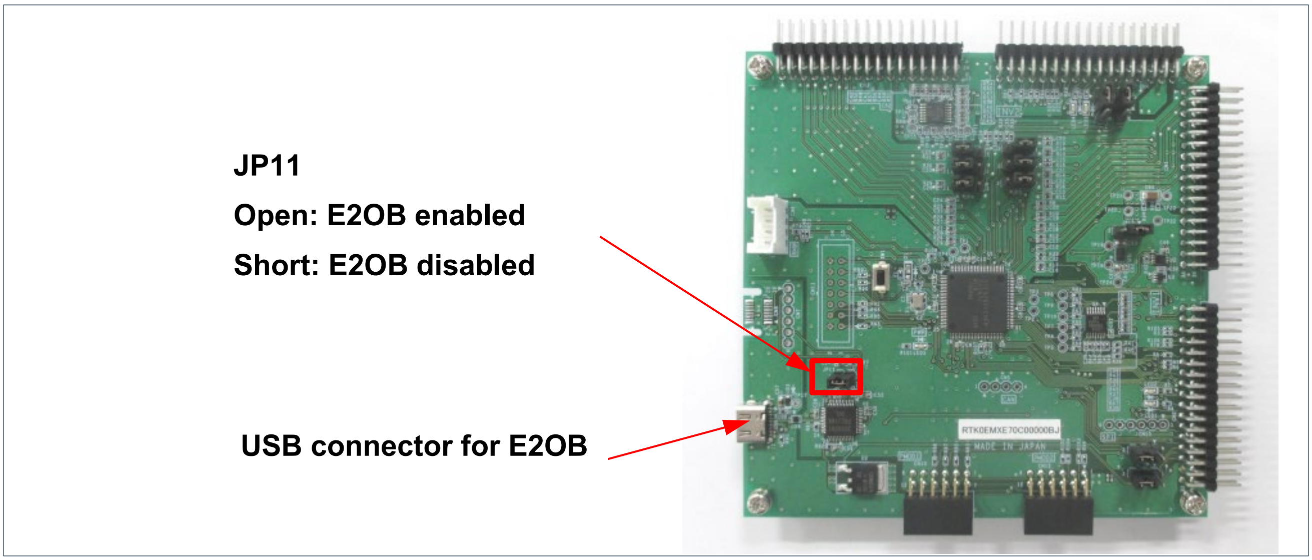


Figure 6-1 On-board debugger

② For motor operation

Connect to the CPU board using MC-COM (RTK0EMXC90Z00000BJ) as shown in Figure 6-2. The CPU board is connected to the PC via UART and can be operated from the PC using a COM port. In this sample program, you can use the board UI (VR and SWs on the inverter board) or RMW to operate the motor. MC-COM provides electrical isolation between the inverter and the PC and can be used safely even in high-voltage environments.

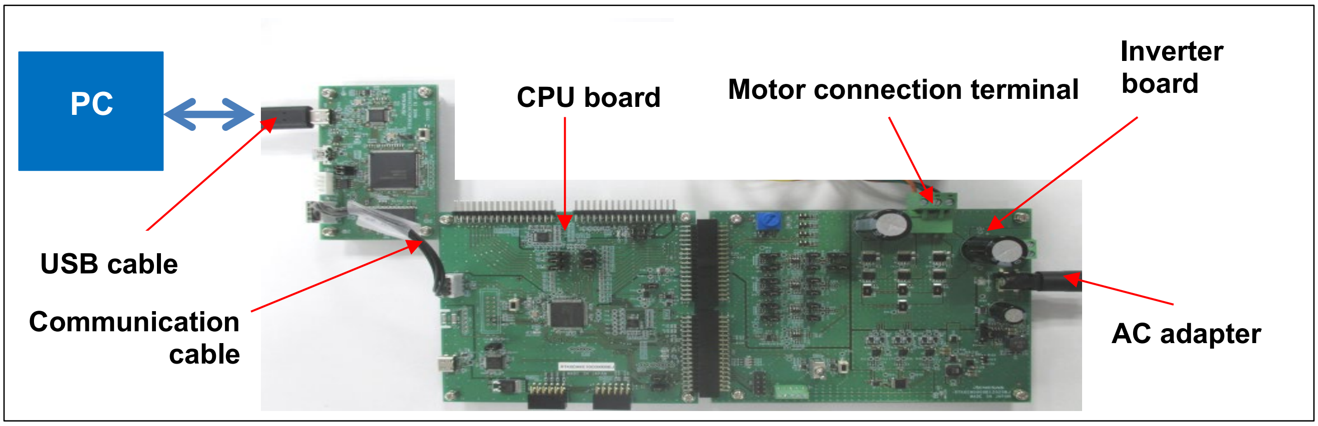


Figure 6-2 Example of wiring for motor operation

6.4 Writing the Sample Program

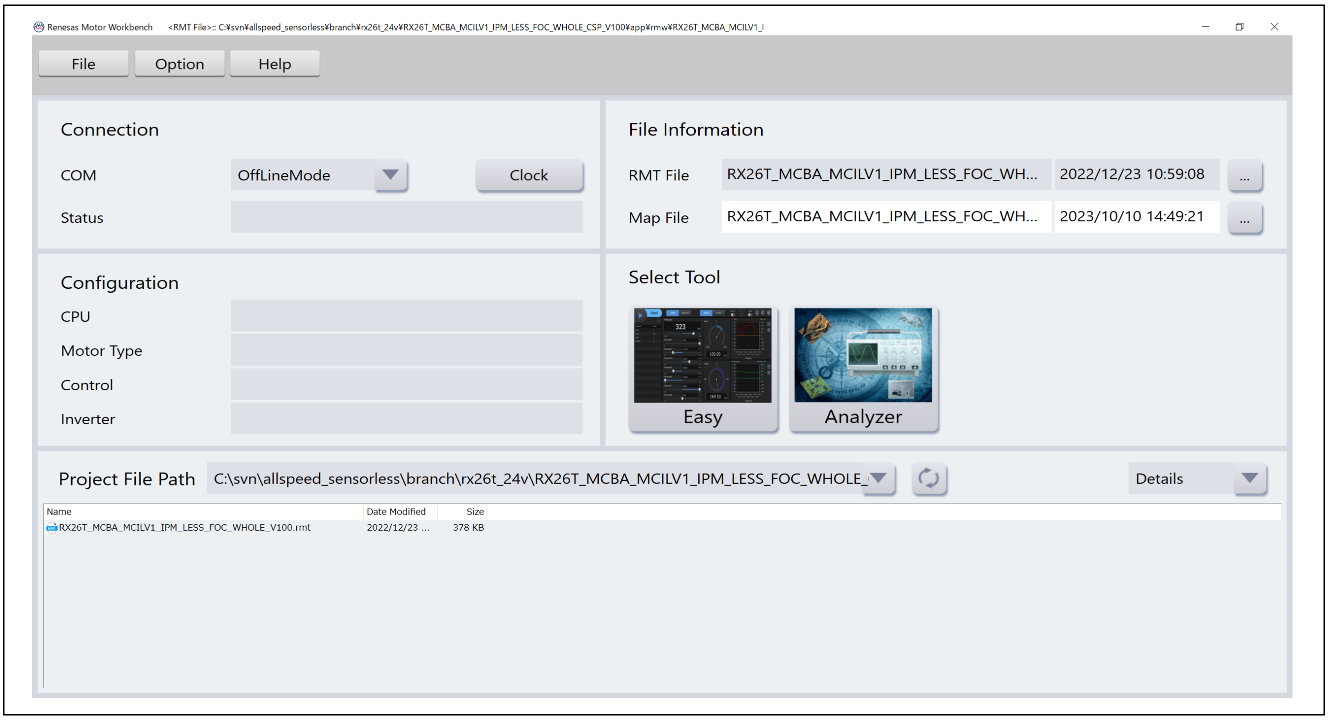
After you have downloaded the sample program from our website, use CS+, e^2 Studio, or Renesas Flash Programmer to write it to the MCU on the CPU board.

For details about how to write programs, see the documentation for CS+, e^2 studio, and Renesas Flash Programmer. To write a program, use the onboard debugger as shown in Figure 6-1 .

6.5 Introducing Renesas Motor Workbench

Renesas Motor Workbench (RMW), a motor control development support tool, is 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.

https://www.renesas.com/en/software-tool/renesas-motor-workbench



<RMT File>:: C:¥svn¥allspeed_sensorless¥branch¥rx26t_24v¥RX26T_MCBA_MCILV1_IPM_LESS_FOC_WHOLE_CSP_V100¥app¥rmw¥RX26T_MCBA_MCILV1_I

COM: OffLineMode

Status:

RMT File: RX26T_MCBA_MCILV1_IPM_LESS_FOC_WH... 2022/12/23 10:59:08

Map File: RX26T_MCBA_MCILV1_IPM_LESS_FOC_WH... 2023/10/10 14:49:21

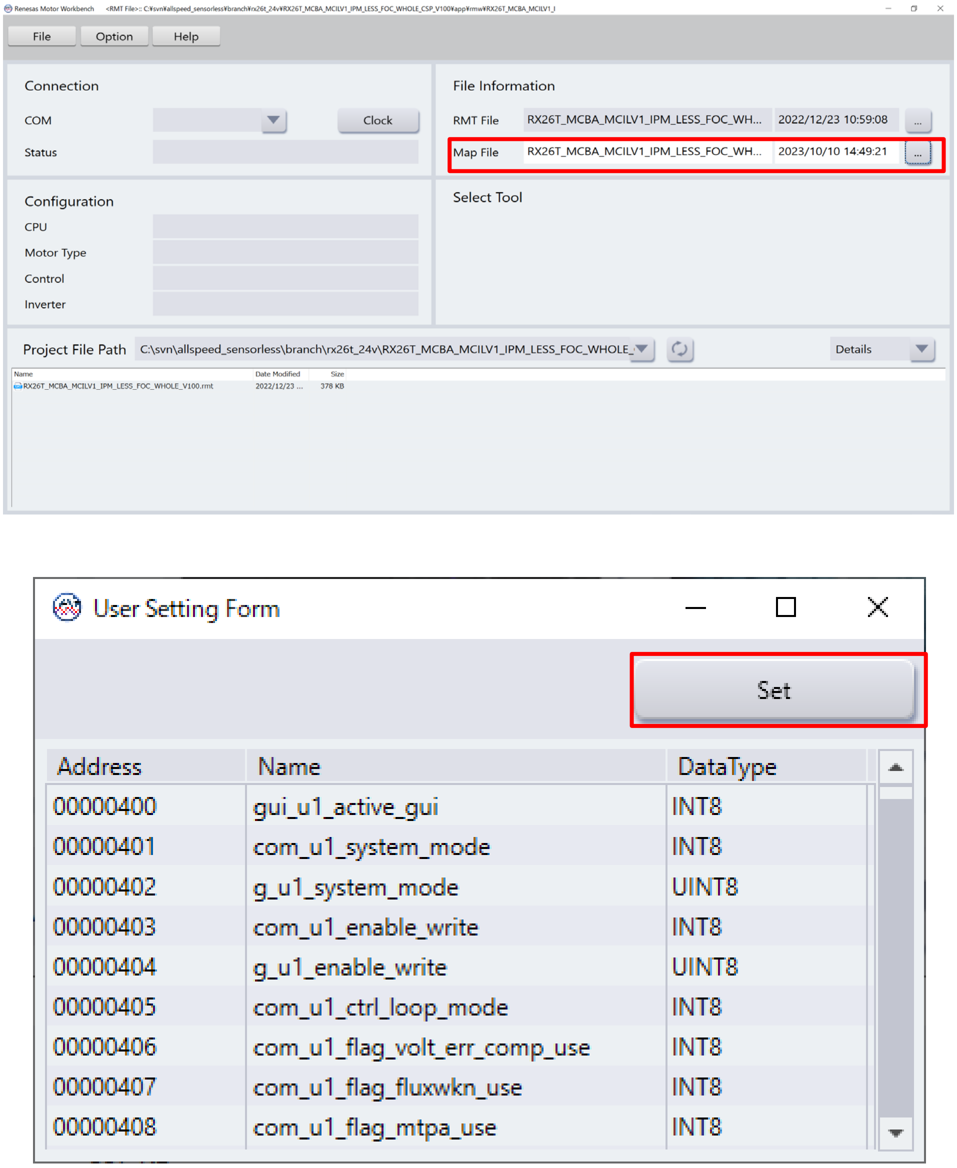
Project File Path: C:\svn\allspeed_sensorless\branch\rx26t_24v\RX26T_MCBA_MCILV1_IPM_LESS_FOC_WHOLE_

Name	Date Modified	Size
RX26T_MCBA_MCILV1_IPM_LESS_FOC_WHOLE_V100.rmt	2022/12/23 ...	378 KB

Figure 6-3 Window of Renesas Motor Workbench

6.6 Registering and Updating the MAP File

When a part of the sample program is changed by the user, the Map file including variables and other information needs to be registered and updated. If the software has not been changed, the Map file does not need to be registered or updated.



Renesas Motor Workbench <RMT File>:: C:¥svn¥allspeed_sensorless¥branch¥rx26t_24v¥RX26T_MCBA_MCILV1_IPM_LESS_FOC_WHOLE_CSP_V100¥app¥rmw¥RX26T_MCBA_MCILV1_I

RMT File RX26T_MCBA_MCILV1_IPM_LESS_FOC_WH... 2022/12/23 10:59:08

Map File RX26T_MCBA_MCILV1_IPM_LESS_FOC_WH... 2023/10/10 14:49:21

Project File Path C:\svn\allspeed_sensorless\branch\rx26t_24v\RX26T_MCBA_MCILV1_IPM_LESS_FOC_WHOLE_

Name	Date Modified	Size
RX26T_MCBA_MCILV1_IPM_LESS_FOC_WHOLE_V100.rmt	2022/12/23 ...	378 KB

User Setting Form

Address	Name	DataType
00000400	gui_u1_active_gui	INT8
00000401	com_u1_system_mode	INT8
00000402	g_u1_system_mode	UINT8
00000403	com_u1_enable_write	INT8
00000404	g_u1_enable_write	UINT8
00000405	com_u1_ctrl_loop_mode	INT8
00000406	com_u1_flag_volt_err_comp_use	INT8
00000407	com_u1_flag_fluxwkn_use	INT8
00000408	com_u1_flag_mtpa_use	INT8

Figure 6-4 Map file registration setup window (upper) and confirmation window (lower) in RMW

6.7 Variables Used for Working with RMW

In this sample program, the motor is controlled using RMW. Table 6-2 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 is the same as the value of the g_u1_enable_write variable. However, note that the variables indicated by an asterisk (*) do not depend on the value of the com_u1_enable_write variable.

Some parameters used for motor control can be changed while the motor is stopped. For details, see Table 8-8.

Note that the variable name prefix (for example, u1 and f4) is an abbreviation of the variable type. RMW automatically recognizes the variable name prefix, automatically selects the type, and displays the numeric value inside the variable in the Control Window.

Table 6-2 List of main input variables for Analyzer function

Analyzer function input variable name	Type	Description
com_u1_system_mode (*)	uint8_t	State Management 0: Stop mode 2: Speed control mode 3: Torque control mode 4: Open-loop control mode 5: Error release 8: Current offset adjustment
com_f4_ref_speed_rpm (*)	float	Speed command value (mechanical angle) [r/min]
com_u1_enable_write	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 variables are the same.

Table 6-3 lists main structure variables that are often observed when driving under speed control is 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.

Table 6-3 List of main variables for speed control

Main speed control variable name	Type	Description
g_st_hall_vector.u2_error_status	uint16_t	An error status. For details, see 6.8(e).
g_st_cc.f4_vdc_ad	float	Inverter bus voltage [V]
g_st_cc.f4_id_ref	float	d-axis current command value [A]
g_st_cc.f4_id_ad	float	d-axis current detection value [A]
g_st_cc.f4_iq_ref	float	q-axis current command value [A]
g_st_cc.f4_iq_ad	float	q-axis current detection value [A]
g_st_cc.f4_iu_ad	float	U-phase current detection value [A]
g_st_cc.f4_iv_ad	float	V-phase current detection value [A]
g_st_cc.f4_iw_ad	float	W-phase current detection value [A]
g_st_cc.f4_vd_ref	float	d-axis voltage command value [V]
g_st_cc.f4_vq_ref	float	q-axis voltage command value [V]
g_st_cc.f4_refu	float	U-phase voltage command value [V]
g_st_cc.f4_refv	float	V-phase voltage command value [V]
g_st_cc.f4_refw	float	W-phase voltage command value [V]
g_st_cc.st_rotor_angle.f4_rotor_angle_rad	float	Estimated magnetic pole position [rad]
g_st_sc.f4_ref_speed_rad_ctrl	float	Speed command value (mechanical angle) [rad/s]
g_st_sc.f4_speed_rad	float	Speed detection value (mechanical angle) [rad/s]

6.8 Controlling the Motor (using RMW)

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

(a) Changing the user interface to use Analyzer

① Confirm that the check box [W?] of "com_u1_system_mode" is selected.
② In the [Write] field, enter "0".
③ Click the [Write] button.

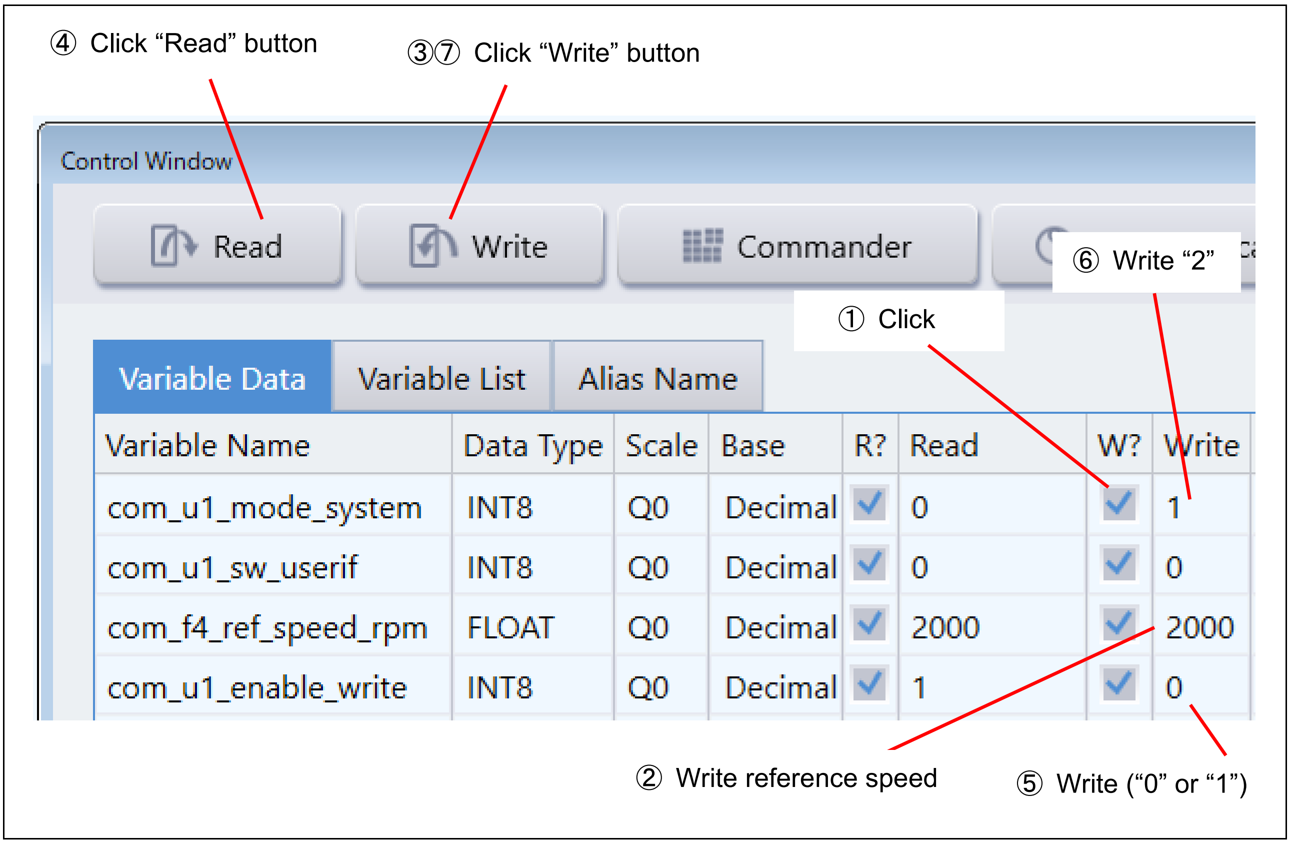
(b) Current offset adjustment (Manual)

The current offset adjustment will start automatically in 0.5 seconds after power on. LED1 lights during the adjustment. Manual adjustment can also be performed by following the steps below. No need to readjust manually in normal conditions.

① Confirm that the check box [W?] of "com_u1_mode_system" is selected.
② In the [Write] field, enter "8".
③ Click the [Write] button.
④ Click the [Read] button to check the [Read] field of the current "com_u1_mode_system".
⑤ Confirm that "com_u1_mode_system" returns to "0".

(c) Start driving the motor

① Confirm that the check boxes [W?] of “com_u1_mode_system”, “com_f4_ref_speed_rpm”, and “com_u1_enable_write” are selected.
② Enter the command rotation speed in the [Write] field of "com_f4_ref_speed_rpm".
③ Click the [Write] button.
④ Click the [Read] button to confirm the current [Read] fields of “com_f4_ref_speed_rpm” and ”g_u1_enable_write”.
⑤ Enter the same value (0 or 1) as the "g_u1_enable_write" confirmed in ④ in the "com_u1_enable_write" and reflect the variable value on the microcontroller.
⑥ Enter “2” in the [Write] field of "com_u1_system_mode".
⑦ Click the [Write] button



④ Click “Read” button

③⑦ Click “Write” button

⑥ Write “2”

① Click

Variable Name	Data Type	Scale	Base	R?	Read	W?	Write
com_u1_mode_system	INT8	Q0	Decimal	☑	0	☑	1
com_u1_sw_userif	INT8	Q0	Decimal	☑	0	☑	0
com_f4_ref_speed_rpm	FLOAT	Q0	Decimal	☑	2000	☑	2000
com_u1_enable_write	INT8	Q0	Decimal	☑	1	☑	0

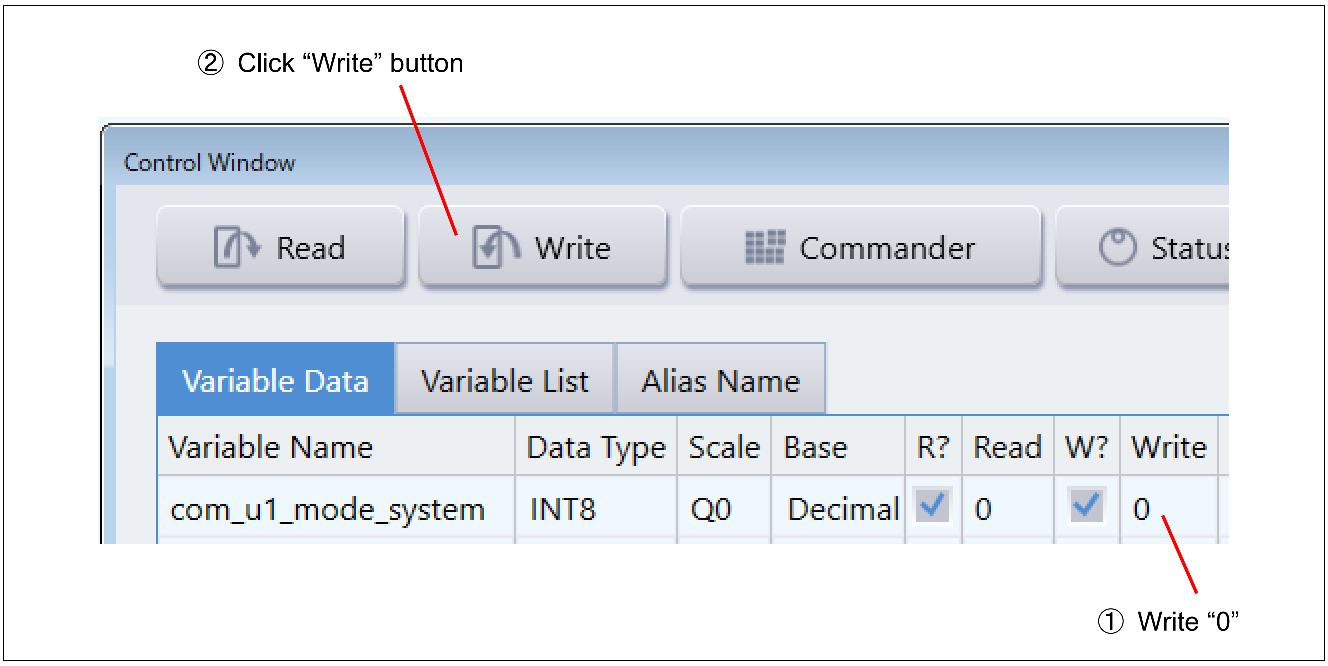
② Write reference speed

⑤ Write (“0” or “1”)

Figure 6-5 Procedure to start driving the motor

(d) Stop the motor

(1) Enter "0" in the [Write] field of "com_u1_mode_system".
(2) Click the [Write] button.



② Click “Write” button

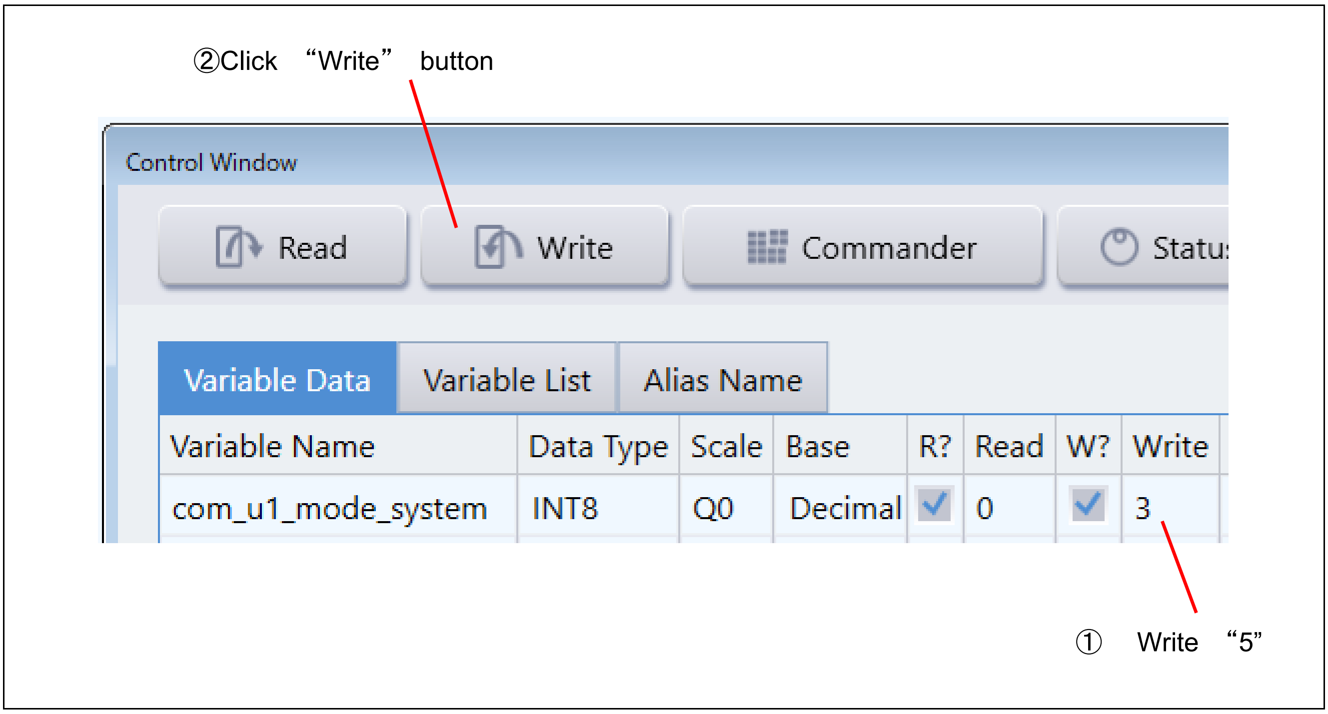
Variable Name	Data Type	Scale	Base	R?	Read	W?	Write
com_u1_mode_system	INT8	Q0	Decimal	☑	0	☑	0

① Write “0”

Figure 6-6 Procedure to stop the motor

(e) What to do in case of motor stop (due to an error)

(1) Enter "5" in the [Write] field of "com_u1_mode_system".
(2) Click the [Write] button.



②Click “Write” button

Variable Name	Data Type	Scale	Base	R?	Read	W?	Write
com_u1_mode_system	INT8	Q0	Decimal	☑	0	☑	3

① Write “5”

Figure 6-7 Procedure to clear errors

Table 6-4 Error Status Explanation

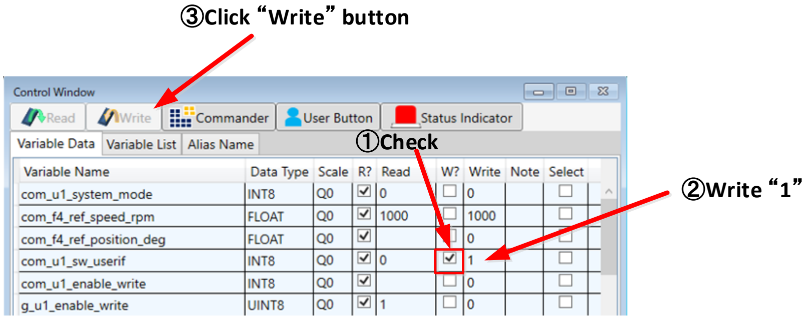
Value	Error Description	Assigned macro name
0x0000	No error	MOTOR_ERROR_NONE
0x0001	Hardware overcurrent error	MOTOR_ERROR_OVER_CURRENT_HW
0x0002	Overvoltage error	MOTOR_ERROR_OVER_VOLTAGE
0x0004	Overspeed error	MOTOR_ERROR_OVER_SPEED
0x0080	Low-voltage error	MOTOR_ERROR_LOW_VOLTAGE
0x0100	Software overcurrent error	MOTOR_ERROR_OVER_CURRENT_SW
0x0200	Overtemperature error	MOTOR_ERROR_OVER_TEMPERATURE
0xffff	Undefined error	MOTOR_ERROR_UNKNOWN

6.9 Controlling the Motor (using board UI)

(a) Switch the user interface

This sample program sets RMW UI as the default user interface. To use the board UI, follow the steps below to switch the UI. This change is reset to default by power ON/OFF or reset operation.

Confirm that the checkbox [W?] of "com_u1_sw_userif" is checked and enter “1” in the [Write] field, then click the [Write] button.



③Click “Write” button

①Check

②Write “1”

Variable Name	Data Type	Scale	R?	Read	W?	Write	Note	Select
com_u1_system_mode	INT8	Q0	☑	0	☐	0		☐
com_f4_ref_speed_rpm	FLOAT	Q0	☑	1000	☐	1000		☐
com_f4_ref_position_deg	FLOAT	Q0	☑		☐	0		☐
com_u1_sw_userif	INT8	Q0	☑	0	☑	1		☐
com_u1_enable_write	INT8	Q0	☑		☐	0		☐
g_u1_enable_write	UINT8	Q0	☑	1	☐	0		☐

Figure 6-8 Procedure for switching between RMW and board UI

(b) Run/Stop the motor

Current offset adjustments will be performed in 0.5s after power on. Please DO NOT run the motor while current offset adjustment is in progress (LED1 light on).

When using board UI, the start and stop of the motor is controlled by the input from SW1 (board UI) on the inverter board. A general-purpose port is assigned to SW1. The pin is read in the mail loop. When the pin is Low, it is judged that the start switch is pressed. When the pin is High, it is judged that the motor is to be stopped.

(c) Current offset adjustment

When using the board UI, current offset adjustment starts by pressing SW2 while the motor is stopped (LED1 is OFF). When current offset adjustment starts, LED1 is turned ON, and when the adjustment ends, LED1 is turned OFF.

(d) Motor rotation speed command value

The command value of motor rotation speed is determined by A/D conversion of the output value (analog value) of VR1 on the inverter board. The A/D converted value of VR1 is used as the rotational speed command value, as shown in the table below.

Note that the board UI does not support the torque control mode.

Table 6-5 Conversion ratio of rotational speed command value

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

6.10 Stopping and Shutting Down the Motor

To stop the motor from operating, follow the procedure below. In an emergency, prioritize the step ② and stop supplying 24 VDC.

① Perform the procedure for stopping motor described in 6.8(b) and 6.8(d).

② After confirming that the motor stops, operate the regulated DC power supply to stop supplying 24 VDC.

7. Motor Control Algorithm

7.1 Overview

This section describes the motor control algorithm of this sample program. Table 7-1 shows the motor control functions.

Table 7-1 Motor control functions of this sample program

Function item	Function description
Control method	Vector control with sensor
Pulse Width Modulation (PWM) method	Space vector modulation method (triangular wave comparison method can also be selected)
Position and speed detection methods	Calculus for Hall sensor
Control Mode	Speed control and torque control
Control and compensation functions	・Voltage error compensation ・Voltage phase lead compensation ・Decoupling control ・Flux-weakening control ・Friction compensation

7.2 Control Block Diagram

The following shows an example block diagram of the entire system for Hall sensor vector control.

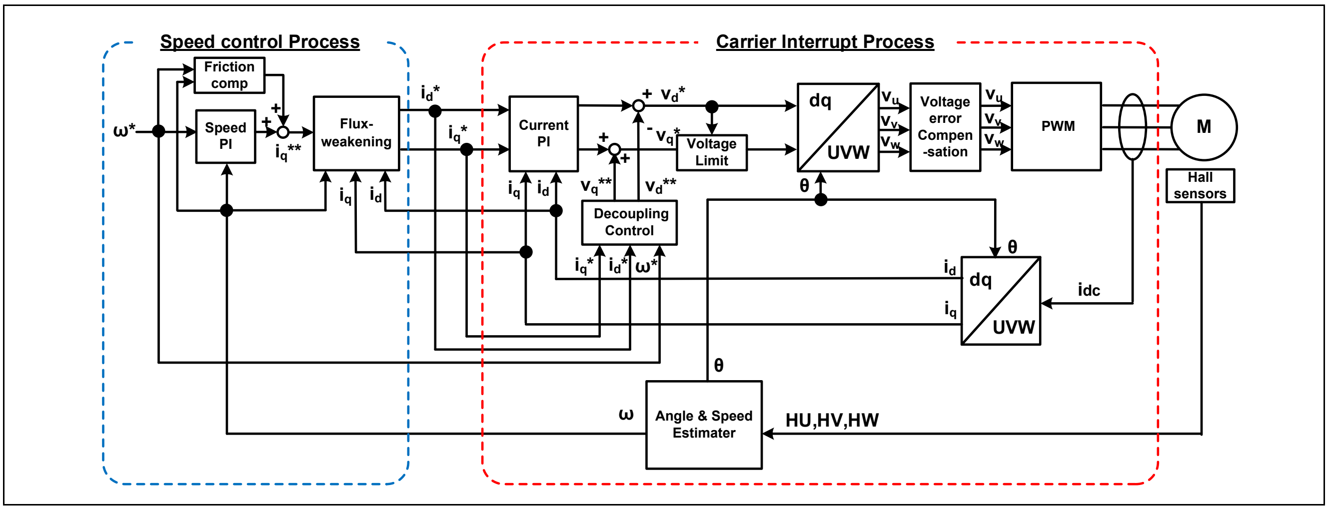


Figure 7-1 Hall sensor vector control schematic block diagram

7.3 Speed Control Function

The speed control function performs PI control so that the motor follows the speed command. When receiving a speed command value, the internal speed regulator outputs a current command value based on the deviation from the estimated speed value. This function also controls the submodules including flux-weakening control.

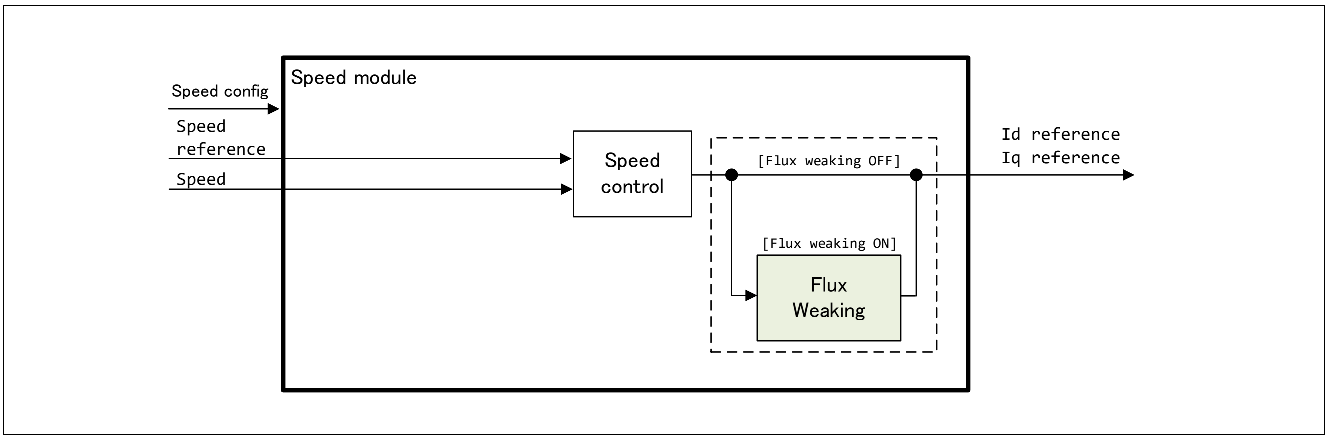


Figure 7-2 Functional block diagram for speed control

7.4 Flux-weakening Control

The flux-weakening control is a method that even under the condition where the inductive voltage (=ωΨ) generated proportional to the PM motor rotation exceeds the maximum output of the bus voltage, it controls the d-axis current in the negative direction and increases the d-axis current command value by the negative value to cancel it (Figure 7-3). By canceling the voltage saturation, it is possible to increase the q-axis current command value required for acceleration, realizing high-speed rotation and improvement of output in the high-speed rotation range.

The flux-weakening control automatically detects the state in which the motor rotation speed is high for the current voltage and the voltage margin is severe, then increases the value of Id* in the negative direction and cancels the inductive voltage according to the voltage equation of the PM motor.

For this control, first, calculate the inductive voltage limit value from the following formula, where R is the resistance value of the motor, and Ia is the sum of squares of the Id and Iq detection values (√(Id*Id+Iq*Iq)). For Vamax, use the maximum value of the voltage vector that has been pre-calculated by voltage error compensation or modulation processing.

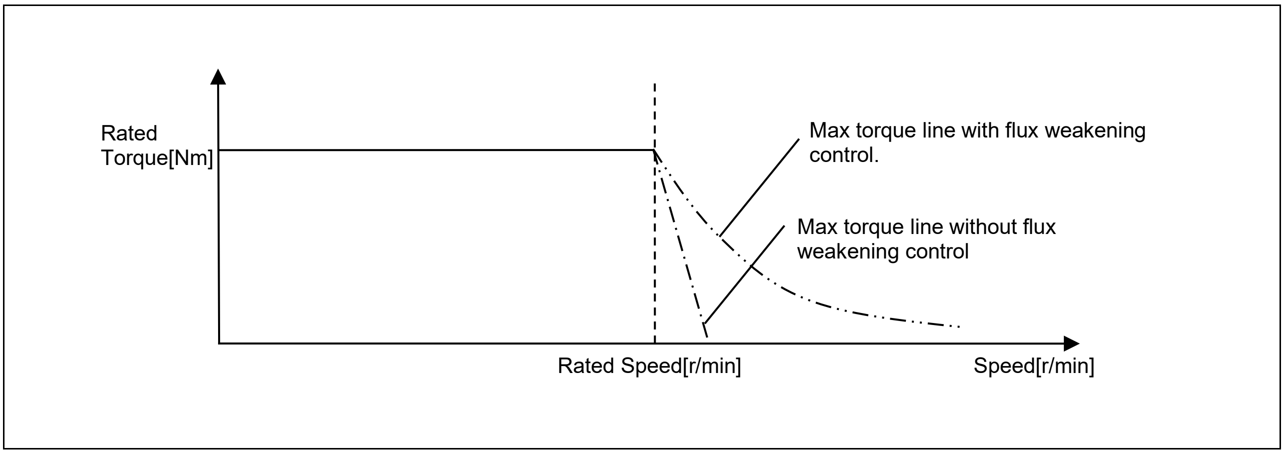


Figure 7-3 Example of the relationship between output torque and speed

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

V_{om}: Inductive voltage limit [V], V_{amax}: Maximum value of voltage vector [V]

I_a: Magnitude of the current vector [A]

Figure 7-4 Formula for calculating the inductive voltage limit

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

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

V_{om}: Inductive voltage limit [V], V_{amax}: Maximum value of voltage vector [V],
I_a: Magnitude of current vector [A]

Figure 7-5 Formula for calculating the d-axis current command value in flux-weakening control

7.5 Current Control Function

The current control function 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. This module also controls submodules including the decoupling, voltage phase lead compensation, voltage error compensation, and speed/angle detection. Figure 7-6 shows the module configuration.

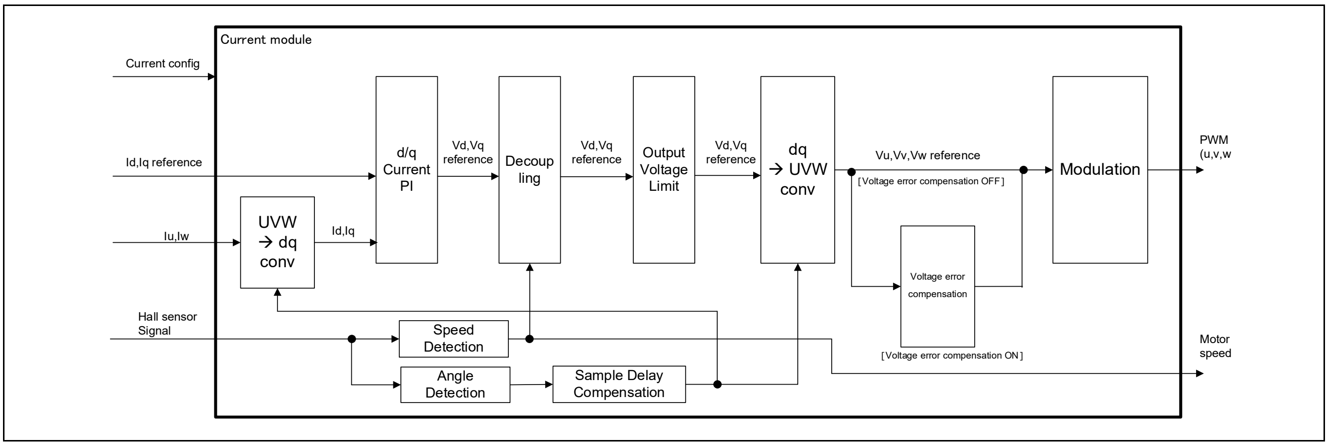


Figure 7-6 Functional block diagram of current control

7.6 Decoupling Control

The decoupling control is used to improve current responsiveness and to prevent currents from interfering with each other between the d- and q-axes, thereby losing stability. The equation used is shown below. It is a voltage equation for a typical PM motor.

$$V_{d\_dec}{}^{*} = RI_d{}^{*} - \omega L_q I_q{}^{*}$$
$$V_{q\_dec}{}^{*} = RI_q{}^{*} + \omega L_d I_d{}^{*} + \omega\Psi$$

id*,Iq*: Current command value [A], ω: Rotational speed (electrical angle) [rad/s],
R: Primary resistance of the motor [Ω], Ld,Lq: Inductance of the motor [H],
Ψ: Magnetic flux linkage of the motor [Wb]

The obtained voltage command values V_{d_dec}* and V_{q_dec}* are added to the voltage command values V_d* and V_q* output from PI regulator.

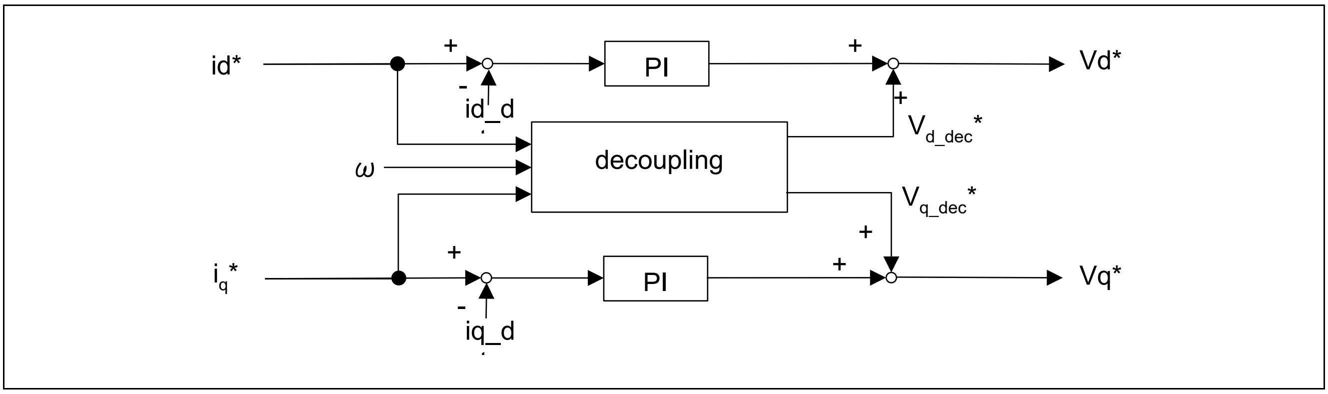


Figure 7-7 Functional block diagram of decoupling control

7.7 Hall Sensor Vector Control Function

7.7.1 Overview

From the signal of the Hall sensor, six pulses can be detected in one rotation of the electrical angle. Normally, six pulses per rotation is a low resolution, so when vector control is performed, angular complementing is required to output arbitrary torque. When a low-resolution encoder such as a Hall sensor is used for vector control, multiple methods are proposed as complementary processing. This sample program uses a simple and easy-to-implement calculus method. In the rotating state, the speed is detected by differentiating the amount of change in the angle of the pulse. Also, assuming that the motor is rotating at the detected speed, the angle can be obtained by integrating the speed by each sampling cycle.

This detection method uses software to detect pulses without using hardware and includes the detection error for the sampling cycle in the speed. Therefore, a parameter is provided so that you can select the method of correction by using a moving average based on the results of multiple detections in the past.

One of the unique starting characteristics of Hall sensor vector control is the phenomenon of sudden speed change at start-up. Since no speed signal can be obtained for a certain period of time between a standstill and the first input of Hall sensor output pulse, the integrator of the speed regulator may be saturated, and the speed may fluctuate due to excessive torque current commands.

To reduce speed fluctuations, use the static friction compensation described later and make adjustments according to the motor and load. Please note that even if it is compensated, it will be difficult in principle to eliminate speed fluctuations completely.

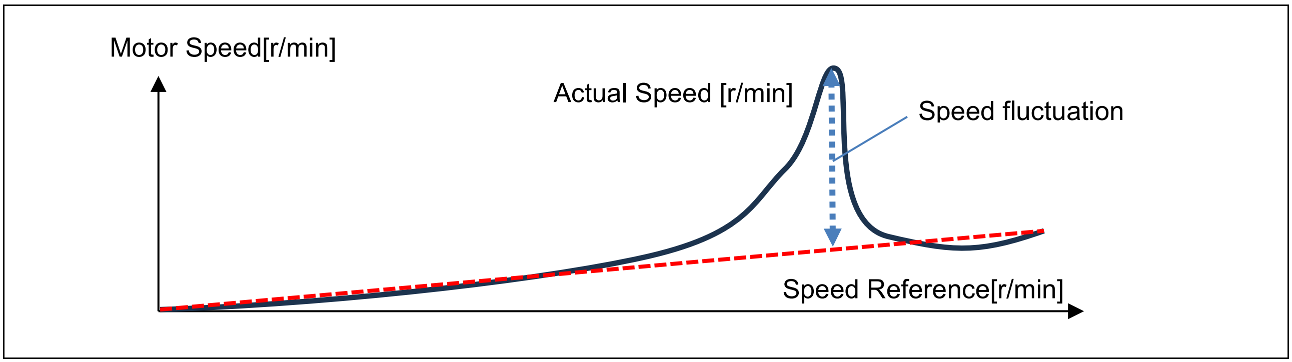


Figure 7-8 Speed variation at starting

7.7.2 Hall Sensor Signals and Hall Pattern Numbers

The Hall sensor consists of three sensors as shown in Figure 7-9, and they are placed every 120 degrees to correspond to the UVW windings. These Hall sensors are defined as HU, HV, and HW, respectively.

The Hall sensor signals consist of three signals: HU, HV, and HW. By using the difference between High/Low of these signals, one rotation of 360 degrees can be represented by the six magnetic pole positions. In other words, it can be regarded as an encoder with six pulses in one rotation of the electric angle.

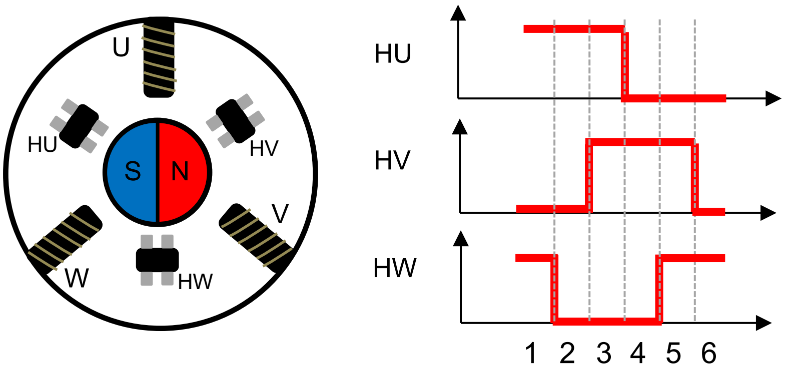


Figure 7-9 Hall sensor placement and Hall sensor signals

The unique Hall signal value is expressed from the 1-bit signal of HU, HV, and HW by the equation as follows.

$$Hall\ signal\ value = HW + HV \times 2 + HU \times 4$$

The relation between HU, HV, and HW signals and the Hall signal values is shown in the table as follows.

Table 7-2 Relation between Hall sensor signal values

HU	HV	HW	Hall signal value
0	0	1	1
1	0	1	5
1	0	0	4
1	1	0	6
0	1	0	2
0	1	1	3

The following table shows the relation between the above signal values and the Hall pattern numbers expressing 360 degrees as 1 to 6.

Table 7-3 Relation between Hall sensor signals and pattern numbers

Hall pattern number	Hall signal value
1	1
2	5
3	4
4	6
5	2

6	3

Based on the Hall pattern numbers, speed detection and angle detection processing are performed.

7.7.3 Speed Detection Method

The rotation speed is detected by checking the change in Hall sensor signals at a current control interval via GPIO, counting the number of the cycles during the change in the Hall sensor signal, and measuring the period T_{60deg} for an electrical angle of 2π/6 [rad] (one interval of Hall signal change). When the current control interval (50 μs) is Ts, T_{60deg} can be expressed by the following equation.

$$T_{60\mathrm{deg}} = Number\ of\ current\ control\ cycle\ k \times T_s$$

Thereby, the electric angular speed ω_e [rad / s] can be obtained from the following equation.

$$\omega_e = \frac{2\pi}{6} \times \frac{1}{T_{60deg}}$$

However, when estimating with one period of Hall sensor signal change, the effect of error due to jitter in the current control interval (50us) is included. In addition, the installation error of the three Hall sensors may appear as a disturbance in the detection speed. Therefore, you can also calculate based on the moving average to reduce the effect of jitter. In this sample program, you can select three detection methods: the past one period, the last six periods (one rotation), and auto-switching.

For example, if using a moving average of the last 6 periods, the electric angular speed ω_{e6}, which is the average of six periods, can be obtained by the following equation.

$$\omega_{e6} = \frac{2\pi}{6} \times \frac{6}{T_{360deg}} = \frac{2\pi}{T_{360deg}}$$

To simplify the angle calculation in the software implementation, the following equation is used for the equivalent calculation.

Electrical angle change in 1 carrier cycle [rad] = 2 π [rad] ÷ number of carrier interrupts (2 π period)

Electrical angular speed [rad/sec] = electrical angle change in 1 carrier cycle [rad] x carrier frequency [Hz]

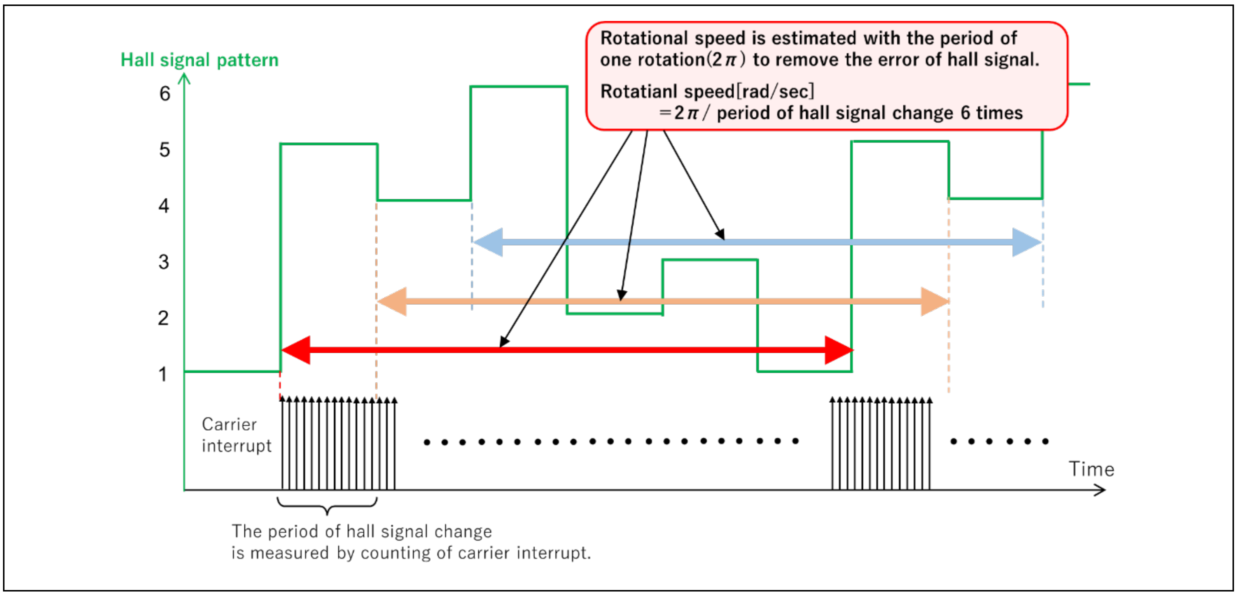
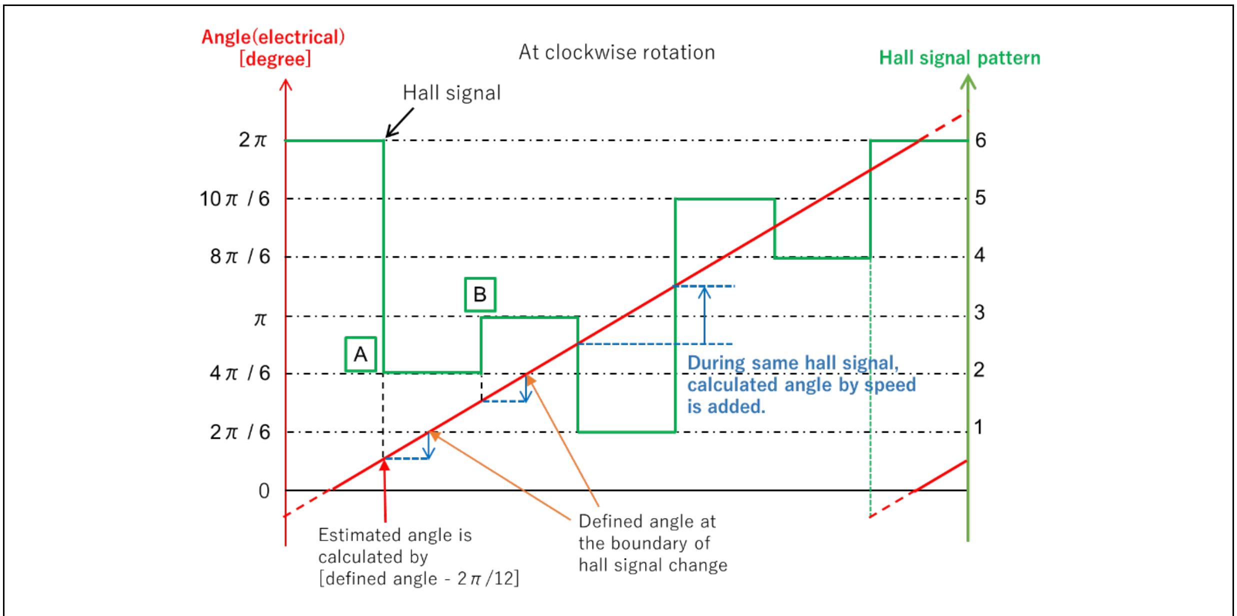


Figure 7-10 Conceptual diagram of speed estimation

7.7.4 Angle Detection Methods

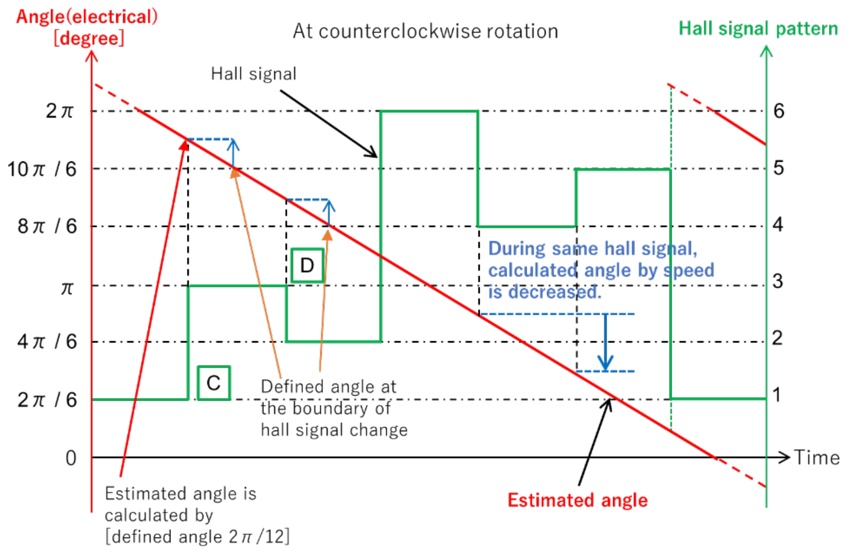
In angle detection, the electric angle is estimated from two pieces of information: the direction of rotation and the estimated rotation speed.

The direction of rotation is detected by the Hall sensor signal pattern. The Hall sensor signal pattern is determined by the motor used, therefore, when the signal changes, the direction of rotation can be detected by comparison between the current and last Hall signal pattern.



Hall signal pattern	1 -> 5 -> 4-> 6-> 2-> 3

Figure 7-11 Electrical angle estimation (CW rotation)



Hall signal pattern	1 -> 3 -> 2 -> 6 -> 4 -> 5

Figure 7-12 Electrical angle estimation (CCW rotation)

At point A in Figure 7-11, the Hall signal changes from 1 to 5. Therefore, the direction of rotation of the motor can be judged as clockwise. At this point A, the electric angle is set as below.

Point A (at clockwise rotation)

Electrical angle [rad] = 2π[rad] x reference angle adjustment value (1/6) + internal angle in same Hall signal [rad] + offset value [rad]

At the boundary of the Hall signal change, the electric angle is estimated with the reference angle 2π/6 [rad]. The reference angle adjustment value is set depending on the Hall signal as shown in Table 7-4.

Table 7-4 List of reference angle adjustment values

Hall Signal	1	5	4	6	2	3
Reference angle adjustment value	0(0/6)	1/6	2/6	3/6	4/6	5/6

At point A in Figure 7-11, the Hall signal changes from 1 to 5, so the reference angle adjustment value is set to 1/6, which corresponds to the Hall signal 5.

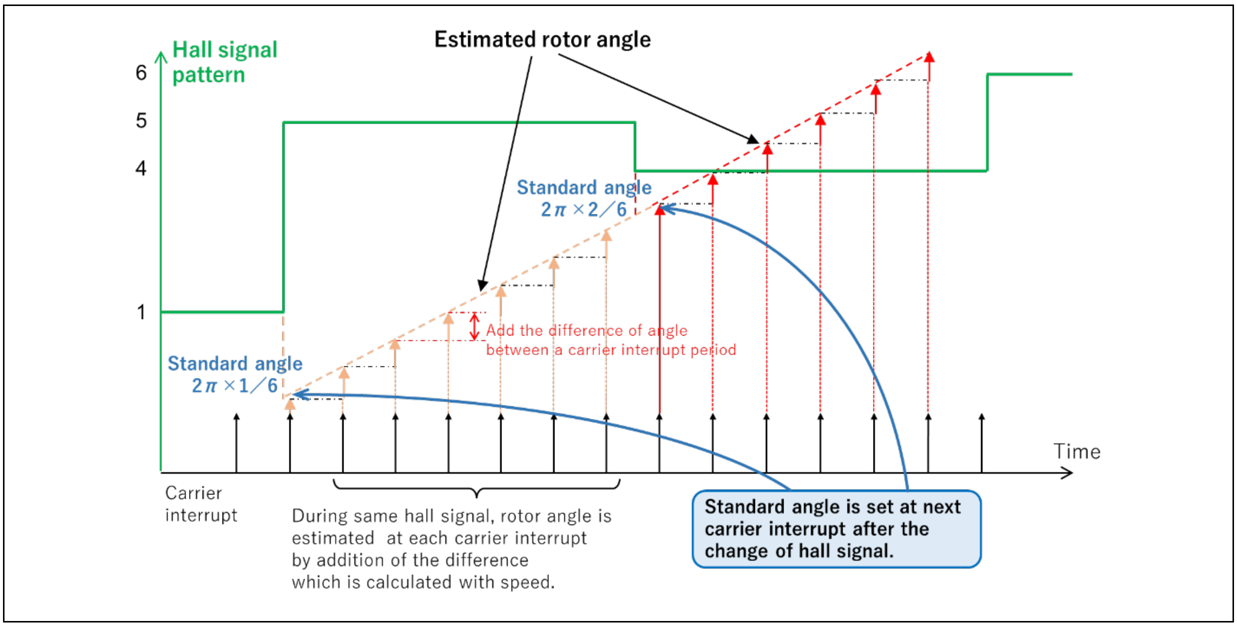


Figure 7-13 Conceptual diagram of Hall sensor internal angle estimation (at CW rotation)

At the boundary of Hall signal change, the internal angle is set to $-2\pi/12$ [rad] at clockwise rotation and $2\pi/12$ [rad] at counterclockwise rotation as the initial value. At each carrier interrupt, the difference is added based on the estimated speed information. However, the difference is limited between $-2\pi/12$ and $2\pi/12$ [rad] with consideration about signal errors and speed fluctuations. Any calculated angle is rounded if it exceeds the range of $-2\pi/12$ to $2\pi/12$.

Table 7-5 Formula for calculating the angle in the Hall signal by direction

Direction	Internal angle in Hall signal [rad]
Clockwise	Initial value ($-2\pi/12$) + estimated speed [rad/sec] x Ts x number of carrier interrupts or Initial value ($-2\pi/12$[rad]) + electrical angle change in one carrier cycle [rad]
Counterclockwise	Initial value ($-2\pi/12$) - estimated speed [rad/sec] x Ts x number of carrier interrupts or Initial value ($2\pi/12$[rad]) - electrical angle change in one carrier cycle [rad]

7.7.5 Friction Compensation

To accelerate smoothly from a standstill at start-up, it is important to detect signals of the speed and position sensors in a short time. However, low-resolution Hall sensors have a characteristic that the sensor signal changes in long intervals, making it difficult to increase the sensitivity of speed detection for control at high sampling cycle, and the speed regulator may not be able to output an appropriate torque, resulting in excessive torque output and sudden changes in speed.

Therefore, this sample program provides a function to preset parameters related to static friction when accelerating at start-up according to the characteristics of the load and motor and to provide feed-forward compensation to the q-axis current command value so as not to generate excessive torque at start-up. Note that if it is a load or motor with excessive static friction, or if the load changes rapidly, it may not be possible to suppress the sudden change in speed to the desired level.

An example of friction compensation is shown in Figure 7-14, where the horizontal axis is the speed ω [rad/s], and the vertical axis is the compensation current value I_{comp}[A]. This compensation is configured with four parameters: the static friction velocity threshold value Vs [rad/s], the static friction compensation value Fs [A], the viscous friction coefficient Fv [A], and the Coulomb friction compensation value Fc [A]. The unit of compensation value is amperes and is used to compensate for the speed regulator.

The static friction velocity threshold Vs is a parameter that determines up to which speed the compensation value Fs is applied when the motor starts to rotate from a standstill. Static friction compensation presets the current for the static friction required to start rotation.

The viscous friction coefficient Fv is a parameter that compensates for the friction that increases according to speed. If you are unsure, set it to 0. If 0 is set, the integrator of the speed regulator performs control instead of this parameter.

The Coulomb friction compensation value is a parameter that compensates for the friction that is always applied to the axis of the motor during rotation. For example, it compensates for frictions caused by bearings or some contact objects.

If the direction of rotation is reversed, the software automatically processes these compensations by reversing positive and negative.

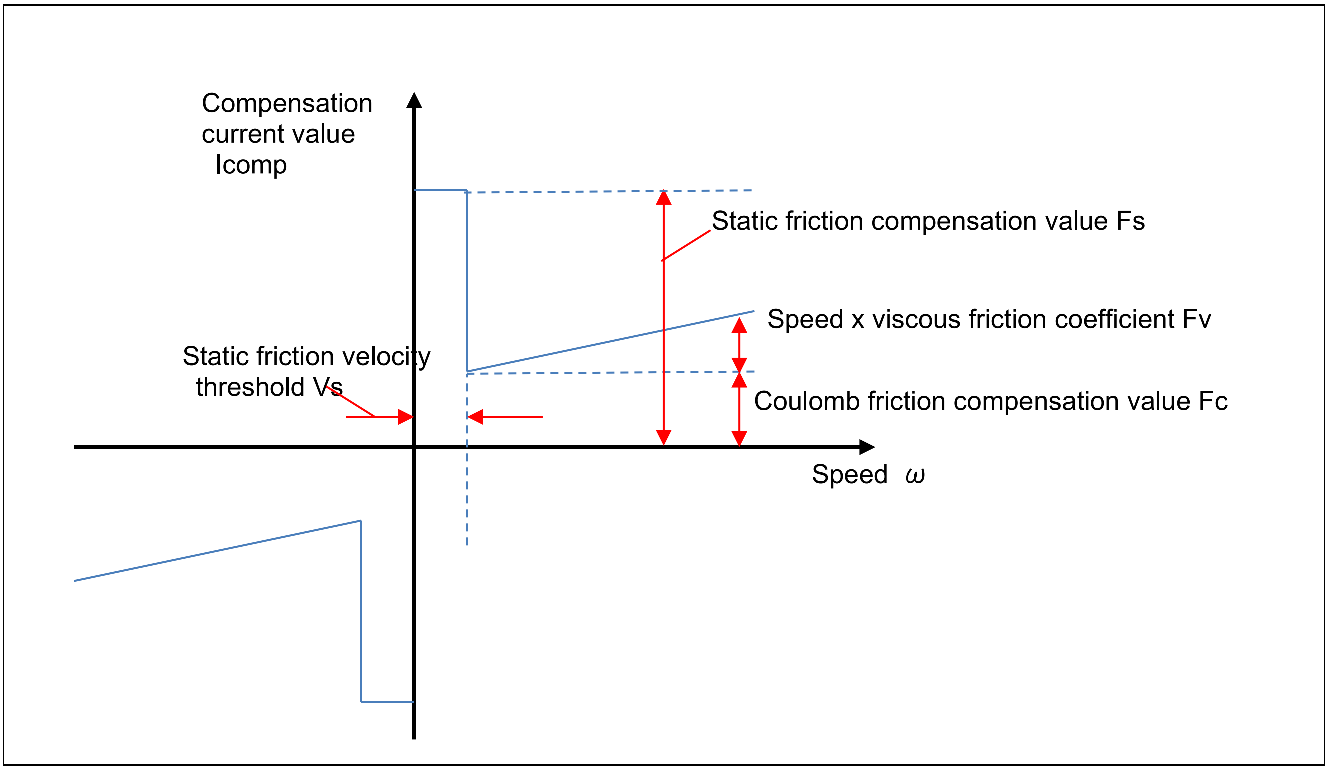


Figure 7-14 Conceptual diagram of friction compensation

The compensation current value I_{comp} that is output by the friction compensation process can be expressed as follows. The input is the speed command value ω_{ref}[rad/s] and the speed detection value ω [rad/s], and the output is the current compensation value [A].

$$I_{comp}(\omega, \omega_{ref}) = \begin{cases} 0\,, & |\omega_{ref}| < V_s \\ sgn\,(\omega)F_s, & |\omega_{ref}| \geq V_s,\ |\omega| < V_s \\ sgn\,(\omega)F_C + F_V\omega & else \end{cases}$$

The following is a list of parameters described above, a guide on how to set them, and setting examples. Since it is assumed that they are applied to the environment where theoretical calculation is difficult, they are methods for parameter calculation assumed to be calculated by experiments using actual equipment.

Table 7-6 How to calculate parameters of friction compensation

Parameter	Setting guide	Setting example
Static friction compensation value Fs	When starting without compensation, set a value of about 80% of the torque command value (q-axis current command value) when the motor starts rotating.	Fs=0.8*Iq_ref
Coulomb friction compensation value Fc	Set the Iq command value on rotating at 40% of the rated speed in an environment close to the actual one where the mechanical system is connected to the motor.	Fc=Iq_ref
Viscous coefficient of friction Fv	Set it to 0 if the load condition has gradual change, if the speed control PI is dominant, or if the conditions are unknown.	Fv=0
Static friction velocity threshold Vs	Set the threshold at which static friction occurs. As a guide, set a small value (e.g., 1.0 [rad/s]) considering the speed detection accuracy when the motor is stopped.	Vs=1.0

7.8 Voltage Phase Lead Compensation

When generating three-phase voltage commands for U, V, and W phases, 2-phase to 3-phase conversion is performed with the angle advanced by 0.5 control interval from the estimated angle. This process improves control stability. Improvement can be achieved for high-speed rotation applications, when the PWM carrier cycle is short, or when skipping is used.

During command calculation, the angle is continuously displaced as the motor rotation advances. This function takes advantage of the fact that the command calculation time is constant to interpolate the advancing angle from the previous angular displacement.

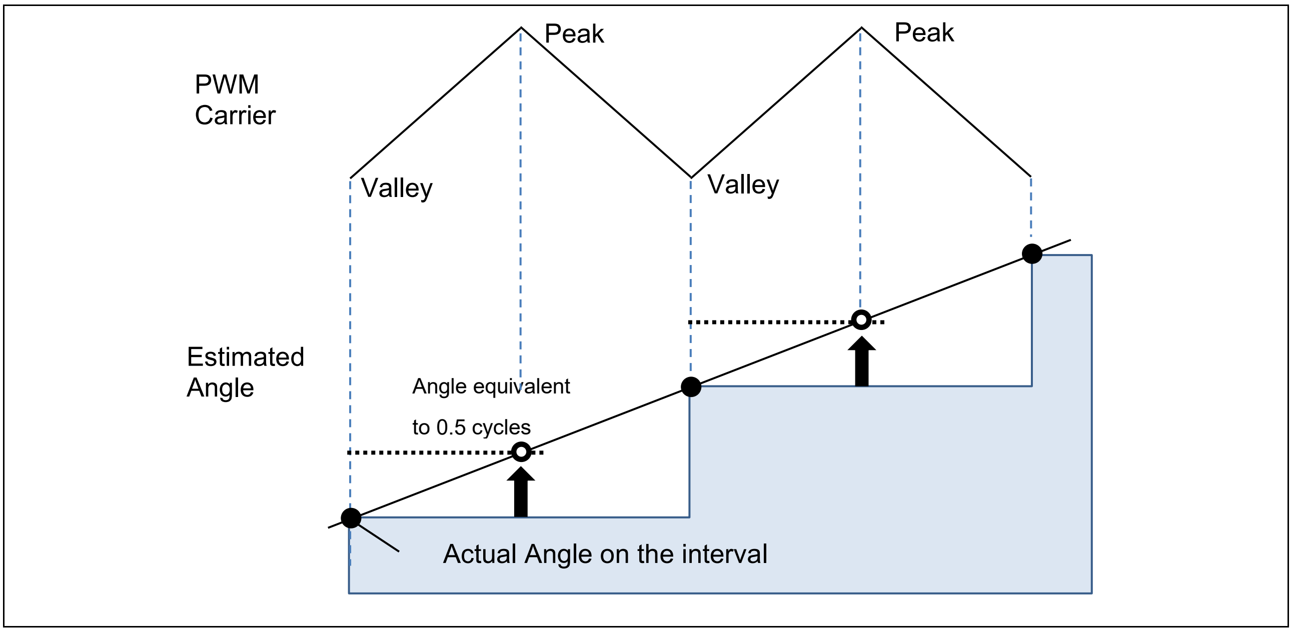


Figure 7-15 Example of the amount of angular advanced in a PWM carrier cycle

7.9 Voltage Error Compensation

In the voltage PWM inverter, 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 actual voltage applied to the motor, thus 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 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.

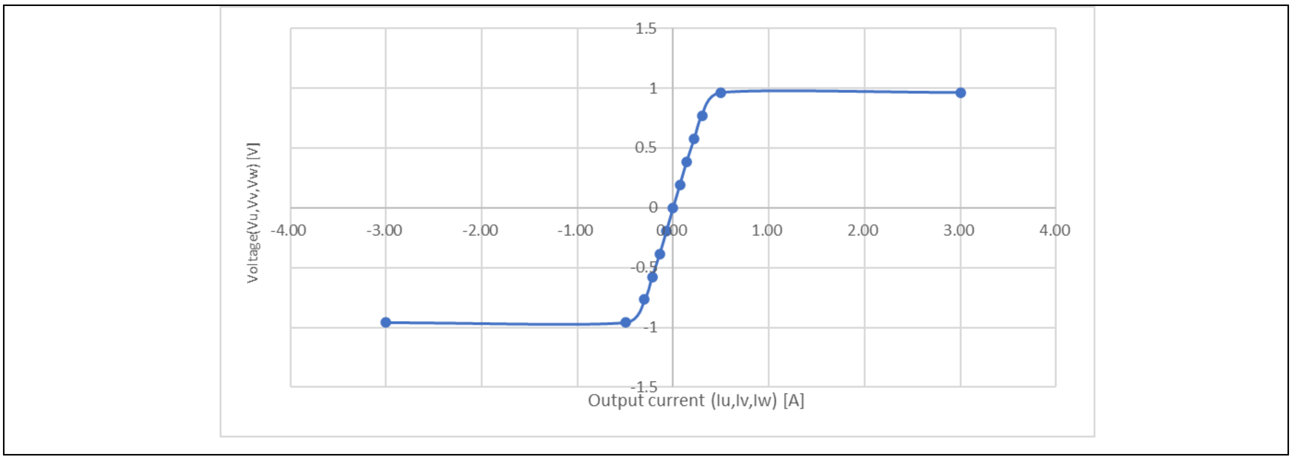


Figure 7-16 Current dependency of voltage error (example)

7.10 Pulse Width Modulation (PWM) Mode

In the sample program, the input voltage to the motor is generated by pulse-width modulation (PWM). In this module, the PWM duty cycle is calculated. In addition, a modulated voltage can be output to improve the voltage utilization. The modulation operation is set from the API of the current control module. In this sample program, one of two pulse-width modulation drive modes can be selected.

(a) Sinusoidal modulation (MOD_METHOD_SPWM)

The modulation rate m is defined as follows:

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

m: Modulation rate, V: Phase voltage command value for PWM generation [V], E: Inverter bus voltage [V]

(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.

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

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

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 (modulated wave)

The modulation rate m is defined as follows:

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

m: Modulation rate, V': Phase voltage command value for PWM generation [V], E: Inverter bus voltage [V]

8. Software Specifications & Configurations

8.1 Software Specifications

The basic specifications of the software of this system are shown below.

Table 8-1 Hall sensor vector control software basic specifications

<table>
<tr><th>Item</th><th colspan="2">Description</th></tr>
<tr><td>Control method</td><td colspan="2">Hall sensor vector control</td></tr>
<tr><td>Motor Control Start/Stop</td><td colspan="2">Determined by input from RMW or board UI</td></tr>
<tr><td>Rotor Magnetic Pole Position Detection</td><td colspan="2">Hall sensor</td></tr>
<tr><td>Input voltage</td><td colspan="2">DC 24V</td></tr>
<tr><td>PWM Carrier Frequency</td><td colspan="2">20 [kHz], Carrier cycle: 50 [μs] (Interrupt at trough)</td></tr>
<tr><td>PWM Modulation Scheme</td><td colspan="2">Triangular wave comparison mode or space vector modulation mode</td></tr>
<tr><td>Dead time</td><td colspan="2">2.0 [μs]</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 set point management</td><td colspan="2">CW: 0 [r/min] to 2400 [r/min]
CCW: 0 [r/min] to -2400 [r/min]</td></tr>
<tr><td>Natural frequencies of each control system</td><td colspan="2">Current control system: 300 Hz
Speed control system: 5 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 methods</td><td>Optimization focusing on the code size (-size) (default)</td></tr>
<tr><td>Protection Shutdown Process</td><td colspan="2">The motor control signal output (six outputs) will be deactivated when any of the following conditions are met:

1. The current peak values of all phases exceed 10 [A] (checked at intervals of 50 [μs]).
2. The current exceeds calculated value (4.723A) from rated current (checked at intervals of 50 [μs]).
3. The inverter bus voltage exceeds 60 [V] (checked at intervals of 50 [μs]).
4. The inverter bus voltage is less than 8 [V] (checked at intervals of 50 [μs]).
5. The rotation speed exceeds 2850 [r/min] (checked at intervals of 50 [μs]).

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

8.2 Overall Configuration of the Software

Figure 8-1 shows the overall configuration of the software.

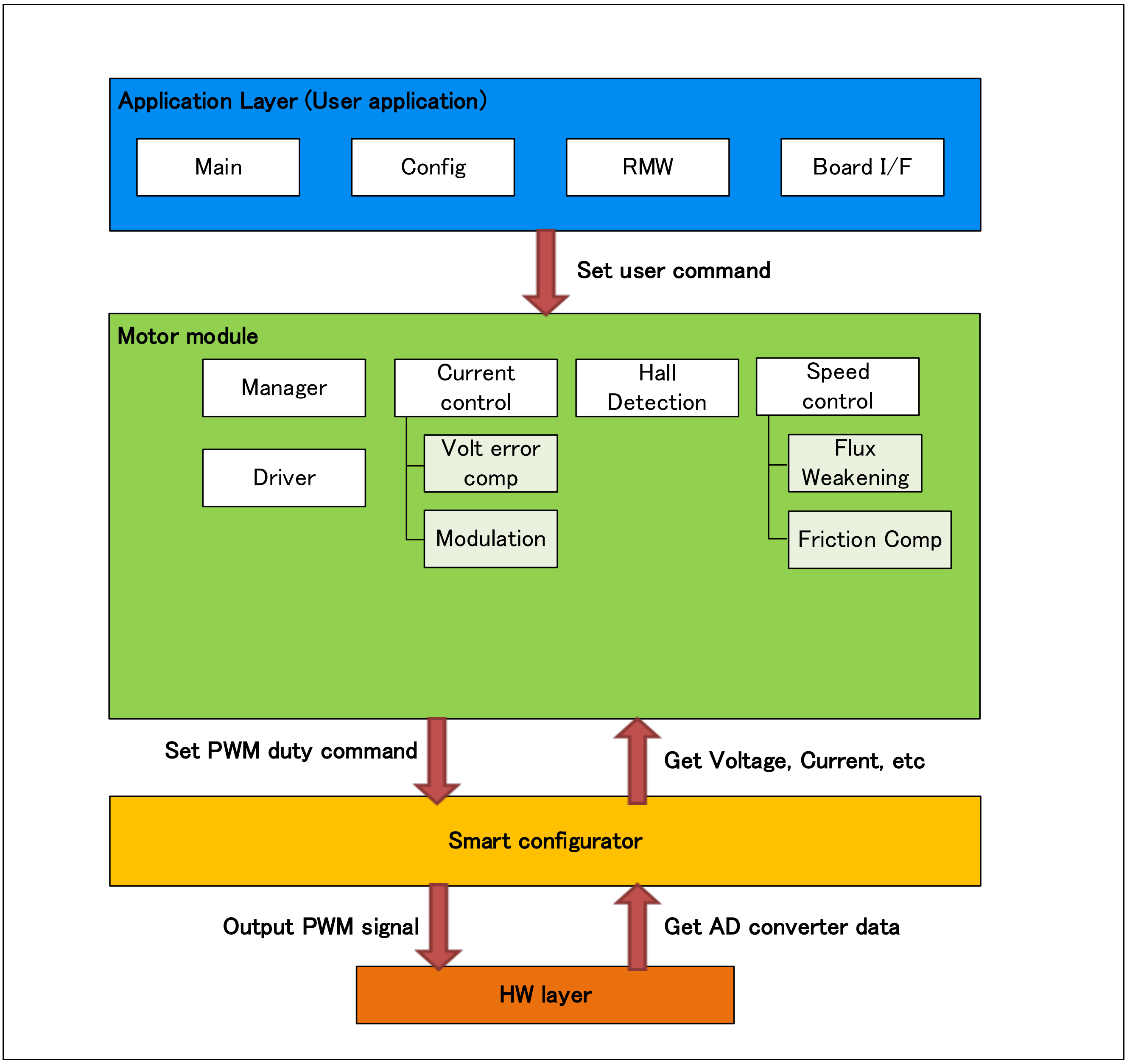


Figure 8-1 Overall configuration of motor control software

8.3 Task Description

The motor control tasks can be divided into two types of tasks: tasks that control the speed with a 0.5 ms cycle and tasks that are synchronized to the PWM carrier cycle (20 kHz, 50us).

Table 8-2 Interrupts and tasks to used

Task	Peripheral function	Interrupt function	Function called
Motor control interrupt (for speed control)	CMT0	r_Config_CMT0_cmi0_interrupt	R_MOTOR_HALL_ VECTOR_SpeedInterrupt
Motor control interrupt (for current control)	MTU3,4	r_Config_MOTOR_ad_interrupt	R_MOTOR_HALL_ VECTOR_CurrentInterrupt
Idle	-	r_app_main_ui_mainloop	R_MOTOR_HALL_VECTOR_StatusGet
At startup 1	-	r_app_main_init_motor_ctrl	R_MOTOR_HALL_VECTOR_Open
At startup 2	-	r_app_main_start_motor_ctrl	(Driver-related only)
At Reset	-	* Performed in the state transition process when recovering from an error.	R_MOTOR_HALL_VECTOR_ MotorReset
RMW operation	-	r_app_rmw_system_mode	R_MOTOR_HALL_VECTOR_ MotorStart MotorStop MotorErrorCancel CtrlTypeSet

8.4 File and Folder Configuration

Table 8-3 shows the folder and file configuration of the sample program.

Table 8-3 File and folder configuration

Folder	Subfolder	File	Remarks
app	main	r_app_main.c/h	User main function
	board_ui	r_app_board_ui.c/h	Definition of functions related to board UI
		r_app_board_ui_ctrl.h	Definition of MCU-dependent board UI functions
		r_app_board_ui_ctrl_rx26t_mcilv1.c	Definition of MCU-dependent board UI functions
	rmw	r_app_rmw.c/h	Definition of functions related to RMW Analyzer UI
		r_app_rmw_interrupt.c	Definition of RMW interrupt functions
		r_app_rmw_mcu.h	RMW hardware related definitions
		ICS2_RX26T.lib/h	Library for RMW communication
	cfg	r_app_control_cfg.h	Configuration definition for the application layer
motor_module	hall_vector	r_motor_vector_action.c	Definition of action functions
		r_motor_hall_vector_api.c/h	Definition of API functions for the manager module
		r_motor_hall_vector_manager.c/h	Definition of local functions for the manager Module
		r_motor_hall_vector_protection.c/h	Definition of functions for the protection function
		r_motor_hall_vector_statemachine.c/h	Definition of functions related to state transition
	current	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_pi_gain_calc.c	Definition of functions for calculating the control gain of the current control module
	speed	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 flux-weakening control module
		r_motor_speed_pi_gain_calc.c	Definition of functions for calculating the control gain of the speed control module
		r_motor_speed_friction_comp.lib/h	Definition of functions for friction compensation module
	sensor/hall	r_motor_sensor_hall_api.c/h	Definition of functions for Hall sensor modules
	driver	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 definitions
	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
src	smc_gen	See Table 8-4.	Drivers and API functions generated by Smart Configurator

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_MCBA_MCILV1_SPM_HALL_FOC_xxx_Vyyy.scfg"
> (xxx: CSP means CS+ version, E2S means e2 studio version.) yyy: revision number)

The following table shows the configuration of the folders and files generated by Smart Configurator.

Table 8-4 Smart Configurator folder file configuration

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	Config_MOTOR.c/h	Definition of functions related to the Motor component
			Config_MOTOR_user.c	Definition of user functions related to the Motor component
		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

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 pinning settings.

8.5 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. In this sample program, RMW (RMW UI) is used for setting and processing. These UIs are also used to control whether to drive or stop the motor and to set control command values.

8.5.1 Functions

Table 8-5 lists the functions that are configured in the application layer.

Table 8-5 Functions available in the application layer

Function	Description
Main processing	Enables or disables each user command in the system.
UI Processing	Manages the board UI and RMW.
Board UI processing	Acquires the speed command values from the switch and VR on the board and sets them.
RMW UI Processing	Acquires and set parameters (including command values).

8.5.2 Configuration

Table 8-6 shows the configurations used in the application layer.

Table 8-6 List of configurations

File name	Macro name	Description
r_app_control_cfg.h	APP_CFG_USE_UI	Initial UI settings RMW: MAIN_UI_RMW BOARD: MAIN_UI_BOARD
	APP_CFG_FREQ_BAND_LIMIT	This item sets the limit value for maintaining separation between the natural frequencies for current control, speed control, and position control.
	APP_CFG_MAX_CURRENT_OMEGA	This item sets the upper limit on the natural frequency for the current control system [Hz].
	APP_CFG_MIN_OMEGA	This item sets the lower limit on natural frequencies [Hz].
	APP_CFG_SCI_CH_SELECT	This item is used to select the SCI channel for RMW.

Table 8-7 List of initial values for configurations

Macro name	Set value
APP_CFG_USE_UI	MAIN_UI_RMW
APP_CFG_FREQ_BAND_LIMIT	3.0f
APP_CFG_MAX_CURRENT_OMEGA	1500.f
APP_CFG_MIN_OMEGA	1.0f
APP_CFG_SCI_CH_SELECT	ICS_SCI6_P81_P80

8.5.3 Structure and Variable Information

Table 8-8 lists the variables that can be used by users in the application layer. Table 8-9 lists the members of the structure provided for updating the motor module parameters by using RMW.

When you use RMW to set a value to a variable shown in Table 8-8, the application layer reflects the updated value in the variable in each control module using the Update function of each module via the structure shown in Table 8-9.

Table 8-8 List of variables

Variable	Description
g_st_rmw_input_buffer	Structure for updating RMW variables
g_u1_update_param_flag	Buffer transfer completion flag
com_u1_system_mode	Variable to switch the system mode for user entry 0: Stop mode 2: Speed control mode 3: Torque control mode 4: Open-loop control mode 5: Error release 8: Current offset adjustment
g_u1_system_mode	System Mode 0: Stop mode 2: Speed control mode 3: Torque control mode 4: Open-loop control mode 5: Error release 8: Current offset adjustment
com_u1_enable_write	Whether to enable rewrite of variables for user entry
g_u1_enable_write	Whether to enable rewrite of variables
com_u1_sw_userif	Variables for UI switching for user entry 0: RMW UI 1: BOARD UI
g_u1_sw_userif	Variables for UI switching
com_u2_offset_calc_time	Current offset value calculation time setting
com_f4_offset_rotor_rad	Magnetic pole position offset [rad]
com_u2_mtr_pp	Number of pole pairs of the motor to be driven
com_f4_mtr_r	Resistance of the motor to be driven [Ω]
com_f4_mtr_ld	d-axis inductance of the motor to be driven [H]
com_f4_mtr_lq	q-axis inductance of the motor to be driven [H]
com_f4_mtr_m	Magnetic flux of the motor to be driven [Wb]
com_f4_mtr_j	Rotor inertia of the motor to be driven [kgm^2]
com_f4_nominal_current_rms	Rated current of the motor to be driven [Arms]
com_f4_max_speed_rpm	Maximum speed (mechanical angle) of the motor to be driven [r/min]
com_f4_current_omega_hz	Natural frequency for current control system [Hz]
com_f4_current_zeta	Attenuation coefficient for the current control system
com_f4_speed_omega_hz	Natural frequency for speed control system [Hz]

Variable	Description
com_f4_speed_zeta	Attenuation coefficient for the speed control system
com_f4_ref_speed_rpm	Speed command value (mechanical angle) [r/min]
com_f4_speed_rate_limit_rpm	Maximum increment/decrement width for the speed command [r/min/s] (used when speed control is enabled, mechanical angle)
com_f4_overspeed_limit_rpm	Speed limit value (mechanical angle) [r/min]
com_u1_flag_volt_err_comp_use	Flag for voltage error compensation setting 0: Disable, 1: Enable
com_u1_flag_fluxwkn_use	Flag when flux weakening is used 0: Disable, 1: Enable
com_f4_friction_comp_static	Static friction coefficient [A]
com_f4_friction_comp_brake_speed	Brake speed during friction [r/min]
com_f4_friction_comp_coulombs	Coulomb friction coefficient [A]
com_f4_friction_comp_viscous	Viscous friction coefficient [A/(rad/s)]
com_f4_hall_speed_select	Selection of Hall sensor speed detection method: 0: Auto-switching 1: Last six intervals (one electrical rotation) 2: Last one interval
com_f4_hall_angle_correct	Angle correction [rad]
com_f4_hall_spdmix_dev_th	Speed deviation tolerance ratio in auto-switching mode (range 0 to 1.0)
com_f4_hall_spdmix_elec_speed_th	The electrical angular speed threshold (electrical angle) [rad/s] at which the speed detection method switches from auto-switching to last six interval method.
s_u1_cnt_ics	ICS watchpoint skip counter
com_f4_torque_ref_nm	Torque command value [Nm]
com_f4_ol_crnt_ref	Current command value during open-loop control [A]

Table 8-9 List of variables of the structure for RMW to update parameters

Structure	Variable	Description
st_rmw_param_buffer_t Structure for updating RMW variables	u2_offset_calc_time	Current offset detection time setting
	f4_offset_rotor_rad	Magnetic pole position offset [rad]
	st_motor	Structure for motor parameters
	f4_max_speed_rpm	Maximum speed [r/min] (mechanical angle)
	f4_friction_comp_static	Static friction coefficient [A]
	f4_friction_comp_brake_speed	Brake speed during friction [r/min]
	f4_friction_comp_coulombs	Coulomb friction coefficient [A]
	f4_friction_comp_viscous	Viscous friction coefficient [A/(rad/s)]
	u1_hall_speed_select	Selection of Hall sensor speed detection method: 0: Auto-switching 1: Last six intervals (one electrical rotation) 2: Last one interval

Structure	Variable	Description
	f4_hall_angle_correct_rad	Angle correction[rad]
	f4_hall_spdmix_elec_speed_th	The electrical angular speed threshold (electrical angle) [rad/s] at which the speed detection method switches from auto-switching to last six interval method.
	f4_hall_spdmix_dev_th	Speed deviation tolerance ratio in auto-switching mode (range 0 to 1.0)
	f4_current_omega_hz	Natural frequency for 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_ref_speed_rpm	Speed command value [r/min] (mechanical angle)
	f4_speed_rate_limit_rpm	Speed variation limit [r/min/s] (mechanical angle)
	f4_overspeed_limit_rpm	Speed limit value [r/min] (mechanical angle)
	f4_torque_ref_nm	Torque command value [Nm]
	f4_ol_crnt_ref	Current command value for open-loop [A]
	u1_flag_volt_err_comp_use	Flag for whether to use voltage error compensation
	u1_flag_fluxwkn_use	Flag for whether to use flux-weakening

8.5.4 Macro Definitions

Table 8-10 lists macros.

Table 8-10 List of macros

File name	Macro name	Defined value	Remarks
r_app_main.h	MAIN_UI_RMW	0	RMW UI is used
	MAIN_UI_BOARD	1	Board UI is used
	MAIN_UI_SIZE	2	The number of selectable UIs
r_app_rmw.h	ICS_DECIMATION	5	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

Note: A macro that defines a channel for communicating with RMW is provided in ICS2_RX26T.h.

8.5.5 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 variables listed in Table 8-8, perform adjustment and configuration from RMW. For details about how to use RMW, see 6.7and Renesas Motor Workbench V3.10 User's Manual (R21UZ0004).

8.6 Manager Modules and Motor Control Modules

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

8.6.1 Functions

Table 8-11 lists the functions of the manager module. Table 8-12 and Table 8-13 list the functions of the motor control module.

Table 8-11 List 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 and position information acquisition	Acquires the speed and position information from the speed control module and current control module.
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.

Table 8-12 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.
Flux-weakening control	Controls the d-axis current to enable the operation over the rated rotational speed.

Table 8-13 List of functions of the current control module

Functions	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.
Forward 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.
Decoupling control	Calculates interference cancellation to prevent interference between the d and q axes.
Sample delay Compensation	Compensates for sample delay for current control cycles when generating a three-phase voltage command value.
Friction compensation	Compensates for friction at start-up and reduces speed fluctuations at start-up caused by low-resolution Hall sensors.

Table 8-14 List of functions of the sensor module

Function	Description
Hall sensor angle/speed estimation	Estimates the angle and speed based on the input Hall sensor signals.
Speed estimation method selection	Can select method of speed estimation by API.

8.6.2 Module Configuration Diagram

Figure 8-2 shows the module configuration.

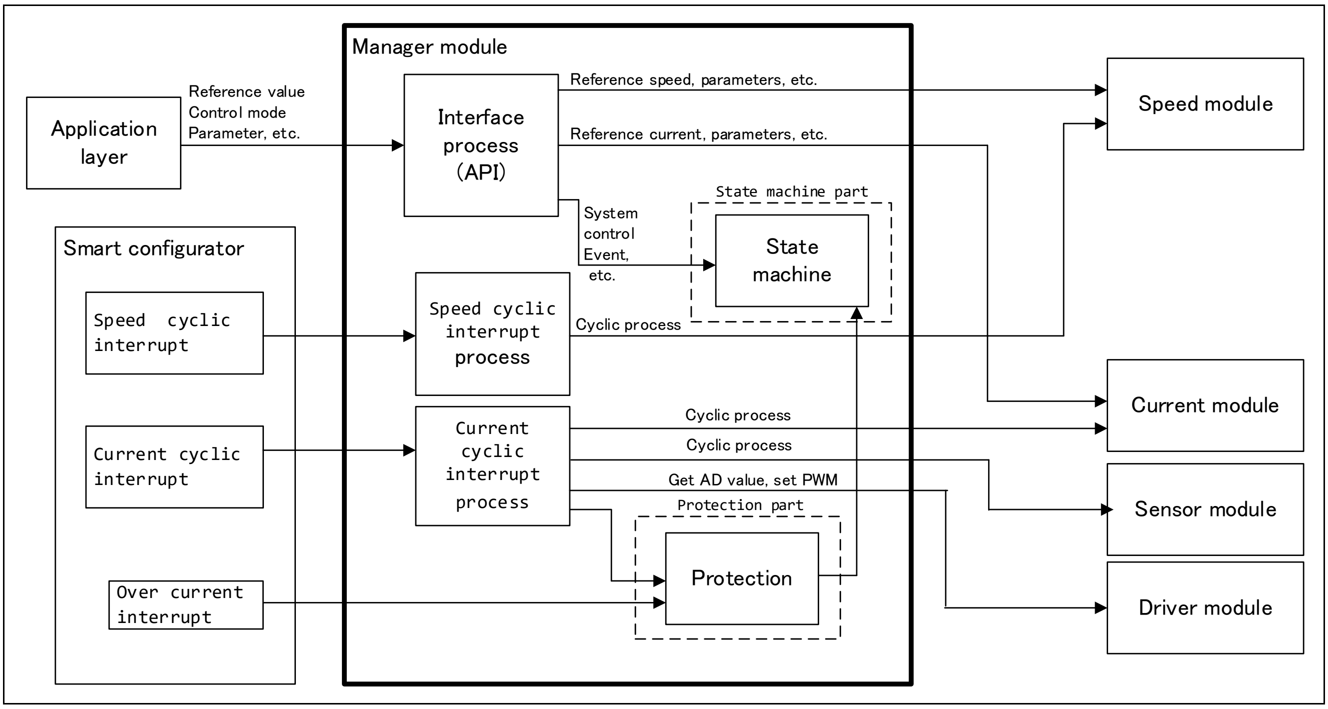
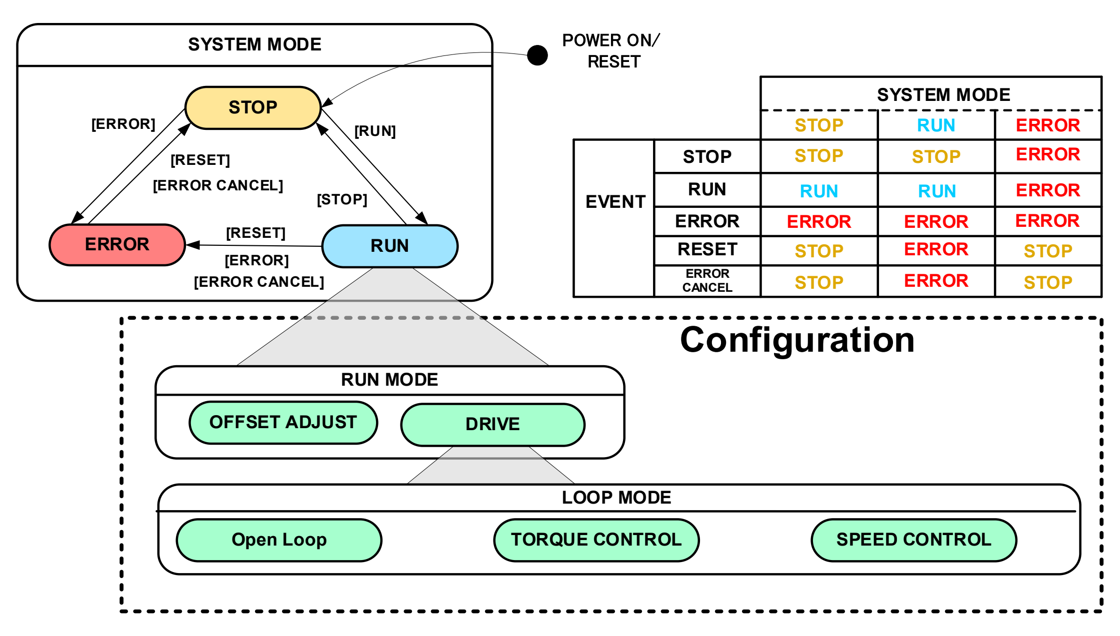


Figure 8-2 Manager module configuration diagram

8.6.3 Mode Management

Figure 8-3 shows the state transition diagram in this sample program. In this sample program, the states are managed by using two types of modes: "SYSTEM MODE" and "RUN MODE".



		SYSTEM MODE		
		STOP	RUN	ERROR
EVENT	STOP	STOP	STOP	ERROR
	RUN	RUN	RUN	ERROR
	ERROR	ERROR	ERROR	ERROR
	RESET	STOP	ERROR	STOP
	ERROR CANCEL	STOP	ERROR	STOP

Figure 8-3 State transition diagram for Hall sensor vector control software

(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).

(2) RUN MODE & LOOP MODE

These modes are used to indicate the motor control state. When the system becomes ACTIVE mode, the motor is driven or calibrated according to the configuration settings shown in Figure 8-3.

(3) EVENT

The matrix table in Figure 8-3 shows how 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 8-15 EVENT LIST

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
ERRORCANCEL	Operation performed by the user

8.6.4 Sequence Description

This sample program has two types of state transitions: one is called the mode and manages ACTIVE, INACTIVE, and ERROR, and the other is called the sequence and manages the operation states for Hall sensor vector control. For more information on the mode, see 8.6.3. This section describes the latter, the sequence which manages the operation states. See the following diagram.

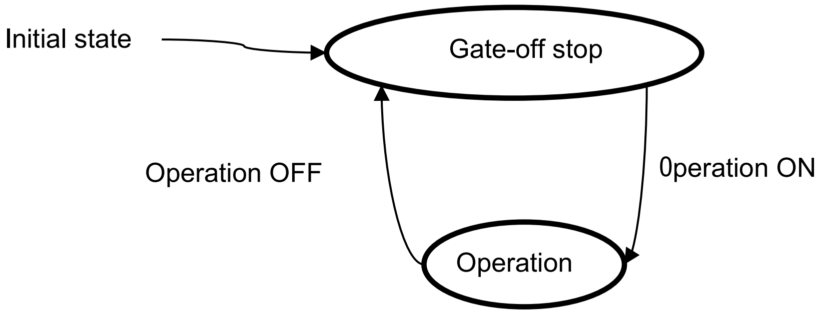


Figure 8-4 State transition diagram for the operation sequence

Table 8-16 Operation sequence states and their description

State	Description
Initial state	This is the state before the CPU is initialized.
Gate-off Stop	This is the state in which the power supply of the CPU board is ON. When the inverter power is OFF, it does not transition to operation ON. If operation is turned on, an error occurs due to undervoltage protection.
Operation	Vector control is performed using a Hall sensor control algorithm.

8.6.5 Start-up

The manager module uses the R_MOTOR_HALL_VECTOR_CtrlTypeSet API to set the control method to be used in rotation or adjust the sensor at start-up. These settings can only be changed in stop state.

The above API sets RUN MODE and LOOP MODE, which determines the input selection for each motor control module and controls the motor accordingly from the next start-up. As an example, Figure 8-5 shows speed control at start-up.

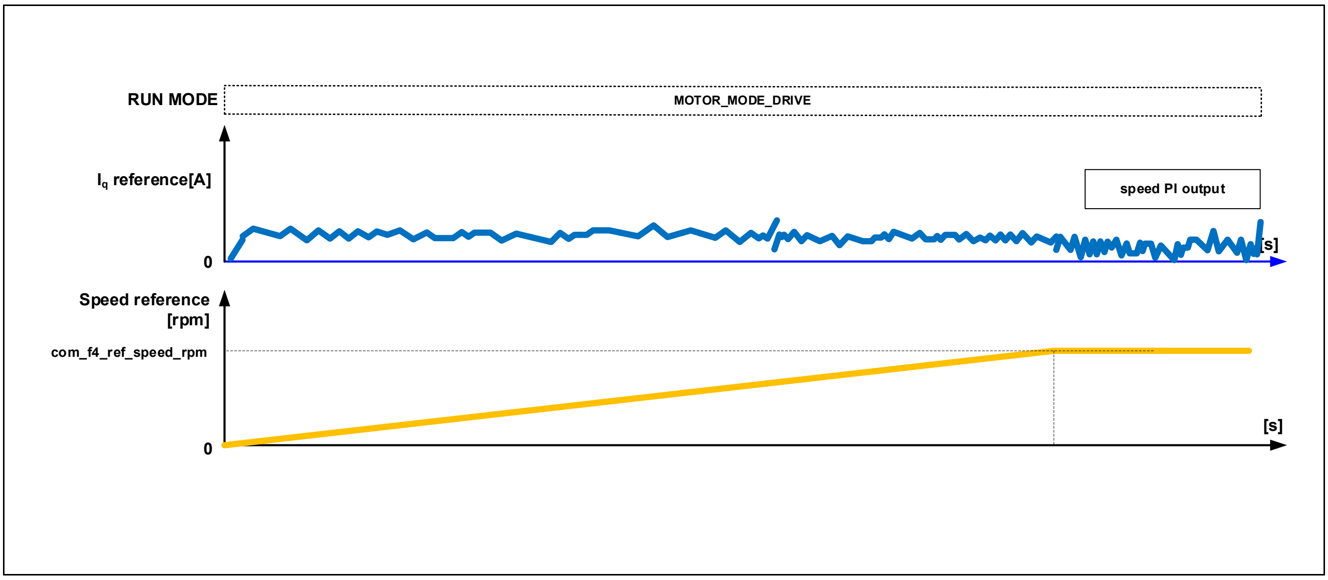


Figure 8-5 Speed control at start-up based on Hall sensor vector control

8.6.6 Protection Function

This control program provides the following error states and implements an emergency stop function in each error case. For details about the values that can specified for the settings of the system protection function, see Table 8-17.

- 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).
  The overcurrent limit value is automatically calculated from the rated current of the motor (MP_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. 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. The low-voltage limit value is preset in consideration of conditions such as an error in the resistor value of the detection circuit.
- Rotational 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.

Table 8-17 Operating conditions and setting values for the system protection functions

Overcurrent error	Overcurrent limit value [A]	10
	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]
Rotational speed error	Speed limit value [r/min]	2850
	Monitoring interval [μs]	Current control interval*[1]

Note: 1. See Table 8-1 Hall sensor vector control software basic specifications

8.6.7 API

Table 8-18 lists the manager module API functions.

Table 8-18 List of API functions

API	Description
R_MOTOR_HALL_VECTOR_Open	Generates an instance of this module and the modules to be used.
R_MOTOR_HALL_VECTOR_Close	Places this module in a reset state.
R_MOTOR_HALL_VECTOR_Reset	Initializes modules.
R_MOTOR_HALL_VECTOR_ParameterUpdate	Updates the control parameter settings of this module. This function also updates the control parameters for the related modules.
R_MOTOR_HALL_VECTOR_MotorStart	Places the motor in the running state.
R_MOTOR_HALL_VECTOR_MotorStop	Places the motor in the stopped state.
R_MOTOR_HALL_VECTOR_MotorReset	Releases the system from the error state and resets the parameters.
R_MOTOR_HALL_VECTOR_MotorCancel	Releases the system from the error state without resetting the parameters.
R_MOTOR_HALL_VECTOR_ErrorSet	Places the system in an error state.
R_MOTOR_HALL_VECTOR_SpeedSet	Sets the speed command value. This function is enabled when speed control is being performed.
R_MOTOR_HALL_VECTOR_SpeedGet	Acquires the speed information.
R_MOTOR_HALL_VECTOR_StatusGet	Acquires the status from the state machine.
R_MOTOR_HALL_VECTOR_ErrorStatusGet	Acquires the error state.
R_MOTOR_HALL_VECTOR_CtrlTypeSet	Sets the control method. To change the control method, place the motor in the stopped state. The control method is defined in the enumeration type e_mtr_ctrl_type_t. 0: Position control (not available) 1: Speed control 2: Torque control 3: Voltage control (not available) 4: Current control (position open-loop control) 5: Current offset adjustment
R_MOTOR_HALL_VECTOR_LoopModeStatusGet	Acquires the control method. The result follows the definition of the enumeration type e_mtr_ctrl_type_t.
R_MOTOR_HALL_VECTOR_RotorAngleOffsetSet	Set rotor angle offset
R_MOTOR_HALL_VECTOR_CurrentOffsetSet	Set AD current offset
R_MOTOR_HALL_VECTOR_CurrentOpenLoopSet	Set commands for constant current open-loop
R_MOTOR_HALL_VECTOR_TorqueSet	Set torque command for torque control mode
R_MOTOR_HALL_VECTOR_OffsetCalibrationStatusGet	Get offset calibration status

API	Description
R_MOTOR_HALL_VECTOR_DriverSet	Set the driver function for hall sensor
R_MOTOR_HALL_VECTOR_SpeedInterrupt	Performs interrupt processing for speed control.
R_MOTOR_HALL_VECTOR_CurrentInterrupt	Performs interrupt processing for current control.
R_MOTOR_HALL_VECTOR_OverCurrentInterrupt	Performs interrupt processing when an overcurrent occurs.

Table 8-19 List of API functions for the current control modules

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	Set the voltage error compensation parameters.
R_MOTOR_CURRENT_CurrentOffsetSet	Set the AD current offset.

Table 8-20 List of API functions for the speed control modules

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_FrictionParameterUpdate	Updates the parameters of the friction compensation.

8.6.8 Structure and Variable Information

Table 8-21 lists the structures and variables for the Manager module. For the manager module, the structure for the manager module (g_st_hall_vector) is defined by securing an instance of the module from the API. Table 8-22 lists the structures and variables that are used for the current control module. Table 8-23 lists the structures and variables for the speed control module. For the current control module and speed control module, the structure for the current control module (g_st_cc) and the structure for the speed module (g_st_sc) are defined by securing an instance of the modules from the API.

Table 8-21 List of structures and variables for the manager module

Structure	Variable	Description
st_hall_vector_control_t Structure for the manager module	u1_state_id_ref	State of the d-axis current command value: 0: Zero command 1: Id command input 2: Manual command input
	u1_state_iq_ref	State of the q-axis current command value: 0: Zero command 1: Iq command input 2: Manual command input
	u1_state_speed_ref	State of the speed command value: 0: Zero command 1: Speed command input 2: Manual command input
	u1_state_open_loop	Open loop state: 0: Calculates the control phase of current control based on the position from the position sensor (closed-loop) 1: Calculates the control phase of current control based on the command speed (open-loop)
	u1_direction	Rotation direction 0: CW 1: CCW
	u1_ctrl_loop_mode	Control loop mode: 0: d-axis current control 1: q-axis current control 2: Speed control 3: Position control
	u1_offset_adjust_mode	Offset removal mode: 0: Current offset 1: Position offset 2: Completion
	u1_flag_offset_calc	Offset calculation flag
	u2_error_status	Error status
	u2_error_status_mask	Error status bit mask, the error that the corresponding bit is set 0 will be ignored
	u2_run_mode	Motor driving mode: 0: Offset removal 1: Motor driving
	u1_flag_charge_cap	Voltage stability flag
	f4_vdc_ad	Bus voltage [V]
	f4_iu_ad	U-phase current [A]

Structure	Variable	Description
	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]
	f4_user_open_loop_current	User open loop mode current command [A]
	f4_user_torque_nm	User torque command [Nm]
	f4_open_loop_angle_rad	Current control phase for open loop control [rad]
	f4_rotor_angle_rad	Current control phase [rad]
	f4_rotor_angle_offset_rad	Position offset [rad]
	HallSignalGet	Pointer to the function that gets hall signal Example: **void** HallSignalGetFunction (uint8_t * u1_bit_u, uint8_t * u1_bit_v, uint8_t * u1_bit_w);
	st_current_output	Structure for the current control module output
	st_speed_output	Structure for the speed control module output
	st_sensor_output	Structure for the sensor module output
	st_stm	Structure for state machine
	st_motor	Motor parameter structure
	*p_st_driver	Instance for the driver module
	*p_st_cc	Instance for the current control module
	*p_st_sc	Instance for the speed control module
	*p_st_sensor	Instance for the sensor module
st_hall_vector_cfg_t Structure for setting control parameters in the manager module	f4_overspeed_limit_rpm	Speed limit value [r/min] (mechanical angle)
	st_motor	Motor parameter structure

Table 8-22 List of structures and variables for current control modules

Structure	Variable	Description
st_current_control_t Structure for current control module	u1_active	The active state of the current control module
	u1_flag_volt_err_comp_use	Enables/disables voltage error compensation function
	u1_state_id_ref	Status of the d-axis input
	u1_state_iq_ref	Status of the q-axis input
	u1_flag_offset_calc	Flags for current offset calculation
	u2_offset_calc_time	Measurement time setting in current offset adjustment
	u2_offset_calc_wait	Current offset adjustment timing counter
	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 [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
	f4_modw	W-phase duty cycle
	f4_speed_rad	Speed [rad/s]
	f4_rotor_angle_input_rad	Rotor angle [rad]

Structure	Variable	Description
	f4_id_ref_manual	d-axis current status: d-axis current command value in d-axis fixed-command mode [A]
	f4_iq_ref_manual	q-axis current status: q-axis current command value in q-axis fixed-command mode [A]
	f4_ref_id_ctrl	d-axis current command value [A]
	f4_ref_iq_ctrl	q-axis current command value [A]
	f4_current_rate_limit	Current increase speed limit value
	f4_va_max	Maximum voltage on the d- and q- axes [V]
	st_mod	Structure for modulation
	st_volt_comp	Structure for the voltage error compensation
	st_pi_id	Structure for the d-axis PI control
	st_pi_iq	Structure for the q-axis PI control
	st_rotor_angle	Rotor information structure
	st_rotor_angle_phasecomp	Rotor Information structure (lead compensation)
	st_motor	Structure for motor parameters
st_current_cfg_t Structure for setting current control module control parameters	u2_offset_calc_time	Offset calculation time setting
	f4_ctrl_period	Control interval [s]
	f4_current_omega_hz	Natural frequency for current control system [Hz]
	f4_current_zeta	Attenuation coefficient for current control system
	u1_flag_volt_err_comp_use	Enables/disables voltage error compensation
	st_motor	Structure for motor parameters
st_current_output_t Structure for the output of the current control module	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_input_t Structure for current control module input	u1_state_id_ref	Status of the d-axis
	u1_state_iq_ref	Status of the q-axis
	f4_rotor_angle_rad	Rotor angle [rad]
	f4_iu_ad	U-phase current value [A]
	f4_iv_ad	V-phase current value [A]
	f4_iw_ad	W-phase current value [A]

Structure	Variable	Description
	f4_vdc_ad	Bus voltage value [V]
	f4_speed_rad	Speed [rad/s]
	f4_id_ref	d-axis current command value [A]
	f4_iq_ref	q-axis current command value [A]
	f4_id_ref_manual	d-axis current status: d-axis current command value in d-axis fixed-command mode [A]
	f4_iq_ref_manual	q-axis current status: q-axis current command value in q-axis fixed-command mode [A]

Table 8-23 List of structures and variables for the speed control module 1

Structure	Variable	Description
st_speed_control_t Structure for speed 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
	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 speed control module during speed 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]
	st_motor	Structure for motor parameters
	st_pi_speed	Structure for PI control
	st_fluxwkn	Structure for magnetic flux-weakening control
	st_friction_comp	Structure for friction compensation

Table 8-24 List of structures and variables for the speed control module 2

Structure	Variable	Description
st_speed_cfg_t Structure for setting speed module control parameters	st_motor	Structure for motor parameters
	u1_flag_fluxwkn_use	Flag for whether to use magnetic flux-weakening control
	f4_max_speed_rpm	Speed control interval [s]
	f4_speed_ctrl_period	Speed variation limit value [r/min] (mechanical angle)
	f4_speed_rate_limit_rpm	Speed variation limit value [rpm]
	f4_speed_omega_hz	Natural frequency for speed control system [Hz]
	f4_speed_zeta	Attenuation coefficient for the speed control system
st_speed_input_t Structure for speed module input	u1_state_speed_ref	Speed command status
	f4_ref_speed_rad	Speed command value [rad/s].
	f4_speed_rad	Input speed [rad/s]
	f4_va_max	Maximum voltage on the d- and q- axes [V]
st_speed_output_t Structure for speed 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 used for PI control [rad/s]

8.6.9 Macro Definitions

Table 8-25 lists the macros for the manager module.

Table 8-25 List of macros

File name	Macro name	Defined value	Remarks
r_motor_hall_vector_api.h	MOTOR_LOOP_CURRENT_ID	0	Magnetic flux control mode
	MOTOR_LOOP_CURRENT_IQ	1	Torque control mode
	MOTOR_LOOP_SPEED	2	Speed control mode
	MOTOR_LOOP_POSITION	3	Position control mode. Not available in this sample program.
	MOTOR_ERROR_ NONE	(0x0000)	Error status: There is no error.
	MOTOR_ERROR_ OVER_CURRENT_HW	(0x0001)	Error status: A hardware overcurrent error has occurred.
	MOTOR_ ERROR_ OVER_VOLTAGE	(0x0002)	Error status: An overvoltage error has occurred.
	MOTOR _ERROR_ OVER_SPEED	(0x0004)	Error status: An overspeed error has occurred.
	MOTOR _ERROR_ UNDER_VOLTAGE	(0x0080)	Error status: An under-voltage error has occurred.
	MOTOR _ERROR_ OVER_CURRENT_SW	(0x0100)	Error status: A software overcurrent error has occurred.
	MOTOR_ERROR_ OVER_TEMPERATURE	(0x0200)	Error status: An inverter overtemperature error has occurred.
	MOTOR_ERROR_ UNKNOWN	(0xffff)	Error status: An error whose error code is unknown has occurred.
r_motor_hall_vector_manager.h	MOTOR_HALL_VECTOR_MODE_ OFFSET_ADJUST	(0x00)	The operation mode for initialization
	MOTOR_HALL_VECTOR_MODE_ DRIVE	(0x01)	The operation mode with the motor driven
	MOTOR_HALL_VECTOR_OFAJ_CURRENT_OFFSET	(0x00)	The operation mode for current offset
	MOTOR_HALL_VECTOR_OFAJ_FIN	(0x01)	Offset adjustment completed
r_motor_hall_vector_api.h	MOTOR_CTRL_TYPE_ POS	0	Macro for switching the control method. Position control mode. Not available in this sample program.
	MOTOR_CTRL_TYPE_ SPEED	1	Macro for switching the control method. Speed control mode.
	MOTOR_CTRL_TYPE_ TORQUE	2	Macro for switching the control method. Torque control mode.
	MOTOR_CTRL_TYPE_ VOLTAGE	3	Not used
	MOTOR_CTRL_TYPE_ CURRENT	4	Macro for switching the control method. Open loop current control.
	MOTOR_CTRL_TYPE_ ADJ_CRNT_OFFSET	5	Macro for switching the control method. Current offset adjustment mode.

8.7 Driver Modules

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

8.7.1 Functions

Table 8-26 lists the functions of the driver module.

Table 8-26 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.

8.7.2 Module Configuration Diagram

Figure 8-6 shows the module configuration of the driver module.

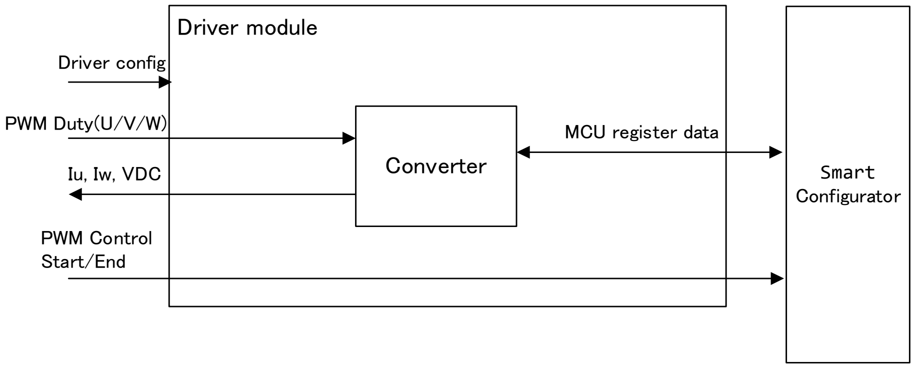


Figure 8-6 Driver module configuration diagram

8.7.3 API

Table 8-27 lists and describes the API functions for the driver module.

Table 8-27 List of API for driver modules

API	Description
R_MOTOR_DRIVER_Open	Generates an instance of the driver module.
R_MOTOR_DRIVER_Close	Place 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.

8.7.4 Configurations

Table 8-28 List of configuration information lists the configuration for the driver module. Set up the functions to be used and the necessary parameters.

Table 8-28 List of configuration information

File name	Macro name	Set value	Description
r_motor_module_cfg.h	DRIVER_CFG_FUNC_PWM_OUTPUT_START	R_Config_xxx_StartTimerCtrl (Smart Configurator function) *1 *2	Sets the function that enables PWM output.
	DRIVER_CFG_FUNC_PWM_OUTPUT_STOP	R_Config_xxx_StopTimerCtrl (Smart Configurator function) *1 *2	Sets the function that disables PWM output.
	DRIVER_CFG_FUNC_ADC_DATA_GET	R_Config_xxx_AdcGetConvVal (Smart Configurator function) *1 *2	Sets the function that acquires the A/D conversion results.
	DRIVER_CFG_FUNC_DUTY_SET	R_Config_xxx_UpdDuty (Smart Configurator function) *1 *2	Sets the function that sets the duty cycle.
r_motor_inverter_cfg.h	INVERTER_CFG_ADC_REF_VOLTAGE	5.0f	Sets the reference voltage for A/D conversion.
r_motor_module_cfg.h	MOTOR_MCU_CFG_ADC_OFFSET	0x7FF	Sets the AD offset value.

Note: 1. For details about the functions shown in the “Set value”, see 10 Smart Configurator Settings .

2. When the Smart Configurator Motor component is used, “xxx” is set to “MOTOR”. When the Motor component is not used, it is the module name used for PWM.

8.7.5 Structure and Variable Information

Table 8-29 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 8-29 Structure list of 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)
st_motor_driver_cfg_t Structure for setting the parameters for controlling driver 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)

8.7.6 Adjustment and Configuration of Parameters

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_config_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 8-30 shows the settings.

Table 8-30 Example of settings specified in the sample program

Variable name	Macro name	File name
*ADCDataGet	DRIVER_CFG_FUNC_ADC_DATA_GET	r_motor_module_cfg.h
*BLDCDutySet	DRIVER_CFG_FUNC_DUTY_SET	
*PWMOutputStop	DRIVER_CFG_FUNC_PWM_OUTPUT_START	
*PWMOutputStart	DRIVER_CFG_FUNC_PWM_OUTPUT_STOP	
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	r_motor_module_cfg.h
f4_pwm_dead_time_cnt	MOTOR_COMMON_DEADTIME_SET	

9. Setting Parameters

9.1 Overview

In this sample program, parameters are defined as macros in the header files shown below. The parameters defined as the macros are set to the variables and structures managed by each function module in the initialization routine at startup and used for the respective processes.

Some parameters can be changed dynamically from RMW or other tool. When changes are made, the parameter update function must be called to reflect the changes. For details, see the description of each function module.

Table 9-1 List of parameter setting files

Macro name	Description
r_motor_module_cfg.h	Defines initial values of parameters related to motor control.
r_motor_inverter_cfg.h	Defines the initial values of parameters related to the inverter.
r_motor_targetmotor_cfg.h	Defines the initial values of parameters related to the motor.

9.2 MCU-related Parameters

Table 9-2 shows a list of parameters that are related to peripheral functions of the MCU. If the peripheral settings of the MCU are changed, these parameters must be changed wherever applicable.

Table 9-2 List of MCU-related parameters

File name	Macro name	Setting value	Description
r_motor_module_cfg.h	MOTOR_MCU_CFG_PWM_TIMER_FREQ	120.0f	PWM timer frequency [MHz]
	MOTOR_MCU_CFG_CARRIER_FREQ	20.0f	Carrier wave frequency [kHz]
	MOTOR_MCU_CFG_INTR_DECIMATION	0	Skipping count for carrier wave interrupts
	MOTOR_MCU_CFG_AD_FREQ	60.0f	ADC operating frequency [MHz]
	MOTOR_MCU_CFG_AD_SAMPLING_CYCLE	2.0f * (7.25f + 63.0f)	ADC sampling cycle [cycle]
	MOTOR_MCU_CFG_AD12BIT_DATA	4095.0f	ADC resolution
	MOTOR_MCU_CFG_ADC_OFFSET	0x7FF	ADC offset value

9.3 List of Control Function Setting Parameters

Table 9-3, Table 9-4, and Table 9-5 show the parameters that enable or disable the functions provided by the motor control software. Items related to motor constants or settings used internally in motor control are described later.

Table 9-3 List of operational parameters (general)

File name	Macro name	Set value	Description
r_motor_module_cfg.h	MOTOR_TYPE_BLDC	MOTOR_TYPE_BLDC	Use as default.
	MOTOR_COMMON_CFG_LOOP_MODE	MOTOR_LOOP_SPEED	Use as default.
	MOTOR_COMMON_CFG_OVERCURRENT_MARGIN_MULT	2.0f	Limit coefficient for overcurrent
	MOTOR_COMMON_CFG_IA_MAX_CALC_MULT	MTR_SQRT_3	Coefficient for calculating the overcurrent limit value Set to √3.
	MOTOR_COMMON_CFG_AUTO_OFFSET_ADJUST_DELAY	(0.5f)	After power on, the delay [s] before starting to perform current offset adjustment.
	MOTOR_MCU_CFG_TFU_OPTIMIZE	MTR_ENABLE	Sets the processing of TFU-specific functions. This parameter will be set automatically according to CC-RX compile option -tfu

Table 9-4 List of operational parameters (related to speed control))

File name	Macro name	Set value	Description
r_motor_module_cfg.h	SPEED_CFG_FLUX_WEAKENING	MTR_ENABLE	Enables or disables the flux-weakening control. Enable: MTR_ENABLE Disable: MTR_DISABLE
	SPEED_CFG_CTRL_PERIOD	0.0005f	Sets the speed control interval [sec]. Set to 0.0005f to make it 0.5ms.

Table 9-5 List of operational parameters (related to current control))

File name	Macro name	Set value	Description
r_motor_module_cfg.h	CURRENT_CFG_VOLT_ERR_COMP	MTR_ENABLE	Enables or disables the voltage error compensation function. Set to MTR_ENABLE.
	CURRENT_CFG_MODULATION_METHOD	MOD_METHOD_SVPWM	See 9.5. Normally, set to MOD_METHOD_SVPWM.
	CURRENT_CFG_OFFSET_CALC_TIME	512.0f	Sets the current offset measurement time.
	CURRENT_CFG_OFFSET_CALC_WAIT	4000	Current offset detection timing

9.4 Protection-related Parameters

The following shows the parameters for the protection functions to ensure safety when operating the motor.

Table 9-6 Motor and inverter parameter settings

File name	Macro name	Set value	Description
r_motor_inverter_cfg.h	INVERTER_CFG_CURRENT_LIMIT	21.4f	Overcurrent limit value for the inverter board [A]
	INVERTER_CFG_OVERVOLTAGE_LIMIT	60.0f	Overvoltage limit [V]
	INVERTER_CFG_UNDERVOLTAGE_LIMIT	8.0f	Low-voltage limit [V]

INVERTER_CFG_CURRENT_LIMIT

Set the current value with a safety margin from the maximum current value that can be output by the inverter.

INVERTER_CFG_OVERVOLTAGE_LIMIT

Sets the voltage at which the overvoltage protection is activated. If the inverter bus voltage exceeds the set voltage, an error occurs and the motor stops operating. Set an appropriate value according to the power supply environment used.

INVERTER_CFG_UNDERVOLTAGE_LIMIT

Sets the voltage at which the undervoltage protection is activated. If the inverter bus voltage falls below the set voltage, an error occurs and the motor stops operating. Set an appropriate value according to the power supply environment used.

9.5 Changing the PWM Carrier Frequency

The PWM carrier frequency is set by the Smart Configurator and by the MOTOR_MCU_CFG_CARRIER_FREQ constant defined in r_motor_module_cfg.h. If the PWM carrier frequency is changed, modify the locations shown in Table 9-7. Some parameters need to be adjusted to match the set value of the PWM carrier frequency.

The default PWM carrier frequency for this sample program is 20 kHz.

Table 9-7 Parameters to be changed when the PWM carrier frequency is changed

Item	Location where change is required
Dead time value	See “9.7 Inverter Parameters”.
Carrier frequency	- In Smart Configurator motor settings, set the PWM carrier frequency - MOTOR_MCU_CFG_CARRIER_FREQ described in 9.2
Motor control-related	Parameters of the current regulator Parameters of Hall sensor control

9.6 Setting the Pulse Modulation Method

In this sample program, one of two pulse-width modulation drive modes can be selected. The default is the spatial vector PWM (MOD_METHOD_SVPWM). Table 9-8 lists the configurations for the modulation function.

If the pulse-width modulation drive mode is changed to sinusoidal PWM, the voltage utilization is limited to 86%, whereby the appropriate voltage cannot be output to the motor, and the inverter bus voltage must be set higher to obtain the desired voltage. When space vector PWM is used, the voltage utilization is 100% with respect to the inverter bus voltage.

Table 9-8 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

Table 9-9 Setting items for the pulse-width modulation drive mode

Setting item for pulse-width modulation drive mode	Value	Pulse-width modulation drive mode
MOD_METHOD_SPWM	0	Sinusoidal PWM
MOD_METHOD_SVPWM	1	Spatial vector PWM

Pulse modulation has the following setting items. Normally, use the default values.

Table 9-10 List of macros

File name	Macro name	Set value	Description
r_motor_current_modulation.h	MOD_DEFAULT_MAX_DUTY	1.0f	Maximum PWM duty cycle. Normally, leave the value at 1.0f.
	MOD_VDC_TO_VAMAX_MULT	0.6124f	Coefficient of the conversion to obtain the maximum possible output voltage at the inverter bus voltage.
	MOD_SVPWM_MULT	1.155f	Only when spatial vector PWM is used. Coefficient for spatial vector PWM

9.7 Inverter Parameters

9.7.1 Overview

When you use the sample program, you need to correctly set the inverter information. Table 9-11shows the inverter parameters set in the sample program.

Table 9-11 Inverter parameter setting

File name	Macro name	Set value	Description
r_motor_inverter_cfg.h	INVERTER_CFG_SHUNT_RESIST	0.010f	Shunt resistance value [ohm]
	INVERTER_CFG_DEADTIME	2.0f	Dead time [μs]
	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_INPUT_V	24.0f	Input voltage [V]
	INVERTER_CFG_ADC_REF_VOLTAGE	5.0f	Analog power supply voltage for the MCU [V]
	INVERTER_CFG_COMP_V0	0.564f	Coefficient for compensation of the voltage error [V]
	INVERTER_CFG_COMP_V1	0.782f	
	INVERTER_CFG_COMP_V2	0.937f	
	INVERTER_CFG_COMP_V3	1.027f	
	INVERTER_CFG_COMP_V4	1.058f	
	INVERTER_CFG_COMP_I0	0.022f	
	INVERTER_CFG_COMP_I1	0.038f	
	INVERTER_CFG_COMP_I2	0.088f	
	INVERTER_CFG_COMP_I3	0.248f	
	INVERTER_CFG_COMP_I4	0.865f	

INVERTER_CFG_DEADTIME

Specify the dead time in μs (microseconds) that is described in the inverter specifications and design document. For the MCK-RX26T motor control kit, 2.0 μs is specified.

INVERTER_CFG_INPUT_V

The default voltage value is 24 V DC.

INVERTER_CFG_ADC_REF_VOLTAGE

Specify the analog voltage of the MCU. In the RX26T CPU board, 5.0 V is specified.

INVERTER_CFG_COMP_Vx, INVERTER_CFG_COMP_Ix

See 9.7.4.

9.7.2 Current Detection Gain

In the MCI-LV-1 inverter, the voltage value input to the ADC is specified by the amount of the current, as shown in Table 9-12.

To set the current detection gain in this sample program, INVERTER_CFG_CURRENT_AMP_GAIN and INVERTER_CFG_SHUNT_RESIST are used. Here, INVERTER_CFG_ADC_REF_VOLTAGE should be left at 5V.

INVERTER_CFG_CURRENT_AMP_GAIN

Sets a coefficient to determine how much current (A) is equivalent per 1 V voltage input by the ADC.

INVERTER_CFG_SHUNT_RESIST

Set the resistor value used in the shunt resistor. When Hole Current Transform is used without using a shunt resistor, specify 1.0.

INVERTER_CFG_ADC_REF_VOLTAGE

Divide by the reciprocal of INVERTER_CFG_SHUNT_RESIST, and then divide by the current value per 1V. In the MCI-LV-1 specification, which is ±12.5 A at 0-5 V (25A at Peak to Peak), 5A per 1V. Assuming that a shunt resistance is 0.01Ω, the reciprocal is 100. It is calculated as (1/100)*(1/20)=5. Set 5 to INVERTER_CFG_ADC_REF_VOLTAGE.

A generalized formula for determining the INVERTER_CFG_ADC_REF_VOLTAGE is shown below.

$$INVERTER\_CFG\_ADC\_REF\_VOLTAGE = \frac{1}{INVERTER\_CFG\_SHUNT\_RESIST[\Omega]} \times \frac{1}{Current\ per\ 1V[A/V]}$$

Table 9-12 Current signal specifications for MCI-LV-1

3-phase output current	ADC input voltage value	ADC conversion values
+12.5A	5V	4095
0A	2.5V	2048
-12.5A	0V	0

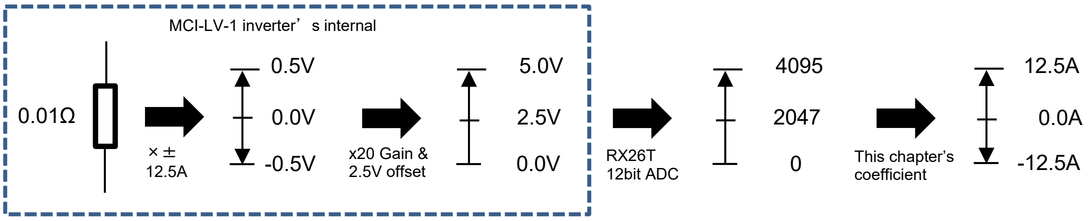


Figure 9-1 Current detection calculation flow

9.7.3 Voltage Detection Gain

The voltage detection gain is set by INVERTER_CFG_VOLTAGE_GAIN.

Set a coefficient to determine how much inverter bus voltage (V) is equivalent per 1 V of voltage input by the ADC. If the voltage of 5V input to ADC is equivalent to 111.383V, 111.383/5=22.2766. Therefore, set 22.2766 to INVERTER_CFG_VOLTAGE_GAIN.

$$\mathrm{INVERTER\_CFG\_VOLTAGE\_GAIN} = \frac{Reference\ of\ Inverter\ bus\ voltage}{Reference\ of\ ADC\ input\ voltage} = \frac{111.383}{5} = 22.2766$$

Table 9-13 Inverter bus voltage signal specifications for MCI-LV-1

Inverter bus voltage value	ADC input voltage value	ADC conversion value
0V	0V	0
111.383V	5V	4095

9.7.4 Voltage Error Compensation Parameters

This section describes how to use and set the voltage error compensation function. The following three settings are required.

① Selecting a dead time value

The dead time value is determined by the characteristics of the power semiconductors used in the inverter. In this sample software, it is set to 2.0μs according to the characteristics of the inverter board (RTK0EM0000B12020BJ) included in the kit. Reflect the selected dead time value in an input location provided in the Smart Configurator's motor settings.

② 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.

③ Setting the voltage compensation table

Performing a switching test using an actual inverter or creating a voltage compensation table using the voltage error value for dead time obtained from the relationship between dead time and carrier period. The relationship between current and voltage obtained in the switching test yields a configurable value for a more effective voltage compensation table.

The limit of the compensation voltage value can be calculated by the following formula.

Compensation voltage limit = (carrier frequency [kHz] × dead time [us] ÷ 1000) × bus-bar voltage value

If the relationship between Iu and Vu (Iv-Vv, Iw-Vw) cannot be obtained experimentally for the slope near the zero cross, it is necessary to calculate the table by considering the above equation and the characteristics of the main circuit.

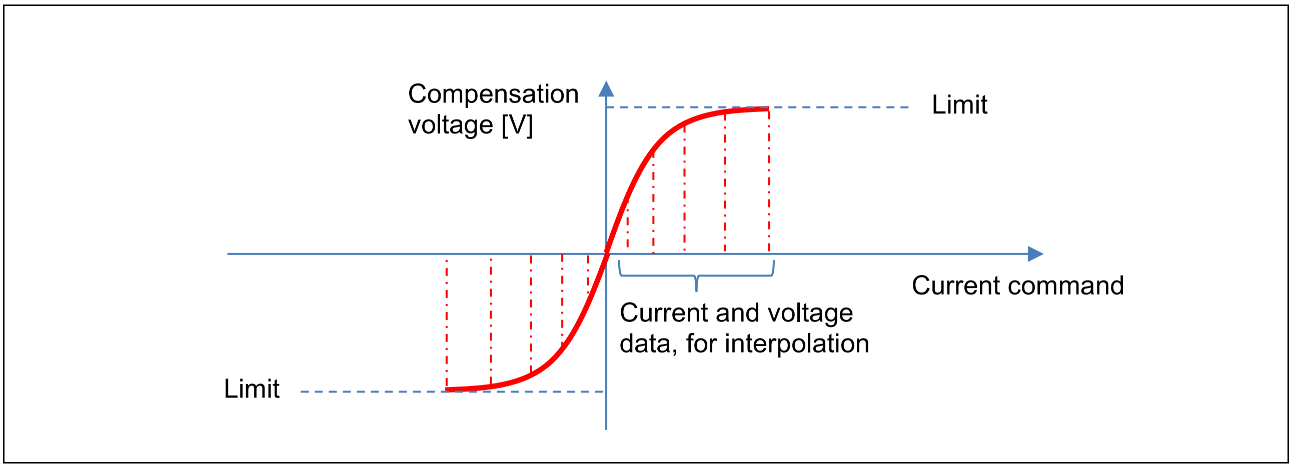


Figure 9-2 Relationship between the compensation voltage value, limit, and current command value

Table 9-14 Interpolation current and voltage data for different carrier cycles

Career cycle		20kHz
	Iu	ΔVu
0	0.00	0.00
1	0.022	0.564
2	0.038	0.782
3	0.088	0.937
4	0.248	1.027
5	0.865	1.058

9.8 Motor Parameters

If motor parameter information is not available from the motor manufacturer, motor parameters R, Ld, and Lq can be obtained simply by using an LCR meter. The inductive voltage can also be obtained simply by using an oscilloscope. The method described here is a simplified method that does not take into account magnetic saturation or other phenomenon and is intended to turn the motor quickly, being subject to individual differences and measurement errors. Therefore, when using the parameters in actual product development, measurement should be performed using measurement equipment whose accuracy is ensured.

The LCR meter should be calibrated periodically, and measurement should be made in a warm-up complete state after at least 30 minutes of power on. In addition, perform open compensation and short compensation in advance to reduce probe errors using the 4-terminal pair method. For details, refer to the LCR meter's instruction manual.

When you use the sample program, you need to correctly set the inverter information and the information about the motor to be used. Table 9-15 shows the values set in the sample program.

Table 9-15 Motor parameter setting

File name	Macro name	Set value	Description
r_motor_targetmotor_cfg.h	MOTOR_CFG_POLE_PAIRS	4	Number of pole pairs
	MOTOR_CFG_MAGNETIC_FLUX	0.01119f	Magnetic flux [wb]
	MOTOR_CFG_RESISTANCE	1.3f	Resistance [ohm]
	MOTOR_CFG_D_INDUCTANCE	0.0013f	d-axis inductance [H]
	MOTOR_CFG_Q_INDUCTANCE	0.0013f	q-axis inductance [H]
	MOTOR_CFG_ROTOR_INERTIA	0.000003666f	Rotor inertia [kg m^2]
	MOTOR_CFG_NOMINAL_CURRENT_RMS	1.67f	Rated current [A]
	MOTOR_CFG_MAX_SPEED_RPM	2400.0f	Maximum speed [r/min]

MOTOR_CFG_POLE_PAIRS

Set the number of pole pairs of the PM motor. The number of pole pairs is 1/2 the number of poles. Refer to the PM motor specifications.

MOTOR_CFG_RESISTANCE

For wiring when measuring with an LCR meter, select two of the motor's three-phase output wires U, V, and W and connect the probes to them. To measure the resistance, use the DC resistance (DCR) mode. Because the resistance value obtained is the composite resistance of the two phases, the resistance value of the motor for one phase can be obtained by halving it. Set the obtained resistance R to MOTOR_CFG_RESISTANCE in r_motor_targetmotor_cfg.h. The unit is Ω.

MOTOR_CFG_D_INDUCTANCE, MOTOR_CFG_Q_INDUCTANCE

For wiring when measuring with an LCR meter, select two of the motor's three-phase output wires U, V, and W and connect the probes to them. For the measurement mode, use the series equivalent circuit mode (Ls). For detailed measurement methods, refer to the LCR meter's instruction manual.

Turn the axis slowly and note down the maximum and minimum inductance values that are displayed. Here, 1/2 of the maximum value is Lq and 1/2 of the minimum value is Ld.

Set the obtained Ld and Lq to MOTOR_CFG_D_INDUCTANCE and MOTOR_CFG_Q_INDUCTANCE in r_motor_targetmotor_cfg.h. The unit is H (henry).

MOTOR_CFG_ROTOR_INERTIA

Specify the inertia (moment of inertia) of the motor's rotor and shaft. The unit is kg m^2. Usually, you can find a description in the documentation provided with the motor. If a load is installed, inertia on the load side should also be added to the setting.

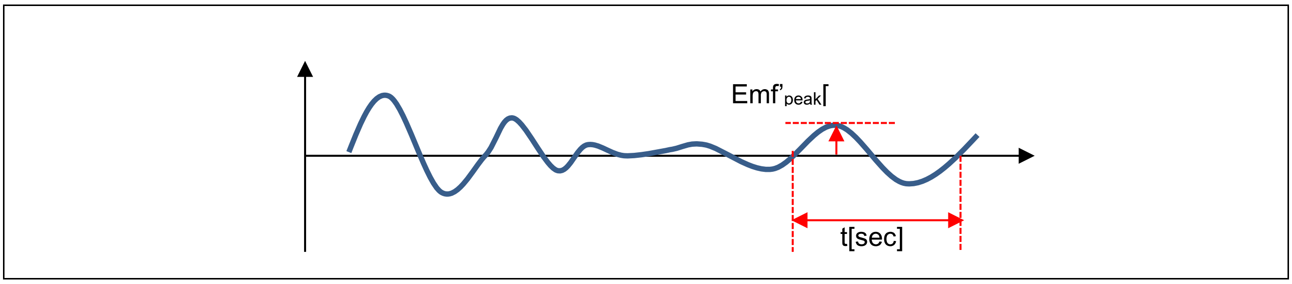
MOTOR_CFG_NOMINAL_CURRENT_RMS

Specify the motor's rated current (RMS). The unit is ampere. It is indicated on the nameplate of the motor or in the accompanying documentation.

MOTOR_CFG_MAGNETIC_FLUX

Select two of the motor's three-phase output wires U, V, and W and connect them to the oscilloscope. For example, connect the oscilloscope probes to the U and V phases so that the voltages can be measured. The U-V phase line voltage value can be obtained by connecting a motor that can rotate at the rated speed to the end of the motor shaft and rotating it at the rated speed. Dividing the line voltage value by √3 gives the peak value of inductive voltage per phase. You can obtain the magnetic flux linkage Ψ from the equation of the inductive voltage = ωΨ. Convert the rated speed to the electrical angular speed frequency f [Hz], substitute ω with 2πf to make the inductive voltage = 2πfΨ, rearrange the equation, substitute a value to obtain the magnetic flux linkage Ψ[Wb].

In cases where a motor cannot be mounted on the end of the shaft, a simple method of obtaining voltage waveforms by quickly rotating the motor by hand can also be used. However, accuracy cannot be guaranteed, and the method can only be used for test run purposes. When turned by hand, the voltage waveform similar to the following image is obtained. In this case, select a cycle close to a constant speed with a sine wave, and find the peak and cycle of the voltage.



In this algorithm, the peak value must be converted to an RMS value. Therefore, divide it by √2 to obtain the RMS value, Emf'rms.

$$Emf'_{rms}[V] = Emf'_{peak}[V] \times \frac{1}{\sqrt{2}}$$

To convert the obtained time t [sec] to Hz, apply the formula f'=1/t. Find the ratio of the obtained f [Hz] to the electrical angular frequency [Hz] obtained from the rated speed of this PM motor, and multiply the voltage Emf'rms[V] obtained simultaneously by the ratio.

$$Emf[V] = Emf'_{rms}[V] \times \frac{\text{electrical angular frequency}[Hz]}{f'[Hz]}$$

As a result, the inductive voltage [V] that is generated when this PM motor rotates at its rated speed can be determined simply. To determine the inductive voltage, it must be measured by rotating the motor shaft at the rated speed using a load test device.

Next, the magnetic flux linkage Ψ[Wb] is obtained from the inductive voltage. In general, inductive voltage and magnetic flux linkage have the relationship as below. f is the electric angular frequency [Hz] at rated speed.

$$Emf[V] = \omega\Psi = 2\pi f\Psi$$

The magnetic flux linkage Ψ[Wb] can be obtained by rearranging the equation and substituting the inductive voltage Emf[V] obtained above and the electric angular frequency [Hz] during rated speed operation.

$$\Psi = \frac{Emf[V]}{2\pi f}$$

Specify the obtained magnetic flux linkage Ψ to MOTOR_CFG_MAGNETIC_FLUX in r_motor_targetmotor_cfg.h.

9.9 Current Control Parameters

Table 9-16 shows the current control parameters. Current control parameters are calculated based on the motor parameters, PWM carrier frequency, and desired current response performance.

The macros for the current control parameters shown in Table 9-16 are set and reflected in the internal variables at start-up, but if adjustments are needed after start-up, the following two parameters can be adjusted from the RMW. See Table 6-2. Note that not all parameters can be changed.

To set the natural frequency for the current control system, use com_f4_current_omega_hz.

To set the attenuation coefficient for the current control system, com_f4_current_zeta

Table 9-16 List of current control parameters

File name	Macro name	Set value	Description
r_motor_target motor_cfg.h	CURRENT_CFG_OFFSET_CALC_ TIME	512.0f	Current offset measurement time setting
	CURRENT_CFG_OMEGA	300.0 f	Natural frequency for the current control system [Hz]
	CURRENT_CFG_ZETA	1.0f	Attenuation coefficient for the current control system
	CURRENT_CFG_RATE_LIMIT	0.0004f	Limit for rate of change in current command [A/period]

CURRENT_CFG_OFFSET_CALC_TIME

Specify the number of times to measure the offset value when measuring the offset of current detection at start-up. Normally, use the default value.

CURRENT_CFG_OMEGA, CURRENT_CFG_ZETA

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 (PWM carrier frequency). However, in many cases, a lower value may be set with a margin in consideration of noise during position detection and current detection.

For example, if the current control frequency is 20 kHz (current control operates at 50 μs intervals), you can specify a current control system intrinsic frequency of 2kHz because it can be set up to 1/10. In practice, however, a high natural frequency may be too sensitive due to the electrical constants of the motor parameters and is often set at a frequency below 2 kHz (for example from 500 Hz to 1 kHz).

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

CURRENT_CFG_RATE_LIMIT

During torque control, the rate of change in current command value is limited so that it does not increase rapidly. Specify the current value [A] that can change per current control cycle. By default, the current control cycle is 50 us, so a value of 0.0004 [A] allows up to 8 A change per second.

9.10 Speed Control Parameters

Table 9-17 shows the speed control parameters. The values set in this file are applied as initial values at system startup. The macros for the speed control parameters shown in Table 9-17 are set and reflected in the internal variables at start-up, but if adjustments are needed after start-up, the following three parameters are available that can be adjusted from the RMW. See Table 6-2. Note that not all speed control parameters can be changed on RMW.

Natural frequency for speed control system: com_f4_speed_omega_hz

Attenuation coefficient for the speed control system: com_f4_speed_zeta

Friction compensation parameter: com_f4_friction_comp_brake_speed, com_f4_friction_comp_static, com_f4_friction_comp_coulombs, com_f4_friction_comp_viscous

Table 9-17 List of speed control parameters

File name	Macro name	Set value	Description
r_motor_target motor_cfg.h	SPEED_CFG_CTRL_PERIOD	0.0005f	Control interval setting [s]
	SPEED_CFG_OMEGA	5.0f	Natural frequency for the speed control system [Hz]
	SPEED_CFG_ZETA	1.0f	Attenuation coefficient for the speed control system
	SPEED_CFG_SPEED_LIMIT_RPM	2850.0f	Speed limit value [r/min] (mechanical angle)
	SPEED_CFG_RATE_LIMIT_RPM	1000.0f	Acceleration limit [r/min/s]

SPEED_CFG_CTRL_PERIOD

Set to 0.0005s (0.5ms). If the value is changed, the timer setting value of Config_CMT0 must be changed to change the speed control cycle.

SPEED_CFG_OMEGA, SPEED_CFG_ZETA

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. If the natural frequency of the current control system is 500 Hz, then 500 Hz/3 = 166 Hz.

Considering that the Hall sensor has a low resolution, default values are set with relatively low tracking capability providing a margin to prevent the divergence of the control system. For example, if a disturbance vibrates at a higher natural frequency, increasing the value of the natural frequency to match the disturbance may improve the tracking of the disturbance to the vibration, and the motor may be driven more stably than the default setting.

For the attenuation coefficient for the speed control system, a value in the range from 0.7 to 1.0 is typically set. Setting a value nearer to 1.0 makes the response more stable and moderate. Make adjustment while checking the speed responsiveness.

SPEED_CFG_RATE_LIMIT_RPM

Sets the rate at which the speed increases (acceleration) when the speed command value is set. The higher the value, the faster the speed increases. When 100 is specified, the speed is increased by 100 r/min per second. The motor reaches standstill to 2000 r/min in 20 seconds.

Table 9-18 List of friction compensation parameters

File name	Macro name	Set value	Unit	com variable, description
r_motor_module_cfg.h	SPEED_CFG_FRICTION_BRAKE_SPEED	1.0f	rad/s	com_f4_friction_comp_brake_speed
	SPEED_CFG_FRICTION_COULOMBS	0.15f	A	com_f4_friction_comp_coulombs
	SPEED_CFG_FRICTION_STATC	0.3f	A	com_f4_friction_comp_static
	SPEED_CFG_FRICTION_VISCOUS_COEF	0.0f	A/(rad/s)	com_f4_friction_comp_viscous

SPEED_CFG_FRICTION_BRAKE_SPEED

Sets the threshold at which static friction occurs. As a guide, consider the speed detection accuracy when the motor is stopped, and set a small value (e.g., 1.0 [rad/s]).

SPEED_CFG_FRICTION_STATIC

When starting without compensation, set a value of about 80% of the torque command value (q-axis current command value) when the motor starts moving.

SPEED_CFG_FRICTION_COULOMBS

Set a value of the Iq command value when the motor rotates at 40% of the rated speed without load in an environment that is close to the actual one where the mechanical system is connected to the motor.

SPEED_CFG_FRICTION_VISCOUS_COEF

Set it to 0 if it is under a condition where the load changes gradually, if the speed control PI is dominant, or if the conditions are unknown.

9.11 Voltage Phase Lead Compensation Parameters

Compensation value to advance the angle to the timing when PWM is actually output based on the current detection timing. In this sample program and inverter configuration, the PWM output timing can be matched by advancing 0.5 samples.

Table 9-19 List of configurations

File name	Macro name	Set value	Description
r_motor_module_cfg.h	CURRENT_CFG_PERIOD_MAG_VALUE	0.5f	Sets the number of samples used for lead compensation. Set to 0.5.

9.12 Hall Sensor Vector Control Parameters

This section describes how to set the parameters required for Hall sensor vector control. Hall sensor vector control may not be able to achieve the desired performance in starting characteristics or low-speed operation if the parameters are set inappropriately.

Table 9-20 lists the parameters for Hall sensor vector control. The parameters related to friction compensation such as SPEED_CFG_FRICTION_BRAKE_SPEED, which are related to speed control parameters, are described in 9.10.

Table 9-20 Setting parameters for Hall sensor vector control

File name	Macro name	Set value	unit	com variable, description
r_motor_module_cfg.h	SPEED_CFG_FRICTION_BRAKE_SPEED	1.0f	rad/s	com_f4_friction_comp_brake_speed
	SPEED_CFG_FRICTION_STATIC	0.15f	A	com_f4_friction_comp_static
	SPEED_CFG_FRICTION_COULOMBS	0.1f	A	com_f4_friction_comp_coulombs
	SPEED_CFG_FRICTION_VISCOUS_COEF	0.0f	A/(rad/s)	com_f4_friction_comp_viscous
	SENSOR_HALL_CFG_ANGLE_CORRECT	0.0f	rad	com_f4_hall_angle_correct
	SENSOR_HALL_CFG_SPEED_DEVIATION_TOLERANCE	0.1f	-	com_f4_hall_spdmix_dev_th Set a value between 0 and 1.0.
	SENSOR_HALL_CFG_SPEED_MIX_TH	Auto	rad/s	com_f4_hall_spdmix_elec_speed_th (SPEED_CFG_OMEGA * MTR_TWOPI * 30.0f / 6.0f)
	SENSOR_HALL_CFG_DEFAULT_COUNTS	4000	-	
	SENSOR_HALL_CFG_MAXIMUM_PERIOD	5000	-	
	SENSOR_HALL_CFG_HALL_POLEPAIRS	Auto		
	SENSOR_HALL_CFG_PATTERN_1	1	-	
	SENSOR_HALL_CFG_PATTERN_2	5	-	
	SENSOR_HALL_CFG_PATTERN_3	4	-	
	SENSOR_HALL_CFG_PATTERN_4	6	-	
	SENSOR_HALL_CFG_PATTERN_5	2	-	
	SENSOR_HALL_CFG_PATTERN_6	3	-	

SENSOR_HALL_CFG_ANGLE_CORRECT

The correction value of the Hall detection angle. Adjust this value if the torque output is clearly lower than the specification, or if there is a large gap in output torque between clockwise and counterclockwise. In other cases, set it to 0.

SENSOR_HALL_CFG_SPEED_DEVIATION_TOLERANCE

The ripple ratio that is allowed at auto-switching of Hall sensor speed detection (range: 0.0 to 1.0). Adjust this parameter starting from 0 referring to the speed ripple ratio at the rated speed.

SENSOR_HALL_CFG_DEFAULT_COUNTS

Set a value of 80% to 100% of the *SENSOR_HALL_CFG_MAXIMUM_PERIOD.*

SENSOR_HALL_CFG_MAXIMUM_PERIOD

The number of counts to judge the speed as 0 (stopped). It depends on the frequency of counting, Set the number of counts equivalent to 0.25 seconds.

SENSOR_HALL_CFG_SPEED_MIX_TH

The speed threshold at which the speed is switched to the speed with high accuracy in the auto-switching mode. Set a value so that the actual sampling rate at this speed should separated from the bandwidth of speed control.

SENSOR_HALL_CFG_PATTERN_1, SENSOR_HALL_CFG_PATTERN_2,

SENSOR_HALL_CFG_PATTERN_3, SENSOR_HALL_CFG_PATTERN_4,

SENSOR_HALL_CFG_PATTERN_5, SENSOR_HALL_CFG_PATTERN_6

Set the pattern number of the Hall sensor. Set these based on the definition of High and Low of the detection circuit and the arrangement of the Hall sensor signal and the electrical angle.

10. Smart Configurator Settings

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.

10.1 Clock settings

Table 10-1 shows the clock settings.

Table 10-1 MCU clock settings

Clock type	Clock setting
Main clock	240Mz
System clock (ICLK)	120MHz
Peripheral module clock (PCLKA)	120MHz
Peripheral module clock (PCLKB)	60MHz
Peripheral module clock (PCLKC)	120MHz
Peripheral module clock (PCLKD)	60MHz
FlashIF clock (FCLK)	60MHz
IWDTCLK	120kHz

10.2 Component settings

Table 10-2 lists the components used and the functions allocated to the components.

Table 10-2 Smart Configurator components and their functions

Function	Component
3-phase PWM output, A/D conversion of current detection	Config_MOTOR
A/D conversion processing (inverter bus voltage detection)	Config_S12AD2
Setting of the port to be used	Config_PORT
Position and speed control interrupt timer	Config_CMT0
Independent watchdog timer	Config_IWDT
Overcurrent detection	Config_POE

10.3 AD settings

A 12-bit AD converter (S12AD) with built-in MCU is used to measure U-phase output current, W-phase output current, and inverter bus voltage. Table 10-3 shows the assigned channels and detection timing.

Since the current detection method of the MCI-LV-1 is based on a shunt resistor, the AD converter is set to acquire current at the trough of the carrier. After the AD converter performs from the start of AD detection to the end of conversion, an AD conversion end interrupt is generated, and current control is executed through the r_Config_MOTOR_ad_interrupt interrupt function.

Table 10-3 AD channels and detection timing settings

Function	Assigned channels	Conversion start trigger
Measurement of inverter bus voltage	AN003	TRG4AN (MTU4. COMPARE MATCH BETWEEN TCNT AND MTU4.TADCOBRA)
Measurement of U-phase current	AN000	
Measurement of V-phase current	AN001	
Measurement of W-phase current	AN002	

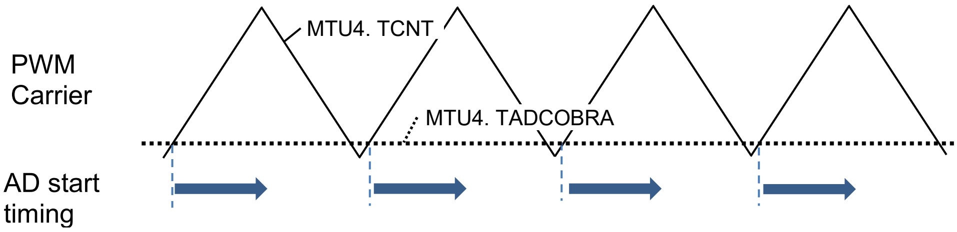


Figure 10-1 AD detection start timing

10.4 Motor settings

In this sample program, MTU3 is used to output PWM. Duty is reflected at the trough of the carrier. Configuration can be done by selecting Config_MOTOR from the component.

Table 10-4 MTU settings

Timer setting		
Cycle setting	Timer operation cycle	50us
	Count clock division ratio	1
	TGRA register value	3240
	Dead time	2.0us
Output pulse and A/D conversion trigger setting	A/D conversion start trigger skipping	Do not skip
	Up	High-level
	Un	High-level
	Vp	High-level
	Vn	High-level
	Wp	High-level
	Wn	High-level
Timer pulse output pin setting	U-phase	MTU4 B-D
	V-phase	MTU4 A-C
	W-phase	MTU3 B-D
A/D converter setting		
A/D conversion setting	Measurement target pin Iu	AN000
	Measurement target pin Iv	AN001
	Measurement target pin Iw	AN002
	Measurement target pin Vdc	AN003

10.5 Interrupts

Table 10-5 shows the information about the interrupts used for the MCUs that use the motor component.

Table 10-5 List of interrupts

Component	Interrupt function	Description
Config_MOTOR	r_Config_MOTOR_ad_interrupt	A/D conversion end interrupt Interrupt level: 12 Multiple interrupt: Enabled
Config_S12AD0/1	without	None
Config_PORT	without	None
Config_CMT0	r_Config_CMT0_cmi0_interrupt	Speed control interrupt Interrupt level: 11 Multiple interrupt: Enabled
Config_IWDT	without	None
Config_POE	r_Config_POE_oei1_interrupt	Hardware overcurrent interrupt Interrupt level: 15 Multiple interrupt: Disabled

10.6 Details of User Code

Table 10-6 lists the functions that are created in the user code area.

Table 10-6 List of functions in the user area

Component	Function	Description
Config_PORT	R_Config_PORT_GetSW1	Acquires SW1 status
	R_Config_PORT_GetSW2	Acquires SW2 status
	R_Config_PORT_Led1_on	Turns LED1 ON
	R_Config_PORT_Led2_on	Turns LED2 ON
	R_Config_PORT_Led1_off	Turns LED1 OFF
	R_Config_PORT_Led2_off	Turns LED2 OFF
	R_Config_PORT_GetHallSignal	Acquires Hall signal

10.7 POE Settings

Table 10-7 shows the POE settings that can be configured in Config_POE. Refer to the signal specifications of your inverter, as output pin settings vary depending on the inverter specifications.

Table 10-7 POE settings

Function			Set Value
POE0#	Request acceptance condition	Accept request on falling edge of the POE0# input	✔
	Number of sampling		-
	Interrupt setting	Output enable interrupt 1 enabled	✔
		Priority (Group BL1)	Level 15 (Max)
MTU3/MTU4	Stops the pin output when the output is short-circuited		✔
	Active level setting enabled		✔
	Output pin setting MTIOC3B. MTIOC3D	MTIOC3B active level	High
		MTIOC3D active level	High
	Output pin setting MTIOC4A. MTIOC4C	MTIOC4A active level	High
		MTIOC4C active level	High
	Output pin setting MTIOC4B. MTIOC4D	MTIOC4B active level	High
		MTIOC4D active level	High

10.8 Pin Settings

Table 10-8 shows the pin interface information.

Table 10-8 Pin Interface

Function	RX26T
Measurement of the inverter bus voltage	P43 / AN003
Measurement of the U-phase current	P40 / AN000
Measurement of the V-phase current	P41 / AN001
Measurement of the W-phase current	P42 / AN002
VR	P50 / AN204 / CMPC42
PWM output (U_n) / "High" active	P76 / MTIOC4D / GTIOC2B
PWM output (V_n) / "High" active	P75 / MTIOC4C / GTIOC1B
PWM output (W_n) / "High" active	P74 / MTIOC3D / GTIOC0B
PWM output (U_p) / "High" active	P73 / MTIOC4B / GTIOC2A
PWM output (V_p) / "High" active	P72 / MTIOC4A / GTIOC1A
PWM output (W_p) / "High" active	P71 / MTIOC3B / GTIOC0A
PWM emergency stop input when an overcurrent is detected	P70 / POE0#
LED2	P20
LED1	P21
SW2	P22
SW1	P23
Hall Sensor W	P24
Hall Sensor V	P27
Hall Sensor U	P30

11. Evaluation Results

11.1 Motor Control Evaluation

11.1.1 Acceleration and Deceleration Characteristics

The following figure shows data of the acceleration characteristics when the motor is driven from a standstill under the speed command sequence below.

[Speed command sequence]

0 → 300r/min → 2400r/min → 300r/min → -300r/min → -2400r/min → -300r/min → 0

Acceleration/deceleration limit: 1500r/min/s, rotation speed (forward = CW, reverse = CCW), no load condition

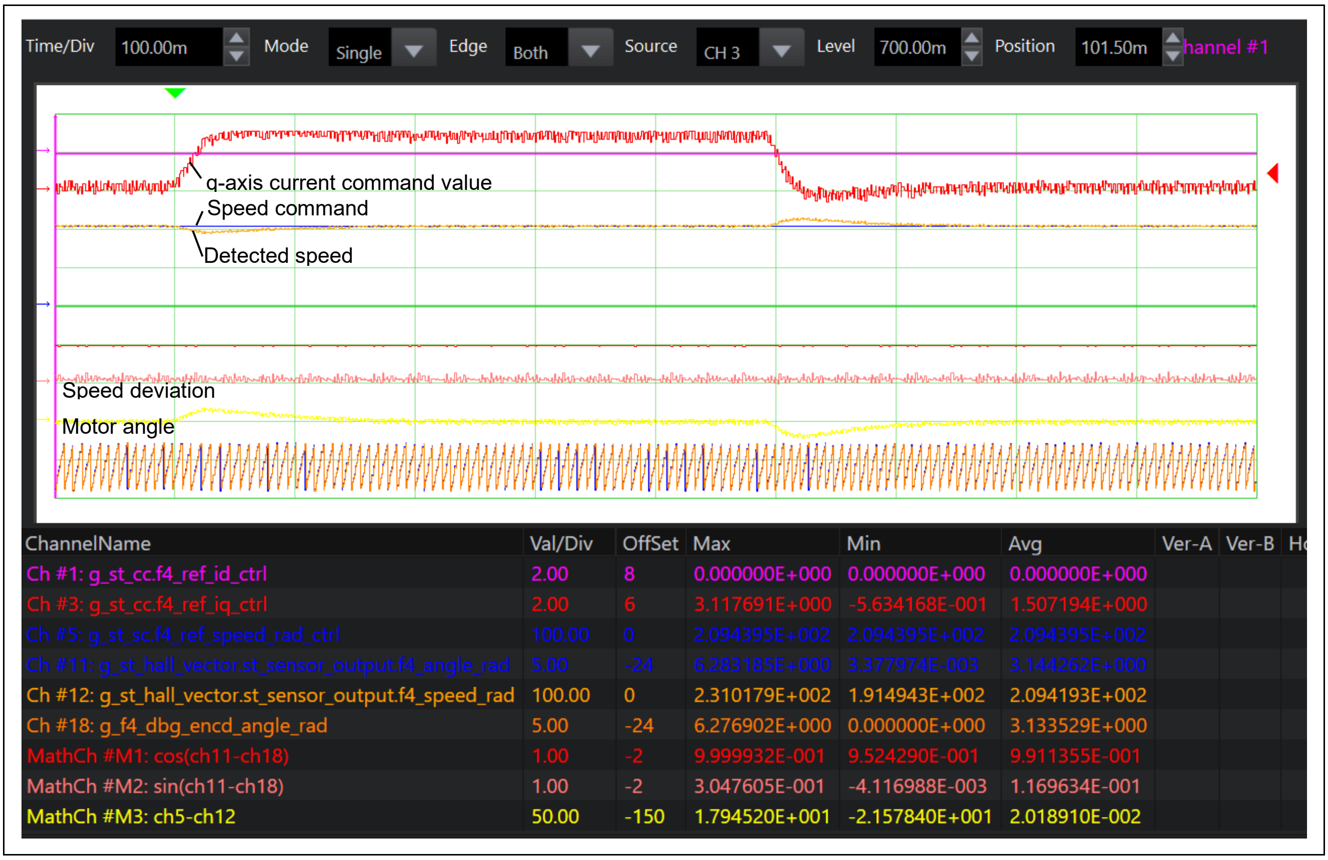
Time/Div 1.00 Mode Single Edge Both Source CH 11 Level 3.00 Position 500.00m

ChannelName	Val/Div	OffSet	Max	Min	Avg	Ver-A
Ch #1: g_st_cc.f4_ref_id_ctrl	500.00m	1	0.000000E+000	0.000000E+000	0.000000E+000	
Ch #3: g_st_cc.f4_ref_iq_ctrl	500.00m	1.5	5.066776E-001	-3.873322E-001	6.071755E-003	
Ch #5: g_st_sc.f4_ref_speed_rad_ctrl	100.00	0	2.513274E+002	-2.513274E+002	1.759626E-001	
Ch #11: g_st_hall_vector.st_sensor_output.f4_angle_rad	5.00	-24	6.278858E+000	1.043081E-007	3.220722E+000	
Ch #12: g_st_hall_vector.st_sensor_output.f4_speed_rad	100.00	0	2.554224E+002	-2.554224E+002	7.539791E-002	
Ch #15: g_st_sc.st_pi_speed.f4_refi	500.00m	1.5	1.123452E-001	-6.465404E-002	3.813474E-003	
MathCh #M1: ch5-ch12	50.00	-100	2.408571E+001	-3.104467E+001	1.005645E-001	

Figure 11-1 Acceleration/deceleration and forward/reverse characteristics

11.1.2 Load Characteristics

The load characteristic waveforms for 2000 r/min operation are shown below. Load tests have been worked from 200 r/min to 2000 r/min to confirm that operation at the rated load is possible.



Time/Div 100.00m
Mode Single
Edge Both
Source CH 3
Level 700.00m
Position 101.50m

ChannelName	Val/Div	OffSet	Max	Min	Avg	Ver-A	Ver-B	H
Ch #1: g_st_cc.f4_ref_id_ctrl	2.00	8	0.000000E+000	0.000000E+000	0.000000E+000			
Ch #3: g_st_cc.f4_ref_iq_ctrl	2.00	6	3.117691E+000	-5.634168E-001	1.507194E+000			
Ch #5: g_st_sc.f4_ref_speed_rad_ctrl	100.00	0	2.094395E+002	2.094395E+002	2.094395E+002			
Ch #11: g_st_hall_vector.st_sensor_output.f4_angle_rad	5.00	-24	6.283185E+000	3.377974E-003	3.144262E+000			
Ch #12: g_st_hall_vector.st_sensor_output.f4_speed_rad	100.00	0	2.310179E+002	1.914943E+002	2.094193E+002			
Ch #18: g_f4_dbg_encd_angle_rad	5.00	-24	6.276902E+000	0.000000E+000	3.133529E+000			
MathCh #M1: cos(ch11-ch18)	1.00	-2	9.999932E-001	9.524290E-001	9.911355E-001			
MathCh #M2: sin(ch11-ch18)	1.00	-2	3.047605E-001	-4.116988E-003	1.169634E-001			
MathCh #M3: ch5-ch12	50.00	-150	1.794520E+001	-2.157840E+001	2.018910E-002			

Figure 11-2 Load characteristics (at 2000r/min operation)

11.2 CPU Utilization

The following table shows the CPU processing time and loading rate for each control interval.

Table 11-1 Loop and CPU loading rate

Control loop type	Control interval	Processing time	CPU loading rate
Current control loop	50 us (0 decimation)	15.44us	30.9%
Speed control loop	500 us	4.9us (up to 17.14us due to interrupt nesting)	1.0%

11.3 Program Size and RAM Usage

The program size (ROM) and RAM usage for this sample program are as below. 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 11-2 Program size and RAM usage

Program size (ROM)	22647 [Bytes]
RAM usage	9704 [Bytes]
Maximum value of stack analysis result	300 [Bytes]
Stack size setting value for IDE environment	5120 [Bytes]

12. FAQ

12.1 Troubleshooting

Table 12-1 shows typical phenomena and their solutions.

Table 12-1 List of phenomena and their solutions

Phenomenon	Solution
The motor does not run. An overcurrent error occurs.	- Check whether there is a break or poor contact of U, V, and W. - Be careful not to break the Hall sensor wires. If the Hall sensor wire is broken or has poor contact, the motor cannot be controlled normally. - Check if the shaft is locked or overloaded.
The bus voltage drops during motor operation.	The output performance of the power supply may be insufficient for the load of the motor, or there may be a short circuit on the motor side.
Speed fluctuations occur during start-up. Is it possible to eliminate speed fluctuations?	In speed control, since vector control is performed from a standstill, the speed is assumed to be zero until a pulse from the Hall sensor is input, and the speed regulator is saturated, resulting in speed fluctuations. It is difficult to eliminate these speed fluctuations because the Hall sensor is a low-resolution encoder with six pulses in one rotation of electrical angle. This problem does not arise in the operation under torque control.
The motor cannot rotate at low speeds of 200 r/min or lower.	In operation at low speeds, a low-resolution Hall sensor has longer intervals of change in detection signal. As a result, the speed detection update cycle may be significantly slowed down, and the speed regulator may not be able to control the speed properly. The speed control mode of Hall sensor vector control is not suitable for applications that require rotation at low speeds. Consider using together with a torque control mode or another external speed detection method.
The motor cannot rotate after stopping due to an error.	For details, see 6.8 (c). It explains how to recover from errors.
The motor stops with an error even after starting.	- See 6.8 (c) for the cause of the error. - Verify that there is no overvoltage or undervoltage in the inverter bus voltage and that the output is not short-circuited. - Verify that sensor-related signal settings are correct. - Verify that the POE settings, motor settings (MTU terminal and active level settings), etc. in the Smart Configurator are properly set according to the inverter circuit specifications.
The value of the variable displayed on the RMW is abnormal.	If you amend the sample program, you should register the Map file generated in the HardwareDebug (e^2 studio version) or DefaultBuild (CS+ version) folder after building with RMW to update the variable status of the sample program. If you skip this process, the variables may not be displayed correctly. For details, see 6.6.
Values set from RMW are not reflected.	Manipulate variables in com_u1_enable_write to rewrite the parameters. When the timing for writing values to com_u1_enable write is prior to writing parameters, the internal reflection process operates first. Address as following: - Put the com_u1_enable_write on the last line - Write the com_u1_enable_write twice (with 1 then 0) or toggle write

Which motor or mechanical system to be chosen for smooth start-up	- When choosing a motor with many pole pairs, the resolution of the Hall sensor is higher, and the performance at low speeds is improved, resulting in smooth start-up. - The motor can start more smoothly with the inertia load than with no load.

Revision History

Rev.	Date of issue	Amendments	
		Page	Point
1.00	2024/12/26	-	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/.