RX Family

Sensorless Vector Control for IPMSM over the Whole Speed Range for MCK

Introduction

This sample program provides a software that can drive 3-phase interior permanent magnet (IPM) motor using sensorless vector control (Sensorless FOC) over the whole speed range from standstill to low, middle, and high speed ranges.

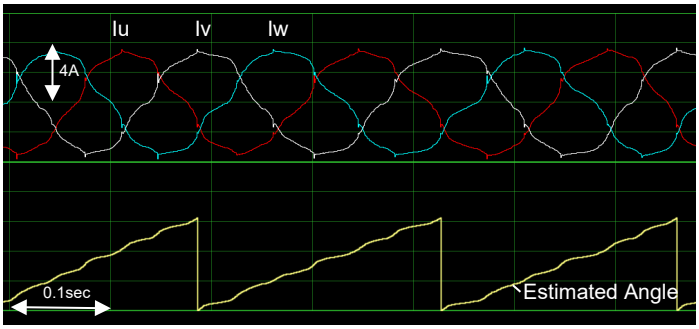
The algorithm estimates the magnetic pole position by injecting a high-frequency pulse voltage (HFI) when the motor is stopped or at lower speeds based on the characteristic difference[1] between d-axis inductance (Ld) and q-axis inductance (Lq), which is a characteristic of IPM motors, thus eliminating the need for a magnetic pole position sensor. At startup, it also determines the N and S poles of the magnetic pole by using the magnetic saturation phenomenon of the IPM motor to accurately estimate the magnetic pole position.

In the middle to high speed range of 300r/min or more, operation is continued using a functionality that automatically switches the operation to sensorless vector control in which the BEMF observer is used.

In the conventional sample program, current draw control is performed by open loop in operation from standstill to low speed range. As a result, there was a time lag of several seconds or more until the operation starts, the rated load could not be applied from standstill to low speed range, and a strong magnetic flux was generated causing issues in terms of energy consumption. This software solves the conventional problems by quickly estimating the magnetic pole position from standstill and controlling the motor in a fully closed-loop manner using sensorless vector control.

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

Figure 1-1 shows the waveforms when operated with this sample program. It shows the estimated angle waveforms and output current waveforms of phases U, V, and W on the RMW with 50% load applied at a speed command of 40 r/min.



(a) 40r/min load current characteristics

Figure 1-1 Example of performance characteristics in sensorless control

Target device

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

- RX26T RAM64KB Version (R5F526TFCDFP)
- RX26T RAM48KB Version (R5F526TACDFM)

[1] This algorithm is not applicable to a surface permanent magnet synchronous motor (SPMSM) that has no characteristic difference between Ld and Lq or an IPM motor having conditions where the characteristic difference becomes less than 20% while the motor is started up, stopped, or operating.

Contents

1. Introduction 6

2. Glossary 8

3. Used hardware and software 9
3.1 List of used hardware 9
3.2 List of used software 10

4. Building a hardware environment 11
4.1 Hardware environment overview 11
4.2 Preparing the power supply 11
4.3 Preparing the motor and load system 11
4.4 Preparing the inverter 13
4.5 Setting up the RX26T CPU board 13
4.6 Connection example of kit (MCK-RX26T) 14
4.7 On-board debugger 15
4.8 Wiring 16
4.9 Using the measurement instruments 17

5. Building a software environment 18
5.1 Using CS+ 18
5.2 Using e^2 studio 18

6. Driving the motor 19
6.1 Precautions before driving the motor 19
6.2 Operation preparation procedure 19
6.3 Connecting 20
6.4 Writing the sample program 21
6.5 Introducing RMW 21
6.6 Registering and updating the Map file 22
6.7 Variables used for working with RMW 23
6.8 Controlling the motor 24
6.9 Stopping and shutting down the motor 26

7. Motor control algorithm 27
7.1 Overview 27
7.2 Control block diagram 27
7.3 Speed control function 28
7.4 Disturbance torque/speed estimation observer 28
7.5 Maximum torque/current control 28
7.6 Current control function 29

7.7 Decoupling control 29
7.8 Sensorless function 30
7.8.1 Overview 30
7.8.2 Low-speed-range sensorless algorithm (HFI) 30
7.8.3 Middle-to-high-speed-range sensorless algorithm 34
7.9 Voltage phase lead compensation 35
7.10 Voltage error compensation 35
7.11 Pulse Width Modulation (PWM) mode 36

8. Software specifications and configuration 37
8.1 Software specifications 37
8.2 Overall configuration of the software 38
8.3 Task description 39
8.4 File and folder configuration 40
8.5 Application layer 42
8.5.1 Functions 42
8.5.2 Configurations 42
8.5.3 Structure and variable information 43
8.5.4 Macro definition 46
8.5.5 Adjustment and configuration of parameters 46
8.6 Manager module and motor control module 47
8.6.1 Functions 47
8.6.2 Module configuration diagram 48
8.6.3 Mode management 49
8.6.4 Sequence description 50
8.6.5 Startup sequence 51
8.6.6 Protection function 51
8.6.7 API 53
8.6.8 Structure and variable information 55
8.6.9 Macro definition 60
8.7 Driver module 61
8.7.1 Functions 61
8.7.2 Module configuration diagram 61
8.7.3 API 62
8.7.4 Configurations 62
8.7.5 Structure and variable information 63
8.7.6 Adjustment and configuration of parameters 64

9. Setting the parameters 65
9.1 Overview 65
9.2 MCU-related parameters 65
9.3 List of control function setting parameters 66

9.4 Protection-related parameters 67
9.5 Changing the PWM carrier frequency 67
9.6 Setting the pulse-width modulation method 68
9.7 Inverter parameters 69
9.7.1 Overview 69
9.7.2 Current detection gain 70
9.7.3 Voltage detection gain 71
9.7.4 Voltage error compensation parameters 71
9.8 Motor parameters 73
9.9 Current control parameters 75
9.10 Speed control parameters 76
9.11 Maximum torque/current control 77
9.12 Disturbance torque/speed estimation observer 77
9.13 Voltage phase lead compensation parameters 78
9.14 Sensorless control parameters 79

10. Smart Configurator setting 83
10.1 Clock settings 83
10.2 Component settings 83
10.3 AD settings 84
10.4 Motor settings 85
10.5 Interrupts 85
10.6 Details of user codes 87
10.7 POE settings 88
10.8 Pin settings 88

11. Evaluation results 90
11.1 Motor control evaluation 90
11.1.1 Acceleration/deceleration characteristics 90
11.1.2 Load characteristics 91
11.2 CPU utilization 91
11.3 Program size and RAM usage 91

12. FAQ 92
12.1 Troubleshooting 92
12.2 Frequently asked questions 93
12.2.1 Is it possible to control SPM motors with sensorless vector control over the whole speed range? 93
12.2.2 Which IPM motors can be controlled with sensorless vector control over the whole speed range? 93
12.2.3 What is magnetic saturation? 93
12.2.4 Does high-frequency pulse injection generate noise? Is there a way I can reduce it? 93
12.2.5 What is stepping out? 93
12.2.6 Is a step-out detection function included? 93

12.2.7 The value of a variable displayed in RMW is abnormal. 93

Revision History 94

1. Introduction

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

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

This sample program can control IS-94BZC (24 VDC IPM motor from Mabuchi Motor Co., Ltd.) without a sensor using RX26T (MCK-RX26T motor control kit from Renesas). This sample program supports Renesas Motor Workbench, a motor control development support tool, and therefore can be used as a user interface (UI) for checking the MCU internal data and controlling a motor. You can use the sample program for reference purposes when selecting the MCU to be used or developing software by checking how MCU functions are allocated, how control is loaded by interrupts, and other information in the sample program.

The sample program described in this application note was developed and evaluated in the IPM motor and inverter environment as described in this document and is not guaranteed to work in your IPM motor or inverter environment. Sensorless control performance may be limited by the current sensor itself; 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, algorithm refinement and parameter tuning are required.

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

- Main equipment and devices used for evaluation
  Motor control kit: MCK-RX26T (Type-A) from Renesas.
  Motor control CPU board: MCB-RX26T-TYPE-C from Renesas.
  Motor: IPM motor IS-94BZC from Mabuchi Motor Co., Ltd..

- Target software
  The following shows the target software for this application:

  — RX26T_MCBA_MCILV1_IPM_LESS_FOC_WHOLE_CSP_V100 (IDE : CS+)
  — RX26T_MCBA_MCILV1_IPM_LESS_FOC_WHOLE_E2S_V100 (IDE : e^2 studio)
  — RX26T_MCBA2_MCILV1_IPM_LESS_FOC_WHOLE_CSP_V100 (IDE : CS+)
  — RX26T_MCBA2_MCILV1_IPM_LESS_FOC_WHOLE_E2S_V100 (IDE : e^2 studio)
  — RX26T_MCBC2_MCILV1_IPM_LESS_FOC_WHOLE_CSP_V100 (IDE : CS+)
  — RX26T_MCBC2_MCILV1_IPM_LESS_FOC_WHOLE_E2S_V100 (IDE : e^2 studio)

- Reference materials
  — RX26T Group User's Manual -- Hardware (R01UH0979)
  — Renesas Motor Workbench User's Guide (R21UZ0004)
  — Smart Configurator User's Manual -- RX API Reference (R20UT4360)
  — RX Smart Configurator User Guide -- CS+ (R20AN0470)
  — RX Smart Configurator User Guide -- e^2 studio (R20AN0451)
  — MCK-RX26T User's Manual (R12UZ0111)
  — MCB-RX26T Type A User's Manual (R12UZ0112)
  — MCB-RX26T Type C User's Manual (R12UZ0127)

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

Table 1-1 List of items to be checked and the corresponding chapters

Items to be checked	Corresponding chapter
Identify and select necessary devices	3
Select a power supply	4.2
Select a motor	4.3
Select an inverter	4.4
Verify the wiring	4.8
Prepare a software development environment	5
Write the sample program to MCU	6.4
Install software for operating the motor on the PC.	6.5
Modify the sample program, and then reflect the changes in RMW	6.6
Review the inverter's internal information on a 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	8
Verify and change the inverter parameters	9.7, 9.4, 10.3, 0, 10.7, and 10.8
Verify and change the motor parameters	9.8, 9.4
Changing the PWM carrier frequency	9.5
Change the sensorless settings	9.14
Change the MCU settings	9.2,10
Check the frequently asked questions Check the troubleshooting tips	12

2. Glossary

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

Table 2-1 Glossary

Term	Description
Component	Refers to a driver module generated by Smart Configurator to facilitate the use of MCU peripheral functions.
Electrical angle	Phase angle of the output current flowing in the motor. It can be converted to a mechanical angle by dividing it by the number of pole pairs of the motor.
E2OB	E2 On-Board. This function has the function of E2Lite on the CPU board.
Emulator	A device used to program an MCU. Also called ICE.
HFI	Refers to application of a high-frequency pulse voltage (High-Frequency Injection). Often used to refer a low-speed-range sensorless algorithm.
IDE	Refers to an Integrated Development Environment, for example, e^2 studio and CS+.
Interior permanent magnet synchronous motor	See IPM motor.
Inverter bus voltage	Refers to the DC voltage fed to the inverter circuit. Also called DC intermediate voltage.
IPM motor	Also called IPMSM. This type of motors has magnets inside the rotor and are considered superior in terms of efficiency, size, and cost. They have saliency, in which the Ld and Lq are different.
Magnetic saturation	Phenomenon in which the motor is magnetically saturated and the magnetic flux is no longer intensified because a current above a certain level is applied. It causes the parameters to change, thus affecting motor control by the inverter.
Mechanical angle	Refers to the rotation angle of the motor axis. One rotation of the axis per minute is 1 r/min.
MC-COM	Refers to a set of communication jigs and tools for displaying waveforms. For details, refer to the following URL. https://www.renesas.com/en/products/microcontrollers-microprocessors/rx-32-bit-performance-efficiency-mcus/rtk0emxc90s00000bj-mc-com-renesas-flexible-motor-control-communication-board
OpenLoop	Refers to a motor control technique that does not need current or position feedback signals to control the voltage.
RMW	Refers to Renesas Motor Workbench, which is the operation software specifically designed for motor control.
salient PMSM	See IPMSM.
Sensorless	In this document, this is used to indicate that there is no magnetic pole position sensor or speed sensor. Omitting the sensors is considered as an advantage because the position sensors and speed sensors present disadvantages in terms of cost and environmental robustness.
SPM motor	Also called SPMSM. This type of motors are used for servo motors that require smooth motion even at low speeds.
Surface permanent magnet synchronous motor	See SPM motor.

3. Used hardware and software

3.1 List of used hardware

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

Table 3-1 List of used hardware devices

Classification	Product used
MCK-RX26T Renesas Flexible Motor Control Kit for RX26T MCU Group	Version.1: RTK0EMXE70S00020BJ Version.2: RTK0EMXE70S00021BJ
Microcontroller / CPU board P/N	RX26T RAM64KB Version (R5F526TFCDFP) / Version 1: RTK0EMXE70C00000BJ Version 2: RTK0EMXE70C00001BJ
	RX26T RAM48KB Version (R5F526TACDFM) / Version 1: RTK0EMXE30C00000BJ Version 2: RTK0EMXE30C00001BJ
Inverter board	Renesas MCI-LV-1 Inverter board for 48V 10A BLDC / RTK0EM0000B12020BJ
MC-COM	Renesas Flexible Motor Control Communication Board RTK0EMXC90Z00000BJ
IPM Motor	IS-94BZC (manufactured by Mabuchi Motor Co., Ltd.)
Torque bench (Load system)	HD-710-8N (manufactured by Magtrol)
Torque bench (Display)	DSP6001 (manufactured by Magtrol)
DC power supply	24V,11A (manufactured by COSEL)
Power meter	WT500 (manufactured by Yokogawa Test & Measurement)

3.2 List of used software

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

Table 3-2 List of used software(RX26T_MCBA_MCILV1_IPM_LESS_FOC_WHOLE_CSP_V100, RX26T_MCBA_MCILV1_IPM_LESS_FOC_WHOLE_E2S_V100)

IDE version	RX smart configurator	Toolchain version
CS+: V8.10.00	Version 2.18.0	CC-RX: V3.05.00
e^2 studio: 2023-07	e^2 studio plug-in version	

Table 3-3 List of used software(RX26T_MCBA2_MCILV1_IPM_LESS_FOC_WHOLE_CSP_V100, RX26T_MCBA2_MCILV1_IPM_LESS_FOC_WHOLE_E2S_V100)

IDE version	RX smart configurator	Toolchain version
CS+：V8.13.00	Version 2.25.0	CC-RX：V3.07.00
e^2 studio：2025-04	e^2 studio plug-in version	

Table 3-4 List of used software(RX26T_MCBC2_MCILV1_IPM_LESS_FOC_WHOLE_CSP_V100, RX26T_MCBC2_MCILV1_IPM_LESS_FOC_WHOLE_E2S_V100)

IDE version	RX smart configurator	Toolchain version
CS+ ：V8.15.00	Version 2.29.0	CC-RX：V3.07.00
e^2 studio ：2026-04.3	e^2 studio plug-in version	

4. Building a hardware environment

4.1 Hardware environment overview

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

In the sections that follow, the power supply (4.2), the motor and load system (4.3), the inverter (4.4), and the RX26T CPU board and its monitoring and programming mechanisms (4.5) are described in detail.

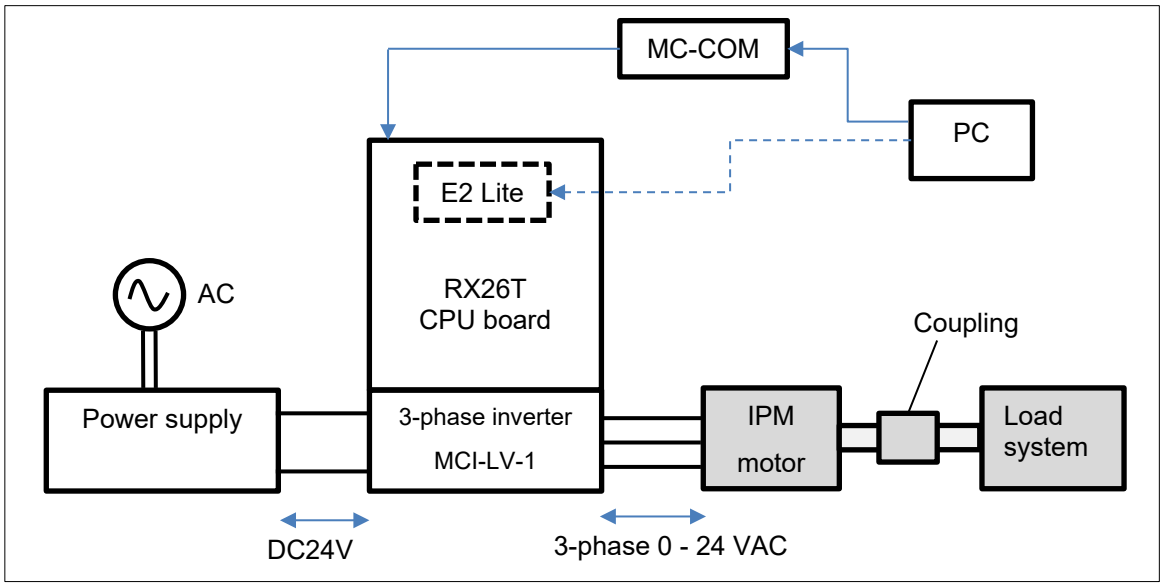


Figure 4-1 Example hardware configuration

4.2 Preparing the power supply

In this sample program, DC stabilized power supply/AC adapter/a control power supply (capable of output of 24V, 10A or more) is used to supply a voltage of 24 VDC from single-phase 100 VAC 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. Select an appropriate type of power supply based on your experimental environment and restrictions and conditions of the power supply to be used.

The inverter introduced here has an output current of 10 Amax, and the motor presented as a control target cannot be operated at the rated load. Please note.

4.3 Preparing the motor and load system

Before connecting the inverter to the motor, obtain the parameters and constants of the IPM motor (required to drive the motor with sensorless vector control) using an LCR meter or other measuring instrument. You may need to contact the manufacturer of the IPM motor to obtain the parameter information as required.

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

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

Table 4-1 shows the parameters of the IS-94BZC motor from Mabuchi Motor Co., Ltd., which we investigated. The parameters are based on our own measurements and may vary between individual motors and depending on the measurement conditions. The accuracy or performance of these parameters is not guaranteed. Note that the magnetic saturation caused by the load current may change the motor parameter values, thus affecting position estimation accuracy or operational performance.

When you perform a load test on an IPM motor, a load tester (load system) or an actual load system capable of performing a load test is required in addition to the IPM motor to verify the control performance. An appropriate load system must be selected based on the required load capacity of the motor and provided by the user. In the evaluation environment for this sample program, evaluation is performed by attaching the IPM motor shaft to a load testing system from MAGTROL. For your reference, Table 4-2 shows the parameter settings.

Table 4-1 IS-94BZC motor parameters (some of which are based on our own measurements)

Items	Value
Primary resistance R	0.045 Ω
d-axis inductance	0.0951mH
q-axis inductance	0.1253mH
Moment of inertia	0.0000294367kgm^2
Magnetic flux linkage Ψ	0.0088Wb (rms)
Inductive voltage Emf	14.2V
Number of poles	14 (7 pairs)
Rated speed	2400r/min
Rated frequency	280Hz (electrical angle), 40Hz (mechanical angle)
Rated current	12.3Arms

Table 4-2 Example settings of the DSP6001 motor load testing system

Items	Value
TSC1 MAXPOWER	0.36 kilowatts
TSC1	HD
TSC1 FILTER	10HZ
INPUT UNITS	N.m
MAX TORQUE	2.0Nm
ENCODER BITS	60BIT
SPEED ALARM	4000RPM

4.4 Preparing the inverter

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

In sensorless vector control, the magnetic pole position is estimated 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 circuit that serves as a path for the signals output from the sensor. When selecting an inverter, careful consideration must be given to the design of the inverter:

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

4.5 Setting up the RX26T CPU board

This section describes how to install the RX26T CPU board, which can be plugged in to MCI-LV-1. You can plug the RX26T CPU board to the top of the MCI-LV-1 board. A connector for MC-COM, connector for an external encoder are also provided.

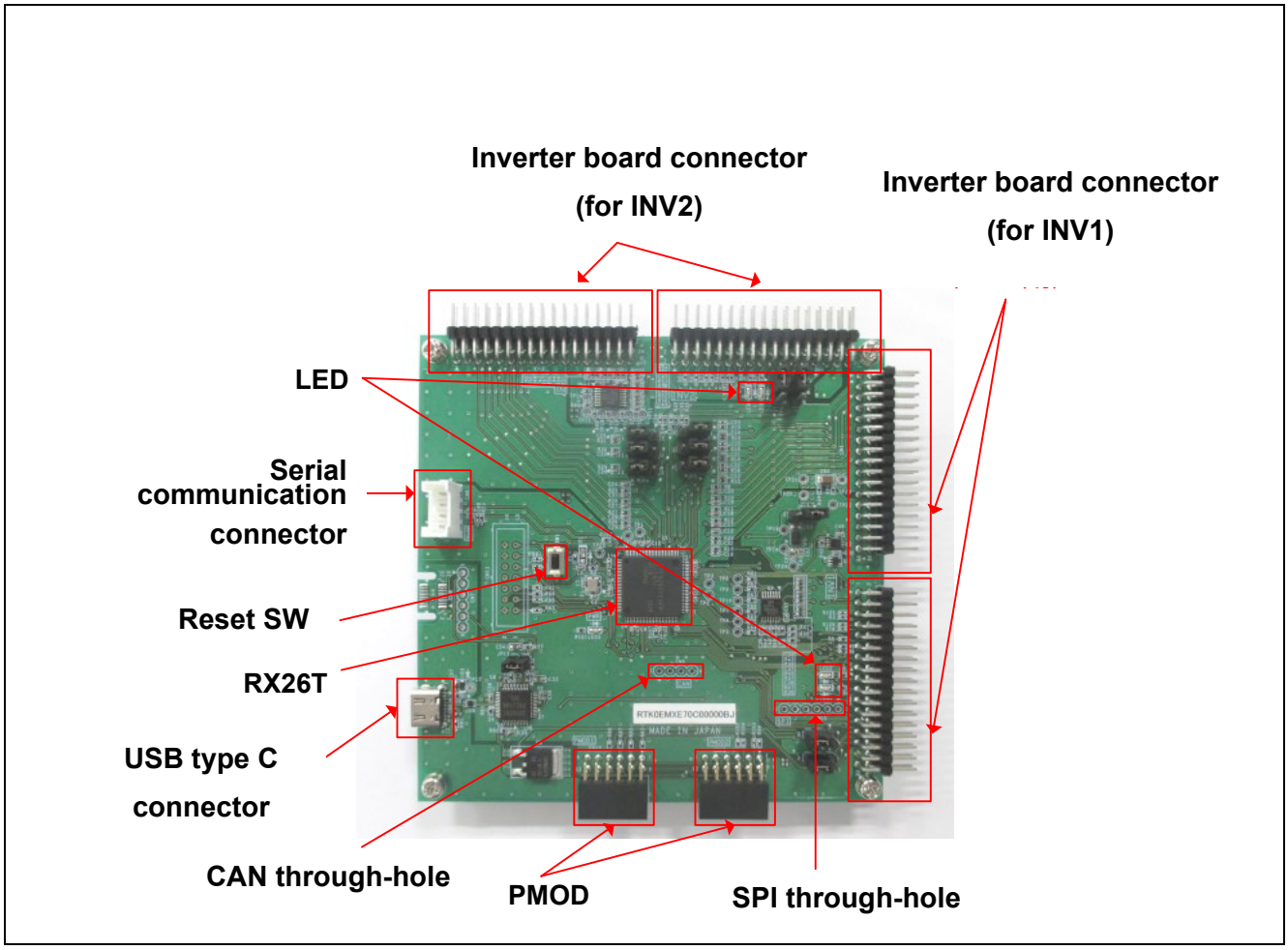


Figure 4-2 RX26T RAM64KB Version CPU board and its interface

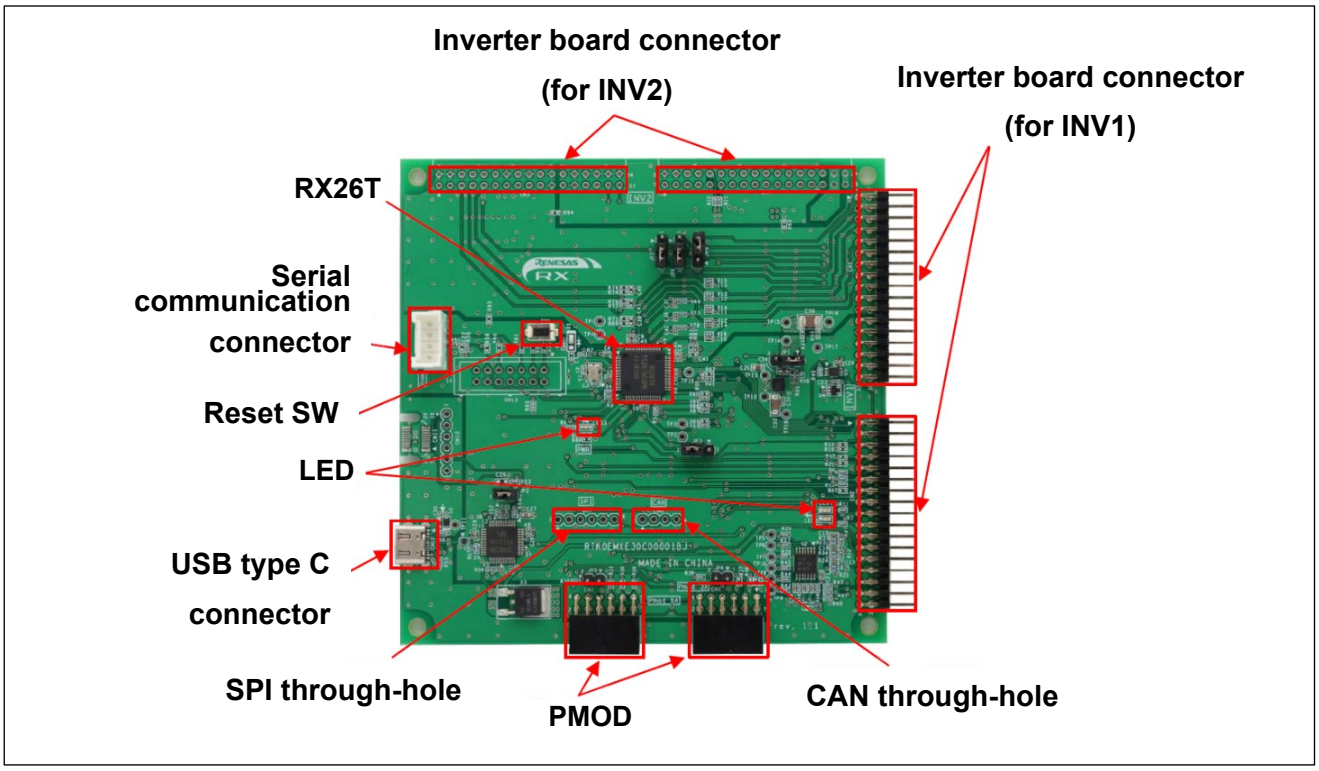


Figure 4-3 RX26T RAM48KB Version CPU board and its interface

4.6 Connection example of kit (MCK-RX26T)

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

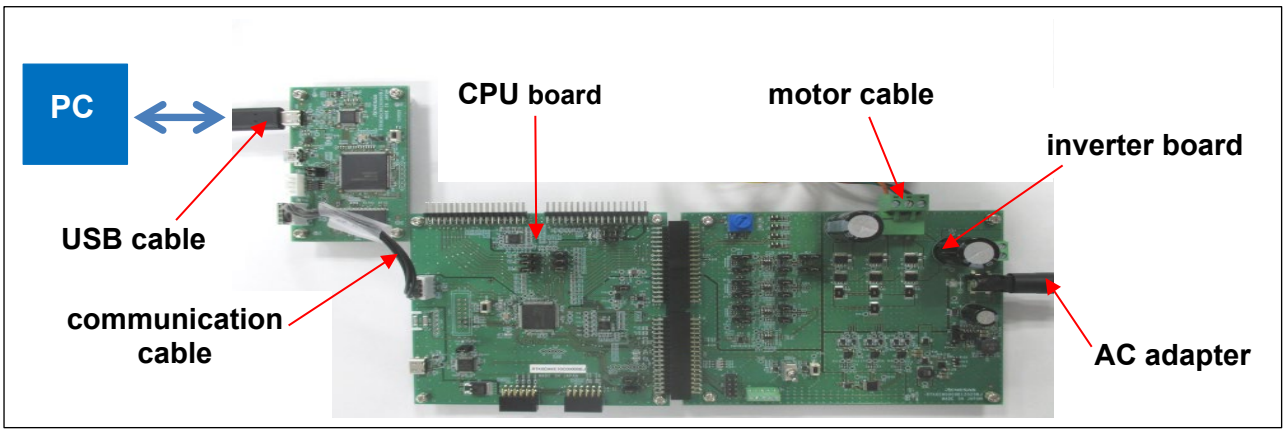


Figure 4-4 Connection example of kit (MCK-RX26T)

4.7 On-board debugger

This product is equipped with an onboard debugger circuit E2 On-Board (hereinafter referred to as E2OB), and the RX26T program is rewritten using E2OB. When rewriting the program, open jumper JP11(RX26T RAM64KB Version) or JP2(RX26T RAM48KB Version) and connect the CPU board and PC with a USB cable. E2OB acts as a debugger equivalent to the E2 emulator Lite. When connecting from an integrated development environment (e.g., e2 studio) or a flash programming tool (e.g. Renesas flash programmer), set the debugger (tool) type to "E2 emulator Lite" at the time of configuration.

After rewriting the program, short JP11 or JP2 to operate the CPU board.

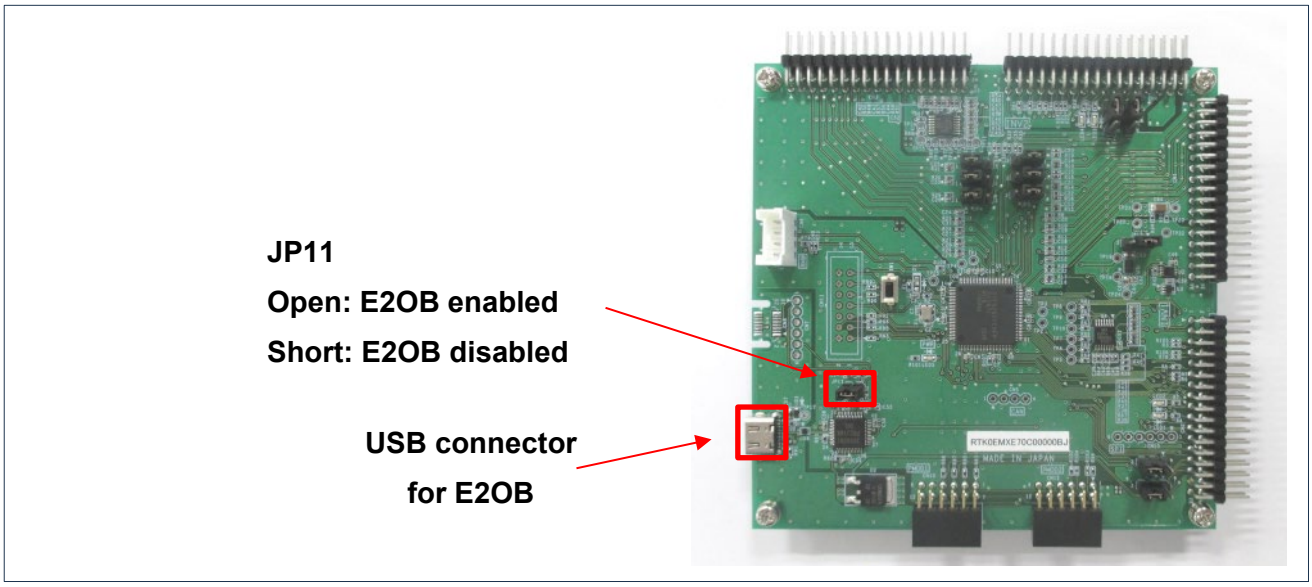


Figure 4-5 On board debugger(RX26T RAM64KB Version)

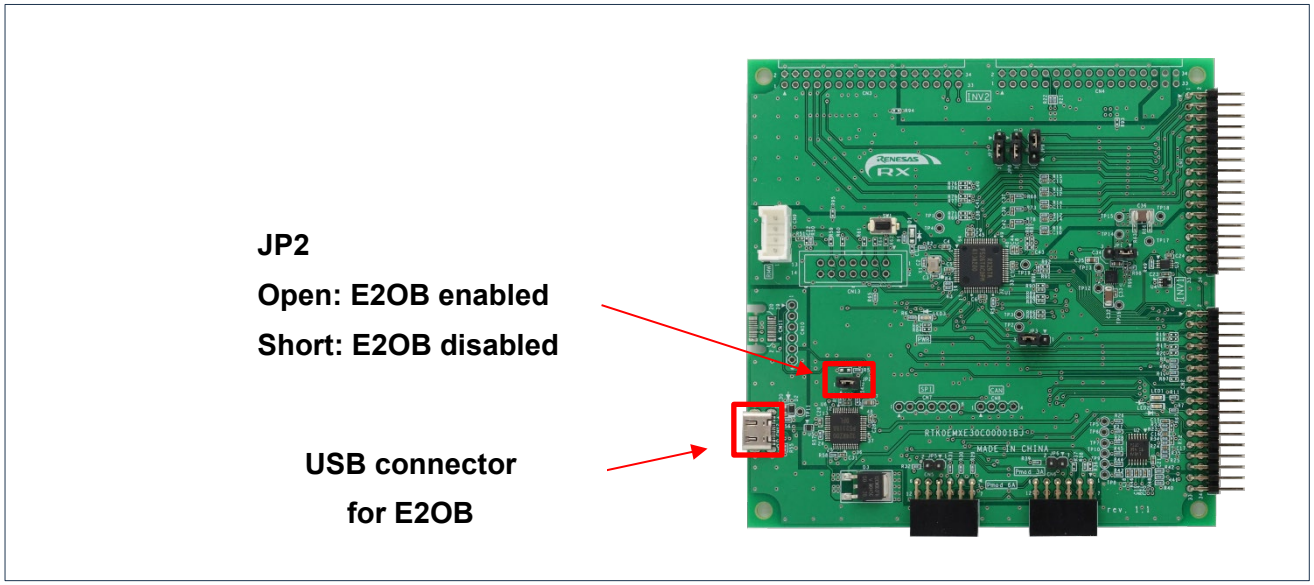


Figure 4-6 On board debugger(RX26T RAM48KB Version)

4.8 Wiring

This section describes how to wire 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 verify the contents and specifications before 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.

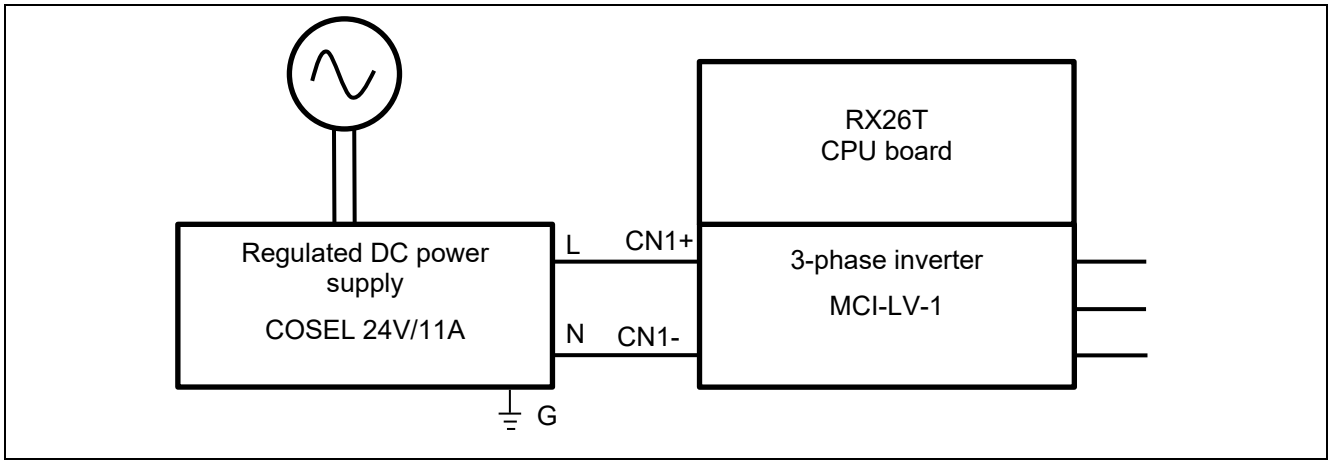


Figure 4-7 Wiring between the power supply and the inverter

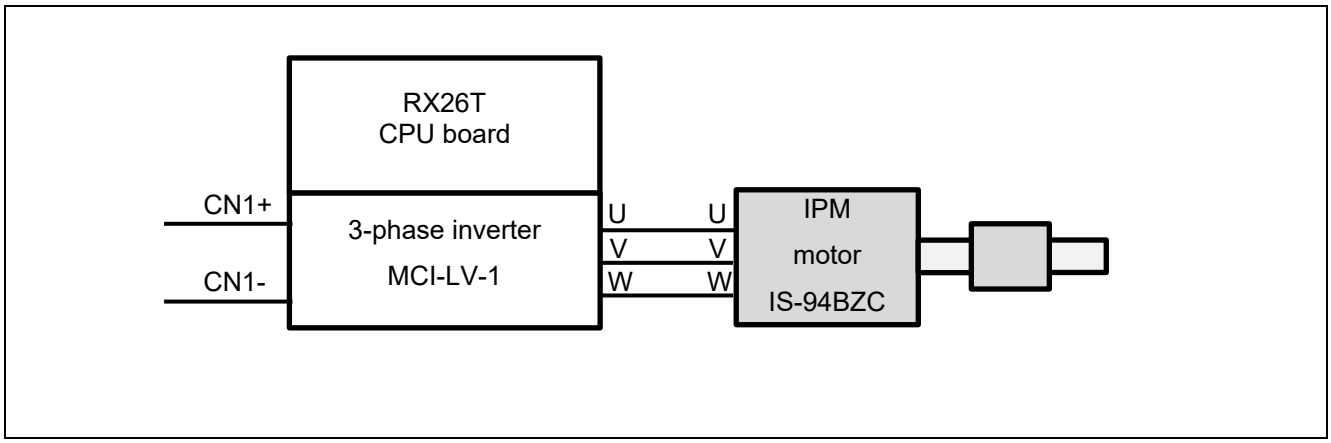


Figure 4-8 Wiring between the inverter and the motor

4.9 Using the measurement instruments

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

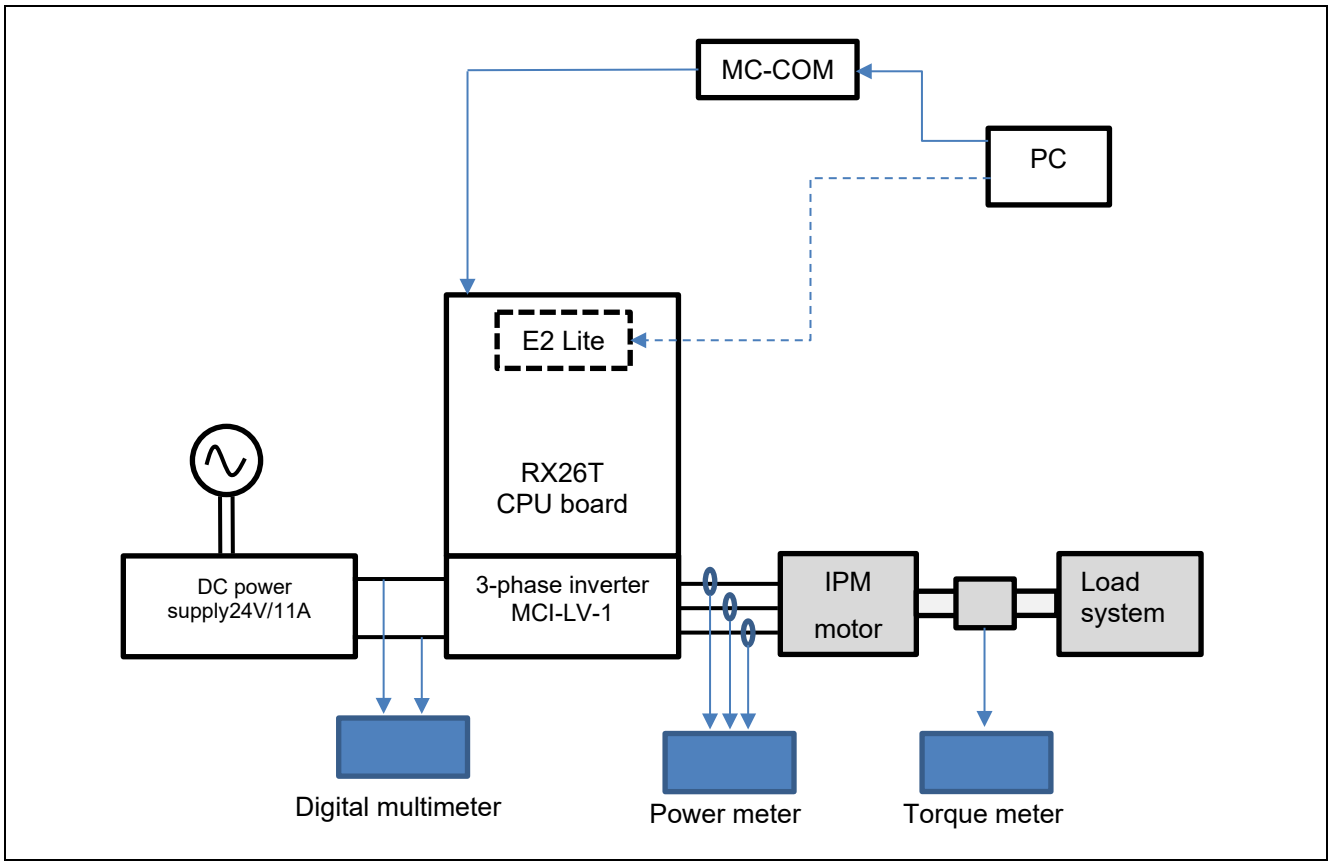


Figure 4-9 Example of adding measurement instruments

5. Building a software environment

5.1 Using CS+

Download CS+ from the following URL:

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

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

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

5.2 Using e^2 studio

Download e^2 studio from the following URL:

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

CC-RX and Smart Configurator must be installed separately. For installation instructions, refer to the PDF manual supplied with CS+ or the video “e² studio tutorial video for RX Family (1/4) - Installation” posted on the above URL.

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

6. Driving the motor

6.1 Precautions before driving the motor

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

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

6.2 Operation preparation procedure

Preparation procedures for operation are as follows:

Table 6-1 Operation preparation procedure

Step	Description	Applicable chapter
1	Insert the CPU board to the inverter in advance.	4.5
2	Install the sample program, development environment (CS+, e^2 studio, etc.), and emulator (E2OB, etc.) to the PC used.	5
3	Connect the PC to the CPU board via the in-circuit-emulator, and supply a 5 V power to the CPU board.	6.3
4	Build the sample program in the development environment.	6.4
5	Write the built sample program to the CPU board.	
6	Remove the cable connecting between the emulator and the CPU board.	6.3
7	Connect MC-COM to the CPU board.	6.3
8	Supply a 24 VDC power to the inverter.	4.2
9	Use RMW installed on the PC to connect to the CPU board via MC-COM and verify that it can be connected properly.	6.5
10	In RMW, verify that the variables of this sample program and sensor information are properly displayed.	6.7
11	Use RMW to operate the motor.	6.8
12	Stop and shut down the motor.	6.9

6.3 Connecting

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

(1) For writing
E2OB is used to write the sample program to the CPU board.

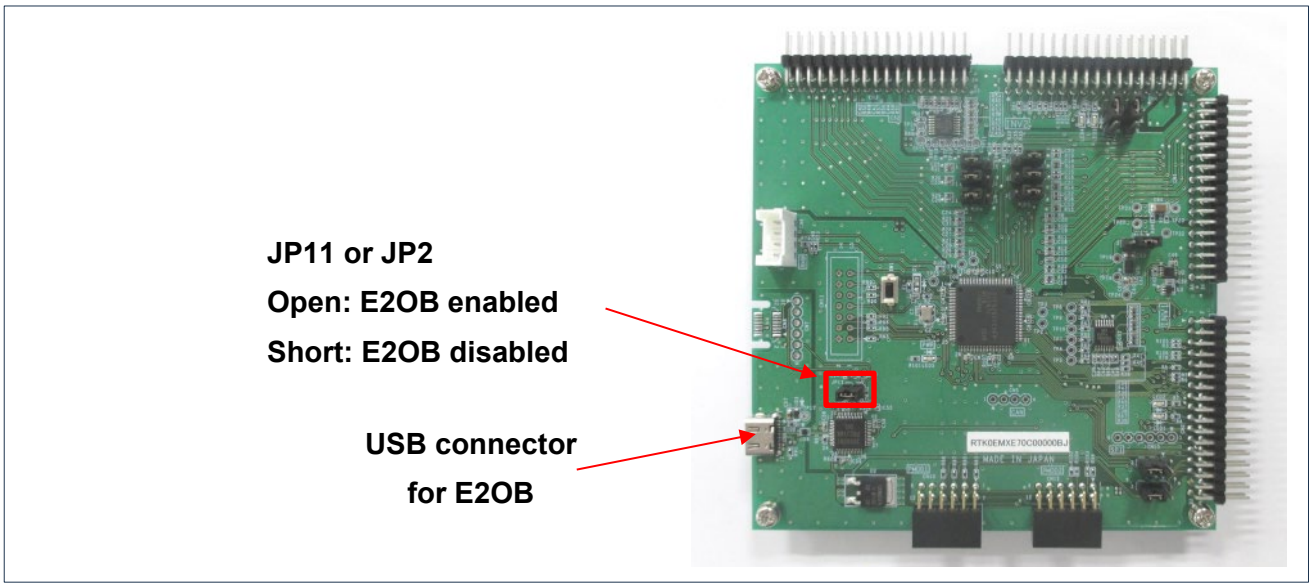


Figure 6-1 On board debugger

(2) For motor operation
Connect to the CPU board using MC-COM(RTK0EMXC90Z00000BJ) as shown in Figure 6-2. The CPU board is connected to the PC via UART and can be operated from the PC using a COM port. RMW can be used to operate the motor. MC-COM provides electrical isolation between the inverter and the PC and can be used safely even in high-voltage environments.

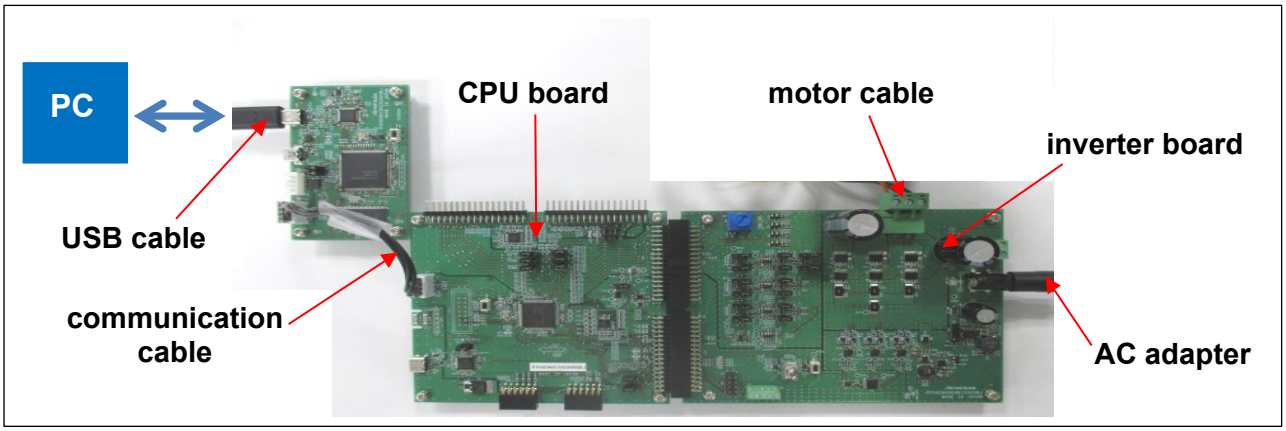


Figure 6-2 Example wiring for motor operation

6.4 Writing the sample program

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

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

6.5 Introducing RMW

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

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

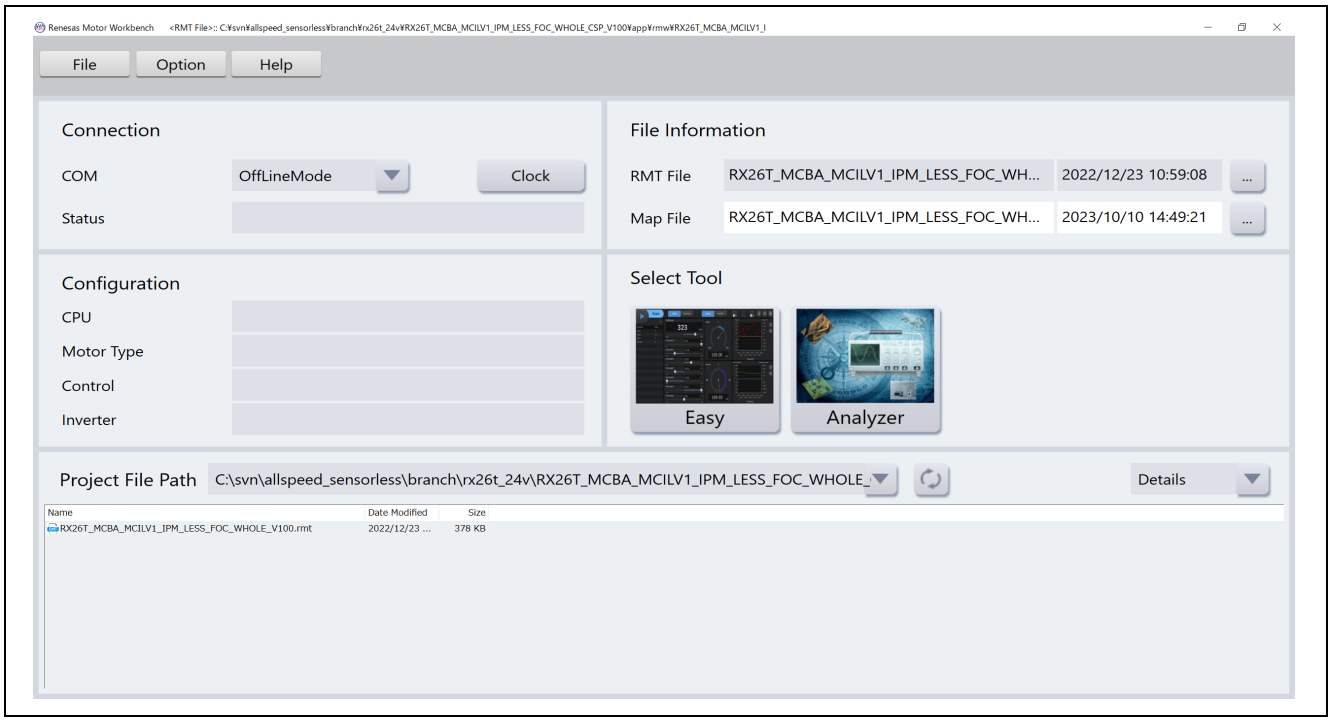


Figure 6-3 Window of Renesas Motor Workbench

6.6 Registering and updating the Map file

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

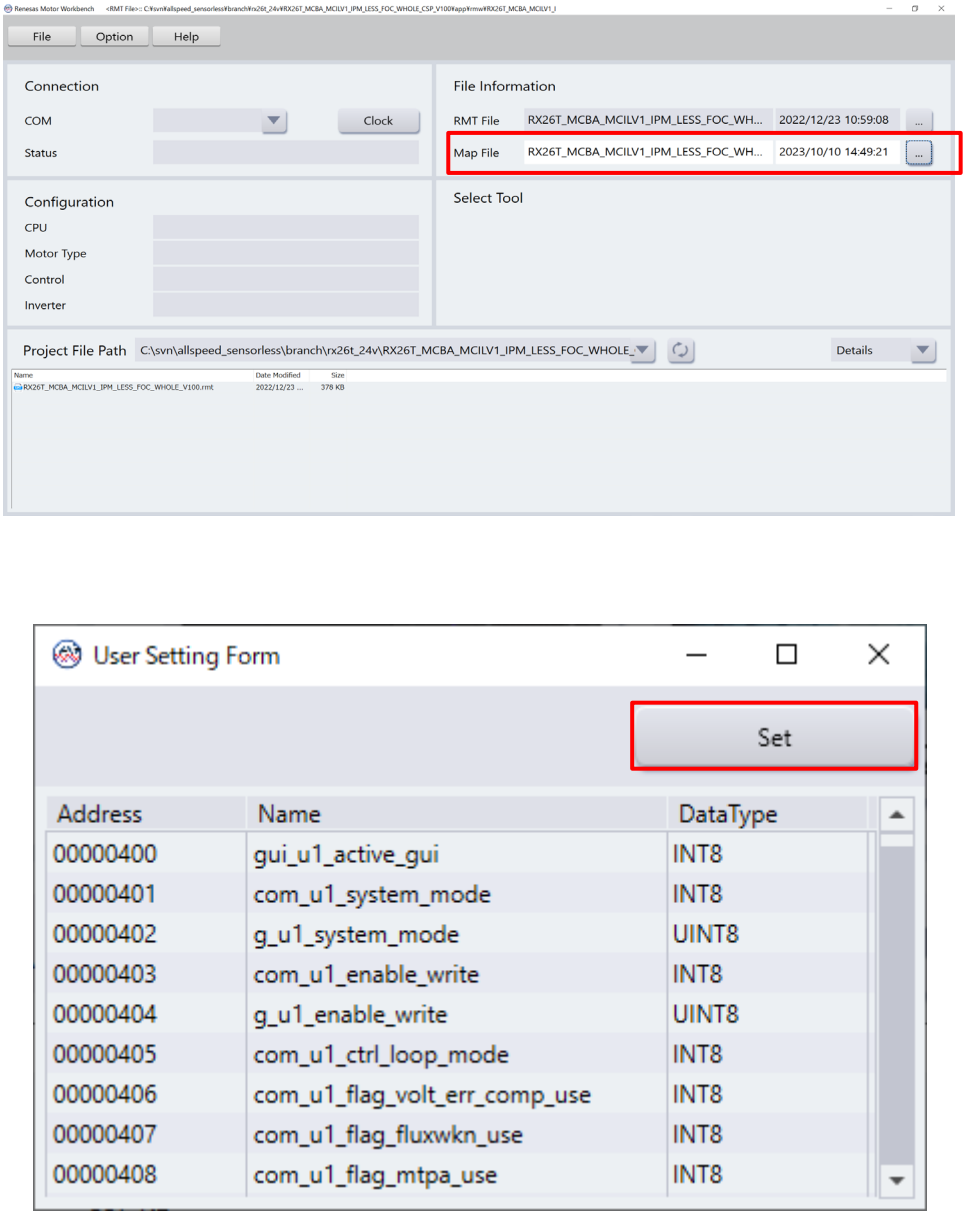


Figure 6-4 Map file registration setup window (upper) and confirmation window (lower) in RMW

6.7 Variables used for working with RMW

In this sample program, the motor is controlled using RMW. Table 6-2 lists the data input variables that are used when the RMW UI is used. The values input to these variables are applied to the corresponding variables in the motor module and then used for controlling the motor if the value written to the com_u1_enable_write variable is the same as the value of the and 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.

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

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

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

Analyzer function input variable name	Type	Description
com_u1_system_mode (*)	uint8_t	Managing the state 0: Stop mode 1: Run mode 3: Reset
com_f4_ref_speed_rpm (*)	float	Speed command value (mechanical angle) [r/min]
com_u1_enable_write	uint8_t	Whether to enable rewrite of variables for user entry The input data is applied if the values of this and g_u1_enable_write variables are the same.

Table 6-3 lists main structure variables that are often observed when the driving under speed control is evaluated. Use this table for reference when the waveform is to be displayed or the values of variables are to be loaded with an Analyzer function.

Table 6-3 List of main variables for sensorless speed control

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

6.8 Controlling the motor

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.

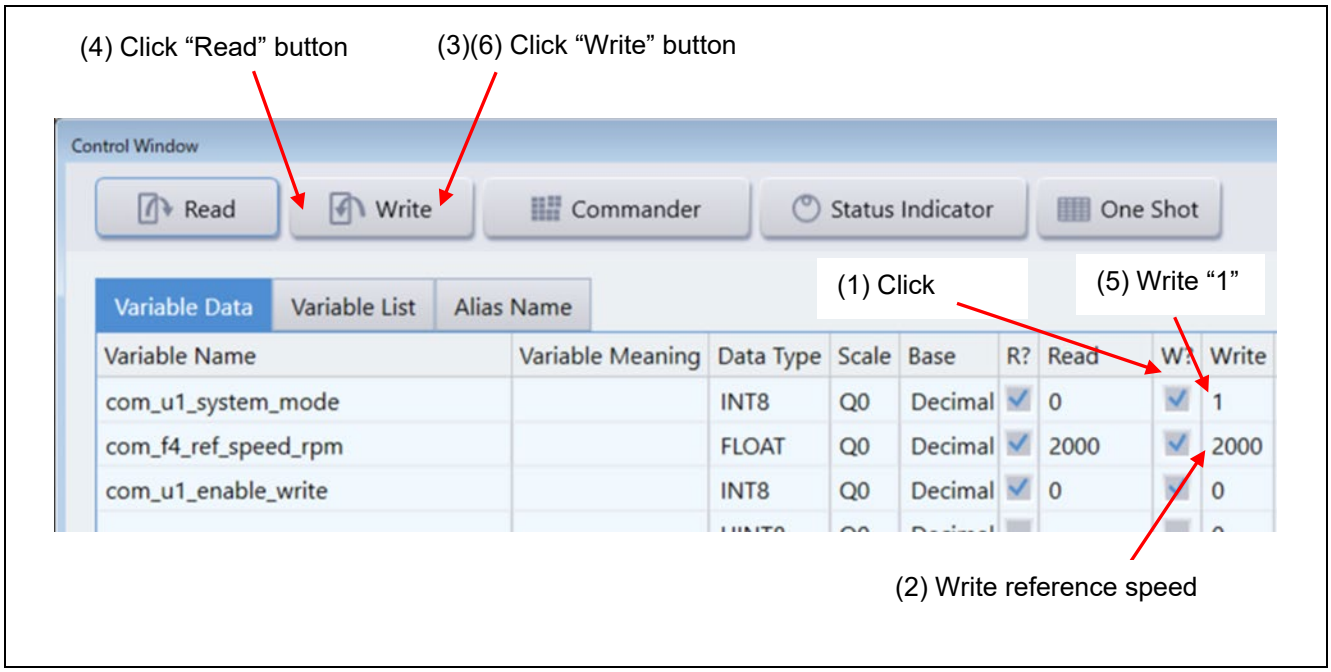


Figure 6-5 Procedure for starting driving of the motor

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.

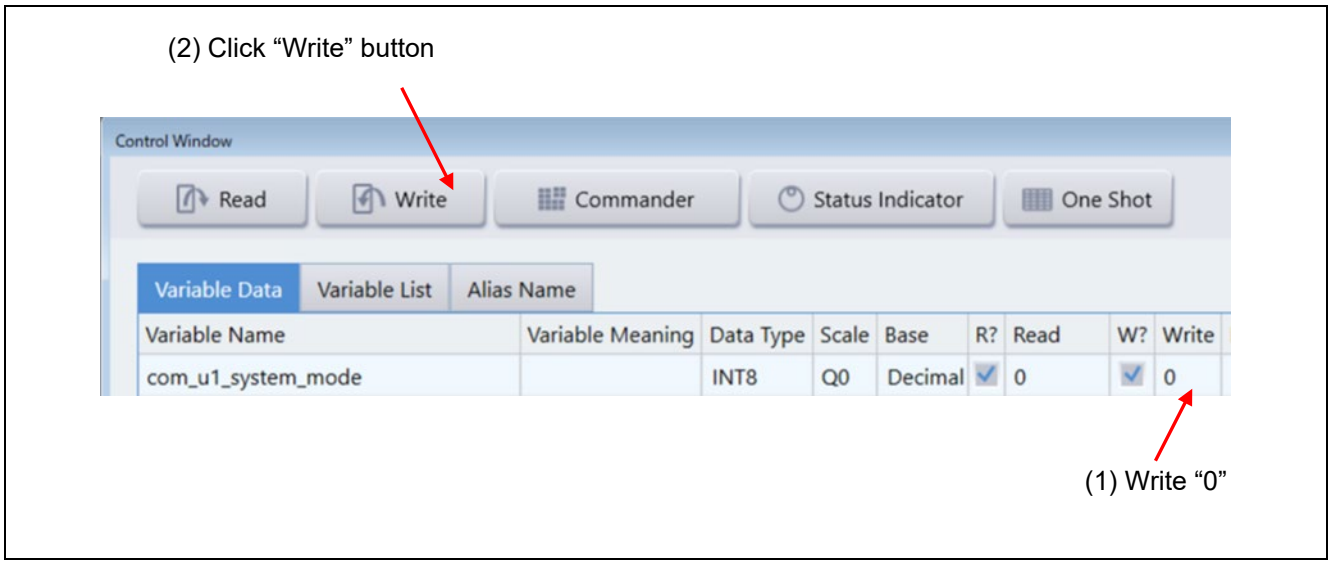


Figure 6-6 Procedure for stopping the motor

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.

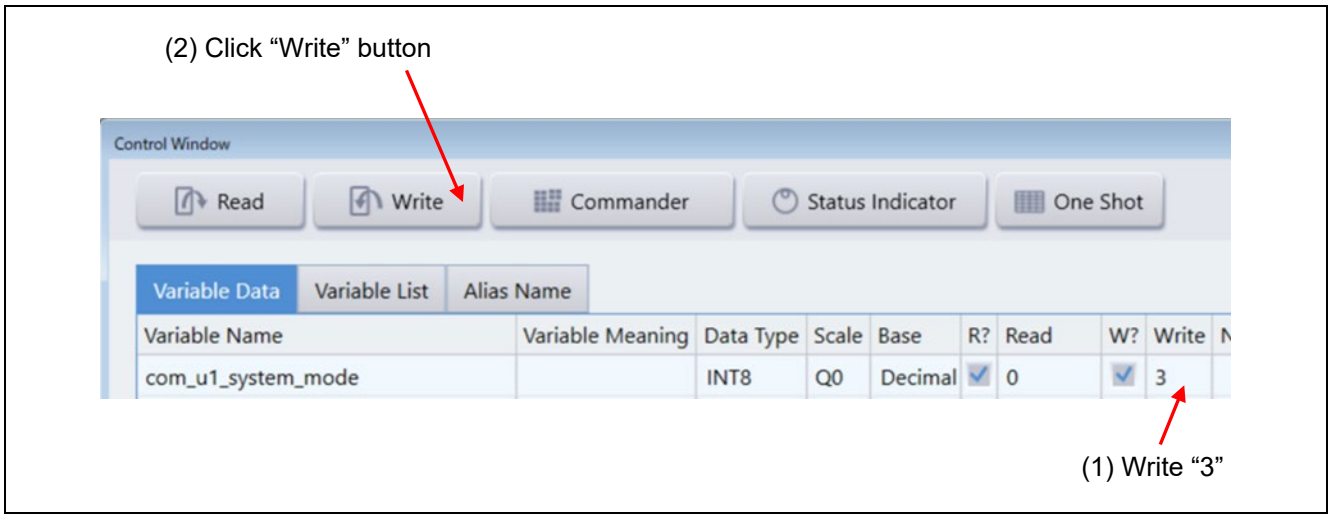


Figure 6-7 Procedure for handling an error

Table 6-4 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
0x0080	Low-voltage error	MOTOR_SENSORLESS_VECTOR_ERROR_LOW_VOLTAGE
0x0100	Software overcurrent error	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_CURRENT_SW
0x0200	Polarity determination error	MOTOR_SENSORLESS_VECTOR_ERROR_FAIL_POLES
0x0400	Magnetic pole position estimation error	MOTOR_SENSORLESS_VECTOR_ERROR_FAIL_POSITION
0xffff	Undefined error	MOTOR_SENSORLESS_VECTOR_ERROR_UNKNOWN

6.9 Stopping and shutting down the motor

To stop the operating motor, follow the procedure below. In an emergency, prioritize the step (C) 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 algorithm

7.1 Overview

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

Table 7-1 Motor control functions of this sample program

Function item	Function description
Control method	Sensorless vector control
Pulse Width Modulation (PWM) method	Space vector modulation method (Triangular-wave comparison method can also be selected)
Position and speed estimation method	Low speed range: High-Frequency pulse Injection(HFI) Middle to high speed range: BEMF observer
Control mode	Only speed control
Compensation function	• Maximum torque/current control (MTPA) • Voltage error compensation • Voltage phase lead compensation • Decoupling control • Disturbance torque/speed estimation observer

7.2 Control block diagram

Sensorless vector control is performed over the whole speed range. The following shows an example block diagram for the overall control system.

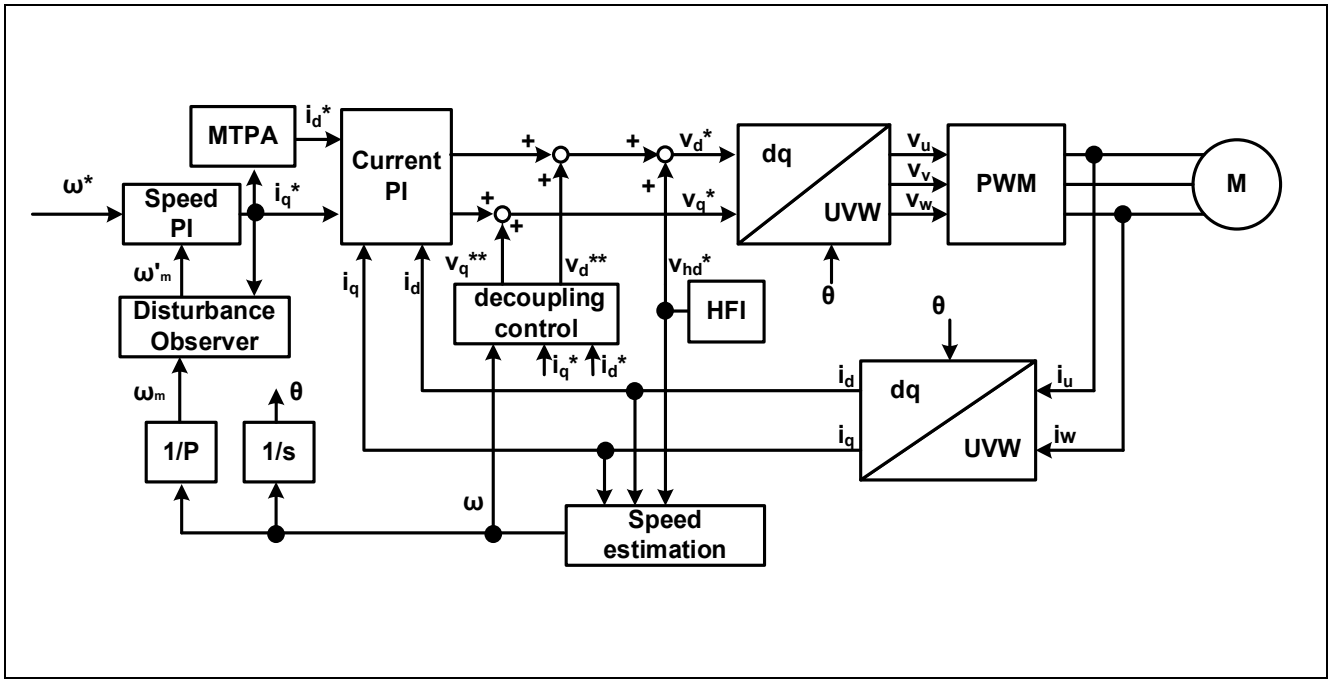


Figure 7-1 System block of vector control

7.3 Speed control function

The speed control function performs PI control so that the motor follows the speed command. When receiving a speed command value, the internal speed regulator outputs a current command value based on the deviation from the estimated speed value. This function also controls the submodules including the disturbance torque/speed estimation observer and maximum torque/current control (MTPA).

The estimated speed value is selected either from the value output from the disturbance torque/speed estimation observer or from the estimated speed passed though the LPF.

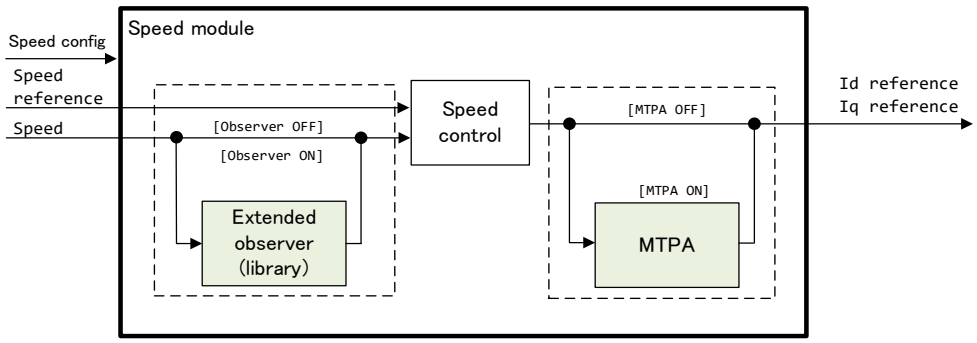


Figure 7-2 Functional block diagram for speed control

7.4 Disturbance torque/speed estimation observer

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

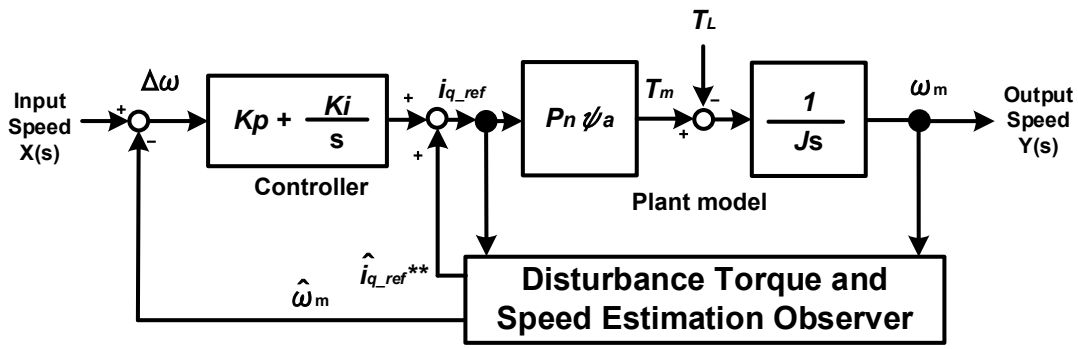


Figure 7-3 Block diagram for the disturbance torque/speed estimation observer

7.5 Maximum torque/current control

An IPM motor having saliency can adjust the torque per unit current to the maximum by applying maximum torque/current control (MTPA). 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]

7.6 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. The module also controls submodules including the decoupling, lead compensation, voltage error compensation, HFI, angle estimation, and phase adjustment. Figure 7-4 shows the module configuration.

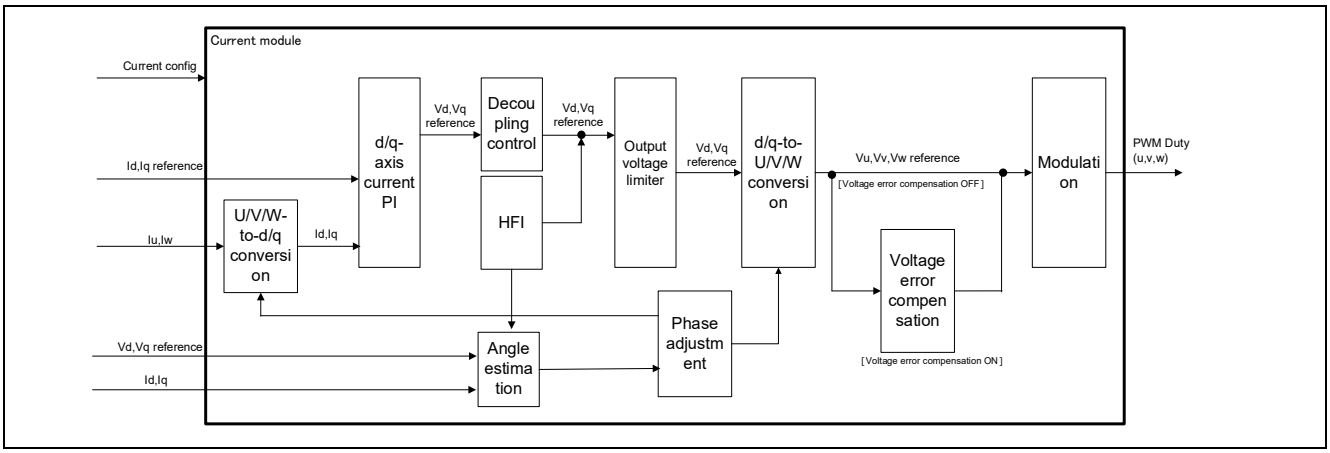


Figure 7-4 Functional block diagram for current control

7.7 Decoupling control

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

$$V_{d_dec}{}^{*} = RI_d{}^{*} - \omega L_q I_q{}^{*}$$
$$V_{q_dec}{}^{*} = RI_q{}^{*} + \omega L_d I_d{}^{*} + \omega\Psi$$

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

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

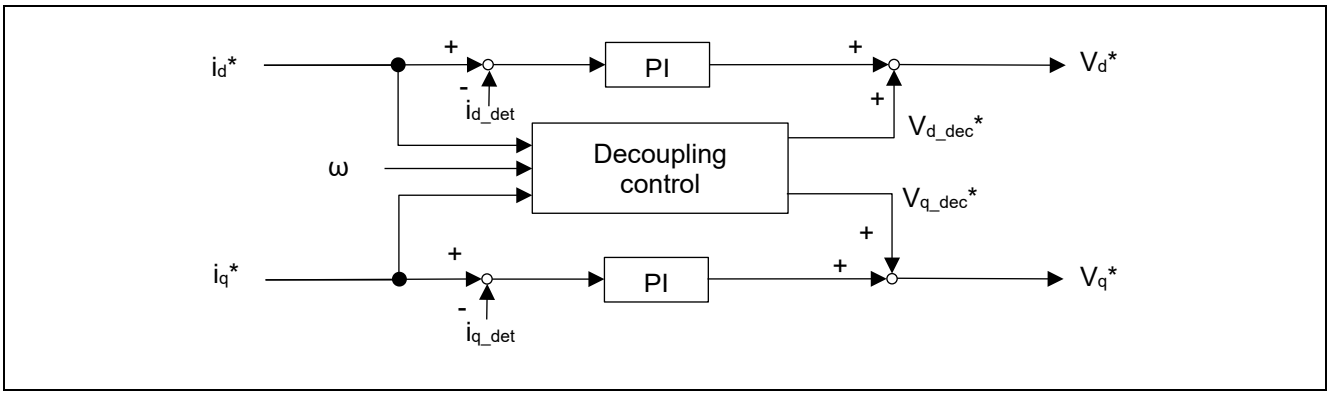


Figure 7-5 Functional block diagram for decoupling

7.8 Sensorless function

7.8.1 Overview

Sensorless full closed loop control is achieved over the whole speed range including the zero speed, by combining the magnetic pole position estimation by High-Frequency pulse voltage Injection (HFI) utilizing saliency from the zero speed to low speed range (300 r/min or less) and magnetic pole position estimation by using an BEMF observer in the middle to high speed range (300 r/min or more).

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

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

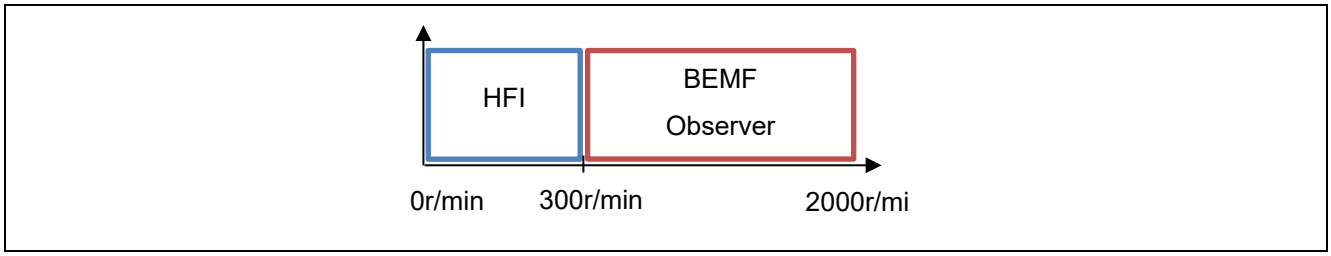


Figure 7-6 Sensorless control algorithm corresponding to the speed range

7.8.2 Low-speed-range sensorless algorithm (HFI)

a) Overview

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

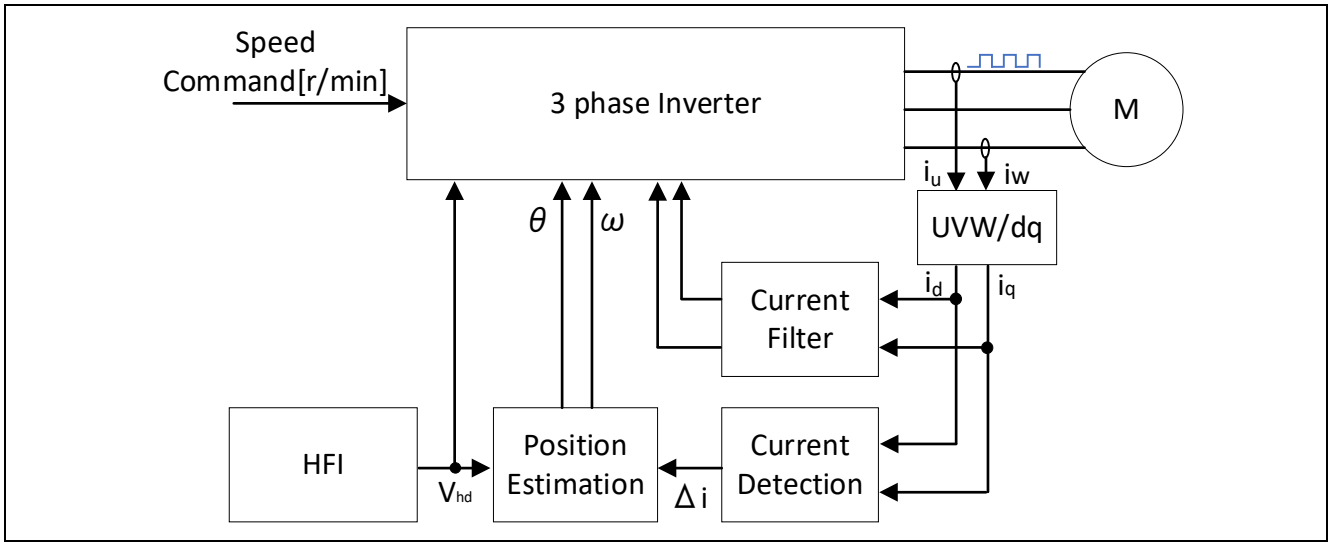


Figure 7-7 Overview of the low-speed-range sensorless algorithm

b) High-frequency pulses and response current

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

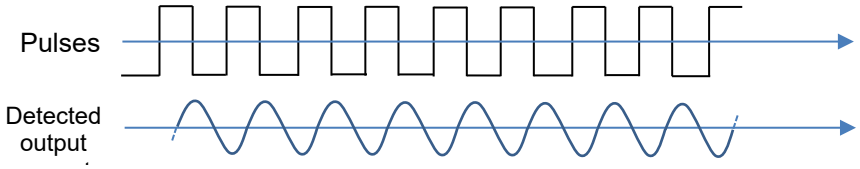


Figure 7-8 Example of pulses and response current

c) Estimating the angle

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

As shown in Figure 7-9, the reference axis for angle estimation is defined as the dc-qc axis. Control is performed by finding Δθ so that the dc-qc axis is aligned with the dq axis.

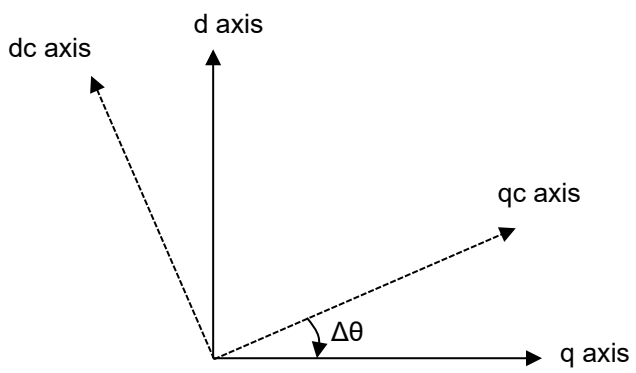


Figure 7-9 Definition of Δθ

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

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

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

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

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

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

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

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

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

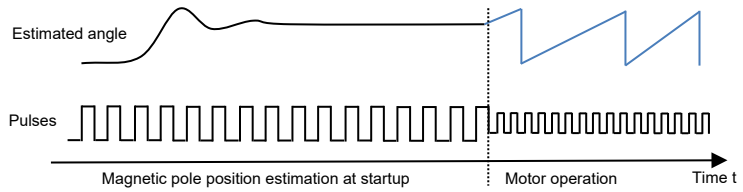


Figure 7-10 Example of the estimated angles and high-frequency pulse injection

d) Polarity determination at startup

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

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

By causing a magnetic saturation phenomenon, there is a slight difference in the direction of the responding pulse voltage and the magnitude of the response current, depending on whether the magnetic pole position faces the N or S pole. When the integrated value of this difference is negative, it can be determined as the N-pole, and when it is positive, it can be determined as the S-pole.

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

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

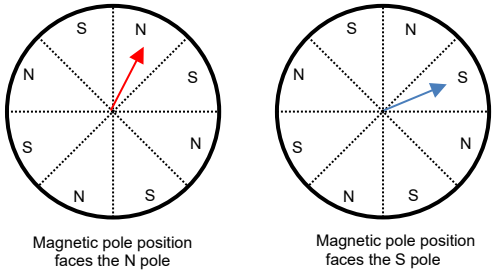


Figure 7-11 Magnetic pole positions and N and S poles in an 8-pole IPM motor

e) Magnetic pole position estimation at startup

Figure 7-12 shows the magnetic pole position estimation operation at startup. At startup, a high-frequency pulse voltage is applied to the d-axis voltage command Vd*. The high-frequency pulse voltage must be set to a value higher than the voltage applied during normal operation so that polarity determination as described above can be performed at the same time during startup. The appropriate pulse voltage value varies depends on the motor.

After applying a high-frequency pulse voltage, wait for a position convergence period of 200 ms, and then determine the convergence of the magnetic pole position with a position estimation convergence determination period of up to 100 ms. The convergence determination condition is when the difference between the previous angle value and the current angle value is confirmed to be within 1 degree for 10 consecutive times. If the position estimation convergence determination formula is not satisfied within 100 ms, it is defined as a magnetic pole position estimation error.

In addition, a polarity determination period is provided in parallel with the position estimation convergence determination period. If the absolute value of the PF value, which is the integral of the difference in response currents obtained by the polarity determination algorithm, is less than 10.0f, polarity determination is considered unsuccessful and is defined as a polarity determination error.

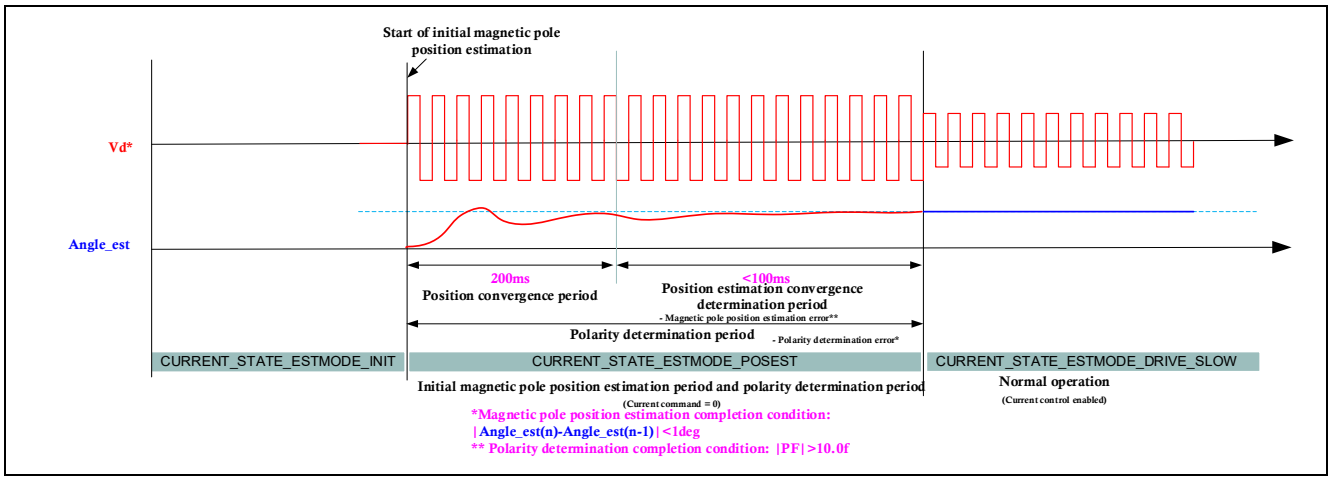


Figure 7-12 Magnetic pole position estimation operation at startup

f) Switching to and from the middle-high speed range

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

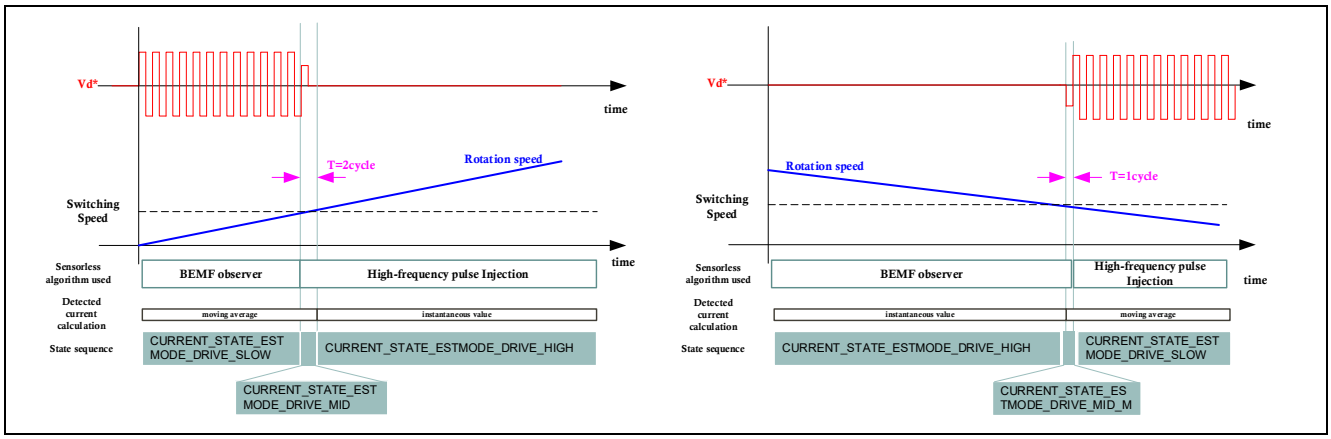


Figure 7-13 Sensorless algorithm switching for acceleration and deceleration

7.8.3 Middle-to-high-speed-range sensorless algorithm

In the middle to high speed range, the motor is controlled by sensorless vector control using an BEMF observer. The algorithm for the BEMF observer is described in detail in “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.

7.9 Voltage phase lead compensation

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

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

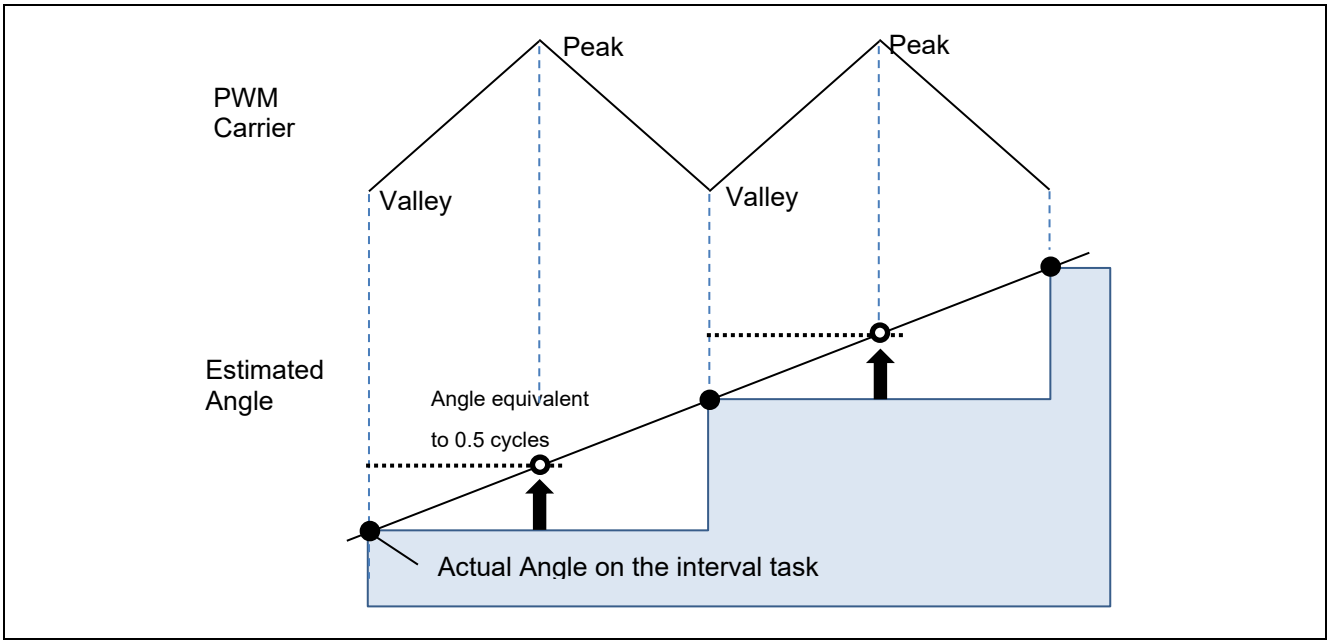


Figure 7-14 Example of the amount of angle advanced in a PWM carrier cycle

7.10 Voltage error compensation

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

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

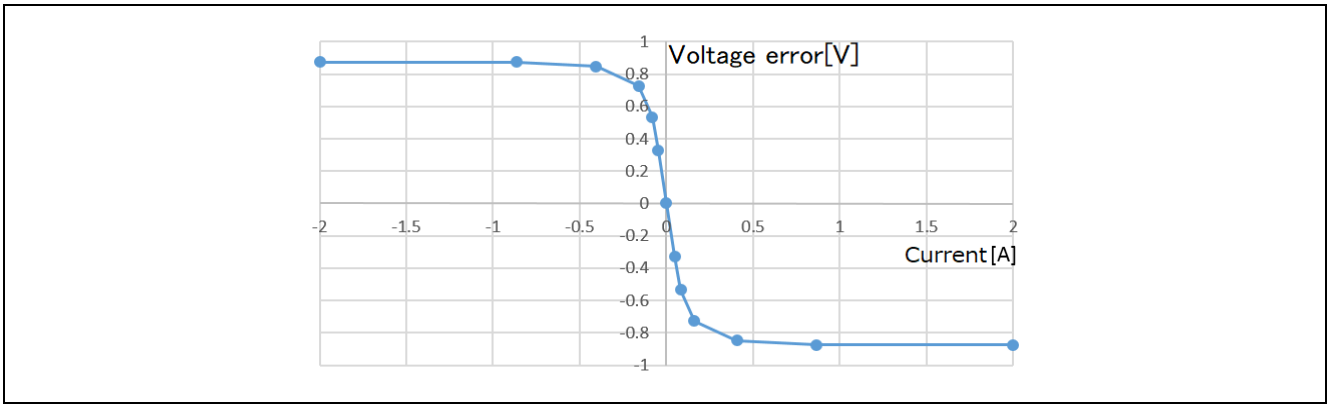


Figure 7-15 Current dependency in the voltage error (example)

7.11 Pulse Width Modulation (PWM) mode

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

a) Sinusoidal modulation (MOD_METHOD_SPWM)
The modulation rate m is defined as follows:

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

m: Modulation rate, V: Command value voltage,
E: Inverter bus voltage

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

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

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

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

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

The modulation rate m is defined as follows:

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

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

8. Software specifications and configuration

8.1 Software specifications

The following shows the basic software specifications of this system.

Table 8-1 Basic specifications of whole-speed-range sensorless vector control software

Item	Description	
Control method	Position sensorless vector control	
Starting/stopping motor control	Determined by input from RMW	
Rotor magnetic pole position detection	Sensorless (High-frequency pulse injection and BEMF observer)	
Input voltage	DC 24V	
PWM carrier frequency	20 [kHz], Carrier cycle: 50 [μs] (Interrupt at trough)	
Pulse Width Modulation (PWM) mode	Triangular wave comparison mode or space vector modulation mode	
Dead time	2.0 [μs]	
Control cycle (current)	50 [μs]	
Control cycle (speed)	500 [μs]	
Speed command value management	CW: 0 to 2000 [r/min] CCW: 0 to -2000 [r/min]	
Natural frequency for each control system	For the current control system: 600 Hz For the speed control system: 10 Hz Disturbance torque/speed estimation observer: 30Hz BEMF observer: 1000Hz Position estimation PLL (for standstill and low speed): 50 Hz Position estimation PLL (for middle to high speed): 20Hz	
Compiler optimization settings	Optimization level	2 (-optimize = 2) (default)
	Optimization method	Optimization focusing on the code size (-size) (default)
Protection stop processing	The motor control signal output (six outputs) will be deactivated when any of the following conditions are met: 1. The current peak values of all phases exceed 10 [A] (checked at intervals of 50 [μs]). 2. The current exceed calculated value (21.4A) from rated current (checked at intervals of 50 [μs]). 3. The inverter bus voltage exceeds 60 [V] (checked at intervals of 50 [μs]). 4. The inverter bus voltage is less than 8 [V] (checked at intervals of 50 [μs]). 5. The rotation speed exceeds 2850 [r/min] (checked at intervals of 50 [μs]). 6. During magnetic pole position estimation, the angle fluctuation (absolute value of the difference from the previous value) does not converge to within 1 degree for 10 consecutive times in a 100 msec period (monitored in the current control cycle). 7. During polarity determination for magnetic pole position estimation, the absolute value of PF value does not reach 10.0 or more in a 100 msec period (monitored in the current control cycle). When the overcurrent detection signal (POE/POEG) from an external circuit or an output short-circuit is detected, the PWM output pin is driven to Inactive.	

8.2 Overall configuration of the software

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

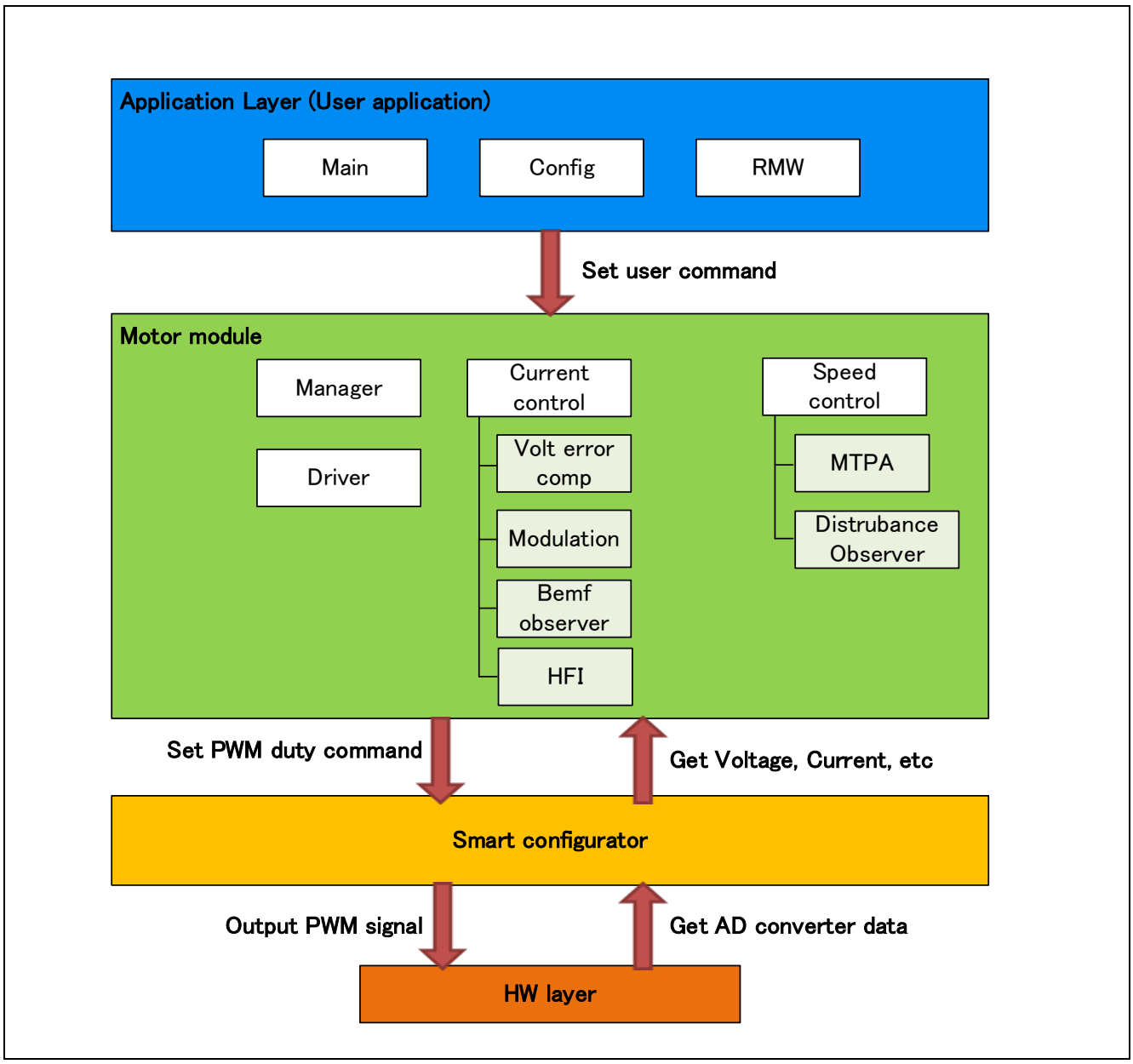


Figure 8-1 Overall configuration of the motor control software

8.3 Task description

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

Table 8-2 Interrupts and tasks used

Task	Peripheral function	Interrupt function	Function called
Motor control interrupt (for speed control)	CMT0	r_Config_CMT0_cmi0_interrupt	R_MOTOR_SENSORLESS_ VECTOR_SpeedInterrupt
Motor control interrupt (for current control)	MTU3,4	r_Config_MOTOR_ad_interrupt	R_MOTOR_SENSORLESS_ VECTOR_CurrentInterrupt
Idle	-	r_app_main_ui_mainloop	R_MOTOR_SENSORLESS_ VECTOR_StatusGet
At startup 1	-	r_app_main_init_motor_ctrl	R_MOTOR_SENSORLESS_ VECTOR_Open
At startup 2	-	r_app_main_start_motor_ctrl	(Driver-related only)
At reset	-	* Performed in the state transition process when recovering from an error.	R_MOTOR_SENSORLESS_ VECTOR_Reset
RMW operation	-	r_app_rmw_system_mode	R_MOTOR_SENSORLESS_VECTOR_ MotorStart MotorStop LoopModeStatusGet SpeedSet MotorReset

8.4 File and folder configuration

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

Table 8-3 File and folder configuration

Folder	Subfolder	File	Remarks
app	main	r_app_main.c/h	User main function
	rmw	r_app_rmw.c/h	Definition of functions related to the RMW Analyzer UI
		r_app_rmw_interrupt.c	Definition of RMW interrupt functions
		ICS2_RX26T.lib/h	Library for RMW communication
	cfg	r_app_control_cfg.h	Configuration definition for the application layer
motor_module	sensorless_ vector_rx	r_motor_sensorless_vector_action.c	Definition of action functions
		r_motor_sensorless_vector_api.c/h	Definition of API functions for the manager module
		r_motor_sensorless_vector_manager.c/h	Definition of local functions for the manager module
		r_motor_sensorless_vector_protection.c/h	Definition of functions for the protection function
		r_motor_sensorless_vector_ statemachine.c/h	Definition of functions related to state transition
	current_rx	r_motor_current_api.c/h	Definition of API functions for the current control module
		r_motor_current.c/h	Definition of local functions for the current control module
		r_motor_current_modulation.c/h	Definition of functions for the modulation module
		r_motor_current_volt_err_comp.c/h	Definition of functions for the voltage error compensation module
		r_motor_current_bemf_observer.lib/h	Definition of functions for the BEMF observer
		r_motor_current_pi_gain_calc.c	Definition of functions for calculating the control gain of the current control module
		r_motor_current_lowspd_sensorless.lib/h	Definition of functions for the low-speed-range sensorless control module
	speed_rx	r_motor_speed_api.c/h	Definition of API functions for the speed control module
		r_motor_speed.c/h	Definition of local functions for the speed control module
		r_motor_speed_extobserver.lib/h	Definition of functions for the disturbance torque/speed estimation observer
		r_motor_speed_pi_gain_calc.c	Definition of functions for calculating the control gain of the speed control module
	driver_rx	r_motor_driver.c/h	Definition of functions for the driver module
	general	r_motor_filter.c/h	Definition of general-purpose filter functions
		r_motor_pi_control.c/h	Definition functions for PI control
		r_motor_common.h	Common definition

Folder	Subfolder	File	Remarks
	cfg	r_motor_inverter_cfg.h	Configuration definition for the inverter
		r_motor_module_cfg.h	Configuration definition for the control module
		r_motor_targetmotor_cfg.h	Configuration definition for the motor
src	smc_gen	See Table 8-4.	Drivers and API functions generated by Smart Configurator

Smart Configurator can be used to generate peripheral drivers easily.

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

RX26T_MCBA_MCILV1_IPM_LESS_FOC_WHOLE_xxx_Vyyy.scfg
RX26T_MCBA2_MCILV1_IPM_LESS_FOC_WHOLE_xxx_Vyyy.scfg
RX26T_MCBC2_MCILV1_IPM_LESS_FOC_WHOLE_xxx_Vyyy.scfg

(In the above file name, the "xxx" portion indicates the edition: CSP indicates the CS+ edition and E2S indicates the e^2 studio edition. The "yyy" portion indicates the revision number.)

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

Table 8-4 Smart Configurator folder and file configurations

Folder	Subfolder	2nd subfolder	File	Remarks
src	smc_gen	Config_S12AD0	Config_S12AD0.c/h	Definition of functions related to 12-bit ADC
			Config_S12AD0_user.c	Definition of user functions related to 12-bit ADC
		Config_PORT	Config_PORT.c/h	Definition of functions related to ports
			Config_PORT_user.c	Definition of user functions related to ports
		Config_CMT0	Config_CMT0.c/h	Definition of functions related to CMT for the control interval
			Config_CMT0_user.c	Definition of user functions related to CMT for the control interval
		Config_MOTOR	Config_MOTOR.c/h	Definition of functions related to the Motor component
			Config_MOTOR_user.c	Definition of user functions related to the Motor component
		Config_IWDT	Config_IWDT.c/h	Definition of functions related to IWDT
			Config_IWDT_user.c	Definition of user functions related to IWDT
		Config_POE	Config_POE.c/h	Definition of functions related to POE
			Config_POE_user.c	Definition of user functions related to POE

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

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

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

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

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

8.5 Application layer

The application layer is used for selecting the user interface (UI), setting command values for controlling motor modules that use RMW, and updating parameters for control modules. In this sample program, RMW (RMW UI) is used for setting and processing. These UIs are also used to control whether to drive or stop the motor and to set control command values.

8.5.1 Functions

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

Table 8-5 Functions available in the application layer

Functions	Description
Main processing	Enables or disables each user command in the system.
UI processing	Manages RMW.
Manager processing	Obtains and sets command values for speed control.
RMW UI processing	Acquires and sets parameters (including command values).

8.5.2 Configurations

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

Table 8-6 List of configurations

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

Table 8-7 List of initial values for configurations

Macro name	Set value	
	RX26T RAM64KB Version	RX26T RAM48KB Version
APP_CFG_USE_UI	MAIN_UI_RMW	
APP_CFG_FREQ_BAND_LIMIT	3.0f	
APP_CFG_MAX_CURRENT_OMEGA	1000.0f	
APP_CFG_MIN_OMEGA	1.0f	
APP_CFG_SCI_CH_SELECT	0x60	0x10

8.5.3 Structure and variable information

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

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

Table 8-8 List of variables

Variable	Description
g_st_rmw_input_buffer	Structure for updating the RMW variables
g_u1_update_param_flag	Buffer transfer completion flag
com_u1_system_mode	Variable to switch the system mode for user entry 0: Stopping the motor 1: Driving the motor 3: Canceling the error
g_u1_system_mode	System mode 0: Stopping the motor 1: Driving the motor 2: Error
com_u1_enable_write	Whether to enable rewrite of variables for user entry
g_u1_enable_write	Whether to enable rewrite of variables
com_u2_offset_calc_time	Current offset value calculation time setting
com_u2_mtr_pp	Number of pole pairs of the motor to be driven
com_f4_mtr_r	Resistance of the motor to be driven [Ω]
com_f4_mtr_ld	d-axis inductance of the motor to be driven [H]
com_f4_mtr_lq	q-axis inductance of the motor to be driven [H]
com_f4_mtr_m	Magnetic flux of the motor to be driven [Wb]
com_f4_mtr_j	Rotor inertia of the motor to be driven [kgm^2]
com_f4_nominal_current_rms	Rated current of the motor to be driven [Arms]
com_f4_max_speed_rpm	Maximum speed (mechanical angle) of the motor to be driven [r/min]
com_u1_ctrl_loop_mode	Switching of the control loop 1: Speed control
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_ref_speed_rpm	Speed command value (mechanical angle) [r/min]
com_f4_speed_rate_limit_rpm	Maximum increment/decrement width for the speed command [r/min/s] (used when speed control is enabled, mechanical angle)

Variable	Description
com_f4_overspeed_limit_rpm	Speed limit value (mechanical angle) [r/min]
com_u1_flag_volt_err_comp_use	Voltage error compensation setting 0: Disable, 1: Enable
com_u1_flag_mtpa_use	Maximum torque/current control setting 0: Disable, 1: Enable
com_u1_flag_extobserver_use	Disturbance torque/speed estimation observer setting 0: Disable, 1: Enable
s_u1_cnt_ics	ICS watchpoint skip counter
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_pll_estlow_omega_hz	Natural frequency for the position estimation system for HFI [Hz]
com_f4_pll_estlow_zeta	Attenuation coefficient for the position estimation system for HFI
com_f4_extobs_omega	Natural frequency of the disturbance torque/speed estimation observer [Hz]
com_f4_spd_low_to_high_threshold	Switching speed from low to high speed
com_f4_spd_high_to_low_threshold	Switching speed from high to low speed

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

Structure	Variable	Description
st_rmw_param_buffer_t Structure for updating the RMW variables	u2_offset_calc_time	Current offset detection time set setting
	st_motor_parameter_t	Structure for motor parameters
	f4_max_speed_rpm	Maximum speed [r/min] (mechanical angle)
	u1_ctrl_loop_mode	Control loop mode (speed control)
	f4_current_omega_hz	Natural frequency for the current control system [Hz]
	f4_current_zeta	Attenuation coefficient for the current control system
	f4_speed_omega_hz	Natural frequency for the speed control system [Hz]
	f4_speed_zeta	Attenuation coefficient for the speed control system
	f4_speed_lpf_hz	Speed LPF cut-off frequency [Hz]
	f4_ref_speed_rpm	Speed command value [r/min] (mechanical angle)
	f4_speed_rate_limit_rpm	Speed variation limit [r/min/s] (mechanical angle)
	f4_overspeed_limit_rpm	Speed limit value [r/min] (mechanical angle)
	u1_flag_volt_err_comp_use	Flag for whether to use voltage error compensation
	u1_flag_mtpa_use	Maximum torque/current control setting
	u1_flag_extobserver_use	Disturbance torque/speed estimation observer setting
	f4_e_obs_omega_hz	Natural frequency for the inductive voltage estimation system [Hz]
	f4_e_obs_zeta	Attenuation coefficient for the inductive voltage estimation system
	f4_pll_est_omega_hz	Natural frequency for the position estimation system [Hz]
	f4_pll_est_zeta	Attenuation coefficient for the position estimation system
	f4_extobs_omega	Natural frequency of the disturbance torque/speed estimation observer [Hz]
	f4_pll_estlow_omega_hz	Natural frequency for HFI [Hz]
	f4_pll_estlow_zeta	Attenuation coefficient for HFI
	f4_highspd_threshold	Threshold speed of changing low to middle-high speed
	f4_lowspd_threshold	Threshold speed of changing middle-high to low speed

8.5.4 Macro definition

Table 8-10 lists macros.

Table 8-10 List of macros

File name	Macro name	Defined value	Remarks
r_app_main.h	MAIN_UI_RMW	0	The RMW UI is used.
	MAIN_UI_BOARD	1	The Board UI is used (not to be used).
	MAIN_UI_SIZE	2	The number of selectable UIs
r_app_rmw.h	ICS_DECIMATION	5	RMW watchpoint skip count
	ICS_INT_LEVEL	6	RMW interrupt priority
	ICS_BRR	251	Communication baud rate for RMW
	ICS_INT_MODE	1	Communication mode selection for RMW
	ICS_SCI_CH_SELECT	APP_CFG_SCI_CH_SELECT	SCI channel to be used

Note: In ICS2_RX26T.h, a macro that defines the channel used for communication via RMW is provided.

Table 8-11 List of macros

Macro name	Set value
ICS_DECIMATION	APP_CFG_SCI_CH_SELECT
ICS_INT_LEVEL	6
ICS_BRR	251
ICS_INT_MODE	1

8.5.5 Adjustment and configuration of parameters

In the application layer, the configurations must be specified by using the r_app_control_cfg.h file.

For the variables listed in Table 8-8, perform adjustment and configuration from RMW. For details about how to use RMW, see 6.7 and the Renesas Motor Workbench V.3.10 User's Guide (R21UZ0004).

8.6 Manager module and motor control module

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

8.6.1 Functions

Table 8-12 lists the functions of the manager module. Table 8-13 and Table 8-14 list the functions of the motor control module.

Table 8-12 List of manager module functions

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

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

Functions	Description
Speed control	Calculates and outputs a current command value that allows the speed to follow the speed command value.
Speed command setting	Sets a speed command value in the speed control module.
Disturbance torque/speed estimation observer	Estimate the disturbance at low speeds based on the speed and current to control and suppress the disturbance.
Maximum torque/current control	Control the d-axis current so that the maximum torque is output according to the load conditions.

Table 8-14 List of functions of the current control module

Functions	Description
Current control	Performs calculation according to the current command value to set the PWM output value.
Current offset adjustment	Calculates the offset value of the current value detected by AD.
Voltage error compensation	Compensates for the effects of output voltage dead time.
Forward and inverse transformation	Performs coordinate transformation for the current value detected to perform vector control. This function also performs inverse transformation of the coordinate for the calculation results to restore the original coordinate axis.
Modulation	Improves the efficiency by modulation to a PWM signal.
Decoupling control	Calculates interference cancellation to prevent interference between the d and q axes.
High-Frequency pulse Injection	Applies high-frequency pulses to the output voltage to estimate the position and speed when the motor is standstill or running at low speeds.
BEMF observer	Uses the BEMF observer to estimate the position and speed when the motor is running at middle to high speeds.

8.6.2 Module configuration diagram

Figure 8-2 shows the module configuration.

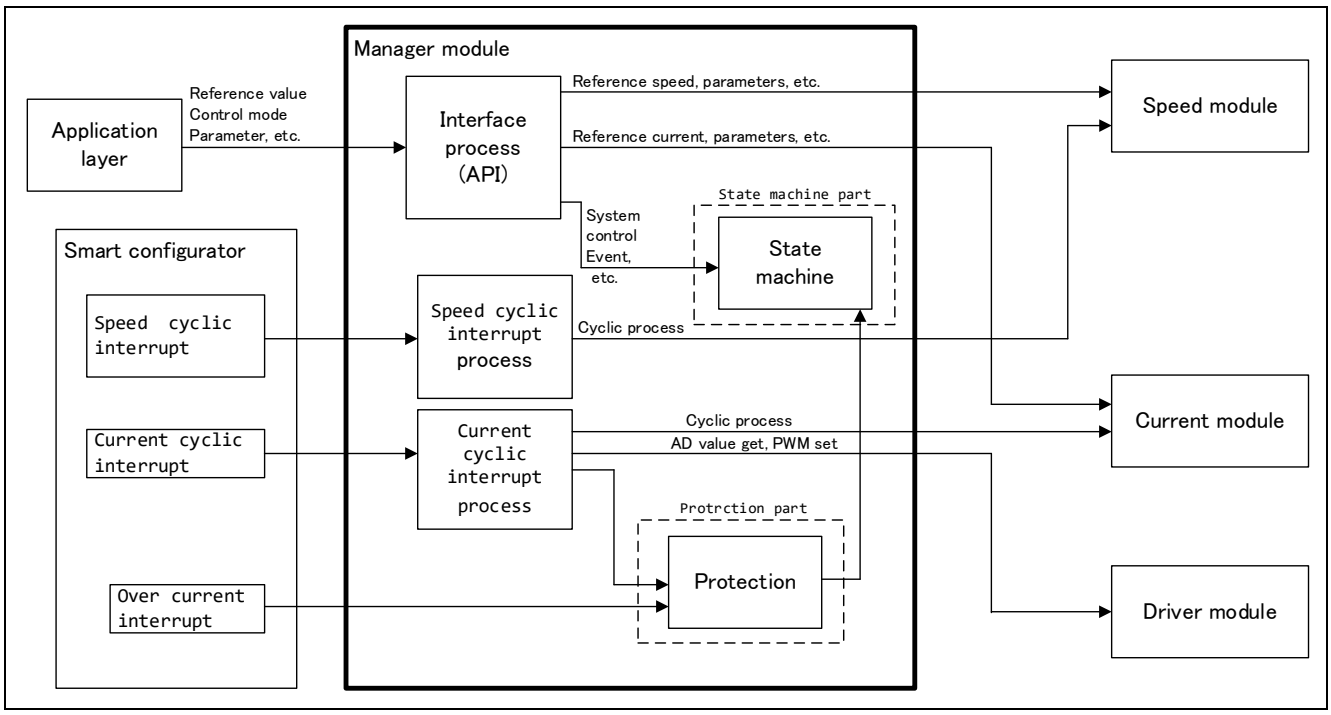
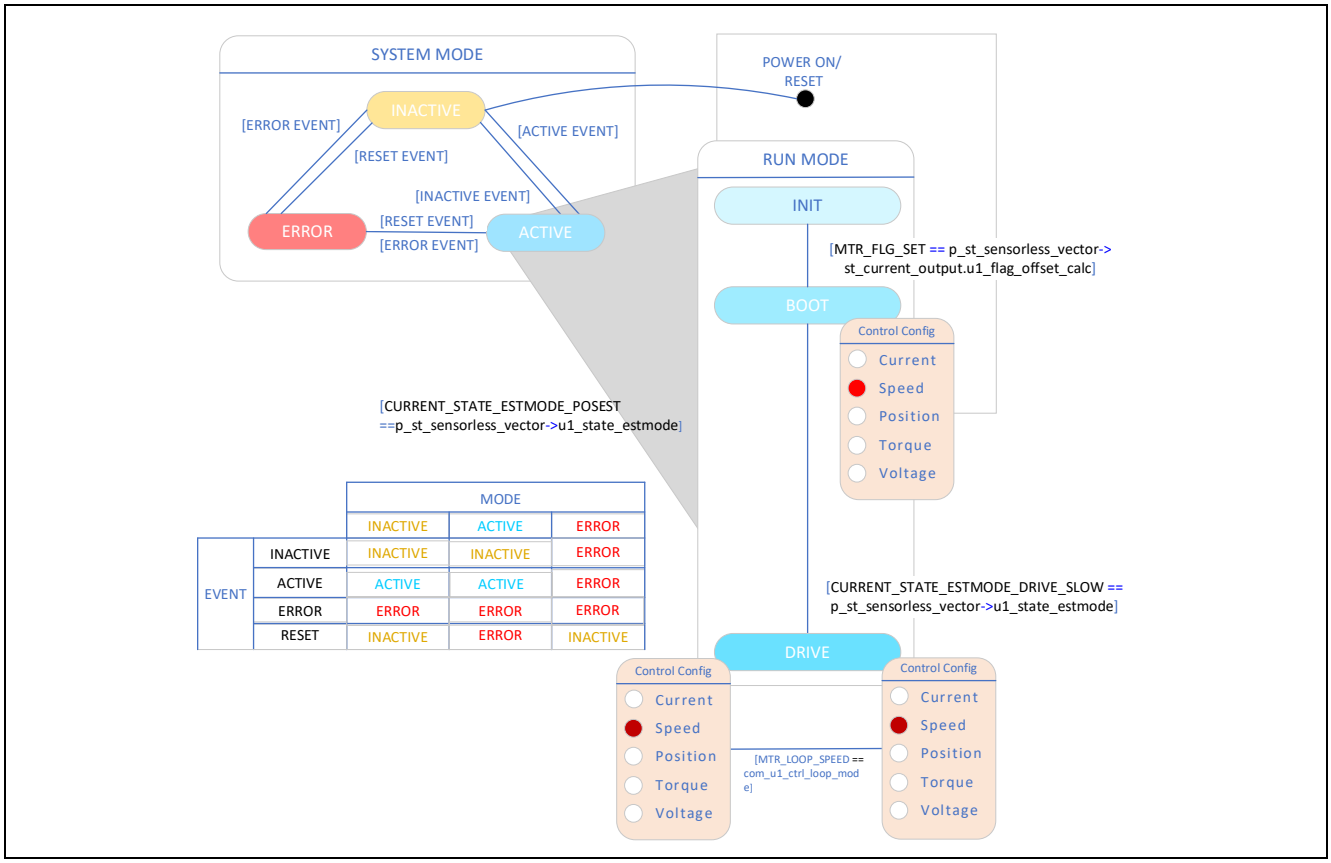


Figure 8-2 Manager module configuration diagram

8.6.3 Mode management

Figure 8-3 shows the state transition diagram for this sample program. For this sample program, the states are managed by using two types of modes: system modes and run modes. Control Config indicates the control systems that are currently active in the software.



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

Figure 8-3 State transition diagram for the sensorless vector control software

(1) SYSTEM MODE

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

(2) RUN MODE

These modes are used to indicate the motor control state. When the system enters ACTIVE mode, the motor state transitions as shown in Figure 8-3.

(3) EVENT

The matrix table in Figure 8-3 shows how the system operation state transitions according to the event that occurs in each system mode. The following table shows the trigger that causes each event to occur.

Table 8-15 List of events

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

8.6.4 Sequence description

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

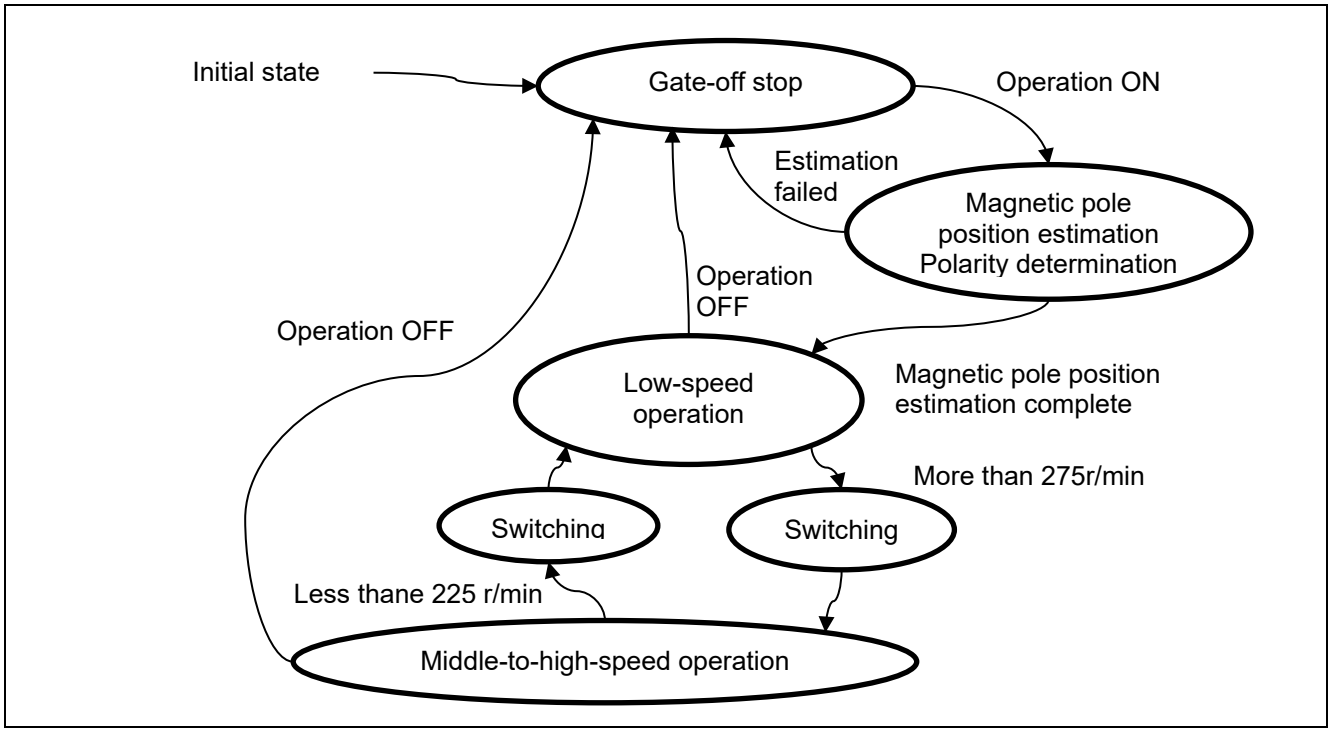


Figure 8-4 State transition diagram for the operation sequence

Table 8-16 Operation sequence states and their descriptions

State	Description
Initial state	This is the state before the MCU is initialized.
Gate-off stop	This is the state in which the power supply of the CPU board is ON. When the inverter power is off, it does not transition to operation ON. If operation is turned on, an error occurs due to undervoltage protection.
Magnetic pole position estimation	A current is applied to the motor to estimate the magnetic pole position. If the estimation succeeds, the state transitions to the low-speed operation mode and rotates at a set speed. If not, it returns to the gate-off stop state.
Low-speed operation	This is the state in which the motor is running from 0 r/min (current is flowing in the motor but the motor is stopped) to approximately 250 r/min (adjustable). Sensorless vector control is performed by using the low-speed sensorless algorithm.
Switching	This is the state in which the sensorless algorithm used is switched from the one used in the low-speed operation to the one used in the middle-to-high-speed operation. During acceleration, the data is transferred to the algorithm for middle-to-high-speed operation. As soon as the data transfer is completed, the state automatically switches to the middle-to-high-speed operation status. During deceleration, the data is transferred to the algorithm for low-speed operation. As soon as the data transfer is completed, the state automatically switches to the low-speed operation status.
Middle-to-high-speed operation	This is the state in which the motor is running from approximately 250 r/min (adjustable) to the motor's rated speed. The motor is controlled by sensorless vector control using the BEMF observer.

8.6.5 Startup sequence

The 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 manager module creates a starting sequence to start the motor.Figure 8-5 shows the startup sequence behavior.

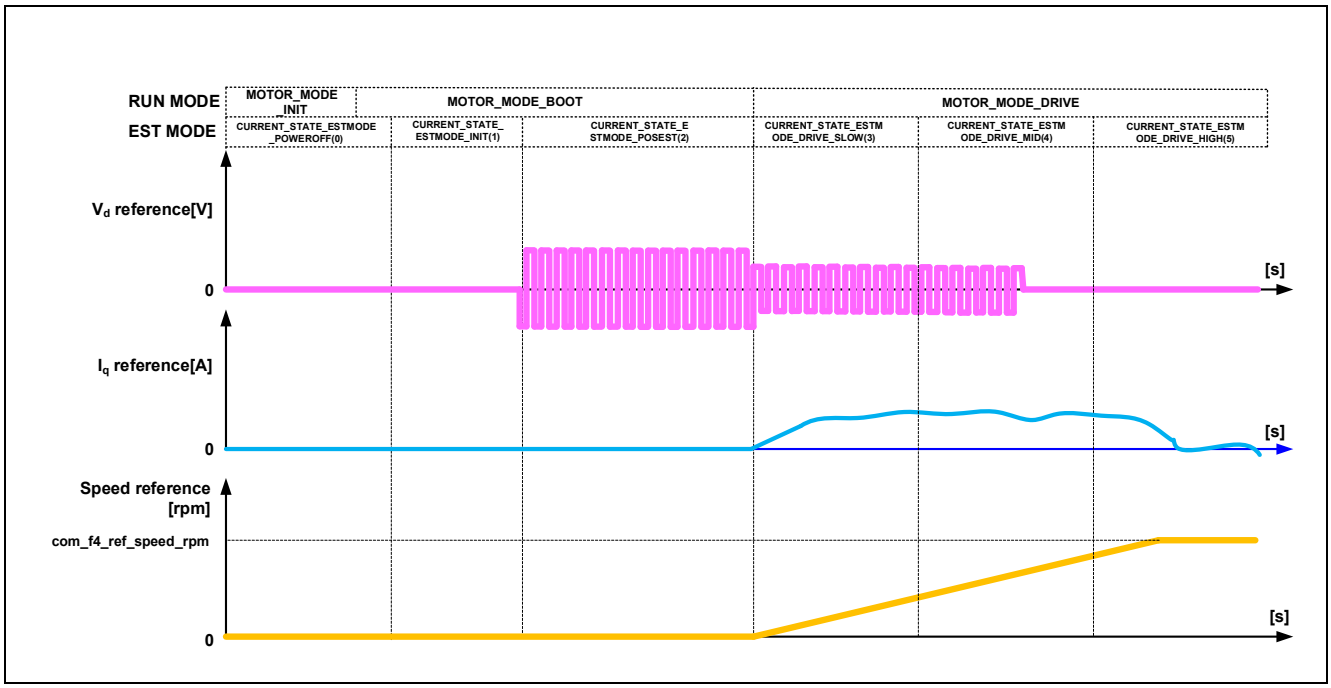


Figure 8-5 Motor start control based on whole-speed-range sensorless vector control

8.6.6 Protection function

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

- Overcurrent error
  Overcurrent errors can be detected on the hardware and in the software.
  A Inactive output is provided to the PWM output pin in response to an emergency stop signal (overcurrent detection) from the hardware. This function monitors U-, V-, and W-phases at the overcurrent monitoring interval. When this function detects an overcurrent (the status in which the current is above the overcurrent limit value), it brings the program to an emergency stop (software detection).
  The overcurrent limit value is automatically calculated from the rated current of the motor (MP_NOMINAL_CURRENT_RMS).

- Overvoltage error
  This function monitors the inverter bus voltage at the overvoltage monitoring interval. When the function detects an overvoltage (that is, a voltage above the overvoltage limit value), it brings the program to an emergency stop. The overvoltage limit value is preset in consideration of conditions such as an error in the resistor value of the detection circuit.

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

- Rotation speed error
  This function monitors the speed at the rotation speed monitoring interval. When the rotation speed exceeds the speed limit value, it brings the program to an emergency stop.

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

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

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

Category	Item	Value
Overcurrent error	Overcurrent limit value [A]	10
	Monitoring interval [μs]	Current control interval*1
Overvoltage error	Overvoltage limit value [V]	60
	Monitoring interval [μs]	Current control interval*1
Low-voltage error	Low-voltage limit value [V]	8
	Monitoring interval [μs]	Current control interval*1
Rotation speed error	Speed limit value [r/min]	2850
	Monitoring interval [μs]	Current control interval*1
Magnetic pole position estimation error	Estimation completion condition	During initial magnetic pole position estimation, the angle fluctuation (absolute value of the difference from the previous value) did not converge to within 1 degree for 10 consecutive times in a 100 msec period.
	Monitoring interval [μs]	Current control interval*1
Polarity determination error	Estimation completion condition	During initial magnetic pole position estimation, the absolute value of PF value did not reach 10.0 or more in a 100 msec period.
	Monitoring interval [μs]	Current control interval*1

Note *1 : See Table 8-1 Basic specifications of whole-speed-range sensorless vector control software.

8.6.7 API

Table 8-18 lists the manager module API functions.

Table 8-18 List of API functions

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

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

API	Description
R_MOTOR_CURRENT_Open	Generates an instance of the current control module.
R_MOTOR_CURRENT_Close	Places the current control module in a reset state.
R_MOTOR_CURRENT_Reset	Initializes the current control module.
R_MOTOR_CURRENT_Run	Activates the current control module.
R_MOTOR_CURRENT_ParameterSet	Inputs the variable information that is used for current control.
R_MOTOR_CURRENT_ParameterGet	Acquires the current control results that are output.
R_MOTOR_CURRENT_ParameterUpdate	Updates the control parameters of the current control module.
R_MOTOR_CURRENT_CurrentCyclic	Performs current control.
R_MOTOR_CURRENT_OffsetCalibration	Performs offset adjustment of current detection.
R_MOTOR_CURRENT_CurrentOffsetRemove	Returns the value with the current detection offset value excluded.
R_MOTOR_CURRENT_VoltErrCompParamSet	Sets the voltage error compensation parameter.
R_MOTOR_CURRENT_BEMFObserverParameterUpdate	Updates the control parameters for the BEMF observer.
R_MOTOR_CURRENT_UpdateAngleNSpole	Updates the rotor angle based on the result of the polarity determination process at startup. Used immediately after the completion of the magnetic pole position estimation process at startup.

Table 8-20 List of API functions of the speed control module

API	Description
R_MOTOR_SPEED_Open	Generates an instance of the speed control module.
R_MOTOR_SPEED_Close	Places the module in a reset state.
R_MOTOR_SPEED_Reset	Initializes the module.
R_MOTOR_SPEED_Run	Activates the module.
R_MOTOR_SPEED_ParameterSet	Inputs the variable information that is used for speed control.
R_MOTOR_SPEED_ParameterGet	Acquires the speed control results that are output.
R_MOTOR_SPEED_ParameterUpdate	Updates the control parameters of the module.
R_MOTOR_SPEED_SpdRefSet	Sets the speed command value.
R_MOTOR_SPEED_SpeedCyclic	Performs speed control.
R_MOTOR_SPEED_ExtObserverParameterUpdate	Updates the control parameters for the disturbance torque/speed estimation observer.

8.6.8 Structure and variable information

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

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

Structure	Variable	Description
st_sensorless_vector_control_t Structure for the manager module	u1_state_speed_ref	Status of the speed command value
	u1_state_estmode	Status of the estimation mode
	u1_direction	Rotation direction
	u1_ctrl_loop_mode	Control mode selection (speed and position)
	u2_error_status	Error status
	u2_run_mode	Run mode
	f4_vdc_ad	Bus voltage [V]
	f4_iu_ad	U-phase current [A]
	f4_iv_ad	V-phase current [A]
	f4_iw_ad	W-phase current [A]
	f4_overcurrent_limit	Overcurrent limit value [A]
	f4_overvoltage_limit	Overvoltage limit value [V]
	f4_undervoltage_limit	Low-voltage limit value [V]
	f4_overspeed_limit_rad	Overspeed limit value [rad/s]
	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	Motor parameter structure
	*p_st_driver	Instance for the driver module
	*p_st_cc	Instance for the current control module
	*p_st_sc	Instance for the speed control module
st_sensorless_vector_cfg_t Structure for setting the manager module control parameters	f4_overspeed_limit_rpm	Speed limit value [r/min] (mechanical angle)
	st_motor	Motor parameter structure

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

Structure	Variable	Description
st_current_control_t Structure for the current control module	u1_active	The active state of the current control module
	u1_flag_volt_err_comp_use	Enables or disables the voltage error compensation function.
	u1_flag_offset_calc	Flag for the current offset calculation
	u2_offset_calc_time	Measurement time setting in current offset adjustment
	u2_crnt_offset_cnt	Measurement count in current offset adjustment
	f4_ctrl_period	Current control interval (period) [s]
	f4_refu	U-phase command voltage [V]
	f4_refv	V-phase command voltage [V]
	f4_refw	W-phase command voltage [V]
	f4_vd_ref	d-axis voltage command value [V]
	f4_vq_ref	q-axis voltage command value [V]
	f4_id_ref	d-axis current command value [A]
	f4_iq_ref	q-axis current command value [A]
	f4_id_ad	d-axis current value [A]
	f4_iq_ad	q-axis current value [A]
	f4_lim_iq	q-axis current limit value [A]
	f4_offset_iu	U-phase offset current value [A]
	f4_offset_iw	W-phase offset current value [A]
	f4_sum_iu_ad	U-phase total current value [A]
	f4_sum_iw_ad	W-phase total current value [A]
	f4_vdc_ad	Bus voltage value [V]
	f4_iu_ad	U-phase current value [A]
	f4_iv_ad	V-phase current value [A]
	f4_iw_ad	W-phase current value [A]
	f4_modu	U-phase duty cycle
	f4_modv	V-phase duty cycle
	f4_modw	W-phase duty cycle
	f4_speed_rad	Speed [rad/s]
	f4_ref_id_ctrl	d-axis current command value [A]
	f4_ref_iq_ctrl	q-axis current command value [A]
	f4_va_max	Maximum voltage on the d and q axes [V]

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

Structure	Variable	Description
st_current_input_t Structure for the current control module input	f4_rotor_angle_rad	Rotor angle [rad]
	f4_iu_ad	U-phase current value [A]
	f4_iv_ad	V-phase current value [A]
	f4_iw_ad	W-phase current value [A]
	f4_vdc_ad	Bus voltage value [V]
	f4_speed_rad	Speed [rad/s]
	f4_id_ref	d-axis current command value [A]
	f4_iq_ref	q-axis current command value [A]
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 8-23 List of structures and variables for the speed control module (1)

Structure	Variable	Description
st_speed_control_t Structure for the speed control module	u1_active	Selects whether to enable the module.
	u1_state_speed_ref	The variable for managing the states that determine the speed command value. It manages the states as shown in "Macro definition" below.
	u1_flag_extobserver_use	Flag for whether to use the disturbance torque/speed estimation observer control
	u1_flag_mtpa_use	Flag for whether to use maximum torque/current control
	f4_speed_ctrl_period	Speed loop control interval [s]
	f4_ref_speed_rad_ctrl	Speed command value for control [rad/s]
	f4_ref_speed_rad	Speed command value output by the position control module during position control [rad/s]
	f4_ref_speed_rad_manual	Speed command value set by the user during speed control [rad/s]
	f4_speed_rad_ctrl	Speed calculated by the speed control module [rad/s]
	f4_speed_rad	Speed that is input [rad/s]
	f4_max_speed_rad	Maximum speed [rad/s]
	f4_speed_rate_limit_rad	Speed variation limit value [rad/s]
	f4_id_ref_output	d-axis current command value [A]
	f4_iq_ref_output	q-axis current command value [A]
	f4_va_max	Maximum voltage on the d and q axes [V]
	f4_id_ad	d-axis current value [A]

	f4_iq_ad	q-axis current value [A]
	f4_torque_current	Torque current [A]
	st_motor_parameter_t	Structure for motor constants
	st_pi_ctrl_t	Structure for PI control
	st_1st_order_lpf_t	Structure for LPF

Table 8-24 List of structures and variables (2)

Structure	Variable	Description
st_speed_cfg_t Structure for setting the parameters for controlling the speed control module	f4_max_speed_rpm	Maximum speed [r/min] (mechanical angle)
	f4_speed_ctrl_period	Speed control interval [s]
	f4_speed_rate_limit_rpm	Speed variation limit value [r/min] (mechanical angle)
	f4_speed_omega_hz	Natural frequency for the speed control system [Hz]
	f4_speed_zeta	Attenuation coefficient for the speed control system
	f4_speed_lpf_hz	LPF for speed control [Hz]
	st_motor_param_t	Structure for motor constants
st_speed_input_t Structure for speed control module input	u1_state_speed_ref	Speed command status
	f4_speed_rad	Speed that is to be input [rad/s]
	f4_va_max	Maximum voltage in the d and q axes [V]
st_speed_output_t Structure for speed control module output	f4_id_ref	d-axis current command value [A]
	f4_iq_ref	q-axis current command value [A]
	f4_ref_speed_rad_ctrl	Speed that is used for PI control [rad/s]
	f4_speed_rad_lpf	Speed after LPF [rad/s]

8.6.9 Macro definition

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

Table 8-25 List of macros

File name	Macro name	Defined value	Remarks
r_motor_sensorless_vector_api.h	MOTOR_LOOP_POSITION	0	Position control mode. * Not supported in this sample program.
	MOTOR_LOOP_SPEED	1	Speed control mode.
	MOTOR_SENSORLESS_VECTOR_ERROR_NONE	(0x0000)	An error status. There is no error.
	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_CURRENT_HW	(0x0001)	An error status. A hardware overcurrent error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_VOLTAGE	(0x0002)	An error status. An overvoltage error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_SPEED	(0x0004)	An error status. An overspeed error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_LOW_VOLTAGE	(0x0080)	An error status. A low-voltage error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_OVER_CURRENT_SW	(0x0100)	An error status. A software overcurrent error has occurred.
	MOTOR_SENSORLESS_VECTOR_ERROR_FAIL_POLES	(0x0200)	An error status. Polarity determination error
	MOTOR_SENSORLESS_VECTOR_ERROR_FAIL_POSITION	(0x0400)	An error status. Magnetic pole position estimation error
	MOTOR_SENSORLESS_VECTOR_ERROR_UNKNOWN	(0xffff)	An error status. An error whose error code is unknown has occurred.
r_motor_sensorless_vector_manager.h	MOTOR_MODE_INIT	(0x00)	INIT run mode
	MOTOR_MODE_BOOT	(0x01)	BOOT run mode
	MOTOR_MODE_DRIVE	(0x02)	DRIVE run mode
r_motor_sensorless_vector_api.h	MOTOR_CTRL_TYPE_POS	0	Macro for switching the control method. Position control mode.
	MOTOR_CTRL_TYPE_SPEED	1	Macro for switching the control method. Speed control mode.

8.7 Driver module

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

8.7.1 Functions

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

Table 8-26 List of functions of the driver module

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

8.7.2 Module configuration diagram

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

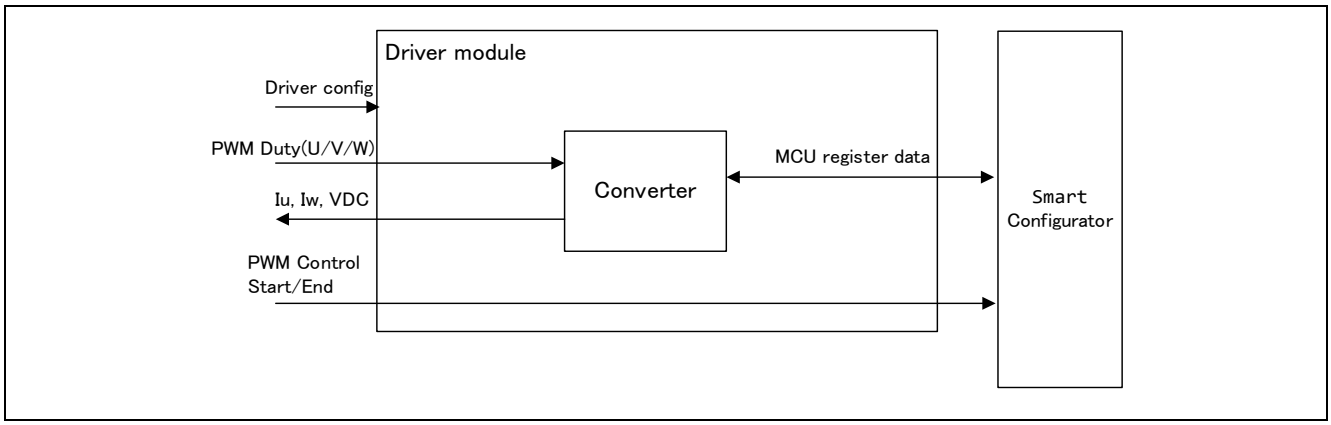


Figure 8-6 Driver module configuration diagram

8.7.3 API

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

Table 8-27 List of API functions for the driver module

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

8.7.4 Configurations

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

Table 8-28 List of configurations

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

Note 1. For details about the functions shown in the "Set value" column, see 10 Smart Configurator setting.
2. When the Smart Configurator Motor component is used, “xxx” is set to “MOTOR”. When the Motor component is not used, it is the module name used for PWM.

8.7.5 Structure and variable information

Table 8-29 lists the structures that are used for the driver module. In the driver module, the structure for the driver module (g_st_driver) is defined by securing an instance of the module from the API.

Table 8-29 List of structures and variables

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

8.7.6 Adjustment and configuration of parameters

In the driver module, parameters that are input from the control parameter configuration (R_MOTOR_DRIVER_ParameterUpdate) are used to associate the motor module and Smart Configurator and to convert data. The parameters are input by using st_speed_config_t (the structure for setting the parameters for controlling the drive module). In the sample program, the information that is defined as configurations is used as the parameter settings. Table 8-30 shows the settings.

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

Variable name	Macro name	File name
*ADCDataGet	DRIVER_CFG_FUNC_ADC_DATA_GET	r_motor_module_cfg.h
*BLDCDutySet	DRIVER_CFG_FUNC_DUTY_SET	
*PWMOutputStop	DRIVER_CFG_FUNC_PWM_OUTPUT_START	
*PWMOutputStart	DRIVER_CFG_FUNC_PWM_OUTPUT_STOP	
f4_shunt_ohm	INVERTER_CFG_SHUNT_RESIST	r_motor_inverter_cfg.h
f4_volt_gain	INVERTER_CFG_VOLTAGE_GAIN	
f4_crnt_amp_gain	INVERTER_CFG_CURRENT_AMP_GAIN	
f4_pwm_period_cnt	MOTOR_COMMON_CARRIER_SET_BASE	r_motor_module_cfg.h
f4_pwm_dead_time_cnt	MOTOR_COMMON_DEADTIME_SET	

9. Setting the parameters

9.1 Overview

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

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

Table 9-1 List of parameter setting files

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

9.2 MCU-related parameters

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

Table 9-2 List of MCU-related parameters

File name	Macro name	Set value	Description
r_motor_module_cfg.h	MOTOR_MCU_CFG_PWM_TIMER_FREQ	120.0f	PWM timer frequency [MHz]
	MOTOR_MCU_CFG_CARRIER_FREQ	20.0f	Carrier wave frequency [kHz]
	MOTOR_MCU_CFG_INTR_DECIMATION	0	Skipping count for carrier wave interrupts
	MOTOR_MCU_CFG_AD_FREQ	60.0f	ADC operating frequency [MHz]
	MOTOR_MCU_CFG_AD_SAMPLING_CYCLE	6+39	ADC sampling cycle [cycles]
	MOTOR_MCU_CFG_AD12BIT_DATA	4095.0f	ADC resolution
	MOTOR_MCU_CFG_ADC_OFFSET	0x7FF	ADC offset value

9.3 List of control function setting parameters

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

Table 9-3 List of operational parameters (general)

File name	Macro name	Set value	Description
r_motor_module_cfg.h	MOTOR_TYPE_BLDC	MOTOR_TYPE_BLDC	Use as default.
	MOTOR_COMMON_CFG_LOOP_MODE	MOTOR_LOOP_SPEED	Use as default.
	MOTOR_COMMON_CFG_OVERCURRENT_MARGIN_MULT	1.5f	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_DISABLE	Sets the processing of TFU-specific functions Set to MTR_DISABLE because RX26T is not supported.

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

File name	Macro name	Set value	Description
r_motor_module_cfg.h	SPEED_CFG_OBSERVER	MTR_ENABLE	Enables or disables the disturbance torque/speed estimation observer. Enable: MTR_ENABLE Disable: MTR_DISABLE
	SPEED_CFG_MTPA	MTR_ENABLE	Maximum torque/current control setting Enable: MTR_ENABLE Disable: MTR_DISABLE
	SPEED_CFG_CTRL_PERIOD	0.0005f	Sets the speed control interval [sec]. Set to 0.0005f to make it 0.5ms.

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

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

9.4 Protection-related parameters

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

Table 9-6 Motor and inverter parameter settings

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

INVERTER_CFG_CURRENT_LIMIT

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

INVERTER_CFG_OVERVOLTAGE_LIMIT

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

INVERTER_CFG_UNDERVOLTAGE_LIMIT

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

9.5 Changing the PWM carrier frequency

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

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

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

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

9.6 Setting the pulse-width modulation method

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

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

Table 9-8 List of configurations

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

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

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

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

Table 9-10 List of macros

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

9.7 Inverter parameters

9.7.1 Overview

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

Table 9-11 Inverter parameter settings

File name	Macro name	Set value	Description
r_motor_inverter_cfg.h	INVERTER_CFG_SHUNT_RESIST	0.010f	Shunt resistance value [ohm]
	INVERTER_CFG_DEADTIME	2.0f	Dead time [μs]
	INVERTER_CFG_VOLTAGE_GAIN	22.2766f	Coefficient for voltage detection
	INVERTER_CFG_CURRENT_AMP_GAIN	20.0f	Gain of the amplifier for current detection
	INVERTER_CFG_INPUT_V	24.0f	Input voltage [V]
	INVERTER_CFG_ADC_REF_VOLTAGE	5.0f	Analog power supply voltage for the MCU [V]
	INVERTER_CFG_COMP_V0	0.564f	Coefficient for compensation of the voltage error [V]
	INVERTER_CFG_COMP_V1	0.782f	Coefficient for compensation of the voltage error [V]
	INVERTER_CFG_COMP_V2	0.937f	Coefficient for compensation of the voltage error [V]
	INVERTER_CFG_COMP_V3	1.027f	Coefficient for compensation of the voltage error [V]
	INVERTER_CFG_COMP_V4	1.058f	Coefficient for compensation of the voltage error [V]
	INVERTER_CFG_COMP_I0	0.022f	Coefficient for compensation of the voltage error [A]
	INVERTER_CFG_COMP_I1	0.038f	Coefficient for compensation of the voltage error [A]
	INVERTER_CFG_COMP_I2	0.088f	Coefficient for compensation of the voltage error [A]
	INVERTER_CFG_COMP_I3	0.248f	Coefficient for compensation of the voltage error [A]
	INVERTER_CFG_COMP_I4	0.865f	Coefficient for compensation of the voltage error [A]

INVERTER_CFG_DEADTIME

Specify the dead time in μs (microseconds) that is described in the inverter specifications and design document. For the MCK-26T motor control kit, 2.0 μs is specified.

INVERTER_CFG_INPUT_V

The default voltage value is the 24 VDC.

INVERTER_CFG_ADC_REF_VOLTAGE

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

INVERTER_CFG_COMP_Vx, INVERTER_CFG_COMP_Ix

See 9.7.4.

9.7.2 Current detection gain

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

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

INVERTER_CFG_CURRENT_AMP_GAIN

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

INVERTER_CFG_SHUNT_RESIST

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

INVERTER_CFG_ADC_REF_VOLTAGE

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

A generalized formula for determining INVERTER_CFG_ADC_REF_VOLTAGE is shown below:

$$INVERTER_CFG_ADC_REF_VOLTAGE = \frac{1}{INVERTER_CFG_SHUNT_RESIST[\Omega]} \times \frac{1}{Current\ per\ 1V[A/V]}$$

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

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

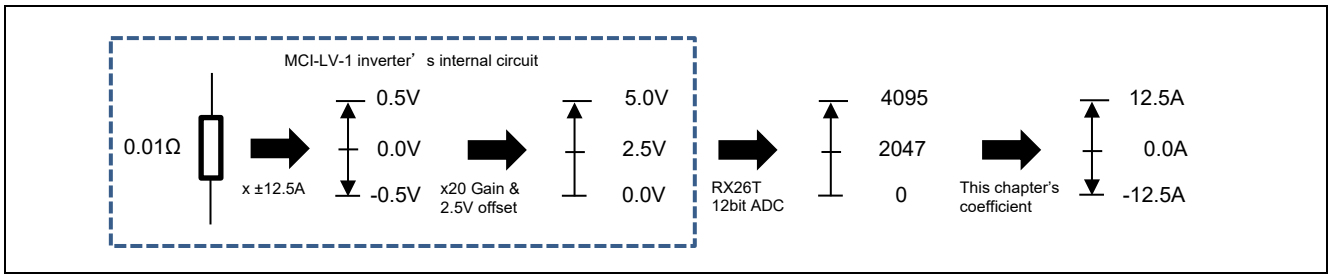


Figure 9-1 Current detection calculation flow

9.7.3 Voltage detection gain

The voltage detection gain is set by INVERTER_CFG_VOLTAGE_GAIN.

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

$$\text{INVERTER_CFG_VOLTAGE_GAIN} = \frac{reference\ of\ Inverter\ bus\ voltage}{reference\ of\ ADC\ input\ voltage} = \frac{111.383}{5} = 22.2766$$

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

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

9.7.4 Voltage error compensation parameters

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

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

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

(3) Setting the voltage compensation table

Performing a switching test using an actual inverter, or creating a voltage compensation table using the voltage error value for dead time obtained from the relationship between dead time and carrier period.

The relationship between current and voltage obtained in the switching test yields a configurable value for a more effective voltage compensation table.

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

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

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

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

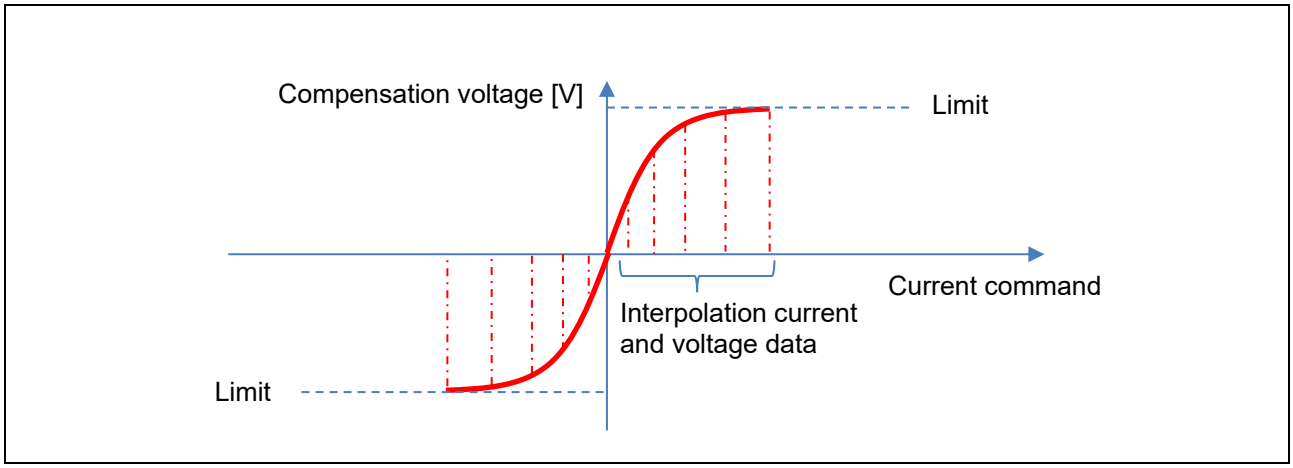


Figure 9-2 Relationship between the compensation voltage value/limit and current command value

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

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

9.8 Motor parameters

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

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

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

Table 9-15 Motor parameter settings

File name	Macro name	Set value	Description
r_motor_targetmotor_cfg.h	MOTOR_CFG_POLE_PAIRS	7	Number of pole pairs
	MOTOR_CFG_MAGNETIC_FLUX	0.0088f	Magnetic flux [wb]
	MOTOR_CFG_RESISTANCE	0.045f	Resistance [ohm]
	MOTOR_CFG_D_INDUCTANCE	0.000095f	d-axis inductance [H]
	MOTOR_CFG_Q_INDUCTANCE	0.000125f	q-axis inductance [H]
	MOTOR_CFG_ROTOR_INERTIA	0.0000294367f	Rotor inertia [kg m^2]
	MOTOR_CFG_NOMINAL_CURRENT_RMS	12.3f	Rated current [A]
	MOTOR_CFG_MAX_SPEED_RPM	2850.0f	Maximum speed [r/min]

MOTOR_CFG_POLE_PAIRS

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

MOTOR_CFG_RESISTANCE

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

MOTOR_CFG_D_INDUCTANCE, MOTOR_CFG_Q_INDUCTANCE

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

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

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

MOTOR_CFG_ROTOR_INERTIA

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

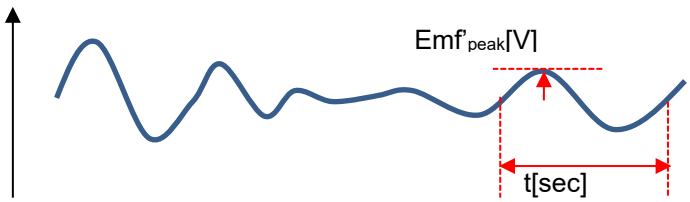
MOTOR_CFG_NOMINAL_CURRENT_RMS

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

MOTOR_CFG_MAGNETIC_FLUX

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

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



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

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

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

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

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

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

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

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

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

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

9.9 Current control parameters

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

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

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

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

Table 9-16 List of current control parameters

File name	Macro name	Set value	Description
r_motor_target motor_cfg.h	CURRENT_CFG_OFFSET_CALC_ TIME	512.0f	Current offset measurement time setting
	CURRENT_CFG_OMEGA	600.0f	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 for the current control system and the attenuation coefficient for the current control system. Set the natural frequency for the current control system

in proportion to the frequency at which to perform current control. The natural frequency can be set to about 1/10 of the current control frequency (PWM carrier frequency). However, in many cases, a lower value may be set with a margin in consideration of noise during position detection and current detection.

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

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

9.10 Speed control parameters

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

To set the natural frequency for the speed control system, use com_f4_speed_omega_hz.

Attenuation coefficient for the speed control system: com_f4_speed_zeta

LPF bandwidth for the speed control system com_f4_speed_lpf_hz

Table 9-17 List of speed control parameters

File name	Macro name	Set value	Description
r_motor_target motor_cfg.h	SPEED_CFG_CTRL_PERIOD	0.0005f	Control interval setting [s]
	SPEED_CFG_OMEGA	10.0f	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	2850.0f	Speed limit value [r/min]
	SPEED_CFG_SPEED_LIMIT_MARGIN	1.1f	Speed limit margin
	SPEED_CFG_RATE_LIMIT_RPM	50.0f	Acceleration limit [r/min/s]

SPEED_CFG_CTRL_PERIOD

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

SPEED_CFG_OMEGA, SPEED_CFG_ZETA

In the speed control module, the control gain is adjusted by tuning the natural frequency for the speed control system and the attenuation coefficient for the speed control system. Increasing the natural frequency for the speed control system improves the responsiveness, expanding the following capability of the speed to the commanded speed. The maximum settable natural frequency for speed control is limited to 1/3 of the maximum settable natural frequency for current control to prevent interference with current control. If the natural frequency of the current control system is 500 Hz, then 500 Hz/3 = 166 Hz. However, in the whole-speed-range sensorless control, because the speed is estimated without using an encoder, set a frequency lower than CURRENT_CFG_PLL_ESTLOW_OMEGA and the natural frequency set by

CURRENT_CFG_PLL_EST_OMEGA. Default values are set with relatively low tracking capability providing a margin. For example, if a disturbance vibrates at a higher natural frequency, increasing the value of the natural frequency to match the disturbance may improve the tracking of the disturbance to the vibration, and the motor may be driven more stably than the default setting.

For the attenuation coefficient for the speed control system, a value in the range from 0.7 to 1.0 is ordinarily 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 can be 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

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

SPEED_CFG_SPEED_LIMIT_MARGIN

Sets the margin percentage for setting the maximum speed. A margin can be set relative to the rated rotation speed. The default value is 1.1f (110%).

9.11 Maximum torque/current control

Maximum torque/current control is an algorithm for adjusting the IPM motor's output to the maximum possible torque. The function is enabled or disabled by SPEED_CFG_MTPA. Normally, this function should be enabled.

Table 9-18 List of configurations

File name	Macro name	Set value	Description
r_motor_module_cfg.h	SPEED_CFG_MTPA	MTR_ENABLE	To use maximum torque/current control, set (MTR_ENABLE). If not, set (MTR_DISABLE).

9.12 Disturbance torque/speed estimation observer

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

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

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

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

The disturbance torque/speed estimation observer is used for motor stability during very low-speed operation around 1 to 30 r/min, so it is not effective at speeds above 100 r/min. To facilitate the switching of the disturbance torque observer, attenuation process (Figure 9-3) is performed by using the parameters SPEED_CFG_SOB_OUTLIM_START_RPM and SPEED_CFG_SOB_OUTLIM_END_RPM.

Table 9-19 List of configurations

File name	Macro name	Set value	Description
r_motor_module_cfg.h	SPEED_CFG_OBSERVER	MTR_ENABLE	Set (MTR_ENABLE) to use the disturbance torque/speed estimation observer. If not, set (MTR_DISABLE).
	SPEED_CFG_SOB_OMEGA	30.0f	The unit is Hz. Approximately 4 to 6 times the natural frequency of the speed control system.
	SPEED_CFG_SOB_OUTLIM_START_RPM	25.0f	Set the start speed [r/min] of the observer's upper limit function.
	SPEED_CFG_SOB_OUTLIM_END_RPM	30.0f	Set the end speed [r/min] of the observer's upper limit function.

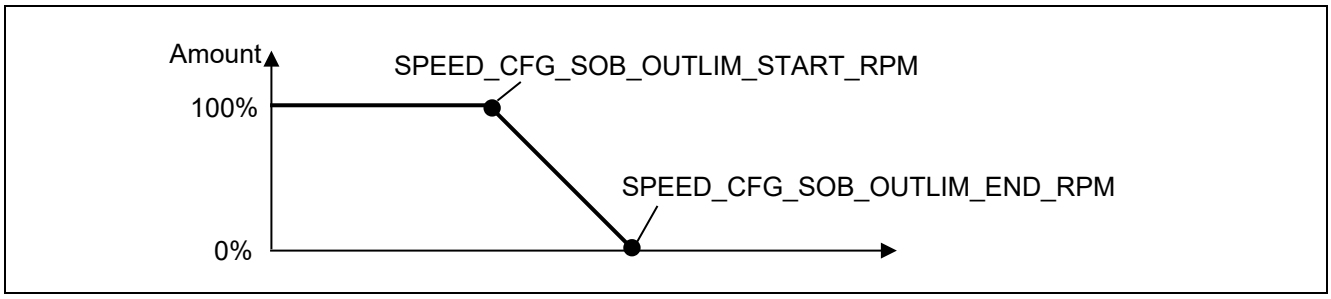


Figure 9-3 Relationship between the amount of compensation and parameters for the disturbance torque/speed estimation observer

9.13 Voltage phase lead compensation parameters

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

Table 9-20 List of configurations

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

9.14 Sensorless control parameters

This section describes how to set parameters required for sensorless control. In sensorless control, a current sensor and predetermined motor parameters and control parameters are used to estimate the magnetic pole position. If the parameters are inappropriate or if an inverter or motor not suited for sensorless control is used, the desired performance may not be achieved.

(1) Low-speed-range sensorless control parameters

Table 9-21 shows the list of parameters to be used for low-speed-range sensorless control.

Table 9-21 Setting parameters for low-speed-range sensorless control

File name	Macro name	Set value	Unit	Description
r_motor_module_cfg.h	CURRENT_CFG_PLL_ESTLOW_OMEGA	50.0f	[Hz]	Natural frequency for the low-speed-range sensorless control PLL [Hz]
	CURRENT_CFG_PLL_ESTLOW_ZETA	1.0f	-	Attenuation coefficient for the low-speed-range sensorless control PLL
	CURRENT_CFG_ESTLOW_PULSEVOLT	8.0f	[V]	Pulse voltage value applied when estimating the magnetic pole position at startup
	CURRENT_CFG_ESTLOW_PULSEVOLT_RUNNING	8.0f	[V]	Pulse voltage value applied during magnetic pole position estimation during operation
	CURRENT_CFG_ESTLOW_ESTTIME	equiv. to 0.2sec	times	Estimation process timeout
	CURRENT_CFG_ESTLOW_ESTTIME_OVER	equiv. to 0.3sec	times	Timeout value for judging estimation processing errors
	CURRENT_CFG_ESTLOW_PULSEFREQ_BOOT	3	times	Pulse application cycle for estimating the magnetic pole position at startup
	CURRENT_CFG_ESTLOW_PULSEFREQ_DRIVE	2	times	Pulse application cycle for estimating the magnetic pole position during operation
	MOTOR_ANGEST_THRESHOLD	0.00872	rad	Threshold for detectability of magnetic pole position estimation
r_motor_current_lowspd_sensorless.h	MOTOR_SENSORLESS_VECTOR_THRESHOLD_HIGHSPEED	28.7979	rad/s	Sets the speed at which the sensorless algorithm switches from the low-speed range to the middle-to-high-speed range.
	MOTOR_SENSORLESS_VECTOR_THRESHOLD_LOWSPEED	23.5619	rad/s	Sets the speed at which the sensorless algorithm switches from the middle-to-high-speed range to the low-speed range.
	MOTOR_SENSORLESS_VECTOR_CURRENT_TABLE_SIZE	8	-	Current buffer table size for estimation. Do not change from 8.
r_motor_sensorless_vector_api.h	CURRENT_SENSORLESS_CHGARGCNT_TOHIGH	2	-	Number of cycles in the current control cycle to be used during sensorless algorithm switching. Set value for low-speed to medium-high-speed.
	CURRENT_SENSORLESS_CHGARGCNT_TOSLOW	1	-	Set value for medium-high-speed to low-speed.

CURRENT_CFG_PLL_ESTLOW_OMEGA

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

CURRENT_CFG_PLL_ESTLOW_ZETA

Sets ζ for the position estimation PLL for low-speed-range sensorless control. Normally, specify 1.0.

CURRENT_CFG_ESTLOW_PULSEVOLT

Specify the magnitude of the pulse (voltage) to be applied at startup. Specify within the range of 0 V to 1/2 of the inverter bus voltage. Since magnetic saturation must be generated to determine polarity, set 8 V or more, depending on the motor specifications. We recommend that you obtain the detailed voltage values experimentally. In this sample program, 8 V is set experimentally.

CURRENT_CFG_ESTLOW_PULSEVOLT_RUNNING

Specify the magnitude of the pulse (voltage) to be applied during operation. Specify within the range of 0 V to 1/2 of the inverter bus voltage. Since generating magnetic saturation phenomenon will adversely affect the operational performance, set the voltage less than half of the voltage set by CURRENT_CFG_ESTLOW_PULSEVOLT. We recommend that you obtain detailed voltage values experimentally, as they will vary from motor to motor.

CURRENT_CFG_ESTLOW_ESTTIME

Specify the time for estimating the magnetic pole position at startup. Multiply by 0.05(The reciprocal of the carrier frequency [kHz]=1/20) to get milliseconds. If 4000 is set, then 4000 x 0.05 = 200 ms. In this case, at least 200 ms is waited for magnetic pole position estimation. Setting it longer can improve the accuracy of magnetic pole position estimation. Set the value to be approximately 200 ms.

CURRENT_CFG_ESTLOW_ESTTIME_OVER

Set the period of time to determine that the results of the magnetic pole position estimation and polarity discrimination have converged and that operation is ready. The value to be set is 6000. The value to be set is 6000, resulting in a waiting time of 6000*0.05=300 ms.

CURRENT_CFG_ESTLOW_PULSEFREQ_BOOT

Specify the cycle of pulses to be applied when estimating the magnetic pole position at startup. The setting range is 1 to 4. For the IPM motor used in this sample program, we have experimentally confirmed that stable estimation can be achieved by setting the pulse application cycle to 3.3 kHz or less. Estimation may fail under conditions where the pulse application period is long while the PWM carrier cycle is long.

Table 9-22 PWM carrier cycle and pulse application cycle settings

PWM carrier cycle	CURRENT_CFG_ESTLOW_PULSEFREQ_BOOT or CURRENT_CFG_ESTLOW_PULSEFREQ_DRIVE value	Pulse injection cycle
20kHz	1	10kHz
	2	5.0kHz
	3	3.3kHz
	4	2.5kHz
12.5kHz	1	6.25kHz
	2	3.12kHz
	3	2.08kHz
	4	1.56kHz
10kHz	1	5.0kHz
	2	2.5kHz
	3	1.66kHz
	4	1.25kHz
5kHz	1	2.5kHz
	2	1.25kHz
	3	0.83kHz
	4	0.62kHz

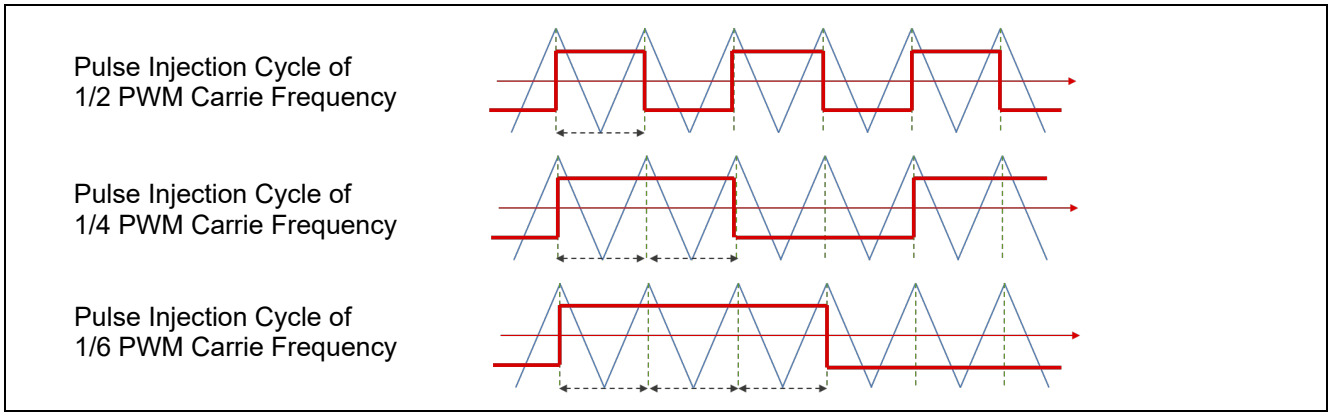


Figure 9-4 Illustration of PWM carrier cycle and pulse application cycle

CURRENT_CFG_ESTLOW_PULSEFREQ_DRIVE

Specify the cycle of pulses to be applied when estimating the magnetic pole position during operation. See Table 9-22 for reference. Normally, specify 1. If the estimation fails during operation, adjust the value of CURRENT_CFG_ESTLOW_PULSEVOLT_RUNNING.

MOTOR_SENSORLESS_VECTOR_THRESHOLD_HIGHSPEED

Sets the speed (in rad/s) at which the speed switches from the low-speed range to the middle-to-high speed range. Above this speed, the high-frequency pulse injection is stopped.

MOTOR_SENSORLESS_VECTOR_THRESHOLD_LOWSPEED

Sets the speed (in rad/s) at which the speed switches from the middle-to-high-speed range to low-speed range. Below this speed, the high-frequency pulse injection is started.

MOTOR_SENSORLESS_VECTOR_CURRENT_TABLE_SIZE

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

MOTOR_SENSORLESS_VECTOR_ PF_START_CNT

This parameter is used to adjust the polarity discrimination start timing. 0-255 is set. One count is made per current control cycle. If 40 is set when the current control cycle is 50us, the polarity discrimination will start after waiting 2ms. Set the value according to the motor constant and the natural frequency of the position estimation control system. Adjustment is necessary when using a motor that fails to discriminate polarity or is difficult to saturate magnetically.

CURRENT_SENSORLESS_CHGARGCNT_TOHIGH

CURRENT_SENSORLESS_CHGARGCNT_TOSLOW

This parameter is used when switching between low and medium/high speed algorithms. Do not change from the default value.

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

Table 9-23 shows the list of BEMF observer parameters to be used for middle-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 9-23 Setting parameters for middle-to-high-speed-range sensorless control

File name	Macro name	Set value	Description
r_motor_module_cfg.h	CURRENT_CFG_E_OBS_OMEGA	1000.0f	Natural frequency for the BEMF observer [Hz]
	CURRENT_CFG_E_OBS_ZETA	1.0f	Attenuation coefficient for the BEMF observer
	CURRENT_CFG_PLL_EST_OMEGA	20.0f	Natural frequency for the middle-to-high-speed-range sensorless control PLL [Hz]
	CURRENT_CFG_PLL_EST_ZETA	1.0f	Attenuation coefficient for the middle-to-high-speed-range sensorless control PLL

10. Smart Configurator setting

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

10.1 Clock settings

Table 10-1 shows the clock settings.

Table 10-1 MCU clock settings

Clock type	Clock setting	
	RX26T RAM64KB Version	RX26T RAM48KB Version
Main clock	240Mz	
System clock (ICLK)	120MHz	
Peripheral module clock (PCLKA)	120MHz	
Peripheral module clock (PCLKB)	60MHz	
Peripheral module clock (PCLKC)	120MHz	
Peripheral module clock (PCLKD)	60MHz	
Flash IF clock (FCLK)	60MHz	
IWDTCLK	120kHz	

10.2 Component settings

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

Table 10-2 Smart Configurator components and their functions

Functions	Component	
	RX26T RAM64KB Version	RX26T RAM48KB Version
3-phase PWM output, A/D conversion of current detection	Config_MOTOR	
A/D conversion processing (inverter bus voltage detection)	Config_S12AD2	
Setting of the port to be used	Config_PORT	
Position and speed control interrupt timer	Config_CMT0	
Independent watchdog timer	Config_IWDT	
Overcurrent detection	Config_POE	

10.3 AD settings

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

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

Table 10-3 AD channels and detection timing settings

Function	Assigned channel		Conversion start trigger
	RX26T RAM64KB Version	RX26T RAM48KB Version	
Measurement of the inverter bus voltage	AN003		TRG4AN (MTU4.TCNT and MTU4.TADCOBRA compare match)
Measurement of the U-phase current	AN000		
Measurement of the W-phase current	AN002		

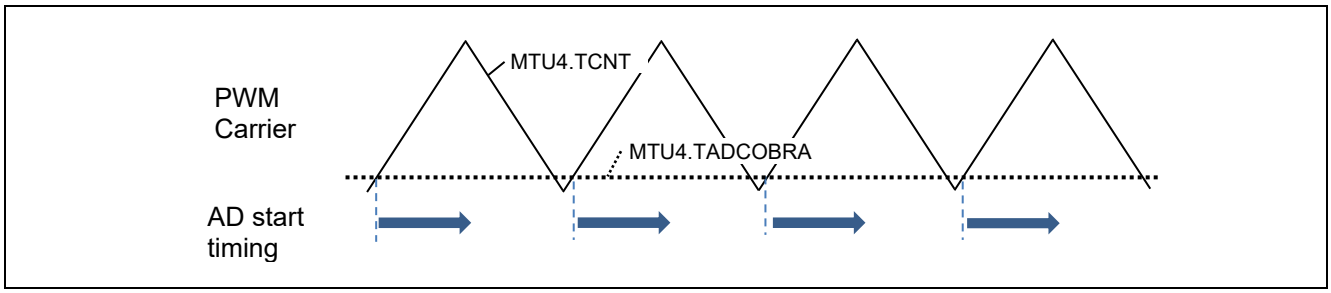


Figure 10-1 AD detection start timing

10.4 Motor settings

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

Table 10-4 MTU settings (RX26T RAM64KB Version)

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

Table 10-5 MTU settings (RX26T RAM48KB Version)

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

10.5 Interrupts

Table 10-6 and Table 10-7 show the information about the interrupts used for the MCUs that use the Motor component.

Table 10-6 List of interrupts (RX26T RAM64KB Version)

Component	Interrupt function	Description
Config_MOTOR	r_Config_MOTOR_ad_interrupt	A/D conversion end interrupt Interrupt level: 10 Multiple interrupt: Enabled
Config_S12AD0/1	None	None
Config_PORT	None	None
Config_CMT0	r_Config_CMT0_cmi0_interrupt	Speed control interrupt Interrupt level: 11 Multiple interrupt: Enabled
Config_IWDT	None	None
Config_POE	r_Config_POE_oei1_interrupt	Hardware overcurrent interrupt Interrupt level: 15 Multiple interrupt: Disabled

Table 10-7 List of interrupts (RX26T RAM48KB Version)

Component	Interrupt function	Description
Config_MOTOR	r_Config_MOTOR_ad_interrupt	A/D conversion end interrupt Interrupt level: 10 Multiple interrupt: Enabled
Config_S12AD0/1	None	None
Config_PORT	None	None
Config_CMT0	r_Config_CMT0_cmi0_interrupt	Speed control interrupt Interrupt level: 11 Multiple interrupt: Enabled
Config_IWDT	None	None
Config_POE	r_Config_POE_oei1_interrupt	Hardware overcurrent interrupt Interrupt level: 15 Multiple interrupt: Disabled
	r_Config_POE_oei2_interrupt	Short-circuited outputs interrupt Interrupt level: 15 Multiple interrupt: Disable

10.6 Details of user codes

Table 10-8 lists the functions that are created in the user code area.

Table 10-8 List of functions in the user area (RX26T RAM64KB Version / RX26T RAM48KB Version)

Component	Function	Description
Config_GPT0	R_Config_GPT0_StartTimerCount	Starts the counting of the PWM timer.
	R_Config_GPT0_StopTimerCount	Stops the counting of the PWM timer.
	R_Config_GPT0_StartTimerCtrl	Enables PWM output.
	R_Config_GPT0_StopTimerCtrl	Disables PWM output.
	R_Config_GPT0_UpdDuty	Writes data to the PWM duty setting register.
	R_Config_GPT0_StartAD	Enables A/D conversion start and end interrupts.
	R_Config_GPT0_StopAD	Disables A/D conversion stop and end interrupts.
	R_Config_GPT0_AdcGetConvVal	Acquires the A/D conversion value.

10.7 POE settings

Table 10-9 and Table 10-10 show the POE settings that can be configured in Config_POE. Refer to the signal specifications of your inverter, as output pin settings vary depending on the inverter specifications.

Table 10-9 POE settings (RX26T RAM64KB Version)

Function			Set value
POE0#	Request acceptance condition	Accept request on falling edge of the POE0# input	✔
	Number of sampling		16 times
	Interrupt setting	Output enable interrupt 1 enabled	✔
		Priority (Group BL1)	Level 15 (Highest)
MTU3/MTU4	Stops the pin output when the output is short-circuited		✔
	Active level setting enabled		✔
	Output pin setting MTIOC3B.MTIOC3D	MTIOC3B active level	High
		MTIOC3D active level	High
	Output pin setting MTIOC4A.MTIOC4C	MTIOC4A active level	High
		MTIOC4C active level	High
	Output pin setting MTIOC4B.MTIOC4D	MTIOC4B active level	High
		MTIOC4D active level	High

Table 10-10 POE settings (RX26T RAM48KB Version)

Function			Set value
POE0#	Request acceptance condition	Accept request on falling edge of the POE0# input	✔
	Number of sampling		16 times
	Interrupt setting	Output enable interrupt 1 enabled	✔
		Priority (Group BL1)	Level 15 (Highest)
MTU3/MTU4	Stops the pin output when the output is short-circuited		✔
	Active level setting enabled		✔
	Output pin setting MTIOC4D.MTIOC4B	MTIOC4D active level	High
		MTIOC4B active level	High
	Output pin setting MTIOC4C.MTIOC4A	MTIOC4C active level	High
		MTIOC4A active level	High
	Output pin setting MTIOC3D.MTIOC3B	MTIOC3D active level	High
		MTIOC3B active level	High

10.8 Pin settings

Table 10-11 and Table 10-12 show the pin interface information.

Table 10-1110-11 Pin interface (RX26T RAM64KB Version)

Function	RX26T RAM64KB Version
Measurement of the inverter bus voltage	P43 / AN003
Measurement of the U-phase current	P40 / AN000
Measurement of the W-phase current	P42 / AN002
PWM output (Up) / "High" active	P76 / MTIOC4D
PWM output (Vp) / "High" active	P75 / MTIOC4C
PWM output (Wp) / "High" active	P74 / MTIOC3D
PWM output (Un) / "High" active	P73 / MTIOC4B
PWM output (Vn) / "High" active	P72 / MTIOC4A
PWM output (Wn) / "High" active	P71 / MTIOC3B
PWM emergency stop input when an overcurrent is detected	P70 / POE0#

Table 10-12 Pin interface (RX26T RAM48KB Version)

Function	RX26T RAM48KB Version
Measurement of the inverter bus voltage	P43 / AN003
Measurement of the U-phase current	P40 / AN000
Measurement of the W-phase current	P42 / AN002
PWM output (Up) / "High" active	P71 / MTIOC3B
PWM output (Vp) / "High" active	P72 / MTIOC4A
PWM output (Wp) / "High" active	P73 / MTIOC4B
PWM output (Un) / "High" active	P74 / MTIOC3D
PWM output (Vn) / "High" active	P75 / MTIOC4C
PWM output (Wn) / "High" active	P76 / MTIOC4D
PWM emergency stop input when an overcurrent is detected	P96 / POE4#

11. Evaluation results

11.1 Motor control evaluation

11.1.1 Acceleration/deceleration characteristics

The acceleration characteristics from standstill to the speed of 800 r/min and the deceleration characteristics from the speed to standstill are shown below. At around 250 r/min, the algorithm is automatically switched from high-frequency pulse injection to BEMF observer, and acceleration/deceleration is performed properly.

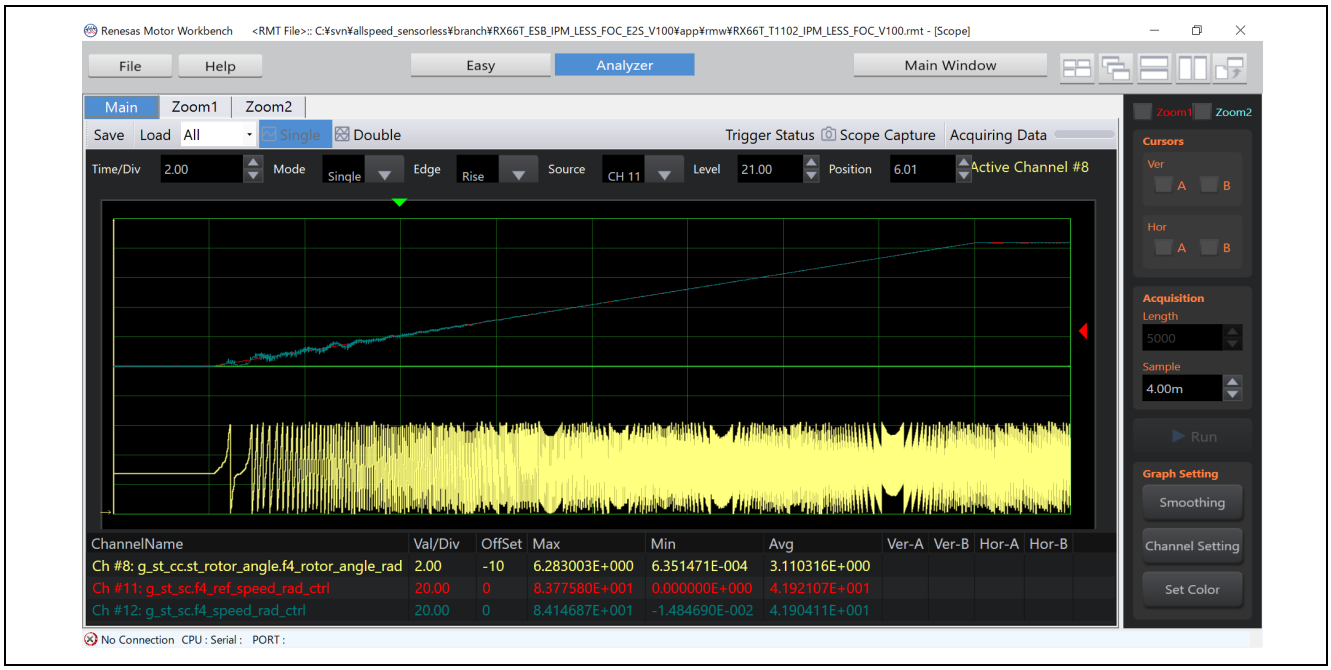


Figure 11-1 Acceleration characteristics

Figure 11-2 Deceleration characteristics

11.1.2 Load characteristics

At a PWM carrier frequency of 20 kHz, we have confirmed that the rated load is achieved.

Due to the current detection function of the MCI-LV-1 used in this environment and the limitation of the output current capacity, load operation is limited to 50%.

11.2 CPU utilization

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

Table 11-1 Control loop and CPU loading rate

CPU board	Control loop type	Control interval	Processing time	CPU loading rate
RX26T RAM64KB Version	Current control loop	50s (0 decimation)	15.6 us	31.2 %
	Speed control loop	500 us	17.5 us	3.5 %
RX26T RAM48KB Version	Current control loop	50s (0 decimation)	15.44 us	30.9%
	Speed control loop	500 us	4.46 us	0.9%

11.3 Program size and RAM usage

The program size (ROM) and RAM usage for this sample program are as below. In the optimization settings of the compiler, the optimization level is set to 2 (-optimize = 2) and the optimization method is set to the one that is code-size oriented (-size).

Table 11-2 Program size and RAM usage

Item	Value	
	RX26T RAM64KB Version	RX26T RAM48KB Version
Program size (ROM)	23836 [Bytes]	23627 [Bytes]
RAM usage	9896 [Bytes]	9904 [Bytes]
Maximum value of stack analysis result	196 [Bytes]	196 [Bytes]
Stack size setting value for IDE environment	5120 [Bytes]	5120 [Bytes]

12. FAQ

12.1 Troubleshooting

Table 12-1 shows typical phenomena and their solutions.

Table 12-1 List of phenomena and their solutions

Phenomena	Solution
When the motor is operated, it makes a beeping sound.	A high-frequency pulse voltage in the audible range is applied to estimate the magnetic pole position. This high-frequency sound is emitted from the motor. It is not abnormal.
When the rotation speed of the motor exceeds a certain speed, the beeping sound stops.	A high-frequency pulse is applied to estimate the magnetic pole position in the low-speed range, but when the set speed is reached, it automatically switches to magnetic pole position estimation by the BEMF observer for the middle-to-high-speed range, and the beeping sound stops.
When the load is applied, the motor stops with an error or steps out.	• Verify that the protection parameters are appropriate. • Verify that the motor parameters are correct. • If the MTPA function is disabled, enable it. • When the current in an IPM motor exceeds a certain level, the magnetic saturation phenomenon causes the inductance to decrease, decreasing Ld to Lq saliency ratio, which may worsen the estimation accuracy and prevent the motor from producing torque.
When a load is applied, the motor keeps turning at a speed other than the setting.	The motor is stepped out. The motor is out of control and must be stopped immediately. Inappropriate motor parameters or control parameters or hardware performance limitations such as sensors may prevent control. Reconsider the design.
Motor cannot rotate after stopping due to an error.	For details, see 6.8 (c). It explains how to recover from errors.
The motor stops with an error even after starting.	• See 6.8 (c) for the cause of the error. • Verify that there is no overvoltage or undervoltage in the inverter bus voltage and that the output is not short-circuited. • Verify that sensor-related signal settings are correct. Verify that the POE settings, motor settings (MTU terminal and active level settings), etc. in the Smart Configurator are properly set according to the inverter circuit specifications.
Values set from RMW are not reflected.	Manipulate variables in com_u1_enable_write to rewrite the parameters. When the timing for writing values to com_u1_enable_write is prior to writing parameters, the internal reflection process operates first. Address as the following: • Put com_u1_enable_write on the last line. • Write com_u1_enable_write twice or toggle write

12.2 Frequently asked questions

12.2.1 Is it possible to control SPM motors with sensorless vector control over the whole speed range?

Almost SPM motors (surface permanent magnet synchronous motors) have a 1:1 ratio of Ld to Lq, and the high-frequency pulse injection method used in this document does not allow sensorless vector control in the standstill to low-speed range. Use the open loop control sample program.

12.2.2 Which IPM motors can be controlled with sensorless vector control over the whole speed range?

The saliency ratio (ratio of Ld to Lq) must be at least 20%. However, control may become difficult under high load or other conditions, when the magnetic saturation phenomenon causes a large change in the saliency ratio and the saliency ratio becomes less than 20%.

12.2.3 What is magnetic saturation?

This is a phenomenon in which a change occurs in the magnet characteristics inside the motor when a certain amount of current is applied to drive the motor. When magnetic saturation occurs, the way current flows changes, and problems such as abnormal heat generation and deterioration of controllability are likely to occur.

In particular, IPM motors are known to be prone to magnetic saturation due to their structural characteristics, and various countermeasure technologies have been researched and devised. This sample program does not include an algorithm that takes magnetic saturation into account, so the user must implement countermeasures if magnetic saturation occurs.

12.2.4 Does high-frequency pulse injection generate noise? Is there a way I can reduce it?

High-frequency noise is generated from the motor during standstill to low-speed operation. During startup, a large current must be applied to estimate the magnetic pole position, resulting in a slightly louder high-frequency noise than while the motor is rotating.

To reduce noise, the pulse applied voltage can be lowered, the carrier frequency can be increased, or the current sensor can be changed from a shunt type to a Hall CT to control at crests and troughs of the carrier.

This sample program does not support sensorless control synchronized with carrier crests and troughs.

12.2.5 What is stepping out?

Stepping out is a condition in which the motor cannot be controlled properly because the magnetic pole position estimated by the motor control software deviate from the actual magnetic pole position of the motor. Overcurrent may cause the motor to stop with an error or to continue rotating in an uncontrollable state. This occurs in the following cases:

- The magnetic pole position at startup failed.
- Due to the design or structural characteristics of the motor, the motor loses its positive polarity at high loads, making it difficult to estimate the magnetic pole position.
- Parameters of the current control system or speed control system are inappropriate.

12.2.6 Is a step-out detection function included?

This sample program does not include a step-out detection function. The user must implement a step-out detection function.

12.2.7 The value of a variable displayed in RMW is abnormal.

When any change is made to the software, it is necessary to register the Map file (that is generated after the build in the HardwareDebug folder) with RMW to update the variable status of the software. If you omit this step, the variable may not display correctly. For details, see 6.6.

Revision History

Rev.	Date of issue	Amendments	
		Page	Point
1.00	2023/11/10	-	First edition issued
1.10	2024/12/10	28, 36	- Fix unclear procedure of RMW operation. - Fix missing axis name(dc/qc).
1.11	2025/06/30	6	Software version and Reference materials upgrade
		9	Table 3-1 and Table 3-2 updated
		13, 17, 19	Corrected the notation of e^2 studio
1.20	2026/06/30	-	RX26T-TYPE-C information added

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

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

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

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

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

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

Notice

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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

© 2026 Renesas Electronics Corporation. All rights reserved.