Renesas RA Family

Sensorless Vector Control for IPMSM over the Whole Speed Range by High-Voltage Inverter

Introduction

This sample program offers the following control algorithms for the RA6T2 CPU board and MCI-HV-1 200-VAC high-voltage inverter from Renesas. These algorithms are mainly for implementing a sensorless vector control function for three-phase interior permanent magnet synchronous motors (IPMSM, hereinafter referred to as IPM motors) that have projecting (salient) poles and principally involving a single PFC (power factor correction) function for use with home appliances.

- Sensorless vector control (sensorless field-oriented control) over the whole speed range of a motor from the standstill state to the low-speed and medium-to-high-speed ranges[1]
- Sensorless vector control of a PM motor through a BEMF observer during medium-to-high-speed operation (3-shunt mode)
- Flux weakening control and maximum torque per current control (maximum torque per ampere, MTPA)[2]
- Torque vibration suppression, step-skipping (stall) detection, and flying start (pick-up control)
- Single PFC control (power factor correction and voltage boost functions)

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.

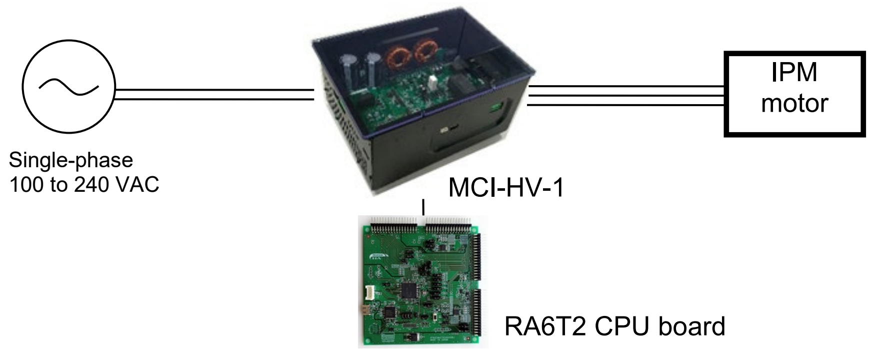


Figure 1-1 Hardware Configuration

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)

---

[1] This algorithm is not applicable to a surface permanent magnet synchronous motor (SPMSM, hereinafter referred to as a SPM motor) that has no characteristics difference between d-axis inductance (Ld) and q-axis inductance (Lq) or an IPM motor having conditions under which the characteristics difference becomes less than 20% while the motor is started up, stopped, or operating.

[2] The MTPA function is only applicable to an IPMSM. It cannot be used with an SPMSM.

Contents

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 permanent magnet synchronous motor with vector control over its whole speed range, including from standstill to the low-speed range. It is also intended to describe the configuration, specifications, and method of control by the software.

Although the conventional sample program for sensorless vector control is applicable to PM motors in general, this sample is only applicable to a certain type of PM motors called IPM motors. This is because this sample program utilizes an IPM motor characteristics called saliency to estimate the magnetic pole position even when the motor is standstill or operating at low speeds. Motors other than IPM motors (for example, SPM motors) are outside the scope of this sample program, because they do not have the saliency characteristic and the program cannot estimate the magnetic pole position without a sensor when the motor is standstill or operating at low speeds.

This sample program can control an EM-AMF 1.5kW motor (a 3-phase 200-VAC PM motor from Mitsubishi Electric Corporation) without a sensor by using the RA6T2 CPU board from Renesas and an MCI-HV-1 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.

For how to set up, use, and check the operation of the MCI-HV-1 inverter manufactured by Renesas, refer to the MCI-HV-1 User's Manual (R12UZ0138).

The sample program described in this application note was developed and evaluated in the environment of the IPM motor and inverter described in this document and is not guaranteed to work with your IPM motor or inverter environment. The sensorless control performance may be limited by the current sensor itself; the PCB design pattern of the signal path; sampling, resolution, and filter specifications; magnetic saturation characteristics of the motor and variations between individual motors. Under the responsibility of the user, refining the 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-1 inverter from Renesas
Motor: PM motor EM-AMF 1.5kW from Mitsubishi Electric Corporation

Target Software

The following shows the target software for this application note.

- RA6T2_MCIHV1_IPM_LESS_FOC_WHOLE_PFC_E2S_V100 (IDE: e^2 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-1 User's Manual (R12UZ0138)

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.	4.4
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.	6.7
Drive the motor.	6.9
Stop the motor.	6.10
Examine the motor control algorithms.	7
Examine the PFC control algorithms.	8
Examine the structure of the sample program.	9
Examine and change the inverter parameters.	10.7, 10.4, 11.8, 11.11, 11.4
Examine and change the motor parameters.	10.8, 10.4
Change the PWM carrier frequency for motor control.	10.5
Change the sensorless control settings.	10.14
Change the MCU settings.	10.2, 11
Check the frequently asked questions. Check the troubleshooting tips.	13

2. Glossary

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

Table 2-1 Glossary

Term	Description
BEMF	Back electromotive force. Refers to an inductive voltage.
HFI	Refers to application of a high-frequency pulse voltage (high-frequency injection). Often used to refer to a low-speed-range sensorless algorithm.
IDE	An integrated development environment such as e^2 studio.
IPM motor	Also called an IPMSM. This type of motor has magnets inside the rotor and is considered superior in terms of efficiency, size, and cost. Such motors also have saliency, in which the Ld and Lq are different.
MC-COM	A set of communication jigs and tools connected for displaying waveforms. For details, refer to the following URL. https://www.renesas.com/en/products/microcontrollers-microprocessors/rx-32-bit-performance-efficiency-mcus/rtk0emxc90s00000bj-mc-com-renesas-flexible-motor-control-communication-board#overview
PFC	Power factor correction. In addition, the boost function is also included as part of the PFC functions in this document.
RMW	Renesas Motor Workbench, which is software specifically designed for motor control operations.
SPM motor	Also called an SPMSM. This type of motor is used for servo motors that require smooth motion even at low speeds.
Salient PMSM	A type of PM motor.
Inverter bus voltage	The DC voltage fed to the inverter circuit. Also known as DC intermediate voltage or bus voltage.
Emulator	A device used to program an MCU. Also called an ICE.
Stack	A driver module generated by the FSP to facilitate the use of MCU peripheral functions.
Sensorless	In this document, this is used to indicate that there is no magnetic pole position sensor or speed sensor.
Feedback control	A method of control that uses feedback signals obtained by current or speed detection.
Interior permanent magnet synchronous motor	An IPMSM or an IPM motor.
Surface permanent magnet synchronous motor	An SPMSM or an SPM motor.
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.
Magnetic saturation	Phenomenon in which the motor is magnetically saturated and the magnetic flux is no longer intensified because a current above a certain

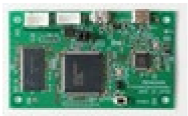
	level is applied. It causes the parameters to change, thus affecting motor control by the inverter.

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-1 RTK0EM0000B14030BJ
Isolated communication board MC-COM 	Renesas	Renesas Flexible Motor Control Communication Board RTK0EMXC90S00000BJ
IPM motor	Mitsubishi Electric	EM-AMF 1.5kW
AC power supply unit	KIKUSUI ELECTRONICS	PCR2000MS
Power meter	Yokogawa Test & Measurement	WT500
Torque meter and load system	Sugawara Laboratories Inc.	TB-5N
Torque meter controller	Sugawara Laboratories Inc.	DMC-3

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 e² studio.

Table 3-2 List of Software Tools that are Used

Manufacturer	Software Tool	Version	Remark
Renesas	e² studio	2024-04	Free version
Renesas	FSP	5.4.0	
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 an IPM motor is operated by using this sample program. Figure 4-1 shows a sample hardware configuration.

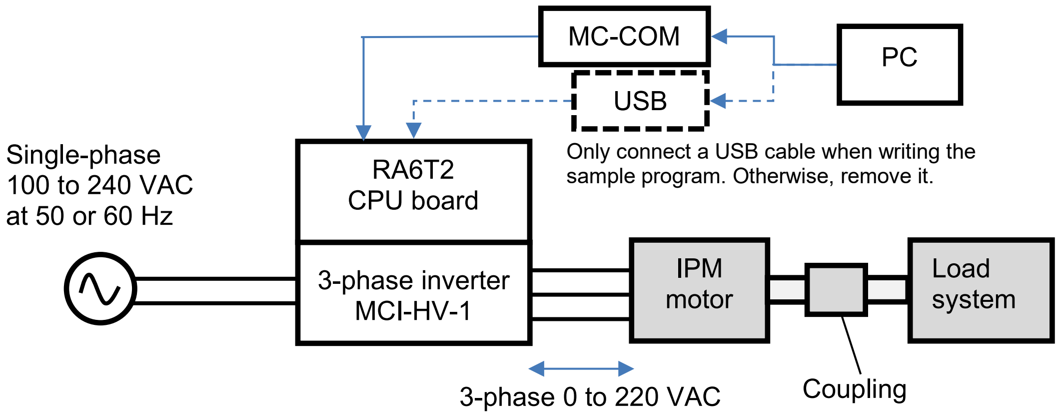


Figure 4-1 Sample Hardware Configuration

4.2 Preparing a Power Supply

The MCI-HV-1 inverter used for this sample program can receive single-phase 100 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 IPM 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 2.0 kVA or more must be prepared so that it can drive a 1.5-kW IPM motor.

4.3 Preparing a Motor

Before connecting the inverter to a motor, obtain the parameters and constants of the IPM motor that are required to drive the motor with sensorless vector control by using a measuring instrument such as an LCR meter. In addition, contact the manufacturer of the IPM motor to obtain the parameter information as required.

If motor parameters are changed, the following parameters for the current regulator, speed regulator, and sensorless control should be changed accordingly.

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

Table 4-1 shows the parameters of the EM-AMF1.5kW motor from Mitsubishi Electric Corporation, 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. Note that the magnetic saturation caused by the load current may change the motor parameter values during operation, thus affecting the position estimation accuracy or operational performance. Therefore, this sample program may not be able to operate properly depending on the motor. (See Section 7.12.2 (d) for details.).

Table 4-1 EM-AMF 1.5kW Motor Parameters (Some Values are Based on Our Own Measurements)

Primary resistance R	0.976375 Ω
d-axis inductance	0.004715 H
q-axis inductance	0.006245 H
Moment of inertia	0.00114 kgm^2
Magnetic flux linkage ψ	0.18 Wb (rms)
Inductive voltage Emf	240 Vpeak
Number of pole pairs	3 (number of poles: 6)
Rated speed	3000 rpm
Maximum speed	4000 rpm
Rated frequency	150 Hz (electrical angle), 50 Hz (mechanical angle)
Rated current	6.1 Arms
Rated torque and maximum torque	4.78 Nm and 9.56 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 a 1-kW or larger load. 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.

4.5 Preparing an Inverter

When preparing an inverter, note the following information. This sample program is configured for the MCI-HV-1 inverter board.

In sensorless vector control, the magnetic pole position is estimated by using the current detection value input from the current sensor. Therefore, the control performance is greatly influenced by the performance of the sensor itself and the accuracy and variations of the circuits that serve as paths for the signals output from the sensor. When selecting an inverter, careful consideration must be given to the design of the inverter:

- Rated capacity (kVA)
- Dead time value (μs)
- 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-1. You can plug the RA6T2 CPU board to the top of the MCI-HV-1 board. A connector for writing the sample program, a connector for MC-COM, and the PG pin for an external encoder are also provided.

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	Pins 1 and 2 are closed.	INV1 PFC current detection (for the inverter board)
JP8 and JP9	—	
JP10	Pins 1 and 2 are closed.	INV1 AC input voltage detection (for the inverter board)
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

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 single-phase 200 VAC power at 50 Hz is used and it is connected to the ACINL and ACINN pins of CN2. 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 CN5. Ground the FG pin of CN2 or CN5 to ensure safety.

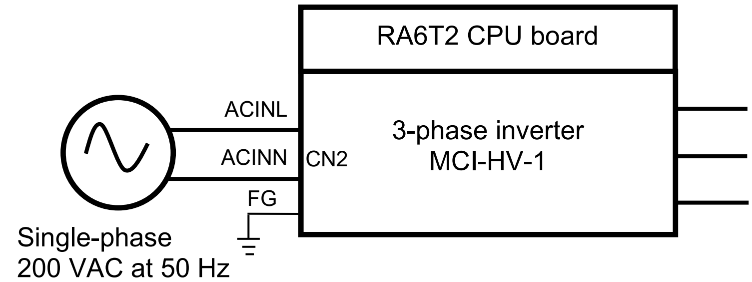


Figure 4-3 Wiring between the Power Supply and Inverter

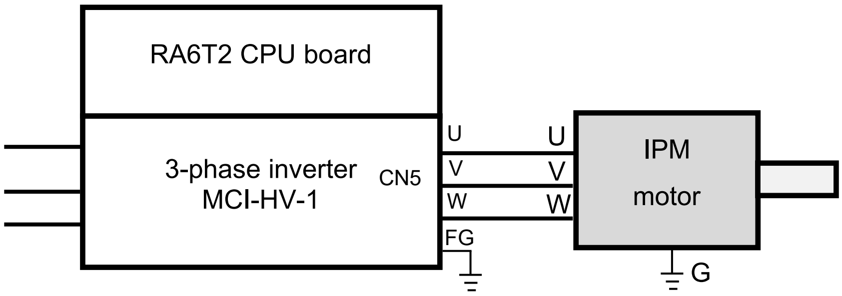


Figure 4-4 Wiring between the Inverter and Motor

4.8 Using Measuring Instruments

When evaluating the sensorless control performance of an IPM motor, using a power meter, a digital multimeter, a torque meter, or an external encoder enables 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.

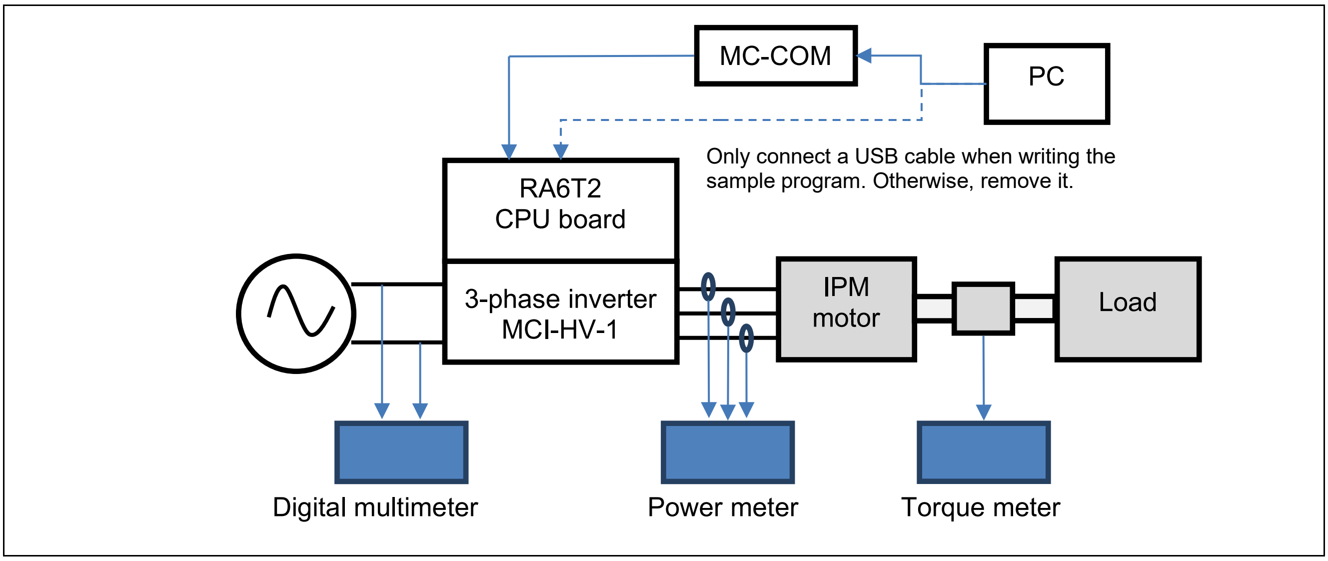


Figure 4-5 Example of Additional Measuring Instruments

5. Configuring a Software Environment

The e² studio is used for development of a system. Download it from the following site. Note that the FSP v5.4.0 used in this sample program is necessary in addition to the e² studio.

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

The "FSP with e² studio" package, which contains both the FSP v5.4.0 and e² 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 e² studio, refer to the PDF manual that you can download from the above e² 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-1 inverter is intended for use in home appliances. The main circuits and CPU board are not isolated. The GND of the CPU board is at the same potential as the N terminal of the main circuit. Consider isolation of the signal and power lines when connecting signals to external devices or instruments.
- Do not apply the power-supply voltage (100 to 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-1.
- 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 while 100 to 200-VAC power is being supplied because the signals are isolated. When the USB connector on the CPU board is used, the GND of the PC and the inverter may be common because the PC and inverter are not isolated, which could lead to an electric shock hazard, the intrusion of noise to the PC, or damage to the PC via the GND.
- 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 IPM motor is still rotating, the IPM motor generates an inductive voltage, thus applying voltage to the U/V/W three-phase wiring. Touching an exposed conductive part may cause an electric shock. If the inverter is stopped during high-speed rotation and the inverter bus voltage is lowered, the energy from the motor flows into the bus of the inverter, which may generate an overvoltage and cause failure of the inverter. In the evaluation environment, place an electromagnetic switch between the inverter and IPM 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^2 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 RA6T2 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 RA6T2 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.

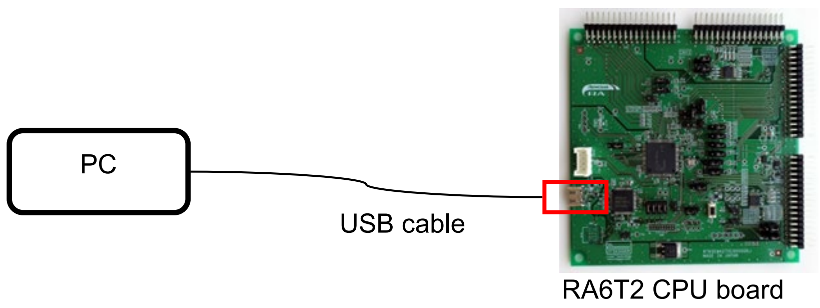


Figure 6-1 Example of Connection for Writing

(2) Motor operation

Use MC-COM (RTK0EMXC90S00000BJ) 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.

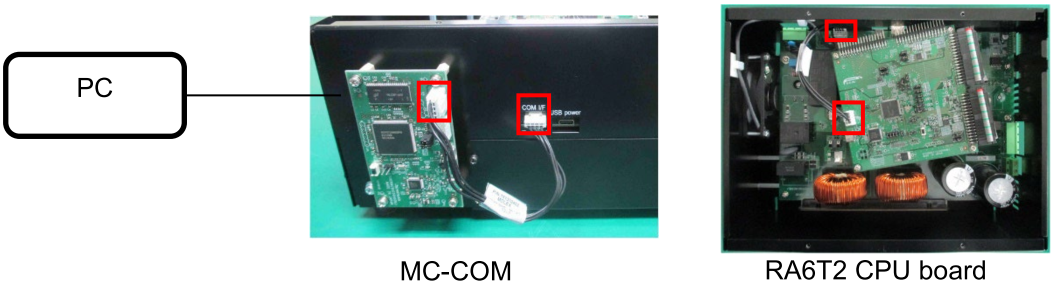


Figure 6-2 Example of Connections for Motor Operation

6.4 Writing the Sample Program

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

For details about how to write programs, see the documentation for the e^2 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^2 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

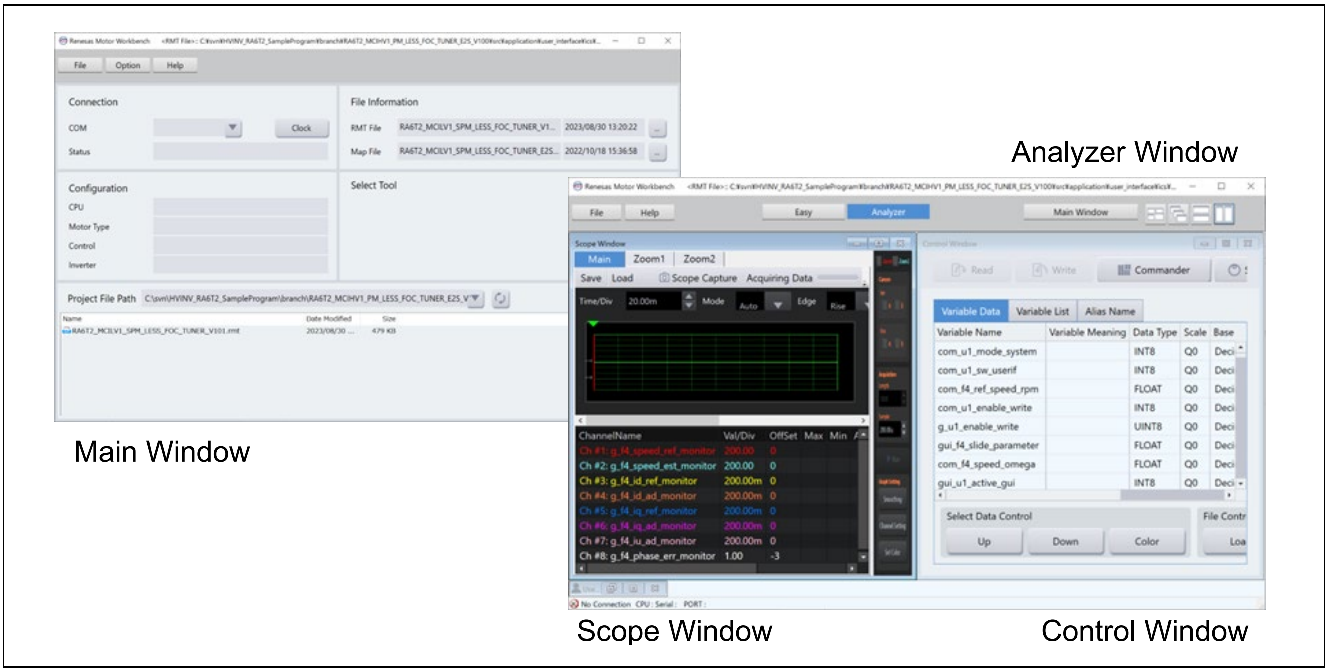


Figure 6-3 Windows of Renesas Motor Workbench

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.

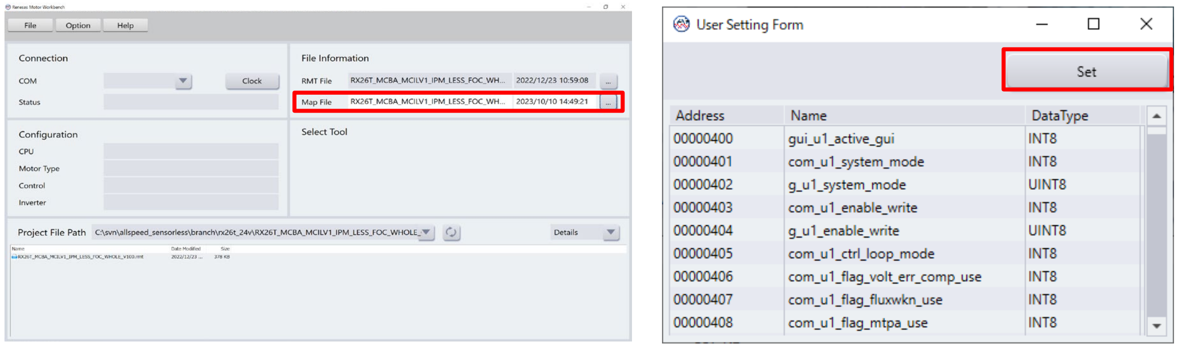


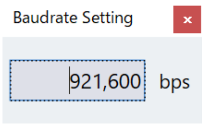
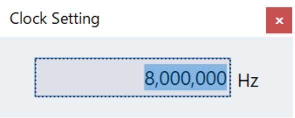
Figure 6-4 Map File Registration Setting (Left) and Setting Confirmation Window (Right)

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	
Clock setting	8,000,000 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_update_param_flag	uint8_t	Buffer transfer completion flag
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_sensorless_vector.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_cc.f4_vdc_ad	float	Inverter bus voltage (V)
g_st_cc.f4_id_ref	float	d-axis current command value (A)
g_st_cc.f4_id_ad	float	d-axis current detection value (A)
g_st_cc.f4_iq_ref	float	q-axis current command value (A)
g_st_cc.f4_iq_ad	float	q-axis current detection value (A)
g_st_cc.f4_iu_ad	float	U-phase current detection value (A)
g_st_cc.f4_iv_ad	float	V-phase current detection value (A)
g_st_cc.f4_iw_ad	float	W-phase current detection value (A)
g_st_cc.f4_vd_ref	float	d-axis voltage command value (V)
g_st_cc.f4_vq_ref	float	q-axis voltage command value (V)
g_st_cc.f4_refu	float	U-phase voltage command value (V)
g_st_cc.f4_refv	float	V-phase voltage command value (V)
g_st_cc.f4_refw	float	W-phase voltage command value (V)
g_st_cc.st_rotor_angle.f4_rotor_angle_rad	float	Estimated magnetic pole position (rad)
g_st_sc.f4_ref_speed_rad_ctrl	float	Speed command value (mechanical angle) (rad/s)
g_st_sc.f4_speed_rad	float	Speed detection value (mechanical angle) (rad/s)

The following com variables can be used to dynamically change the constants, gains, and other 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_mtr_r	Resistance of the motor to be driven* (Ω)
com_f4_mtr_ld	d-axis inductance of the motor to be driven* (H)
com_f4_mtr_lq	q-axis inductance of the motor to be driven* (H)
com_f4_mtr_m	Magnetic flux of the motor to be driven* (Wb)
com_f4_mtr_j	Rotor inertia of the motor to be driven (kgm^2)
com_f4_nominal_current_rms	Rated current of the motor to be driven (Arms)
com_f4_max_speed_rpm	Maximum speed (mechanical angle) of the motor to be driven (rpm)
com_f4_current_omega_hz	Natural frequency for the current control system (Hz)
com_f4_current_zeta	Attenuation coefficient for the current control system
com_f4_speed_omega_hz	Natural frequency for the speed control system (Hz)
com_f4_speed_zeta	Attenuation coefficient for the speed control system

Variable	Description
com_f4_speed_lpf_hz	Speed LPF cut-off frequency (Hz)
com_f4_speed_rate_limit_rpm	Maximum width for incrementation and decrementation (of mechanical angle) per control interval in response to the speed command (rpm) (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_u1_flag_mtpa_use	Maximum torque per current control setting 0: Disable, 1: Enable
com_u1_flag_fluxwkn_use	Flux weakening control setting 0: Disable, 1: Enable
com_u1_flag_flying_start_use	Flying start setting 0: Disable, 1: Enable
com_u1_flag_stall_detection_use	Step-skipping (stall) detection setting 0: Disable, 1: Enable
com_u1_flag_trq_vibration_comp_use	Torque vibration suppression setting 0: Disable, 1: Enable
com_f4_e_obs_omega_hz	Natural frequency for the inductive voltage estimation system (Hz)
com_f4_e_obs_zeta	Attenuation coefficient for the inductive voltage estimation system
com_f4_pll_est_omega_hz	Natural frequency for the position estimation system (Hz)
com_f4_pll_est_zeta	Attenuation coefficient for the position estimation system
com_f4_id_hpf_time	Step-skipping (stall) detection: Time constant of d-axis current HPF (s)
com_f4_iq_hpf_time	Step-skipping (stall) detection: Time constant of q-axis current HPF (s)
com_f4_threshold_level	Step-skipping (stall) detection: Detection level (A)
com_f4_threshold_time	Step-skipping (stall) detection: Detection time (s)
com_f4_timelead	Torque vibration suppression: Output phase adjustment value
com_f4_tf_lpf_time	Torque vibration suppression: Filter constant
com_f4_output_gain	Torque vibration suppression: Output gain
com_u1_flag_trqvib_comp_learning	Torque vibration suppression: Learning function enabling flag
com_f4_input_weight2	Torque vibration suppression: Input weight 2
com_f4_input_weight1	Torque vibration suppression: Input weight 1
com_f4_input_weight0	Torque vibration suppression: Input weight 0
com_f4_restart_speed	Flying start: Restart judgement speed (rpm)
com_f4_off_time	Flying start: Switched-off time (s)
com_f4_over_time	Flying start: Limit time for being switched on (s)
com_f4_active_brake_time	Flying start: Brake time (s)
com_f4_on_current_th	Flying start: On-time current threshold (A)
com_f4_pll_estlow_omega_hz	Natural frequency of the low-speed-range sensorless control PLL (Hz)
com_f4_pll_estlow_zeta	Attenuation coefficient of the low-speed-range sensorless control PLL

Variable	Description
com_u1_flag_extobserver_use	Enables or disables the disturbance torque/speed estimation observer.
com_f4_extobs_omega	Natural frequency of the disturbance torque/speed estimation observer (Hz)
com_f4_spd_low_to_high_threshold	Sets the speed at which the sensorless algorithm switches from the low-speed range to the medium-to-high-speed range (rad/s).
com_f4_spd_high_to_low_threshold	Sets the speed at which the sensorless algorithm switches from the medium-to-high-speed range to the low-speed range (rad/s).

Note: The values marked with * can only be reflected in the variables while the motor is stopped.

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 RA6T2 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 100-VAC or 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 inverter bus voltage is automatically boosted to 390 V.

(c) Starting the RMW

Start the RMW installed in your PC. After that, select the COM port corresponding to the RA6T2 CPU board and select "Analyzer" to establish a connection.

(d) Starting rotation of the motor

The correct operation of the PFC control requires checking. Confirm that "g_st_pfc_manager.u2_run_mode" is set to 3.

After this confirmation, follow the steps below.

(1) Click on the [Read] button and confirm that a voltage of approximately 390 V is applied to “g_st_sensorless_vector.f4_vdc_ad”.
(2) Confirm that "g_st_sensorless_vector.u2_error_status" is 0. If it is not 0, perform the operation described in (f) given later 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.

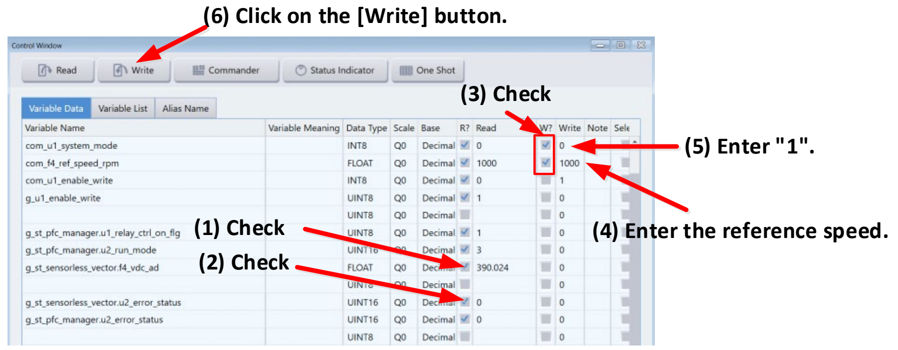


Figure 6-5 Procedure for Starting Rotation of the Motor

(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.

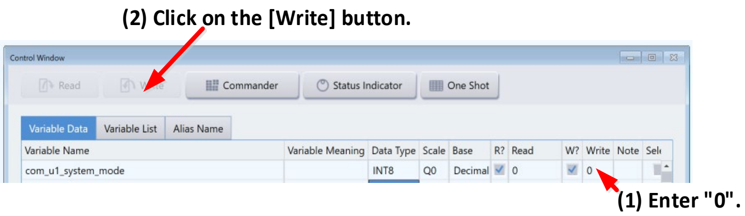


Figure 6-6 Procedure for Stopping the Motor

(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 value of “g_st_sensorless_vector.u2_error_status” while referring to Table 6-6 and take action in response to the cause.
For an error in PFC control, check the value of “g_st_pfc_manager.u2_error_status” while referring to Table 6-7.
(2) In the "com_u1_system_mode" row, enter "3" in the [Write] column.
(3) Click on the [Write] button.

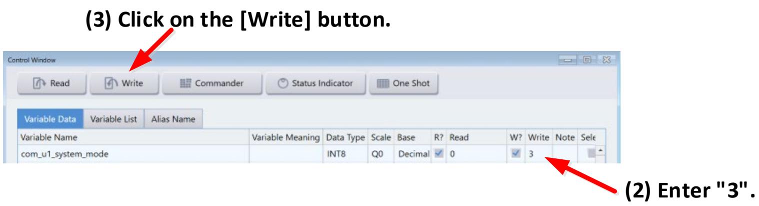


Figure 6-7 Procedure for Clearing the Error Condition

Table 6-6 Description of Errors in Motor Control

Value	Error Description	Assigned Macro Name
0x0000	No error	MOTOR_SENSORLESS_VECTOR_ERROR_NONE
0x0001	Hardware overcurrent error	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_CURRENT_HW
0x0002	Overvoltage error	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_VOLTAGE
0x0004	Overspeed error	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_SPEED
0x0008, 0x0010, 0x0020, 0x0040	Reserved	-
0x0080	Low-voltage error	MOTOR_SENSORLESS_VECTOR_ERROR_LOW_VOLTAGE
0x0100	Software overcurrent error	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_CURRENT_SW
0x0200	Step-skipping (stall) error	MOTOR_SENSORLESS_VECTOR_ERROR_STALL_DETECTED
0x0400	PFC overcurrent error	MOTOR_SENSORLESS_VECTOR_ERROR_PFC
0x0800	Axis difference error	MOTOR_SENSORLESS_VECTOR_ERROR_FAIL_POLES
0x1000	Position estimation error	MOTOR_SENSORLESS_VECTOR_ERROR_FAIL_POSITION
0xffff	Undefined error	MOTOR_SENSORLESS_VECTOR_ERROR_UNKNOWN

Table 6-7 Description of Errors in PFC Control

Value	Error Description	Assigned Macro Name
0x0000	No error	PFC_MANAGER_ERROR_NONE
0x0001	Vac overvoltage error	PFC_MANAGER_ERROR_AC_OVER_VOLTAGE
0x0002	Vdc overvoltage error	PFC_MANAGER_ERROR_BUS_OVER_VOLTAGE
0x0004	Vdc low-voltage error	PFC_MANAGER_ERROR_BUS_LOW_VOLTAGE
0x0008	Software overcurrent error	PFC_MANAGER_ERROR_OVER_CURRENT_SW
0x0010	Hardware overcurrent error	PFC_MANAGER_ERROR_OVER_CURRENT_HW
0x0020	Overheat error	PFC_MANAGER_ERROR_OVER_HEATING
0xffff	Undefined error	PFC_MANAGER_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 inverter bus voltage has dropped from 390 VDC to less than 42 VDC.

7. Motor Control Algorithms

7.1 Overview

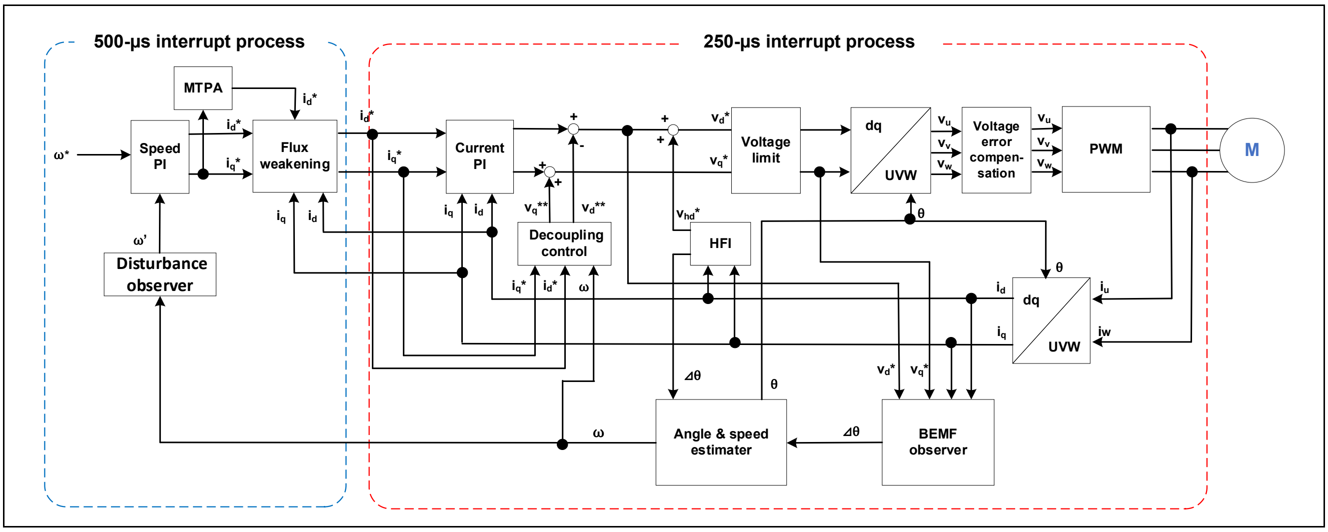
This section describes the motor control algorithms of this sample program. Table 7-1 lists the motor control functions.

Table 7-1 Motor Control Functions of This Sample Program

Function	Description
Control method	Sensorless vector control
Pulse width modulation (PWM) method	Space vector modulation method (sinusoidal modulation can also be selected)
Position and speed estimation method	Low-speed range: High-frequency pulse voltage injection (HFI) Medium-to-high-speed range: BEMF observer
Control mode	Only speed control
Compensation functions	• Maximum torque per current control (MTPA) • Flux weakening control • Disturbance torque/speed estimation observer • Voltage error compensation • Sample delay compensation • Decoupling control • Torque vibration suppression • Flying start • Step-skipping (stall) detection

7.2 Control Block Diagram

The following shows a sample block diagram of the overall control system.



Note: While a high-frequency pulse voltage is being applied, a feedback current after removal of the current ripple generated in the pulses is injected into the current proportional–integral (PI) controller and BEMF observer.

Figure 7-1 Schematic Block Diagram of Sensorless Vector Control System (Sensorless Control)

7.3 Speed Control Function

The speed control function performs PI control so that the motor follows the speed command. In response to an input speed command value, the internal speed regulator outputs a q-axis current command value based on the deviation from the estimated speed value. This function module also controls the submodules for flux weakening control and maximum torque per current control (MTPA).

The estimated speed after having been passed through the LPF is used as the estimated speed value.

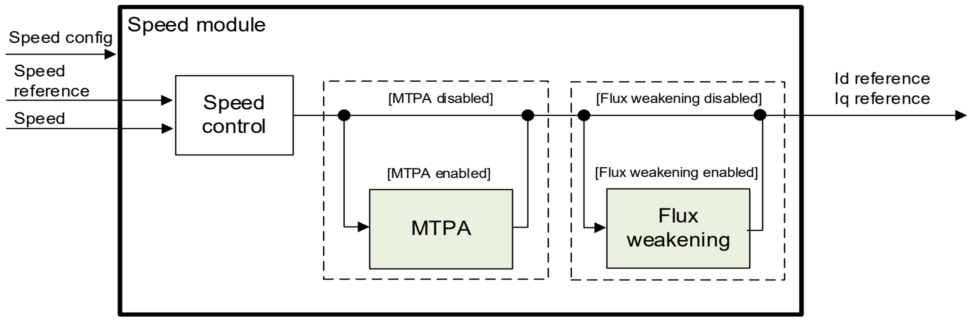


Figure 7-2 Functional Block Diagram of Speed Control

7.4 Maximum Torque per Current Control (MTPA)

For a PM motor having saliency like an IPM motor, maximum torque per current control (MTPA) can be applied. MTPA uses the reluctance torque, which is not used in control with Id = 0, and is therefore capable of adjusting the torque per unit current to the maximum. Note that the reluctance torque for an SPM motor, which does not have saliency, is difficult to use due to the motor's structure and therefore cannot be used with MTPA. When the motor in use is an SPM motor, the Ld and Lq values of which are about the same, be sure to disable MTPA. Otherwise, this sample program will not work correctly. The sample program does not automatically determine whether to enable MTPA by monitoring the Ld and Lq values.

The equation used is shown below. The d-axis current command value can be obtained using the q-axis current command value Iq* output by the speed regulator as input.

$$I^*{}_d = \frac{\Psi}{2(L_q - L_d)} - \sqrt{\left(\frac{\Psi}{2(L_q - L_d)}\right)^2 + {I_q^*}^2}$$

Ψ: Magnetic flux linkage (Wb), L_d, L_q: d-axis inductance and q-axis inductance of the motor (H)

7.5 Flux Weakening Control

The flux weakening control function controls the d-axis current in the negative direction. Even under conditions where the inductive voltage (= $\omega\psi$) generated in proportion to the rotation of the PM motor exceeds the voltage that can be output from the inverter bus voltage, this function increases the d-axis current command value in the negative direction to reduce the inductive voltage (Figure 7-3). Through this reduction in the voltage saturation region, the q-axis current command value, which is necessary for acceleration, can be increased and the output torque in the high-speed rotation region and acceleration of rotation can thus be improved.

This function automatically detects the state where the speed of motor rotation has become fast and the margin of the available voltage in comparison with the current voltage has become small. In this state, this function increases the Id* value in the negative direction and cancels the inductive voltage according to the voltage equation of the PM motor.

To achieve this, the equation shown in Figure 7-4 is used to obtain the maximum limit on the inductive voltage. R is the resistance value of the motor and Ia is the square root of the sum of squares of the detected Id and Iq values (√ (Id × Id + Iq × Iq)). For Vamax, the maximum magnitude of the voltage vector that has been calculated in voltage error compensation or modulation processing is used.

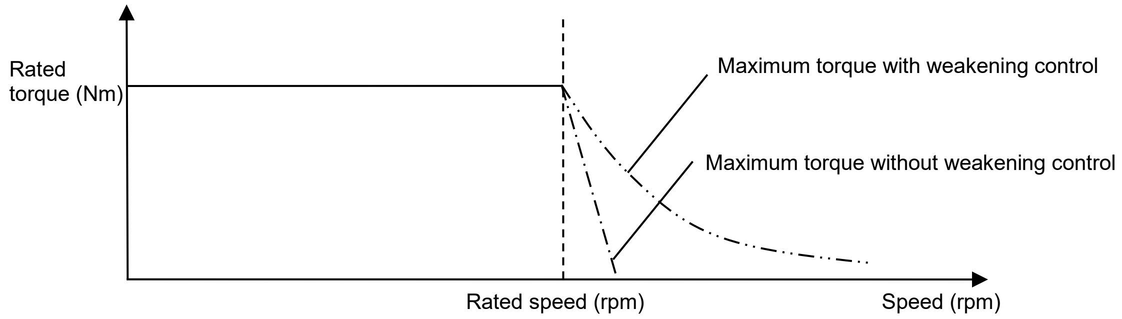


Figure 7-3 Relationship between the Available Output Torque and Speed

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

V_{om}: Maximum limit on inductive voltage (V), V_{amax}: Maximum magnitude of voltage vector (V),

I_a: Magnitude of current vector (A)

Figure 7-4 Equation for Calculating the Maximum Limit on the Inductive Voltage

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

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

V_{om}: Maximum limit on inductive voltage (V), V_{amax}: Maximum magnitude of voltage vector (V),
I_a: Magnitude of current vector (A)

Figure 7-5 Equation for Calculating the d-axis Current Command Value in Flux Weakening Control

7.6 Disturbance Torque/Speed Estimation Observer

This function is for reducing ripples in speed in the very low-speed range from around 1 to 30 rpm by applying an observer-based speed estimation algorithm. The observer takes the torque and speed (ω) calculated from the q-axis command value (I_{q_ref}) as input, and obtains an estimated speed ($\hat{\omega}$) and disturbance torque based on the plant model. The observer can reduce speed ripple and has less influence on the control system than ordinary filter processing. It is also possible to reduce the impact by the sensor's quantization error and speed ripple due to noise.

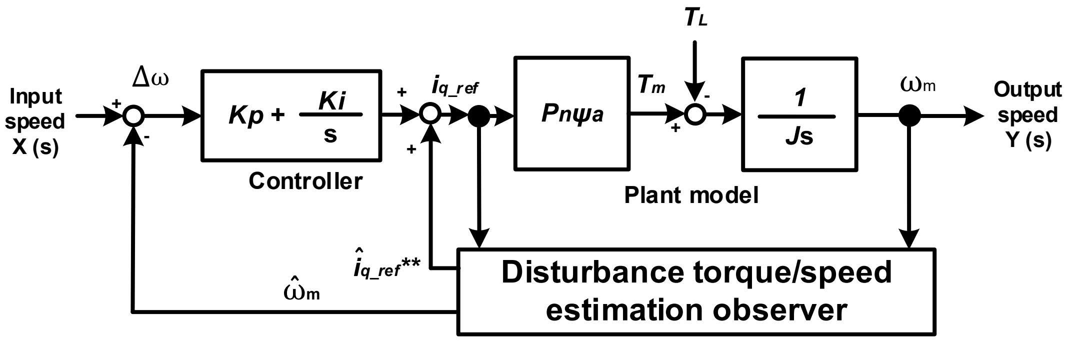


Figure 7-6 Block Diagram for the Disturbance Torque/Speed Estimation Observer

7.7 Current Control Function

The current control function uses the value of the incoming current to perform coordinate transformation and feedback control that are necessary for vector control and then calculates the voltage of the PWM output. This function module also controls submodules for decoupling, sample delay compensation, voltage error compensation, and BEMF observer processing. Figure 7-7 shows the configuration of the module.

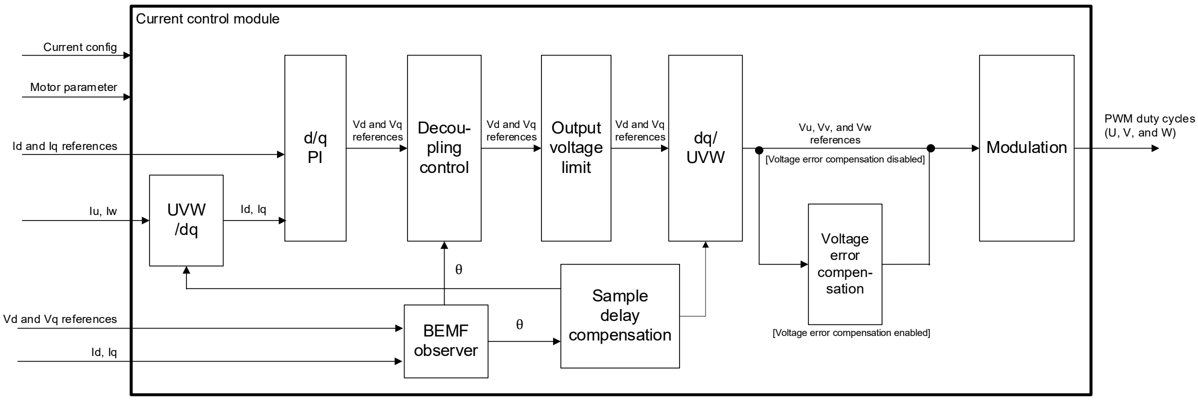


Figure 7-7 Functional Block Diagram for Current Control

RENESAS

7.8 Decoupling Control

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

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

Id*, Iq*: Current command values (A), ω: Rotational velocity (electrical angle) (rad/s),
R: Primary resistance of the motor (Ω),
Ld, Lq: Inductances of the motor (H), Ψ: Magnetic flux linkage of the motor (Wb)

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

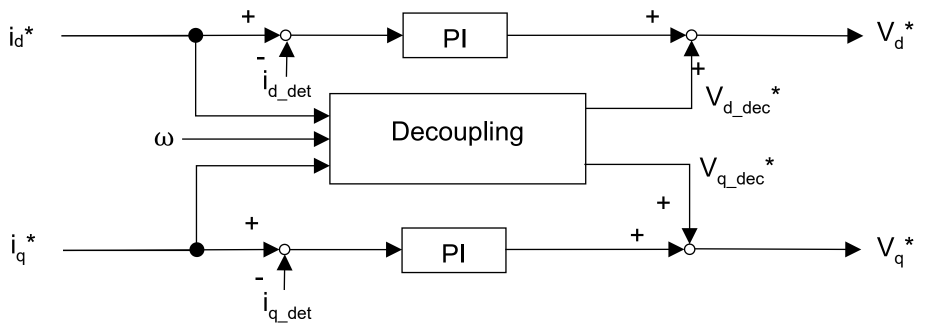


Figure 7-8 Functional Block Diagram of Decoupling Control

7.9 Step-Skipping (Stall) Detection

Step-skipping (stall) is a condition in which the magnetic pole position estimated by the motor control software deviates from the actual magnetic pole position of the motor. This produces serious problems such as rapid decreases in speed, overcurrents, or rotation at unexpected speeds.

The step-skipping (stall) detection function judges the step-skipping (stalled) condition based on the change in the oscillation of current and stops the motor for protection. Note that this function does not guarantee that all step-skipping (stalled) conditions are detected. Prepare multiple protection measures as required.

This function focuses on the AC component of the detected Id or Iq value; it detects and accumulates the component through a high-pass filter (HPF) and uses the result for judgement. In general, the detected current values on the d and q axes correspond to the rotation frequency and appear as DC values. If the estimated magnetic pole position deviates from the actual position, the values are detected as AC values and this function is based on this fact. Therefore, when the motor has stalled or the actual speed is slower than the internally detected speed, this function may not detect the situation in some cases. In cases where this function does not work as expected, other functions such as the overcurrent protection function will handle the situation.

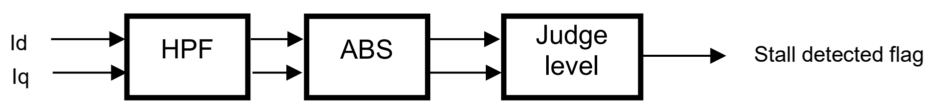


Figure 7-9 Functional Block Diagram of Step-Skipping (Stall) Detection

7.10 Torque Vibration Suppression

The torque vibration suppression function is aimed at suppressing torque vibration at a frequency of one rotation in mechanical degrees that may occur in a single-rotary or reciprocating compressor used in an air conditioner or a refrigerator.

In low-to-medium-speed operation in particular, the vibration of a compressor causes problems such as noise; use this function to reduce such vibration. Note that the algorithm for this function uses the speed estimated by sensorless control and performs feedforward control to estimate and cancel the torque vibration. Suppression of vibration may in some cases not be fully effective due to the parameter settings or the configuration of the compressor or equipment.

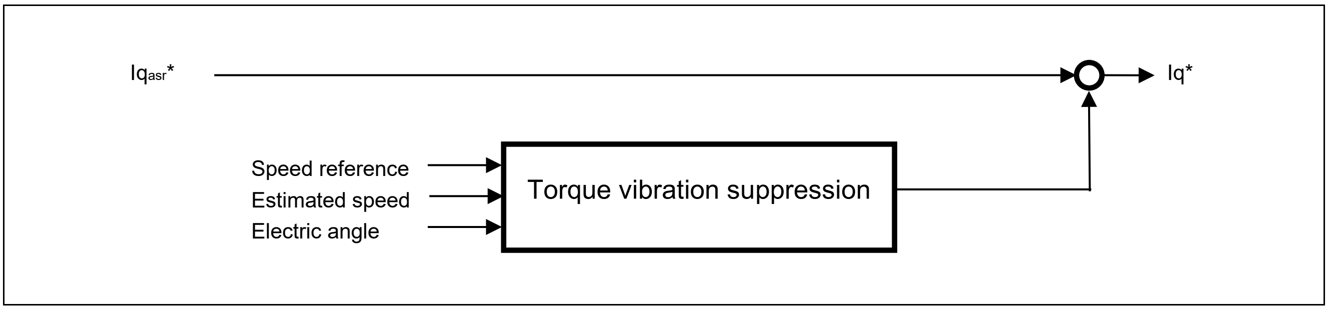


Figure 7-10 Functional Block Diagram of Torque Vibration Suppression

This sample program performs the following feedforward control: it first uses a simplified Fourier transform and a repetitive controller to detect only the frequency component that is synchronized with one rotation in mechanical degrees, then obtains the current command value for cancelling the torque vibration, and adds it to the q-axis current command value.

The repetitive controller should be considered as a discrete time system. A repetitive controller that periodically operates N times in one cycle can be represented as shown in the figure below. To implement this controller, a table (array) for mechanical angles obtained by dividing one rotation in mechanical degrees by N is prepared and the input values for the individual angles produced by division are stored in the table.

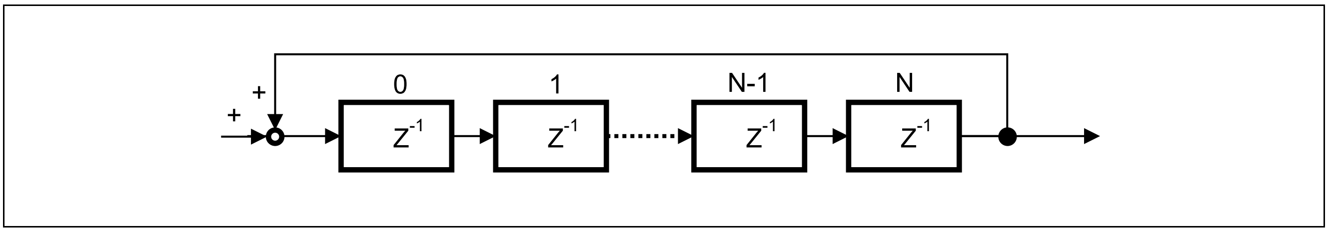


Figure 7-11 Block Diagram of Repetitive Control (Discrete Time System)

• Advance compensation

The advance compensation processing uses the table of N entries held by the repetitive controller and obtains and outputs the value of the element offset by j which corresponds to the angles specified for advance compensation, from the i-th value which is the location of the mechanical angle calculated from the current magnetic pole position.

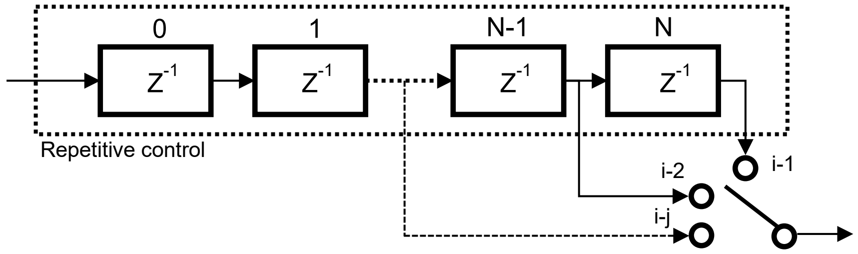


Figure 7-12 Block Diagram of Advance Compensation (Discrete Time System)

• Learning enabling or disabling function

If the repetitive control is continued, overlearning will occur, leading not only to vibration not effectively being suppressed but possible also causing overcompensation that increases the vibration. Therefore, a function for enabling or disabling learning is provided to turn the input to the repetitive controller on or off.

7.11 Flying Start

The flying start function is used in the state where the inverter has stopped (all switches have been turned off) during motor rotation but the motor is still rotating and the control system estimates the rotational velocity and magnetic pole position of the motor and re-activates the inverter.

When the inverter is restarted from the stopped state, the switching elements of the lower side of the three-phase inverter are turned on twice (Figure 7-14) and the vector of the current flowing through the switches due to the inductive voltage of the rotor is used to estimate the initial rotational velocity and magnetic pole position. Figure 7-13 shows the processing for a flying start. The switching elements of the three-phase lower side of the inverter are simultaneously turned on in the periods from t1 to t2 and from t3 to t4 and the rotational velocity and magnetic pole position are estimated from the phases of the rotation current vectors at times t2 and t4. At time t5, the calculated initial rotational velocity and magnetic pole position are used to set the initial values in the position and speed estimating system and speed PI control system and start the inverter.

The algorithm for this function defines restarting as being allowed while the rotational velocity is within the range in which operation of the BEMF observer is possible. If the estimated rotational velocity is beyond the range in which the BEMF observer can operate, the switching elements of the three-phase lower side are turned on for a specified period to generate the brake torque and stop the motor, after which normal activation processing proceeds.

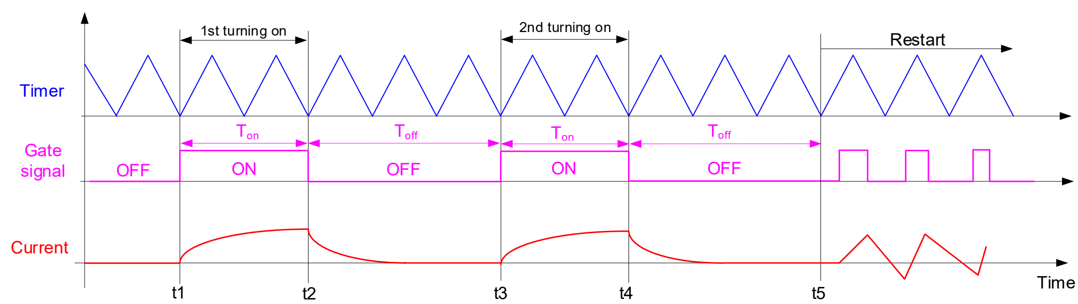


Figure 7-13 Sample Waveform of Flying Start Operation

(1) Detection of rotational velocity

Figure 7-14 shows the relationship between the phases of the rotation current vectors obtained by turning on twice. Two-phase currents $i\alpha$ and $i\beta$ are calculated from the three-phase currents iu, iv, and iw and the phase angles $\theta 1$ and $\theta 2$ of the current vectors at the times of the first and second turning on are calculated by using a trigonometric function (atan2). From the current vector phase angles $\theta 1$ and $\theta 2$ and the pulse-on and off times Ton and Toff, the electrical angular velocity of rotation ω is calculated by using equation 7.11.1.

$$\omega = \frac{\theta_2 - \theta_1}{T_{on} + T_{off}}$$ Equation 7.11.1

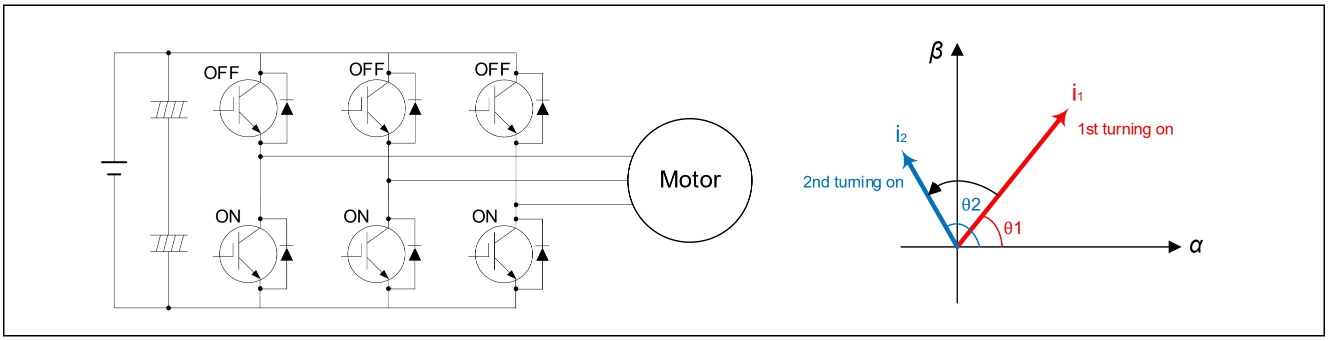


Figure 7-14 Trajectory of Current Vector by Turning on Twice

If the rotation current vector is rotated by π (180 degrees) or more when the elements are turned on twice, the direction of rotation cannot be determined, so (Ton + Toff) needs to satisfy the following condition. Here, ω_{max} is the maximum electrical angular velocity of rotation.

$$Ton + Toff < \pi/\omega_{max}$$ Condition 7.11.2

(2) Detection of magnetic pole position

The voltage equation in the dq-axis rotation coordinate system is given as equation 7.11.3.

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R + pL_d & -\omega L_q \\ \omega L_d & R + pL_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \omega\psi \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$ Equation 7.11.3

Here, v_d and v_q are the d-axis and q-axis voltages, i_d and i_q are the d-axis and q-axis currents, R is the winding resistance, L_d and L_q are the d-axis and q-axis inductances, ψ is the inductive voltage coefficient for the rotor, and p is a differential operator. When the three-phase elements are turned on ($v_d = 0$ and $v_q = 0$) in the above equation, equation 7.11.4 is obtained. Note that the turning-on time Ton is short enough with respect to the electrical time constant L_q/R and R is approximated by 0.0 ($R \simeq 0.0$).

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} pL_d & -\omega L_q \\ \omega L_d & pL_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \omega\psi \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$ Equation 7.11.4

A Laplace transform is applied to the above equation with the condition of initial current i(0) = 0 to calculate the current vector i(T) at time T and equation 7.11.5 is obtained.

$$i(T) = \begin{bmatrix} i_d(T) \\ i_q(T) \end{bmatrix} = \begin{bmatrix} -\frac{\psi}{L_d}(1 - \cos\omega T) \\ -\frac{\psi}{L_q}\sin\omega T \end{bmatrix}$$ Equation 7.11.5

The current vector phase angle θ_a in the dq-axis rotation coordinate system is calculated from the angular velocity of rotation ω and turning-on time Ton by using the following equation.

$$\theta_a = \mathrm{atan2}\left(\frac{i_q}{i_d}\right) = \mathrm{atan2}\left(\frac{-\frac{\psi}{L_q}\sin\omega T_{on}}{-\frac{\psi}{L_d}(1-\cos\omega T_{on})}\right) = \mathrm{atan2}\left(\frac{L_d \sin\omega T_{on}}{L_q(1-\cos\omega T_{on})}\right) \quad \text{Equation 7.11.6}$$

The dq coordinate system of the rotor in the vector control system is a rotation coordinate system based on the α axis (U phase) of the αβ coordinate system, so the magnetic pole position θ_r on the d axis is finally calculated as follows.

$$\theta_r = \theta_i - \theta_a = \mathrm{atan2}\left(\frac{i_\beta}{i_\alpha}\right) - \mathrm{atan2}\left(\frac{i_q}{i_d}\right) \quad \text{Equation 7.11.7}$$

Figure 7-15 shows the relationship of the phases between the rotation current vector and magnetic pole position in the case of turning on for the second time. θ_a is the phase angle of the current vector I_a from the d axis and θ_i is the phase angle of the current vector I_a from the α axis.

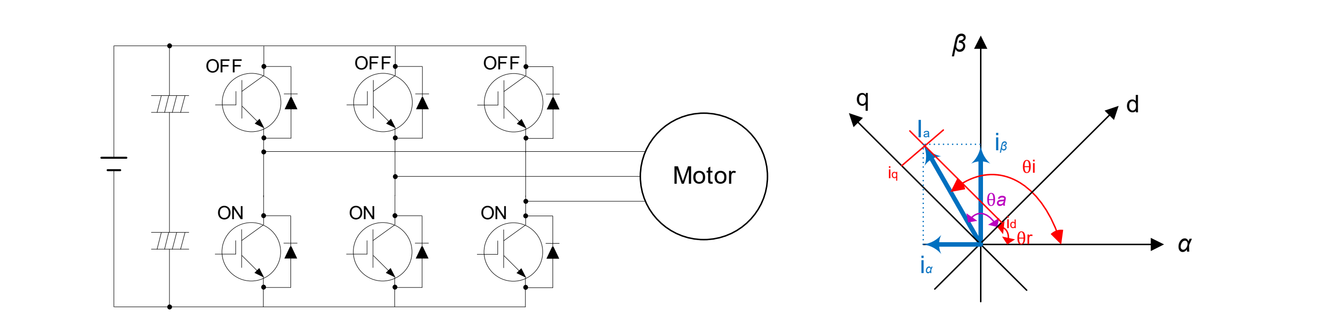


Figure 7-15 Relationship between the Current Vectors and Magnetic Pole Positions at the Second Time of Turning on

(3) Design of control parameters

Design the parameters related to the Ton and Toff times as follows.

Table 7-2 Design of Parameters for Controlling Flying Start

Maximum time of Ton + Toff ((Ton + Toff)max)	The following describes the relationship between (Ton + Toff)max and rotational velocity by using equation 7.11.1. (Ton + Toff)max at the maximum rotational velocity of 4000 rpm is equal to the time required for 0.5 of a rotation in electrical degrees, which is 3.75 ms. Therefore, (Ton + Toff) needs to be designed so that it never exceeds 3.75 ms over the entire range of rotational velocity. To achieve this, determine this parameter according to the target motor and the specifications of the maximum rotational velocity.
Ton time	See the description of SENSORLESS_VECTOR_FLY_START_CURRENT_TH in section 10.16.
Toff time	See the description of SENSORLESS_VECTOR_FLY_START_OFF_TIME_SEC in section 10.16.

7.12 Sensorless Function

7.12.1 Overview

Sensorless full closed loop control is achieved over the whole speed range including the zero speed, by combining the magnetic pole position estimation by high-frequency pulse voltage injection (HFI) utilizing saliency from the zero speed to low-speed range (500 rpm or less) and magnetic pole position estimation by using an BEMF observer in the medium-to-high-speed range (500 rpm or more).

In open loop control, the load limit was restricted to about half to prevent the motor from stepping out. Whereas in whole-speed-range sensorless control, you can apply as high a load as required. The problem regarding energy consumption due to the continuous flow of excitation current is also cleared.

The algorithm must be switched when the speed is increased from the low range to the medium-to-high range. This switching is performed automatically when the speed set by a parameter is reached.

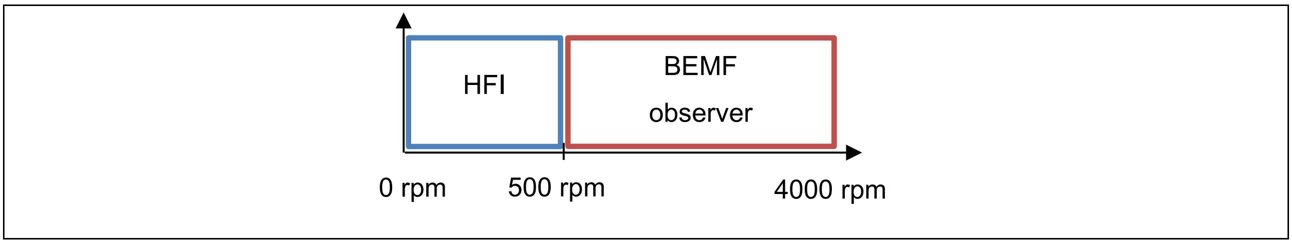


Figure 7-16 Sensorless Control Algorithm Corresponding to the Speed Range

7.12.2 Low-speed-range Sensorless Algorithm (HFI)

(a) Overview

The low-speed-range sensorless algorithm applies a high-frequency pulse voltage when the motor is standstill or running at low speeds to estimate the magnetic pole position of the IPM motor from its response. The high-frequency pulse voltage is applied to the d-axis voltage command value, which has little effect on the motor's rotative force.

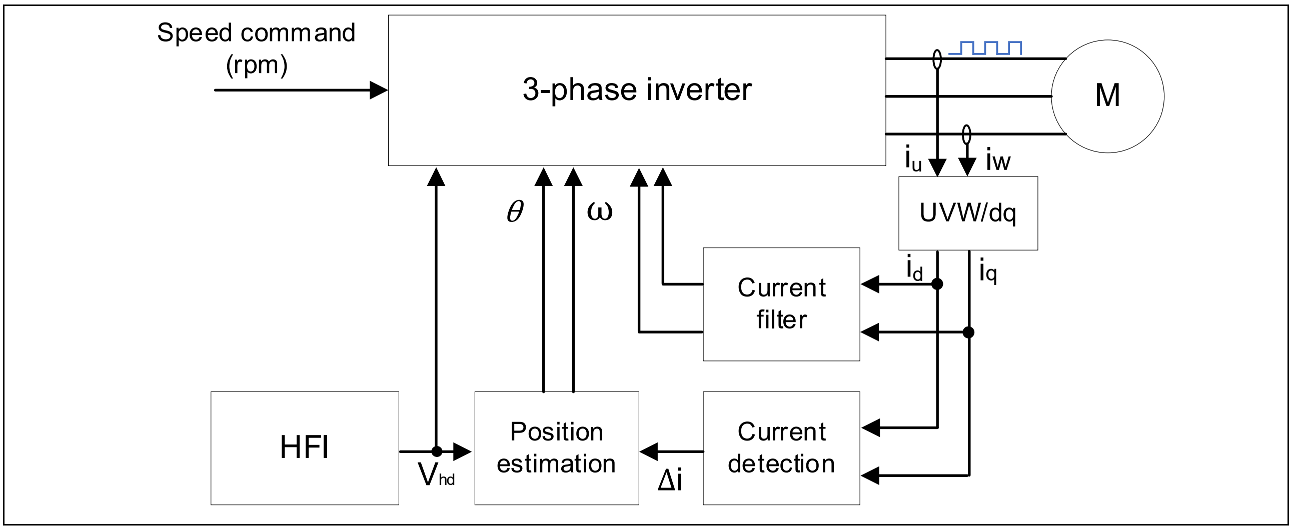


Figure 7-17 Overview of the Low-speed-range Sensorless Algorithm

(b) High-frequency pulses and response current

The voltage of positive and negative high-frequency pulses with a period of 1/2 to 1/8 of the PWM carrier cycle is applied to the d-axis voltage command value. Since the IPM motor has different Ld and Lq values due to its inherent structural characteristics, the current values Id and Iq that flow in response to the high-frequency pulses vary with the magnetic pole position of the IPM motor, according to the difference in the ratio of Ld and Lq. This phenomenon is used to estimate the magnetic pole position of the IPM motor from the detected current values Id and Iq, Ld and Lq, and pulse voltage values. Note that this algorithm is only applicable to IPM motors, because Ld and Lq values are the same for SPM motors, and therefore angle-related current change according to the magnetic pole position does not occur.

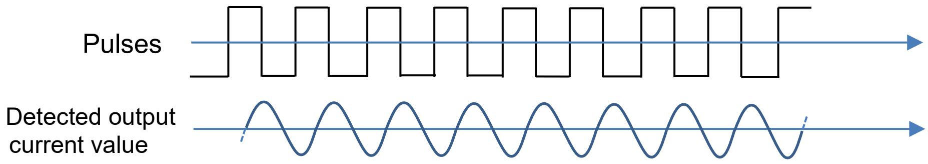


Figure 7-18 Example of Pulses and Response Current

(c) Estimating the angle

When the motor is running at low speeds or is standstill, the motor does not generate an inductive voltage, so that angle estimation using the BEMF observer etc. cannot be performed. Therefore, angle estimation is performed by intentionally applying high-frequency pulses, without using an inductive voltage.

As shown in Figure 7-19, the reference axis for angle estimation is defined as the dc-qc axis. Control is performed by finding Δθ so that the dc-qc axis is aligned with the dq axis.

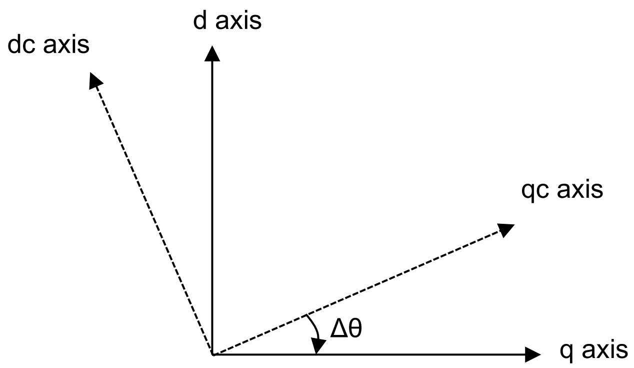


Figure 7-19 Definition of Δθ

Assuming that the rotational speed is 0 or slow (ω ≒ 0), the equation of state for the current at low speed or standstill is derived from the voltage equation of the PM motor.

$$\frac{d}{dt}\begin{bmatrix} i_{dc} \\ i_{qc} \end{bmatrix} = \frac{1}{L_{dc}L_{qc}}\begin{bmatrix} L_{qc} + (L_{qc} - L_{dc})sin^2\Delta\theta & (L_{dc} - L_{qc})sin\Delta\theta\ cos\Delta\theta \\ (L_{qc} - L_{dc})sin\Delta\theta\ cos\Delta\theta & L_{dc} + (L_{dc} - L_{qc})sin^2\Delta\theta \end{bmatrix}\begin{bmatrix} v_{dc} \\ v_{qc} \end{bmatrix} - \frac{R_s}{L_{dc}L_{qc}}\begin{bmatrix} L_{qc} + (L_{qc} - L_{dc})sin^2\Delta\theta & (L_{dc} - L_{qc})sin\Delta\theta\ cos\Delta\theta \\ (L_{qc} - L_{dc})sin\Delta\theta\ cos\Delta\theta & L_{dc} + (L_{dc} - L_{qc})sin^2\Delta\theta \end{bmatrix}\begin{bmatrix} i_{dc} \\ i_{qc} \end{bmatrix}$$

Using the equation of state for the current at low speed or standstill, the angle estimation error Δθ is derived from current response when a pulse voltage is applied to the dc-axis voltage command value. Focusing on the derivatives of current d/dt • i_{dc}, d/dt • i_{qc} when the high-frequency pulses are applied, the following equation is derived when i_{dc}=i_{qc}=0, the pulse voltage = V_{dc}, and V_{qc}=0.

$$\frac{d}{dt}\begin{bmatrix} i_{dc} \\ i_{qc} \end{bmatrix} = \frac{1}{L_{dc}L_{qc}}\begin{bmatrix} L_{qc} + (L_{qc} - L_{dc})sin^2\Delta\theta & (L_{dc} - L_{qc})sin\Delta\theta\ cos\Delta\theta \\ (L_{qc} - L_{dc})sin\Delta\theta\ cos\Delta\theta & L_{dc} + (L_{dc} - L_{qc})sin^2\Delta\theta \end{bmatrix}\begin{bmatrix} v_{dc} \\ 0 \end{bmatrix}$$

Calculate focusing on the q-axis current derivative d/dt • i_{qc}.

$$\frac{d}{dt}i_{qc} = \frac{(L_{qc} - L_{dc})sin\Delta\theta\ cos\Delta\theta}{L_{dc}L_{qc}}v_{dc} = \frac{(L_{qc} - L_{dc})sin2\Delta\theta}{2L_{dc}L_{qc}}v_{dc}$$

When Δθ is small enough, sin2Δθ can be approximated by 2Δθ. This equation can be transformed into an equation for Δθ and derived as follows:

$$\Delta\theta = \frac{L_{dc}L_{qc}}{(L_{qc} - L_{dc}) \cdot v_{dc}}\frac{d}{dt}i_{qc}$$

A PLL of Δθ yields ω, which can be further integrated to derive the estimated angle θ.

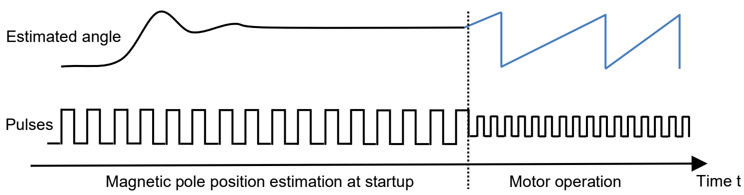


Figure 7-20 Example of the Estimated Angles and High-frequency Pulse Injection

(d) Polarity determination at startup

In the magnetic pole position estimation by applying a high-frequency pulse voltage, positions can be estimated in the range of ±90° (180°). Therefore, if the magnetic pole position is located at a 180° reversed position relative to the estimated angle, the motor will rotate in the opposite direction. This is a problem in that the magnetic pole positions estimated in the previous section cannot be used as they are.

Therefore, during startup, a voltage higher than the high-frequency pulse voltage to be applied in normal operation is applied to estimate whether the motor's magnetic pole position is facing the N pole or S pole based on the magnetic saturation phenomenon. Therefore, immediately before starting the motor, the motor may generate a slightly louder high-frequency noise than during operation. This function waits for the estimation results of the magnetic pole position estimation process to stabilize before performing this function, so set a waiting time of several milliseconds. The waiting time depends on the natural frequency of the position estimation PLL and the motor.

Causing a magnetic saturation phenomenon generates a slight difference in the direction of the responding pulse voltage and the magnitude of the response current, depending on whether the magnetic pole position faces the N or S pole. When the integrated value of this difference is negative, the magnetic pole position is judged to face the N pole, and when it is positive, the position is judged to face the S pole.

If the N- or S-pole position estimation fails, the motor will rotate in the opposite direction and steps out. To prevent the motor from stepping out, a large current that is sufficient to cause a magnetic saturation phenomena must be applied.

Polarity determination is not performed when the motor is rotating. This is because, if the magnetic pole position deviates by 180° or more, the motor cannot rotate properly and generates errors such as overcurrent or steps out, preventing it from continuing operation. Therefore, it is considered sufficient to perform polarity determination only at startup.

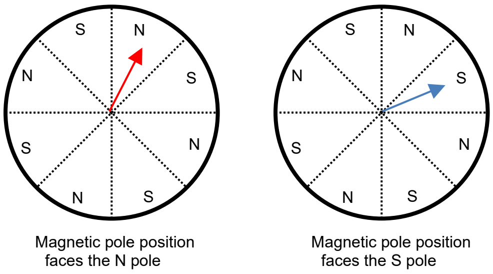


Figure 7-21 Magnetic Pole Positions and N and S Poles in an 8-pole IPM Motor

IPM motors have id-Id characteristics as shown in Figure 7-22. In the case of general IPM motors, the polarity can be determined by the relationship between large and small. However, in the case of special IPM motors, the relationship between large and small may be reversed, and the polarity determination may fail.

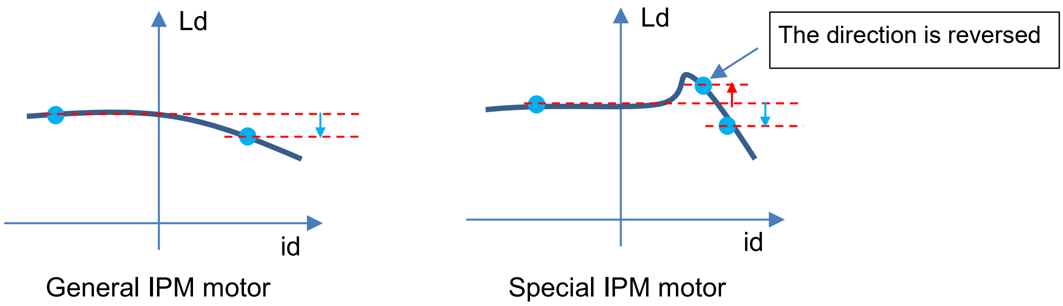


Figure 7-22 id-Ld characteristics of IPM motor during polarity determination

(e) Magnetic pole position estimation at startup

Figure 7-22 shows the magnetic pole position estimation operation at startup. At startup, a high-frequency pulse voltage is applied to the d-axis voltage command value Vd*. The high-frequency pulse voltage must be set to a value higher than the voltage applied during normal operation so that polarity determination as described above can be performed at the same time during startup. The appropriate pulse voltage value varies depending on the motor.

After having applied a high-frequency pulse voltage, the sample program waits for the position convergence period of 200 ms and then determines the convergence of the magnetic pole position estimation within the position estimation convergence determination period of up to 100 ms. The convergence is determined when the following condition is met: the difference between the previous angle value and the current angle value is confirmed to be within 1 degree 10 consecutive times. If the condition for the position estimation convergence determination is not met within 100 ms, it is defined as a magnetic pole position estimation error.

In addition, a polarity determination period is set at the same time as the position estimation convergence determination period. If the absolute value of the PF value, which is the integral of the difference in response currents obtained by the polarity determination algorithm, is no greater than 2.5f, polarity determination is considered unsuccessful, which is defined as a polarity determination error.

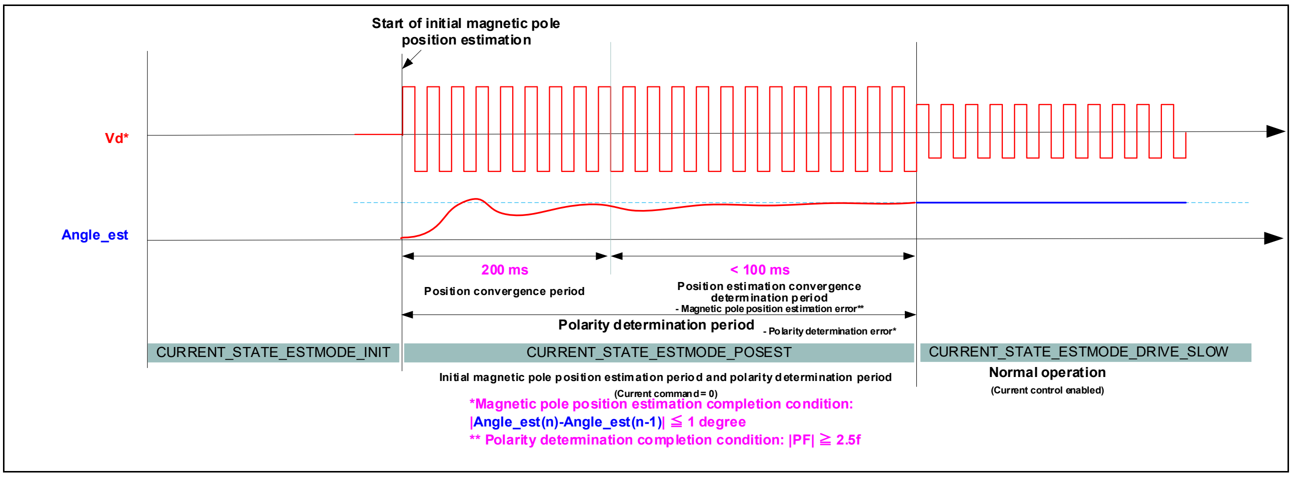


Figure 7-23 Magnetic Pole Position Estimation Operation at Startup

(f) Switching to and from the medium-to-high-speed range

The algorithm is switched when the switching speed is reached. When the sensorless algorithm switches from the low-speed range to the medium-to-high-speed range, the state sequence is made to operate so that the high-frequency pulse injection is halved to reduce current fluctuations. When the speed at which the sensorless control algorithm switches from the medium-to-high-speed range to the low-speed range has been reached, a sequence for starting high-frequency pulse injection is performed. The switching speed can be adjusted using the parameters described below.

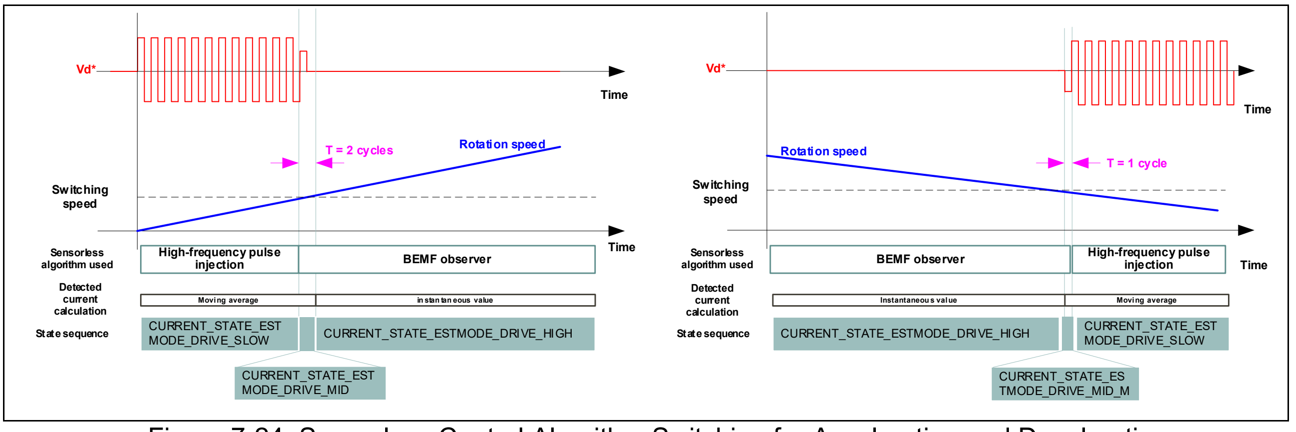


Figure 7-24 Sensorless Control Algorithm Switching for Acceleration and Deceleration

7.12.3 Sensorless Control Algorithm for the Medium-to-high-speed Range

In the medium-to-high-speed range, the motor is controlled by sensorless vector control using a BEMF observer. The algorithm for the BEMF observer is described in detail in section 5.6, Inductive voltage observer (current control module), of the application note “Sensorless Vector Control of a Permanent Magnet Synchronous Motor for the Evaluation System for BLDC Motor (R01AN6307EJ0110)” on which this sample program is based.

7.13 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.

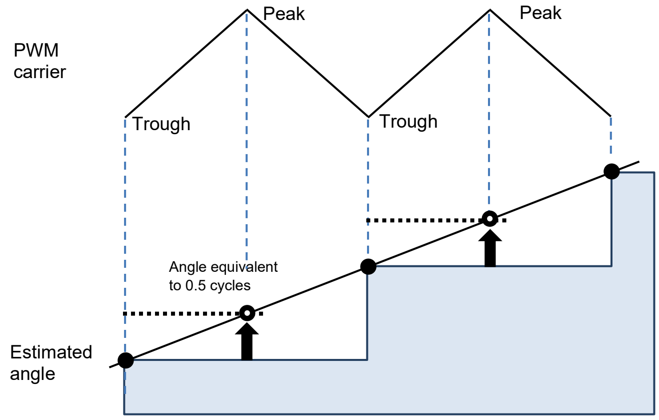


Figure 7-25 Example of the Amount by which the Angle is Advanced in a PWM Carrier Cycle

7.14 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.

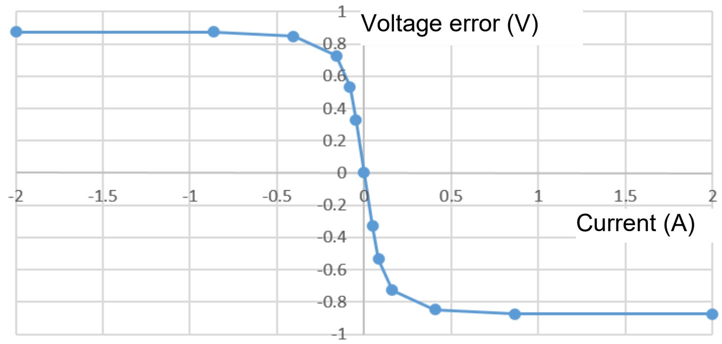


Figure 7-26 Example of the Dependency of the Voltage Error on the Current

7.15 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 line voltage) 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 line voltage) 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 line voltage is unchanged.

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

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

V_u, V_v, V_w: 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. Power Factor Correction (PFC) Control Algorithms

8.1 Overview

The PFC function corrects the power factor of the input current and boosts the inverter bus voltage. The MCI-HV-1 has circuits intended for interleaved PFC but this sample program only provides a single PFC function.

The sample program receives AC voltage Vac, PFC control current Ipfc, and inverter bus voltage Vdc as input and boosts the inverter bus voltage to a specified level while controlling the power factor. The following sections show the block diagram of this control and describe the internal control algorithms.

8.2 Block Diagram of PFC Control

Figure 8-1 is a block diagram of the overall PFC control circuits. The outer loop is a voltage control system that inputs the difference between the target voltage command value and the PFC output voltage to a controller and calculates the current command values to be input to the inner-loop current control system.

The inner-loop current control system detects the instantaneous value of the current flowing through the reactor (Lf) and applies PI control to the current so that the current follows the current command value generated by the voltage control system. The inner loop also obtains the feedforward duty compensation values that are proportional to the AC input voltage and DC output voltage, uses the value to generate the duty cycle of the gate driving signal for the PFC circuits, and controls the phases of the input voltage and current (power factor correction).

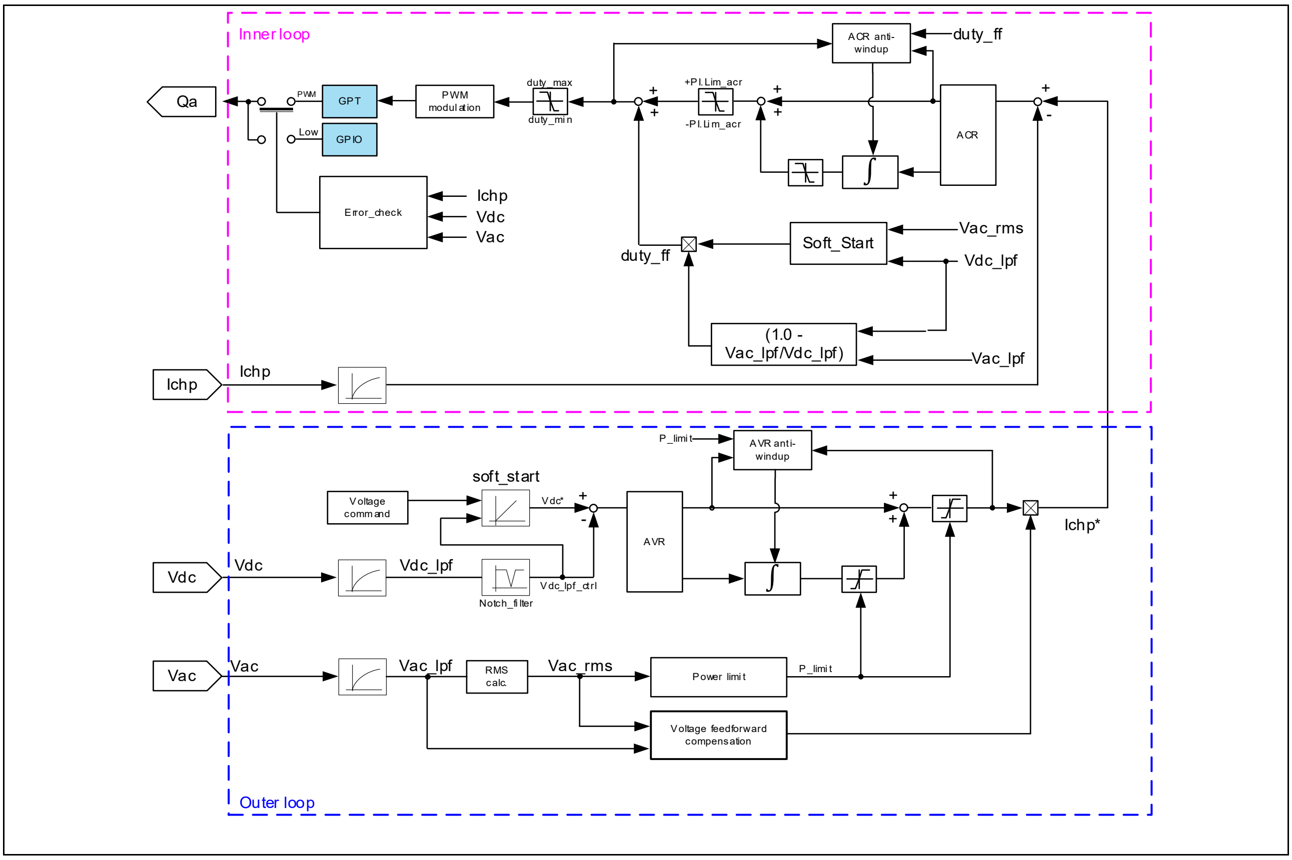


Figure 8-1 Block Diagram of PFC Control

8.3 Voltage Control Function

This facility calculates the differences between the PFC output voltage command values and the detected output voltage values and inputs them to the PI control processing function. Although upper and lower limits are placed within the PI control processing function on the absolute values of the integral term and output of the PI controller, a limiter for the output values is also in place outside the PI control processing function so that the minimum output value is 0.

As a description of gain calculation, Figure 8-2 is a simplified block diagram of voltage PI control.

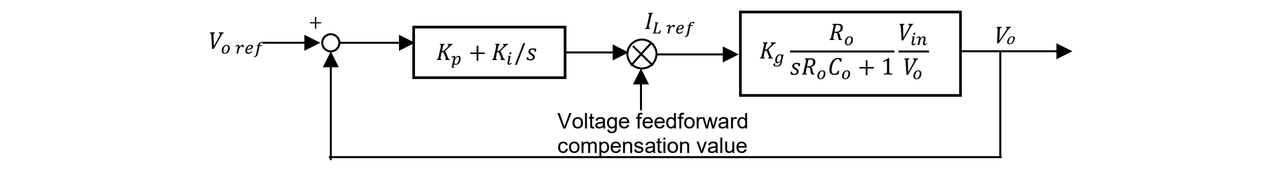


Figure 8-2 Block Diagram of Voltage PI Control

The output from the PI controller is the power to be output from the PFC circuits. This value is multiplied by the voltage feedforward compensation value to generate the current command value.

Here, the target of control can be expressed as $K_g \frac{R_o}{sR_oC_o+1}\frac{V_{in}}{V_o}$.

Let Kp/Ki = RoCo. The pole-zero pair can then be canceled out and the above control can be represented by the following first-order transfer characteristic.

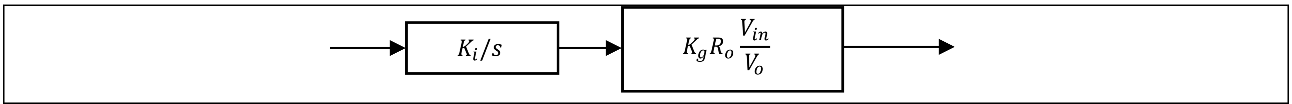


Figure 8-3 Open-Loop Characteristic of Voltage PI Control

By calculating the open-loop characteristic Go(s) and comparing its coefficient with that of the standard first-order transfer characteristic Go(s) = ω/s, the proportional gain Kp and integral gain Ki of voltage control are expressed as follows.

$$K_p = \omega_v \frac{R_oC_o}{K_gR_o(V_{in}/V_o)}$$

$$K_i = \omega_v \frac{1}{K_gR_o(V_{in}/V_o)}T_s$$

Here, ωv is the natural frequency of the voltage control system, Ro is the output resistance, Co is the capacitance, Kg is a constant, Vin is the input voltage, Vo is the output bus voltage, and Ts is the interval of control. When the backward Euler method is used for discretization, the integral term is multiplied by Ts, so the above Ki will also be multiplied by Ts.

8.4 Power Limitation

The maximum and minimum values of the PFC output power are calculated by multiplying the RMS of input voltage by the slope coefficient and the limiting value of the integral term of the voltage PI control module and the limiting value of the PI output are updated.

When the input voltage is 200 Vrms or higher, the maximum output power is limited to 1 kW. When the input is 100 Vrms to 200 Vrms, the power linearly increases. When the input is lower than 100 Vrms, the minimum power is limited to 500 W. In consideration of the loss in the PFC or inverter circuits, a margin coefficient is provided so that the power can be controlled within the range from ×1.0 to ×1.4. Therefore, the maximum and minimum power values and slope coefficient are expressed by the following equations.

Maximum power value = 200 × slope coefficient

Minimum power value = 100 × slope coefficient

Slope coefficient = (maximum power value – minimum power value)/(200 – 100)

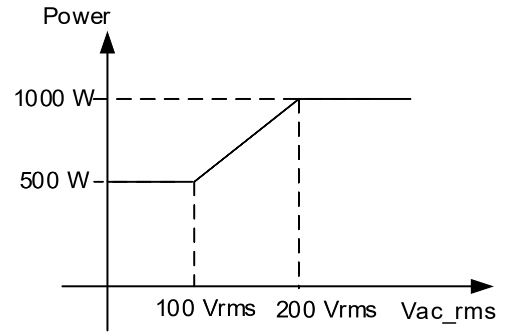


Figure 8-4 Power Limitation

8.5 Voltage Anti-Windup Control

The anti-windup control function prevents excess winding up of the integral term when the output from the voltage PI controller is limited by a limiter and the PFC output voltage does not follow the command value. This function places priority on the proportional term of the PI controller; when the PI output is saturated, this function forcibly sets the integrator to the value of the voltage difference between the PI output limiter value and the proportional term and controls the PI output so that the PFC output voltage follows the command value.

8.6 Voltage Feedforward Compensation

As the output from the voltage PI controller is used as the PFC output power, the voltage feedforward compensation coefficient shown below is calculated and the power (W) is converted to the AC current command value (A).

Voltage feedforward compensation coefficient
= (instantaneous value of input voltage)/(RMS of input voltage × RMS of input voltage)

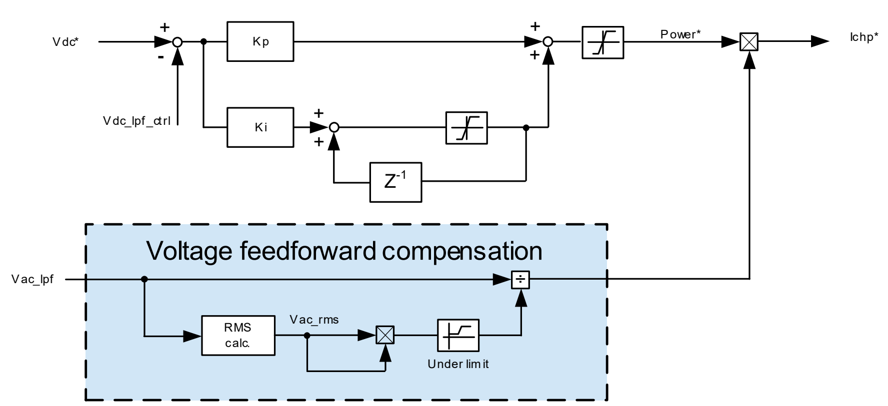


Figure 8-5 Block Diagram of Voltage Feedforward Compensation

8.7 Current Control Function

Current PI control in the current control function involves adjusting the power factor of the input AC voltage and input AC current by making the input current (reactor current) follow the current command value generated by the voltage PI control system. Figure 8-6 shows a block diagram of the current control system.

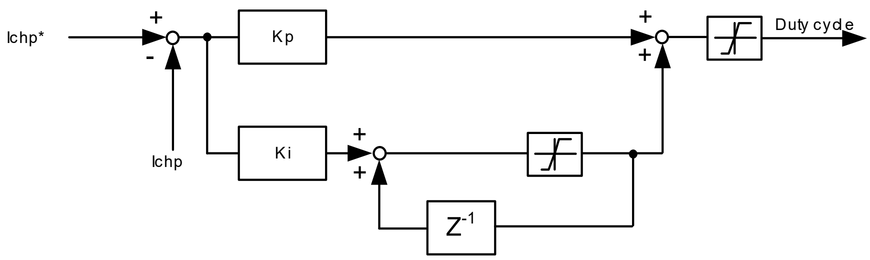


Figure 8-6 Block Diagram of Current PI Control

8.8 Current Anti-Windup Control

The anti-windup control module in the current control system prevents excess winding up of the integral term when the output from the current PI controller is limited by a limiter and the input current (reactor current) does not follow the command value. This function places priority on the proportional term of the PI controller; when the PI output is saturated, this function forcibly sets the integrator to the value of the duty difference between the PI output limiter value and the proportional term and controls the PI output so that the input current follows the command value.

8.9 Duty Feedforward Compensation

The duty cycle of the output voltage with respect to the input voltage in the steady state satisfies the following relationship.

Duty cycle = 1.0 – input voltage/output voltage

To improve the transient response to the change in the input voltage or output voltage, the above equation is used to compensate for the duty cycle in a feedforward manner. In addition, to reduce the transient change in current on startup of this duty feedforward compensation function, the soft start function is used to raise the compensation value to be in the range from 0.0 to 1.0 in the specified soft start time.

9. Software Specifications and Configuration

9.1 Software Specifications

The following shows the basic specifications of this software.

Table 9-1 Basic Specifications of this Software

Item	Description	
Motor control method	Position sensorless vector control	
Starting and stopping of motor control	Determined by input from the RMW	
PFC control method	Single-phase current continuous mode	
Starting and stopping of PFC control	Starting is automatic in response to the power supply being turned on and stopping is in response to a condition for a protective stop having been met.	
Rotor magnetic pole position detection	Sensorless algorithm: HFI (high-frequency pulse voltage injection) at standstill and low speed BEMF observer at medium-to-high speed	
Input voltage	Single-phase 100 to 240 VAC at 50 or 60 Hz	
Inverter bus voltage	390 VDC	
PWM carrier frequencies	Motor control	4 kHz, 250-µs cycle (interrupts in troughs)
	PFC control	32 kHz, 31.25-µs cycle
PWM mode	Sinusoidal modulation mode or space vector modulation mode	
Dead time	2.0 µs	
Control cycle	PFC	31.25 µs
	Current	250 µs
	Speed	500 µs
	System manager	1.0 ms
Speed command value management	CW: 0 to 4000 rpm CCW: 0 to –4000 rpm	
Natural frequency for each control system	Motor control system	Current control system: 150 Hz Speed control system: 3 Hz BEMF observer: 400Hz Position estimation PLL: 20 Hz
	PFC control system	Current control system: 1500 Hz Voltage control system: 12 Hz
Protective stop processing	The motor control signal outputs (six lines) will be deactivated when any of the following conditions is met. 1. The peak current value for any phase exceeds 17.25 A (checking is at 250-µs intervals). 2. The inverter bus voltage exceeds 450 V (checking is at 250-µs intervals). 3. The inverter bus voltage is lower than 100 V (checking is at 250-µs intervals). 4. The rotational velocity exceeds 4200 rpm (checking is at 250-µs intervals). 5. An abnormal temperature is detected in the IPM or PFC control system (checking is at 31.25-µs intervals). 6. The overcurrent detection signal (POE/POEG) is detected. 7. A step-skipping (stalled) state is detected if the step-skipping (stall) detection function is enabled (checking is at 250-µs intervals). 8. Any error related to PFC control listed on the following page is detected (checking is at 1.0-ms intervals). 9. During magnetic pole position estimation, the angle fluctuation (absolute value of the difference from the previous value) does not converge to within	

	1 degree 10 consecutive times within the period of 100 msec (checking is at the current control interval). 10. During polarity determination at magnetic pole position estimation, the absolute value of the PF value is less than 2.5 within the period of 100 msec (checking is at the current control interval). The PWM signal (one line) from the PFC control system will be deactivated when any of the following conditions is met. 11. The PFC output voltage exceeds 450 V (checking is at the PFC control interval). 12. The PFC input voltage exceeds 388 V (checking is at the PFC control interval). 13. The PFC current exceeds 19 A (checking is at the PFC control interval). 14. The PFC output is lower than 80 V (checking is at the PFC control interval). 15. The PFC current exceeds 49.09 A (indication by an external interrupt). 16. An abnormal temperature is detected in the IPM or PFC control system (checking is at the PFC control interval).

9.2 Overall Configuration of the Software

Figure 9-1 shows the overall configuration of the software.

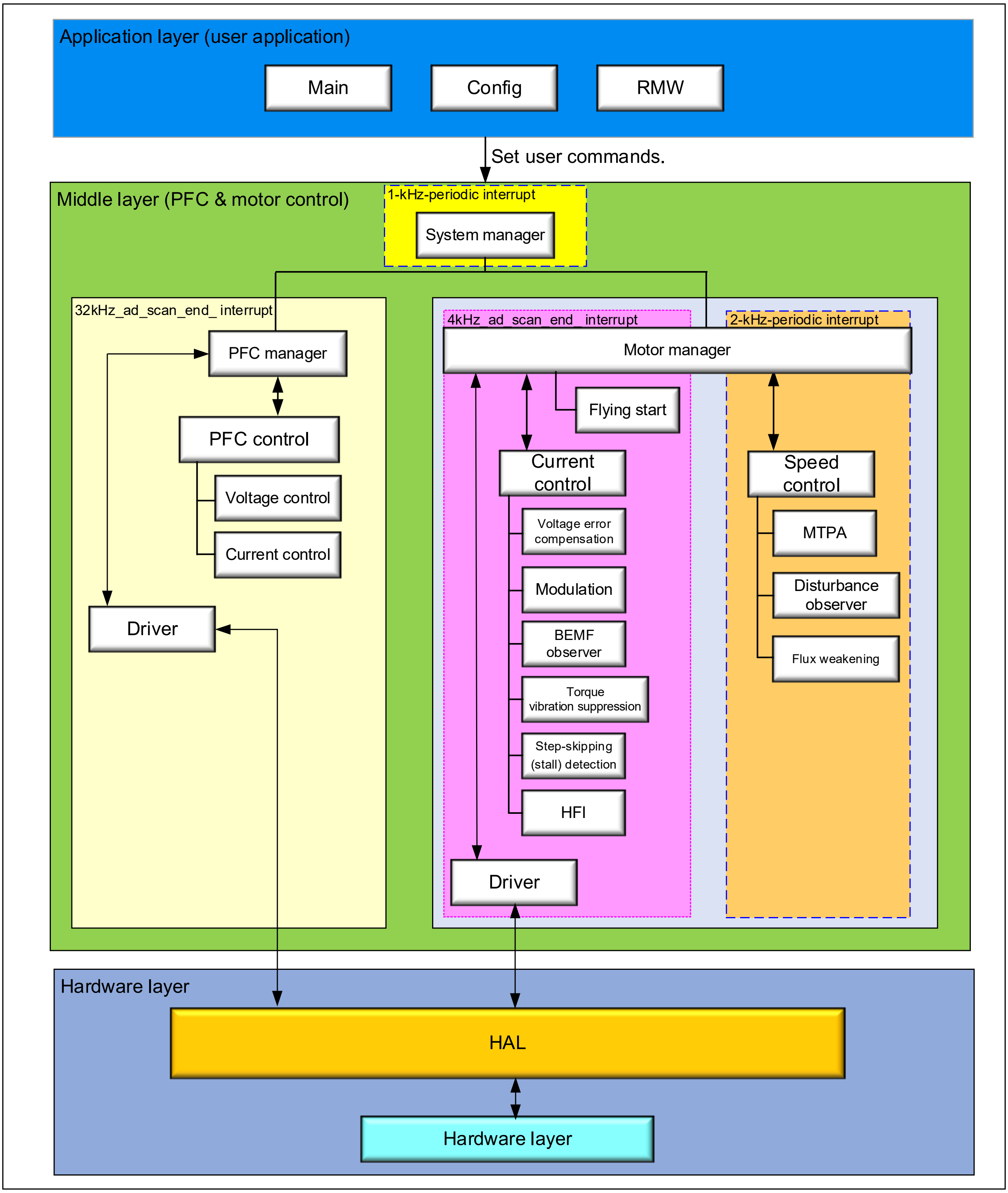


Figure 9-1 Overall Configuration of the Sample Program

9.3 Task Descriptions

For motor control, a task for speed control at 0.5-ms intervals and that for current control (4 kHz, 250 μs) are used. For PFC control, a task for PFC control interrupt processing (32 kHz, 31.25 μs) is used.

Table 9-2 Interrupts and Tasks Used

Task	Peripheral Module	Interval	Interrupt Function	Description
Motor control interrupt (for speed control)	agt0	500 μs	callback_agt_motor_speed_cyclic	
PFC control interrupt	adc0	31.25 μs	callback_gpt_adc_cyclic	These two tasks run upon an ADC conversion end interrupt. As they use a common interrupt function, the mask passed from the ADC stack in the FSP is checked to judge the task to be executed.
Motor control interrupt (for current control)	adc0	250 μs		
Periodic system manager interrupt	agt1	1 ms	callback_agt_system_manager_cyclic	
Task on a reset	—		Note: Executed in the state transition processing when recovering from an error.	
PFC overcurrent error interrupt	External IRQ		callback_irq2_pfc_error	
Motor output overcurrent error interrupt	External IRQ		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.
RMW operation	—		r_app_rmw_ui_mainloop	

9.4 Configuration of Folders and Files

Table 9-3 shows the configuration of the folders and files of the sample program.

Table 9-3 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		hal_entry.c	Startup routine module
src/application/main		mtr_main.c/h	Main module
src/application/motor_module	sensorless vector	r_motor_sensorless_vector_action.c	Definitions of action functions
		r_motor_sensorless_vector_api.c/h	Definitions of API functions for the motor manager module
		r_motor_sensorless_vector_flyingstart.lib/h	Flying start module
		r_motor_sensorless_vector_manager.c/h	Definitions of local functions for the manager module
		r_motor_sensorless_vector_protection.c/h	Definitions of functions for the protection facility
		r_motor_sensorless_vector_statemachine.c/h	Definitions of functions related to state transition
	current	r_motor_current_api.c/h	Definitions of API functions for the current control module
		r_motor_current.c/h	Definitions of local functions for the current control module
		r_motor_current_modulation.c/h	Definitions of functions for the modulation module
		r_motor_current_volt_err_comp.lib/h	Definitions of functions for the voltage error compensation module
		r_motor_current_bemf_observer.lib/h	Definitions of functions for the BEMF observer
		r_motor_current_pi_gain_calc.c	Definitions of functions for calculating the control gain of the current control module
		r_motor_current_stall_detection.lib/h	Step-skipping (stall) detection module
		r_motor_current_trq_vib_comp.lib/h	Torque vibration suppression module
		r_motor_current_lowspd_sensorless.lib/h	Definition of functions for the low-speed-range sensorless control module
	speed	r_motor_speed_api.c/h	Definitions of API functions for the speed control module
		r_motor_speed.c/h	Definitions of local functions for the speed control module
		r_motor_speed_fluxwkn.lib/h	Flux weakening control module
		r_motor_speed_mtpa.c/h	MTPA module
		r_motor_speed_extobserver.lib/h	Definition of functions for the disturbance torque/speed estimation observer

Folder	Subfolder	File	Remarks
		r_motor_speed_pi_gain_calc.c	Definitions of functions for calculating the control gain of the speed control module
	driver	r_motor_driver.c/h	Definitions of functions for the driver module
		r_motor_driver_fsp.c/h	Definitions of relay functions of the FSP for the driver module
	general	r_motor_filter.c/h	Definitions of general-purpose filter functions
		r_motor_pi_control.c/h	Definitions of PI control functions
		r_motor_common.h	Common definitions
	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
src/application/pfc_module	pfc_cfg	r_pfc_cfg.h	Definitions of the PFC-related configuration
	pfc_ctrl	r_pfc_ctrl.c/h r_pfc_ctrl_api.c/h	PFC control module
	pfc_driver	r_pfc_driver.c/h	PFC-related driver module
	pfc_general	r_pfc_common.h r_pfc_filter.c/h r_pfc_pi_control.c/h	Common modules related to PFC
	pfc_systask	r_pfc_manager.c/h r_pfc_manager_api.c/h r_pfc_manager_protection.c/h	PFC manager module
src/application/system_module	system_manager	r_system_manager.c/h r_system_manager_api.c/h	System manager module
src/application/user_interface	ics	r_mtr_ics.c/h	Definitions of interface functions for the RMW
		ICS2_RA6T2.o/h	Communications library for the RMW
		convert.bat	Batch file for MAP file generation
		ElfMapConverter.exe	MAP file generation tool
		ICS2_RA6T2_Built_in.o	Object file for use as built-in to the RMW

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^2 studio. The following table shows the configuration of the folders and files generated by the FSP.

Table 9-4 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.

9.5 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.

9.5.1 Functions

Table 9-5 lists the functions that are performed in the application layer.

Table 9-5 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.
LED processing	LED operation functions that the user can freely use are provided.

9.5.2 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 9-6 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 9-6.

Table 9-6 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_parameter_t	Structure for motor parameters
	f4_max_speed_rpm	Maximum speed (rpm) (mechanical angle)
	u1_ctrl_loop_mode	Control loop mode (speed control)
	f4_current_omega_hz	Natural frequency for the current control system (Hz)
	f4_current_zeta	Attenuation coefficient for the current control system
	f4_speed_omega_hz	Natural frequency for the speed control system (Hz)
	f4_speed_zeta	Attenuation coefficient for the speed control system
	f4_speed_lpf_hz	Speed LPF cut-off frequency (Hz)

Structure	Variable	Description
	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.
	u1_flag_fluxwkn_use	Enables or disables flux weakening control.
	u1_flag_extobserver_use	Enables or disables the disturbance torque/speed estimation observer.
	f4_extobs_omega	Natural frequency of the disturbance torque/speed estimation observer (Hz)
	u1_flag_mtpa_use	Enables or disables maximum torque per current control.
	u1_flag_flying_start_use	Enables or disables flying start.
	u1_flag_stall_detection_use	Enables or disables step-skipping (stall) detection.
	u1_flag_trq_vibration_comp_use	Enables or disables torque vibration suppression.
	f4_e_obs_omega_hz	Natural frequency for the inductive voltage estimation system (Hz)
	f4_e_obs_zeta	Attenuation coefficient for the inductive voltage estimation system
	f4_pll_est_omega_hz	Natural frequency for the position estimation system (Hz)
	f4_pll_est_zeta	Attenuation coefficient for the position estimation system
	f4_pll_estlow_omega_hz	Natural frequency for the position estimation system for HFI (Hz)
	f4_pll_estlow_zeta	Attenuation coefficient for the position estimation system for HFI
	f4_highspd_threshold	Switching speed from low to high speed
	f4_lowspd_threshold	Switching speed from high to low speed
	f4_id_hpf_time	Step-skipping (stall) detection: Constant of HPF for Id oscillation detection
	f4_iq_hpf_time	Step-skipping (stall) detection: Constant of HPF for Iq oscillation detection
	f4_threshold_level	Step-skipping (stall) detection: Threshold value (A)
	f4_threshold_time	Step-skipping (stall) detection: Monitoring time (s)
	f4_timelead	Torque vibration suppression: Phase adjustment value
	f4_tf_lpf_time	Torque vibration suppression: Extraction filter constant
	f4_output_gain	Torque vibration suppression: Output gain
	f4_input_weight2	Torque vibration suppression: Weight 2
	f4_input_weight1	Torque vibration suppression: Weight 1
	f4_input_weight0	Torque vibration suppression: Weight 0

Structure	Variable	Description
	f4_restart_speed	Flying start: Restart speed (rpm) (mechanical angle)
	f4_off_time	Flying start: Switched-off time (s)
	f4_over_time	Flying start: Limit time for being switched on (s)
	f4_active_brake_time	Active brake time (s)
	f4_on_current_th	Current threshold for switching on (A)

9.5.3 Macro Definitions

Table 9-7 lists the macros used in the RMW.

Table 9-7 List of Macros

File Name	Macro Name	Defined Value	Description
r_mtr_ics.h	ICS_DECIMATION	3	RMW watchpoint skip count
	ICS_BRR	19	RMW communications rate
	ICS_INT_MODE	1	RMW communications mode

Note: A macro that defines the channel used for communications via the RMW is provided in ICS2_RA6T2.h.

9.5.4 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 PFC 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).

9.6 System Manager

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

9.6.1 Functions

The following lists the functions of the system manager.

Table 9-8 List of Functions of the System Manger

Function	Description
System manager processing	Handles processing of command values and parameters supplied by the application layer and passes and receives callback functions. This facility also passes and receives detected values and states of processing to and from the motor manager and PFC manager.
Motor manager processing	Acquires and specifies command values for speed control.
PFC manager processing	Performs PFC control.

9.6.2 Module Configuration Diagram

Figure 9-2 shows the module configuration.

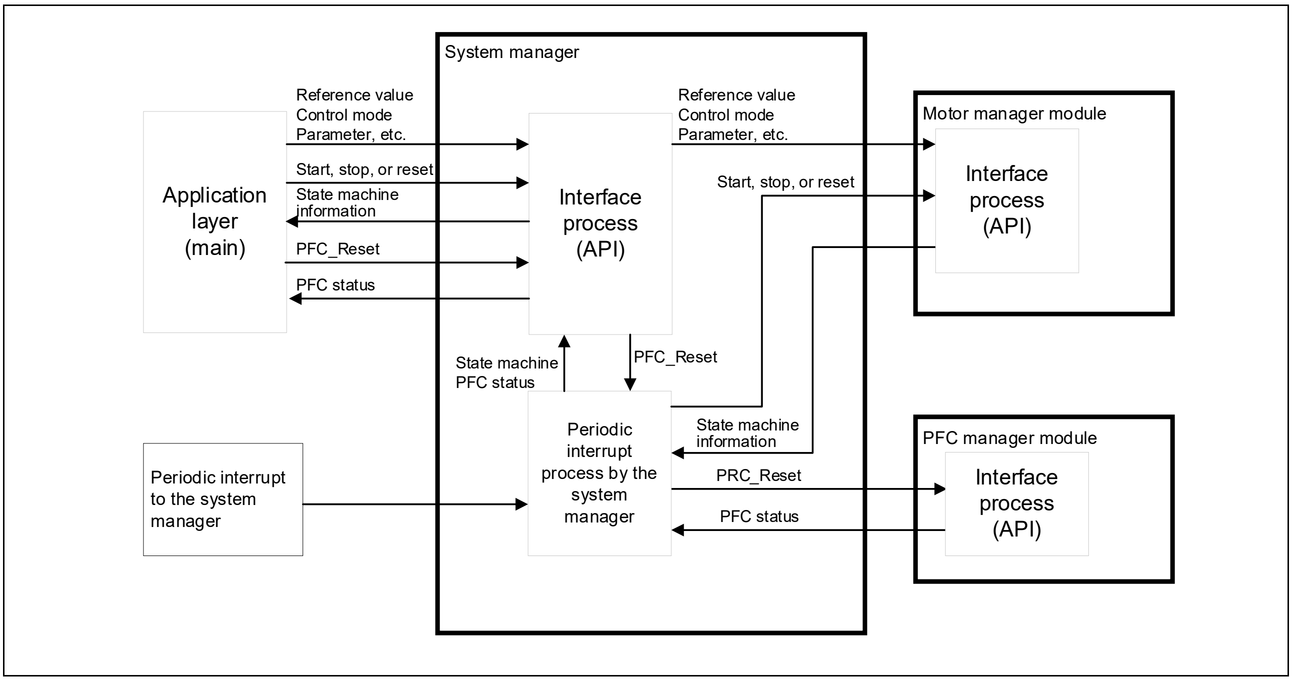


Figure 9-2 Module Configuration of the System Manager

9.7 Motor Manager

The motor manager (r_motor_sensorless_vector_manager) uses specific control modules that the motor control module includes 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.

9.7.1 Functions

Table 9-9 lists the functions of the motor manager module. Table 9-10 and Table 9-11 list the functions of the motor control modules.

Table 9-9 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.
Control method management	Acquires and sets the states of speed control and current control.
Speed and position information acquisition	Acquires the speed and position information from the speed control module and current control module.
Control module command value setting	Selects the command values to be input to the current control module and speed control module based on the control states.
Flying start	Starts the motor when it is already rotating.
Interrupt processing	Assigns processing to appropriate modules in response to callback functions (interrupts) set through the FSP.

Table 9-10 List of Functions of the Speed Control Module

Function	Description
Speed control	Calculates and outputs a current command value so that the speed follows the speed command value.
Speed command setting	Sets a speed command value in the speed control module.
Flux weakening control	Controls the d-axis current so that the motor is capable of operating above its rated rotational velocity.
Disturbance torque/speed estimation observer	Estimates the disturbance at low speeds based on the speed and current to control and suppresses the disturbance.
Maximum torque per current control	Controls the d-axis current so that the maximum torque is output according to the load conditions.

Table 9-11 List of Functions of the Current Control Module

Function	Description
Current control	Calculates and sets PWM output values so that the current follows the current command value.
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.

Decoupling control	Calculates interference cancellation to prevent interference between the d and q axes.
Sample delay compensation	Compensates for the delay in sampling by the current-control cycle in the generation of three-phase voltage command values.
Torque vibration suppression	Detects and suppresses the vibration of the load in synchronization with rotation through one cycle of mechanical angle.
Step-skipping (stall) detection	Detects a displacement of the magnetic pole position as estimated in the control system from the actual position in the motor and stops the motor.
High-frequency pulse injection	Applies high-frequency pulses to the output voltage to estimate the position and speed when the motor is standstill or running at low speeds.
BEMF observer	Uses the BEMF observer to estimate the position and speed when the motor is running at medium to high speeds.

9.7.2 Module Configuration Diagram

Figure 9-3 shows the module configuration.

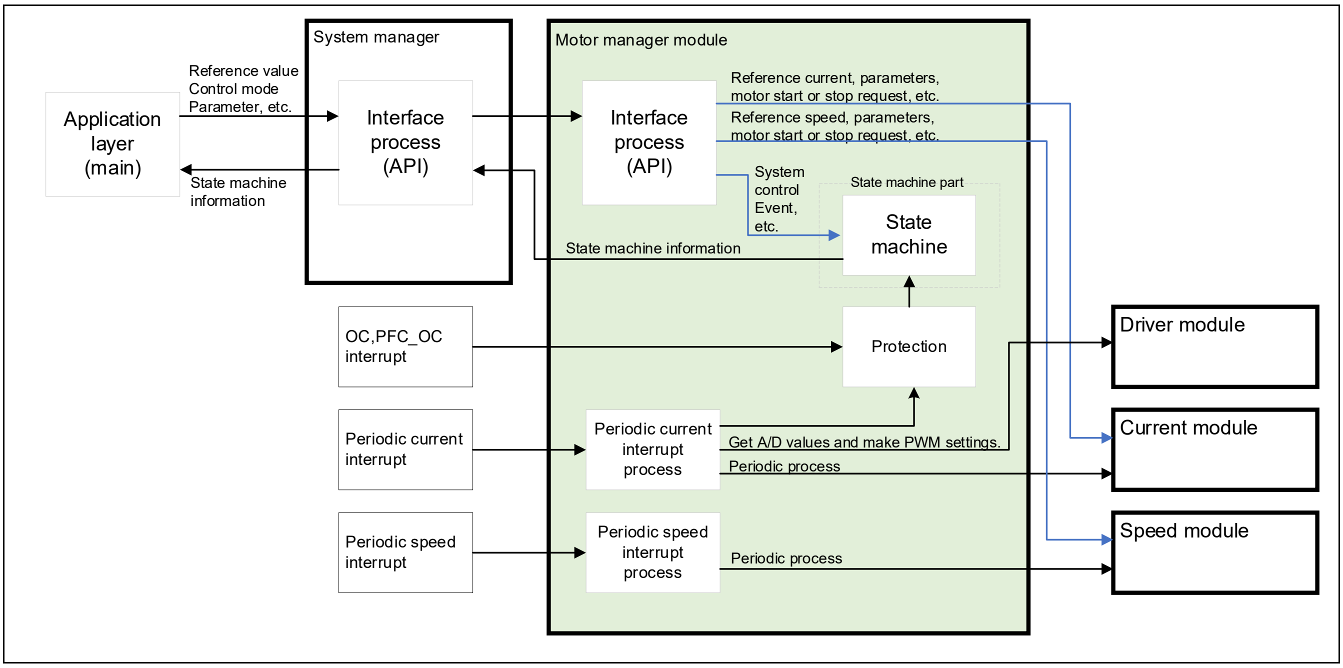


Figure 9-3 Module Configuration of the Motor Manager

9.7.3 Mode Management

Figure 9-4 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.

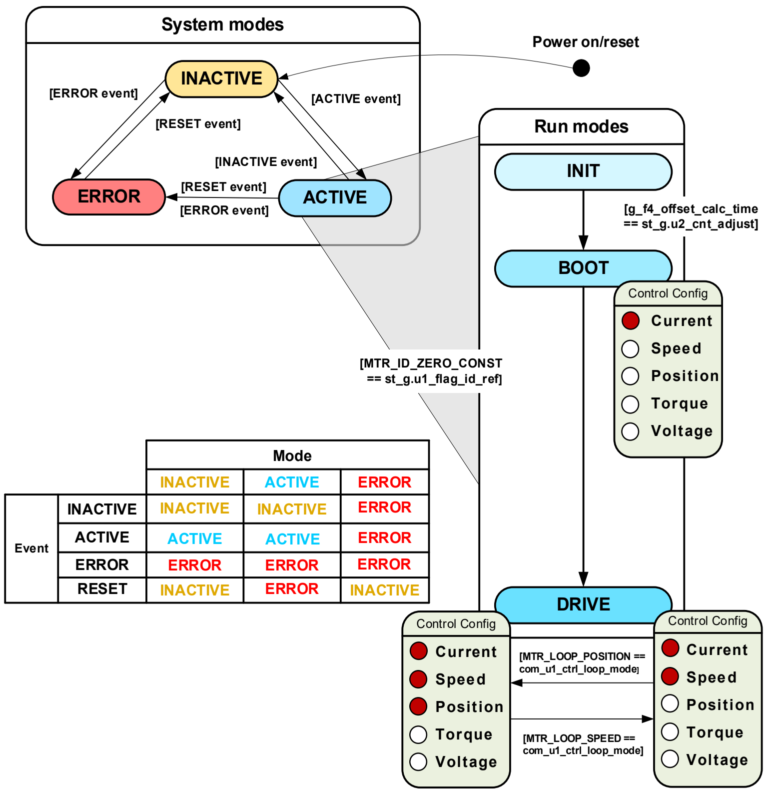


Figure 9-4 State Transition Diagram of the Motor Control Software

(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 9-4.

(3) Events

The matrix table in Figure 9-4 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 9-12 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

9.7.4 Sequence Descriptions

This sample program has two types of state transition. One is referred to as the mode (ACTIVE, INACTIVE, or ERROR) and involves the management of the corresponding transitions. The other is called the sequence and involves managing the state of operation for sensorless control, with the main point of control being switching between two methods according to the speed relative to the specified speed. For more information on the mode, see section 9.7.3. This section describes the latter, that is, the sequence, which involves managing the state of operation. See the following diagram.

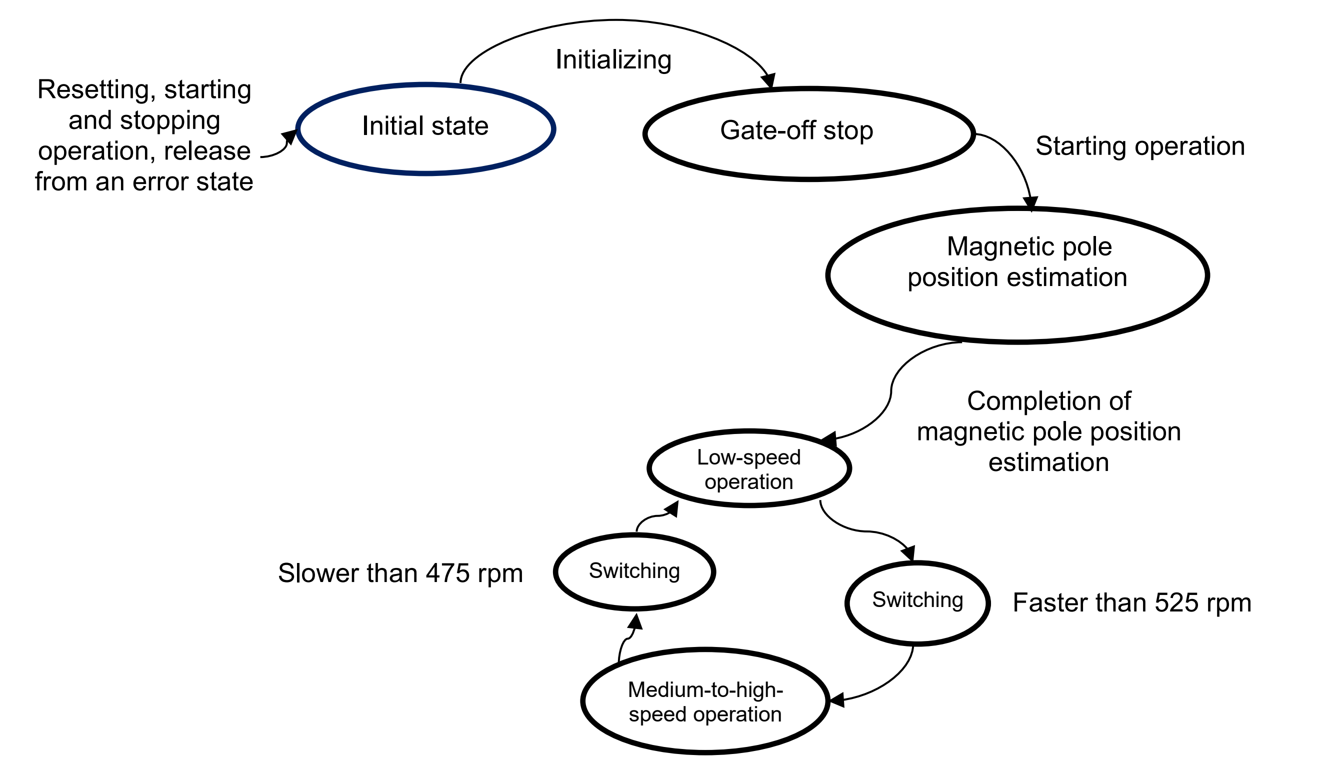


Figure 9-5 State Transition Diagram of the Operation Sequence

Table 9-13 Operation Sequence States and their Descriptions

State	Description
Initial state	Initialized state. The sequence enters this state after the motor has started or stopped operating, or control has been released from an error state following a reset.
Gate-off stop	This is the state in which the power supply to the CPU board is turned on. When the power supply to the inverter is turned off, the motor will not be turned on. Attempting to turn it on leads to an error due to undervoltage protection.
Magnetic pole position estimation	A current is applied to the motor to estimate the magnetic pole position. The successful completion of estimation switches the sequence to the low-speed operation mode and the motor starts rotating at a specified speed. If the estimation fails, the protection function disables output and the sequence switches to the initial state through the processing for release from the error state.
Low-speed operation	This is the state in which the motor is running from 0 rpm (current is flowing in the motor but the motor is stopped) to approximately 500 rpm (adjustable). Sensorless vector control is performed by using the low-speed sensorless algorithm.
Switching	This is the state in which the sensorless algorithm used is switched from the one used in the low-speed operation to the one used in the medium-to-high-speed operation. During acceleration, the data is transferred to the algorithm for medium-to-high-speed operation. As soon as the data transfer is completed, the sequence automatically switches to the medium-to-high-speed operation state.

	During deceleration, the data is transferred to the algorithm for low-speed operation. As soon as the data transfer is completed, the sequence automatically switches to the low-speed operation state.
Medium-to-high-speed operation	This is the state in which the motor is running from approximately 500 rpm (adjustable) to the motor's rated speed. The motor is controlled by sensorless vector control using the BEMF observer.

9.7.5 Startup Sequence

The motor manager module controls the motor by changing the flag settings that manage the speed command value according to the run mode. Also, by changing these command values appropriately, the motor manager module creates a startup sequence to start the motor. Figure 9-6 shows the behavior in the startup sequence.

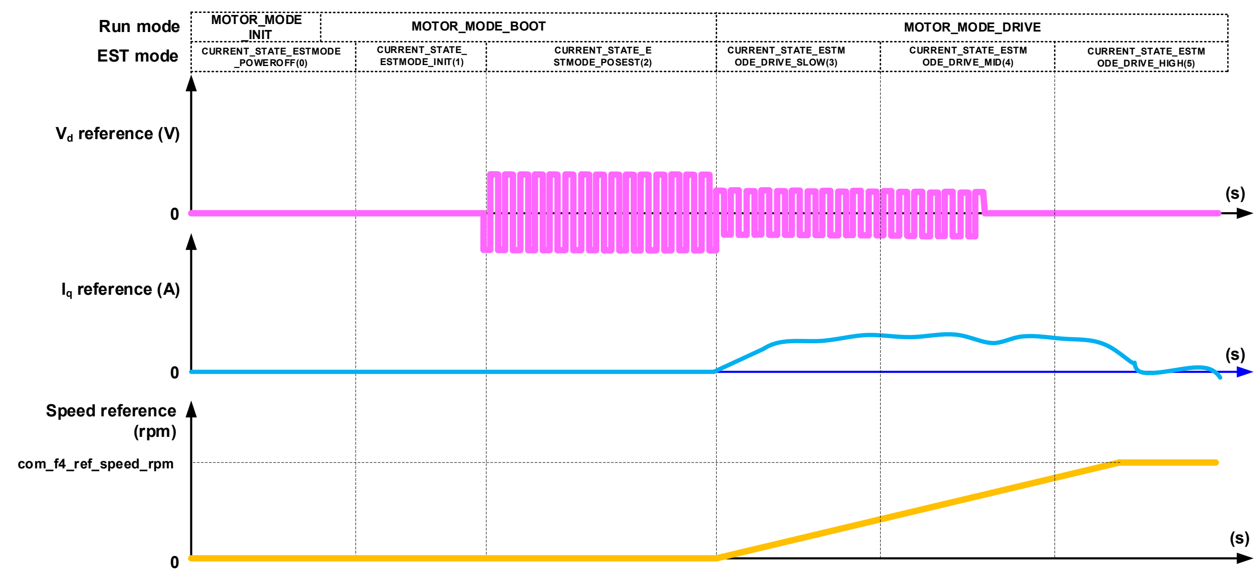


Figure 9-6 Behavior in the Startup Sequence

9.7.6 Protection Function

This control program has the following error states and implements an emergency stop function in each error state. For details about the values that can be specified for the system protection function, see Table 9-14.

- Overcurrent error

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

The PWM output pins are placed in the high-impedance state in response to an emergency stop signal (due to overcurrent detection) from the hardware. The U-, V-, and W-phase currents are monitored at the overcurrent monitoring 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).

- Overvoltage error

The inverter bus voltage is monitored at the overvoltage monitoring 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 inverter bus voltage is monitored at the low-voltage monitoring 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 rotation speed monitoring interval. If the rotation speed exceeds the speed limit value, the motor is brought to an emergency stop.

- Step-skipping (stall) detection error

If the symptom for detecting step-skipping (stall) is detected during sensorless vector control, the motor is brought to an emergency stop. Use of the step-skipping (stall) detection function requires that it be explicitly enabled.

- Magnetic pole position estimation error

This process estimates the magnetic pole position of the IPM motor at startup. If the estimation completion condition is not satisfied, it pauses and stops the operation.

- Polarity determination error

This process determines whether the magnetic pole position of the IPM motor faces the N pole or S pole at startup. If the estimation completion condition is not satisfied, it pauses and stops the operation.

Table 9-14 Operating Conditions and Settings for the System Protection Functions

Overcurrent error	Overcurrent limit value (A)	17.25
	Monitoring interval (μs)	Current control interval*
Overvoltage error	Overvoltage limit value (V)	450
	Monitoring interval (μs)	Current control interval*
Low-voltage error	Low-voltage limit value (V)	100
	Monitoring interval (μs)	Current control interval*
Rotation speed error	Speed limit value (rpm)	4200
	Monitoring interval (μs)	Current control interval*
Step-skipping (stall) detection error	Condition of occurrence	Detection of step-skipping (stall)
	Monitoring interval (μs)	Current control interval*
Magnetic pole position estimation error	Estimation completion condition	During initial magnetic pole position estimation, the angle fluctuation (absolute value of the difference from

		the previous value) did not converge to within 1 degree 10 consecutive times within the period of 100 msec.
	Monitoring interval (μs)	Current control interval*
Polarity determination error	Estimation completion condition	During initial magnetic pole position estimation, the absolute value of the PF value was no greater than 2.5 within the period of 100 msec.
	Monitoring interval (μs)	Current control interval*

Note * See Table 9-1, Basic Specifications of this Software.

9.7.7 API

Table 9-15 lists the API functions of the motor manager module.

Table 9-15 List of API Functions

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

Table 9-16 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 of 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_BEMFObserverParameterUpdate	Updates the control parameters for the BEMF observer.
R_MOTOR_CURRENT_UpdateAngleNSpole	Updates the rotor angle based on the result of the polarity determination process at startup. Used immediately after the completion of the magnetic pole position estimation process at startup.

Table 9-17 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 module in the reset state.
R_MOTOR_SPEED_Reset	Initializes the module.
R_MOTOR_SPEED_Run	Activates the 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_ExtObserverParameterUpdate	Updates the control parameters for the disturbance torque/speed estimation observer.

9.7.8 Structure and Variable Information

Table 9-18 lists the structures and their member variables for the motor manager module. In this module, the structure for the motor manager module (g_st_sensorless_vector) is defined by the API function for securing an instance of the module. Table 9-19 lists the structures and their member variables that are used in the current control module. Table 9-20 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 9-18 List of Structures and Variables for the Motor Manager Module

Structure	Variable	Description
st_sensorless_vector_control_t Structure for the motor manager module	u1_state_speed_ref	State of the speed command value
	u1_state_estmode	State of the magnetic pole position estimation
	u2_estmode_state_chg_cnt	Counter for the magnetic pole position estimation state
	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	Inverter bus voltage (V)
	f4_iu_ad	U-phase current (A)
	f4_iv_ad	V-phase current (A)
	f4_iw_ad	W-phase current (A)
	f4_overcurrent_limit	Overcurrent limit value (A)
	f4_overvoltage_limit	Overvoltage limit value (V)
	f4_undervoltage_limit	Low-voltage limit value (V)
	f4_overspeed_limit_rad	Overspeed limit value (rad/s)
	u2_est_timeout_cnt	Timeout counter for magnetic pole position estimation
	f4_ctrl_period	Current loop control interval (s)
	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 for motor parameters
	*p_st_driver	Structure for the driver module
	st_current_control_t	Structure for the current control module
	st_speed_control_t	Structure for the speed control module
st_sensorless_vector_cfg_t Structure for setting the motor manager module control parameters	f4_nominal_current_rms	Current limit value (A)
	f4_overspeed_limit_rpm	Speed limit value (rpm) (mechanical angle)
	st_motor	Structure for motor parameters

Table 9-19 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_ref	d-axis current command value (A)
	f4_iq_ref	q-axis current command value (A)
	f4_id_ad	d-axis current value (A)
	f4_iq_ad	q-axis current value (A)
	f4_lim_iq	q-axis current limit value (A)
	f4_offset_iu	U-phase offset current value (A)
	f4_offset_iw	W-phase offset current value (A)
	f4_sum_iu_ad	U-phase total current value (A)
	f4_sum_iw_ad	W-phase total current value (A)
	f4_vdc_ad	Inverter bus voltage value (V)
	f4_iu_ad	U-phase current value (A)
	f4_iv_ad	V-phase current value (A)
	f4_iw_ad	W-phase current value (A)
	f4_modu	U-phase duty cycle
	f4_modv	V-phase duty cycle
	f4_modw	W-phase duty cycle
	f4_speed_rad	Speed (rad/s)
	f4_ref_id_ctrl	d-axis current command value (A)
	f4_ref_iq_ctrl	q-axis current command value (A)
	f4_va_max	Maximum voltage on the d and q axes (V)
	f4_ed	Estimated d-axis inductive voltage value
	f4_eq	Estimated q-axis inductive voltage value

Structure	Variable	Description
	st_mod_t	Structure for the modulation module
	st_volt_comp_t	Structure for the voltage error compensation module
	st_bemf_observer_t	Structure for the BEMF observer
	st_pll_est_t	Structure for position and speed estimation (BEMF observer)
	st_pll_est_low_t	Structure for position and speed estimation (HFI)
	st_pi_ctrl_t	Structure for d-axis PI control
	st_pi_ctrl_t	Structure for q-axis PI control
	st_rotor_angle_t	Structure for rotor information
	st_rotor_angle_phasecomp	Structure for rotor information (lead compensation)
	st_motor_parameter_t	Structure for motor parameters
	st_lowspd	Structure for HFI function
st_current_cfg_t Structure for setting the control parameters for the current control module	u2_offset_calc_time	Offset calculation time setting
	f4_ctrl_period	Control interval (s)
	f4_current_omega_hz	Natural frequency for the current control system (Hz)
	f4_current_zeta	Attenuation coefficient for the current control system
	u1_flag_volt_err_comp_use	Enables or disables voltage error compensation.
	st_motor	Structure for motor parameters
st_current_output_t Structure for the current control module output	u1_flag_offset_calc	Current offset flag
	f4_modu	U-phase duty cycle
	f4_modv	V-phase duty cycle
	f4_modw	W-phase duty cycle
	f4_neutral_duty	Duty cycle in offset measurement
	f4_va_max	Maximum voltage on the d and q axes (V)
	f4_ref_id_ctrl	d-axis current command value
	f4_speed_rad	Estimated speed (rad/s)
	f4_ed	Estimated d-axis inductive voltage value
	f4_eq	Estimated q-axis inductive voltage value
st_current_input_t Structure for the current control module input	f4_rotor_angle_rad	Rotor angle (rad)
	f4_iu_ad	U-phase current value (A)
	f4_iv_ad	V-phase current value (A)
	f4_iw_ad	W-phase current value (A)
	f4_vdc_ad	Inverter bus voltage value (V)

Structure	Variable	Description
	f4_speed_rad	Speed (rad/s)
	f4_id_ref	d-axis current command value (A)
	f4_iq_ref	q-axis current command value (A)
st_bemf_observer_cfg_t	f4_e_obs_omega_hz	Natural frequency for the inductive voltage estimation system (Hz)
Structure for the BEMF observer module input	f4_e_obs_zeta	Attenuation coefficient for the inductive voltage estimation system
	f4_pll_est_omega_hz	Natural frequency for the position estimation system (Hz)

Table 9-20 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 9.7.9, Macro Definitions.
	u1_flag_extobserver_use	Flag for indicating whether to use disturbance torque/speed estimation observer control
	u1_flag_mtpa_use	Flag for indicating whether to use maximum torque per current control
	f4_speed_ctrl_period	Speed loop control interval (s)
	f4_ref_speed_rad_ctrl	Speed command value for control (rad/s)
	f4_ref_speed_rad	Speed command value output by the position control module during position control (rad/s)
	f4_ref_speed_rad_manual	Speed command value set by the user during speed control (rad/s)
	f4_speed_rad_ctrl	Speed calculated by the speed control module (rad/s)
	f4_speed_rad	Speed 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_va_max	Maximum voltage on the d and q axes (V)
	f4_id_ad	d-axis current value (A)
	f4_iq_ad	q-axis current value (A)
	f4_torque_current	Torque current (A)
	st_motor_parameter_t	Structure for motor constants
	st_pi_ctrl_t	Structure for PI control
	st_1st_order_lpf_t	Structure for LPF
st_speed_config_t	f4_max_speed_rpm	Maximum speed (rpm) (mechanical angle)
	f4_speed_ctrl_period	Speed control interval (s)

Structure	Variable	Description
Structure for setting the control parameters for the speed control module	f4_speed_rate_limit_rpm	Speed variation limit value (rpm) (mechanical angle)
	f4_speed_omega_hz	Natural frequency for the speed control system (Hz)
	f4_speed_zeta	Attenuation coefficient for the speed control system
	f4_speed_lpf_hz	LPF for speed control (Hz)
	st_motor_param_t	Structure for motor constants
st_speed_input_t Structure for speed control module input	u1_state_speed_ref	Speed command state
	f4_speed_rad	Speed to be input (rad/s)
	f4_va_max	Maximum voltage on the d and q axes (V)
st_speed_output_t Structure for speed control module output	f4_id_ref	d-axis current command value (A)
	f4_iq_ref	q-axis current command value (A)
	f4_ref_speed_rad_ctrl	Speed used for PI control (rad/s)
	f4_speed_rad_lpf	Speed after LPF processing (rad/s)

9.7.9 Macro Definitions

Table 9-21 lists the macros for the motor manager module.

Table 9-21 List of Macros

File Name	Macro Name	Defined Value	Description
r_motor_sensorless_vector_api.h	MOTOR_LOOP_POSITION	0	Position control mode Note: Not supported in this sample program
	MOTOR_LOOP_SPEED	1	Speed control mode
	MOTOR_SENSORLESS_VECTOR_ERROR_NONE	0x0000	Error state There is no error.
	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_CURRENT_HW	0x0001	Error state A hardware overcurrent error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_VOLTAGE	0x0002	Error state An overvoltage error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_SPEED	0x0004	Error state An overspeed error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_LOW_VOLTAGE	0x0080	Error state A low-voltage error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_CURRENT_SW	0x0100	Error state A software overcurrent error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_STALL_DETECTED	0x0200	Error state Step-skipping (stall) has been detected.
	MOTOR_SENSORLESS_VECTOR_ERROR_PFC	0x0400	Error state PFC error
	MOTOR_SENSORLESS_VECTOR_ERROR_FAIL_POLES	0x0800	Error state A polarity determination error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_FAIL_POSITION	0x1000	Error state A magnetic pole position estimation error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_UNKNOWN	0xffff	Error state An error whose error code is unknown has occurred.
r_motor_sensorless_vector_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_sensorless_vector_api.h	MOTOR_CTRL_TYPE_POS	0	Macro for switching the control method Position control mode
	MOTOR_CTRL_TYPE_SPEED	1	Macro for switching the control method Speed control mode

9.8 PFC Manager

9.8.1 Functions

The PFC manager (r_pfc_manager) is activated by A/D conversion end interrupts, which are generated at a frequency of 32 kHz. It drives a relay, detects errors, controls the PFC output voltage, and adjusts the power factors of the input voltage and input current according to the A/D-converted values of the input voltage, PFC output voltage, and PFC reactor current. If an error occurs during PFC, it is reset through the system manager.

9.8.2 Module Configuration Diagram

Figure 9-7 shows the functional blocks of the PFC manager.

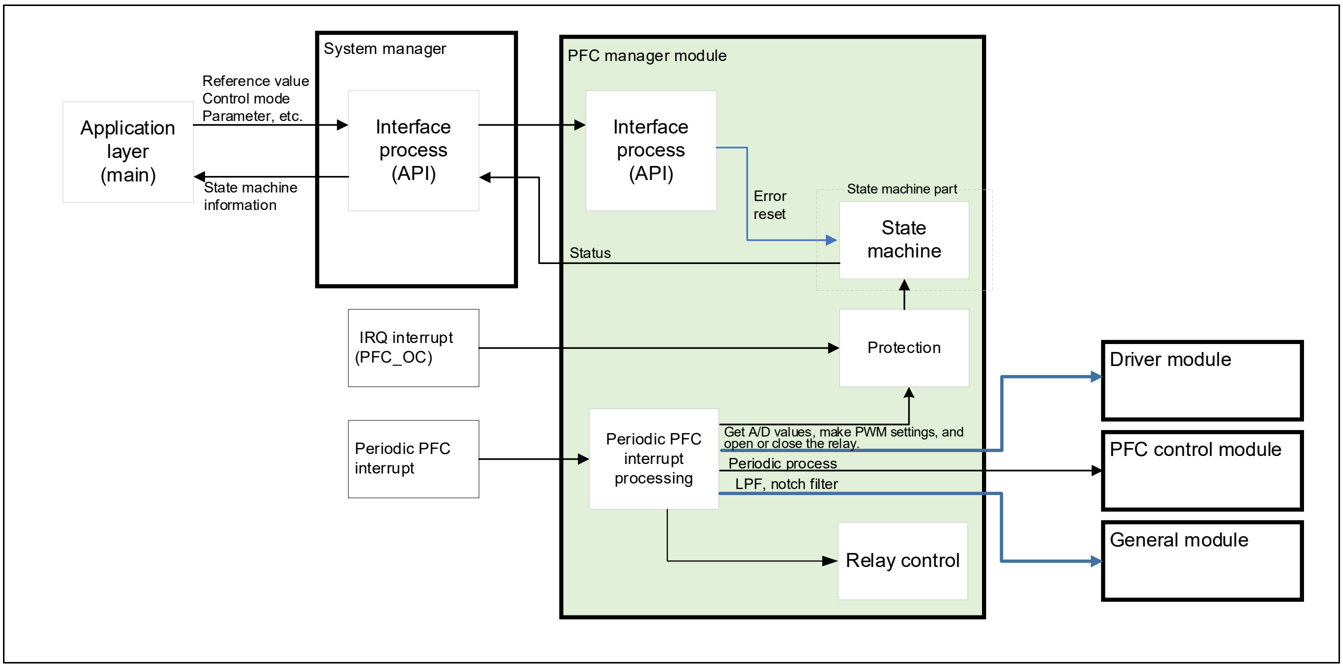


Figure 9-7 Functional Blocks of the PFC Manager

9.8.3 Sequence Descriptions

The PFC manager controls two sequences: the startup sequence and stop sequence.

• Startup sequence

When the inverter bus voltage reaches the specified level and the relay is turned on while none of the errors listed in Table 9-22, List of Target Errors for Protective Stopping, has been generated, the PFC manager waits for a specified period (100 ms only when the relay is shifted from the off state to the on state) and then enables PFC control. After PFC control is enabled, PFC activation is completed when the output voltage command value satisfies the startup conditions. Figure 9-8 is a state transition diagram that includes the startup sequence.

• Stop sequence

If any of the errors listed in Table 9-22, List of Target Errors for Protective Stopping, occurs, the PFC manager stops PFC control and changes the PWM output pins to operate as GPIO output pins so that the low level (the inactive level) is output from the pins.

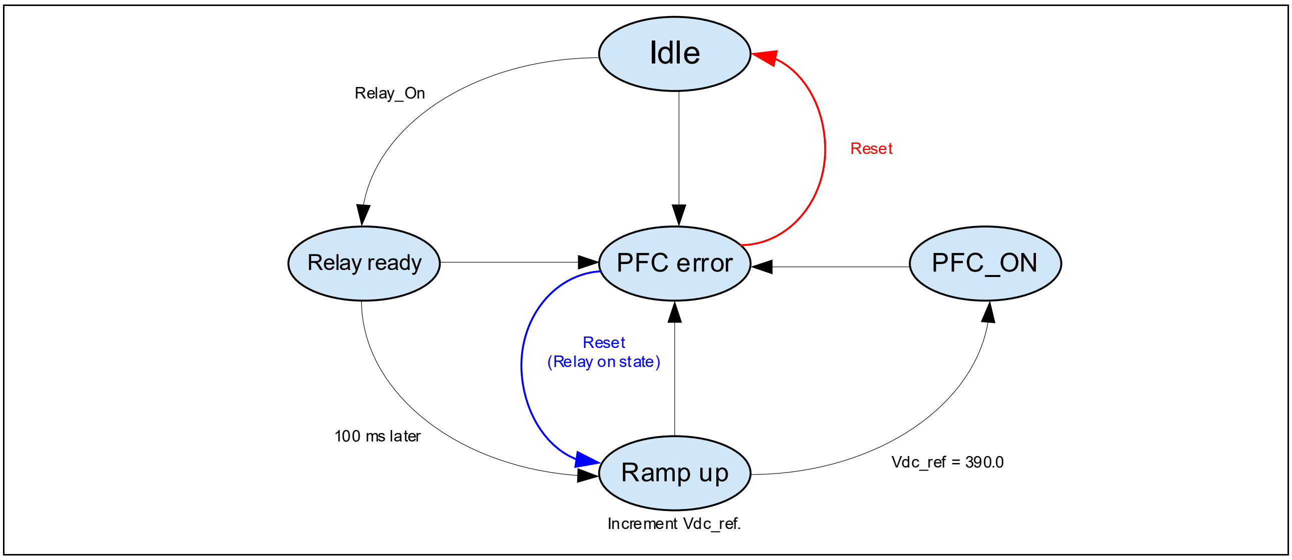


Figure 9-8 State Transition Diagram of PFC

9.8.4 Protection Function

This function judges and processes voltage, current, and temperature errors related to the PFC hardware. If any of the states listed in Table 9-22 is detected, the PWM gate signals are cut off and processing to stop driving of the inverter proceeds in the PFC manager.

The DC bus overvoltage, DC bus low voltage, PFC-related defect 2 (overvoltage input), PFC-related defect 3 (OC_PFC_SW), and PFC temperature error states are periodically monitored in the periodic PFC manager interrupt processing. PFC-related defect 1 (OC_PFC_HW) is monitored by hardware and the hardware overcurrent interrupt processing is executed in response to the associated IRQ interrupt.

Table 9-22 List of Target Errors for Protective Stopping

Error	Pin	Detection Interval	Detection Level	Unit	Protective Operation
DC bus overvoltage	AN006	PFC carrier cycle	450	V	The motor inverter and PFC gate signals are cut off.
DC bus low voltage	AN006	PFC carrier cycle	80	V	
PFC-related defect 1 (OC_PFC_HW)	P001/IRQ 2	IRQ interrupt	49.09	A	
PFC-related defect 2 (overvoltage input)	AN028	PFC carrier cycle	388	V	
PFC-related defect 3 (OC_PFC_SW)	AN027	PFC carrier cycle	19	A	
PFC temperature error	PD07	PFC carrier cycle	Low *	-	

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

9.8.5 API

Table 9-23 lists the API functions of the PFC manager module.

Table 9-23 List of API Functions

API Function	Description
R_PFC_MANAGER_Open	Generates instances of this module and the modules it is to use.
R_PFC_MANAGER_Close	Closes this module.

API Function	Description
R_PFC_MANAGER_Reset	Places this module in the reset state.
R_PFC_MANAGER_ErrorCancel	Releases the system from the error state.
R_PFC_MANAGER_StatusGet	Acquires the internal state.
R_PFC_MANAGER_Main	Executes the interrupt processing required for processing by the PFC manager.
R_PFC_MANAGER_OverCurrentInterrupt	Executes the interrupt processing in response to an overcurrent error.

9.8.6 Structure and Variable Information

Table 9-24 List of Structures and Variables for the PFC Manager Module

Structure	Variable	Description
st_pfc_manager_t Structure for the PFC manager module	u1_pfc_ctrl_enable_flg	Flag for enabling PFC control
	u1_relay_ctrl_on_flg	Flag for enabling relay control
	u1_error_cancel_flg	Flag for cancelling errors
	u2_error_status	Error state
	u2_run_mode	Run mode
	f4_vac_ad	AC voltage (V)
	f4_vdc_ad	Inverter bus voltage (V)
	f4_ichp_ad	PFC current value (A)
	f4_vac_ad_lpf	AC voltage after LPF processing (V)
	f4_vdc_ad_lpf	Bus voltage after LPF processing (V)
	f4_ref_vdc_ctrl	Inverter bus voltage command value (V)
	f4_vdc_up_step	Step-up value for the bus voltage (V)
	f4_target_vdc	Target value of the bus voltage (V)
	f4_ac_overvoltage_limit	AC voltage: Overvoltage limit (V)
	f4_bus_overvoltage_limit	Bus voltage: Overvoltage limit (V)
	f4_bus_undervoltage_limit	Bus voltage: Low-voltage limit (V)
	f4_overcurrent_limit	Overcurrent limit (A)
	u1_overheat_detect_level	Pin level stored when an overheat error was detected
	st_vac_ad_lpf	Structure of LPF parameters for the AC voltage
	st_vdc_ad_lpf	Structure of LPF parameters for the bus voltage
	st_vdc_notch_fil	Structure of notch filter parameters for the inverter bus voltage
	st_pfc_ctrl_output	Structure of parameters for PFC control output
	st_ac_fil	Structure of AC voltage RMS data
	st_relay_ctrl	Structure of relay control data
	p_st_pfc_driver	Structure for the PFC driver
	p_st_pfc_ctrl	Structure for PFC control

9.8.7 Macro Definitions

Table 9-25 lists the macros used by the PFC manager.

Table 9-25 List of Macros

File Name	Macro Name	Defined Value	Description
r_pfc_manager.h	PFC_MODE_IDLE	0x00	Wait mode
	PFC_MODE_RELAY_READY	0x01	The relay is ready.
	PFC_MODE_RAMP_UP	0x02	Ramping up is in progress.
	PFC_MODE_PFC_ON	0x03	PFC is active.
	PFC_MODE_PFC_ERROR	0x04	PFC error
r_pfc_manager_api.h	PFC_MANAGER_ERROR_NONE	0x0000	No error
	PFC_MANAGER_ERROR_AC_OVER_VOLTAGE	0x0001	Vac overvoltage error
	PFC_MANAGER_ERROR_BUS_OVER_VOLTAGE	0x0002	Vdc overvoltage error
	PFC_MANAGER_ERROR_BUS_LOW_VOLTAGE	0x0004	Vdc low-voltage error
	PFC_MANAGER_ERROR_OVER_CURRENT_SW	0x0008	Software overcurrent error
	PFC_MANAGER_ERROR_OVER_CURRENT_HW	0x0010	Hardware overcurrent error
	PFC_MANAGER_ERROR_OVER_HEATING	0x0020	Overheat error
	PFC_MANAGER_ERROR_UNKNOWN	0xffff	Undefined error

9.9 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.

9.9.1 Functions

Table 9-26 lists the functions of the driver module.

Table 9-26 List of Functions of the Driver Module

Function	Description
Acquisition of A/D-converted values	Acquires A/D values such as the phase currents and inverter 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.

9.9.2 Module Configuration Diagram

Figure 9-9 shows the configuration of the driver module.

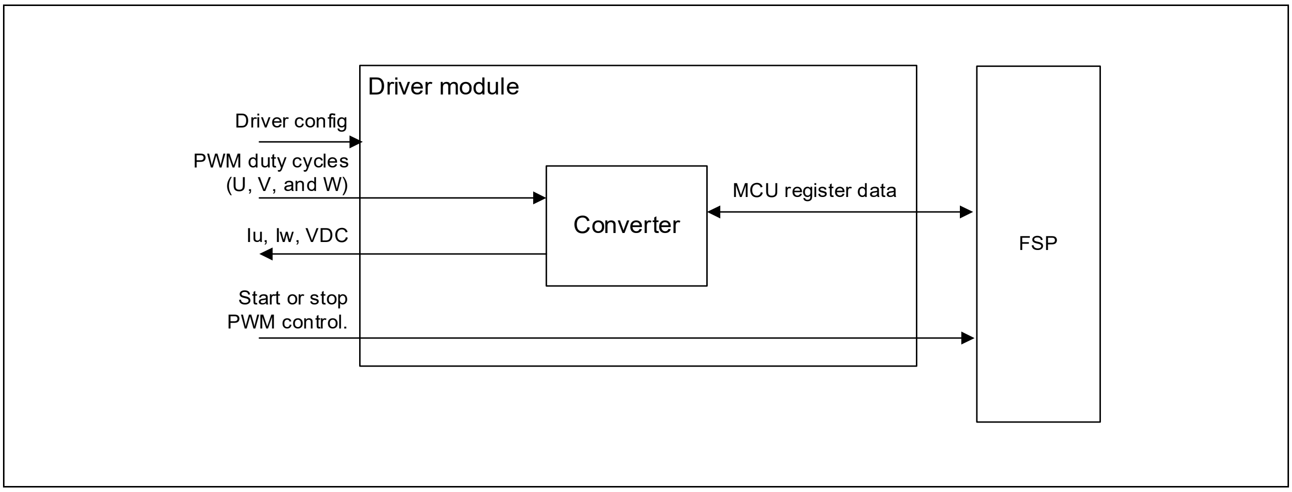


Figure 9-9 Configuration of the Driver Module

9.9.3 API

Table 9-27 lists and describes the API functions of the driver module.

Table 9-27 List of API Functions of the Driver Module

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

9.9.4 Configuration Items

Table 9-28 lists the configuration items of the driver module. Set up the functions to be used and the required parameters.

Table 9-28 Lit of Configuration Items

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) *	Sets the function for enabling PWM outputs.
	DRIVER_CFG_FUNC_PWM_OUTPUT_STOP	R_Config_MOTOR_StopTimerCtrl (API relay function of the FSP) *	Sets the function for disabling PWM outputs.
	DRIVER_CFG_FUNC_ADC_DATA_GET	R_Config_MOTOR_AdcGetConvVal (API relay function of the FSP) *	Sets the function for acquiring the A/D conversion results
	DRIVER_CFG_FUNC_DUTY_SET	R_Config_MOTOR_UpdDuty (API relay function of the FSP) *	Sets the function for setting the duty cycle
	DRIVER_CFG_FUNC_ZERO_DUTY_SET	R_Config_MOTOR_UpdZeroDuty (API relay function of the FSP) *	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) *	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.

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

9.9.5 Structure and Variable Information

Table 9-29 lists the structures that are used for the driver module. In the driver module, the structure for the driver module (g_st_driver) is defined by the API function for securing an instance of the module.

Table 9-29 List of Structures and Variables

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)
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 (ohms) (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)

9.9.6 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 9-30 lists the settings.

Table 9-30 Example of Settings in the Sample Program

Variable Name	Macro Name	File Name
*ADCDataGet	DRIVER_CFG_FUNC_ADC_DATA_GET	r_motor_module_cfg.h
*BLDCDutySet	DRIVER_CFG_FUNC_DUTY_SET	
*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	
f4_pwm_period_cnt	MOTOR_COMMON_CARRIER_SET_BASE	r_motor_module_cfg.h
f4_pwm_dead_time_cnt	MOTOR_COMMON_DEADTIME_SET	

10. Parameter Settings

10.1 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 10-1 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_pfc_cfg.h	Defines the initial values of parameters related to PFC.

10.2 MCU-Related Parameters

Table 10-2 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 10-2 List of MCU-Related Parameters

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	MOTOR_MCU_CFG_PWM_TIMER_FREQ	120.0	PWM timer frequency (MHz)
	MOTOR_MCU_CFG_CARRIER_FREQ	4.0	Carrier wave frequency (kHz)
	MOTOR_MCU_CFG_INTR_DECIMATION	0	Value to count for the skipping of carrier wave interrupts
	MOTOR_MCU_CFG_AD_FREQ	60.0	ADC operating frequency (MHz)
	MOTOR_MCU_CFG_AD_SAMPLING_CYCLE	2.0 × (7.25 + 120.0)	ADC sampling interval (cycles)
	MOTOR_MCU_CFG_AD12BIT_DATA	4095.0	ADC resolution
	MOTOR_MCU_CFG_ADC_OFFSET	0x7FF	ADC offset value

10.3 List of Parameters for Setting Control Functions

Table 10-3, Table 10-4, and Table 10-5 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 10-3 List of Operational Parameters (General)

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	MOTOR_TYPE_BLDC	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_IA_MAX_CALC_MULT	MTR_SQRT_3	Coefficient for calculating the overcurrent limit value. Set to √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 10-4 List of Operational Parameters (Related to Speed Control)

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	SPEED_CFG_OBSERVER	MTR_ENABLE	Enables or disables the disturbance torque/speed estimation observer. Enable: MTR_ENABLE Disable: MTR_DISABLE
	SPEED_CFG_MTPA	MTR_ENABLE	Setting of maximum torque per current control. Enable: MTR_ENABLE Disable: MTR_DISABLE For the motor in which Ld = Lq (an SPM motor), be sure to set this to MTR_DISABLE.
	SPEED_CFG_CTRL_PERIOD	0.0005f	Setting of the speed control interval (s). Set this to 0.0005f to specify 0.5 ms.

Table 10-5 List of Operational Parameters (Related to Current Control)

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_VOLT_ERR_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 10.6. Set this to MOD_METHOD_SVPWM in most cases.
	CURRENT_CFG_OFFSET_CALC_TIME	512	Sets the current offset measurement time.

10.4 Protection-Related Parameters

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

Table 10-6 Settings of the Motor Parameters and Inverter Parameters

File Name	Macro Name	Setting	Description
r_motor_inverter_cfg.h	INVERTER_CFG_CURRENT_LIMIT	21.2	Overcurrent limit value for the inverter board (A)
	INVERTER_CFG_OVERVOLTAGE_LIMIT	450.0	Overvoltage limit (V)
	INVERTER_CFG_UNDERVOLTAGE_LIMIT	100.0	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.

10.5 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 10-7 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 4.0 kHz.

Table 10-7 Parameters to be Modified When the PWM Carrier Frequency is Changed

Item	Item that Requires Change
Dead time value	See section 10.7, Inverter Parameters.
Carrier frequency	• Setting for the three-phase PWM GPT described in section 11.6 • MOTOR_MCU_CFG_CARRIER_FREQ described in section 10.2
Motor control-related parameters	Parameters for the following processing • Current regulator • Sensorless control • Flying start • Torque vibration suppression • Step-skipping (stall) detection

10.6 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 10-8 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 10-8 List of 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 10-9 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 10-10 List of Macros

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

10.7 Inverter Parameters

10.7.1 Overview

When you use the sample program, you need to correctly set the inverter information. Table 10-11 lists the inverter parameters set in the sample program.

Table 10-11 Settings of the Inverter Parameters

File Name	Macro Name	Setting	Description
r_motor_inverter_cfg.h	INVERTER_CFG_SHUNT_RESIST	0.01	Shunt resistance value (ohms)
	INVERTER_CFG_DEADTIME	2.0	Dead time (μs)
	INVERTER_CFG_VOLTAGE_GAIN	174.913	Coefficient for voltage detection
	INVERTER_CFG_CURRENT_AMP_GAIN	4.17	Gain of the amplifier for current detection
	INVERTER_CFG_INPUT_V	390.0	Input voltage (V)
	INVERTER_CFG_ADC_REF_VOLTAGE	3.3	Analog power-supply voltage for the MCU (V)
	INVERTER_CFG_COMP_V0	0.624	Coefficient for compensation of the voltage error (V)
	INVERTER_CFG_COMP_V1	1.248	Coefficient for compensation of the voltage error (V)
	INVERTER_CFG_COMP_V2	1.872	Coefficient for compensation of the voltage error (V)
	INVERTER_CFG_COMP_V3	2.496	Coefficient for compensation of the voltage error (V)
	INVERTER_CFG_COMP_V4	3.120	Coefficient for compensation of the voltage error (V)
	INVERTER_CFG_COMP_I0	0.084	Coefficient for compensation of the voltage error (A)
	INVERTER_CFG_COMP_I1	0.168	Coefficient for compensation of the voltage error (A)
	INVERTER_CFG_COMP_I2	0.264	Coefficient for compensation of the voltage error (A)
	INVERTER_CFG_COMP_I3	0.360	Coefficient for compensation of the voltage error (A)
	INVERTER_CFG_COMP_I4	0.600	Coefficient for compensation of the voltage error (A)

INVERTER_CFG_DEADTIME

Specify the dead time in μs (microseconds) that is described in the inverter specifications and design document. For the MCI-HV-1 inverter, 2.0 μs is specified.

INVERTER_CFG_INPUT_V

The default voltage value is 390 VDC, which is obtained by boosting single-phase 200 VAC through the PFC circuit.

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 10.7.4.

10.7.2 Current Detection Gain

In the MCI-HV-1 inverter, the voltage input to the ADC is specified by the amount of the current as shown in Table 10-12.

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. When a Hall current transformer is used instead of a shunt resistor, 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-1 specifications prescribe that the output current range is ±39.6 A (79.2 A peak-to-peak) for the voltage range from 0 V to 3.3 V, that is, 79.2 A/3.3 V = 24 A per volt. Assuming that the shunt resistance is 0.01 Ω, the coefficient becomes (1/0.01) × (1/24) = 4.166.

In that case, therefore, set INVERTER_CFG_CURRENT_AMP_GAIN to 4.166.

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

Table 10-12 Current Signal Specifications for the MCI-HV-1

3-Phase Output Current	ADC Input Voltage	A/D-Converted Value
+39.6 A	3.3 V	4095
0 A	1.65 V	2048
-39.6 A	0.0 V	0

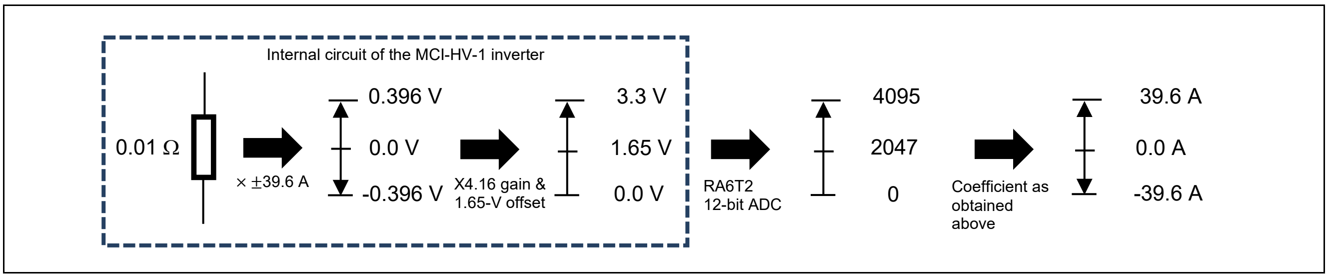


Figure 10-1 Flow of Calculation for Current Detection

10.7.3 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. If the input of 3.3 V to the ADC leads to the output of 577.2 V, 577.2/3.3 = 174.9 is obtained. Therefore, set INVERTER_CFG_VOLTAGE_GAIN to 174.9.

$$\text{INVERTER\_CFG\_VOLTAGE\_GAIN} = \frac{Reference\ inverter\ bus\ voltage}{Reference\ ADC\ input\ voltage} = \frac{577.2}{3.3} = 174.9$$

Table 10-13 Specifications of Inverter Bus Voltage Signal for the MCI-HV-1

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

10.7.4 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 μs 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 enabling or disabling the voltage error compensation function) to MTR_FLG_SET when R_MOTOR_CURRENT_ParameterUpdate (function for setting the control parameters for the current control module) is called. To disable the facility, set this flag to MTR_FLG_CLR.

(3) Setting the voltage compensation table

Perform a switching test on an actual inverter with the current flowing through it to create a voltage compensation table. Once the relationship between the current and voltage is obtained from the switching test, the obtained values can be converted to values that can be set in the voltage compensation table.

Figure 10-3 shows the U-phase voltage data acquired with an inverter bus voltage of 311 V and PWM carrier frequency of 16 kHz. From the data, six representative points showing the relationship between the current and voltage can be plotted as Figure 10-4. The five (current, voltage) points excluding (0,0) are the information used in voltage error compensation. Negative values can be omitted by calculating them as absolute values internally because they are point symmetrical to the positive values.

Note that the voltage error data shown here are for a PWM carrier frequency of 16 kHz; if the PWM carrier frequency setting is changed, the values in the voltage compensation table also require changing. If the PWM carrier frequency is 8 kHz, the values in the table must be converted to 8/16 (1/2) of their former values.

The limit on the compensation voltage value can be calculated by the following equation.

Compensation voltage limit = (carrier frequency [kHz] × dead time [μs] ÷ 1000) × inverter bus voltage value

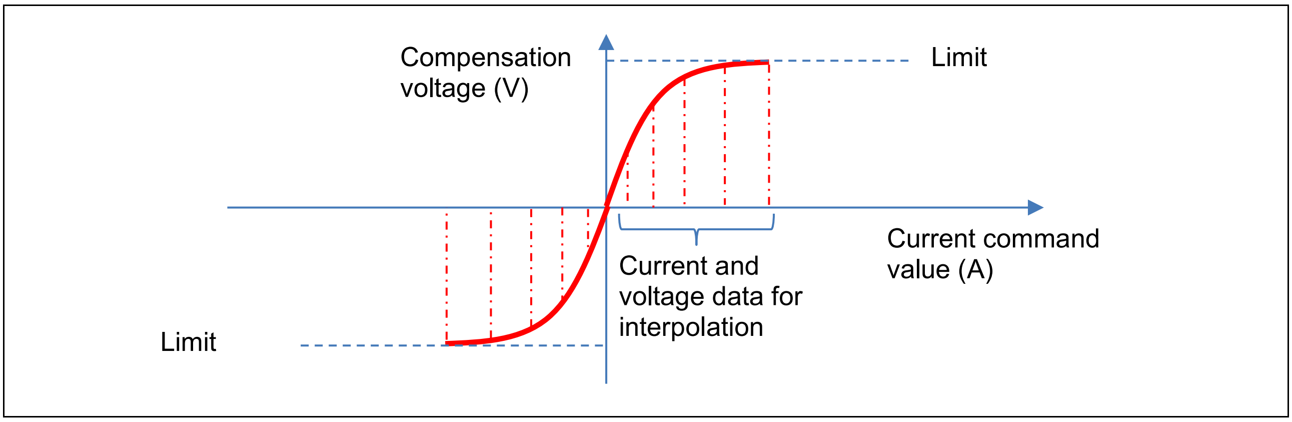


Figure 10-2 Relationship between the Compensation Voltage, Limit, and Current Command Values

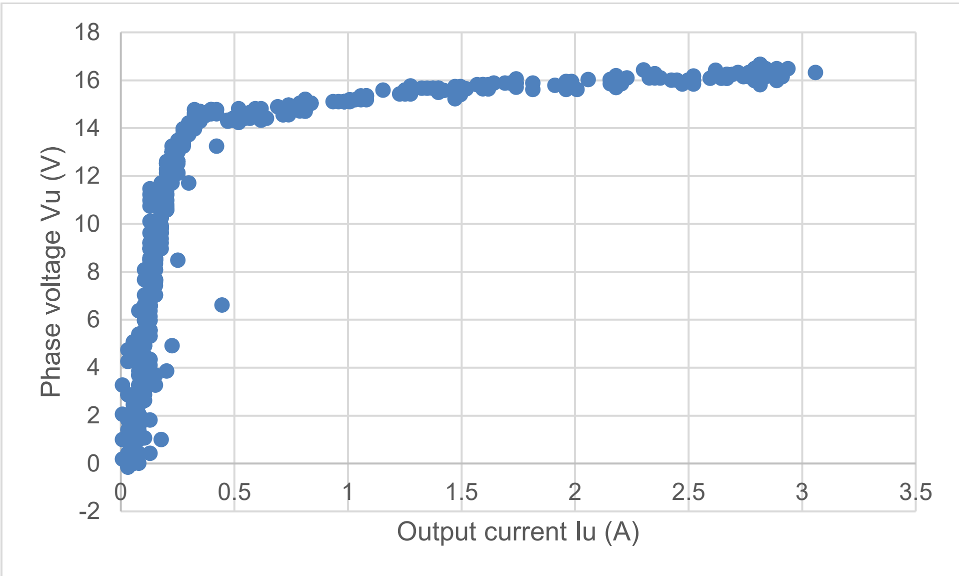


Figure 10-3 Example of Voltage Error Data

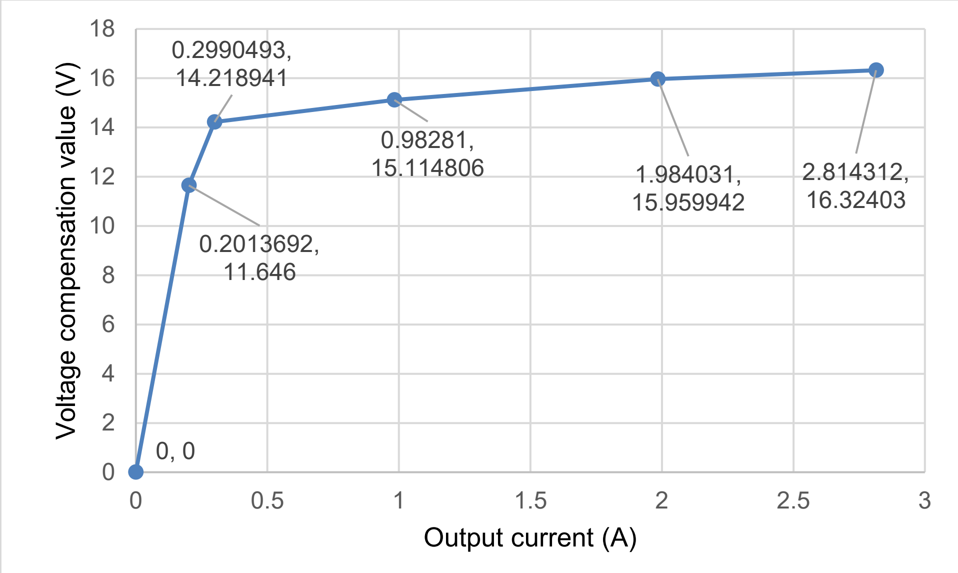


Figure 10-4 Example of Points for Dead Time Compensation Table Selected from Voltage Error Data

Table 10-14 Current and Voltage Data for Interpolation for Different Carrier Cycles

Carrier Frequency		**8 kHz**	**4 kHz**
	Iu	ΔVu	ΔVu
0	0.00	0.000	0.000
1	0.07	1.248	0.624
2	0.14	2.496	1.248
3	0.22	3.744	1.872
4	0.30	4.992	2.496
5	0.50	6.240	3.120

10.8 Motor Parameters

If the motor parameter information is not obtainable from the motor manufacturer, simple values for motor parameters R, Ld, and Lq can be obtained by using an LCR meter. A simple value for inductive voltage can also be obtained by using an oscilloscope. The methods described above are simplistic in that they do not take into account magnetic saturation or other phenomenon and are intended to quickly start the motor rotating, and the resulting values are subject to individual differences and measurement errors. Therefore, when the parameters are to be used in actual product development, measure the parameters by using measuring equipment having guaranteed accuracy.

The LCR meter should be calibrated periodically and measurement should proceed in a warm-up completed state after power having been supplied for at least 30 minutes. In addition, perform open-circuit compensation and short-circuit compensation in advance to reduce probe errors by using the 4-terminal pair method. For details, refer to the LCR meter's instruction manual.

When using the sample program, correctly set the information about the inverter and motor to be used. Table 10-15 lists the settings in the sample program.

Table 10-15 Motor Parameter Settings

File Name	Macro Name	Setting	Description
r_motor_targetmotor_cfg.h	MOTOR_CFG_POLE_PAIRS	3	Number of pole pairs
	MOTOR_CFG_MAGNETIC_FLUX	0.18f	Magnetic flux (wb)
	MOTOR_CFG_RESISTANCE	0.976375f	Resistance (ohms)
	MOTOR_CFG_D_INDUCTANCE	0.004715f	d-axis inductance (H)
	MOTOR_CFG_Q_INDUCTANCE	0.006245f	q-axis inductance (H)
	MOTOR_CFG_ROTOR_INERTIA	0.00114f	Rotor inertia (kgm^2)
	MOTOR_CFG_NOMINAL_CURRENT_RMS	6.1f	Rated current (A)
	MOTOR_CFG_MAX_SPEED_RPM	4000.0f	Maximum speed (rpm)

MOTOR_CFG_POLE_PAIRS

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

MOTOR_CFG_RESISTANCE

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

MOTOR_CFG_D_INDUCTANCE, MOTOR_CFG_Q_INDUCTANCE

For the wiring for measurement with an LCR meter, select two among the motor's three-phase output lines U, V, and W and connect the probes to them. Use the series equivalent circuit mode (Ls) as the measurement mode. For detailed measurement methods, refer to the LCR meter's instruction manual.

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

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

MOTOR_CFG_ROTOR_INERTIA

Specify the moment of inertia of the motor's rotor and shaft. The unit is kgm^2. Usually, you can find the value in the documentation provided with the motor. If a load is installed, the inertia of the load should be added to the setting.

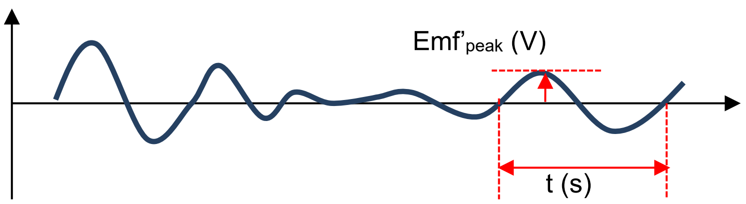
MOTOR_CFG_NOMINAL_CURRENT_RMS

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

MOTOR_CFG_MAGNETIC_FLUX

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

In cases where a motor cannot be connected to the end of the shaft, a simplistic method of obtaining the voltage waveforms by quickly rotating the shaft by hand can also be used. However, the accuracy cannot be guaranteed with this method so it is only suitable for test run purposes. When the shaft is turned by hand, a voltage waveform similar to the following image will be obtained. Select a cycle that is close to a sine wave at a constant speed and find the peak of the voltage and the period of the cycle.



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

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

To convert the unit (seconds) of the obtained time t to Hz, apply the formula f' = 1/t. Find the ratio of the electrical angular frequency (Hz) obtained from the rated speed of this PM motor to the obtained f (Hz) and multiply the ratio by the voltage Emf'_{rms} (V) that was obtained at the same time.

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

As a result, a rough value for the inductive voltage (V) that is generated when this PM motor is rotating at its rated speed can be obtained. To actually determine the inductive voltage, it must be measured by rotating the motor shaft at the rated speed using a load test device.

Next, obtain the magnetic flux linkage Ψ (Wb) from the inductive voltage. In general, the inductive voltage and magnetic flux linkage have the relationship indicated below, with f as the electrical angular frequency (Hz) at the rated speed.

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

The magnetic flux linkage Ψ (Wb) can be obtained by re-arranging the equation and assigning the value for the inductive voltage Emf (V) obtained above and the electrical angular frequency (Hz) during rated-speed operation.

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

Set the obtained magnetic flux linkage Ψ as MOTOR_CFG_MAGNETIC_FLUX in r_motor_targetmotor_cfg.h.

10.9 Current Control Parameters

Table 10-16 lists the current control parameters. The current control parameters should be calculated based on the motor parameters, PWM carrier frequency, and desired current response performance.

The macros for the current control parameters listed in Table 10-16 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 current control parameters can be changed.

Table 10-16 List of Current Control Parameters

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_OFFSET_CALC_TIME	512	Current offset measurement time
	CURRENT_CFG_OMEGA	150.0f	Natural frequency for the current control system (Hz)
	CURRENT_CFG_ZETA	1.0f	Attenuation coefficient for the current control system

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.

CURRENT_CFG_OMEGA, CURRENT_CFG_ZETA

The control gain is adjusted by tuning the natural frequency and attenuation coefficient for the current control system. Set the natural frequency for the current control system to be proportional to the frequency of current control. The natural frequency can be set to up to about 1/10 of the current control frequency (PWM carrier frequency). However, in many cases, a lower value may be set to leave a margin in consideration of noise during position detection and current detection.

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

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

10.10 Maximum Torque per Current Control

Maximum torque per current control is an algorithm used to adjust the output torque of a PM motor that has saliency (a PM motor satisfying Lq > Ld) to be the maximum possible value. This function can be enabled or disabled by the SPEED_CFG_MTPA setting. In general, enable it.

However, when the Ld and Lq values of the PM motor are the same or almost the same, this function cannot be used. In such cases, be sure to disable it (specify MTR_DISABLE).

Table 10-17 List of Configuration Information

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	SPEED_CFG_MTPA	MTR_ENABLE	Set this to MTR_ENABLE to use the maximum torque per current control function. When it is not to be used, be sure to set it to MTR_DISABLE.

10.11 Speed Control Parameters

Table 10-18 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 10-18 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 10-18 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_OMEGA	3.0f	Natural frequency for the speed control system (Hz)
	SPEED_CFG_ZETA	1.0f	Attenuation coefficient for the speed control system
	SPEED_CFG_LPF_OMEGA	25.0f	LPF bandwidth for the speed control system (Hz)
	SPEED_CFG_SPEED_LIMIT_RPM	4200.0f	Speed limit value (rpm) (mechanical angle)
	SPEED_CFG_RATE_LIMIT_RPM	300.0f	Acceleration limit per control interval (rpm)

SPEED_CFG_CTRL_PERIOD

Set to 0.0005 s (0.5 ms). If this setting needs to be changed, the AGT0 timer setting of 11.7 must also be changed to specify a new speed control interval.

SPEED_CFG_OMEGA, SPEED_CFG_ZETA

In the speed control module, the control gain is adjusted by tuning the natural frequency and attenuation coefficient for the speed control system. Increasing the natural frequency for the speed control system improves the responsiveness and the capability of the speed to follow the requests of the speed commands.

The maximum natural frequency for speed control should be no more than 1/3 of the maximum natural frequency for current control to prevent interference with current control. If the natural frequency for the current control system is 500 Hz, the natural frequency for speed control is 500 Hz/3 = 166 Hz. However, this sample program estimates the speed without using an encoder, so set a frequency lower than the natural frequency set by CURRENT_CFG_PLL_EST_OMEGA. The default value specifies a frequency with relatively low capability of following the requests of the speed commands to provide a margin. For example, if a disturbance oscillation is at a frequency above the natural frequency, increasing the value of the natural frequency to match that of the disturbance may improve the capability of following the latter oscillation and driving of the motor may be more stable than with the default setting in some cases.

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

SPEED_CFG_LPF_OMEGA

A filter is set for the estimated speed to suppress fluctuations. If the value is too small, the speed responsiveness deteriorates and the motor will not be able to follow a sudden change in speed.

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.

10.12 Disturbance Torque/Speed Estimation Observer

The disturbance torque/speed estimation observer is a function to reduce cogging torque and vibration during very low-speed operation. The function is enabled or disabled by SPEED_CFG_OBSERVER. Normally, set to MTR_ENABLE (enabled).

Set the disturbance torque/speed estimation observer control parameters of the speed control module API by using R_MOTOR_SPEED_ExtObserverParameterUpdate (API function for updating the parameters). This module sets the following three types of parameters:

- Motor inertia
- Natural frequency of the disturbance torque/speed estimation observer
- Sampling interval of the observer

For the motor inertia and the sampling interval of the observer, make sure that you set correct values that are actually used for control. Decreasing the natural frequency for the disturbance torque/speed estimation observer further reduces speed ripple but degrades responsiveness to change of the commanded speed. Make adjustment while checking the speed responsiveness. As a guideline, the natural frequency for the disturbance observer becomes about four to six times the natural frequency for the speed control system.

The disturbance torque/speed estimation observer is used for motor stability during very low-speed operation around 1 to 30 rpm, so it is not effective at speeds above 100 rpm. To facilitate the switching of the disturbance torque/speed estimation observer, attenuation processing (Figure 10-5) proceeds with the use of the parameters SPEED_CFG_SOB_OUTLIM_START_RPM and SPEED_CFG_SOB_OUTLIM_END_RPM.

Table 10-19 List of Configurations

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	SPEED_CFG_OBSERVER	MTR_ENABLE	Set this to MTR_ENABLE to use the disturbance torque/speed estimation observer. When it is not to be used, set it to MTR_DISABLE.
	SPEED_CFG_SOB_OMEGA	7.5	The unit is Hz. Approximately 4 to 6 times the natural frequency of the speed control system.

	SPEED_CFG_SOB_ZETA	1	Sets the attenuation coefficient. Normally, specify 1.
	SPEED_CFG_SOB_OUTLIM_START_RPM	25	Set the start speed (rpm) of the observer's upper limit function.
	SPEED_CFG_SOB_OUTLIM_END_RPM	30	Set the end speed (rpm) of the observer's upper limit function.

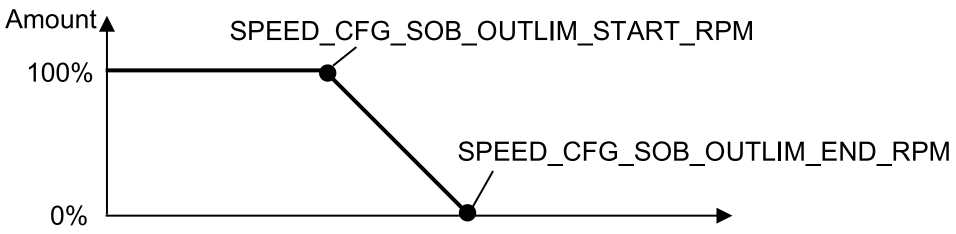


Figure 10-5 Relationship between the Amount of Compensation and Parameters for the Disturbance Torque/Speed Estimation Observer

10.13 Sample Delay 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 10-20 List of Configuration Information

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

10.14 Sensorless Control Parameters

This section describes how to set the parameters required for sensorless control. In sensorless control, a current sensor and predetermined motor and control parameters are used to start the motor. If the parameters are inappropriate or the inverter or motor in use is not suited for sensorless control, the desired performance may not be achieved.

(1) Low-speed-range sensorless control parameters

Table 10-21 shows the list of parameters to be used for low-speed-range sensorless control.

Table 10-21 Setting Parameters for Low-Speed-Range Sensorless Control

File Name	Macro Name	Setting	Unit	Description
r_motor_module_cfg.h	CURRENT_CFG_PLL_ESTLOW_OMEGA	50	Hz	Natural frequency for the low-speed-range sensorless control PLL (Hz)
	CURRENT_CFG_PLL_ESTLOW_ZETA	1	-	Attenuation coefficient for the low-speed-range sensorless control PLL
	CURRENT_CFG_ESTLOW_PULSEVOLT	100	V	Pulse voltage value applied when estimating the magnetic pole position at startup
	CURRENT_CFG_ESTLOW_PULSEVOLT_RUNNING	50	V	Pulse voltage value applied during magnetic pole position estimation during operation
	CURRENT_CFG_ESTLOW_ESTTIME	Equivalent to 0.2 s	times	Estimation process timeout
	CURRENT_CFG_ESTLOW_ESTTIME_OVER	Equivalent to 0.3 s	times	Timeout value for judging estimation processing errors
	CURRENT_CFG_ESTLOW_PULSEFREQ_BOOT	3	times	Pulse application cycle for estimating the magnetic pole position at startup
	CURRENT_CFG_ESTLOW_PULSEFREQ_DRIVE	1	times	Pulse application cycle for estimating the magnetic pole position during operation
	MOTOR_ANGEST_THRESHOLD	0.00872	rad	Threshold for detectability of magnetic pole position estimation
r_motor_current_lowspd_sensorless.h	MOTOR_SENSORLESS_VECTOR_THRESHOLD_HIGHSPEED	65.9734	rad/s	Sets the speed at which the sensorless algorithm switches from the low-speed range to the medium-to-high-speed range.
	MOTOR_SENSORLESS_VECTOR_THRESHOLD_LOWSPEED	59.6902	rad/s	Sets the speed at which the sensorless algorithm switches from the medium-to-high-speed range to the low-speed range.
	MOTOR_SENSORLESS_VECTOR_CURRENT_TABLE_SIZE	8	-	Current buffer table size for estimation. Do not change the setting from 8.
	MOTOR_SENSORLESS_VECTOR_PF_START_CNT	100	count	Parameter for adjusting the timing with which polarity determination starts
r_motor_sensorless_vector_api.h	CURRENT_SENSORLESS_CHGARGCNT_TOHIGH	2	-	Number of current control cycles to be used in switching between the sensorless algorithms. Set a fixed value for switching from low speed to medium-to-high speed.
	CURRENT_SENSORLESS_CHGARGCNT_TOSLOW	1	-	Number of current control cycles to be used in switching between the sensorless algorithms. Set a fixed

				value for switching from medium-to-high speed to low speed.

CURRENT_CFG_PLL_ESTLOW_OMEGA

Set the natural frequency of the position estimation PLL for low-speed-range sensorless control. The default is 50 Hz. Specify this value considering the current control cycle and the frequency bandwidth of the current response.

CURRENT_CFG_PLL_ESTLOW_ZETA

Set the attenuation coefficient of the position estimation PLL for low-speed-range sensorless control. Normally, specify 1.0.

CURRENT_CFG_ESTLOW_PULSEVOLT

Specify the magnitude of the pulse (voltage) to be applied at startup. Specify within the range of 0 V to 1/2 of the inverter bus voltage. Since magnetic saturation must be generated to determine polarity, set 100 V or more, depending on the motor specifications. We recommend that you obtain the detailed voltage values experimentally. In this sample program, 100 V is set experimentally.

CURRENT_CFG_ESTLOW_PULSEVOLT_RUNNING

Specify the magnitude of the pulse (voltage) to be applied during operation. Specify within the range of 0 V to 1/2 of the inverter bus voltage. Since generating magnetic saturation phenomenon will adversely affect the operational performance, set the voltage no greater than half of the voltage set by CURRENT_CFG_ESTLOW_PULSEVOLT. We recommend that you obtain detailed voltage values experimentally, as they will vary from motor to motor.

CURRENT_CFG_ESTLOW_ESTTIME

Specify the time for estimating the magnetic pole position at startup. Multiply by 0.08 to get milliseconds. If 2500 is set, then 2500 x 0.08 = 200 ms. In this case, at least 200 ms is waited for magnetic pole position estimation. Setting it longer can improve the accuracy of magnetic pole position estimation. Set the value to be approximately 200 ms.

CURRENT_CFG_ESTLOW_ESTTIME_OVER

Set the period of time to determine that the results of the magnetic pole position estimation and polarity determination have converged and that operation is ready. To define condition that the determination is completed within 100 ms following 200 ms set in CURRENT_CFG_ESTLOW_ESTTIME, specify a value equivalent to 300 ms. The value to be set is 3750, resulting in a waiting time of 3750 × 0.08 = 300 ms.

CURRENT_CFG_ESTLOW_PULSEFREQ_BOOT

Specify the cycle of pulses to be applied when estimating the magnetic pole position at startup. The setting range is 1 to 8. For the IPM motor used in this sample program, we have experimentally confirmed that stable estimation can be achieved by setting the pulse application cycle to 2.5 kHz or less. Estimation may fail under conditions where the pulse application cycle is long while the PWM carrier cycle is long.

Table 10-22 PWM Carrier Cycle and Pulse Application Cycle Settings

PWM Carrier Cycle	CURRENT_CFG_ESTLOW_PULSEFREQ_BOOT or CURRENT_CFG_ESTLOW_PULSEFREQ_DRIVE Value	Pulse Application Cycle
8 kHz	1	4.00 kHz
	2	2.00 kHz
	3	1.33 kHz
	4	1.00 kHz
	5	0.80 kHz
	6	0.67 kHz
	7	0.57 kHz
	8	0.50 kHz
4 kHz	1	2.00 kHz
	2	1.00 kHz
	3	0.67 kHz
	4	0.50 kHz
	5	0.40 kHz
	6	0.33 kHz
	7	0.29 kHz
	8	0.25 kHz
2 kHz	1	1.00 kHz
	2	0.50 kHz
	3	0.33 kHz
	4	0.25 kHz
	5	0.20 kHz
	6	0.17 kHz
	7	0.14 kHz
	8	0.13 kHz

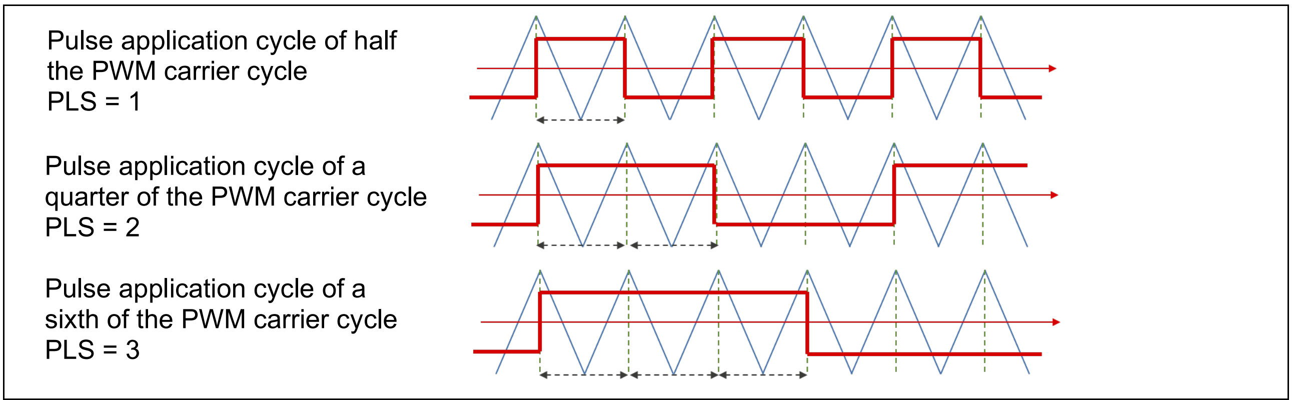


Figure 10-6 Illustration of PWM Carrier Cycle and Pulse Application Cycle

CURRENT_CFG_ESTLOW_PULSEFREQ_DRIVE

Specify the cycle of pulses to be applied when estimating the magnetic pole position during operation. See Table 10-22 for reference. Normally, specify 1. If the estimation fails during operation, adjust the value of CURRENT_CFG_ESTLOW_PULSEVOLT_RUNNING.

MOTOR_SENSORLESS_VECTOR_THRESHOLD_HIGHSPEED

Set the speed (in rad/s) at which the sensorless algorithm switches from the low-speed range to the medium-to-high- speed range. Above this speed, the high-frequency pulse voltage injection is stopped.

MOTOR_SENSORLESS_VECTOR_THRESHOLD_LOWSPEED

Set the speed (in rad/s) at which the sensorless algorithm switches from the medium-to-high-speed range to the low-speed range. Below this speed, the high-frequency pulse voltage injection is started.

MOTOR_SENSORLESS_VECTOR_CURRENT_TABLE_SIZE

Set the current buffer table size for the pulse cycle. Do not change its default value and leave it at 8.

MOTOR_SENSORLESS_VECTOR_ PF_START_CNT

This parameter is used to adjust the timing with which polarity determination starts. A vlaue from 0 to 255 is specifiable. One count is made per current control cycle. If 25 is set when the current control cycle is 80 µs, the polarity determination will start after waiting for 2 ms. Set the value according to the motor constants and the natural frequency for the position estimation control system. Adjusting the timing is required when polarity determination fails or a motor in use is difficult to saturate magnetically.

CURRENT_SENSORLESS_CHGARGCNT_TOHIGH

CURRENT_SENSORLESS_CHGARGCNT_TOSLOW

These parameters are used in switching between the low- and medium-to-high-speed-range algorithms. Do not change their default values.

(2) Medium-to-high-speed-range sensorless control parameters

Table 10-23 lists the BEMF observer parameters to be used for medium-to-high-speed-range sensorless control.

For details on how to set the parameters, refer to “Sensorless Vector Control of a Permanent Magnet Synchronous Motor for the Evaluation System for BLDC Motor (R01AN6307EJ0110)” on which this sample program is based.

Table 10-23 Setting Parameters for Medium-to-High-Speed-Range Sensorless Control

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_E_OBS_OMEGA	400	Natural frequency for the BEMF observer (Hz)
	CURRENT_CFG_E_OBS_ZETA	1	Attenuation coefficient for the BEMF observer
	CURRENT_CFG_PLL_EST_OMEGA	20	Natural frequency for the medium-to-high-speed-range sensorless control PLL (Hz)
	CURRENT_CFG_PLL_EST_ZETA	1	Attenuation coefficient for the medium-to-high-speed-range sensorless control PLL

CURRENT_CFG_E_OBS_OMEGA

Specify the natural frequency for the BEMF observer. The frequency should be sufficiently separated from the CURRENT_CFG_OMEGA setting for the current regulator, with around ×2 to ×3 of CURRENT_CFG_OMEGA as a suggested range. Specifying a value that is relatively close to CURRENT_CFG_OMEGA will make obtaining correct values for angle impossible due to oscillation of the output from the BMEF observer.

CURRENT_CFG_E_OBS_ZETA
Specify the attenuation coefficient for the BEMF observer. Specify 1.0 in general.

CURRENT_CFG_PLL_EST_OMEGA
Specify the natural frequency for the PLL for use in calculating speeds by applying PLL operation to the angle errors obtained by the BEMF observer. The frequency should be around 1/10 of that for the BEMF observer and higher than SPEED_CFG_OMEGA for the speed regulator. If an inappropriate value is specified, correct estimation of angles will not be possible and the output may oscillate.

CURRENT_CFG_PLL_EST_ZETA
Specify the attenuation coefficient for the PLL for use in calculating speeds by applying PLL operation to the angle errors obtained by the BEMF observer. Specify 1.0 in general.

10.15 Flux Weakening Control Parameters

The only parameter provided for flux weakening control is for enabling or disabling it; no other control parameters are used. The operating conditions such as the inverter bus voltage and speed are monitored and control automatically begins when the necessary conditions are satisfied.

Table 10-24 List of Configuration Information

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	SPEED_CFG_FLUX_WEAKENING	MTR_ENABLE	Set this to MTR_ENABLE to use the flux weakening control function. When it is not to be used, set it to MTR_DISABLE.

10.16 Flying Start Parameters

The following describes the parameters for flying start operation.

Table 10-25 List of Configuration Information

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_FLYING_START	MTR_DISABLE	Set this to MTR_ENABLE to use the flying start function in starting up the motor. Even if MTR_DISABLE has been specified, the setting can later be changed through the com variable com_u1_flag_flying_start_use.

Table 10-26 List of Flying Start Parameters

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	SENSORLESS_VECTOR_FLY_START_CURRENT_TH	2.0f	Specify the threshold (A) for the switched-on current.
	SENSORLESS_VECTOR_FLY_START_OVER_TIME_SEC	0.005f	
	SENSORLESS_VECTOR_FLY_START_OFF_TIME_SEC	0.0005f	
	SENSORLESS_VECTOR_FLY_START_ACTIVE_BRAKE_TIME_SEC	1.0f	
	SENSORLESS_VECTOR_FLY_START_RESTART_SPEED_LIMIT	600.0f	Specify the minimum speed at which restarting through flying start control is allowed.

SENSORLESS_VECTOR_FLY_START_CURRENT_TH

Specify the threshold for the current of switched-on elements to be used in judging the Ton time. This parameter determines the Ton time.

To reduce the effects of the resolution of current detection on the rotational velocity and estimation of the pole position, the elements on the lower side of the three-phase inverter are simultaneously turned on for the Ton time, that is, until the detected current vector (Ia) reaches 2.0 A due to conditions 1 and 2 below (0.96 A < Ia < 5.7 A). A longer Ton time increases the approximation error of equation 7.11.5, so 2.0 A is used as the threshold current for the Ton time.

Condition 1	Determine the threshold such that the magnitude of the detected current vector (Ia) is around at least 100 times greater than the current detection resolution. For example, when

	the current detection resolution of the MCI-HV-1 inverter board is 79.2 A/12 bits (19.3 mA/LSB), the threshold will be 19.3 mA × 100 = 1.93 A or a greater value. Note that if the switched-on current is slow to rise (the slope of the rise is gentle) and the (Ton + Toff)max condition for flying start is not satisfied due to the impedance of the wiring or the constants of the motor, experimentally lower the current threshold value by roughly halving it.
Condition 2	The magnitude of the detected current vector (Ia) must be no greater than the target motor's rated RMS phase current × √3. As the target motor's rated RMS phase current (3.3 A) × √3 ≃ 5.7 A, the detected current vector (Ia) must be no greater than 5.7 A.

SENSORLESS_VECTOR_FLY_START_OVER_TIME_SEC

Specify the maximum time for waiting until the threshold of the switched-on current is reached in seconds. If this time has elapsed, the motor is assumed to be stopped or rotating at a low speed and active braking is applied.

SENSORLESS_VECTOR_FLY_START_OFF_TIME_SEC

Specify the Toff time for a flying start. The Toff time is required to satisfy conditions 1 and 2 below (0.61 ms < Toff < 3.5 ms), so Toff is set to 2 ms.

Condition 1	To satisfy the approximation condition i(0) = 0 for equation 7.11.5 in the dq-axis rotation coordinate system, the three-phase currents require attenuation to zero after the elements have been turned on for the Ton time. As a result of circuit simulation for obtaining the Toff time until the current vector Ia is attenuated from 2 A to 0 A, the maximum Toff time at the maximum rotation speed 4000 rpm is about 0.61 ms. Therefore, specify Toff > 0.61 ms.
Condition 2	As a result of circuit simulation for obtaining the switched-on time Ton after the start of turning the three-phase lower side on until the threshold current of 2 A is reached, Ton = 0.25 ms can be obtained at the maximum rotation speed 4000 rpm. Here, the (Ton + Toff)max time that can be converted from the rotation speed is 3.75 ms, so Toff < (3.75 – 0.25) ms = 3.5 ms must be satisfied.

SENSORLESS_VECTOR_FLY_START_ACTIVE_BRAKE_TIME_SEC

If the motor is rotating at a speed at which a flying start is not possible, the motor is stopped through active braking. Specify the time (s) for applying active braking to stop the motor.

10.17 Torque Vibration Suppression Parameters

The torque vibration suppression function can be used while the motor is running under sensorless vector control with the use of the BEMF observer.

As this control function is mainly for use in the low-speed range and the conditions for use will depend on the characteristics of the source (such as a compressor) of the torque vibration, this function is manually enabled or disabled by modifying the com_u1_flag_trq_vibration_comp_use variable through the RMW. The following table lists the steps of the procedure for operating the torque vibration suppression function through the RMW.

Table 10-27 Procedure for Operating the Torque Vibration Suppression Function

Step	Operation	Manipulation in the RMW
1	Start the torque vibration suppression function.	Set com_u1_flag_trq_vibration_comp_use to 1. Note: Only enable this com variable under sensorless vector control.
2	After the torque vibration suppression function has started operating, enable the learning function at a desired time.	Set com_u1_flag_trqvib_comp_learning to 1.

3	Disable the learning function when the torque vibration suppression function has reduced the fluctuations in speed.	Clear com_u1_flag_trqvib_comp_learning to 0.
4	Continue running the torque vibration suppression function with the speed kept constant.	—
5	Disable the torque vibration suppression function when the speed requires changing or the function is otherwise to be disabled.	Clear com_u1_flag_trq_vibration_comp_use to 0.
6	Return to step 1 as required after the speed has been changed.	—

Table 10-28 List of Torque Vibration Suppression Parameters

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_TRQVIB_OUTPUT_GAIN	0.001	Output gain
	CURRENT_CFG_TRQVIB_TIMELEAP	0.0	
	CURRENT_CFG_TRQVIB_LPF_GAIN	0.0005	
	CURRENT_CFG_TRQVIB_INPUT_WEIGHT_2	1.0	These values are used to specify the weights for the input signals. Specify them to suit the characteristics of the motor and load.
	CURRENT_CFG_TRQVIB_INPUT_WEIGHT_1	0.0	
	CURRENT_CFG_TRQVIB_INPUT_WEIGHT_0	0.0	

CURRENT_CFG_TRQVIB_OUTPUT_GAIN
Specify the gain to be added to the q-axis current command. Specifying a small value decreases the feedforward values in response to the q-axis current command values. However, the internal algorithm of torque vibration suppression includes an element of integration, so the feedback value is kept unchanged in a steady state regardless of the gain although the times in transient states will change.

CURRENT_CFG_TRQVIB_TIMELEAP
This parameter adjusts the output phase. Specify it within the range from 0 to 2π (6.28) in radians.

CURRENT_CFG_TRQVIB_LPF_GAIN
Specify the constant for the filter that extracts the oscillation component. Use the following equation.

$$Gain = \frac{250\mu s}{LPF\ Time\ [\mu s]}$$

CURRENT_CFG_TRQVIB_INPUT_WEIGHT_0,
CURRENT_CFG_TRQVIB_INPUT_WEIGHT_1,
CURRENT_CFG_TRQVIB_INPUT_WEIGHT_2
Specify the weight values to be used for storage of the moving averages in the internal table. Adjust these values when the variations in the vibration suppression effects are strong.

10.18 Step-Skipping (Stall) Detection Parameters

The following lists the step-skipping (stall) detection parameters.

Table 10-29 List of Configuration Information

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_STALL_DETECTION	MTR_ENABLE	Set this to MTR_ENABLE to use the step-skipping (stall) detection function. When it is not to be used, set it to MTR_DISABLE.

Table 10-30 List of Step-Skipping (Stall) Detection Parameters

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_STALL_D_HPF_GAIN	0.00025	Specify the gain of the HPF for extracting the oscillation components from the detected d- and q-axis current values.
	CURRENT_CFG_STALL_Q_HPF_GAIN	0.00025	
	CURRENT_CFG_STALL_THRESHOLD_LEVEL	5.0	Specify the threshold (A) for the level of current to be judged as representing step-skipping (stall).
	CURRENT_CFG_STALL_THRESHOLD_TIME	0.1	Specify the time (s) for which the level of current continuing to exceed the threshold is to be judged as representing step-skipping (stall).

CURRENT_CFG_STALL_D_HPF_GAIN
CURRENT_CFG_STALL_Q_HPF_GAIN
The gain of the HPF can be calculated from the following equation. When the current control interval at which the step-skipping (stall) detection processing operates is 250 µs and the time constant of the HPF is 8 ms, the gain of the HPF is about 0.016.

$$HPFGain = \frac{Tc\ [s]}{HPF\ Time\ [s]} = \frac{250\ \mu s}{8\ ms} = 0.03125$$

10.19 PFC Control Parameters

(1) General parameters

The following describes the general parameters and their use in calculations for PFC control. Table 10-31 lists the general parameters for PFC settings.

Table 10-31 Parameters for PFC Settings (General Parameters)

Macro Name	Setting	Unit	Description
PFC_MCU_CFG_PWM_TIMER_FREQ	120.0	MHz	Frequency of the PWM timer
PFC_MCU_CFG_CARRIER_FREQ	32.0	kHz	Carrier frequency
PFC_MCU_CFG_AD12BIT_DATA	4095.0	-	Resolution of the ADC
PFC_MCU_CFG_ADC_OFFSET	0x7FF	-	Offset value for the ADC
PFC_CFG_ADC_REF_VOLTAGE	3.3	V	Reference voltage for the ADC
PFC_CFG_SHUNT_RESIST	0.005	Ohm	Shunt resistance
PFC_CFG_AC_VOLTAGE_GAIN	426.5319149	-	Gain of AC voltage conversion
PFC_CFG_BUS_VOLTAGE_GAIN	174.9130435	-	Gain of bus voltage conversion
PFC_CFG_CURRENT_AMP_GAIN	8.333333333	-	Gain of current conversion
PFC_CFG_BUS_VOLTAGE_OFFSET	0	-	Offset adjustment value for the bus voltage
PFC_CFG_INPUT_VOLTAGE_OFFSET	0	-	Offset adjustment value for the AC voltage
PFC_CFG_CURRENT_OFFSET	0	-	Offset adjustment value for the current

PFC_MCU_CFG_PWM_TIMER_FREQ

Specify the clock frequency for the GPT to be used in PFC control. The default value is 120 MHz.

PFC_MCU_CFG_CARRIER_FREQ

Specify the PWM carrier frequency for PFC control. The default value is 32 kHz. When changing it, be sure to specify an integer multiple of the PWM carrier frequency for use in motor control.

PFC_MCU_CFG_AD12BIT_DATA

This macro is used to specify a maximum value that can be obtained after conversion by the 12-bit ADC. Do not change the value.

PFC_MCU_CFG_ADC_OFFSET

This macro is used to specify a constant for offsetting 1.65 V to 0. The required value is 0x7FF (2047) when the maximum input voltage is 3.3 V. Do not change the value.

PFC_CFG_ADC_REF_VOLTAGE

This macro is used to specify the reference voltage for the ADC. The required value is 3.3 V for the RA6T2. Do not change the value.

PFC_CFG_SHUNT_RESIST

Specify the shunt resistance (ohms) used for current detection in PFC control.

PFC_CFG_AC_VOLTAGE_GAIN

Specify the gain of input AC voltage detection for use in PFC control. The range of the input voltage Vac from 0 V to 3.3 V is assumed to correspond to the range from –703.8 V to +703.8 V. As the resolution of the ADC is 12 bits, values within the range from 0 to 4095 are detectable.

Table 10-32 Relationship between the Input AC Voltage and Value Detected through the ADC

Voltage Input to ADC	Value Detected through ADC (12 Bits)	AC Voltage
0.0 V	0	-703.8 V
1.65 V	2048	0 V
3.3 V	4095	+703.8 V

Obtain the detection gain as follows.

$$Gain = \frac{1}{3.3} \times |703.8 - (-703.8)| = 426.5$$

PFC_CFG_BUS_VOLTAGE_GAIN

Specify the gain of bus voltage detection for use in PFC control. The range of the bus voltage (PFC output voltage) from 0 V to 3.3 V is assumed to correspond to the range from 0.0 V to +577.2 V. As the resolution of the ADC is 12 bits, values within the range from 0 to 4095 are detectable.

Table 10-33 Relationship between the Input AC Voltage and Value Detected through the ADC

Voltage Input to ADC	Value Detected through ADC (12 Bits)	Bus Voltage
0.0 V	0	0.0 V
3.3 V	4095	+577.2 V

Obtain the detection gain as follows.

$$Gain = \frac{1}{3.3} \times 577.2 = 174.9$$

PFC_CFG_CURRENT_AMP_GAIN

Specify the gain of current detection for use in PFC control. This value can be calculated in a similar way to INVERTER_CFG_CURRENT_AMP_GAIN. The MCI-HV-1 specifications prescribe that the range of current is ±39.6 A (79.2 A peak-to-peak) for the range of voltage from 0 V to 3.3 V; that is, 79.2 A/3.3 V = 24 A per volt. Assuming that the shunt resistance PFC_CFG_SHUNT_RESIST is 0.005 Ω, the gain becomes (1/0.005) × (1/24) = 8.333.

In that case, therefore, set PFC_CFG_CURRENT_AMP_GAIN to 8.333.

PFC_CFG_BUS_VOLTAGE_OFFSET

PFC_CFG_INPUT_VOLTAGE_OFFSET

PFC_CFG_CURRENT_OFFSET

Specify the offset values to adjust for the individual variations between voltage and current detectors. As these values are used to adjust the offset values of the ADC and operational amplifier, they can be specified in steps of the voltage or current that correspond to one bit of the 12-bit detected values after conversion through the ADC.

Use a highly accurate voltmeter and the RMW to acquire the voltages, ADC-detected values, and internally obtained bus voltages at regular intervals in the range from 0 V to the maximum value seen in usage. From these results, adjust the offset values to maximize the accuracies of the slopes and offsets of the voltages in the voltage range that will most frequently be used.

(2) Command and limit values

Specify the target command values and limit values on the output power to be applied in PFC control.

Table 10-34 Parameters for PFC Settings in r_pfc_cfg.h (Command Values and Limit Values)

Macro Name	Setting	Unit	Description
VAC_FREQ	50.0	Hz	Frequency of the input AC voltage
DATA_ARR_SIZE	320	-	Number of elements in the array for storing AC voltages
VDC_TARGET_VALUE	390.0	V	Target bus voltage
PFC_OUT_MAX_POWER	1000.0	W	Maximum PFC output
PFC_OUT_MIN_POWER	500.0	W	Minimum PFC output
PFC_OUT_POWER_COEF	1.4	-	Coefficient for output adjustment

VAC_FREQ
Specify the frequency of the input AC voltage. This is 50 Hz or 60 Hz in Japan, depending on the region. If this is changed from the default value (50 Hz), the DATA_ARR_SIZE value also requires changing as described below.

DATA_ARR_SIZE
Adjust this value according to the frequency of the input AC voltage. The size is 320 for 50 Hz or 267 for 60 Hz. More specifically, when the PFC control cycle Fc is 32 kHz and the frequency (f) of the input AC voltage is 50 Hz, this value is obtained as follows.

$$DATA\ ARR\ SIZE = \frac{Fc\ [Hz]}{2 \times f\ [Hz]} = \frac{32000}{2 \times 50} = 320$$

VDC_TARGET_VALUE
Specify the target command value (V) for the bus voltage produced by boosting through the PFC circuits. The default value is 390 V. The bus voltage specified here may differ from the actual measured value due to the variation between individual bus voltage detectors. To correct for errors of this type, the user should manually adjust the gain or offset of bus voltage detection.

PFC_OUT_MAX_POWER
This macro is used to specify the maximum power (W) that can be output by the PFC circuits when the input voltage is 200 VAC. This value is preset in accordance with the MCI-HV-1 characteristics, so do not modify it.

PFC_OUT_MIN_POWER
This macro is used to specify the minimum power (W) that can be output by the PFC circuits when the input voltage is 100 VAC. This value is preset in accordance with the MCI-HV-1 characteristics, so do not modify it.

PFC_OUT_POWER_COEF
This is a coefficient for power adjustment. Specify a margin to compensate for the loss of the circuits within the range from 1.0 to 1.4.

(3) Voltage and current control systems

Specify the gains of the AVR (voltage regulator) and ACR (current regulator) used inside the PFC controller and some related parameters. Since these values affect PFC control, take care in considering and correctly designing them. The following tables and passages describe how to calculate them.

Table 10-35 Parameters of PFC Settings in r_pfc_cfg.h

Macro Name	Setting	Unit	Description
PFC_AVR_KP	32.9	-	AVR proportional gain
PFC_AVR_KI	0.003	-	AVR integral gain
PFC_AVR_LIMIT	500.0	W	AVR output limit
PFC_ACR_KP	0.019	-	ACR proportional gain
PFC_ACR_KI	0.003	-	ACR integral gain
PFC_ACR_LIMIT	1.0	-	ACR output limit
PFC_AVR_FF_COMP_MIN_LIMIT	10.0	Vrms	Lower limit on the RMS input voltage in AVR feedforward compensation
PFC_ACR_FF_COMP_MIN_LIMIT	10.0	V	Lower limit on the bus voltage in ACR feedforward compensation
PFC_ACR_DUTY_FF_CTRL_TIME	0.1	s	Soft-start time for duty feedforward compensation
PFC_ACR_DUTY_FF_COMP_COEF	0.85	-	Compensation coefficient for duty feedforward compensation
PFC_DUTY_MAX	0.968	-	Maximum duty cycle
PFC_DUTY_MIN	0.003	-	Minimum duty cycle

PFC_AVR_KP
Specify the proportional gain of the voltage regulator to be used to follow the bus voltage, that is, the PFC output voltage. Calculate it by using the following constants. Select design values that suit the inverter circuits to be used.

Table 10-36 Constants from which the Proportional and Integral Gains of AVR and ACR are to be Calculated

Constant	Variable Name	Design Value	Unit
Input AC voltage	Vin	100	Vrms
Output bus voltage	Vout	390	Vdc
Output power	Pout	500	W
Switching frequency	Fsw	32	kHz
Internal capacitance	C	1120	μF

Internal inductance	L	400	μH

When the natural frequency Fv of the AVR is 12 Hz and Kg is 0.01, Kp is calculated as follows.

$$AVR\ K_p = \frac{2\pi F_v C}{K_g (V_{in}/V_{out})} = \frac{2\pi \times 12 \times 0.00112}{0.01 \times (100/390)} = 32.9$$

PFC_AVR_KI
Specify the integral gain of the voltage regulator to be used to follow the bus voltage, that is, the PFC output voltage. Use the constants previously listed in Table 10-36. Let AVR natural frequency Fv = 12 Hz, Kg = 0.01, and Ts = 1/Fsw = 31.25 μs. Ro is determined as follows.

$$R_o = V_{out} \times \frac{V_{out}}{P_{out}} = 390 \times \frac{390}{500} = 304.2\ [Ohm]$$

In this case, Ki is calculated by the following equation.

$$AVR\ K_i = \frac{2\pi F_v T_s}{K_g R_o (V_{in}/V_{out})} = \frac{2\pi \times 12 \times 0.00003125}{0.01 \times 304.2 \times (100/390)} = 0.003$$

PFC_AVR_LIMIT
Specify the limit on the output from the voltage regulator to be used to follow the bus voltage, that is, the PFC output voltage.

PFC_ACR_KP
Specify the proportional gain of the current regulator to be used to follow the input AC current. Use the constants previously listed in Table 10-36. Let the natural frequency Fc = 1500 Hz and Kg = 1.0. Fc should be set to around 1/20 of the carrier frequency so that a sufficient number of samples can be obtained with respect to the sampling frequency (carrier frequency) for current detection and that the input AC current can be followed. Here, Kp is calculated as follows.

$$ACR\ K_p = \frac{2\pi F_c L}{K_g V_{out}} = \frac{2\pi \times 12 \times 0.0004}{0.01 \times 390} = 0.019$$

PFC_ACR_KI
Specify the integral gain of the current regulator to be used to follow the input AC current. Use the constants previously listed in Table 10-36. Let the natural frequency Fc = 1500 Hz and Kg = 1.0 in the same way as PFC_ACR_KP. In addition, let Ts = 1/Fsw = 31.25 μs. Here, Ki is calculated as follows.

$$ACR\ K_i = \frac{2\pi F_c L}{K_g V_{out}} T_s = \frac{2\pi \times 12 \times 0.0004}{0.01 \times 390} \times 0.00003125 = 0.00284$$

PFC_ACR_LIMIT
Specify the limit on the output from the current regulator to be used to follow the input AC current.

PFC_AVR_FF_COMP_MIN_LIMIT
Specify an RMS input voltage value (Vrms) as the lower limit on feedforward compensation in the voltage regulator.

PFC_ACR_FF_COMP_MIN_LIMIT
Specify a bus voltage value (V) as the lower limit on feedforward compensation in the current regulator.

PFC_ACR_DUTY_FF_CTRL_TIME
Specify the soft-start time (s) for duty-cycle feedforward compensation.

PFC_ACR_DUTY_FF_COMP_COEF
Specify the compensation coefficient for duty-cycle feedforward compensation. Specify a value from 0.0 to 1.0.

PFC_DUTY_MAX
Specify the maximum duty cycle of the PWM output under PFC control. Specify a value from 0.0 to 1.0.

PFC_DUTY_MIN
Specify the minimum duty cycle of the PWM output under PFC control. A high value for the minimum duty cycle may boost the bus voltage if the load is very small. Specify a value from 0.0 to 1.0.

(4) Relay control

The following parameters are used to specify the conditions for operating the relay that prevents an inrush current. Specify values that suit the target environment.

Table 10-37 Parameters of PFC Settings in r_pfc_cfg.h

Macro Name	Setting	Unit	Description
RELAY_ON_VAC_RMS_MIN	78.0	Vrms	RMS input voltage at which the relay is turned on
RELAY_OFF_VAC_RMS_MIN	70.0	Vrms	RMS input voltage at which the relay is turned off
RELAY_ON_DIV_MIN	10.0	V	Voltage difference at which the relay is turned on
RELAY_OFF_DIV_MAX	100.0	V	Voltage difference at which the relay is turned off
RELAY_ON_DELAY_TIME	0.1	s	Waiting time until the relay is turned on
RELAY_OFF_DELAY_TIME	0.03	s	Waiting time until the relay is turned off

RELAY_ON_VAC_RMS_MIN
Specify the minimum RMS value (Vrms) of the input AC voltage at which the relay is turned on.

RELAY_OFF_VAC_RMS_MIN
Specify the minimum RMS value (Vrms) of the input AC voltage at which the relay is turned off.

RELAY_ON_DIV_MIN
Specify the minimum width (V) of voltage fluctuations in response to which the relay is turned on.

RELAY_ON_DIV_MAX
Specify the maximum width (V) of voltage fluctuations in response to which the relay is turned off.

RELAY_ON_DELAY_TIME
Specify the delay (s) until the relay is turned on after the conditions for turning on the relay are satisfied while the relay is off.

RELAY_OFF_DELAY_TIME
Specify the delay (s) until the relay is turned off after the conditions for turning off the relay are satisfied while the relay is on.

(5) Protection

The following parameters are used to specify the PFC protection functions.

Table 10-38 Parameters of PFC Settings in r_pfc_cfg.h

Macro Name	Setting	Unit	Description
VAC_OVER_VOLTAGE	388.0	Vpeak	Input AC voltage to be detected as an overvoltage
VDC_OVER_VOLTAGE	450.0	V	Bus voltage to be detected as an overvoltage
VDC_UNDER_VOLTAGE	80.0	V	Bus voltage to be detected as a low voltage
ICHP_OVER_CURRENT	19.0	Apeak	PFC control current to be detected as an overcurrent
PFC_VAC_DIP_DETECT_LEVEL	70.0	Vrms	Detection level for input voltage dip

VAC_OVER_VOLTAGE
Specify the single-phase input AC voltage (Vpeak) to be detected as an overvoltage. That is, this parameter specifies the allowable peak voltage. The specified value should usually be no greater than the maximum input voltage for the inverter.

VDC_OVER_VOLTAGE
Specify the bus voltage (V), that is, the PFC output voltage, to be detected as an overvoltage. The specified value should usually be no greater than the maximum input voltage for the inverter.

VDC_UNDER_VOLTAGE
Specify the bus voltage (V), that is, the PFC output voltage, to be detected as a low voltage.

ICHP_OVER_CURRENT
Specify the PFC control current (Apeak) to be detected as an overcurrent. That is, this parameter specifies the allowable peak current.

PFC_VAC_DIP_DETECT_LEVEL
Specify the detection level (Vrms) of input voltage dip.

(6) Parameters related to the detection filters

The following parameters are used for settings of the detection filters for PFC control

Table 10-39 Parameters of PFC Settings in r_pfc_cfg.h

Macro Name	Setting	Unit	Description
VDC_NOTCH_FILTER_D	0.01	-	Depth of the notch filter
VDC_NOTCH_FILTER_ZETA	0.05	-	Bandwidth of the notch filter
VAC_LPF_CUT_FREQ	2000.0	Hz	Cutoff frequency of the LPF for the AC voltage
VDC_LPF_CUT_FREQ	800.0	Hz	Cutoff frequency of the LPF for the bus voltage
ICHP_LPF_CUT_FREQ	0.0	Hz	Cutoff frequency of the LPF for the PFC current

VDC_NOTCH_FILTER_D
Specify the depth of the notch filter. The value specified here can be converted to a depth in dB by the following equation. A depth setting of 0.01 corresponds to -40 db.

$$D[db] = 20\log_{10} VDC\_NOTCH\_FILTER\_D$$

VDC_NOTCH_FILTER_ZETA
Specify the bandwidth of the notch filter. This value is a coefficient that represents half the ratio of the bandwidth Bw (Hz) to the notch filter frequency Fn (Hz). When the notch filter frequency is 50 Hz and the bandwidth is 5 Hz, the value is 0.05.

$$\xi = \frac{B_w}{2F_n}$$

VAC_LPF_CUT_FREQ
Specify the cutoff frequency of the LPF to be used in detection of the input AC voltage in PFC control. This value must correspond to a sufficiently high frequency; specifically, the set frequency should be no lower than ten times the frequency of the input AC voltage.

VDC_LPF_CUT_FREQ
Specify the cutoff frequency of the LPF to be used in detection of the inverter bus voltage in PFC control. This setting is only used to control the AVR; it is not used for bus voltage detection in motor control.

ICHP_LPF_CUT_FREQ
Specify the cutoff frequency of the LPF to be used in detection of the PFC current.

11. Settings for the FSP

11.1 Overview of the FSP

Figure 11-1 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 PFC 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.

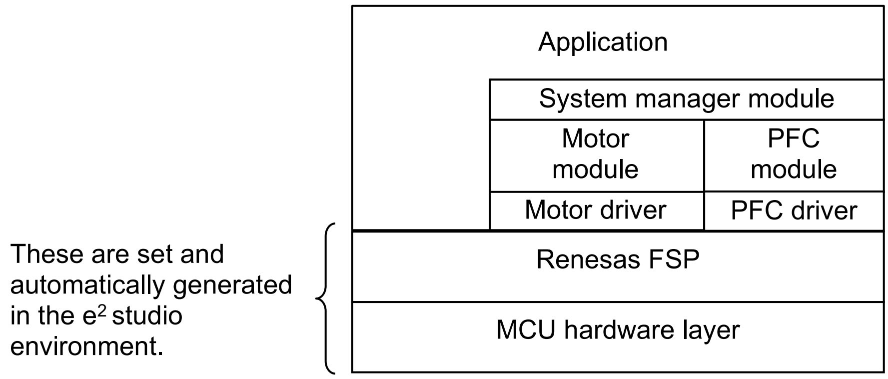


Figure 11-1 Software Architecture of This Sample Program

11.2 Setting FSP Stacks

The FSP provides functional modules for each peripheral function, which are referred to as stacks. Table 11-1 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.

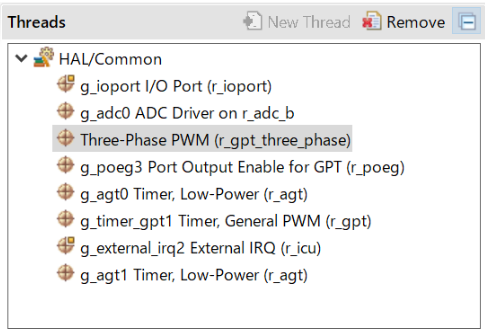


Figure 11-2 List of FSP Stacks

Table 11-1 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)
A/D conversion for PFC (detection of the inverter bus voltage, detection of the PFC input voltage and current)	g_adc0 ADC Driver on r_adc_b (adc0, sub group1)
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)
Interrupt for the system manager (1-ms intervals)	g_agt1 Timer, Low-Power (r_agt)
PWM control in the PFC circuit	g_timer_gpt1 Timer, General PWM (r_gpt)
External interrupt (IRQ2)	g_external_irq2 External IRQ (r_icu)
Overcurrent detection	g_poeg3 Port Output Enable for GPT (r_poeg)

11.3 Callback Interrupts

The FSP defines callback functions as functions to be called for the interrupt processing. Table 11-2 lists the interrupts.

The files which handle the actual processing for the callback functions in the table are in src/application/main/mtr_main.c.

Table 11-2 List of Interrupts

FSP Stack	Callback Function	Description
g_adc0	callback_gpt_adc_cyclic()	This function is for use in both 32-kHz-periodic PFC control and 4-kHz-periodic motor current control. The function separates these two tasks by internally masking one or the other.
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_agt_motor_speed_cyclic()	
agt1	callback_agt_system_manager_cyclic()	
irq2	callback_irq2_pfc_error()	

11.4 Pin Settings

Table 11-3 lists the information on pin interfaces.

Table 11-3 Pin Interfaces

Function	**Pin Name**	**Peripheral Function**	**Pin to Which the Function is Allocated**	**Remarks**
LED1	PD01	GPIO	-	These allow use of the LEDs on the CPU board by the user.
LED2	PD02	GPIO	-	
LED3	PD03	GPIO	-	
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 input AC voltage for PFC	PB10	S12AD	AN028	
Measurement of the current for PFC	PE15	S12AD	AN027	
Measurement of the inverter bus voltage for use in control over PFC and the motor	PA06	S12AD	AN006	
Abnormal inverter temperature	PD07	GPIO	-	The low level indicates the abnormal state.
PFC overcurrent	P001	IRQ	IRQ2	A falling edge of the signal on the pin indicates the abnormal state.
PFC PWM output	PB14	GPT	GTIOC1A	
PWM emergency stop input in response to an overcurrent being detected	PC13	POEG	GTETRGD	The low level indicates the 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	-	

11.5 GPT Settings for PFC

The channel 1 GPT is used in PFC control by the FSP. The main settings are listed in the table below. The PWM carrier frequency is set to 32 kHz (control period: 31.25 μs).

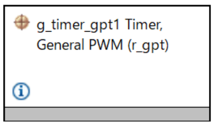


Figure 11-3 GPT Stack for PFC

Table 11-4 GPT Settings for PFC

Function and Item for Setting				**Setting**
Module	General	Name		g_timer_gpt1
		Channel		1
		Mode		Triangle-Wave PWM (symmetric, Mode1)
		Period		31250
		Period Unit		Nanoseconds
	Output	Custom Waveform		Enabled
		Custom Waveform/ GTIOA	Initial Output Level	Pin Level High
			Cycle End Output Level	Pin Level Retain
			Compare Match Output Level	Pin Level Toggle
			Retain Output Level at Count Stop	Disabled
		Custom Waveform/ GTIOB	Initial Output Level	Pin Level Low
			Cycle End Output Level	Pin Level Retain
			Compare Match Output Level	Pin Level Toggle
			Retain Output Level at Count Stop	Disabled
		Duty Cycle Percent (only applicable in PWM mode)		50
		GTIOCA Output Enabled		False
		GTIOCA Stop Level		Pin Level Low
		GTIOCB Output Enabled		False
		GTIOCB Stop Level		Pin Level Low
	Input			Not in use
	Interrupts			Not in use
	Extra Features	Output Disable		Not in use
		ADC Trigger/ Start Event Trigger		Trigger Event A/D Converter Start Request A During Down Counting
		Dead Time		Not in use
		ADC Trigger (Channels with GTADTRA only)		ADC A Compare Match (Raw Counts) = 0
		ADC Trigger (Channels with GTADTRB only)		ADC B Compare Match (Raw Counts) = 0
		Interrupt Skipping (Channels with GTITC only)		Not in use
		Extra Features		Enabled
Pins	GTIOC1A			PB14
	GTIOC1B			None

11.6 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 v5.4.0 does not support it.

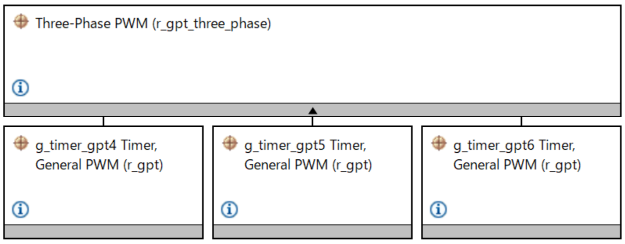


Figure 11-4 Stacks Related to the Three-Phase PWM GPT Stack

Table 11-5 Three-Phase PWM Settings

Function and Item for Setting			Setting
General	Name		g_three_phase0
	Mode		Triangle-Wave Symmetric PWM
	Period		250
	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)	240
		Dead Time Count Down (Raw Counts)	240

Table 11-6 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 11-7 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 11-8 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

11.7 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.

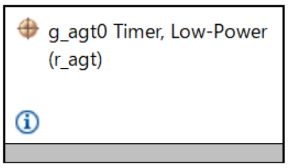


Figure 11-5 AGT0 Stack

Table 11-9 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_agt_motor_speed_cyclic
	Underflow Interrupt Priority		Priority 9
Pins	AGTEED		<unavailable>
	AGTIO0		<unavailable>
	AGTO0		<unavailable>
	AGTOA0		<unavailable>
	AGTOB0		<unavailable>

11.8 AGT1 Settings (Setting the Interval for the Activation of Control by the System Manager)

The AGT1 is used to set the interval for the activation of control by the system manager. This is generally set for 1-ms periodic operation.

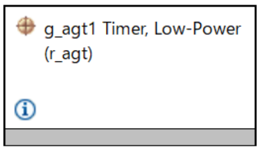


Figure 11-6 AGT1 Stack

Table 11-10 AGT1 Settings for the Interval for the Activation of Control by the System Manager

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_agt_system_manager_cyclic
	Underflow Interrupt Priority		Priority 10
Pins	AGTEED		<unavailable>
	AGTIO0		<unavailable>
	AGTO0		<unavailable>
	AGTOA0		<unavailable>
	AGTOB0		<unavailable>

11.9 ADC Settings

The 12-bit A/D converters in the MCU are used to measure the U-, V-, and W-phase output currents, PFC current, input AC voltage, and inverter bus voltage. Table 11-11 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.

Since the MCI-HV-1 inverter employs the current detection method with the use of a shunt resistor, 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. In addition, the A/D conversion end interrupt is generated after completion of conversion following the start of A/D detection. In PFC control, the PFC current, AC voltage, and inverter bus voltage are detected at intervals of 32 kHz, after which A/D conversion end interrupts are generated in the same way as in motor current control. Since the callback function to be used for the interrupt processing is common to these two types of control, the function identifies the two different intervals by checking which of PFC control or motor current control is currently masked within the function.

Table 11-11 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 PFC current	ADC0 channel 27	
Measurement of the input AC voltage	ADC0 channel 28	
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	

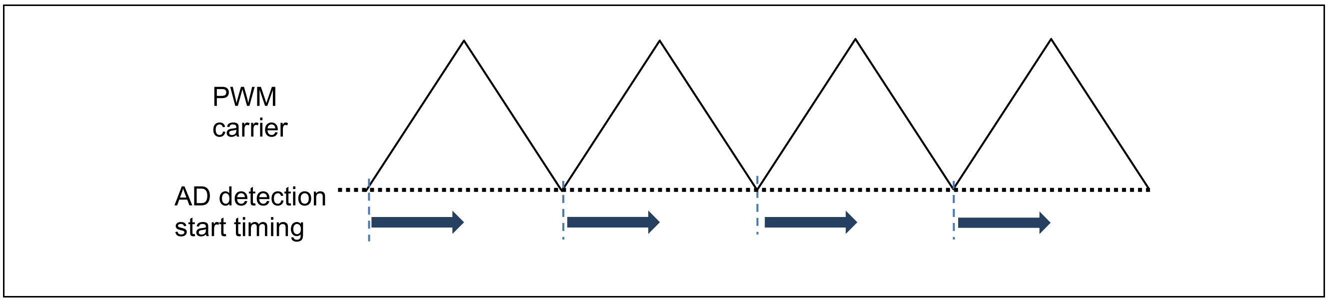


Figure 11-7 A/D Detection Start Timing (Trough-Sensed Interrupt)

Table 11-12 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	100

Function and Item for Setting			Setting
		Period Cycle	
	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_gpt_adc_cyclic
Digital Filter			Not in use (by default)
Sample and Hold	Enable Unit	Unit 0	☑
		Unit 1	☑
		Unit 2	☑
		Units 4 to 6	☐
	Analog Channels 0 to 5	Sampling Time	120
		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	**Scan Group 0**
		Channel Select	**AN000**
		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	**Scan Group 0**
		Channel Select	**AN002**
		Sampling State Table ID	Sampling State Entry 0

Function and Item for Setting			Setting
		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	**Scan Group 0**
		Channel Select	**AN004**
		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 3	Scan Group	**Scan Group 1**
		Channel Select	**AN027**
		Sampling State Table ID	Sampling State Entry 2
		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 4	Scan Group	**Scan Group 1**
		Channel Select	**AN028**
		Sampling State Table ID	Sampling State Entry 2
		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 5	Scan Group	**Scan Group 1**
		Channel Select	**AN006**
		Sampling State Table ID	Sampling State Entry 1
		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

Function and Item for Setting				**Setting**
	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	☐
		External Trigger Enable	External Trigger Input 1 (ADTRG1) Enable	☐
		ELC Trigger Enable		Not in use
		GPT Trigger Enable	GPT Channel 0 Request A	☐
		GPT Trigger Enable	GPT Channel 1 Request A	☐
		GPT Trigger Enable	GPT Channel 2 Request A	☐
		GPT Trigger Enable	GPT Channel 3 Request A	☐
		GPT Trigger Enable	GPT Channel 4 Request A	☑
		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	☐
		External Trigger Enable	External Trigger Input 1 (ADTRG1) Enable	☐
		ELC Trigger Enable		Not in use
		GPT Trigger Enable	GPT Channel 0 Request A	☐
		GPT Trigger Enable	GPT Channel 1 Request A	☑
		GPT Trigger Enable	GPT Channel 2 Request A	☐
		GPT Trigger Enable	GPT Channel 3 Request A	☐
		GPT Trigger Enable	GPT Channel 4 Request A	☐
		GPT Trigger Enable	GPT Channels 5 to 9 Request A/B	Not in use
		Enable		Enable
		Converter Selection		**ADC 1**
		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

11.10 IRQ Settings for PFC Overcurrents

The MCI-HV-1 circuit has a function for hardware detection of PFC overcurrents and generates an active-low signal when that error condition is satisfied. This sample program uses the external interrupt function (IRQ) to detect a PFC overcurrent error in response to generation of the active-low signal and executes the callback_irq2_pfc_error() callback function.

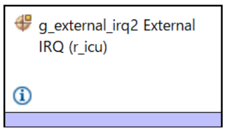


Figure 11-8 IRQ Stack

Table 11-13 IRQ2 Settings Related to an External Interrupt Due to a PFC Overcurrent

Function and Item for Setting			Setting
	Name		g_external_irq2
	Channel		2
	Trigger		Falling
	Digital Filtering		Enabled
	Digital Filtering Sample Clock		PCLK / 64
	Callback		callback_irq2_pfc_error
	Pin Interrupt Priority		Priority 0 (highest)

11.11 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 11-14 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 11-14 POEG Settings

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

12. Results of Evaluation

12.1 Evaluation of PFC Control

We have confirmed that the voltage rises up to DC 390 V under the condition that AC 200 V at 50 Hz is being supplied.

Figure 12-1 Example of Waveforms in the State Where the Voltage is Rising Up to 390 V under PFC Control

12.2 Evaluation of Motor Control

12.2.1 Magnetic Pole Position Estimation Accuracy

Under the evaluation environment with a 12-bit resolution angle sensor attached to the motor shaft, we have confirmed that the magnetic pole position estimation accuracy in the stopped state is within ±10° compared to the angle sensor.

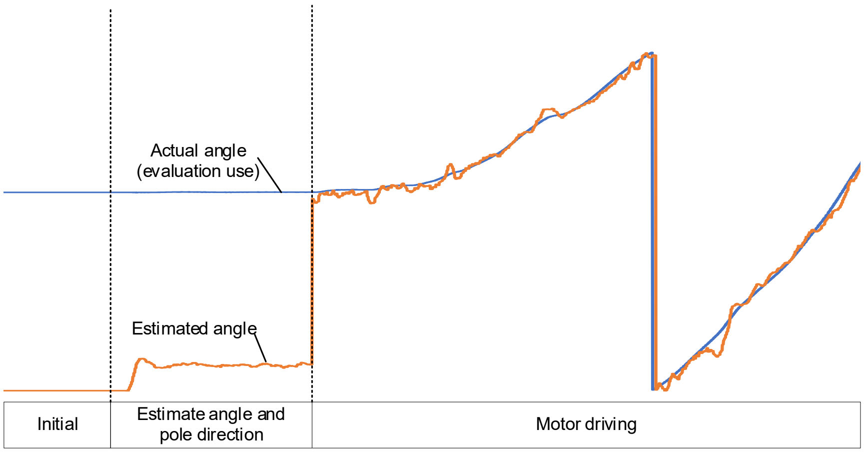


Figure 12-2 Example of Angular Waveform for Magnetic Pole Position Estimation

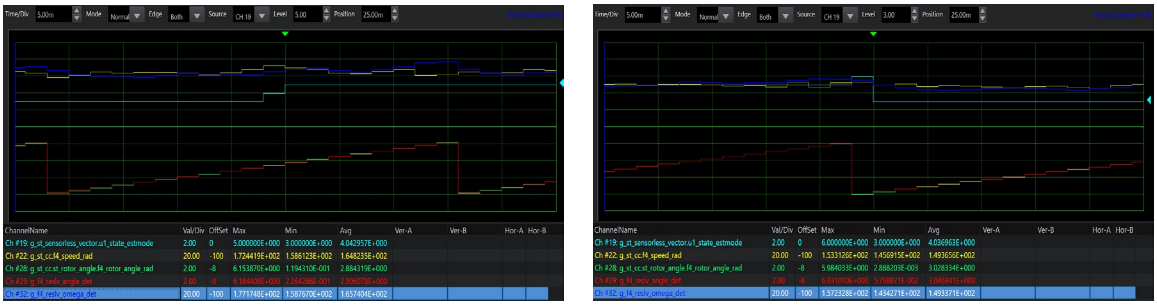
12.2.2 Starting Characteristics

We have confirmed that the motor is accelerated up to the specified speed of 1000 rpm after position estimation has proceeded with the motor stopped at the time of start-up.

Figure 12-3 Waveforms Indicating the Characteristics during Start-up under Sensorless Control

12.2.3 Control Switching Characteristics

We have confirmed the characteristics during switching of the method of estimating the magnetic pole position between high-frequency pulse voltage injection (HFI) for the standstill state and low-speed range and the BEMF observer for the medium-to-high-speed range.



Low-speed range (HFI) -> Medium-to-high-speed range (BEMF) Medium-to-high-speed range (BEMF) -> Low-speed range (HFI)

Figure 12-4 Waveforms during Switching the Sensorless Vector Control Method between High-frequency Pulse Voltage Injection (HFI) and the BEMF Observer

12.2.4 Acceleration/Deceleration Characteristics

The waveforms indicating the characteristics in acceleration from the speed of 200 rpm to 800 rpm and in deceleration from 800 rpm to 200 rpm are shown in the figures below. As shown in the figures, both acceleration and deceleration proceeded correctly under sensorless vector control.

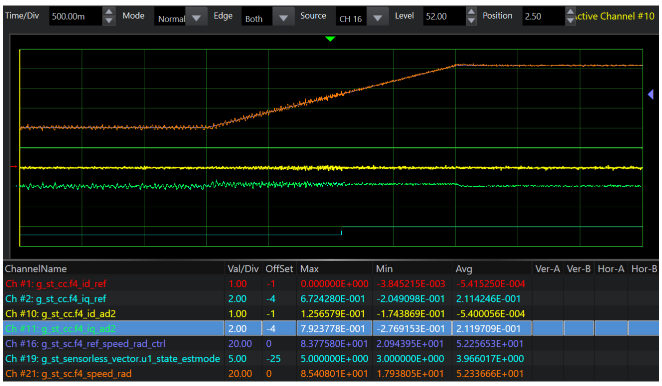


Figure 12-5 Acceleration Characteristics

Figure 12-6 Deceleration Characteristics

12.2.5 High-Speed Operation Characteristics

We show an example waveform when operating the motor under flux weakening control in the speed range of 3000 to 4000 rpm.

Figure 12-7 Example of Waveforms during Operation under Flux Weakening Control

12.2.6 Load Characteristics

At a PWM carrier frequency of 4 kHz, we show an example waveform when a rated load of 1500 W using MTPA is performed during rotation at the rated speed of 3000 rpm.

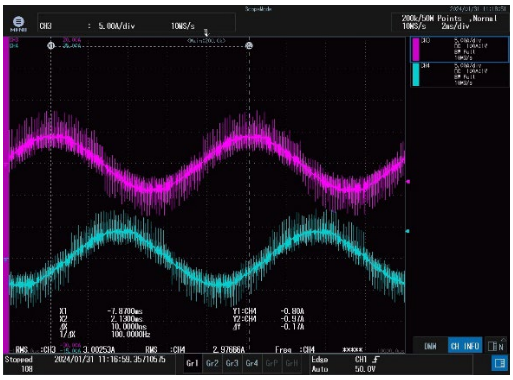


Figure 12-8 Example of Output current waveforms during Rated Load Operation

12.2.7 Evaluation of Operation in Flying Start Mode

We have confirmed stopping by active braking and starting of the motor at around 300 rpm while the motor is being decelerated in the free-running state. We have also confirmed restarting of the motor in the free-running state at around 700 rpm by using the flying start function.

Figure 12-9 Current Waveforms during Operation in Flying Start Mode (Left: Active Braking; Right: Flying Start)

12.2.8 Evaluation of the Step-Skipping (Stall) Detection Function

We have confirmed that the step-skipping (stall) detection function can be used to detect the pseudo-stalled state, which was generated by giving the Lq motor parameter an incorrect value, approximately two seconds after its generation during operation with the use of the BEMF observer at a speed of 1000 rpm. We have also confirmed that disabling the step-skipping (stall) detection function results in the motor continuing to stall for approximately eight seconds until an overcurrent error occurs following the generation of the pseudo-stalled state.

Figure 12-10 Example of Current Waveforms on Detection of the Step-Skipping (Stalled) State

12.3 CPU Utilization

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

Table 12-1 Control Loops and CPU Loading Rates

Control Loop Type	**Control Interval**	**Processing Time**	**CPU Loading Rate**
PFC control loop	31.125 μs	10.0 μs	32.0%
Current control loop in motor control	250 μs (no decimation)	33.5 μs	13.4%
Speed control loop in motor control	500 μs	4.8 μs	1.0%

12.4 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 12-2 Program Size and RAM Usage

Program size (ROM)	40600 bytes
RAM usage	6880 bytes
Maximum value of stack analysis results	428 bytes
Stack size setting in the IDE	1024 bytes

13. FAQ

Table 13-1 lists typical problems and examples of their solutions.

Table 13-1 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^2 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.
Application of the flying start function terminated operation of the motor.	When the flying start function is in use, the program uses active braking to forcibly terminate operation of the motor if it judges the speed of rotation to be below the specified value since the inductive voltage is not high enough for the normal estimation of speed and angle. To avoid this, review the specified speed.
Application of the flying start function generated an overcurrent error.	Appropriately design and set the value of threshold current for the flying start function, taking into consideration the effects of the motor parameters and the impedance of the wiring.
Step-skipping (stall) detection does not work.	Since the step-skipping (stall) detection function is complementary to the overcurrent protection function, an overcurrent error may be generated before the stalled state is detected. In addition, the step-skipping (stalled) state is not detected if a fluctuation in current which would normally have been generated in that state has not been generated. Consider the use of the step-skipping (stall) detection function in its combination with the overcurrent protection function by appropriately designing the threshold value for use in detecting an overcurrent error.
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.
Attempted application of the torque vibration suppression function did not have its desired effect.	The generation of vibration depends on the characteristics, structure, and combinations of the peripheral machine parts, their junctions, vibration control components, etc., as well as on the motor and compressor. The effectiveness of using the torque vibration suppression function greatly differs according to the relationship between the rotation speed of the motor and these elements of the mechanism. Apply countermeasures such as reviewing the structure, selection of the elements, and rotation speed to be used to empirically derive the conditions for a suitable solution.
I was unable to switch the control method to sensorless vector.	When a surface permanent magnet (SPM) motor is in use, enabling the MTPA function leads to incorrect operation of the software for controlling the motor. The MTPA function is only usable with IPM motors, so be sure to disable it when an SPM motor is in use.
When the motor is operating, it makes a beeping sound.	A high-frequency pulse voltage in the audible range is applied to estimate the magnetic pole position. This high-frequency sound is emitted from the motor. It is not abnormal.

Problem	Example of Solution
When the rotation speed of the motor exceeds a certain speed, the beeping sound stops.	A high-frequency pulse is applied to estimate the magnetic pole position in the low-speed range, but when the set speed is reached, it automatically switches to magnetic pole position estimation by the BEMF observer for the medium-to-high-speed range, and the beeping sound stops.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Nov. 29, 2024	—	First edition issued

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

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

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

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

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

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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