Sensorless Vector Control of PM Motor

For Renesas Flexible Motor Control Series, Equipped with Flying Start

Introduction

This sample program offers the following control algorithms for the Renesas Flexible Motor Control Kit with RA Family and RX26T CPU boards. These algorithms are mainly for implementing a sensorless vector control function for permanent magnet motors (PM) with additional functions for use in industrial and home appliances.

- Starting with open-loop control from the standstill state or during low-speed operation (current-drawn control)
- Sensorless vector control of a PM motor through a BEMF observer during medium-to-high-speed operation
- Flux weakening control and maximum torque per current control (maximum torque per ampere, MTPA)[1]
- Torque vibration suppression, step-skipping (stall) detection, and flying start (pick-up control)

This application note describes how to set up and use the combination of the sample program and the inverter. The specifications of the internal program are described to assist users in evaluating Renesas MCUs and semiconductor devices in inverter development projects. Figure 1-1 shows the hardware configuration to use with this sample program.

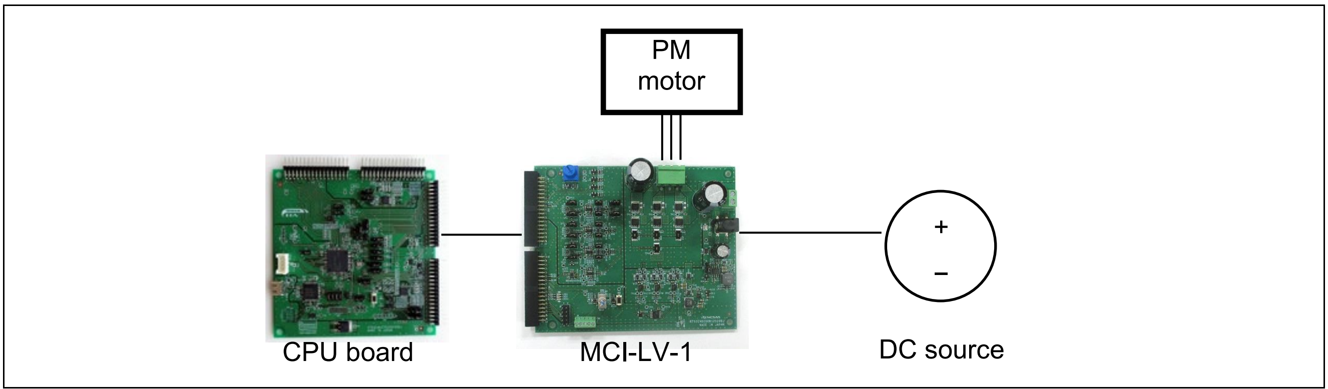


Figure 1-1 Hardware Configuration

The sample program provided with this application note is only for the purpose of evaluation use. Renesas Electronics Corporation does not guarantee the operations of this sample program. Users are advised to conduct thorough evaluation in an appropriate environment before using this sample program.

Target Devices

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

- RA6T2 (R7FA6T2BD3CFP)
- RA8T1 (R7FA8T1AHECBD)
- RA8T2 (R7KA8T2LFLCAC)
- RX26T 64-Kbyte RAM version (R5F526TFCDFP)

[1] The MTPA function is only applicable to an IPMSM (interior permanent magnet synchronous motor). It cannot be used with an SPMSM (surface permanent magnet synchronous motor).

Contents

1. Introduction

This application note is intended to explain how to use the sample program that employs RA Family or RX26T microcontroller (MCU) manufactured by Renesas, to drive a permanent magnet synchronous motor with sensorless vector control with the MCI-LV-1 inverter board. It is also intended to describe the configuration, specifications, and method of control by the software.

Refer to the MCI-LV-1 User's Manual (R12UZ0092) to learn the set-up, usage and operation of the MCI-LV-1 inverter manufactured by Renesas.

This sample program can control R42BLD30L3, a brushless DC motor (Product of 'MOONS') without a sensor by using RA Family CPU board or RX26T CPU board from Renesas and an MCI-LV-1 inverter from Renesas. This sample program supports the Renesas Motor Workbench, a motor control development support tool, which can be used as a user interface (UI) for checking the MCU internal data and controlling the motor. You can use the sample program as a reference to check how MCU functions are assigned, how control is loaded on interrupts, and other information in the sample program when selecting an MCU to be used or developing software.

The sample program described in this application note was developed and evaluated in the environment of the PM motor and inverter described in this document and is not guaranteed to work with your PM 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-LV-1 inverter from Renesas
Motor: R42BLD30L3 (Product of 'MOONS')

Target Software

The following shows the target software for this application note.

- RA6T2_MCB2_MCILV1_PM_LESS_FOC_WFS_E2S_V100 (IDE: e^2 studio)
- RA8T1_MCB_MCILV1_PM_LESS_FOC_WFS_E2S_V100 (IDE: e^2 studio)
- RA8T2_MCB_MCILV1_PM_LESS_FOC_WFS_E2S_V100 (IDE: e^2 studio)
- RX26T_MCBA2_MCILV1_PM_LESS_FOC_WFS_E2S_V100 (IDE: e^2 studio)
- RX26T_MCBA2_MCILV1_PM_LESS_FOC_WFS_CSP_V100 (IDE: CS+)

Reference Documents

- RA6T2 Group User's Manual — Hardware (R01UH0951)
- RA8T1 Group User's Manual — Hardware (R01UH1016)
- RA8T2 Group User's Manual — Hardware (R01UH1067)
- RX26T Group User's Manual — Hardware (R01UH0979)
- RA Flexible Software Package Documentation (Release v6.2.0)
- Smart Configurator User's Manual: RX API Reference (R20UT4360)
- Smart Configurator User Guide: e^2 studio (R20AN0451)
- Smart Configurator User Guide: CS+ (R20AN0470)
- Renesas Motor Workbench User's Manual (R21UZ0004)
- MCB-RA6T2 Version 2 User's Manual (R12UZ0099)
- MCB-RA8T1 User's Manual (R12UZ0134)
- MCB-RA8T2 User's Manual (R12UZ0172)
- MCB-RX26T Type A Version 2 User's Manual (R12UZ0112)
- MCI-LV-1 User's Manual (R12UZ0092)

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

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

Item for Checking	Reference Section
Identify and select necessary devices.	3
Select a power supply.	4.2
Select a motor.	4.3
Select an inverter.	4.5
Check the wiring.	4.10
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.	6.8
Stop the motor.	6.9
Examine the motor control algorithms.	7
Examine the structure of the sample program.	9
Examine and change the inverter parameters.	10.7, 10.4, 11.8, 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.	10.13
Change the MCU settings.	10.2, 11, 12
Check the frequently asked questions. Check the troubleshooting tips.	14

2. Glossary

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

Table 2-1 Glossary

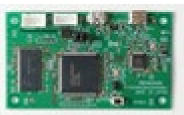
Term	Description
IDE	An integrated development environment such as e[2] studio.
IPM motor	Also called an IPMSM. This type of motor has magnets inside the rotor and is considered superior in terms of efficiency, size, and cost. Such motors also have saliency, in which the Ld and Lq are different.
MC-COM	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
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 called DC intermediate voltage.
Emulator	A device used to program an MCU. Also called an ICE.
Open loop	A motor control technique that does not require positional feedback signals to control the voltage.
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.
FSP	Flexible Software Package, a software framework for embedded system development using Renesas RA MCU.
SC	Smart Configurator, which is a tool for automatically generating a program for initial settings of an MCU, specifically RX MCU.

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 MCB-RA6T2 Version 2 	Renesas	RTK0EMA270C00002BJ *1 MCU product code RA6T2, R7FA6T2BD3CFP
RA8T1 CPU board MCB-RA8T1 	Renesas	RTK0EMA5K0C00000BJ MCU product code RA8T1, R7FA8T1AHECBD
RA8T2 CPU board MCB-RA8T2 	Renesas	RTK0EMA6L0C00000BJ MCU product code RA8T2, R7FA8T2LFLCAC
RX26T CPU board MCB-RX26T Type A Version 2 	Renesas	RTK0EMXE70C00001BJ MCU product code RX26T 64-Kbyte RAM version, R5F526TFCDFP
Inverter board 	Renesas	MCI-LV-1 RTK0EM0000B12020BJ
Isolated communication board MC-COM 	Renesas	Renesas Flexible Motor Control Communication Board RTK0EMXC90S00000BJ
PM motor	MOONS'	R42BLD30L3
DC power supply unit	KIKUSUI ELECTRONICS	PCR1000MS
Torque meter and load system	MAGTROL	DSP6001

Note *1 Please note that RTK0EMA270C00000BJ is not supported.

3.2 List of Software Tools Used

The following tables list the software tools and their versions used in evaluating this sample program. This sample program can be used with the limitations of Renesas development environment e^2 studio and CS+ evaluation edition.

Table 3-2 List of Software Tools used for the RA Family MCU

Manufacturer	Software Tool	Version	Remark
Renesas	e^2 studio	2025-10	Free version
Renesas	FSP	6.2.0	
Renesas	Renesas Motor Workbench	3.2.0	

Table 3-3 List of Software Tools used for the RX26T

Manufacturer	Software Tool	Version	Remark
Renesas	e^2 studio	2025-10	Free version
Renesas	CS+ for CC	V8.14.00	Free version
Renesas	RX Smart Configurator	V2.27.0	
Renesas	CC-RX	V3.07.00	Free evaluation version
Renesas	Renesas Motor Workbench	3.2.0	

4. Configuring a Hardware Environment

4.1 Overview of Hardware Environment

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

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

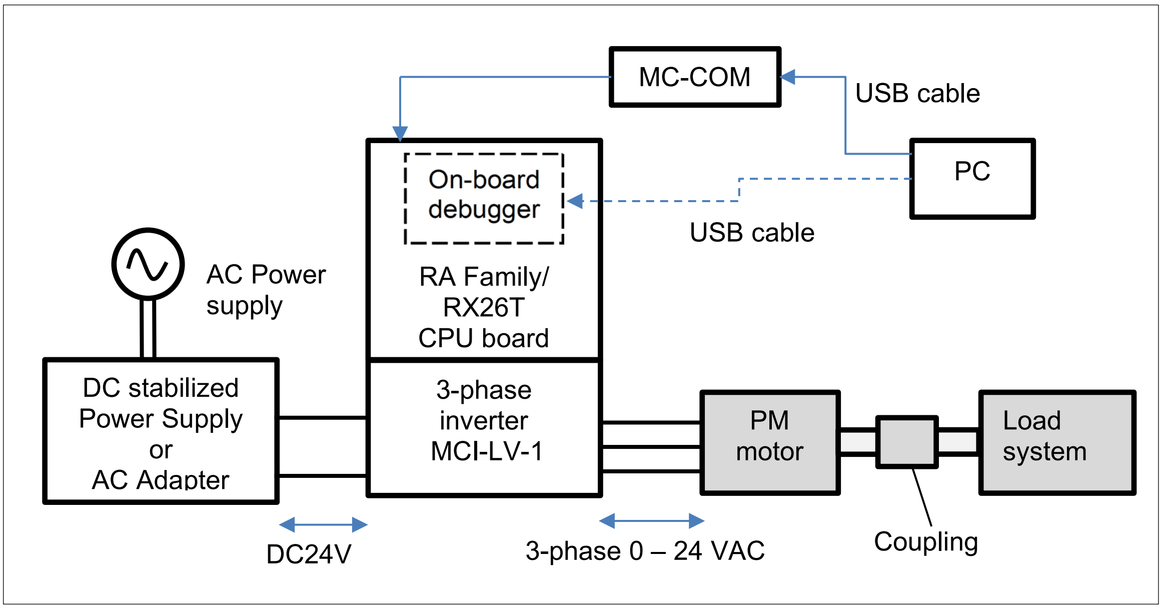


Figure 4-1 Hardware Configuration Example

4.2 Preparing a Power Supply

In this sample program, DC stabilized power supply/AC adapter/a control power supply (capable of output of 24V, 2.5A or more) is used to supply a voltage of 24 VDC to the 3-phase inverter MCI-LV-1.

The voltage supplied to the inverter varies depending on the inductive voltage, rating conditions, maximum load conditions of the motor to be used. Please select an appropriate type of power supply based on your experimental environment and restrictions and conditions of AC power supply to be used.

The inverter introduced here has an output current of 10A max. Please note that if the motor is changed, the rated operation may not be possible due to the rating of the new motor.

4.3 Preparing a Motor

Before connecting the inverter to a motor, obtain the parameters and constants of the PM 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 PM 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 R42BLD30L3 motor from MOONS', 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.

Table 4-1 MOONS' R42BLD30L3 Motor Parameters (Some Values are Based on Our Own Measurements)

Primary resistance R	1.3 Ω
d-axis inductance	1.3 mH
q-axis inductance	1.3 mH
Moment of inertia	0.000003666 kgm^2
Magnetic flux linkage ψ	0.01119 Wb (rms)
Number of pole pairs	4 (8 poles)
Rated speed	4000 rpm
Maximum speed	4500 rpm
Rated torque and maximum torque	0.08 Nm and 0.16 Nm
Rated frequency	266.67 Hz (electrical angle), 66.67 Hz (mechanical angle)
Rated voltage	36 V
Rated current	1.67 Arms
Rated output power	30 W

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.

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-LV-1 inverter board and must be changed if you use another inverter.

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

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

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

Item	Interface components	Function
Rotational position/speed	Volume (VR1)	Rotation speed command value input (analog value)
START/STOP	Toggle switch (SW1)	Motor rotation start/stop command
ERROR RESET	Push Switch (SW2)	Command to recover from an error status (when an error occurs)
LED1	Orange LED	・At the time of motor rotation: ON ・At the time of motor stop: OFF
LED2	Orange LED	・At the time of error detection: ON ・At the time of normal operation: OFF
RESET	Push switch (RESET1)	System reset

4.6 Setting up the RA Family CPU Board

This section describes how to install the RA Family CPU board, which can be plugged in MCI-LV-1 directly. “Inverter board connector (for INV1)” or “Inverter connector” as shown in Figure 4-2 to Figure 4-5 should be connected to MCI-LV-1’s “CPU board connector”.

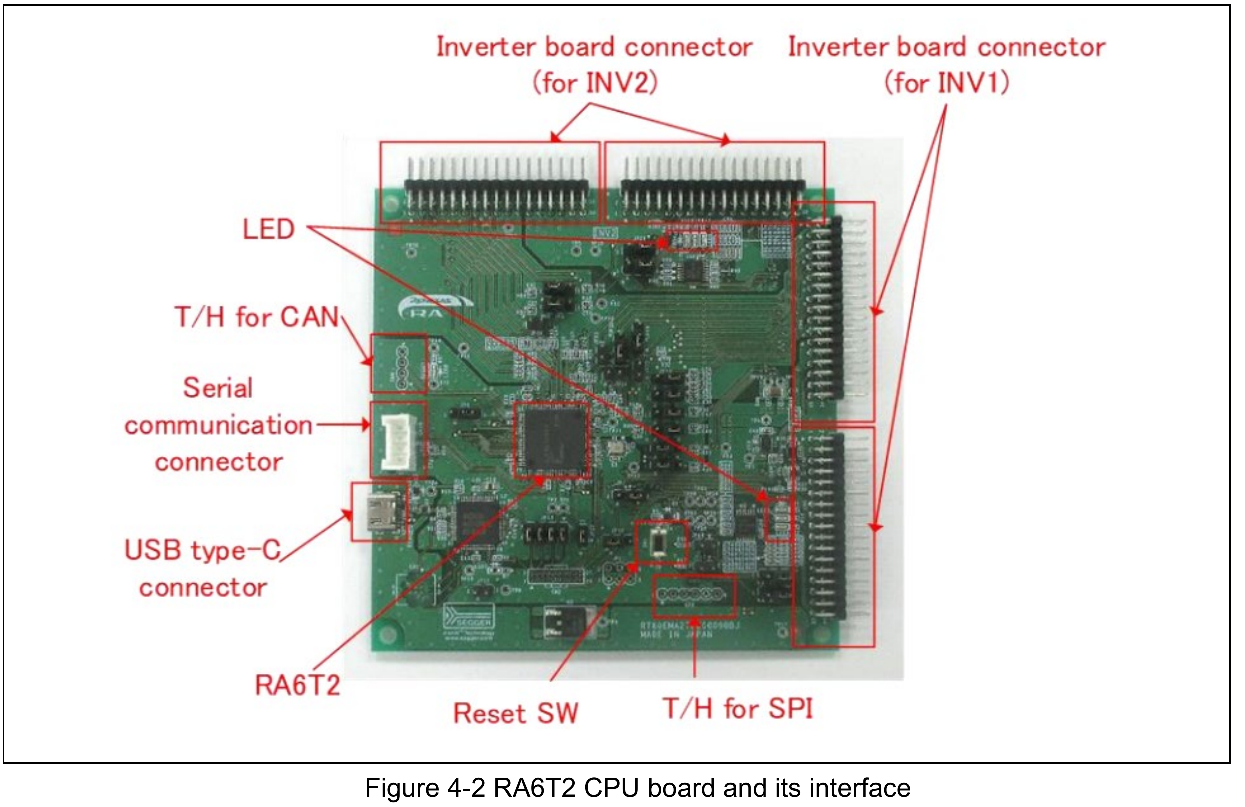


Figure 4-2 RA6T2 CPU board and its interface

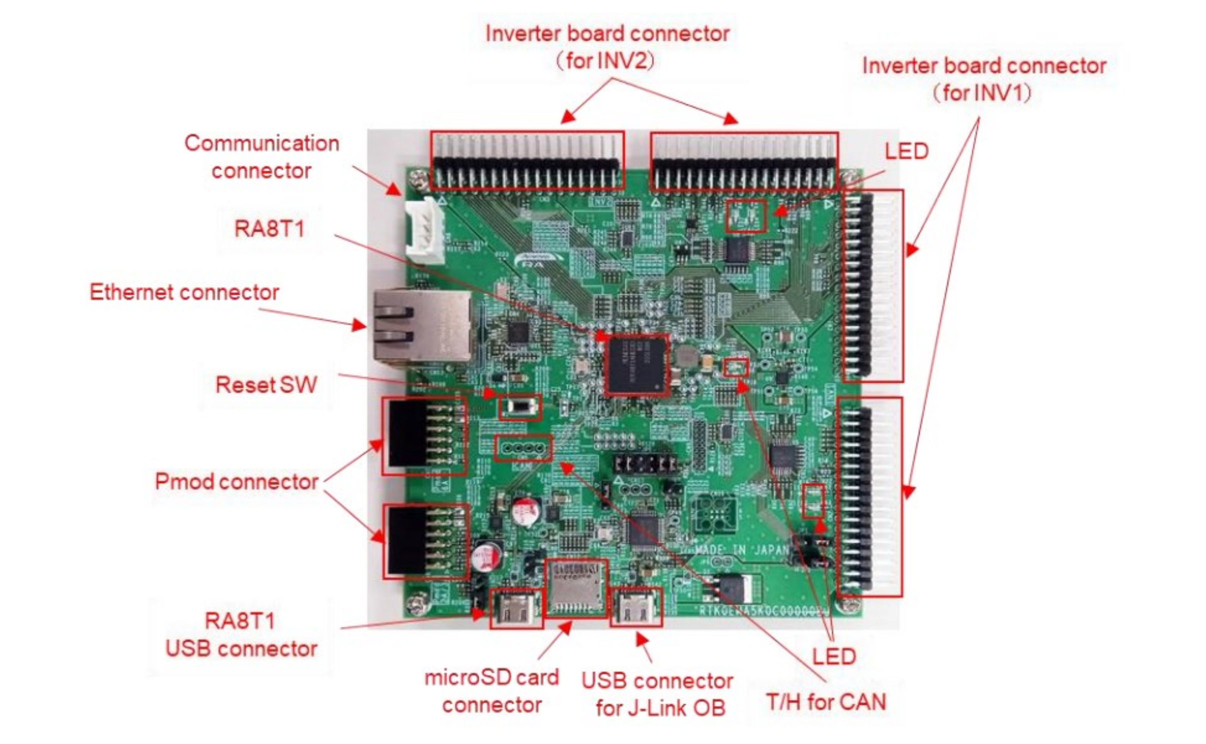


Figure 4-3 RA8T1 CPU board and its interface

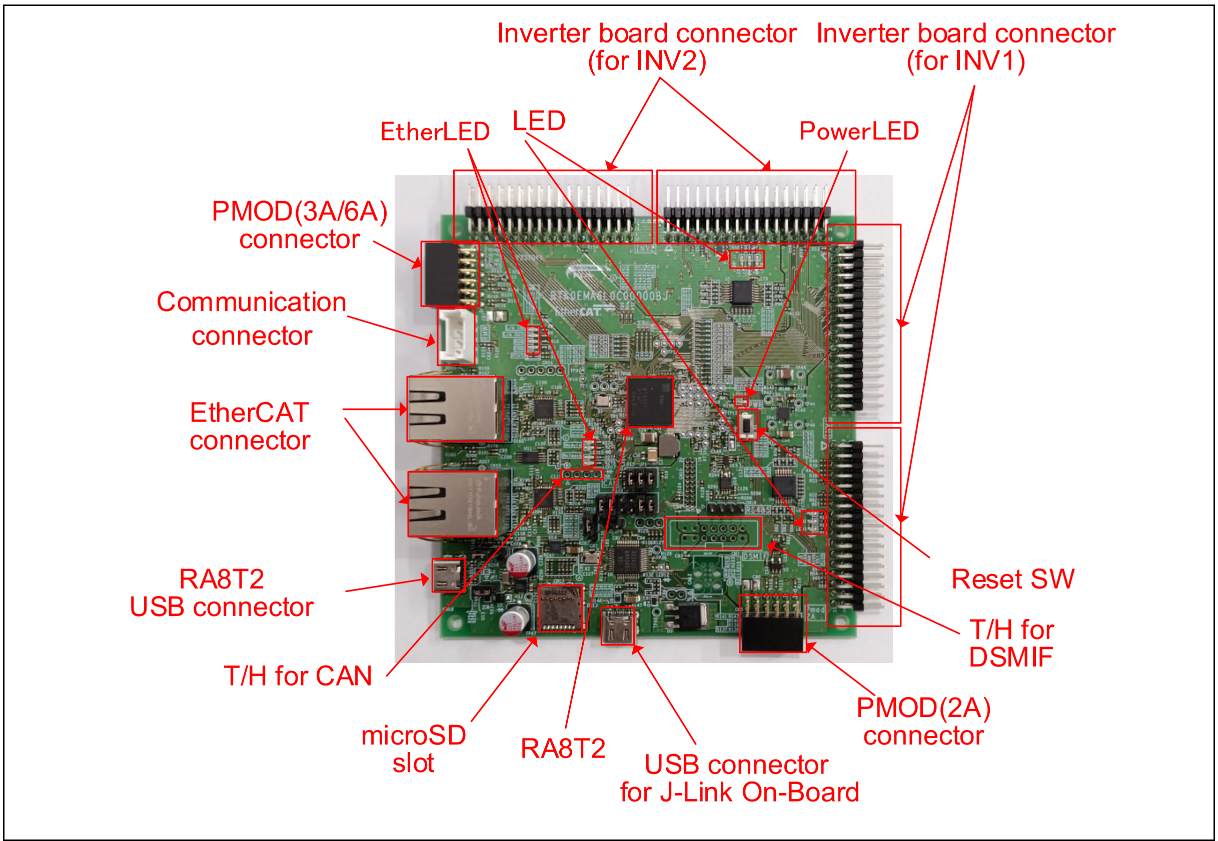


Figure 4-4 RA8T2 CPU board and its interface

4.7 Setting up the RX26T CPU Card

This section describes how to install the RX26T CPU board, which can be plugged in MCI-LV-1 directly.

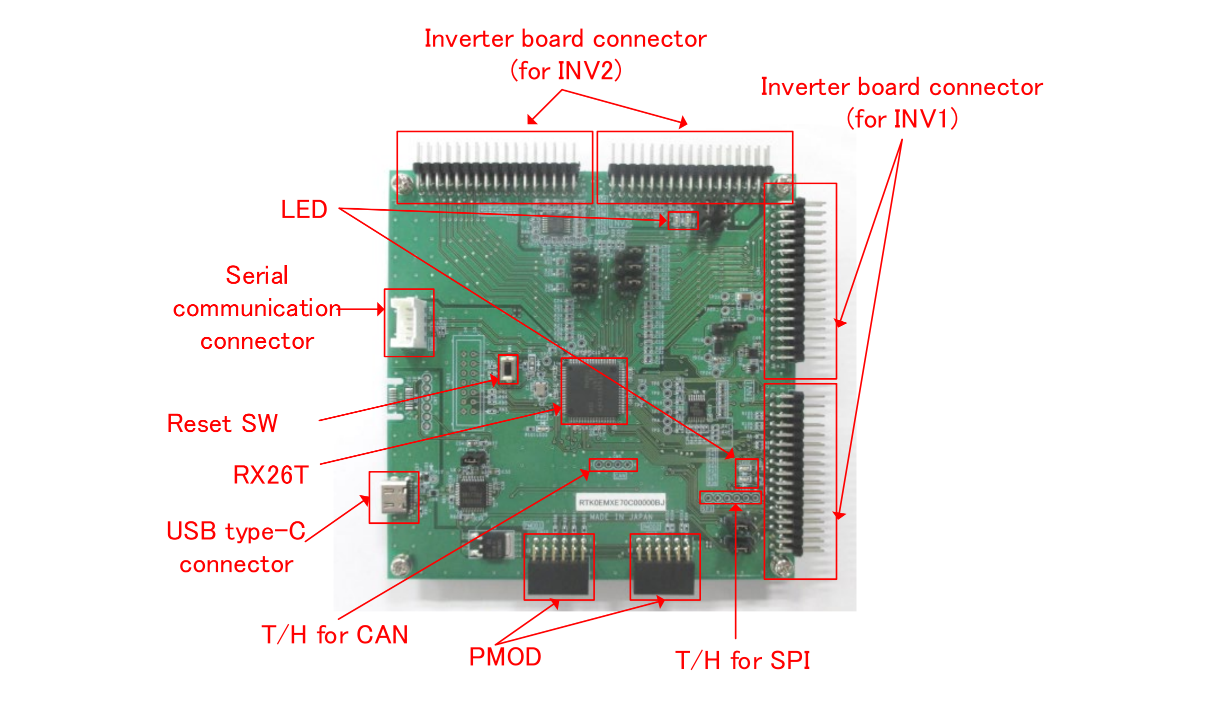


Figure 4-5 RX26T Type A CPU board and its interface

4.8 Connection Example of Kit

Figure 4-6 shows an example of the connection of a CPU board in combination with an inverter board kit (MCI-LV-1) and a communication board kit (MC-COM, model name: RTK0EMXC90Z00000BJ).

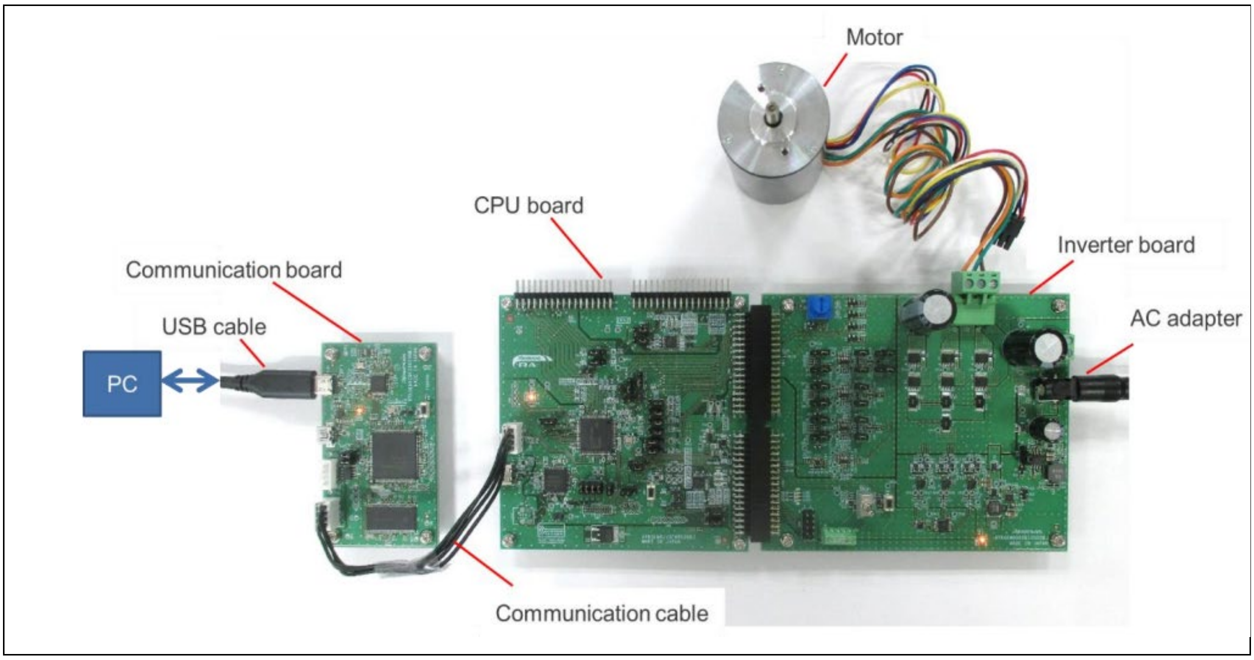


Figure 4-6 Connection example of kit

4.9 On-board Debugger

The RA Family CPU board includes the circuit of on-board debugger, J-Link OB (referred to as “JLOB” hereafter). Updating of program is performed through JLOB. To perform program update, please connect CPU board and your PC via a USB cable. The USB connector on RA Family CPU board is indicated as “USB Type-C connector” in Figure 4-2 and “USB connector for J-link On-board” in Figure 4-3 to Figure 4-4.

Similarly, the RX26T CPU board includes the circuit of on-board debugger E2 Lite (referred to as “E2OB” hereafter). Updating of program is performed through E2OB. To update the program, please connect RX26T CPU board and your PC via a USB cable. The USB connector on RX26T CPU board is indicated as “USB Type-C connector” in Figure 4-5.

4.10 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 4-7 shows an example of wiring between the power supply and the inverter. Here, the output terminals of the regulated DC power supply are connected to the P and GND terminals of the inverter. Be careful not to connect with the wrong polarity. Figure 4-8 shows an example of wiring between the inverter and the motor.

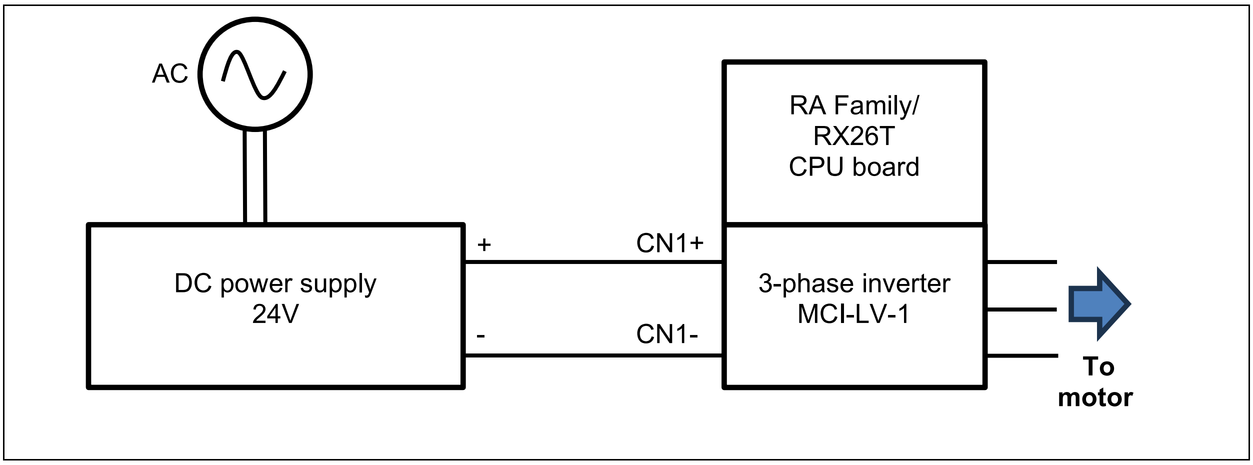


Figure 4-7 Wiring between the power supply and the inverter

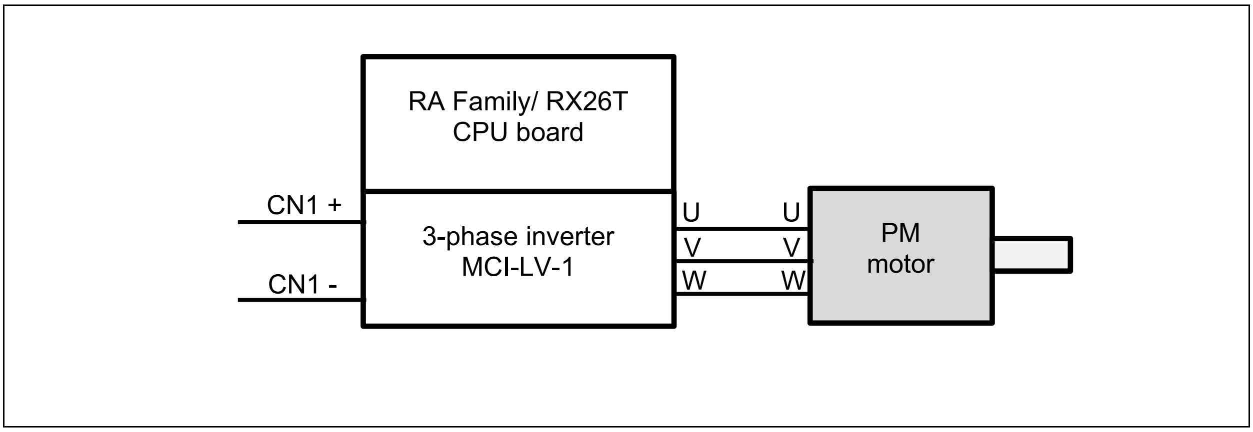


Figure 4-8 Wiring between the inverter and the motor

4.11 Using Measuring Instruments

When evaluating the sensorless control performance of a PM 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.

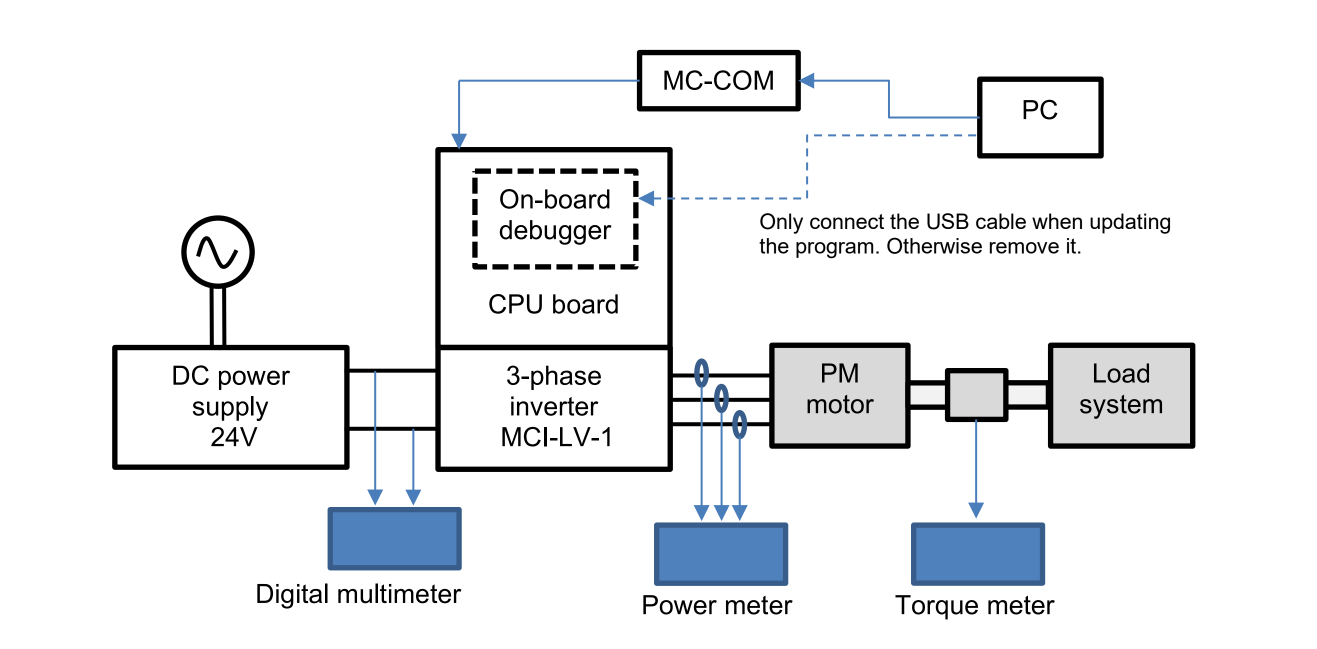


Figure 4-9 Example of Additional Measuring Instruments

5. Configuring the Software Environment

The following describes the software development environment for RA Family MCU and RX26T MCU.

How to upload or update program is described in section 6.4.

5.1 Using e^2 studio for RA Family MCU

When the e^2 studio is used with the RA Family CPU board, the "FSP with e^2 studio" package, which contains both the FSP v6.2.0 and e^2 studio for easy installation, can be used. Access the following FSP page on the Renesas site or github site.

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

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

For more information on how to use the e^2 studio, refer to the PDF manuals that you can download from the above page or the videos on the page.

5.2 Using e^2 studio for RX26T MCU

Download e^2 studio from the following URL:

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

For installation instructions, refer to the PDF manual supplied with e^2 studio or the video "e^2 studio Tutorial Video for RX Family (1/4) - Installation" posted on the web.

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

5.3 Using CS+ for RX26T MCU

Download CS+ from the following URL:

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

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

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

6. Driving the Motor

6.1 Precautions before Driving the Motor

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

- 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 during operation because the signals are isolated. When a similar device is used, the GND of the PC and the inverter may be common, which could cause an electric shock hazard via the GND.
- Design the experimental facility so that the motor can be stopped in an emergency.
- If the motor is operated at high rotation speed, 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 PM motor is still rotating, the PM 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.

6.2 Procedures of Preparing for Operation

The procedures of preparing for operations 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, 4.7
2	Install the sample program and development environment software (e^2 studio / CS+) 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.	6.4
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 24-VDC 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.	6.8
12	Stop and shut down the motor.	6.9

6.3 Connections

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

(1) For writing of program

The RA Family CPU board includes the JLOB circuit while the RX26T CPU board includes the E2OB circuit. USB cable can be connected directly from PC to the CPU board to download or write the program without another environment. Remove the USB cable after downloading the program.

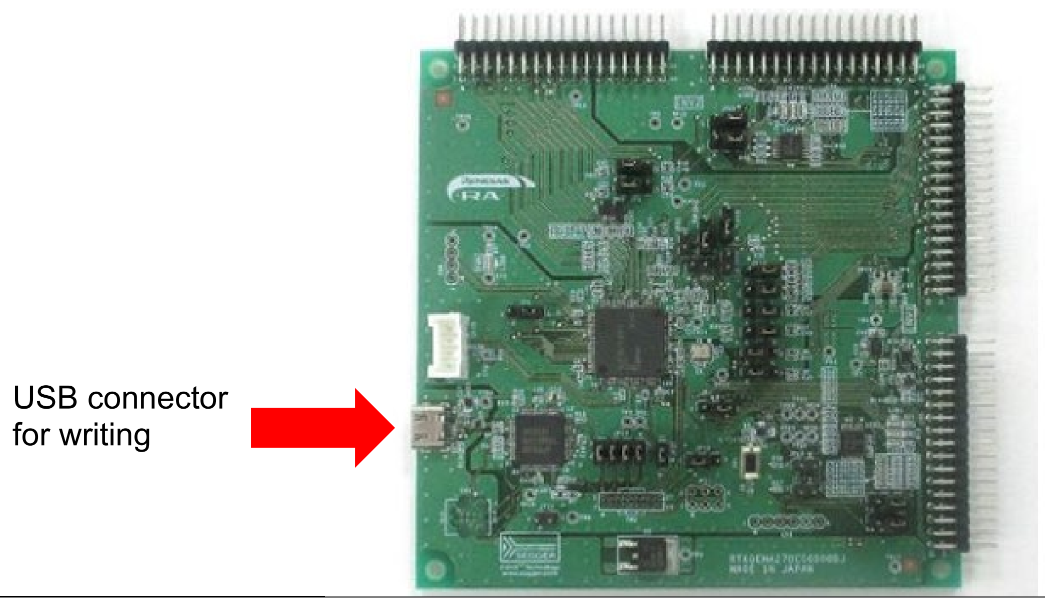


Figure 6-1 RA6T2 CPU board USB connector for writing

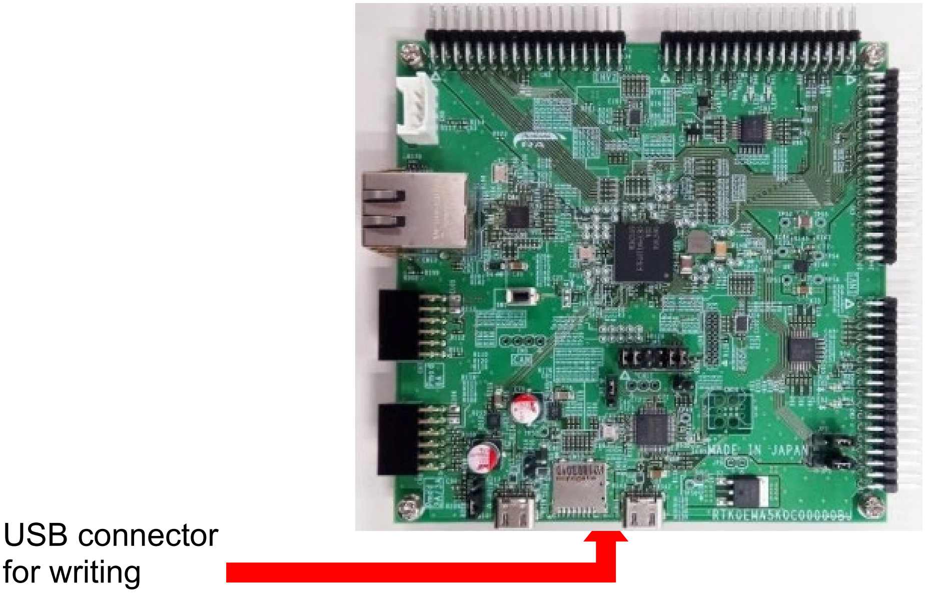


Figure 6-2 RA8T1 CPU board USB connector for writing

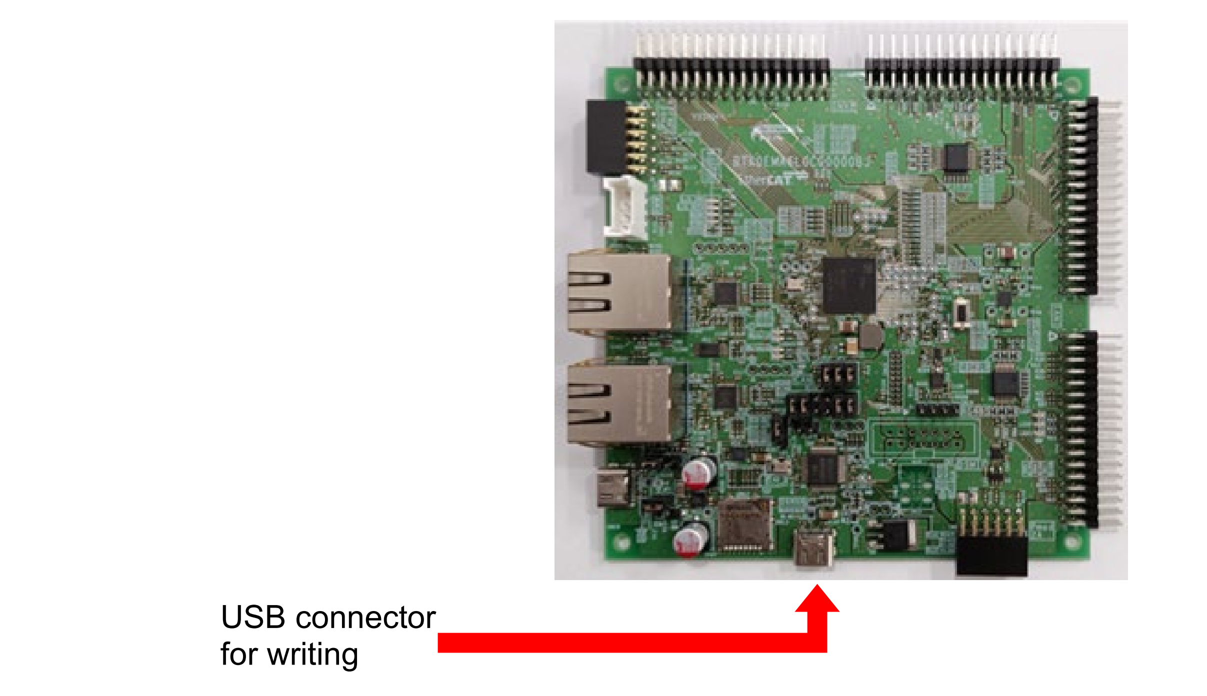


Figure 6-3 RA8T2 CPU board USB connector for writing

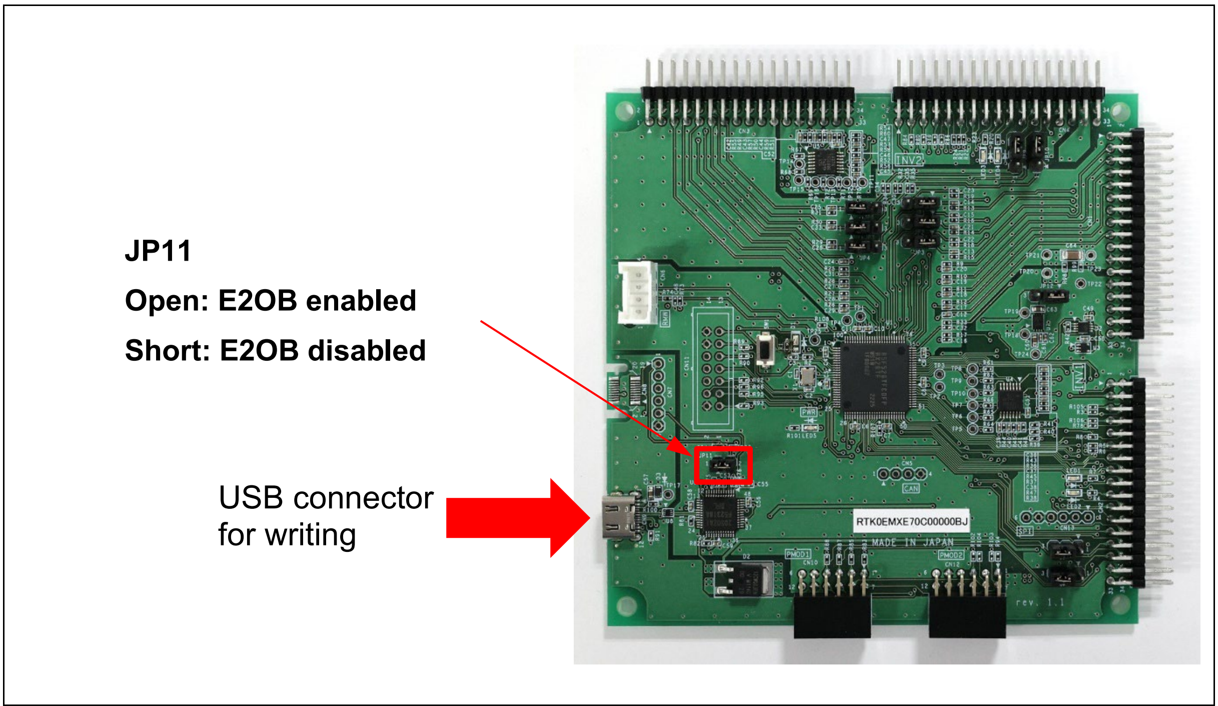


Figure 6-4 RX26T Type A CPU board USB connector for writing

(2) For motor operation

Connect PC to communication board, MC-COM(RTK0EMXC90Z00000BJ) as shown in Figure 6-5. The CPU board is connected to the PC via UART and can be operated from the PC using a COM port. Ensure the bundled communication cable is connected as shown. RMW can be used to operate the motor. MC-COM provides electrical isolation between the inverter and the PC and can be used safely.

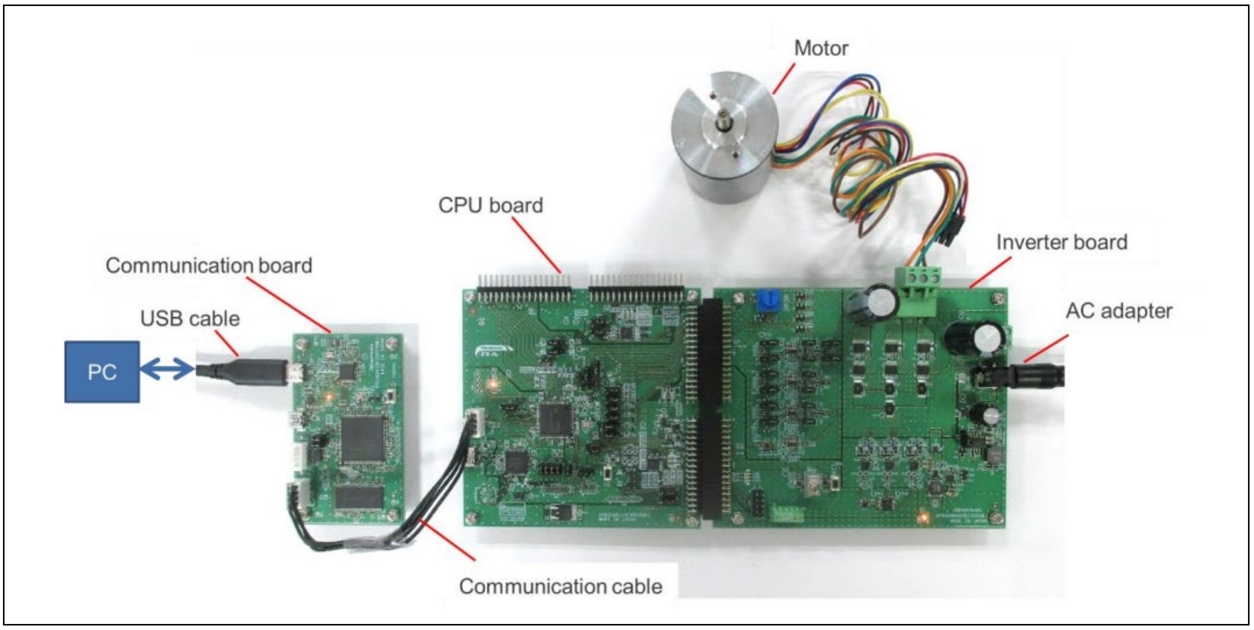


Figure 6-5 Example of wiring for motor operation

(3) Confirmation of jumper on MCI-LV-1 inverter board

Please confirm the jumper settings as shown below as the jumper position may have changed due to operation of MCI-LV-1 in other environment for other control type.

・Inverter board

Jumper	Connection
JP8	Short 1-2pin
JP11	Short 1-2pin

JP8 and JP11 are settings for the type of shunt current detection circuit. Refer to MCI-LV-1 User's manual (R12UZ0092) for more details.

6.4 Writing the sample program

After downloading the sample program from our website, users can write the program to the MCU on the CPU board with e^2 studio or CS+ (for RX26T only).

6.4.1 Installation of e^2 studio for RA Family or RX26T MCU

For RA Family MCU, please download e^2 studio which supports FSP configurator from below URL and install it to your PC.

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

For RX26T MCU, please download e^2 studio or CS+ from below URL and install it to your PC.
https://www.renesas.com/en/software-tool/e-studio
https://www.renesas.com/en/software-tool/cs

6.4.2 Importing a project in e^2 studio

Only importing a project in e^2 studio is described below. For importing a project in in CS+, please refer to CS+ manual.

1. Left click “File” TAB.

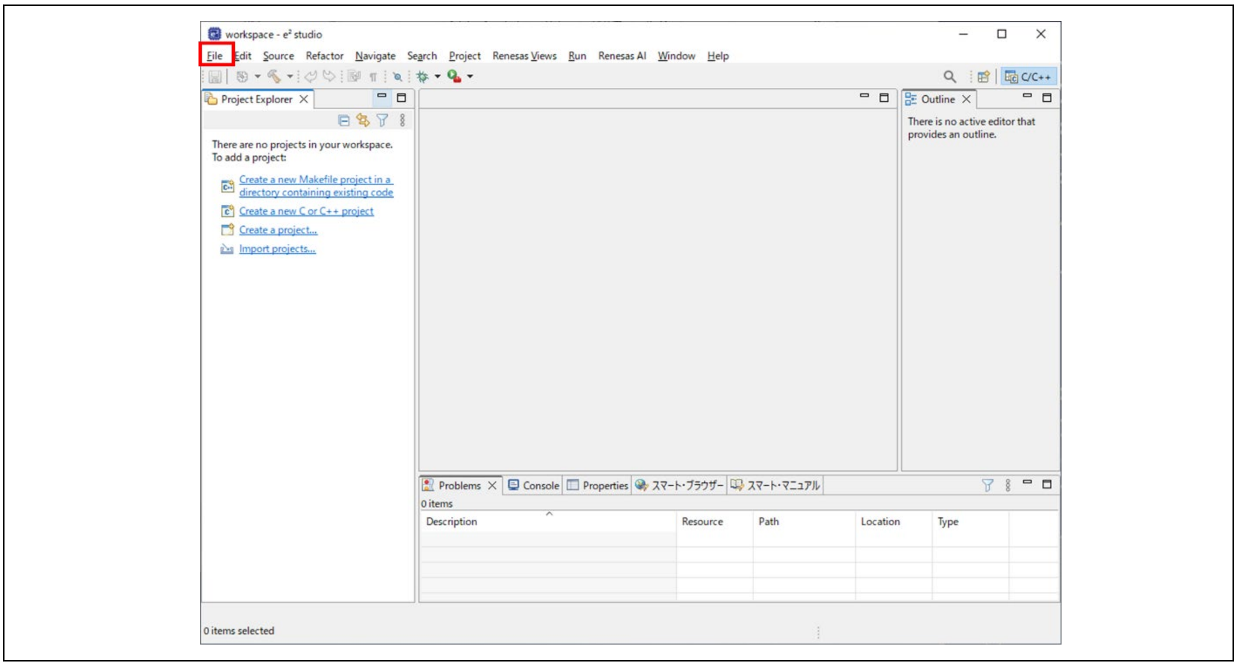


Figure 6-6 Selection of the target project (1)

2. Pull down menu is displayed. Then, select “Import” and left click.

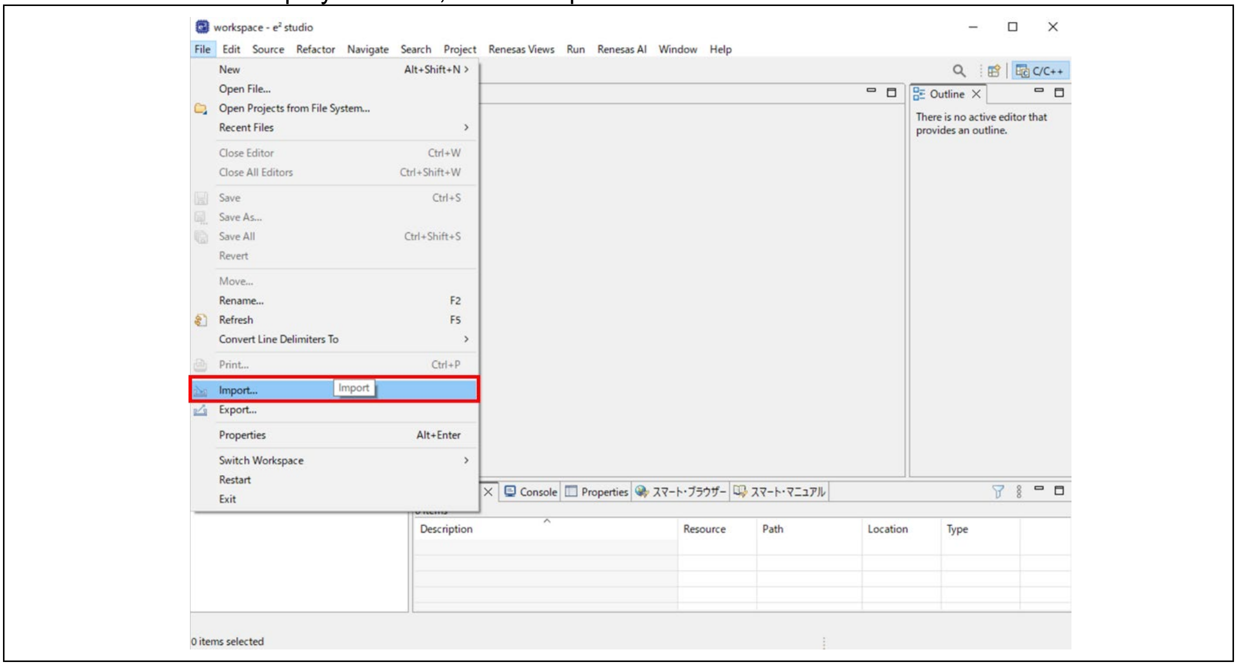


Figure 6-7 Selection of the target project (2)

3. Import window opens. Then, select “Existing project into Workspace” and left click.

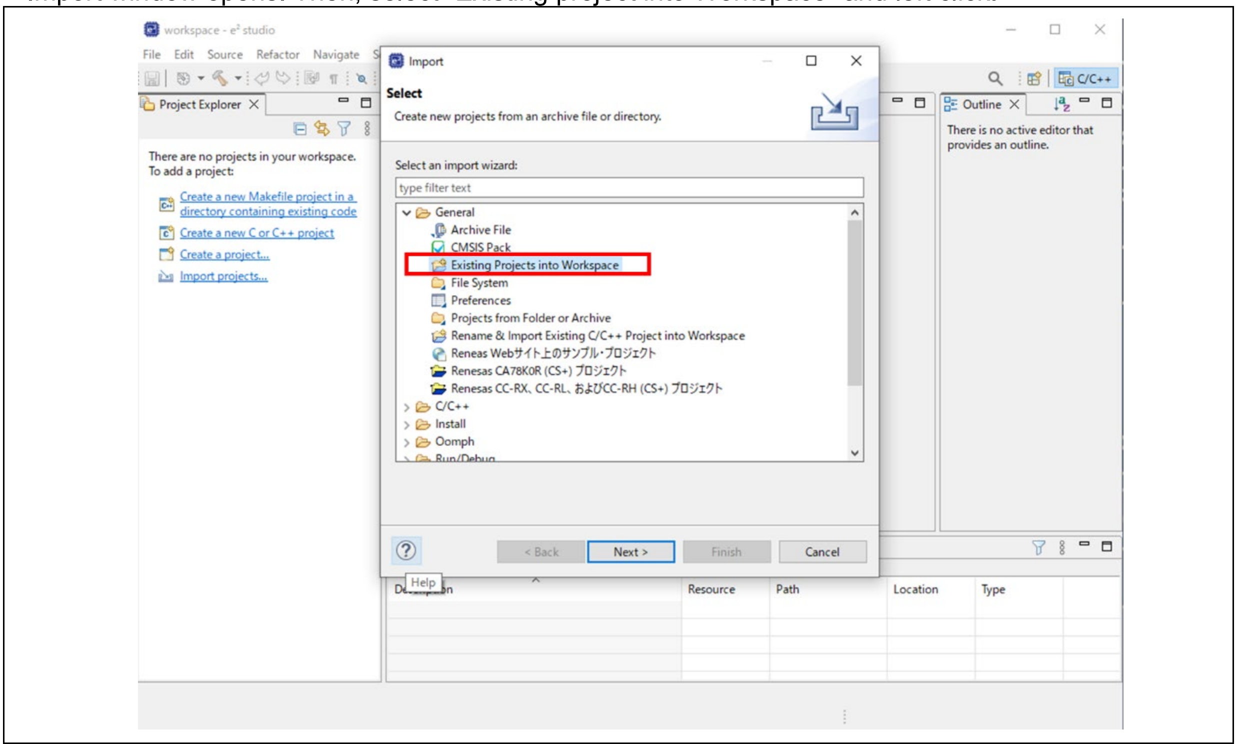


Figure 6-8 Selection of the target project (3)

4. “Import Projects” window opens. Then, left click “Browse”.

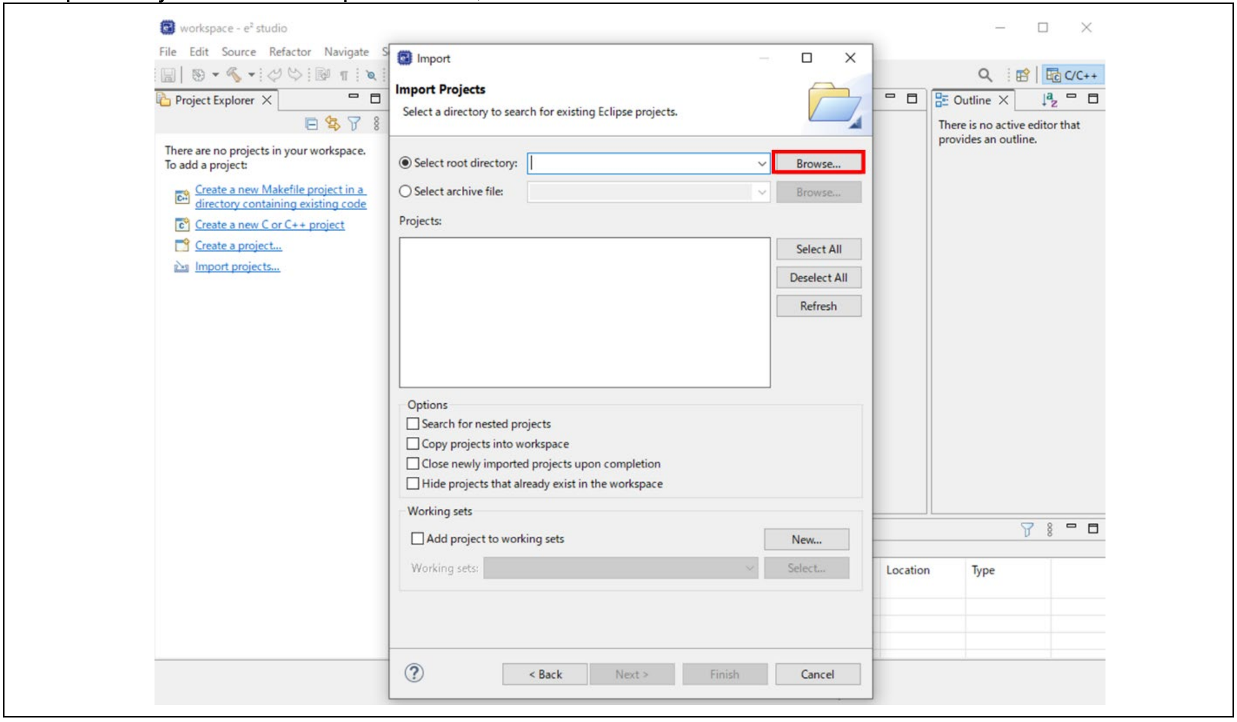


Figure 6-9 Selection of the target project (4)

5. Folder selection window opens. Select the target folder and left click “Select Folder”.

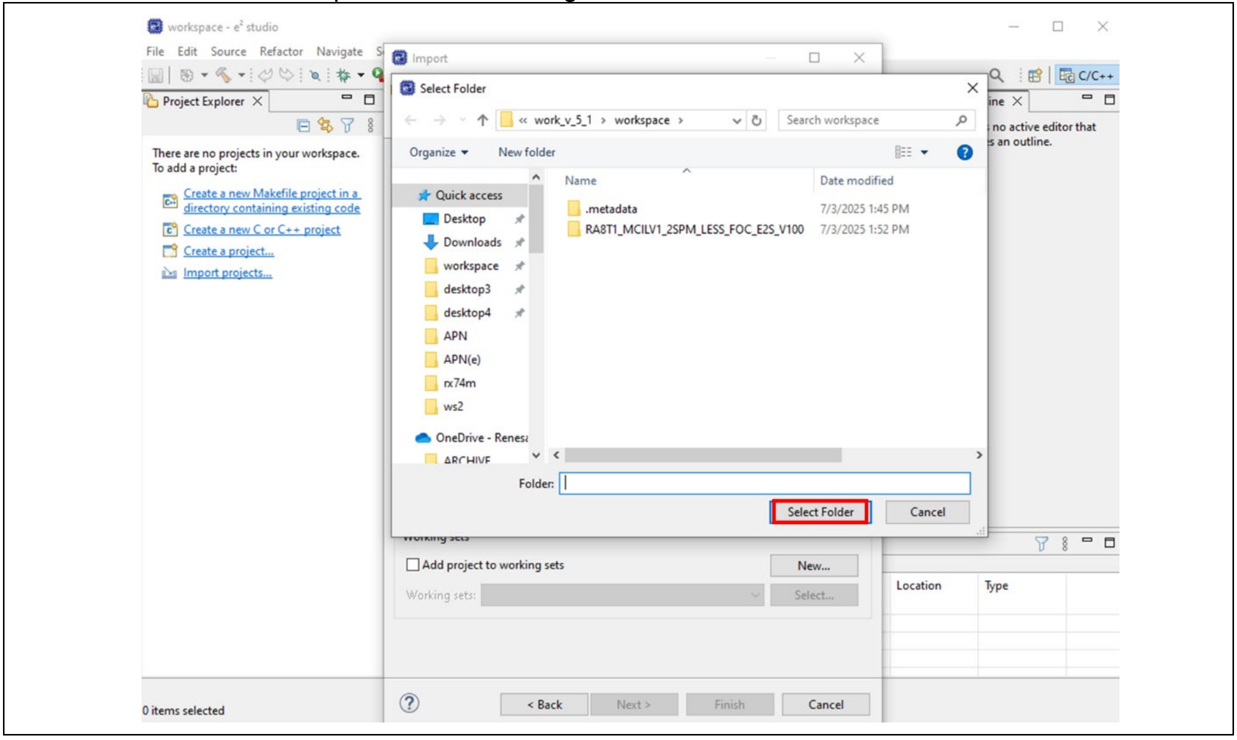


Figure 6-10 Selection of the target project (5)

6. When the target project is selected correctly, the display becomes like below. After confirmation, left click “Finish”.

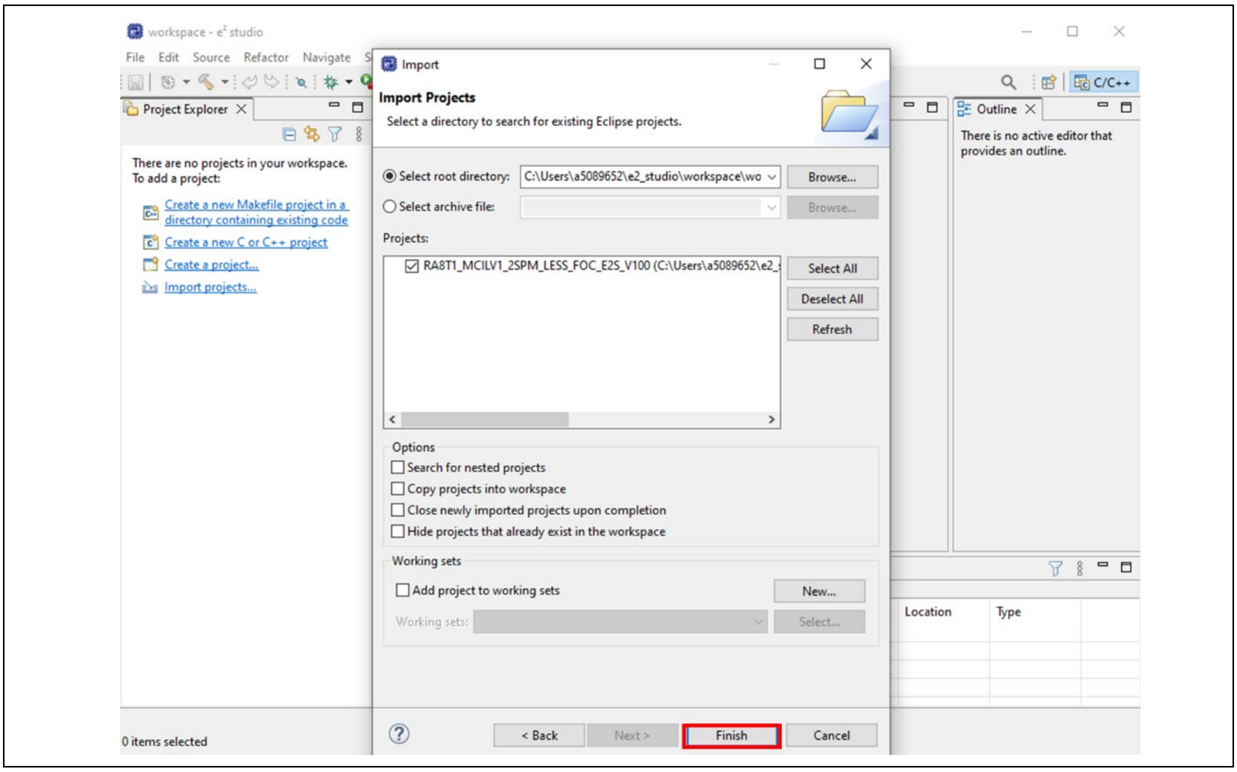


Figure 6-11 Selection of the target project (6)

7. Confirm the target project is imported correctly into e² studio.

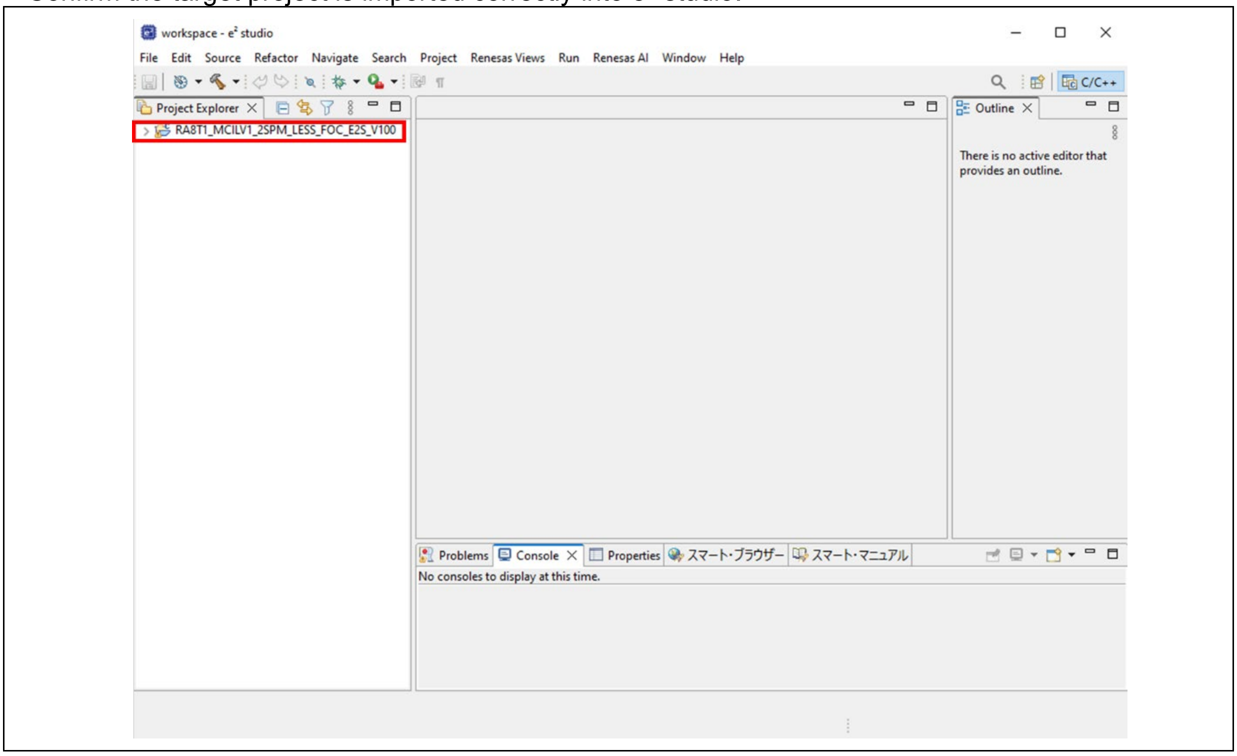


Figure 6-12 Confirmation of import the target project

6.4.3 Build the target project in e² studio

Only build process in e² studio is described in the following. Please refer to CS+ manual for building a project.

1. Right click the imported target project in e² studio window.

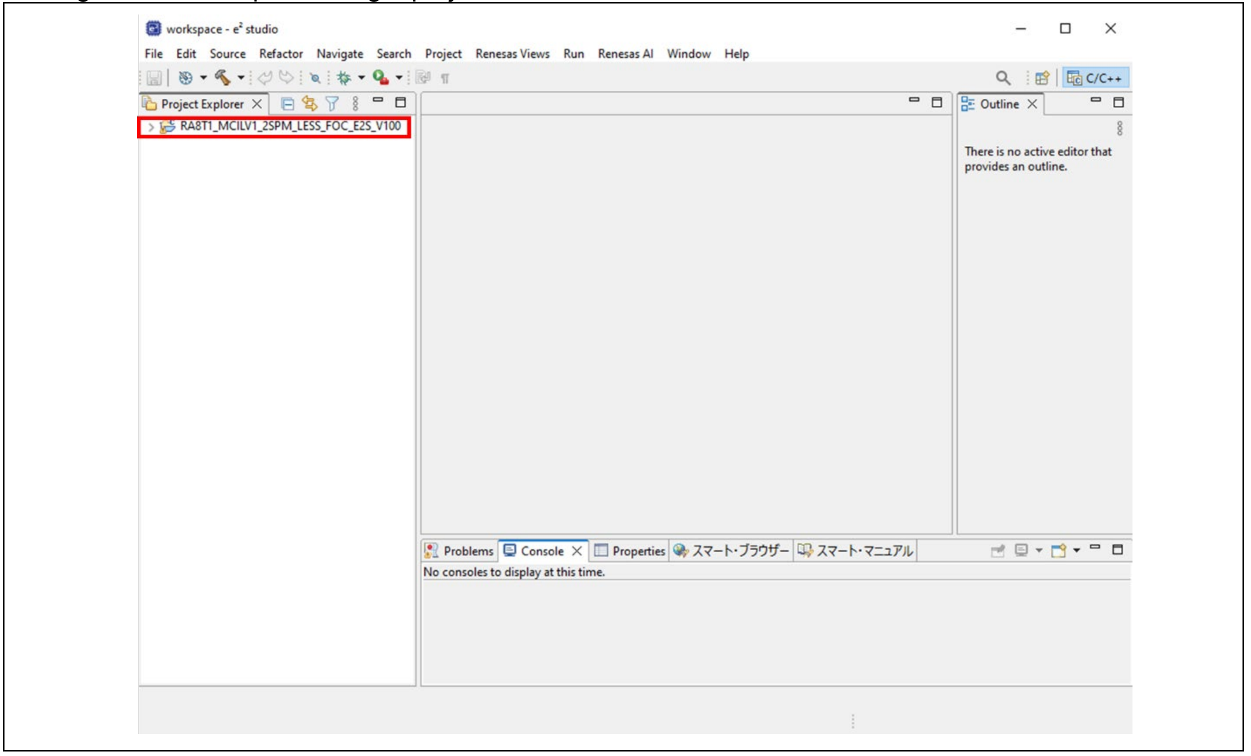


Figure 6-13 Selection of the target project

2. Pulldown menu appears. Then, left click "Build Project".

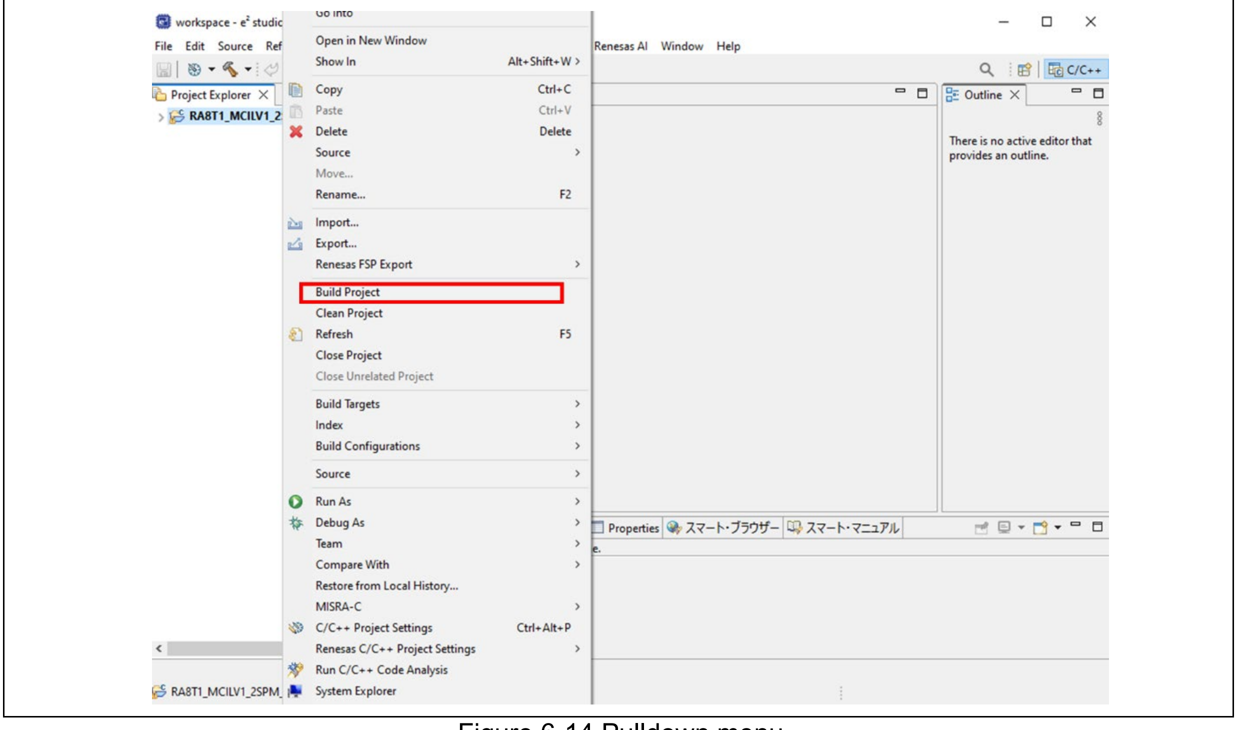


Figure 6-14 Pulldown menu

3. Target build is performed, and build process are displayed in console window. Confirm the completion of build with no error.

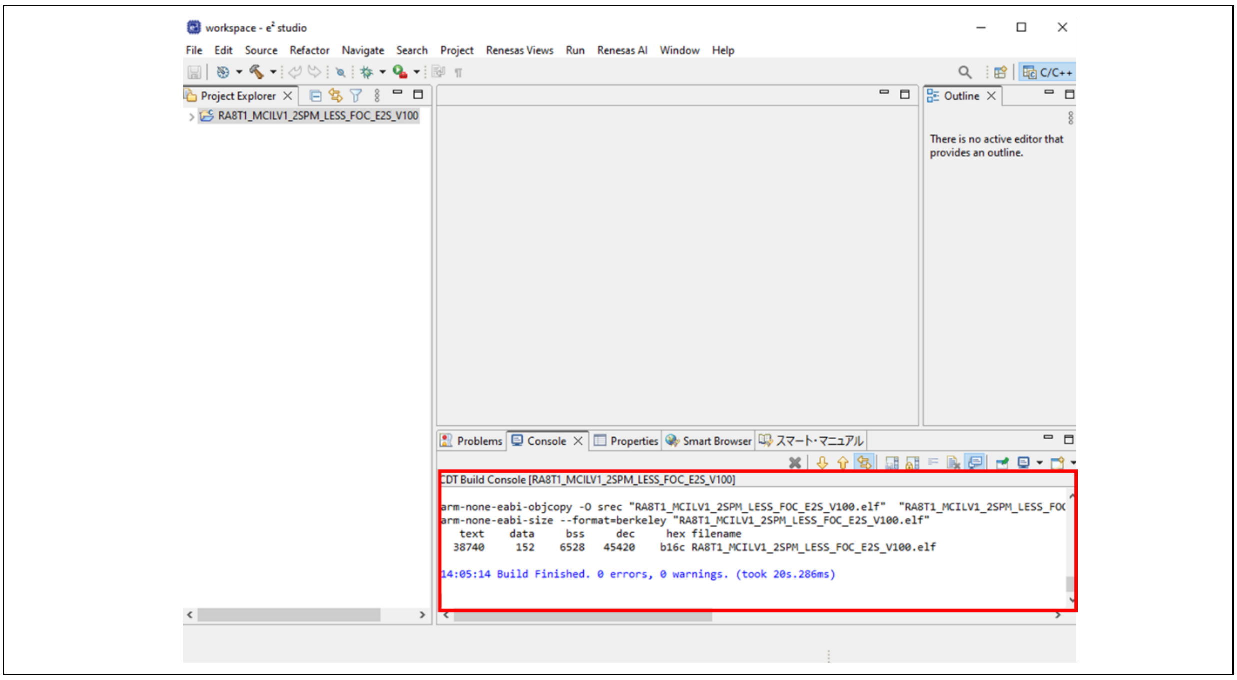


Figure 6-15 Confirmation of finish of build

6.4.4 Connection between PC and the target CPU board via USB cable

Please connect PC and CPU board via USB cable like below. The description uses RA6T2 CPU board as an example.

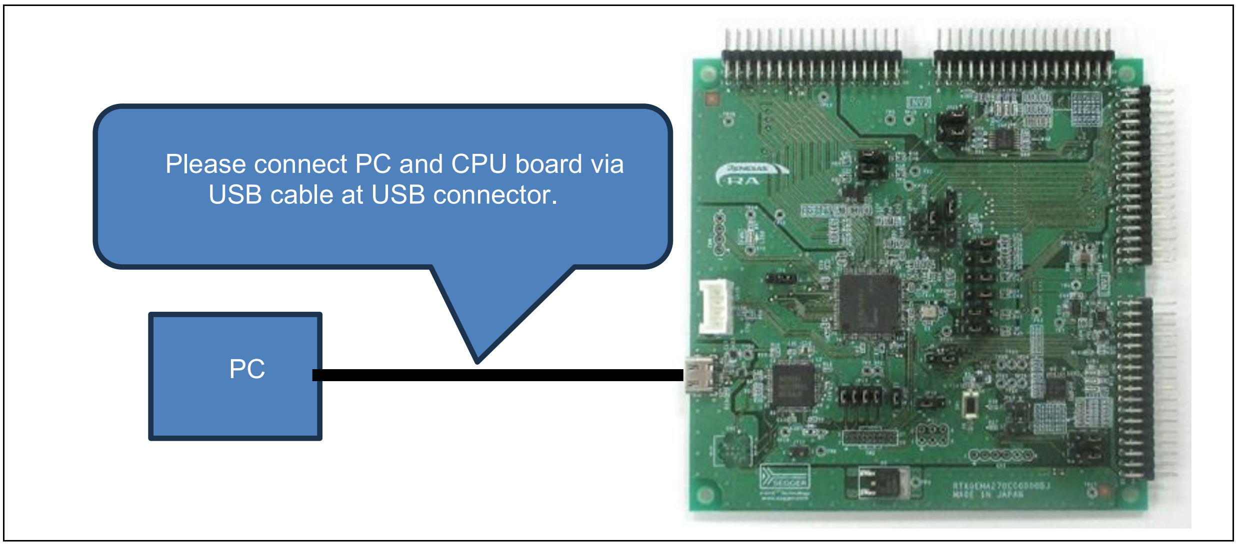


Figure 6-16 Confirmation between PC and CPU board (RA6T2)

6.4.5 Writing to the target board (with built program)

Only writing steps in e[2] studio are described in the following. Please refer to CS+ manual for writing procedures.

1. Select target project and right click.

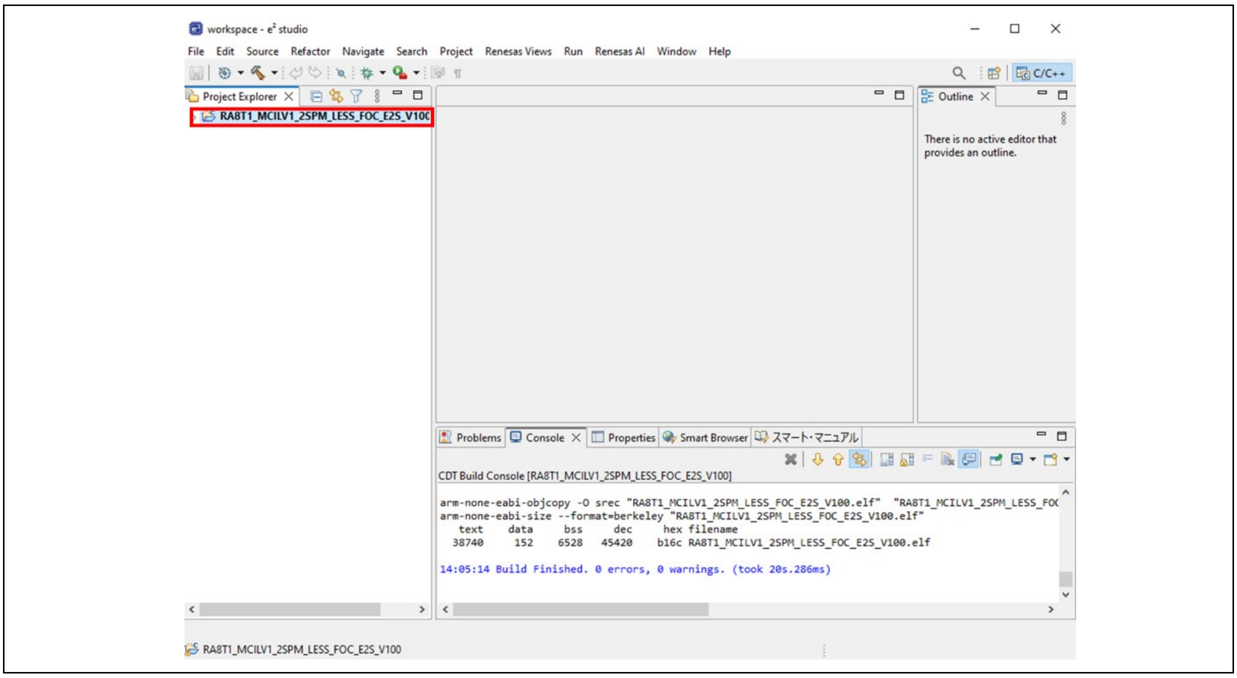


Figure 6-17 Selection of target project

2. Pull down menu is displayed. Hover cursor on "Debug As" to display available options. Select "Renesas GDB Hardware Debugging" and left click.

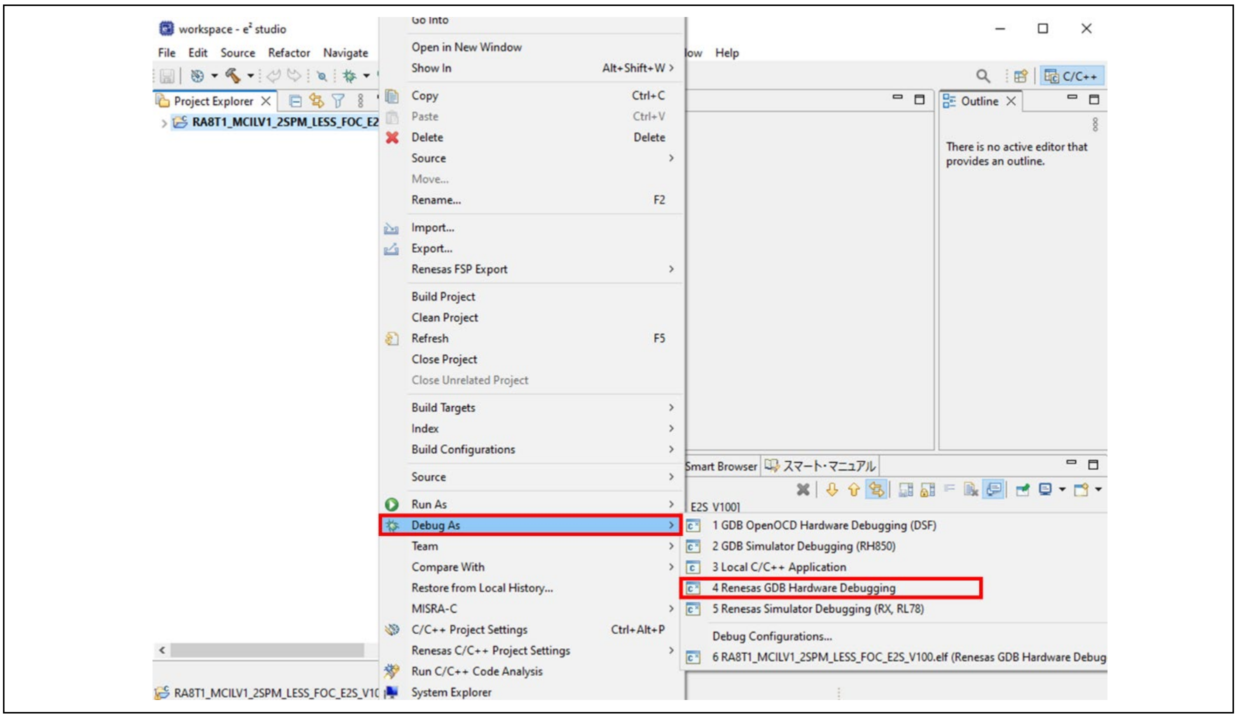


Figure 6-18 Selection of debug method

3. If the connection is correct and program is downloaded successfully, the display changes to "Debug" mode as shown below.

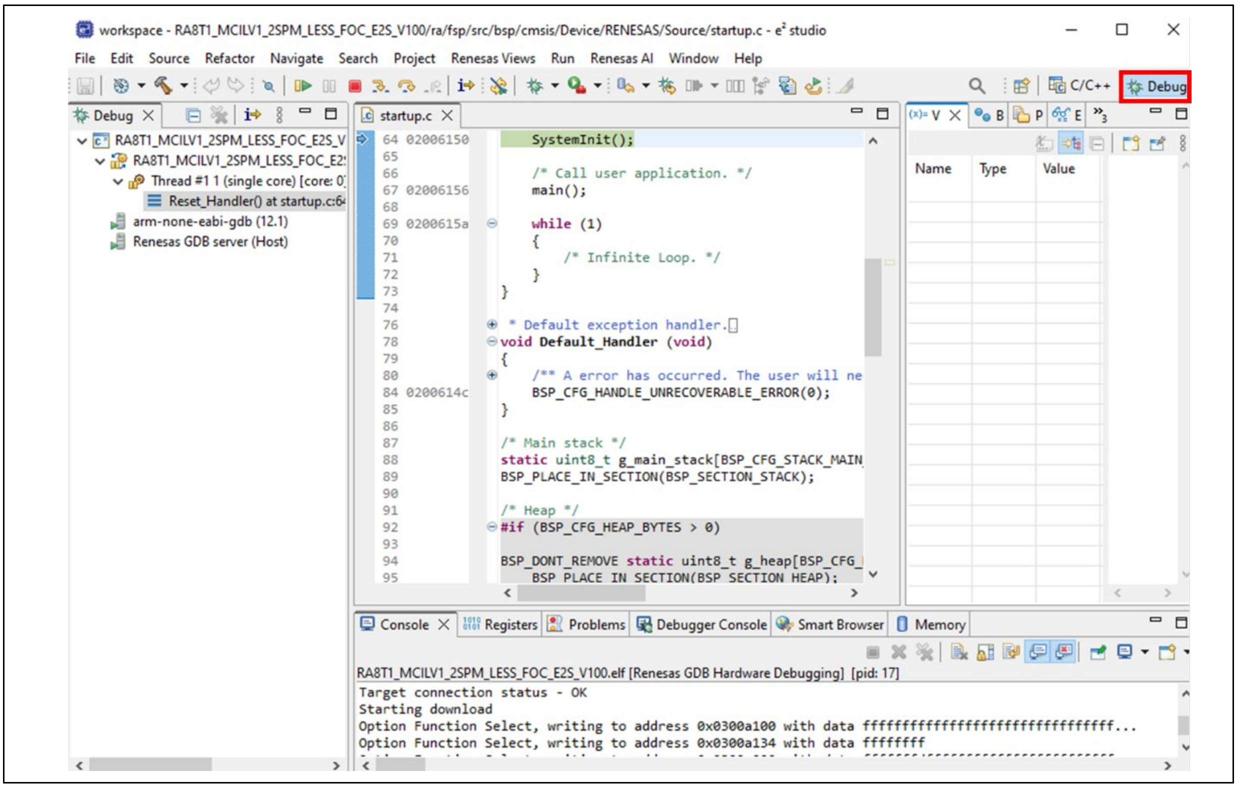


Figure 6-19 "Debug" mode

4. Left click "■" in debug mode to disconnect from target board, then remove the USB cable. All writing processes are completed.

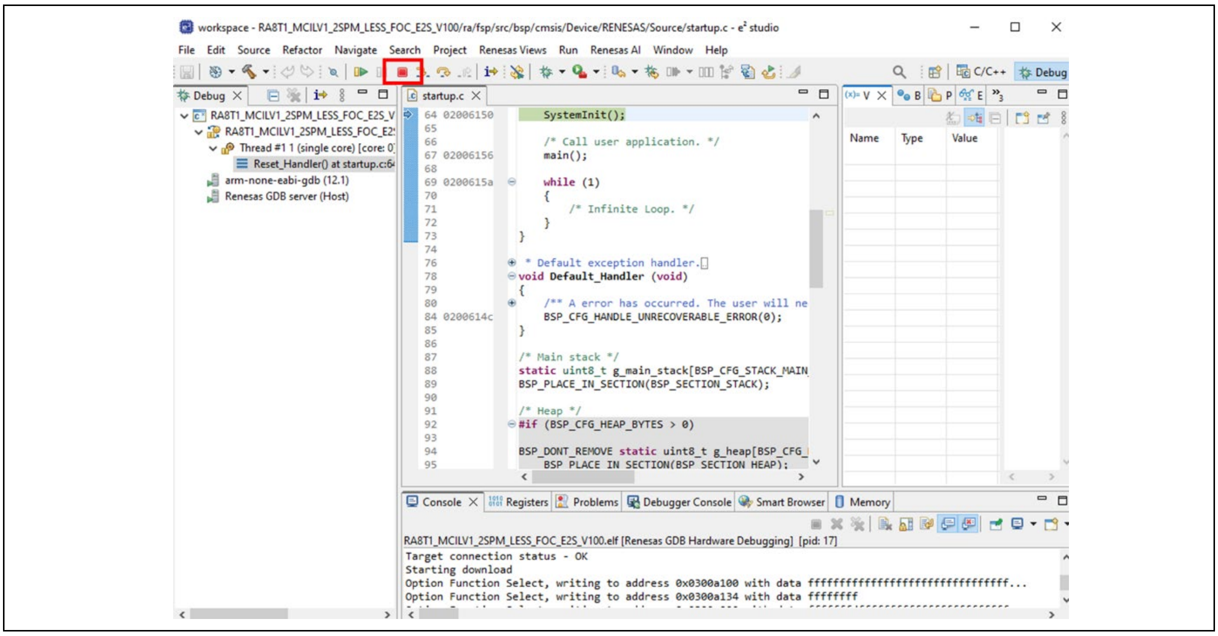


Figure 6-20 Disconnect from target board

6.5 Introducing RMW

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

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

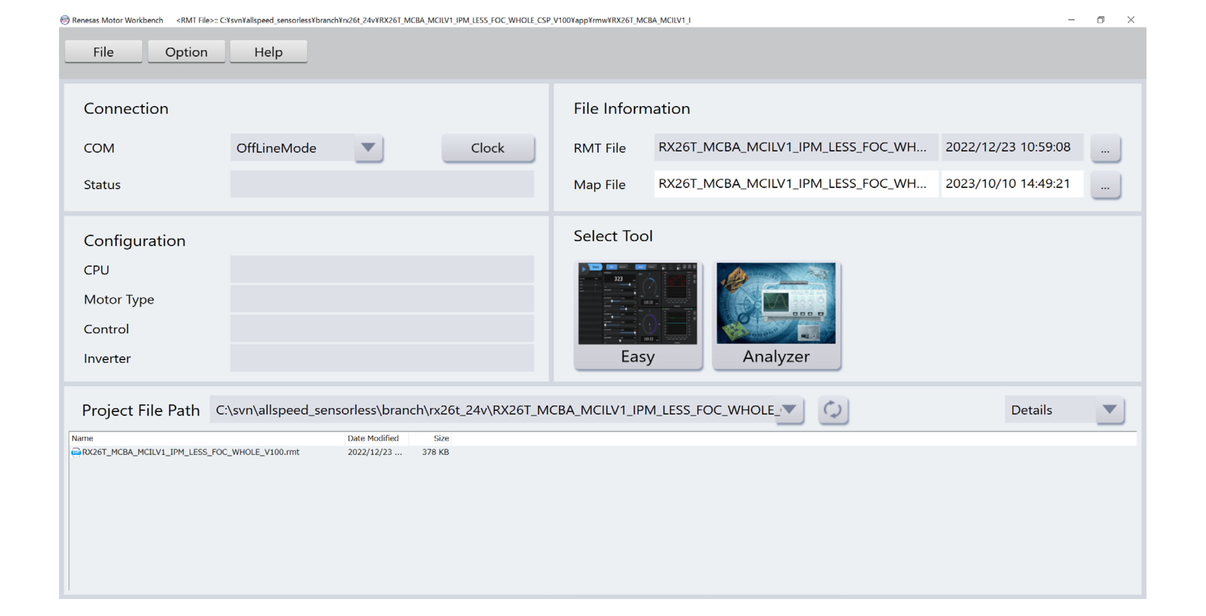


Figure 6-21 Window of Renesas Motor Workbench

6.6 Registering and updating the Map file

When a part of the sample program is changed by the user, the Map file including variables and other information needs to be registered and updated. If the software has not been changed, the Map file does not need to be registered nor updated.

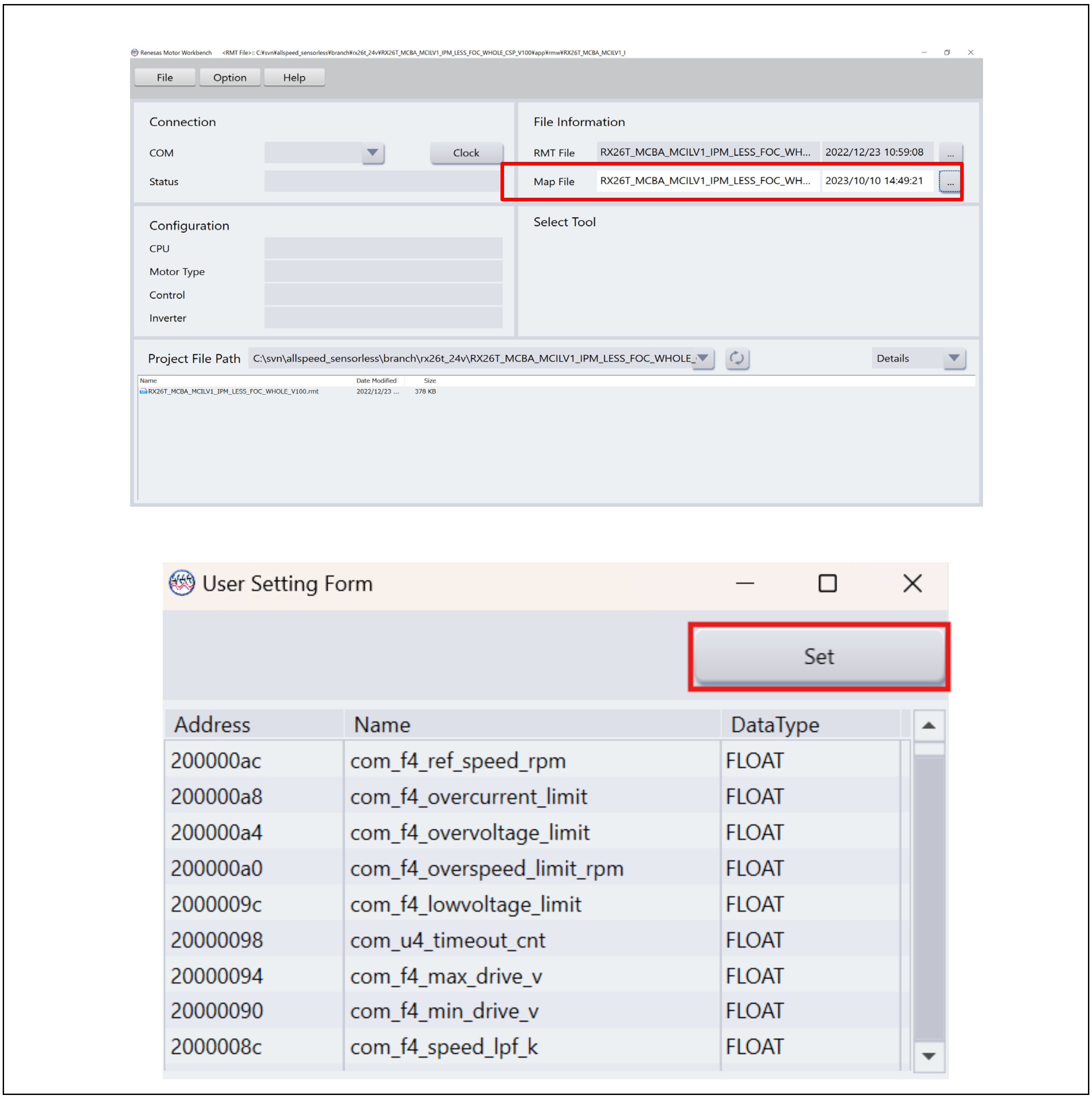


Figure 6-22 Map file registration setup window (upper) and confirmation window (lower) in RMW

6.7 Variables used for working with RMW

In this sample program, the motor can be controlled using RMW. Table 6-2 lists the data input variables that are used when the RMW UI is used. The values input to these variables are applied to the corresponding variables in the motor module and then used for controlling the motor if the value written to the com_u1_enable_write variable is the same as the value of the g_u1_enable_write variable. Note, however, that the variables indicated by an asterisk (*) do not depend on the value of the com_u1_enable_write variable.

For the other parameters for controlling the motor, see Table 6-4.

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

Table 6-2 List of Main Input Variables for 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 6-3 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 6-3 List of Main Variables

Name of Main Variable	Type	Description
g_st_sensorless_vector.u2_error_status	uint16_t	Error status. For details, see section 6.8 c), 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 6-4 List of com Variables

Variable	Description
com_u2_offset_calc_time	Setting of the time for calculating the current offset value
com_u2_charge_bootstrap_time	Charging time for the bootstrap circuit (cnt)
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 c (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
com_f4_speed_lpf_hz	Speed LPF cut-off frequency (Hz)
com_f4_speed_rate_limit_rpm	Maximum increment and decrement width (mechanical angle) for the speed command (rpm/s) (for use when speed control is enabled)
com_f4_overspeed_limit_rpm	Speed limit value (mechanical angle) (rpm)

Variable	Description
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_u1_flag_less_switch_use	Setting of the switching function in open-loop control 0: Disable, 1: Enable
com_u1_flag_openloop_damping_use	Setting of damping control in open-loop control 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_switch_phase_err_deg	The threshold for the angle error in sensorless switching judgment (degree)
com_f4_opl2less_sw_time	Switching time to sensorless
com_f4_phase_err_lpf_cut_freq	Angle-error LPF cutt-off frequency [Hz]
com_f4_ed_hpf_omega	Damping control: HPF coefficient
com_f4_ol_ref_id	Open-loop control: d-axis current command value (A)
com_f4_id_up_time	D-axis current increase time (cnt)
com_f4_id_down_time	D-axis current decrease time (cnt)
com_f4_id_down_speed_rpm	Speed for switching the motor control method (accelerating) (rpm)
com_f4_id_up_speed_rpm	Speed for switching the motor control method (decelerating) (rpm)
com_f4_ol_damping_zeta	Damping control: Damping compensation ζ
com_f4_ol_damping_fb_limit_rate	Damping control: Feedback limit rate for damping compensation
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_1f/2f	Torque vibration suppression: Phase lead (rad)
com_f4_tf_lpf_omega	Torque vibration suppression: Natural frequency for the LPF in the tracking filter (TF) (Hz)
com_f4_output_gain1f/2f	Torque vibration suppression: Gains for the value input to the repetitive controllers
com_u1_flag_trqvib_comp_learning	Torque vibration suppression: Learning function enabling flag

Variable	Description
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_suppression_th_1f/2f	Torque vibration suppression: Goal value for suppression
com_f4_abnormal_output_th_1f/2f	Torque vibration suppression: Threshold for the ratio of abnormal output from the TF
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)

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

6.8 Controlling the motor

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

a) Start driving of the motor
The motor can be rotated by performing the following steps:
(1) Confirm that the check boxes in the [W?] column are selected on the "com_u1_system_mode" and "com_f4_ref_speed_rpm" rows.
(2) On the "com_f4_ref_speed_rpm" row, in the [Write] column, enter the command rotation speed.
(3) Click the [Write] button (At this time, the com_u1_system_mode field remains at "0").
(4) Click the [Read] button. Confirm that the boxes in the [Read] column on the "com_f4_ref_speed_rpm" rows.
(5) On the "com_u1_system_mode" row, in the [Write] column, enter "1".
(6) Click the [Write] button.

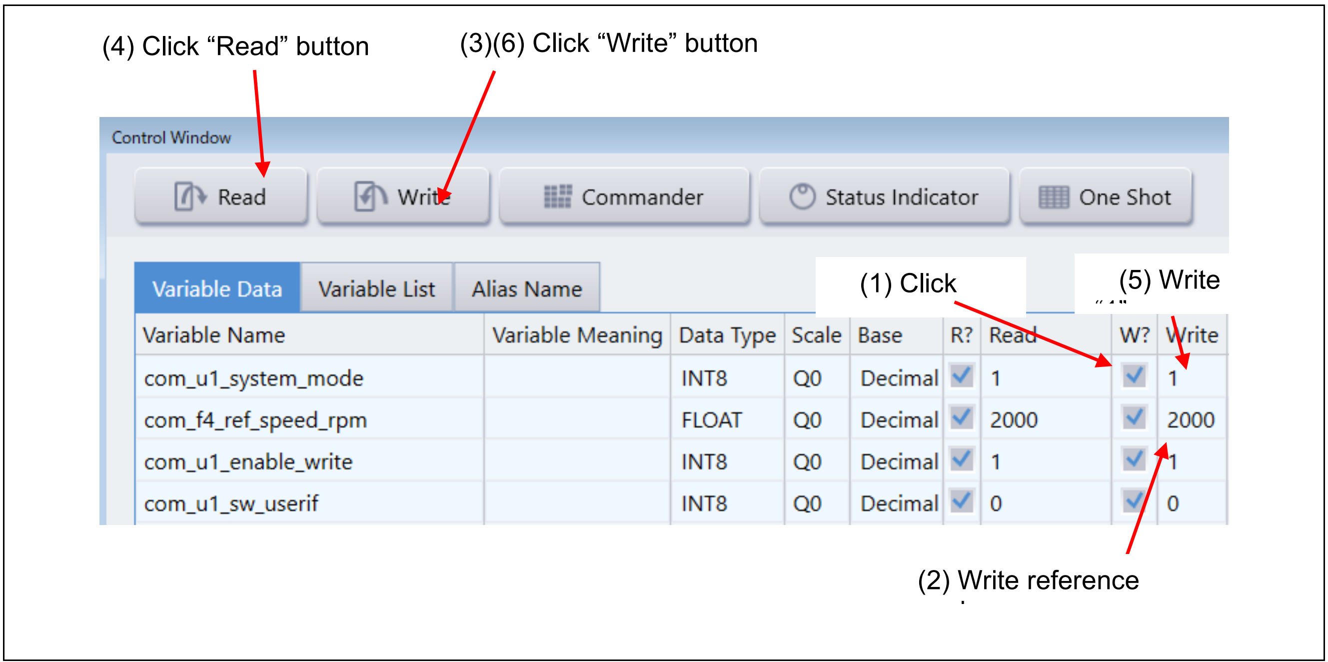


Figure 6-23 Procedure for driving of the motor

b) Stop the motor
   (1) On the "com_u1_system_mode" row, in the [Write] column, enter "0".
   (2) Click the [Write] button.
   (3) Confirm that the motor has stopped.

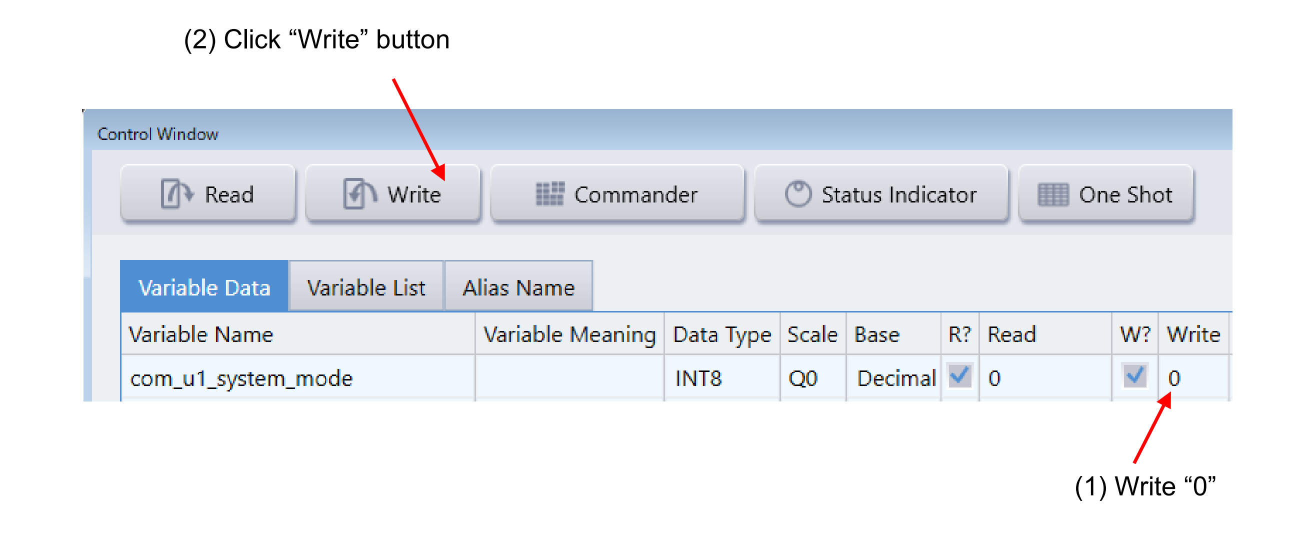


Figure 6-24 Procedure for stopping the motor

c) What to do in case of motor stop (due to an error)
   (1) On the "com_u1_system_mode" row, in the [Write] column, enter "3".
   (2) Click the [Write] button.

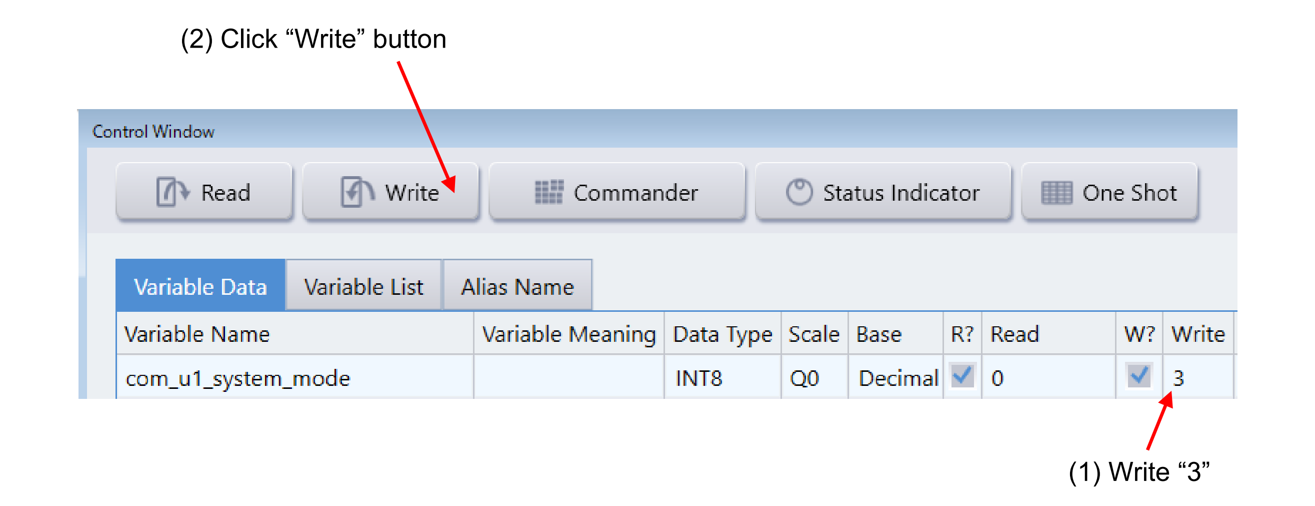


Figure 6-25 Procedure for handling an error

Table 6-5 Error status description

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	Hall signal timeout error (Not happen in sensorless vector)	MOTOR_ SENSORLESS_VECTOR_ERROR_HALL_TIMEOUT
0x0010	BEMF signal timeout error (Not happen in sensorless vector)	MOTOR_ SENSORLESS_VECTOR_ERROR_BEMF_TIMEOUT
0x0020	Hall signal pattern error (Not happen in sensorless vector)	MOTOR_ SENSORLESS_VECTOR_ERROR_HALL_PATTERN
0x0040	BEMF signal pattern error (Not happen in sensorless vector)	MOTOR_ SENSORLESS_VECTOR_ERROR_BEMF_PATTERN
0x0080	Low-voltage error	MOTOR_ SENSORLESS_VECTOR_ERROR_LOW_VOLTAGE
0x0100	Software overcurrent error	MOTOR_ SENSORLESS_VECTOR_ERROR_OVER_CURRENT_SW
0x0200	Stall error	MOTOR_ SENSORLESS_VECTOR_ERROR_STALL_DETECTED
0xFFFF	Undefined error	MOTOR_ SENSORLESS_VECTOR_ERROR_UNKNOWN

6.9 Stopping and shutting down the motor

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

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

② After confirming that the motor stops, operate the regulated DC power supply to stop supplying 24 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
Current detection method	3-shunt detection
Pulse width modulation (PWM) method	Space vector modulation method (sinusoidal modulation can also be selected)
Position and speed estimation method	Low speed range: Current-drawn control (open loop) Medium to high speed range: BEMF observer
Control mode	Only speed control
Compensation functions	• Maximum torque per current control (MTPA) and flux weakening control • Voltage error compensation and 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.

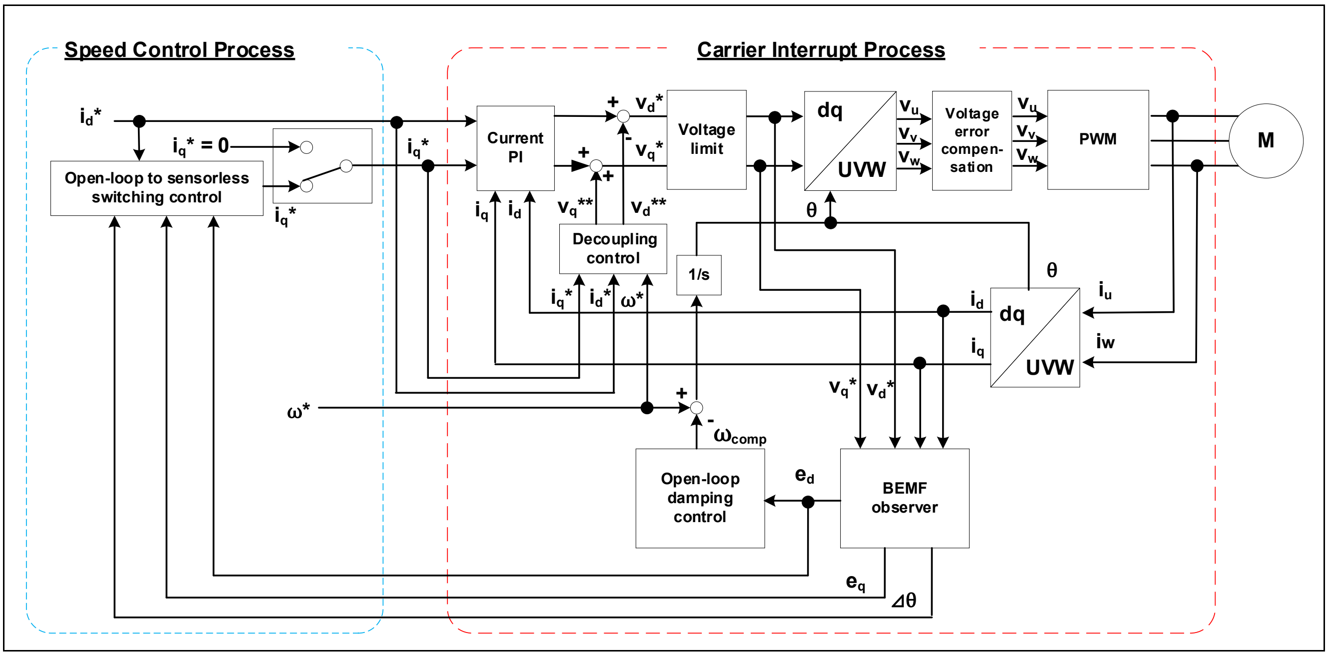


Figure 7-1 Schematic Block Diagram of Sensorless Vector Control System (Open-Loop Control)

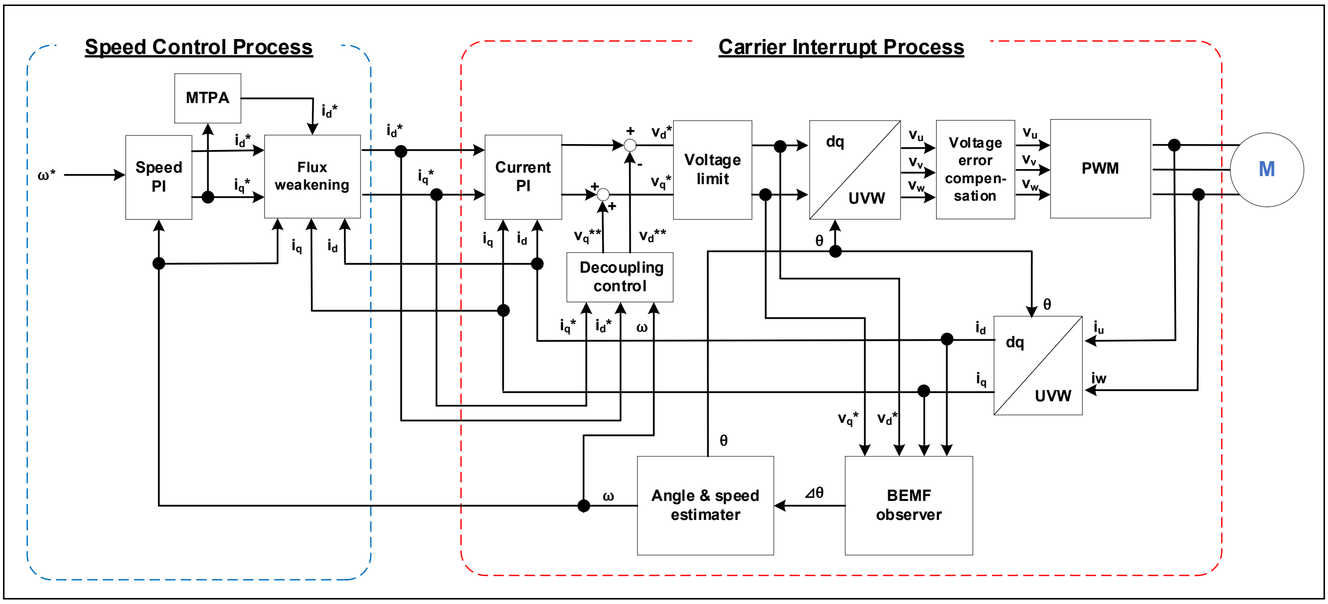


Figure 7-2 Schematic Block Diagram of Sensorless Vector Control System (Sensorless Control)

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.

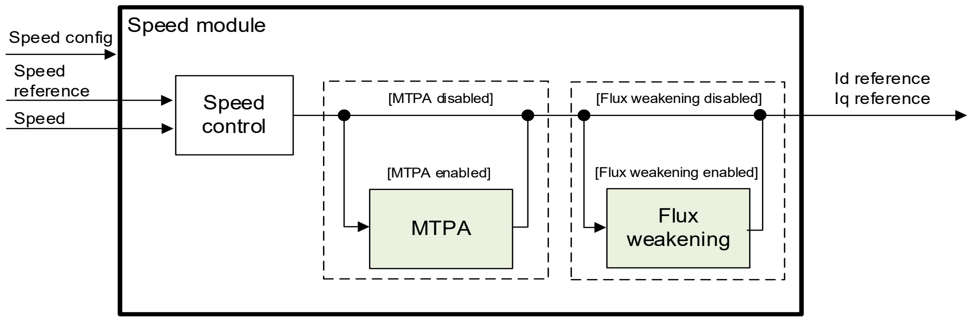


Figure 7-3 Functional Block Diagram of Speed Control

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 (= ωψ) generated in proportion to the rotation of the PM motor exceeds the voltage that can be output from the bus voltage, this function increases the d-axis current command value in the negative direction to reduce the inductive voltage (Figure 7-4). 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 7-5 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.

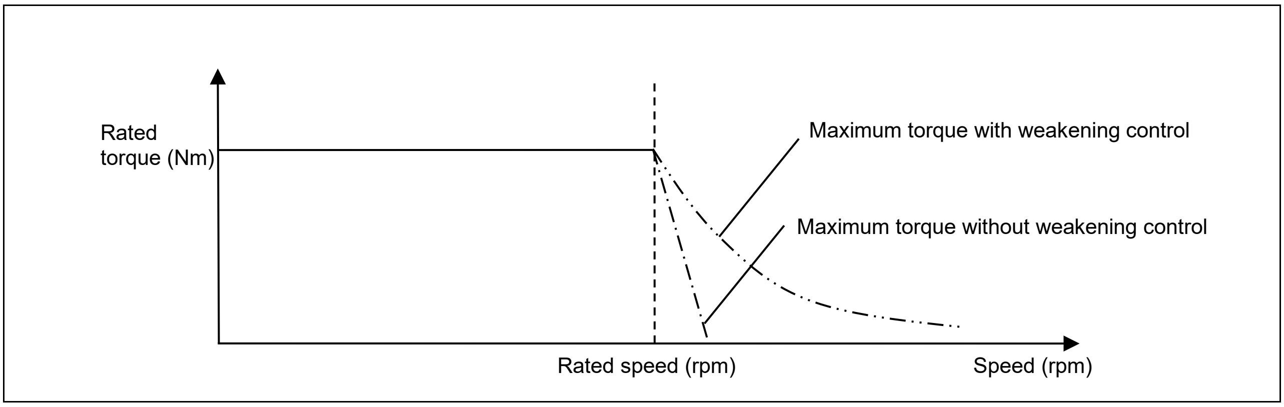


Figure 7-4 Relationship between the Available Output Torque and Speed

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

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

I_a: Magnitude of current vector (A)

Figure 7-5 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 7-6 Equation for Calculating the d-axis Current Command Value in Flux Weakening Control

7.6 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.

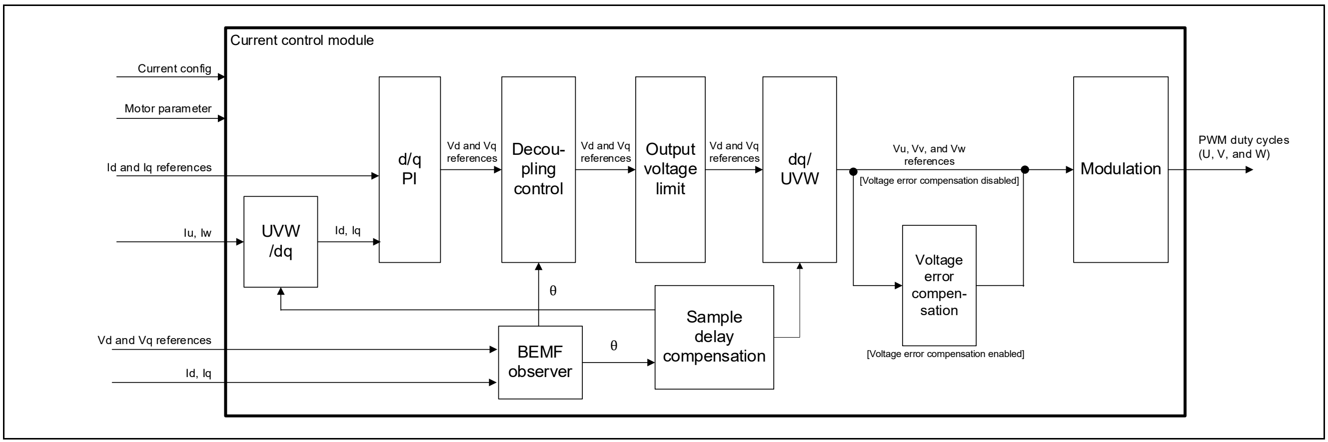


Figure 7-7 Functional Block Diagram for Current Control

7.7 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.

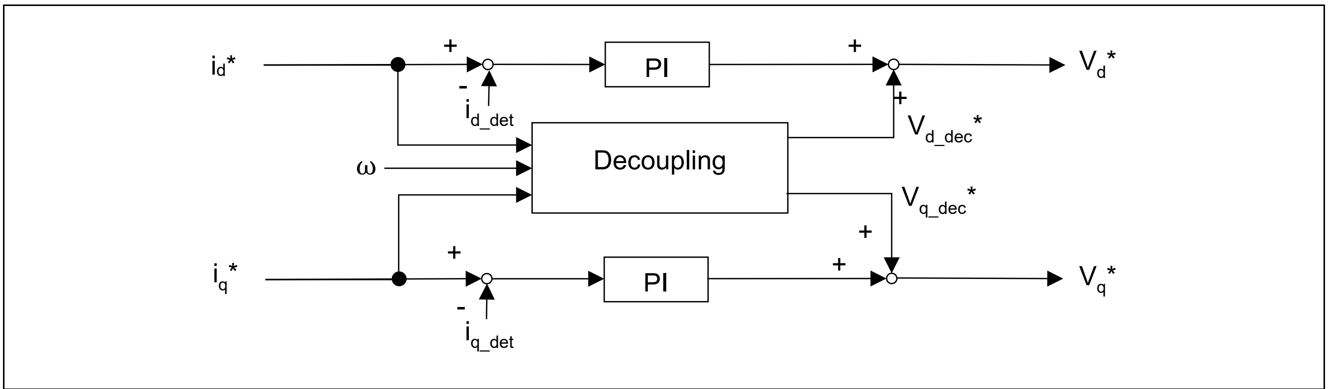


Figure 7-8 Functional Block Diagram of Decoupling Control

7.8 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.

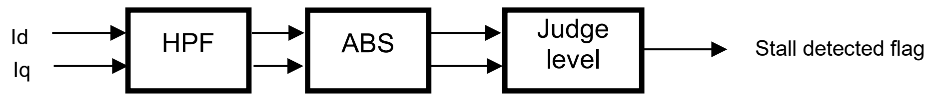


Figure 7-9 Functional Block Diagram of Step-Skipping (Stall) Detection

7.9 Torque Vibration Suppression

7.9.1 Overview

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 a compressor, the rotation of a motor is used to compress refrigerant gas or air and the load fluctuates according to the progress in the compression, exhaust, expansion, and intake processes. Such fluctuations in load (= torque vibration) cause fluctuations in the motor's speed of rotation. This can make the output from a compressor unstable or generate the vibration of devices, which leads to noise or failures. Using the torque vibration suppression function reduces fluctuations in the motor's speed of rotation due to torque vibration and effects such as stabilization of the output from the compressor and reduction of the vibration of devices can be expected. 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. Therefore, suppression of vibration may in some cases not be fully effective due to the parameter settings or the configuration of the compressor or equipment.

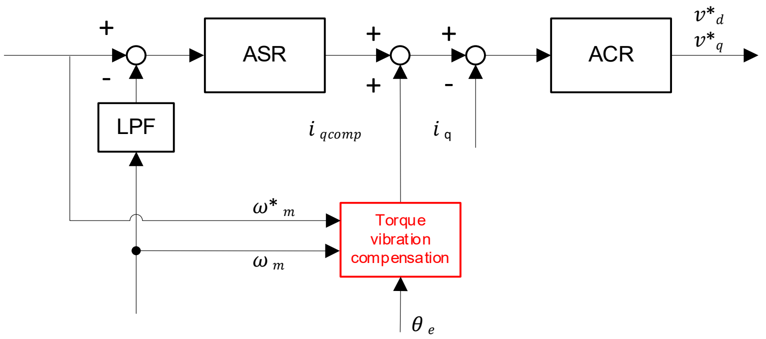


Figure 7-10 Functional Block Diagram of Torque Vibration Suppression

7.9.2 Principles of Control

This section gives an overview of the algorithm used to suppress torque vibration that is implemented in this sample program. Figure 7-11 shows an outline of the torque vibration suppression function. The torque vibration of a single-rotary or reciprocating compressor is proportional to the speed of rotation and depends on the mechanical angle, so its periodicity is related to the rotation frequency (hereafter, the value of the motor rotation speed command converted from mechanical degrees to frequency is referred to as the rotation frequency). To reduce the effect of the torque vibration of a compressor, a compensating signal for cancelling the torque vibration is generated according to the mechanical angle and added to the q-axis current command values (for torque current) in a feedforward manner. The sample software uses a tracking filter and repetitive control to generate the compensating signal. In repetitive control, the compensating signal for the one-cycle earlier vibration component of the input signal is added to the current vibration component and the result is output (hereafter, the controller of this process is referred to as the repetitive controller). As the vibration component is damped through suppression control, the input is also attenuated and a stationary compensating signal can be learned. Here, the vibration component contained in the speed fluctuations is the target for suppression. The load characteristic of a compressor is generated at integer multiples of the rotation frequency and the first-order component, second-order component, and the like of the rotation frequency are the relevant vibration components. Therefore, a tracking filter (TF) is used to divide the speed fluctuation component into vibration components, repetitive control is applied to each of these, and compensating signals for the individual components are generated to obtain a final compensating signal for negating the load fluctuations in a compressor. For details of the repetitive control and tracking filter, refer to technical documents on those topics.

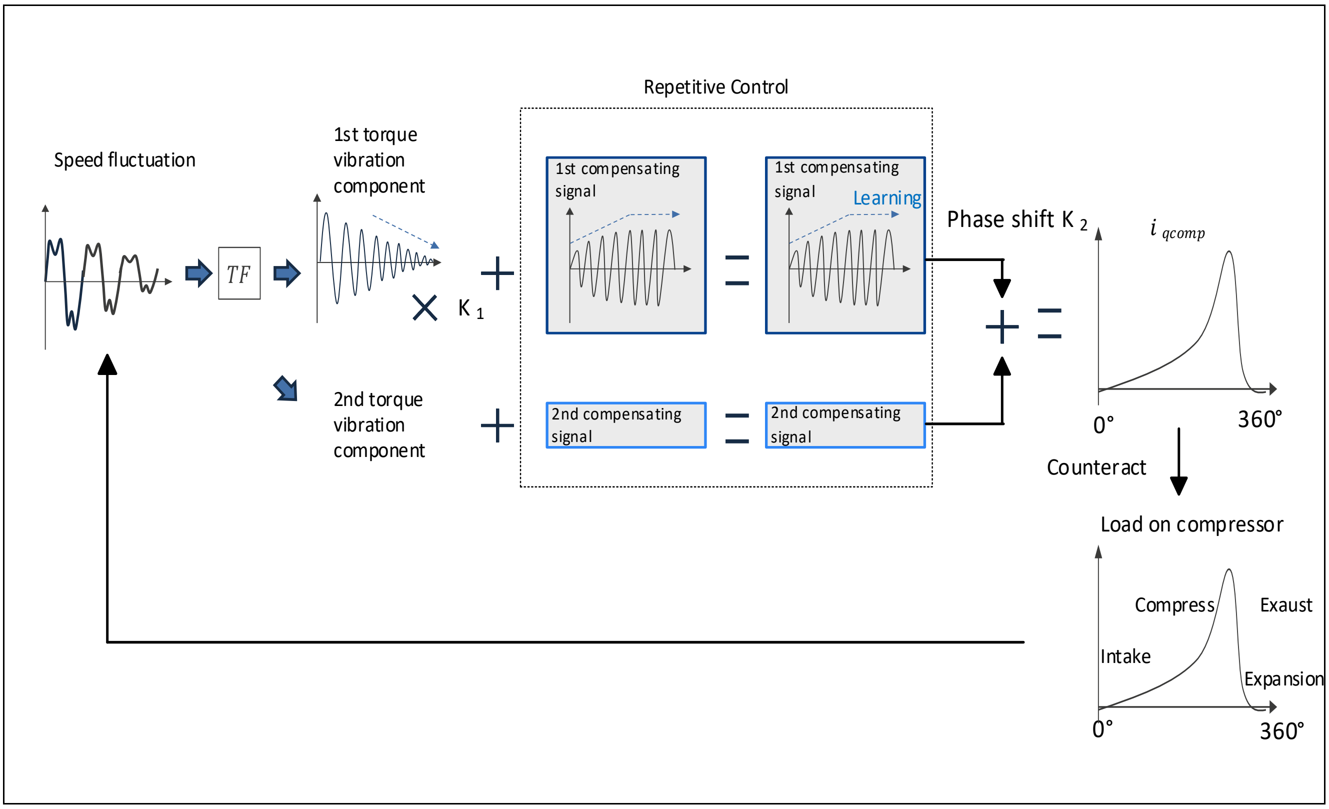


Figure 7-11 Outline of Torque Vibration Suppression by Using a Tracking Filter and Repetitive Control

7.9.3 Implementing a Repetitive Controller in the MCU

A repetitive controller should have a structure that adds the output value from one cycle of the controller frequency earlier than the current cycle to the current input to the controller and outputs the result. Therefore, the one-cycler earlier compensating signals must be retained. As the MCU handles data as discrete values, a table (array) of compensation values obtained by dividing one cycle in mechanical degrees by N is prepared and the compensation values for the individual angles are stored in the table to retain the compensation values for one cycle.

Table 7-2 Table of Compensation Values (N = 256)

Index	Angle	Compensation Value (A)
0	0	0.01
1	1.41	0.02
2	2.81	0.02
:	:	:
254	357.18	–0.0005
255	358.59	0.0

This sample program provides two methods of implementing the generation of a compensating signal from a table of compensation values as shown in Table 7-2. One is a look-up table (LUT) method, in which the table of compensation values is read at every desired angle for the output of a compensating signal and the values read are output as the compensating signal. The other is a polynomial approximation technique (PAT) method, in which polynomial approximations with a compensating signal taken as a function of the pole position proceed and the polynomial coefficients are obtained for use in generating the compensating signal. The PAT method can generate a higher-resolution compensating signal in a smaller area of memory in comparison with the LUT method. A macro is provided for switching between these two methods.

(a) Look-up table (LUT) method

With this method, the table of compensation values is read at desired angles for the output of a compensating signal and the values are output as the compensating signal. The size of the table (N) corresponds to the resolution of mechanical angle for the compensating signal.

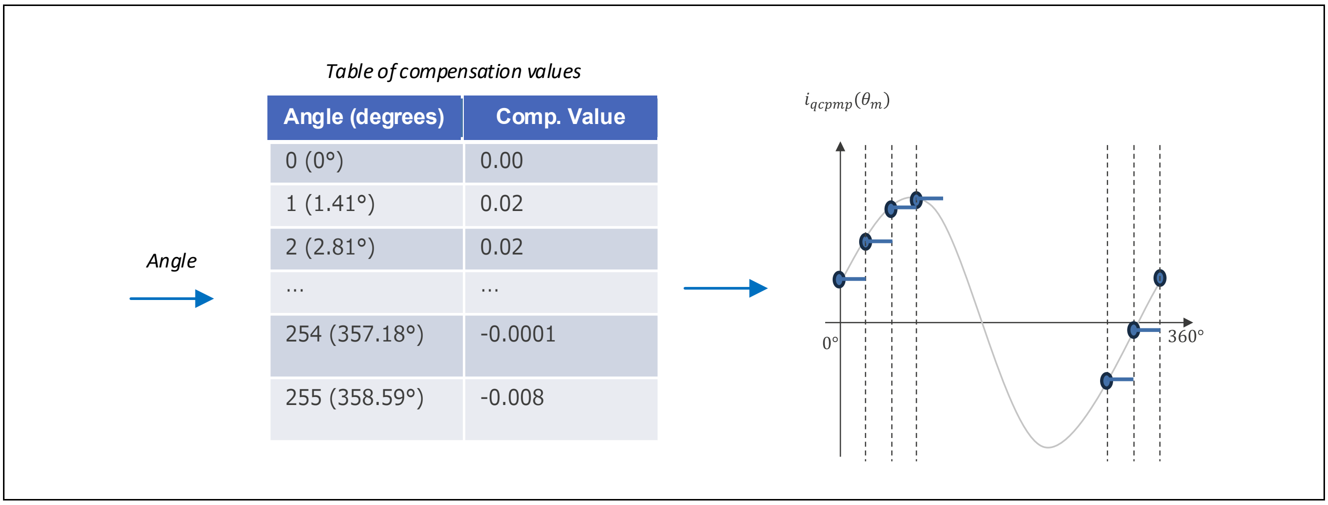


Figure 7-12 LUT Method

(b) Polynomial approximation technique (PAT) method

This method differs from the LUT method in that polynomial approximations are used to generate the compensating signal from the table of compensation values. It can suppress vibration with the use of less RAM (a smaller number of array elements) and can also suppress the effect of harmonic currents through complementation by approximations.

Table of compensation values

Angle (degrees)	Comp. Value
0 (0°)	0.00
1 (72.0°)	0.045
2 (144.0°)	0.02
3 (216.0°)	0.00
4 (288.0°)	-0.01
5 (360.0°)	0.00

Solve a linear system

$$Ax = y$$

$$\begin{bmatrix} 1 & i_0 & i_0^2 & i_0^3 & i_0^4 & i_0^5 \\ 1 & i_1 & i_1^2 & i_1^3 & i_1^4 & i_1^5 \\ 1 & i_2 & i_2^2 & i_2^3 & i_2^4 & i_2^5 \\ 1 & i_3 & i_3^2 & i_3^3 & i_3^4 & i_3^5 \\ 1 & i_4 & i_4^2 & i_4^3 & i_4^4 & i_4^5 \\ 1 & i_5 & i_5^2 & i_5^3 & i_5^4 & i_5^5 \end{bmatrix} \begin{bmatrix} f \\ e \\ d \\ c \\ b \\ a \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}$$

Solve

$$x = A^{-1}y$$

i : Index of mechanical angle

a to e : Polynomial coefficients

y : *Compensation value*

$i_q(\theta_m)$: *Compensation signal*

Polynomial approximation

$$I_{qcomp}(\theta_m) = a \times (\theta_m)^5 + b \times (\theta_m)^4 + c \times (\theta_m)^3 + d \times (\theta_m)^2 + e \times (\theta_m) + f$$

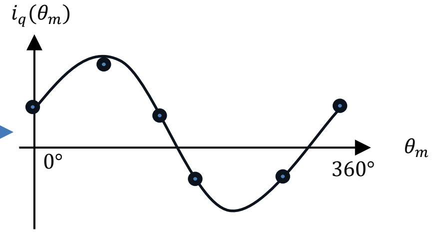


Figure 7-13 PAT Method

7.9.4 Advance Compensation

Torque vibration in a compressor has multiple frequency components because it is generated at integer multiples of the rotation frequency. Therefore, the phase characteristics of speed fluctuations and of the compensating signal may differ in some cases. In the sample software, a mechanical angle (reference mechanical angle) to which advance compensation has been applied is used in generating the compensating signal for the stable suppression of torque vibration. Specifically, in the LUT method, a reference mechanical angle that is offset from the current mechanical angle by the lead angle K_2 is used in the reading and output of compensation values from the table of compensation values. Likewise, in the PAT method, a reference mechanical angle that is offset from the current mechanical angle by the lead angle K_2 is used together with the polynomial approximation coefficients in generating the compensating signal.

7.9.5 Learning Function

The torque vibration suppression function implemented in this sample software performs the learning of compensating signals during motor operation. The purpose of this learning is to generate compensating signals from relative mechanical angles obtained every time driving operation proceeds instead of from absolute mechanical angles. This is because using an expensive position sensor may be undesirable due to the desire for low-cost and environmentally resistant compressors and absolute mechanical angles cannot be detected in such compressors. Figure 7-14 is a block diagram of the torque vibration suppression function implemented in this sample software. The target vibration components for control by the sample software are defined as the first-order and second-order components of the rotation frequency. Accordingly, two repetitive controllers for generating compensating signals are placed in parallel. To simplify the system, only a single tracking filter is used to detect the torque vibration components (TVC) from the speed fluctuations.

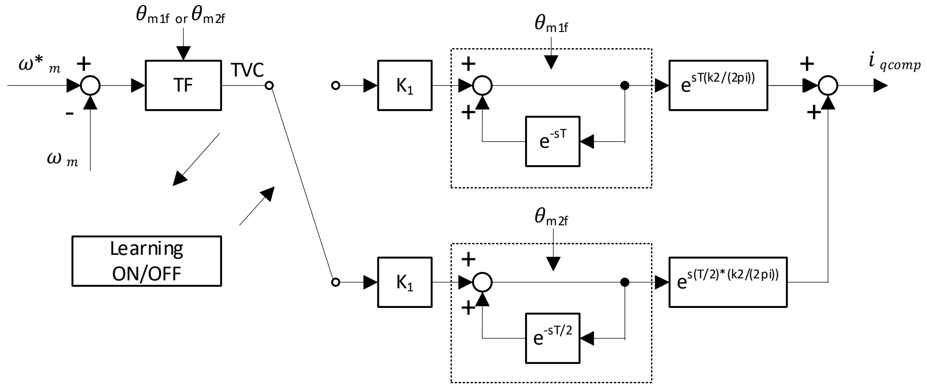


Figure 7-14 Block Diagram of the Torque Vibration Suppression Function

Whether to start or stop learning and whether to switch the target vibration component for compensation require judgment during motor operation. The following describes the conditions for starting and stopping learning.

- Condition for starting learning

  The vibration component output from the tracking filter (TF) is stable.

- Conditions for stopping learning

  Learning stops if either of the following conditions is satisfied.

(a) The monitored amplitude of the vibration component having reached the goal value for suppression

The amplitude of the vibration component output from the TF is monitored. If this amplitude has reached the prespecified goal value for suppression, learning is stopped. As described above, low cost is important for a compressor and using a speed sensor, a vibration meter, or an accelerometer for evaluating the attenuation of the vibration component is not desirable. Therefore, this system judges the effect of suppression from the amplitude of the extracted vibration component.

(b) The ratio of abnormal output from the TF having exceeded the threshold value

The ratio of abnormal output from the TF is monitored. If this ratio has exceeded the prespecified threshold value, learning is stopped. Although the TF can detect a desired frequency component through a simple system, if a large component exists around the target frequency component to be extracted, a component that is not desired may be extracted together with it. The torque suppression function attenuates the vibration component, and the magnitude of the vibration component relative to the other frequency components becomes smaller as suppression proceeds. Especially during low-speed operation, the interval between the first-order and second-order components of the rotation frequency is close and the effect of the second-order component appears larger in the output from the TF as the first-order component is made smaller by suppression. As a result, depending on the phase of the compensating signal, erroneous learning of a compensating signal may occur and unexpected types of operation, such as the output of an overcurrent, may be generated by the accumulation of compensation values. In the sample software, the effect of undesired vibration components on the target vibration components in the output from the TF is defined as the ratio of abnormal output from the TF. Figure 7-15 shows the ratio of abnormal output from the TF, the output from the TF, and the compensating signals in learning of the compensating signal for the first-order component during low-speed operation. The ratio of abnormal output from the TF increases with progress in suppression of the vibration component. If learning continues after the ratio has exceeded 1.0, the second-order component, which is not the target of learning, will become dominant in the output from the TF. Therefore, stopping learning before the ratio exceeds 1.0 can prevent erroneous learning of a compensating signal.

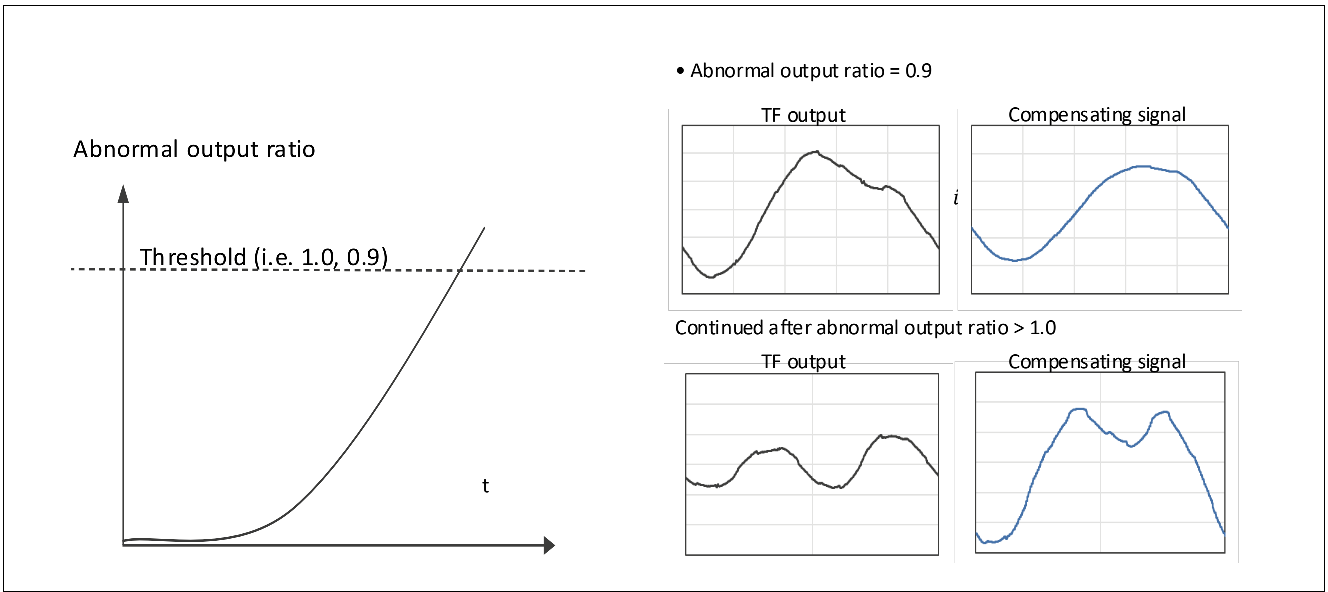


Figure 7-15 Ratio of Abnormal Output from the TF, Outputs from the TF, and Compensating Signals during Suppression Control

Table 7-3 lists the states of the torque vibration suppression function and Table 7-4 lists the actions that cause changes in the state. Figure 7-16 shows an image of the suppression of fluctuations in the motor's rotation speed in the individual states of the torque vibration suppression function.

Table 7-3 States of Torque Vibration Suppression

State	Description	Output of Compensating Signal
IDLE	The torque vibration suppression function is disabled.	None
1F STANDBY	Waiting for extraction of the first-order component of the rotation frequency.	None
1F LEARNING	Learning of the compensating signal for the first-order component of the rotation frequency is in progress.	Compensating signal for the first-order component of the rotation frequency
2F STANDBY	Waiting for extraction of the second-order component after learning of the compensating signal for the first-order component of rotation frequency is complete.	Compensating signal for the first-order component of the rotation frequency
2F LEARNING	Learning of the compensating signal for the second-order component is in progress after learning of the compensating signal for the first-order component of the rotation frequency is complete.	Compensating signal for the first-order component + second-order component of the rotation frequency
COMPLETE	Learning of the compensating signals was completed.	Compensating signal for the first-order component + second-order component of the rotation frequency

Table 7-4 Actions in Torque Vibration Suppression

Action	Description
RESET	The torque vibration suppression function is disabled. The compensating signals and compensation tables are reset.
START	The torque vibration suppression function is enabled.
LEARNING_ON	The condition for starting learning of a compensating signal has been satisfied and learning is started.
LEARNING_OFF	A condition for stopping learning of a compensating signal has been satisfied and learning is stopped.

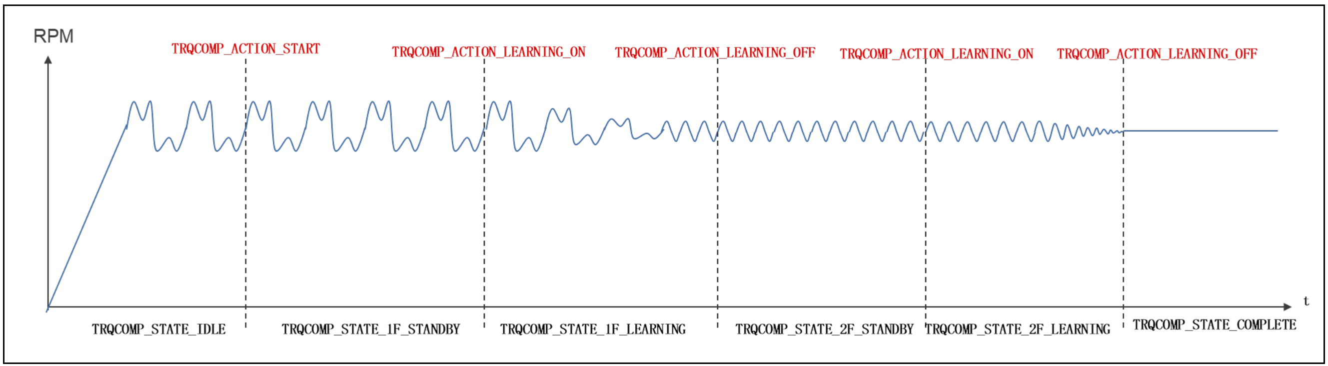


Figure 7-16 Transitions between States in Torque Vibration Suppression

7.10 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 7-18) 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 7-17 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.

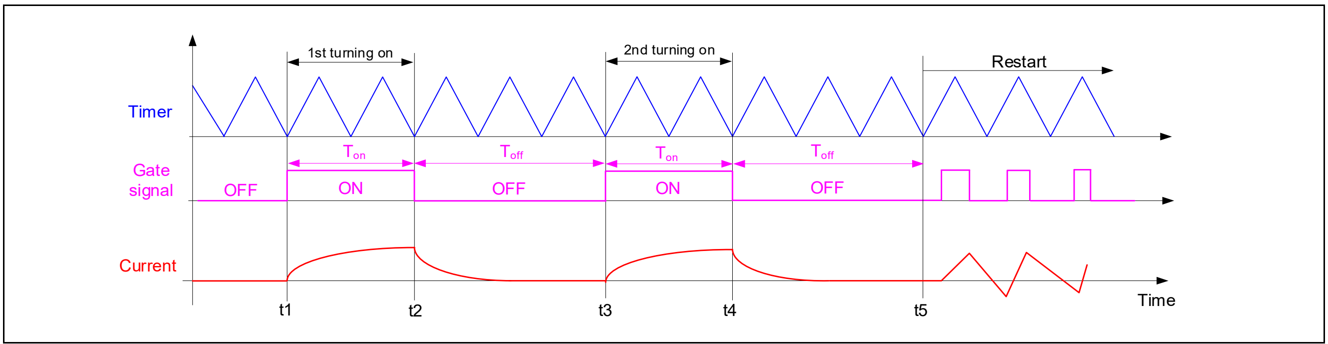


Figure 7-17 Sample Waveform of Flying Start Operation

(1) Detection of rotational velocity

Figure 7-18 shows the relationship between the phases of the rotation current vectors obtained by turning on twice. Two-phase currents iα and iβ are calculated from the three-phase currents iu, iv, and iw and the phase angles θ1 and θ2 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 θ1 and θ2 and the pulse-on and off times Ton and Toff, the electrical angular velocity of rotation ω is calculated by using equation 7.10.1.

$$\omega = \frac{\theta_2 - \theta_1}{T_{on} + T_{off}}$$ Equation 7.10.1

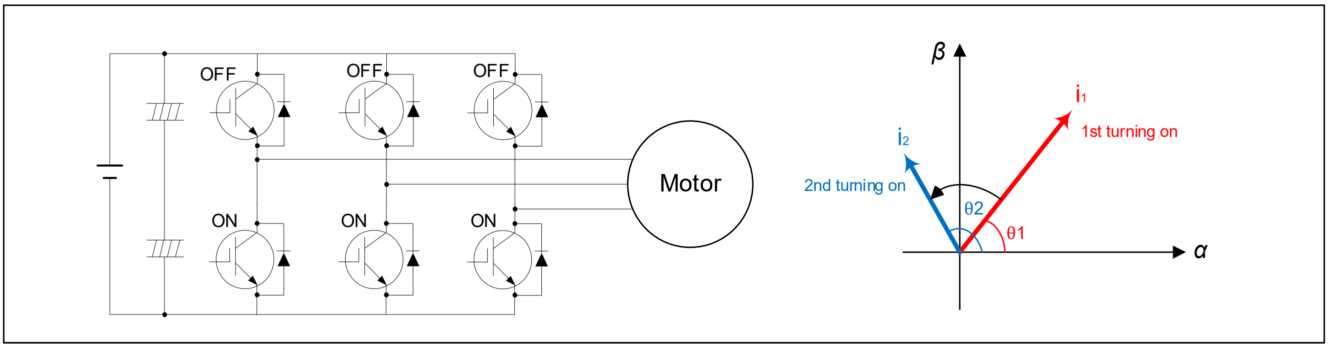


Figure 7-18 Trajectory of Current Vector by Turning on Twice

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.

$$\mathrm{Ton} + \mathrm{Toff} < \pi/\omega_{max} \qquad \text{Condition 7.10.2}$$

(2) Detection of magnetic pole position

The voltage equation in the dq-axis rotation coordinate system is given as equation 7.10.3.

$$\begin{bmatrix} \mathrm{v_d} \\ \mathrm{v_q} \end{bmatrix} = \begin{bmatrix} \mathrm{R} + \mathrm{pL_d} & -\omega \mathrm{L_q} \\ \omega \mathrm{L_d} & \mathrm{R} + \mathrm{pL_q} \end{bmatrix} \begin{bmatrix} \mathrm{i_d} \\ \mathrm{i_q} \end{bmatrix} + \omega\psi \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \text{Equation 7.10.3}$$

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 7.10.4 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 ($R \simeq 0.0$).

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathrm{pL_d} & -\omega \mathrm{L_q} \\ \omega \mathrm{L_d} & \mathrm{pL_q} \end{bmatrix} \begin{bmatrix} \mathrm{i_d} \\ \mathrm{i_q} \end{bmatrix} + \omega\psi \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \text{Equation 7.10.4}$$

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 7.10.5 is obtained.

$$\mathrm{i(T)} = \begin{bmatrix} \mathrm{i_d(T)} \\ \mathrm{i_q(T)} \end{bmatrix} = \begin{bmatrix} -\frac{\psi}{\mathrm{L_d}}(1 - \cos\omega \mathrm{T}) \\ -\frac{\psi}{\mathrm{L_q}}\sin\omega \mathrm{T} \end{bmatrix} \qquad \text{Equation 7.10.5}$$

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.

$$\theta_\mathrm{a} = \mathrm{atan2}\left(\frac{\mathrm{i_q}}{\mathrm{i_d}}\right) = \mathrm{atan2}\left(\frac{-\frac{\psi}{\mathrm{L_q}}\sin\omega \mathrm{T_{on}}}{-\frac{\psi}{\mathrm{L_d}}(1-\cos\omega \mathrm{T_{on}})}\right) = \mathrm{atan2}\left(\frac{\mathrm{L_d}\sin\omega \mathrm{T_{on}}}{\mathrm{L_q}(1-\cos\omega \mathrm{T_{on}})}\right) \qquad \text{Equation 7.10.6}$$

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.

$$\theta_r = \theta_i - \theta_a = \mathrm{atan2}\left(\frac{i_\beta}{i_\alpha}\right) - \mathrm{atan2}\left(\frac{i_q}{i_d}\right) \qquad \text{Equation 7.10.7}$$

Figure 7-19 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 Iₐ from the d axis and θ_i is the phase angle of the current vector Iₐ from the α axis.

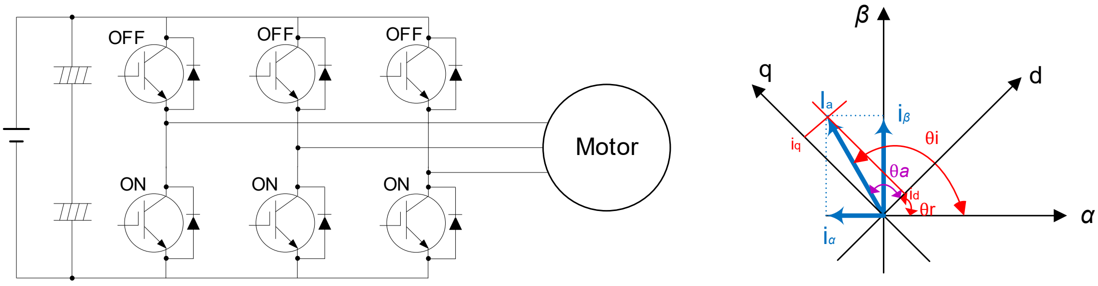


Figure 7-19 Relationship between the Current Vectors and Magnetic Pole Positions at the Second Time of Turning on

(3) Design of control parameters

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

Table 7-5 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 7.10.1. (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 10.15.
Toff time	See the description of SENSORLESS_VECTOR_FLY_START_OFF_TIME_SEC in section 10.15.

7.11 Sensorless Control Function

7.11.1 Overview

The open-loop startup processing through current-drawn control is done in the zero-speed to low-speed range (600 rpm and slower). The BEMF observer is used for magnetic pole position estimation in the medium-to-high-speed range (600 rpm and faster). A combination of these methods achieves sensorless vector control.

In open-loop control, the load should be limited to half or less of the rated load to prevent step-skipping of (stalling by) the motor. Note that the motor may rotate by up to 180 electrical degrees at startup during operation under open-loop control.

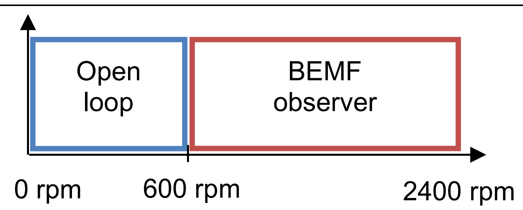


Figure 7-20 Sensorless Control Algorithms Corresponding to the Speed Ranges

7.11.2 Current-Drawn Control (Open Loop)

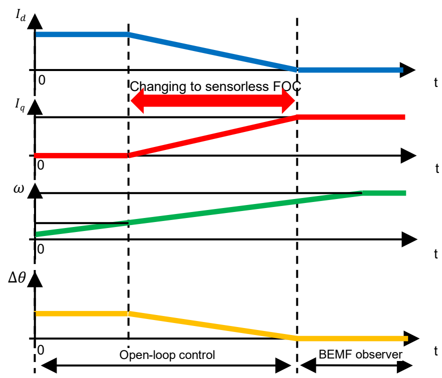
(a) Overview

In the low-speed range, current-drawn control is performed by supplying a positive-direction current through the d axis to cause forced excitation. As this is open-loop control, the applicable loads are only several tens of percent of the rated load. To apply the rated load, drive the motor in the later-described mode of sensorless vector control (current feedback control) with the use of the BEMF observer in the medium-to-high-speed range.

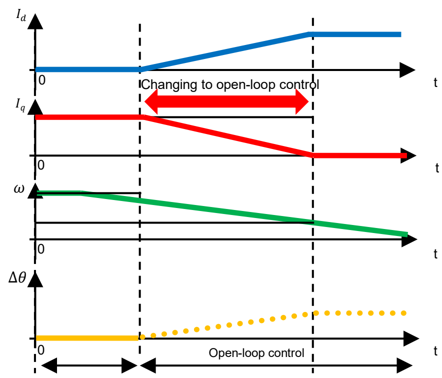
(b) Switching to and from the medium-to-high-speed range

After the motor is started, control is switched to the sensorless method (closed-loop speed control) when the speed has reached a high enough level for estimation of the inductive voltage. Note that hunting of the current and speed may occur due to a phase error when control is switched from the open-loop method to the sensorless method. Therefore, the load torque is estimated from the phase error Δθ and the processing for switching to sensorless control proceeds as shown in Table 7-6. When the sensorless control algorithm switches from the low-speed range to the medium-to-high-speed range, the state sequence is made to operate so that current fluctuations are reduced by adjusting the d-axis and q-axis current commands. On the other hand, 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, operation is switched to open-loop control. The speeds for switching control during acceleration and deceleration need to be sufficiently separated so that switching does not occur frequently. These speeds can be adjusted by using parameters described later. Hunting of the current and speed at the time of switching of control can thus be reduced.

Table 7-6 Behavior of Physical Quantities in the Switching of Sensorless Control during Acceleration and Deceleration



Acceleration



Deceleration

7.11.3 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.

7.12 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 any number of samples of the control interval from the estimated angle. This process improves the stability of control.

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.

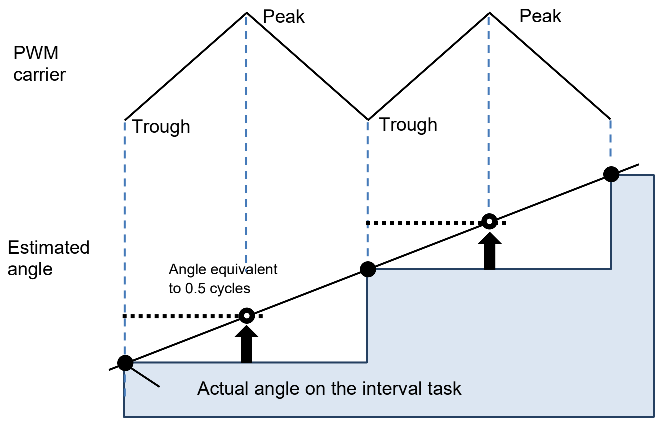


Figure 7-21 Example of the Amount by which the Angle is Advanced in a PWM Carrier Cycle

7.13 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.

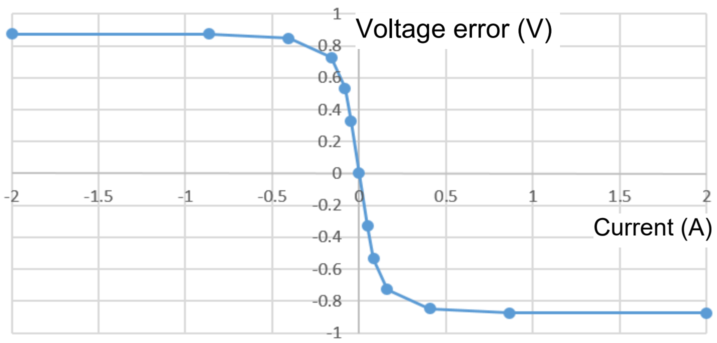


Figure 7-22 Example of the Dependency of the Voltage Error on the Current

7.14 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.

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

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.

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

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

V_u, V_v, V_w: 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.

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

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

8. Hardware Specifications

8.1 User interface

User interface in board circuit is listed below Table 8-1.

Table 8-1 User interface in board circuit

Item	Interface	Function
Rotation speed	Variable resister (VR1)	Input speed reference
START/STOP	Toggle switch (SW1)	Start/Stop motor rotation
ERROR RESET	Push switch (SW2)	Reset from error state.
LED1	Orange LED (LED1)	・Motor driven : lighten ・Motor stop : lights out
LED2	Orange LED (LED2)	・Error happen : lighten ・No error : lights out
LED3	Orange LED (LED3)	No use
RESET	Push switch (RESET1)	System reset

Pin interface of this sample software is listed below Table 8-2 and Table 8-3.

Table 8-2 Pin interface [1/2]

Function	RA6T2	RA8T1	RA8T2
Measure inverter bus voltage	PA07 / AN007	P008 / AN008	P007 / AN007
Input speed reference (VR1)	P000 / AN016	P014 / AN007	P015 / AN015
START/STOP Toggle switch (SW1)	PD04	PA15	PA00
Error reset push switch (SW2)	PD07	PA13	PA07
LED1 Light control	PD01	PA12	P614
LED2 Light control	PD02	PA14	PA15
Measure current of U-phase	PA04 / AN004	P004 / AN000	P006 / AN006
Measure current of V-phase	PA02 / AN002	P005 / AN001	P008 / AN008
Measure current of W-phase	PA00 / AN000	P006 / AN002	P010 / AN010
PWM Output (Up)	PB04 / GTIOC4A	P115 / GTIOC5A	P605 / GTIOC8A
PWM Output (Vp)	PB06 / GTIOC5A	P113 / GTIOC2A	P603 / GTIOC7A
PWM Output (Wp)	PB08 / GTIOC6A	P300 / GTIOC3A	P612 / GTIOC9A
PWM Output (Un)	PB05 / GTIOC4B	P609 / GTIOC5B	P604 / GTIOC8B
PWM Output (Vn)	PB07 / GTIOC5B	P114 / GTIOC2B	P602 / GTIOC7B
PWM Output (Wn)	PB09 / GTIOC6B	P112 / GTIOC3B	P613 / GTIOC9B
Emergency PWM stop input at overcurrent	PC13 / GTETRGD	P613 / GTETRGA	P112 / GTETRGA

Table 8-3 Pin interface [2/2]

Functtion	RX26T Type A
Measure inverter bus voltage	P43 / AN003
Input speed reference (VR1)	P50 / AN204
START/STOP Toggle switch (SW1)	P23
Error reset push switch (SW2)	P22
LED1 Light control	P21
LED2 Light control	P20
Measure current of U-phase	P40 / AN000
Measure current of V-phase	P41 / AN001
Measure current of W-phase	P42 / AN002
PWM Output (Up)	P73 / GTIOC2A
PWM Output (Vp)	P72 / GTIOC1A
PWM Output (Wp)	P71 / GTIOC0A
PWM Output (Un)	P76 / GTIOC2B
PWM Output (Vn)	P75 / GTIOC1B
PWM Output (Wn)	P74 / GTIOC0B
Emergency PWM stop input at overcurrent	P70 / GTETRGB

8.2 Peripheral functions

Peripheral functions which are used in sample program are listed below Table 8-4 and Table 8-5.

Table 8-4 Peripheral functions [1/2]

Peripheral	Purpose	RA6T2	RA8T1	RA8T2
A/D converter	Measure current of U-phase	AN004	AN000	AN006
	Measure current of V-phase	AN002	AN001	AN008
	Measure current of W-phase	AN000	AN002	AN010
	Measure inverter bus voltage	AN007	AN008	AN007
	Measure VR input	AN016	AN007	AN015
AGTW	Interval timer for speed control	AGT0	AGT0	AGT0
GPT	PWM output of U-phase	CH4	CH5	CH8
	PWM output of V-phase	CH5	CH2	CH7
	PWM output of W-phase	CH6	CH3	CH9
POEG	Emergency stop input of overcurrent	Group D	Group A	Group A

Table 8-5 Peripheral functions [2/2]

Peripheral	Purpose	RX26T Type A
A/D converter	Measure current of U-phase	AN000
	Measure current of V-phase	AN001
	Measure current of W-phase	AN002
	Measure inverter bus voltage	AN003
	Measure VR input	AN204
CMT	Interval timer for speed control	CMT0
GPT	PWM output of U-phase	CH2
	PWM output of V-phase	CH1
	PWM output of W-phase	CH0
POEG	Emergency stop input of overcurrent	Group B

(1) A/D converter

A/D converter measure current of U-phase(Iu), current of V-phase(Iv), current of W-phase(Iw), inverter bus voltage(Vdc), and Speed reference input (VR) with “Single scan mode” (use hardware trigger).

A/D conversion is synchronized with underflow of GPT (trough of PWM). The interrupt of A/D conversion finish is used as current control period interrupt.

(2) General asynchronous timer (AGTW)

AGTW is used as cyclic interval timer to generate speed control period interrupt for RA Family MCU.

(3) Compare Match Timer (CMT)
CMT is used as cyclic interval timer to generate speed control period interrupt for RX26T.

(4) General purpose timer (GPT)
GPT outputs complementally PWM by its original mode with dead time.

(5) Port output enable for GPT (POEG)
When overcurrent is detected (low level input at GTETRGx pin), set PWM output ports are all high-impedance state.

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	SW1 input or input from the RMW	
Rotor magnetic pole position detection	Sensorless control (current-drawn control and BEMF observer)	
Input voltage	24 [VDC]	
Main clock frequency	RA6T2: 240 [MHz] RA8T1: 480 [MHz] RA8T2: 1 [GHz] RX26T: 120 [MHz]	
PWM carrier frequencies	20 kHz (Carrier period: 50 [μs])	
PWM mode	Space vector modulation (able to select sinusoidal modulation)	
Dead time	1.0 [μs]	
Control cycle	Current control period	RA6T2: 50 [μs] RA8T1: 50 [μs] RA8T2 :50[μs] RX26T: 50 [μs]
	Speed control period	RA6T2: 500 [μs] RA8T1: 500 [μs] RA8T2: 500 [μs] RX26T: 500 [μs]
Speed command value management	CW: 0 to 2400 [rpm] CCW: 0 to –2400 [rpm] However, 600 [rpm] or less is driven by a speed open loop.	
Natural frequency for each control system	Current control system: 300 [Hz] Speed control system: 5 [Hz] BEMF observer: 1000 [Hz] Position estimation PLL: 20 [Hz]	
Optimization setting of compiler	Optimization level	Optimize more (-O2) (default setting)
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 4.72(=1.67*sqrt (2)*2.0) [A] (monitored in current control period). 2. The inverter bus voltage exceeds 60 [V] (monitored in current control period). 3. The inverter bus voltage is lower than 8 [V] (monitored in current control period). 4. The rotational velocity exceeds 4500 [rpm] (monitored in current control period). 5. The overcurrent detection signal (POE/POEG) is detected. When an external over current signal is detected (when a low level is detected), the PWM output ports are set to high impedance state. 6. A step-skipping (stalled) state is detected if the step-skipping (stall) detection function is enabled (monitored in current control period).	

9.2 Overall Configuration of the Software

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

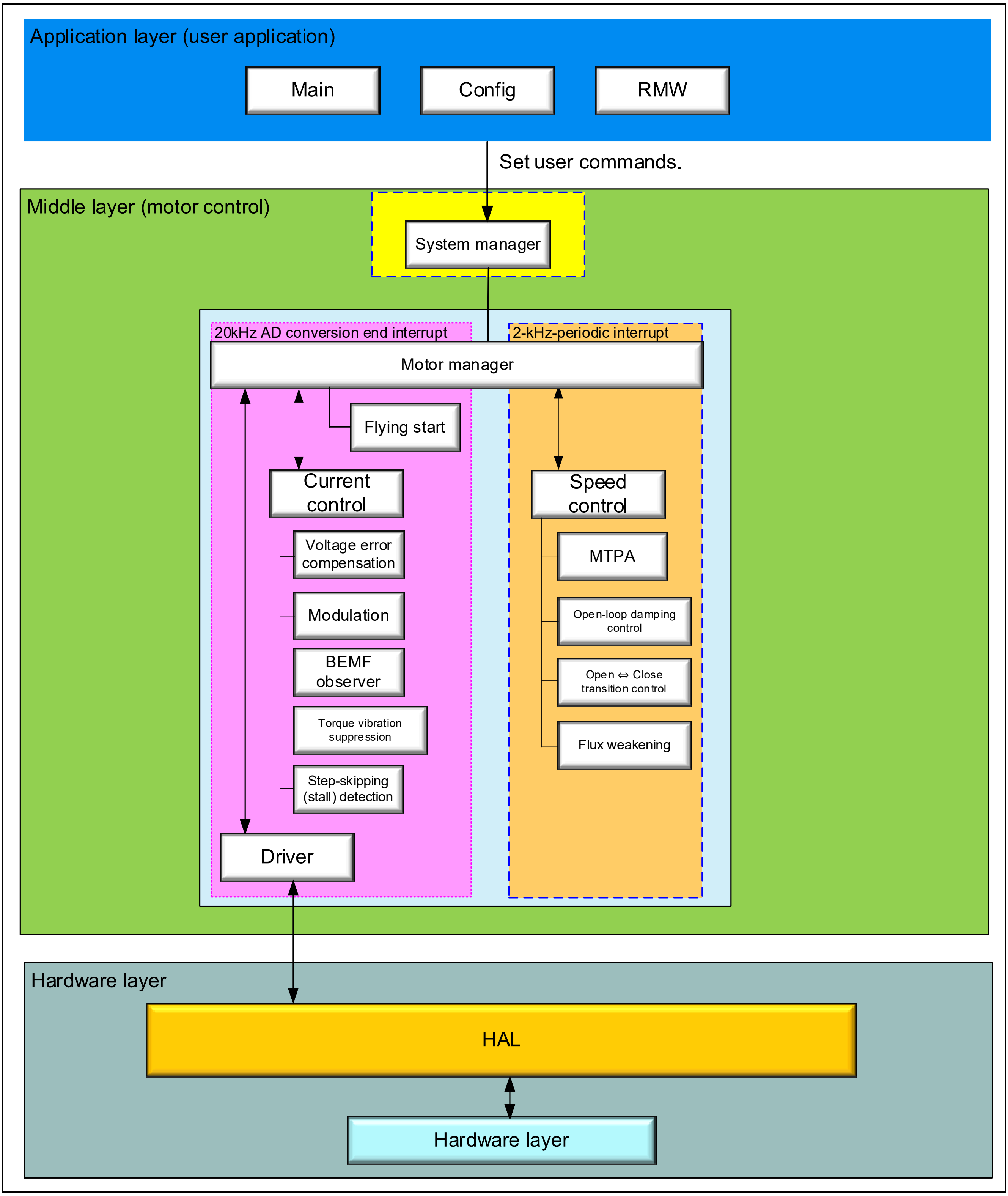


Figure 9-1 Overall Configuration of the Sample Program

9.3 Task Descriptions

For motor control, a task for speed control at 0.5ms intervals and that for current control at 20 kHz, 50 μs are used.

Table 9-2 Interrupts and Tasks Used for RA Family MCU

Task	Peripheral Module	Interval	Interrupt Function	Description
Motor control interrupt (for speed control)	agt0	500 μs	callback_agt_motor_speed_cyclic	
Motor control interrupt (for current control through 3-shunt current detection)	adc0	50 μs	callback_gpt_adc_cyclic	
Task on a reset	—		Note: Executed in the state transition processing when recovering from an 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	

Table 9-3 Interrupts and Tasks Used for RX26T

Task	Peripheral Module	Interval	Interrupt Function	Description
Motor control interrupt (for speed control)	CMT0	500 μs	r_Config_CMT0_cmi0_interrupt	
Motor control interrupt (for current control through 3-shunt current detection)	S12AD0	50 μs	r_Config_S12AD0_interrupt	
Task on a reset	—		Note: Executed in the state transition processing when recovering from an error.	
Motor output overcurrent error interrupt	POEGB		r_Config_POEG_poeggbi_interrupt	
RMW operation	—		r_app_rmw_ui_mainloop	

9.4 Configuration of Folders and Files

Table 9-4 shows the configuration of the folders and files of the sample program for RA Family MCU.

Table 9-4 Configuration of Folders and Files for RA Family MCU

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_interrupt.c/h	Interrupt functions
		mtr_main.c/h	Main module
src/application/mcu	ra6t2 / ra8t1 / ra8t2	r_app_mcu.c/h	HAL dependency main module
		r_app_mcu_callback.c	Callback processing module
		r_motor_driver_fsp.c	Motor related HAL driver
		r_motor_driver_hal.h	Motor related HAL driver definition
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.a/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.a/h	Definitions of functions for the voltage error compensation module
		r_motor_current_bemf_observer.a/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.a/h	Step-skipping (stall) detection module
		r_motor_current_trq_vib_comp.a/h	Torque vibration suppression 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.a/h	Flux weakening control module
		r_motor_speed_mtpa.c/h	MTPA module
		r_motor_speed_opl_damp_ctrl.a/h	Damping control module

Folder	Subfolder	File	Remarks
		r_motor_speed_opl2less.a/h	Dfinition of functions for switching-to-sensorless processing
		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
	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/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_RAYYY[2].o/h	Communications library for the RMW
		convert.bat	Batch file for MAP file generation
		ElfMapConverter.exe	MAP file generation tool
		ICS2_RAYYY[2]_Built_in.o	Object file for use as built-in to the RMW
		XXX[1].rmt	RMT project file for the RMW
	board_ui	r_app_board_ui.c/h	Definition of MCU-dependent board UI functions

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. Note that:

1. 'XXX' varies depending on the project file.
2. ‘YYY’ depending on RA MCU type. For example, the communication library files for RA6T2 are ICS2_RA6T2.o/h.

Table 9-5 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.

Table 9-6 shows the configuration of the folders and files of the sample program for RX26T.

Table 9-6 Configuration of Folders and Files for RX26T

Folder	**Subfolder**	**File**	**Remarks**
smc_gen	Config_CMT0	Config_CMT0_user.c	Definition of user functions related to CMT0 for the control interval
		Config_CMT0.c/h	Definition of functions related to CMT0 for the control interval
	Config_GPT0	Config_GPT0_user.c	Definition of user functions related to GPT0
		Config_GPT0.c/h	Definition of functions related to GPT0
	Config_GPT1	Config_GPT1_user.c	Definition of user functions related to GPT1
		Config_GPT1.c/h	Definition of functions related to GPT1
	Config_GPT2	Config_GPT2_user.c	Definition of user functions related to GPT2
		Config_GPT2.c/h	Definition of functions related to GPT2
	Config_IWDT	Config_IWDT_user.c	Definition of user functions related to IWDT
		Config_IWDT.c/h	Definition of functions related to IWDT
	Config_POEG	Config_POEG_user.c	Definition of user functions related to POEG
		Config_POEG.c/h	Definition of functions related to POEG
	Config_PORT	Config_PORT_user.c	Definition of user functions related to PORT
		Config_PORT.c/h	Definition of functions related to PORT
	Config_S12AD0	Config_S12AD0_user.c	Definition of user functions related to S12AD0
		Config_S12AD0.c/h	Definition of functions related to S12AD0
	Config_S12AD2	Config_S12AD2_user.c	Definition of user functions related to S12AD2
		Config_S12AD2.c/h	Definition of functions related to S12AD2
	general	General	Modification of the files in these folders is prohibited.
	r_bsp	Board support Package	
	r_config	Configuration for SC	
	r_pincfg	Pin configuration for SC	
src/application		main.c	Startup routine module
src/application/main		mtr_interrupt.c/h	Interrupt functions
		mtr_main.c/h	Main module
src/application/mcu	rx26t	r_app_mcu.c/h	HAL dependency main module
		r_motor_driver_smc.c	Motor related HAL driver
		r_motor_driver_hal.h	Motor related HAL driver definition

Folder	Subfolder	File	Remarks
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
	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_opl_damp_ctrl.lib/h	Damping control module
		r_motor_speed_opl2less.lib/h	Dfinition of functions for switching-to-sensorless processing
		r_motor_speed_pi_gain_calc.c	Definitions of functions for calculating the control gain of the speed control module
src/application/motor_module	driver	r_motor_driver.c/h	Definitions of functions 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

Folder	Subfolder	File	Remarks
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_RX26T.h	Communications library for the RMW
		XXX.rmt	RMT project file for the RMW
		ICS2_RX26T.lib	Object file for use as built-in to the RMW
	Board_ui	r_app_board_ui.c/h	Definition of board UI functions

By using SC, peripheral function drivers can be easily generated from the GUI screen.

SC stores configuration information such as microcontrollers, peripheral functions, and pin functions used in the project in the project file (RX26T_XXX.scfg). To check the peripheral function settings of this sample program, refer to the SC setting screen on e^2 studio. Note that 'XXX' varies depending on the project file.

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 9-7 lists the functions that are performed in the application layer.

Table 9-7 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.
UI processing	Selects of Board UI or RMW UI, and manages these.
RMW UI processing	Manages the RMW and acquires and sets parameters including command values.
Initial settings of the MCU	The FSP/SC is used to make initial settings for the MCU. Calibration and other settings to suit the application also proceed.
Bridge to the FSP/SC	Defines the callback functions assigned to peripheral functions, which are specified through the FSP/SC, and passes them to lower-level modules through the system manager.

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 6-4 in section 6.7. Table 9-8 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 9-8.

Table 9-8 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)
	u2_charge_bootstrap_time	Charging time for the bootstrap circuit (cnt)
	st_motor	Structure for motor parameters
	f4_max_speed_rpm	Maximum speed (rpm) (mechanical angle)
	u1_ctrl_loop_mode	Control loop mode (speed control)
	f4_ol_ref_id	Open-loop control: Id current command value
	f4_id_up_time	Id increase time (cnt)
	f4_id_down_time	Id decrease time (cnt)
	f4_id_down_speed_rpm	Speed for switching the motor control method (accelerating) (rpm)
	f4_id_up_speed_rpm	Speed for switching the motor control method (decelerating) (rpm)
	f4_current_omega_hz	Natural frequency for the current control system (Hz)

Structure	Variable	Description
	f4_current_zeta	Attenuation coefficient for the current control system
	f4_speed_omega_hz	Natural frequency for the speed control system (Hz)
	f4_speed_zeta	Attenuation coefficient for the speed control system
	f4_speed_lpf_hz	Speed LPF cut-off frequency (Hz)
	f4_ref_speed_rpm	Speed command value (rpm) (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_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
	u1_flag_trq_vibration_comp_mode	Torque vibration suppression: Compensation signal generation method
	f4_e_obs_omega_hz	Natural frequency for the inductive voltage estimation system (Hz)
	f4_e_obs_zeta	Attenuation coefficient for the inductive voltage estimation system
	f4_pll_est_omega_hz	Natural frequency for the position estimation system (Hz)
	f4_pll_est_zeta	Attenuation coefficient for the position estimation system
	u1_flag_less_switch_use	Enables or disables the sensorless control soft switching function
	f4_switch_phase_err_deg	Sensorless switching angle error (degrees)
	f4_opl2less_sw_time	Sensorless switching time (s)
	f4_phase_err_lpf_cut_freq	Angle-error LPF frequency (Hz)
	u1_flag_openloop_damping_use	Enables or disables the damping control function
	f4_ed_hpf_omega	Damping control: Natural frequency for HPF (Hz)
	f4_ol_damping_zeta	Damping control: Attenuation coefficient
	f4_ol_damping_fb_limit_rate	Damping control: Feedback limit rate
	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

Structure	Variable	Description
	f4_threshold_level	Step-skipping (stall) detection: Threshold value (A)
	f4_threshold_time	Step-skipping (stall) detection: Monitoring time (s)
	u1_target_2f	Torque vibration suppression: Enables or disables the suppression of secondary components
	f4_timelead_1f	Torque vibration suppression: Phase adjustment value for fundamental components (rad)
	f4_timelead_2f	Torque vibration suppression: Phase adjustment value for secondary components (rad)
	f4_tf_lpf_omega	Torque vibration suppression: LPF cutoff frequency for the TF (Hz)
	f4_output_gain_1f	Torque vibration suppression: TF output gain for fundamental components
	f4_output_gain_2f	Torque vibration suppression: TF output gain for fundamental components
	f4_input_weight2	Torque vibration suppression: Weight 2
	f4_input_weight1	Torque vibration suppression: Weight 1
	f4_input_weight0	Torque vibration suppression: Weight 0
	f4_suppression_th_1f	Torque vibration suppression: Suppression Target for fundamental components
	f4_suppression_th_2f	Torque vibration suppression: Suppression Target for secondary components
	f4_abnormal_output_th_1f	Torque vibration suppression: TF output abnormality for fundamental components
	f4_abnormal_output_th_2f	Torque vibration suppression: TF output abnormality for secondary components
	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	Flying start: Active brake time (s)
	f4_on_current_th	Flying start: Current threshold for switching on (A)

9.5.3 Macro Definitions

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

Table 9-9 List of Macros

File Name	Macro Name	Defined Value	Description
r_app_mcu.h	ICS_DECIMATION	RA6T2, RA8T1, RA8T2, RX26T : 5	RMW watchpoint skip count
	ICS_BRR	RA6T2, RA8T1, RA8T2, RX26T : 19	RMW communications rate
	ICS_INT_MODE	RA6T2, RA8T1, RA8T2, RX26T : 1	RMW communications mode

Note: A macro that defines the channel used for communications via the RMW is provided in ICS2_RA*yyy*2.h or ICS2_RX26T.h, where '*yyy*' is RA MCU type number.

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 or motor 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 6.8 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 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 9-10 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.
Motor manager processing	Acquires and specifies command values for speed control.

9.6.2 Module Configuration Diagram

Figure 9-2 shows the module configuration.

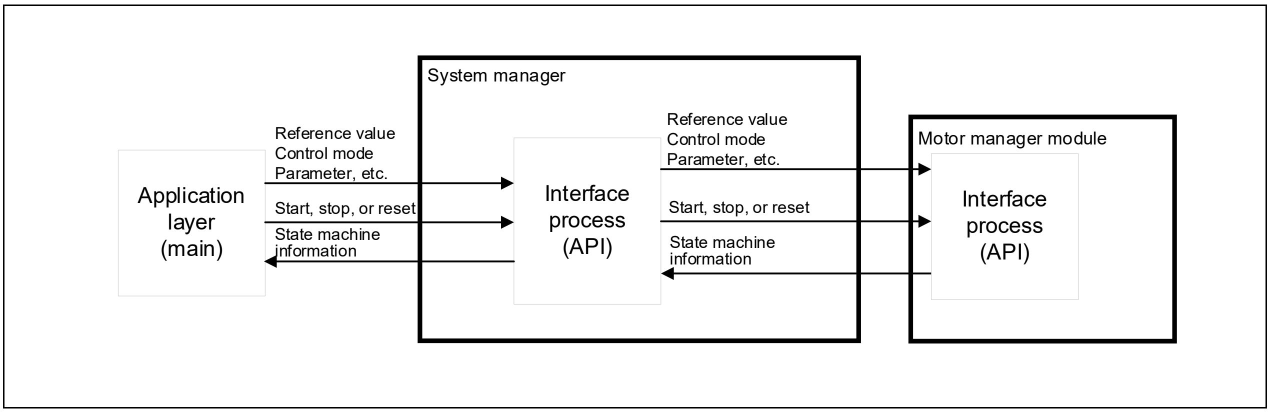


Figure 9-2 Module Configuration of the System Manager

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 9-11 lists the functions of the motor manager module. Table 9-12 and Table 9-13 list the functions of the motor control modules.

Table 9-11 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/SC.

Table 9-12 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.
Maximum torque per current control	Controls the d-axis current so that the maximum torque is output according to the load conditions.

Table 9-13 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.
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.

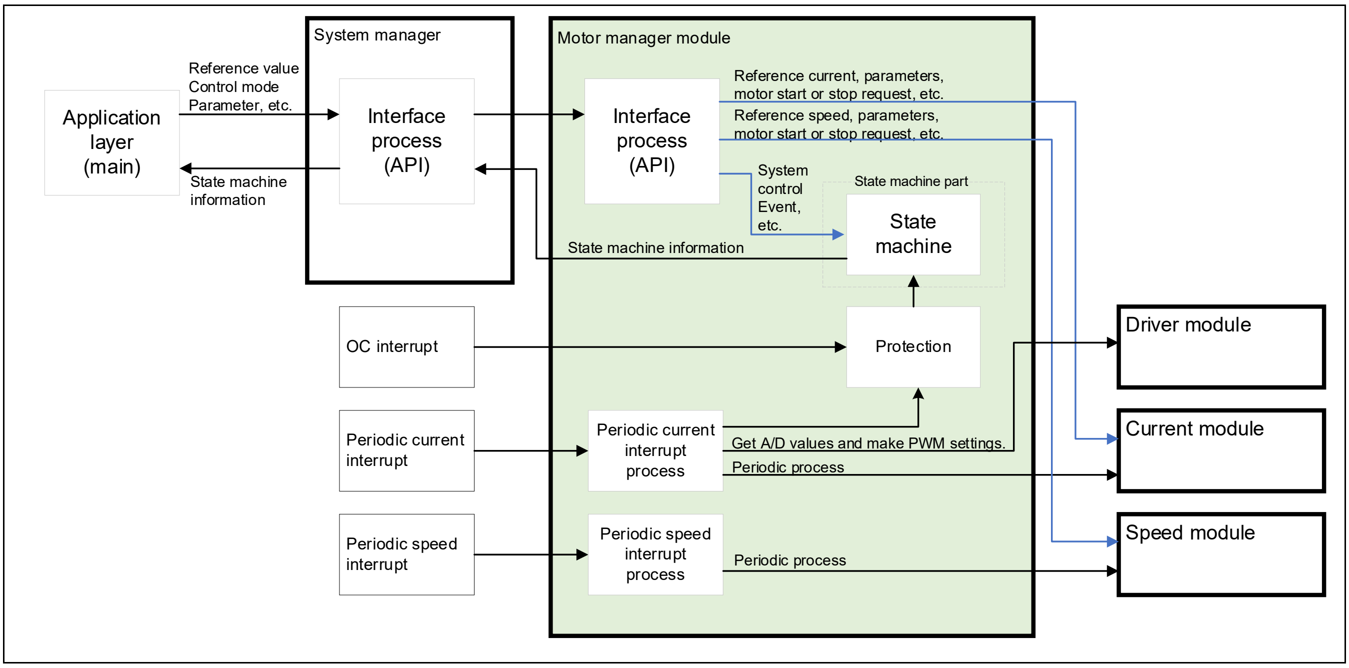


Figure 9-3 Module Configuration of the Motor Manager

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.

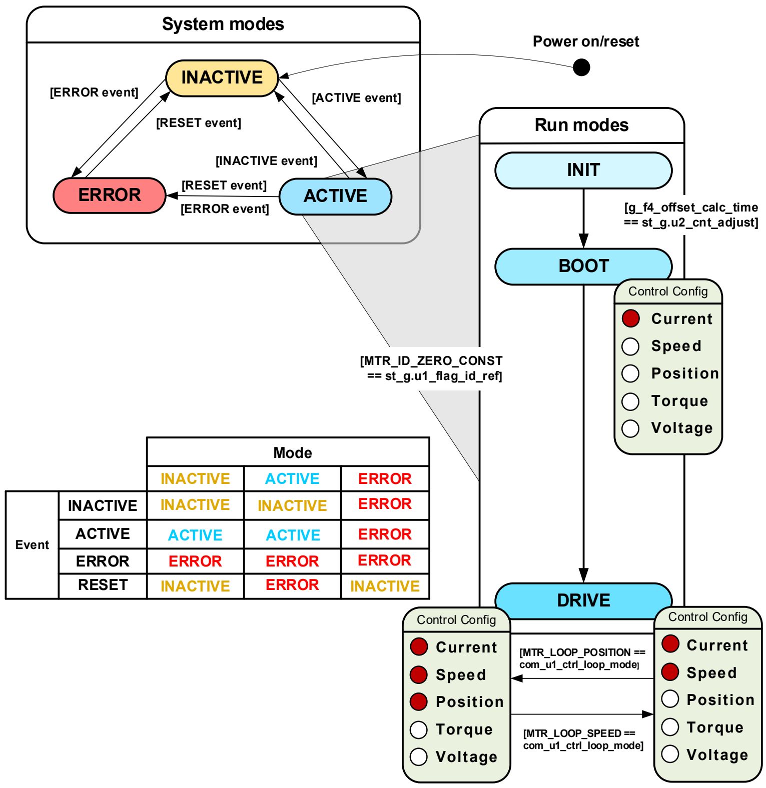


Figure 9-4 State Transition Diagram of the Motor Control Software

(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 9-14 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.

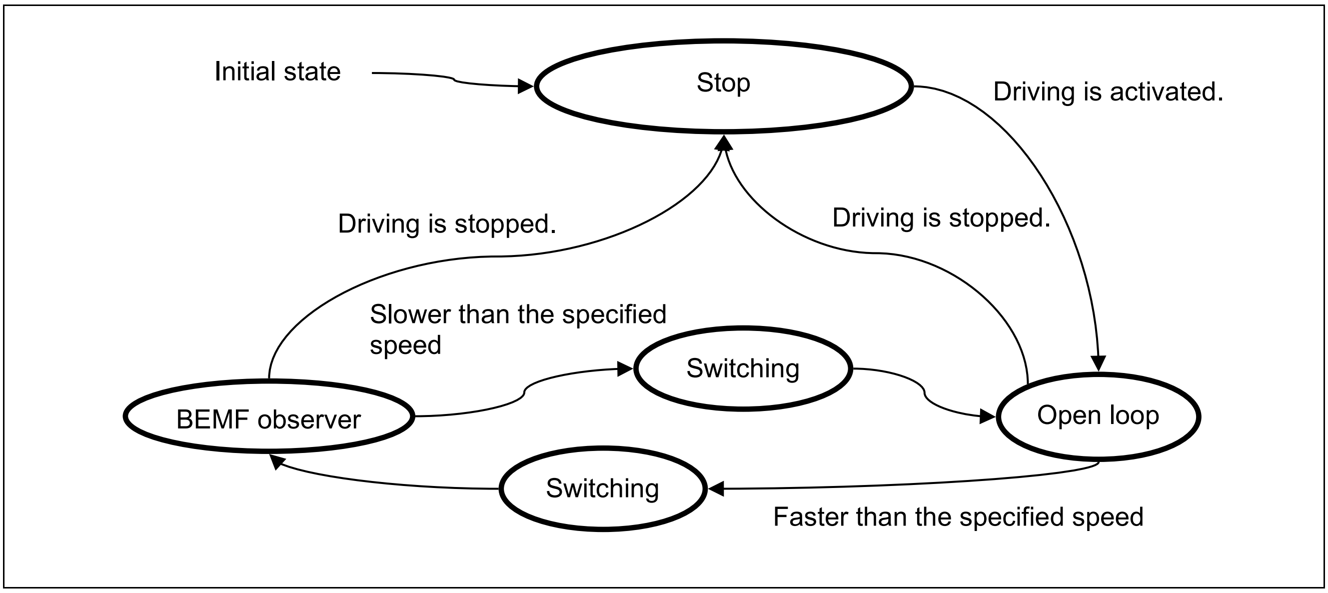


Figure 9-5 State Transition Diagram of the Operation Sequence

Table 9-15 Operation Sequence States and their Descriptions

State	Description
Initial state	This is the state before the CPU is initialized.
Stop	This is the state in which the power supply of the CPU board is turned on and the CPU board is activated. The motor is stopped.
Open loop	This is the state in which the motor is running in the range from 0 rpm (a current is flowing in the motor but the motor is stopped) to approximately 600 rpm (this is adjustable). When running within this range, the motor is controlled by using the open-loop control algorithm.
Switching	This is the state in which the control algorithm used is switched from open-loop control used in the low-speed operation to the sensorless algorithm used in the medium-to-high-speed operation. During acceleration, data are 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, data are transferred to the algorithm for open-loop control. As soon as the data transfer is completed, the sequence automatically switches to the low-speed operation state.
BEMF observer	This is the state in which the motor is running within the specified range of speed for operation under sensorless vector control up to the motor's rated speed. The motor is controlled by sensorless vector control with the use of 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.

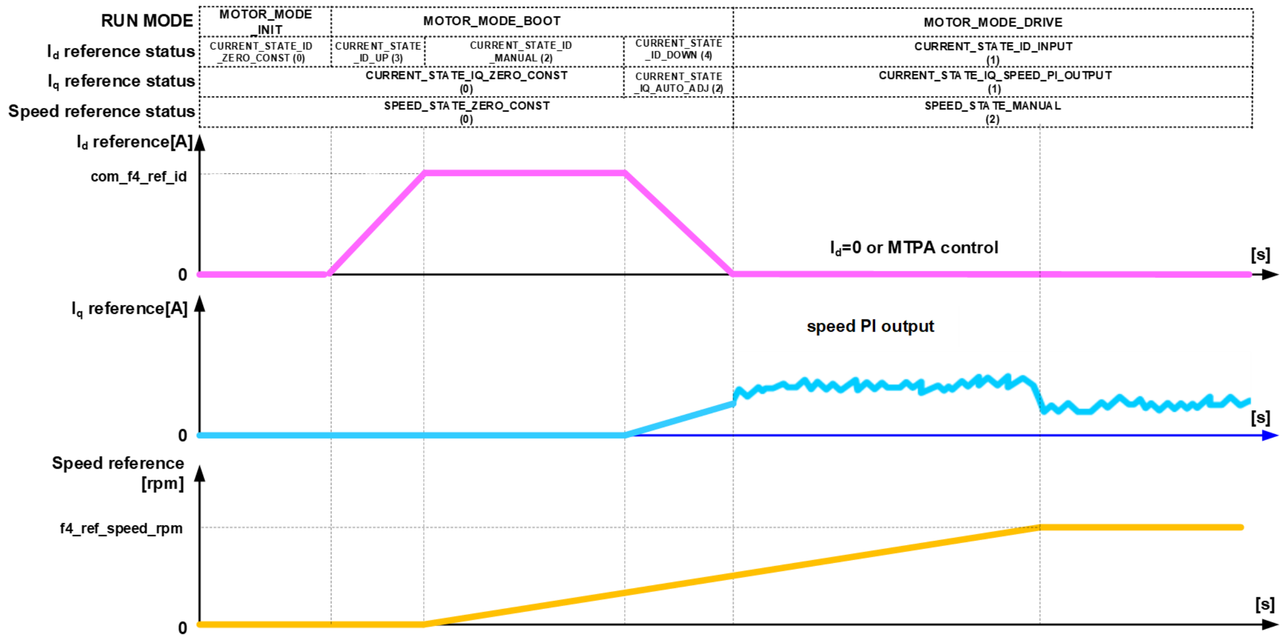


Figure 9-6 Behavior in the Startup Sequence

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 9-16.

- 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.

Table 9-16 Operating Conditions and Settings for the System Protection Functions

Overcurrent error	Overcurrent limit value (A)	4.72
	Monitoring interval (μs)	Current control interval*
Overvoltage error	Overvoltage limit value (V)	60
	Monitoring interval (μs)	Current control interval*
Low-voltage error	Low-voltage limit value (V)	8
	Monitoring interval (μs)	Current control interval*
Rotation speed error	Speed limit value (rpm)	4500
	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*

Note * See Table 9-1.

9.7.7 API

Table 9-17 lists the API functions of the motor manager module.

Table 9-17 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. (3shunt current detection)
R_MOTOR_SENSORLESS_VECTOR_OverCurrentInterrupt	Performs interrupt processing when an overcurrent is detected.

Table 9-18 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_HuntingSupress	To reduce vibration during sensorless switching, set the initial value of the PLL integral term for position and speed estimation.
R_MOTOR_CURRENT_PLLSpeedSet	Set the initial value of the PLL integral term for position and speed estimation.
R_MOTOR_CURRENT_RotorAngleSet	Set the rotor angle for structure of rotor information.
R_MOTOR_CURRENT_RefstateSet	Set the status of current control.
R_MOTOR_CURRENT_BEMFObserverParameterSet	Set the q-axis voltage disturbance of the induced voltage observer.
R_MOTOR_CURRENT_ChargeBootstrap	Perform the process for charging the bootstrap circuit.

Table 9-19 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_Opl2lessReferenceIqCalc	Calculate the q-axis current command value during sensorless control switching.

R_MOTOR_SPEED_Opl2lessPreprocess	Perform the preprocessing during sensorless control switching.
R_MOTOR_SPEED_OplDampCtrl	Perform open-loop damping control.
R_MOTOR_SPEED_OplDampReset	Reset the variable information used in open-loop damping control.
R_MOTOR_SPEED_HuntingSuppress	Set the initial value of the integral term for speed PI control.
R_MOTOR_SPEED_SwitchingFlagSet	Set the sensorless control switching flag.
R_MOTOR_SPEED_ControlParamSet	Set the speed value used in speed control.
R_MOTOR_SPEED_RefstateSet	Set the current control status used in speed control.

9.7.8 Structure and Variable Information

Table 9-20 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 9-21 List of Structures and Variables for the Current Control Module lists the structures and their member variables that are used in the current control module. Table 9-22 List of Structures and Variables for the Speed Control Module 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 9-20 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_flag_flying_start_use	Enables or disables flying start
	u1_flag_less_switch_use	Enables or disables the sensorless control soft switching function
	u1_flag_openloop_damping_use	Enables or disables the damping compensation function
	u1_flag_down_to_ol	Switching flag for open-loop control
	u1_flag_stop_transition	Flag during transition to STOP state
	u1_state_id_ref	D-axis current control status
	u1_state_iq_ref	Q-axis current control status
	u1_state_speed_ref	State of the speed command value
	u1_direction	Rotation direction
	u1_ctrl_loop_mode	Control mode selection (speed or position)
	u2_error_status	Error state
	u2_run_mode	Run mode
	f4_vdc_ad	Bus voltage (V)
	f4_iu_ad	U-phase current (A)
	f4_iv_ad	V-phase current (A)
	f4_iw_ad	W-phase current (A)
	f4_ibus_a_ad	Inverter bus current at point A (1 shunt only)
	f4_ibus_b_ad	Inverter bus current at point B (1 shunt only)
	f4_overcurrent_limit	Overcurrent limit value (A)
	f4_overvoltage_limit	Overvoltage limit value (V)
	f4_undervoltage_limit	Low-voltage limit value (V)
	f4_overspeed_limit_rad	Overspeed limit value (rad/s)
	f4_phase_err_rad_lpf	LPF output for angle error (rad)
	f4_switch_phase_err_rad	Switching angle error range (rad)
	f4_id_down_speed_rad	Speed for switching the motor control method (accelerating) (rad/s)
	f4_id_up_speed_rad	Speed for switching the motor control method (decelerating) (rad/s)

Structure	Variable	Description
	f4_id_damp_comp_speed	Open-loop damping compensation speed (rad/s)
	f4_ol_speed_rad	Speed during open-loop control (rad/s)
	st_phase_err_lpf	Structure for Angle-error LPF
	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
	st_flystart	Structure for flying start
	*p_st_driver	Structure for the driver module
	*p_st_cc	Structure for the current control module
	*p_st_sc	Structure for the speed control module
st_sensorless_vector_cfg_t Structure for setting the motor manager module control parameters	u1_flag_flying_start_use	Enables or disables flying start
	u1_flag_less_switch_use	Enables or disables the sensorless control soft switching function
	u1_flag_openloop_damping_use	Enables or disables the damping control function
	u2_off_time_cnt	Flying start: Switched-off time (cnt)
	f4_overspeed_limit_rpm	Speed limit value (rpm) (mechanical angle)
	f4_switch_phase_err_deg	Sensorless switching angle error (degrees)
	f4_phase_err_lpf_cut_freq	Angle-error LPF frequency (Hz)
	f4_id_down_speed_rpm	Speed for switching the motor control method (accelerating) (rpm)
	f4_id_up_speed_rpm	Speed for switching the motor control method (decelerating) (rpm)
	f4_ctrl_period	Current control interval (period) (s)
	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	Flying start: Active brake time (s)
	f4_on_current_th	Flying start: Current threshold for switching on (A)
	st_motor	Structure for motor parameters

Table 9-21 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_stall_detection_use	Enables or disables step-skipping (stall) detection
	u1_flag_trq_vibration_comp_use	Enables or disables torque vibration suppression
	u1_flag_trq_vibration_comp_mode	Torque vibration suppression: Compensation signal generation method
	u1_state_id_ref	D-axis current control status
	u1_state_iq_ref	Q-axis current control status
	u1_flag_offset_calc	Flag for current offset calculation
	u1_flag_charge_bootstrap	Charging completion flag for the bootstrap circuit
	u2_offset_calc_time	Measurement time setting in current offset adjustment
	u2_crnt_offset_cnt	Measurement count in current offset adjustment
	u2_charge_bootstrap_time	Charging time for the bootstrap circuit (cnt)
	u2_charge_bootstrap_cnt	Charging count for the bootstrap circuit (cnt)
	f4_ctrl_period	Current control interval (period) (s)
	f4_refu	U-phase command voltage (V)
	f4_refv	V-phase command voltage (V)
	f4_refw	W-phase command voltage (V)
	f4_vd_ref	d-axis voltage command value (V)
	f4_vq_ref	q-axis voltage command value (V)
	f4_id_ref	d-axis current command value (A)
	f4_iq_ref	q-axis current command value (A)
	f4_id_ad	d-axis current value (A)
	f4_iq_ad	q-axis current value (A)
	f4_lim_iq	q-axis current limit value (A)
	f4_offset_iu	U-phase offset current value (A)
	f4_offset_iw	W-phase offset current value (A)
	f4_sum_iu_ad	U-phase total current value (A)
	f4_sum_iw_ad	W-phase total current value (A)
	f4_vdc_ad	Bus voltage value (V)
	f4_iu_ad	U-phase current value (A)
	f4_iv_ad	V-phase current value (A)

Structure	Variable	Description
	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_id_up_step	Increase in d-axis current command value (per current control period) (A)
	f4_id_down_step	Decrease in d-axis current command value (per current control period) (A)
	f4_iq_down_step	Decrease in q-axis current command value (per current control period) (A)
	f4_iq_down_step_inv	Reciprocal of the decrease time of q-axis current command value (cnt)
	f4_ol_ref_id	Open loop current (A)
	f4_va_max	Maximum voltage on the d and q axes (V)
	f4_ed	Estimated d-axis inductive voltage value
	f4_eq	Estimated q-axis inductive voltage value
	f4_phase_err_rad	Angle error (rad)
	f4_ol_speed_rad	Speed during open-loop control (rad/s)
	f4_ref_speed_rad_ctrl	Speed command value (after LPF) (rad/s)
	st_mod	Structure for the modulation module
	st_volt_comp	Structure for the voltage error compensation module
	st_bemf_observer	Structure for the BEMF observer
	st_pll_est	Structure for position and speed estimation (BEMF observer)
	st_pi_id	Structure for d-axis PI control
	st_pi_iq	Structure for q-axis PI control
	st_rotor_angle	Structure for rotor information
	st_rotor_angle_phasecomp	Structure for rotor information (lead compensation)
	st_motor	Structure for motor parameters
	st_stalldet	Structure for stall detection
	st_trqvib_comp	Structure for torque vibration suppression control
st_current_cfg_t	u1_flag_volt_err_comp_use	Enables or disables voltage error compensation
Structure for setting the control parameters for	u1_flag_stall_detection_use	Enables or disables step-skipping (stall) detection

Structure	Variable	Description
the current control module	u1_flag_trq_vibration_comp_use	Enables or disables torque vibration suppression
	u1_flag_trq_vibration_comp_mode	Torque vibration suppression: Compensation signal generation method
	u1_target_2f	Torque vibration suppression: Enables or disables the suppression of secondary components
	u2_offset_calc_time	Offset calculation time setting
	u2_charge_bootstrap_time	Charging time for the bootstrap circuit (cnt)
	f4_ctrl_period	Current control period (s)
	f4_current_omega_hz	Natural frequency for the current control system (Hz)
	f4_current_zeta	Attenuation coefficient for the current control system
	f4_id_up_step	Increase in d-axis current command value (per current control period) (A)
	f4_id_down_step	Decrease in d-axis current command value (per current control period) (A)
	f4_iq_down_step_time_inv	Decrease in q-axis current command value (per current control period) (A)
	f4_ol_ref_id	Open loop current (A)
	f4_id_hpf_time	Step-skipping (stall) detection: Time constant of d-axis current HPF
	f4_iq_hpf_time	Step-skipping (stall) detection: Time constant of q-axis current HPF
	f4_threshold_level	Step-skipping (stall) detection: Detection level (A)
	f4_threshold_time	Step-skipping (stall) detection: Detection time (s)
	f4_timelead_1f	Torque vibration suppression: Phase adjustment value for fundamental components (rad)
	f4_timelead_2f	Torque vibration suppression: Phase adjustment value for secondary components (rad)
	f4_tf_lpf_omega	Torque vibration suppression: LPF cutoff frequency for the TF (Hz)
	f4_output_gain_1f	Torque vibration suppression: TF output gain for fundamental components
	f4_output_gain_2f	Torque vibration suppression: TF output gain for fundamental components
	f4_input_weight2	Torque vibration suppression: Weight 2
	f4_input_weight1	Torque vibration suppression: Weight 1
	f4_input_weight0	Torque vibration suppression: Weight 0
	f4_suppression_th_1f	Torque vibration suppression: Suppression Target for fundamental components
	f4_suppression_th_2f	Torque vibration suppression: Suppression Target for secondary components
	f4_abnormal_output_th_1f	Torque vibration suppression: TF output abnormality for fundamental components

Structure	Variable	Description
	f4_abnormal_output_th_2f	Torque vibration suppression: TF output abnormality for secondary components
	st_motor	Structure for motor parameters
st_current_output_t Structure for the current control module output	u1_flag_offset_calc	Current offset flag
	u1_flag_charge_bootstrap	Charging completion flag for the bootstrap circuit
	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
	f4_phase_err_rad	Angel error [rad]
st_current_input_t Structure for the current control module input	u1_state_id_ref	D-axis current control status
	u1_state_iq_ref	Q-axis current control status
	f4_iu_ad	U-phase current value (A)
	f4_iv_ad	V-phase current value (A)
	f4_iw_ad	W-phase current value (A)
	f4_vdc_ad	Bus voltage value (V)
	f4_id_ref	d-axis current command value (A)
	f4_iq_ref	q-axis current command value (A)
	f4_ol_speed_rad	Speed during open-loop control (rad/s)
	f4_ref_speed_rad_ctrl	Speed command value (after LPF) (rad/s)
st_bemf_observer_cfg_t Structure for the BEMF observer module input	f4_e_obs_omega_hz	Natural frequency for the inductive voltage estimation system (Hz)
	f4_e_obs_zeta	Attenuation coefficient for the inductive voltage estimation system
	f4_pll_est_omega_hz	Natural frequency for the position estimation system (Hz)
	f4_pll_est_zeta	Attenuation coefficient for the position estimation system

Table 9-22 List of Structures and Variables for the Speed Control Module

Structure	Variable	Description
st_speed_control_t Structure for the speed control module	u1_state_id_ref	D-axis current control status
	u1_state_iq_ref	Q-axis current control status
	u1_active	Selection of whether to enable the 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.
	u1_flag_fluxwkn_use	Enables or disables flux weakening control
	u1_flag_switching	Switching flag for sensorless 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 (after LPF) (rad/s)
	f4_ref_speed_rad	Speed command value (rad/s)
	f4_ref_speed_rad_manual	Speed command value set by the user during speed control (rad/s)
	f4_speed_rad_ctrl	Speed (after LPF) (rad/s)
	f4_speed_rad	Input speed to the speed module (rad/s)
	f4_max_speed_rad	Maximum speed (rad/s)
	f4_speed_rate_limit_rad	Speed variation limit value (rad/s)
	f4_id_ref_output	d-axis current command value (A)
	f4_iq_ref_output	q-axis current command value (A)
	f4_va_max	Maximum voltage on the d and q axes (V)
	f4_id_ad	d-axis current value (A)
	f4_iq_ad	q-axis current value (A)
	f4_opl2less_sw_time	Sensorless soft switching time (s)
	f4_torque_current	Torque current (A) (used in soft switching calculation for sensorless control)
	st_motor	Structure for motor constants
	st_pi_speed	Structure for PI control
	st_fluxwkn	Structure for flux weakening control
	st_opl_damp	Structure for step-skipping (stall) detection
	st_1st_order_lpf_t	Structure for LPF
	st_mtpa	Structure for maximum torque per current control
st_speed_cfg_t	u1_flag_fluxwkn_use	Enables or disables flux weakening control
	u1_flag_mtpa_use	Flag for indicating whether to use maximum torque per current control

Structure	Variable	Description
Structure for setting the control parameters for the speed control module	f4_max_speed_rpm	Maximum speed (rpm) (mechanical angle)
	f4_speed_ctrl_period	Speed control interval (s)
	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)
	f4_opl2less_sw_time	Sensorless switching time (s)
	f4_ed_hpf_omega	Damping control: Natural frequency for HPF (Hz)
	f4_ol_damping_zeta	Damping control: Attenuation coefficient
	f4_ol_damping_fb_limit_rate	Damping control: Feedback limit rate
	f4_ol_ref_id	Open-loop control: Id current command value
	f4_id_down_speed_rpm	Speed for switching the motor control method (accelerating) (rpm)
	st_motor	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)
	u1_state_id_ref	D-axis current control status
	u1_state_iq_ref	Q-axis current control status
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 9-23 lists the macros for the motor manager module.

Table 9-23 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_ERROR_HALL_TIMEOUT	0x0008	Timeout error of hall signal detection (Does not happen in sensorless vector)
	MOTOR_ERROR_BEMF_TIMEOUT	0x0010	Timeout error of BEMF signal detection (Does not happen in sensorless vector)
	MOTOR_ERROR_HALL_PATTERN	0x0020	Hall signal pattern error (Does not happen in sensorless vector)
	MOTOR_ERROR_BEMF_PATTERN	0x0040	BEMF signal pattern error (Does not happen in sensorless vector)
	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_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
	MOTOR_MODE_ANALYSIS	0x03	Analysis mode (Not used)
	MOTOR_MODE_TUNE	0x04	Tuning mode (Not used)
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

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.

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 or SC, the parameters related to the changed settings must also be modified.

Table 10-2 List of MCU-Related Parameters

File Name	Macro Name	RA Family Setting	RX26T Setting	Description
r_motor_module_cfg.h	MOTOR_MCU_CFG_PWM_TIMER_FREQ	RA6T2, RA8T1 : 120.0 RA8T2 : 250	120.0	PWM timer frequency (MHz)
	MOTOR_MCU_CFG_CARRIER_FREQ	20.0	20.0	Carrier wave frequency (kHz)
	MOTOR_MCU_CFG_INTR_DECIMATION	0	0	Value to count for the skipping of carrier wave interrupts
	MOTOR_MCU_CFG_AD_FREQ	RA6T2, RA8T1 : 60.0 RA8T2 : 41.447	60.0	ADC operating frequency (MHz)
	MOTOR_MCU_CFG_AD_SAMPLING_CYCLE	(6 + 39)	(6 + 39)	ADC sampling interval (cycles)
	MOTOR_MCU_CFG_AD12BIT_DATA	4095.0	4095.0	ADC resolution
	MOTOR_MCU_CFG_ADC_OFFSET	0x7FF	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_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.
	CURRENT_CFG_CHARGE_BOOTSTRAP_TIME	144	Sets the charging time for the bootstrap circuit [cnt].#

Note # : Depending on the circuit configuration of the user's own inverter board (such as complementary type where the upper and lower arms are a combination of n-channel and p-channel), a bootstrap circuit may not be necessary. In such cases, please set the macro value to 0.

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	21.4	Overcurrent limit value for the inverter board (A)
	INVERTER_CFG_OVERVOLTAGE_LIMIT	60.0	Overvoltage limit (V)
	INVERTER_CFG_UNDERVOLTAGE_LIMIT	8.0	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/SC 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 20.0 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 11.5 or 12.6 • 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

<table>
<tr><th>File Name</th><th>Macro Name</th><th>RA MCU setting</th><th>RX26T Setting</th><th>Description</th></tr>
<tr><td rowspan="16">r_motor_inverter_cfg.h</td><td>INVERTER_CFG_SHUNT_RESIST</td><td>0.01</td><td>0.01</td><td>Shunt resistance value (ohms)</td></tr>
<tr><td>INVERTER_CFG_DEADTIME</td><td>1.0</td><td>1.0</td><td>Dead time (μs)</td></tr>
<tr><td>INVERTER_CFG_VOLTAGE_GAIN</td><td>22.2766</td><td>22.2766</td><td>Coefficient for voltage detection</td></tr>
<tr><td>INVERTER_CFG_CURRENT_AMP_GAIN</td><td>20.0</td><td>20.0</td><td>Gain of the amplifier for current detection</td></tr>
<tr><td>INVERTER_CFG_INPUT_V</td><td>24.0</td><td>24.0</td><td>Input voltage (V)</td></tr>
<tr><td>INVERTER_CFG_ADC_REF_VOLTAGE</td><td>3.3</td><td>5.0</td><td>Analog power-supply voltage for the MCU (V)</td></tr>
<tr><td>INVERTER_CFG_COMP_V0</td><td colspan="2">0.096</td><td>Coefficient for compensation of the voltage error (V)</td></tr>
<tr><td>INVERTER_CFG_COMP_V1</td><td colspan="2">0.192</td><td>Coefficient for compensation of the voltage error (V)</td></tr>
<tr><td>INVERTER_CFG_COMP_V2</td><td colspan="2">0.288</td><td>Coefficient for compensation of the voltage error (V)</td></tr>
<tr><td>INVERTER_CFG_COMP_V3</td><td colspan="2">0.384</td><td>Coefficient for compensation of the voltage error (V)</td></tr>
<tr><td>INVERTER_CFG_COMP_V4</td><td colspan="2">0.480</td><td>Coefficient for compensation of the voltage error (V)</td></tr>
<tr><td>INVERTER_CFG_COMP_I0</td><td colspan="2">0.07</td><td>Coefficient for compensation of the voltage error (A)</td></tr>
<tr><td>INVERTER_CFG_COMP_I1</td><td colspan="2">0.14</td><td>Coefficient for compensation of the voltage error (A)</td></tr>
<tr><td>INVERTER_CFG_COMP_I2</td><td colspan="2">0.22</td><td>Coefficient for compensation of the voltage error (A)</td></tr>
<tr><td>INVERTER_CFG_COMP_I3</td><td colspan="2">0.30</td><td>Coefficient for compensation of the voltage error (A)</td></tr>
<tr><td>INVERTER_CFG_COMP_I4</td><td colspan="2">0.50</td><td>Coefficient for compensation of the voltage error (A)</td></tr>
</table>

INVERTER_CFG_DEADTIME

Specify the dead time in μs (microseconds) that is described in the inverter specifications and design document. For the MCI-LV-1 inverter, 1.0 μs is specified.

INVERTER_CFG_INPUT_V

The default voltage value is 24 VDC, which is supplied by a regulated DC Voltage Source.

INVERTER_CFG_ADC_REF_VOLTAGE

Specify the analog voltage of the MCU. 3.3 V is specified for the RA Family MCU CPU board and 5.0 V is specified for the RX26T CPU board.

INVERTER_CFG_COMP_Vx, INVERTER_CFG_COMP_Ix

See section 10.7.4.

10.7.2 Current Detection Gain

In the MCI-LV-1 inverter, the voltage input to the ADC is specified by the amount of the current as shown in Table 10-12 for RA Family MCU or Table 10-13 for RX26T

To set the current detection in this sample program, INVERTER_CFG_CURRENT_AMP_GAIN and INVERTER_CFG_SHUNT_RESIST are used.

INVERTER_CFG_CURRENT_AMP_GAIN

Set the current amplifier gain which determines how much current (A) is equivalent per 1 V voltage input to the ADC. The current amplifier gain depends on the current detection circuit and by default the gain value is 20 according to MCI-LV-1 User's Manual (R12UZ0092).

INVERTER_CFG_SHUNT_RESIST

Set the resistance used for the shunt resistor. The shunt resistor used in MCI-LV-1 inverter current detection circuit is 0.01ohm. Refer to MCI-LV-1 User's Manual (R12UZ0092) for the details of current detection circuit.

When a Hall current transformer is used instead of a shunt resistor, specify 1.0.

INVERTER_CFG_ADC_REF_VOLTAGE

For RA Family MCU, set to 3.3 because the ADC reference voltage is 3.3 V in the RA Family MCU.

For RX26T MCU, set to 5.0 because the ADC reference voltage is 5.0V in the RX MCU.

The current equivalent per volt input to the ADC is governed by the equation shown below.

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

As such, in this sample program using MCI-LV-1 inverter, the current per volt [A/V] is 5A per 1V.

This results in the current detection range of 5A/V × 3.3V = 16.5 A (peak-to-peak) or ±8.25 A for RA Family MCU. For RX26T MCU, the current detection range is 5A/V × 5.0V = 25.0 A (peak-to-peak) or ±12.5 A.

The following descriptions assume that the RA6T2 is in use whereby the reference voltage is 3.3 V.

Table 10-12 Current Signal Specifications for the MCI-LV-1 for the RA6T2 MCU Using a 3.3-V Reference Voltage

3-Phase Output Current	ADC Input Voltage	A/D-Converted Value
+8.25 A	3.3 V	4095
0 A	1.65 V	2047
-8.25 A	0.0 V	0

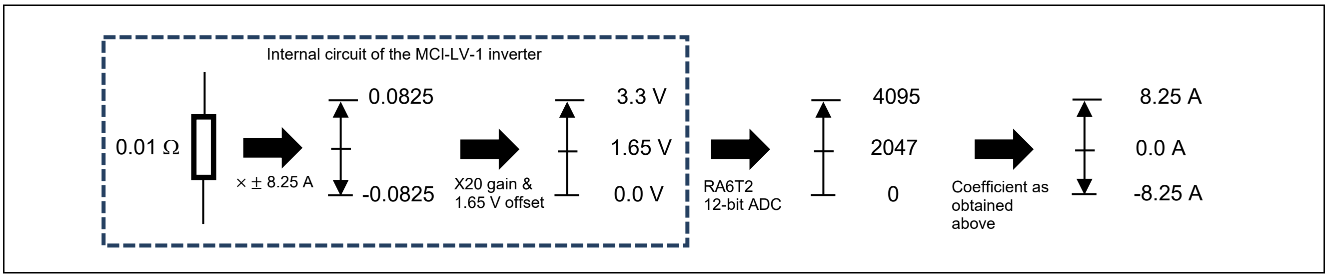


Figure 10-1 Flow of Calculation for Current Detection for the RA6T2 MCU Using a 3.3-V Reference Voltage

The following descriptions assume that the RX26T is in use whereby the reference voltage is 5.0 V

Table 10-13 Current Signal Specifications for the MCI-LV-1 for the RX26T MCU Using a 5.0-V Reference Voltage

3-Phase Output Current	ADC Input Voltage	A/D-Converted Value
+12.5 A	5.0 V	4095
0 A	2.5 V	2047
-12.5 A	0.0 V	0

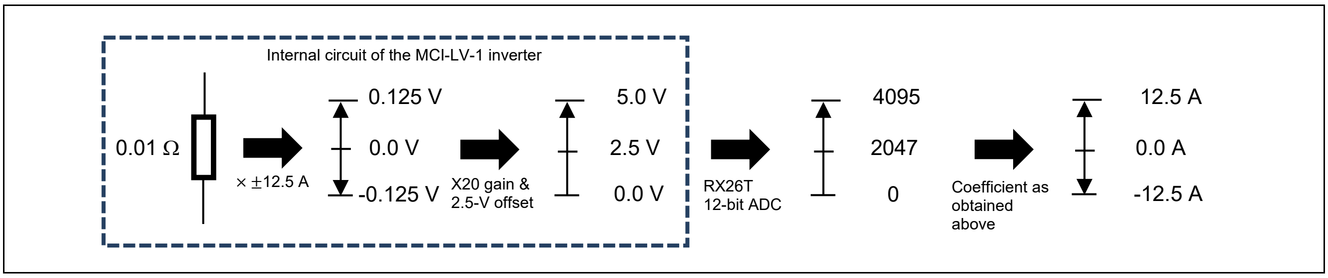


Figure 10-2 Flow of Calculation for Current Detection for the RX26T MCU Using a 5.0-V Reference Voltage

10.7.3 Voltage Detection Gain

The voltage detection gain is set by INVERTER_CFG_VOLTAGE_GAIN.

Like section 10.7.2, the following descriptions assume that RA Family MCU is in use, thus the reference ADC voltage is 3.3 V. When the RX26T is in use, assume the ADC reference voltage as 5.0 V.

Set the coefficient for use in calculating the inverter bus voltage (V) per volt input to the ADC.

For MCI-LV-1 used with RA Family MCU, when ADC input is 3.3 V, detected voltage corresponds to 73.51 V (this ratio depends on the inverter board circuit with DC input source of 24 V). This is the reference inverter bus voltage. In the case of RX26T, when ADC input is 5.0 V, the reference inverter bus voltage is 111.383 V.

The INVERTER_CFG_VOLTAGE_GAIN is obtained from the equation below.

RA Family MCU : $\mathrm{INVERTER\_CFG\_VOLTAGE\_GAIN} = \frac{Reference\ inverter\ bus\ voltage}{Reference\ ADC\ input\ voltage} = \frac{73.51}{3.3} = 22.276$

RX26T MCU : $\mathrm{INVERTER\_CFG\_VOLTAGE\_GAIN} = \frac{Reference\ inverter\ bus\ voltage}{Reference\ ADC\ input\ voltage} = \frac{111.383}{5.0} = 22.276$

Therefore, set INVERTER_CFG_VOLTAGE_GAIN to 22.276.

Table 10-14 Specifications of Inverter Bus Voltage Signal for the MCI-LV-1

Inverter Bus Voltage		**ADC Input Voltage**		**A/D-Converted Value**
0.0 V		0.0 V		0
RA Family MCU	73.51 V	RA Family MCU	3.3 V	4095
RX26T MCU	111.383 V	RX26T MCU	5.0 V	

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 and the design of the gate driving circuits determine the dead time. When Si-MOSFET is used, a value roughly in or around the range from 0.5 to 1 μs is selected. Reflect the selected dead-time value in the dedicated input location provided for motor settings in the FSP/SC. In this sample software, it is set to 1.0μs according to the characteristics of the inverter board (RTK0EM0000B12020BJ) included in the kit.

(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 function, set this flag to MTR_FLG_CLR.

(3) Setting the voltage compensation table

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

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

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

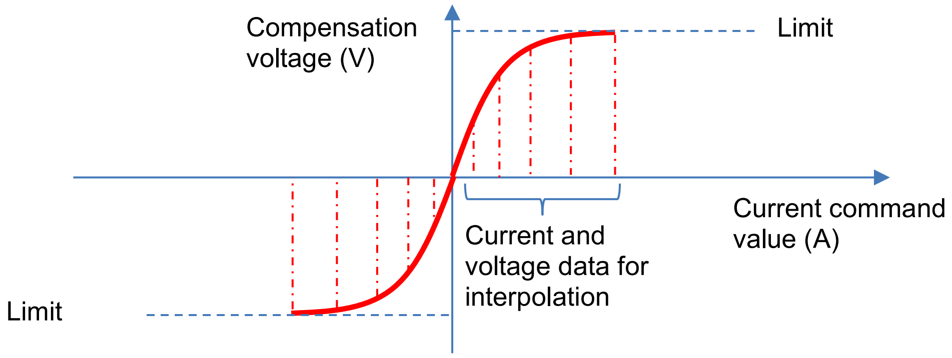


Figure 10-3 Relationship between the Compensation Voltage, Limit, and Current Command Values

Table 10-15 Current and Voltage Data for Interpolation for Different Carrier Cycles

Carrier Frequency		**20 kHz**
	Iu	ΔVu
0	0.00	0.000
1	0.07	0.096
2	0.14	0.192
3	0.22	0.288
4	0.30	0.384
5	0.50	0.480

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 10-16 lists the settings in the sample program.

Table 10-16 Motor Parameter Settings

File Name	Macro Name	Setting	Description
r_motor_targetmotor_cfg.h	MOTOR_CFG_POLE_PAIRS	4	Number of pole pairs
	MOTOR_CFG_MAGNETIC_FLUX	0.01119f	Magnetic flux (wb)
	MOTOR_CFG_RESISTANCE	1.3f	Resistance (ohms)
	MOTOR_CFG_D_INDUCTANCE	0.0013f	d-axis inductance (H)
	MOTOR_CFG_Q_INDUCTANCE	0.0013f	q-axis inductance (H)
	MOTOR_CFG_ROTOR_INERTIA	0.000003666f	Rotor inertia (kgm^2)
	MOTOR_CFG_NOMINAL_CURRENT_RMS	1.67f	Rated current (A)
	MOTOR_CFG_MAX_SPEED_RPM	2400.0f	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.

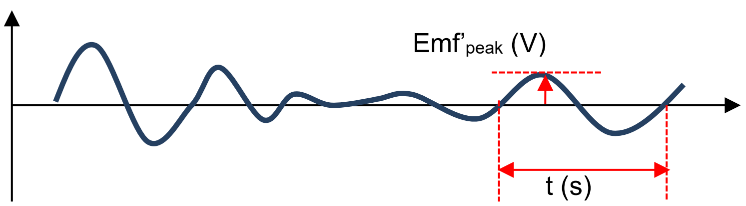
MOTOR_CFG_NOMINAL_CURRENT_RMS

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 10-17 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 10-17 are set and reflected in the internal variables at startup but if adjustments are needed after startup, the parameters listed in Table 6-4 can be changed from the RMW. Note that not all of the current control parameters can be changed.

Table 10-17 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	300.0f	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 10-18 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 10-19 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 10-19 are set and reflected in the internal variables at startup but if adjustments are needed after startup, the parameters listed in Table 6-4 can be changed from the RMW. Note that not all of the speed control parameters can be changed.

Table 10-19 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	5.0f	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	4500.0f	Speed limit value (rpm) (mechanical angle)
	SPEED_CFG_RATE_LIMIT_RPM	1000.0f	Acceleration limit (rpm/s)

SPEED_CFG_CTRL_PERIOD

Set to 0.0005 s (0.5 ms). If this setting needs to be changed, the AGT0 timer setting of chapter 11.6 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 an interfering oscillation is at a frequency above the natural frequency, increasing the value of the natural frequency to match that of the interference 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 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 any number of samples.

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.

10.13 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) Open-loop control parameters

Table 10-21 lists the parameters to be used for open-loop control.

Table 10-21 Setting Parameters for Open-Loop Control

File Name	Macro Name	Setting	Unit	Description
r_motor_module_cfg.h	CURRENT_CFG_REF_ID_OPENLOOP	0.5f	A	
	CURRENT_CFG_ID_UP_STEP_TIME	2560.0f	Cycles	
	CURRENT_CFG_ID_DOWN_STEP_TIME	500.0f	Cycles	
	SPEED_OPL2LESS_SWITCH_TIME	0.025f	S	
	SPEED_OPL_DAMP_ED_HPF_OMEGA	2.5f	Hz	
	SPEED_OPL_DAMP_ZETA	1.0f	—	
	SPEED_OPL_DAMP_FB_SPEED_LIMIT_RATE	0.5f	—	
	SENSORLESS_VECTOR_ID_DOWN_SPEED_RPM	600.0f	rpm	
	SENSORLESS_VECTOR_ID_UP_SPEED_RPM	400.0f	rpm	
	SENSORLESS_VECTOR_OPL2LESS_SWITCH_PHASE_ERR_DEG	10.0f	Degrees	
	SENSORLESS_VECTOR_OPL2LESS_SWITCH_PHASE_ERR_LPF_CUT_FREQ	10.0f	Hz	

CURRENT_CFG_REF_ID_OPENLOOP

Specify the current that is to flow through the d axis when open-loop control is started. This value should be no greater than the rated current. If the motor has a large moment of inertia, specifying a small value may lead to the motor being unable to start.

To adjust this value during operation, use the com variable com_f4_ol_ref_id.

CURRENT_CFG_ID_UP_STEP_TIME

Specify the time required to increase the current that is to flow through the d axis when open-loop control is started. The unit of this value is the cycle of current control (the value 1 specifies the period of one current control cycle).

CURRENT_CFG_ID_DOWN_STEP_TIME

Specify the time required to decrease the current that is to flow through the d axis in order to switch the operation to sensorless vector control after open-loop control has started. The unit of this value is the cycle of current control (the value 1 specifies the period of one current control cycle).

SPEED_OPL2LESS_SWITCH_TIME

Specify the time for switching the operation from open-loop control to closed-loop control (vector control by using the BEMF observer).

SPEED_OPL_DAMP_ED_HPF_OMEGA

Specify the cutoff frequency (Hz) of the HPF for estimating the voltage induced along the d axis.

SPEED_OPL_DAMP_ZETA

Specify the default attenuation coefficient for the open-loop damping control system. Specify a value from 0.8 to 1.0.

SPEED_OPL_DAMP_FB_SPEED_LIMIT_RATE

Specify the default constant for calculating the limit on the compensation value for output speed commands.

SENSORLESS_VECTOR_ID_DOWN_SPEED_RPM

Specify the mechanical angular velocity (rpm) at which the operation is to be switched from open-loop control to BEMF observer control.

SENSORLESS_VECTOR_ID_UP_SPEED_RPM

Specify the mechanical angular velocity (rpm) at which the operation is to be switched from BEMF observer control to open-loop control. Specify a velocity that is sufficiently lower than SENSORLESS_VECTOR_ID_DOWN_SPEED_RPM.

SENSORLESS_VECTOR_OPL2LESS_SWITCH_PHASE_ERR_DEG

Specify the threshold value for the angle error to be used when the operation is switched from open-loop control to BEMF observer control. The unit of this value is electrical angle (degrees).

SENSORLESS_VECTOR_OPL2LESS_SWITCH_PHASE_ERR_LPF_CUT_FREQ

Specify the cutoff frequency of the LPF for the angle error obtained by the BEMF observer to obtain smooth switching from open-loop control to sensorless vector control with the use of the BEMF observer.

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

Table 10-22 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 10-22 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	1000	Natural frequency for the BEMF observer (Hz)
	CURRENT_CFG_E_OBS_ZETA	1	Attenuation coefficient for the BEMF observer
	CURRENT_CFG_PLL_EST_OMEGA	10	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 accumulating the angle errors obtained by the BEMF observer and in calculating angles. 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 accumulating the angle errors obtained by the BEMF observer and in calculating angles. Specify 1.0 in general.

10.14 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 bus voltage and speed are monitored and control automatically begins when the necessary conditions are satisfied.

Table 10-23 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.

10.15 Flying Start Parameters

The following describes the parameters for flying start operation.

Table 10-24 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 10-25 List of Flying Start Parameters

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	SENSORLESS_VECTOR_FLY_START_CURRENT_TH	0.1f	Specify the threshold (A) for the switched-on current.
	SENSORLESS_VECTOR_FLY_START_OVER_TIME_SEC	0.0009f	
	SENSORLESS_VECTOR_FLY_START_OFF_TIME_SEC	0.0001f	
	SENSORLESS_VECTOR_FLY_START_ACTIVE_BRAKE_TIME_SEC	0.005f	
	SENSORLESS_VECTOR_FLY_START_RESTART_SPEED_LIMIT	660.0f	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 0.1 A due to conditions 1 and 2 below (0.1 A < Ia < 2.9 A). A longer Ton time increases the approximation error of equation 7.10.4, so 0.1 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-LV-1 inverter board is 16.5 A/12 bits (4.03 mA/LSB), the threshold will be 4.03 mA × 100 = 0.4 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. Note that the Ton and Toff periods are motor drive control stop periods, and particularly during the Ton period, braking is applied, so when inertia is small, the motor stops immediately. Therefore, experimentally lower the current threshold value to approximately one-quarter.
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 To(1.67 A) × √3 ≃ 2.9 A, the detected current vector (Ia) must be no greater than 2.9 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 (8.25 us < Toff < 3.0 ms), so Toff is set to 0.1 ms.

Condition 1	To satisfy the approximation condition i(0) = 0 for equation 7.10.4 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 0.1 A to 0 A, the maximum Toff time at the maximum rotation speed 2400 rpm is about 8.25 us. Therefore, specify Toff > 8.25 us.
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 0.1 A is reached, Ton = 0.126 ms can be obtained at the maximum rotation speed 2400 rpm. Here, the (Ton + Toff)max time that can be converted from the rotation speed is 3.125 ms, so Toff < (3.125 – 0.126) ms = 3.0 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.

10.16 Torque Vibration Suppression Parameters

The torque vibration suppression function can be used in the steady state, that is, while the motor is running under sensorless vector control with the use of the BEMF observer. It cannot be used while the motor is under open-loop control or during acceleration or deceleration. If the torque vibration suppression function is enabled during open-loop control, acceleration, or deceleration, note that unexpected operation may occur. As this control function is mainly for use in the low-to-medium-speed range and the conditions for use will depend on the characteristics of the source (such as a compressor) of the torque vibration, use the com_u1_flag_trq_vibration_comp_use variable and monitor the motor driving waveform through the RMW to check the operation.

Table 10-26 Procedure for Operating the Torque Vibration Suppression Function

Step	Operation	Manipulation in the RMW
1	Check that the motor is in the steady state.	—
2	Set the parameters for the torque vibration suppression function.	Modify the necessary parameters through the RMW. The following lists the major parameters for adjustment. • Phase lead K_2 (rad): com_f4_timelead_1f/2f • Natural frequency for the LPF in the TF (Hz): com_f4_tf_lpf_omega • Gain K_1 for the value input to the repetitive controller: com_f4_tf_output_gain_1f/2f • Input signal weight: com_f4_input_weight0/1/2 • Goal value for suppression: com_f4_suppression_th_1f/2f

		• Threshold for the ratio of abnormal output from the TF: com_f4_abnormal_output_th_1f/2f
3	Start the torque vibration suppression function.	Set com_u1_flag_trq_vibration_comp_use to 1. Update the variables according to the procedures for an RMW operation. * Only enable this com variable under sensorless vector control.
4	Generation of the compensating signal ends.	Check the state of torque vibration suppression through an RMW operation. State: u2_trq_comp_state * If the compensating signal diverges or a desired effect of suppression is not obtainable, proceed to step 5 and then adjust the parameter values as described in step 2.
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. Update the variables according to the procedures for an RMW operation.
6	Return to step 1 as required after the speed has been changed.	—

Table 10-27 List of Torque Vibration Suppression Parameters

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_TRQVIB_TARGET_2F	MTR_FLG_SET	0: Only the first-order component of the rotation frequency is suppressed. 1: The first-order and second-order components of the rotation frequency are suppressed.
	CURRENT_CFG_TRQVIB_COMP_MODE	TRQCOMP_MODE_PAT	Compensation signal generation method: TRQCOMP_MODE_LUT, TRQCOMP_MODE_PAT
	CURRENT_CFG_TRQVIB_OUTPUT_GAIN_1F	0.005f	K_1: Gain for the value input to the repetitive controller (for the first-order component of the rotation frequency)
	CURRENT_CFG_TRQVIB_OUTPUT_GAIN_2F	0.005f	K_1: Gain for the value input to the repetitive controller (for the second-order component of the rotation frequency)
	CURRENT_CFG_TRQVIB_TIMELEAP_1F	0.0f	K_2: Phase lead (rad) (for the first-order component of the rotation frequency)
	CURRENT_CFG_TRQVIB_TIMELEAP_2F	4.0f	K_2: Phase lead (rad) (for the second-order component of the rotation frequency)
	CURRENT_CFG_TRQVIB_TF_LPF_OMEGA	0.6f	Natural frequency for the LPF in the tracking filter (Hz)
	CURRENT_CFG_TRQVIB_INPUT_WEIGHT_2	1.0f	These values are used to specify the weights for the input signals. Specify them to suit the
	CURRENT_CFG_TRQVIB_INPUT_WEIGHT_1	0.0f	

	CURRENT_CFG_TRQVIB_INPUT_WEIGHT_0	0.0f	characteristics of the motor and load (only with the LUT method).
	CURRENT_CFG_TRQVIB_SUPP_TH_1F	0.05f	Goal value for suppression X (for the first-order component of the rotation frequency)
	CURRENT_CFG_TRQVIB_SUPP_TH_2F	0.1f	Goal value for suppression X (for the second-order component of the rotation frequency)
	CURRENT_CFG_TRQVIB_ABNORMAL_TH_1F	0.9f	Ratio of abnormal output from the TF (for the first-order component of the rotation frequency)
	CURRENT_CFG_TRQVIB_ABNORMAL_TH_2F	0.9f	Ratio of abnormal output from the TF (for the second-order component of the rotation frequency)

CURRENT_CFG_TRQVIB_TARGET_2F
This macro can be used to include the second-order component of the rotation frequency as the target of compensation in torque vibration suppression control.
0: Only the first-order component of the rotation frequency is suppressed.
1: The first-order and second-order components of the rotation frequency are suppressed.

CURRENT_CFG_TRQVIB_OUTPUT_GAIN_1F/2F
Specify the gain K_1 for the value input to the repetitive controller. Specifying a large value for K_1 shortens the time until learning of the compensating signal is completed but the compensating signal may be divergent depending on the conditions. Specifying a small value for K_1 makes the value input to the repetitive controller smaller, which increases the time until learning of the compensating signal is completed, but a stable compensating signal can be expected. In addition, the internal algorithm of torque vibration suppression includes an element of integration, so the feedback value is kept unchanged in a steady state with stable K_1 regardless of the gain although the times in transient states will change.

CURRENT_CFG_TRQVIB_TIMELEAP_1F/2F (rad)
This parameter adjusts the output phase. Specify it within the range from 0 to 2π (6.28) in radians corresponding to one cycle of the vibration component.

CURRENT_CFG_TRQVIB_TF_LPF_OMEGA (Hz)
Specify the natural frequency for the LPF in the TF. The purpose of the LPF in the TF is to only pass the DC components. Take the delay in extraction into consideration when setting this value.

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 (only with the LUT method).

CURRENT_CFG_TRQVIB_SUPP_TH_1F /2F
This macro can be used to specify the goal value for suppression X, which is to be used in judging the end of learning.

$$X = \frac{Amplitude\ of\ the\ vibration\ component\ after\ suppression}{Amplitude\ of\ the\ vibration\ component\ before\ suppression}$$

CURRENT_CFG_TRQVIB_ABNORMAL_TH_1F/2F
This macro can be used to specify the ratio of abnormal output from the TF, which is to be used in judging the end of learning. The ratio of abnormal output from the TF indicates the components in output from the TF that are not target vibration components for extraction. If learning continues after the ratio exceeds the value 1.0, components that are not required will also be learned. Therefore, setting the ratio to a value no greater

than 1.0 is recommended. Note that the correct calculation of the ratio may not be possible, depending on the phase of the reference mechanical angle used by the TF. Adjust the judgement of the end of learning through a combination of the goal value for suppression described above and the ratio of abnormal output from the TF.

10.17 Step-Skipping (Stall) Detection Parameters

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

Table 10-28 List of Configuration Information

File Name	Macro Name	Setting	Description
r_motor_module_cfg.h	CURRENT_CFG_STALL_DETECTION	MTR_DISABLE	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 10-29 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	2.2	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 50 μs and the time constant of the HPF is 8 ms, the gain of the HPF is about 0.0063.

$$HPFGain = \frac{Tc\ [s]}{HPF\ Time\ [s]} = \frac{50\ \mu s}{8\ ms} = 0.00625 \approx 0.0063$$

11. Settings for the FSP for the RA Family MCU

11.1 Overview of the FSP

Figure 11-1 shows the software architecture of this sample program for the RA Family MCU. 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 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.

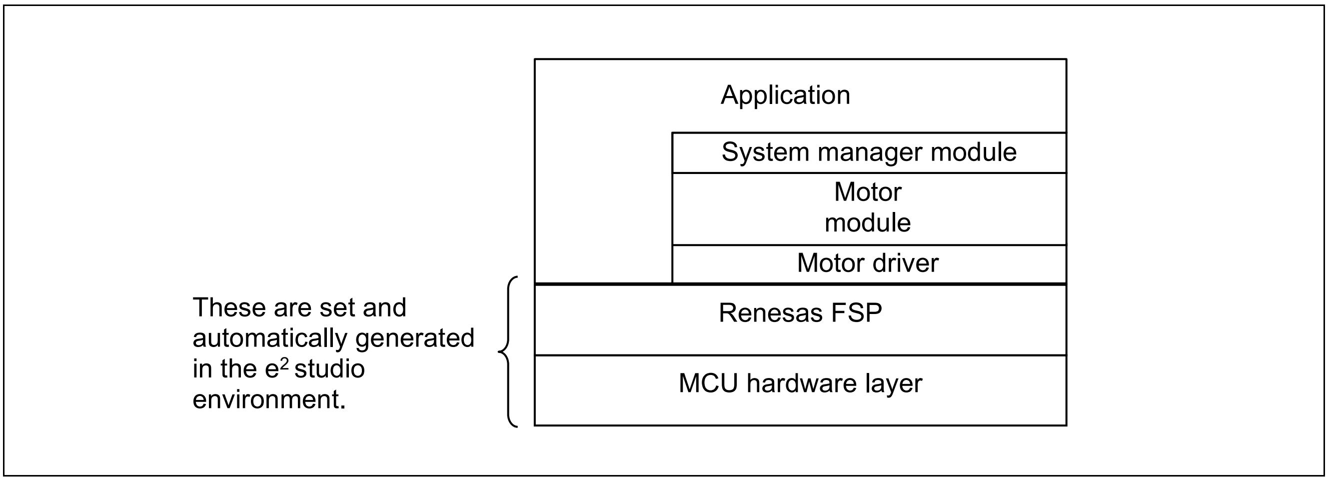


Figure 11-1 Software Architecture of This Sample Program for the RA Family MCU

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.

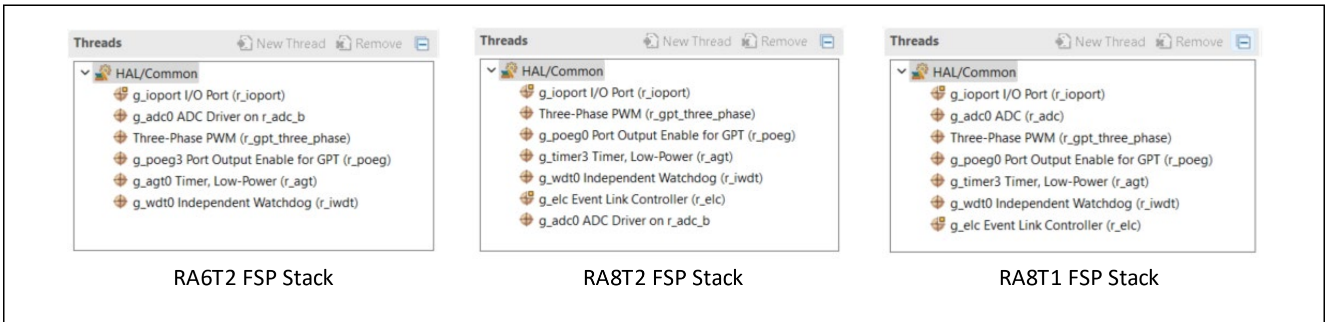


Figure 11-2 List of FSP Stacks

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, inverter bus voltage, VR input)	g_adc0 ADC Driver on r_adc_b g_adc0 ADC (r_adc)
Setting port pins to be used	g_ioport I/O Port (r_ioport)
Speed control interrupt timer	g_agt0 Timer, Low-Power (r_agt)
Overcurrent detection	g_poeg3 Port Output Enable for GPT (r_poeg)
Independent Watchdog Timer	g_wdt0 Independent Watchdog (r_iwdt)
Event Link Controller#	g_elc Event Link Controller (r_elc)

Note # : This function is not used in RA6T2.

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 (in the Case of 3-Shunt Detection for the RA MCU)

FSP Stack	Callback Function	Description
g_adc0	callback_gpt_adc_cyclic()	This function is for use in 20-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()	—

11.4 Pin Settings

Table 11-3 to Table 11-6 list the information on pin interfaces.

Table 11-3 Pin Interfaces for RA6T2

Function	**Pin Name**	**Peripheral Function**	**Pin to Which the Function is Allocated**	**Remarks**
LED1	PD01	GPIO	—	Low level indicates motor in run state
LED2	PD02	GPIO	—	Low level indicates error
SW1	PD04	GPIO	—	Board User interface to start / stop motor switch
SW2	PD07	GPIO	—	RESET switch
Detection of the U-phase current	PA04	S12AD	AN004	—
Detection of the V-phase current	PA02	S12AD	AN002	—
Detection of the W-phase current	PA00	S12AD	AN000	—
Detection of the Inverter bus voltage	PA07	S12AD	AN007	—
Volume Switch	P000	S12AD	AN016	Board user interface for Speed command
Detection of the overcurrent in the inverter hardware	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

Table 11-4 Pin Interfaces for RA8T1

Function	**Pin Name**	**Peripheral Function**	**Pin to Which the Function is Allocated**	**Remarks**
LED1	PA12	GPIO	—	Low level indicates motor in run state
LED2	PA14	GPIO	—	Low level indicates error
SW1	PA15	GPIO	—	Board User interface to start / stop motor switch
SW2	PA13	GPIO	—	RESET switch
Detection of the U-phase current	P004	S12AD	AN000	—
Detection of the V-phase current	P005	S12AD	AN001	—
Detection of the W-phase current	P006	S12AD	AN002	—
Detection of the Inverter bus voltage	P008	S12AD	AN008	—
Volume Switch	P014	S12AD	AN007	Board user interface for Speed command
Detection of the overcurrent in the inverter hardware	P613	POEG	GTETRGA	The low level indicates the abnormal state.
PWM output (U_p)	P115	GPT	GTIOC5A	Active high
PWM output (U_n)	P609	GPT	GTIOC5B	Active high
PWM output (V_p)	P113	GPT	GTIOC2A	Active high
PWM output (V_n)	P114	GPT	GTIOC2B	Active high
PWM output (W_p)	P300	GPT	GTIOC3A	Active high
PWM output (W_n)	P112	GPT	GTIOC3B	Active high

Table 11-5 Pin Interfaces for RA8T2

Function	Pin Name	Peripheral Function	Pin to Which the Function is Allocated	Remarks
LED1	P614	GPIO	—	Low level indicates motor in run state
LED2	PA15	GPIO	—	Low level indicates error
SW1	PA00	GPIO	—	Board User interface to start / stop motor switch
SW2	PA07	GPIO	—	RESET switch
Detection of the U-phase current	P006	S12AD	AN006	—
Detection of the V-phase current	P008	S12AD	AN008	—
Detection of the W-phase current	P010	S12AD	AN010	—
Detection of the Inverter bus voltage	P007	S12AD	AN007	—
Volume Switch	P015	S12AD	AN015	Board user interface for Speed command
Detection of the overcurrent in the inverter hardware	P112	POEG	GTETRGA	The low level indicates the abnormal state.
PWM output (U_p)	P605	GPT	GTIOC8A	Active high
PWM output (U_n)	P604	GPT	GTIOC8B	Active high
PWM output (V_p)	P603	GPT	GTIOC7A	Active high
PWM output (V_n)	P602	GPT	GTIOC7B	Active high
PWM output (W_p)	P612	GPT	GTIOC9A	Active high
PWM output (W_n)	P613	GPT	GTIOC9B	Active high

Table 11-6 Pin Interfaces for RX26T-A

Function	Pin Name	Peripheral Function	Pin to Which the Function is Allocated	Remarks
LED1	P21	GPIO	—	Low level indicates motor in run state
LED2	P20	GPIO	—	Low level indicates error
SW1	P23	GPIO	—	Board User interface to start / stop motor switch
SW2	P22	GPIO	—	RESET switch
Detection of the U-phase current	P40	S12AD	AN000	—
Detection of the V-phase current	P41	S12AD	AN001	—
Detection of the W-phase current	P42	S12AD	AN002	—
Detection of the Inverter bus voltage	P43	S12AD	AN003	—
Volume Switch	P50	S12AD	AN204	Board user interface for Speed command
Detection of the overcurrent in the inverter hardware	P70	POEG	GTETRGB / POE0#	The low level indicates the abnormal state.
PWM output (U_p)	P73	GPT	GTIOC2A	Active high
PWM output (U_n)	P76	GPT	GTIOC2B	Active high
PWM output (V_p)	P72	GPT	GTIOC1A	Active high
PWM output (V_n)	P75	GPT	GTIOC1B	Active high
PWM output (W_p)	P71	GPT	GTIOC0A	Active high
PWM output (W_n)	P74	GPT	GTIOC0B	Active high

11.5 Settings for the Three-Phase PWM GPT

The three-phase PWM GPT is used in motor control.

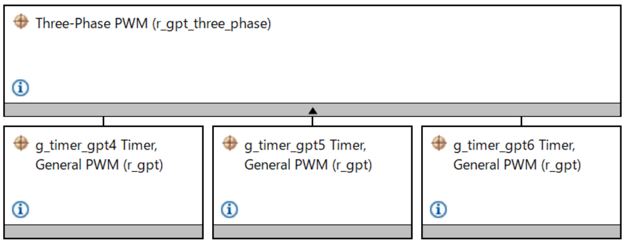


Figure 11-3 Stacks Related to the Three-Phase PWM GPT Stack

Table 11-7 Three-Phase PWM Settings

Function and Item for Setting			**Setting**		
			RA6T2	**RA8T1**	**RA8T2**
General	Name		g_three_phase0		
	Mode		Triangle-Wave Symmetric PWM		
	Period		50		
	Period Unit		Microseconds		
	GPT U-Channel		4	5	8
	GPT V-Channel		5	2	7
	GPT W-Channel		6	3	9
	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)	120	120	250
		Dead Time Count Down (Raw Counts)	120	120	250

Table 11-8 U-Phase GPT Settings

Function and Item for Setting				**Setting**		
				RA6T2	**RA8T1**	**RA8T2**
Module <Name> Timer	General	Name		g_timer_gpt4	g_timer0	
	Extra Features	Output Disable	POEG Link	POEG Channel 3	POEG Channel 0	POEG Channel 0
			GTIOCA Disable setting	Set HiZ	Set HiZ	Set HiZ
			GTIOCB Disable Setting	Set HiZ	Set Hi,Z	Set HiZ
		ADC Trigger		Start Event Trigger • Request A during Down-Counting	—	
				ADC A Compare Match: 0	—	
		Interrupt Skipping	Interrupt to Count	None	None	None
			Interrupt Skip Count	0	0	0
Pins				GTIOC4A (PB04)	GTIOC5A (P115)	GTIOC8A (P605)
				GTIOC4B (PB05)	GTIOC5B (P609)	GTIOC8B (P604)

Table 11-9 V-Phase GPT Settings

Function and Item for Setting			**Setting**		
			RA6T2	**RA8T1**	**RA8T2**
Module <Name> Timer	General	Name	g_timer_gpt5	g_timer1	
	The settings of the other items are omitted because they are automatically made as part of the three-phase PWM settings.				
Pins			GTIOC5A (PB06)	GTIOC2A (P113)	GTIOC7A (P603)
			GTIOC5B (PB07)	GTIOC2B (P114)	GTIOC7B (P602)

Table 11-10 W-Phase GPT Settings

Function and Item for Setting			**Setting**		
			RA6T2	**RA8T1**	**RA8T2**
Module <Name> Timer	General	Name	g_timer_gpt6	g_timer2	
	The settings of the other items are omitted because they are automatically made as part of the three-phase PWM settings.				
Pins			GTIOC6A (PB08)	GTIOC3A (P300)	GTIOC9A (P612)
			GTIOC6B (PB09)	GTIOC3B (P112)	GTIOC9B (P613)

11.6 AGT0 Settings (Setting the Interval for Speed Control)

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

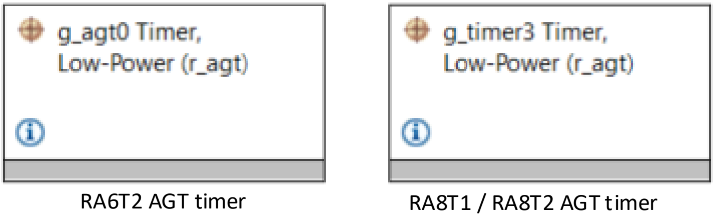


Figure 11-4 RA Family MCU AGT Timer stack

Table 11-11 AGT Settings for the Speed Control Interval

Function and Item for Setting		**Setting**		
		RA6T2	**RA8T1**	**RA8T2**
General	Name	g_agt0	g_timer3	g_timer3
	Counter Bit Width	AGT 32-bit	AGT 16-bit	AGT-16bit
	Channel	0		
	Mode	Periodic		
	Period	500	500	500
	Period Unit	Microseconds	Microseconds	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>		

11.7 ADC Settings

The 12-bit A/D converters in the MCU are used to measure the U-, V-, and W-phase output currents (in 3-shunt detection), variable resistor voltage for board user interface and inverter bus voltage. Table 11-12 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.

With the 3-shunt current detection method, 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.

Table 11-12 Settings for ADC Channels to Which the Respective Functions are Allocated and Timing of Detection for the RA Family MCU

Function	**Channel to be Allocated (ADC0 used)**			**Trigger for Starting A/D Conversion**
	RA6T2	**RA8T1**	**RA8T2**	
Detection of the inverter bus voltage	7	8	7	Counting down reaching 0
Detection of the U-phase current	4	0	6	
Detection of the V-phase current	2	1	8	
Detection of the W-phase current	0	2	10	
Detection of the variable resistor for Board UI speed command	16	7	15	

RA6T2 and RA8T2 use the ADC_B peripheral for ADC0 while the other RA Family MCU (RA8T1) uses the ADC12 peripheral of the device.

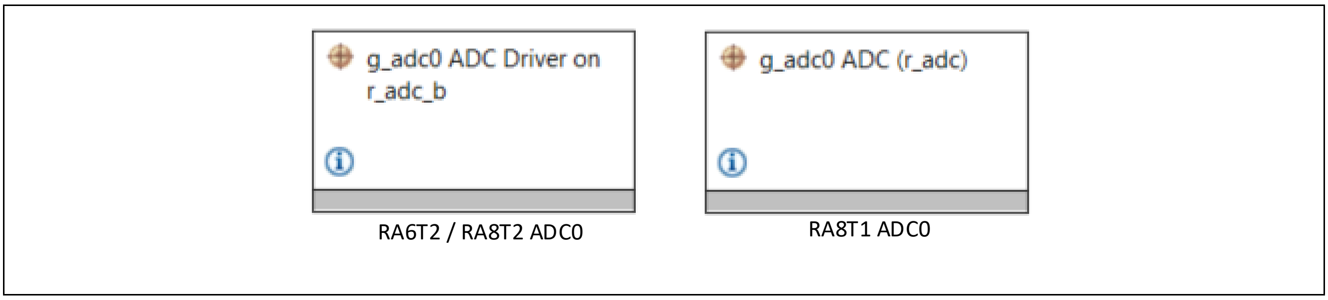


Figure 11-5 ADC Stack

Table 11-13 ADC Settings for RA6T2 and RA8T2

Function and Item for Setting			**Setting**	
			RA6T2	**RA8T2**
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	5
		ADC1	6	5
	Synchronous Operation	Enable for ADC 0	Disable	
		Enable for ADC 1	Disable	
		Synchronous Operation Period Cycle	100	
	Calibration/A/D Calibration	Sampling Time	10	
		Conversion Time	6	
	Calibration/Sample and Hold Calibration	Sampling Time	25	95
		Hold Time	3	5
	Sampling State Table	Entry 0	10	10
		Entry 1	4	95
		Entry 2	24	95
		Entries 3 to 15	95	
	Name		g_adc0	
Clock Configuration	Divider		Div /1	Div /3
	Source		PCLKC	PCLKA
Interrupts	Limiter Clip Priority		All interrupts disabled	
	Conversion Error Priority		All interrupts disabled	
	Overflow Priority		All interrupts disabled	
	Calibration End Priority		Priority 12	Priority 15
	Scan End Priority	Group 0	Priority 5	
		Group 1	Disabled	
		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	☑	☑
		Unit 4	☐	☐
		Unit 5	☐	☐
		Unit 6	☐	☐
	Analog Channels 0 to 5	Sampling Time	60	95
		Hold Time	3	5
	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**	**ADC Channel 6**
		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	
		Input Mode Selection	—	Single-ended input
		Data Sign Selection	—	Unsigned
	Virtual Channel 1	Scan Group	**Scan Group 0**	
		Channel Select	**AN002**	**ADC Channel 8**
		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	
		Input Mode Selection	—	Single-ended input
		Data Sign Selection	—	Unsigned
	Virtual Channel 2	Scan Group	**Scan Group 0**	
		Channel Select	**AN004**	**ADC Channel 10**
		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	
		Input Mode Selection	—	Single-ended input
		Data Sign Selection	—	Unsigned
	Virtual Channel 3	Scan Group	**Scan Group 1**	
		Channel Select	**AN007**	**ADC Channel 7**
		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	
		Input Mode Selection		—	Single-ended input
		Data Sign Selection		—	Unsigned
	Virtual Channel 4	Scan Group		**Scan Group 0**	**Scan Group 1**
		Channel Select		**AN016**	**ADC Channel 15**
		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	
		Input Mode Selection		—	Single-ended input
		Data Sign Selection		—	Unsigned
	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 Channel 5 Request A	☐	☐
		GPT Trigger Enable	GPT Channel 6 Request A	☐	☐

		GPT Trigger Enable	GPT Channel 7 Request A	☐	☐
		GPT Trigger Enable	GPT Channel 8 Request A	☐	☑
		GPT Trigger Enable	GPT Channel 9 Request A	☐	☐
		GPT Trigger Enable		Not in use • GPT Channel 0 to 9 Request B	Not in use • GPT Channel 9 to 13 Request A • GPT Channel 0 to 13 Request B
		Enable		Enable	
		Converter Selection		ADC 0	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 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 Channel 5 Request A	☐	☐
		GPT Trigger Enable	GPT Channel 6 Request A	☐	☐
		GPT Trigger Enable	GPT Channel 7 Request A	☐	☐

		GPT Trigger Enable	GPT Channel 8 Request A	☐	☑
		GPT Trigger Enable	GPT Channel 9 Request A	☐	☐
		GPT Trigger Enable		Not in use • GPT Channel 0 to 9 Request B	Not in use • GPT Channel 9 to 13 Request A GPT Channel 0 to 13 Request B
		Enable		Enable	
		Converter Selection		**ADC 1**	
		Start Trigger Delay		0	
		Scan End Interrupt Enable		Disable	
		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	

Table 11-14 ADC Settings for RA8T1

<table>
<tr><th colspan="4" rowspan="2">Function and Item for Setting</th><th>Settings</th></tr>
<tr><th>RA8T1</th></tr>
<tr><td rowspan="7">General</td><td>Name</td><td colspan="2"></td><td>g_adc0</td></tr>
<tr><td>Unit</td><td colspan="2"></td><td>0</td></tr>
<tr><td>Resolution</td><td colspan="2"></td><td>12-bit</td></tr>
<tr><td>Alignment</td><td colspan="2"></td><td>Right</td></tr>
<tr><td>Clear after read</td><td colspan="2"></td><td>Off</td></tr>
<tr><td>Mode</td><td colspan="2"></td><td>Single Scan</td></tr>
<tr><td>Double-trigger</td><td colspan="2"></td><td>Disabled</td></tr>
<tr><td rowspan="31">Input</td><td rowspan="12">Channel Scan Mask</td><td colspan="2">Channel 0</td><td>☑</td></tr>
<tr><td colspan="2">Channel 1</td><td>☐</td></tr>
<tr><td colspan="2">Channel 2</td><td>☑</td></tr>
<tr><td colspan="2">Channel 3</td><td>☐</td></tr>
<tr><td colspan="2">Channel 4</td><td>☐</td></tr>
<tr><td colspan="2">Channel 5</td><td>☐</td></tr>
<tr><td colspan="2">Channel 6</td><td>☐</td></tr>
<tr><td colspan="2">Channel 7</td><td>☑</td></tr>
<tr><td colspan="2">Channel 8</td><td>☑</td></tr>
<tr><td colspan="2">Channel 9 to 28</td><td>Not in use (by default)</td></tr>
<tr><td colspan="2">Temperature Sensor</td><td>☐</td></tr>
<tr><td colspan="2">Internal Reference Voltage</td><td>☐</td></tr>
<tr><td rowspan="3">Group B Scan Mask</td><td colspan="2">Channel 0 to 28</td><td>Not in use (by default)</td></tr>
<tr><td colspan="2">Temperature Sensor</td><td>☐</td></tr>
<tr><td colspan="2">Internal Reference Voltage</td><td>☐</td></tr>
<tr><td rowspan="3">Addition/ Averaging Mask</td><td colspan="2">Channel 0 to 28</td><td>Not in use (by default)</td></tr>
<tr><td colspan="2">Temperature Sensor</td><td>☐</td></tr>
<tr><td colspan="2">Internal Reference Voltage</td><td>☐</td></tr>
<tr><td rowspan="4">Sample and Hold</td><td rowspan="3">Sample and Hold Channels</td><td>Channel 0</td><td>☑</td></tr>
<tr><td>Channel 1</td><td>☐</td></tr>
<tr><td>Channel 2</td><td>☑</td></tr>
<tr><td colspan="2">Sample Hold States</td><td>24</td></tr>
<tr><td rowspan="6">Window Compare</td><td colspan="2">Window A</td><td>Not in use (by default)</td></tr>
<tr><td colspan="2">Window B</td><td>Not in use (by default)</td></tr>
<tr><td colspan="2">Window Mode</td><td>Disable</td></tr>
<tr><td colspan="2">Event Output</td><td>OR</td></tr>
<tr><td colspan="2">Add/Average Count</td><td>Disabled</td></tr>
<tr><td colspan="2">Reference Voltage control</td><td>VREFH0/VREFH</td></tr>
<tr><td rowspan="2">Interrupts</td><td>Normal/Group A Trigger</td><td colspan="2"></td><td>GPT5 COUNTER UNDERFLOW (Underflow)</td></tr>
<tr><td>Group B Trigger</td><td colspan="2"></td><td>Disabled</td></tr>
</table>

	Group Priority		Group A cannot interrupt Group B
	Callback		callback_gpt_adc_cyclic
	Scan End Interrupt Priority		Priority 5
	Scan End Group B Interrupt Priority		Disabled
	Window Compare A Interrupt Priority		Disabled
	Window Compare B Interrupt Priority		Disabled
Extra	ADC Ring Buffer		Disabled
Pins	ADTRG0		—
	AN000		P004
	AN001		P005
	AN002		P006
	AN004		P007
	AN005		P010
	AN006		P009
	AN007		P014
	AN008		P008
	AN011		—
	AN012		—
	AN013		—
	AN016		P513
	AN017		P805
	AN018		P806
	AN019		P502
	PGAVSS000		—

11.8 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 11-15 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.

Figure 11-6 POEG Stack

Table 11-15 POEG Settings

Function and Item for Setting			**Setting**		
			RA6T2	**RA8T1**	**RA8T2**
General	Trigger	GTETRG Pin	☑		
		GPT Output Level	☐		
		Oscillation Stop	☐		
		ACMPHS0	☐		
		ACMPHS1	☐		
		ACMPHS2	☐	—	☐
		ACMPHS3	☐	—	☐
		DSMIF0	—		☐
		DSMIF1			☐
	Name		g_poeg3	g_poeg0	
	Channel		3	0	0
Input	GTETRG Polarity		Active Low		
	GTETRG Noise Filter		PCLKB/32	PCLKB/128	PCLKB/32
Interrupts	Callback		callback_poe_overcurrent		
	Interrupt Priority		Priority 0 (highest)		

11.9 ELC Settings

The ELC is used to link ADC scan end interrupt request in RA8T1 to GPT counter underflow. ELC is not required for RA6T2 and RA8T2.

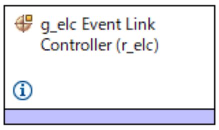


Figure 11-7 ELC Stack

Table 11-16 ELC Settings

Function and Item for Setting			Setting
Module	Name		g_elc

Table 11-17 ELC Settings (Event Links Tab)

Event Links Configuration / Allocations	
Peripheral Function	Event
ADC12A0	RA8T1 : • GPT5 COUNTER UNDERFLOW (Underflow)
others	No allocation

11.10 IWDT Settings

The IWDT consists of a 14-bit down counter that can reset the MCU when the system goes out of control. The IWDT stack is added to use the API, but the settings themselves are configured on the BSP tab.

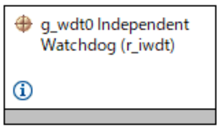


Figure 11-8 IWDT Stack

Table 11-18 IWDT Settings

Function and Item for Setting			Setting
Module	Name		g_wdt0

Table 11-19 IWDT Settings (BSP Tab) for RA6T2

Function and Item for Setting			Setting
RA6T2 Family	OFS0 register	Start Mode	IWDT is automatically activated after a reset (Autostart mode)
		Timeout Period	2048 cycles
		Dedicated Clock Frequency Divisor	1
		Window End Position	0% (no window end position)
		Window Start Position	100% (no window start position)

		Reset Interrupt Request Select	Reset is enabled
		Stop Control	Stop counting when in Sleep, Snooze mode, or Software Standby

Table 11-20 IWDT Settings (BSP Tab) for RA8T1, RA8T2

Function and Item for Setting			**Setting**
RA8T1 Family RA8T2 Device Options	OFS0 register	Start Mode	IWDT is stopped after a reset
		Timeout Period	2048 cycles
		Dedicated Clock Frequency Divisor	128
		Window End Position	0% (no window end position)
		Window Start Position	100% (no window start position)
		Reset Interrupt Request Select	Reset is enabled
		Stop Control	Stop counting when in Sleep, Snooze mode, or Software Standby

12. Settings for the Smart Configurator for the RX26T

12.1 Overview of the SC

In the sample program for the RX26T, the Smart Configurator (SC) is used to create a project for the hardware abstraction layer (HAL). This section describes the components used in the sample program and the functions added to the user area. For the MCUs of the RA family, the flexible software package (FSP) instead of the SC is used to set up the HAL.

The only differences between the sample programs for the RA Family and RX26T are the HAL and MCU; most modules in the application layer are used in common for both MCUs.

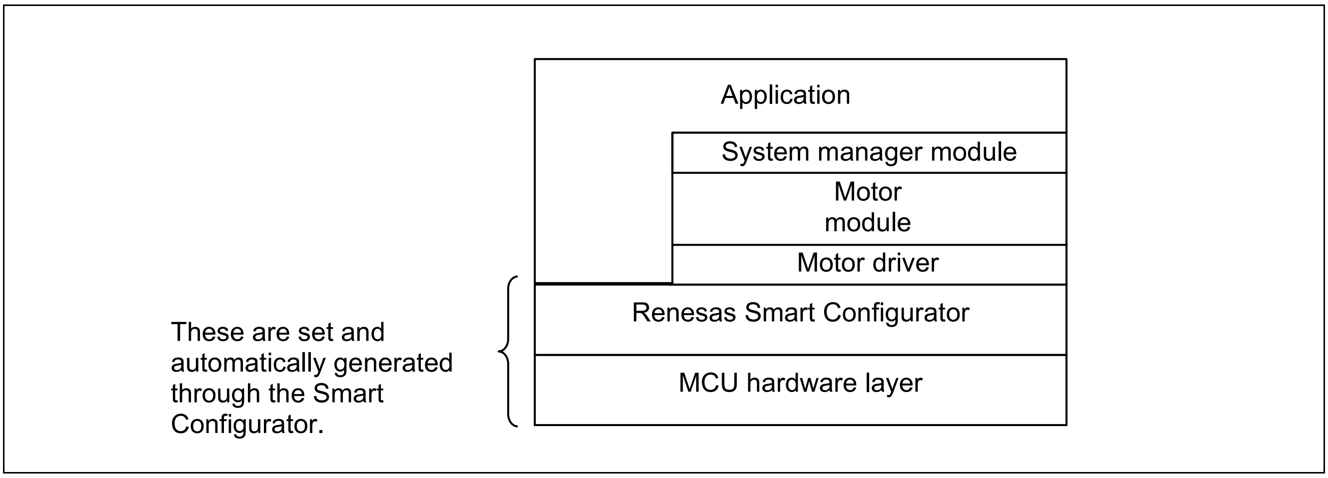


Figure 12-1 Software Architecture of This Sample Program for the RX26T

12.2 Clock Settings

Table 12-1 lists the clock settings which can be set in the Clocks tab of SC.

Table 12-1 MCU Clock Settings

Type of Clock	Setting
Main clock	10 MHz
System clock (ICLK)	120 MHz
Peripheral module clock (PCLKA)	120 MHz
Peripheral module clock (PCLKB)	60 MHz
Peripheral module clock (PCLKC)	120 MHz
Peripheral module clock (PCLKD)	60 MHz
Flash memory interface clock (FCLK)	60 MHz
IWDTCLK	120 kHz

12.3 Component Settings

Table 12-2 lists the components for use with this sample program and the functions allocated to each of them.

Table 12-2 Smart Configurator Components and the Functions Allocated to Each of Them

Function	**Component**
W-phase PWM output	Config_GPT0
V-phase PWM output	Config_GPT1
U-phase PWM output	Config_GPT2
A/D conversion processing (phase currents, inverter bus voltage)	Config_S12AD0
A/D conversion processing (variable resistor for Board UI speed command voltage)	Config_S12AD2
Setting port pins to be used	Config_PORT
Speed control interrupt timer	Config_CMT0
Independent watchdog timer	Config_IWDT
Inverter hardware overcurrent detection	Config_POEG

12.4 AD Settings

The 12-bit A/D converter (S12AD) in the MCU is used to measure the U- and W-phase output currents, inverter bus voltage, reactor current, input AC voltage, and inverter bus current. Table 12-3 shows the channels to which the respective functions are allocated and the timing of detection.

Table 12-3 Settings for AD Channels to Which the Respective Functions are Allocated and Timing of Detection for the RX26T

Function	**Channel**		**Trigger for Starting A/D Conversion**
Detection of the U-phase current (3-shunt)	AN000		Compare match between the GTCNT and GTADTRA in GPT0. The GTADTRA is set to 0 to start A/D conversion at the troughs of the carrier wave (20 kHz).
Detection of the V-phase current (3-shunt)	AN001		
Detection of the W-phase current (3-shunt)	AN002		
Detection of the inverter bus voltage	AN003		
Detection of Variable resistor voltage for Board UI speed command	RX26T Type A	AN204	

12.5 CMT Settings

In this sample program, a Compare Match Timer (CMT) is used to generate a speed-controlled periodic interrupt.

Table 12-4 Settings for Compare Match Timer

Component	Item for settings	Settings
Config_CMT0	Clock settings	PCLK/8
	Interval times	500 μs
	Allow Compare Match Interrupt (CMI0)	☑
	Priority	Level 6

12.6 GPT Settings

This sample program uses the GPT to output PWM waveforms.

Table 12-5 Settings for GPT0 to GPT2 for Motor Control

Item for Setting		**Setting (3-Shunt)**
Settings of Counting	Clock source	120 MHz
	Timer operation period	50 μs
	Period setting register value	3000
	Buffer operation	Single buffer
	Count direction	Counting up
	Initial counter value	0
GTCCRA	GTCCRA function	Compare match
	Buffer operation	Single buffer
	GTIOC0A function	PWM output pin
	GTIOC0A output duty cycle	Determined by a compare match

	GTIOC0A pin negating control	Hi-Z
	Output level when counting begins or stops	0 is output when counting begins. 0 is output when counting stops.
	Output level at a compare match	Toggled output
	Output level at the end of a cycle	The output level is retained.
GTCCRB	GTCCRB function	Compare match
	Buffer operation	Single buffer
	GTIOC0B function	PWM output pin
	GTIOC0B output duty cycle	Determined by a compare match
	GTIOC0B pin negating control	Hi-Z
	Output level when counting begins or stops	1 is output when counting begins. 1 is output when counting stops.
	Output level at a compare match	Toggled output
	Output level at the end of a cycle	The output level is retained.
GTCCRC function		GTCCRA buffer register
GTCCRD function		Compare match
GTCCRE function		GTCCRB buffer register
GTCCRF function		Compare match
Starting counting by a software source		☑
Stopping counting by a software source		☑
Automatic dead time setting	Automatic setting	☑
	GTDVU value	120
	GTDVD = GTDVU	☑
A/D conversion start request settings	GTADTRA (Only for GPT0)	During counting up
	GTADTRA compare-match value (Only for GPT0)	1
	GTATDRB	—
	GTADTRB compare-match value	—
Output stop group	Group selection	B

12.7 POEG Settings

Table 12-6 shows the POEG functions that are specifiable through Config_POEG. The output pin settings depend on the specifications of the inverter. Confirm the signal specifications of the inverter you are using.

Table 12-6 POEG Settings for the RX26T

Function		**Setting**
GTETRGB settings	Inversion of input	☑
	Noise filter	☑ Sampling at every PCLK_GPTB clock cycle. Number of sampling: 3 times
	Detection mode	Detection on edges
	Output stop request from the GTETRGB pin	☑
Interrupt setting	Enabling the POEGGBI interrupt	Level 15

12.8 IWDT Settings

Table 12-7 shows the IWDT functions that are specifiable through Config_IWDT.

Table 12-7 IWDT Settings for the RX26T

Function		Setting
Start mode setting	-	Register start mode
IWDTCLK clock setting	Clock division ratio selection	IWDTCLK
	Frequency	120 kHz
	Timeout cycle	1024 cycle
	Timeout period	8.533333 ms
Window position	Window start position	100 %
	Window end position	0 %
Sleep mode count stop control setting	-	Enabled
Reset interrupt request select	-	Reset output

12.9 Interrupt Settings

Table 12-8 lists the information on interrupts in the MCU to be specified through motor components.

Table 12-8 List of Interrupts for the RX26T

Component	Interrupt Function	Description
Config_POEG	r_Config_POEG_poeggbi_interrupt	Inverter hardware overcurrent interrupt Interrupt level: 15
Config_S12AD	r_Config_S12AD0_interrupt	Motor A/D conversion end interrupt Interrupt level: 10
Config_CMT0	r_Config_CMT0_cmi0_interrupt	Speed control interrupt Interrupt level: 6

12.10 Pin Settings

Table 12-9 list the information on pin interfaces.

Table 12-9 Pin Interfaces for RX26T

Function	Pin Name	Peripheral Function	Pin to Which the Function is Allocated	Remarks
LED1	P21	PORT	—	These allow use of the LEDs on the CPU board by the user.
LED2	P20	PORT	—	
SW1	P23	PORT	—	Board User interface to start / stop motor switch
SW2	P22	PORT	—	RESET switch
Detection of the U-phase current	P40	S12AD0	AN000	—
Detection of the V-phase current	P41	S12AD0	AN001	—
Detection of the W-phase current	P42	S12AD0	AN002	—
Detection of the Inverter bus voltage	P43	S12AD0	AN003	—
Volume Switch	P50	S12AD2	AN204	Board user interface for Speed command
Detection of the Overcurrent in the inverter hardware	P70	POEG	GTETRGB	A falling edge of the signal on the pin indicates the abnormal state.
PWM output (U_p)	P73	GPT2	GTIOC2A	Active high
PWM output (U_n)	P76	GPT2	GTIOC2B	Active high
PWM output (V_p)	P72	GPT1	GTIOC1A	Active high
PWM output (V_n)	P75	GPT1	GTIOC1B	Active high
PWM output (W_p)	P71	GPT0	GTIOC0A	Active high
PWM output (W_n)	P74	GPT0	GTIOC0B	Active high

13. Results of Evaluation

13.1 Evaluation of Motor Control

13.1.1 Transition from startup to closed loop speed control

In sensorless vector control, where the rotor position is estimated using the BEMF observer, it is necessary to accelerate the speed to a range where a sufficiently large inductive voltage can be generated for accurate estimation. Therefore, after aligning the d-axis with the motor's U-phase axis, the speed is accelerated to the transition speed using open-loop control. Once the transition speed is reached, the control transitions to closed-loop speed control (PI control), enabling high torque and high-efficiency drive. Figure 13-1 shows the process of accelerating to 600 r/min using open-loop control and then transitioning to sensorless vector control using the BEMF observer.

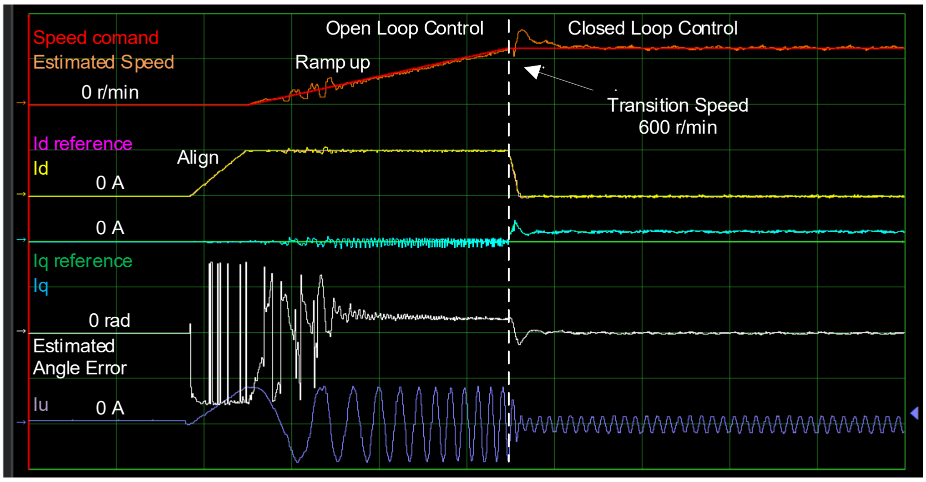


Figure 13-1 Waveforms during the transition from startup to closed-loop speed control

13.1.2 Load Characteristics

The load characteristic waveforms for 2000 r/min operation are shown below. Load tests have been worked from 600 r/min to 2000 r/min to confirm that operation at the rated load is possible.

Figure 13-2 Load characteristics (at 2000r/min operation)

13.1.3 Flying Start

Flying start (free-run restart) is a function that starts sensorless vector control (closed-loop speed control) from a state where the motor is rotating with the inverter off. When attempting to start sensorless vector control without knowing the rotor position and speed, unexpected behaviors such as overcurrent may occur, leading to startup failure. Therefore, it is common to stop the motor with a brake or similar method, perform open-loop control, and then transition to sensorless vector control to drive the motor at the desired speed. On the other hand, flying start allows direct initiation of sensorless vector control by estimating the rotor position and speed of the rotating motor, significantly reducing startup time. Figure 13-3 shows the "active brake restart" and "flying start restart" during free-run deceleration.

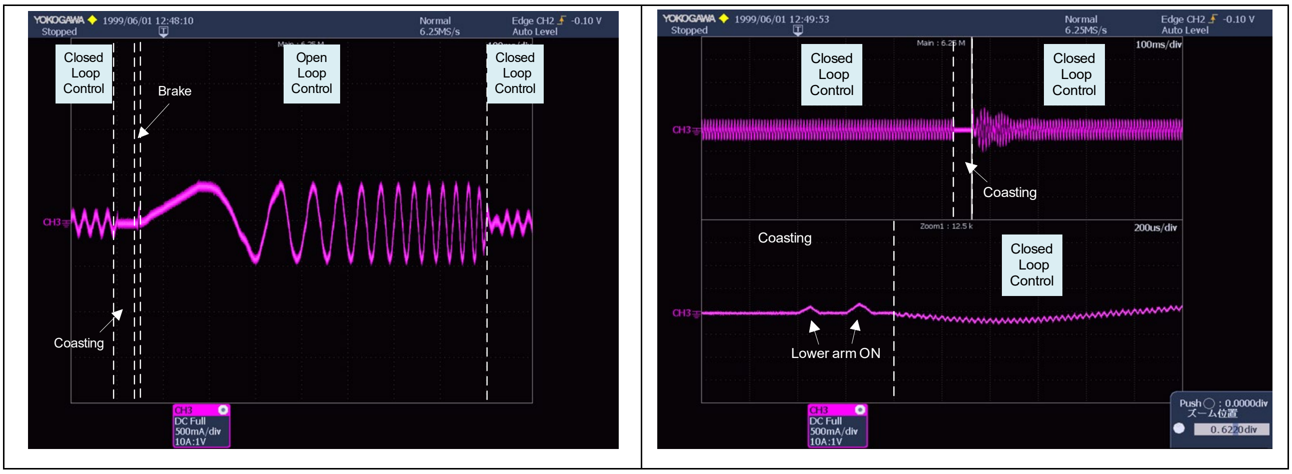


Figure 13-3 Current Waveforms during Operation in Flying Start
(Left: Active Braking; Right: Flying Start)

13.2 CPU Utilization

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

Table 13-1 Control Loops and CPU Loading Rates (RA Family MCU)

Target Device	Control Loop Type	Control Interval	Processing Time	CPU Loading Rate
RA6T2	Current control loop in motor control	50 μs (no decimation)	17.3 μs	34.6 %
	Speed control loop in motor control	500 μs	4.4 μs	0.9 %
RA8T1	Current control loop in motor control	50 μs (no decimation)	15.0 μs	29.9 %
	Speed control loop in motor control	500 μs	4.0 μs	0.8 %
RA8T2	Current control loop in motor control	50 μs (no decimation)	2.8 μs	5.5 %
	Speed control loop in motor control	500 μs	0.7 μs	0.1 %

Table 13-2 Control Loops and CPU Loading Rates (RX26T)

Target Device	Control Loop Type	Control Interval	Processing Time	CPU Loading Rate
RX26T-A	Current control loop in motor control	50 μs (no decimation)	13.7 μs	27.4 %
	Speed control loop in motor control	500 μs	6.5 μs	1.3 %

13.3 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 13-3 Program Size and RAM Usage (RA Family MCU)

Target Device	**Memory**	**Size**
RA6T2	Program size (ROM)	37576 bytes
	RAM usage	6360 bytes
	Maximum value of stack analysis results	488 bytes
	Stack size setting in the IDE	1024 bytes
RA8T1	Program size (ROM)	36828 bytes
	RAM usage	6396 bytes
	Maximum value of stack analysis results	572 bytes
	Stack size setting in the IDE	1024 bytes
RA8T2	Program size (ROM)	38880 bytes
	RAM usage	6432 bytes
	Maximum value of stack analysis results	572 bytes
	Stack size setting in the IDE	1024 bytes

Table 13-4 Program Size and RAM Usage (RX26T)

Target Device	Memory	Size
RX26T-A	Program size (ROM)	29641 bytes
	RAM usage	12993 bytes
	Maximum value of stack analysis results	476 bytes
	Stack size setting in the IDE	5120 bytes

14. FAQ

Table 14-1 lists typical problems and examples of their solutions.

Table 14-1 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.
I attempted to start up the motor, but it did not run.	The load being higher or the inertia being greater than intended with respect to the motor under control may lead to failure to start up the motor under open-loop control. Review the d-axis current reference and the amounts of increase and decrease in speed under open-loop control. See 10.13 (1) for details.
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.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jan. 30, 2026	—	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.

Corporate Headquarters

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

Trademarks

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/.