

## Renesas RA Family

### IM V/f Control by High-Voltage Inverter

#### Introduction

The sample program covered by this application note provides induction motor V/f control algorithms for the RA6T2 CPU board and MCI-HV-2-3PH three-phase 200-VAC input high-voltage inverter from Renesas for general industrial use.

V/f control of induction motors involves simple algorithms for open-loop control and does not require detailed motor parameters. Therefore, handling the operation only requires minimal settings. V/f control is implemented in general-purpose inverters, etc., that are widely used in general industrial applications in factories or buildings. Among such applications, it is mainly used in those that do not require a large starting torque, such as fans or pumps.

This sample program provides the following basic functions for V/f control and inverter control.

- V/f control and torque boost compensation for induction motors
- Dead-time (voltage error) compensation and sample delay compensation
- Control of the relay for connecting the inrush-current prevention resistor
- Protection against overvoltage and overcurrent and detection of overheating in the inverter

This application note describes how to set up and use the combination of the sample program and the inverter and also describes the specifications of the internal program for the user to apply in evaluating Renesas MCUs and semiconductor devices in inverter development projects. Figure 1-1 shows the hardware configuration for use with this sample program.

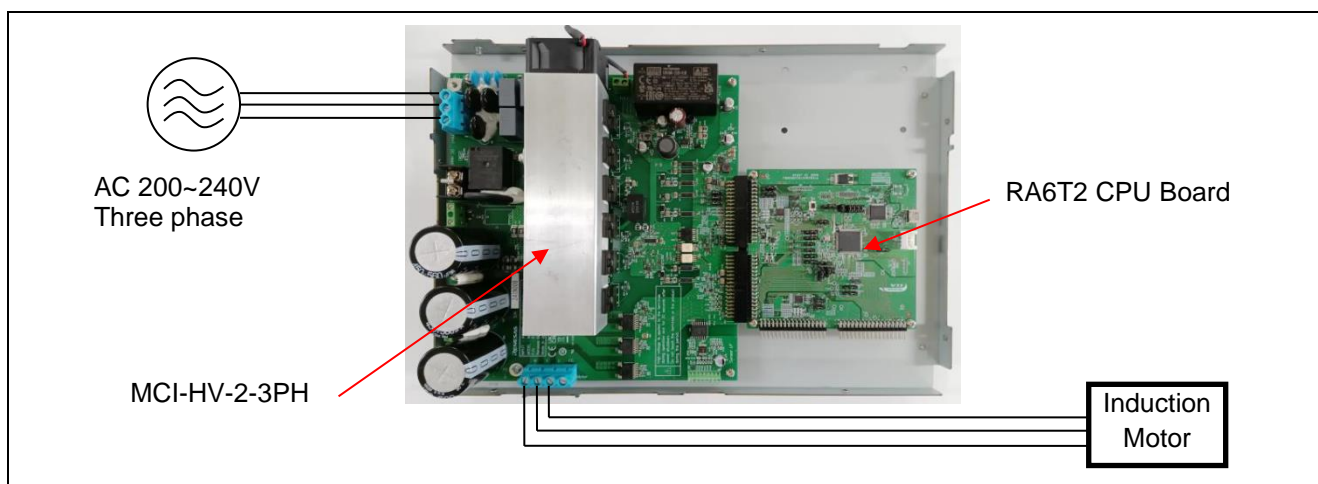


Figure 1-1 Hardware Configuration

The sample program provided with this application note is only for use in evaluation. Renesas Electronics Corporation does not guarantee the desired performance or operation. Before using this sample program, conduct thorough evaluation in an appropriate environment.

#### Target Device

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

- RA6T2 (R7FA6T2BD3CFP)

---

Contents

1. Introduction.....	5
2. Glossary .....	7
3. Hardware Devices and Software Tools That are Used.....	8
3.1 List of Hardware Devices That are Used.....	8
3.2 List of Software Tools That are Used .....	8
4. Configuring a Hardware Environment .....	9
4.1 Overview of Hardware Environment.....	9
4.2 Preparing a Power Supply .....	9
4.3 Preparing a Motor .....	9
4.4 Preparing a Load System .....	10
4.5 Preparing an Inverter.....	10
4.6 Setting up the RA6T2 CPU Board.....	11
4.7 Wiring .....	12
4.8 Using Measuring Instruments.....	13
5. Configuring a Software Environment .....	14
6. Driving the Motor .....	15
6.1 Points to Note before Driving the Motor .....	15
6.2 Procedures of Preparing for Operation .....	15
6.3 Connections.....	16
6.4 Writing the Sample Program .....	17
6.5 Installing the RMW .....	17
6.6 Updating Registration of the Map File .....	18
6.7 Settings of Communications in the RMW.....	18
6.8 Variables Used for Operating the RMW .....	19
6.9 Operating the Motor.....	20
6.10 Stopping and Shutting Down the Motor .....	23
7. Motor Control Algorithms .....	24
7.1 V/f Control for an Induction Motor .....	24
7.1.1 Principles of Torque Generation and Slip in an Induction Motor.....	24
7.1.2 Overview of V/f Control .....	25
7.2 Overview of the Implemented Functions .....	27
7.3 Control Block Diagram .....	27
7.4 Speed Control Function.....	28
7.5 Current Control Function .....	28
7.6 V/f Control.....	28
7.7 Torque Boost Compensation.....	29
7.8 Flux Weakening Control .....	30
7.9 Sample Delay Compensation .....	31
7.10 Voltage Error Compensation .....	31

7.11	Pulse-Width Modulation (PWM) Mode .....	32
8.	Software Specifications and Configuration.....	33
8.1	Software Specifications .....	33
8.2	Overall Configuration of the Software .....	34
8.3	Task Descriptions .....	35
8.4	Configuration of Folders and Files .....	36
8.5	Application Layer .....	38
8.5.1	Functions .....	38
8.5.2	Structure and Variable Information.....	38
8.5.3	Adjustment and Configuration of Parameters .....	39
8.6	System Manager .....	40
8.6.1	Functions .....	40
8.6.2	Module Configuration Diagram.....	40
8.7	Motor Manager and Motor Control Modules .....	41
8.7.1	Functions .....	41
8.7.2	Module Configuration Diagram.....	42
8.7.3	Mode Management .....	43
8.7.4	Protection Function .....	44
8.7.5	API .....	45
8.7.6	Structure and Variable Information.....	47
8.7.7	Macro Definitions.....	52
8.8	Relay Manager and Relay Control Module .....	54
8.8.1	Functions .....	54
8.8.2	Module Configuration Diagram.....	54
8.8.3	Sequence Descriptions .....	54
8.8.4	Protection Function .....	55
8.8.5	API .....	55
8.8.6	Structure and Variable Information.....	56
8.8.7	Macro Definitions.....	57
8.9	Driver Module .....	58
8.9.1	Functions .....	58
8.9.2	Module Configuration Diagram.....	58
8.9.3	API .....	59
8.9.4	Configuration Items .....	60
8.9.5	Structure and Variable Information.....	62
8.9.6	Adjustment and Configuration of Parameters .....	64
9.	Parameter Settings.....	65
9.1	Overview.....	65
9.2	MCU-Related Parameters .....	65
9.3	List of Parameters for Setting Control Functions .....	66
9.4	Protection-Related Parameters .....	67
9.5	Changing the PWM Carrier Frequency for Motor Control .....	67
9.6	Setting the Pulse-Width Modulation Method .....	68
9.7	Inverter Parameters.....	69
9.7.1	Overview.....	69

9.7.2	Current Detection Gain.....	70
9.7.3	Voltage Detection Gain .....	71
9.7.4	Voltage Error Compensation Parameters .....	71
9.8	Motor Parameters.....	73
9.9	Current Control Parameters .....	73
9.10	Speed Control Parameters .....	74
9.11	Voltage Phase Lead Compensation Parameter .....	74
9.12	V/f Control Parameters .....	75
10.	Settings for the FSP.....	76
10.1	Overview of the FSP.....	76
10.2	Setting FSP Stacks.....	76
10.3	Callback Interrupts.....	77
10.4	Pin Settings .....	78
10.5	Settings for the Three-Phase PWM GPT .....	79
10.6	AGT0 Settings (Setting the Interval for Speed Control) .....	80
10.7	AGT1 Settings (Setting the Interval for Relay Control) .....	81
10.8	ADC Settings .....	82
10.9	POEG Settings .....	86
11.	Results of Evaluation .....	87
11.1	Evaluation of Inrush-Current Prevention Relay Control .....	87
11.2	Evaluation of Motor Control.....	88
11.2.1	Acceleration/Deceleration Characteristics .....	88
11.2.2	Evaluation of Motor Characteristics.....	88
11.3	CPU Utilization .....	89
11.4	Program Size and RAM Usage .....	89
12.	FAQ.....	90

## 1. Introduction

This application note is intended to explain how to use the sample program that employs an RA6T2, a microcontroller (MCU) manufactured by Renesas, to drive a three-phase 200-VAC induction motor with V/f control. It is also intended to describe the configuration, specifications, and method of control by the software.

For how to set up, use, and check the operation of the MCI-HV-2-3PH inverter manufactured by Renesas, refer to the MCI-HV-2-3PH User's Manual (R12UZ0170).

This sample program can control a 3.7-kW MLU1115D motor (a three-phase 200-VAC induction motor from Fuji Electric Co., Ltd.) through V/f control by using the RA6T2 CPU board from Renesas and an MCI-HV-2-3PH inverter from Renesas. This sample program supports the Renesas Motor Workbench, a motor control development support tool, and therefore can be used as a user interface (UI) for checking the MCU internal data and controlling a motor. You can use the sample program for reference to check how MCU functions are assigned, how control is loaded on interrupts, and other information in the sample program when selecting an MCU to be used or developing software.

The sample program described in this application note was developed and evaluated in the environment of the induction motor and inverter described in this document and is not guaranteed to work with your induction motor or inverter environment. The control performance may be limited by the current sensor itself; the PCB design pattern of the signal path; sampling, resolution, and filter specifications; load characteristics of the motor and variations between individual motors. Under the responsibility of the user, refining algorithms and using parameters will be 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 Equipment and Devices Used for Evaluation

Inverter: MCI-HV-2-3PH inverter from Renesas

Motor: 3.7-kW MLU1115D induction motor from Fuji Electric Co., Ltd.

## Target Software

The following shows the target software for this application note.

- RA6T2\_MCIHV2\_IM\_LESS\_VF\_E2S\_V100 (IDE: e<sup>2</sup> studio)

## Reference Documents

- RA6T2 Group User's Manual — Hardware (R01UH0951)
- Renesas Motor Workbench User's Manual (R21UZ0004)
- MCB-RA6T2 User's Manual (R12UZ0099)
- MCI-HV-2-3PH User's Manual (R12UZ0170)

The following shows a summary of the items for frequent checking in this application note and the corresponding section for each.

Table 1-1 List of Items for Checking and the Corresponding Sections

Item for Checking	Reference Section
Identify and select necessary devices.	3
Select a power supply.	4.2
Select a motor.	4.3
Select an inverter.	4.5
Check the wiring.	4.7
Prepare a software development environment for the sample program.	5
Write the sample program to the MCU.	6.3, 6.4
Install software for operating the motor on a PC.	6.5
Modify the sample program and then reflect the changes in the Renesas Motor Workbench (RMW).	6.6
Review the internal information of the sample program on the PC.	6.8
Drive the motor.	6.9
Stop the motor.	6.10
Examine the motor control algorithms.	7
Examine the structure of the sample program.	8
Examine and change the inverter parameters.	9.7
Examine and change the motor parameters.	9.8
Change the PWM carrier frequency for motor control.	9.5
Change the MCU settings.	10
Check the 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
IDE	An integrated development environment such as e <sup>2</sup> studio.
induction motor	Also called Alternating current induction motor (ACIM). In this document, the term is used to indicate a three-phase squirrel-cage induction motor.
MC-COM	A set of communication jigs and tools connected for displaying waveforms. For details, refer to the following URL.  <a href="https://www.renesas.com/en/products/microcontrollers-microprocessors/rx-32-bit-performance-efficiency-mcus/rtk0emxc90s00000bj-mc-com-renesas-flexible-motor-control-communication-board#overview">https://www.renesas.com/en/products/microcontrollers-microprocessors/rx-32-bit-performance-efficiency-mcus/rtk0emxc90s00000bj-mc-com-renesas-flexible-motor-control-communication-board#overview</a>
RMW	Renesas Motor Workbench, which is software specifically designed for motor control operations.
rms	Root-mean-square, which is used as an index representing a DC quantity that is required to obtain an effect equivalent to that of a given AC quantity. This value multiplied by $\sqrt{2}$ is the peak value. Values for rms voltage and rms current can be obtained. $V_{rms}$ indicates an rms voltage.
Inverter bus voltage	The DC voltage fed to the inverter circuit. Also called DC intermediate voltage.
Voltage phase lead compensation	Also called sample delay compensation.
Emulator	A device used to program an MCU. Also called an ICE.
Open-loop control	A control technique that does not require feedback signals. V/f control is a type of open-loop control.
Stack	A driver module generated by the FSP to facilitate the use of MCU peripheral functions.
Electrical angle	The 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	The rotation angle of the motor axis. One rotation of the axis per minute is 1 rpm.

### 3. Hardware Devices and Software Tools That are Used

#### 3.1 List of Hardware Devices That are Used

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

Table 3-1 List of Hardware Devices That are Used

Hardware	Manufacturer	Product Code
RA6T2 CPU board 	Renesas	RTK0EMA270C00000BJ  MCU product code RA6T2, R7FA6T2BD3CFP
Inverter board	Renesas	MCI-HV-2-3PH RTK0EM0000B16030BJ
Isolated communication board MC-COM 	Renesas	Renesas Flexible Motor Control Communication Board RTK0EMXC90S00000BJ
Induction motor for testing	Fuji Electric	200-VAC 3.7-kW MLU1115D
AC power supply unit	Chroma ATE	12-kVA 61812
Power meter	Yokogawa Test & Measurement	Main unit: WT1800E-5A6-50A0-HE-D/G5/MTR/PD2 Current sensor: CT200
Load system and load motor	Fuji Electric	Load inverter: FRN18.5VG1S-2J Load motor: MVK8185A
Torque meter and monitor	Unipulse	Torque meter: UTMIII-100Nm Monitor: TM320

#### 3.2 List of Software Tools That are Used

The following lists the software tools and their versions used in evaluating this sample program. This sample program can be used within limitations of Renesas development environment e<sup>2</sup> studio.

Table 3-2 List of Software Tools That are Used

Manufacturer	Software Tool	Version	Remark
Renesas	e <sup>2</sup> studio	2024-10	Free version
Renesas	FSP	5.6.0	
Renesas	Renesas Motor Workbench	3.1.2	

## 4. Configuring a Hardware Environment

### 4.1 Overview of Hardware Environment

This section describes the hardware environment in which a PM motor is operated by using this sample program. Figure 4-1 shows a sample hardware configuration.

In the sections that follow, the power supply (section 4.2), the motor and load system (sections 4.3 and 4.4), the inverter (section 4.5), and the RA6T2 CPU board and its monitoring and programming mechanisms (section 4.6) are described in detail.

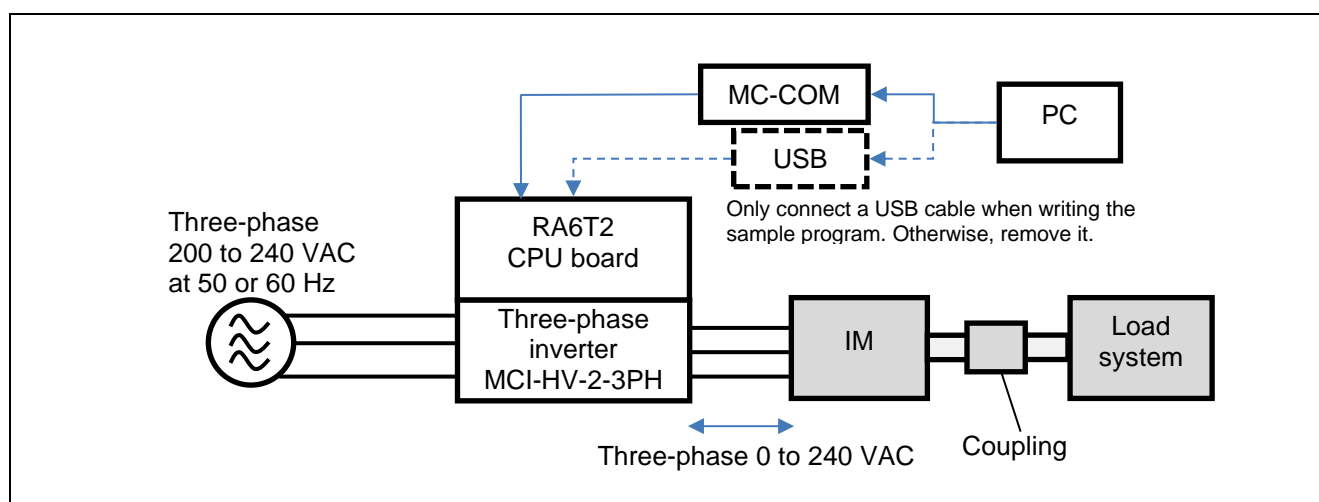


Figure 4-1 Sample Hardware Configuration

### 4.2 Preparing a Power Supply

The MCI-HV-2-3PH inverter used for this sample program can receive three-phase 200 to 240 VAC power at 50 or 60 Hz as the input power supply. The voltage supplied to the inverter varies depending on the inductive voltage, rating conditions, and maximum load conditions of the motor to be used. A power supply with sufficient capacity to drive a 200-VAC induction motor must be prepared. Select an appropriate type of power supply based on your experimental environment and restrictions and conditions of the power supply to be used.

For this sample program, a system that can supply 5.4 kVA or more is used so that it can drive a 2.2-kW (rating) to 3.7-kW (max.) induction motor.

### 4.3 Preparing a Motor

Before connecting the inverter to a motor, check the parameters of the motor that are required to drive the motor with V/f control on the nameplate of the motor. In addition, contact the manufacturer of the motor to obtain the parameter information as required.

If motor parameters are changed, the following parameters for the inverter and V/f control should be changed accordingly.

- Rated values (current, voltage, and speed)
- Number of poles

Table 4-1 shows the parameters of the 3.7-kW MLU1115D motor from Fuji Electric Co., Ltd., which we investigated. The parameters are based on our own measurements and may vary between individual motors and depending on the measurement conditions. The accuracy of these parameters or performance of the motor is not guaranteed.

Table 4-1 3.7-kW MLU1115D Motor Parameters (Some Values are Based on Our Own Measurements)

Parameter	Value	Remark
Primary resistance R	0.556 $\Omega$	For the resistor for one phase
Moment of inertia	0.000543 kgm <sup>2</sup>	
Capacity	3.7 kW	
Number of poles	4	2 pole pairs
Rated speed	1465 rpm at 50 Hz 1755 rpm at 60 Hz	
Rated frequency	100 Hz (electrical angle) 50 Hz (mechanical angle)	
Rated slip	35 rpm at 50 Hz 45 rpm at 60 Hz	
Rated current	15.5 A at 50 Hz 13.5 A at 60 Hz	
Rated torque	24.1 Nm	

#### 4.4 Preparing a Load System

Evaluation of the control of the inverter and motor requires acquisition of the output characteristics and a load system is required. The user should prepare the load system. Select a load system that can be connected to the target motor for evaluation and couple it to the motor. In addition, connect a torque and speed meter that can measure the torque and speed between the load system and motor so that accurate torque and speed characteristics can be obtained.

This evaluation is based on the use of equipment that allows the stable 2.2-kW or larger rated load of the inverter. For continuous testing, using a regenerative load tester is recommended to enable feedback to the inverter under testing. Before using a load tester that uses a particle brake or a hysteresis brake, check the restrictions on continuous operation due to heat generation in the test equipment, etc.

#### 4.5 Preparing an Inverter

When preparing an inverter, note the following information. This sample program is configured for the MCI-HV-2-3PH inverter board.

V/f control implemented in the sample program does not include a control algorithm that uses the current detection value input from the current sensor. When the motor control performance needs to be improved through the implementation of additional compensation functions by the user, this is strongly influenced by the performance of the sensor itself and the accuracy and variations of the circuits that serve as paths for the signals output by the sensor. When selecting an inverter, careful consideration must be given to the design of the inverter:

- Rated capacity (kVA)
- Dead time value ( $\mu$ s)
- Type, characteristics, and signal specifications of the current sensor
- Characteristics data of the current sensor including gain and offset values, relationship between the current and voltage, and linearity of the signals
- Characteristics data of the voltage sensor including gain and offset values and linearity of the signals

## 4.6 Setting up the RA6T2 CPU Board

This section describes how to install the RA6T2 CPU board (RTK0EMA270C00000BJ), which can be plugged into MCI-HV-2-3PH. You can plug the RA6T2 CPU board to a side of the MCI-HV-2-3PH board. A connector for writing the sample program and a connector for MC-COM are also provided.

Connector for MC-COM

USB connector for writing  
the sample program



Figure 4-2 RA6T2 CPU Board and Its Interfaces

Table 4-2 Settings of the Jumpers on the CPU Board

Jumper	Setting	Description of the Setting
JP1 to JP6	—	
JP7	—	
JP8 and JP9	—	
JP10	—	
JP11	—	
JP12	Pins 1 and 2 are closed: Setting for operating the motor  Pins 1 and 2 are open: Setting for writing the sample program	Closed: J-Link OB is disabled. Open: J-Link OB is enabled.
JP13	—	
JP14	Pins 1 and 2 are closed.	RA6T2 is enabled.
JP15 and JP16	—	
JP17	Pins 2 and 3 are closed.	INV1 encoder A
JP18	Pins 2 and 3 are closed.	INV1 encoder B
JP19	Pins 1 and 2 are closed.	INV1 W-phase voltage detection
JP20	Pins 1 and 2 are closed.	INV1 V-phase voltage detection
JP21	—	
JP22	—	
JP23 to JP25	Pins 1 and 2 are open.	The LPF for current detection is enabled.

## 4.7 Wiring

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

Figure 4-3 shows an example of wiring between the power supply and the inverter. In this example, an AC power supply unit that can output three-phase 200 VAC power at 50 Hz is used and it is connected to the R, S, and T pins of CN1. Figure 4-4 shows an example of wiring between the inverter and the motor. The wires from the motor are connected to the U, V, and W pins of CN6. Ground the FG pin of CN1 or CN6 to ensure safety.

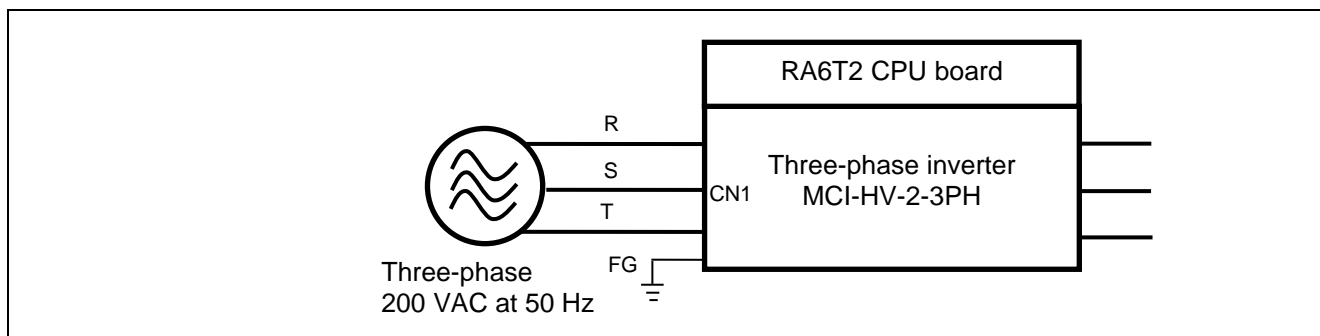


Figure 4-3 Wiring between the Power Supply and Inverter

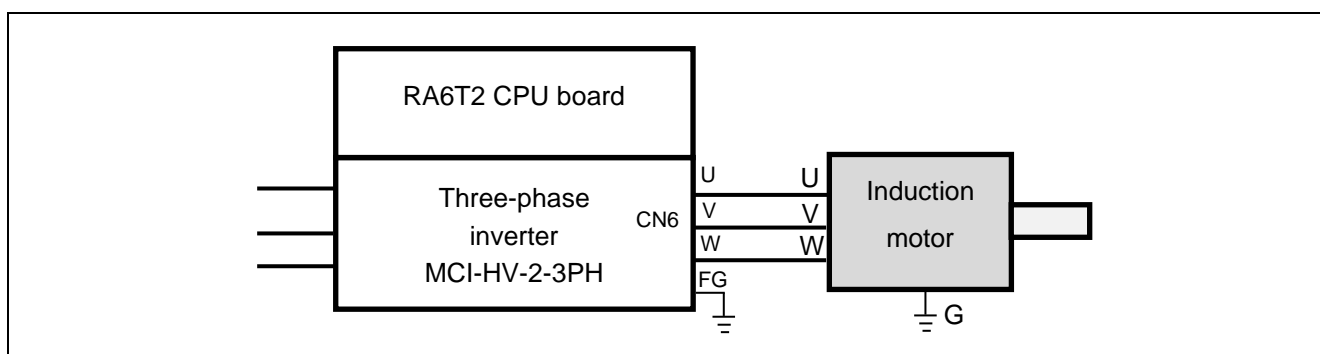


Figure 4-4 Wiring between the Inverter and Motor

#### 4.8 Using Measuring Instruments

When evaluating the control performance of a motor, using a power meter, a digital multimeter, a torque meter, or an external encoder enables more detailed analysis of control of the inverter and motor. Consider which measuring instruments are required according to the user environment, required measurement accuracy, and target performance specifications.

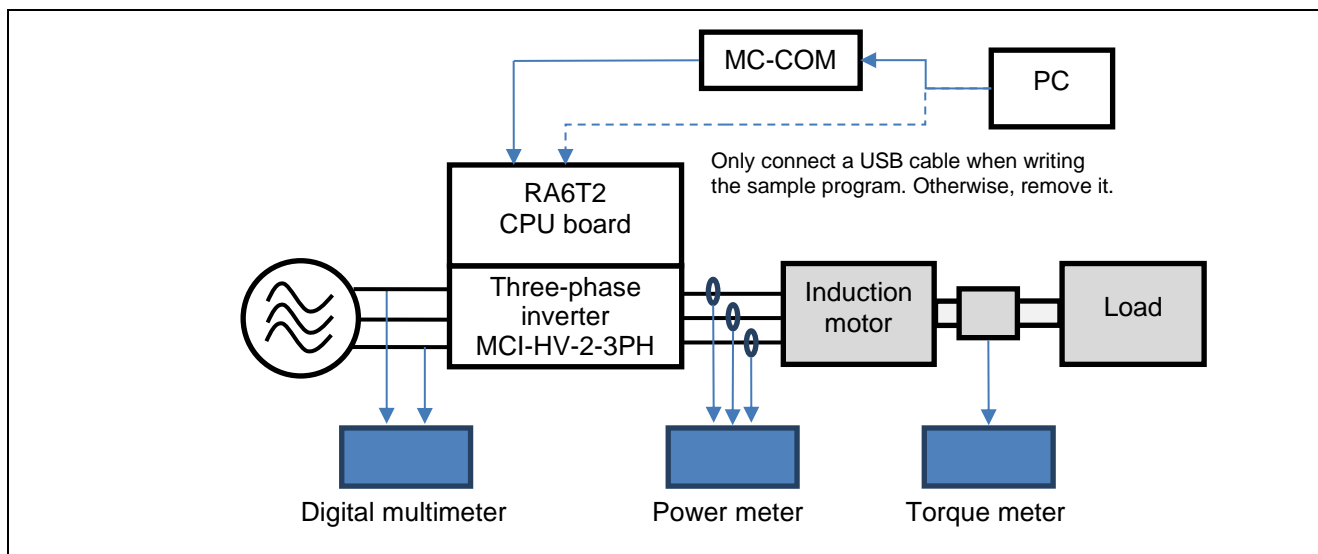


Figure 4-5 Example of Additional Measuring Instruments

## 5. Configuring a Software Environment

The e<sup>2</sup> studio is used for development of a system. Download it from the following site. Note that the FSP used in this sample program is necessary in addition to the e<sup>2</sup> studio.

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

The "FSP with e<sup>2</sup> studio" package, which contains both the FSP and e<sup>2</sup> studio for easy installation, can also be used. Access the following FSP page on the Renesas site or github site.

<https://www.renesas.com/en/software-tool/flexible-software-package-fsp>

<https://github.com/renesas/fsp/releases>

For more information on how to use the e<sup>2</sup> studio, refer to the PDF manual that you can download from the above e<sup>2</sup> studio page or the videos on the page.

## 6. Driving the Motor

### 6.1 Points to Note before Driving the Motor

When running the motor, note the following points. Improper use may cause an electric shock or lead to devices breaking down.

- The MCI-HV-2-3PH inverter is intended for use in industrial applications. The main circuits and CPU board are isolated. Consider isolation of the signal and power lines when connecting signals to external devices or instruments.
- Do not apply the power-supply voltage (three-phase 200 VAC) for the main circuits for the inverter when you write the sample program to the CPU board. The power to be used in writing the sample program to the CPU board should be that supplied from the PC through a USB cable or from the control power terminal of the MCI-HV-2-3PH.
- Do not control the motor under conditions where tracing and breakpoints are set. Doing so may lead to a sudden stop, which may cause the inverter to operate abnormally. Use the RMW and MC-COM to perform debugging under conditions where the safety functions are working properly.
- Remove the USB cable from the USB connector on the CPU board before driving the motor. The USB connector on the CPU board is not electrically isolated, which may cause adverse effects or failure on the PC through GND if the inverter operates abnormally.
- MC-COM can be safely used even during operation of the motor while 200-VAC power is being supplied to the inverter because the signals are isolated. When the USB connector on the CPU board is used to write the sample software, an electric shock hazard, the intrusion of noise to the PC, or damage to the PC via the GND may occur.
- Design the facility for testing the motor operation so that the motor can be stopped and the power can be cut off under any circumstances in an emergency. Make sure that an emergency stop button for the facility is placed close to the operator.
- The motor shaft rotates at high speeds, so be sure to install a cover over the coupling section as a guard. Parts such as couplings may scatter outside the rotating shaft if they are damaged during rotation.
- If the inverter is stopped but the motor is still rotating, do not touch or move close to the motor until it stops rotating. In the evaluation environment, place an electromagnetic switch between the inverter and motor so that the connection between them can be cut off in an emergency.

### 6.2 Procedures of Preparing for Operation

The procedures of preparing for operation are shown below.

Table 6-1 Procedures of Preparing for Operation

Step	Description	Reference Section
1	Insert the CPU board to the inverter board in advance.	4.6
2	Install the sample program and development environment software (e <sup>2</sup> studio) on the PC used.	5
3	Connect the PC to the CPU board via a USB cable and supply 5-V power to the CPU board.	6.3
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 PC and the CPU board.	6.3
7	Connect MC-COM to the CPU board.	6.3

8	Supply 200-VAC 50-Hz power to the inverter.	4.2
9	Use the 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	Verify that the variables of this sample program and sensor information are properly displayed on the RMW.	6.7
11	Use the RMW to operate the motor.	6.9
12	Stop and shut down the motor.	6.10

### 6.3 Connections

Note that the device to be used between the CPU board and the PC differs between writing and operating. The connections for (1) writing and (2) motor operation are described below.

#### (1) Writing

The CPU board has a dedicated circuit for use in writing, so an external in-circuit emulator (ICE) is not necessary. The USB port of the CPU board is not electrically isolated. Therefore, for your safety, be sure to remove the USB cable from the CPU board during operation after writing.

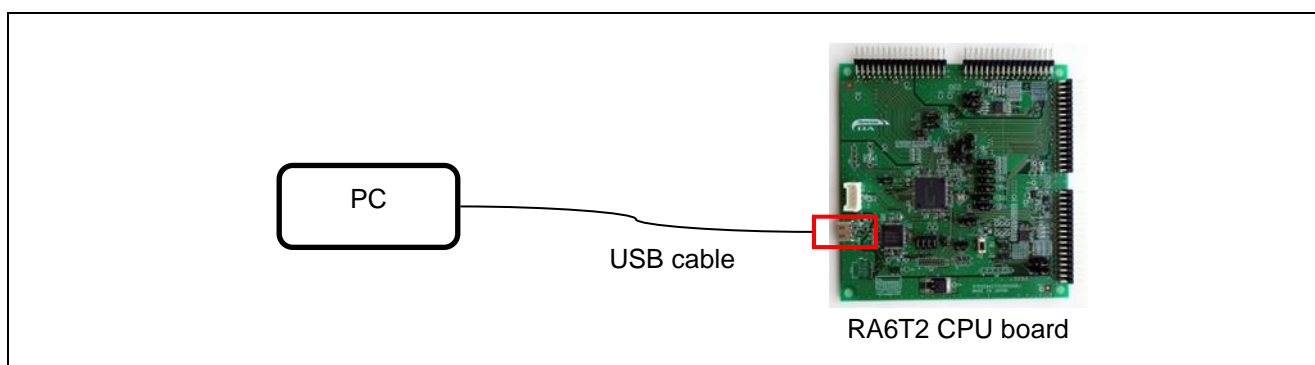


Figure 6-1 Example of Connection for Writing

#### (2) Motor operation

Use MC-COM to connect the PC to the CPU board. The CPU board is connected to the PC via UART and can be operated from the PC through a COM port. The RMW is used 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.

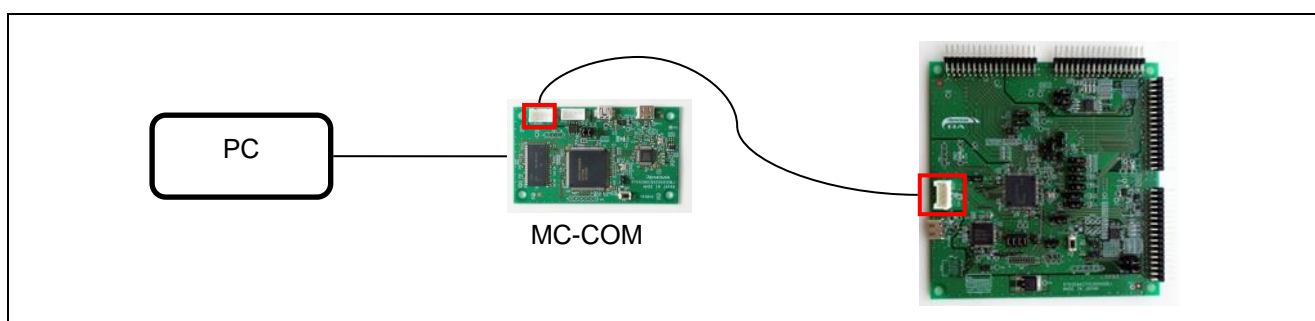


Figure 6-2 Example of Connections for Motor Operation

## 6.4 Writing the Sample Program

After you have downloaded the sample program from our website, use the e<sup>2</sup> studio to write it to the MCU on the CPU board.

For details about how to write programs, see the documentation for the e<sup>2</sup> studio.

As the RA6T2 CPU board includes circuits equivalent to those of an emulator, there is no need to purchase a separate dedicated emulator product for writing programs. Connect the RA6T2 CPU board and PC through a USB cable, and the debugging and programming functions of the e<sup>2</sup> studio can then be used to write the sample program to the RA6T2 CPU board.

## 6.5 Installing the RMW

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

Renesas Motor Workbench website:

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

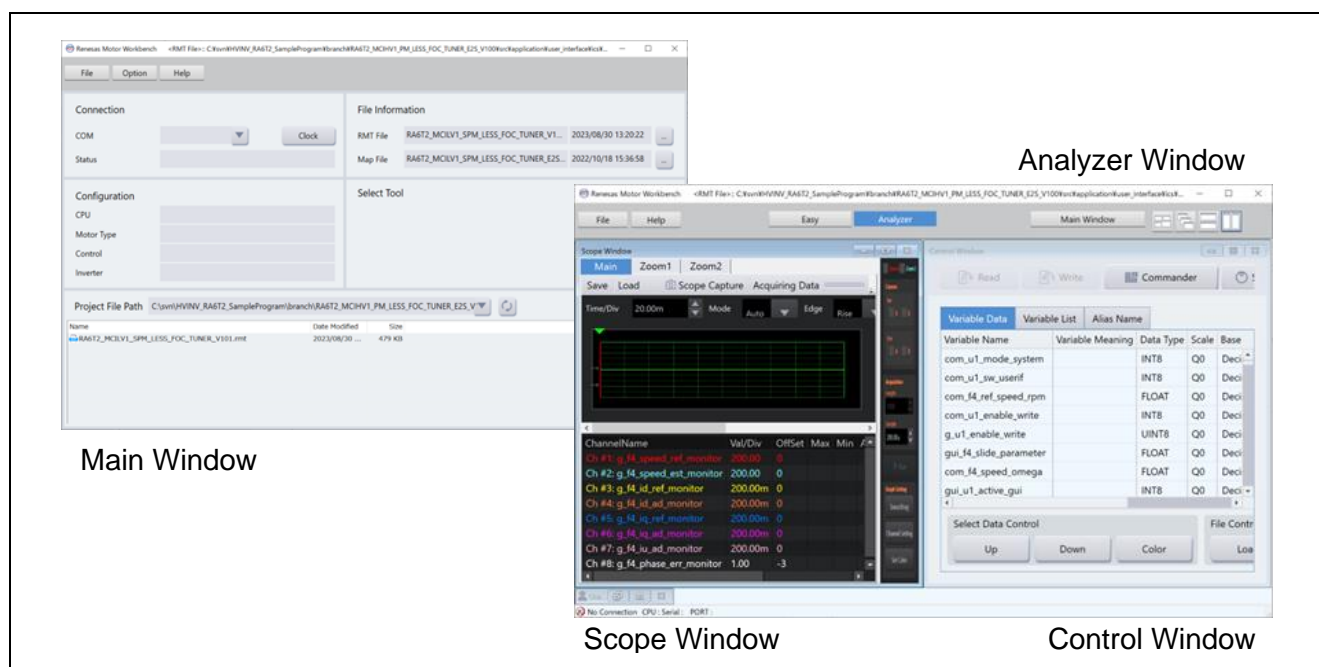


Figure 6-3 Windows of Renesas Motor Workbench

## 6.6 Updating Registration of the Map File

If a part of the sample program has been modified and the sample program has then been rebuilt by the user, information such as the addresses of variables may have changed. Registration of the Map file to include the changed information requires updating. If the sample program has not been modified, registration of the Map file does not require updating.

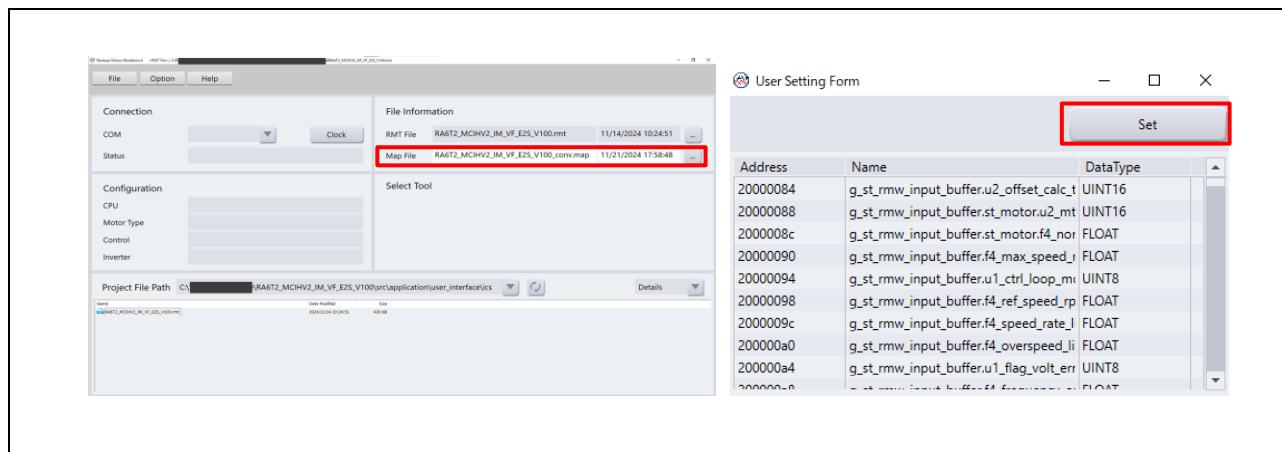


Figure 6-4 Example of Map File Registration Setting (Left) and Setting Confirmation Window (Right)

## 6.7 Settings of Communications in the RMW

This sample program sets up the RMW communications as shown in Table 6-2.

Even when the settings are correct, communications may not proceed as expected. This depends on the state of activation of the CPU board. In such cases, turning the power for controlling the CPU board or inverter off and then on may improve the state of communications.

Table 6-2 Settings of Communications in the RMW

Item	Setting	Example of Setting Window
Transfer rate	921,600 bps	Baudrate Setting <span style="color: red;">✖</span> <div style="border: 1px solid black; padding: 2px; display: inline-block;">921,600</div> bps
Clock setting	8,000,000 Hz	Clock Setting <span style="color: red;">✖</span> <div style="border: 1px solid black; padding: 2px; display: inline-block;">8,000,000</div> Hz

## 6.8 Variables Used for Operating the RMW

The RMW is used to control the motor in this sample program. Table 6-3 lists the input variables that are used when the RMW UI is in use. Input values can only be reflected in the corresponding variables in the motor module for use in controlling the motor when the values are written with the com\_u1\_enable\_write value being toggled between 0 and 1 each time. Note, however, that the variables marked with an asterisk (\*) in the table are written regardless of the state of toggling of the com\_u1\_enable\_write variable.

For the other parameters for controlling the motor, see Table 6-5.

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

Table 6-3 List of Main Input Variables for the Analyzer Functions

Name of the Input Variable for the Analyzer Functions	Type	Description
com_u1_system_mode (*)	uint8_t	Manages the inverter state. 0: Motor stop mode 1: Motor driving mode 3: Error reset
com_f4_ref_speed_rpm (*)	float	Speed command value (mechanical angle) (rpm)
com_u1_enable_write	uint8_t	Enables the rewriting of the user input variables. Input data are only reflected in variables when this value is toggled and matches the value of the g_u1_enable_write variable.
g_u1_system_mode	uint8_t	System mode 0: Motor stop 1: Motor driving 2: Error
g_u1_enable_write	uint8_t	Enables the rewriting of variables.

Table 6-4 lists main structure variables that are often monitored in the evaluation of driving under speed control. The waveforms of these values can be displayed by the Analyzer functions. Use this table for reference when the values of variables are to be loaded.

Table 6-4 List of Main Variables

Name of Main Variable	Type	Description
g_st_imvf.u2_error_status	uint16_t	Error state. For details, see section 6.9 (f), What to do in case of the motor stopping (due to an error).
g_st_relay_manager.u2_error_status	uint16_t	Error state of the relay module
g_st_cc.f4_vdc_ad	float	Inverter bus voltage (V)
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	Angle command value (electrical angle) (rad)
g_st_sc.f4_ref_speed_rad_ctrl	float	Speed command value (mechanical angle) (rad/s)

The following com variables can be used to dynamically change the constants and parameters of the motor through the RMW. Note that the written values are cleared when the power is turned on or off or the MCU is reset.

Table 6-5 List of com Variables

Variable	Description
com_u2_offset_calc_time	Setting of the time for calculating the current offset value
com_u2_mtr_pp	Number of the pole pairs of the motor to be driven
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 (rpm)
com_f4_speed_rate_limit_rpm	Maximum increment and decrement width (mechanical angle) for the speed command (rpm/s) (for use when speed control is enabled)
com_f4_overspeed_limit_rpm	Speed limit value (mechanical angle) (rpm)
com_u1_flag_volt_err_comp_use	Voltage error compensation setting 0: Disable, 1: Enable
com_f4_frequency_output_max	Maximum output frequency (Hz)
com_f4_rated_frequency	Rated motor frequency (Hz)
com_f4_voltage_output_max	Maximum output voltage (V)
com_f4_rated_voltage	Rated motor voltage (V)
com_f4_torque_boost	Torque boost compensation value (1% = 0.01)

## 6.9 Operating the Motor

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

### (a) Writing the sample program

The sample program having been written to the CPU board is assumed. For the method of writing the sample program, refer to section 6.4.

### (b) Turning on the power supply

The inverter having been connected to your PC via MC-COM by using a USB cable is assumed. Supply three-phase 200-VAC 50-Hz power to the inverter. After the power is turned on, the relay for preventing an inrush current is turned on, after which the bus voltage is automatically stabilized at a full-wave rectified value.

### (c) Starting the RMW

Start the RMW installed in your PC. After that, select the COM port corresponding to the CPU board and select "Analyzer" to establish a connection.

### (d) Starting rotation of the motor

Confirm that no error has occurred and then follow the steps on the following page.

- (1) Click on the [Read] button and confirm that a voltage of approximately 280 V or higher is being applied as "g\_st\_imvf.f4\_vdc\_ad".
- (2) Confirm that "g\_st\_imvf.u2\_error\_status" and "g\_st\_relay\_manager.u2\_error\_status" are 0. If they are not 0, perform the operation described in (f) on the following page to clear the error state.
- (3) Confirm that the check boxes in the [W?] column are selected in the "com\_u1\_system\_mode" and "com\_f4\_ref\_speed\_rpm" rows.
- (4) In the "com\_f4\_ref\_speed\_rpm" row, enter the command rotation speed in the [Write] column.
- (5) In the "com\_u1\_system\_mode" row, enter "1" in the [Write] column.
- (6) Click on the [Write] button.
- (7) Confirm that the motor has started rotation.

**(6) Click on the [Write] button.**

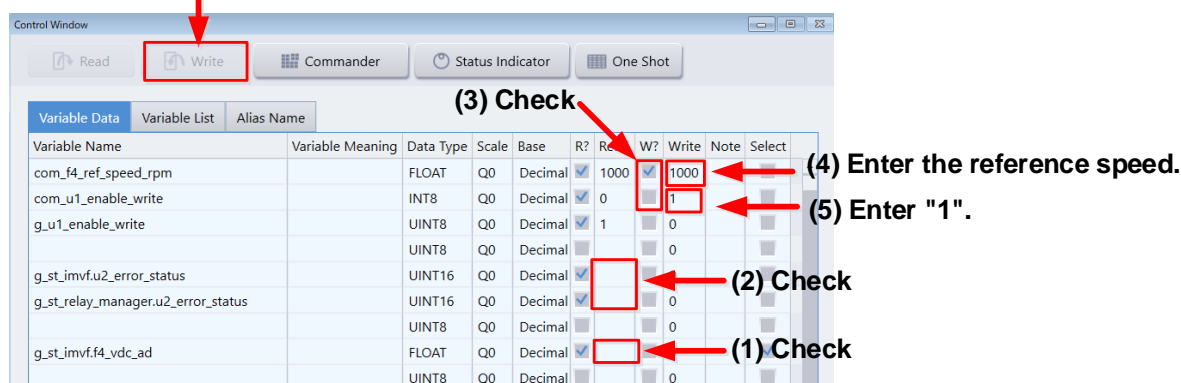


Figure 6-5 Procedure for Starting Rotation of the Motor

#### (e) Stopping the motor

Follow the steps below to stop the motor.

- (1) In the "com\_u1\_system\_mode" row, enter "0" in the [Write] column.
- (2) Click on the [Write] button.
- (3) Confirm that the motor has stopped.

**(2) Click on the [Write] button.**

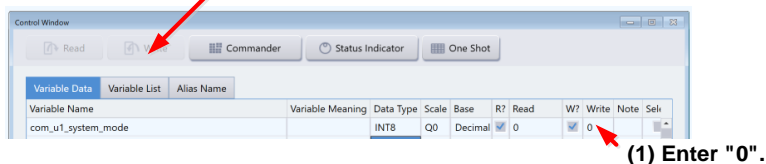


Figure 6-6 Procedure for Stopping the Motor

## (f) What to do in case of the motor stopping (due to an error)

The error condition can be cleared by performing the steps below. After that, operation can be resumed. Note that you cannot proceed with the motor operation in (d) unless you have not performed the following steps.

- (1) For an error in motor control, check the values of "g\_st\_imvf.u2\_error\_status" and "g\_st\_relay\_manager.u2\_error\_status" while referring to Table 6-6 and Table 6-7 and take action in response to the cause of the error.
- (2) In the "com\_u1\_system\_mode" row, enter "3" in the [Write] column.
- (3) Click on the [Write] button.

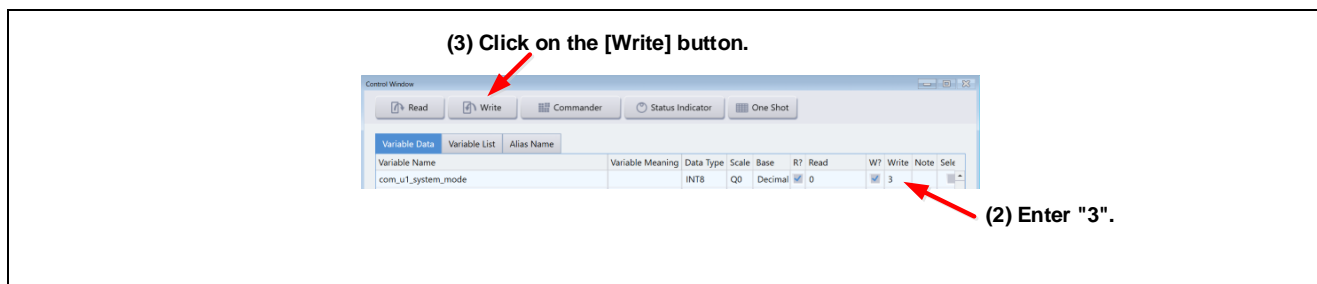


Figure 6-7 Procedure for Clearing the Error Condition

Table 6-6 Description of Errors in Motor Control

Value	Error Description	Assigned Macro Name
0x0000	No error	MOTOR_IMVF_ERROR_NONE
0x0001	Hardware overcurrent error	MOTOR_IMVF_ERROR_OVER_CURRENT_HW
0x0002	Overvoltage error	MOTOR_IMVF_ERROR_OVER_VOLTAGE
0x0004	Overspeed error	MOTOR_IMVF_ERROR_OVER_SPEED
0x0008, 0x0010, 0x0020, 0x0040	Reserved	—
0x0080	Low-voltage error	MOTOR_IMVF_ERROR_LOW_VOLTAGE
0x0100	Software overcurrent error	MOTOR_IMVF_ERROR_OVER_CURRENT_SW
0x0200	Reserved	—
0x0400	Relay control error	MOTOR_IMVF_ERROR_RELAY
0xffff	Undefined error	MOTOR_IMVF_ERROR_UNKNOWN

Table 6-7 Description of Errors in Relay Control

Value	Error Description	Assigned Macro Name
0x0000	No error	RELAY_MANAGER_ERROR_NONE
0x0001, 0x0002, 0x0004, 0x0008, 0x0010	Reserved	—
0x0020	Overheat error	RELAY_MANAGER_ERROR_OVER_HEATING
0xffff	Undefined error	MOTOR_IMVF_ERROR_UNKNOWN

## 6.10 Stopping and Shutting Down the Motor

To stop the operating motor, follow the steps below. In an emergency, prioritize step (2) and stop supplying 200 VAC.

- (1) Perform the procedure for stopping the motor described in 6.9 (e).
- (2) After confirming that the motor has stopped, operate the power breaker to stop the supply of power.
- (3) To prevent an electrical shock after the motor has stopped, do not connect or disconnect wires or cables or open the casing of the inverter before confirming that the bus voltage has dropped to less than 42 VDC.

## 7. Motor Control Algorithms

### 7.1 V/f Control for an Induction Motor

For users who are unfamiliar with induction motors, section 7.1.1 explains the principles of torque generation in an induction motor and "slip", which is a unique characteristic of induction motors, and section 7.1.2 gives an overview of V/f control.

#### 7.1.1 Principles of Torque Generation and Slip in an Induction Motor

The principles of torque generation in a three-phase squirrel-cage induction motor can be explained by using Fleming's left-hand and right-hand rules.

- (1) Applying three-phase AC voltage to the stator of an induction motor generates a rotating flux in the same way as in a permanent magnet (PM) motor.
- (2) When the rotating flux crosses the cage-shaped conductor, a voltage is induced in the conductor according to Fleming's right-hand rule.
- (3) The inductive voltage causes a current (a secondary current) to flow through the cage-shaped conductor.
- (4) An electromagnetic force is generated from the secondary current and rotating magnetic field according to Fleming's left-hand rule and the motor rotates.

Here, voltage induction occurs due to the relative movement between the rotating magnetic field and the cage-shaped conductor. Therefore, rotation of the rotor due to the torque thus generated is slower than that of the rotating magnetic field. This relative movement is generally called slip,  $s$ . This slip,  $s$ , is calculated from the speeds of rotation (synchronous speed)  $\omega$  of the rotating magnetic field, of rotation  $\omega_r$  of the rotor, and of slip  $\omega_s$  by using the following equation.

$$\text{Slip } s = \frac{\omega - \omega_r}{\omega} = \frac{\omega_s}{\omega}$$

$s$ : Slip,  $\omega_r$ : Rotation speed of the rotor,  $\omega$ : Synchronous speed,  $\omega_s$ : Slip speed

The general relationship between the torque and slip in an induction motor operating at the rated voltage and frequency is shown in Figure 7-1. In the absence of a load, the rotation speed of the rotor is equal to the synchronous speed and the slip becomes 0. Applying a load increases the torque and the slip also increases. The figure shows that the relationship between the slip and torque is close to linear until the torque reaches the rated value. Even when the rated load is applied, the slip is several %, which indicates that the rotor is rotating at a speed close to the synchronous speed.

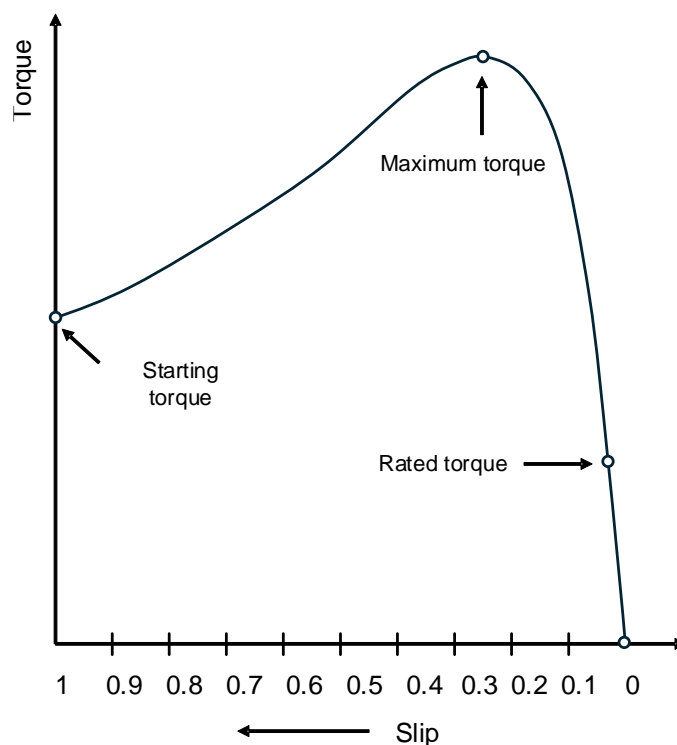
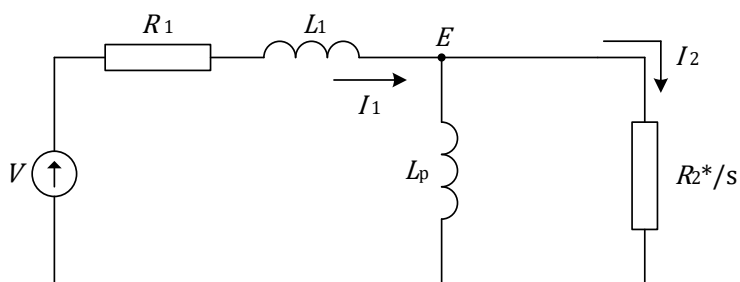


Figure 7-1 General Relationship between the Slip and Torque in an Induction Motor

### 7.1.2 Overview of V/f Control

V/f control is a type of average value control and an equivalent circuit is often used in describing the characteristics of the control. The equivalent circuit of an induction motor can be drawn by arranging the equivalent circuit of a transformer as shown in Figure 7-2. Note that the equivalent circuit shown is only for a single phase of an induction motor. The number of phases needs to be taken into account in the calculation of output.



$V$ : Stator voltage (V),  $R_1$ : Primary winding resistance ( $\Omega$ ),  $L_1$ : Primary inductance (H),  
 $L_p$ : Excitation inductance (H),  $R_2^*$ : Equivalent secondary resistance ( $\Omega$ ),  $s$ : Slip,  $E$ : Excitation voltage (V)  
 $I_1$ : Primary current (A),  $I_2$ : Secondary current (A)

Figure 7-2 Simplified Equivalent Circuit Diagram for a Single Phase of an Induction Motor in the Stationary State

V/f control is for control to obtain a constant ratio of the stator voltage  $V$  to its frequency  $f$ . Here,  $L_p$  in the equivalent circuit is an excitation inductance, that is, the inductance component related to generation of a rotating flux, and the rotating flux is expressed by the following equation, which does not take the voltage drop due to the primary winding resistance  $R_1$  and primary leakage inductance  $L_1$  into account.

$$\Phi = \frac{E}{2\pi f} \cong \frac{V}{2\pi f}$$

$\Phi$ : Magnetic flux,  $E$ : Excitation voltage,  $V$ : Stator voltage,  $f$ : Output frequency of the inverter

This equation shows that flux  $\Phi$  can be controlled to be constant by obtaining a fixed  $V/f$  ratio.

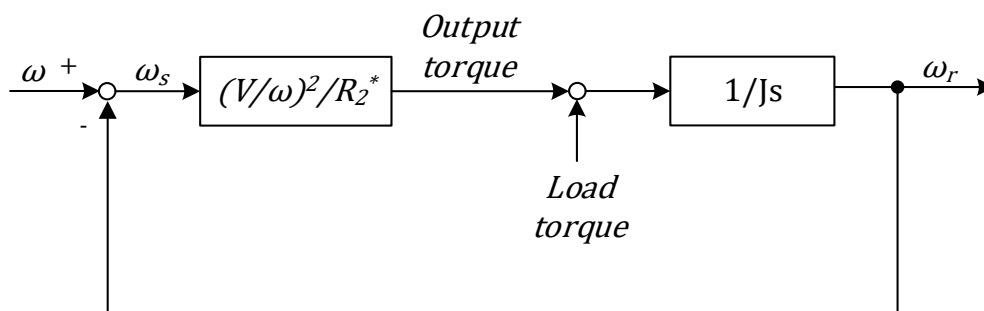
The output torque is generated by interaction between the secondary current and rotating flux and is expressed by the following equation.

$$T = I_2 \times \Phi \cong \left(\frac{V}{\omega}\right)^2 \frac{\omega_s}{R_2^*} = \frac{\Phi^2 \omega_s}{R_2^*}$$

$T$ : Output torque,  $I_2$ : Secondary current,  $\Phi$ : Magnetic flux,  $R_2^*$ : Secondary resistance,  $\omega_s$ : Slip frequency

Proportionality of the torque to the slip frequency and the square of the magnetic flux and that the torque can be made proportional to the slip by controlling the magnetic flux to be constant can be confirmed from this equation. Therefore, V/f control can easily be used to control the speed to be close to the commanded value over a wide range of speed by simply changing the stator voltage  $V$  in response to the speed command value  $f$ .

In addition, when the load torque increases under V/f control, the rotation speed of the motor decreases and the slip increases as shown in the block diagram of Figure 7-3. As the slip increases, the equivalent negative feedback operation is generated in the motor for the output of a larger torque. Therefore, V/f control can handle changes in the load torque although it does not involve direct control over the state of the load on the motor.



$J$ : Moment of inertia ( $\text{kgm}^2$ ),  $s$ : Laplace operator

Figure 7-3 Negative Feedback Operation and the Relationship between the Load Torque, Slip, and Rotation Speed

## 7.2 Overview of the Implemented Functions

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

Table 7-1 Motor Control Functions of This Sample Program

Function	Description
Control method	V/f control
Pulse-width modulation (PWM) method	Space vector modulation method (sinusoidal modulation can also be selected)
Control mode	Only speed control
Compensation functions	<ul style="list-style-type: none"> <li>• Voltage error compensation</li> <li>• Sample delay compensation</li> <li>• Torque boost compensation</li> <li>• Flux weakening control</li> </ul>

## 7.3 Control Block Diagram

Figure 7-4 shows a sample block diagram of the overall V/f control system.

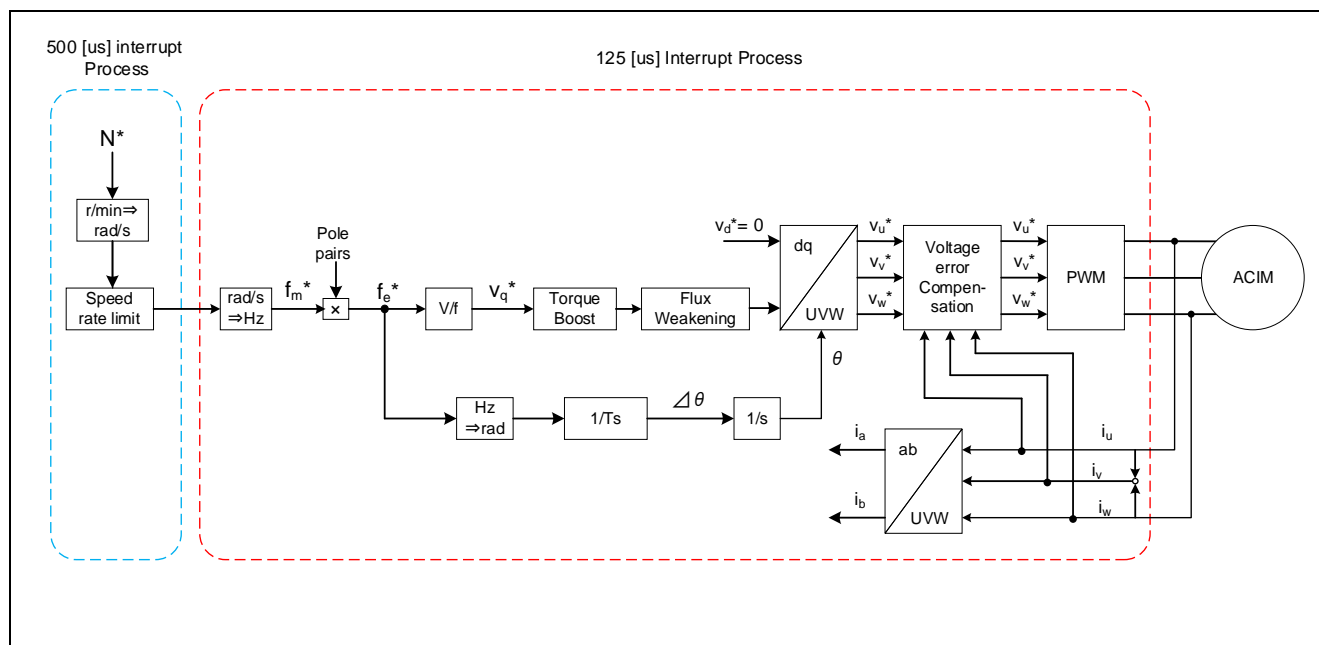


Figure 7-4 Schematic Block Diagram of V/f Control

## 7.4 Speed Control Function

The speed control function creates the speed commands for the motor. The increases and decreases of the speed are adjusted within a limited range to prevent sudden acceleration or deceleration.

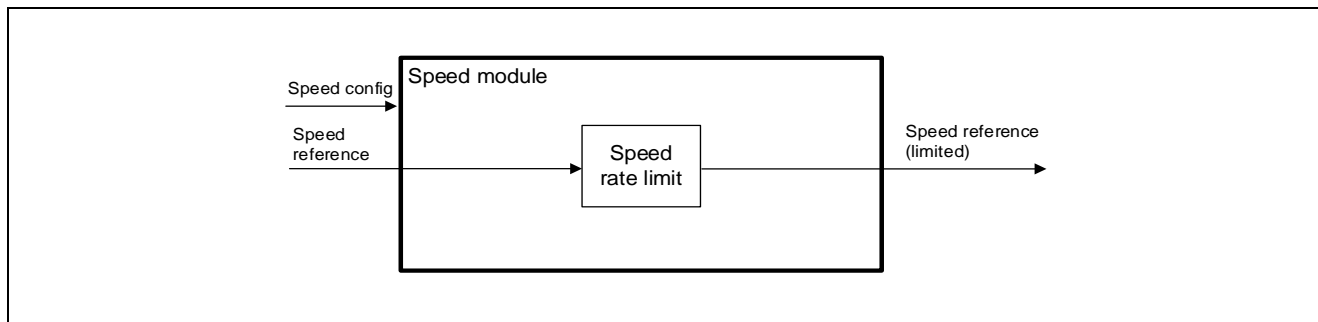


Figure 7-5 Functional Block Diagram of Speed Control

## 7.5 Current Control Function

The current control function uses the values of the input frequency commands to generate the angle and voltage commands that are necessary for V/f control and then calculates the voltage of the PWM outputs. Figure 7-6 shows the configuration of the module.

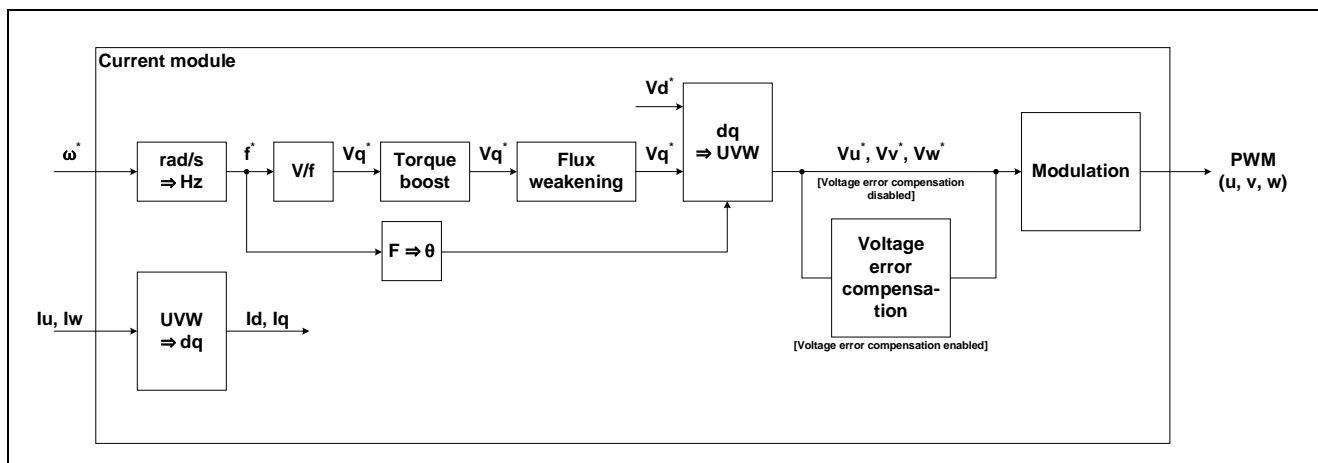


Figure 7-6 Functional Block Diagram for Current Control

## 7.6 V/f Control

The following describes the typical procedure of V/f control processing. The control commands for inverters in general that are used with V/f control are usually electrical angle frequency commands (Hz). However, the electrical angle frequency commands require conversion to mechanical angles to obtain the actual speed of motor rotation. In this sample program, speed commands are received as mechanical angles (rpm) and internally converted to frequency commands (Hz).

$$f_{ref} = \frac{N_{ref}}{60} * P$$

$f_{ref}$ : Frequency command (Hz),  $N_{ref}$ : Speed command (rpm), P: Number of pole pairs

In the current-control loop, two signals, voltage command and angle command signals, are generated from the frequency commands. In control of the induction motor, the position of the rotor is internally generated

and controlled by the inverter. The angles required for motor rotation are generated from the frequency commands.

The following equation is used to obtain the voltage commands. The motor in use determines the rated frequency and voltage. Note that the lower limit on the voltage command values is determined through torque boost compensation, to be described in section 7.7.

$$V_{ref} = \frac{V_{rated}}{f_{rated}} \times f_{ref}$$

$f_{rated}$ : Rated frequency (Hz),  $V_{rated}$ : Rated voltage (Vrms)

The angle command  $\theta$  (rad) is calculated by obtaining and integrating  $\Delta\theta$  (rad). The angle  $\Delta\theta$ , which is the angle to move per sampling interval, can be obtained by dividing the frequency command (Hz) by the sampling interval for angle generation (current control interval in most cases, otherwise, speed control interval).

$$\Delta\theta = 2\pi \times \frac{f_{ref}}{T_s}$$

$$\theta = \int \Delta\theta$$

The value of the voltage command is used as  $V_q$  and an inverse Park transformation (inverse dq transformation) based on the angle command value  $\theta$  is used to obtain the  $\alpha$  and  $\beta$  axes of the stator coordinate system from the d and q axes.

The value of the voltage command has been multiplied by  $\sqrt{(2/3)}$  through absolute conversion from  $V_q^*$  to  $V_u^*$ ,  $V_v^*$ , and  $V_w^*$ . This multiplication factor is the same as that used for conversion from the rms voltage between lines to the peak phase voltage and matches that used with settings for the GPT or MTU to obtain the phase voltage command values. Therefore, this value can be used as it is.

The obtained  $V_u^*$ ,  $V_v^*$ , and  $V_w^*$  values are divided by the bus voltage in the same way as in general PWM control for an inverter to obtain the modulation rate (duty cycle).

## 7.7 Torque Boost Compensation

In low-speed operation, the voltage drop due to the winding resistance in the induction motor comes to have such a large effect that sufficient magnetic flux cannot be generated by the voltage between lines, which leads to a smaller torque. To prevent such a situation, by compensating the reference voltage showed in Figure 7-7 Characteristic of Torque Boost Compensation and V/f torque in low speed can be increased. This function is called torque boost compensation.

Setting larger values for torque compensation may increase torque in low-speed, on the other hand, larger current flows. Therefore, settings must be adjusted appropriately in the range from approximately 0 to 5%.

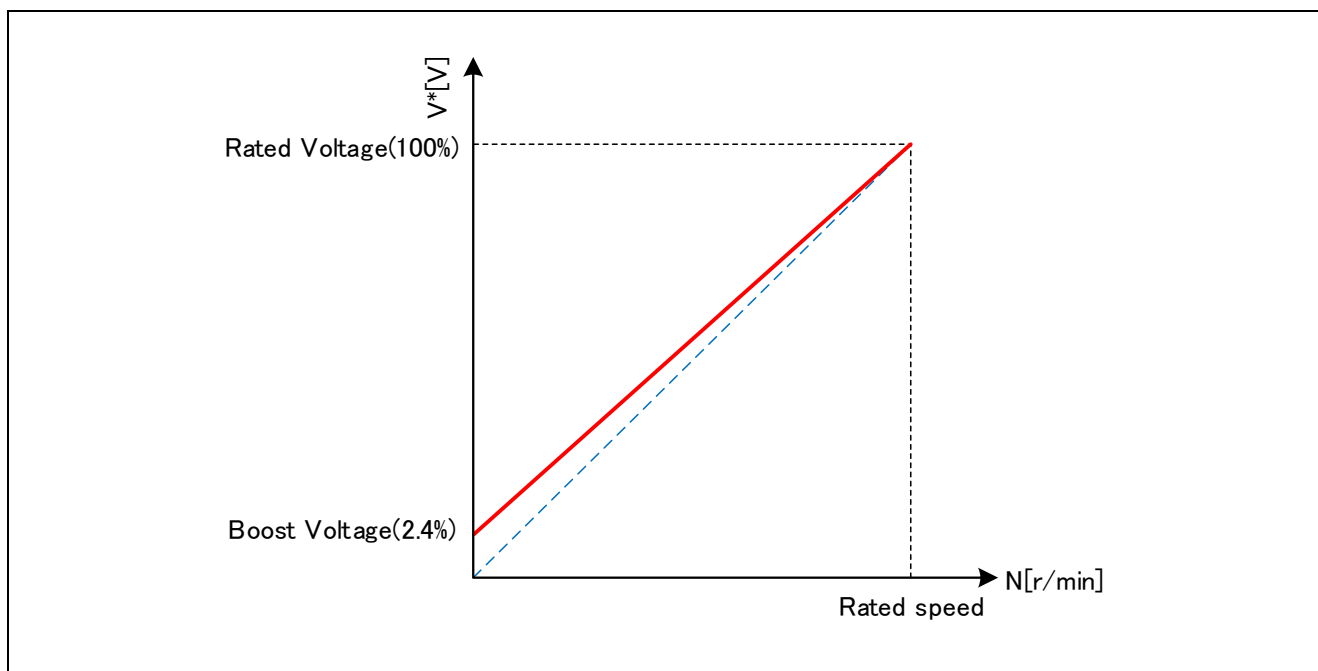


Figure 7-7 Characteristic of Torque Boost Compensation and V/f

## 7.8 Flux Weakening Control

Flux weakening control enables the operation of a motor in a higher range of speeds than the rated speed. In this operation, the product of the rotation speed of the rotor and the torque is limited to the rated output, so the torque requires control to make it inversely proportional to the rotation speed. This function is implemented in V/f control by limiting the applied voltage to the rated voltage when speeds enter a higher range than the rated speed.

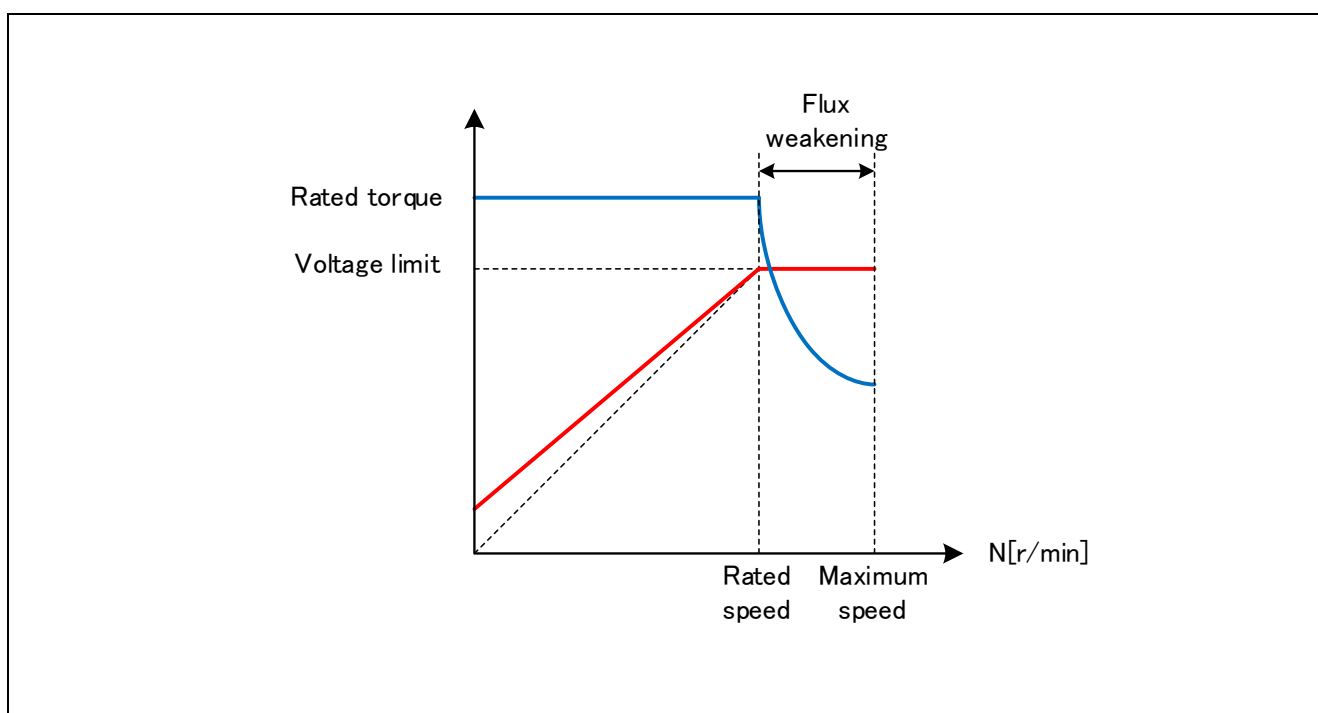


Figure 7-8 Flux Weakening Control

## 7.9 Sample Delay Compensation

To generate three-phase voltage commands for the U, V, and W phases, two-phase to three-phase conversion is performed with the angle advanced by 0.5 of the control interval from the estimated angle. This process improves the stability of control. For high-speed rotation applications or when the PWM carrier cycle is short, the motor control processing is skipped.

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

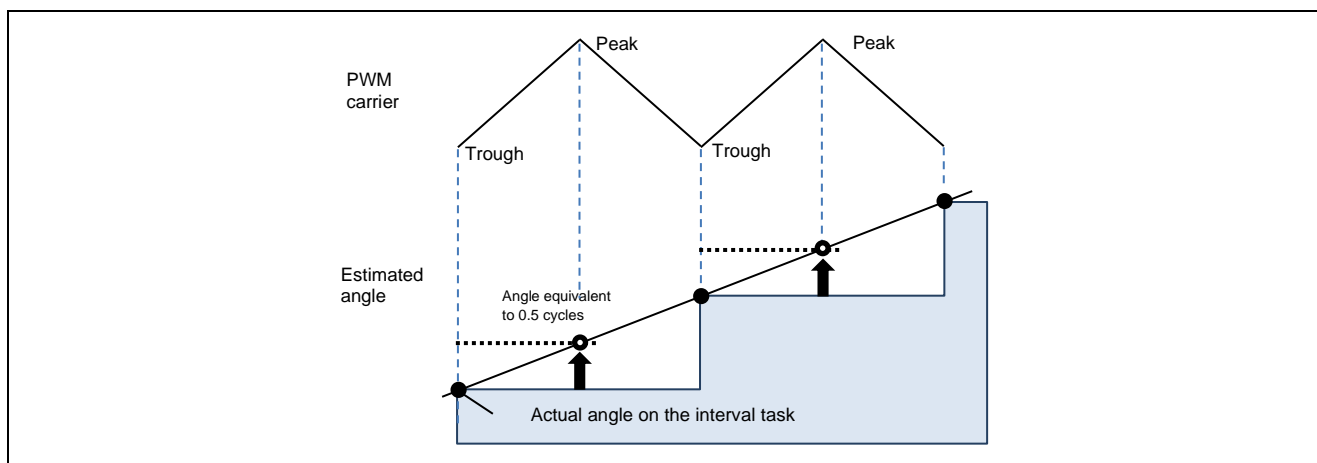


Figure 7-9 Example of the Amount by which the Angle is Advanced in a PWM Carrier Cycle

## 7.10 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 voltage that is actually being applied to the motor, degrading the accuracy of control. Voltage error compensation is implemented to reduce this error.

The voltage error depends on the current (direction and magnitude), dead time, and the switching characteristics of the power elements to be used and this dependency 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.

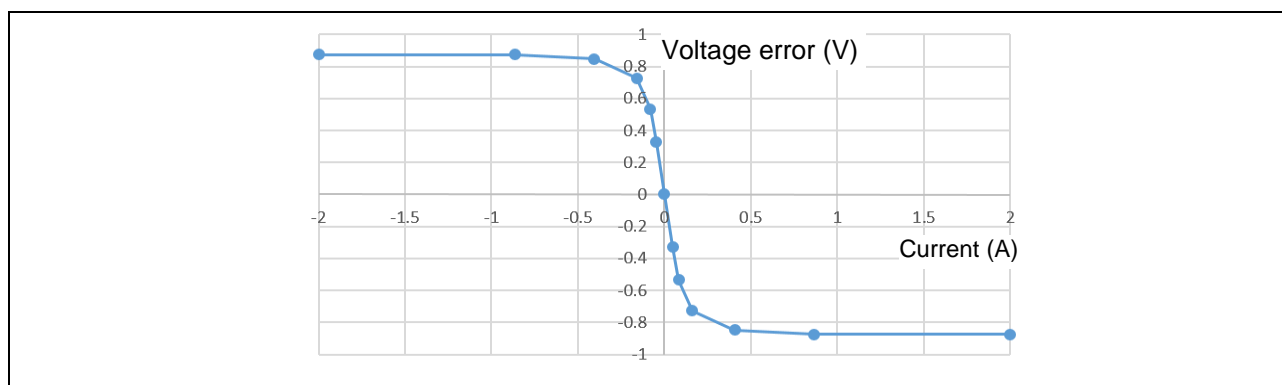


Figure 7-10 Example of the Dependency of the Voltage Error on the Current

### 7.11 Pulse-Width Modulation (PWM) Mode

In the sample program, the voltage to be input to the motor is generated by pulse-width modulation (PWM). A module in this sample program calculates the PWM duty cycle. In addition, a modulated voltage can be output to improve the voltage utilization. The modulation operation is specified through 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)

In vector control of a permanent magnet synchronous motor, the desired voltage command values for each phase will generally be generated sinusoidally. The voltage utilization as applied to the motor (in terms of the voltage between lines) is limited to a maximum of 86.7% with respect to the inverter bus voltage. When the sinusoidal modulation mode is used, the voltage unitization does not reach 100%, so the best performance may not be obtained from the inverter.

The modulation rate  $m$  is defined as follows in this mode.

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

M: Modulation rate   V: Command value voltage   E: Inverter bus voltage

#### (b) Space vector modulation (MOD\_METHOD\_SVPWM)

In the sinusoidal modulation mode, if the generated value is used as-is for the modulation wave for PWM generation, the voltage utilization as applied to the motor (in terms of the voltage between lines) is limited to a maximum of 86.7% with respect to the inverter bus voltage.

Therefore, as shown in the following expression, the average of the maximum and minimum voltage command values is calculated for 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 the voltage utilization becomes 100% and the voltage between lines is unchanged.

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

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

$V_u, V_v, V_w$ : Voltage command values of U, V, and W phases

$V'_u, V'_v, V'_w$ : Voltage command values of U, V, and W phases for PWM generation (modulation wave)

The modulation rate  $m$  is defined as follows.

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

M: Modulation rate    $V'$ : Phase voltage command for PWM generation   E: Inverter bus voltage

## 8. Software Specifications and Configuration

### 8.1 Software Specifications

The following shows the basic specifications of this software.

Table 8-1 Basic Specifications of This Software

Item	Description	
Motor control method	V/f control	
Starting and stopping of motor control	Input from the RMW	
Input voltage	Three-phase 200 to 240 VAC at 50 or 60 Hz	
DC bus voltage	282.8 VDC (Three-phase 200 VAC input)	
PWM carrier frequencies	Motor control	8 kHz, 125- $\mu$ s cycle (interrupts in troughs)
PWM mode	Sinusoidal modulation mode or space vector modulation mode	
Dead time	3.0 $\mu$ s	
Control interval	Current control	125 $\mu$ s
	Speed control	500 $\mu$ s
	Relay control	1.0 ms
Speed range	CW: 0 to 1800 rpm CCW: 0 to -1800 rpm (at the rated frequency 60 Hz)	
Protective stop processing	<p>■ Software protection function</p> <p>The motor control signal outputs (six lines) will be deactivated when any of the following conditions is met.</p> <ol style="list-style-type: none"> <li>1. The current for any phase exceeds 30.94 A (calculated from inverter current limit) or the value calculated from the motor rated current (limit value for overcurrent protection; checking is at the current control interval).</li> <li>2. The bus voltage exceeds 430 V (limit value for overvoltage protection; checking is at the current control interval).</li> <li>3. The bus voltage is lower than 186 V (limit value for low-voltage protection; checking is at the current control interval).</li> <li>4. The rotational velocity exceeds 1800 rpm (limit value for overspeed protection; checking is at the current control interval).</li> <li>5. A low level is input to the abnormal IGBT temperature detection port pin (checking is at the current control interval).</li> </ol> <p>■ Hardware protection function</p> <p>The PWM output pins will be placed in the high-impedance state and the motor control signal outputs (six lines) will be deactivated when an external overcurrent detection signal (low-level input to the GTETRGD pin) is detected or short-circuiting of output signals is detected.</p>	

## 8.2 Overall Configuration of the Software

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

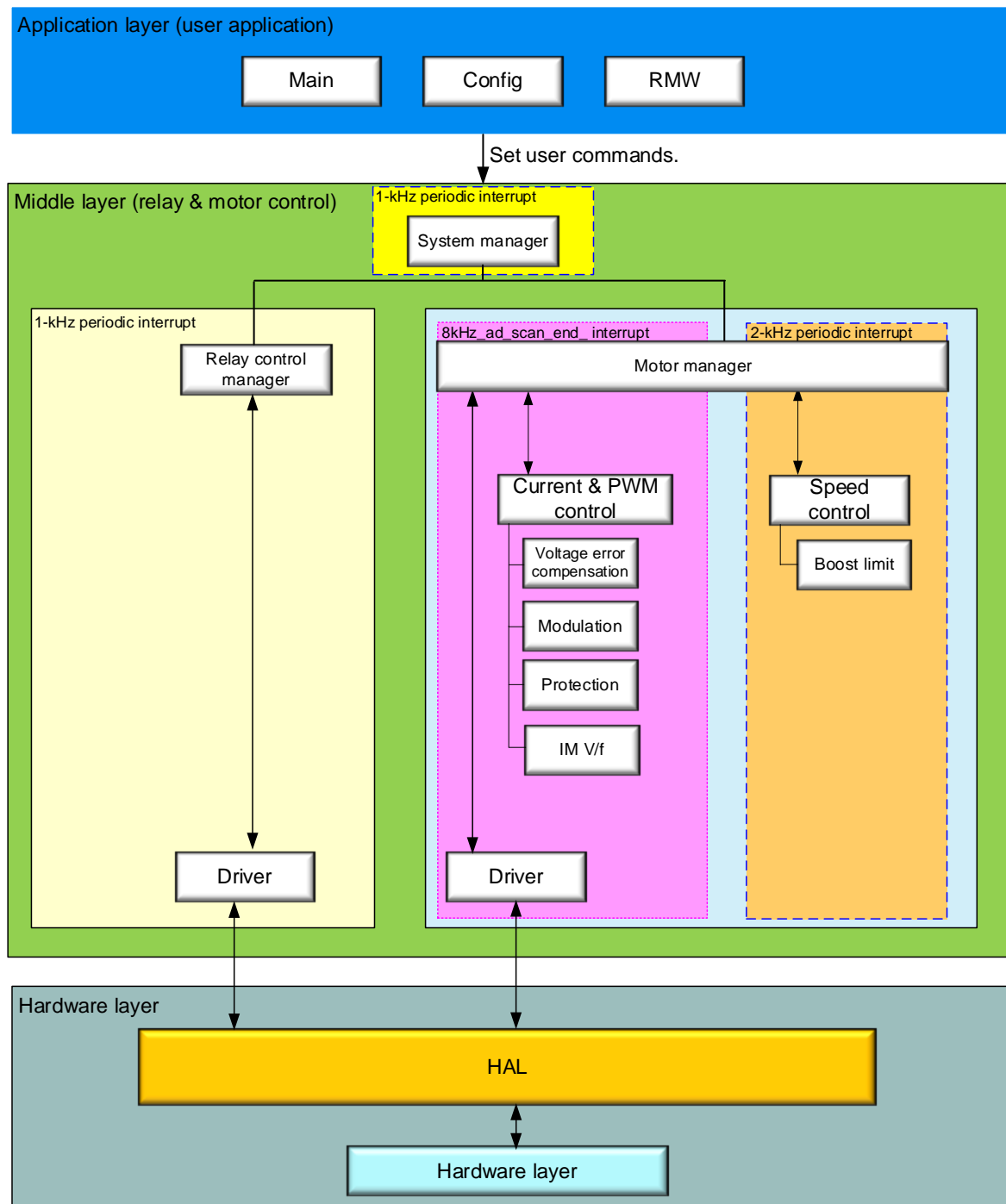


Figure 8-1 Overall Configuration of the Sample Program

### 8.3 Task Descriptions

For motor control, a task for speed control at 500- $\mu$ s intervals and that for voltage control (8 kHz, 125  $\mu$ s) are used.

Table 8-2 Interrupts and Tasks Used

Task	Peripheral Module	Interval	Interrupt Function	Description
500- $\mu$ s control interrupt (for speed control)	AGT0	500 $\mu$ s	callback_motor_speed_cyclic_timer	
A/D conversion end interrupt (for current control)	ADC0	125 $\mu$ s	callback_motor_current_cyclic_adc	This task runs upon an ADC conversion end interrupt.
1-ms periodic interrupt	AGT1	1 ms	callback_system_manager_cyclic_timer	
Reset	—		Note: Executed in the state transition processing when recovering from an error.	
POEG interrupt (hardware overcurrent error interrupt)	POEG		callback_system_manager_overcurrent_poe	Be sure to call R_POEG_Reset() from within the callback function for the POEG stack to reset the flag. If this is not done, the other processing may be stopped depending on the interrupt priority level.
RMW operation	—		r_app_rmw_ui_mainloop	

## 8.4 Configuration of Folders and Files

Table 8-3 shows the configuration of the folders and files of the sample program.

Table 8-3 Configuration of Folders and Files

Folder	Subfolder		File	Remarks
ra	—	—	FSP library and middleware body files	Modification of the files in these folders is prohibited.
ra_cfg	—	—	Header files of the FSP library settings	
ra_gen	—	—	HAL-related and automatically generated files	
script	—	—	Linker script files for the FSP	
src /application	main	—	mtr_interrupt.c/h	Main processing of the application
			mtr_main.c/h	
	mcu	ra6t2	r_app_mcu_callback.c	Definitions of the MCU-dependent callback functions
			r_app_mcu.c/h	Definitions of the wrapper functions related to MCU peripheral facilities
			r_motor_driver_fsp.c	MCU-dependent driver related to the motor module
			r_motor_driver_hal.h	HAL-dependent driver related to the motor module
			r_relay_driver_fsp.c	MCU-dependent driver related to the relay module
			r_relay_driver_hal.h	HAL-dependent driver related to the relay module
	motor_m odule	cfg	r_motor_inverter_cfg.h	Definitions of the inverter configuration
			r_motor_module_cfg.h	Definitions of the control module configuration
			r_motor_targetmotor_cfg.h	Definitions of the motor configuration
		current	r_motor_current_api.c/h	Definitions of API functions for the current control module
			r_motor_current_modulation.c/h	Definitions of functions for the modulation module
			r_motor_current_vf.c/h	Definitions of functions for the V/f module
			r_motor_current_volt_err_comp.lib/h	Voltage error compensation module
			r_motor_current.c/h	Definitions of local functions for the current control module
		driver	r_motor_driver.c/h	Definitions of functions for the driver module
		general	r_motor_common.h	Common definitions
			r_motor_filter.c/h	Definitions of general-purpose filter functions
		imvf	r_motor_imvf_action.c	Definitions of action functions
			r_motor_imvf_api.c/h	Definitions of API functions for the motor manager module
			r_motor_imvf_manager.c/h	Definitions of local functions for the manager module
			r_motor_imvf_protection.c/h	Definitions of functions for the protection facility
			r_motor_imvf_statemachine.c/h	Definitions of functions related to state transition
		speed	r_motor_speed_api.c/h	Definitions of API functions for the speed control module

Folder	Subfolder	File	Remarks
		r_motor_speed.c/h	Definitions of local functions for the speed control module
	relay_module	cfg	Definitions of the relay-related configuration
		driver	Relay-related driver module
		general	Common modules related to the relay module
		systask	Relay manager module
		r_relay_manager.c/h	
	system_module	system_manager	r_system_manager_api.c/h
			r_system_manager.c/h
	user_interface	ics	ICS2_RA6T2.o/h
			r_mtr_ics.c/h
			convert.bat
			ElfMapConverter.exe
			ICS2_RA6T2_Built_in.o
	—	—	hal_entry.c
Debug	—	—	Products of building

The FSP can be used to generate peripheral drivers easily through the GUI windows.

The FSP saves the settings information about the microcontroller, peripheral functions, pin functions, and other items that are used in the current project in a project file (configuration.xml). To check the settings of the peripheral functions for the sample program, see the FSP configuration window on the e<sup>2</sup> studio. The following table shows the configuration of the folders and files generated by the FSP.

Table 8-4 Configuration of Folders Generated by the FSP

Folder	Description
ra	This folder contains various module and library files related to the FSP. The folder is automatically generated and the configuration and contents of the subfolders and files it contains must not be changed.
ra_cfg	This folder contains the header files related to the FSP library settings. The folder is automatically generated and the configuration and contents of the subfolders and files it contains must not be changed.
ra_gen	This folder contains the hardware abstraction layer (HAL) files that serve as a bridge between the FSP library and user application. The values specified by the user through the FSP for use in the application are generated as modules. The subfolders and files in this folder are always automatically generated and their configuration and contents must not be changed.
script	This folder contains script files for registering the FSP modules to the linker.

## 8.5 Application Layer

The application layer is used for processing to control the system manager and RMW, which serves as the user interface, including the setting of command values for control by the system manager and updating of parameters for control modules. In this sample program, the RMW (RMW UI) is used for these settings and processes. This UI is 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 performed in the application layer.

Table 8-5 Functions Available in the Application Layer

Function	Description
Main processing	Enables or disables the operation of the application system in response to commands from the user.
RMW UI processing	Manages the RMW and acquires and sets parameters including command values.
Initial settings of the MCU	The FSP is used to make initial settings for the MCU. Calibration and other settings to suit the application also proceed.
Bridge to the FSP	Defines the callback functions assigned to peripheral functions, which are specified through the FSP, and passes them to lower-level modules through the system manager.

### 8.5.2 Structure and Variable Information

The variables that can be used by the user in the application layer are defined and managed in the system manager. For convenience of use of the sample software, they are also listed in Table 6-5 in section 6.8, Variables Used for Operating the RMW.

Table 8-6 lists the members of the structure provided for updating the motor module parameters by using the RMW.

When you use the RMW to specify a value for a variable, the application layer reflects the updated value in the variable in each of the relevant control modules by using the Update functions of the modules via the structure shown in Table 8-6.

Table 8-6 List of Variables in the Structure for Updating Parameters through the RMW

Structure	Variable	Description
st_rmw_param_buffer_t  Structure for updating parameters through the RMW	u2_offset_calc_time	Current offset detection time (s)
	st_motor	Structure of motor parameters
	f4_max_speed_rpm	Maximum speed (rpm) (mechanical angle)
	u1_ctrl_loop_mode	Control loop mode (speed control)
	f4_ref_speed_rpm	Speed command value (rpm) (mechanical angle)
	f4_speed_rate_limit_rpm	Speed variation limit (rpm/s) (mechanical angle)
	f4_overspeed_limit_rpm	Speed limit value (rpm) (mechanical angle)
	u1_flag_volt_err_comp_use	Enables or disables voltage error compensation.
	f4_frequency_output_max	Maximum output frequency (Hz) (electrical angle)
	f4_rated_frequency	Base frequency for V/f control (Hz) (electrical angle)

Structure	Variable	Description
	f4_voltage_output_max	Maximum output voltage (V)
	f4 Rated_voltage	Rated voltage for V/f control (V)
	f4_torque_boost	Torque boost compensation ratio (-)

### 8.5.3 Adjustment and Configuration of Parameters

The com variables used in the RMW are parameters that are only specifiable in the application layer. For the parameters used by the system manager, motor manager, or relay manager, refer to the corresponding sections.

During motor operation, adjust and configure variables through the RMW. For details about how to use the RMW, see section 6.7 and the Renesas Motor Workbench User's Manual (R21UZ0004).

## 8.6 System Manager

The system manager (r\_system\_manager) sets command values for the motor manager and relay manager and updates parameters for control modules on the basis of the command values, parameter settings, and callback notifications supplied by the application layer.

### 8.6.1 Functions

The following lists the functions of the system manager.

Table 8-7 List of Functions of the System Manger

Function	Description
Processing for monitoring the states of relay and motor operation and enabling and stopping motor operation	Monitors the states of relay and motor operation and enables or stops motor operation depending on the state on the relay control side.
Error canceling processing	Performs reset processing if an error has occurred on the relay control side or motor control side.
Interface between the application layer and manager layer	This function is placed between the application layer and motor manager to connect their interfaces. It also makes settings for the parameters input through RMW operations.

### 8.6.2 Module Configuration Diagram

Figure 8-2 shows the module configuration.

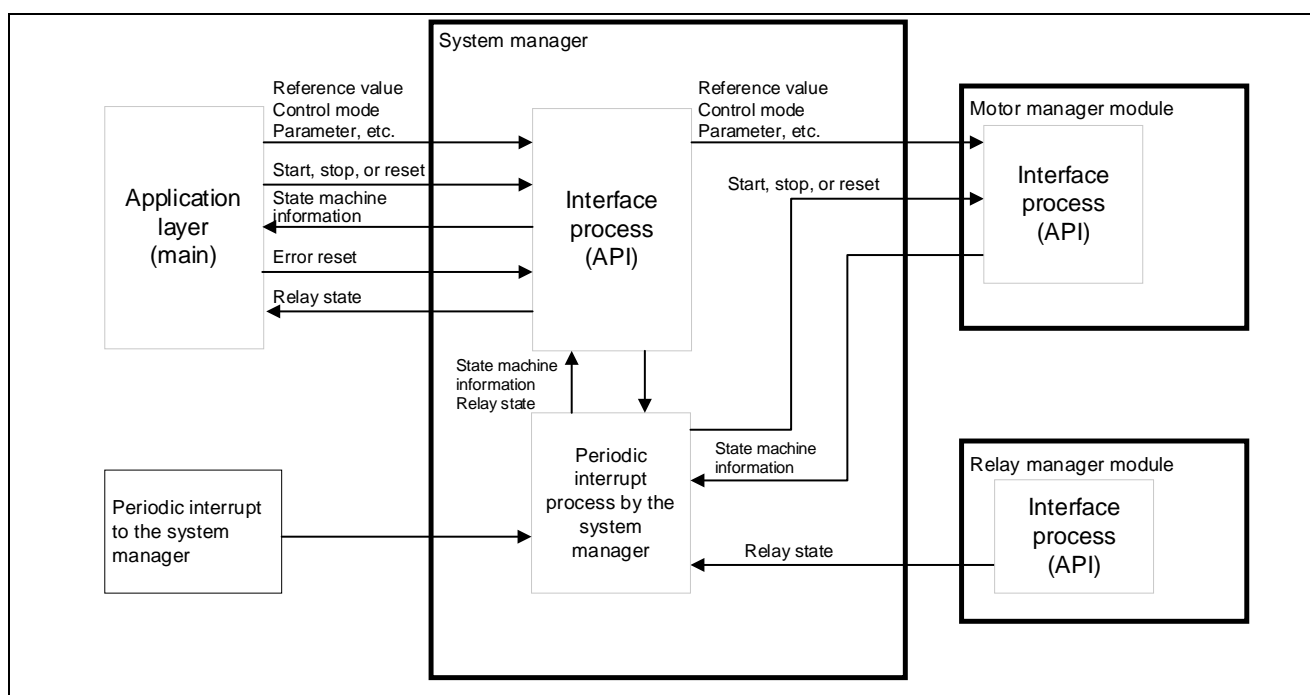


Figure 8-2 Module Configuration of the System Manager

## 8.7 Motor Manager and Motor Control Modules

The motor manager (r\_motor\_imvf\_manager) uses specific control modules (current control and speed control modules) to control the motor. Its processing includes the management and protection of the overall system for the interfaces with each of the modules and for motor control.

### 8.7.1 Functions

Table 8-8 lists the functions of the motor manager module. Table 8-9 and Table 8-10 list the functions of the motor control modules.

Table 8-8 List of Functions of the Motor Manger Module

Function	Description
Mode management	Switches the operating mode of the system in response to user commands for controlling the motor.
Protection function	Handles errors by using the system protection function.
Periodic current control processing	Performs processing in response to interrupts at the current control interval. The processing for error judgement and that related to communications with the RMW are started from this processing.
Periodic speed control processing	Performs processing in response to interrupts at the speed control interval.

Table 8-9 List of Functions of the Speed Control Module

Function	Description
Speed control	Applies the limit on the speed command values.
Speed command setting	Sets a speed command value in the speed control module.

Table 8-10 List of Functions of the Current Control Module

Function	Description
Current control	Calculates a voltage command value from the speed command value and sets PWM output values.
Current offset adjustment	Calculates the offset value of the current value detected by A/D conversion.
Voltage error compensation	Compensates for the effects of dead time on the output voltage.
Forward and inverse transformation	Transforms coordinates for the current value detected to perform vector control. This function also applies inverse transformation of coordinates to the calculation results to restore the original coordinate axes.
PWM modulation	Applies the desired frequency and voltage to the motor through the modulation of PWM signals.
V/f control	Creates a voltage command value from the frequency command value, rated voltage, and rated frequency.
Limit voltage vector	Limit voltage vector from detected bus voltage.
Torque boost function	Sets the lower limit on the voltage command value to compensate for the drop in the torque in the low-speed range.
Flux weakening control	Limits the q-axis voltage so that the motor operates with the rated output in a higher-speed range than the rated rotation speed.
Voltage phase lead compensation	Compensates for the delay in sampling by the current-control cycle in the generation of three-phase voltage command values.

## 8.7.2 Module Configuration Diagram

Figure 8-3 shows the module configuration.

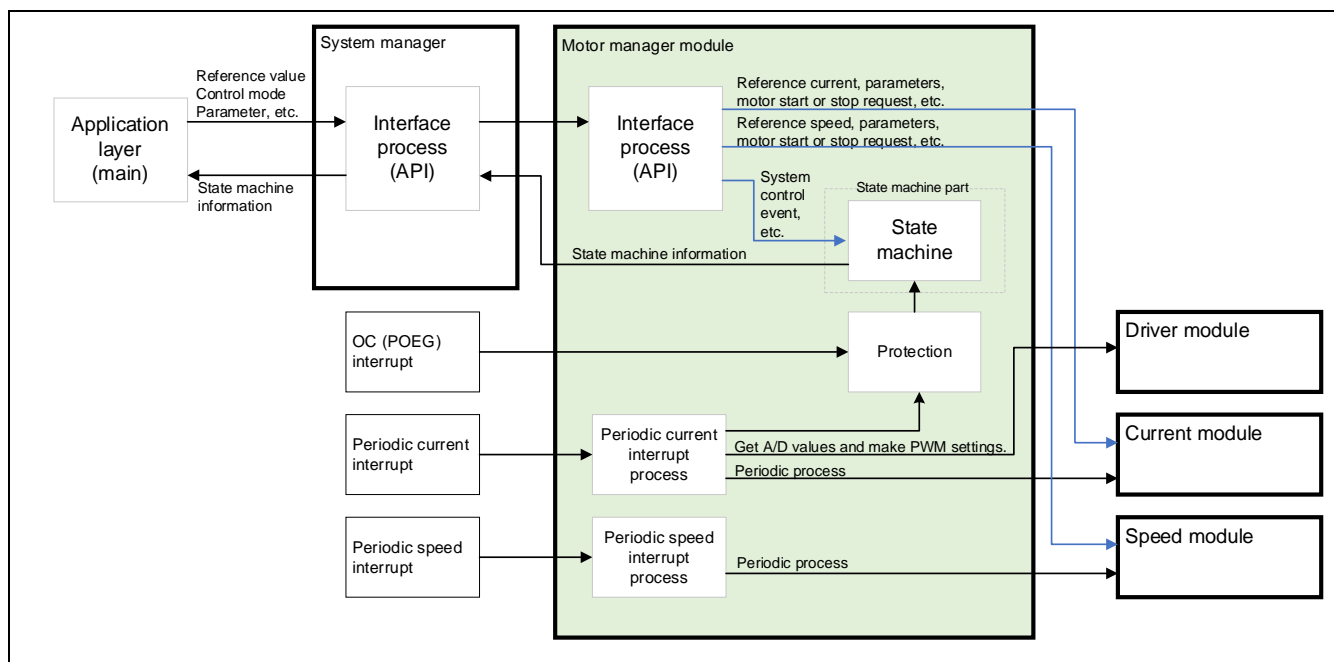


Figure 8-3 Module Configuration of the Motor Manager

### 8.7.3 Mode Management

Figure 8-4 shows the state transition diagram of this sample program. In this sample program, the states of control are managed by using two types of modes: system modes and run modes. Control Config indicates the control systems that are currently active in the software.

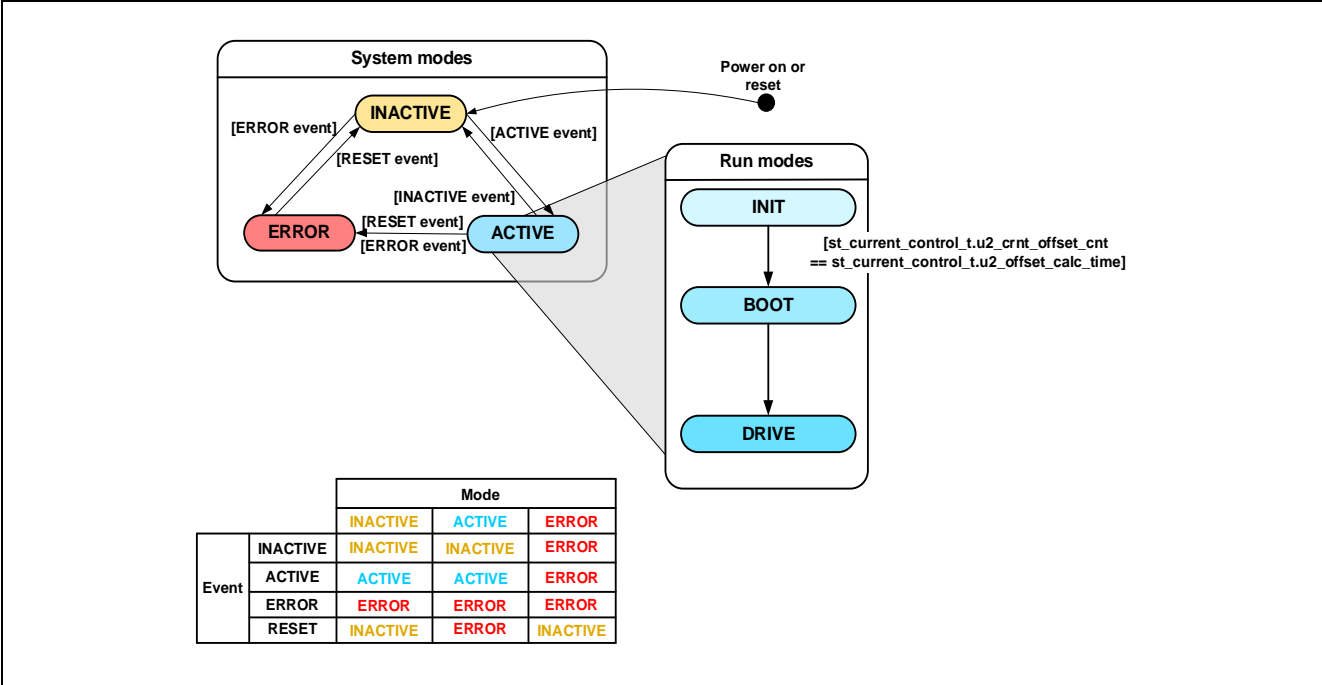


Figure 8-4 State Transition Diagram of the Motor Control Software

#### (1) System modes

The system modes indicate the state of system operation. The system makes a transition between states in response to the event corresponding to a new state occurring. There are three system modes: INACTIVE (the motor is stopped), ACTIVE (the motor is running), and ERROR (an error has occurred).

#### (2) Run modes

The run modes indicate the state of motor control. When the system enters ACTIVE mode, the motor makes a transition between run modes as shown in Figure 8-4.

#### (3) Events

The matrix table in Figure 8-4 shows how the system operation makes a transition between states in response to the event occurring in each system mode. The following table shows the trigger that causes each event to occur.

Table 8-11 List of Events

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

#### 8.7.4 Protection Function

This control program has the following error states and implements an emergency stop function in each error state.

##### (1) Software protection functions

The following errors are checked at 125- $\mu$ s monitoring intervals. If any of the conditions is satisfied, the outputs for the motor control signals are deactivated.

- Overcurrent error

The U-, V-, and W-phase currents are monitored at the current control interval. If an overcurrent (a current above the overcurrent limit value) is detected, the motor is brought to an emergency stop (software detection).

The overcurrent limit value is automatically calculated from the rated current of the motor (MOTOR\_CFG\_NOMINAL\_CURRENT\_RMS). If the limit value is greater than the rated current of the inverter (INVERTER\_CFG\_CURRENT\_LIMIT), the rated current is used as the limit.

- Overvoltage error

The bus voltage is monitored at the current control interval. If an overvoltage (a voltage above the overvoltage limit value) is detected, the motor is brought to an emergency stop. The overvoltage limit value is preset in consideration of conditions such as errors in the resistor value of the detection circuit.

- Low-voltage error

The bus voltage is monitored at the current control interval. If a low voltage (a voltage below the low-voltage limit value) is detected, the motor is brought to an emergency stop. The low-voltage limit value is preset in consideration of conditions such as errors in the resistor value of the detection circuit.

- Rotation speed error

The speed is monitored at the current control interval. If the rotation speed exceeds the speed limit value, the motor is brought to an emergency stop.

Note: See Table 8-1 for the settings of the limit values for the individual protection functions.

##### (2) POEG protection function (hardware protection)

Detection of an overcurrent error is driven by the generation of an external interrupt (POEG interrupt). Upon detection, the PWM output pins are placed in the high-impedance state.

## 8.7.5 API

Table 8-12 lists the API functions of the motor manager module.

Table 8-12 List of API Functions

API Function	Description
R_MOTOR_IMVF_Open	Generates instances of this module and the modules it is to use.
R_MOTOR_IMVF_Close	Places this module in the reset state.
R_MOTOR_IMVF_Reset	Initializes this module.
R_MOTOR_IMVF_ParameterUpdate	Updates the control parameter settings of this module. This function also updates the control parameters for the related modules.
R_MOTOR_IMVF_MotorStart	Places the motor in the running state.
R_MOTOR_IMVF_MotorStop	Places the motor in the stopped state.
R_MOTOR_IMVF_MotorReset	Releases the system from the error state.
R_MOTOR_IMVF_ErrorSet	Places the system in an error state.
R_MOTOR_IMVF_SpeedSet	Sets the speed command value.
R_MOTOR_IMVF_SpeedGet	Acquires the speed information.
R_MOTOR_IMVF_StatusGet	Acquires the state from the state machine.
R_MOTOR_IMVF_ErrorStatusGet	Acquires the error state.
R_MOTOR_IMVF_SpeedInterrupt	Performs interrupt processing for speed control.
R_MOTOR_IMVF_CurrentInterrupt	Performs interrupt processing for current control.
R_MOTOR_IMVF_OverCurrentInterrupt	Performs interrupt processing when an overcurrent is detected.

Table 8-13 List of API Functions of the Current Control Module

API Function	Description
R_MOTOR_CURRENT_Open	Generates an instance of the current control module.
R_MOTOR_CURRENT_Close	Places the current control module in the reset state.
R_MOTOR_CURRENT_Reset	Initializes the current control module.
R_MOTOR_CURRENT_Run	Activates the current control module.
R_MOTOR_CURRENT_ParameterSet	Specifies 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 for the current control module.
R_MOTOR_CURRENT_CurrentCyclic	Performs current control.
R_MOTOR_CURRENT_OffsetCalibration	Adjusts the offset for current detection.
R_MOTOR_CURRENT_CurrentOffsetRemove	Returns the detected current value with the offset value removed.
R_MOTOR_CURRENT_VoltErrCompParamSet	Sets the parameters for voltage error compensation.
R_MOTOR_CURRENT_RotorAngleSet	Updates the angle information for the current control module.

Table 8-14 List of API Functions of the Speed Control Module

API Function	Description
R_MOTOR_SPEED_Open	Generates an instance of the speed control module.
R_MOTOR_SPEED_Close	Places the speed control module in the reset state.
R_MOTOR_SPEED_Reset	Initializes the speed control module.
R_MOTOR_SPEED_Run	Activates the speed control module.
R_MOTOR_SPEED_ParameterSet	Specifies 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_ControlParamSet	Presets the information of the variables to be used for speed control.

## 8.7.6 Structure and Variable Information

Table 8-15 lists the structures and their member variables for the motor manager module. In this module, the structure for the motor manager module (g\_st\_imvf) is defined by the API function for securing an instance of the module. Table 8-16 lists the structures and their member variables that are used in the current control module. Table 8-17 lists the structures and their member variables used in 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 control module (g\_st\_sc) are defined by the API function for securing an instance of each module.

Table 8-15 List of Structures and Variables for the Motor Manager Module

Structure	Variable	Description
st_imvf_control_t  Structure of the variables for controlling the motor manager module	u1_state_speed_ref	State of the speed command value
	u1_direction	Rotation direction
	u1_ctrl_loop_mode	Control mode selection (speed or position)
	u2_error_status	Error state
	u2_run_mode	Run mode
	f4_vdc_ad	Bus voltage at the current control interval (V)
	f4_relay_vdc_ad	Bus voltage at the relay control interval (V)
	f4_iu_ad	U-phase current (A)
	f4_iv_ad	V-phase current (A)
	f4_iw_ad	W-phase current (A)
	f4_overcurrent_limit	Overcurrent limit value (A)
	f4_overvoltage_limit	Overvoltage limit value (V)
	f4_undervoltage_limit	Low-voltage limit value (V)
	f4_overspeed_limit_rad	Overspeed limit value (rad/s)
	u1_relay_first_on	Turning the relay on at the time of startup
	st_current_output	Structure for current control module output
	st_speed_output	Structure for speed control module output
	st_stm	Structure for the state machine
	st_motor	Structure of motor parameters
	*p_st_driver	Address of the structure of the variables for controlling the driver module
st_imvf_cfg_t  Structure for setting the parameters for controlling the motor manager module	*p_st_cc	Address of the structure of the variables for controlling the current control module
	*p_st_sc	Address of the structure of the variables for controlling the speed control module
	f4_overspeed_limit_rpm	Speed limit value (rpm) (mechanical angle)
	f4_ctrl_period	Speed control interval (ms)
	st_motor	Structure of motor parameters

Table 8-16 List of Structures and Variables for the Current Control Module

Structure	Variable	Description
st_current_control_t  Structure for the current control module	u1_active	Active state of the current control module
	u1_flag_volt_err_comp_use	Enables or disables the voltage error compensation function.
	u1_flag_offset_calc	Flag for current offset calculation
	u2_offset_calc_time	Measurement time setting in current offset adjustment
	u2_crnt_offset_cnt	Measurement count in current offset adjustment
	f4_ctrl_period	Current control interval (period) (s)
	f4_refu	U-phase command voltage (V)
	f4_refv	V-phase command voltage (V)
	f4_refw	W-phase command voltage (V)
	f4_vd_ref	d-axis voltage command value (V)
	f4_vq_ref	q-axis voltage command value (V)
	f4_id_ad	d-axis current value (A)
	f4_iq_ad	q-axis current value (A)
	f4_offset_iu	U-phase offset current value (A)
	f4_offset_iw	W-phase offset current value (A)
	f4_sum_iu_ad	U-phase total current value (A)
	f4_sum_iw_ad	W-phase total current value (A)
	f4_vdc_ad	Bus voltage detection value (V)
	f4_iu_ad	U-phase current detection value (A)
	f4_iv_ad	V-phase current detection value (A)
	f4_iw_ad	W-phase current detection value (A)
	f4_modu	U-phase duty cycle
	f4_modv	V-phase duty cycle
	f4_modw	W-phase duty cycle
	f4_speed_rad_ctrl	Speed command value (electrical angle) (rad/s)
	f4_va_max	Maximum voltage vector value (V)
	f4_ref_id_ctrl	d-axis current command value (A)
	f4_ref_iq_ctrl	q-axis current command value (A)
	st_mod	Structure for PWM modulation
	st_volt_comp	Structure for voltage error compensation
	st_rotor_angle	Structure of rotor information
	st_rotor_angle_phasecomp	Structure of rotor information (lead compensation)

Structure	Variable	Description
	st_motor	Structure of motor parameters
	st_current_imvf	Structure of the variables for IM V/f control
st_current_cfg_t  Structure for setting the parameters for controlling the current control module	u1_flag_volt_err_comp_use	Enables or disables the voltage error compensation function.
	u2_offset_calc_time	Current offset measurement time (number of measurements)
	f4_ctrl_period	Current control interval (ms)
	st_current_vf_cfg	Structure for V/f control configuration
	st_motor	Structure of motor parameters
st_current_output_t  Structure for the current control module output	u1_flag_offset_calc	Flag for indicating detection or non-detection of a current offset
	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_ref_id_ctrl	d-axis current command value
	f4_speed_rad	Estimated speed (rad/s)
st_current_input_t  Structure for the current control module input	f4_iu_ad	U-phase current detection value (A)
	f4_iv_ad	V-phase current detection value (A)
	f4_iw_ad	W-phase current detection value (A)
	f4_vdc_ad	Bus voltage detection value (V)
	f4_ref_speed_rad_ctrl	Speed (rad/s)
st_rotor_angle_t  Angle information for the current control module	f4_rotor_angle_rad	Rotor angle (electrical angle) (rad)
	f4_sin	Sine coefficient for reducing the number of calculations
	f4_cos	Cosine coefficient for reducing the number of calculations
st_current_vf_t  Structure of the variables for V/f control	f4_frequency_output_max	Maximum output frequency (Hz)
	f4 Rated frequency	Rated frequency (Hz)
	f4_voltage_output_max	Maximum output voltage (Vrms)
	f4 Rated voltage	Rated voltage (Vrms)
	f4_torque_boost	Torque boost (%)
	f4_vq_boost	Lower limit for torque boost compensation
	f4_ref_speed_freq_ctrl	Frequency command value (Hz)
st_current_vf_cfg_t  Structure for the V/f control configuration	f4_frequency_output_max	Maximum output frequency (Hz)
	f4 Rated frequency	Rated frequency (Hz)
	f4_voltage_output_max	Maximum output voltage (Vrms)
	f4 Rated voltage	Rated voltage (Vrms)

Structure	Variable	Description
	f4_torque_boost	Torque boost (%)
st_mod_t  Structure for PWM modulation	f4_vdc	Bus voltage (V)
	f4_1_div_vdc	Reciprocal of the bus voltage
	f4_voltage_error_ratio	Voltage error ratio
	f4_max_duty	Maximum duty cycle
	f4_min_duty	Minimum duty cycle
	f4_neutral_duty	Neutral duty cycle (duty cycle that produces 0-V output)
st_volt_comp_t  Structure for voltage error compensation	f4_comp_v[VERR_COMP_ARY_SIZE]	Table of voltages for voltage error compensation (V)
	f4_comp_i[VERR_COMP_ARY_SIZE]	Table of currents for voltage error compensation (A)
	f4_slope[VERR_COMP_ARY_SIZE+1]	Table of slopes of the linear interpolation equation for voltage error compensation (V/A)
	f4_intcept[VERR_COMP_ARY_SIZE+1]	Table of intercepts of the linear interpolation equation for voltage error compensation (V)
	f4_volt_comp_array[3]	Voltage error compensation values for the U, V, and W phases (V)
	f4_vdc	Bus voltage detection value (V)
	f4_volt_comp_limit	Limit on the voltage error compensation value (V)
	f4_volt_comp_limit_ratio	Limit on the voltage error compensation period
	u1_volt_err_comp_enable	Enabling or disabling of voltage error compensation
	u1_volt_comp_use_motor_type	Selection of three phases, two phases, or no phases for voltage error compensation

Table 8-17 List of Structures and Variables for the Speed Control Module

Structure	Variable	Description
st_speed_control_t	u1_active	Selection of whether to enable the module
Structure for the speed control module	u1_state_speed_ref	Variable for managing the states that determine the speed command value. The states to be managed are shown in section 8.7.7, Macro Definitions.
	f4_speed_ctrl_period	Speed loop control interval (s)
	f4_ref_speed_rad_ctrl	Speed command value for control (rad/s)
	f4_ref_speed_rad	Speed command value output by the position control module during position control (rad/s)
	f4_ref_speed_rad_manual	Speed command value set by the user during speed control (rad/s)
	f4_speed_rad_ctrl	Speed calculated by the speed control module (rad/s)
	f4_speed_rad	Speed to be 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_id_ad	d-axis current value (A)
	f4_iq_ad	q-axis current value (A)
	st_motor	Structure of motor parameters
st_speed_cfg_t	f4_max_speed_rpm	Maximum speed command value (electrical angle) (rpm)
	f4_speed_ctrl_period	Speed control interval (s)
	f4_speed_rate_limit_rpm	Speed variation limit per cycle (rad/s/step)
	st_motor	Structure of motor parameters
st_speed_input_t	u1_state_speed_ref	Speed command value control state
	f4_speed_rad	Speed (electrical angle) (rad/s)
st_speed_output_t	f4_ref_speed_rad_ctrl	Speed command value (electrical angle) (rad/s)

## 8.7.7 Macro Definitions

Table 8-18 lists the macros for the motor manager module.

Table 8-18 List of Macros

File Name	Macro Name	Defined Value	Description
r_motor_imvf_api.h	MOTOR_LOOP_POSITION	0	Position control mode Note: Not supported in this sample program
	MOTOR_LOOP_SPEED	1	Speed control mode
	MOTOR_IMVF_ERROR_NONE	0x0000	Error state There is no error.
	MOTOR_IMVF_ERROR_OVER_CURRENT_HW	0x0001	Error state A hardware overcurrent error has occurred.
	MOTOR_IMVF_ERROR_OVER_VOLTAGE	0x0002	Error state An overvoltage error has occurred.
	MOTOR_IMVF_ERROR_OVER_SPEED	0x0004	Error state An overspeed error has occurred.
	MOTOR_IMVF_ERROR_LOW_VOLTAGE	0x0080	Error state A low-voltage error has occurred.
	MOTOR_IMVF_ERROR_OVER_CURRENT_SW	0x0100	Error state A software overcurrent error has occurred.
	MOTOR_IMVF_ERROR_RELAY	0x0400	Error state A relay control error has occurred.
	MOTOR_IMVF_ERROR_UNKNOWN	0xFFFF	Error state An error whose cause is unknown has occurred.
r_motor_imvf_manager.h	MOTOR_MODE_INIT	0x00	Run mode for initialization
	MOTOR_MODE_BOOT	0x01	Run mode for preparation for driving
	MOTOR_MODE_DRIVE	0x02	Run mode for motor driving state
r_motor_imvf_statemachine.h	STATEMACHINE_STATE_STOP	0x00	State of the state machine: The motor has stopped.
	STATEMACHINE_STATE_RUN	0x01	State of the state machine: The motor is rotating.
	STATEMACHINE_STATE_ERROR	0x02	State of the state machine: An error has occurred.
	STATEMACHINE_STATE_SIZE	3	State machine (array of states)
	STATEMACHINE_EVENT_STOP	0x00	Switching of the state machine: Stopping the motor
	STATEMACHINE_EVENT_RUN	0x01	Switching of the state machine: Starting the motor
	STATEMACHINE_EVENT_ERROR	0x02	Switching of the state machine: Giving information on error settings
	STATEMACHINE_EVENT_RESET	0x03	Switching of the state machine: Releasing the motor from the error state
	STATEMACHINE_EVENT_SIZE	4	State machine (array of events)
	STATEMACHINE_ERROR_NONE	0x00	Flag for the state machine: No error
	STATEMACHINE_ERROR_EVENT_OUT_BOUND	0x01	Flag for the state machine: Access was outside the array of events.

File Name	Macro Name	Defined Value	Description
	STATEMACHINE_ERROR_STATE_OUT_BOUND	0x02	Flag for the state machine: Access was outside the array of states.
	STATEMACHINE_ERROR_ACTION_EXCEPTION	0x04	Flag for the state machine: Transition has failed.

## 8.8 Relay Manager and Relay Control Module

### 8.8.1 Functions

The relay manager (r\_relay\_manager) is activated by AGT interrupts, which are generated at a frequency of 1 kHz. The manager turns a relay on or off according to the detected bus voltage. The following table lists the functions of the relay manager.

Table 8-19 List of Functions of the Relay Manager

Function	Description
State transition (state machine) processing	Handles transitions between the relay operation, stop, and error states.
Driving of the relay	Turns the relay on or off in response to the bus voltage.
Periodic relay control processing	Monitors the bus voltage and controls the processing for turning the relay on or off.
Protection function	Detects an abnormal IGBT temperature error.

### 8.8.2 Module Configuration Diagram

Figure 8-5 shows the functional blocks of the relay manager.

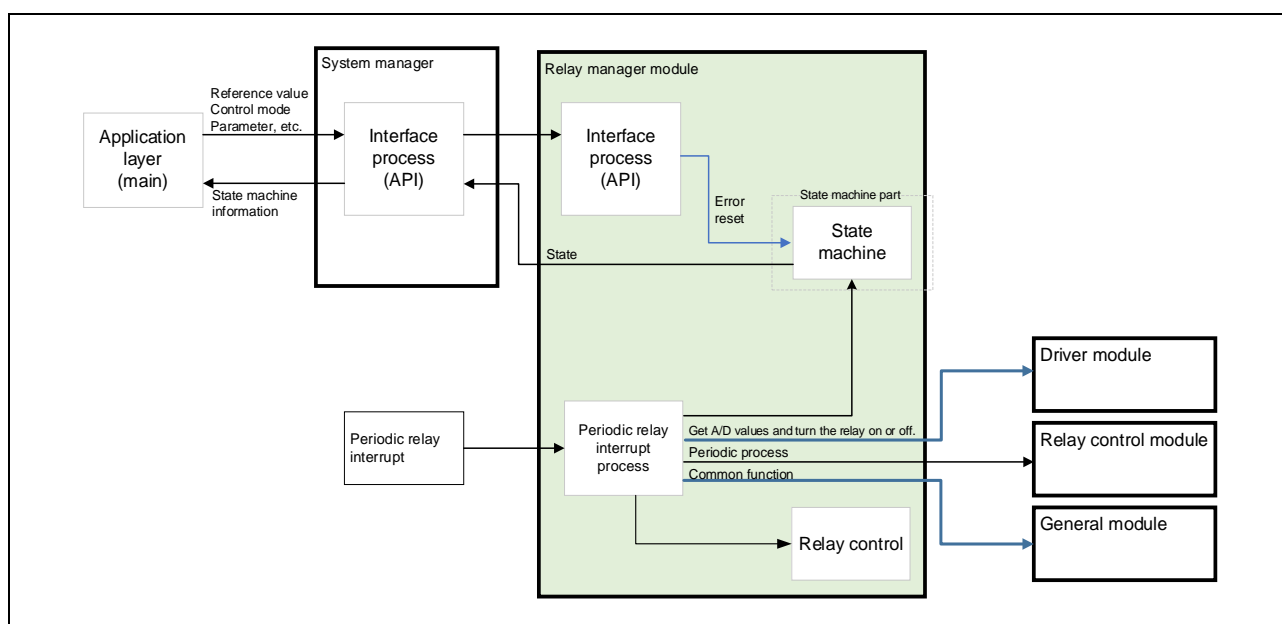


Figure 8-5 Functional Blocks of the Relay Manager

### 8.8.3 Sequence Descriptions

The relay is placed in the Idle state when the supply of power is started. The relay manager monitors the bus voltage and turns the relay on or off when the appropriate one of the two conditions described below is satisfied. In the case of the relay being shifted from the off (open) state to the on (closed) state, the transition is preceded by 100 ms of waiting. After the relay is placed in the RelayClose state, the manager enables motor operation. Figure 8-6 is a state transition diagram.

- Turning the relay on

After the bus voltage through the LPF remains at 230.0 VDC or higher and the deviation between the previous detected bus voltage remains within the range  $\pm 5.0$  V over the specified period (100 ms) while the

relay is off, the relay manager turns the relay on (by output of the high level through the PE01 pin). The motor operation is only enabled while the relay is on.

- Turning the relay off

After the bus voltage through the LPF remains at a voltage below 186.0 VDC for 60 ms while the relay is on, the relay manager turns the relay off (by output of the low level through the PE01 pin). Motor operation is not permitted while the relay is off. Therefore, if the relay is turned off during motor rotation, the manager stops the motor.

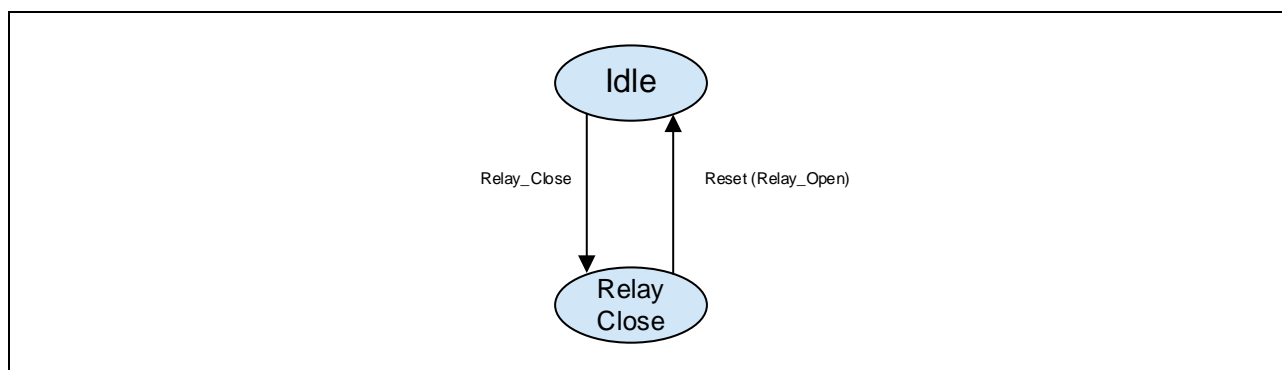


Figure 8-6 State Transition Diagram of the Relay

#### 8.8.4 Protection Function

This function judges the need for protective stopping due to an abnormal IGBT temperature error and handles error processing. If any of the states listed in Table 8-20 is detected, the PWM gate signals are cut off and processing to stop driving of the inverter proceeds.

Table 8-20 Error for Protective Stopping

Error	Pin	Detection Interval	Detection Level	Unit	Protective Operation
IGBT temperature error	PD07	Relay control interval	Low *	—	

Note: \* The level on the port pin is checked. The low level being detected on the pin indicates that an error has occurred.

#### 8.8.5 API

Table 8-21 lists the API functions of the relay manager module.

Table 8-21 List of API Functions of the Relay Manager

API Function	Description
R_RELAY_MANAGER_Open	Generates instances of this module and the modules it is to use.
R_RELAY_MANAGER_Close	Closes this module.
R_RELAY_MANAGER_Reset	Places this module in the reset state.
R_RELAY_MANAGER_ErrorCancel	Releases the system from the error state.
R_RELAY_MANAGER_StatusGet	Acquires the internal state.
R_RELAY_MANAGER_Main	Executes the interrupt processing required for processing by the relay manager.
R_RELAY_MANAGER_CurrentInterrupt	Checks abnormal temperature errors.

## 8.8.6 Structure and Variable Information

Table 8-22 lists the structure and its member variables for the relay manager module.

Table 8-22 List of Structure and Variables for the Relay Manager Module

Structure	Variable	Description
st_relay_manager_t  Structure of the variables for relay control	u1_relay_ctrl_on_flg	Flag for judging whether the relay is to be turned on or off
	u1_error_cancel_flg	Flag for requesting release from an error state
	u2_error_status	Error state
	u2_run_mode	Run mode
	f4_vdc_ad	Bus voltage detection value (V)
	f4_vdc_dt	Bus voltage deviation (V)
	f4_vdc_ad_lpf	Bus voltage detection value after LPF processing (V)
	f4_vdc_ad_last	Previous bus voltage detection value after LPF processing (V)
	f4_pre_lpf_output	Previous bus voltage detection value for LPF calculation (V)
	u1_dt_ct_relay_on	Counter for judging when to turn the relay on
	u1_dt_ct_relay_off	Counter for judging when to turn the relay off
	u1_overheat_detect_level	Pin level stored when an overheat error was detected
	p_st_relay_driver	Address of the structure of the variables for relay driver control

## 8.8.7 Macro Definitions

Table 8-23 lists the macros used by the relay manager.

Table 8-23 List of Macros Used by the Relay Manager

File Name	Macro Name	Defined Value	Description
r_relay_manager.h	RELAY_ON_VDC	230.0f	Bus voltage threshold for judging whether the relay is to be turned on (V)
	RELAY_ON_DT	5.0f	Bus voltage deviation for judging whether the relay is to be turned on (V)
	RELAY_ON_DT_CT	100	Period for judging whether the relay is to be turned on (ms)
	RELAY_OFF_VDC	186.0f	Bus voltage threshold for judging whether the relay is to be turned off (V)
	RELAY_OFF_DT_CT	60	Period for judging whether the relay is to be turned off (ms)
	RELAY_LPF_K	0.1f	LPF gain
r_relay_manager_api.h	RELAY_MANAGER_ERROR_NONE	0x0000	State of error detection for relay control: No error
	RELAY_MANAGER_ERROR_AC_OVER_VOLTAGE *	0x0001	State of error detection for relay control: AC voltage error
	RELAY_MANAGER_ERROR_BUS_OVER_VOLTAGE *	0x0002	State of error detection for relay control: Overvoltage error
	RELAY_MANAGER_ERROR_BUS_LOW_VOLTAGE *	0x0004	State of error detection for relay control: Low-voltage error
	RELAY_MANAGER_ERROR_OVER_CURRENT_SW *	0x0008	State of error detection for relay control: Overcurrent error detected by software
	RELAY_MANAGER_ERROR_OVER_CURRENT_HW *	0x0010	State of error detection for relay control: Overcurrent error detected by hardware
	RELAY_MANAGER_ERROR_OVER_HEATING	0x0020	State of error detection for relay control: Abnormal IGBT temperature
	RELAY_MANAGER_ERROR_UNKNOWN	0xffff	State of error detection for relay control: Error due to an unknown cause
	RELAY_MODE_IDLE	0x00	Run mode for relay control: Idle
	RELAY_MODE_RELAY_ON	0x01	Run mode for relay control: Relay is turned on.
	RELAY_MODE_RELAY_ERROR	0x02	Run mode for relay control: Error in the relay

Note: The macros marked with \* are not used in this software.

## 8.9 Driver Module

The driver module provides an interface between the manager modules, which is equivalent to the middleware of the sample program, and the FSP required to access the peripheral modules in the MCU. Appropriately configuring the driver module allows you to assign MCU functions and accommodate differences in specifications between boards without modifying the motor module or relay module.

This section describes the driver settings for the motor module and relay module.

### 8.9.1 Functions

The following tables list the functions of the driver module.

Table 8-24 List of Functions of the Driver Module for the Motor Module

Function	Description
Acquisition of A/D-converted values	Acquires A/D values such as the phase currents and inverter board bus voltage via an API function of the FSP.
PWM duty cycle settings	Sets the duty cycles of PWM output in the U-, V-, and W-phases via an API function of the FSP.
PWM start and stop	Controls whether to start or stop PWM output via an API function of the FSP.

Table 8-25 List of Functions of the Driver Module for the Relay Module

Function	Description
Acquisition of A/D-converted values	Acquires A/D values such as the inverter board bus voltage via an API function of the FSP.
Setting the output for turning the relay on	Sets the output port pin to turn the relay on via an API function of the FSP.
Setting the output for turning the relay off	Sets the output port pin to turn the relay off via an API function of the FSP.
Receiving the input for abnormal IGBT temperature detection	Obtains the value input from the abnormal IGBT temperature detection port pin via an API function of the FSP.

### 8.9.2 Module Configuration Diagram

Figure 8-7 shows the configuration of the driver module.

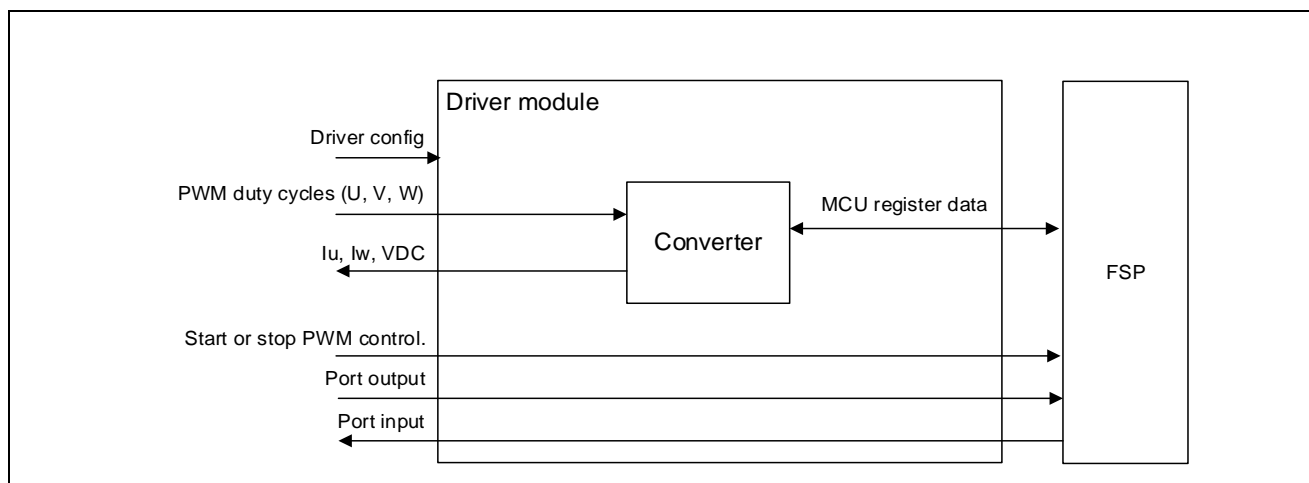


Figure 8-7 Configuration of the Driver Module

## 8.9.3 API

The following tables list and describe the API functions of the driver module.

Table 8-26 List of API Functions of the Driver Module for the Motor Module

API Function	Description
R_MOTOR_DRIVER_Open	Generates an instance of the driver module.
R_MOTOR_DRIVER_Close	Places the module in the closed 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 cycle.
R_MOTOR_DRIVER_BldcZeroDutySet	Forcibly fixes the GPT control mode to output 0.
R_MOTOR_DRIVER_BldcCompareDutySet	Changes the GPT control mode to PWM mode.
R_MOTOR_DRIVER_PWMControlStop	Stops PWM control.
R_MOTOR_DRIVER_PWMControlStart	Starts PWM control.

Table 8-27 List of API Functions of the Driver Module for the Relay Module

API Function	Description
R_RELAY_DRIVER_Open	Generates an instance of the driver module.
R_RELAY_DRIVER_Close	Places the module in the closed state.
R_RELAY_DRIVER_Reset	Places the module in the reset state.
R_RELAY_DRIVER_ParameterUpdate	Inputs the variable information that is to be used inside the module.
R_RELAY_DRIVER_AnalogGet	Acquires the A/D conversion results.
R_RELAY_DRIVER_RelayOn	Turns the relay on.
R_RELAY_DRIVER_RelayOff	Turns the relay off.
R_RELAY_DRIVER_OverheatCheck	Checks the abnormal IGBT temperature.

## 8.9.4 Configuration Items

The following tables list the configuration items of the driver module. Set up the functions to be used and the required parameters.

Table 8-28 List of Configuration Items for the Motor Module

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	DRIVER_CFG_FUNC_PWM_OUTPUT_START	R_Config_MOTOR_StartTimerCtrl (API relay function of the FSP) *1	Sets the function for enabling PWM outputs.
	DRIVER_CFG_FUNC_PWM_OUTPUT_STOP	R_Config_MOTOR_StopTimerCtrl (API relay function of the FSP) *1	Sets the function for disabling PWM outputs.
	DRIVER_CFG_FUNC_ADC_DATA_GET *2	R_Config_MOTOR_AdcGetConvVal (API relay function of the FSP) *1	Sets the function for acquiring the A/D conversion results
	DRIVER_CFG_FUNC_DUTY_SET *2	R_Config_MOTOR_UpdDuty (API relay function of the FSP) *1	Sets the function for setting the duty cycle
	DRIVER_CFG_FUNC_ZERO_DUTY_SET	R_Config_MOTOR_UpdZeroDuty (API relay function of the FSP) *1	Sets the function for fixing the outputs to 0
	DRIVER_CFG_FUNC_COMPARE_DUTY_SET	R_Config_MOTOR_UpdCompareDuty (API relay function of the FSP) *1	Sets restoration of the outputs to PWM output
r_motor_inverter_cfg.h	INVERTER_CFG_ADC_REF_VOLTAGE	3.3f	Sets the reference voltage for A/D conversion
r_motor_module_cfg.h	MOTOR_MCU_CFG_ADC_OFFSET	0x7FF	Sets the A/D offset value.

Notes: 1. For details about the functions shown in the "Setting" column, see the *Renesas Flexible Software Package (FSP) User's Manual*.

2. These macros are not used in this software.

Table 8-29 List of Configuration Items for the Relay Module

File Name	Macro Name	Setting	Description
r_relay_cfg.h	RELAY_DRIVER_CFG_FUNC_ADC_DATA_GET	R_Config_RELAY_AdcGetConvVal (API relay function of the FSP) *	Sets the function for acquiring the A/D conversion results
	RELAY_DRIVER_CFG_FUNC_RELAY_ON	R_Config_RELAY_RelayOn (API relay function of the FSP) *	Makes the settings of the output port pin for turning the relay on.
	RELAY_DRIVER_CFG_FUNC_RELAY_OFF	R_Config_RELAY_RelayOff (API relay function of the FSP) *	Makes the settings of the output port pin for turning the relay off.
	RELAY_DRIVER_CFG_FUNC_OVERHEAT_CHECK	R_Config_RELAY_OverheatCheck (API relay function of the FSP) *	Acquires the abnormal IGBT temperature state.
	RELAY_CFG_ADC_REF_VOLTAGE	3.3f	Sets the reference voltage for A/D conversion
	RELAY_CFG_BUS_VOLTAGE_GAIN	297.879f	Sets the voltage gain.
	RELAY_CFG_BUS_VOLTAGE_OFFSET	2047.0f	Sets the A/D-converted value corresponding to 0-V

File Name	Macro Name	Setting	Description
			detection.
	RELAY_MCU_CFG_AD12BIT_DATA	4095.0f	Sets the A/D-converted value corresponding to maximum voltage detection.
	RELAY_CFG_BUS_VOLTAGE_OFFSET	0x7FF	Sets the A/D offset value.
	RELAY_OVERHEAT_DETECTION_LEVEL	RELAY_LOW	Sets the level for detecting the abnormal IGBT temperature.

Note: \* For details about the functions shown in the "Setting" column, see the *Renesas Flexible Software Package (FSP) User's Manual*.

## 8.9.5 Structure and Variable Information

The following tables list the structures that are used for the driver module. In the driver module, the structures for the driver module (g\_st\_driver and g\_st\_relay\_driver) are defined by the API function for securing an instance of the module.

Table 8-30 List of Structures and Variables for the Motor Module

Structure	Variable	Description
st_motor_driver_t  Structure for the driver module	*ADCDataGet	Pointer to a relay function of the FSP This variable specifies the function that acquires the results of A/D conversion.
	*BLDCDutySet	Pointer to a relay function of the FSP This variable specifies the function that enables PWM output.
	*BLDCZeroDutySet *	Pointer to a relay function of the FSP This variable specifies the function that sets the outputs from the lower side and upper side of the inverter to 100% and 0% respectively.
	*BLDCCompareDutySet *	Pointer to a relay function of the FSP This variable specifies the function that sets the duty cycle to be dependent on the compare match.
	*PWMOutputStop	Pointer to a relay function of the FSP This variable specifies the function that disables PWM output.
	*PWMOutputStart	Pointer to a relay function of the FSP This variable specifies 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	Value to count for one interval of the PWM counter (information for the duty-cycle setting)
	f4_pwm_dead_time_cnt	Value to count for the dead time (information for the duty-cycle setting)
	s2_bus_volt_offset	Offset for adjusting bus voltage detection
st_motor_driver_cfg_t  Structure for setting the parameters for controlling the drive module	*ADCDataGet	Pointer to a relay function of the FSP
	*BLDCDutySet	Pointer to a relay function of the FSP
	*PWMOutputStop	Pointer to a relay function of the FSP
	*PWMOutputStart	Pointer to a relay function of the FSP
	f4_shunt_ohm	Shunt resistance value ( $\Omega$ ) (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	Value to count for one interval of the PWM counter (information for the duty-cycle setting)
	f4_pwm_dead_time_cnt	Value to count for the dead time (information for the duty-cycle setting)
	s2_bus_volt_offset	Offset for adjusting bus voltage detection

Note: The variables marked with \* are not used in this software.

Table 8-31 List of Structures and Variables for the Relay Module

Structure	Variable	Description
st_relay_driver_t	f4_ad_vdc_per_digit	Scale for A/D conversion of voltages
	s2_bus_volt_offset	Bus voltage offset
st_relay_driver_cfg_t	f4_bus_volt_gain	Bus voltage gain
	s2_bus_volt_offset	Bus voltage offset
st_relay_adc_t	u2_vdc_ad	Bus voltage (A/D-converted value)

### 8.9.6 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 FSP and to convert data. The parameters are input by using st\_motor\_driver\_cfg\_t (the structure for setting the parameters for controlling the driver module). In the sample program, the values defined as configuration items are used to set up the parameters. Table 8-32 lists the settings.

For the relay module, read R\_RELAY\_DRIVER\_ParameterUpdate above as R\_MOTOR\_DRIVER\_ParameterUpdate and st\_relay\_driver\_cfg\_t as st\_motor\_driver\_cfg\_t.

Table 8-32 Example of Settings in the Sample Program for the Motor Module

Variable Name	Macro Name	File Name
*ADCDataGet	DRIVER_CFG_FUNC_ADC_DATA_GET	r_motor_module_cfg.h
*BLDCDutySet	DRIVER_CFG_FUNC_DUTY_SET	
*BLDCZeroDutySet *	DRIVER_CFG_FUNC_ZERO_DUTY_SET	
*BLDCCompareDutySet *	DRIVER_CFG_FUNC_COMPARE_DUTY_SET	
*PWMOutputStop	DRIVER_CFG_FUNC_PWM_OUTPUT_STOP	
*PWMOutputStart	DRIVER_CFG_FUNC_PWM_OUTPUT_START	
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	
s2_bus_volt_offset	INVERTER_CFG_BUS_VOLTAGE_OFFSET	
f4_pwm_period_cnt	MOTOR_COMMON_CARRIER_SET_BASE	r_motor_module_cfg.h
f4_pwm_dead_time_cnt	MOTOR_COMMON_DEADTIME_SET	

Note: The variables marked with \* are not used in this software.

Table 8-33 Example of Settings in the Sample Program for the Relay Module

Variable Name	Macro Name	File Name
*ADCDataGet	RELAY_DRIVER_CFG_FUNC_ADC_DATA_GET	r_relay_cfg.h
*RelayOn	RELAY_DRIVER_CFG_FUNC_RELAY_ON	
*RelayOff	RELAY_DRIVER_CFG_FUNC_RELAY_OFF	
*OverheatCheck	RELAY_DRIVER_CFG_FUNC_OVERHEAT_CHECK	
f4_bus_volt_gain	RELAY_CFG_BUS_VOLTAGE_GAIN	
s2_bus_volt_offset	RELAY_CFG_BUS_VOLTAGE_OFFSET	

## 9. Parameter Settings

### 9.1 Overview

In this sample program, parameters are defined as macros in the header files shown below. The parameter values defined as the macros are set in 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 the RMW or other tools. 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

Header File Name	Description
r_motor_module_cfg.h	Defines the 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.
r_relay_cfg.h	Defines the initial values of parameters related to the relay for the inrush-current prevention resistor.

### 9.2 MCU-Related Parameters

Table 9-2 lists parameters that are related to peripheral functions of the MCU. If the peripheral settings of the MCU are changed through the FSP, the parameters related to the changed settings must also be modified.

Table 9-2 List of MCU-Related Parameters

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	MOTOR_MCU_CFG_PWM_TIMER_FREQ	120.0f	PWM timer frequency (MHz)
	MOTOR_MCU_CFG_CARRIER_FREQ	8.0f	Carrier wave frequency (kHz)
	MOTOR_MCU_CFG_INTR_DECOMPOSITION	0	Value to count for the skipping of carrier wave interrupts
	MOTOR_MCU_CFG_AD_FREQ	60.0f	ADC operating frequency (MHz)
	MOTOR_MCU_CFG_AD_SAMPLING_CYCLE	2.0*(7.25+63.0)	ADC sampling interval (cycles)
	MOTOR_MCU_CFG_AD12BIT_DATA	4095.0f	ADC resolution
	MOTOR_MCU_CFG_ADC_OFFSET	0x7FF	ADC offset value

### 9.3 List of Parameters for Setting Control Functions

Table 9-3, Table 9-4, and Table 9-5 list the parameters used to enable or disable the functions provided by the motor control program. The items related to motor constants or settings used internally for motor control are described later.

Table 9-3 List of Operational Parameters (General)

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	MOTOR_TYPE_BLDC	-	Use the default value.
	MOTOR_COMMON_CFG_LOOP_MODE	MOTOR_LOOP_SPEED	Use the default value.
	MOTOR_COMMON_CFG_OVERCURRENT_MARGIN_MULT	2.0f	Limit coefficient for overcurrent
	MOTOR_COMMON_CFG_IAMAX_CALC_MULT	MTR_SQRT_3	Coefficient for calculating the overcurrent limit value. Set to $\sqrt{3}$ .
	MOTOR_MCU_CFG_TFU_OPTIMIZE	MTR_ENABLE	Setting of the TFU (trigonometric function unit)-specific function processing. It is automatically set to ENABLE.

Table 9-4 Operational Parameter (Related to Speed Control)

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	SPEED_CFG_CTRL_PERIOD	0.0005f	Setting of the speed control interval (s). Set this to 0.0005f to specify 0.5 ms.

Table 9-5 List of Operational Parameters (Related to Current Control)

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_VOLT_ERROR_COMP	MTR_ENABLE	Enables or disables the voltage error compensation function. Set this to MTR_ENABLE.
	CURRENT_CFG_MODULATION_METHOD	MOD_METHOD_SVPWM	See section 9.6. Set this to MOD_METHOD_SVPWM in most cases.
	CURRENT_CFG_OFFSET_CALC_TIME	512.0f	Sets the current offset measurement time.

## 9.4 Protection-Related Parameters

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

Table 9-6 Settings of the Motor Parameters and Inverter Parameters

File Name	Macro Name	Setting	Description
r_motor_inverter_cfg.h	INVERTER_CFG_CURRENT_LIMIT	25.5f	Overcurrent limit value for the inverter board (A)
	INVERTER_CFG_OVERVOLTAGE_LIMIT	430.0f	Overvoltage limit (V)
	INVERTER_CFG_UNDERVOLTAGE_LIMIT	186.0f	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*

Set 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*

Set the voltage at which the low-voltage 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 for Motor Control

The PWM carrier frequency for motor control is set by the FSP and by the MOTOR\_MCU\_CFG\_CARRIER\_FREQ constant defined in r\_motor\_module\_cfg.h. If the PWM carrier frequency is changed, the items listed in Table 9-7 also require changing. Some parameters require adjustment to match the settings of the PWM carrier frequency.

The default PWM carrier frequency for motor control in this sample program is 8.0 kHz.

Table 9-7 Parameters to be Modified When the PWM Carrier Frequency is Changed

Item	Item that Requires Change
Dead time value	See section 9.7, Inverter Parameters.
Carrier frequency	<ul style="list-style-type: none"> <li>Setting for the three-phase PWM GPT described in section 10.5</li> <li>MOTOR_MCU_CFG_CARRIER_FREQ described in section 9.2</li> </ul>
Motor control-related parameters	<ul style="list-style-type: none"> <li>Current control parameters</li> <li>V/f control parameters</li> </ul>

## 9.6 Setting the Pulse-Width Modulation Method

In this sample program, one of two pulse-width modulation drive modes can be selected. The default setting is the space vector PWM (MOD\_METHOD\_SVPWM) mode. Table 9-8 shows the configuration item 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 Configuration Item

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_MODULATION_METHOD	(MOD_METHOD_SVPWM)	Pulse-width modulation drive mode

Table 9-9 Items to be Set for the Pulse-Width Modulation Drive Mode

Item	Value	Pulse-Width Modulation Drive Mode
MOD_METHOD_SPWM	0	Sinusoidal PWM
MOD_METHOD_SVPWM	1	Space vector PWM

The pulse-width modulation has the following macros to be set. Use the default values in most cases.

Table 9-10 List of Macros

File Name	Macro Name	Setting	Description
r_motor_current_modulation.h	MOD_DEFAULT_MAX_DUTY	1.0f	Maximum PWM duty cycle. Leave the value at 1.0f in most cases.
	MOD_VDC_TO_VAMAX_MULT	0.6124f	Coefficient for the conversion to obtain the maximum possible output voltage at the inverter bus voltage.
	MOD_SVPWM_MULT	1.155f	This setting is only necessary when space vector PWM is to be used. Coefficient for space 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 9-11 lists the inverter parameters set in the sample program.

Table 9-11 Settings of the Inverter Parameters

File Name	Macro Name	Setting	Description
r_motor_inverter_cfg.h	INVERTER_CFG_SHUNT_RESIST	1.0f	Shunt resistance value ( $\Omega$ ) Set to 1.0 when a Hall current transformer is to be used.
	INVERTER_CFG_DEADTIME	3.0f	Dead time ( $\mu$ s)
	INVERTER_CFG_VOLTAGE_GAIN	297.879f	Coefficient for voltage detection
	INVERTER_CFG_CURRENT_AMP_GAIN	-0.05333f	Gain of the amplifier for current detection
	INVERTER_CFG_INPUT_V	282.8f	Input voltage (V)
	INVERTER_CFG_ADC_REF_VOLTAGE	3.3f	Analog power-supply voltage for the MCU (V)
	INVERTER_CFG_BUS_VOLTAGE_OFFSET	2047.0f	Offset for adjusting bus voltage detection
	INVERTER_CFG_COMP_V0	1.35744f	Coefficient for compensation of the voltage error (V)
	INVERTER_CFG_COMP_V1	2.71488f	Coefficient for compensation of the voltage error (V)
	INVERTER_CFG_COMP_V2	4.07232f	Coefficient for compensation of the voltage error (V)
	INVERTER_CFG_COMP_V3	5.42976f	Coefficient for compensation of the voltage error (V)
	INVERTER_CFG_COMP_V4	6.7872f	Coefficient for compensation of the voltage error (V)
	INVERTER_CFG_COMP_I0	0.21f	Coefficient for compensation of the voltage error (A)
	INVERTER_CFG_COMP_I1	0.42f	Coefficient for compensation of the voltage error (A)
	INVERTER_CFG_COMP_I2	0.66f	Coefficient for compensation of the voltage error (A)
	INVERTER_CFG_COMP_I3	0.90f	Coefficient for compensation of the voltage error (A)
	INVERTER_CFG_COMP_I4	1.50f	Coefficient for compensation of the voltage error (A)

#### *INVERTER\_CFG\_DEADTIME*

Specify the dead time in  $\mu$ s (microseconds) that is described in the inverter specifications and design document. For the MCI-HV-2-3PH inverter, 3.0  $\mu$ s is specified.

#### *INVERTER\_CFG\_INPUT\_V*

The default voltage value is 282.8 VDC, which is obtained by rectifying three-phase 200 VAC through the full-wave rectifying circuit in the inverter.

*INVERTER\_CFG\_ADC\_REF\_VOLTAGE*

Specify the analog voltage of the MCU. For the RA6T2 CPU board, 3.3 V is specified.

*INVERTER\_CFG\_COMP\_Vx, INVERTER\_CFG\_COMP\_Ix*

See section 9.7.4.

## 9.7.2 Current Detection Gain

In the MCI-HV-2-3PH inverter, the voltage 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.

*INVERTER\_CFG\_ADC\_REF\_VOLTAGE*

Set to 3.3 because the ADC reference voltage is 3.3 V in the RA6T2.

*INVERTER\_CFG\_SHUNT\_RESIST*

Set the resistance used for the shunt resistor. A Hall current transformer may be used instead of a shunt resistor in inverters with a medium or higher capacity, that is, 1 kW or larger, such as the MCI-HV-2-3PH. In such a case, specify 1.0.

*INVERTER\_CFG\_CURRENT\_AMP\_GAIN*

Set the coefficient for use in calculating the current (A) per volt input to the ADC. The MCI-HV-2-3PH specifications prescribe that the output current range is  $\pm 30.94$  A (61.88 A peak-to-peak) for the voltage range from 0 V to 3.3 V, that is,  $61.88 \text{ A} / 3.3 \text{ V} = 18.75 \text{ A per volt}$ . Assuming that the shunt resistance value is 1.0 because a Hall current transformer is used in the MCI-HV-2-3PH, the coefficient becomes  $(1/1.0) \times (1/18.75) = 0.05333$ .

In that case, therefore, set *INVERTER\_CFG\_CURRENT\_AMP\_GAIN* to -0.05333.

$$INVERTER\_CFG\_CURRENT\_AMP\_GAIN = \frac{1}{INVERTER\_CFG\_SHUNT\_RESIST [\Omega]} \times \frac{1}{Current \text{ per volt } [A/V]}$$

Table 9-12 Current Signal Specifications for the MCI-HV-2-3PH

Three-Phase Output Current	ADC Input Voltage	A/D-Converted Value
-30.94 A	3.3 V	4095
0 A	1.65 V	2047
+30.94 A	0.0 V	0

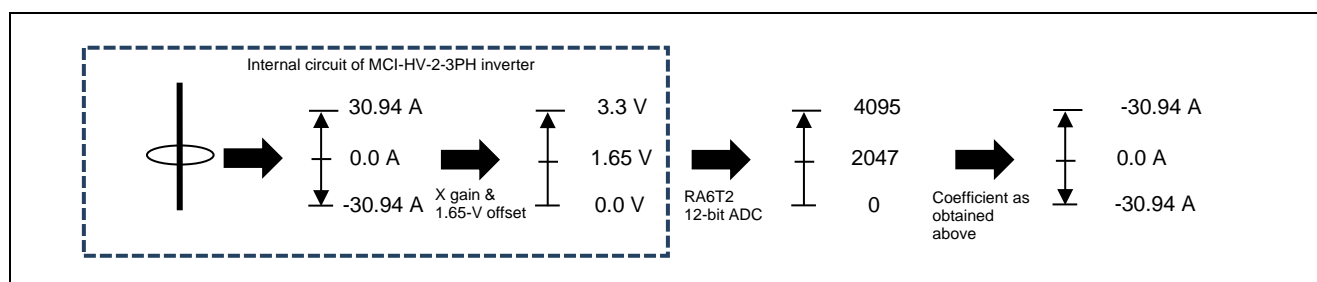


Figure 9-1 Flow of Calculation for Current Detection

### 9.7.3 Voltage Detection Gain

The voltage detection gain is set by `INVERTER_CFG_VOLTAGE_GAIN`.

Set the coefficient for use in calculating the inverter bus voltage (V) per volt input to the ADC. In the MCI-HV-2-3PH, the input to the ADC is 1.65 V to 3.3 V and the bus voltage between 0 V to 491.5 V is to be detected. In this case,  $491.5 / (3.3 - 1.65) = 297.879$ , so set `INVERTER_CFG_VOLTAGE_GAIN` to 297.879.

$$\text{INVERTER\_CFG\_VOLTAGE\_GAIN} = \frac{\text{Reference inverter bus voltage}}{\text{Reference ADC input voltage}} = \frac{491.5}{3.3 - 1.65} = 297.879$$

Table 9-13 Specifications of Inverter Bus Voltage Signal for the MCI-HV-2-3PH

Inverter Bus Voltage	ADC Input Voltage	A/D-Converted Value
0.0 V	1.65 V	0
491.5 V	3.3 V	4095

### 9.7.4 Voltage Error Compensation Parameters

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

#### (1) Selecting a dead time value

The characteristics of the power semiconductor devices used in the inverter determine the dead time. When Si-IGBT is used, a value roughly in or around the range from 2 to 3  $\mu\text{s}$  is selected. Reflect the selected dead time value in the dedicated input location provided for motor settings in the FSP.

#### (2) Setting the flag for enabling 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 (μs) ÷ 1000) × bus voltage value

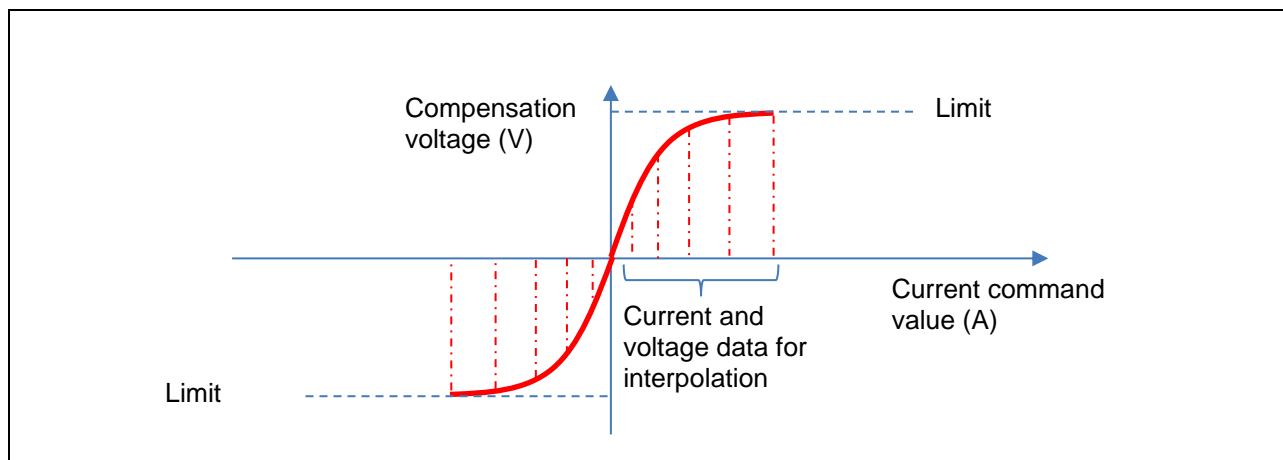


Figure 9-2 Relationship between the Compensation Voltage, Limit, and Current Command Values

Table 9-14 Current and Voltage Data for Interpolation for Different Carrier Cycles

Carrier Frequency		8 kHz
	$I_u$	$\Delta V_u$
0	0.00f	0.00f
1	0.21f	1.35744f
2	0.42f	2.71488f
3	0.66f	4.07232f
4	0.90f	5.42976f
5	1.50f	6.7872f

## 9.8 Motor Parameters

The motor resistance, inductance, and certain other motor parameters are not used in V/f control. When using this sample program, only set the parameters listed below with reference to the specifications or values on the nameplate of the motor to be used.

Table 9-15 Motor Parameter Settings

File Name	Macro Name	Setting	Description
r_motor_targetmotor_cfg.h	MOTOR_CFG_POLE_PAIRS	2	Number of pole pairs
	MOTOR_CFG_NOMINAL_CURRENT_RMS	15.5f	Rated current (A)
	MOTOR_CFG_MAX_SPEED_RPM	1800.0f	Maximum speed (rpm)

### *MOTOR\_CFG\_POLE\_PAIRS*

Set the number of pole pairs of the motor. The number of pole pairs is 1/2 the number of poles.

### *MOTOR\_CFG\_NOMINAL\_CURRENT\_RMS*

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

### *MOTOR\_CFG\_MAX\_SPEED\_RPM*

Specify the maximum speed of the motor based on the highest of the rated frequency values indicated on the nameplate or in the specifications of the motor.

## 9.9 Current Control Parameters

Table 9-16 shows the current control parameter. The current control software is not usually used in V/f control but the processing for detecting currents and related processing are used. Therefore, only some of the current control parameters are used. The current control parameters are calculated based on the motor parameters, PWM carrier frequency, and desired current response performance.

The macro for the current control parameter shown in Table 9-16 is set and reflected in the internal variable at startup but if adjustments are needed after startup, the parameters listed in Table 6-5 can be changed from the RMW. Note that not all of the current control parameters can be changed.

Table 9-16 Current Control Parameter

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_OFFSET_CALC_TIME	512.0f	Current offset measurement time

### *CURRENT\_CFG\_OFFSET\_CALC\_TIME*

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

## 9.10 Speed Control Parameters

Table 9-17 lists 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 startup but if adjustments are needed after startup, the parameters listed in Table 6-5 can be changed from the RMW. Note that not all of the speed control parameters can be changed.

Table 9-17 List of Speed Control Parameters

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	SPEED_CFG_CTRL_PERIOD	0.0005f	Control interval setting (s)
	SPEED_CFG_SPEED_LIMIT_RPM	1800.0f	Speed limit value (rpm) (mechanical angle)
	SPEED_CFG_RATE_LIMIT_RPM	500.0f	Acceleration limit (rpm/s)

### *SPEED\_CFG\_CTRL\_PERIOD*

Set to 0.0005 s (0.5 ms). If this setting needs to be changed, the AGT0 timer setting of 10.6 must also be changed to specify a new speed control interval.

### *SPEED\_CFG\_RATE\_LIMIT\_RPM*

Set the rate at which the speed increases (acceleration rate) when a speed command value is set. The higher the value, the faster the speed will increase. When 100 is specified, the speed is increased by 100 rpm per second. In this case, the speed will reach 2000 rpm from the standstill state in 20 seconds.

## 9.11 Voltage Phase Lead Compensation Parameter

This is the compensation value for advancing the angle to the actual timing of PWM output based on the timing of current detection. In this sample program and inverter configuration, the PWM output timing can be adjusted through advancement by 0.5 of a sample.

Table 9-18 Configuration Information

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_PERIOD_MAG_VALUE	0.5f	This sets the number of samples used for lead compensation. Set to 0.5.

## 9.12 V/f Control Parameters

Specify the parameters to be used in V/f control. If the motor in use is changed, the following parameters will also require the appropriate changes.

Table 9-19 Setting Parameters for V/f Control

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_IMVF_FREQUENCY_OUTPUT_MAX	60.0f	Electrical angle frequency equivalent to the maximum possible output speed
	CURRENT_CFG_IMVF_RATED_FREQUENCY	50.0f	Rated frequency (base frequency) to be used for V/f control
	CURRENT_CFG_IMVF_VOLTAGE_OUTPUT_MAX	200.0f	Maximum possible output voltage
	CURRENT_CFG_IMVF_RATED_VOLTAGE	200.0f	Rated voltage to be used for V/f control
	CURRENT_CFG_IMVF_TORQUE_BOOST	0.024f	Torque boost compensation value

### *CURRENT\_CFG\_IMVF\_FREQUENCY\_OUTPUT\_MAX*

Specify the maximum frequency that can be output. This frequency is convertible to the speed of rotation by using the number of poles and the speed thus obtained is equivalent to the maximum speed that the inverter can output.

### *CURRENT\_CFG\_IMVF\_RATED\_FREQUENCY*

Specify the rated frequency (Hz), which is indicated on the nameplate of the target induction motor to be controlled.

### *CURRENT\_CFG\_IMVF\_VOLTAGE\_OUTPUT\_MAX*

Specify the maximum voltage (Vrms) that the inverter can output.

### *CURRENT\_CFG\_IMVF\_RATED\_VOLTAGE*

Specify the rated voltage (Vrms) to be output at the rated speed, which is indicated on the nameplate of the target induction motor to be controlled.

### *CURRENT\_CFG\_IMVF\_TORQUE\_BOOST*

During low-speed operation, the voltage will drop due to the winding resistance in the induction motor and the motor operation will become unstable. To compensate for the voltage drop and suppress the instability of operation, specify a torque boost compensation value (%) which determines the minimum output voltage. The setting 1.00f specifies 100% as the compensation value. The value specified is relative to the rated voltage, with the rated voltage taken as 100%. The setting is generally a value in the range from 0% to 5%.

10. Settings for the FSP

10.1 Overview of the FSP

Figure 10-1 shows the software architecture of this sample program. The flexible software package (FSP) makes the settings of the registers the MCU has that are specific to peripheral modules such as the ADC, GPT, and SCI and controls interrupts from those modules. The FSP provides the middleware functionality such as the file system and the protocol stack for use in communications as well as that of the hardware abstraction layer (HAL).

This sample program only uses the functions of the abstracted register settings and interrupts for the peripheral functions of the MCU, such as the timers, ADC, and GPIO, from among the typical functions of the FSP. The functions for motor control and relay control are implemented in the application layer. This allows users to freely change software such as that for the algorithm and sequence of motor control.

Although the FSP also provides the motor middleware as a standard function, which is analogous to that of the sample program, note that its functionality is not compatible with that of the sample program in terms of the interfaces such as the software’s internal configuration, functions, and parameters.

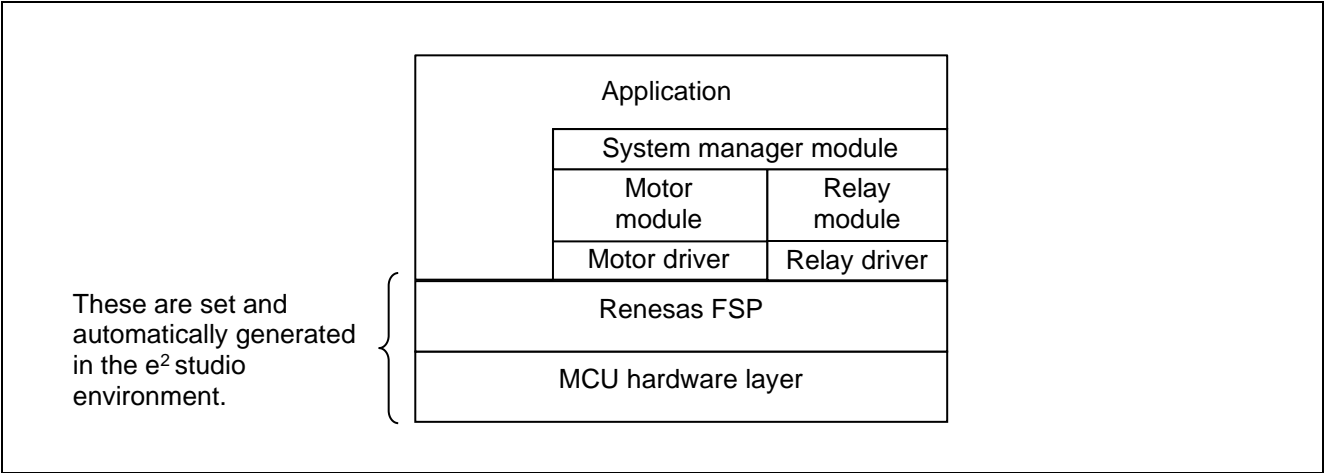


Figure 10-1 Software Architecture of This Sample Program

10.2 Setting FSP Stacks

The FSP provides functional modules for each peripheral function, which are referred to as stacks. Table 10-1 lists the FSP stacks for use with this sample program and the functions allocated to each of them.

Opening the [Stacks Configuration] page for the FSP or changing the property in a stack automatically generates the hal\_data.c/h and other files in the ra\_gen folder. Execution of [Generate Project Content] automatically generates or updates the FSP-related modules in the ra folder.

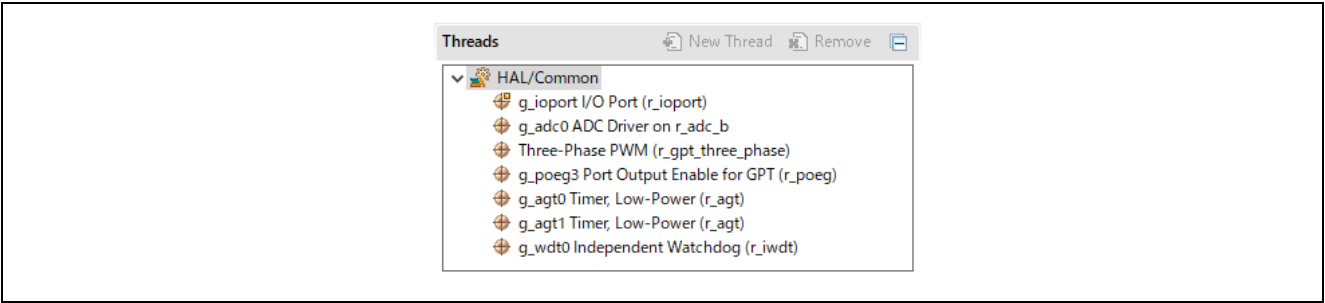


Figure 10-2 List of FSP Stacks

Table 10-1 FSP Stacks and the Functions Allocated to Each of Them

Function	FSP Stack
Three-phase PWM output	Three-Phase PWM (r_gpt_three_phase)
A/D conversion for the motor (detection of U-, V-, and W-phase output currents)	g_adc0 ADC Driver on r_adc_b (adc0, sub group0)
Setting port pins to be used	g_ioport I/O Port (r_ioport)
Speed control interrupt timer (0.5-ms intervals)	g_agt0 Timer, Low-Power (r_agt)
Relay control interrupt timer (1-ms intervals)	g_agt1 Timer, Low-Power (r_agt)
Overcurrent detection	g_poeg3 Port Output Enable for GPT (r_poeg)
Independent Watchdog Timer	g_wdt0_Independent_Watchdog(r_iwtdt)

### 10.3 Callback Interrupts

The FSP defines callback functions as functions to be called for the interrupt processing. Table 10-2 lists the interrupts.

The files which handle the actual processing for the callback functions in the table are in src/application/mcu/ra6t2/r\_app\_mcu\_callback.c.

Table 10-2 List of Interrupts

FSP Stack	Callback Function	Description
g_adc0	callback_motor_current_cyclic_adc ()	The periodic callback function called at the motor current control interval (8 kHz) internally checks whether interrupt processing is to be executed.
poeg	callback_poe_overcurrent()	Be sure to call R_POEG_Reset() from within the callback function for the POEG stack to reset the flag. If this is not done, the other processing may be stopped depending on the interrupt priority level.
agt0	callback_motor_speed_cyclic_timer()	Speed control interval (2kHz)
agt1	callback_system_manager_cyclic_timer()	System manager interval (1kHz)

## 10.4 Pin Settings

Table 10-3 lists the information on pin interfaces.

Table 10-3 Pin Interfaces

Function	Pin Name	Peripheral Function	Pin to Which the Function is Allocated	Remarks
Measurement of the U-phase current	PA04	S12AD	AN004	
Measurement of the V-phase current	PA02	S12AD	AN002	
Measurement of the W-phase current	PA00	S12AD	AN000	
Measurement of the bus voltage	PA06	S12AD	AN006	
Abnormal IGBT temperature	PD07	GPIO	—	The low level indicates an abnormal state.
PWM emergency stop input in response to an overcurrent being detected	PC13	POEG	GTETRGD	The low level indicates an abnormal state.
PWM output ( $U_p$ )	PB04	GPT	GTIOC4A	Active high
PWM output ( $U_n$ )	PB05	GPT	GTIOC4B	Active high
PWM output ( $V_p$ )	PB06	GPT	GTIOC5A	Active high
PWM output ( $V_n$ )	PB07	GPT	GTIOC5B	Active high
PWM output ( $W_p$ )	PB08	GPT	GTIOC6A	Active high
PWM output ( $W_n$ )	PB09	GPT	GTIOC6B	Active high
Relay control to prevent inrush currents	PE01	GPIO	—	

## 10.5 Settings for the Three-Phase PWM GPT

The three-phase PWM GPT is used in motor control. In the FSP, triangle-wave PWM mode 1 (32-bit transfer at troughs) is specifiable. Note that the complementary mode is not selectable because the FSP 5.6.0 does not support it.

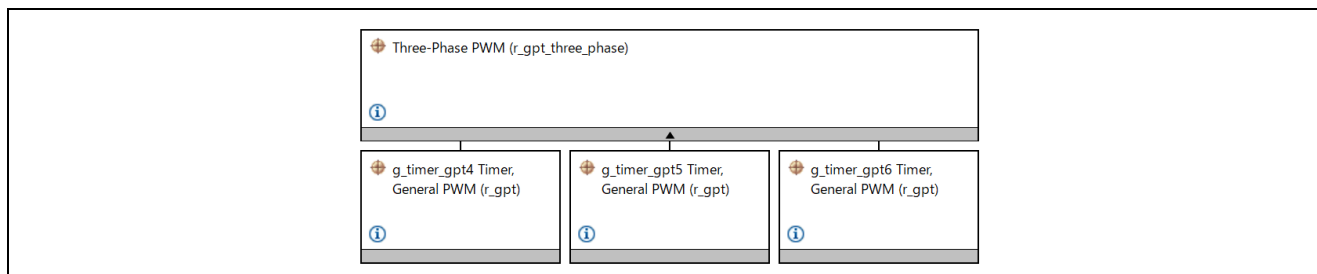


Figure 10-3 Stacks Related to the Three-Phase PWM GPT Stack

Table 10-4 Three-Phase PWM Settings

Function and Item for Setting			Setting
General	Name		g_three_phase0
	Mode		Triangle-Wave Symmetric PWM
	Period		125
	Period Unit		Microseconds
	GPT U-Channel		4
	GPT V-Channel		5
	GPT W-Channel		6
	Callback Channel		U-Channel
	Buffer Mode		Single Buffer
	GTIOCA Stop Level		Pin Level Low
	GTIOCB Stop Level		Pin Level High
Extra Features	Dead Time	Dead Time Count Up (Raw Counts)	360
		Dead Time Count Down (Raw Counts)	360

Table 10-5 U-Phase GPT Settings

Function and Item for Setting			Setting
Module g_timer_gpt4 timer	General	Name	g_timer_gpt4
	The settings of the other items are omitted because they are automatically made as part of the three-phase PWM settings.		
Pins		GTIOC4A	PB04
		GTIOC4B	PB05

Table 10-6 V-Phase GPT Settings

Function and Item for Setting			Setting
Module g_timer_gpt5 timer	General	Name	g_timer_gpt5
	The settings of the other items are omitted because they are automatically made as part of the three-phase PWM settings.		
Pins		GTIOC5A	PB06
		GTIOC5B	PB07

Table 10-7 W-Phase GPT Settings

Function and Item for Setting			Setting
Module g_timer_gpt6 timer	General	Name	g_timer_gpt6
	The settings of the other items are omitted because they are automatically made as part of the three-phase PWM settings.		
Pins		GTIOC6A	PB08
		GTIOC6B	PB09

## 10.6 AGT0 Settings (Setting the Interval for Speed Control)

The asynchronous general purpose timer (AGT) is used to set the interval between interrupts for use in 0.5-ms-periodic speed control. The table below shows an example of the AGT settings.

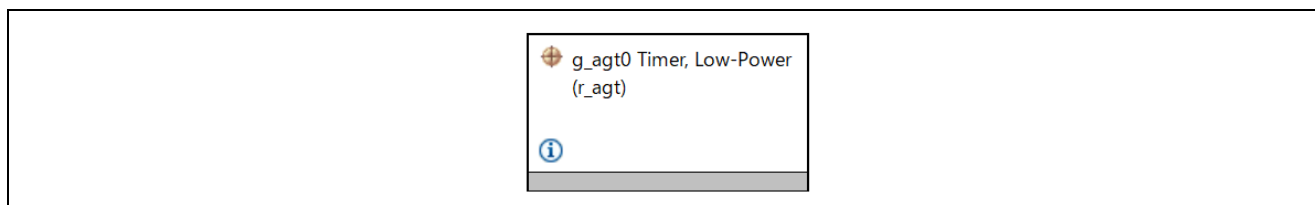


Figure 10-4 AGT0 Stack

Table 10-8 AGT0 Settings for the Speed Control Interval

Function and Item for Setting			Setting
General	Name		g_agt0
	Counter Bit Width		AGT 32-bit
	Channel		0
	Mode		Periodic
	Period		500
	Period Unit		Microseconds
	Count Source		PCLKB
Output	Duty Cycle Percent		50
	AGTOA Output		Disabled
	AGTOB Output		Disabled
	AGTO Output		Disabled
Input	Measurement Mode		Measure Disabled
	Input Filter		No Filter
	Enable Pin		Enable Pin Not Used
	Trigger Edge		Trigger Edge Rising
Interrupts	Callback		callback_motor_speed_cyclic_timer
	Underflow Interrupt Priority		Priority 9
Pins	AGTEED		<unavailable>
	AGTIO0		<unavailable>
	AGTO0		<unavailable>
	AGTOA0		<unavailable>
	AGTOB0		<unavailable>

## 10.7 AGT1 Settings (Setting the Interval for Relay Control)

The AGT1 is used to set the 1-ms interval for relay control.

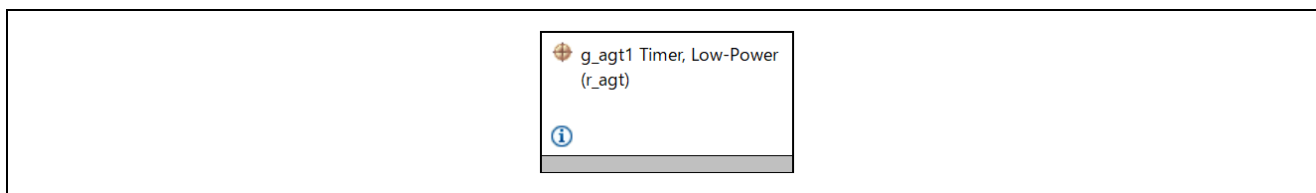


Figure 10-5 AGT1 Stack

Table 10-9 AGT1 Settings for the Interval for Relay Control

Function and Item for Setting			Setting
General	Name		g_agt1
	Counter Bit Width		AGT 32-bit
	Channel		1
	Mode		Periodic
	Period		1
	Period Unit		Milliseconds
	Count Source		PCLKB
Output	Duty Cycle Percent		50
	AGTOA Output		Disabled
	AGTOB Output		Disabled
	AGTO Output		Disabled
Input	Measurement Mode		Measure Disabled
	Input Filter		No Filter
	Enable Pin		Enable Pin Not Used
	Trigger Edge		Trigger Edge Rising
Interrupts	Callback		callback_system_manager_cyclic_timer
	Underflow Interrupt Priority		Priority 10
Pins	AGTEED		<unavailable>
	AGTIO0		<unavailable>
	AGTO0		<unavailable>
	AGTOA0		<unavailable>
	AGTOB0		<unavailable>

## 10.8 ADC Settings

The 12-bit A/D converters in the MCU are used to measure the U-, V-, and W-phase output currents and inverter bus voltage. Table 10-10 shows the channels to which the respective functions are allocated and the timing of detection.

With the initial settings of the ADC immediately after the MCU is started up, the processing to wait for the completion of calibration always proceeds. Attempting to start scanning by the ADC without waiting for the completion of calibration leads to the results detected by the ADC being incorrect and the possibility of operation also being incorrect. The calibration is completed after approximately several milliseconds have elapsed.

The times at which the motor current is detected are set to the troughs of the GPT carrier counter. Specifically, the trigger for detection of the motor current is generated on a compare match when counting down reaches 0.

Table 10-10 Settings for ADC Channels to Which the Respective Functions are Allocated and Timing of Detection

Function	Channel to be Allocated	Trigger for Starting A/D Conversion
Measurement of the inverter bus voltage	ADC0 channel 6	Counting down reaching 0
Measurement of the U-phase current	ADC0 channel 4	
Measurement of the V-phase current	ADC0 channel 2	
Measurement of the W-phase current	ADC0 channel 0	

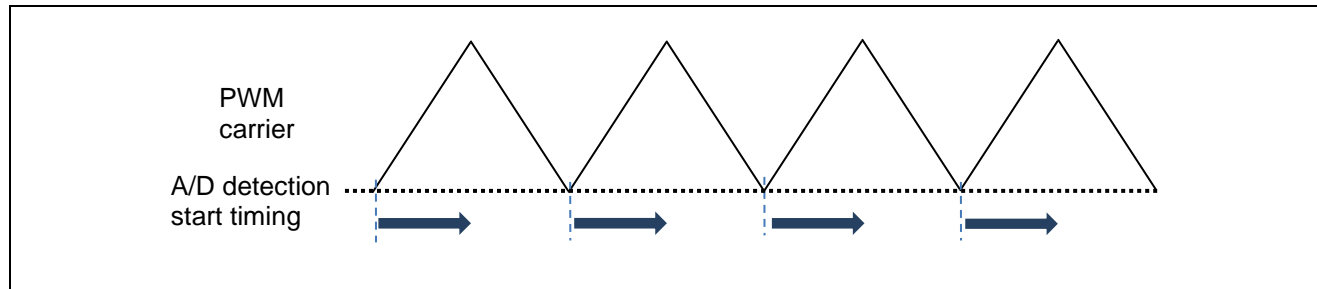


Figure 10-6 A/D Detection Start Timing (Trough-Sensed Interrupt)

Table 10-11 ADC Settings

Function and Item for Setting			Setting
General	Operation/ADC0	Conversion Method	SAR Mode
		Scan Mode	Single Scan
	Operation/ADC1	Conversion Method	SAR Mode
		Scan Mode	Single Scan
	ADC Successive Approximation Time	ADC0	6
		ADC1	6
	Synchronous Operation	Enable for ADC 0	Disable
		Enable for ADC 1	Disable
		Synchronous Operation Period Cycle	100
	Calibration/A/D Calibration	Sampling Time	10
		Conversion Time	6

	Calibration/Sample and Hold Calibration	Sampling Time	25
		Hold Time	3
	Sampling State Table	Entry 0	10
		Entry 1	4
		Entry 2	24
		Entries 3 to 15	95
	Name		g_adc0
Clock Configuration	Divider		Div /1
	Source		PCLKC
Interrupts	Limiter Clip Priority		All interrupts disabled
	Conversion Error Priority		All interrupts disabled
	Overflow Priority		All interrupts disabled
	Calibration End Priority		Priority 12
	Scan End Priority	Group 0	Priority 5
		Group 1	Priority 3
		Groups 2 to 8	Disabled
	FIFO Priorities		All interrupts disabled
	Callback		callback_motor_current_cyclic_adc
Digital Filter			Not in use (by default)
Sample and Hold	Enable Unit	Unit 0	<input checked="" type="checkbox"/>
		Unit 1	<input checked="" type="checkbox"/>
		Unit 2	<input checked="" type="checkbox"/>
		Unit 4	<input type="checkbox"/>
		Unit 5	<input type="checkbox"/>
		Unit 6	<input checked="" type="checkbox"/>
	Analog Channels 0 to 5	Sampling Time	60
		Hold Time	3
	Analog Channels 6 to 11	Sampling Time	95
		Hold Time	5
Programmable Gain Amplifier			Not in use (by default)
User Offset Table			Not in use (by default)
User Gain Table			Not in use (by default)
Limiter Clipping			Not in use (by default)
Virtual Channels	Virtual Channel 0	Scan Group	<b>Scan Group 0</b>
		Channel Select	<b>AN000</b>
		Sampling State Table ID	Sampling State Entry 0
		Channel Gain Table	Disabled
		Channel Offset Table	Disabled
		Add/Average Mode	Disabled
		Add/Average Count	1-time conversion (Normal Conversion)
		Limit Clip Table ID	Disabled
		Conversion Data Format Select	12-bit Data Format
		Digital Filter Selection	Disabled
	Virtual Channel 1	Scan Group	<b>Scan Group 0</b>
		Channel Select	<b>AN002</b>
		Sampling State Table ID	Sampling State Entry 0

		Channel Gain Table		Disabled
		Channel Offset Table		Disabled
		Add/Average Mode		Disabled
		Add/Average Count		1-time conversion (Normal Conversion)
		Limit Clip Table ID		Disabled
		Conversion Data Format Select		12-bit Data Format
		Digital Filter Selection		Disabled
	Virtual Channel 2	Scan Group		<b>Scan Group 0</b>
		Channel Select		<b>AN004</b>
		Sampling State Table ID		Sampling State Entry 0
		Channel Gain Table		Disabled
		Channel Offset Table		Disabled
		Add/Average Mode		Disabled
		Add/Average Count		1-time conversion (Normal Conversion)
		Limit Clip Table ID		Disabled
		Conversion Data Format Select		12-bit Data Format
		Digital Filter Selection		Disabled
	Virtual Channels 3 and 4			Not in use
	Virtual Channel 5	Scan Group		<b>Scan Group 1</b>
		Channel Select		<b>AN006</b>
		Sampling State Table ID		Sampling State Entry 0
		Channel Gain Table		Disabled
		Channel Offset Table		Disabled
		Add/Average Mode		Disabled
		Add/Average Count		1-time conversion (Normal Conversion)
		Limit Clip Table ID		Disabled
		Conversion Data Format Select		12-bit Data Format
		Digital Filter Selection		Disabled
	Virtual Channels 6 to 36			Not in use
Scan Groups	Scan Group 0	Self Diagnosis	Voltage Selection	Self-Diagnosis Mode Disabled
		External Trigger Enable	External Trigger Input 0 (ADTRG0) Enable	<input type="checkbox"/>
		External Trigger Enable	External Trigger Input 1 (ADTRG1) Enable	<input type="checkbox"/>
		ELC Trigger Enable		Not in use
		GPT Trigger Enable	GPT Channel 0 Request A	<input type="checkbox"/>
		GPT Trigger Enable	GPT Channel 1 Request A	<input type="checkbox"/>
		GPT Trigger Enable	GPT Channel 2 Request A	<input type="checkbox"/>
		GPT Trigger Enable	GPT Channel 3 Request A	<input type="checkbox"/>
		GPT Trigger Enable	GPT Channel 4 Request A	<input checked="" type="checkbox"/>
		GPT Trigger Enable	GPT Channels 5 to 9 Request A/B	Not in use

		Enable		Enable
		Converter Selection		ADC 0
		Start Trigger Delay		0
		Scan End Interrupt Enable		Enable
		Limit Clip Interrupt Enable		Disable
		FIFO Enable		Disable
		FIFO Interrupt Enable		Disable
		FIFO Interrupt Generation Level		0
	Scan Group 1	Self Diagnosis	Voltage Selection	Self-Diagnosis Mode Disabled
		External Trigger Enable	External Trigger Input 0 (ADTRG0) Enable	<input type="checkbox"/>
		External Trigger Enable	External Trigger Input 1 (ADTRG1) Enable	<input type="checkbox"/>
		ELC Trigger Enable		Not in use
		GPT Trigger Enable	GPT Channel 0 Request A	<input type="checkbox"/>
		GPT Trigger Enable	GPT Channel 1 Request A	<input type="checkbox"/>
		GPT Trigger Enable	GPT Channel 2 Request A	<input type="checkbox"/>
		GPT Trigger Enable	GPT Channel 3 Request A	<input type="checkbox"/>
		GPT Trigger Enable	GPT Channel 4 Request A	<input checked="" type="checkbox"/>
		GPT Trigger Enable	GPT Channels 5 to 9 Request A/B	Not in use
		Enable		Enable
		Converter Selection		<b>ADC 1</b>
		Start Trigger Delay		0
		Scan End Interrupt Enable		Enable
		Limit Clip Interrupt Enable		Disable
		FIFO Enable		Disable
		FIFO Interrupt Enable		Disable
		FIFO Interrupt Generation Level		0
	Scan Groups 2 to 8			Not in use

## 10.9 POEG Settings

The POEG is a peripheral function of the MCU and quickly switches the PWM gating signal under control to the high-Z state when an error has occurred in an inverter circuit for use in motor control. Table 10-12 shows the specifiable functions of the POEG FSP stack. The output pin settings depend on the specifications of the inverter. Confirm the signal specifications of the inverter you are using.

Table 10-12 POEG Settings

Function and Item for Setting			Setting
General	Trigger	GTETRG Pin	<input checked="" type="checkbox"/>
		GPT Output Level	<input type="checkbox"/>
		Oscillation Stop	<input type="checkbox"/>
		ACMPHS0	<input type="checkbox"/>
		ACMPHS1	<input type="checkbox"/>
		ACMPHS2	<input type="checkbox"/>
		ACMPHS3	<input type="checkbox"/>
	Name		g_poeg3
Input	Channel		3
	GTETRG Polarity		Active Low
	GTETRG Noise Filter		PCLKB/32
Interrupts	Callback		callback_poe_overcurrent
	Interrupt Priority		Priority 0 (highest)

## 11. Results of Evaluation

This section describes results of evaluation by using this sample software.

### 11.1 Evaluation of Inrush-Current Prevention Relay Control

From the following waveforms, we have confirmed that the relay control operation is switched at 230 V, which is the bus voltage threshold for judging the timing of turning on the relay, after the input voltage has risen. This turning on of the relay can also be checked by seeing that LED5 on the inverter board is turned on at the same point.

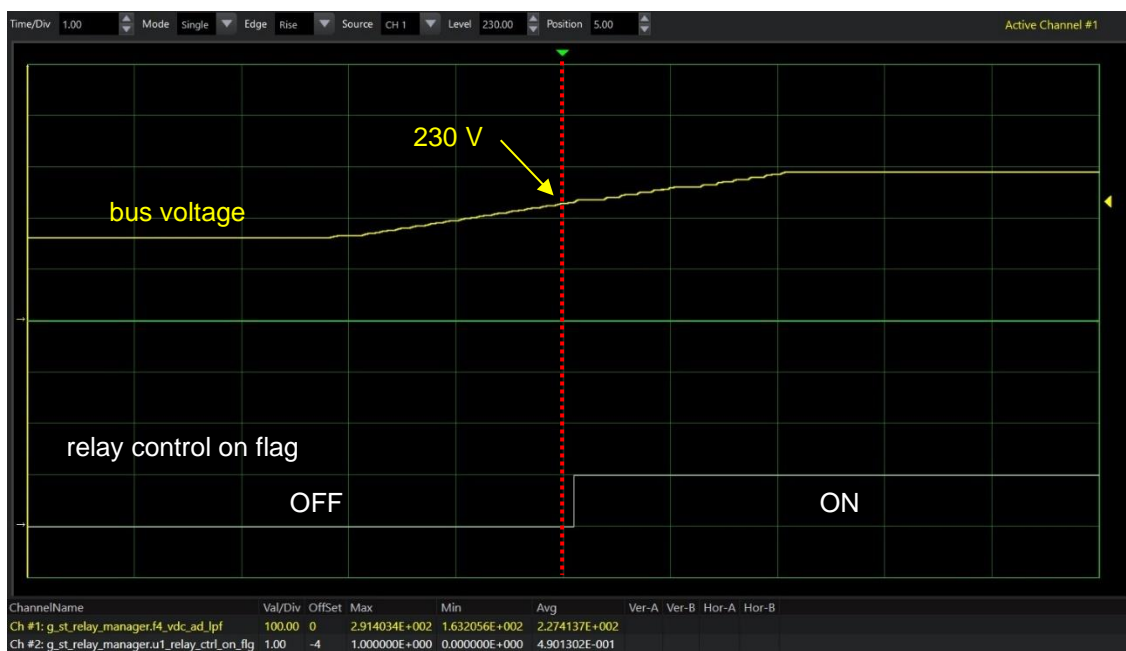


Figure 11-1 Waveforms of Inrush-Current Prevention Relay Control

## 11.2 Evaluation of Motor Control

The following shows the results of evaluation of three-phase 200-VAC 3.7-kW induction motor operation by using this sample software.

### 11.2.1 Acceleration/Deceleration Characteristics

The waveforms obtained in acceleration from the speed at rest (0 rpm) to the Maximum speed of 1800 rpm with no load are shown below. It can be seen from the figure below that q-axis voltage command value changes proportionally to speed command value due to V/f control and hits a plateau due to flux weakening control in the range from 1500rpm to 1800rpm. Input voltage was set to 240V when acquiring this data.

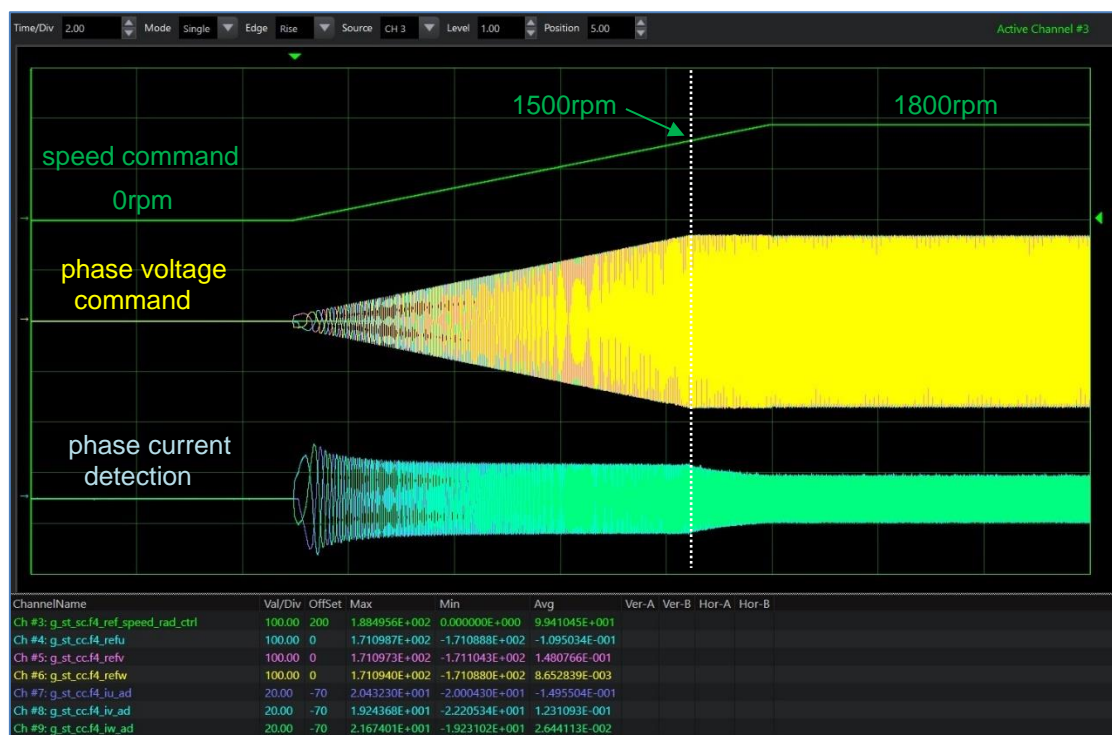


Figure 11-2 Waveforms in Acceleration with No Load

### 11.2.2 Evaluation of Motor Characteristics

The following shows the results of evaluation of the motor characteristics in operation with various loading rates. Input voltage was set to 200V when acquiring this data.

Table 11-1 Results of Evaluation of 3.7-kW Induction Motor Characteristics

Speed command value (rpm)	Rated voltage (V)	Loading Rate (%)	Phase Current (A)	Speed of Axial Rotation* (rpm)
1500	200	0	6.99	1500
		25	7.73	1491
		50	10.1	1481
		75*	13.2	1470
		100*	17.2	1457

Note: \* Data for Loading Rate 100, 75% was measured for short time, since Rated power for MCI-HV-2-3PH is 2.2kW.

### 11.3 CPU Utilization

The following table shows the CPU processing times and loading rates for each control interval.

Table 11-2 Control Loops and CPU Loading Rates

Control Loop Type	Control Interval	Processing Time	CPU Loading Rate
Relay-control loop	1 ms	2.0μs	0.2 %
Current-control loop in motor control	125 μs (no decimation)	14.96 μs	11.968 %
Speed-control loop in motor control	500 μs	1.14 μs	0.228 %

### 11.4 Program Size and RAM Usage

The program size (ROM) and RAM usage for this sample program are as listed below. In the optimization settings for the compiler, the optimization level is set to 2 (-O2).

Table 11-3 Program Size and RAM Usage

Program size (ROM)	25024 bytes
RAM usage	3040 bytes
Maximum value of stack analysis results	160 bytes
Stack size setting in the IDE	1024 bytes

## 12. FAQ

Table 12-1 lists typical problems and examples of their solutions.

Table 12-1 Problems and Examples of Their Solutions

Problem	Example of Solution
An error message indicating that the FSP version is different appeared.	Opening the project in an environment where the version of the FSP is different from that for use with the e <sup>2</sup> studio which is specified in this sample program leads to the display of an error message indicating that the FSP version is different. Download the version of the FSP environment which is specified in this sample program from the Renesas Electronics Web site and install it on the PC you are using. With a different version of the FSP, the specifications of the API functions, etc. may change, and this may require modifications by users. In addition, the result of executing the program or its behavior may change. Note that we cannot support cases where you run the program in an environment where a different version of the FSP is in use.
I attempted to start up the motor, but it did not run.	The load being higher or the inertia being greater than intended for the motor under control may lead to the startup torque being insufficient to start up the motor. Check whether the motor has sufficient capacity for the load and consider the use of a startup method other than V/f control.
A value detected by the ADC immediately after start-up was incorrect.	The ADC_B peripheral function in the RA6T2 requires self-calibration during start-up, a reset, etc. For this reason, skipping the processing for waiting for the completion of calibration during ADC initial settings after start-up may lead to a value detected by the ADC being incorrect. Be sure to include the processing for waiting for the completion of ADC calibration in the processing to be done during start-up.
The motor does not rotate at the low speed.	V/f control only allows a narrow range of operation (the available speeds of operation are up to about 1/10 of the rated speed) and is aimed at applications that only require a small startup torque. The use of V/f control may degrade the characteristics of motor operation, especially at low speeds. This may require the consideration of a control algorithm other than V/f control.
The motor does not run when the acceleration time (speed limiting rate) is short.	When the acceleration time is short, the torque required to produce acceleration becomes too large and, depending on the inertia, acceleration and operation of the motor may be impossible. Determine the appropriate acceleration time for the inertia or use a large-capacity inverter that can output sufficient torque for acceleration.
The bus voltage increases during rapid deceleration of the motor.	Regeneration occurs during deceleration and this raises the bus voltage. Add a regenerative resistor to suppress the rise of the bus voltage or set a longer time for deceleration.
The current waveform becomes distorted when the motor speed is increased.	The current waveform may be distorted if the V/f ratio is not correct, the input voltage is too low, the bus voltage drops, or the motor is rapidly accelerated or decelerated.
The speed decreases when a load is applied.	In V/f control, the speed command values and the actual rotation speed of the axis do not match due to the nature of this method of control, specifically, the effect of slip. The rotation speed will decrease by the speed corresponding to the rated slip under the rated load condition.  If the rated slip and the actual speed of rotation differ significantly, the reason may be an incorrect V/f ratio or an operating condition that exceeds the maximum load of the inverter or motor. In such a case, the amount of slip may increase with the result of decreased speed. If the amount of slip exceeds the maximum value, control of operation will not be possible and the motor will stop.
An overcurrent error occurs when a load is suddenly applied.	V/f control cannot track rapid changes of load. This may require the consideration of a different method of control.

Revision History

Rev.	Date	Description	
		Page	Summary
1.0	Mar. 31, 2025	—	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](http://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/](http://www.renesas.com/contact/)