Sensorless vector control for permanent magnetic synchronous motor - 1shunt current detection

For Renesas Flexible Motor Control Series

Introduction

This application note describes the sample program to drive a permanent magnetic synchronous motor with sensorless vector control and 1-shunt current detection based on Renesas microcontroller.

The targeted software for this note is only to be used for reference purposes and Renesas Electronics Corporation does not guarantee the operations. Please use this after carrying out a thorough evaluation in a suitable environment.

Contents

1. Introduction

This application note explains how to implement the vector control software that drives permanent magnetic synchronous motor using the microcontroller (MCU) Renesas RA series. This sample program can drive a motor with the electrical kit ‘Renesas Flexible Motor Control Series’. And, this program supports the motor control development support tool, ‘Renesas Motor Workbench’. User can confirm the parameters of program with this tool and can use as User Interface to control motor driven. Please consider selecting which MCU or develop your program with refer to the period of interrupt process.

This sample program supports ‘QE for Motor’. Please use the workflow to develop motor software simply by following the steps.

Target software

The following shows the target software for this application note:

- RA6T2_MCILV1_SPM_LESS_FOC_1SHUNT _E2S_V121
- RA6T2_MCB2_MCILV1_SPM_LESS_FOC_1SHUNT _E2S_V100
- RA6T3_MCILV1_SPM_LESS_FOC_1SHUNT _E2S_V111
- RA4T1_MCILV1_SPM_LESS_FOC_1SHUNT _E2S_V111
- RA8T1_MCILV1_SPM_LESS_FOC_1SHUNT _E2S_V111
- RA8T2_MCILV1_SPM_LESS_FOC_1SHUNT _E2S_V101

Reference materials

RA6T2 Group User’s Manual: Hardware (R01UH0951)
RA6T3 Group User’s Manual: Hardware (R01UH0998)
RA4T1 Group User’s Manual: Hardware (R01UH0999)
RA8T1 Group User’s Manual: Hardware (R01UH1016)
RA8T2 Group User’s Manual: Hardware (R01UH1067)
RA Flexible Software Package Documentation (Release v6.2.0)
Application note: ‘Sensorless vector control for permanent magnet synchronous motor (Algorithm)’ (R01AN3786)
Renesas Motor Workbench User’s Manual (R21UZ0004)
Renesas Motor Workbench Quick start guide (R21QS0011)
MCK-RA6T2 User's Manual (R12UZ0091)
MCK-RA6T3 User's Manual (R12UZ0114)
MCK-RA4T1 User's Manual (R12UZ0115)
MCK-RA8T1 User's Manual (R12UZ0133)
MCK-RA8T2 User's Manual (R12UZ0172)

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 environments	3
Confirm the wiring	4.7
Prepare a software development environment	5
Write the sample program to MCU	6.3
Install software for operating the motor on the PC.	6.4
Modify the sample program, and then reflect the changes in RMW	6.5
Drive the motor	6.7
Stop the motor	6.8
Examine the motor control algorithms	7
Examine the structure of the sample program	8
Verify and change the motor parameters	10.11
Changing the PWM carrier frequency	10.8
Change the control parameters	10.12, 10.13
Check the frequently asked questions Check the troubleshooting tips	12

2. Glossary

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

Table 2-1 Glossary

Term	Description
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/design-resources/boards-kits/mc-com
RMW	Refers to Renesas Motor Workbench, which is the GUI operation software on PC specifically designed for motor control.
QE for Motor	QE for Motor is a software development support tool for motors that allows you to develop motor software by simply following the workflow. For details, refer to the following URL. https://www.renesas.com/software-tool/qe-motor-development-assistance-tool-motor-applications
Inverter bus voltage	Refers to the DC voltage fed to the inverter circuit. Also called DC intermediate voltage.
Open loop	Refers to a motor control technique that does not need current or position feedback signals to control the voltage.
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.
Electrical angle	Phase angle of the output current flowing in the motor. It can be converted to a mechanical angle by dividing it by the number of pole pairs of the motor.
Mechanical angle	Refers to the physical rotation angle of the motor axis. One rotation of the axis per minute is 1rpm.

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 Development Environment : Hardware

Category	Product used
Microcontroller / CPU board product type	RA6T2 (R7FA6T2BD3CFP) / MCB2 Ver.1 RTK0EMA270C00000BJ RA6T2 (R7FA6T2BD3CFP) / MCB2 Ver.2 RTK0EMA270C00002BJ RA6T3 (R7FA6T3BB3CFM) / RTK0EMA330C00000BJ RA4T1 (R7FA4T1BB3CFM) / RTK0EMA430C00000BJ RA8T1 (R7FA8T1AHECBD) / RTK0EMA5K0C00000BJ RA8T2 (R7KA8T2LFLCAC) / RTK0EMA6L0C00000BJ
Inverter board	MCI-LV-1 Inverter board / RTK0EM0000B12020BJ
Motor	R42BLD30L3 (Product of 'MOONS')

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 e^2 studio evaluation edition.

Table 3-2 Software Development Environment

e^2 studio version	FSP version	Toolchain version
e^2 studio：2025-10	V6.2.0	GCC ARM Embedded :13.2.1.arm-13-7

4. Building a hardware environment

4.1 Overview : Hardware environments

This section describes hardware environments in which an SPM motor is operated using this sample program. Figure 4-1 shows an example of hardware configuration.

In the sections that follow, the power supply (4.2), the inverter (4.3), the CPU boards(4.4) and the on board debugger (4.6) are described in detail.

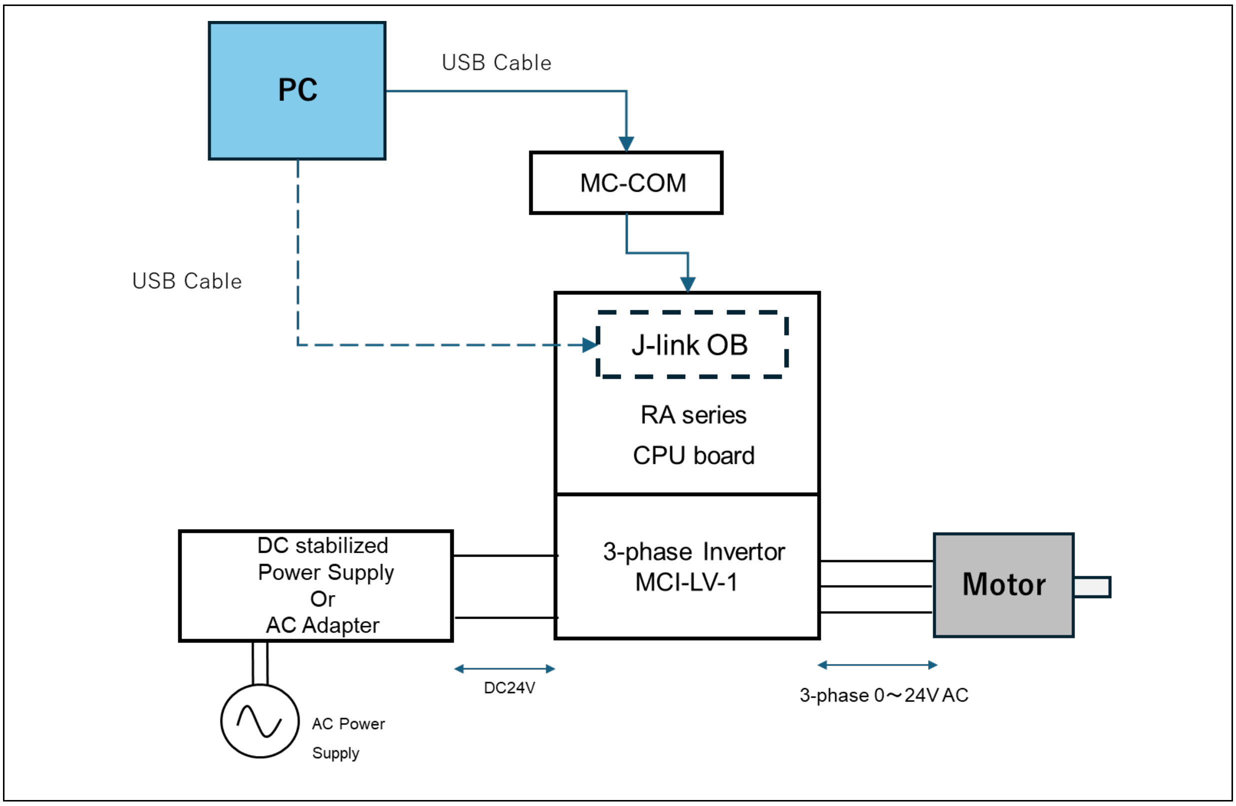


Figure 4-1 Example of hardware configuration

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, 2.5A or more) is used to supply a voltage of 24 VDC to the 3-phase inverter MCI-LV-1.

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

The inverter introduced here has an output current of 10 A max.

4.3 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. Please consider the selection and design of the inverter carefully.

- 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

4.4 Setting up the RA series CPU board

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

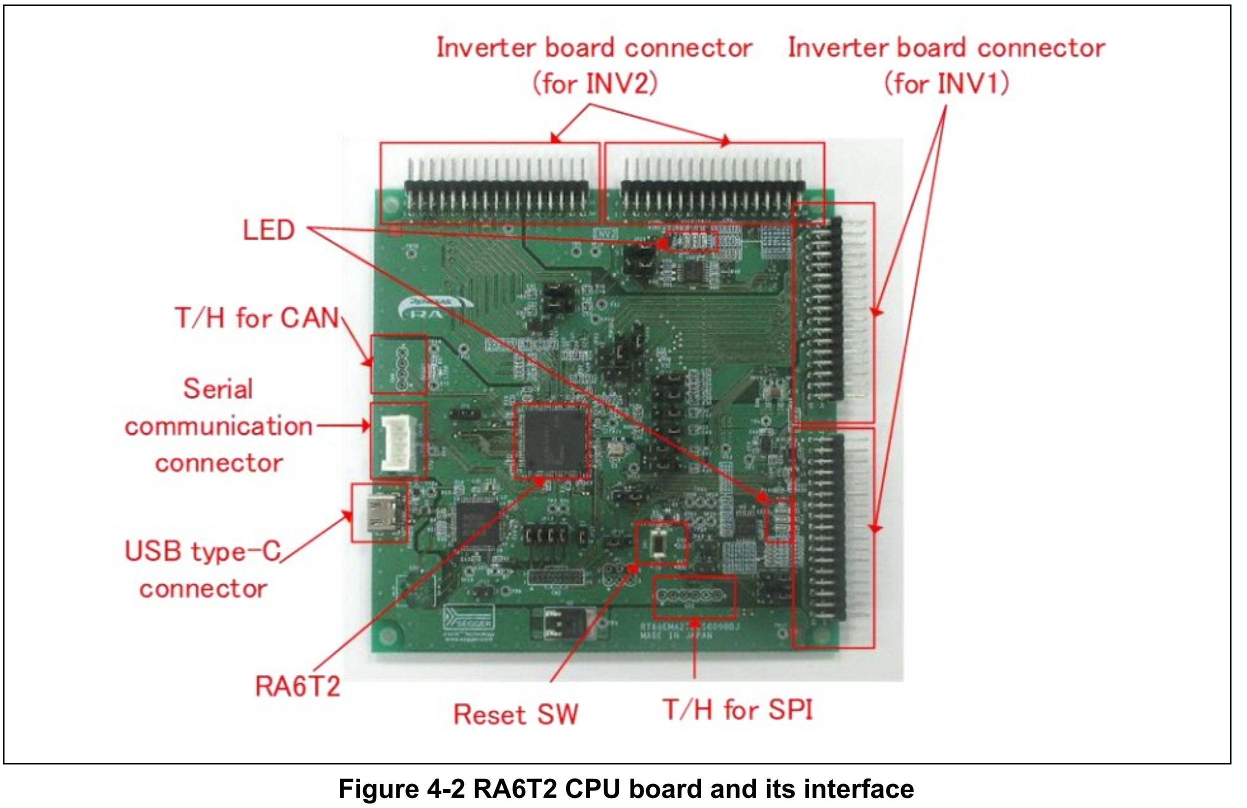


Figure 4-2 RA6T2 CPU board and its interface

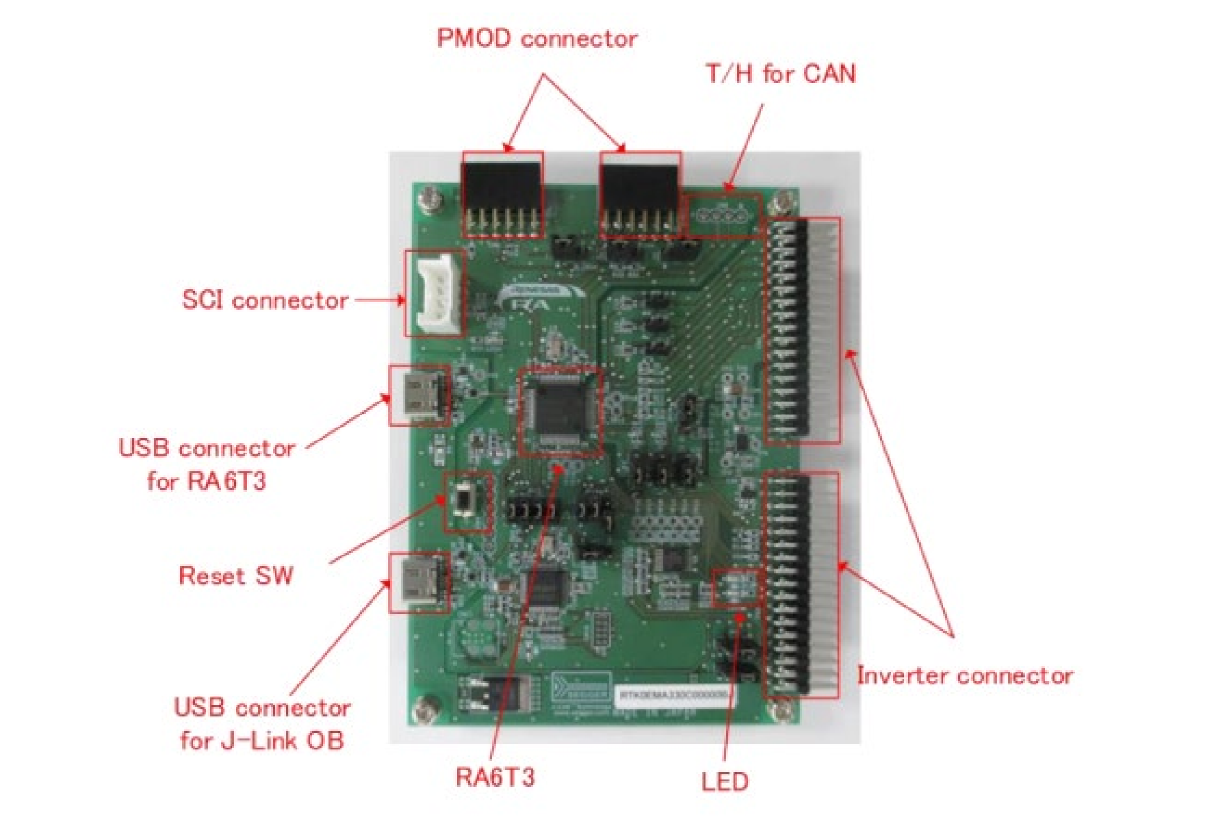


Figure 4-3 RA6T3 CPU board and its interface

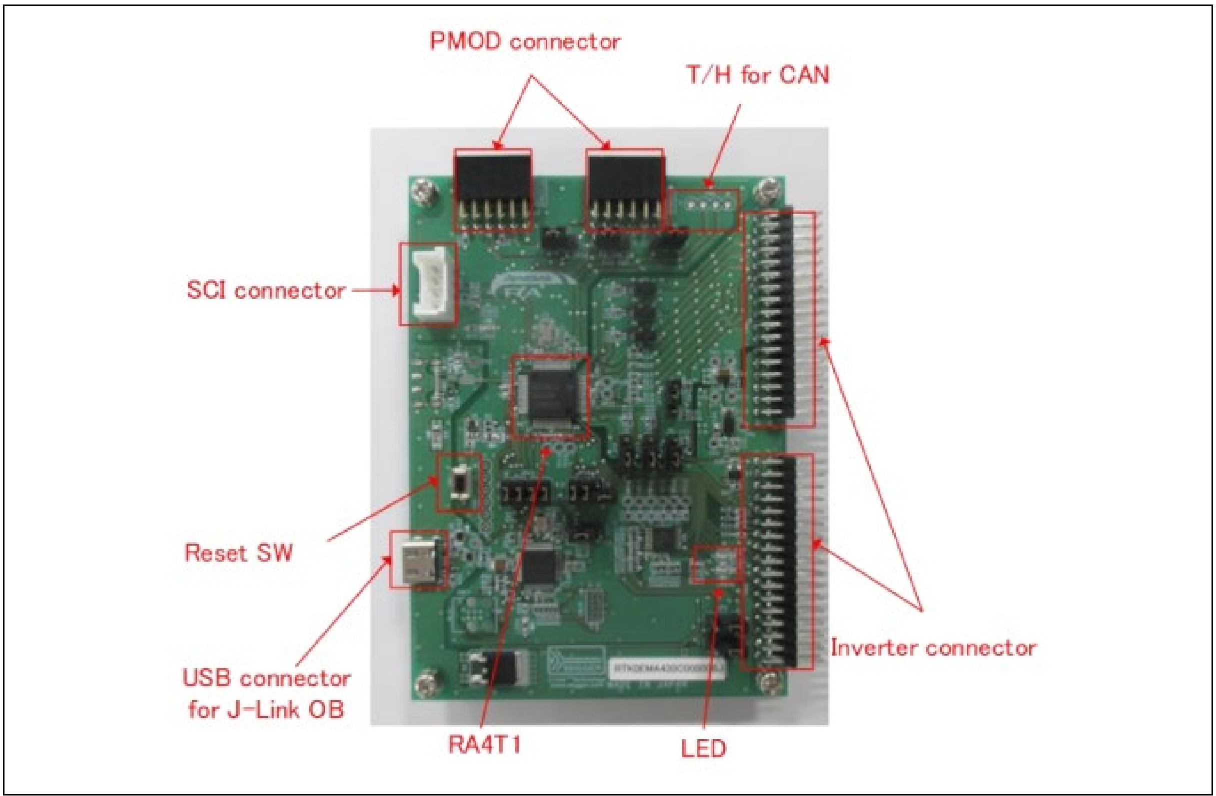


Figure 4-4 RA4T1 CPU board and its interface

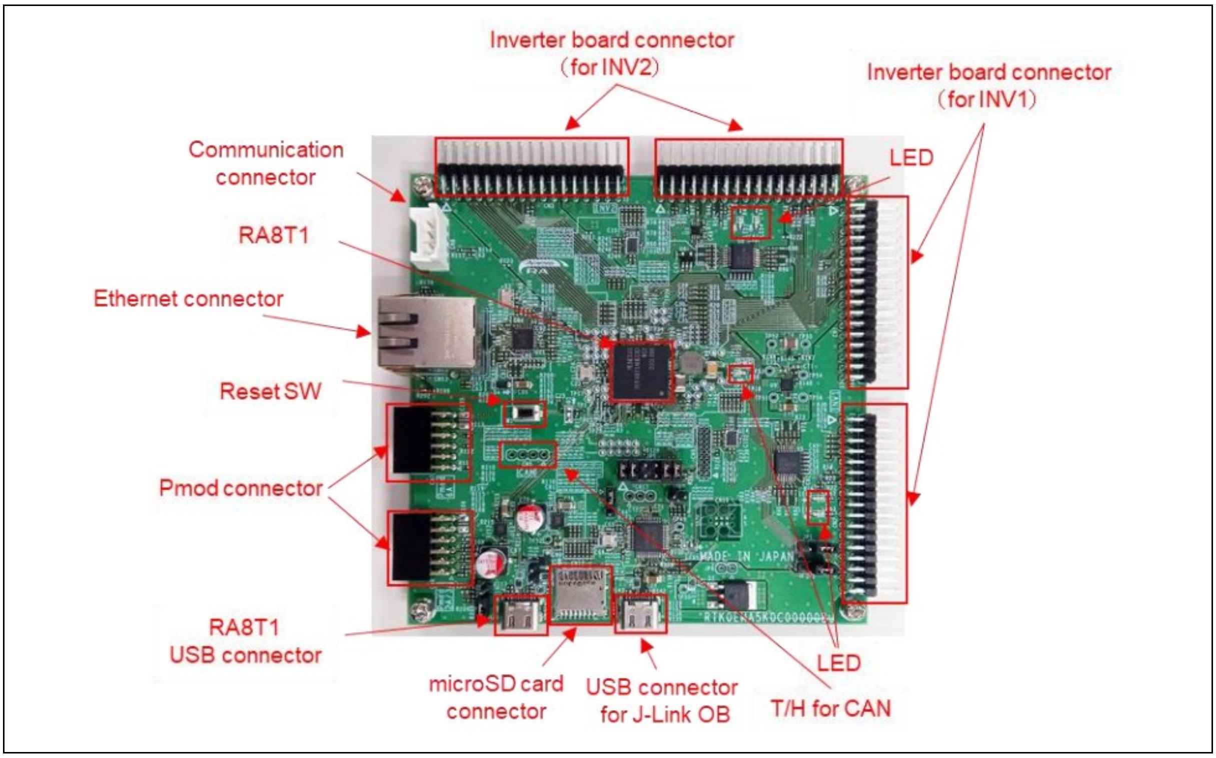


Figure 4-5 RA8T1 CPU board and its interface

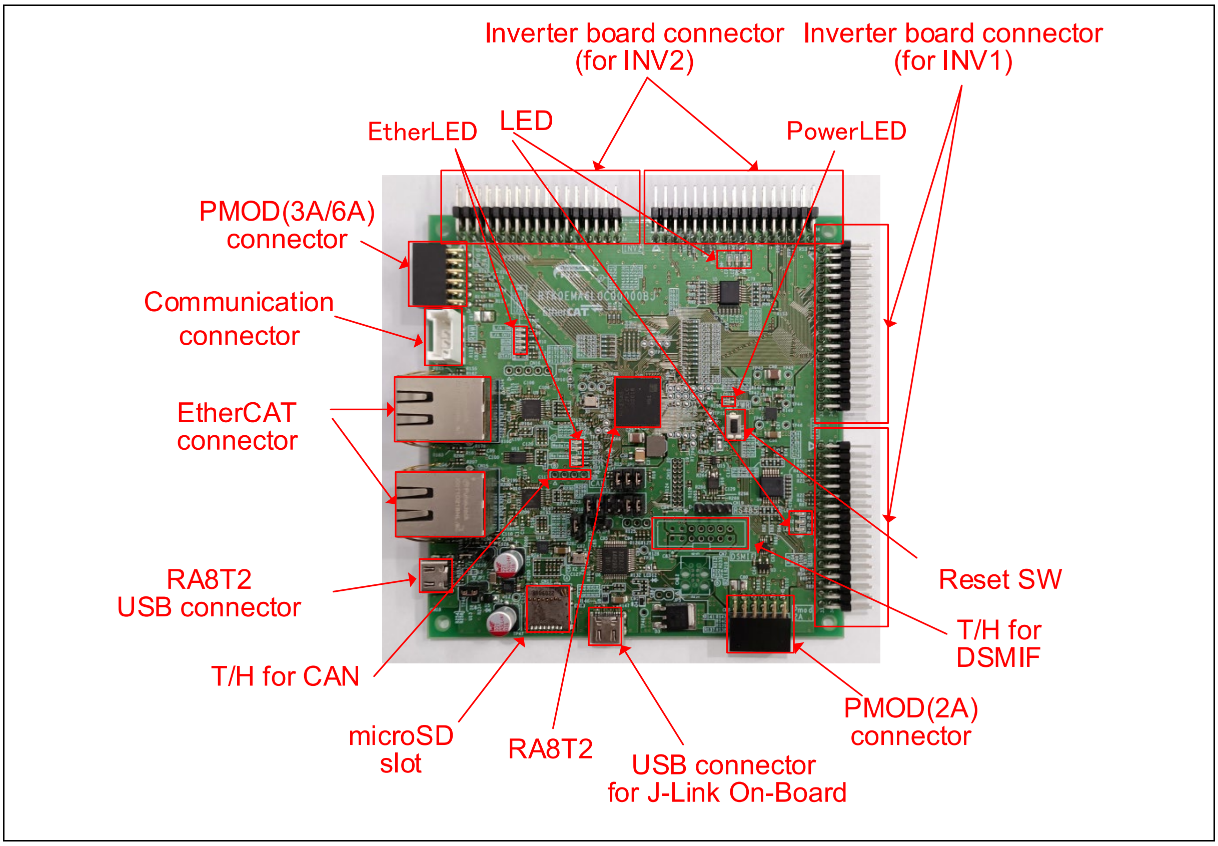


Figure 4-6 RA8T2 CPU board and its interface

4.5 Connection example of kit

Figure 4-7 shows an example of the connection of a CPU board in combination with an inverter board kit (MCI-LV-1) and a communication board kit (MC-COM, model name: RTK0EMXC90Z00000BJ).

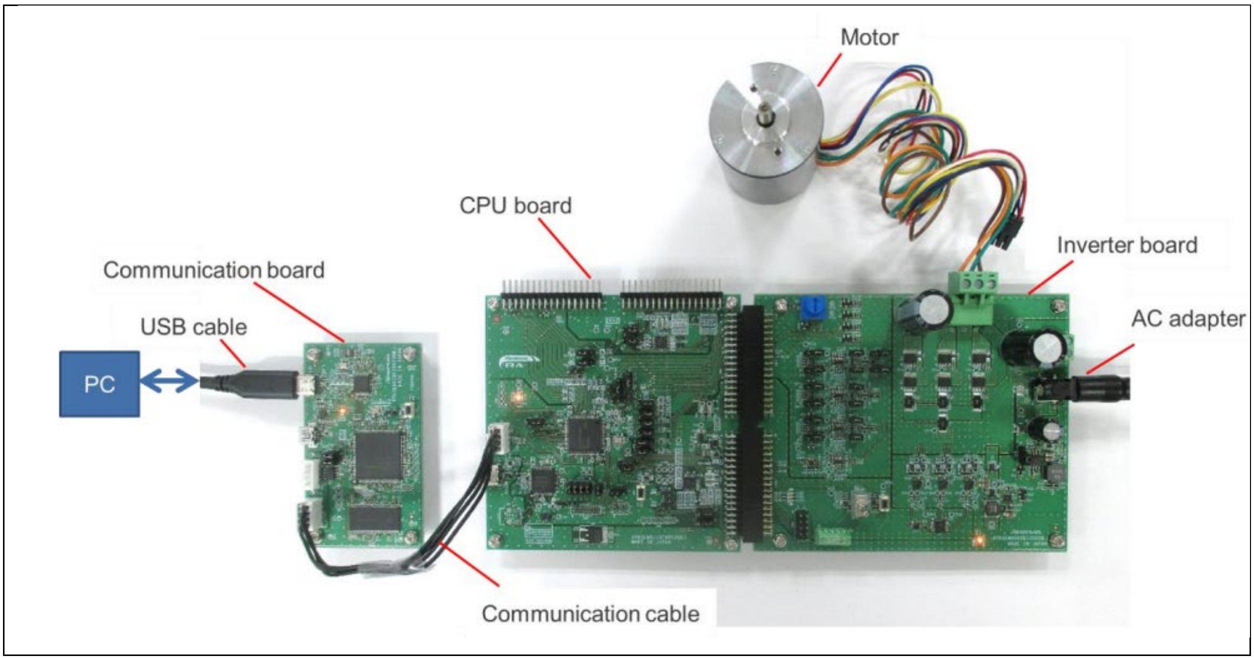


Figure 4-7 Connection example of kit

4.6 On board debugger

The RA series CPU boards include the circuit of “On board debugger J-Link OB (after here, JLOB). The update of program is performed through JLOB. When update, please connect CPU board and your PC via an USB cable.

4.7 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 4-8 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 4-9 shows an example of wiring between the inverter and the motor.

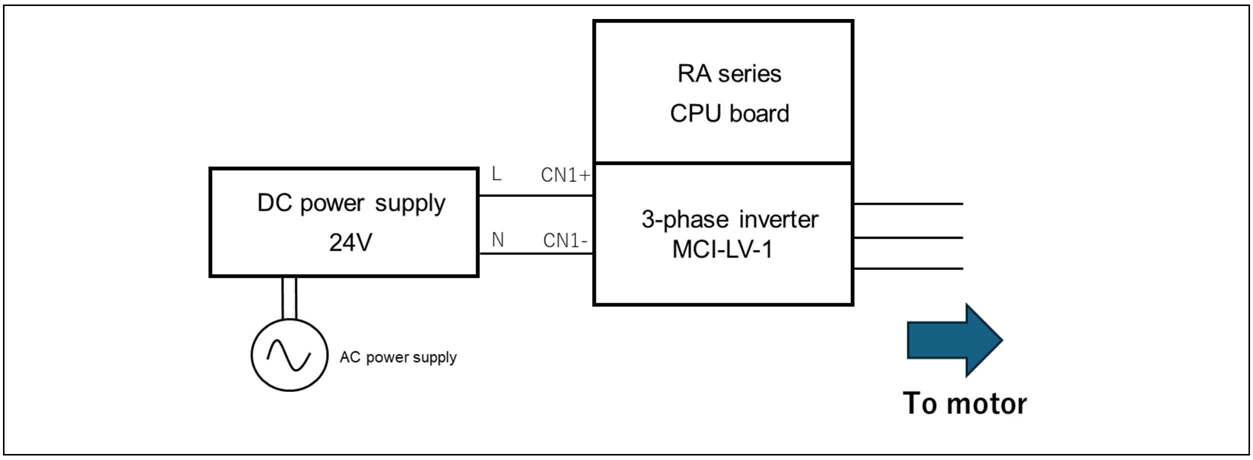


Figure 4-8 Wiring between the power supply and the inverter

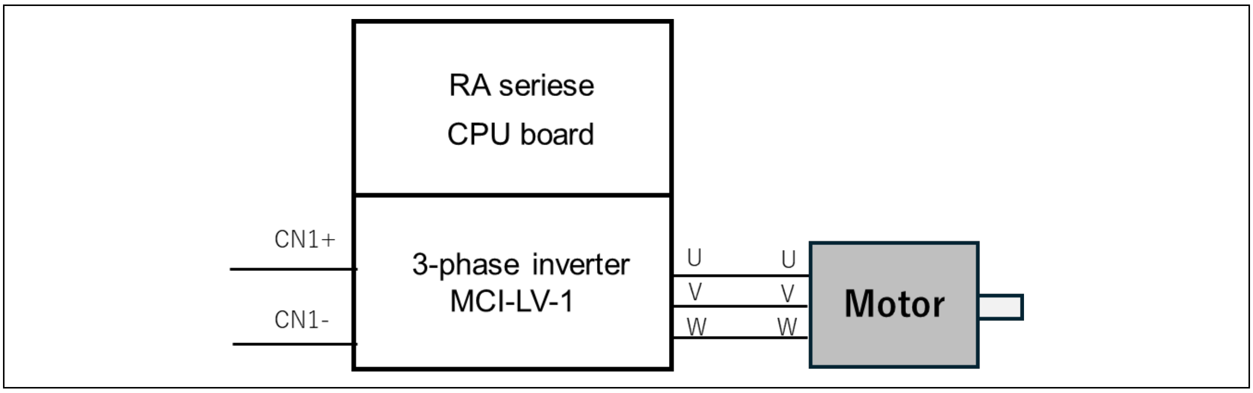


Figure 4-9 Wiring between the inverter and the motor

5. Environment to build the sample software

To develop the sample program in this application note, we will use e^2 studio with RA FSP as the development environment. Please download the installation environment from the following.

Download e^2 studio from the following URL:

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

For installation instructions, refer to the PDF manual included with e^2 studio.

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

How to upload the program to MCU is describes in detail on “6.3 Writing the sample program”.

6. Driving the motor

This chapter describes how to drive a motor. Please confirm “6.1 Precautions before driving the motor” and operate according to”6.2 Connecting”.

6.1 Precautions before driving the motor

When driving 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.
- 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 Connecting

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

(1) For writing
CPU board includes the circuit of JLOB, therefore USB cable can be connected directly, and the program can download without another environment. After download, please connect out the cable.

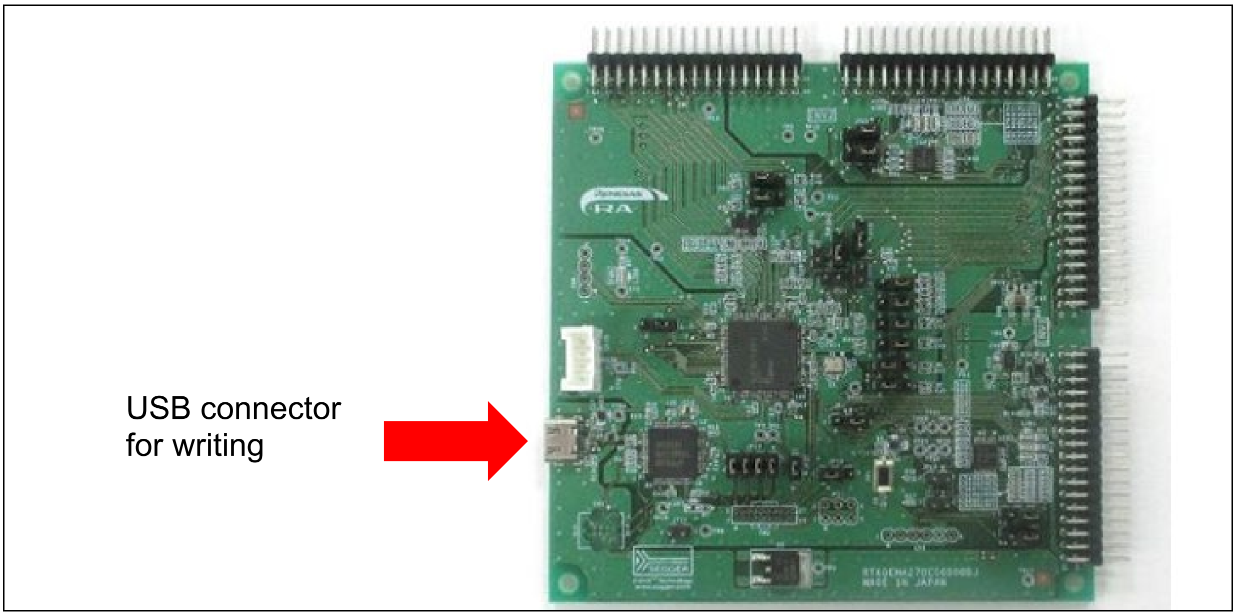


Figure 6-1 RA6T2 CPU board USB connector for writing

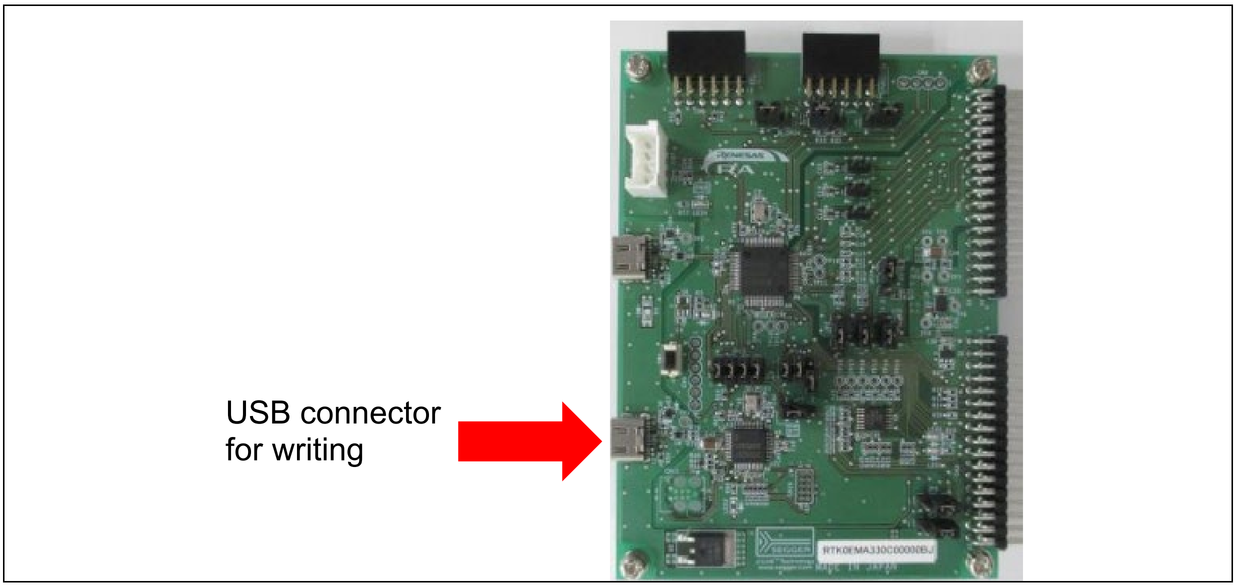


Figure 6-2 RA6T3 CPU board USB connector for writing

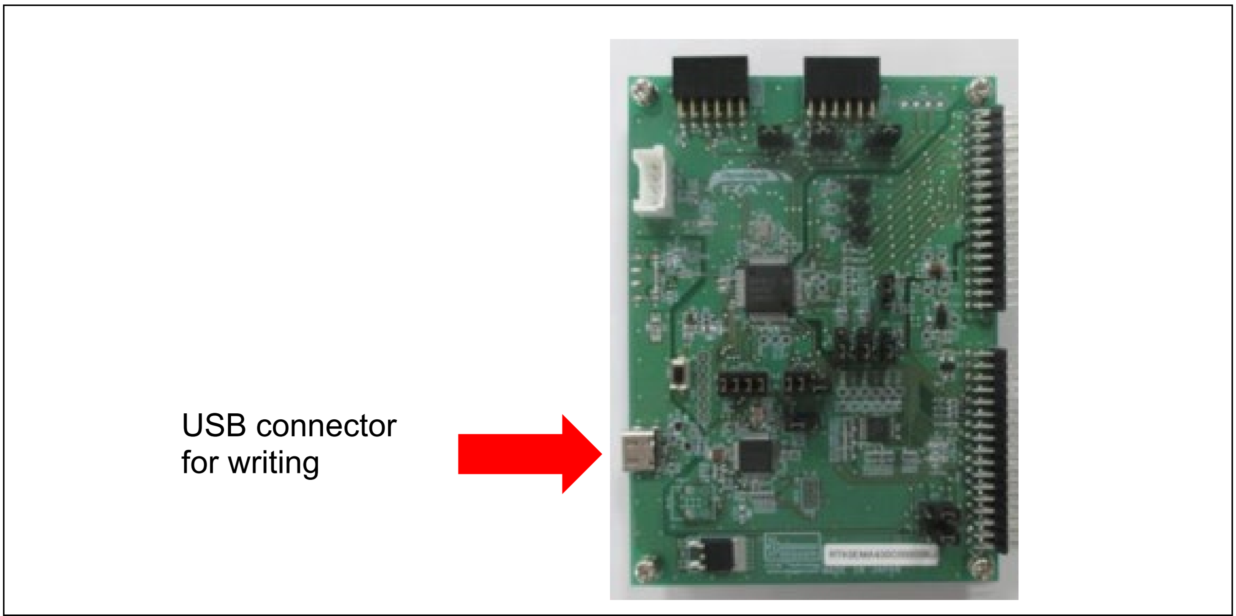


Figure 6-3 RA4T1 CPU board USB connector for writing

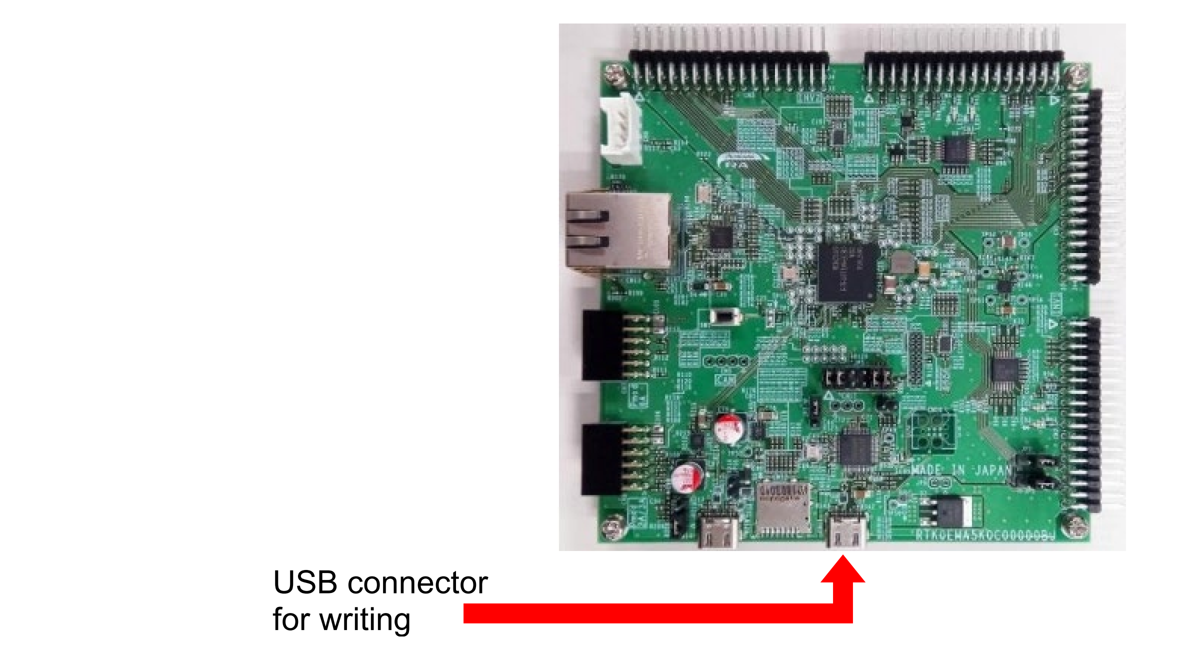


Figure 6-4 RA8T1 CPU board USB connector for writing

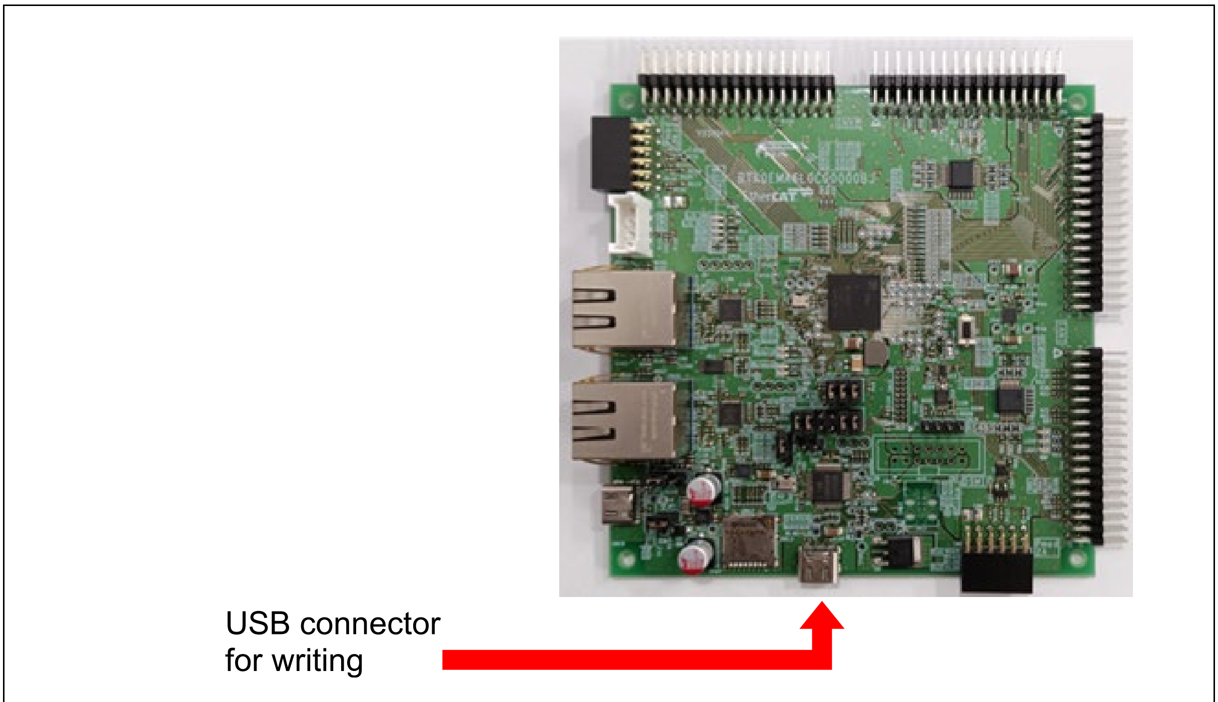


Figure 6-5 RA8T2 CPU board USB connector for writing

(2) For motor operation

Connect to the PC using MC-COM(RTK0EMXC90Z00000BJ) as shown in Figure 6-6. 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.

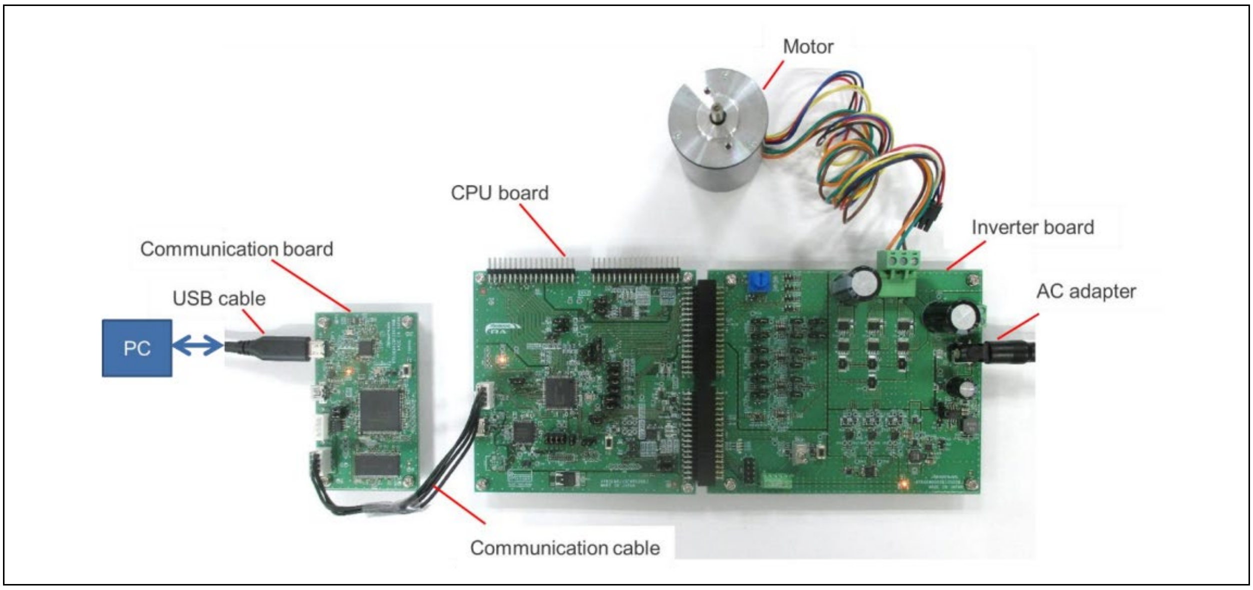


Figure 6-6 Example of connection at investigation

(3) Confirmation of jumper

Please confirm the jumper settings as shown below. After you used the environment for other type control, jumper position is different from below condition.

・Inverter board

Number of jumper	Connection
JP8	Connect 2-3
JP11	Connect 2-3

6.3 Writing the sample program

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

6.3.1 Install e^2 studio

Please download e^2 studio which supports FSP configurator from our below WEB site, and install to your PC.

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

6.3.2 Import a project

1. Left click “File” TAB.

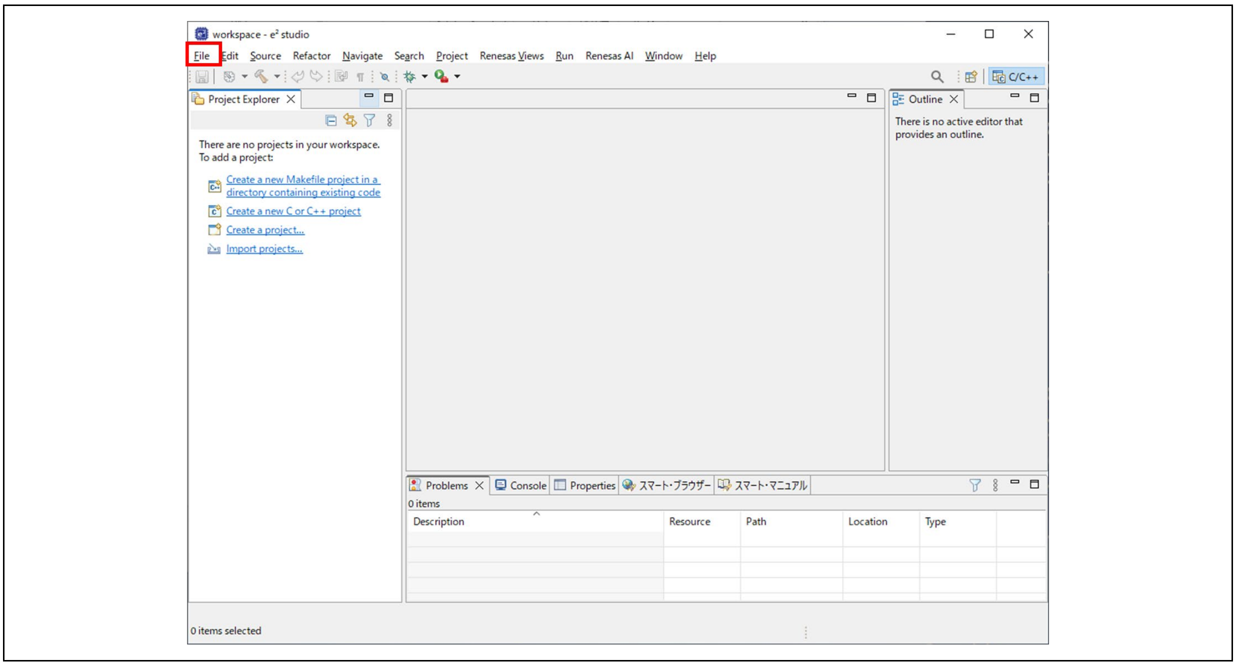


Figure 6-7 Selection of the target project (1)

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

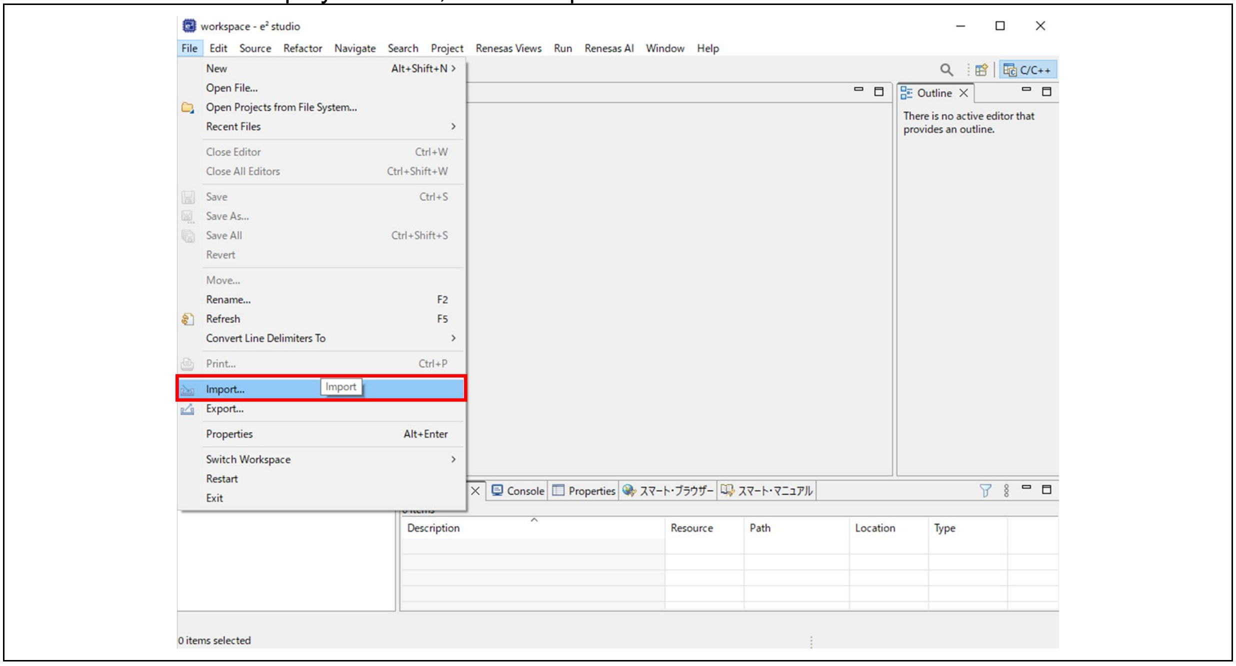


Figure 6-8 Selection of the target project (2)

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

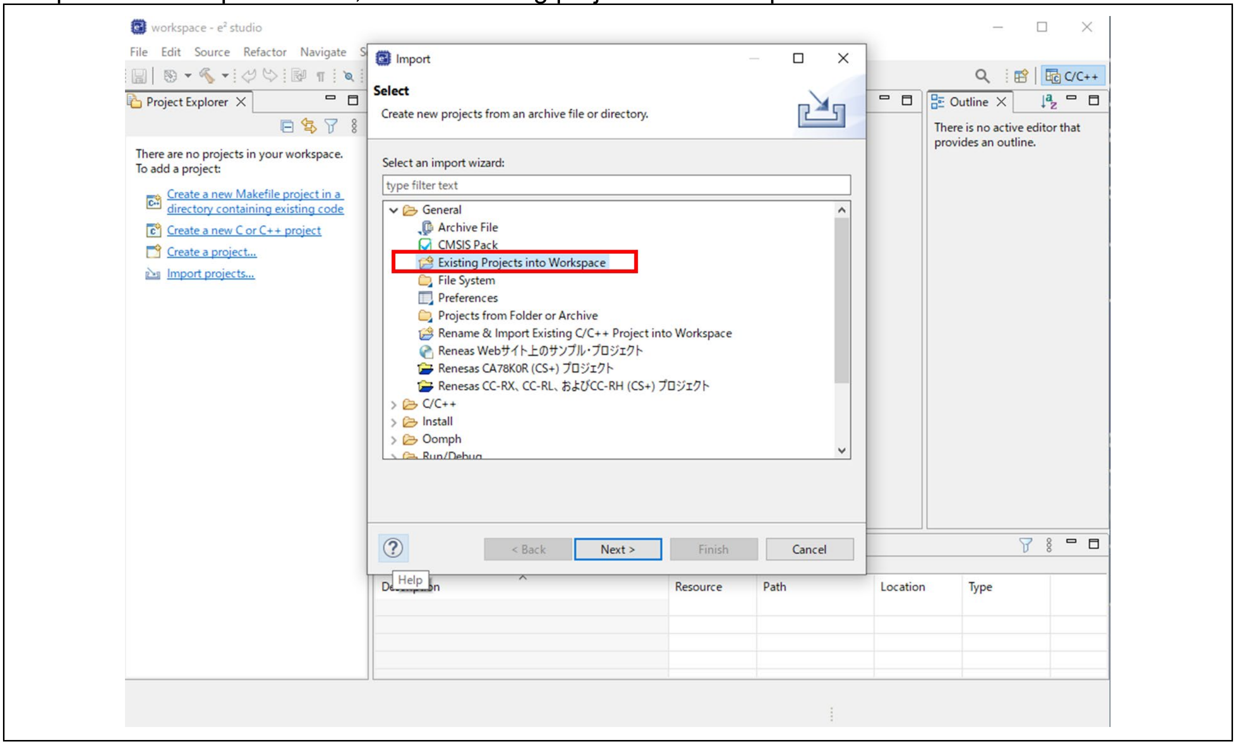


Figure 6-9 Selection of the target project (3)

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

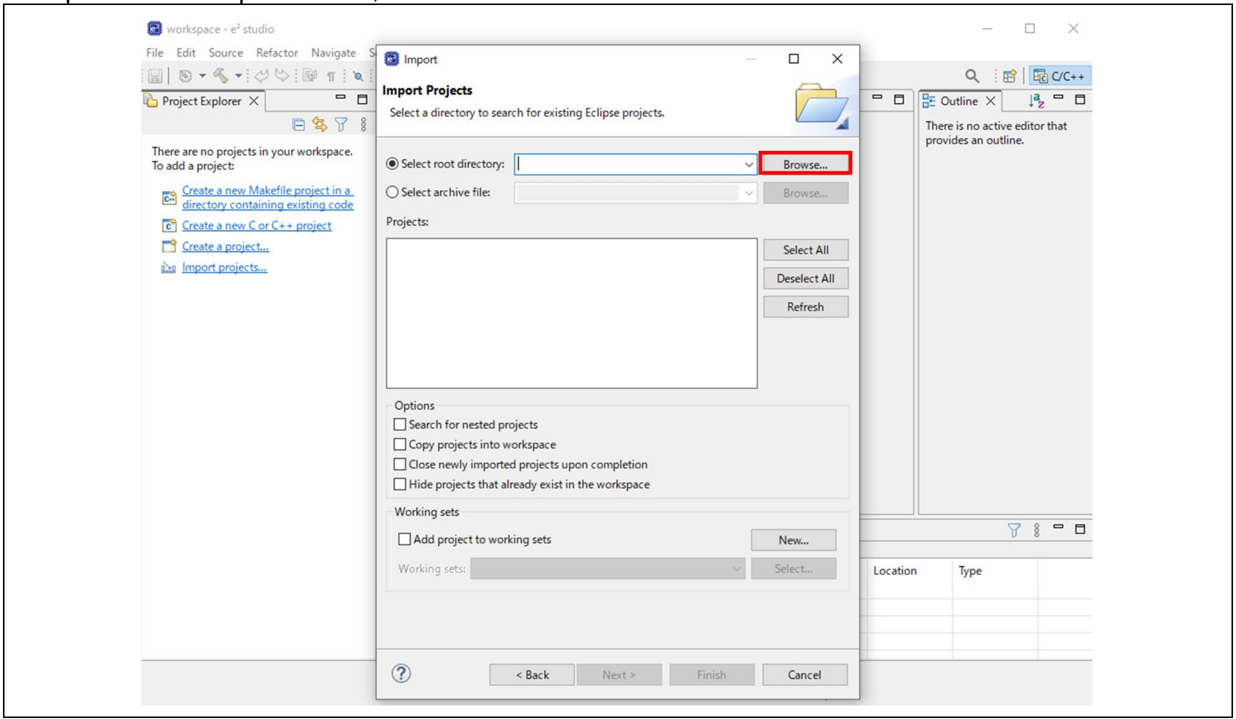


Figure 6-10 Selection of the target project (4)

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

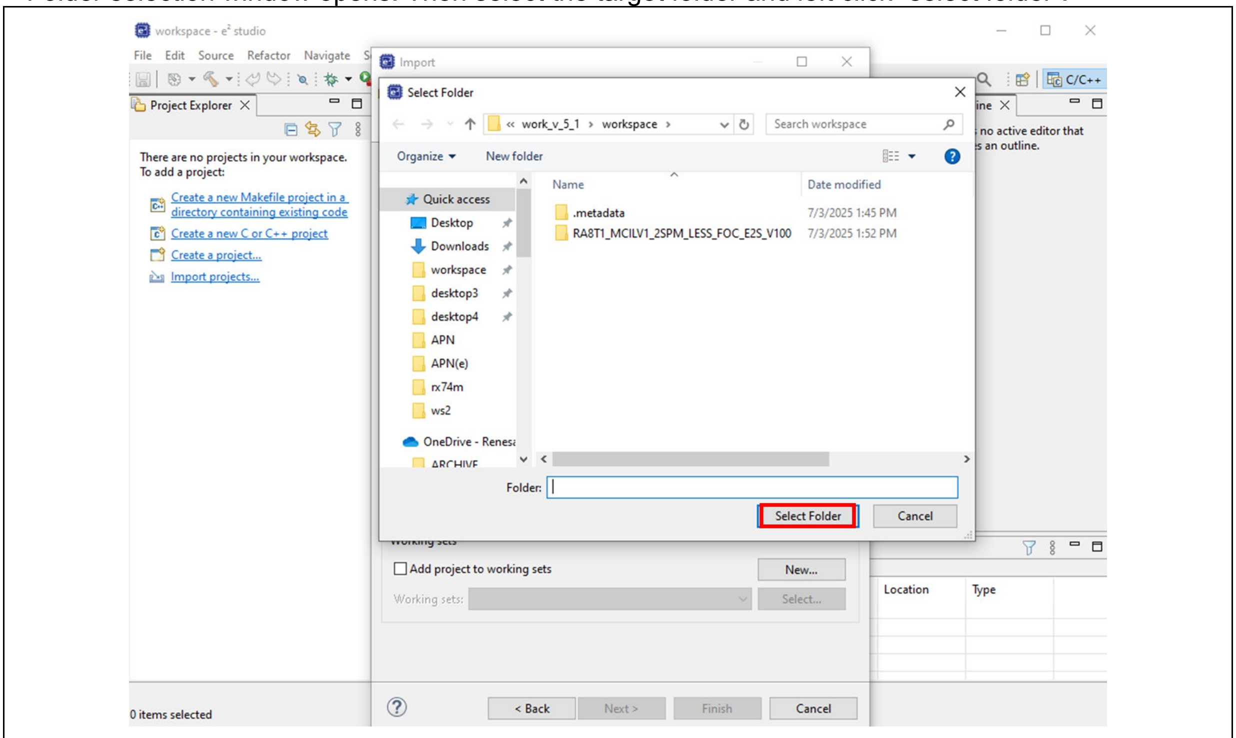


Figure 6-11 Selection of the target project (5)

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

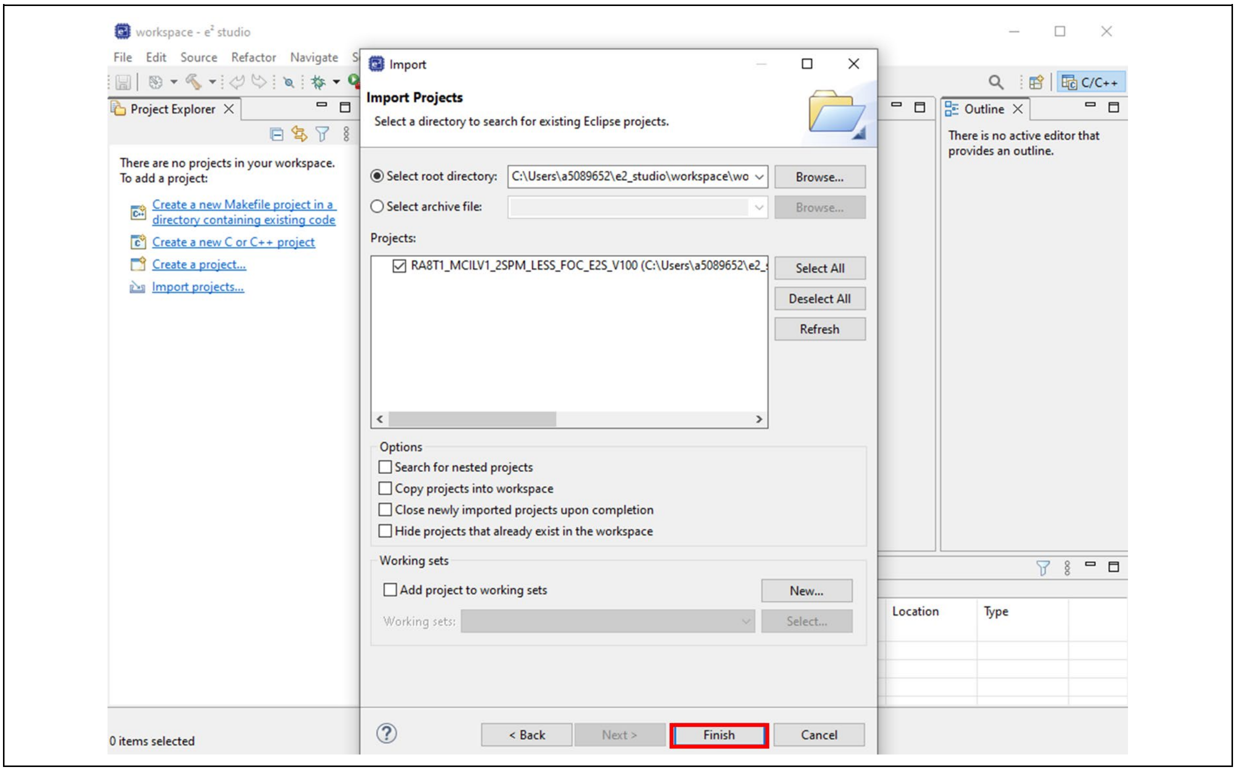


Figure 6-12 Selection of the target project (6)

7. Confirm the target project is imported correctly into e^2 studio.

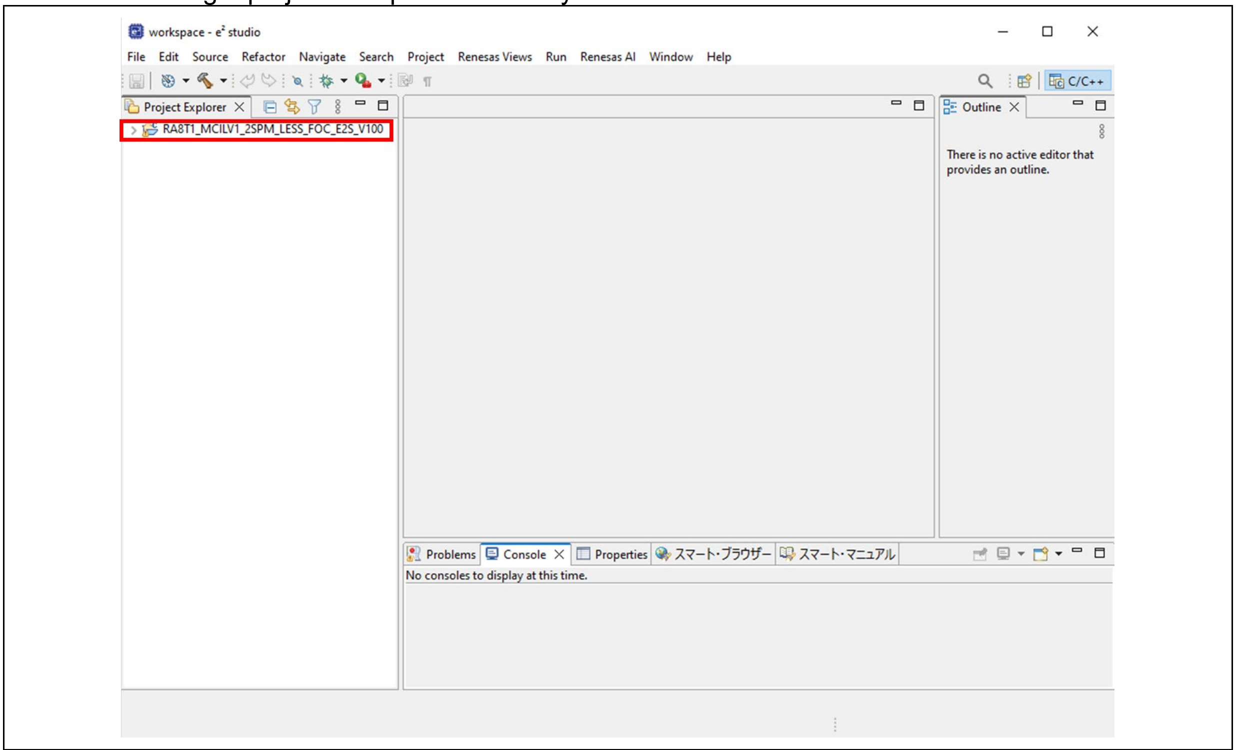


Figure 6-13 Confirmation of import the target project

6.3.3 Build the target project

1. Right click the imported target project in e[2] studio window.

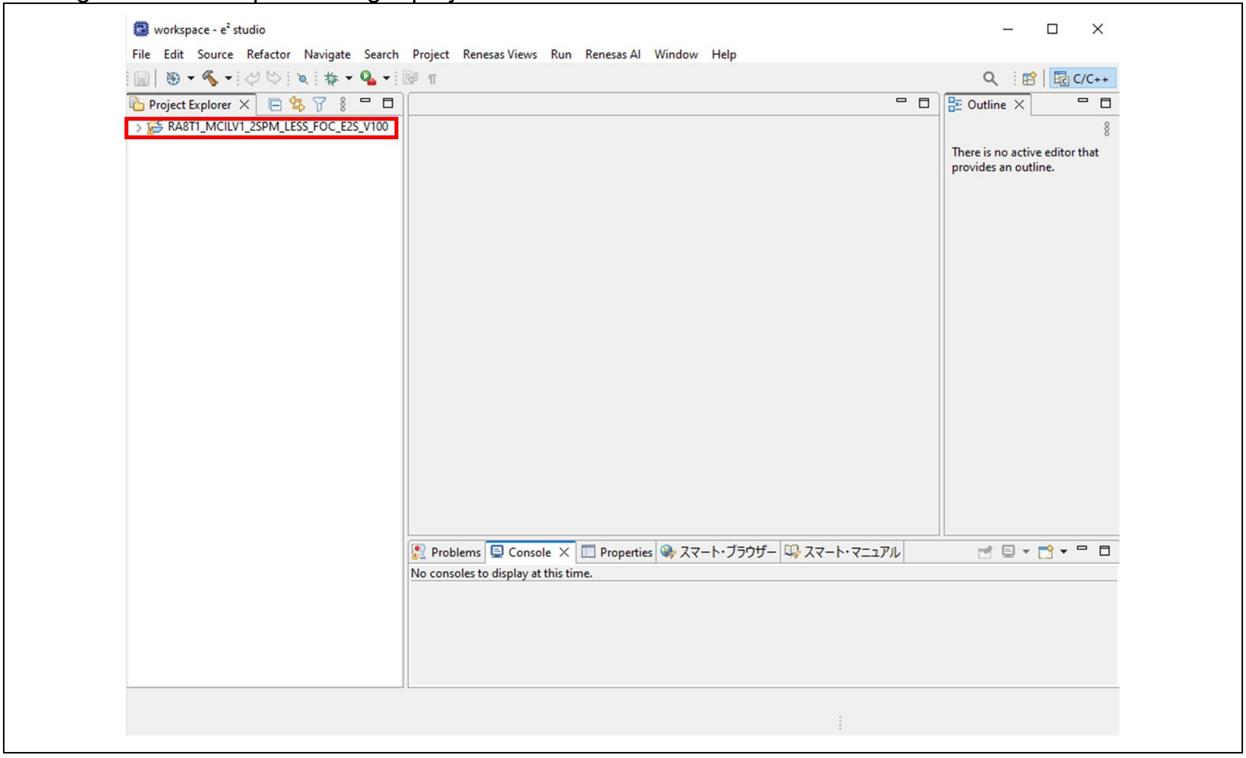


Figure 6-14 Selection of the target project

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

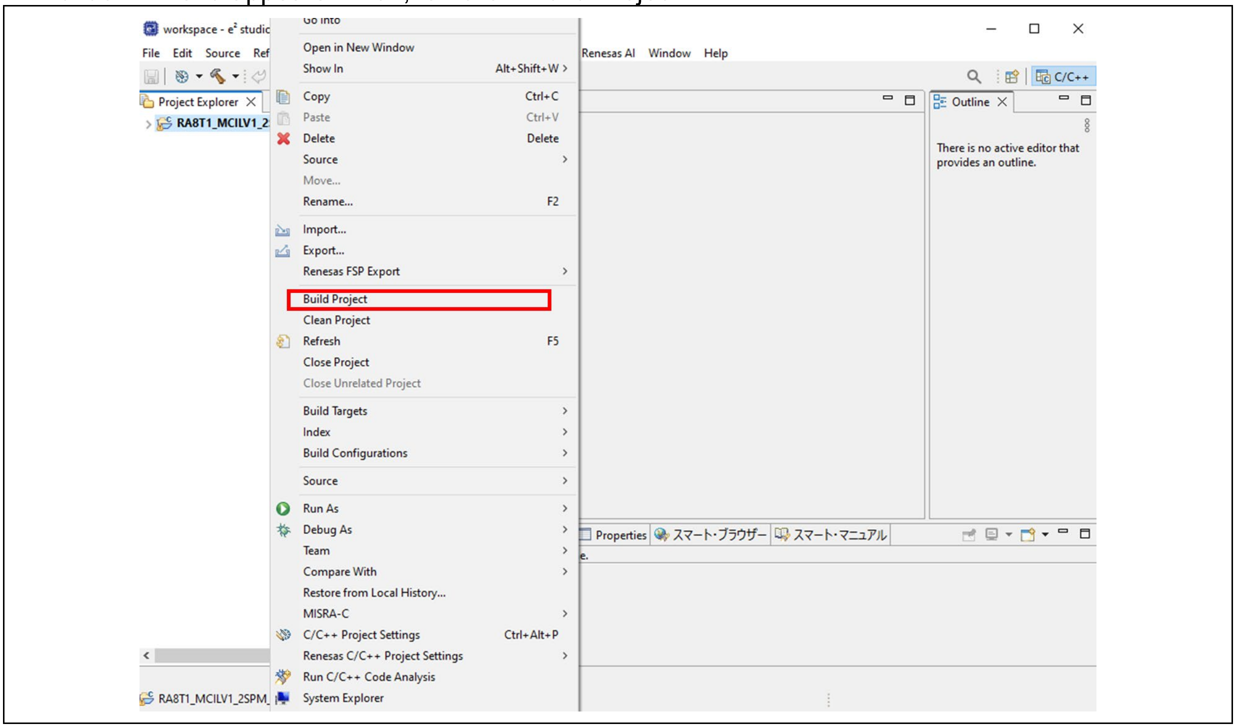


Figure 6-15 Pulldown menu

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

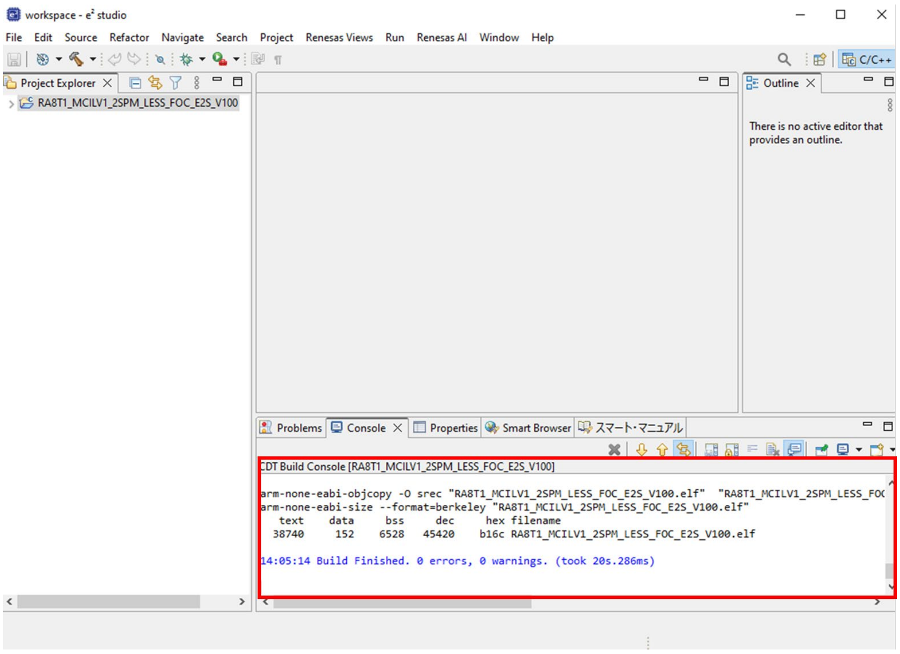


Figure 6-16 Confirmation of finish of build

6.3.4 Connection between PC and the target CPU board via an USB cable

Please connect PC and CPU board via an USB cable like below.

(In below figure, target board is RA6T2.)

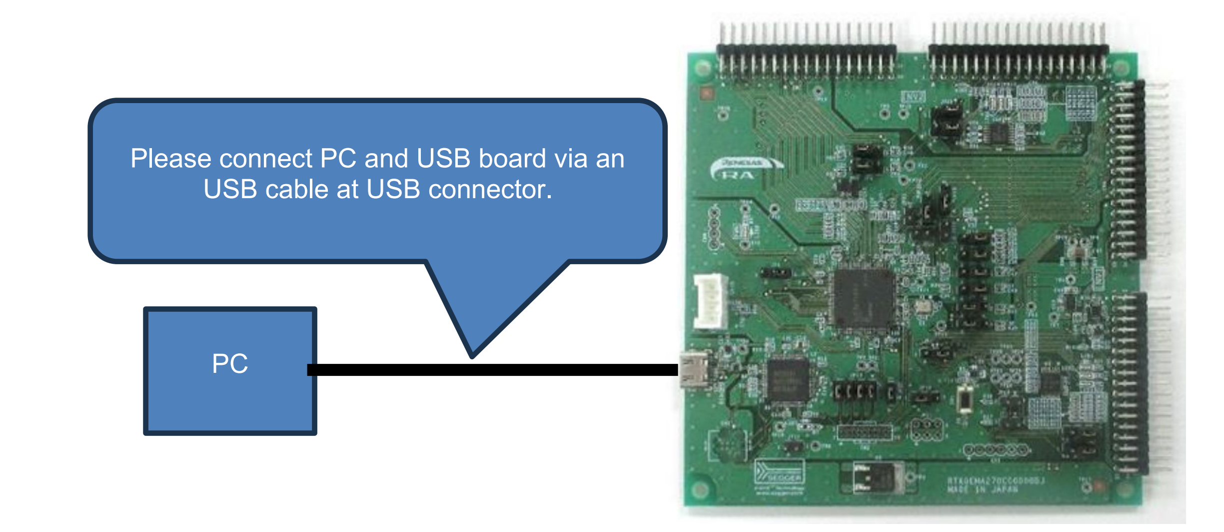


Figure 6-17 Confirmation between PC and CPU board (RA6T2)

6.3.5 Writing to the target board (with built program)

1. Select target project and right click.

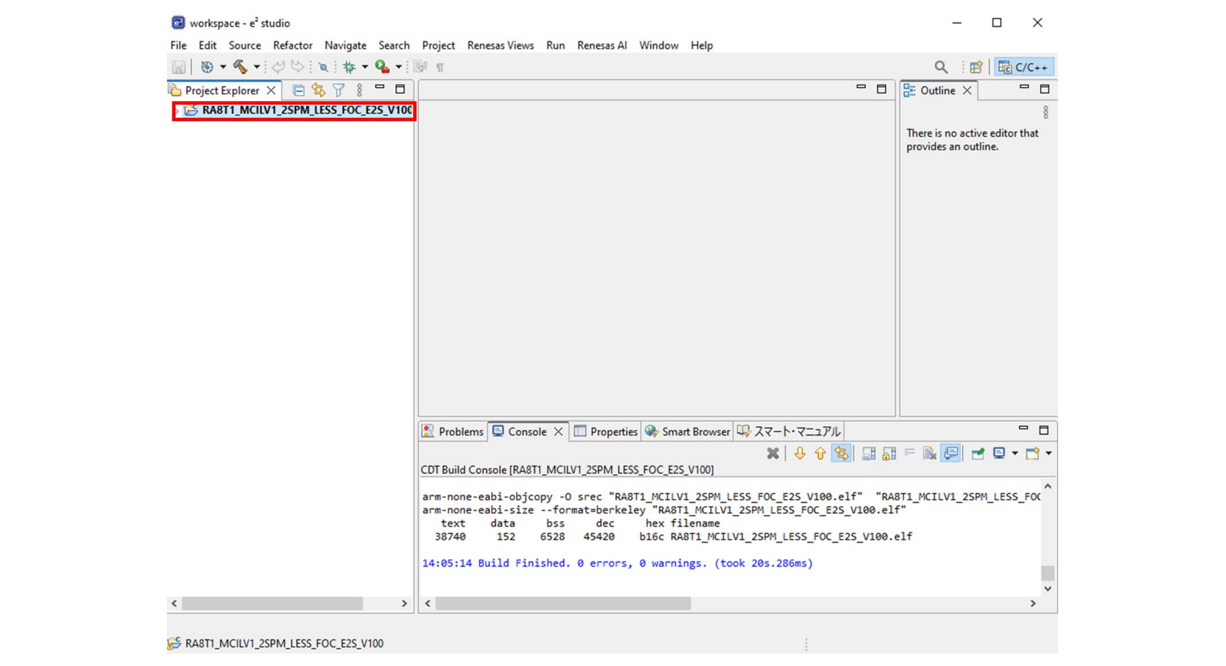


Figure 6-18 Selection of target project

2. Pull down menu display. Put mouse cursor on “Debug As”, after that new list window appears. Select “Renesas GDB Hardware Debugging” and left click.

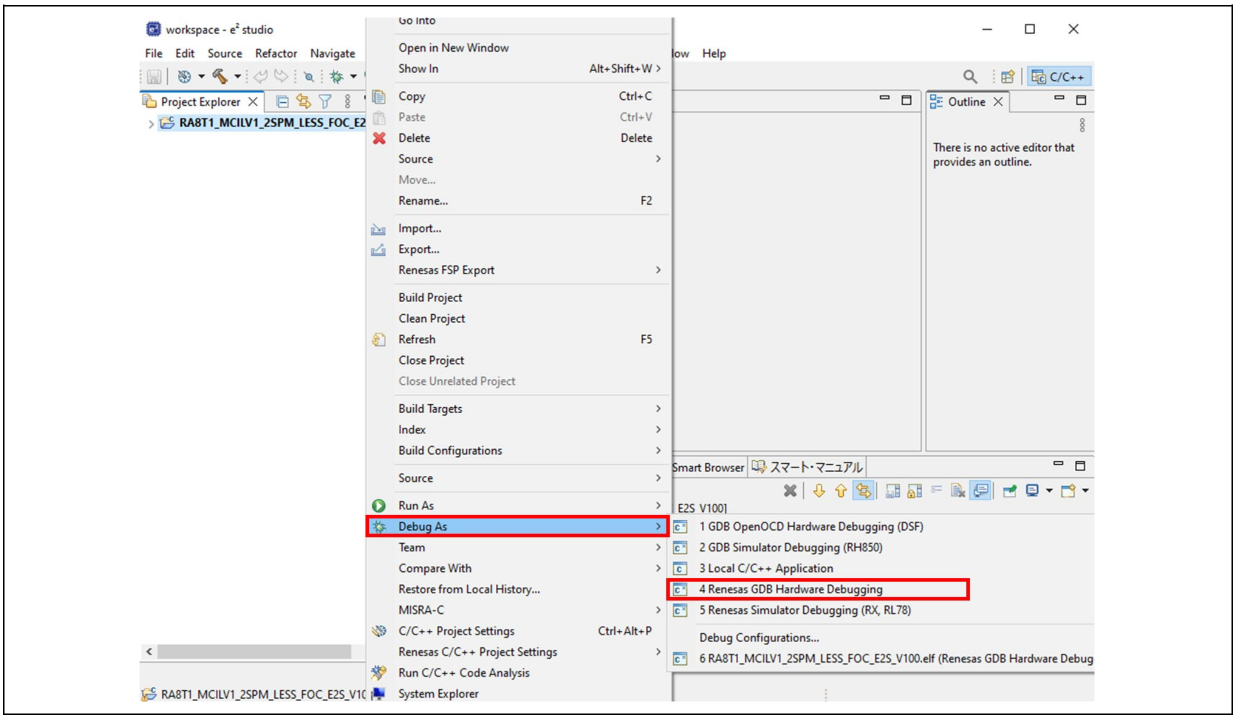


Figure 6-19 Selection of debug method

3. If the connection is correct and program is downloaded successfully, the display changes to “Debug” mode like below.

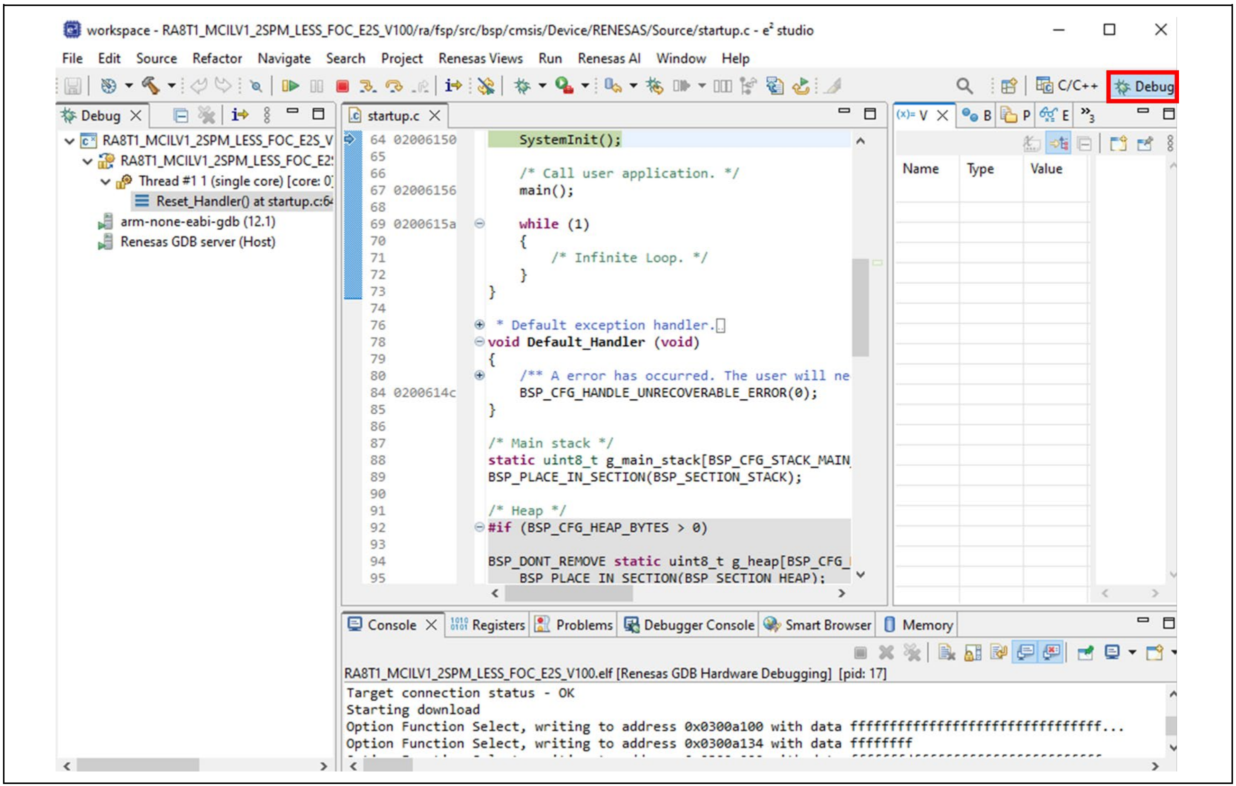


Figure 6-20 “Debug” mode

4. Left click “■“ in debug mode to disconnect from target board. After that disconnect USB cable. All process is finished.

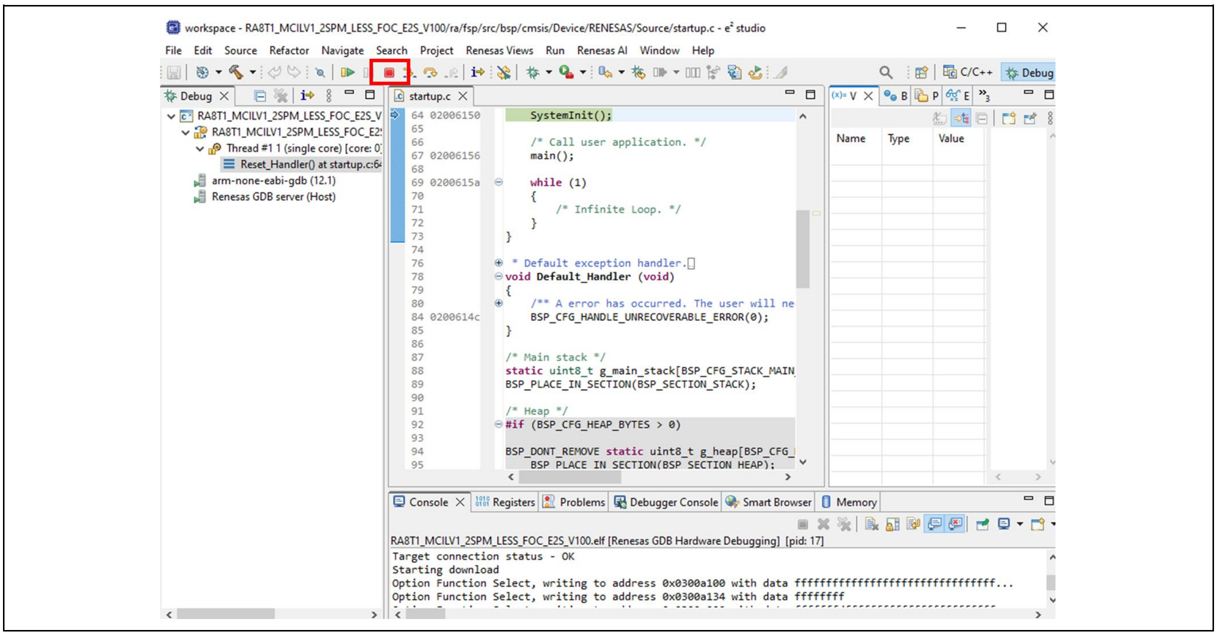


Figure 6-21 Disconnect from target board

6.4 Introducing RMW

Renesas Motor Workbench, a motor control development support tool, is used as an 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

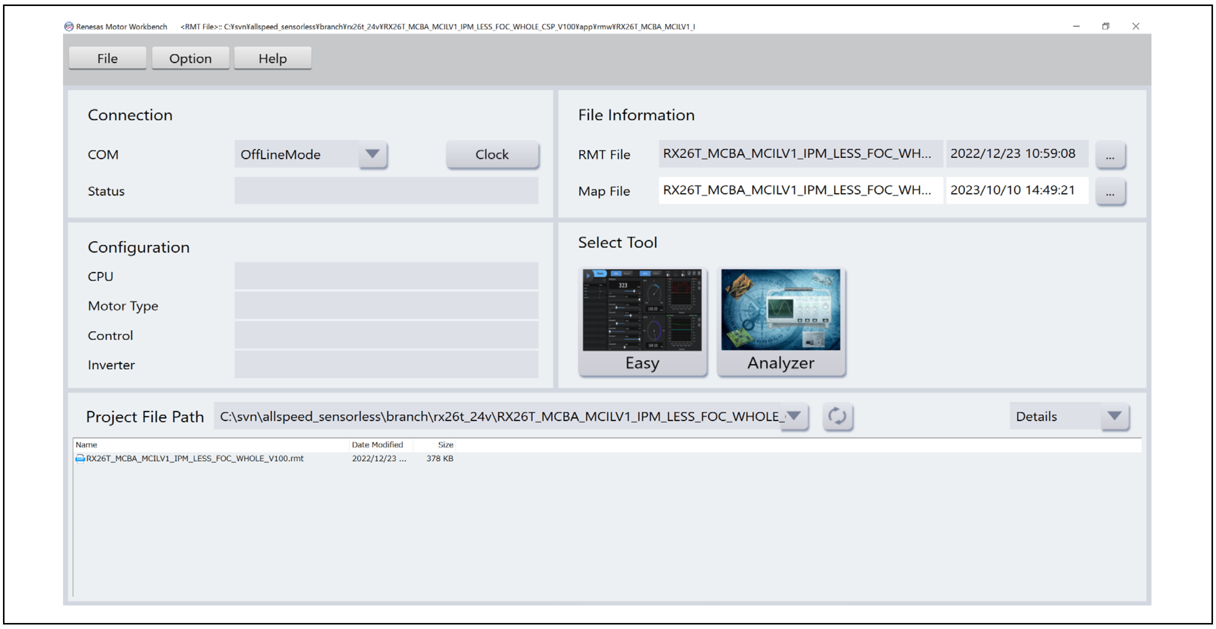


Figure 6-22 Window of Renesas Motor Workbench

6.5 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.

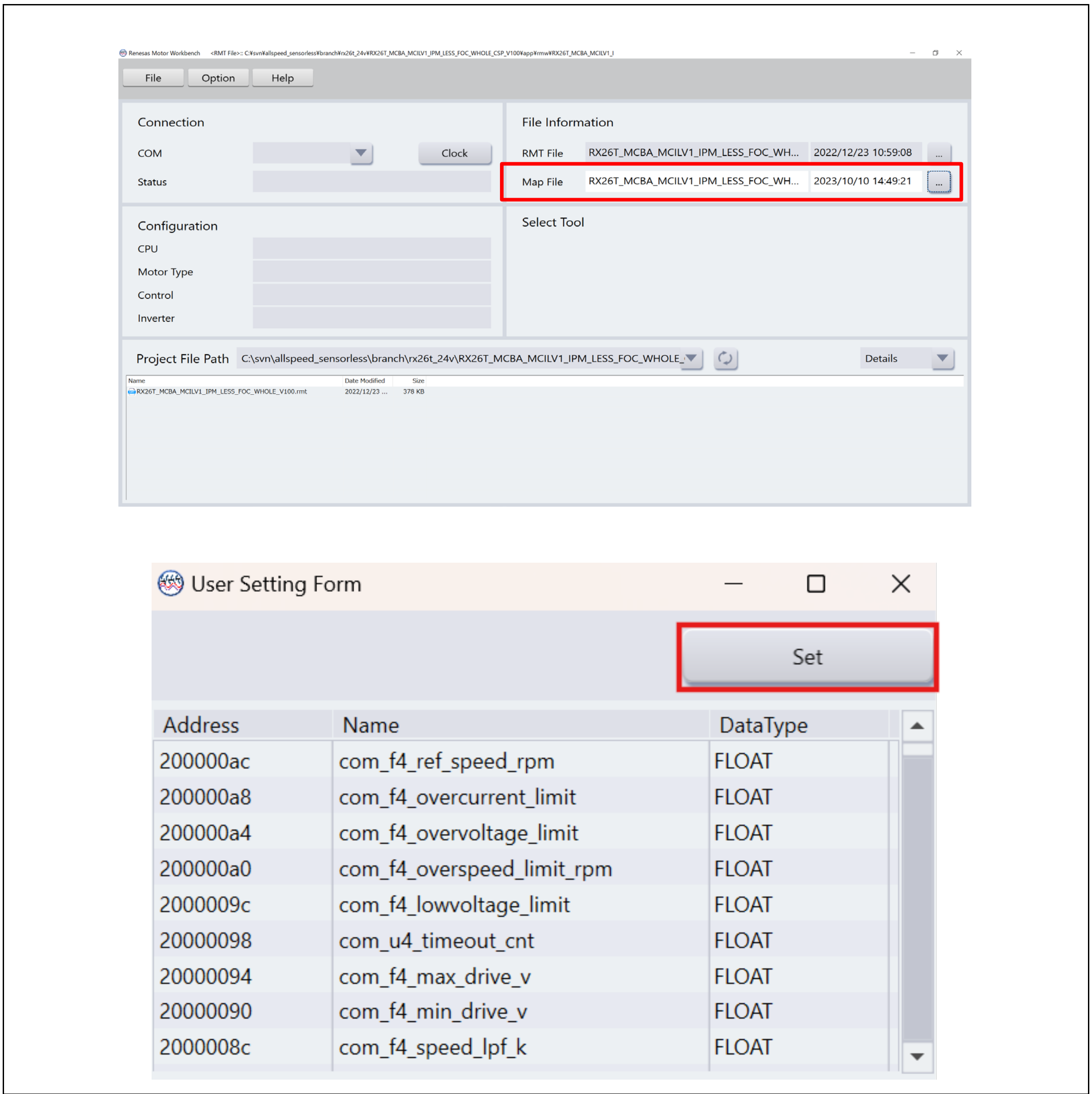


Figure 6-23 Map file registration setup window (upper) and confirmation window (lower) in RMW

6.6 Variables used for working with RMW

In this sample program, the motor can be controlled using RMW. Table 6-1 lists the data input variables that are used when the RMW UI is used. The values input to these variables are applied to the corresponding variables in the motor module and then used for controlling the motor if the value written to the com_u1_enable_write variable is the same as the value of the g_u1_enable_write variable. Note, however, that the variables indicated by an asterisk (*) do not depend on the value of the com_u1_enable_write variable.

Some parameters used for motor control can be changed while the motor is stopped. For details, see Table 9-8 and Table 9-9.

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

Table 6-1 List of main input variables for Analyzer functions

Analyzer function input variable name	Type	Description
com_u1_mode_system (*)	uint8_t	Managing the state 0: Stop mode 1: Run mode 3: Reset
com_f4_ref_speed_rpm (*)	float	Speed command value (mechanical) [rpm]
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 6-2 lists main variables that are often observed when driving under speed control is evaluated. Use this table for reference when the waveform is to be displayed, or the values of variables are to be loaded with an Analyzer function.

Table 6-2 List of main variables for sensorless vector control

Sensorless vector control variable name	Type	Substance
g_f4_id_ref_monitor	float	d-axis current reference [A]
g_f4_id_ad_monitor	float	d-axis measured current [A]
g_f4_iq_ref_monitor	float	q-axis current reference [A]
g_f4_iq_ad_monitor	float	q-axis measured current [A]
g_f4_iu_ad_monitor	float	U phase measured current [A]
g_f4_iv_ad_monitor	float	V phase measured current [A]
g_f4_iw_ad_monitor	float	W phase measured current [A]
g_f4_vdc_ad_monitor	float	Measured inverter bus voltage [V]
g_f4_vd_ref_monitor	float	d-axis voltage reference [V]
g_f4_vq_ref_monitor	float	q-axis voltage reference [V]
g_f4_refu_monitor	float	U phase voltage reference [V]
g_f4_refv_monitor	float	V phase voltage reference [V]
g_f4_refw_monitor	float	W phase voltage reference [V]
g_f4_angle_rad_monitor	float	Rotor angle (electrical) [rad]
g_f4_speed_est_monitor	float	Rotation speed (electrical) [rad/s]
g_f4_speed_ref_monitor	float	Speed reference (electrical) [rad/s]
g_f4_speed_rpm_monitor	float	Rotation speed (mechanical) [rpm]

6.7 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_mode_system" 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_mode_system 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_mode_system" row, in the [Write] column, enter "1".

(6) Click the [Write] button.

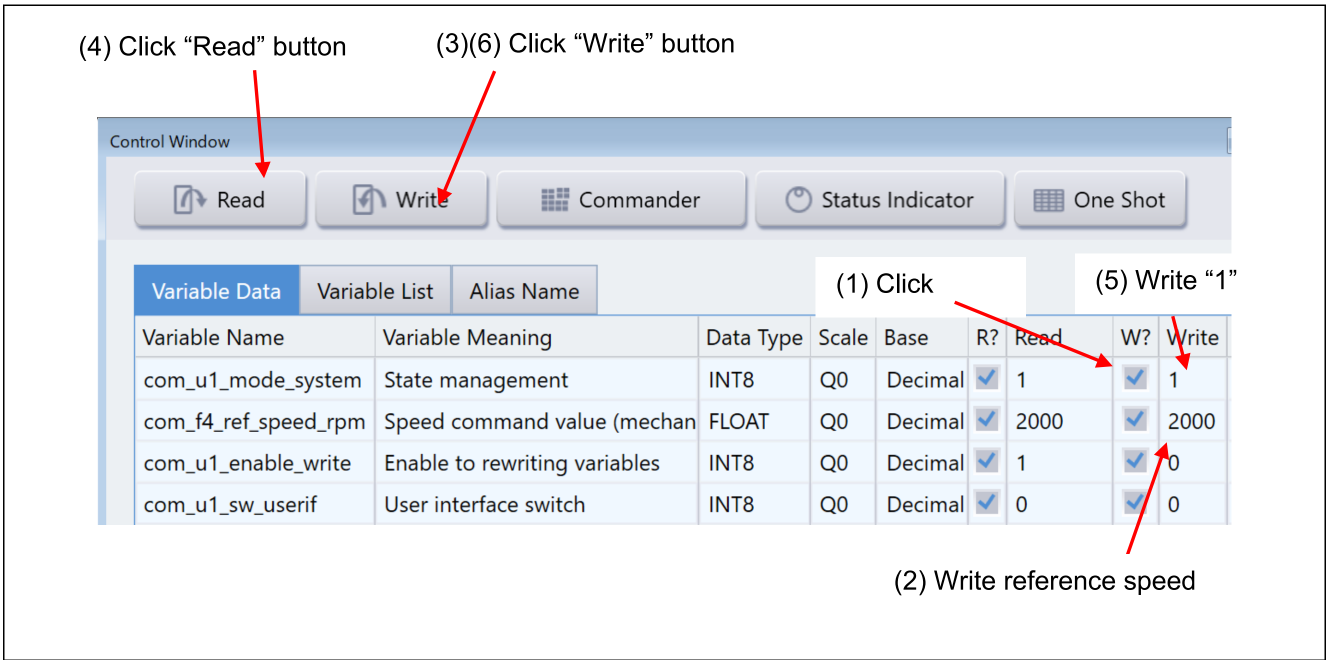


Figure 6-24 Procedure for driving of the motor

b) Stop the motor
   (1) On the "com_u1_mode_system" row, in the [Write] column, enter "0".
   (2) Click the [Write] button.
   (3) Confirm that the motor has stopped.

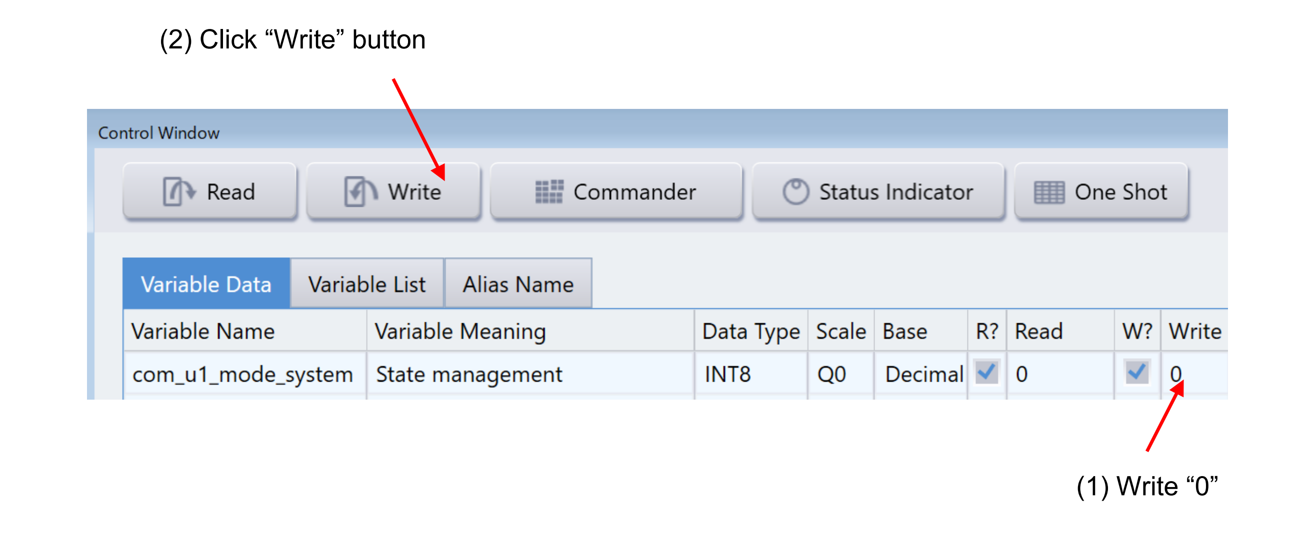


Figure 6-25 Procedure for stopping the motor

c) What to do in case of motor stop (due to an error)
   (1) On the "com_u1_mode_system" row, in the [Write] column, enter "3".
   (2) Click the [Write] button.

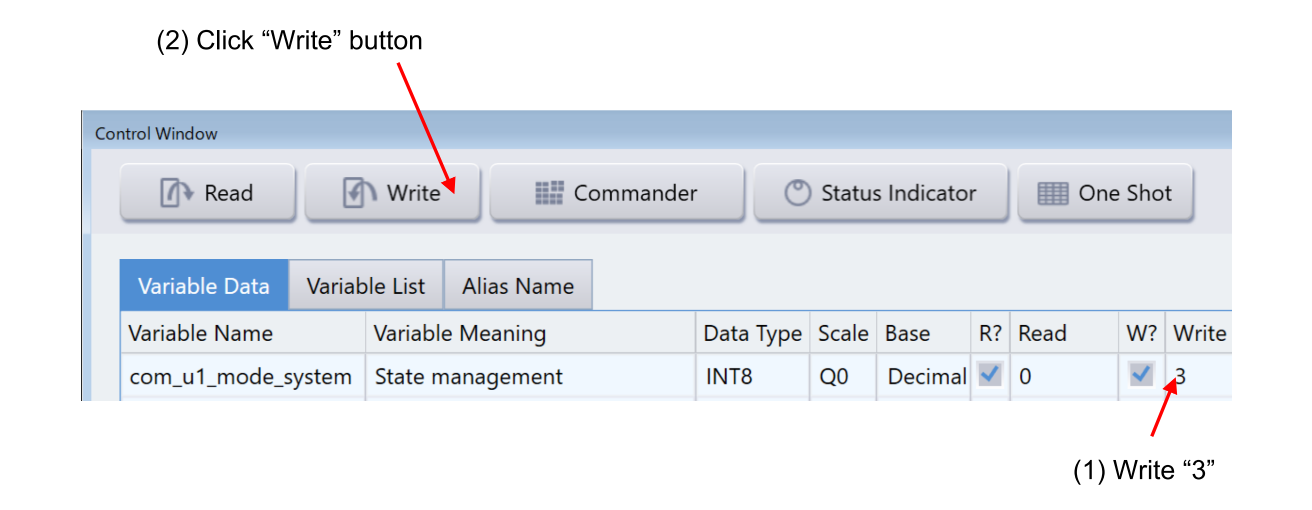


Figure 6-26 Procedure for handling an error

Table 6-3 Error status description

Value	Error Description	Assigned macro name
0x0000	No error	MOTOR_ERROR_NONE
0x0001	Hardware overcurrent error	MOTOR_ERROR_OVER_CURRENT_HW
0x0002	Overvoltage error	MOTOR_ERROR_OVER_VOLTAGE
0x0004	Overspeed error	MOTOR_ERROR_OVER_SPEED
0x0008	Hall signal timeout error (Not happen in sensorless vector)	MOTOR_ERROR_HALL_TIMEOUT
0x0010	BEMF signal timeout error (Not happen in sensorless vector)	MOTOR_ERROR_BEMF_TIMEOUT
0x0020	Unused	MOTOR_ERROR_HALL_PATTERN
0x0040	BEMF signal pattern error (Not happen in sensorless vector)	MOTOR_ERROR_BEMF_PATTERN
0x0080	Low-voltage error	MOTOR_ERROR_LOW_VOLTAGE
0x0100	Software overcurrent error	MOTOR_ERROR_OVER_CURRENT_SW
0x0200	Induction sensor error in calibration (Not happen in sensorless vector)	MOTOR_ERROR_INDUCTION_CORRECT
0xFFFF	Undefined error	MOTOR_ERROR_UNKNOWN

6.8 Stopping and shutting down the motor

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

(1) Perform the procedure for stopping the motor described in 6.7b).

(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 (Sinusoidal modulation can also be selected)
Position and speed estimation method	BEMF observer
Control mode	Only speed control
Compensation function	• Voltage error compensation • Voltage phase lead compensation • Decoupling control

7.2 Control block diagram

The block diagram of sensorless vector control is shown below.

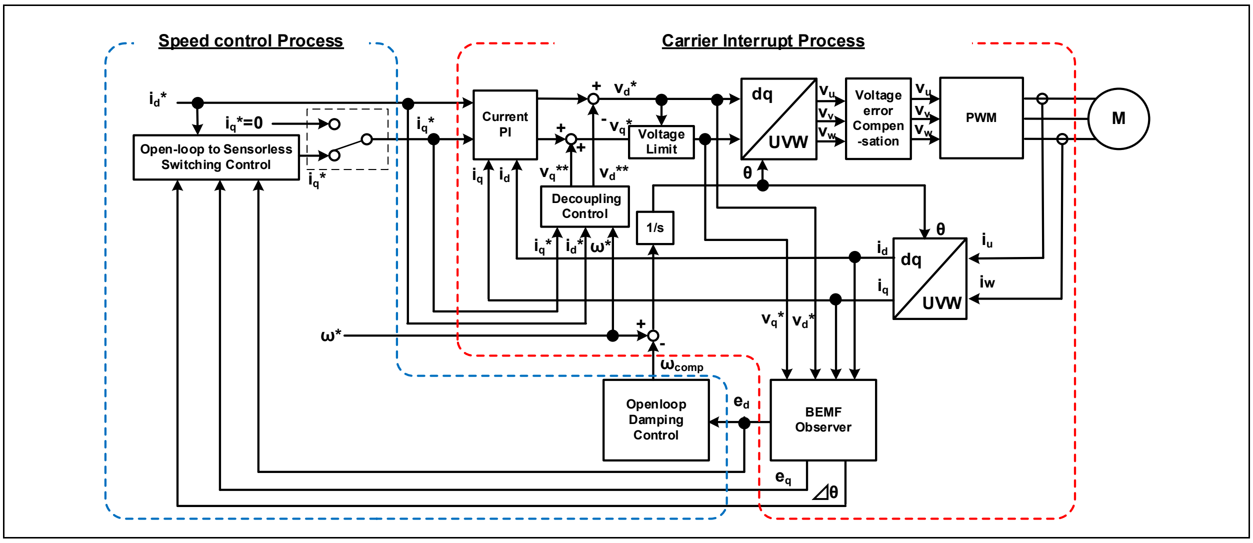


Figure 7-1 Block Diagram of Sensorless Vector Control (Open loop Control)

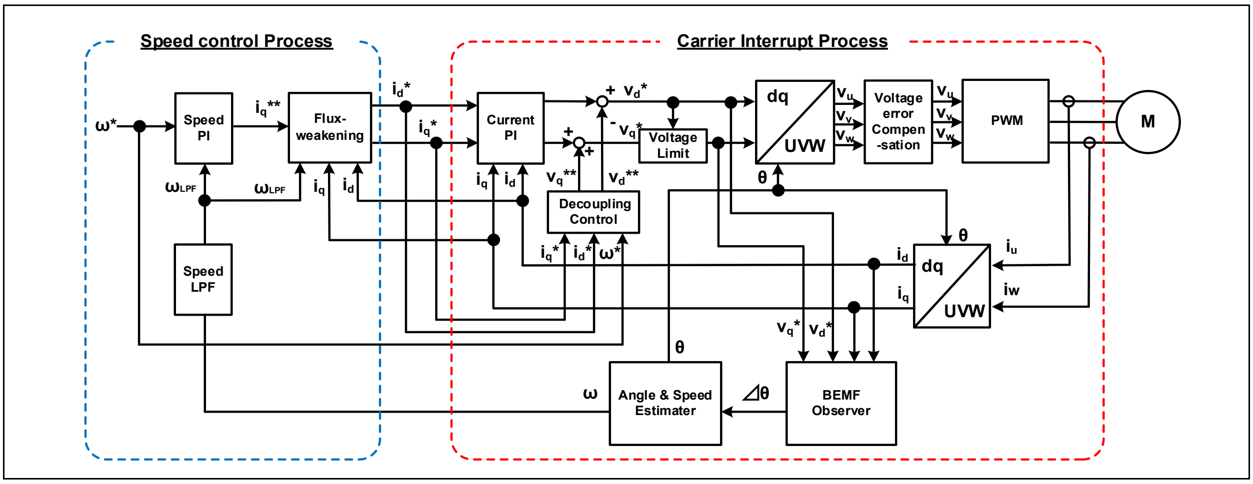


Figure 7-2 Block Diagram of Sensorless Vector Control (Sensorless Control)

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.

The estimated speed value is output from the estimated speed passed though the LPF.

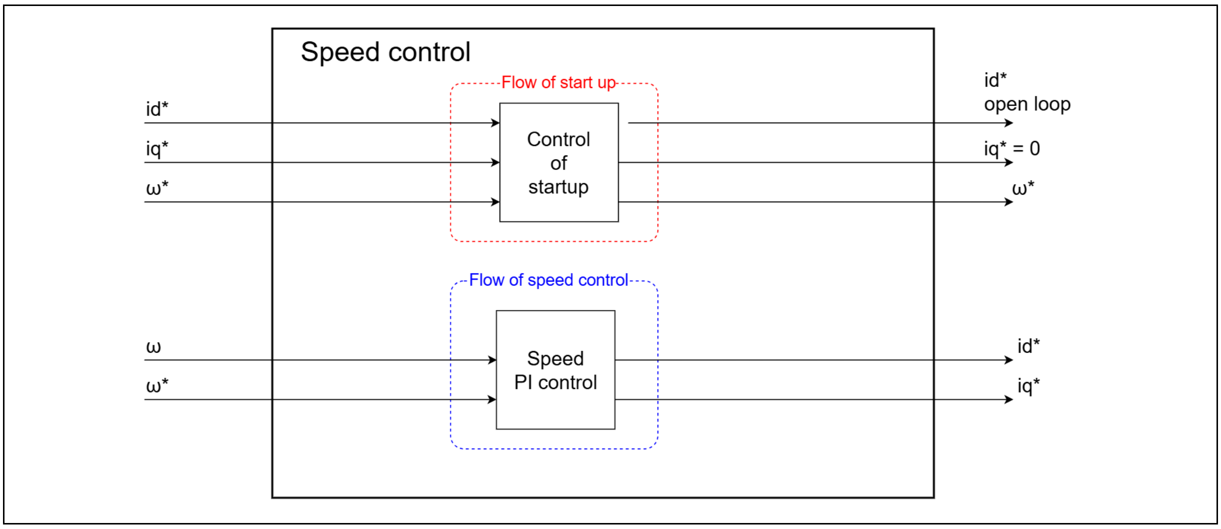


Figure 7-3 Functional block diagram for speed control

7.4 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.

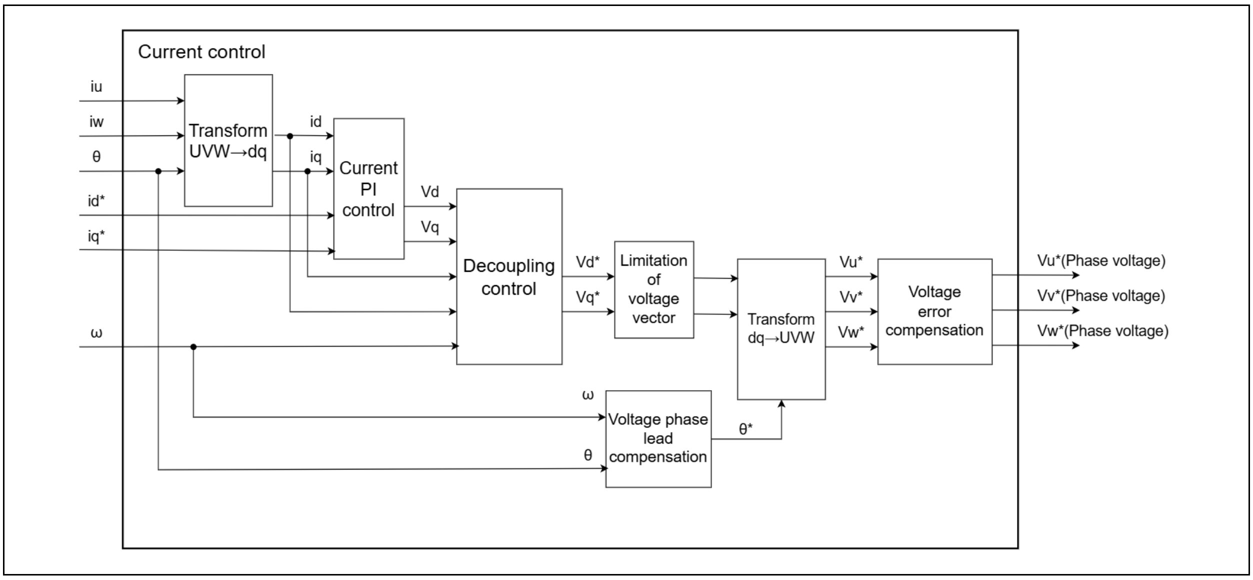


Figure 7-4 Functional block diagram for current control

7.5 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 [ohm], 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.

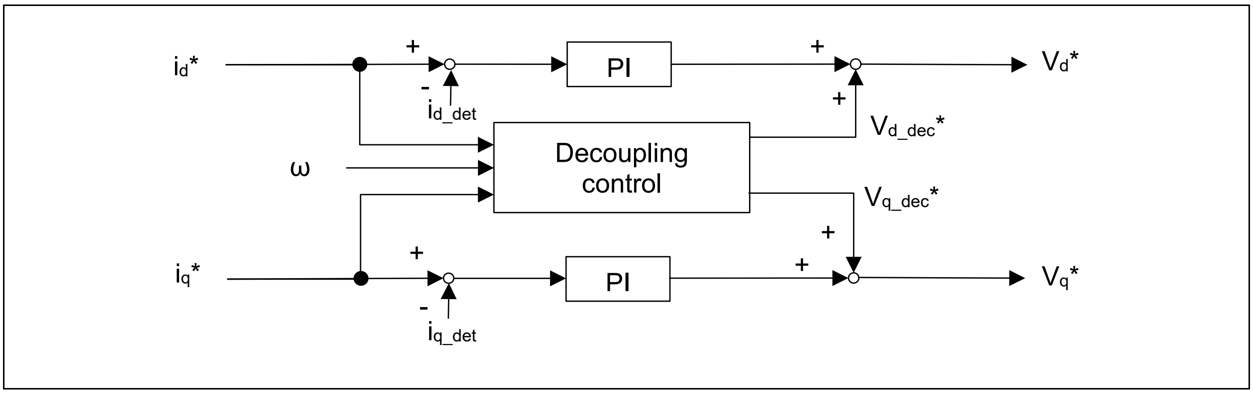


Figure 7-5 Functional block diagram for decoupling control

7.6 Sensorless function

When PI control is implemented, the motor is controlled by sensorless vector control using a BEMF observer.

When an inductive voltage observer is used, the observer is used to estimate the inductive voltage. Then the phase error between the estimated d/q axis and the actual d/q axis is calculated to obtain the position and speed.

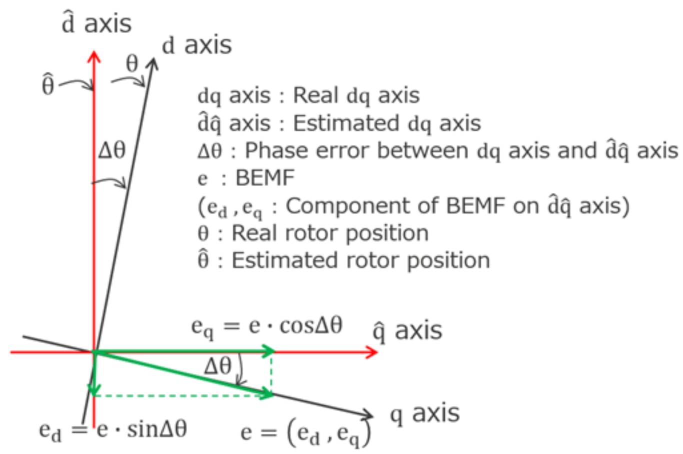


Figure 7-6 Induced voltage at estimated d/q-axis

From Figure 7-6, the voltage equations of the estimated d and q axes can be represented as follows:

$$v_d^* = (R + sL_d)i_d - \omega^* L_q i_q + e_d$$
$$v_q^* = (R + sL_q)i_q + \omega^* L_d i_d + e_q$$

Here, $-\omega^* L_q i_q + e_d$ and $\omega^* L_d i_d + e_q$ are treated as voltage disturbance and put to $-d_d, -d_q$.

$$v_d^* = (R + sL_d)i_d - d_d$$
$$v_q^* = (R + sL_q)i_q - d_q$$

An estimate equation for the d-axis inductive voltage is derived first. Rearrange the d-axis voltage equation as follows:

$$si_d = \frac{v_d^*}{L_d} - \frac{R}{L_d} i_d + \frac{d_d}{L_d}$$

Based on the above equation, an equation of state is composed with the state variables i_d (d-axis current) and d (voltage disturbance).

$$si_d = -\frac{R}{L_d} i_d + \frac{d}{L_d} + \frac{v_d^*}{L_d}$$
$$sd = sd_d$$

Here, if the estimated value for i_d and d is $\hat{\iota_d}$ and $\hat{d}$, respectively, the estimated equation of state for the observer side can be represented as follows, by multiplying the estimated gains K_{Ed1} and K_{Ed2} to the estimated error.

$$s\widehat{\iota_d} = -\frac{R}{L_d}\widehat{\iota_d} + \frac{\hat{d}}{L_d} + \frac{v_d^*}{L_d} + K_{Ed1}(i_d - \widehat{\iota_d})$$
$$s\hat{d} = K_{Ed2}(i_d - \widehat{\iota_d})$$

From the above equation, $\widehat{\iota_d}$ and $\hat{d}$ can be represented as follows:

$$\widehat{\iota_d} = \frac{\frac{K_{Ed2}}{L_d}}{s^2 + \left(\frac{R}{L_d} + K_{Ed1}\right)s + \frac{K_{Ed2}}{L_d}}\left\{\left(1 + \frac{K_{Ed1}}{K_{Ed2}} L_d s\right) i_d + \frac{s}{K_{Ed2}} v_d^*\right\}$$
$$\hat{d} = \widehat{d_d} = \frac{\frac{K_{Ed2}}{L_d}}{s^2 + \left(\frac{R}{L_d} + K_{Ed1}\right)s + \frac{K_{Ed2}}{L_d}}\{(L_d s + R) i_d - v_d^*\}$$

Looking at the above equation, $\widehat{\iota_d}$ and $\widehat{d_d}$ can be represented in a quadratic form, with the inputs i_d and v_d^*. Also, the natural frequency ω_n and attenuation coefficient ζ are as follows:

$$\omega_n = \sqrt{\frac{K_{Ed2}}{L_d}}$$
$$\zeta = \frac{\frac{R}{L_d} + K_{Ed1}}{2\sqrt{\frac{K_{Ed2}}{L_d}}}$$

The estimated gains K_{Ed1} and K_{Ed2} of the d-axis inductive voltage estimation system can be represented as follows by using ω_n and ζ.

$$K_{Ed1} = 2\zeta_{EG}\omega_{EG} - \frac{R}{L_d}$$

$$K_{Ed2} = \omega_{EG}^2 L_d$$

ω_{EG}: Natural frequency for the inductive voltage estimation system

ζ_{EG}: Attenuation coefficient for the inductive voltage estimation system

The estimated equation of state is further rearranged as follows:

$$\hat{i_d} = \frac{1}{s}\left\{-\frac{R}{L_d}\hat{i_d} + \frac{\widehat{d_d}}{L_d} + \frac{v_d^*}{L_d} + K_{Ed1}(i_d - \hat{i_d})\right\}$$

$$\widehat{d_d} = \frac{1}{s}\{K_{Ed2}(i_d - \hat{i_d})\}$$

From the above equation, the block diagram of d-axis inductive voltage estimation is as shown in Figure 7-7.

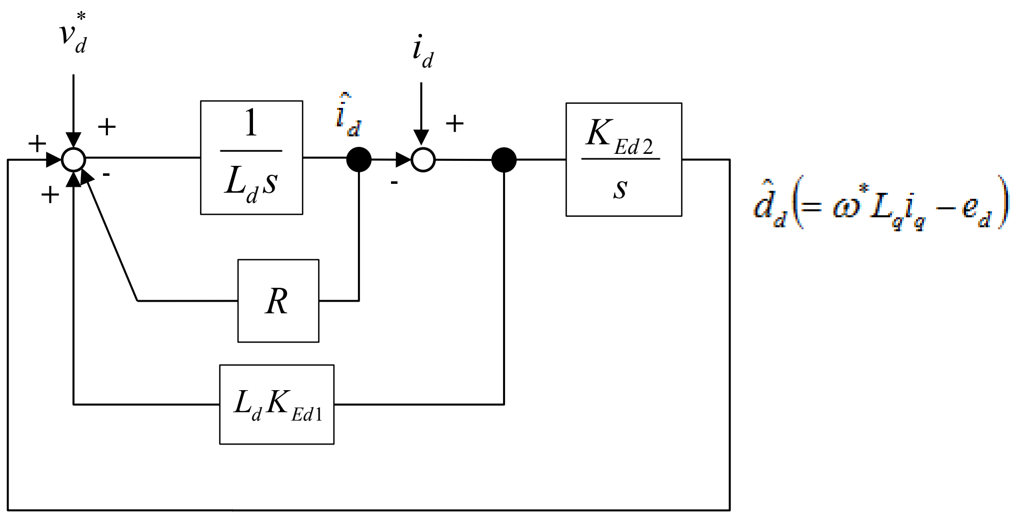


Figure 7-7 Block diagram of the d-axis inductive voltage observer

The q-axis side can be calculated in the same way, and $\hat{\iota_q}$ and $\hat{d}$ can be represented as follows:K_{Eq1} and K_{Eq2} are the estimated gains for the q axis side.

$$\widehat{\iota_q} = \frac{\frac{K_{Eq2}}{L_q}}{s^2 + \left(\frac{R}{L_q} + K_{Eq1}\right)s + \frac{K_{Eq2}}{L_q}} \left\{\left(1 + \frac{K_{Eq1}}{K_{Eq2}} L_q s\right) i_q + \frac{s}{K_{Eq2}} v_q^*\right\}$$

$$\hat{d} = \widehat{d_q} = \frac{\frac{K_{Eq2}}{L_q}}{s^2 + \left(\frac{R}{L_q} + K_{Eq1}\right)s + \frac{K_{Eq2}}{L_q}} \left\{\left(L_q s + R\right) i_q - v_q^*\right\}$$

In the same way as the d axis, looking at the above equation, $\widehat{\iota_q}$ and $\widehat{d_q}$ can be represented in a quadratic form, with the inputs i_q and v_q^*. Also, the natural frequency ω_n and attenuation coefficient ζ are as follows:

$$\omega_n = \sqrt{\frac{K_{Eq2}}{L_q}}$$

$$\zeta = \frac{\frac{R}{L_q} + K_{Eq1}}{2\sqrt{\frac{K_{Eq2}}{L_q}}}$$

Therefore, the estimated gains K_{Eq1} and K_{Eq2} of the q-axis inductive voltage estimation system can be represented as follows:

$$K_{Eq1} = 2\zeta_{EG}\omega_{EG} - \frac{R}{L_q}$$

$$K_{Eq2} = \omega_{EG}^2 L_q$$

ω_{EG}: Natural frequency for the inductive voltage estimation system

ζ_{EG}: Attenuation coefficient for the inductive voltage estimation system

In the same way as the d axis, the estimated equation of state can be represented as follows:

$$\widehat{i_q} = \frac{1}{s}\left\{-\frac{R}{L_q}\widehat{i_q} + \frac{\widehat{d_q}}{L_q} + \frac{v_q^*}{L_q} + K_{Eq1}\left(i_q - \widehat{i_q}\right)\right\}$$

$$\widehat{d_q} = \frac{1}{s}\left\{K_{Eq2}\left(i_q - \widehat{i_q}\right)\right\}$$

From the above equation, the block diagram of q-axis inductive voltage estimation is as shown in Figure 7-8.

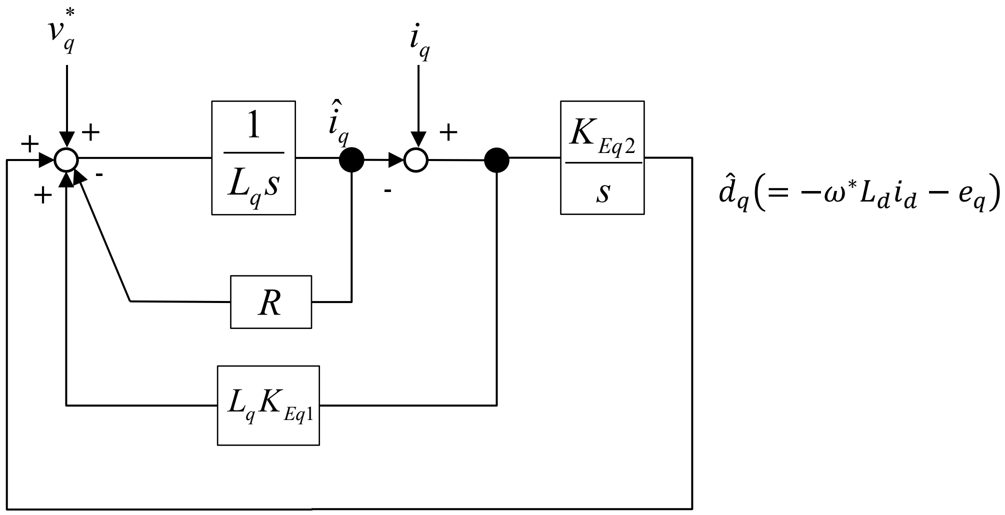


Figure 7-8 Block diagram of the q-axis inductive voltage observer

Next, the inductive voltages can be calculated from voltage disturbances $\widehat{d_d}$ and $\widehat{d_q}$.

$$e_d = -\widehat{d_d} + \omega^* L_q i_q$$

$$e_q = -\widehat{d_q} - \omega^* L_d i_d$$

$$\Delta\theta = \operatorname{atan}\left(\frac{e_d}{e_q}\right)$$

From the above, the phase error $\Delta\theta$ between the actual d/q axis and the estimated d/q axis can be obtained.

Finally, reflect the phase error $\Delta\theta$ to the estimated d/q-axis phase. The reflection is performed according to the block diagram shown in Figure 7-9.

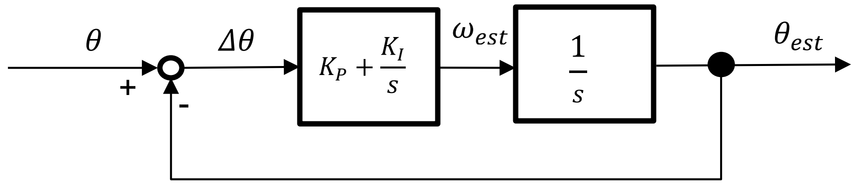


Figure 7-9 Block diagram of the magnetic pole estimation system

Here, the closed-loop transfer function from the magnetic pole position θ of the motor to the estimated magnetic pole position θ_{est} can be represented as follows:

$$\frac{\theta_{est}(s)}{\theta(s)} = \frac{K_I\left(s\frac{K_P}{K_I}+1\right)}{s^2 + K_P s + K_I}$$

Therefore, the transfer function from the motor's actual magnetic pole position to the estimated magnetic pole position is in quadratic form, and the natural frequency ω_n and attenuation coefficient ζ of the magnetic pole position estimation system are determined by the following formulas.

$$\omega_n = \sqrt{K_I}$$

$$\zeta = \frac{K_P}{2\sqrt{K_I}}$$

As a result, the control gains $K_{P\_phase\_error}$ and $K_{I\_phase\_error}$ for the magnetic pole position estimation system can be represented as follows:

$$K_{P\_phase\_error} = 2\zeta_{\Delta\theta}\omega_{\Delta\theta}$$

$$K_{I\_phase\_error} = \omega_{\Delta\theta}^2$$

$\omega_{\Delta\theta}$: Natural frequency for the magnetic pole position estimation system

$\zeta_{\Delta\theta}$: Attenuation coefficient for the magnetic pole position estimation system

As above, position and speed estimation is complete.

7.7 Flux weakening control function

The flux weakening control function controls the d-axis current in the negative direction. Even under conditions where the inductive voltage (= ωψ) generated in proportion to the rotation of the PM motor exceeds the voltage that can be output from the inverter bus voltage, this function increases the d-axis current command value in the negative direction to reduce the inductive voltage (Figure 7-10). Through this reduction in the voltage saturation region, the q-axis current command value, which is necessary for acceleration, can be increased and the output torque in the high-speed rotation region and acceleration of rotation can thus be improved.

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

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

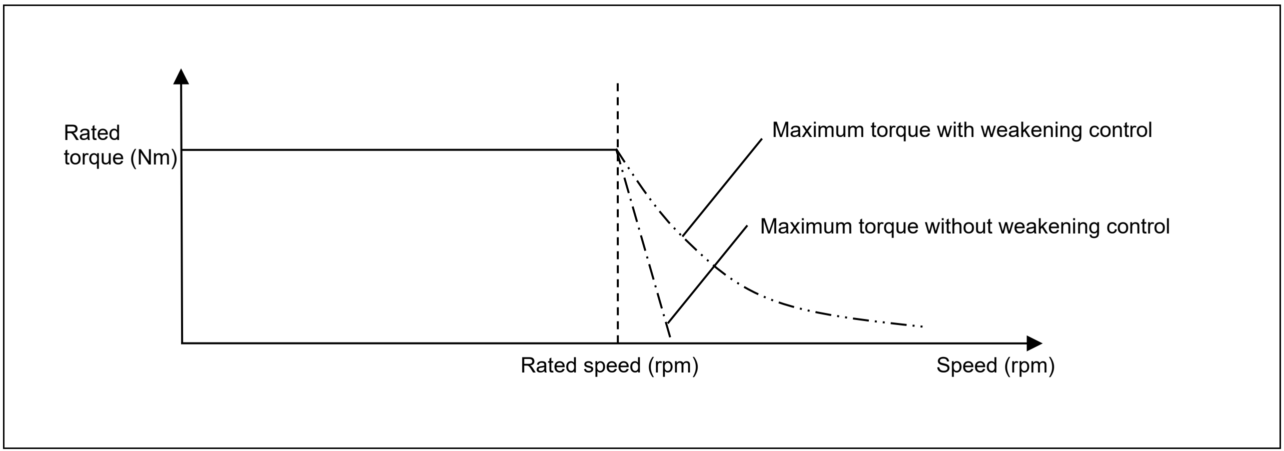


Figure 7-10 Relationship between the Available Output Torque and Speed

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

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

I_a: Magnitude of current vector (A)

Figure 7-11 Equation for Calculating the Maximum Limit on the Inductive Voltage

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

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

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

Figure 7-12 Equation for Calculating the d-axis Current Command Value in Flux Weakening Control

7.8 Voltage phase lead compensation

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

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

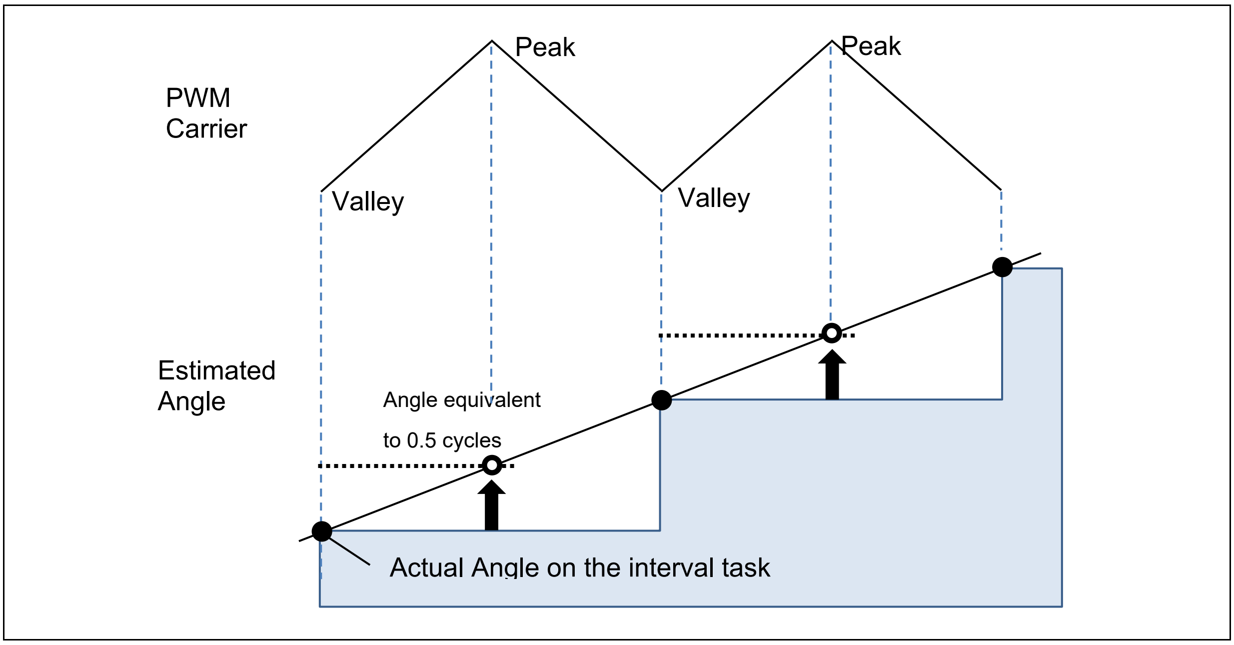


Figure 7-13 Example of the amount of angle advanced in a PWM carrier cycle

7.9 Voltage error compensation

In the voltage PWM inverter, to prevent the switching elements of the upper and lower sides from creating a short circuit, a dead time during which the two elements are simultaneously turned off is set. Therefore, an error arises between the voltage command value and the 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.

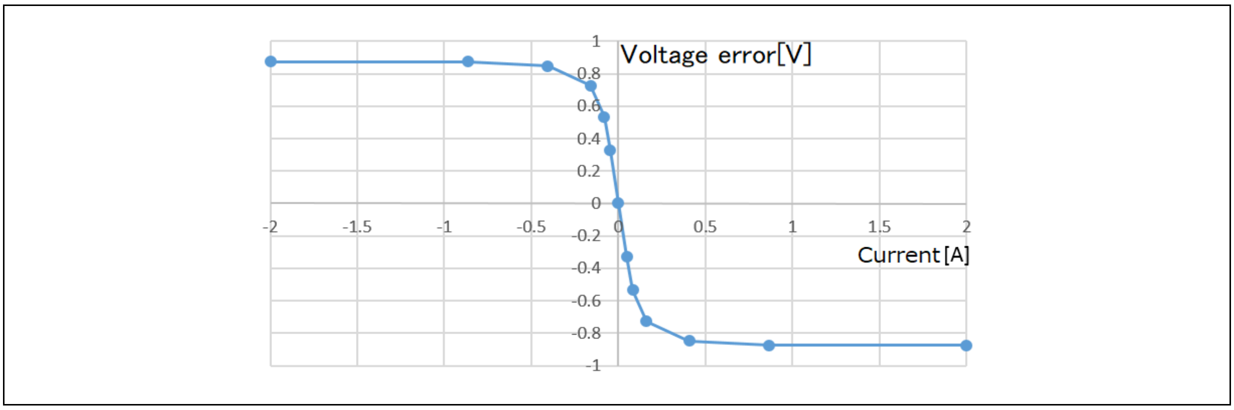


Figure 7-14 Current dependency in the voltage error (example)

7.10 Pulse Width Modulation (PWM) mode

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

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

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

m: Modulation rate, V: 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

7.11 Shunt Resistor Current Measurement Method

This section explains the current measurement method using 1 shunt resistor as employed in the sample software.

7.11.1 Current Measurement Timing with 1 Shunt Resistor

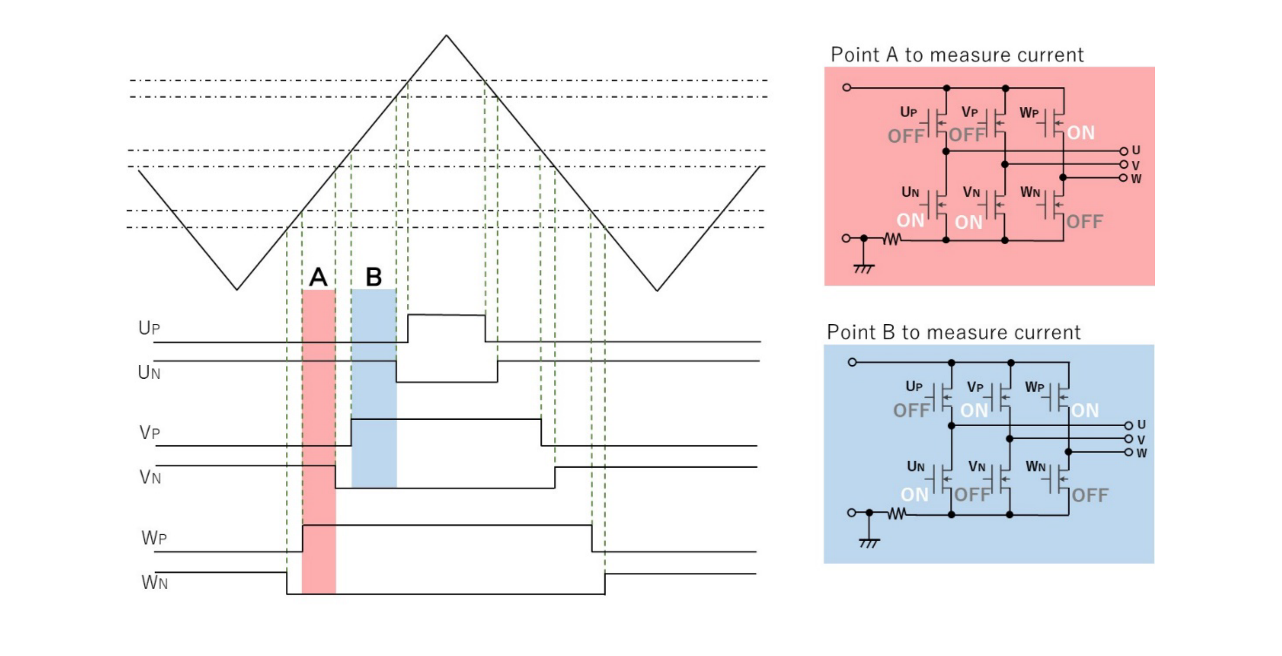


Figure 7-15 Complementary PWM Waveform Diagram (Example: Duty Ratio Relationship W > V > U)

In the sample software, control is performed using the GPT unit to output three-phase PWM signals with dead time in complementary PWM mode. Figure 7-15 shows the waveform of the complementary PWM (example: duty ratio relationship W > V > U).

At point A in the figure, only the upper arm of phase W is turned ON. At this time, the current flowing through the 1 shunt resistor (Idc) and the W-phase current (Iw) have the relationship: Idc = Iw.

At point B in the figure, only the lower arm of phase U is turned ON. At this time, the relationship between Idc and the U-phase current (Iu) is: Idc = -Iu.

The current of the remaining phase V (Iv) can be reconstructed using Kirchhoff's first law as:Iv = -Iu - Iw.

Therefore, if the current values flowing through the 1 shunt resistor at points A and B in the figure can be obtained, the three-phase currents can be determined.

The above example assumes a duty ratio relationship of W > V > U. As the PWM output changes, there are a total of six possible combinations of duty ratio order. The relationship between the detected current values at points A and B and the corresponding phases varies depending on the pattern, so it is necessary to assign the detected values to the respective phases accordingly. Since the duty ratio relationship is already known at the time of setting the duty values, it is possible to switch the assignment of the detected current values to each phase based on that information.

Table 7-2 Duty Pattern and Phase Currents

Duty pattern	Point A	Point B
W > V > U	Iw	-Iu
W > U > V	Iw	-Iv
V > W > U	Iv	-Iu
V > U > W	Iv	-Iw
U > W > V	Iu	-Iv
U > V > W	Iu	-Iw

7.11.2 1 Shunt Resistor Current Measurement Method

When measuring the current through a single shunt resistor as described in 7.11.1, it is necessary to control the A/D converter's conversion timing according to the PWM duty settings. In the sample software, this is achieved by using the A/D conversion start request function triggered by compare match between the GPT module's GTADTRA and GTADTRB registers and the GTCNT counter.

7.11.3 Duty Adjustment

Current detection using 1 shunt resistor is possible when the timing shown in 7.11.1 can be ensured. However, depending on the PWM duty conditions during operation, sufficient time for A/D conversion may not be available, making it impossible to accurately acquire the current values. To handle such cases where timing cannot be ensured, the following two countermeasures have been implemented:

(1). When the switching timings of two phases are close to each other:

When the switching timings of two phases are too close together and sufficient time for A/D conversion cannot be secured, the PWM duty is left unchanged. Instead, the switching timing of the phase that changes later is delayed by the amount of time required for A/D conversion, thereby ensuring enough time for accurate current sampling.

(2). When timing delay is not possible:

When the PWM switching timing is delayed as described above, if the duty cycle is wide enough that the delayed timing reaches the end of the PWM carrier period, the timing delay cannot be applied. In such cases, since the modulation index is close to 1, a limit is imposed on the modulation index to ensure that the PWM switching timing does not extend beyond the end of the carrier period.

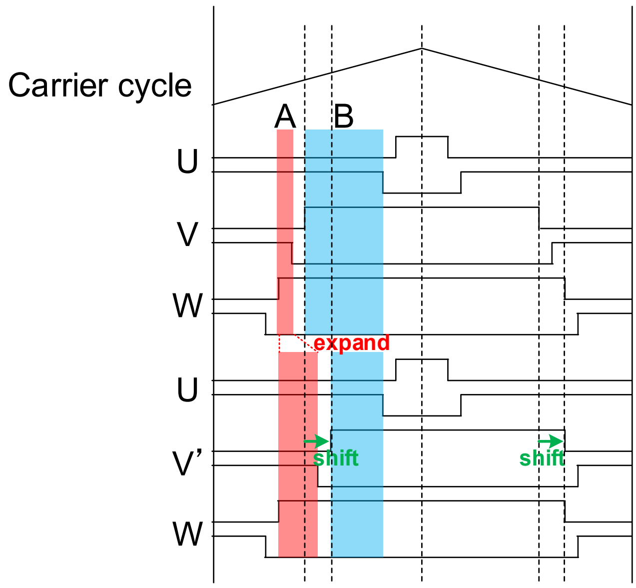


Figure 7-16 Duty Adjustment

7.11.4 AD Trigger

The timing of the A/D trigger and the scan group is shown below.

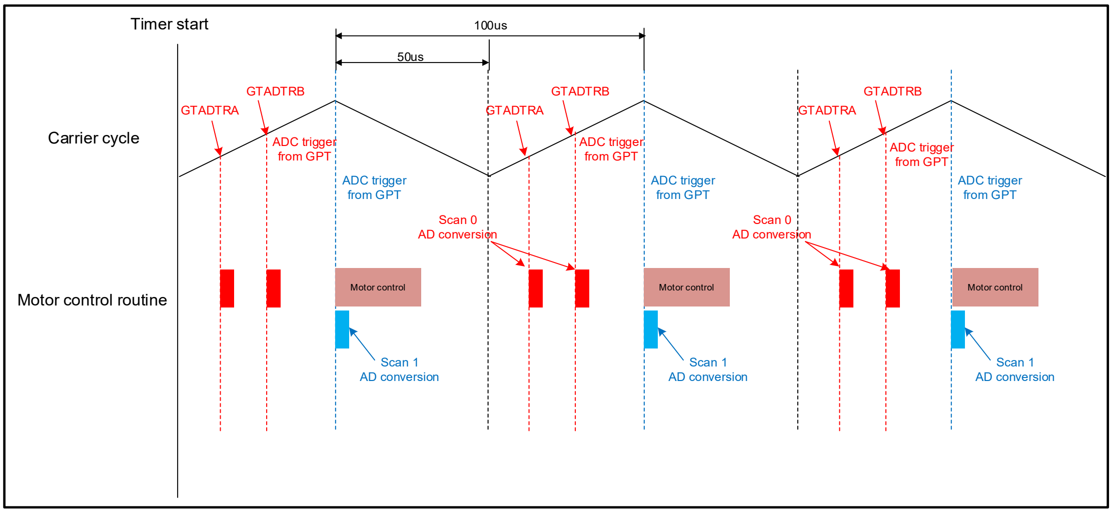


Figure 7-17 Trigger Timing

8. Hardware specifications

8.1 User interface

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

Table 8-1 User interface in board circuit

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

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

Table 8-2 Pin interface [1/2]

Function	RA6T2	RA6T3	RA4T1
Measure inverter bus voltage	Ver.1: PA06 / AN006 Ver.2: PA07 / AN007	P004 / AN004	P004 / AN004
Input speed reference (VR1)	Ver.1: PB00 / AN008 Ver.2: P000 / AN016	P005 / AN005	P005 / AN005
START/STOP Toggle switch (SW1)	PD04	P304	P304
Error reset push switch (SW2)	PD07	P200	P200
LED1 Light control	PD01	P113	P113
LED2 Light control	PD02	P106	P106
1 Shunt Current Measurement	PA04 / AN004	P000 / AN000	P000 / AN000
PWM Output (Up)	PB04 / GTIOC4A	P409 / GTIOC1A	P409 / GTIOC1A
PWM Output (Vp)	PB06 / GTIOC5A	P103 / GTIOC2A	P103 / GTIOC2A
PWM Output (Wp)	PB08 / GTIOC6A	P111 / GTIOC3A	P111 / GTIOC3A
PWM Output (Un)	PB05 / GTIOC4B	P408 / GTIOC1B	P408 / GTIOC1B
PWM Output (Vn)	PB07 / GTIOC5B	P102 / GTIOC2B	P102 / GTIOC2B
PWM Output (Wn)	PB09 / GTIOC6B	P112 / GTIOC3B	P112 / GTIOC3B
Emergency PWM stop input at overcurrent	PC13 / GTETRGD	P104 / GTETRGB	P104 / GTETRGB

Table 8-3 Pin interface [2/2]

Function	RA8T1	RA8T2
Measure inverter bus voltage	P008 / AN008	P007 / AN007
Input speed reference (VR1)	P014 / AN007	P015 / AN015
START/STOP Toggle switch (SW1)	PA15	PA00
Error reset push switch (SW2)	PA13	PA07
LED1 Light control	PA12	P614
LED2 Light control	PA14	PA15
1 Shunt Current Measurement	P004 / AN000	P006 / AN006
PWM Output (Up)	P115 / GTIOC5A	P605 / GTIOC8A
PWM Output (Vp)	P113 / GTIOC2A	P603 / GTIOC7A
PWM Output (Wp)	P300 / GTIOC3A	P612 / GTIOC9A
PWM Output (Un)	P609 / GTIOC5B	P604 / GTIOC8B
PWM Output (Vn)	P114 / GTIOC2B	P602 / GTIOC7B
PWM Output (Wn)	P112 / GTIOC3B	P613 / GTIOC9B
Emergency PWM stop input at overcurrent	P613 / GTETRGA	P112 / GTETRGA

8.2 Peripheral functions

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

Table 8-4 Peripheral functions [1/2]

Peripheral	**Purpose**	**RA6T2**	**RA6T3**	**RA4T1**
A/D converter	1 Shunt Current Measurement	AN004	AN000	AN000
	Measure inverter bus voltage	Ver.1: AN006 Ver.2: AN007	AN004	AN004
	Measure VR input	Ver.1: AN008 Ver.2: AN016	AN005	AN005
AGTW	Interval timer for speed control	AGT0	AGT0	AGT0
GPT	PWM output of U-phase	CH4	CH1	CH1
	PWM output of V-phase	CH5	CH2	CH2
	PWM output of W-phase	CH6	CH3	CH3
POEG	Emergency stop input of overcurrent	Group D	Group B	Group B

Table 8-5 Peripheral functions [2/2]

Peripheral	**Purpose**	**RA8T1**	**RA8T2**
A/D converter	1 Shunt Current Measurement	AN000	AN006
	Measure inverter bus voltage	AN008	AN007
	Measure VR input	AN007	AN015
AGTW	Interval timer for speed control	AGT0	AGT0
GPT	PWM output of U-phase	CH5	CH8
	PWM output of V-phase	CH2	CH7
	PWM output of W-phase	CH3	CH9
POEG	Emergency stop input of overcurrent	Group A	Group A

(1) A/D converter

A/D converter measure current of 1 shunt(I), inverter bus voltage(Vdc), and Speed reference input (VR) with "Single scan mode" (use hardware trigger).

(2) General asynchronous timer (AGTW)

AGTW is used as cyclic interval timer to generate speed control period interrupt.

(3) General purpose timer (GPT)

GPT outputs complementally PWM by its original mode with dead time.

(4) Port output enable for GPT (POEG)

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

9. Software specifications and configuration

9.1 Software specifications

The following shows the basic software specifications of this system.

Table 9-1 Basic Specifications of Sensorless Vector Control Software

Item	Content	
Control method	Vector control	
Motor rotation start/stop	SW1 input or input from ‘Renesas Motor Workbench’	
Position detection method	Sensorless (BEMF observer)	
Input voltage	24 [VDC]	
Main clock frequency	RA6T2: 240 [MHz] RA6T3: 200 [MHz] RA4T1: 100 [MHz] RA8T1: 480 [MHz] RA8T2: 1 [GHz]	
Carrier frequency (PWM)	10 [kHz] (Carrier period: 100 [μs])	
Modulation method of PWM	Space vector modulation (enable to select sinsoidal modulation)	
Dead time	2 [μs]	
Current control period	RA6T2: 100 [μs] RA6T3: 100 [μs] RA4T1: 200 [μs] RA8T1: 100 [μs] RA8T2 :100[μs]	
Speed control period	RA6T2: 1000 [μs] RA6T3: 1000 [μs] RA4T1: 2000 [μs] RA8T1: 1000 [μs] RA8T2: 1000 [μs]	
Rotation speed control range	CW: 0 [rpm] to 2400 [rpm] CCW: 0 [rpm] to 2400 [rpm] However, 500 [rpm] or less is driven by a speed open loop.	
Natural frequency of each control system	Current control system : 300 [Hz] Speed control system : 3 [Hz] BEMF estimation system : 1000 [Hz] (500 [Hz] only RA4T1) Position estimation system : 20 [Hz]	
Optimization setting of compiler	Optimization level	Optimize more(-O2) (default setting)
Processing stop for protection	Disables the motor control signal output (six outputs), under any of the following conditions. 1. Instantaneous value of current of any phase exceeds 3.54(=1.67*sqrt (2)*1.5) [A] (monitored in current control period) 2. Inverter bus voltage exceeds 60 [V] (monitored in current control period) 3. Inverter bus voltage is less than 8 [V] (monitored in current control period) 4. Rotation speed exceeds 4500 [rpm] (monitored in current control period) When an external over current signal is detected (when a low level is detected), the PWM output ports are set to high impedance state.	

9.2 Overall configuration of the software

The overall configuration of the software is shown below.

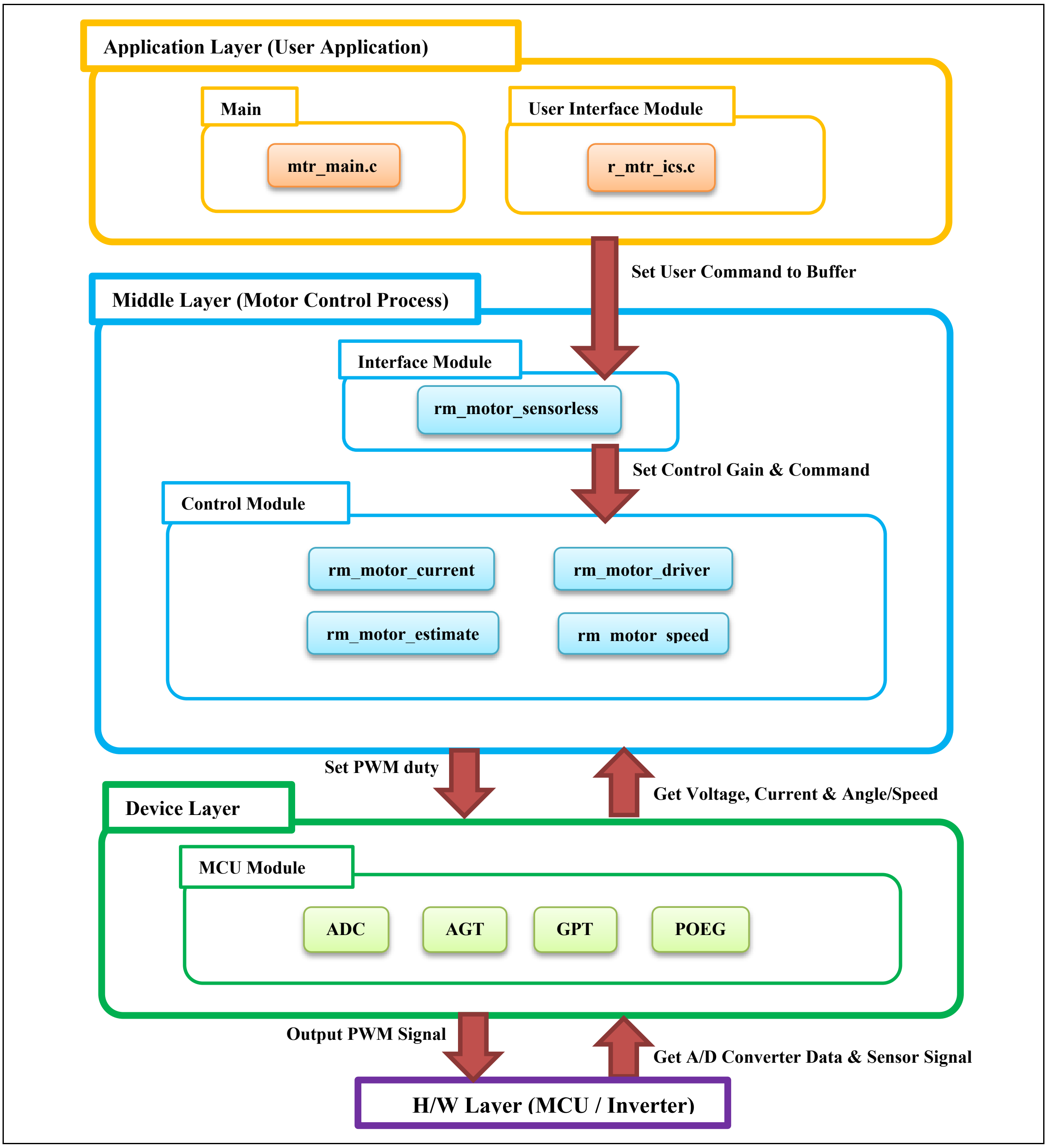


Figure 9-1 Module Configuration

9.3 Explanation about interrupt

In the sample program of this document, main process are performed at speed control period interrupt and current control period interrupt. UI functions as interface layer are performed in main routine. POEG hardware interrupt is used as emergency stop at detection of overcurrent by hardware.

Table 9-2 Interrupt priority

Interrupt level	Priority	function
15	Min	
14		
13		
12		
11		
10		AGT0 INT Speed control period interrupt
9		
8		
7		
6		
5		GPT4 COUNTER OVERFLOW (RA6T2) GPT1 COUNTER OVERFLOW (RA6T3, RA4T1) GPT5 COUNTER OVERFLOW (RA8T1) GPT8 COUNTER OVERFLOW (RA8T2) Current control period Interrupt (GPT Overflow Interrupt)
4		
3		
2	Max	
1		
0		POEG3 EVENT(RA6T2) POEG1 EVENT(RA6T3, RA4T1) POEG0 EVENT(RA8T1, RA8T2) Over current detection interrupt

9.4 File and folder configuration

The lists of folder and file configuration of the sample program are shown below.

Table 9-3 File and folder configuration[1/2]

Folder	Subfolder	File	Remarks
ra_cfg			Generated config header
ra_gen			Generated register setting, main function etc.
ra	arm		CMSIS source code
	board		Function definition for board (bsp)
	fsp/inc/api	bsp_api.h	BSP API definition
		fsp_common_api.h	Common API definition
		r_adc_api.h	AD API definition
		r_elc_api.h(RA6T3, RA4T1, RA8T1)	ELC API definition
		r_ioport_api.h	I/O API definition
		r_poeg_api.h	POEG API definition
		r_three_phase_api.h	3phase PWM API definition
		r_timer_api.h	Timer API definition
		r_transfer_api.h	Transfer API definition
		rm_motor_angle_api.h	Angle detection API definition
		rm_motor_api.h	Motor control API definition
		rm_motor_current_api.h	Current control API definition
		rm_motor_driver_api.h	Motor driver API definition
		rm_motor_position_api.h	Position control API definition
		rm_motor_speed_api.h	Speed control API definition
	fsp/inc/instances	r_adc_b.h(RA6T2, RA8T2) r_adc.h(RA6T3, RA4T1 and RA8T1)	Function definition for ADC
		r_agt.h	Function definition for AGT
		r_elc.h(Only RA6T3, RA4T1 and RA8T1)	Function definition for ELC
		r_gpt.h	Function definition for GPT
		r_gpt_three_phase.h	Function definition for 3 Phase PWM
		r_ioport.h	Function definition for I/O
		r_poeg.h	Function definition for POEG
		rm_motor_current.h	Function definition for current control
		rm_motor_driver.h	Function definition for motor driver
		rm_motor_estimate.h	Function definition for angle estimate
		rm_motor_sensorless.h	Function definition for Sensorless control
		rm_motor_speed.h	Function definition for speed control

Table 9-4 File and folder configuration[2/2]

Folder	Subfolder	File	Remarks
ra	fsp/lib		Library files
	fsp/src	bsp	BSP driver folder
		r_adc_b/r_adc_b.c(RA6T2, RA8T2) r_adc/r_adc.c(RA6T3, RA4T1 and RA8T1)	ADC driver
		r_agt/r_agt.c	AGT driver
		r_elc/r_elc.c(Only RA6T3, RA4T1 and RA8T1)	ELC driver
		r_gpt/r_gpt.c	GPT driver
		r_gpt_three_phase/ r_gpt_three_phase.c	3 phase PWM driver
		r_ioport/r_ioport.c	I/O driver
		r_poeg/r_poeg.c	POEG driver
		rm_motor_current/rm_motor_current.c	Current control driver
		rm_motor_current/rm_motor_current_library.h	Current control library API definition
		rm_motor_driver/rm_motor_driver.c	Motor driver
		rm_motor_estimate/rm_motor_estimate.c	Angle estimate driver
		rm_motor_estimate/rm_motor_estimate_library.h	Angle estimate library API definition
		rm_motor_sensorless/rm_motor_sensorless.c	Sensorless control driver
		rm_motor_speed/rm_motor_speed.c	Speed control driver
		rm_motor_speed/rm_motor_speed_library.h	Speed control library API definition
src	application/main	mtr_main.h , mtr_main.c	User main function
		r_mtr_control_parameter.h	Control parameters definition
		r_mtr_motor_parameter.h	Motor parameters definition
	application/rmw	r_mtr_rmw.h , r_mtr_rmw.c	Function definition for Analyzer UI
		ICS2_RA6T2.h , ICS2_RA6T3.h , ICS2_RA4T1.h ICS2_RA8T1.h , ICS2_RA8T2.h	Function definition for RMW communication
		ICS2_RA6T2.o , ICS2_RA6T3.o , ICS2_RA4T1.o ICS2_RA8T1.o , ICS2_RA8T2.o	Communication library for GUI tool

FSP can be used to generate peripheral drivers easily.

FSP saves the settings information about the microcontrollers, peripheral functions, pin functions, and other items that are used for the project in a Configuration Settings File (configuration.xml), and references the information saved in the file. Settings of configuration can be changed with an operation in e^2 studio.

Below folders are also generated automatically by FSP at structed a program project.

- ra

Information about selected target board, header and code files of selected FSP modules are installed below this folder.

If you want to change code for adding functions, maintain the header files of modules and C code files under ra/fsp/inc and ra/fsp/src.

- ra_cfg

Settings (as like “compile option”) about selected functions of FSP modules are registered below this folder. Do not edit. These settings are able to be changed only by FSP operation.

- ra_gen

Files which include variables, and these initial values which are generated with configuration settings(Pins, interrupts, property of each module and so on), set by FSP operation, are registered below this folder. These files are always generated at program build. Therefore, it is no need to modify directly.

9.5 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.

9.5.1 Functions

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

Table 9-5 Functions available in the application layer

Functions	Description
Main processing	Enables or disables each user command in the system.
UI processing	Selects of Board UI or RMW UI, and manages these.
Manager processing	Manages motor start/stop. Reads speed reference and reflects.
RMW UI processing	Acquires and sets parameters (including command values).

9.5.2 Configurations

Application layer is a user interface layer of motor control, which uses generated FSP modules (Speed, Current and so on). Therefore, Application layer program of this sample program is implemented only as a sample. Configurations of this application layer are implemented in “mtr_main.h” file as MACRO definitions.

Table 9-6 shows the configurations used in the application layer.

Table 9-6 List of configurations

File name	Macro name	Description
mtr_main.h	CHATTERING_CNT	Chattering counts for switch read.
	MTR_MAX_SPEED_RPM	Maximum limit of speed reference If you set large value of this, speed reference is limited with this value.
	CONFIG_DEFAULT_UI	Selection of Board UI/RMW UI at reset start.
	MTR_ADCH_VR1	AD conversion channel of VR for speed reference.

Table 9-7 List of initial values for configurations

Macro name	Set value
CHATTERING_CNT	10
MTR_MAX_SPEED_RPM	2400
CONFIG_DEFAULT_UI	BOARD_UI
MTR_ADCH_VR1	RA6T2 Ver.1 : 8 RA6T2 Ver.2 : 16 RA6T3, RA4T1 : 5 RA8T1 : 7 RA8T2 : 15

9.5.3 Structure and variable information

Table 9-8 and Table 9-9 list the variables that can be used by users in the application layer.

The variables which are listed in Table 9-8 and Table 9-9 can be changed by users.

When you change these values by RMW, these are reflected to variables listed in Table 9-10. The application layer reflects these settings with each module parameter update process.

Table 9-8 List of variables[1/2]

Variable	Description
g_u1_trig_enable_write	An internal flag displays enable of value update
com_u1_mode_system	Change system mode 0: Stop motor rotation 1: Start motor rotation 3: Reset errors
g_u1_mode_system	System mode 0: Stop motor rotation 1: Start motor rotation 2: Error
com_u1_enable_write	A flag displays user command for value update (When this variable is set same value with “g_u1_enable_write”, set values are reflected to each variables.)
g_u1_enable_write	A flag displays enable of value update
com_u2_mtr_pp	Motor pole pairs
com_f4_mtr_r	Resistance [ohm]
com_f4_mtr_ld	Inductance of d-axis [H]
com_f4_mtr_lq	Inductance of q-axis [H]
com_f4_mtr_m	Magnetic flux [Wb]
com_f4_mtr_j	Inertia [kgm^2]
com_f4_current_omega	Natural frequency of current control [Hz]
com_f4_current_zeta	Attenuation coefficient of current control
com_f4_speed_omega	Natural frequency of speed control [Hz]
com_f4_speed_zeta	Attenuation coefficient of speed control
com_f4_e_obs_omega	Natural frequency of BEMF observer [Hz]
com_f4_e_obs_zeta	Attenuation coefficient of BEMF observer
com_f4_pll_est_omega	Natural frequency of speed PLL [Hz]
com_f4_pll_est_zeta	Attenuation coefficient of speed PLL
com_f4_ref_id	Maximum current of d-axis at open loop [A]
com_f4_ol_id_up_step	Update step of d-axis current at open loop [A]
com_f4_ol_id_down_step	Down step of d-axis current at open loop [A]

Table 9-9 List of variables[2/2]

Variable	Description
com_f4_id_down_speed_rpm	Speed to start declaration of current of d-axis at open loop (mechanical) [rpm]
com_f4_id_up_speed_rpm	Speed to start update of current of d-axis at open loop (mechanical) [rpm]
com_f4_max_speed_rpm	Maximum speed reference (mechanical) [rpm]
com_f4_overspeed_limit_rpm	Threshold of over speed detection (mechanical) [rpm]
com_f4_overcurrent_limit	Threshold of software over current detection [A]
com_f4_iq_limit	Limit of q-axis current [A]
com_f4_limit_speed_change	Step of speed change at acceleration/deceleration (mechanical) [rpm]
com_f4_nominal_current	Nominal current [A]

Table 9-10 List of variables of the structure for RMW to update parameters

Structure	Description
g_user_motor_sensorless_extended_cfg	Structure of configuration for parameters about interface module which can be changed by user
g_user_motor_speed_extended_cfg	Structure of configuration for parameters about speed module which can be changed by user
g_user_motor_current_extended_cfg	Structure of configuration for parameters about current module which can be changed by user
g_user_motor_estimate_extended_cfg	Structure of configuration for parameters about angle/speed estimation module which can be changed by user

9.5.4 Macro definition

Macro definitions are listed below.

Table 9-11 List of macros [1/2] (mtr_main.h)

Macro Name	RA6T2	RA6T3	RA4T1
SW_ON	0	0	0
SW_OFF	1	1	1
SW1_ON	1	1	1
SW1_OFF	0	0	0
SW2_ON	0	0	0
SW2_OFF	1	1	1
CHATTERING_CNT	10	10	10
MTR_CW	-	-	-
MTR_CCW	-	-	-
WAIT_STOP_COUNT	-	-	-
MTR_LED_ON	BSP_IO_LEVEL_LOW	BSP_IO_LEVEL_LOW	BSP_IO_LEVEL_LOW
MTR_LED_OFF	BSP_IO_LEVEL_HIGH	BSP_IO_LEVEL_HIGH	BSP_IO_LEVEL_HIGH
ICS_UI	0	0	0
BOARD_UI	1	1	1
LOOP_SPEED	0	0	0
LOOP_POSITION	1	1	1
MTR_MAX_SPEED_RPM	2400	2400	2400
STOP_RPM	400	400	400
MTR_AD12BIT_DATA	4095.0f	4095.0f	4095.0f
VR1_SCALING	(MTR_MAX_SPEED_RPM + 100) / (MTR_AD12BIT_DATA * 0.5f)	(MTR_MAX_SPEED_RPM + 100) / (MTR_AD12BIT_DATA * 0.5f)	(MTR_MAX_SPEED_RPM + 100) / (MTR_AD12BIT_DATA * 0.5f)
VR1_SCALING_POS	-	-	-
VR1_180	-	-	-
VR1_POSITION_DEAD_BAND	-	-	-
ADJUST_OFFSET	0x7FF	0x7FF	0x7FF
MTR_FLG_CLR	0	0	0
MTR_FLG_SET	1	1	1
CONFIG_DEFAULT_UI	BOARD_UI	BOARD_UI	BOARD_UI
CONFIG_LOOP_MODE	-	-	-
MTR_ADCH_VR1	8	5	5
MTR_PORT_SW1	BSP_IO_PORT_13_PIN_04	BSP_IO_PORT_03_PIN_04	BSP_IO_PORT_03_PIN_04
MTR_PORT_SW2	BSP_IO_PORT_13_PIN_07	BSP_IO_PORT_02_PIN_00	BSP_IO_PORT_02_PIN_00
MTR_PORT_LED1	BSP_IO_PORT_13_PIN_01	BSP_IO_PORT_01_PIN_13	BSP_IO_PORT_01_PIN_13
MTR_PORT_LED2	BSP_IO_PORT_13_PIN_02	BSP_IO_PORT_01_PIN_06	BSP_IO_PORT_01_PIN_06
MTR_PORT_LED3	BSP_IO_PORT_13_PIN_03	-	-

Table 9-12 List of macros [2/2] (mtr_main.h)

Macro Name	RA8T1	RA8T2
SW_ON	0	0
SW_OFF	1	1
SW1_ON	1	1
SW1_OFF	0	0
SW2_ON	0	0
SW2_OFF	1	1
CHATTERING_CNT	10	10
MTR_CW	-	-
MTR_CCW	-	-
WAIT_STOP_COUNT	-	-
MTR_LED_ON	BSP_IO_LEVEL_LOW	BSP_IO_LEVEL_LOW
MTR_LED_OFF	BSP_IO_LEVEL_HIGH	BSP_IO_LEVEL_HIGH
ICS_UI	0	0
BOARD_UI	1	1
LOOP_SPEED	0	0
LOOP_POSITION	1	1
MTR_MAX_SPEED_RPM	2400	2400
STOP_RPM	400	400
MTR_AD12BIT_DATA	4095.0f	4095.0f
VR1_SCALING	(MTR_MAX_SPEED_RPM + 100) / (MTR_AD12BIT_DATA * 0.5f)	(MTR_MAX_SPEED_RPM + 100) / (MTR_AD12BIT_DATA * 0.5f)
VR1_SCALING_POS	-	-
VR1_180	-	-
VR1_POSITION_DEAD_BAND	-	-
ADJUST_OFFSET	0x7FF	0x7FF
MTR_FLG_CLR	0	0
MTR_FLG_SET	1	1
CONFIG_DEFAULT_UI	BOARD_UI	BOARD_UI
CONFIG_LOOP_MODE	-	-
MTR_ADCH_VR1	7	15
MTR_PORT_SW1	BSP_IO_PORT_10_PIN_15	BSP_IO_PORT_10_PIN_00
MTR_PORT_SW2	BSP_IO_PORT_10_PIN_13	BSP_IO_PORT_10_PIN_07
MTR_PORT_LED1	BSP_IO_PORT_10_PIN_12	BSP_IO_PORT_06_PIN_14
MTR_PORT_LED2	BSP_IO_PORT_10_PIN_14	BSP_IO_PORT_10_PIN_15
MTR_PORT_LED3	-	BSP_IO_PORT_10_PIN_04

Table 9-13 List of macros [1/2] (r_mtr_rmw.h)

Macro Name	RA6T2	RA6T3	RA4T1
USE_BUILT_IN	0	0	0
MTR_ICS_DECIMATION	5	5	3
ICS_BRR	19	250	250
ICS_INT_MODE	1	1	1
MTR_SQRT_2	1.41421356f	1.41421356f	1.41421356f
MTR_TWO_PI	6.28318531f	6.28318531f	6.28318531f
MTR_RAD_RPM	60/MTR_TWO_PI	60/MTR_TWO_PI	60/MTR_TWO_PI
MTR_RAD_DEGREE	360/MTR_TWO_PI	360/MTR_TWO_PI	360/MTR_TWO_PI
MTR_OVERCURRENT_MARGIN_MULT	1.5f	1.5f	1.5f

Table 9-14 List of macros [2/2] (r_mtr_rmw.h)

Macro Name	RA8T1	RA8T2
USE_BUILT_IN	0	0
MTR_ICS_DECIMATION	5	5
ICS_BRR	19	19
ICS_INT_MODE	1	1
MTR_SQRT_2	1.41421356f	1.41421356f
MTR_TWO_PI	6.28318531f	6.28318531f
MTR_RAD_RPM	60/MTR_TWO_PI	60/MTR_TWO_PI
MTR_RAD_DEGREE	360/MTR_TWO_PI	360/MTR_TWO_PI
MTR_OVERCURRENT_MARGIN_MULT	1.5f	1.5f

9.6 Interface Module

The interface module manages motor rotation with specific control modules (speed and current). This module controls data swapping between speed and current modules, total system mode of this project, and protection functions.

9.6.1 Functions

Table 9-15 lists the functions of interface module.

Table 9-15 List of interface 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 current control module.
Control module command value setting	Set speed reference to speed control module via API
Interrupt processing	Perform user functions which are registered as a callback at each interrupts (speed and current).

9.6.2 Module configuration diagram

Figure 9-2 shows the module configuration.

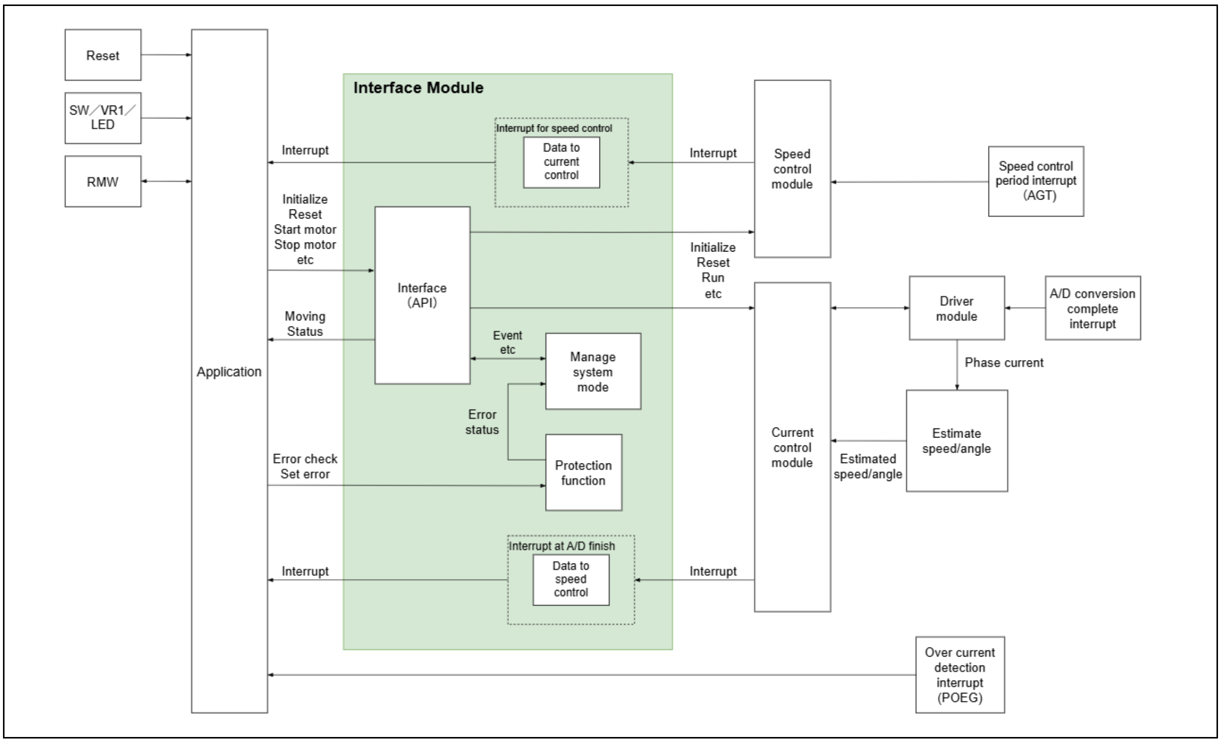
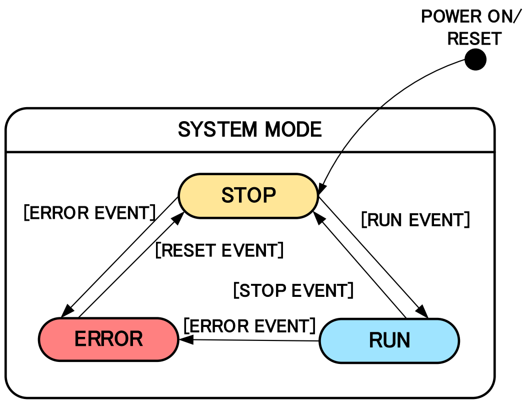


Figure 9-2 Module configuration diagram of interface module

9.6.3 State transition

Figure 9-3 is a state transition diagram of sample software. In the target software of this application note, the software state is managed by “SYSTEM MODE”.



		MODE		
		STOP	RUN	ERROR
EVENT	STOP	STOP	STOP	ERROR
	RUN	RUN	RUN	ERROR
	ERROR	ERROR	ERROR	ERROR
	RESET	STOP	RUN	STOP

Figure 9-3 State Transition Diagram of Sample Software

(1). SYSTEM MODE

“SYSTEM MODE” indicates the operating states of the system. The state transits on occurrence of each event (EVENT). “SYSTEM MODE” has 3 states that are motor drive stop (STOP), motor drive (RUN), and abnormal condition (ERROR).

(2). EVENT

When “EVENT” occurs in each “SYSTEM MODE”, “SYSTEM MODE” changes as shown the table in Figure 9-3, according to that “EVENT”. The occurrence factors of each event are shown below.

Table 9-16 List of EVENT

EVENT name	occurrence factor
STOP	by user operation
RUN	by user operation
ERROR	when the system detects an error
RESET	by user operation

9.6.4 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 9-17.

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

- 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.

Table 9-17 Operating conditions and setting values for the system protection functions

Category	Item	Value
Overcurrent error	Overcurrent limit value [A]	1.67
	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 (mechanical) [rpm]	4500
	Monitoring interval [μs]	Current control interval*1

*1 Refer to Table 9-1 Basic Specifications of Sensorless Vector Control Software.

9.6.5 API

Table 9-18 lists API functions of interface module.

Table 9-18 List of API functions

API	Description
RM_MOTOR_SENSORLESS_Open	Generates instances of this module and the modules to be used.
RM_MOTOR_SENSORLESS_Close	Close instances of this module and the modules to be used.
RM_MOTOR_SENSORLESS_Run	Run the motor.
RM_MOTOR_SENSORLESS_Stop	Stop the motor.
RM_MOTOR_SENSORLESS_Reset	Reset this module and the modules to be used.
RM_MOTOR_SENSORLESS_ErrorSet	Set error state.
RM_MOTOR_SENSORLESS_SpeedSet	Set speed reference (mechanical) [rpm]
RM_MOTOR_SENSORLESS_StatusGet	Get current state of the project.
RM_MOTOR_SENSORLESS_AngleGet	Get rotor angle [rad]
RM_MOTOR_SENSORLESS_SpeedGet	Get rotation speed (mechanical) [rpm]
RM_MOTOR_SENSORLESS_ErrorCheck	Check error occurrence
RM_MOTOR_SENSORLESS_PositionSet	Set position reference [degree] (Unsupported in sensorless vector)
RM_MOTOR_SENSORLESS_WaitStopFlagGet	Get a flag of waiting motor stop (Unsupported in sensorless vector)
RM_MOTOR_SENSORLESS_FunctionSelect	Select to use servo function (Unsupported in sensorless vector)

9.6.6 Structure and variable information

The structures and variables for interface module are listed below.

Table 9-19 List of structures and variable for interface module (rm_motor_api.h)

Structure	Members	Description
motor_callback_args_t	*p_context	Address of context data for callback function
	event	Event data of callback
motor_cfg_t	*p_motor_speed_instance	Address of speed control module instance
	*p_motor_current_instance	Address of current control module instance
	*p_callback	Address of registered callback function
	*p_context	Address of context data for registered callback function
	*p_extend	Address of structure of extended configuration data to refer
motor_api_t	*open	Function address to open module
	*close	Function address to close module
	*run	Function address to activate module (start motor rotation)
	*stop	Function address to inactivate module (stop motor rotation)
	*reset	Function address to reset module
	*errorSet	Function address to set error data
	*speedSet	Function address to set speed reference (mechanical) [rpm]
	*positionSet	Function address to set position reference (Unsupported in sensorless vector)
	*statusGet	Function address to get moving status
	*angleGet	Function address to get rotor angle
	*speedGet	Function address to get rotation speed (mechanical) [rpm]
	*waitStopFlagGet	Function address to get a flag for waiting “STOP” (Unsupported in sensorless vector)
	*errorCheck	Function address to check error occurrence
	*functionSelect	Function address to select servo function (Unsupported in sensorless vector)
motor_instance_t	*p_ctrl	Address of structure of variables to be used in the module
	*p_cfg	Address of structure of configuration data
	*p_api	Address of structure of API functions

Table 9-20 List of structures and variable for interface module (rm_motor_sensorless.h)

Structure	Members	Description
motor_sensorless_statemachine_t	u1_status	Moving status
	u1_status_next	Next moving status
	u1_current_event	Current happened event
	u2_error_status	Error status
motor_sensorless_extended_cfg_t	f_overcurrent_limit	Limit (threshold) of over current detection [A]
	f_overvoltage_limit	Limit (threshold) of over voltage detection [V]
	f_overspeed_limit	Limit (threshold) of over speed detection (mechanical) [rpm]
	f_lowvoltage_limit	Limit (threshold) of low voltage detection [V]
motor_sensorless_instance_ctrl_t	open	Module opened information
	u2_error_info	Error status
	st_statem	Structure of state machine
	st_speed_input	Structure of input data from speed control module
	st_speed_output	Structure of output data to speed control module
	st_current_input	Structure of input data from current control module
	st_current_output	Structure of output data to current control module
	*p_cfg	Address of structure of configuration to refer

9.6.7 Macro and enumeration definition

The macros and enumerations for interface module are listed below.

Table 9-21 List of macros for interface module

File name	Name of MACRO	Defined value	Description
rm_motor_sensorless.c	MOTOR_SENSORLESS_OPEN	('M' << 24U) | ('T' << 16U) | ('S' << 8U) | ('L' << 0U)	Opened information
	MOTOR_SENSORLESS_RAD2RPM	30.0F / 3.1415926535F	To use rad/s => rpm translation
	MOTOR_SENSORLESS_FLG_CLR	0	Clear a flag
	MOTOR_SENSORLESS_FLG_SET	1	Set a flag
	MOTOR_SENSORLESS_STATEMACHINE_SIZE_STATE	3	Size of status of state machine
	MOTOR_SENSORLESS_STATEMACHINE_SIZE_EVENT	4	Size of event of state machine
	MOTOR_SENSORLESS_STATEMACHINE_ERROR_NONE	0x00	No error happened
	MOTOR_SENSORLESS_STATEMACHINE_ERROR_EVENTOUTBOUND	0x01	Error of outbound of event
	MOTOR_SENSORLESS_STATEMACHINE_ERROR_STATEOUTBOUND	0x02	Error of outbound of status
	MOTOR_SENSORLESS_STATEMACHINE_ERROR_ACTIONEXCEPTION	0x04	Error of exception about action

Table 9-22 List of Enumeration type for interface module [1/2] (rm_motor_api.h)

Enumeration type name	Members	Defined value	Description
motor_error_t	MOTOR_ERROR_NONE	0x0000	No error happened
	MOTOR_ERROR_OVER_CURRENT_HW	0x0001	Over current error detected by hardware
	MOTOR_ERROR_OVER_VOLTAGE	0x0002	Over voltage error
	MOTOR_ERROR_OVER_SPEED	0x0004	Over speed error
	MOTOR_ERROR_HALL_TIMEOUT	0x0008	Timeout error of hall signal detection (Not happen in sensorless vector)
	MOTOR_ERROR_BEMF_TIMEOUT	0x0010	Timeout error of BEMF signal detection (Not happen in sensorless vector)
	MOTOR_ERROR_HALL_PATTERN	0x0020	Unused
	MOTOR_ERROR_BEMF_PATTERN	0x0040	BEMF signal pattern error (Not happen in sensorless vector)
	MOTOR_ERROR_LOW_VOLTAGE	0x0080	Low voltage error
	MOTOR_ERROR_OVER_CURRENT_SW	0x0100	Over current error detected by software
	MOTOR_ERROR_INDUCTION_CORRECT	0x0200	Induction calibration error (Not happen in sensorless vector)
	MOTOR_ERROR_UNKNOWN	0xFFFF	Unknown error
motor_callback_event_t	MOTOR_CALLBACK_EVENT_SPEED_FORWARD	1	Callback event before cyclic speed control process
	MOTOR_CALLBACK_EVENT_SPEED_BACKWARD	2	Callback event after cyclic speed control process
	MOTOR_CALLBACK_EVENT_CURRENT_FORWARD	3	Callback event before cyclic current control process
	MOTOR_CALLBACK_EVENT_CURRENT_BACKWARD	4	Callback event after cyclic current control process
	MOTOR_CALLBACK_EVENT_ADC_FORWARD	5	Callback event before A/D conversion end interrupt process (Not happen in sensorless vector)
	MOTOR_CALLBACK_EVENT_ADC_BACKWARD	6	Callback event after A/D conversion end interrupt process (Not happen in sensorless vector)
	MOTOR_CALLBACK_EVENT_CYCLE_FORWARD	7	Callback event before cyclic speed control process (Not happen in sensorless vector)
	MOTOR_CALLBACK_EVENT_CYCLE_BACKWARD	8	Callback event after cyclic speed control process (Not happen in sensorless vector)
motor_wait_stop_flag_t (No use in sensorless vector)	MOTOR_WAIT_STOP_FLAG_CLEAR	0	Clear a flag for waiting stop
	MOTOR_WAIT_STOP_FLAG_SET	1	Set a flag for waiting stop

Table 9-23 List of Enumeration type for interface module [2/2] (rm_motor_api.h)

Enumeration type name	Members	Defined value	Description
motor_function_select_t (No use in sensorless vector)	MOTOR_FUNCTION_SELECT_NONE	0	No servo function is selected
	MOTOR_FUNCTION_SELECT_INERTIA_ESTIMATE	1	Inertia estimation function is selected
	MOTOR_FUNCTION_SELECT_RETURN_ORIGIN	2	Return origin position function is selected

Table 9-24 List of Enumeration type for interface module (rm_motor_sensorless.h)

Enumeration type name	Members	Defined value	Description
motor_sensorless_ctrl_status_t	MOTOR_SENSORLESS_CTRL_STOP	0	Stop state
	MOTOR_SENSORLESS_CTRL_RUN	1	Run (active) state
	MOTOR_SENSORLESS_CTRL_ERROR	2	Error state
motor_sensorless_ctrl_event_t	MOTOR_SENSORLESS_CTRL_EVENT_STOP	0	Event to stop (inactive)
	MOTOR_SENSORLESS_CTRL_EVENT_RUN	1	Event to run (active)
	MOTOR_SENSORLESS_CTRL_EVENT_ERROR	2	Event of error
	MOTOR_SENSORLESS_CTRL_EVENT_RESET	3	Event to reset

9.7 Speed Control Module

Speed control module calculates speed PI with speed reference which is set by user and rotation speed from current module. And it outputs current reference to current module.

9.7.1 Functions

Functions of speed control module are listed below.

Table 9-25 Functions of speed control module

Function	Description
Speed control	Calculate speed PI and output current reference
Startup process	Open loop control at motor startup

9.7.2 Module configuration diagram

Figure 9-4 shows the module configuration diagram.

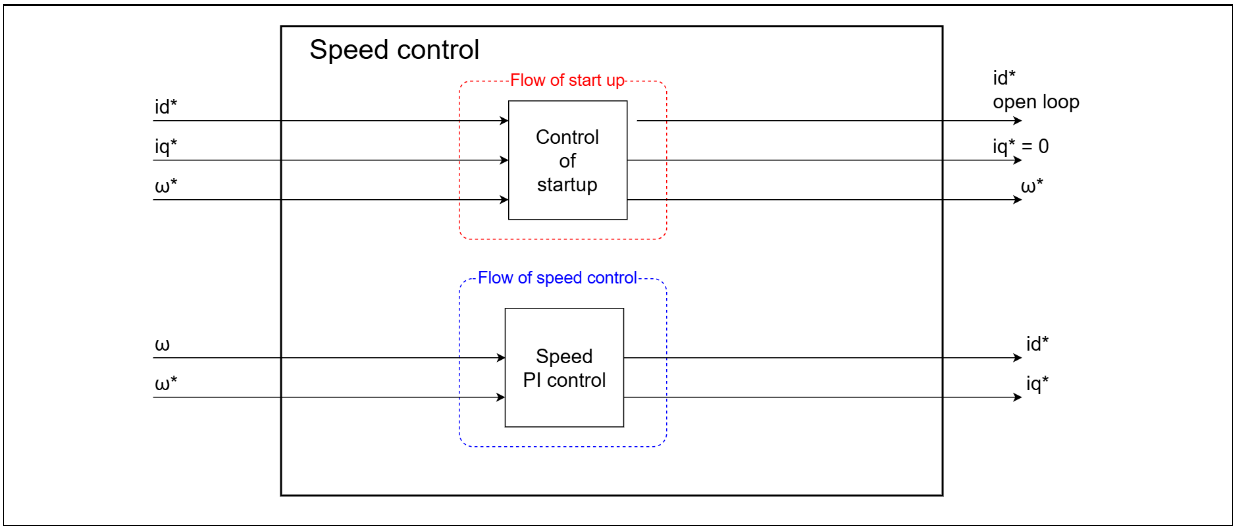


Figure 9-4 Module configuration diagram of speed control module

9.7.3 Start-up method

Figure 9-5 shows startup control of sensorless vector control software. Each mode is controlled by flags managing each reference of the d-axis current, q-axis current, and speed. Speed control module manages startup process.

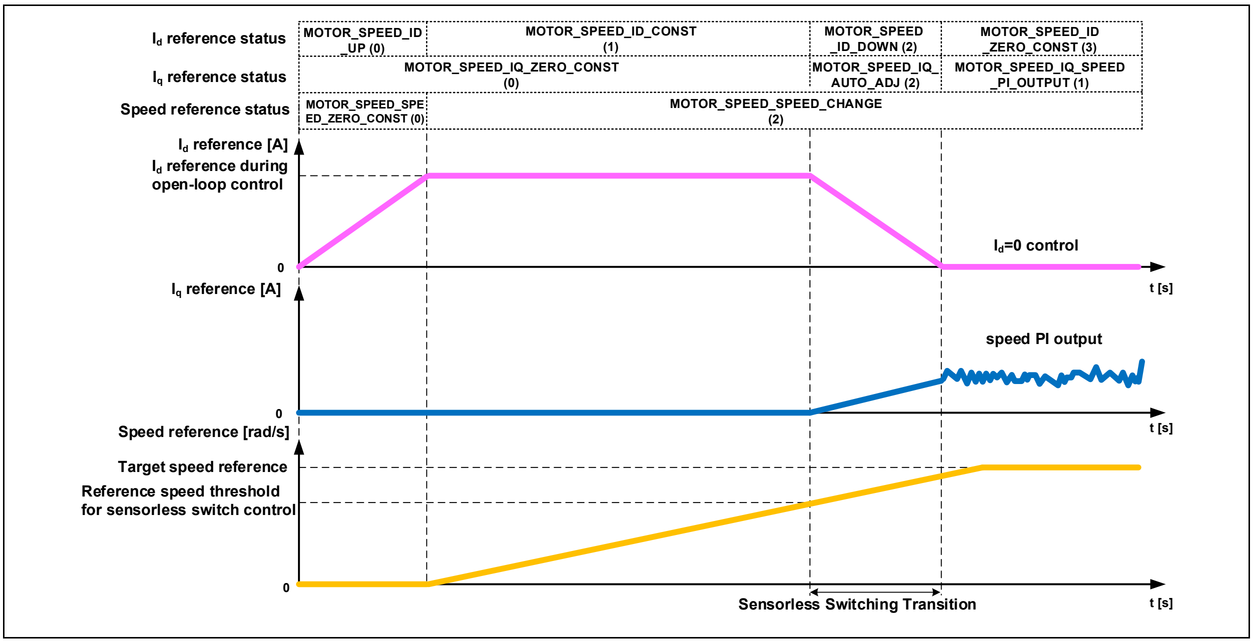


Figure 9-5 Startup Control of Sensorless Vector Control Software

9.7.4 API

Table 9-26 lists API functions of speed control module.

Table 9-26 List of API functions of the speed control module

API	Description
RM_MOTOR_SPEED_Open	Generates instances of speed control module and the modules to be used.
RM_MOTOR_SPEED_Close	Close instances of speed control module and the modules to be used.
RM_MOTOR_SPEED_Reset	Initializes the module.
RM_MOTOR_SPEED_Run	Activates the module.
RM_MOTOR_SPEED_SpeedReferenceSet	Set speed reference.
RM_MOTOR_SPEED_PositionReferenceSet	Set position reference. (Unsupported in sensorless vector)
RM_MOTOR_SPEED_ParameterSet	Set parameters from current control module.
RM_MOTOR_SPEED_SpeedControl	Perform speed control process.
RM_MOTOR_SPEED_ParameterGet	Get parameters of speed control module. (These parameters are set to current control module via interface module.)
RM_MOTOR_SPEED_ParameterUpdate	Updates the control parameters by user setting.

9.7.5 Structure and variable information

Structures and variables of speed control module are listed below.

Table 9-27 List of structures and variables of speed control module [1/2] (rm_motor_speed_api.h)

Structure	Members	Description
motor_speed_callback_args_t	*p_context	Address of context data for callback function
	event	Event data of callback
motor_speed_input_t	f_id	d-axis electrical current
	f_iq	q-axis electrical current
	f_vamax	Maximum value of vector current [A]
	f_speed_rad	Rotation speed (electrical) [rad/s]
	f_position_rad	Rotor position (angle) [rad]
	f_ed	Estimated d-axis voltage [V]
	f_eq	Estimated q-axis voltage [V]
	f_phase_err_rad	Phase error [rad]
	u1_flag_get_iref	Flag of enable getting current reference
	u1_adjust_status	Adjusting (draw in) status at start up (No use in sensorless vector)
	u1_adjust_mode	Adjusting (draw in) mode at start up (No use in sensorless vector)
	u1_adjust_count_full	Flag of full of counter (No use in sensorless vector)
	u1_openloop_status	Open loop status at induction calibration (No use in sensorless vector)
	f_openloop_speed	Speed of open loop at induction calibration (No use in sensorless vector)
	f_openloop_id_ref	d-axis current reference at induction calibration (No use in sensorless vector)
motor_speed_output_t	f_id_ref	d-axis current reference
	f_iq_ref	q-axis current reference
	f_ref_speed_rad_ctrl	Speed reference internally (electrical) [rad/s]
	f_damp_comp_speed	Damping speed at open loop (electrical) [rad/s]
	u1_flag_pi	Flag of started PI control
motor_speed_position_data_t (No use in sensorless vector)	e_step_mode	Position control mode
	e_loop_mode	Control method (Speed or Position)
	position_reference_degree	Position reference [degree]

Table 9-28 List of structures and variables of speed control module [2/2] (rm_motor_speed_api.h)

Structure	Members	Description
motor_speed_cfg_t	*st_input	Address of structure of input data from current control module
	*st_output	Address of structure of output data to current control module
	*p_timer_instance	Address of instance of cyclic timer module
	*p_position_instance	Address of instance of position control module (No use in sensorless vector)
	*p_callback	Address of registered callback function
	*p_context	Address of context data for registered callback function
	*p_extend	Address of structure of extended configuration data to refer
motor_speed_api_t	*open	Function address to open module
	*close	Function address to close module
	*reset	Function address to reset module
	*run	Function address to run (activate)
	*speedReferenceSet	Function address to set speed reference
	*positionReferenceSet	Function address to set position reference (Unsupported in sensorless vector)
	*parameterSet	Function address to set data to speed control module
	*speedControl	Function address to perform speed control
	*parameterGet	Function address to get parameters for setting to current control module
	*parameterUpdate	Function address to update configuration
motor_speed_instance_t	*p_ctrl	Address of structure to be used in the module
	*p_cfg	Address of structure of configuration data
	*p_api	Address of structure of API functions

Table 9-29 List of structures and variables of speed control module [1/5] (rm_motor_speed.h)

Structure	Members	Description
motor_speed_pi_params_t	f_err	Error
	f_kp	Gain parameter for proportional term
	f_ki	Gain parameter for integral term
	f_refi	Integral term
	f_ilimit	Limit of integral term
motor_speed_design_params_t	f_speed_omega	Natural frequency of speed PI [Hz]
	f_speed_zeta	Attenuation coefficient of speed PI
	f_ed_hpf_omega	Natural frequency of HPF [Hz]
	f_ol_damping_zeta	Attenuation coefficient of damping
	f_phase_err_lpf_cut_freq	Cut off frequency of LPF of phase error
	f_observer_omega	Natural frequency of speed observer [Hz] (No use in sensorless vector)
	f_observer_zeta	Attenuation coefficient of speed observer (No use in sensorless vector)
motor_speed_lpf_t	f_pre_output	Previous (last) output data
	f_pre_input	Previous (last) input data
	f_omega_t	Natural frequency of LPF [Hz]
	f_gain_ka	Gain of LPF
	f_gain_kb	Gain of LPF
motor_speed_2nd_order_lpf_t	f4_pre_output	Previous (last) output data
	f4_pre2_output	2 times previous output data
	f4_pre_input	Previous (last) input data
	f4_pre2_input	2 times previous input data
	f4_omega_t	Calculated value
	f4_omega2_t	Calculated value
	f4_omega2_t2	Calculated value
	f4_gain_ka	Gain of LPF
	f4_gain_kb	Gain of LPF
	f4_gain_kc	Gain of LPF
motor_speed_oldamp_sub_t	st_ed_lpf	Structure for LPF
	f_damp_comp_gain	Gain of damping at open loop
	f_fb_speed_limit_rate	Limit rate of speed feedback

Table 9-30 List of structures and variables of speed control module [2/5] (rm_motor_speed.h)

Structure	Members	Description
motor_speed_oldamp_t	f4_ol_id_up_step	Step of increase of d-axis current [A]
	f4_ol_id_down_step	Step of decrease of d-axis current [A]
	f4_ol_iq_down_step_ratio	Ratio of decrease of q-axis current
	f4_ol_id_ref	d-axis current reference at open loop [A]
	f4_id_down_speed_rpm	Threshold of speed to judge decrease current (electrical) [rpm]
	f4_id_up_speed_rpm	Threshold of speed to judge increase current (electrical) [rpm]
	f4_opl2less_sw_time	Period to change to sensorless from open loop
	f4_switch_phase_err_rad	Phase error at change to sensorless
motor_speed_motor_parameter_t	u2_mtr_pp	Pole pairs
	f4_mtr_r	Resistance [ohm]
	f4_mtr_ld	d-axis inductance [H]
	f4_mtr_lq	q-axis inductance [H]
	f4_mtr_m	Magnetic flux [Wb]
	f4_mtr_j	Inertia [kgm^2]
motor_speed_flux_weakening_t	*pmotor	Address of structure of motor parameters
	f4_ia_max	Maximum current of d/q-axis [A]
	f4_va_max	Maximum voltage of d/q-axis [V]
	f4_vfw_ratio	Maximum voltage ratio in flux weakening
	f4_id_demag	Demagnetization current
	f4_id_min	Minimum current of d-axis [A]
	f4_v_fw	Limit of voltage vector in flux weakening
	u2_fw_status	Status of flux weakening

Table 9-31 List of structures and variables of speed control module [3/5] (rm_motor_speed.h)

Structure	Members	Description
motor_speed_observer_t (No use in sensorless vector)	f4_speed_rad	Rotation speed (electrical) [rad/s]
	f4_ref_torque	Torque reference
	f4_ref_pre_torque	Previous (last) torque reference
	f4_ref_speed_rad	Previous speed (electrical) [rad/s]
	f4_ref_pre_speed_rad	Previous (last) speed reference (electrical) [rad/s]
	f4_hpf_k1	Coefficient of HPF #1
	f4_hpf_k2	Coefficient of HPF #2
	f4_hpf_k3	Coefficient of HPF #3
	f4_k1	Coefficient of observer #1
	f4_k2	Coefficient of observer #2
	f4_hpf_ref_speed_rad	Speed reference in HPF (electrical) [rad/s]
	f4_hpf_ref_pre_speed_rad	Previous (last) speed reference in HPF (electrical) [rad/s]
	f4_hpf_omega	Natural frequency of HPF [Hz]
	st_lpf	Structure for LPF
motor_speed_disturbance_observer_t (No use in sensorless vector)	f4_gain_distubance_estimate	Gain of disturbance
	f4_gain_speed_estimate	Gain of speed
	f4_estimated_distubance	Estimated disturbance
	f4_estimated_speed	Estimated speed (electrical) [rad/s]
	f4_inertia	Motor inertia
	f4_ctrl_period	Period of observer process
	st_lpf	Structure for LPF

Table 9-32 List of structures and variables of speed control module [4/5] (rm_motor_speed.h)

Structure	Members	Description
motor_speed_extended_cfg_t	u1_ctrl_method	Control method (Speed or Position) (Only speed is effective in sensorless)
	f_speed_ctrl_period	Period of speed control [Hz]
	f_limit_speed_change	Step of speed at speed change (electrical) [rad/s]
	f_maximum_speed_rpm	Maximum rotation speed (electrical) [rpm]
	f_omega_t	Natural frequency of speed PI [Hz]
	f_id_up_speed_rad	Speed to judge increase d-axis current
	f_iq_limit	Limit of q-axis current [A]
	f_ol_fb_speed_limit_rate	Limit of speed reference
	f_natural_frequency	Natural frequency of observer (No use in sensorless vector)
	u1_openloop_damping	Flag to select open loop damping (enable/disable)
	u1_flux_weakening	Flag to select flux weakening control function (enable/disable)
	u1_less_switch	Flag to select the tolerance function from open loop to sensorless (enable/disable)
	u1_observer_swtich	Flag to select speed observer function (No use in sensorless vector)
	observer_select	Selected method of speed observer (No use in sensorless vector)
	ol_param	Structure of open loop control
	ol_sub_param	Structure of sub data of open loop control
	d_param	Designed parameters for PI control
	control_type	Method of motor control
	mtr_param	Structure of motor parameters

Table 9-33 List of structures and variables of speed control module [5/5] (rm_motor_speed.h)

Structure	Members	Description
motor_speed_instance_ctrl_t	u1_active	Flag of active
	u1_state_speed_ref	State of setting speed reference
	u1_flag_get_iref	Flag to get current reference
	u1_state_id_ref	State of calculation of d-axis current
	u1_state_iq_ref	State of calculation of q-axis current
	f_rpm2rad	Parameter to transform rpm => rad/s
	f_ref_speed_rad_ctrl	Internal speed reference (electrical) [rad/s]
	f_ref_speed_rad	Speed reference (electrical) [rad/s]
	f_speed_lpf_rad	Speed in LPF
	e_status	Status of speed control
	u1_flag_down_to_ol	Flag to switch to open loop
	f_ol_iq_down_step	Step of declaration of q-axis current at open loop [A]
	f_phase_err_rad_lpf	Phase error after LPF
	f_init_phase_err_rad	Initial value of phase error
	f_opl_torque_current	Torque current value to change open loop
	f_damp_comp_speed	Damping speed in open loop (electrical) [rad/s]
	f_damp_comp_gain	Damping gain of open loop
	f_fb_speed_limit_rate	Limit of speed feedback
	u1_enable_flux_weakning	Flag of selection of flux weakening control function (enable/disable)
	st_flxwkn	Structure for flux weakening control function
	*p_cfg	Address of structure of configuration data to refer
	pi_param	Structure for PI control
	st_input	Structure of input data from current control
	st_speed_lpf	Structure for speed LPF
	st_phase_err_lpf	Structure for phase error LPF
	st_observer	Structure of speed observer (No use in sensorless vector)
	st_disturbance_observer	Structure of disturbance speed observer (No use in sensorless vector)
	openloop_sub	Structure of sub data for open loop
	st_position_data	Structure of position control (No use in sensorless vector)
	timer_args	Argument data for timer callback

9.7.6 Macro and enumeration definition

The macros and enumerations for speed control module are listed below.

Table 9-34 List of macros of speed control module [1/2]

File name	Name of MACRO	Defined value	Descriprion
rm_motor_speed.c	MOTOR_SPEED_OPEN	('M' << 24U) | ('T' << 16U) | ('S' << 8U) | ('P' << 0U)	Module opened
	MOTOR_SPEED_FLAG_CLEAR	0	Clear a flag
	MOTOR_SPEED_FLAG_SET	1	Set a flat
	MOTOR_SPEED_MULTIPLE_2	2.0F	2.0
	MOTOR_SPEED_TWOPI	2.0F * 3.1415926535F	2PI
	MOTOR_SPEED_TWOPI_60	MOTOR_SPEED_TWOPI / 60.0F	To transform rpm => rad/s
	MOTOR_SPEED_DIV_8BIT	1.0F / 256.0F	Divided by full 8bit
	MOTOR_SPEED_RAD_TRANS	3.1415926535F / 180.0F	To transform speed
	MOTOR_SPEED_ROOT3	1.7320508F	√3
	MOTOR_SPEED_SPEED_ZERO_CONST	0	Speed reference state Speed set 0 constancy
	MOTOR_SPEED_POSITION_CONTROL	1	Speed reference state Position control (No use in sensorless vector)
	MOTOR_SPEED_SPEED_CHANGE	2	Speed reference state PI control
	MOTOR_SPEED_OPEN_LOOP_INDUCTION	3	Speed reference state Induction calibration open loop (No use in sensorless vector)
	MOTOR_SPEED_ID_UP	0	d-axis reference state d-axis increasing
	MOTOR_SPEED_ID_CONST	1	d-axis reference state d-axis constancy
	MOTOR_SPEED_ID_DOWN	2	d-axis reference state d-axis decreasing
	MOTOR_SPEED_ID_ZERO_CONST	3	d-axis reference state d-axis 0 constancy
	MOTOR_SPEED_ID_FLUXWKN	4	d-axis reference state using flux weakening process
	MOTOR_SPEED_ID_OPENLOOP	5	d-axis reference state Induction calibration open loop
	MOTOR_SPEED_IQ_ZERO_CONST	0	q-axis reference state q-axis 0 constancy
	MOTOR_SPEED_IQ_SPEED_PI_OUTPUT	1	q-axis reference state PI control
	MOTOR_SPEED_IQ_AUTO_ADJ	2	q-axis reference state Draw in at start up
	MOTOR_SPEED_IQ_DOWN	3	q-axis reference state q-axis decreasing

Table 9-35 List of macros of speed control module [2/2]

File name	Name of MACRO	Defined value	Description
rm_motor_speed.c	MOTOR_SPEED_CALCULATE_ANGLE_ADJUST_90DEG	1	Draw in to 90 degrees direction at start up (No use in sensorless vector)
	MOTOR_SPEED_CALCULATE_ANGLE_ADJUST_0DEG	2	Draw in to 0 degree direction at start up (No use in sensorless vector)
	MOTOR_SPEED_CALCULATE_ANGLE_ADJUST_FIN	3	Draw in finished (No use in sensorless vector)
	MOTOR_SPEED_CALCULATE_ANGLE_ADJUST_OPENLOOP	4	Open loop runs at induction calibration (No use in sensorless vector)

Table 9-36 List of Enumeration type for speed control module (rm_motor_speed_api.h)

Enumeration type name	Members	Defined value	Description
motor_speed_event_t	MOTOR_SPEED_EVENT_FORWARD	1	Event before cyclic speed control process
	MOTOR_SPEED_EVENT_BACKWARD	2	Event after cyclic speed control process
	MOTOR_SPEED_EVENT_ENCODER_CYCLIC	3	Event for cyclic encoder process
	MOTOR_SPEED_EVENT_ENCODER_ADJUST	4	Event for draw in process at using encoder sensor
motor_speed_loop_mode_t	MOTOR_SPEED_LOOP_MODE_SPEED	0	Speed control
	MOTOR_SPEED_LOOP_MODE_POSITION	1	Position control (No use in sensorless vector)
motor_speed_step_t (No use in sensorless vector)	MOTOR_SPEED_STEP_DISABLE	0	Calculate position reference
	MOTOR_SPEED_STEP_ENABLE	1	Linear follow position reference

Table 9-37 List of Enumeration type for speed control module (rm_motor_speed.h)

Enumeration type name	Members	Defined value	Description
motor_speed_control_type_t	MOTOR_SPEED_CONTROL_TYPE_SENSORLESS	0	Sensorless
	MOTOR_SPEED_CONTROL_TYPE_ENCODER	1	Use encoder sensor
	MOTOR_SPEED_CONTROL_TYPE_HALL	2	Use hall sensor
	MOTOR_SPEED_CONTROL_TYPE_INDUCTION	3	Use induction sensor
motor_speed_openloop_damping_t	MOTOR_SPEED_OPENLOOP_DAMPING_DISABLE	0	Disable open loop damping function
	MOTOR_SPEED_OPENLOOP_DAMPING_ENABLE	1	Enable open loop damping function
motor_speed_flux_weaken_t	MOTOR_SPEED_FLUX_WEAKEN_DISABLE	0	Disable flux weakening control function
	MOTOR_SPEED_FLUX_WEAKEN_ENABLE	1	Enable flux weakening control function
motor_speed_less_switch_t	MOTOR_SPEED_LESS_SWITCH_DISABLE	0	Disable switching function at open loop to PI control
	MOTOR_SPEED_LESS_SWITCH_ENABLE	1	Enable switching function at open loop to PI control
motor_speed_observer_switch_t (No use in sensorless vector)	MOTOR_SPEED_OBSERVER_SWITCH_DISABLE	0	Disable speed observer function
	MOTOR_SPEED_OBSERVER_SWITCH_ENABLE	1	Enable speed observer function
motor_speed_observer_select_t (No use in sensorless vector)	MOTOR_SPEED_OBSERVER_SELECT_NORMAL	0	Normal observer
	MOTOR_SPEED_OBSERVER_SELECT_DISTURBANCE	1	Disturbance observer
motor_speed_ctrl_status_t	MOTOR_SPEED_CTRL_STATUS_INIT	0	Status of speed control Initialize
	MOTOR_SPEED_CTRL_STATUS_BOOT	1	Status of speed control Boot
	MOTOR_SPEED_CTRL_STATUS_RUN	2	Status of speed control Run (PI control)

9.8 Current Control Module

Current control module calculates PI with current reference from speed control module and detected current by driver module. And it sets calculated PWM duty to driver module. This module has angle module (estimate module). This module sets parameters from driver module to angle module and acquire speed and angle data from angle module. These data present to speed control module and interface module.

9.8.1 Functions

Functions of current control module are listed below.

Table 9-38 Functions of current control module

Function	Description
Current control	Calculate current PI and output PWM duty
Offset adjustment of current A/D	Calculate A/D offset at each phase current detection
Voltage error compensation	Compensation for the error of voltage by PWM deadtime
Transform of axis	Transform current data from U/V/W to d/q-axis. And retransform from d/q-axis to U/V/W.
Modulation	Modulate to PWM signal to improve efficiency.
Decoupling control	Avoid the interference between d-axis and q-axis.
BEMF observer	Estimates angle and rotational speed using BEMF observer.

9.8.2 Module configuration diagram

Figure 9-6 shows the module configuration diagram.

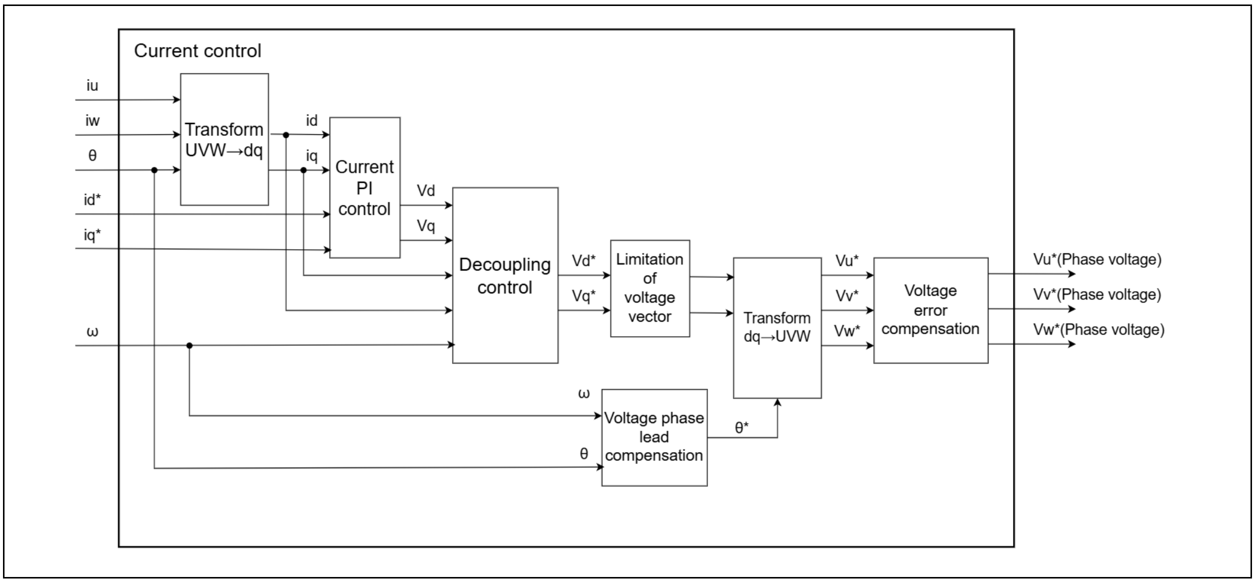


Figure 9-6 Module configuration diagram of current control module

9.8.3 API

API functions for current control are listed below.

Table 9-39 List of API functions for the current control module

API	Description
RM_MOTOR_CURRENT_Open	Generates instances of current control module and the modules to be used.
RM_MOTOR_CURRENT_Close	Close instances of current control module and the modules to be used.
RM_MOTOR_CURRENT_Reset	Initializes current control module and the modules to be used.
RM_MOTOR_CURRENT_Run	Activates current control module.
RM_MOTOR_CURRENT_ParameterSet	Set parameters from speed control module.
RM_MOTOR_CURRENT_CurrentReferenceSet	Set current reference from speed control module.
RM_MOTOR_CURRENT_SpeedPhaseSet	Set speed and phase information.
RM_MOTOR_CURRENT_CurrentSet	Set detected current data.
RM_MOTOR_CURRENT_ParameterGet	Get parameters about current control module. (These parameters are set to speed control module via interface module.)
RM_MOTOR_CURRENT_CurrentGet	Get current and voltage data.
RM_MOTOR_CURRENT_PhaseVoltageGet	Get phase voltage data.
RM_MOTOR_CURRENT_ParameterUpdate	Updates the control parameters by user setting.

9.8.4 Structure and variable information

Structures and variables of current control module are listed below.

Table 9-40 List of structures and variables for the current control module [1/2] (rm_motor_current_api.h)

Structure	Members	Description
motor_current_callback_args_t	*p_context	Address of context data for callback function
	event	Event data of callback
motor_current_output_t	f_id	d-axis electrical current [A]
	f_iq	q-axis electrical current [A]
	f_vamax	Limit of voltage vector [V]
	f_speed_rad	Rotation speed (electrical) [rad/s]
	f_speed_rpm	Rotation speed (mechanical) [rpm]
	f_rotor_angle	Rotor angle [rad]
	f_position_rad	Position data [rad]
	f_ed	Estimated d-axis voltage [V]
	f_eq	Estimated q-axis voltage [V]
	f_phase_err_rad	Phase error [rad]
	u1_flag_get_iref	Flag of start to calculate current reference
	u1_adjust_status	Adjusting (draw in) status at start up (No use in sensorless vector)
	u1_adjust_count_full	Flag of full of counter (No use in sensorless vector)
	u1_openloop_status	Open loop status at induction calibration (No use in sensorless vector)
	f_openloop_speed	Speed of open loop at induction calibration (No use in sensorless vector)
	f_openloop_id_ref	d-axis current reference at induction calibration (No use in sensorless vector)
motor_current_input_t	f_id_ref	d-axis current reference [A]
	f_iq_ref	q-axis current reference [A]
	f_ref_speed_rad_ctrl	Speed reference internally (electrical) [rad/s]
	f_damp_comp_speed	Speed at open loop damping (electrical) [rad/s]
	u1_flag_pi	Flag of start PI control
motor_current_input_current_t	iu	U phase electrical current [A]
	iv	V phase electrical current [A]
	iw	W phase electrical current [A]
motor_current_input_voltage_t	vdc	Inverter bus voltage [V]
	va_max	Limit of voltage vector [V]

Table 9-41 List of structures and variables for the current control module [2/2] (rm_motor_current_api.h)

Structure	Members	Description
motor_current_get_voltage_t	u_voltage	U phase voltage [V] (No use in sensorless vector)
	v_voltage	V phase voltage [V] (No use in sensorless vector)
	w_voltage	W phase voltage [V] (No use in sensorless vector)
	vd_reference	Reference of d-axis voltage
	*vq_reference	Address for reference of q-axis voltage
motor_current_cfg_t	*p_motor_driver_instance	Address of instance of driver module
	*p_motor_angle_instance	Address of instance of angle (estimate) module
	*p_callback	Address of registered callback function
	*p_context	Address of context data for registered callback function
	*p_extend	Address of structure of extended configuration data to refer
motor_current_api_t	*open	Function address to open module
	*close	Function address to close module
	*reset	Function address to reset module
	*run	Function address to run module (start motor rotation)
	*parameterSet	Function address to set parameters from speed control module
	*currentReferenceSet	Function address to set current reference
	*speedPhaseSet	Function address to set rotation speed and rotor angle
	*currentSet	Function address to set each phase current
	*parameterGet	Function address to get parameters to output to speed control module
	*currentGet	Function address to get current data
	*phaseVoltageGet	Function address to get voltage data
	*parameterUpdate	Function address to update configuration
motor_current_instance_t	*p_ctrl	Address of structure to be used in the module
	*p_cfg	Address of structure of configuration data
	*p_api	Address of structure of API functions

Table 9-42 List of structures and variables for the current control module [1/4] (rm_motor_current.h)

Structure	Members	Description
motor_current_pi_params_t	f_err	Error
	f_kp	Gain parameter for proportional term
	f_ki	Gain parameter for integral term
	f_refi	Integral term
	f_ilimit	Limit of integral term
motor_current_design_parameter_t	f_current_omega	Natural frequency of current PI [Hz]
	f_current_zeta	Attenuation coefficient of current PI
motor_current_motor_parameter_t	u2_mtr_pp	Pole pairs
	f4_mtr_r	Resistance [ohm]
	f4_mtr_ld	d-axis inductance [H]
	f4_mtr_lq	q-axis inductance [H]
	f4_mtr_m	Magnetic flux [Wb]
	f4_mtr_j	Inertia [kgm^2]
motor_currnt_voltage_compensation_t	f_comp_v	Correction of voltage
	f_comp_i	Correction of current
	f_slope	Coefficient of line interpolation
	f_intcept	Coefficient of line interpolation
	f_volt_comp_array	Array of voltage compensation
	f_vdc	Inverter bus voltage [V]
	f_volt_comp_limit	Limit of voltage compensation
	u1_volt_err_comp_enable	Selection of voltage error compensation function (enable/disable)

Table 9-43 List of structures and variables for the current control module [2/4] (rm_motor_current.h)

Structure	Members	Description
motor_current_extended_cfg_t	u1_control_type	Sensor type (sensorless/encoder/hall/induction)
	shunt	Shunt type (1/2/3)
	f_comp_v	Table of voltage for voltage error compensation
	f_comp_i	Table of current for voltage error compensation
	vcomp_enable	Selection of voltage compensation function (enable/disable)
	u1_sample_delay_comp_enable	Selection of sample delay compensation function (enable/disable)
	f_period_magnitude_value	Ratio of period for sample delay compensation
	f_current_ctrl_period	Period of current control [sec]
	f_ilimit	Limit of current [A]
	*p_motor_parameter	Address of structure of motor parameters
	*p_design_parameter	Address of structure of current PI control

Table 9-44 List of structures and variables for the current control module [3/4] (rm_motor_current.h)

Structure	Members	Description
motor_current_instance_ctrl_t	open	Module opened information
	u1_active	A flag of module active
	f_vd_ref	d-axis voltage reference [V]
	f_vq_ref	q-axis voltage reference [V]
	f_id_ref	d-axis current reference [A]
	f_iq_ref	q-axis current reference [A]
	f_iu_ad	U phase current [A]
	f_iv_ad	V phase current [A]
	f_iw_ad	W phase current [A]
	f_id_ad	d-axis current [A]
	f_iq_ad	q-axis current [A]
	f_vdc_ad	Inverter bus voltage [V]
	f_speed_rad	Rotation speed (electrical) [rad/s]
	f_rotor_angle	Rotor angle [rad]
	f_position_rad	Position data [rad]
	f_refu	U phase voltage reference [V]
	f_refv	V phase voltage reference [V]
	f_refw	W phase voltage reference [V]
	f_va_max	Limit of voltage vector
	f_ed	Estimated d-axis voltage
	f_eq	Estimated q-axis voltage
	f_phase_err	Phase error
	u1_flag_crnt_offset	Flag of getting current offset
	f_sin_ad_data	Sine input data of induction sensor (No use in sensorless vector)
	f_cos_ad_data	Cosine input data of induction sensor (No use in sensorless vector)
	*p_cfg	Address of structure of configuration data to refer

Table 9-45 List of structures and variables for the current control module [4/4] (rm_motor_current.h)

Structure	Members	Description
motor_current_instance_ctrl_t	st_pi_id	Structure for d-axis PI control
	st_pi_iq	Structure for q-axis PI control
	st_vcomp	Structure for voltage error compensation
	st_input	Structure for input data from speed control module
	*p_angle_instance	Address of instance of angle (estimate) module
	*p_driver_instance	Address of instance of driver module

9.8.5 Macro and enumeration definition

The macros and enumerations for current control module are listed below.

Table 9-46 List of macros of current control module

File name	Name of MACRO	Defined value	Description
rm_motor_current.c	MOTOR_CURRENT_OPEN	('M' << 24U) | ('T' << 16U) | ('C' << 8U) | ('T' << 0U)	Opened information
	MOTOR_CURRENT_FLG_CLR	0	Clear a flag
	MOTOR_CURRENT_FLG_SET	1	Set a flag
	MOTOR_CURRENT_TWOPI	2.0F * 3.1415926535F	2PI
	MOTOR_CURRENT_60_TWOPI	60.0F / MOTOR_CURRENT_TWOPI	To transform rad/s => rpm
	MOTOR_CURRENT_SQRT_2	1.41421356F	√2
	MOTOR_CURRENT_SQRT_3	1.7320508F	√3
	MOTOR_CURRENT_DIV_KHZ	0.001F	To transform kHz => Hz

Table 9-47 List of Enumeration type for current control module (rm_motor_current_api.h)

Enumeration type name	Members	Defined value	Description
motor_current_event_t	MOTOR_CURRENT_EVENT_FORWARD	1	Callback event before cyclic current control process
	MOTOR_CURRENT_EVENT_DATA_SET	2	Callback event for cyclic current control process
	MOTOR_CURRENT_EVENT_BACKWARD	3	Callback event after cyclic current control process

Table 9-48 List of Enumeration type for current control module (rm_motor_current_api.h)

Enumeration type name	Members	Defined value	Description
motor_current_control_type_t	MOTOR_CURRENT_CONTROL_TYPE_SENSORLESS	0	Sensorless
	MOTOR_CURRENT_CONTROL_TYPE_ENCODER	1	Encoder
	MOTOR_CURRENT_CONTROL_TYPE_HALL	2	Hall
	MOTOR_CURRENT_CONTROL_TYPE_INDUCTION	3	Induction
motor_current_shunt_type_t	MOTOR_CURRENT_SHUNT_TYPE_1_SHUNT	1	1shunt
	MOTOR_CURRENT_SHUNT_TYPE_2_SHUNT	2	2shunts
	MOTOR_CURRENT_SHUNT_TYPE_3_SHUNT	3	3shunts
motor_current_voltage_compensation_select_t	MOTOR_CURRENT_VOLTAGE_COMPENSATION_SELECT_DISABLE	0	Disable voltage compensation function
	MOTOR_CURRENT_VOLTAGE_COMPENSATION_SELECT_ENABLE	1	Enable voltage compensation function
motor_current_sample_delay_compensation_t	MOTOR_CURRENT_SAMPLE_DELAY_COMPENSATION_DISABLE	0	Disable sample delay compensation function
	MOTOR_CURRENT_SAMPLE_DELAY_COMPENSATION_ENABLE	1	Enable sample delay compensation function

9.9 Driver Module

The driver module works as an interface between each motor modules and MCU peripherals. Appropriately configuring the driver module allows you to use microcontroller function allocation and the differences of the board to be used without modifying the motor module.

9.9.1 Functions

Table 9-49 lists the functions of the driver module.

Table 9-49 List of functions of the driver module

Functions	Description
Acquisition of the A/D conversion value	Acquires A/D converted values such as the phase current and inverter bus voltage.
PWM duty setting	Sets the PWM duty value that is to be output to U, V, and W-phases.
PWM start/stop	Controls whether to start or stop (active or inactive) of PWM output.

9.9.2 Module configuration diagram

The module configuration of driver module is shown below.

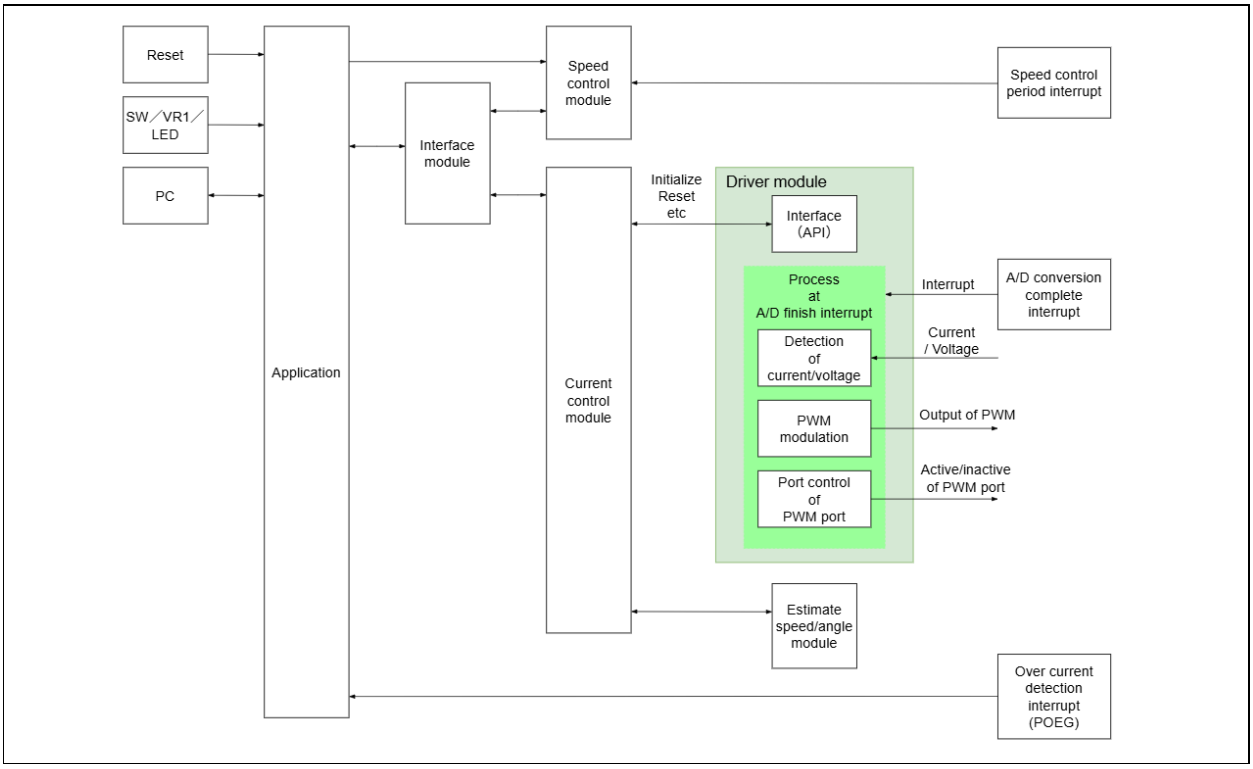


Figure 9-7 Module configuration diagram of driver module

9.9.3 API

Table 9-50 lists and describes the API functions for driver module.

Table 9-50 List of API functions for the driver module

API	Description
RM_MOTOR_DRIVER_Open	Generate instances of driver module and the modules to be used.
RM_MOTOR_DRIVER_Close	Close instances of driver module and the modules to be used.
RM_MOTOR_DRIVER_Reset	Reset the module.
RM_MOTOR_DRIVER_PhaseVoltageSet	Set phase and PWM voltage of each phase.
RM_MOTOR_DRIVER_CurrentGet	Get each phase current and inverter bus voltage.
RM_MOTOR_DRIVER_FlagCurrentOffsetGet	Perform the measurement of A/D offset and get the finish flag of measurement.
RM_MOTOR_DRIVER_CurrentOffsetRestart	Restart current offset measurement.
RM_MOTOR_DRIVER_ParameterUpdate	Updates the control parameters by user setting.

9.9.4 Structure and variable information

Structures and variables of driver module are listed below.

Table 9-51 List of structures and variables for driver module [1/2] (rm_motor_driver_api.h)

Structure	Members	Description
motor_driver_callback_args_t	event	Event data of callback
	*p_context	Address of context data for callback function
motor_driver_current_get_t	iu	U phase electrical current [A]
	iv	V phase electrical current [A]
	iw	W phase electrical current [A]
	vdc	Inverter bus voltage [V]
	va_max	Limit of voltage vector
	sin_ad	sine signal input of induction sensor (No use in sensorless vector)
	cos_ad	cosine signal input of induction sensor (No use in sensorless vector)
motor_driver_cfg_t	*p_adc_instance	Address of instance of A/D conversion module
	iu_ad_ch	A/D channel for U phase current
	iv_ad_ch	A/D channel for V phase current
	iw_ad_ch	A/D channel for W phase current
	vdc_ad_ch	A/D channel for inverter bus voltage
	sin_ad_ch	A/D channel for sine signal of induction sensor (No use in sensorless vector)
	cos_ad_ch	A/D channel for cosine signal of induction sensor (No use in sensorless vector)
	*p_adc2_instance	Address of instance of 2nd A/D conversion module (This is used at 2unit ADC peripheral using.)
	shunt	Shunt type
	*p_three_phase_instance	Address of instance of three phase PWM control module
	*p_callback	Address of registered callback function
	*p_context	Address of context data for registered callback function
	*p_extend	Address of structure of extended configuration data to refer

Table 9-52 List of structures and variables for driver module [2/2] (rm_motor_driver_api.h)

Structure	Members	Description
motor_driver_api_t	*open	Function address to open module
	*close	Function address to close module
	*reset	Function address to reset module
	*phaseVoltageSet	Function address to set each phase voltage
	*currentGet	Function address to get current and voltage data
	*flagCurrentOffsetGet	Function address to perform measurement of current offset and to get finish flag
	*currentOffsetRestart	Function address to restart measurement of current offset
	*parameterUpdate	Function address to update configuration
motor_driver_instance_t	*p_ctrl	Address of structure to be used in the module
	*p_cfg	Address of structure of configuration data
	*p_api	Address of structure of API functions

Table 9-53 List of structures and variables for driver module [1/4] (rm_motor_direver.h)

Structure	Members	Description
motor_driver_modulation_t	f4_vdc	Inverter bus voltage [V]
	f4_1_div_vdc	Divided by inverter bus voltage (to calculate)
	f4_voltage_error_ratio	Error ratio of voltage
	f4_max_duty	Maximum PWM duty
	f4_min_duty	Minimum PWM duty
	f4_neutral_duty	Neutral PWM duty (0[V] output)
	u1_sat_flag	Saturation flag
motor_driver_shared_instance_ctrl_t	open	Shared module open information
	registered_motor_count	Registered motor count in shared module
	*p_context	Address of context data for shared module
motor_driver_extended_shared_cfg_t	*p_adc_instance_first	Address of instance of 1st A/D conversion module
	*p_adc_instance_second	Address of instance of 2nd A/D conversion module
	*p_shared_instance_ctrl	Address of structure of variables to be used by shared module

Table 9-54 List of structures and variables for driver module [2/4] (rm_motor_direver.h)

Structure	Members	Description
motor_driver_extended_cfg_t	u2_pwm_timer_freq	PWM timer clock frequency [MHz]
	u2_pwm_carrier_freq	PWM carrier frequency [kHz]
	pwm_carrier_freq	PWM carrier frequency [kHz]
	u2_deadtime	Dead time
	f_current_range	Maximum detected current value [A]
	f_vdc_range	Maximum detected inverter bus voltage value [V]
	f_ad_resolution	Resolution of A/D conversion
	f_ad_current_offset	Current offset at A/D port (0[A] digits)
	f_ad_voltage_conversion	A/D conversion level
	u2_offset_calc_count	Counts to measure current offset
	modulation_method	Modulation method
	port_up	PWM U upper phase port number
	port_un	PWM U lower phase port number
	port_vp	PWM V upper phase port number
	port_vn	PWM V lower phase port number
	port_wp	PWM W upper phase port number
	port_wn	PWM W lower phase port number
	f_ad_current_adjust	Adjustment value for current conversion for 1shunt detection
	s4_difference_minimum	Minimum difference of PWM duty
	s4_adjust_adc_delay	Adjustment delay of A/D sampling timing for 1shunt detection
	trigger_phase	Trigger phase of interrupt to get 1shunt current

Table 9-55 List of structures and variables for driver module [3/4] (rm_motor_direver.h)

Structure	Members	Description
motor_driver_extended_cfg_t	adc_group	Which ADC scan group to use in ADC_B module
	iu_ad_unit	A/D unit number for U phase current
	iv_ad_unit	A/D unit number for V phase current
	iw_ad_unit	A/D unit number for W phase current
	vdc_ad_unit	A/D unit number for inverter bus voltage
	sin_ad_unit	A/D unit number for sine signal of induction sensor (No use in sensorless vector)
	cos_ad_unit	A/D unit number for cosine signal of induction sensor (No use in sensorless vector)
	mod_param	Structure of modulation variables
	interrupt_adc	Which ADC module happens A/D conversion end interrupt
	*p_shared_cfg	Address of configuration data of shared module

Table 9-56 List of structures and variables for driver module [4/4] (rm_motor_direver.h)

Structure	Members	Description
motor_driver_instance_ctrl_t	open	Module open information
	u2_carrier_base	PWM carrier counts
	u2_deadtime_count	Deadtime counts
	f_iu_ad	U phase electrical current [A]
	f_iv_ad	V phase electrical current [A]
	f_iw_ad	W phase electrical current [A]
	f_vdc_ad	Inverter bus voltage [V]
	f_sin_ad	sine signal input of induction sensor (No use in sensorless vector)
	f_cos_ad	cosine signal input of induction sensor (No use in sensorless vector)
	f_refu	U phase voltage reference [V]
	f_refv	V phase voltage reference [V]
	f_refw	W phase voltage reference [V]
	u1_flag_offset_calc	Flag of finish of current offset measurement
	u2_offset_calc_times	Counter to measure current offset
	f_offset_iu	Current offset of U phase current
	f_offset_iv	Current offset of V phase current
	f_offset_iw	Current offset of W phase current
	f_sum_iu_ad	Summation of current offset to calculate for U phase current
	f_sum_iv_ad	Summation of current offset to calculate for V phase current
	f_sum_iw_ad	Summation of current offset to calculate for W phase current
	min_phase	Minimum duty phase (for 1shunt)
	mid_phase	Medium duty phase (for 1shunt)
	u4_gtioca_low_cfg	GTIOCA configuration for low output
	u4_gtiocb_low_cfg	GTIOCB configuration for low output
	u1_flag_port_enable	Flag of PWM port active/inactive
	st_modulation	Structure of modulation
	*p_cfg	Address of configuration data to refer
	adc_callback_args	Arguments for A/D conversion end interrupt callback
	timer_callback_args	Arguments for timer interrupt callback
	*p_shared_instance_ctrl	Address of structure to be used in shared module

9.9.5 Macro and enumeration definition

The macros and enumerations for driver module are listed below.

Table 9-57 List of macros of driver module

File name	Name of MACRO	Defined value	Description
rm_motor_driver.c	MOTOR_DRIVER_OPEN	('M' << 24U) | ('T' << 16U) | ('D' << 8U) | ('R' << 0U)	Open information of driver module
	MOTOR_DRIVER_SHARED_ADC_OPEN	('M' << 24U) | ('T' << 16U) | ('S' << 8U) | ('A' << 0U)	Open information of shared module
	MOTOR_DRIVER_FLG_CLR	0	Clea a flag
	MOTOR_DRIVER_FLG_SET	1	Set a flag
	MOTOR_DRIVER_KHZ_TRANS	1000U	To transform kHz => Hz
	MOTOR_DRIVER_DEF_HALF	0.5F	0.5
	MOTOR_DRIVER_MULTIPLE_TWO	2.0F	2.0
	MOTOR_DRIVER_ADC_DATA_MASK	0x00000FFF	Mask of A/D converted data
	MOTOR_DRIVER_METHOD_SPWM	0	Modulation method SPWM
	MOTOR_DRIVER_METHOD_SVPWM	1	Modulation method SVPWM
	MOTOR_DRIVER_SATFLAG_BITU	1 << 0	Shift data of Saturation bit for U phase
	MOTOR_DRIVER_SATFLAG_BITV	1 << 1	Shift data of Saturation bit for V phase
	MOTOR_DRIVER_SATFLAG_BITW	1 << 2	Shift data of Saturation bit for W phase
	MOTOR_DRIVER_VDC_TO_VAMAX_MULT	0.6124F	Coefficient to transform limit of voltage vector
	MOTOR_DRIVER_SVPWM_MULT	1.155F	√4/3
	MOTOR_DRIVER_IO_PORT_CFG_LOW	0x3000004	IO port configuration for output low level
	MOTOR_DRIVER_IO_PORT_CFG_HIGH	0x3000005	IO port configuration for output high level
	MOTOR_DRIVER_IO_PORT_PERIPHERAL_MASK	0x0010000	Mask of IO port configuration for peripheral mode
	MOTOR_DRIVER_IO_PORT_GPIO_MASK	0xFFEFFFF	Mask of IO port configuration for general mode

Table 9-58 List of Enumeration type for driver module (rm_motor_driver_api.h)

Enumeration	Members	Defined value	Description
motor_driver_event_t	MOTOR_DRIVER_EVENT_FORWARD	1	Callback event before driver module process
	MOTOR_DRIVER_EVENT_CURRENT	2	Callback event for calculation of A/D converted data
	MOTOR_DRIVER_EVENT_BACKWARD	3	Callback event after driver module process
motor_driver_shunt_type_t	MOTOR_DRIVER_SHUNT_TYPE_1_SHUNT	1	1shunt
	MOTOR_DRIVER_SHUNT_TYPE_2_SHUNT	2	2shunts
	MOTOR_DRIVER_SHUNT_TYPE_3_SHUNT	3	3shunts

Table 9-59 List of Enumeration type for driver module (rm_motor_driver.h)

Enumeration	Members	Defined value	Description
motor_driver_select_adc_instance_t	MOTOR_DRIVER_SELECT_ADC_INSTANCE_FIRST	0	1st ADC module instance
	MOTOR_DRIVER_SELECT_ADC_INSTANCE_SECOND	1	2nd ADC module instance
motor_driver_modulation_method_t	MOTOR_DRIVER_MODULATION_METHOD_SPWM	0	Modulation method SPWM
	MOTOR_DRIVER_MODULATION_METHOD_SVPWM	1	Modulation method SVPWM
motor_driver_phase_t	MOTOR_DRIVER_PHASE_U_PHASE	0	U phase
	MOTOR_DRIVER_PHASE_V_PHASE	1	V phase
	MOTOR_DRIVER_PHASE_W_PHASE	2	W phase

9.10 Estimation of Angle/Speed Module

Estimate module estimates rotation speed and rotor angle with BEMF observer with detected phase currents.

9.10.1 Functions

Table 9-60 lists the functions of the estimate module.

Table 9-60 List of functions of the estimate module

Functions	Description
Estimation of speed and angle	Estimate rotation speed and rotor angle with BEMF observer.

9.10.2 Module configuration diagram

Figure 9-6 shows the module configuration diagram.

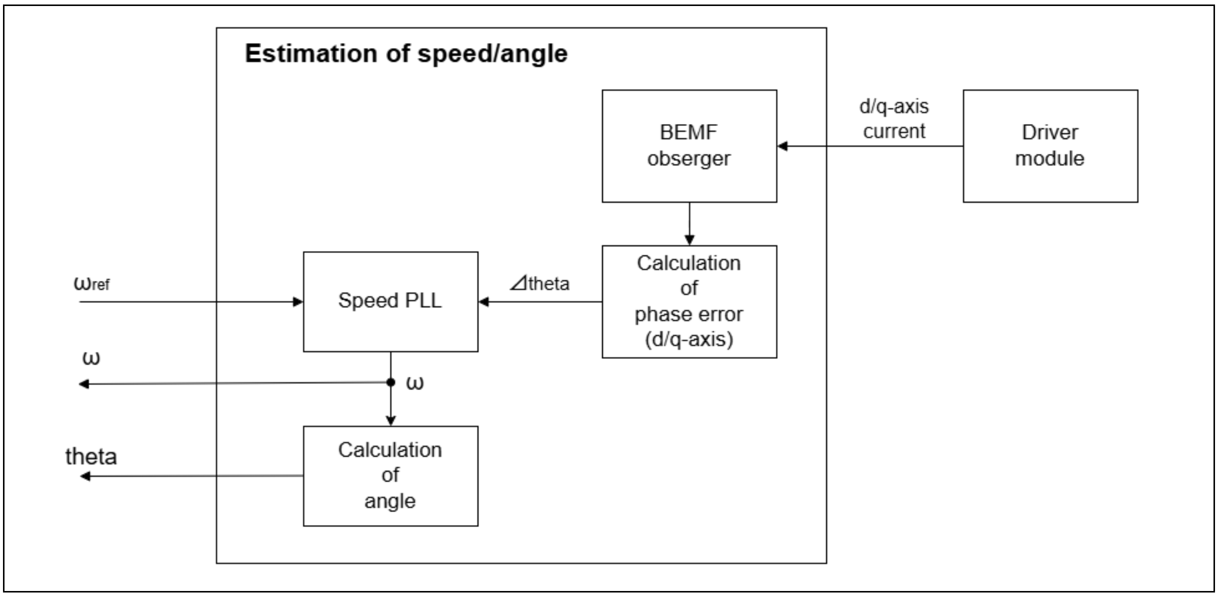


Figure 9-8 Module configuration diagram of estimate module

9.10.3 API

Table 9-61 lists and describes the API functions for estimate module.

Table 9-61 List of API functions for estimate module

API	Description
RM_MOTOR_ESTIMATE_Open	Generate instance of estimate module.
RM_MOTOR_ESTIMATE_Close	Close instance of estimate module.
RM_MOTOR_ESTIMATE_Reset	Reset the module.
RM_MOTOR_ESTIMATE_CurrentSet	Set current data.
RM_MOTOR_ESTIMATE_SpeedSet	Set speed (electrical) [rad/s] data.
RM_MOTOR_ESTIMATE_FlagPiCtrlSet	Set a flag to start PI control.
RM_MOTOR_ESTIMATE_AngleSpeedGet	Get estimated rotor angle and rotation speed.
RM_MOTOR_ESTIMATE_EstimatedComponentGet	Get estimated voltages.
RM_MOTOR_ESTIMATE_ParameterUpdate	Updates the control parameters by user setting.
RM_MOTOR_ESTIMATE_InternalCalculate	Perform internal calculation. (Unsupported in sensorless vector)
RM_MOTOR_ESTIMATE_AngleAdjust	Perform a draw in at start up. (Unsupported in sensorless vector)
RM_MOTOR_ESTIMATE_EncoderCyclic	Perform cyclic process for encoder. (Unsupported in sensorless vector)
RM_MOTOR_ESTIMATE_InfoGet	Get information about calibration of induction sensors. (Unsupported in sensorless vector)
RM_MOTOR_ESTIMATE_CyclicProcess	Perform cyclic process at period interrupt. (Unsupported in sensorless vector)
RM_MOTOR_ESTIMATE_SensorDataSet	Set data of induction sensors. (Unsupported in sensorless vector)

9.10.4 Structure and variable information

Structures and variables of estimate module are listed below.

Table 9-62 Structures of estimate module [1/2] (rm_motor_angle_api.h)

Structure	Member	Description
motor_angle_cfg_t	*p_context	Address of context data of estimation of angle/speed module
	*p_extend	Address of extended configuration data to refer
motor_angle_current_t	id	d-axis electrical current [A]
	iq	q-axis electrical current [A]
motor_angle_voltage_reference_t	vd	d-axis voltage reference [V]
	vq	q-axis voltage reference [V]
motor_angle_ad_data_t (No use in sensorless vector)	sin_ad_data	sin signal data of induction sensor
	cos_ad_data	cos signal data of induction sensor
motor_angle_encoder_info_t (No use in sensorless vector)	e_adjust_status	Draw in status at start up
	u1_adjust_count_full	Flag of counter full about time manage at start up
	e_open_loop_status	Open loop status of induction calibration
	f_openloop_speed	Speed of open loop at induction calibration
	f_openloop_id_ref	d-axis current of open loop at induction calibration

Table 9-63 Structures of estimate module [2/2] (rm_motor_angle_api.h)

Structure	Member	Description
motor_angle_api_t	*open	Function address to open module
	*close	Function address to close module
	*reset	Function address to reset module
	*currentSet	Function address to set current data
	*speedSet	Function address to set speed (electrical) [rad/s] data
	*flagPiCtrlSet	Function address to set a flag of start PI control
	*internalCalculate	Function address to perform internal calculation (Unsupported in sensorless vector)
	*angleSpeedGet	Function address to get rotation speed and rotor angle
	*angleAdjust	Function address to perform draw in at start up (Unsupported in sensorless vector)
	*encoderCyclic	Function address to perform cyclic process of encoder (Unsupported in sensorless vector)
	*cyclicProcess	Function address to perform cyclic process (Unsupported in sensorless vector)
	*sensorDataSet	Function address to set induction sensor data (Unsupported in sensorless vector)
	*estimatedComponentGet	Function address to get estimated component
	*infoGet	Function address to get information about induction calibration (Unsupported in sensorless vector)
	*parameterUpdate	Function address to update configuration
motor_angle_instance_t	*p_ctrl	Address of structure to be used in the module
	*p_cfg	Address of structure of configuration data
	*p_api	Address of structure of API functions

Table 9-64 Structures of estimate module (rm_motor_estimate.h)

Structure	Member	Description
motor_estimate_bemf_obs_axis_t	f4_k_e_obs_1	Proportional parameter for BEMF observer #1
	f4_k_e_obs_2	Proportional parameter for BEMF observer #2
	f4_i_pre	Previous (last) current value
	f4_i_est_pre	Previous (last) estimated current value
	f4_d_est	Estimated disturbance voltage
	f4_d_est_pre	Previous estimated disturbance voltage
	f4_d_est_limit	Integrated limit of estimated disturbance voltage
motor_estimate_motor_parameter_t	u2_mtr_pp	Pole pairs
	f4_mtr_r	Resistance [ohm]
	f4_mtr_ld	d-axis inductance [H]
	f4_mtr_lq	q-axis inductance [H]
	f4_mtr_m	Magnetic flux [Wb]
	f4_mtr_j	Inertia [kgm^2]
	f4_mtr_nominal_current	Nominal current [A]
motor_estimate_bemf_observer_t	f4_dt	Period of calculate BEMF observer
	st_motor_params	Structure for motor parameters
	st_d_axis	Estimated d-axis value
	st_q_axis	Estimated q-axis value
motor_estimate_pll_est_t	f4_kp_est_speed	Proportional value of speed PLL
	f4_ki_est_speed	Integrated value of speed PLL
	f4_i_est_speed	Integrated part of speed PLL
motor_estimate_input_t	f_vd_ref	d-axis voltage reference [V]
	f_vq_ref	q-axis voltage reference [V]
	f_id	d-axis electrical current [A]
	f_iq	q-axis electrical current [A]
	f4_ref_speed_rad_ctrl	Speed reference (electrical) [rad/s]
	f4_damp_comp_speed	Speed of open loop damping (electrical) [rad/s]
motor_estimate_extended_cfg_t	openloop_damping	Structure for open loop damping
	f_e_obs_omega	Natural frequency of BEMF observer [Hz]
	f_e_obs_zeta	Attenuation coefficient of BEMF observer
	f_pll_est_omega	Natural frequency of speed PLL [Hz]
	f_pll_est_zeta	Attenuation coefficient of speed PLL
	f4_ctrl_period	Control period [Hz]
	st_motor_params	Structure of motor parameters
motor_estimate_instance_ctrl_t	open	Open information
	f4_ed	d-axis estimated voltage [V]
	f4_eq	q-axis estimated voltage [V]
	f4_speed_rad	Speed reference (electrical) [rad/s]
	f4_phase_err_rad	Phase error [rad]
	f4_angle_rad	Rotor angle [rad]
	u1_flg_pi_ctrl	Flag of started PI control
	u1_flg_pll_start	Flag of started speed PLL
	st_bemf_obs	Structure for BEMF observer
	st_pll_est	Structure for speed PLL
	st_input	Structure of input data from current control module
	*p_cfg	Address of configuration data to refer

9.10.5 Macro and enumeration definition

The macros and enumerations for estimate module are listed below.

Table 9-65 List of macros of estimate module

File name	Name of MACRO	Defined value	Description
rm_motor_estimate.c	MOTOR_ESTIMATE_OPEN	('M' << 24U) | ('T' << 16U) | ('E' << 8U) | ('S' << 0U)	Open information
	MOTOR_ESTIMATE_FLG_CLR	0U	Clear a flag
	MOTOR_ESTIMATE_FLG_SET	1U	Set a flag
	MOTOR_ESTIMATE_TWOPI	2.0F * 3.1415926535F	2PI

Table 9-66 List of enumeration of estimate module (rm_motor_angle_api.h)

Enumeration	Member	Value	Description
motor_sense_encoder_angle_adjust_t (No use in sensorless vector)	MOTOR_SENSE_ENCODER_ANGLE_ADJUST_90_DEGREE	1	Draw in 90-degree direction at start up
	MOTOR_SENSE_ENCODER_ANGLE_ADJUST_0_DEGREE	2	Draw in 0 degree direction at start up
	MOTOR_SENSE_ENCODER_ANGLE_ADJUST_FINISH	3	Draw in finished
	MOTOR_SENSE_ENCODER_ANGLE_ADJUST_OPENLOOP	4	Open loop at induction calibration
motor_angle_open_loop_t (No use in sensorless vector)	MOTOR_ANGLE_OPEN_LOOP_INACTIVE	0	Open loop is inactive at Induction calibration
	MOTOR_ANGLE_OPEN_LOOP_ACTIVE	1	Open loop is active at induction calibration
motor_angle_error_t (No use in sensorless vector)	MOTOR_ANGLE_ERROR_NONE	0	Induction calibration is success
	MOTOR_ANGLE_ERROR_INDUCTION	1	Induction calibration is error

Table 9-67 List of enumeration of estimate module (rm_motor_estimate.h)

Enumeration	Member	Value	Description
motor_estimate_openloop_damping_t	MOTOR_ESTIMATE_OPENLOOP_DAMPING_DISABLE	0	Disable open loop damping
	MOTOR_ESTIMATE_OPENLOOP_DAMPING_ENABLE	1	Enable open loop damping

10. Setting the parameters

10.1 Overview

In this sample program, initial data of each module can be set by FSP configurator. Set values are automatically reflected to common_data.c/h and hal_data.c/h at code generation. Set values are also included to need parameters of each module at initial process (open process).

A part of parameter can be changed by RMW. Target parameters are listed in “9.5.3”. How to change values is referred to RMW manual.

10.2 List of Interface module parameters

The changeable parameters that are used in Interface module are listed in Table 10-1. And initial value are listed in Table 10-2 and Table 10-3. Setting can be changed in “Property” TAB on e^2 studio FSP. About other modules, same method can be used.

Table 10-1 Configuration Options (rm_motor_sensorless)

Option name	Description
Limit of over current (A)	When electric current over this value, PWM ports are set to OFF.
Limit of over voltage (V)	When entered inverter bus voltage over this value, PWM ports are set to OFF.
Limit of over speed (rpm)	When rotation speed over this value, PWM ports are set to OFF.
Limit of low voltage (V)	When entered inverter bus voltage below this value, PWM ports are set to OFF.
Callback	Callback function to be called at Speed/Current cyclic process.

Table 10-2 Initial value of configuration options [1/2] (rm_motor_sensorless)

Option name	RA6T2	RA6T3	RA4T1
Limit of over current (A)	1.67	1.67	1.67
Limit of over voltage (V)	60.0	60.0	60.0
Limit of over speed (rpm)	4500.0	4500.0	4500.0
Limit of low voltage (V)	8.0	8.0	8.0
Callback	mtr_callback_event	mtr_callback_event	mtr_callback_event

Table 10-3 Initial value of configuration options [2/2] (rm_motor_sensorless)

Option name	RA8T1	RA8T2
Limit of over current (A)	1.67	1.67
Limit of over voltage (V)	60.0	60.0
Limit of over speed (rpm)	4500.0	4500.0
Limit of low voltage (V)	8.0	8.0
Callback	mtr_callback_event	mtr_callback_event

10.3 List of Speed control module parameters

The changeable parameters that are used in speed control module and these initial values are listed below.

Table 10-4 Configuration Options (rm_motor_speed)

Options	Description
Common | Position support	Support position control
General | Speed control period (sec)	The period of speed control process [sec].
General | Step of speed climbing (rpm)	The step of speed fluctuation (mechanical) [rpm]. Program controls speed by this step at acceleration and deceleration.
General | Maximum rotational speed (rpm)	Maximum rotational speed (mechanical) [rpm]
General | Speed LPF omega	Speed LPF parameter omega [Hz].
General | Limit of q-axis current (A)	Limit of q-axis current [A].
General | Step of speed feedback at open-loop	Rate of reference speed for feedback speed limiter at Open loop.
General | Natural frequency	Natural frequency for disturbance speed observer.
General | Open-loop damping	Select enable/disable of damping control at Open loop.
General | Flux weakening	Select enable/disable of flux weakening control at high speed.
General | Torque compensation for sensorless transition	Select enable/disable of soft switching at the transition from Open loop to PI control.
General | Speed observer	Select enable/disable of speed observer process
General | Selection of speed observer	Select the method of speed observer
General | Control method	Select the position control method.
Open-Loop | Step of d-axis current climbing	The d-axis current reference ramping up rate [A/msec].
Open-Loop | Step of d-axis current descending	The d-axis current reference ramping down rate [A/msec].
Open-Loop | Step of q-axis current descending ratio	The q-axis current reference ramping down proportion to reference before open loop [A/msec].
Open-Loop | Reference of d-axis current	The d-axis current reference in open loop drive [A].
Open-Loop | Threshold of speed control descending	The speed threshold to ramp down the d-axis current (mechanical) [rpm].
Open-Loop | Threshold of speed control climbing	The speed threshold to ramp up the d-axis current (mechanical) [rpm].
Open-Loop | Period between open-loop to BEMF (sec)	Time to switch open loop to sensorless [sec].
Open-Loop | Phase error(degree) to decide sensor-less switch timing	Phase error to decide sensorless switch timing (electrical angle) [degree].
Design parameter | Speed PI loop omega	Speed PI Control parameter omega.
Design parameter | Speed PI loop zeta	Speed PI Control parameter zeta.
Design parameter | Estimated d-axis HPF omega	Natural frequency [Hz] for HPF in open loop damping gain design.
Design parameter | Open-loop damping zeta	Damping ratio for open loop damping gain design.
Design parameter | Cutoff frequency of phase error LPF	The cut-off frequency [Hz] of phase error LPF gain design.
Design parameter | Speed observer omega	Speed observer omega.
Design parameter | Speed observer zeta	Speed observer zeta.
Motor Parameter | Pole pairs	Pole pairs of target motor.
Motor Parameter | Resistance (ohm)	Resistance of motor [ohm].
Motor Parameter | Inductance of d-axis (H)	D-axis inductance [H].
Motor Parameter | Inductance of q-axis (H)	Q-axis inductance [H].
Motor Parameter | Permanent magnetic flux (Wb)	Magnetic flux [Wb].
Motor Parameter | Rotor inertia (kgm^2)	Rotor inertia [kgm^2].

Table 10-5 Configuration Options initial value [1/2] (rm_motor_speed)

Options	RA6T2	RA6T3	RA4T1
Common | Position support	-	-	-
General | Speed control period (sec)	0.001	0.001	0.002
General | Step of speed climbing (rpm)	0.5	0.5	1.0
General | Maximum rotational speed (rpm)	2400.0	2400.0	2400.0
General | Speed LPF omega	10.0	10.0	10.0
General | Limit of q-axis current (A)	1.67	1.67	1.67
General | Step of speed feedback at open-loop	0.2	0.2	0.2
General | Natural frequency	100.0	100.0	100.0
General | Open-loop damping	Enable	Enable	Enable
General | Flux weakening	Disable	Disable	Disable
General | Torque compensation for sensorless transition	Enable	Enable	Enable
General | Speed observer	-	-	-
General | Selection of speed observer	-	-	-
General | Control method	-	-	-
Open-Loop | Step of d-axis current climbing	0.6	0.6	1.2
Open-Loop | Step of d-axis current descending	0.6	0.6	1.2
Open-Loop | Step of q-axis current descending ratio	1.0	1.0	1.0
Open-Loop | Reference of d-axis current	0.3	0.3	0.3
Open-Loop | Threshold of speed control descending	700.0	700.0	700.0
Open-Loop | Threshold of speed control climbing	500.0	500.0	500.0
Open-Loop | Period between open-loop to BEMF (sec)	0.025	0.025	0.025
Open-Loop | Phase error(degree) to decide sensor-less switch timing	10.0	10.0	10.0
Design parameter | Speed PI loop omega	3.0	3.0	3.0
Design parameter | Speed PI loop zeta	1.0	1.0	1.0
Design parameter | Estimated d-axis HPF omega	2.5	2.5	2.5
Design parameter | Open-loop damping zeta	1.0	1.0	1.0
Design parameter | Cutoff frequency of phase error LPF	10.0	10.0	10.0
Design parameter | Speed observer omega	-	-	-
Design parameter | Speed observer zeta	-	-	-
Motor Parameter | Pole pairs	4	4	4
Motor Parameter | Resistance (ohm)	1.3	1.3	1.3
Motor Parameter | Inductance of d-axis (H)	0.0013	0.0013	0.0013
Motor Parameter | Inductance of q-axis (H)	0.0013	0.0013	0.0013
Motor Parameter | Permanent magnetic flux (Wb)	0.01119	0.01119	0.01119
Motor Parameter | Rotor inertia (kgm^2)	0.000003666	0.000003666	0.000003666

Table 10-6 Configuration Options initial value [2/2] (rm_motor_speed)

Options	RA8T1	RA8T2
Common | Position support	-	-
General | Speed control period (sec)	0.001	0.001
General | Step of speed climbing (rpm)	0.5	0.5
General | Maximum rotational speed (rpm)	2400.0	2400.0
General | Speed LPF omega	10.0	10.0
General | Limit of q-axis current (A)	1.67	1.67
General | Step of speed feedback at open-loop	0.2	0.2
General | Natural frequency	100.0	100.0
General | Open-loop damping	Enable	Enable
General | Flux weakening	Disable	Disable
General | Torque compensation for sensorless transition	Enable	Enable
General | Speed observer	-	-
General | Selection of speed observer	-	-
General | Control method	-	-
Open-Loop | Step of d-axis current climbing	0.6	0.6
Open-Loop | Step of d-axis current descending	0.6	0.6
Open-Loop | Step of q-axis current descending ratio	1.0	1.0
Open-Loop | Reference of d-axis current	0.3	0.3
Open-Loop | Threshold of speed control descending	700.0	700.0
Open-Loop | Threshold of speed control climbing	500.0	500.0
Open-Loop | Period between open-loop to BEMF (sec)	0.025	0.025
Open-Loop | Phase error(degree) to decide sensor-less switch timing	10.0	10.0
Design parameter | Speed PI loop omega	3.0	3.0
Design parameter | Speed PI loop zeta	1.0	1.0
Design parameter | Estimated d-axis HPF omega	2.5	2.5
Design parameter | Open-loop damping zeta	1.0	1.0
Design parameter | Cutoff frequency of phase error LPF	10.0	10.0
Design parameter | Speed observer omega	-	-
Design parameter | Speed observer zeta	-	-
Motor Parameter | Pole pairs	4	4
Motor Parameter | Resistance (ohm)	1.3	1.3
Motor Parameter | Inductance of d-axis (H)	0.0013	0.0013
Motor Parameter | Inductance of q-axis (H)	0.0013	0.0013
Motor Parameter | Permanent magnetic flux (Wb)	0.01119	0.01119
Motor Parameter | Rotor inertia (kgm^2)	0.000003666	0.000003666

10.4 List of Current control module parameters

The changeable parameters that are used in current control module and these initial values are listed below.

Table 10-7 Configuration options of current control (rm_motor_current)

Option name	Description
General | Shunt type	Selects how many shunt resistances to use current detection.
General | Current control decimation	Counts of decimation about current control period interrupt
General | PWM carrier frequency (kHz)	PWM carrier frequency [kHz]
General | Input voltage (V)	Input voltage [V]
General | Sample delay compensation	Selects whether to “enable” or “disable” sample delay compensation
General | Period magnification value	Period magnification value for sampling delay compensation.
General | Voltage error compensation	Selects whether to “enable” or “disable” voltage error compensation.
General | Voltage error compensation table of voltage 1	Table of voltage error compensation about voltage #1
General | Voltage error compensation table of voltage 2	Table of voltage error compensation about voltage #2
General | Voltage error compensation table of voltage 3	Table of voltage error compensation about voltage #3
General | Voltage error compensation table of voltage 4	Table of voltage error compensation about voltage #4
General | Voltage error compensation table of voltage 5	Table of voltage error compensation about voltage #5
General | Voltage error compensation table of current 1	Table of voltage error compensation about current #1
General | Voltage error compensation table of current 2	Table of voltage error compensation about current #2
General | Voltage error compensation table of current 3	Table of voltage error compensation about current #3
General | Voltage error compensation table of current 4	Table of voltage error compensation about current #4
General | Voltage error compensation table of current 5	Table of voltage error compensation about current #5
Design Parameter | Current PI loop omega	Current PI control omega parameter [Hz].
Design Parameter | Current PI loop zeta	Current PI control zeta parameter.
Motor Parameter | Pole pairs	Pole pairs of target motor.
Motor Parameter | Resistance (ohm)	Resistance of motor [ohm].
Motor Parameter | Inductance of d-axis (H)	D-axis inductance [H].
Motor Parameter | Inductance of q-axis (H)	Q-axis inductance [H].
Motor Parameter | Permanent magnetic flux (Wb)	Magnetic flux [Wb].
Motor Parameter | Rotor inertia (kgm^2)	Rotor inertia [kgm^2].

Table 10-8 Configuration Options initial value [1/2] (rm_motor_current)

Option name	RA6T2	RA6T3	RA4T1
General | Shunt type	1shunt	1shunt	1shunt
General | Current control decimation	0	0	1
General | PWM carrier frequency (kHz)	10.0	10.0	10.0
General | Input voltage (V)	24.0	24.0	24.0
General | Sample delay compensation	Enable	Enable	Enable
General | Period magnification value	1.5	1.5	2.0
General | Voltage error compensation	Enable	Enable	Enable
General | Voltage error compensation table of voltage 1	0.477	0.477	0.477
General | Voltage error compensation table of voltage 2	0.742	0.742	0.742
General | Voltage error compensation table of voltage 3	0.892	0.892	0.892
General | Voltage error compensation table of voltage 4	0.979	0.979	0.979
General | Voltage error compensation table of voltage 5	1.009	1.009	1.009
General | Voltage error compensation table of current 1	0.021	0.021	0.021
General | Voltage error compensation table of current 2	0.034	0.034	0.034
General | Voltage error compensation table of current 3	0.064	0.064	0.064
General | Voltage error compensation table of current 4	0.158	0.158	0.158
General | Voltage error compensation table of current 5	0.400	0.400	0.400
Design Parameter | Current PI loop omega	300.0	300.0	300.0
Design Parameter | Current PI loop zeta	1.0	1.0	1.0
Motor Parameter | Pole pairs	4	4	4
Motor Parameter | Resistance (ohm)	1.3	1.3	1.3
Motor Parameter | Inductance of d-axis (H)	0.0013	0.0013	0.0013
Motor Parameter | Inductance of q-axis (H)	0.0013	0.0013	0.0013
Motor Parameter | Permanent magnetic flux (Wb)	0.01119	0.01119	0.01119
Motor Parameter | Rotor inertia (kgm^2)	0.000003666	0.000003666	0.000003666

Table 10-9 Configuration Options initial value [2/2] (rm_motor_current)

Option name	RA8T1	RA8T2
General | Shunt type	1shunt	1shunt
General | Current control decimation	0	0
General | PWM carrier frequency (kHz)	10.0	10.0
General | Input voltage (V)	24.0	24.0
General | Sample delay compensation	Enable	Enable
General | Period magnification value	1.5	1.5
General | Voltage error compensation	Enable	Enable
General | Voltage error compensation table of voltage 1	0.477	0.477
General | Voltage error compensation table of voltage 2	0.742	0.742
General | Voltage error compensation table of voltage 3	0.892	0.892
General | Voltage error compensation table of voltage 4	0.979	0.979
General | Voltage error compensation table of voltage 5	1.009	1.009
General | Voltage error compensation table of current 1	0.021	0.021
General | Voltage error compensation table of current 2	0.034	0.034
General | Voltage error compensation table of current 3	0.064	0.064
General | Voltage error compensation table of current 4	0.158	0.158
General | Voltage error compensation table of current 5	0.400	0.400
Design Parameter | Current PI loop omega	300.0	300.0
Design Parameter | Current PI loop zeta	1.0	1.0
Motor Parameter | Pole pairs	4	4
Motor Parameter | Resistance (ohm)	1.3	1.3
Motor Parameter | Inductance of d-axis (H)	0.0013	0.0013
Motor Parameter | Inductance of q-axis (H)	0.0013	0.0013
Motor Parameter | Permanent magnetic flux (Wb)	0.01119	0.01119
Motor Parameter | Rotor inertia (kgm^2)	0.000003666	0.000003666

10.5 List of Driver module parameters

The changeable parameters that are used in driver control module and these initial values are listed below.

Table 10-10 Configuration options of driver module [1/2] (rm_motor_driver)

Option name	Description
Common | ADC_B Support	ADC_B module support
Common | Shared ADC support	Selection of using shared ADC module
Common | Supported Motor Number	Number of driven motors
General | Shunt type	Current detection method selection
General | Modulation method	Selection of the method of modulation
General | PWM output port UP	Port setting of U phase upper arm
General | PWM output port UN	Port setting of U phase lower arm
General | PWM output port VP	Port setting of V phase upper arm
General | PWM output port VN	Port setting of V phase lower arm
General | PWM output port WP	Port setting of W phase upper arm
General | PWM output port WN	Port setting of W phase lower arm
General | PWM Timer Frequency (MHz)	PWM Timer Clock Frequency [MHz]
General | PWM Carrier Period (Microseconds)	PWM Carrier Period [Micro seconds]
General | Dead Time (Raw Counts)	PWM Dead time [raw counts]
General | Current Range (A)	Measurement Range of Electric current [A]
General | Voltage Range (V)	Measurement Range of Inverter Voltage [V]
General | Counts for current offset measurement	Counts of measurement the offset of A/D Conversion at electric current input.
General | A/D conversion channel for U Phase current	A/D channel for U-phase current
General | A/D conversion channel for W Phase current	A/D channel for W-phase current
General | A/D conversion channel for Main Line Voltage	A/D channel for inverter bus voltage
General | A/D conversion channel for V Phase current	A/D channel for V-phase current
General | A/D conversion channel for sin signal	A/D channel for sin signal
General | A/D conversion channel for cos signal	A/D channel for cos signal
General | Using ADC scan group	Set ADC scan group according to ADC module setting.
General | A/D conversion unit for U Phase current	Select the A/D conversion module for U phase current
General | A/D conversion unit for W Phase current	Select the A/D conversion module for W phase current
General | A/D conversion unit for main line voltage	Select the A/D conversion module for inverter bus voltage
General | A/D conversion unit for V Phase current	Select the A/D conversion module for V phase current

Table 10-11 Configuration options of driver module [2/2] (rm_motor_driver)

Option name	Description
General | A/D conversion unit for sin signal	Select the A/D conversion module for sin signal
General | A/D conversion unit for cos signal	Select the A/D conversion module for cos signal
General | ADC interrupt module	Select from which module ADC interrupt happens
General | Adjustment value to current A/D	Current A/D timing adjustment (for 1shunt)
General | Minimum difference of PWM duty	Minimum difference of PWM duty setting (for 1shunt)
General | Adjustment delay of A/D conversion	A/D conversion delay timing adjustment (for 1shunt)
General | 1shunt interrupt phase	Which phase is used to detect 1shunt current (for 1shunt)
General | Input Voltage (V)	Range of input for inverter bus voltage
General | Resolution of A/D conversion	Resolution of A/D conversion Please set same value with ADC module setting.
General | Offset of A/D conversion for current	Offset level of A/D conversion input for current Please set according to the circuit.
General | Conversion level of A/D conversion for voltage	Conversion level of A/D conversion for voltage Please set when the MCU main voltage is different.
General | GTIOCA stop level	Output level of upper arm at stop status
General | GTIOCB stop level	Output level of lower arm at stop status
Modulation | Maximum duty	Maximum duty of PWM Maximum duty except dead time.

Table 10-12 Configuration Options initial value [1/4] (rm_motor_driver)

Option name	RA6T2	RA6T3	RA4T1
Common | ADC_B Support	Enabled	-	-
Common | Shared ADC support	Disabled	Disabled	Disabled
Common | Supported Motor Number	1	1	1
General | Shunt type	1shunt	1shunt	1shunt
General | Modulation method	SVPWM	SVPWM	SVPWM
General | PWM output port UP	BSP_IO_PORT_11_PIN_04	BSP_IO_PORT_04_PIN_09	BSP_IO_PORT_04_PIN_09
General | PWM output port UN	BSP_IO_PORT_11_PIN_05	BSP_IO_PORT_04_PIN_08	BSP_IO_PORT_04_PIN_08
General | PWM output port VP	BSP_IO_PORT_11_PIN_06	BSP_IO_PORT_01_PIN_03	BSP_IO_PORT_01_PIN_03
General | PWM output port VN	BSP_IO_PORT_11_PIN_07	BSP_IO_PORT_01_PIN_02	BSP_IO_PORT_01_PIN_02
General | PWM output port WP	BSP_IO_PORT_11_PIN_08	BSP_IO_PORT_01_PIN_11	BSP_IO_PORT_01_PIN_11
General | PWM output port WN	BSP_IO_PORT_11_PIN_09	BSP_IO_PORT_01_PIN_12	BSP_IO_PORT_01_PIN_12
General | PWM Timer Frequency (MHz)	120	100	100
General | PWM Carrier Period (Microseconds)	100	100	100
General | Dead Time (Raw Counts)	240	200	200
General | Current Range (A)	16.5	16.5	16.5
General | Voltage Range (V)	73.51	73.51	73.51
General | Counts for current offset measurement	500	500	500
General | A/D conversion channel for U Phase current	4	0	0
General | A/D conversion channel for W Phase current	-	-	-
General | A/D conversion channel for Main Line Voltage	Ver.1: 6 Ver.2: 7	4	4
General | A/D conversion channel for V Phase current	-	-	-
General | A/D conversion channel for sin signal	-	-	-
General | A/D conversion channel for cos signal	-	-	-
General | Using ADC scan group	0	-	-
General | A/D conversion unit for U Phase current	-	0	0
General | A/D conversion unit for W Phase current	-	-	-
General | A/D conversion unit for main line voltage	-	0	0

Table 10-13 Configuration Options initial value [2/4] (rm_motor_driver)

Option name	RA8T1	RA8T2
Common | ADC_B Support	-	Enabled
Common | Shared ADC support	Enabled	Disabled
Common | Supported Motor Number	1	1
General | Shunt type	1shunt	1shunt
General | Modulation method	SVPWM	SVPWM
General | PWM output port UP	BSP_IO_PORT_01_PIN_15	BSP_IO_PORT_06_PIN_05
General | PWM output port UN	BSP_IO_PORT_06_PIN_09	BSP_IO_PORT_06_PIN_04
General | PWM output port VP	BSP_IO_PORT_01_PIN_13	BSP_IO_PORT_06_PIN_03
General | PWM output port VN	BSP_IO_PORT_01_PIN_14	BSP_IO_PORT_06_PIN_02
General | PWM output port WP	BSP_IO_PORT_03_PIN_00	BSP_IO_PORT_06_PIN_12
General | PWM output port WN	BSP_IO_PORT_01_PIN_12	BSP_IO_PORT_06_PIN_13
General | PWM Timer Frequency (MHz)	120	250
General | PWM Carrier Period (Microseconds)	100	100
General | Dead Time (Raw Counts)	240	500
General | Current Range (A)	16.5	16.5
General | Voltage Range (V)	73.51	73.51
General | Counts for current offset measurement	500	500
General | A/D conversion channel for U Phase current	-	6
General | A/D conversion channel for W Phase current	-	-
General | A/D conversion channel for Main Line Voltage	8	7
General | A/D conversion channel for V Phase current	1	-
General | A/D conversion channel for sin signal	-	-
General | A/D conversion channel for cos signal	-	-
General | Using ADC scan group	-	0
General | A/D conversion unit for U Phase current	-	-
General | A/D conversion unit for W Phase current	-	-
General | A/D conversion unit for main line voltage	0	-

Table 10-14 Configuration Options initial value [3/4] (rm_motor_driver)

Option name	RA6T2	RA6T3	RA4T1
General | A/D conversion unit for V Phase current	-	-	-
General | A/D conversion unit for sin signal	-	-	-
General | A/D conversion unit for cos signal	-	-	-
General | ADC interrupt module	-	1st	1st
General | Adjustment value to current A/D	0.0	20.0	20.0
General | Minimum difference of PWM duty	900	500	500
General | Adjustment delay of A/D conversion	600	250	250
General | 1shunt interrupt phase	PHASE_U	PHASE_U	PHASE_U
General | Input Voltage (V)	24.0	24.0	24.0
General | Resolution of A/D conversion	0xFFF	0xFFF	0xFFF
General | Offset of A/D conversion for current	0x7FF	0x7FF	0x7FF
General | Conversion level of A/D conversion for voltage	1.0	1.0	1.0
General | GTIOCA stop level	Pin Level Low	Pin Level Low	Pin Level Low
General | GTIOCB stop level	Pin Level High	Pin Level High	Pin Level High
Modulation | Maximum duty	0.9375	0.9375	0.9375

Table 10-15 Configuration Options initial value [4/4] (rm_motor_driver)

Option name	RA8T1	RA8T2
General | A/D conversion unit for V Phase current	0	-
General | A/D conversion unit for sin signal	-	-
General | A/D conversion unit for cos signal	-	-
General | ADC interrupt module	1st	-
General | Adjustment value to current A/D	10.0	0
General | Minimum difference of PWM duty	500	1800
General | Adjustment delay of A/D conversion	250	1200
General | 1shunt interrupt phase	PHASE_V	PHASE_U
General | Input Voltage (V)	24.0	24.0
General | Resolution of A/D conversion	0xFFF	0xFFF
General | Offset of A/D conversion for current	0x7FF	0x7FF
General | Conversion level of A/D conversion for voltage	1.0	1.0
General | GTIOCA stop level	Pin Level Low	Pin Level Low
General | GTIOCB stop level	Pin Level High	Pin Level High
Modulation | Maximum duty	0.9375	0.9375

10.6 List of Estimate speed and angle module parameters

The changeable parameters that are used in estimate speed and angle module and these initial values are listed below.

Table 10-16 Configuration options of estimate module (rm_motor_estimate)

Option name	Description
Motor Parameter | Pole pairs	Pole pairs of target motor.
Motor Parameter | Resistance (ohm)	Resistance of motor [ohm].
Motor Parameter | Inductance of d-axis (H)	D-axis inductance [H].
Motor Parameter | Inductance of q-axis (H)	Q-axis inductance [H].
Motor Parameter | Permanent magnetic flux (Wb)	Magnetic flux [Wb].
Motor Parameter | Rotor inertia (kgm^2)	Rotor inertia [kgm^2].
Motor Parameter | Nominal current (Arms)	Nominal current [Arms]
Openloop damping	Select enable/disable of Open loop Damping Control
Natural frequency of BEMF observer	Natural frequency for BEMF observer [Hz].
Damping ratio of BEMF observer	Damping ratio for BEMF observer.
Natural frequency of PLL Speed estimate loop	Natural frequency for rotor position Phase-Locked Loop [Hz].
Damping ratio of PLL Speed estimate loop	Damping ratio for rotor position Phase-Locked Loop.
Control period	Period of Speed Control [sec]

Table 10-17 Configuration Options initial value [1/2] (rm_motor_estimate)

Option name	RA6T2	RA6T3	RA4T1
Motor Parameter | Pole pairs	4	4	4
Motor Parameter | Resistance (ohm)	1.3	1.3	1.3
Motor Parameter | Inductance of d-axis (H)	0.0013	0.0013	0.0013
Motor Parameter | Inductance of q-axis (H)	0.0013	0.0013	0.0013
Motor Parameter | Permanent magnetic flux (Wb)	0.01119	0.01119	0.01119
Motor Parameter | Rotor inertia (kgm^2)	0.000003666	0.000003666	0.000003666
Motor Parameter | Nominal current (Arms)	1.67	1.67	1.67
Openloop damping	Enable	Enable	Enable
Natural frequency of BEMF observer	1000.0	1000.0	500.0
Damping ratio of BEMF observer	1.0	1.0	1.0
Natural frequency of PLL Speed estimate loop	20.0	20.0	20.0
Damping ratio of PLL Speed estimate loop	1.0	1.0	1.0
Control period	0.0001	0.0001	0.0002

Table 10-18 Configuration Options initial value [2/2] (rm_motor_estimate)

Option name	RA8T1	RA8T2
Motor Parameter | Pole pairs	4	4
Motor Parameter | Resistance (ohm)	1.3	1.3
Motor Parameter | Inductance of d-axis (H)	0.0013	0.0013
Motor Parameter | Inductance of q-axis (H)	0.0013	0.0013
Motor Parameter | Permanent magnetic flux (Wb)	0.01119	0.01119
Motor Parameter | Rotor inertia (kgm^2)	0.000003666	0.000003666
Motor Parameter | Nominal current (Arms)	1.67	1.67
Openloop damping	Enable	Enable
Natural frequency of BEMF observer	1000.0	1000.0
Damping ratio of BEMF observer	1.0	1.0
Natural frequency of PLL Speed estimate loop	20.0	20.0
Damping ratio of PLL Speed estimate loop	1.0	1.0
Control period	0.0001	0.0001

10.7 Parameters about protection function

rm_motor_sensorless / Limit of over current
Set the threshold to judge overcurrent. Actual value is calculated by "input value * sqrt(2) * 1.5".

rm_motor_sensorless / Limit of over voltage
Set the threshold to judge over voltage error. Please select reasonable value with your environment.

rm_motor_sensorless / Limit of low voltage
Set the threshold to judge low voltage error. Please select reasonable value with your environment.

10.8 Changing the PWM carrier frequency

If you want to change PWM carrier frequency, please change below items.

rm_motor_driver / PWM Carrier Period
Set the period of PWM carrier.

rm_motor_current / PWM carrier frequency
Set the frequency of PWM carrier.

rm_motor_estimate / Control period
Set the period of PWM carrier.

10.9 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). You can select the method at "Modulation method" option in driver module.

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.

rm_motor_driver / Maximum duty
Maximum duty of PWM. Please use default setting "0.9375" normally.

10.10 Parameters for inverter

10.10.1 Deadtime

rm_motor_driver / Dead Time (Raw Counts)

Set dead time which is written in specification or design document about the inverter as counts of the timer (GPT). For example, if the timer clock is “120[MHz]” and deadtime is “2[μs]”, please set “240”.

10.10.2 Current detection gain

rm_motor_driver / Current Range (A)

Set the range to detect electrical current. In the specification of MCILV-1, when 0.0 to 3.3[V] input, ±8.25[A] (peak to peak 16.5[A]) is detected. Please set value of “peak to peak”.

Table 10-19 Current signal specifications for MCI-LV-1

3-phase output current value	ADC Input voltage value	ADC conversion value
+8.25A	3.3V	4095
0A	1.65V	2047
-8.25A	0V	0

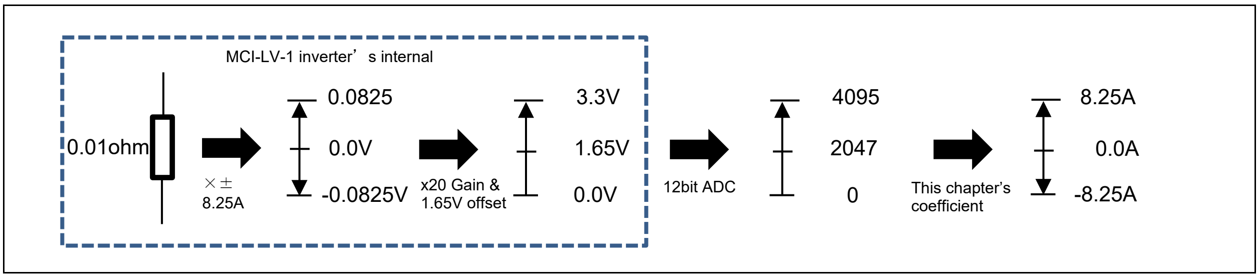


Figure 10-1 Current detection calculation flow

10.10.3 Voltage detection gain

rm_motor_driver / Voltage Range (V)

Set the value when maximum digits (12bit conversion : 4095) is detected, how match inverter bus voltage is inputted. For MCI-LV-1, when ADC input is 3.3[V], detected voltage is correspond to 73.51[V] (this ratio depends on circuit). Therefore, set “73.51” in this configuration.

Table 10-20 Inverter bus voltage signal specifications for MCI-LV-1

Inverter bus voltage value	ADC Input voltage value	ADC conversion value
0V	0V	0
73.51V	3.3V	4095

10.10.4 Voltage error compensation parameters

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

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

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

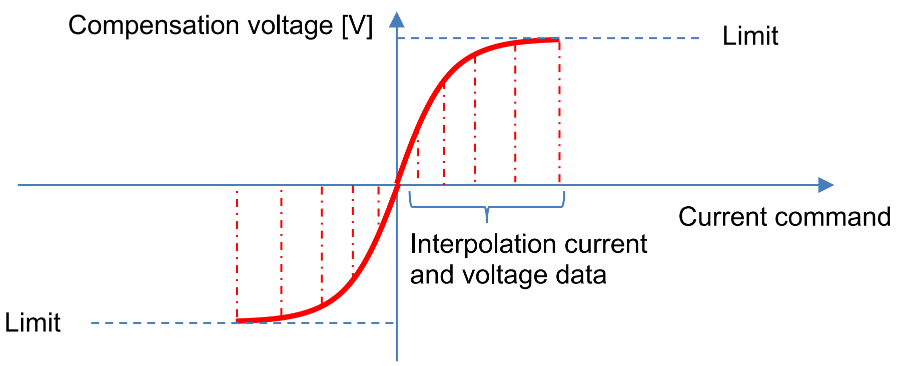


Figure 10-2 Relationship between the compensation voltage value/limit and current command value

Table 10-21 Example of setting for compensation about current and voltage

No.	rm_motor_current/Voltage error compensation table of current	rm_motor_current/Voltage error compensation table of voltage
0	0.477	0.021
1	0.742	0.034
2	0.892	0.064
3	0.979	0.158
4	1.009	0.400

10.11 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.

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

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. The unit is ohm.

Inductance of d-axis, Inductance of q-axis
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.

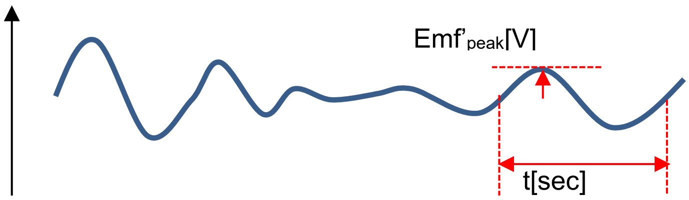
Set the obtained Ld and Lq. The unit is H (henry).

Roter 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.

Nominal current
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.

Permanent 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 $\sqrt{3}$ gives the peak value of inductive voltage per phase. You can obtain the magnetic flux linkage Ψ from the equation of the inductive voltage = $\omega\Psi$. Convert the rated speed to the electrical angular speed frequency f[Hz], substitute ω with 2πf to make the inductive voltage = $2\pi f\Psi$, 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}.

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

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.

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

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.

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

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.

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

Specify the obtained magnetic flux linkage Ψ to Magnetic Flux of each module.

10.12 Parameters for current control

rm_motor_current / Current PI loop omega, rm_motor_current / Current PI loop 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.

10.13 Parameters for speed control

rm_motor_speed/Speed PI loop omega, rm_motor_speed/Speed PI loop zeta
In the speed control module, the control gain is adjusted by tuning the natural frequency for the speed control system(Speed PI loop omega) and the attenuation coefficient for the speed control system(Speed PI loop zeta). 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 Natural frequency of PLL Speed estimate loop. 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 stable 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.

rm_motor_speed/Step of speed climbing
Set the rate at which the speed increases (acceleration) when the speed command value is set. If set value is big, speed increases faster. When “1” is set, the speed is increased by 1rpm per speed control period.

10.14 Parameters for sample delay compensation

rm_motor_current/Period magnification value

Set the value which compensates the period from current detection timing to PWM duty set timing.

11. Control flowcharts

11.1 Main process

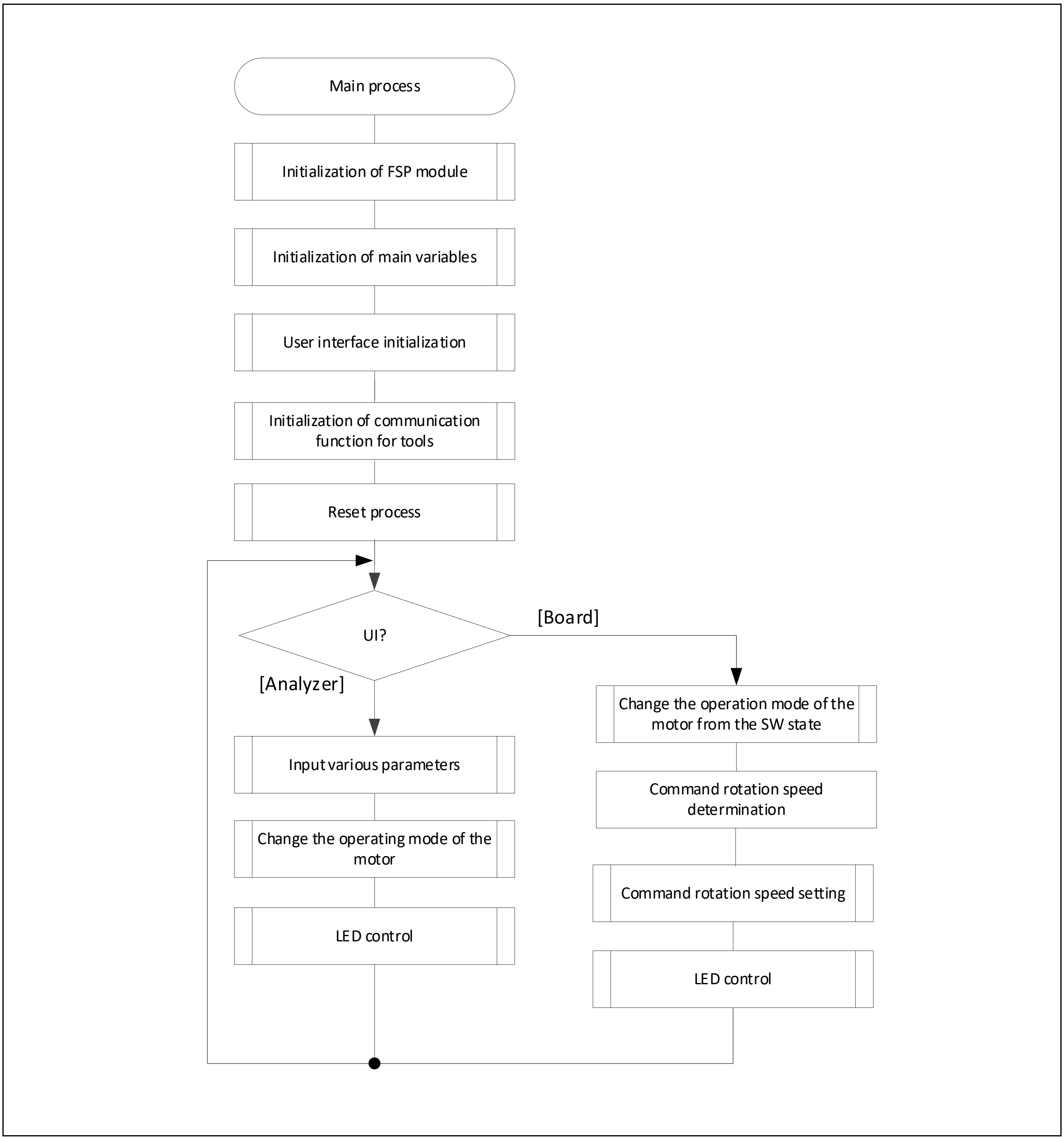


Figure 11-1 Main Process Flowchart

11.2 Current Control Period Interrupt Process

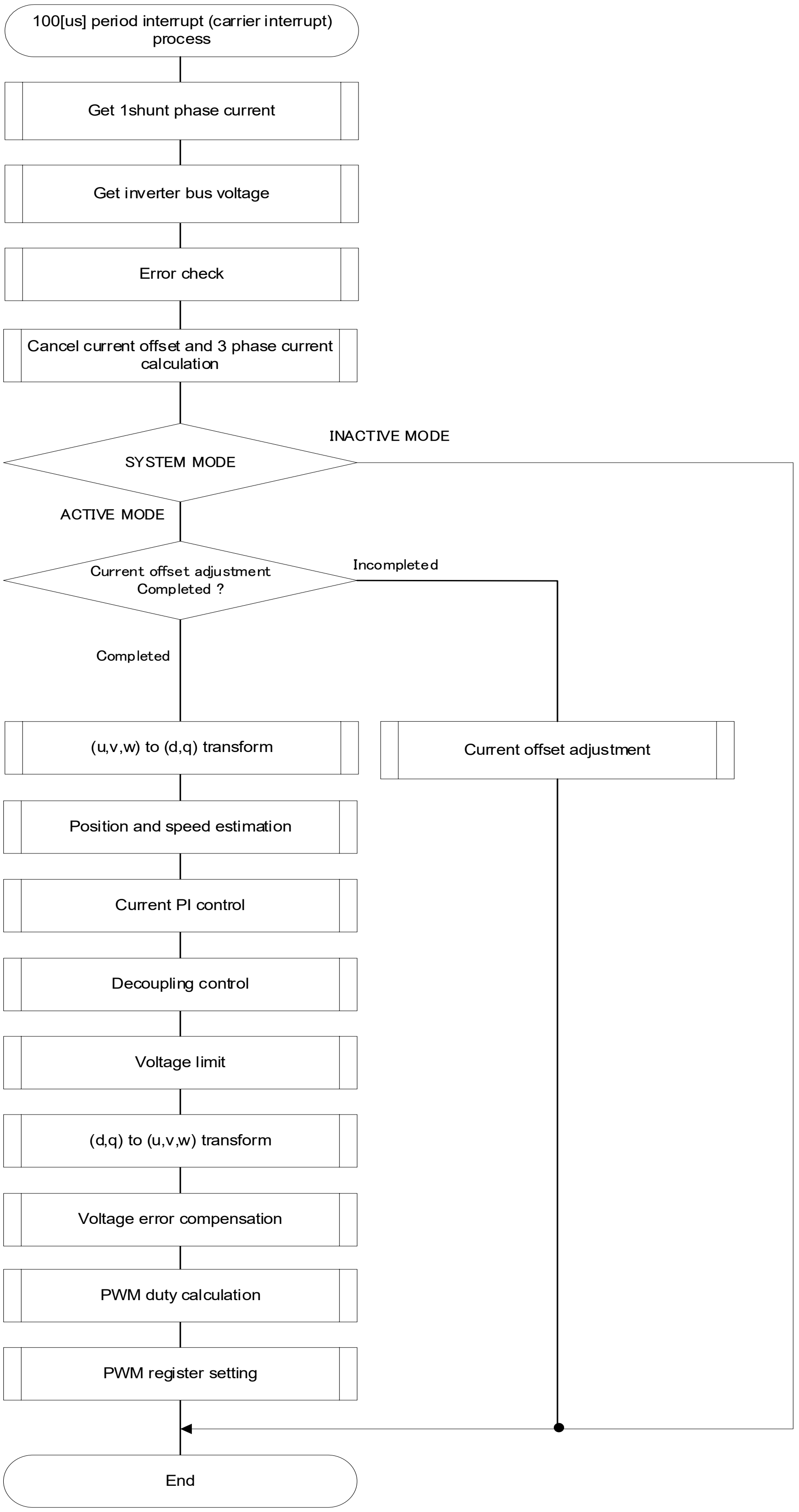


Figure 11-2 Current control Period Interrupt Process Flowchart

11.3 Speed control Period Interrupt Process

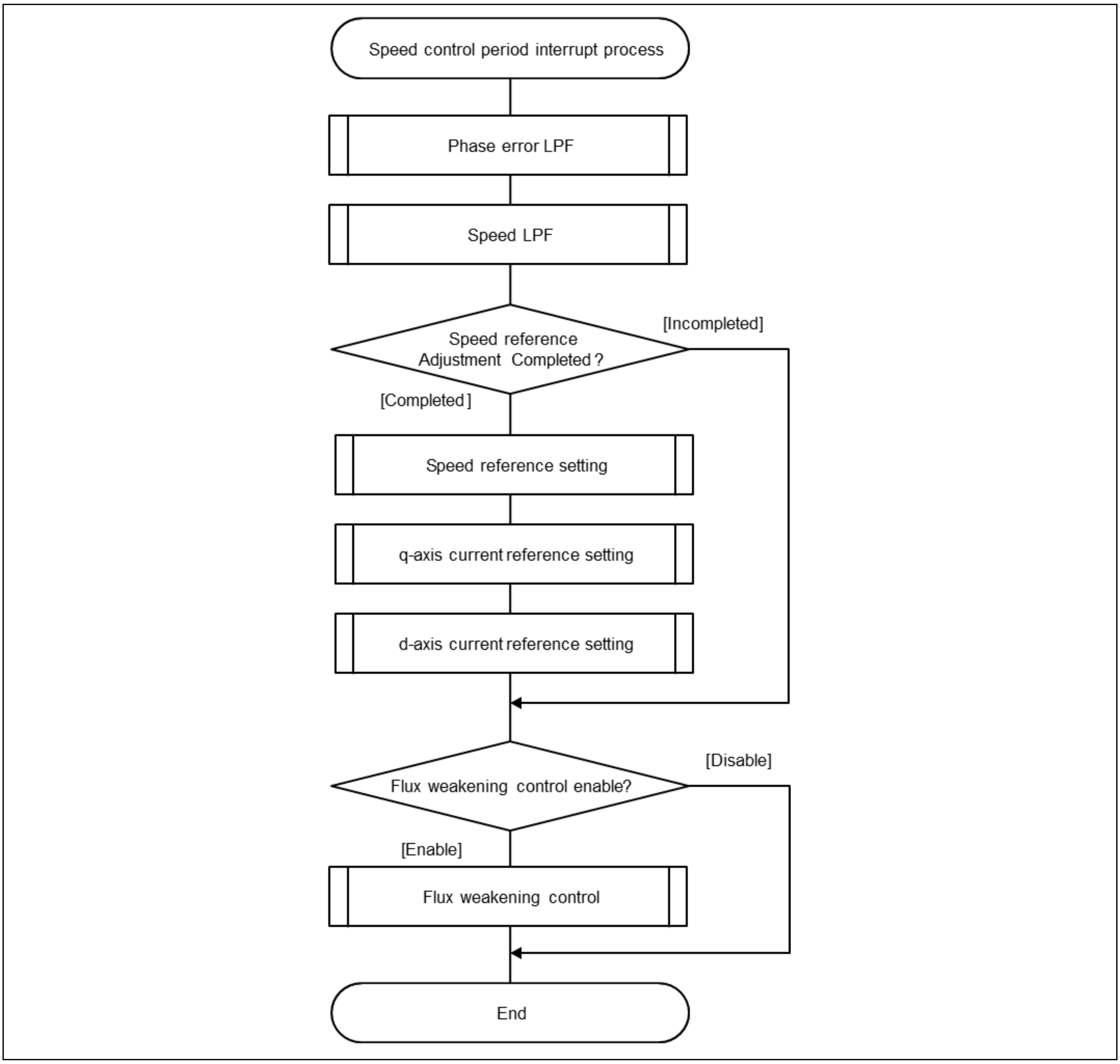


Figure 11-3 Speed Control Period Interrupt Process Flowchart

11.4 Over Current Detection Interrupt Process

The overcurrent detection interrupt is an interrupt that occurs when an external overcurrent detection signal is input. The PWM output terminal are put in the high impedance state. Therefore, at the start of execution of this interrupt processing, the PWM output terminal is already in the high impedance state and the output to the motor had been stopped.

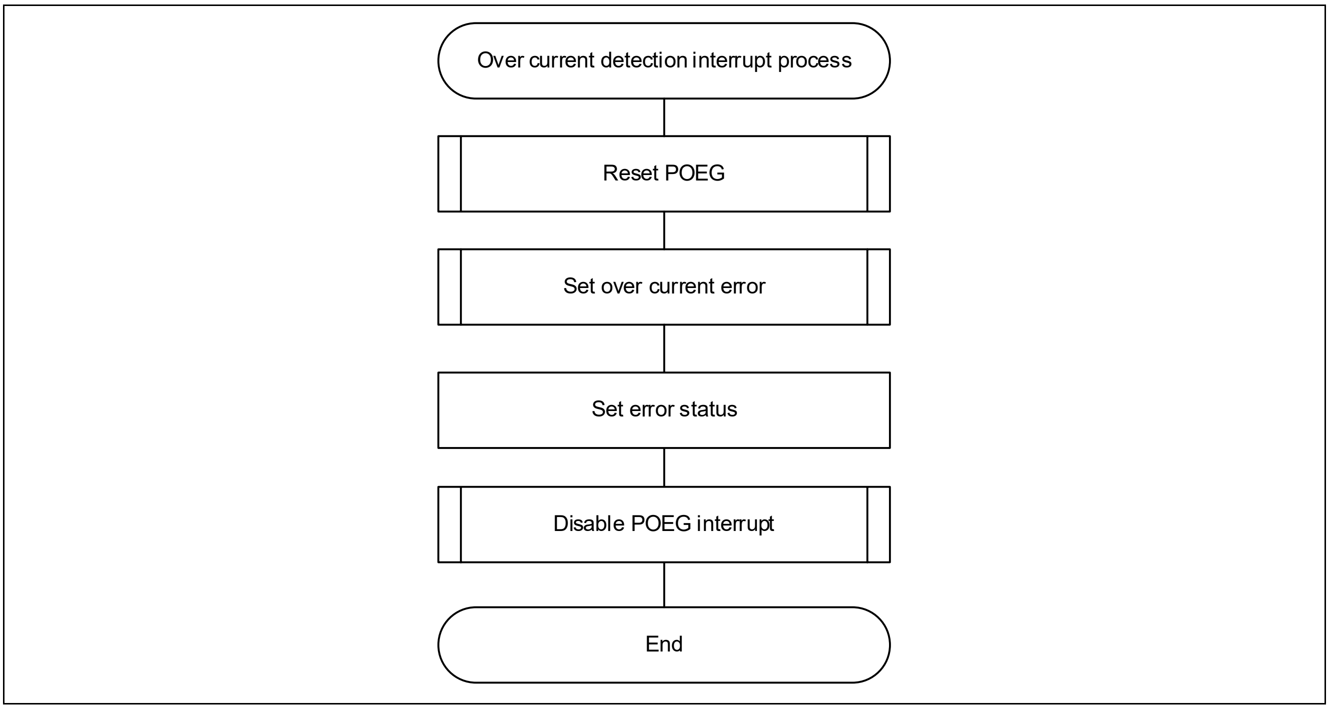


Figure 11-4 Over Current Detection Interrupt Process Flowchart

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 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 6.7 c). It explains how to recover from errors.
The motor stops with an error even after starting.	See 6.7 c) for the cause of the error. After that, please confirm settings about the cause of errors.
The 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 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 "src/application/rmw/*.map" with RMW to update the variable status of the software. If you omit this step, the variable may not display correctly. For details, see 6.5.

Revision History

Rev.	Date of issue	Amendments	
		Page	Point
1.00	May 23, 2023	-	First edition issued
1.10	Jan 23, 2024	-	Added description related to RA8T1
1.11	Dec 23, 2024	-	Update target software
1.20	Sep 2, 2025		- Added RA8T2 - Updated chapter titles
1.21	Oct 31, 2025	-	- Added RA6T2 ver.2 - Fixed incorrect entries

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

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

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Corporate Headquarters

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

Trademarks

Contact information

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