120-degree conducting control of permanent magnetic synchronous motor using hall sensors

For Renesas Flexible Motor Control Series

Introduction

This application note describes the sample program to drive a permanent magnetic synchronous motor with 120-degree conducting method using hall sensors 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 4

2. Glossary 6

3. Used hardware and software 7
3.1 List of used hardware 7
3.2 List of used software 7

4. Building a hardware environment 8
4.1 Overview : Hardware environments 8
4.2 Preparing the power supply 8
4.3 Preparing the inverter 9
4.4 Setting up the RA series CPU board 9
4.5 Connection example of kit 12
4.6 Connection of Hall sensor 13
4.7 On board debugger 13
4.8 Wiring 14

5. Environment to build the sample software 15

6. Driving the motor 16
6.1 Precautions before driving the motor 16
6.2 Connecting 16
6.3 Writing the sample program 20
6.3.1 Install e^2 studio 20
6.3.2 Import a project 20
6.3.3 Build the target project 24
6.3.4 Connection between PC and the target CPU board via an USB cable 26
6.3.5 Writing to the target board (with built program) 27

6.4 Introducing RMW ... 29
6.5 Registering and updating the Map file ... 30
6.6 Variables used for working with RMW ... 31
6.7 Controlling the motor ... 33
6.8 Stopping and shutting down the motor ... 35

7. Motor control algorithm ... 36
7.1 Overview ... 36
7.2 Detection of rotor angle with hall sensors ... 37
7.3 Speed control function ... 38
7.4 Voltage control by PWM ... 39
7.5 Start-up method ... 41

8. Hardware specifications ... 42
8.1 User interface ... 42
8.2 Peripheral functions ... 44

9. Software specifications and configuration ... 46
9.1 Software specifications ... 46
9.2 Overall configuration of the software ... 47
9.3 Explanation about interrupt ... 48
9.4 File and folder configuration ... 49
9.5 Application layer ... 52
9.5.1 Functions ... 52
9.5.2 Configurations ... 52
9.5.3 Structure and variable information ... 53
9.5.4 Macro definition ... 55
9.6 Interface Module ... 58
9.6.1 Functions ... 58
9.6.2 Module configuration diagram ... 59
9.6.3 State transition ... 60
9.6.4 Protection function ... 61
9.6.5 API ... 62
9.6.6 Structure and variable information ... 63
9.6.7 Macro and enumeration definition ... 65
9.7 120-degree hall Control Module ... 68
9.7.1 Functions ... 68
9.7.2 Module configuration diagram ... 68
9.7.3 API ... 69
9.7.4 Structure and variable information ... 70
9.7.5 Macro and enumeration definition ... 74

9.8 120-degree driver Module 77
9.8.1 Functions 77
9.8.2 Module configuration diagram 77
9.8.3 API 78
9.8.4 Structure and variable information 79
9.8.5 Macro and enumeration definition 83

10. Setting the parameters 87
10.1 Overview 87
10.2 List of Interface module parameters 87
10.3 List of 120-degree control module parameters 88
10.4 List of 120-degree driver module parameters 91
10.5 Parameters about protection function 95
10.6 Changing the PWM carrier frequency 95
10.7 modulation method (chopping pattern) 95
10.8 Parameters for inverter 96
10.8.1 Deadtime 96
10.8.2 Current detection gain 96
10.8.3 Voltage detection gain 96
10.9 Motor parameters 97

11. Control flowcharts 99
11.1 Main process 99
11.2 Carrier periodic interrupt process 100
11.3 Speed control Periodic Interrupt Process 101
11.4 Hall signal interrupt process 102
11.5 Over Current Detection Interrupt Process 103

12. FAQ 104
12.1 Troubleshooting 104
12.2 Frequently asked questions 105
12.2.1 The value of a variable displayed in RMW is abnormal. 105

Revision History 106

1. Introduction

This application note explains how to implement the 120-degree conducting 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_HALL_120_E2S_V120
- RA6T2_MCB2_MCILV1_SPM_HALL_120_E2S_V100
- RA6T2_MCILV1_SPM_HALL_120_E2S_V120
- RA6T3_MCILV1_SPM_HALL_120_E2S_V110
- RA4T1_MCILV1_SPM_HALL_120_E2S_V110
- RA8T1_MCILV1_SPM_HALL_120_E2S_V110
- RA8T2_MCILV1_SPM_HALL_120_E2S_V100

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 v6.1.0)
120-degree conducting control of permanent magnetic synchronous motor (Algorithm) (R01AN2657)
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	4.8
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	10.9
Changing the PWM carrier frequency	10.6
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) / MCB Ver.1 RTK0EMA270C00000BJ RA6T2 (R7FA6T2BD3CFP) / MCB 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：2025-07	V6.1.0	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 (4.7) are described in detail.

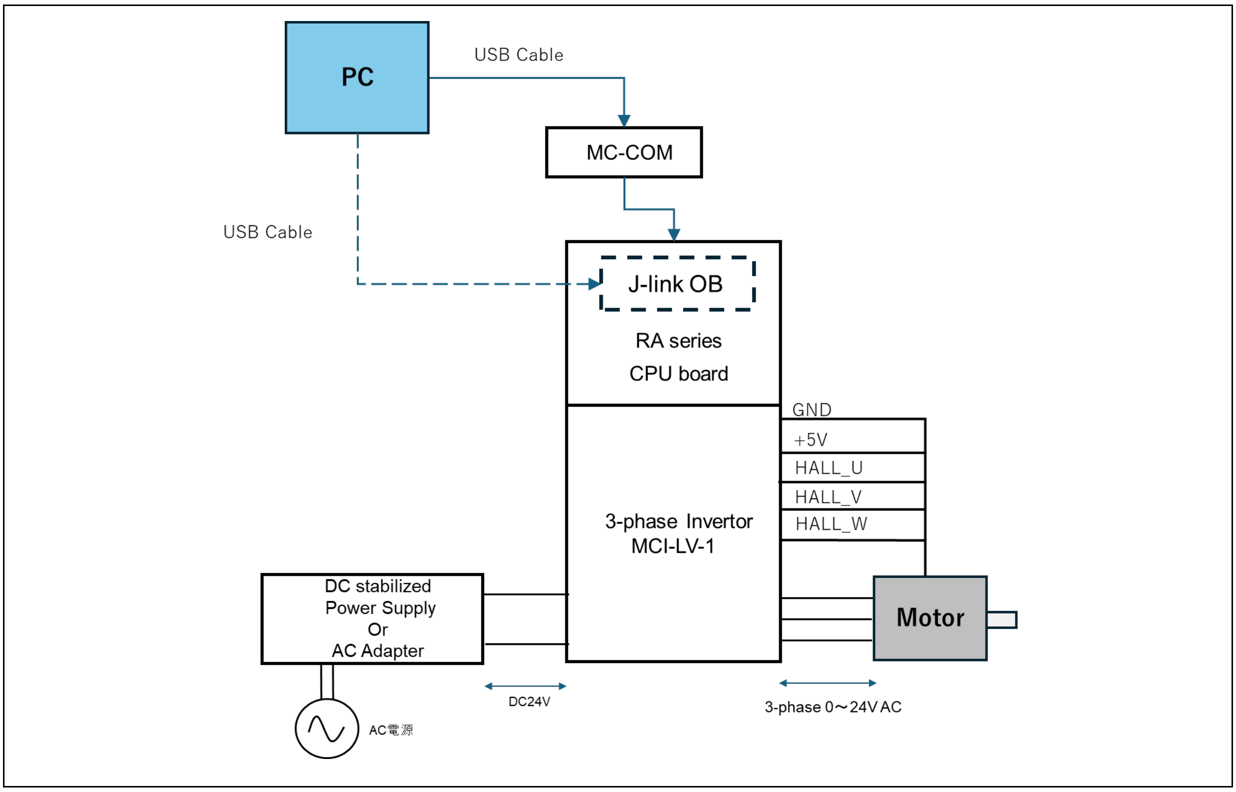


Figure 4-1 Example of hardware configuration

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.

- 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.

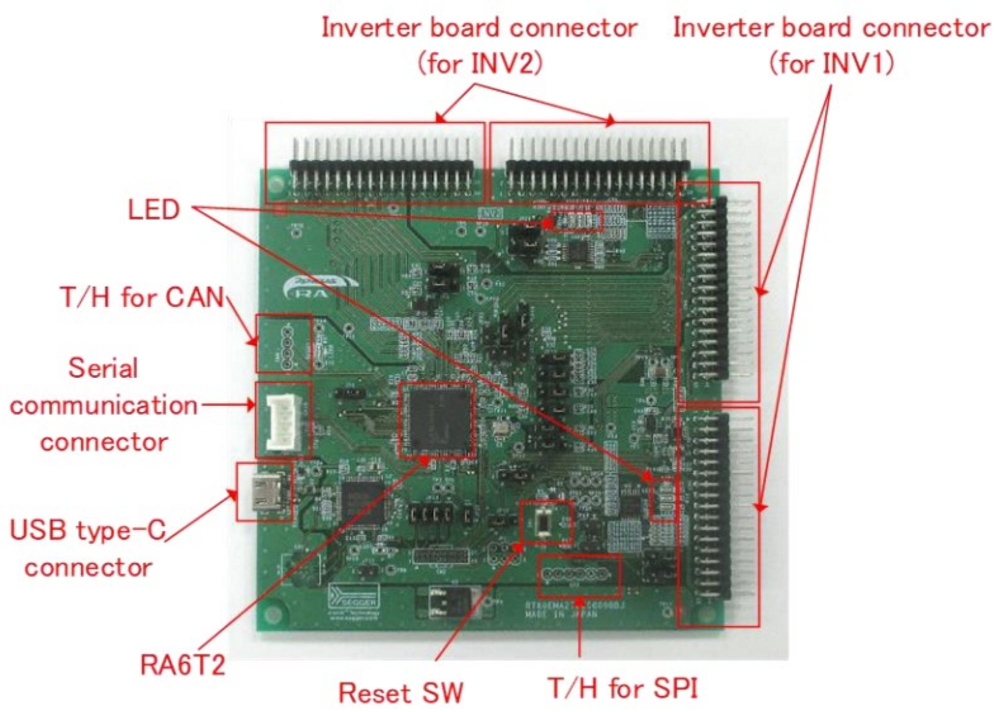


Figure 4-2 RA6T2 CPU board and its interface

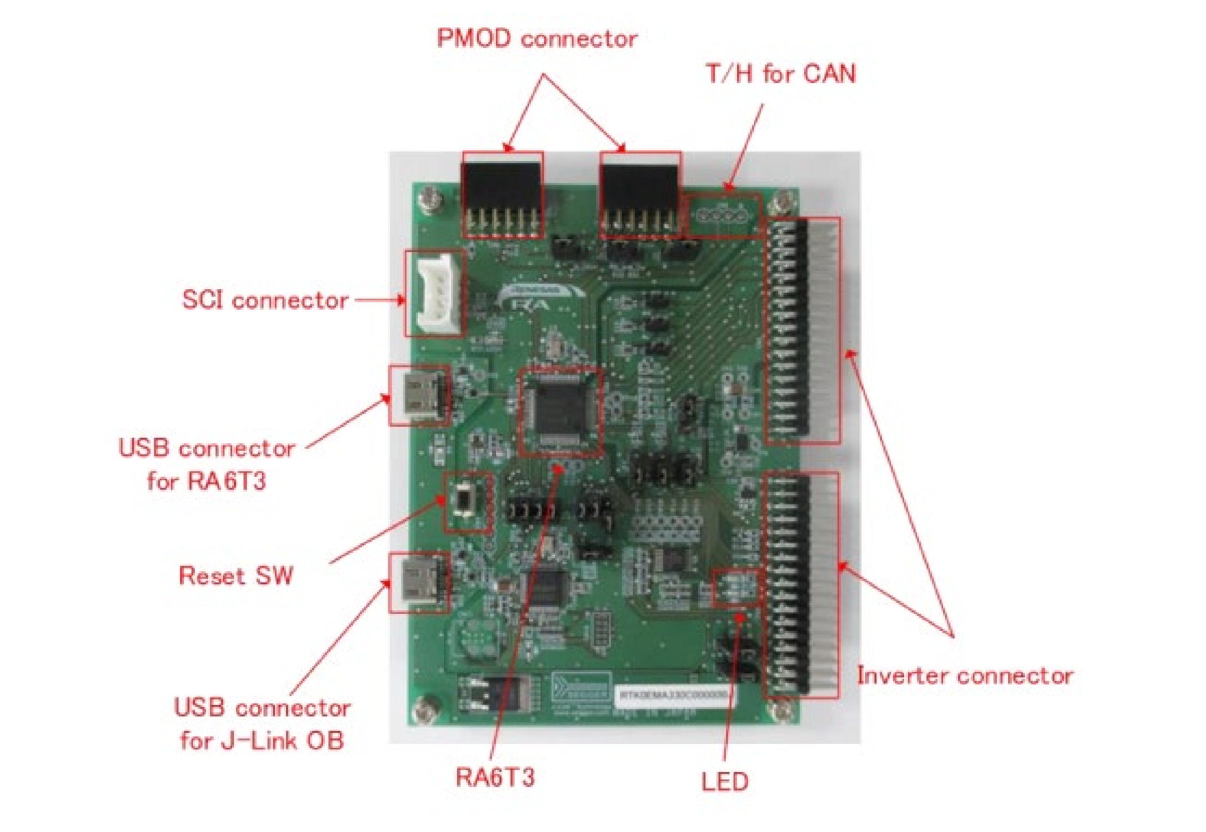


Figure 4-3 RA6T3 CPU board and its interface

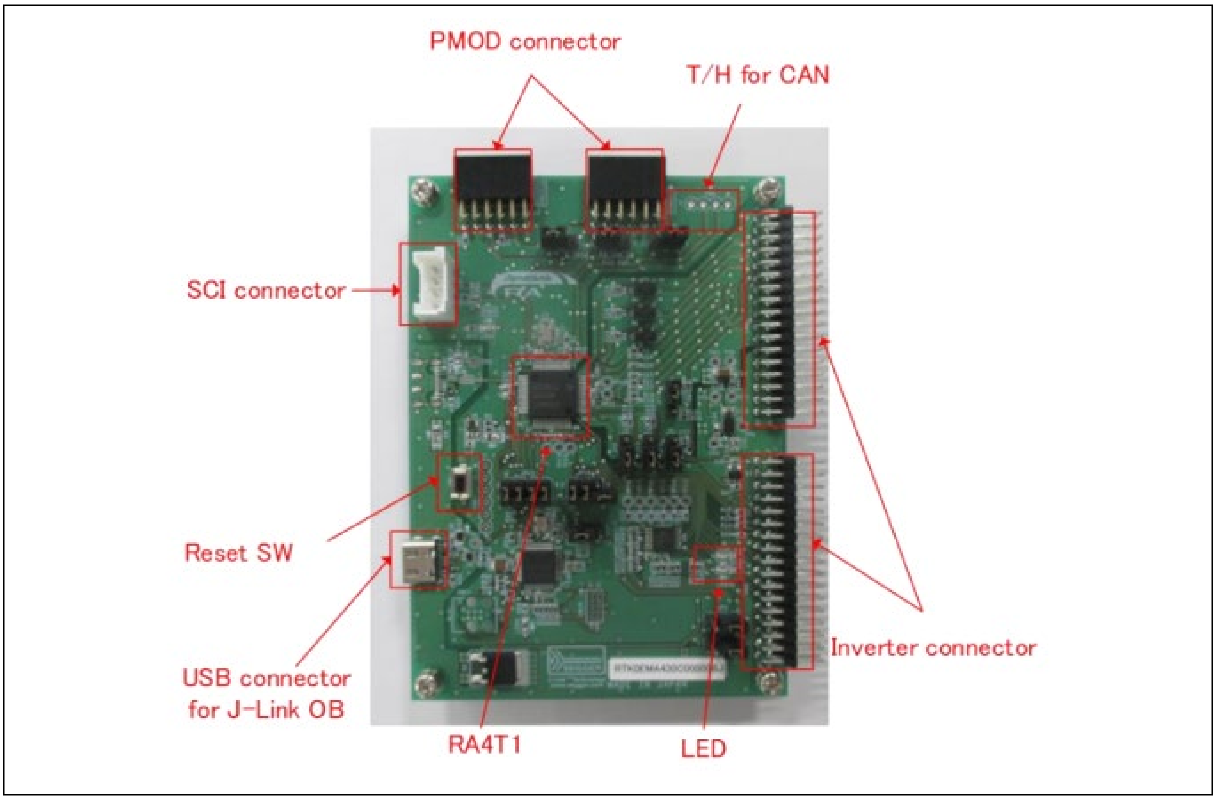


Figure 4-4 RA4T1 CPU board and its interface

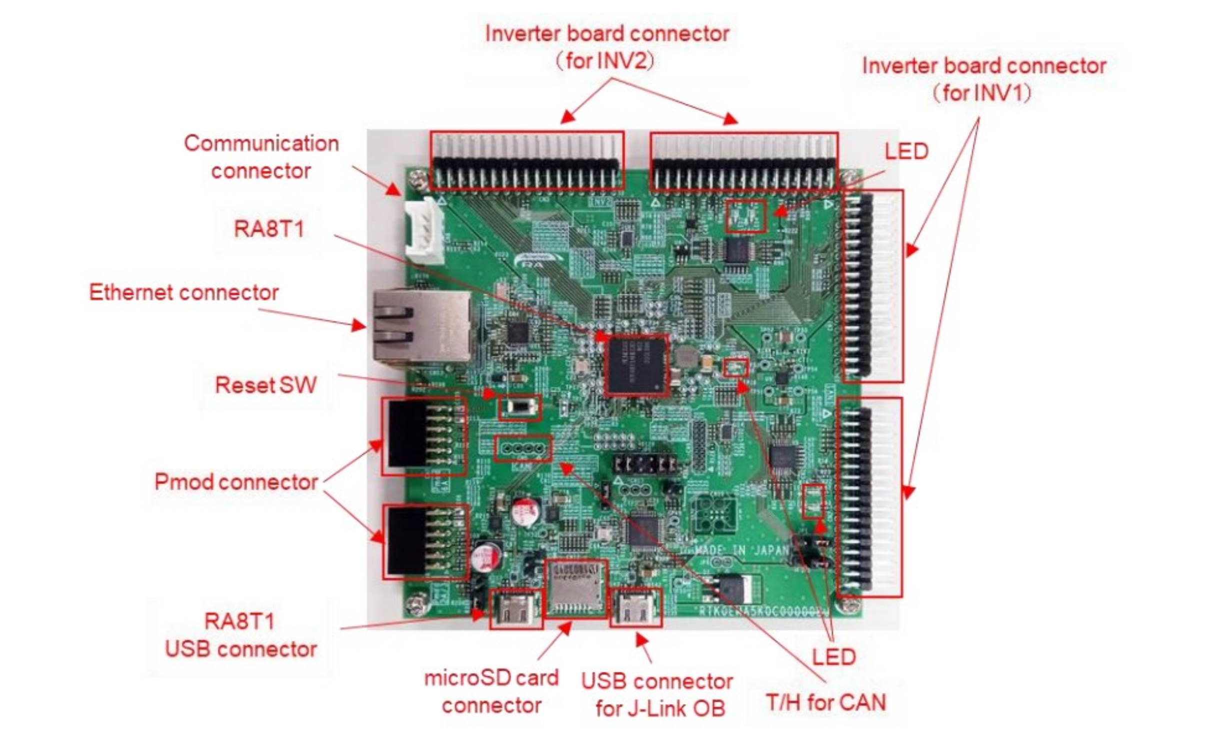


Figure 4-5 RA8T1 CPU board and its interface

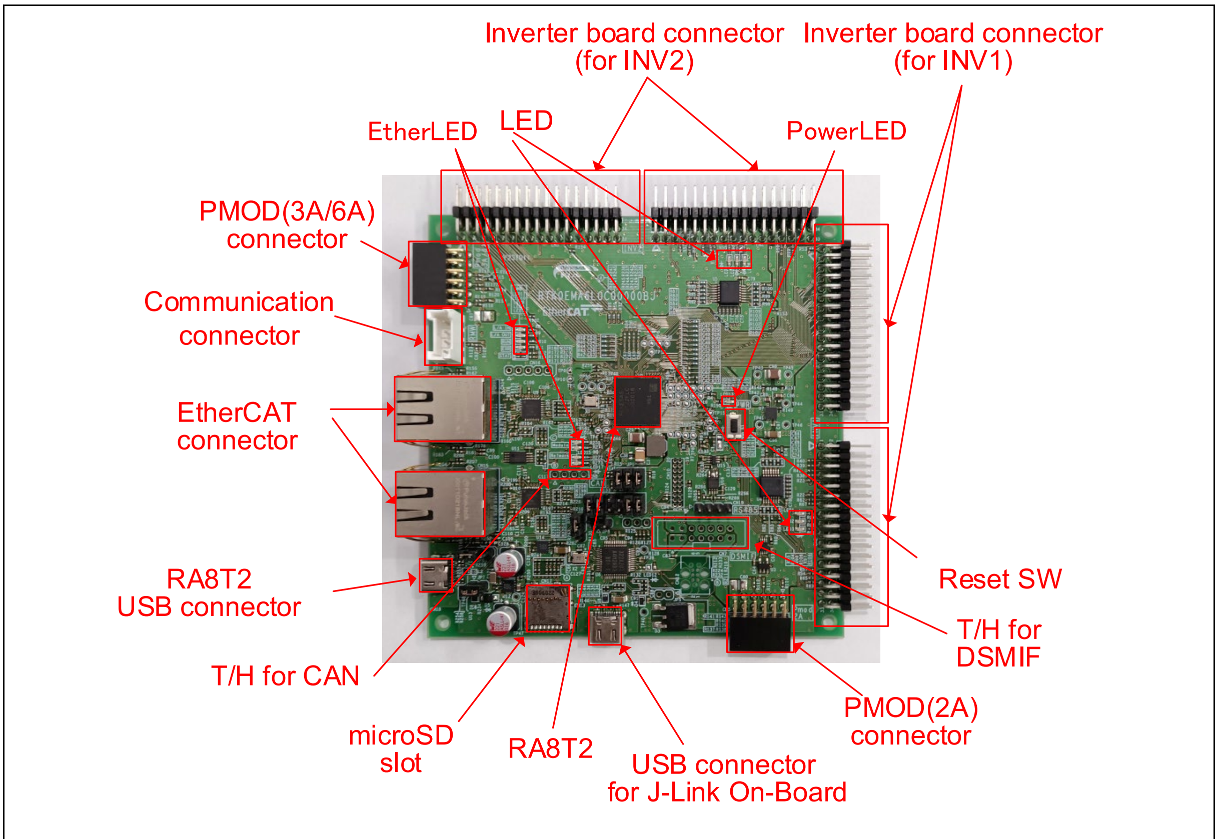


Figure 4-6 RA8T2 CPU board and its interface

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).

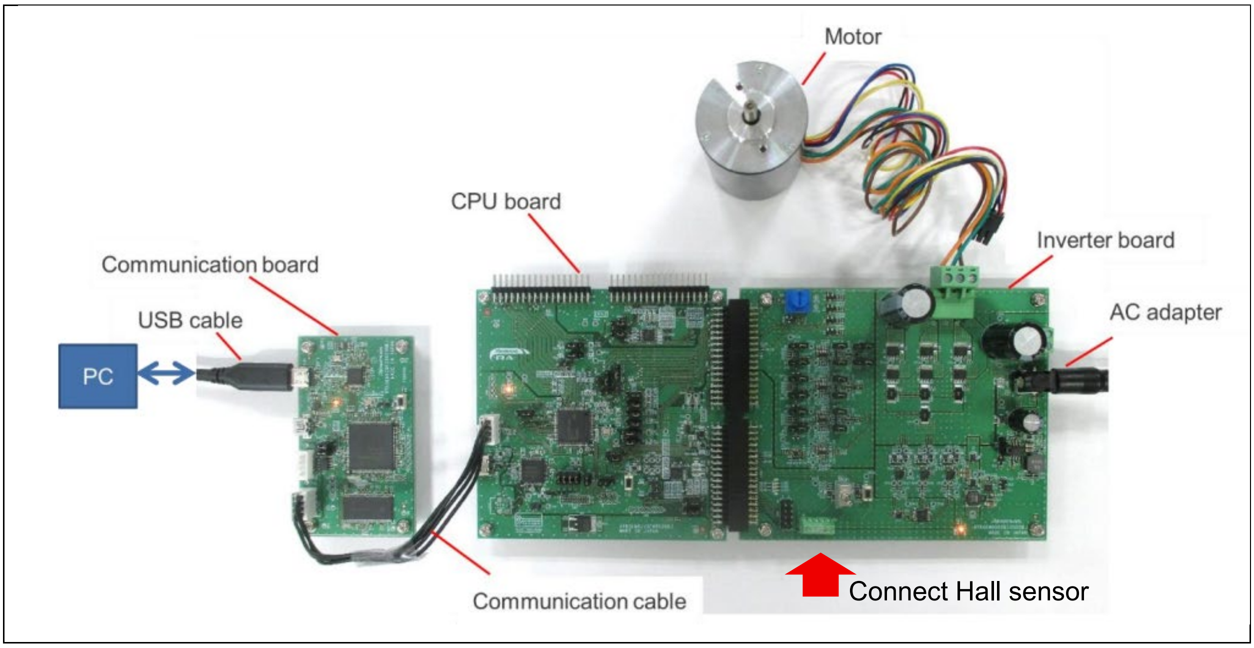


Figure 4-7 Connection example of kit

4.6 Connection of Hall sensor

Please connect the cables of hall sensor signals and connector for hall sensor signal according to Table 4-1. (Please refer Figure 4-8.)

Table 4-1 Connection of signals

Function	Connector port number (in MCI-LV-1)
GND	CN6 1pin
+5V	CN6 2pin
Hall sensor input (HW)	CN6 3pin
Hall sensor input (HV)	CN6 4pin
Hall sensor input (HU)	CN6 5pin

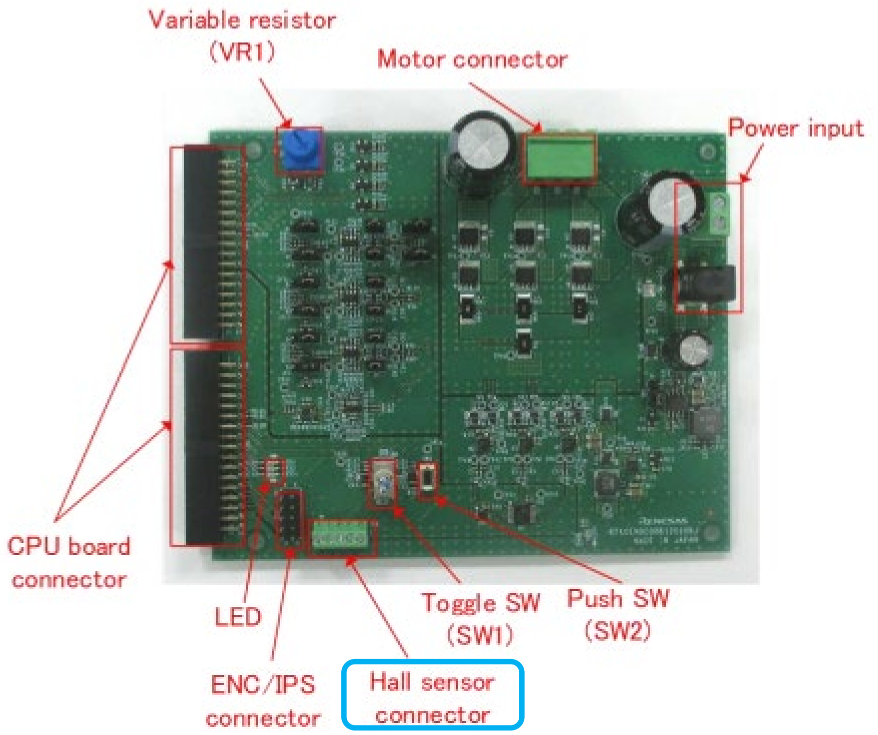


Figure 4-8 Invertor board (MCI-LV-1)

4.7 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.

4.8 Wiring

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

Figure 4-9 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 4-10 shows an example of wiring between the inverter and the motor.

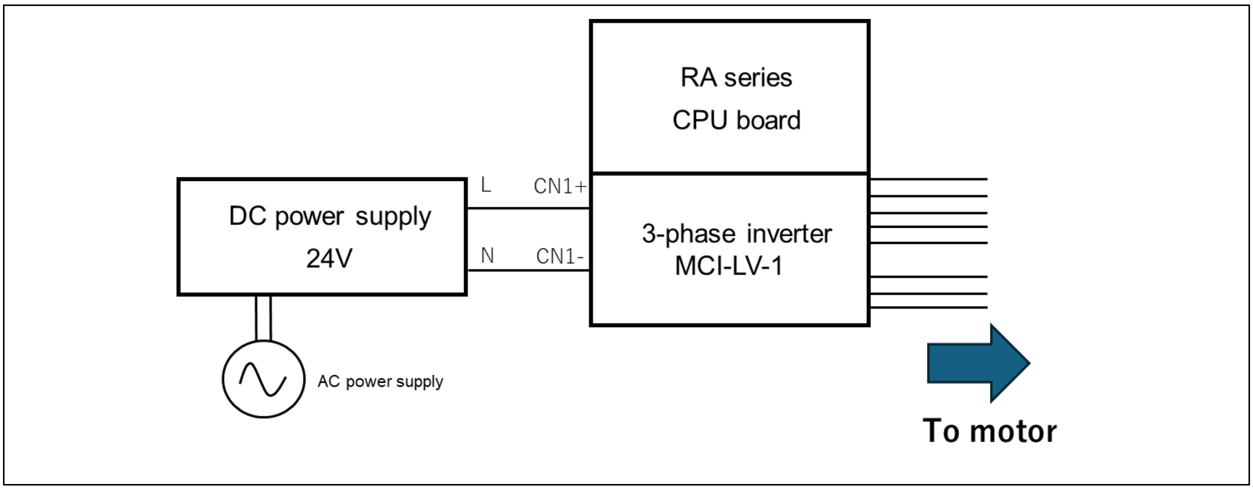


Figure 4-9 Wiring between the power supply and the inverter

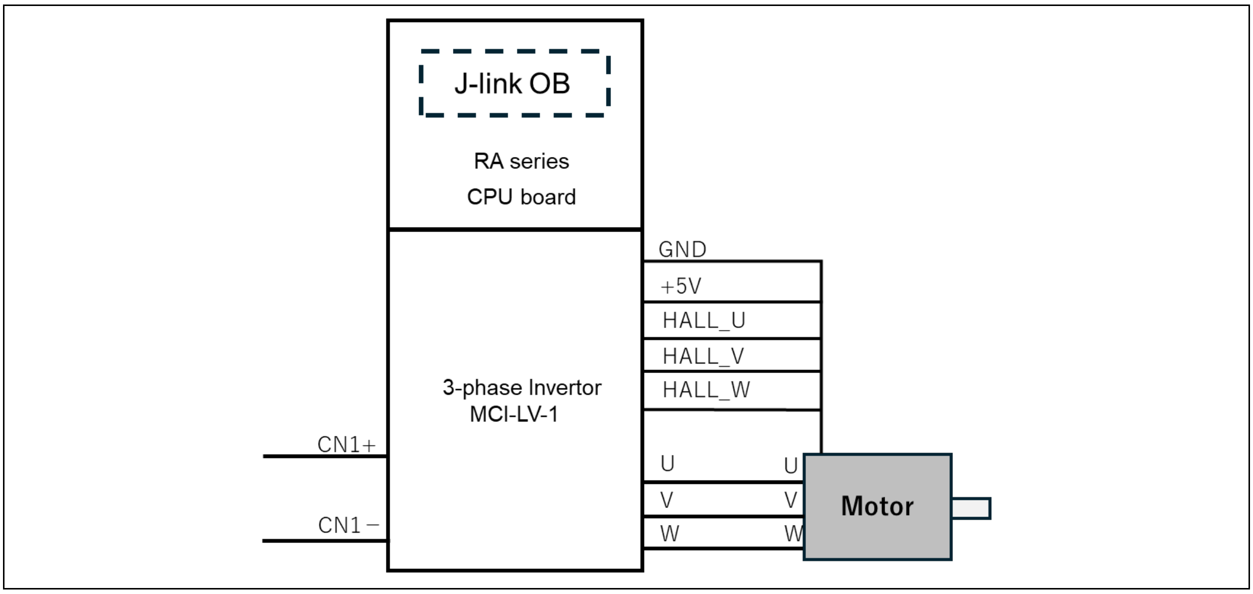


Figure 4-10 Wiring between the inverter and the motor

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.

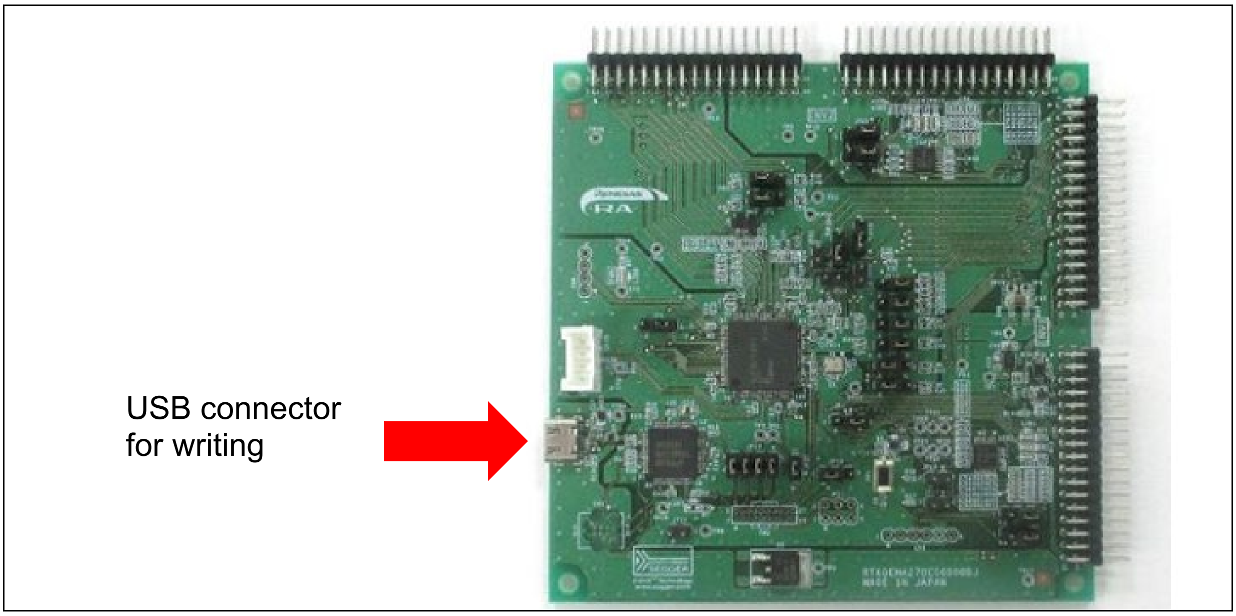


Figure 6-1 RA6T2 CPU board USB connector for writing

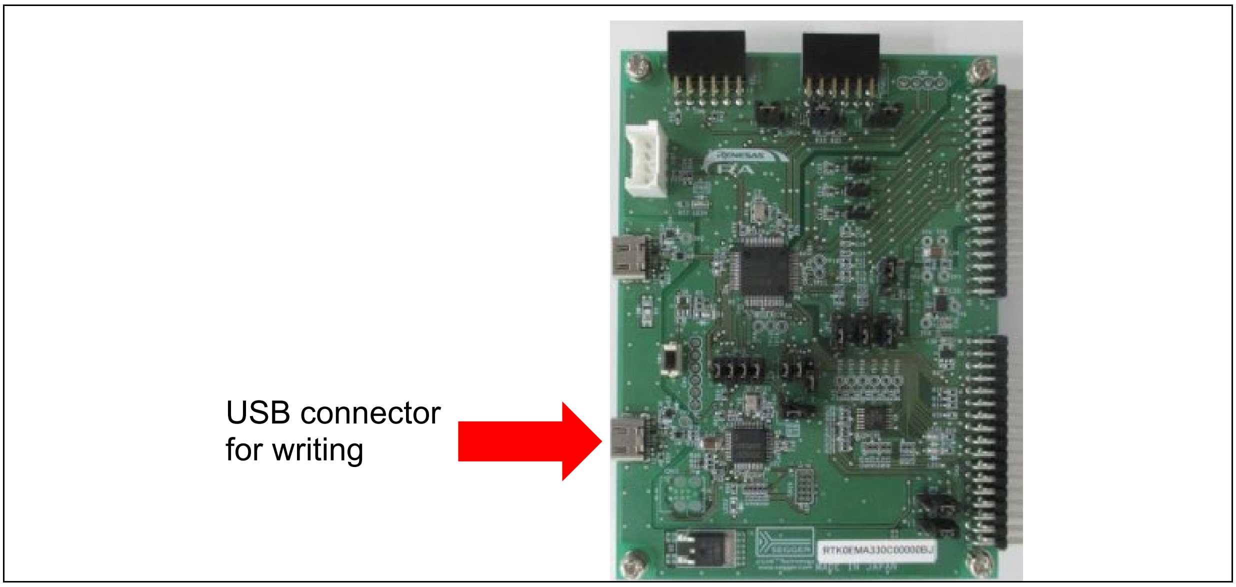


Figure 6-2 RA6T3 CPU board USB connector for writing

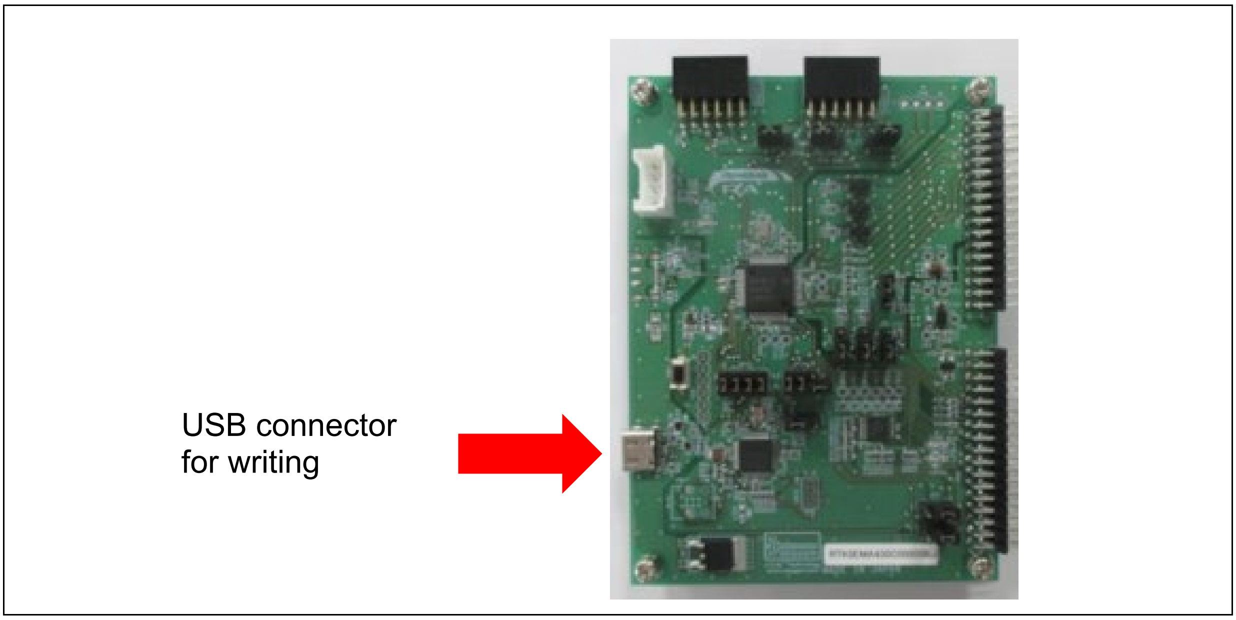


Figure 6-3 RA4T1 CPU board USB connector for writing

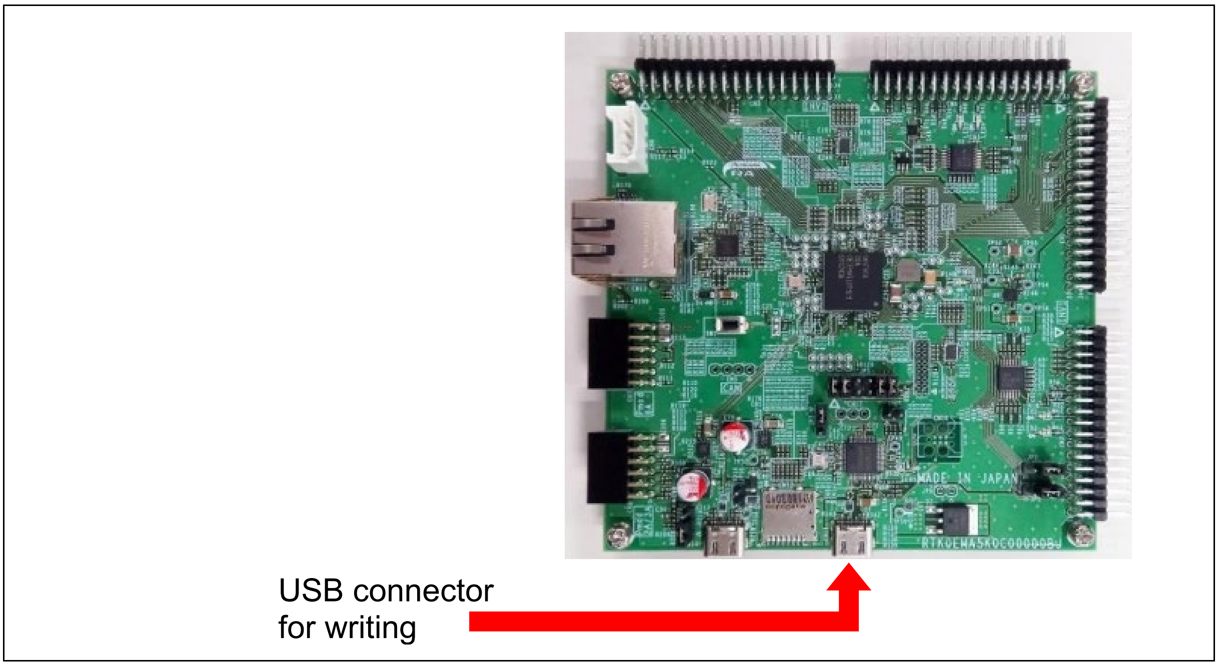


Figure 6-4 RA8T1 CPU board USB connector for writing

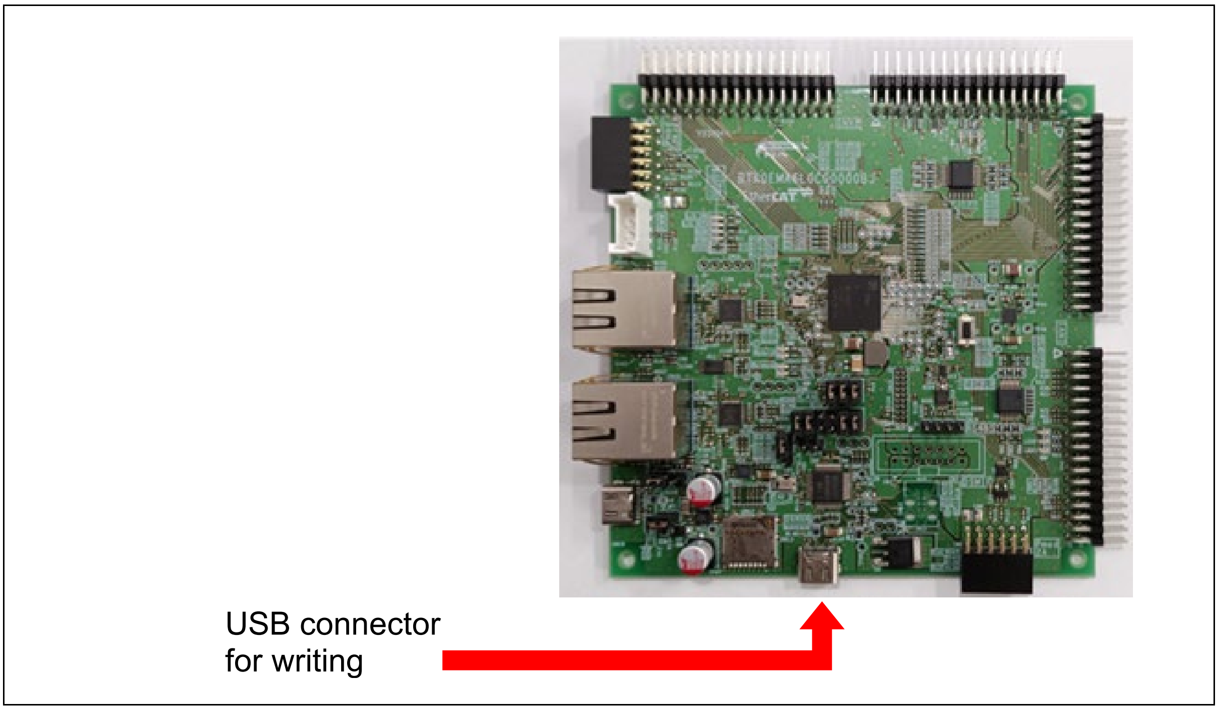


Figure 6-5 RA8T2 CPU board USB connector for writing

(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.

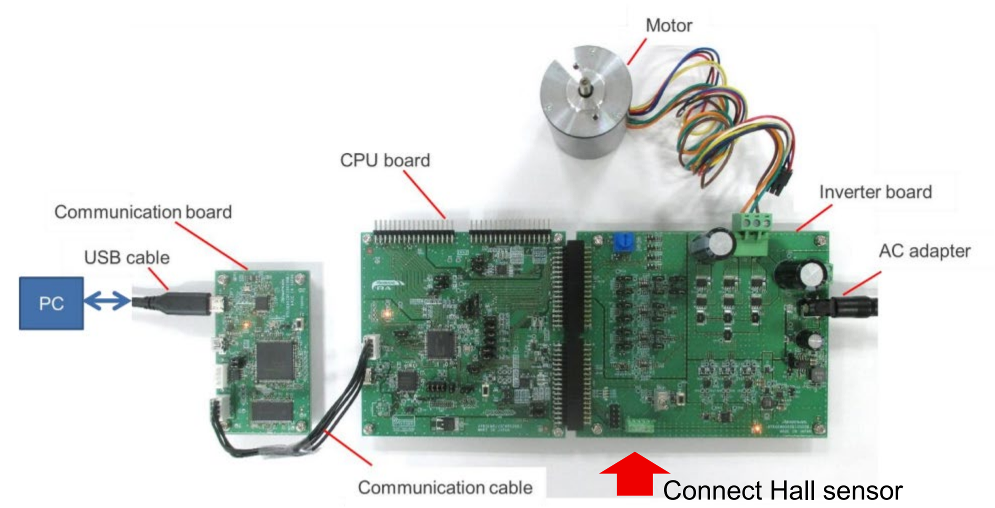


Figure 6-6 Example of connection at investigation

(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 1-2
JP11	Connect 1-2

・RA6T2 board

Number of jumper	Connection
JP4	Connect 1-2
JP5	Connect 1-2

・RA6T3/RA4T1 board

Number of jumper	Connection
JP2	Connect 1-2
JP3	Connect 1-2
JP4	Connect 1-2

・RA8T1/RA8T2 board

There is no change of jumper settings with other sample programs. Therefore, please use as default.

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.

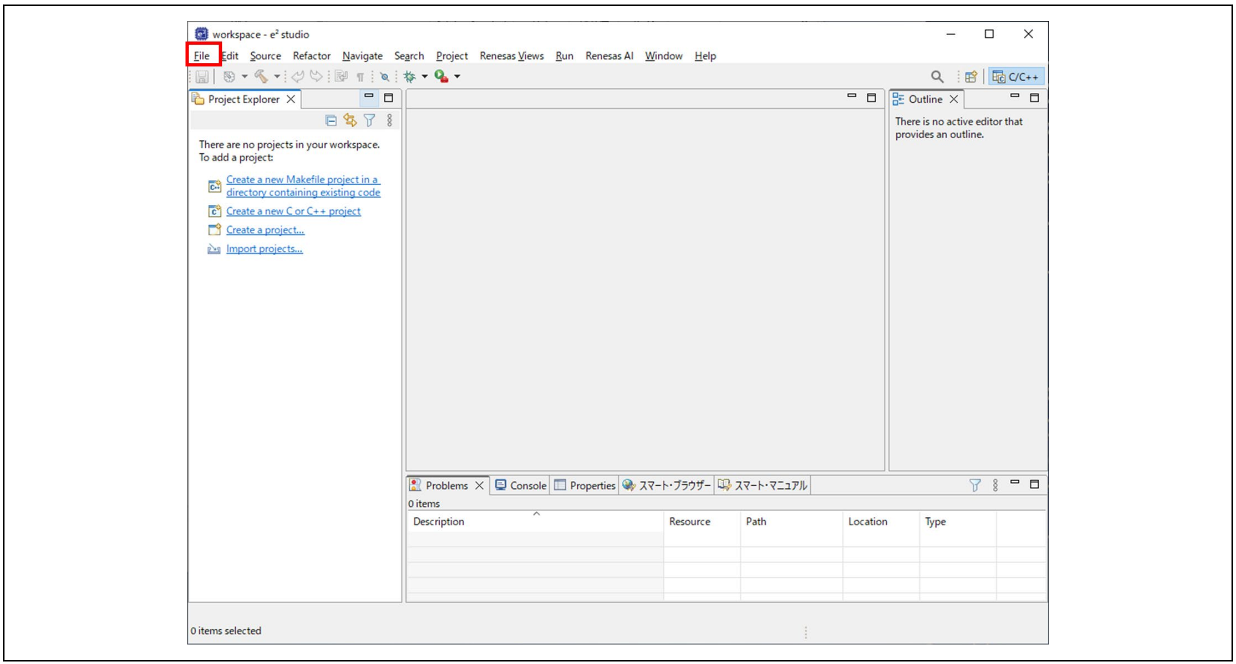


Figure 6-7 Selection of the target project (1)

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

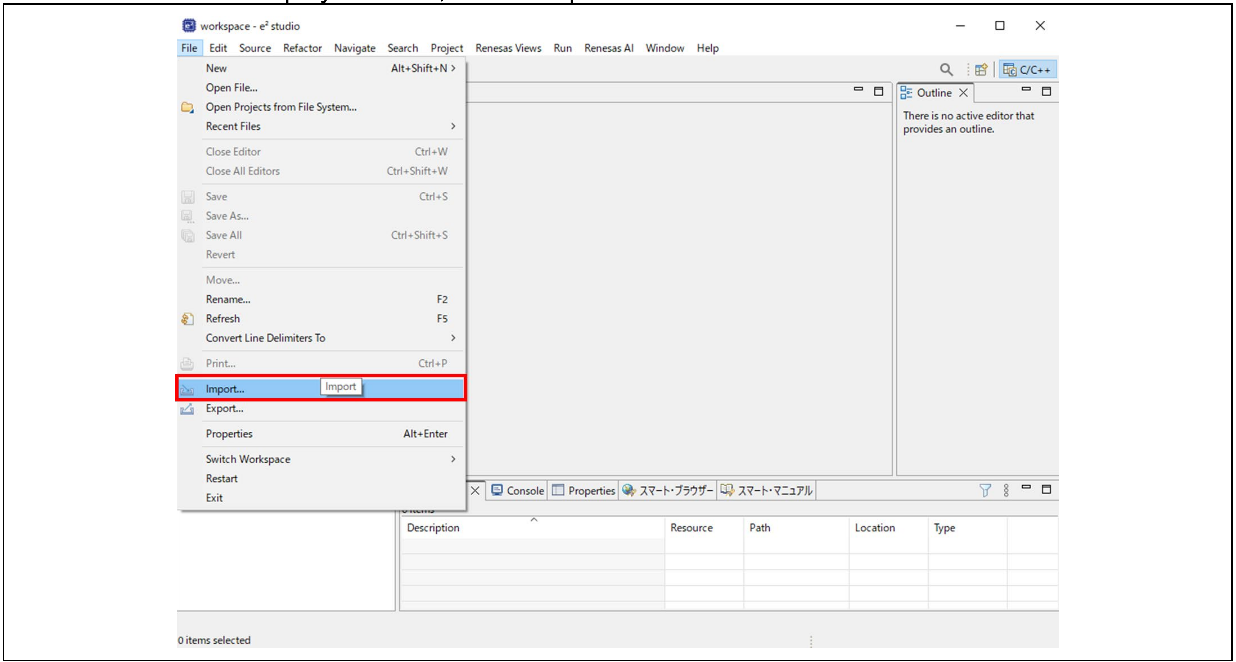


Figure 6-8 Selection of the target project (2)

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

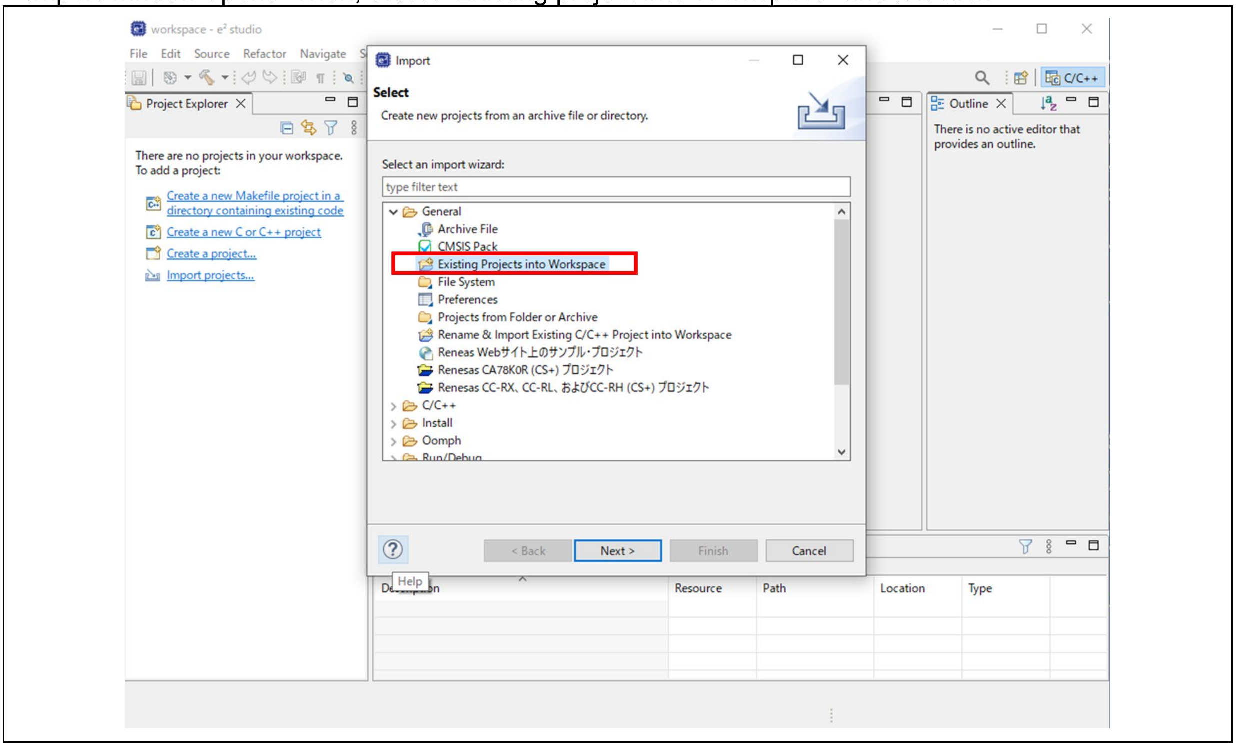


Figure 6-9 Selection of the target project (3)

4. "Import" window opens. Then, left click "Browse".

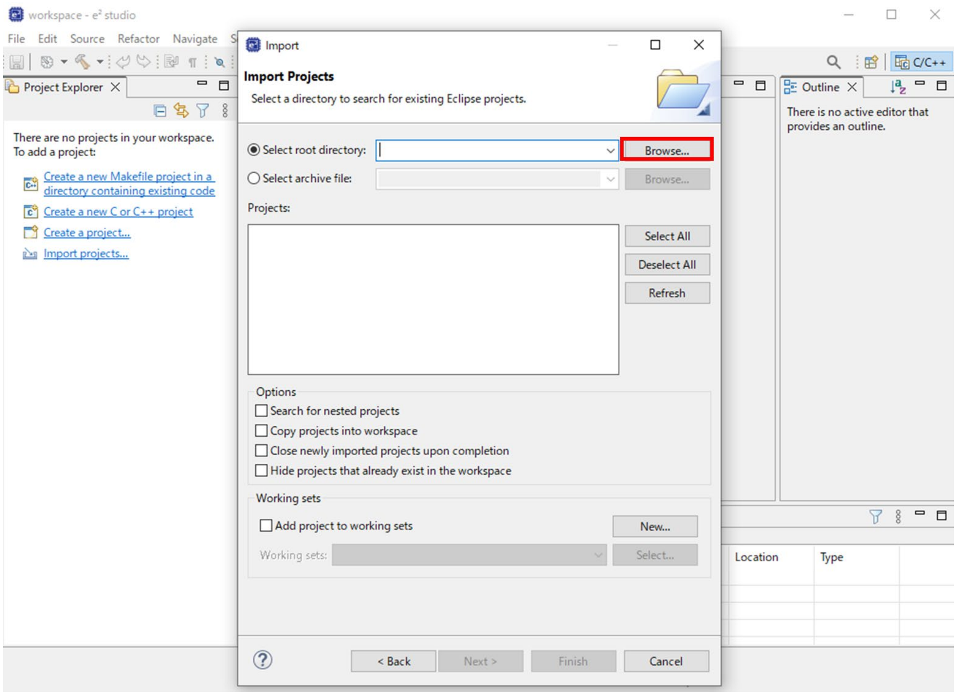


Figure 6-10 Selection of the target project (4)

5. Folder selection window opens. Then select the target folder and left click "Select Folder".

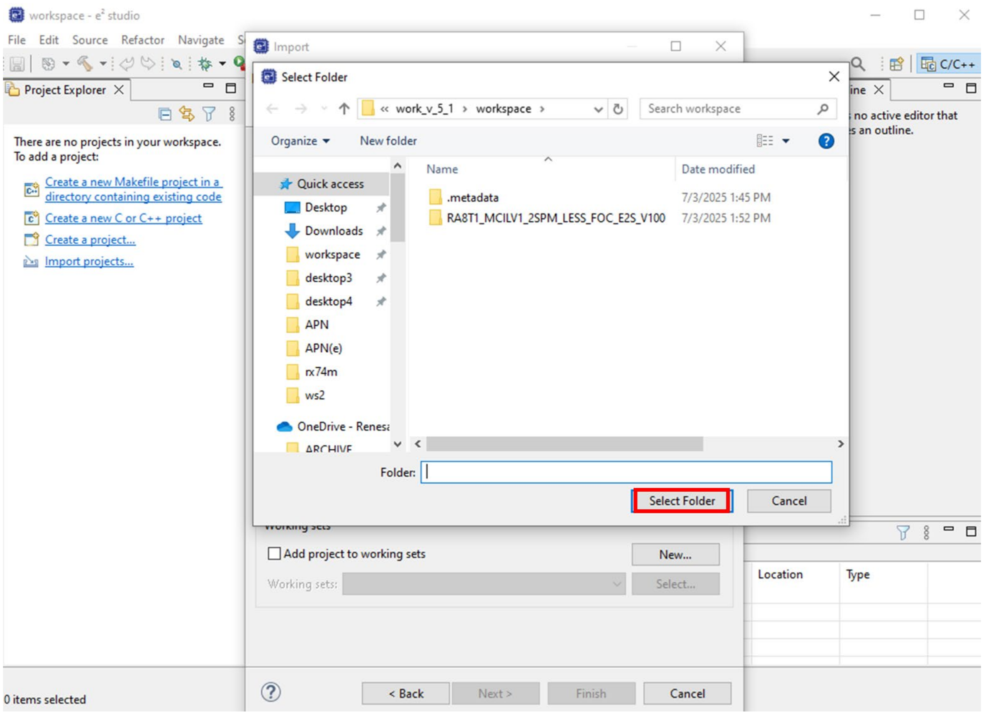


Figure 6-11 Selection of the target project (5)

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

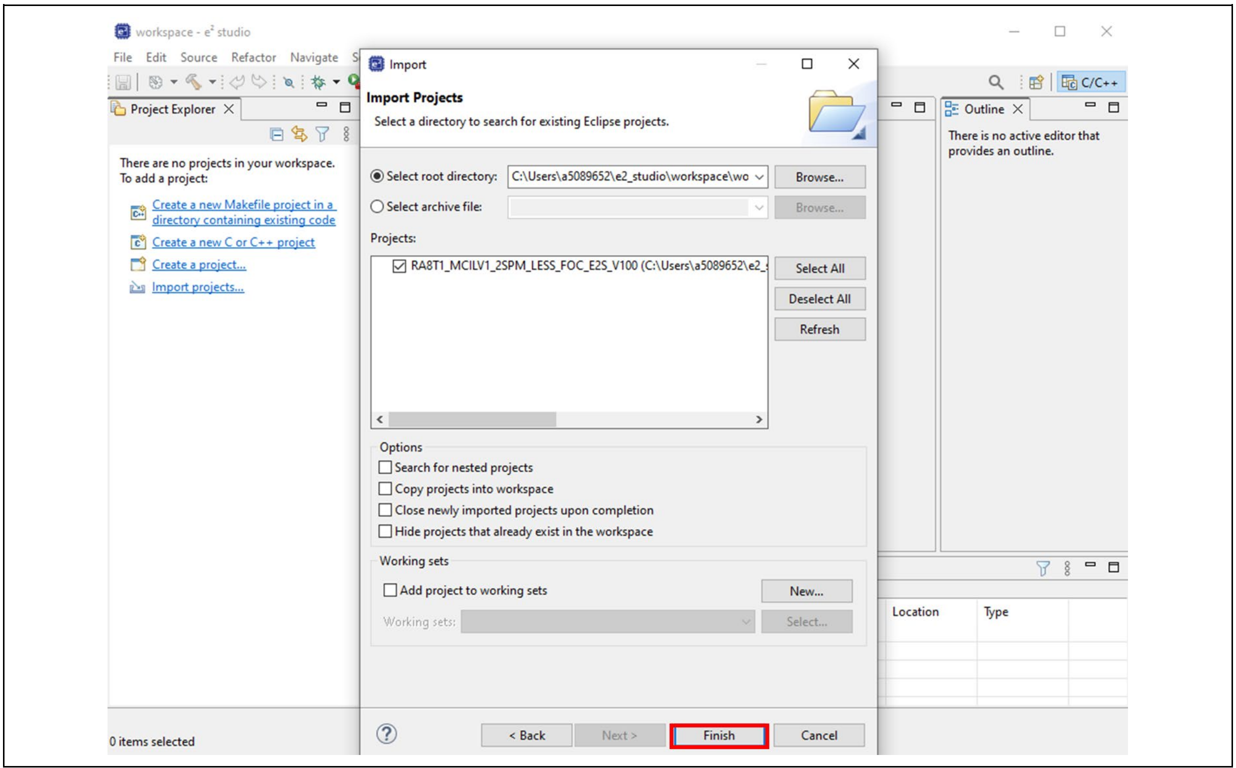


Figure 6-12 Selection of the target project (6)

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

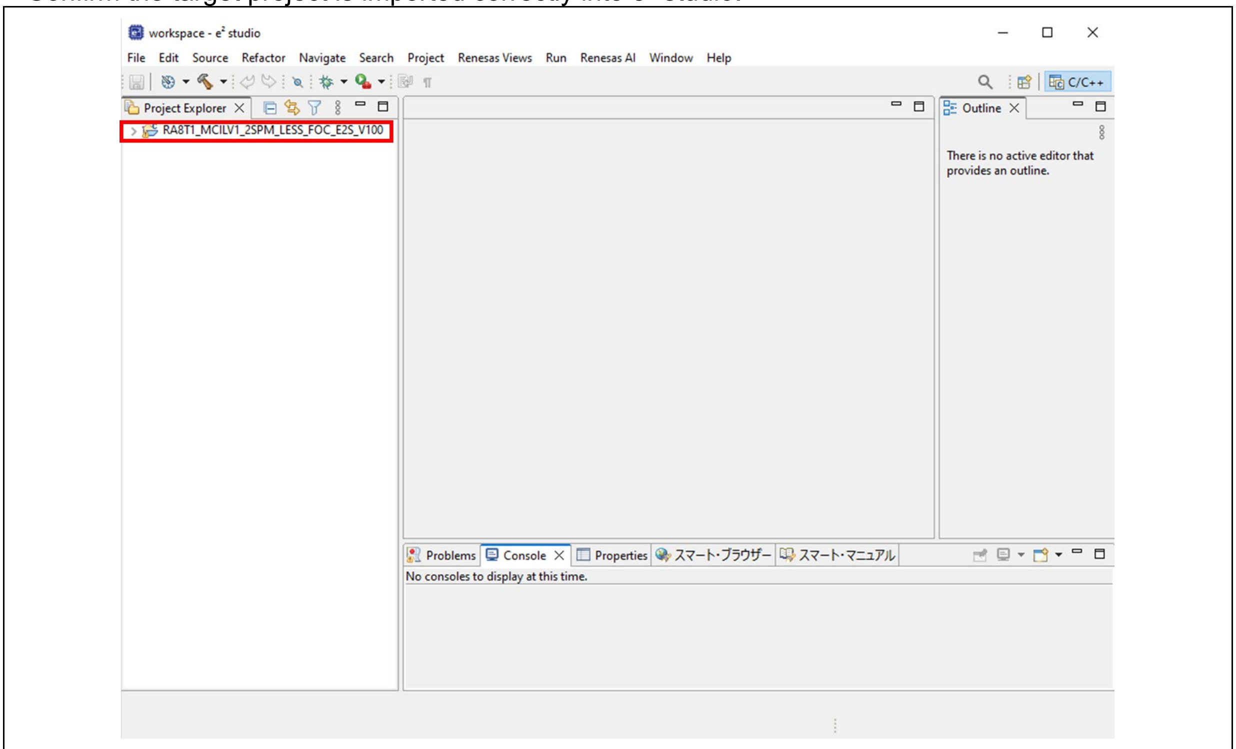


Figure 6-13 Confirmation of import the target project

6.3.3 Build the target project

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

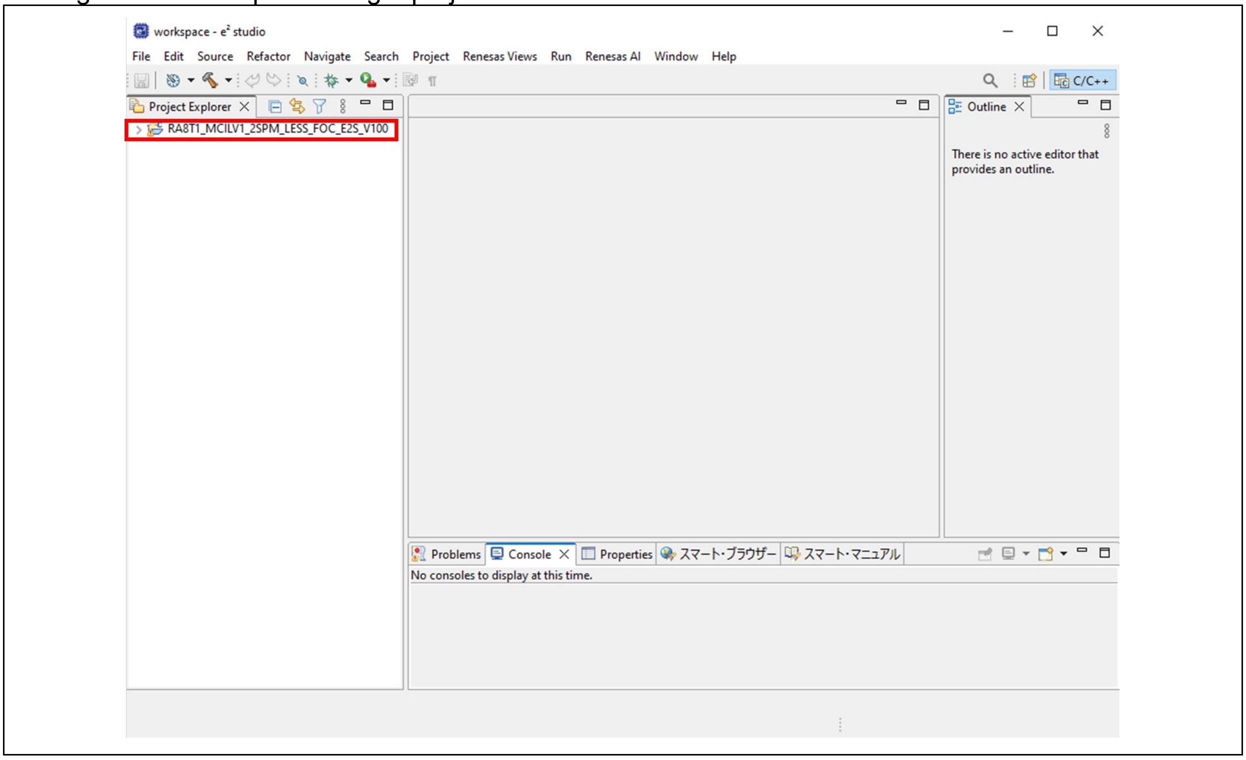


Figure 6-14 Selection of the target project

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

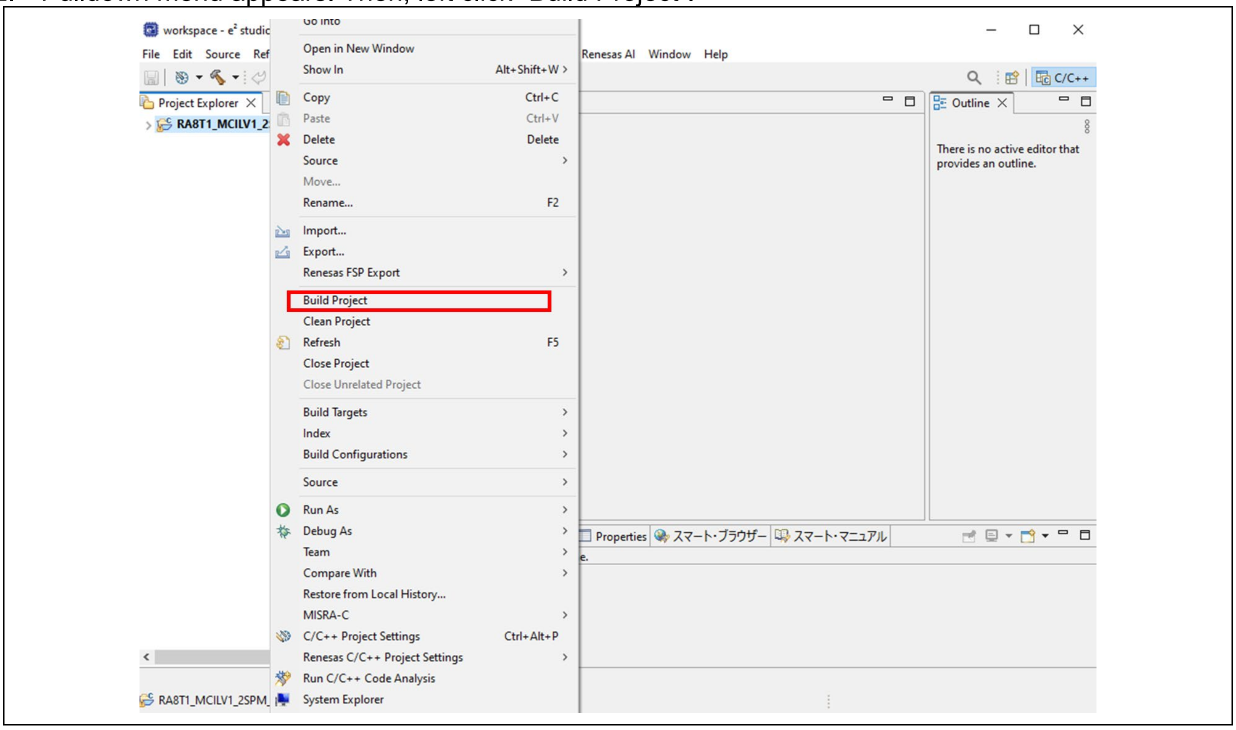


Figure 6-15 Pulldown menu

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

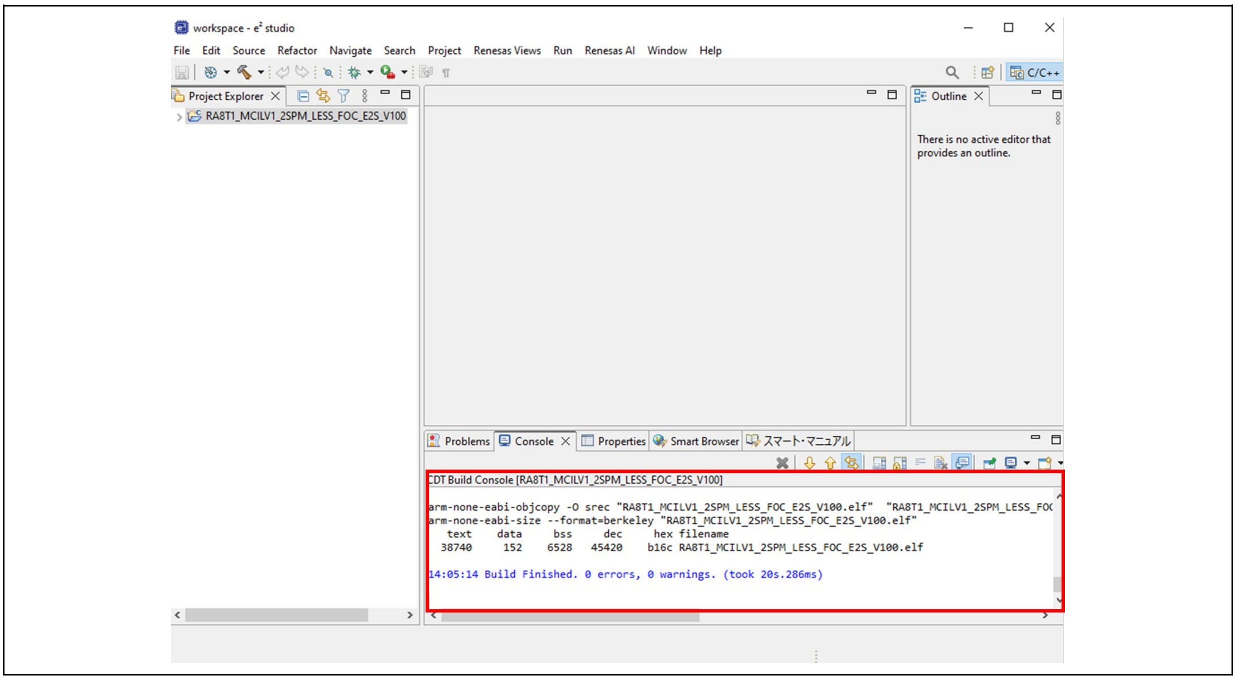


Figure 6-16 Confirmation of finish of build

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.)

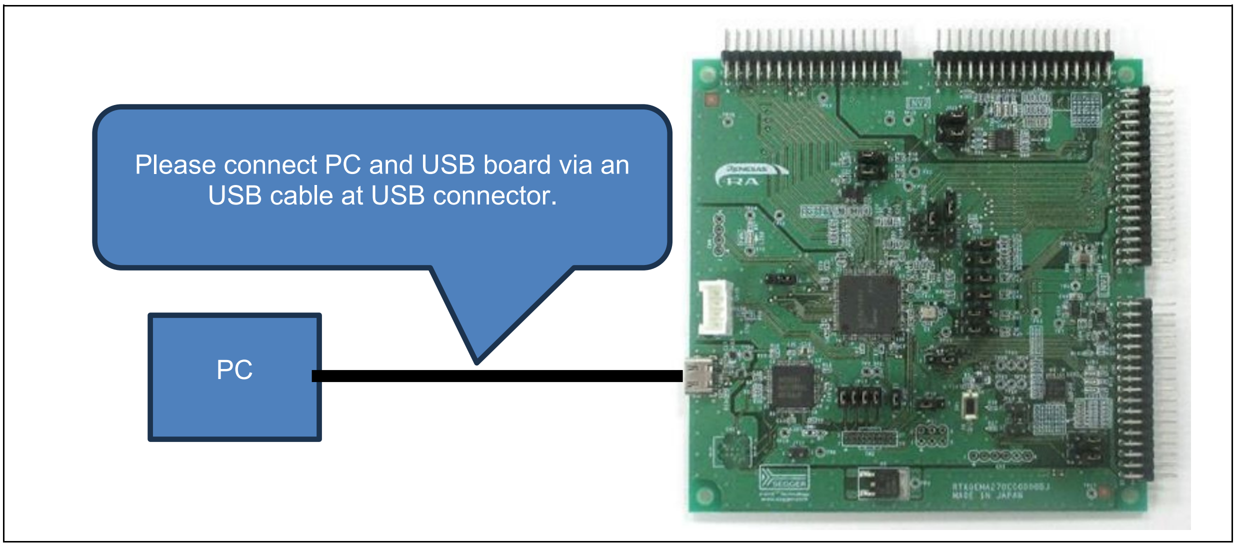


Figure 6-17 Confirmation between PC and CPU board (RA6T2)

6.3.5 Writing to the target board (with built program)

1. Select target project and right click.

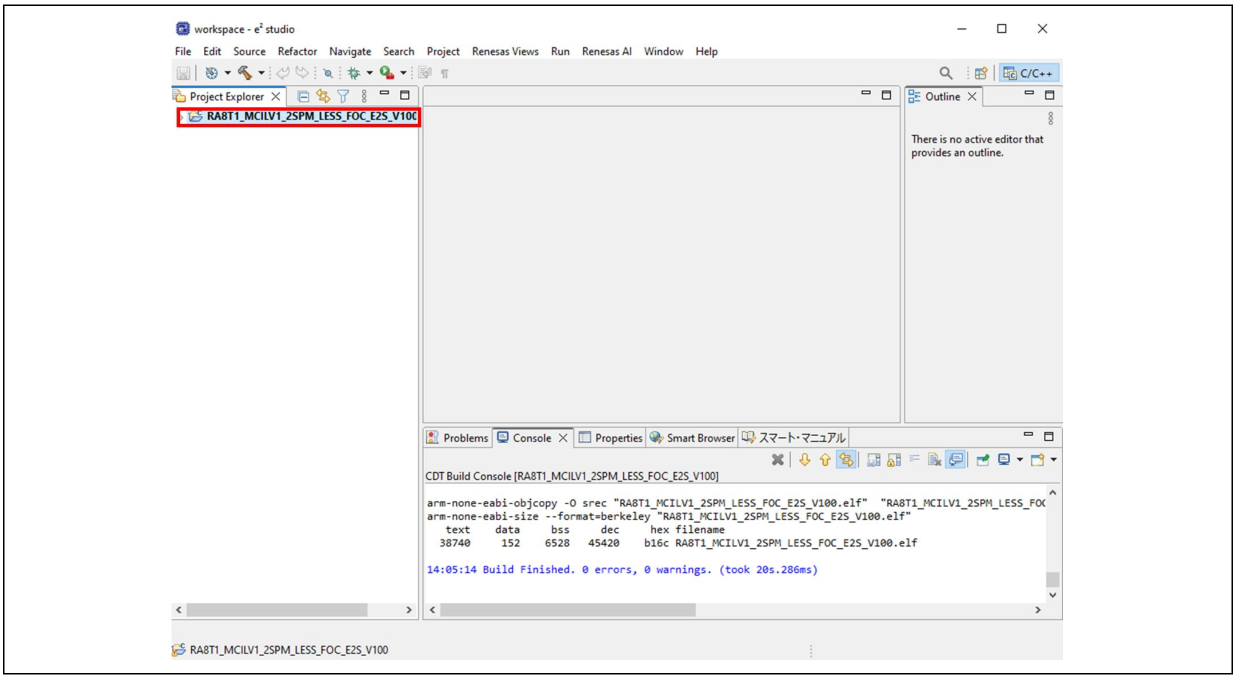


Figure 6-18 Selection of target project

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.

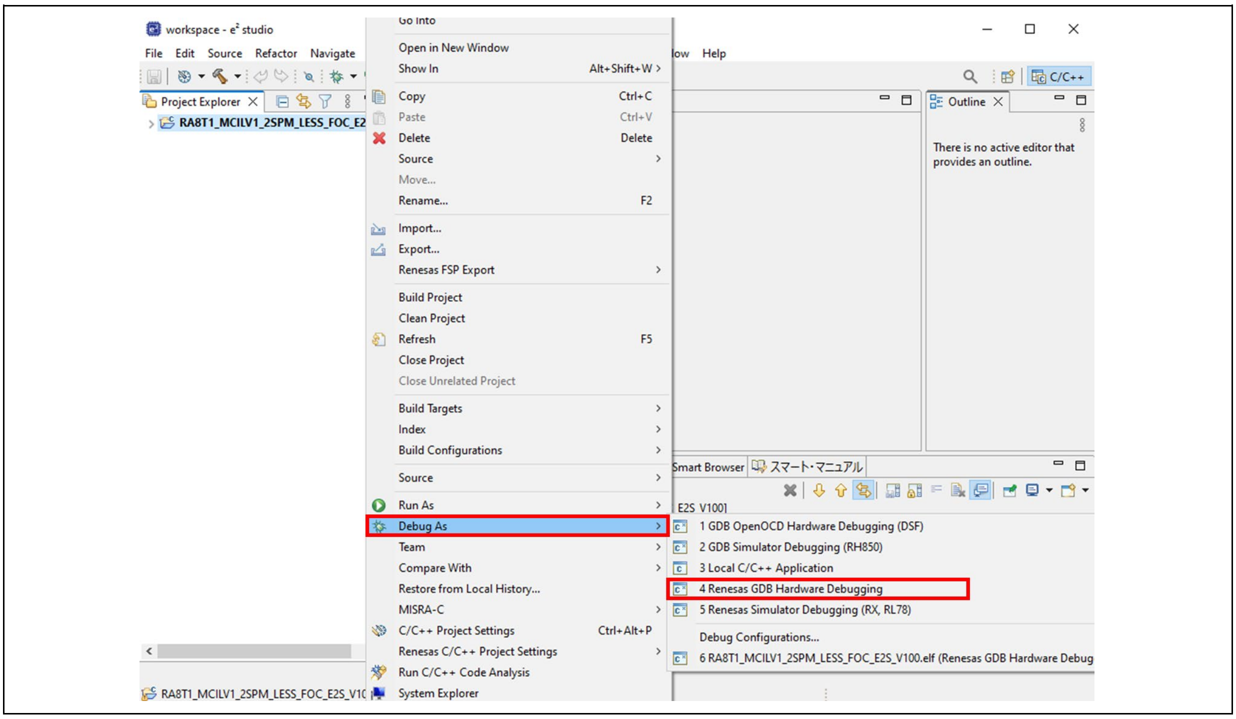


Figure 6-19 Selection of debug method

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

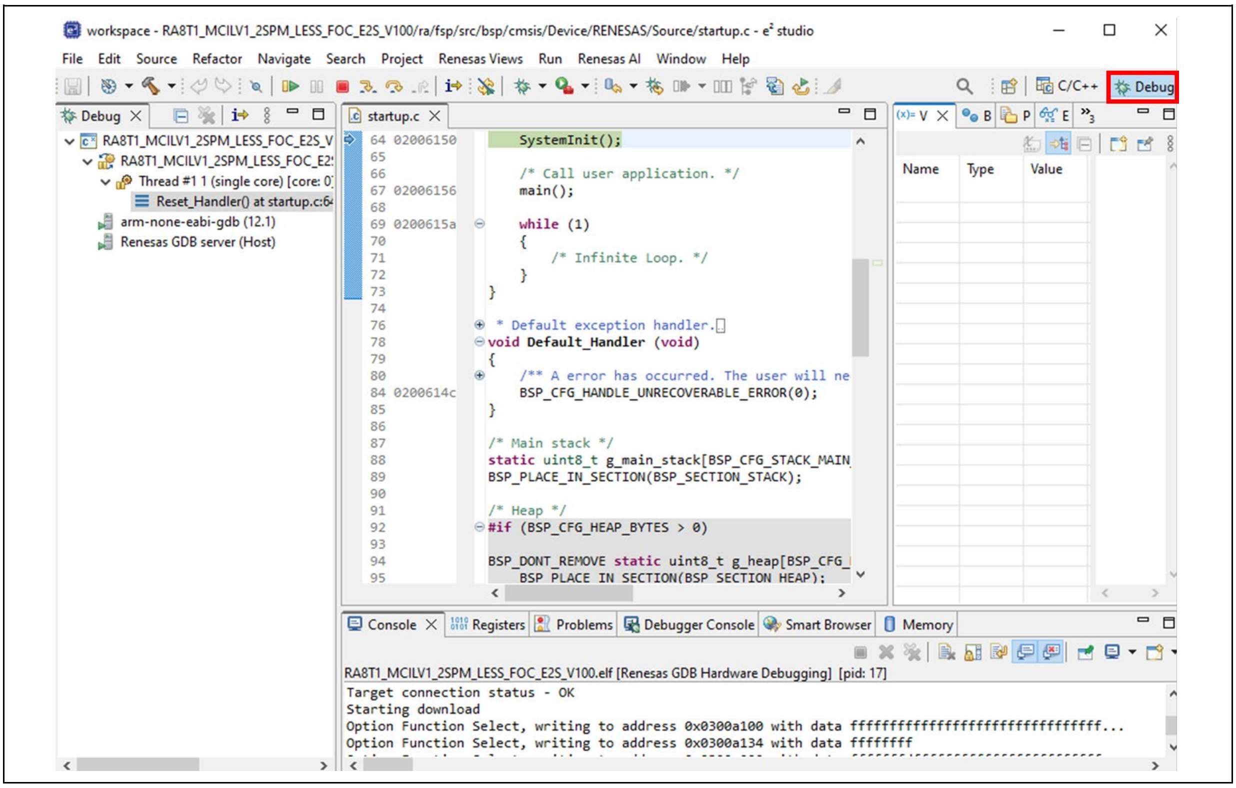


Figure 6-20 “Debug” mode

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

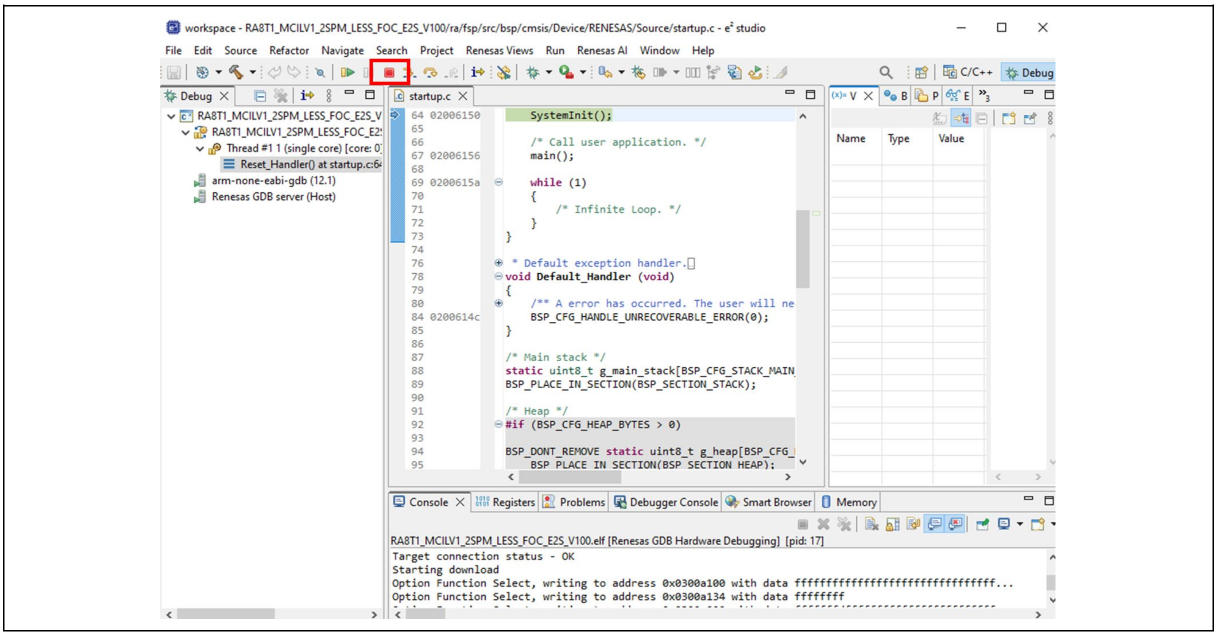


Figure 6-21 Disconnect from target board

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

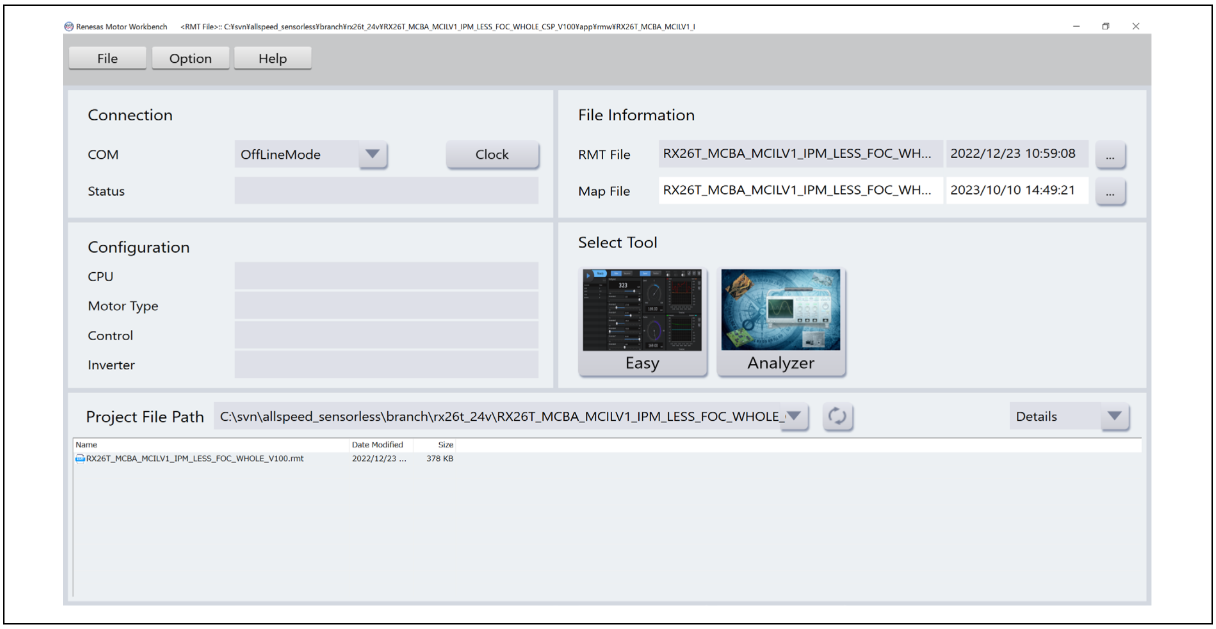


Figure 6-22 Window of Renesas Motor Workbench

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.

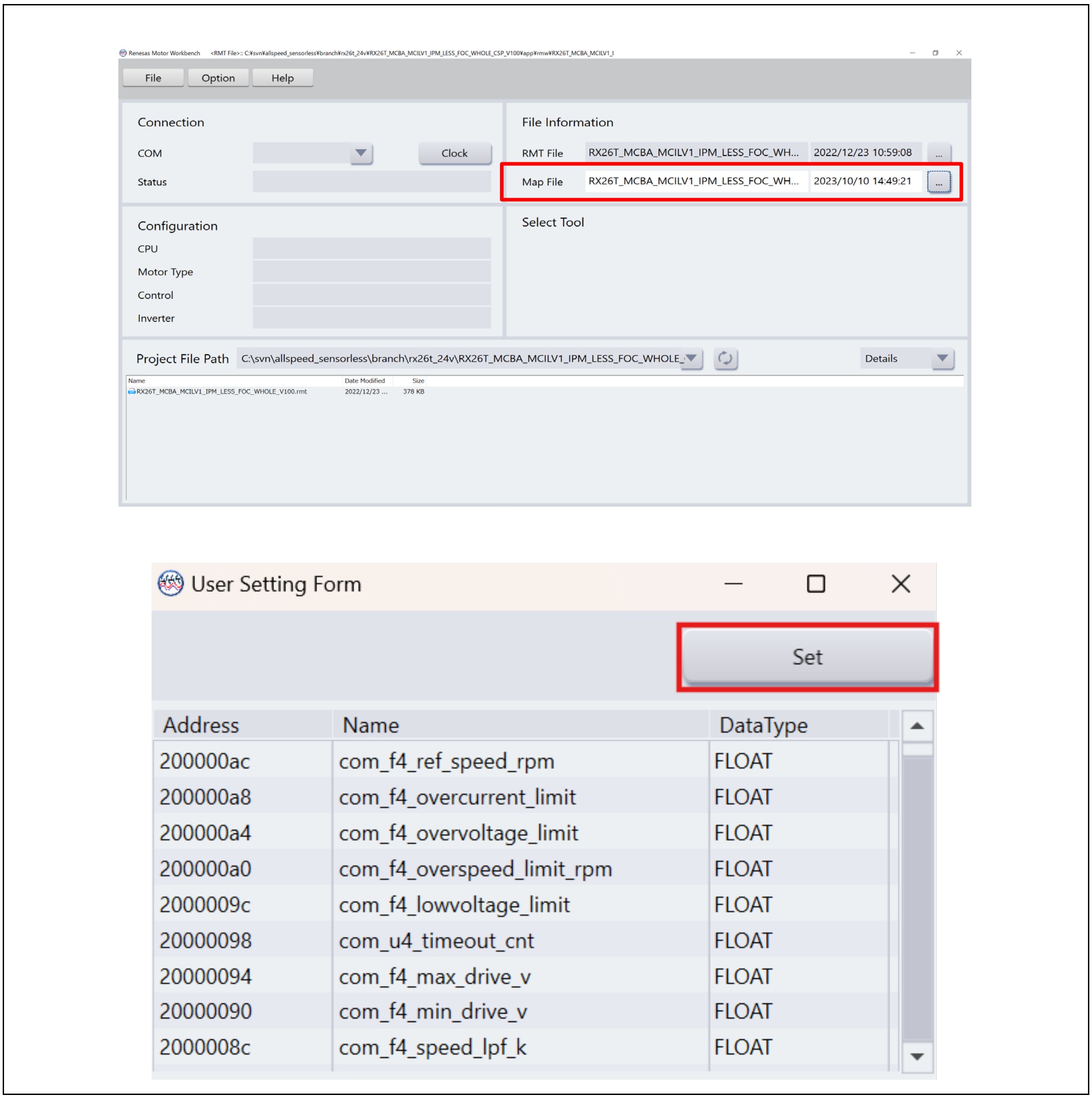


Figure 6-23 Map file registration setup window (upper) and confirmation window (lower) in RMW

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.

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 hall 120-degree conducting control

Hall 120-degree conducting control variable name	Type	Substance
g_f4_iu_offset_monitor	float	U phase offset [A]
g_f4_iw_offset_monitor	float	W phase offset [A]
g_f4_iu_ad_monitor	float	U 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_speed_rad_monitor	float	Rotation speed (electrical) [rad/s]
g_f4_speed_rpm_monitor	float	Rotation speed (mechanical) [rpm]
g_f4_speed_ref_monitor	float	Speed reference (electrical)[rad/s]
g_f4_v_ref_monitor	float	Voltage reference [V]
g_u1_direction_monitor	uint8_t	Rotation direction

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.

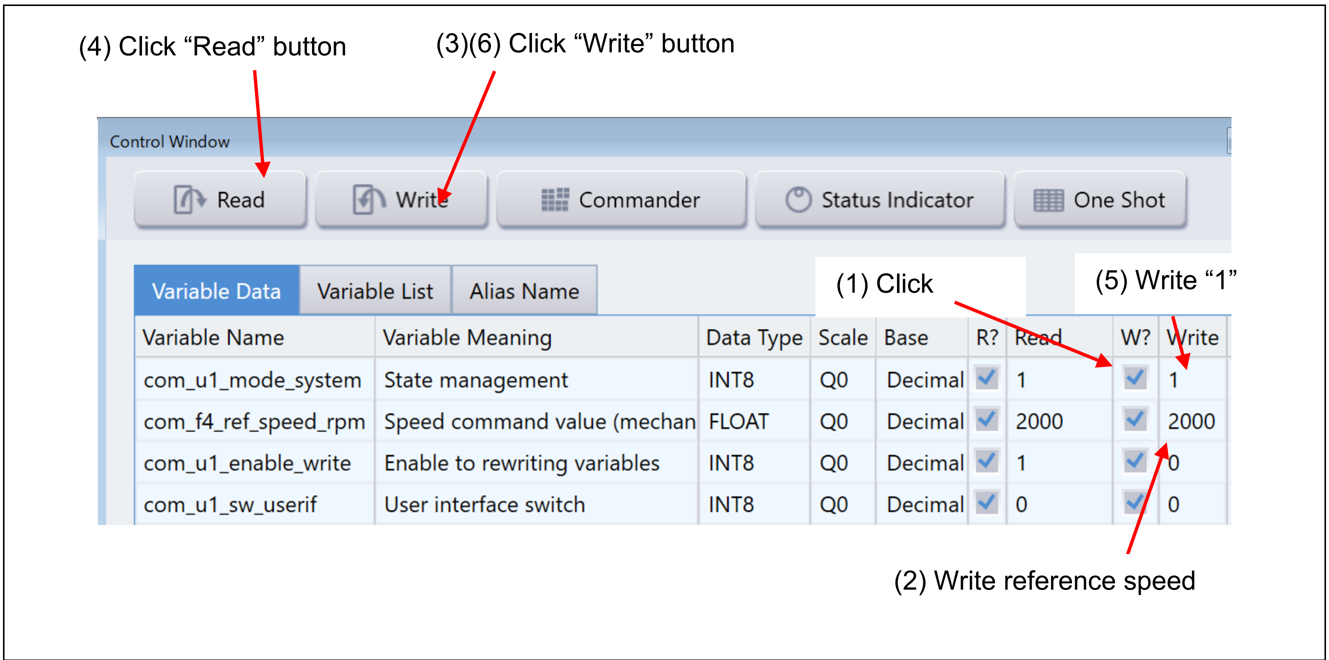


Figure 6-24 Procedure for driving of the motor

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.

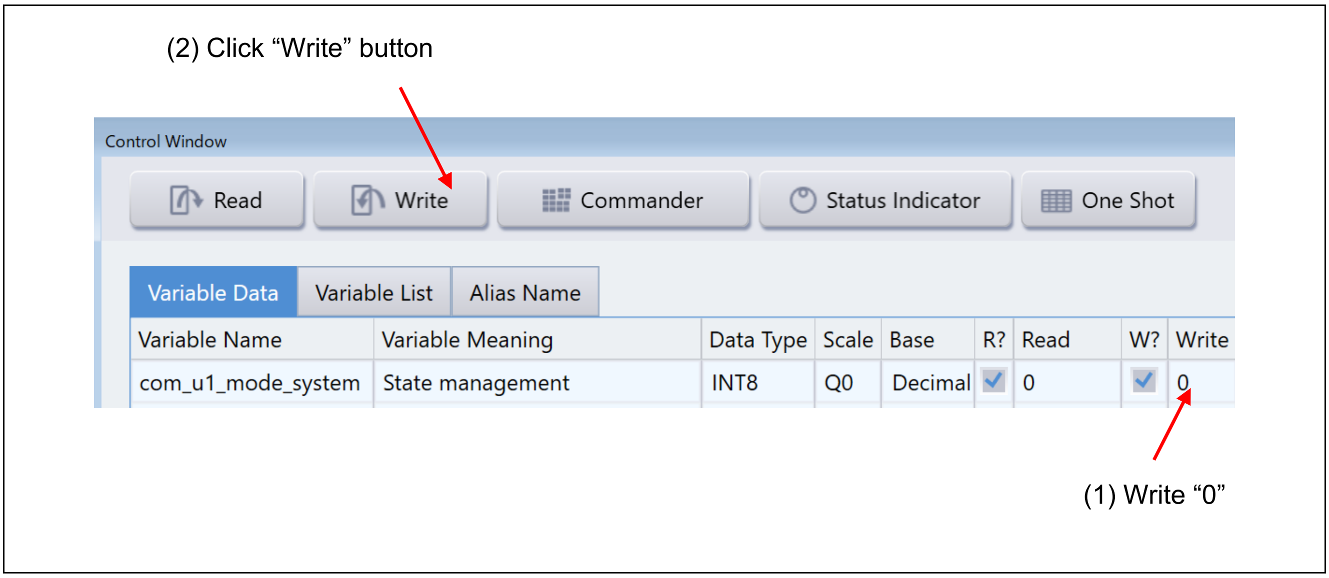


Figure 6-25 Procedure for stopping the motor

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.

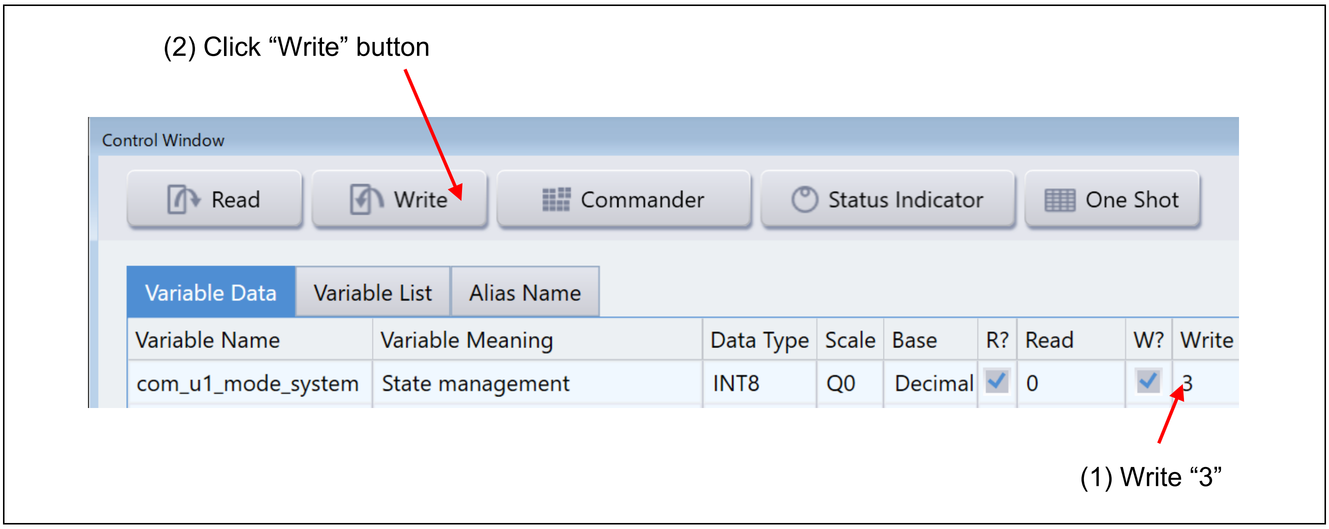


Figure 6-26 Procedure for handling an error

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	MOTOR_ERROR_HALL_TIMEOUT
0x0010	BEMF signal timeout error (Not happen in 120-degree hall)	MOTOR_ERROR_BEMF_TIMEOUT
0x0020	Unused	MOTOR_ERROR_HALL_PATTERN
0x0040	Hall signal pattern error	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 120-degree hall)	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	120-degree conducting control using hall sensors
Pulse Width Modulation (PWM) method	First 60 degree chopping
Position and speed detection method	Detected with hall sensors
Control mode	Only speed control

7.2 Detection of rotor angle with hall sensors

The Hall effect sensors are used to detect the position of the permanent magnet, and the signals from the Hall effect sensors are inputted to the microcontroller as position information.

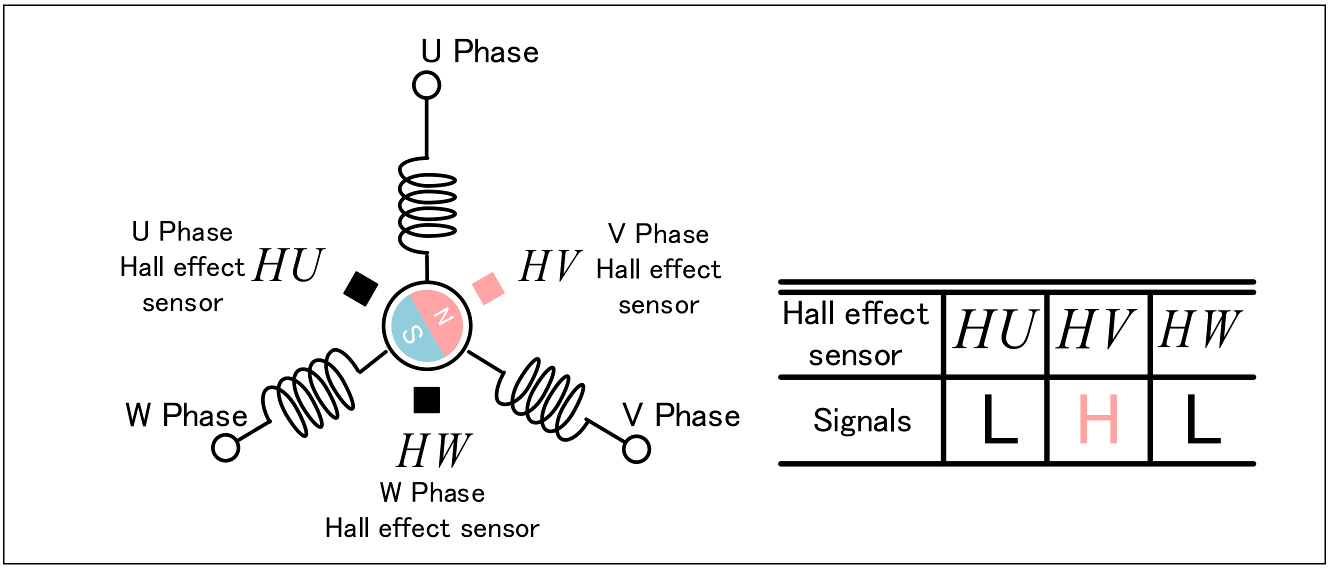


Figure 7-1 Example of Hall effect sensors (HU, HV, HW) position and signals

As shown in Figure 7-1, the Hall effect sensors are allocated every 120 degrees and the respective Hall effect sensor signals are switched depending on change in magnetic poles of the permanent magnet. Combining these signals of three Hall effect sensors enables to obtain position information every 60 degrees (six patterns for one cycle).
At the switching timing of Hall effect sensor signals, the conduction patterns of each phase are changed as shown in Figure 7-2.

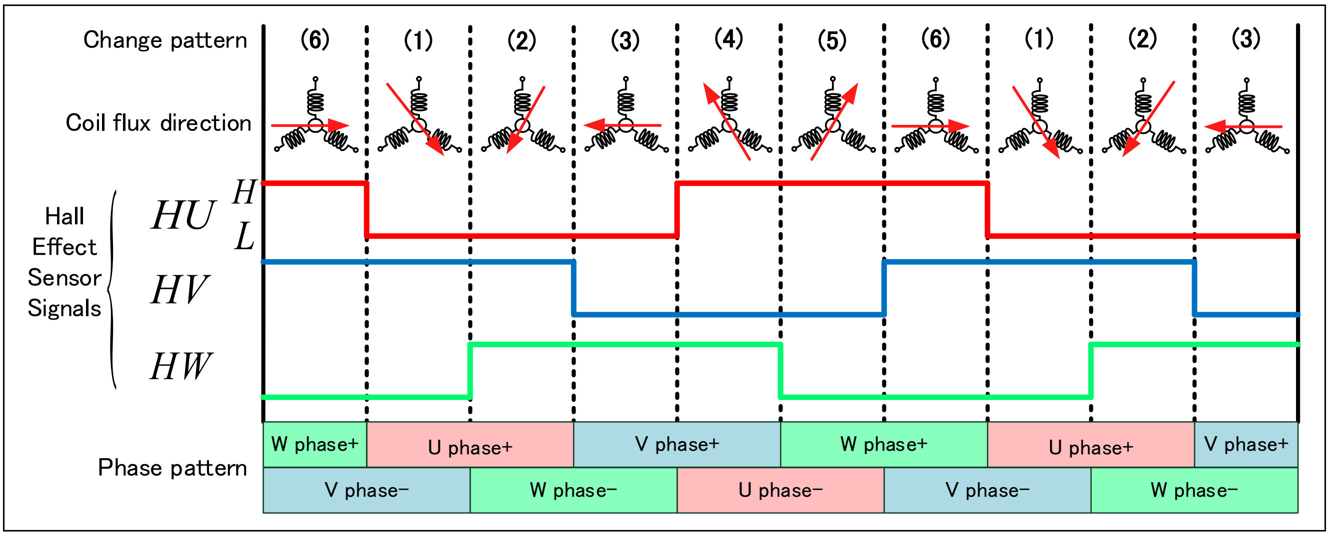


Figure 7-2 Relation between Hall effect sensor signals and conduction patterns

(Rotation direction: CW)

7.3 Speed control function

In this system, the motor rotation speed is calculated from a difference between the current timer value and the timer value 2π [rad] before. The timer values are obtained from free run timer at every hall sensor interrupt. And values which are obtained last 6 times are used as period of 2π..

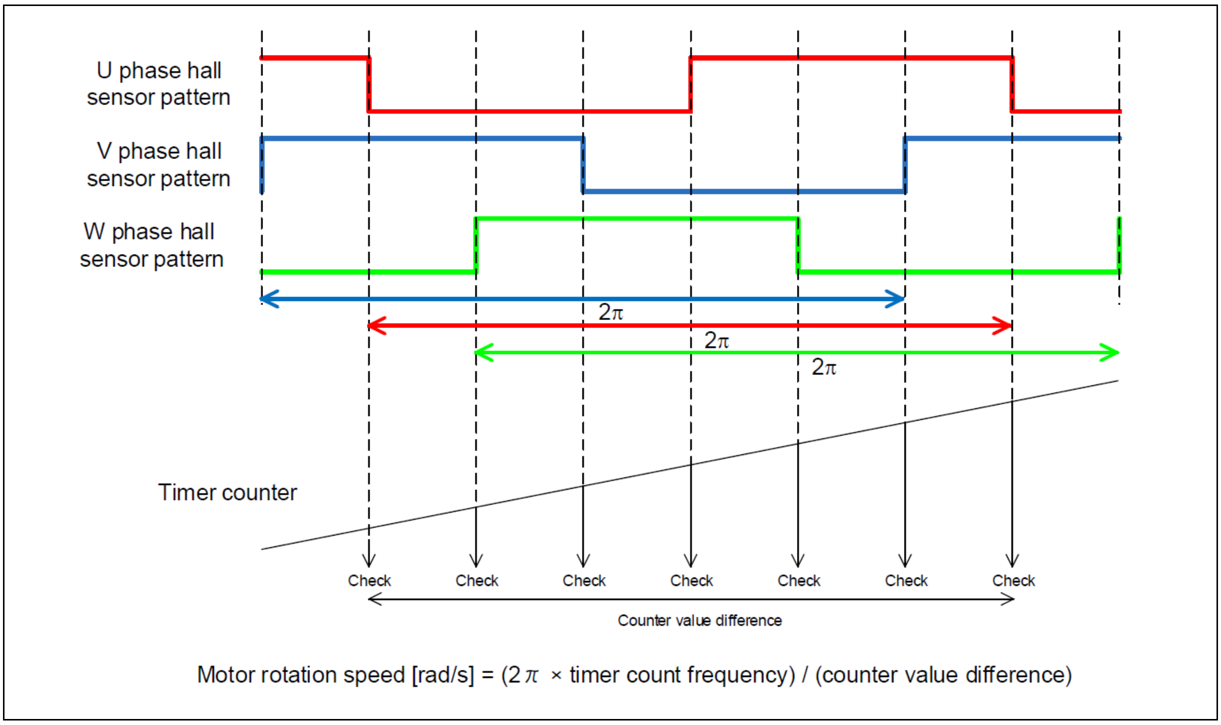


Figure 7-3 Method to calculate rotation speed

Speed control in the software targeted by this application note is performed by PI control. Obtain the voltage command value by the following speed control PI control.

$$v^{*} = (K_{P\omega} + \frac{K_{I\omega}}{s})(\omega^{*} - \omega)$$

v^*: Voltage command value, ω^*: Speed command value, ω: Rotation speed

$Kp\omega$: Speed PI proportional gain, $KI\omega$: Speed PI integral gain, s: Laplace operator

Please refer specialized books about detail of PI control.

7.4 Voltage control by PWM

PWM control is used for controlling output voltage. The PWM control is a control method that continuously adjusts the average voltage by varying the duty of pulse, as shown in.

$$\text{Duty} = \frac{T_{ON}}{T_{ON} + T_{OFF}} \times 100\ [\%]$$

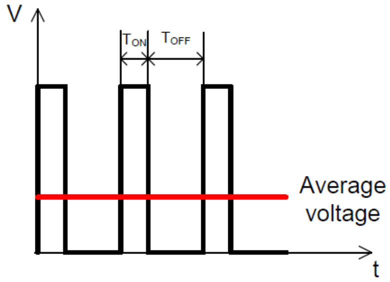


Figure 7-4 PWM control

Here, modulation factor m is defined as follows.

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

m: Modulation factor V: Command value voltage E: Inverter bus voltage

This modulation factor is reflected in the setting value of the register that determines the PWM duty.

In the target software of this application note, first-60-degree chopping is used to control the output voltage and speed. Figure 7-5 shows an example of motor control signal output waveforms at non-complimentary first 60-degree Chopping. Figure 7-6 shows an example of motor control signal output waveforms at Complimentary first 60-degree Chopping.

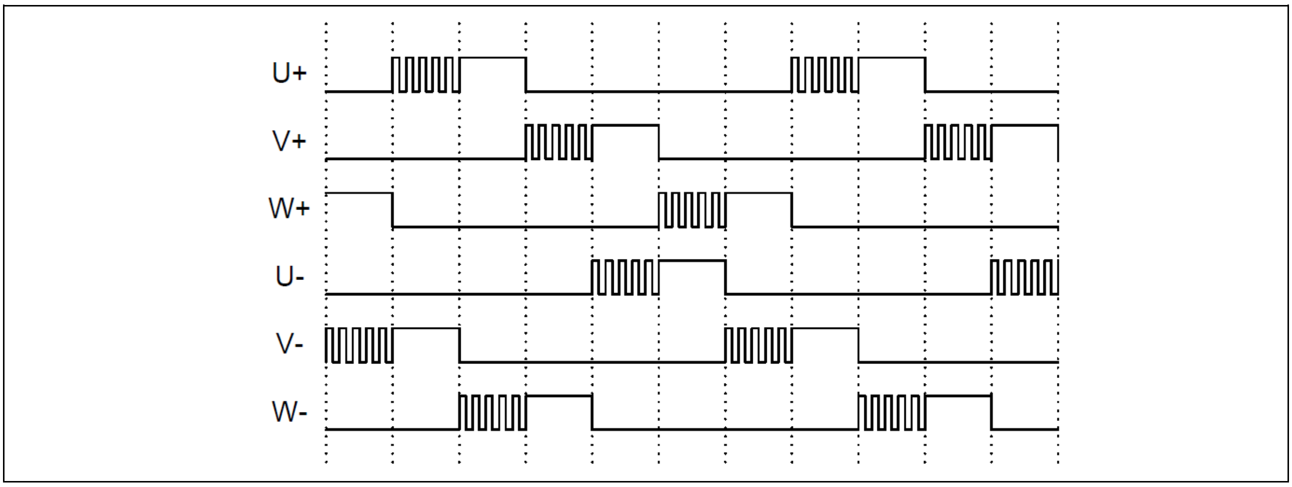


Figure 7-5 non-complimentary first 60-degree Chopping

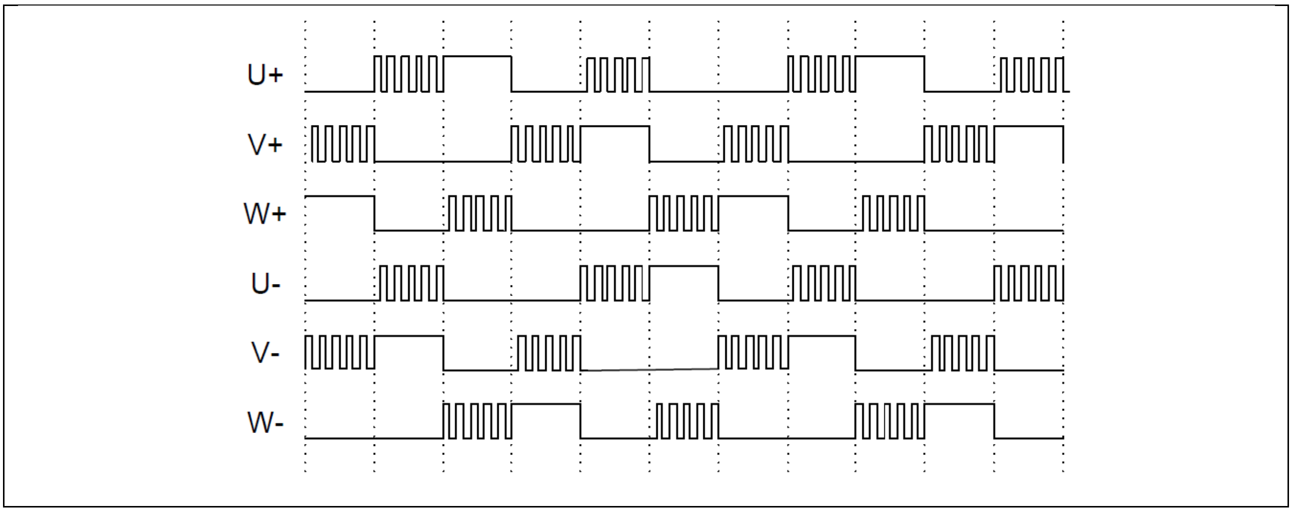


Figure 7-6 Complimentary first 60-degree Chopping

7.5 Start-up method

In the case of 120-degree conducting control using hall sensors, the rotor position can be determined by hall sensors' signals. Therefore, the conduction pattern at start-up is also determined.

When the control is changed to PI control, at least the motor needs to rotate one time (refer to 7.3). At start-up the motor is controlled in open loop with a constant voltage until the motor rotate one time.

Figure 7-7 shows the start-up method in this sample software. In "MOTOR_120_CONTROL_RUN_MODE_BOOT ", open loop with a constant voltage.

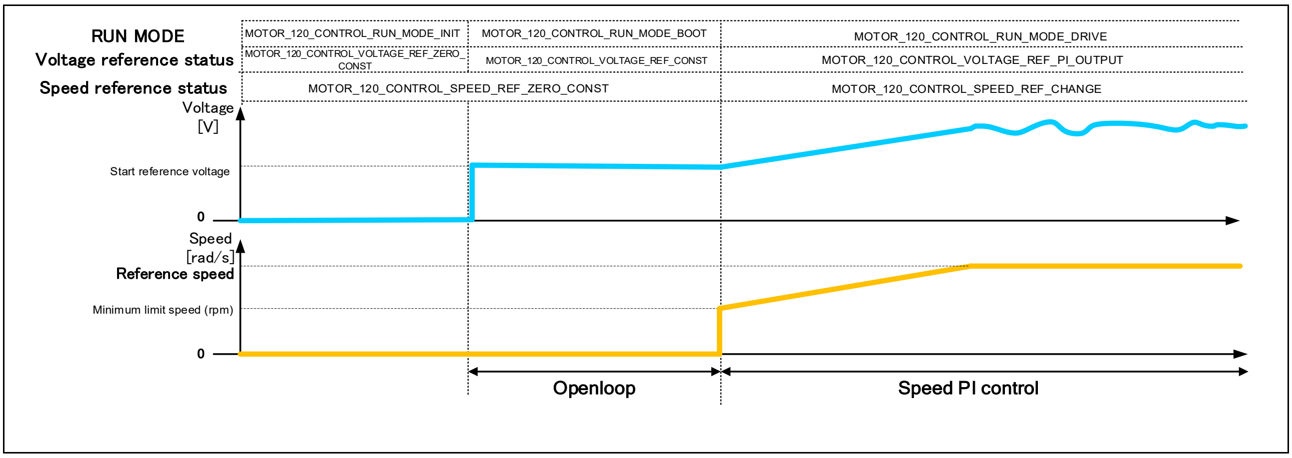


Figure 7-7 Start-up method (example)

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
Measure current of U-phase	PA04 / AN004	P000 / AN000	P000 / AN000
Measure current of V-phase	PA02 / AN002	P001 / AN001	P001 / AN001
Measure current of W-phase	PA00 / AN000	P002 / AN002	P002 / AN002
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
Hall sensor input (HU)	Ver.1: PC04 / IRQ10 Ver.2: PB02 / IRQ15-DS	P008 / IRQ12-DS	P008 / IRQ12-DS
Hall sensor input (HV)	Ver.1: PC05 / IRQ11 Ver.2: PC00 / IRQ11-DS	P006 / IRQ11-DS	P006 / IRQ11-DS
Hall sensor input (HW)	Ver.1: PB01 / IRQ1 Ver.2: PB10 / IRQ10-DS	P015 / IRQ13	P015 / IRQ13
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
Measure current of U-phase	P004 / AN000	P006 / AN006
Measure current of V-phase	P005 / AN001	P008 / AN008
Measure current of W-phase	P006 / AN002	P010 / AN010
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
Hall sensor input (HU)	P907 / IRQ10	P907 / IRQ10
Hall sensor input (HV)	P905 / IRQ8	P905 / IRQ8
Hall sensor input (HW)	P906 / IRQ9	P906 / IRQ9
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	Measure current of U-phase	AN004	AN000	AN000
	Measure current of V-phase	AN002	AN001	AN001
	Measure current of W-phase	AN000	AN002	AN002
	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
	Free run timer for measuring speed	-	AGT1	AGT1
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
	Free run timer for measuring speed	CH0	-	-
IRQ	U-phase Hall sensor signal edge detection	Ver.1: CH10 Ver.2: CH15-DS	CH12-DS	CH12-DS
	V-phase Hall sensor signal edge detection	Ver.1: CH11 Ver.2: CH11-DS	CH11-DS	CH11-DS
	W-phase Hall sensor signal edge detection	Ver.1: CH1 Ver.2: CH10-DS	CH13	CH13
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	Measure current of U-phase	AN000	AN006
	Measure current of V-phase	AN001	AN008
	Measure current of W-phase	AN002	AN010
	Measure inverter bus voltage	AN008	AN007
	Measure VR input	AN007	AN015
AGTW	Interval timer for speed control	AGT0	AGT0
	Free run timer for measuring speed	-	AGT1
GPT	PWM output of U-phase	CH5	CH8
	PWM output of V-phase	CH2	CH7
	PWM output of W-phase	CH3	CH9
	Free run timer for measuring speed	CH0	-
IRQ	U-phase Hall sensor signal edge detection	CH10	CH6
	V-phase Hall sensor signal edge detection	CH8	CH4
	W-phase Hall sensor signal edge detection	CH9	CH5
POEG	Emergency stop input of overcurrent	Group A	Group A

(1) A/D converter

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

A/D conversion is synchronized with underflow of GPT (trough of PWM). A/D conversion end interrupt is used as carrier period interrupt.

(2) General asynchronous timer (AGTW)

AGTW is used as cyclic interval timer to generate speed control period interrupt.
With RA6T3/RA4T1/RA8T2, AGTW is also used as a free run timer to calculate rotation speed.

(3) General purpose timer (GPT)

GPT outputs complementally PWM by its original mode with dead time.
With RA6T2/RA8T1, GPT is also used as a free run timer to calculate rotation speed.

(4) External interrupt (IRQ)

The hall sensors’ signals are inputted for detection of rotor position.
Both edge mode is used.
When the interrupt occurs, following operations are performed.

- detection of rotor position
- rotation speed measurement
- conduction pattern change

(5) Port output enable for GPT (POEG)

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

9. Software specifications and configuration

9.1 Software specifications

The following shows the basic software specifications of this system.

Table 9-1 Basic Specifications of 120-degree Hall Control Software

Item	Content	
Control method	120-degree conducting control	
Motor rotation start/stop	SW1 input or input from 'Renesas Motor Workbench'	
Position detection method	Position detection using hall sensors	
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)	20 [kHz] (Carrier period: 50 [μs])	
Dead time	2 [μs]	
Control period (speed)	1 [ms]	
Rotation speed control range	CW: 550 [rpm] to 2400 [rpm] CCW: 550 [rpm] to 2400 [rpm]	
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.

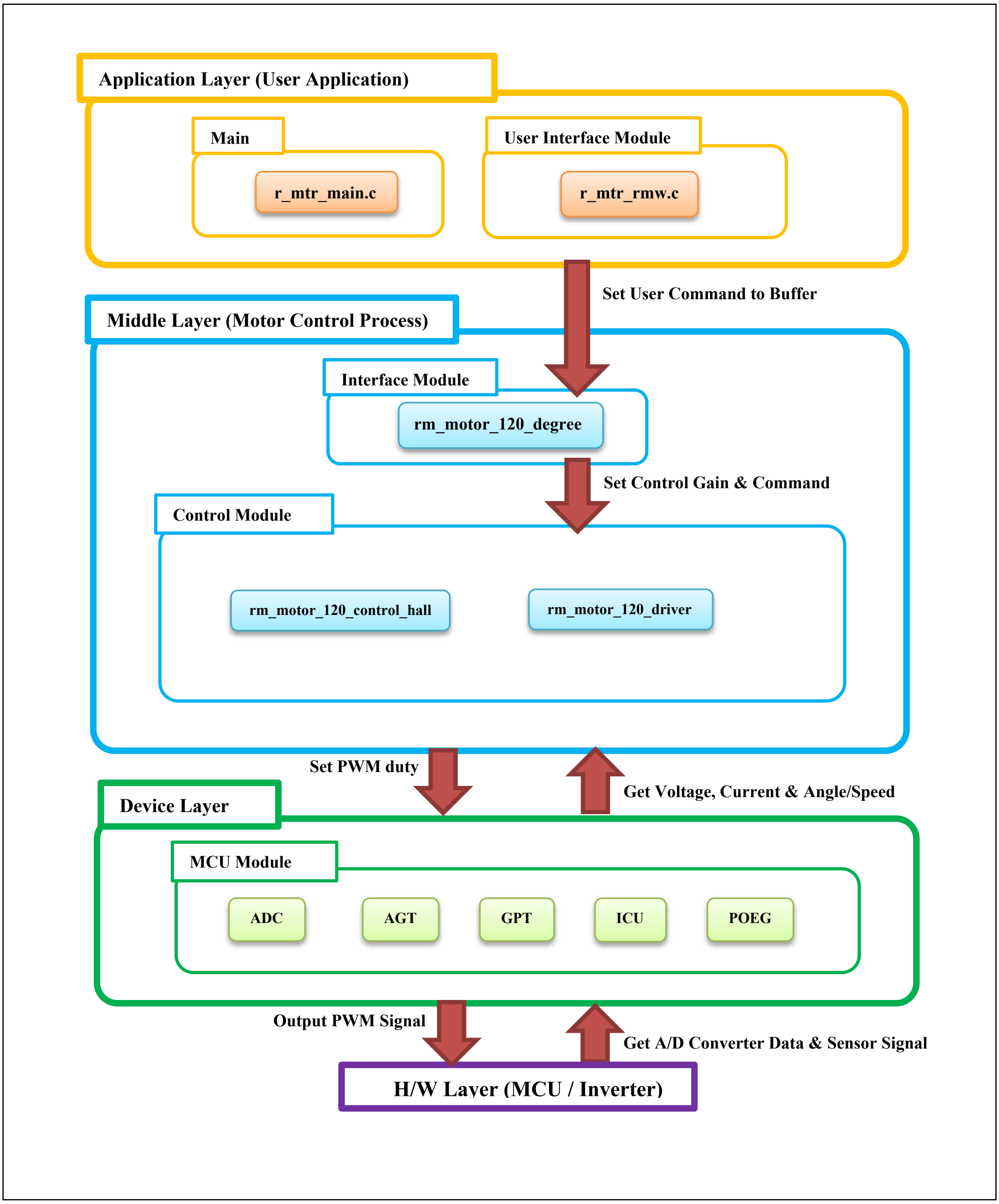


Figure 9-1 Module Configuration

9.3 Explanation about interrupt

In the sample program of this document, main processes are performed at speed control period interrupt and carrier 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		ADC0 ADI0(RA6T2, RA8T2) ADC0 SCAN END(RA6T3, RA4T1, RA8T1) Carrier period interrupt (A/D conversion end interrupt)
4		
3	Max	ICU IRQ11 , ICU IRQ10 , ICU IRQ1 (RA6T2) ICU IRQ12 , ICU IRQ11 , ICU IRQ13 (RA6T3, RA4T1) ICU IRQ10 , ICU IRQ8 , ICU IRQ9 (RA8T1, RA8T2) Hall signal interrupt
2		
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_external_irq_api.h	External IRQ 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_120_control_api.h	120-degree conducting control API definition
		rm_motor_120_driver_api.h	120-degree conducting control driver 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_120_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_icu.h	Function definition for external IRQ
		r_ioport.h	Function definition for I/O
		r_poeg.h	Function definition for POEG
		rm_motor_120_control_hall.h	Function definition for 120-degree hall control
		rm_motor_120_degree.h	Function definition for 120-degree motor
		rm_motor_120_driver.h	Function definition for 120-degree motor driver

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

Folder	Subfolder	File	Remarks
	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_icu/r_icu.c	External IRQ driver
		r_ioport/r_ioport.c	I/O driver
		r_poeg/r_poeg.c	POEG driver
		rm_motor_120_control_hall/rm_motor_120_control_hall.c	120-degree hall control driver
		rm_motor_120_degree/rm_motor_120_degree.c	120-degree driver
		rm_motor_120_driver/rm_motor_120_driver.c	120-degree motor driver
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
		r_mtr_rmw_display_cfg.h	Display configuration for RMW
	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 lists the variables that can be used by users in the application layer.

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

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

Table 9-8 List of variables

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 variable.)
g_u1_enable_write	A flag displays enable of value update
com_f4_ref_speed_rpm	Speed command value (mechanical) [rpm]
com_f4_overcurrent_limit	High current limit value [A]
com_f4_overvoltage_limit	High voltage limit value [V]
com_f4_overspeed_limit_rpm	Speed limit value (mechanical) [rpm]
com_f4_lowvoltage_limit	Low voltage limit value [V]
com_u4_timeout_cnt	Timeout count limit
com_f4_max_drive_v	Maximum command voltage [V]
com_f4_min_drive_v	Minimum command voltage [V]
com_f4_speed_lpf_k	Speed LPF parameter
com_f4_limit_speed_change	Command speed changing limit (mechanical) [rpm]
com_f4_start_refv	Command voltage at startup
com_f4_pi_ctrl_kp	Speed PI proportional gain
com_f4_pi_ctrl_ki	Speed PI Integral gain
com_f4_pi_ctrl_ilimit	Voltage PI control output limit value [V]

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

Structure	Description
g_user_motor_120_degree_extended_cfg	Structure of configuration for parameters about interface module which can be changed by user
g_user_motor_120_control_extended_cfg	Structure of configuration for parameters about 120-degree hall control module which can be changed by user
g_user_motor_120_driver_extended_cfg	Structure of configuration for parameters about 120-degree motor driver module which can be changed by user

9.5.4 Macro definition

Macro definitions are listed below.

Table 9-10 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
SPIKE_CNT	128	128	128
SPIKE_OC_RATE	0.9F	0.9F	0.9F
MTR_CW	0	0	0
MTR_CCW	1	1	1
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)
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 9-11 List of macros [2/2] (mtr_main.h)

Macro Name	RA8T1	R8T2
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
SPIKE_CNT	128	128
SPIKE_OC_RATE	0.9F	0.9F
MTR_CW	0	0
MTR_CCW	1	1
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)
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 9-12 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 9-13 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. This module controls data transfer between modules, total system mode of this project, and protection functions.

9.6.1 Functions

Table 9-14 lists the functions of interface module.

Table 9-14 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.
Speed information acquisition	Acquires the speed information from 120-degree hall control module.
Control module command value setting	Set speed reference to 120-degree hall control module via API
Interrupt processing	Perform user functions which are registered as a callback at each interrupts (speed and carrier).

9.6.2 Module configuration diagram

Figure 9-2 shows the module configuration.

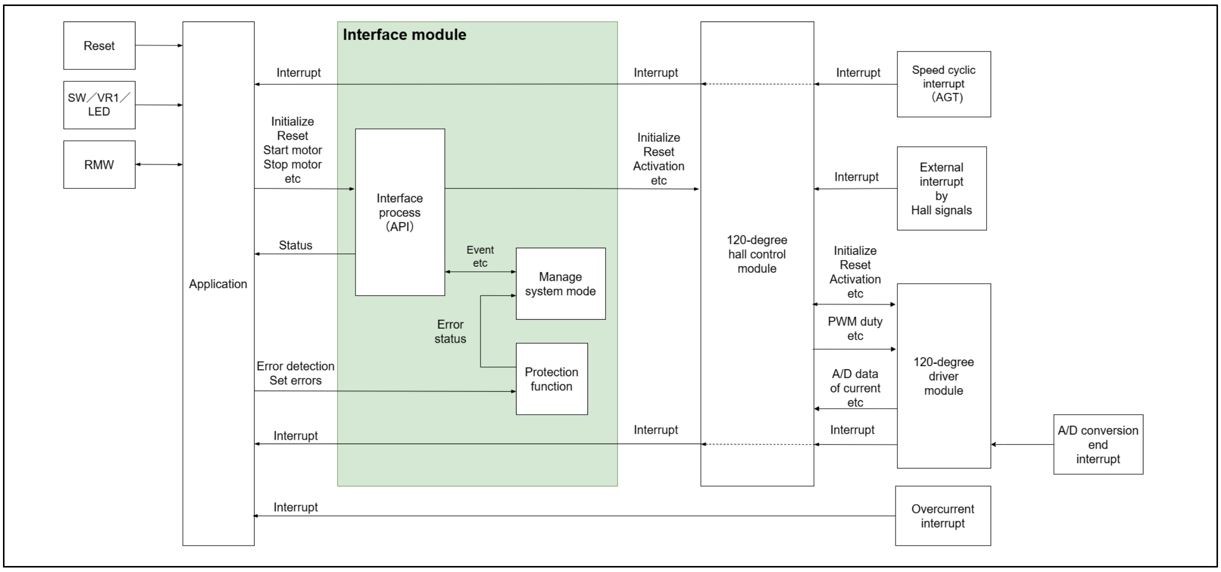
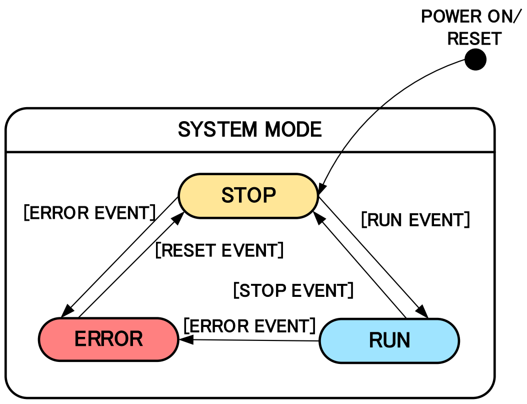


Figure 9-2 Module configuration diagram of interface module

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 9-15 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 9-16.

- 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 9-16 Operating conditions and setting values for the system protection functions

Category	Item	Value
Overcurrent error	Overcurrent limit value [A]	1.67
	Monitoring interval [μs]	Carrier interrupt interval*1
Overvoltage error	Overvoltage limit value [V]	60
	Monitoring interval [μs]	Carrier interrupt interval*1
Low-voltage error	Low-voltage limit value [V]	8
	Monitoring interval [μs]	Carrier interrupt interval*1
Rotation speed error	Speed limit value (mechanical) [rpm]	4500
	Monitoring interval [μs]	Carrier interrupt interval*1

*1 Refer to Table 9-1 Basic Specifications of 120-degree Hall Control Software.

9.6.5 API

Table 9-17 lists API functions of interface module.

Table 9-17 List of API functions

API	Description
RM_MOTOR_120_DEGREE_Open	Generates instances of this module and the modules to be used.
RM_MOTOR_120_DEGREE_Close	Close instances of this module and the modules to be used.
RM_MOTOR_120_DEGREE_Run	Run the motor.
RM_MOTOR_120_DEGREE_Stop	Stop the motor.
RM_MOTOR_120_DEGREE_Reset	Reset this module and the modules to be used.
RM_MOTOR_120_DEGREE_ErrorSet	Set error state.
RM_MOTOR_120_DEGREE_SpeedSet	Set speed reference (mechanical) [rpm]
RM_MOTOR_120_DEGREE_StatusGet	Get current state of the project.
RM_MOTOR_120_DEGREE_AngleGet	Get rotor angle [rad] (Unsupported in 120-degree hall)
RM_MOTOR_120_DEGREE_SpeedGet	Get rotation speed (mechanical) [rpm]
RM_MOTOR_120_DEGREE_ErrorCheck	Check error occurrence
RM_MOTOR_120_DEGREE_PositionSet	Set position reference [degree] (Unsupported in 120-degree hall)
RM_MOTOR_120_DEGREE_WaitStopFlagGet	Get a flag of waiting motor stop
RM_MOTOR_120_DEGREE_FunctionSelect	Select to use servo function (Unsupported in 120-degree hall)

9.6.6 Structure and variable information

The structures and variables for interface module are listed below.

Table 9-18 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 (No use in 120-degree hall)
	*p_motor_current_instance	Address of current control module instance (No use in 120-degree hall)
	*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 (mechanical) [rpm] reference
	*positionSet	Function address to set position reference (Unsupported in 120-degree hall)
	*statusGet	Function address to get moving status
	*angleGet	Function address to get rotor angle (Unsupported in 120-degree hall)
	*speedGet	Function address to get rotation speed (mechanical) [rpm]
	*waitStopFlagGet	Function address to get a flag for waiting “STOP”
	*errorCheck	Function address to check error occurrence
	*functionSelect	Function address to select servo function (Unsupported in 120-degree hall)
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 9-19 List of structures and variable for interface module (rm_motor_120-degree.h)

Structure	Members	Description
motor_120_degree_statemachine_t	status	Moving status
	status_next	Next moving status
	current_event	Current happened event
	u2_error_status	Error status
motor_120_degree_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_120_degree_instance_ctrl_t	open	Module opened information
	u2_error_info	Error status
	f_speed_rpm	Rotation speed (mechanical) [rpm]
	st_statem	Structure of state machine
	*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 9-20 List of macros for interface module

File name	Name of MACRO	Defined value	Description
rm_motor_120_degree.c	MOTOR_120_DEGREE_OPEN	0x4D314C53L	Module opened information
	MOTOR_120_DEGREE_STATEMACHINE_SIZE_STATE	3	Size of status of state machine
	MOTOR_120_DEGREE_STATEMACHINE_SIZE_EVENT	4	Size of event of state machine
	MOTOR_120_DEGREE_STATEMACHINE_ERROR_NONE	0x00	No error happened
	MOTOR_120_DEGREE_STATEMACHINE_ERROR_EVENTOUTBOUND	0x01	Error of outbound of event
	MOTOR_120_DEGREE_STATEMACHINE_ERROR_STATEOUTBOUND	0x02	Error of outbound of status
	MOTOR_120_DEGREE_STATEMACHINE_ERROR_ACTIONEXCEPTION	0x04	Error of exception about action

Table 9-21 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
	MOTOR_ERROR_BEMF_TIMEOUT	0x0010	Timeout error of BEMF signal detection (Not happen in 120-degree hall)
	MOTOR_ERROR_HALL_PATTERN	0x0020	Unused
	MOTOR_ERROR_BEMF_PATTERN	0x0040	Hall signal pattern error
	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 120-degree hall)
	MOTOR_ERROR_UNKNOWN	0xFFFF	Unknown error
motor_callback_event_t	MOTOR_CALLBACK_EVENT_SPEED_FORWARD	1	Callback event before cyclic speed control process (Not happen in 120-degree hall)
	MOTOR_CALLBACK_EVENT_SPEED_BACKWARD	2	Callback event after cyclic speed control process (Not happen in 120-degree hall)
	MOTOR_CALLBACK_EVENT_CURRENT_FORWARD	3	Callback event before cyclic current control process (Not happen in 120-degree hall)
	MOTOR_CALLBACK_EVENT_CURRENT_BACKWARD	4	Callback event after cyclic current control process (Not happen in 120-degree hall)
	MOTOR_CALLBACK_EVENT_ADC_FORWARD	5	Callback event before A/D conversion end interrupt process
	MOTOR_CALLBACK_EVENT_ADC_BACKWARD	6	Callback event after A/D conversion end interrupt process
	MOTOR_CALLBACK_EVENT_CYCLE_FORWARD	7	Callback event before cyclic speed control process
	MOTOR_CALLBACK_EVENT_CYCLE_BACKWARD	8	Callback event after cyclic speed control process
motor_wait_stop_flag_t	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 9-22 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 120-degree hall)	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 9-23 List of Enumeration type for interface module (rm_motor_120_degree.h)

Enumeration type name	Members	Defined value	Description
motor_120_degree_ctrl_status_t	MOTOR_120_DEGREE_CTRL_STATUS_STOP	0	Stop state
	MOTOR_120_DEGREE_CTRL_STATUS_RUN	1	Run (active) state
	MOTOR_120_DEGREE_CTRL_STATUS_ERROR	2	Error state
motor_120_degree_ctrl_event_t	MOTOR_120_DEGREE_CTRL_EVENT_STOP	0	Event to stop (inactive)
	MOTOR_120_DEGREE_CTRL_EVENT_RUN	1	Event to run (active)
	MOTOR_120_DEGREE_CTRL_EVENT_ERROR	2	Event of error
	MOTOR_120_DEGREE_CTRL_EVENT_RESET	3	Event to reset

9.7 120-degree hall Control Module

120-degree hall control module calculates speed PI with speed reference which is set by user and rotation speed. And it outputs drive voltage reference to 120-degree motor driver module.

9.7.1 Functions

Functions of 120-degree hall control module are listed below.

Table 9-24 Functions of 120-degree hall control module

Function	Description
120-degree conducting control	Set 120-degree chipping pattern according to the Hall sensor signal. Chopping pattern is selected from non-complimentary 60 degree chopping or complimentary 60 degree chopping.
Detection of rotation speed	Detect rotation speed with Hall sensor signal and free run timer counter.
Speed PI control	Calculate PWM duty with speed reference and estimated speed by PI control. Set the PWM duty to 120-degree motor driver module.

9.7.2 Module configuration diagram

Figure 9-4 shows the module configuration diagram.

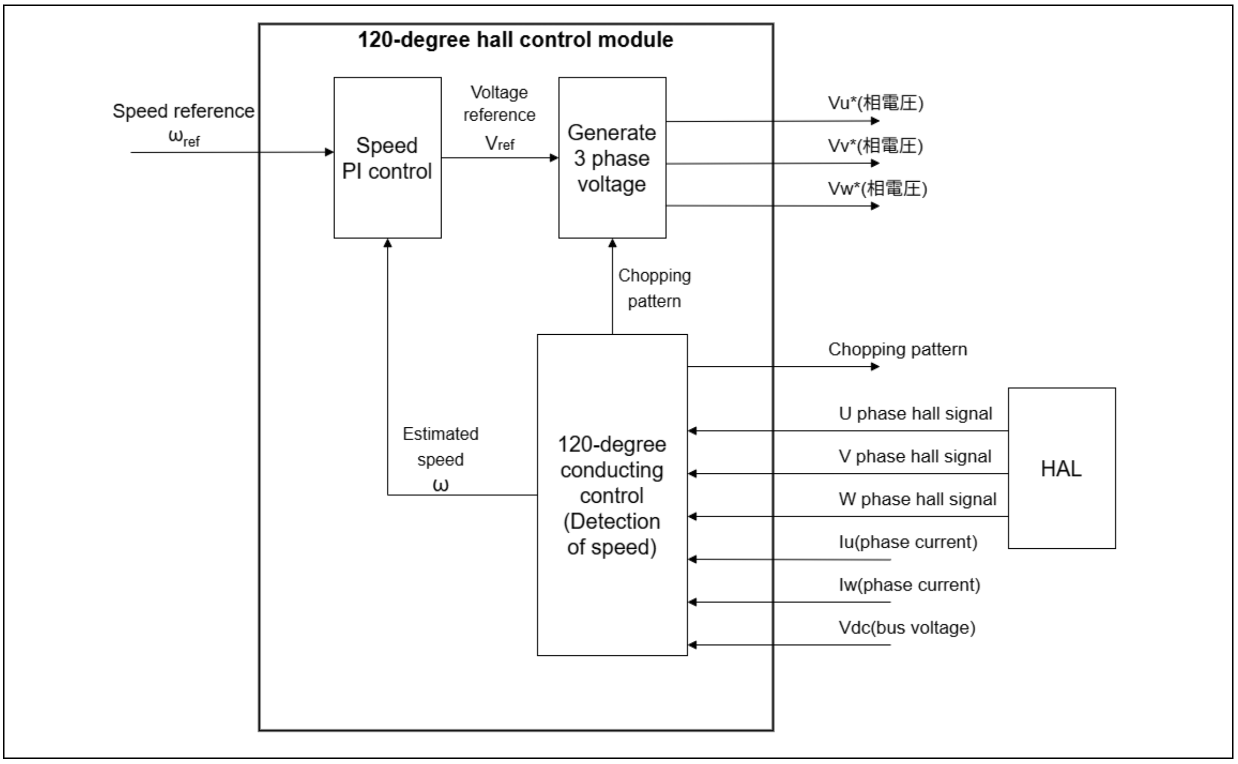


Figure 9-4 Module configuration diagram of 120-degree hall control module

9.7.3 API

Table 9-25 lists API functions of 120-degree hall control module.

Table 9-25 List of API functions of the 120-degree hall control module

API	Description
RM_MOTOR_120_CONTROL_HALL_Open	Generates instances of 120-degree hall control module and the modules to be used.
RM_MOTOR_120_CONTROL_HALL_Close	Close instances of 120-degree hall control module and the modules to be used.
RM_MOTOR_120_CONTROL_HALL_Run	Start motor rotation.
RM_MOTOR_120_CONTROL_HALL_Stop	Stop motor rotation.
RM_MOTOR_120_CONTROL_HALL_Reset	Reset this module and the modules to be used.
RM_MOTOR_120_CONTROL_HALL_SpeedSet	Set speed (mechanical) [rpm] reference.
RM_MOTOR_120_CONTROL_HALL_SpeedGet	Get estimated rotation speed (mechanical) [rpm].
RM_MOTOR_120_CONTROL_HALL_CurrentGet	Get detected phase current, inverter bus voltage.
RM_MOTOR_120_CONTROL_HALL_WaitStopFlagGet	Get the flag to wait motor stop.
RM_MOTOR_120_CONTROL_HALL_TimeoutErrorFlagGet	Get the flag of error occurrence about Hall time out.
RM_MOTOR_120_CONTROL_HALL_PatternErrorFlagGet	Get the flag of error occurrence about hall pattern
RM_MOTOR_120_CONTROL_HALL_VoltageRefGet	Get voltage reference.
RM_MOTOR_120_CONTROL_HALL_ParameterUpdate	Update the configuration parameters of this module.

9.7.4 Structure and variable information

Structures and variables of 120-degree hall control module are listed below.

Table 9-26 List of structures and variables of 120-degree control module [1/2] (rm_motor_120_control_api.h)

Structure	Members	Description
motor_120_control_callback_args_t	*p_context	Address of context data for callback function
	event	Event data of callback
motor_120_control_motor_parameter_t	u4_motor_pp	Pole pairs
	f4_motor_r	Resistance [ohm]
	f4_motor_ld	d-axis inductance [H]
	f4_motor_lq	q-axis inductance [H]
	f4_motor_m	Magnetic flux [Wb]
	f4_motor_j	Inertia [kgm^2]
motor_120_control_cfg_t	conduction_type	Selection of chopping pattern
	u4_timeout_cnt	Time out counts to detect BEMF pattern
	f4_max_drive_v	Maximum drive voltage [V]
	f4_min_drive_v	Minimum drive voltage [V]
	u4_speed_pi_decimation	Decimation number of speed control
	u4_free_run_timer_freq	Free run timer frequency for speed detection [MHz]
	f4_speed_lpf_k	Coefficient of speed LPF
	f4_limit_speed_change	Additional step of speed change (mechanical) [rpm]
	f4_pi_ctrl_kp	Proportional parameter of speed PI
	f4_pi_ctrl_ki	Integrated parameter of speed PI
	f4_pi_ctrl_ilimit	Limit of speed PI
	*p_callback	Address of callback function
	*p_context	Address of context data for callback function
	*p_extend	Address to refer extended configuration structure

Table 9-27 List of structures and variables of 120-degree control module [2/2] (rm_motor_120_control_api.h)

Structure	Members	Description
motor_120_control_api_t	*open	Function address to open module
	*close	Function address to close module
	*run	Function address to start motor rotation
	*stop	Function address to stop motor rotation
	*reset	Function address to reset module
	*speedSet	Function address to set speed (mechanical) [rpm] reference
	*speedGet	Function address to get estimated speed (mechanical) [rpm]
	*currentGet	Function address to get phase current and inverter bus voltage
	*waitStopFlagGet	Function address to get the flag to wait motor stop
	*timeoutErrorFlagGet	Function address to get the flag of occurrence of time out error
	*patternErrorFlagGet	Function address to get the flag of occurrence of BEMF pattern error
	*voltageRefGet	Function address to get voltage reference
	*parameterUpdate	Function address to update module parameters
motor_120_control_instance_t	*p_ctrl	Address of module variable structure
	*p_cfg	Address of module configuration structure
	*p_api	Address of API function structure

Table 9-28 List of structures and variables of 120-degree hall control module [1/2] (rm_motor_120_control_hall.h)

Structure	Members	Description
motor_120_control_hall_extended_cfg_t	port_hall_sensor_u	Port number of U phase hall signal
	port_hall_sensor_v	Port number of V phase hall signal
	port_hall_sensor_w	Port number of W phase hall signal
	f4_start_refv	Voltage reference at start up [V]
	u4_hall_wait_cnt	Counts to judge motor rotate at start up
	u4_stop_judge_time	Time to judge motor stop (counts)
	u4_min_speed_rpm	Minimum rotation speed (mechanical) [rpm]
	u4_hall_interrupt_mask_value	Mask value to avoid error signal in hall signal (counts)
	*p_motor_120_driver_instance	Address of 120-degree driver module instance
	*p_speed_cyclic_timer_instance	Address of timer instance for cyclic speed control
	*p_speed_calc_timer_instance	Address of timer instance for free run
	*p_u_hall_irq_instance	Address of external interrupt module instance for U phase
	*p_v_hall_irq_instance	Address of external interrupt module instance for V phase
	*p_w_hall_irq_instance	Address of external interrupt module instance for W phase
motor_120_control_hall_instance_ctrl_t	open	Module opened information
	active	System state (active/inactive)
	run_mode	Management of rotation mode
	timeout_error_flag	The flag for timeout error
	pattern_error_flag	The flag for pattern error

Table 9-29 List of structures and variables of 120-degree hall control module [2/2] (rm_motor_120_control_hall.h)

Structure	Members	Description
motor_120_control_hall_instance_ctrl_t	direction	Rotation direction
	f4_speed_calc_base	Base counts to calculate rotation speed
	f_rpm2rad	Translate for rpm=>rad/s
	f4_v_ref	Voltage reference [V]
	f4_ref_speed_rad	Speed reference (electrical) [rad/s]
	f4_ref_speed_rad_ctrl	Internal speed reference (electrical) [rad/s]
	f4_speed_rad	Rotation speed (electrical) [rad/s]
	u4_cnt_speed_pi	Counter for speed PI control
	flag_wait_stop	The flag to wait motor stop
	u4_cnt_wait_stop	Counter to wait motor stop
	v_pattern	Chopping pattern
	flag_speed_ref	State of speed reference
	flag_voltage_ref	State of voltage reference
	u4_cnt_timeout	Counter to judge timeout
	u4_hall_timer_cnt	Free run counts at hall interrupt
	u4_pre_hall_timer_cnt	Previous free run counts
	s4_timer_cnt_ave	Free run counts for 2 π
	u4_timer_cnt_buf	Buffer of free run counts to calculate 2 π
	u4_timer_cnt_num	Number to manage above buffer array
	f4_pi_ctrl_err	Error of speed PI control
	f4_pi_ctrl_refi	Integral value of speed PI control
	u4_hall_intr_cnt	Counter of hall interrupt at startup
	u4_adc_interrupt_cnt	Counter of AD end interrupt to avoid error of hall signal
	*p_cfg	Address of structure of module configuration to refer
	timer_direction	Timer direction of free run timer
	hall_interrupt_args	Arguments for hall interrupt callback
	timer_args	Arguments for timer interrupt callback

9.7.5 Macro and enumeration definition

The macros and enumerations for 120-degree hall control module are listed below.

Table 9-30 List of macros of 120-degree hall control module (rm_motor_120_control_hall.c)

File name	Name of MACRO	Defined value	Descriprion
rm_motor_120_control_hall.c	MOTOR_120_CONTROL_HALL_OPEN	('1' << 24U) | ('2' << 16U) | ('H' << 8U) | ('L' << 0U)	Module opened information
	MOTOR_120_CONTROL_HALL_TWOPI	2.0F * 3.1415926535F	2π
	MOTOR_120_CONTROL_HALL_TWOPI_DIV_60	MOTOR_120_CONTROL_HALL_TWOPI / 60.0F	2π/60 (To use translation from rpm to rad/s)
	MOTOR_120_CONTROL_HALL_HZ_TRANS	1000U	To use translation from kHz to Hz
	MOTOR_120_CONTROL_HALL_PATTERN_CW_V_U	2	Hall pattern V->U at clockwise
	MOTOR_120_CONTROL_HALL_PATTERN_CW_W_U	3	Hall pattern W->U at clockwise
	MOTOR_120_CONTROL_HALL_PATTERN_CW_W_V	1	Hall pattern W->V at clockwise
	MOTOR_120_CONTROL_HALL_PATTERN_CW_U_V	5	Hall pattern U->V at clockwise
	MOTOR_120_CONTROL_HALL_PATTERN_CW_U_W	4	Hall pattern U->W at clockwise
	MOTOR_120_CONTROL_HALL_PATTERN_CW_V_W	6	Hall pattern V->W at clockwise
	MOTOR_120_CONTROL_HALL_PATTERN_CCW_V_U	5	Hall pattern V->U at counterclockwise
	MOTOR_120_CONTROL_HALL_PATTERN_CCW_V_W	1	Hall pattern W->U at counterclockwise
	MOTOR_120_CONTROL_HALL_PATTERN_CCW_U_W	3	Hall pattern W->V at counterclockwise
	MOTOR_120_CONTROL_HALL_PATTERN_CCW_U_V	2	Hall pattern U->V at counterclockwise
	MOTOR_120_CONTROL_HALL_PATTERN_CCW_W_V	6	Hall pattern U->W at counterclockwise
	MOTOR_120_CONTROL_HALL_PATTERN_CCW_W_U	4	Hall pattern V->W at counterclockwise

Table 9-31 List of Enumeration type for 120-degree control module (rm_motor_120_control_api.h) [1/2]

Enumeration type name	Members	Defined value	Description
motor_120_control_event_t	MOTOR_120_CONTROL_EVENT_ADC_FORWARD	1	Event before carrier interrupt process
	MOTOR_120_CONTROL_EVENT_ADC_BACKWARD	2	Event after carrier interrupt process
	MOTOR_120_CONTROL_EVENT_CYCLE_FORWARD	3	Event before speed control process
	MOTOR_120_CONTROL_EVENT_CYCLE_BACKWARD	4	Event after speed control process
motor_120_conduction_type_t	MOTOR_120_CONDUCTION_TYPE_FIRST60	0	None-complementary 1st 60 degree chopping
	MOTOR_120_CONDUCTION_TYPE_COMPLEMENTARY	1	Complementary 1st 60 degree chopping
motor_120_control_status_t	MOTOR_120_CONTROL_STATUS_INACTIVE	0	Inactive
	MOTOR_120_CONTROL_STATUS_ACTIVE	1	Active
motor_120_control_run_mode_t	MOTOR_120_CONTROL_RUN_MODE_INIT	0	Initialize state
	MOTOR_120_CONTROL_RUN_MODE_BOOT	1	Boot state
	MOTOR_120_CONTROL_RUN_MODE_DRIVE	2	Drive state
motor_120_control_rotation_direction_t	MOTOR_120_CONTROL_ROTATION_DIRECTION_CW	0	Direction clockwise
	MOTOR_120_CONTROL_ROTATION_DIRECTION_CCW	1	Direction counterclockwise
	MOTOR_120_CONTROL_ROTATION_DIRECTION_MAX	2	Maximum of direction
motor_120_control_wait_stop_flag_t	MOTOR_120_CONTROL_WAIT_STOP_FLAG_CLEAR	0	Clear the flag to wait motor stop
	MOTOR_120_CONTROL_WAIT_STOP_FLAG_SET	1	Set the flag to wait motor stop
motor_120_control_timeout_error_flag_t	MOTOR_120_CONTROL_TIMEOUT_ERROR_FLAG_CLEAR	0	Clear the flag of time out error
	MOTOR_120_CONTROL_TIMEOUT_ERROR_FLAG_SET	1	Set the flag of time out error
motor_120_control_pattern_error_flag_t	MOTOR_120_CONTROL_PATTERN_ERROR_FLAG_CLEAR	0	Clear the flag of BEMF pattern error
	MOTOR_120_CONTROL_PATTERN_ERROR_FLAG_SET	1	Set the flag of BEMF pattern error

Table 9-32 List of Enumeration type for 120-degree control module (rm_motor_120_control_api.h) [2/2]

Enumeration type name	Members	Defined value	Description
motor_120_control_speed_ref_t	MOTOR_120_CONTROL_SPEED_REF_ZERO_CONST	0	State to set speed reference to zero
	MOTOR_120_CONTROL_SPEED_REF_OPENLOOP_1	1	State of speed reference at open loop #1
	MOTOR_120_CONTROL_SPEED_REF_OPENLOOP_2	2	State of speed reference at open loop #2
	MOTOR_120_CONTROL_SPEED_REF_OPENLOOP_3	3	State of speed reference at open loop #3
	MOTOR_120_CONTROL_SPEED_REF_CHANGE	4	State of speed reference at PI control
motor_120_control_voltage_ref_t	MOTOR_120_CONTROL_VOLTAGE_REF_ZERO_CONST	0	State to set voltage reference to zero
	MOTOR_120_CONTROL_VOLTAGE_REF_UP	1	State to increase voltage reference
	MOTOR_120_CONTROL_VOLTAGE_REF_CONST	2	State to set voltage reference constant
	MOTOR_120_CONTROL_VOLTAGE_REF_OPENLOOP	3	State of voltage reference at open loop
	MOTOR_120_CONTROL_VOLTAGE_REF_PI_OUTPUT	4	State of voltage reference at PI control

9.8 120-degree driver Module

The 120-degree driver module works as an interface between each motor modules and MCU peripherals. Appropriately configuring the 120-degree driver module allows you to use microcontroller function allocation and the differences of the board to be used without modifying the motor module.

9.8.1 Functions

Table 9-33 lists the functions of the 120-degree driver module.

Table 9-33 List of functions of the 120-degree 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.
Offset adjustment of current and voltage	Calculate A/D offset at each phase current and voltage detection
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.

9.8.2 Module configuration diagram

The module configuration of 120-degree driver module is shown below.

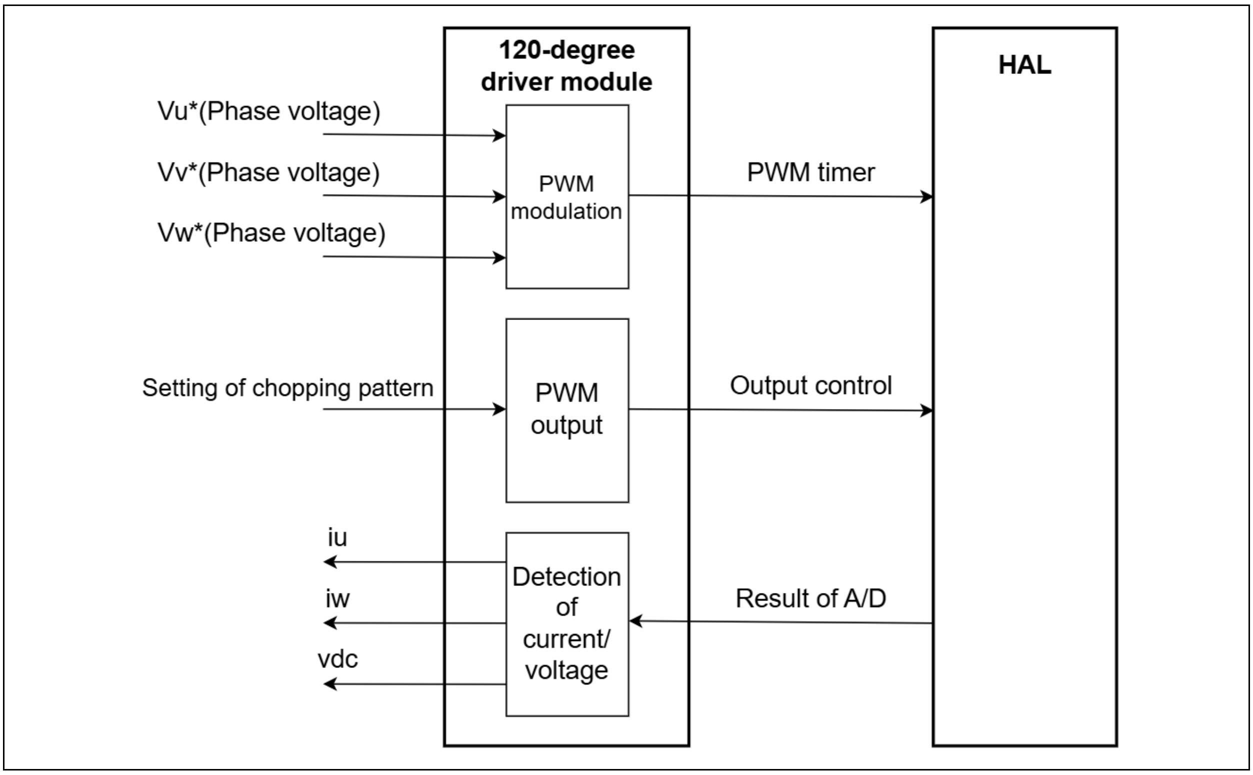


Figure 9-5 Module configuration diagram of 120-degree driver module

9.8.3 API

Table 9-34 lists and describes the API functions for 120-degree driver module.

Table 9-34 List of API functions for the 120-degree driver module

API	Description
RM_MOTOR_120_DRIVER_Open	Generate instances of 120-degree driver module and the modules to be used.
RM_MOTOR_120_DRIVER_Close	Close instances of 120-degree driver module and the modules to be used.
RM_MOTOR_120_DRIVER_Run	Active the module (Start motor rotation)
RM_MOTOR_120_DRIVER_Stop	Inactive the module (Stop motor rotation)
RM_MOTOR_120_DRIVER_Reset	Reset the module and the modules to be used.
RM_MOTOR_120_DRIVER_PhaseVoltageSet	Set phase voltage reference
RM_MOTOR_120_DRIVER_PhasePatternSet	Set phase chopping pattern
RM_MOTOR_120_DRIVER_CurrentGet	Get A/D converted data of phase current and inverter bus voltage
RM_MOTOR_120_DRIVER_CurrentOffsetCalc	Calculate A/D offset
RM_MOTOR_120_DRIVER_FlagCurrentOffsetGet	Get the flag of A/D offset state (active/finish)
RM_MOTOR_120_DRIVER_ParameterUpdate	Update configuration parameters of the module

9.8.4 Structure and variable information

Structures and variables of 120-degree driver module are listed below.

Table 9-35 List of structures and variables for 120-degree driver module (rm_motor_120_driver_api.h)

Structure	Members	Description
motor_120_driver_callback_args_t	event	Event data of callback
	*p_context	Address of context data for callback function
motor_120_driver_current_status_t	iu	U phase electrical current [A]
	iv	V phase electrical current [A]
	iw	W phase electrical current [A]
	vdc	Inverter bus voltage [V]
	vu	U phase voltage [V] (No use in 120-degree hall)
	vv	V phase voltage [V] (No use in 120-degree hall)
	vw	W phase voltage [V] (No use in 120-degree hall)
motor_120_driver_cfg_t	*p_callback	Address of set callback function
	*p_context	Address of context data for callback function
	*p_extend	Address to refer the extended configuration structure
motor_120_driver_api_t	*open	Address of open function
	*close	Address of close function
	*run	Address of function to active the module (start motor rotation)
	*stop	Address of function to inactive the module (stop motor rotation)
	*reset	Address of reset function
	*phaseVoltageSet	Address of function to set phase voltage reference
	*phasePatternSet	Address of function to set chopping pattern
	*currentGet	Address of function to get phase current and inverter bus voltage
	*currentOffsetCalc	Address of function to perform calculation of A/D offset
	*flagCurrentOffsetGet	Address of function to get the flag of A/D offset
	*parameterUpdate	Address of function to update configuration parameters.
motor_120_driver_instance_t	*p_ctrl	Address of structure of module variables
	*p_cfg	Address of structure of module configuration
	*p_api	Address of structure of API function address

Table 9-36 List of structures and variables for 120-degree driver module [1/3] (rm_motor_120_driver.h)

Structure	Members	Description
motor_120_driver_shared_instance_ctrl_t	open	Module opened information
	registered_motor_count	Registered motor counts
	*p_context	Address of context to be referred
motor_120_driver_extended_shared_cfg_t	*p_adc_instance_1st	Address of ADC instance #1
	*p_adc_instance_2nd	Address of ADC instance #2
	*p_shared_instance_ctrl	Address of structure of shared module variables
motor_120_driver_modulation_t	f4_vdc	Inverter bus voltage
	f4_max_duty	Maximum of PWM duty
	f4_min_duty	Minimum of PWM duty
	f4_neutral_duty	Value of PWM to output neutral
motor_120_driver_extended_cfg_t	*p_adc_instance	Address of ADC instance
	*p_three_phase_instance	Address of 3phase PWM instance
	motor_120_type	120 sensorless/hall type
	iu_ad_ch	A/D channel for U phase current
	iw_ad_ch	A/D channel for W phase current
	vdc_ad_ch	A/D channel for inverter bus voltage
	vu_ad_ch	A/D channel for U phase voltage (No use in 120-degree hall)
	vv_ad_ch	A/D channel for V phase voltage (No use in 120-degree hall)
	vw_ad_ch	A/D channel for W phase voltage (No use in 120-degree hall)
	iu_ad_unit	A/D unit number for U phase current
	iw_ad_unit	A/D unit number for W phase current
	vdc_ad_unit	A/D unit number for inverter bus voltage
	vu_ad_unit	A/D unit number for U phase voltage (No use in 120-degree hall)
	vv_ad_unit	A/D unit number for V phase voltage (No use in 120-degree hall)
	vw_ad_unit	A/D unit number for W phase voltage (No use in 120-degree hall)
	port_up	Port number for U phase upper side
	port_un	Port number for U phase lower side
	port_vp	Port number for V phase upper side
	port_vn	Port number for V phase lower side
	port_wp	Port number for W phase upper side
	port_wn	Port number for W phase lower side

Table 9-37 List of structures and variables for 120-degree driver module [2/3] (rm_motor_120_driver.h)

Structure	Members	Description
motor_120_driver_extended_cfg_t	u4_pwm_timer_freq	PWM timer frequency [MHz]
	pwm_carrier_freq	PWM carrier frequency [kHz]
	u4_deadtime	Value of dead time (counter value)
	f_current_range	Maximum to detect electrical current [A]
	f_vdc_range	Maximum to detect inverter bus voltage [V]
	f_ad_resolution	Resolution of A/D conversion
	f_ad_current_offset	Center value of A/D for current detection
	f_ad_voltage_conversion	Value to convert A/D data
	u4_offset_calc_count	Counts to measure A/D offset
	mod_param	Structure for modulation
	interrupt_adc	A/D unit number which occurs A/D end interrupt
	*p_shared_cfg	Address of A/D shared module configuration
motor_120_driver_instance_ctrl_t	open	Module opened information
	u1_active	Status of active/inactive of the module
	u4_carrier_base	Base counts to calculate PWM duty
	u4_deadtime_count	Couts to calculate dead time
	f_iu_ad	Detected U phase electrical current [A]
	f_iw_ad	Detected V phase electrical current [A]
	f_vdc_ad	Detected inverter bus voltage [V]
	f_refu	U phase voltage reference [V]
	f_refv	V phase voltage reference [V]
	f_refw	W phase voltage reference [V]
	f_vu_ad	Detected U phase voltage [V] (No use in 120-degree hall)
	f_vv_ad	Detected V phase voltage [V] (No use in 120-degree hall)
	f_vw_ad	Detected W phase voltage [V] (No use in 120-degree hall)
	u1_flag_offset_calc	The flag of state of calculate A/D offset
	u4_offset_calc_times	Counter to measure A/D offset

Table 9-38 List of structures and variables for 120-degree driver module [3/3] (rm_motor_120_driver.h)

Structure	Members	Description
motor_120_driver_instance_ctrl_t	f_offset_iu	A/D offset for U phase current
	f_offset_iw	A/D offset for W phase current
	f_sum_iu_ad	Summation of A/D offset for U phase current
	f_sum_iw_ad	Summation of A/D offset for V phase current
	f_offset_vu	A/D offset for U phase voltage (No use in 120-degree hall)
	f_offset_vv	A/D offset for V phase voltage (No use in 120-degree hall)
	f_offset_vw	A/D offset for W phase voltage (No use in 120-degree hall)
	f_offset_off_vu	A/D offset for U phase voltage at PWM is off (No use in 120-degree hall)
	f_offset_off_vv	A/D offset for V phase voltage at PWM is off (No use in 120-degree hall)
	f_offset_off_vw	A/D offset for W phase voltage at PWM is off (No use in 120-degree hall)
	f_sum_vu_ad	Summation of A/D offset for U phase voltage (No use in 120-degree hall)
	f_sum_vv_ad	Summation of A/D offset for V phase voltage (No use in 120-degree hall)
	f_sum_vw_ad	Summation of A/D offset for W phase voltage (No use in 120-degree hall)
	u4_gtioca_general_low_cfg	Setting data of GTIOCA as general port to output low signal
	u4_gtioca_general_high_cfg	Setting data of GTIOCA as general port to output high signal
	u4_gtioca_periheral_low_cfg	Setting data of GTIOCA as peripheral port to output low signal
	u4_gtioca_periheral_high_cfg	Setting data of GTIOCA as peripheral port to output high signal
	u4_gtiocb_general_low_cfg	Setting data of GTIOCB as general port to output low signal
	u4_gtiocb_general_high_cfg	Setting data of GTIOCB as general port to output high signal
	u4_gtiocb_periheral_low_cfg	Setting data of GTIOCB as peripheral port to output low signal
	u4_gtiocb_periheral_high_cfg	Setting data of GTIOCB as peripheral port to output high signal
	st_modulation	Structure of modulation
	*p_cfg	Address to refer module configuration
	adc_callback_args	Callback argument for ADC callback
	timer_callback_args	Callback argument for timer callback
	*p_shared_ctrl	Address of structure of A/D shared module variables

9.8.5 Macro and enumeration definition

The macros and enumerations for 120-degree driver module are listed below.

Table 9-39 List of macros of 120-degree driver module

<table>
<tr><th>File name</th><th>Name of MACRO</th><th>Defined value</th><th>Description</th></tr>
<tr><td rowspan="8">rm_motor_120_driver.c</td><td>MOTOR_120_DRIVER_OPEN</td><td>('M' << 24U) | ('1' << 16U) | ('D' << 8U) | ('R' << 0U)</td><td>Module opened information</td></tr>
<tr><td>MOTOR_120_DRIVER_SHARED_ADC_OPEN</td><td>('M' << 24U) | ('1' << 16U) | ('S' << 8U) | ('A' << 0U)</td><td>Module opened information of shared module</td></tr>
<tr><td>MOTOR_120_DRIVER_DEV_HALF</td><td>0.5F</td><td>0.5</td></tr>
<tr><td>MOTOR_120_DRIVER_KHZ_TRANS</td><td>1000U</td><td>To transform kHz => Hz</td></tr>
<tr><td>MOTOR_120_DRIVER_GENERAL_IO_PORT_L</td><td>0x3000004</td><td>Value to set a port to general low output</td></tr>
<tr><td>MOTOR_120_DRIVER_GENERAL_IO_PORT_H</td><td>0x3000005</td><td>Value to set a port to general high output</td></tr>
<tr><td>MOTOR_120_DRIVER_PERIPHERAL_IO_PORT_L</td><td>0x3010004</td><td>Value to set a port to peripheral low output</td></tr>
<tr><td>MOTOR_120_DRIVER_PERIPHERAL_IO_PORT_H</td><td>0x3010005</td><td>Value to set a port to peripheral high output</td></tr>
</table>

Table 9-40 List of Enumeration type for 120-degree driver module [1/2] (rm_motor_120_driver_api.h)

Enumeration	Members	Defined value	Description
motor_120_driver_event_t	MOTOR_120_DRIVER_EVENT_FORWARD	0	Callback event before 120-degree driver module process (A/D conversion end interrupt)
	MOTOR_120_DRIVER_EVENT_120_CONTROL	1	Callback event for 120-degre driver module process (A/D conversion end interrupt)
	MOTOR_120_DRIVER_EVENT_BACKWARD	2	Callback event after 120-degree driver module process (A/D conversion end interrupt)
motor_120_driver_flag_offset_calc_t	MOTOR_120_DRIVER_FLAG_OFFSET_CALC_CLEAR	0	No get A/D offset
	MOTOR_120_DRIVER_FLAG_OFFSET_CALC_OFF_FINISH	1	A/D offset is gotten at port off
	MOTOR_120_DRIVER_FLAG_OFFSET_CALC_ALL_FINISH	2	Finish to get A/D offset
motor_120_driver_phase_pattern_t	MOTOR_120_DRIVER_PHASE_PATTERN_ERROR	0	Error of 3 phase chopping pattern
	MOTOR_120_DRIVER_PHASE_PATTERN_UP_PWM_VN_ON	1	U phase upper side : PWM V phase lower side : ON
	MOTOR_120_DRIVER_PHASE_PATTERN_UP_PWM_WN_ON	2	U phase upper side : PWM W phase lower side : ON
	MOTOR_120_DRIVER_PHASE_PATTERN_VP_PWM_UN_ON	3	V phase upper side : PWM U phase lower side : ON
	MOTOR_120_DRIVER_PHASE_PATTERN_VP_PWM_WN_ON	4	V phase upper side : PWM W phase lower side : ON
	MOTOR_120_DRIVER_PHASE_PATTERN_WP_PWM_UN_ON	5	W phase upper side : PWM U phase lower side : ON
	MOTOR_120_DRIVER_PHASE_PATTERN_WP_PWM_VN_ON	6	W phase upper side : PWM V phase lower side : ON
	MOTOR_120_DRIVER_PHASE_PATTERN_UP_ON_VN_PWM	7	U phase upper side : ON V phase lower side : PWM
	MOTOR_120_DRIVER_PHASE_PATTERN_UP_ON_WN_PWM	8	U phase upper side : ON W phase lower side : PWM
	MOTOR_120_DRIVER_PHASE_PATTERN_VP_ON_UN_PWM	9	U phase upper side : ON V phase lower side : PWM
	MOTOR_120_DRIVER_PHASE_PATTERN_VP_ON_WN_PWM	10	U phase upper side : ON V phase lower side : PWM
	MOTOR_120_DRIVER_PHASE_PATTERN_WP_ON_UN_PWM	11	U phase upper side : ON V phase lower side : PWM
	MOTOR_120_DRIVER_PHASE_PATTERN_WP_ON_VN_PWM	12	W phase upper side : ON V phase lower side : PWM
	MOTOR_120_DRIVER_PHASE_PATTERN_U_PWM_VN_ON	13	U phase upper side : ON V phase lower side : PWM
	MOTOR_120_DRIVER_PHASE_PATTERN_U_PWM_WN_ON	14	U phase : PWM W phase lower side : ON
	MOTOR_120_DRIVER_PHASE_PATTERN_V_PWM_UN_ON	15	V phase : PWM U phase lower side : ON
	MOTOR_120_DRIVER_PHASE_PATTERN_V_PWM_WN_ON	16	V phase : PWM W phase lower side : ON

Table 9-41 List of Enumeration type for 120-degree driver module [2/2] (rm_motor_120_driver_api.h)

Enumeration	Members	Defined value	Description
motor_120_driver_phase_pattern_t	MOTOR_120_DRIVER_PHASE_PATTERN_W_PWM_UN_ON	17	W phase : PWM U phase lower side : ON
	MOTOR_120_DRIVER_PHASE_PATTERN_W_PWM_VN_ON	18	W phase : PWM V phase lower side : ON
	MOTOR_120_DRIVER_PHASE_PATTERN_UP_ON_V_PWM	19	U phase upper side : ON V phase : PWM
	MOTOR_120_DRIVER_PHASE_PATTERN_UP_ON_W_PWM	20	U phase upper side : ON W phase : PWM
	MOTOR_120_DRIVER_PHASE_PATTERN_VP_ON_U_PWM	21	V phase upper side : ON U phase : PWM
	MOTOR_120_DRIVER_PHASE_PATTERN_VP_ON_W_PWM	22	V phase upper side : ON W phase : PWM
	MOTOR_120_DRIVER_PHASE_PATTERN_WP_ON_U_PWM	23	W phase upper side : ON U phase : PWM
	MOTOR_120_DRIVER_PHASE_PATTERN_WP_ON_V_PWM	24	W phase upper side : ON V phase : PWM

Table 9-42 List of Enumeration type for 120-degree driver module (rm_motor_120_driver.h)

Enumeration	Members	Defined value	Description
motor_120_driver_select_adc_instance_t	MOTOR_120_DRIVER_SELECT_ADC_INSTANCE_1ST	0	1st ADC module instance
	MOTOR_120_DRIVER_SELECT_ADC_INSTANCE_2ND	1	2nd ADC module instance
motor_120_driver_status_t	MOTOR_120_DRIVER_STATUS_INACTIVE	0	Inactive state (Motor stops)
	MOTOR_120_DRIVER_STATUS_ACTIVE	1	Active state (Motor rotates)
motor_120_driver_type_t	MOTOR_120_DRIVER_TYPE_SENSORLESS	0	120-degree sensorless
	MOTOR_120_DRIVER_TYPE_HALL	1	120-degree hall

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_120_degree)

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_120_degree)

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_120_degree)

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 120-degree control module parameters

The changeable parameters that are used in 120-degree control module and these initial values are listed below.

Table 10-4 Configuration Options (rm_motor_120_control_hall)

Options	Description
General | Conduction type	Type of chopping pattern
General | Timeout counts (msec)	Couts to judge timeout [ms]
General | Maximum voltage (V)	Maximum voltage [V]
General | Minimum voltage (V)	Minimum voltage [V]
General | Speed PI decimation	Decimation value of speed process interrupt
General | Free run timer frequency (MHz)	Frequency of free run timer [MHz]
General | Speed LPF	Coefficient of speed LPF
General | Step of speed reference change	Changing step of speed reference
General | Start reference voltage (V)	Voltage reference at start up [V]
General | Hall wait counts	Counts to judge motor rotation at start up
General | Minimum limit speed (rpm)	Minimum of rotation speed (mechanical) [rpm]
General | PI control KP	Proportional parameter of speed PI
General | PI control KI	Integral parameter of speed PI
General | PI control limit	Integral limit of speed PI [V]
General | Hall interrupt mask value	Counts to mask error of hall signal
Motor Parameter | Pole pairs	Pole pairs
Motor Parameter | Resistance (ohm)	Resistance [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)	Inertia [kgm^2]
Hall sensor port U	Port number of U phase hall sensor
Hall sensor port V	Port number of V phase hall sensor
Hall sensor port W	Port number of W phase hall sensor

Table 10-5 Configuration Options initial value [1/2] (rm_motor_120_control_hall)

Options	RA6T2	RA6T3	RA4T1
General | Conduction type	Complementary First 60 degree PWM	Complementary First 60 degree PWM	Complementary First 60 degree PWM
General | Timeout counts (msec)	200	200	200
General | Maximum voltage (V)	22.0	22.0	22.0
General | Minimum voltage (V)	3.0	3.0	3.0
General | Speed PI decimation	0	0	0
General | Free run timer frequency (MHz)	120.0	50.0	50.0
General | Speed LPF	1.0	1.0	1.0
General | Step of speed reference change	0.2	0.2	0.2
General | Start reference voltage (V)	5.8	5.8	5.8
General | Hall wait counts	12	12	12
General | Minimum limit speed (rpm)	550	550	550
General | PI control KP	0.02	0.02	0.02
General | PI control KI	0.0005	0.0005	0.0005
General | PI control limit	24.0	24.0	24.0
General | Hall interrupt mask value	15	15	15
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
Hall sensor port U	BSP_IO_PORT_12_PIN_04	BSP_IO_PORT_00_PIN_08	BSP_IO_PORT_00_PIN_08
Hall sensor port V	BSP_IO_PORT_12_PIN_05	BSP_IO_PORT_00_PIN_06	BSP_IO_PORT_00_PIN_06
Hall sensor port W	BSP_IO_PORT_11_PIN_01	BSP_IO_PORT_00_PIN_15	BSP_IO_PORT_00_PIN_15

Table 10-6 Configuration Options initial value [2/2] (rm_motor_120_control_hall)

Options	RA8T1	RA8T2
General | Conduction type	Complementary First 60 degree PWM	Complementary First 60 degree PWM
General | Timeout counts (msec)	200	200
General | Maximum voltage (V)	22.0	22.0
General | Minimum voltage (V)	3.0	3.0
General | Speed PI decimation	0	0
General | Free run timer frequency (MHz)	120.0	250.0
General | Speed LPF	1.0	1.0
General | Step of speed reference change	0.2	0.2
General | Start reference voltage (V)	5.8	5.8
General | Hall wait counts	12	12
General | Minimum limit speed (rpm)	550	550
General | PI control KP	0.02	0.02
General | PI control KI	0.0005	0.0005
General | PI control limit	24.0	24.0
General | Hall interrupt mask value	15	15
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
Hall sensor port U	BSP_IO_PORT_09_PIN_07	BSP_IO_PORT_10_PIN_08
Hall sensor port V	BSP_IO_PORT_09_PIN_05	BSP_IO_PORT_10_PIN_10
Hall sensor port W	BSP_IO_PORT_09_PIN_06	BSP_IO_PORT_10_PIN_09

10.4 List of 120-degree driver module parameters

The changeable parameters that are used in 120-degree driver control module and these initial values are listed below.

Table 10-7 Configuration options of driver module [1/2] (rm_motor_120_driver)

Option name	Description
Common | ADC_B Support	ADC_B module support
Common | Shared ADC Support	Selection of using shared ADC module
General | PWM output port UP	PWM output (Up) port
General | PWM output port UN	PWM output (Un) port
General | PWM output port VP	PWM output (Vp) port
General | PWM output port VN	PWM output (Vn) port
General | PWM output port WP	PWM output (Wp) port
General | PWM output port WN	PWM output (Wn) port
General | PWM timer frequency (MHz)	PWM timer frequency [MHz]
General | PWM carrier period (Microseconds)	PWM carrier frequency [Microseconds]
General | Dead time (Raw counts)	Dead time count [Raw counts]
General | Current range (A)	Current detection range [A]
General | Voltage range (V)	Voltage detection range [V]
General | Resolution of A/D conversion	A/D conversion value
General | Offset of A/D conversion for current	A/D conversion offset
General | Conversion level of A/D conversion for voltage	Voltage A/D conversion rate
General | Counts for current offset measurement	Offset value calculation count
General | Input voltage	Inverter bus voltage
General | A/D conversion channel for U phase current	U phase current detection channel
General | A/D conversion channel for W phase current	W phase current detection channel
General | A/D conversion channel for main line voltage	Inverter bus voltage detection channel
General | A/D conversion channel for U phase voltage	U phase voltage detection channel
General | A/D conversion channel for V phase voltage	V phase voltage detection channel
General | A/D conversion channel for W phase voltage	W phase voltage detection channel
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

Table 10-8 Configuration options of driver module [2/2] (rm_motor_120_driver)

Option name	Description
General | A/D conversion unit for U phase voltage	Select the A/D conversion module for U phase voltage
General | A/D conversion unit for V phase voltage	Select the A/D conversion module for V phase voltage
General | A/D conversion unit for W phase voltage	Select the A/D conversion module for W phase voltage
General | GTIOCA stop level	Level when the upper arm is stopped
General | GTIOCB stop level	Level when lower arm is stopped
General | ADC interrupt module	A/D unit number which occurs interrupt
Modulation | Maximum duty	PWM maximum duty

Table 10-9 Configuration Options initial value [1/2] (rm_motor_120_driver)

Option name	RA6T2	RA6T3	RA4T1
Common | ADC_B Support	Enabled	Disabled	Disabled
Common | Shared ADC Support	Disabled	Disabled	Disabled
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)	50	50	50
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 | 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 | Counts for current offset measurement	500	500	500
General | Input voltage	24.0	24.0	24.0
General | A/D conversion channel for U phase current	4	0	0
General | A/D conversion channel for W phase current	0	2	2
General | A/D conversion channel for main line voltage	Ver.1: 6 Ver.2: 7	4	4
General | A/D conversion channel for U phase voltage	-	-	-
General | A/D conversion channel for V phase voltage	-	-	-
General | A/D conversion channel for W phase voltage	-	-	-
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	-	-	-
General | A/D conversion unit for U phase voltage	-	-	-
General | A/D conversion unit for V phase voltage	-	-	-
General | A/D conversion unit for W phase voltage	-	-	-
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
General | ADC interrupt module	-	-	-
Modulation | Maximum duty	0.9375	0.9375	0.9375

Table 10-10 Configuration Options initial value [2/2] (rm_motor_120_driver)

Option name	RA8T1	RA8T2
Common | ADC_B Support	Disabled	Enabled
Common | Shared ADC Support	Enabled	Disabled
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)	50	50
General | Dead time (Raw counts)	240	500
General | Current range (A)	16.5	16.5
General | Voltage range (V)	73.51	73.51
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 | Counts for current offset measurement	500	500
General | Input voltage	24.0	24.0
General | A/D conversion channel for U phase current	0	6
General | A/D conversion channel for W phase current	2	10
General | A/D conversion channel for main line voltage	8	7
General | A/D conversion channel for U phase voltage	-	-
General | A/D conversion channel for V phase voltage	-	-
General | A/D conversion channel for W phase voltage	-	-
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	-	-
General | A/D conversion unit for U phase voltage	-	-
General | A/D conversion unit for V phase voltage	-	-
General | A/D conversion unit for W phase voltage	-	-
General | GTIOCA stop level	Pin Level Low	Pin Level Low
General | GTIOCB stop level	Pin Level High	Pin Level High
General | ADC interrupt module	1st	1st
Modulation | Maximum duty	0.9375	0.9375

10.5 Parameters about protection function

rm_motor_120_degree / Limit of over current
Set the threshold to judge overcurrent. Actual value is calculated by "input value * sqrt(2) * 1.5".

rm_motor_120_degree / Limit of over voltage
Set the threshold to judge over voltage error. Please select reasonable value with your environment.

rm_motor_120_degree / Limit of low voltage
Set the threshold to judge low voltage error. Please select reasonable value with your environment.

10.6 Changing the PWM carrier frequency

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

rm_motor_120_driver / PWM Carrier frequency
Set the frequency of PWM carrier.

10.7 modulation method (chopping pattern)

rm_motor_120_control_hall / Conduction type
In this sample program, 2 types chopping pattern can be selected. Default setting is "Complimentary first 60 chopping".

10.8 Parameters for inverter

10.8.1 Deadtime

rm_motor_120_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".

10.8.2 Current detection gain

rm_motor_120_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 10-11 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

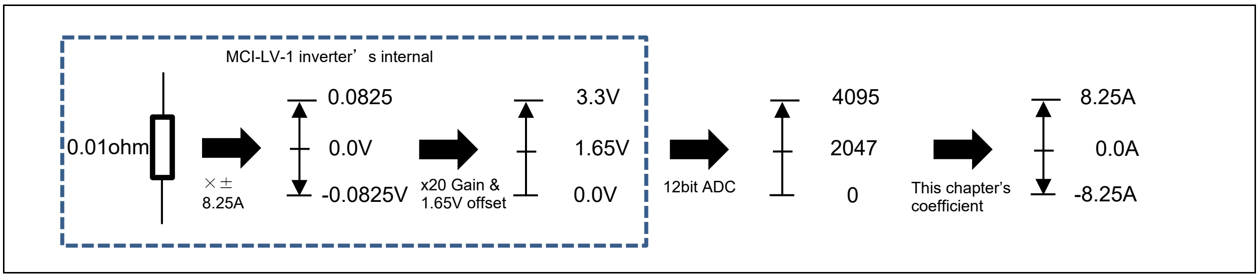


Figure 10-1 Current detection calculation flow

10.8.3 Voltage detection gain

rm_motor_120_driver / Voltage Range (V)

Set the value when maximum digits (12bit conversion : 4095) is detected, how match 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 10-12 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

10.9 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.

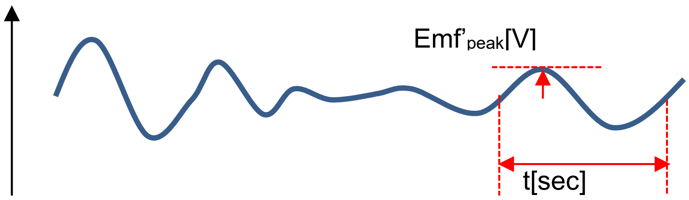
Nominal current

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}.

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

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

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

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

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

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

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

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

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

11. Control flowcharts

11.1 Main process

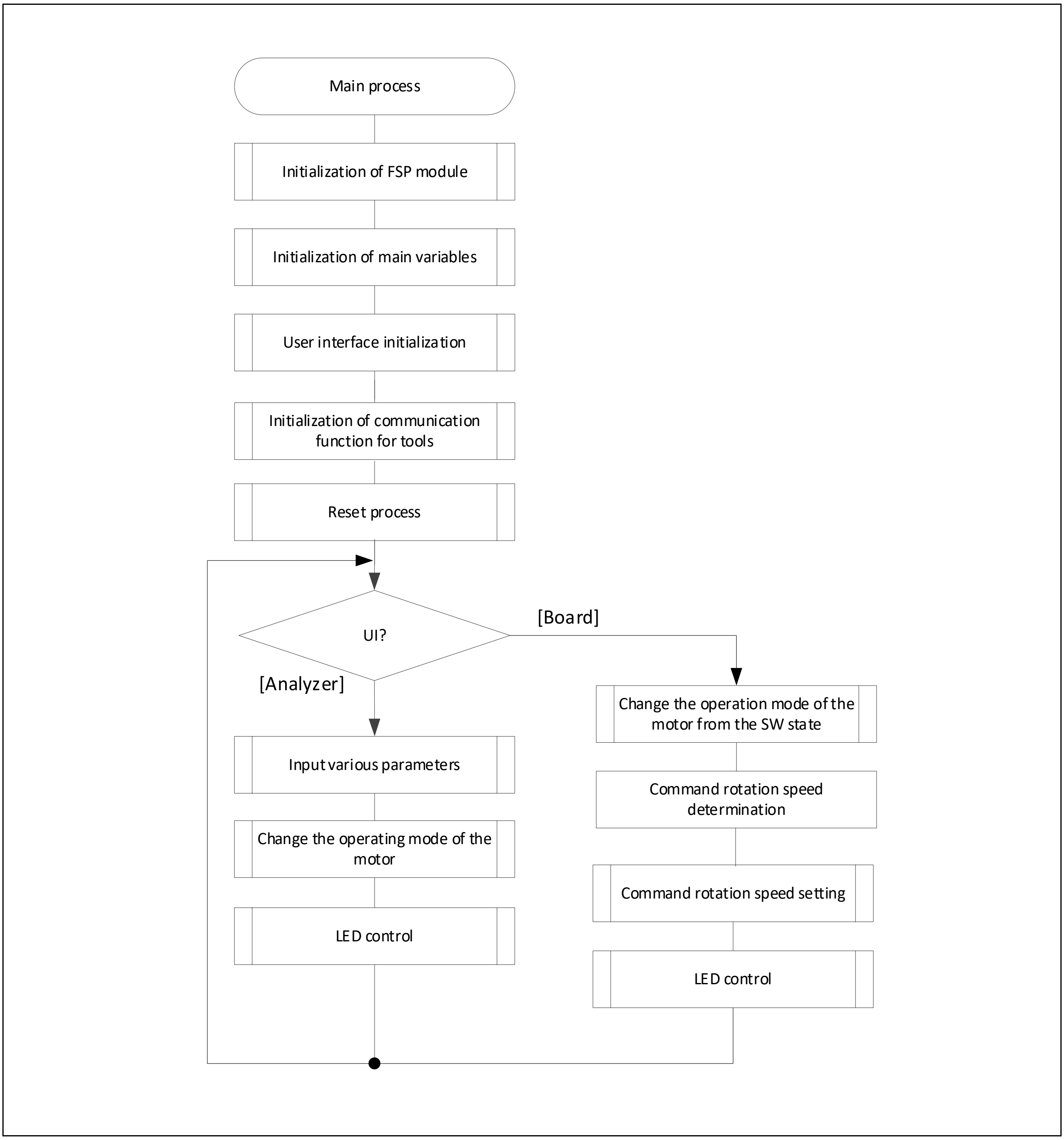


Figure 11-1 Main Process Flowchart

11.2 Carrier periodic interrupt process

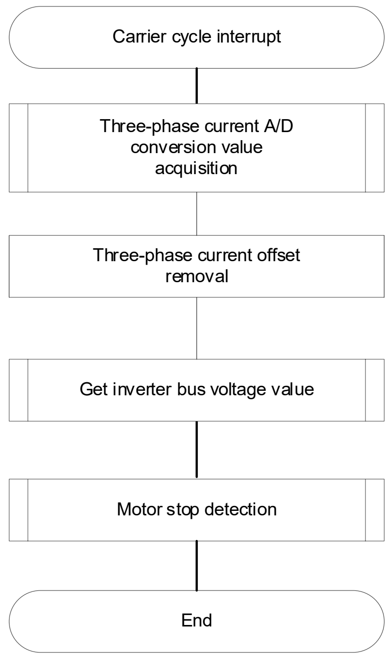


Figure 11-2 Carrier Periodic Interrupt Process Flowchart

11.3 Speed control Periodic Interrupt Process

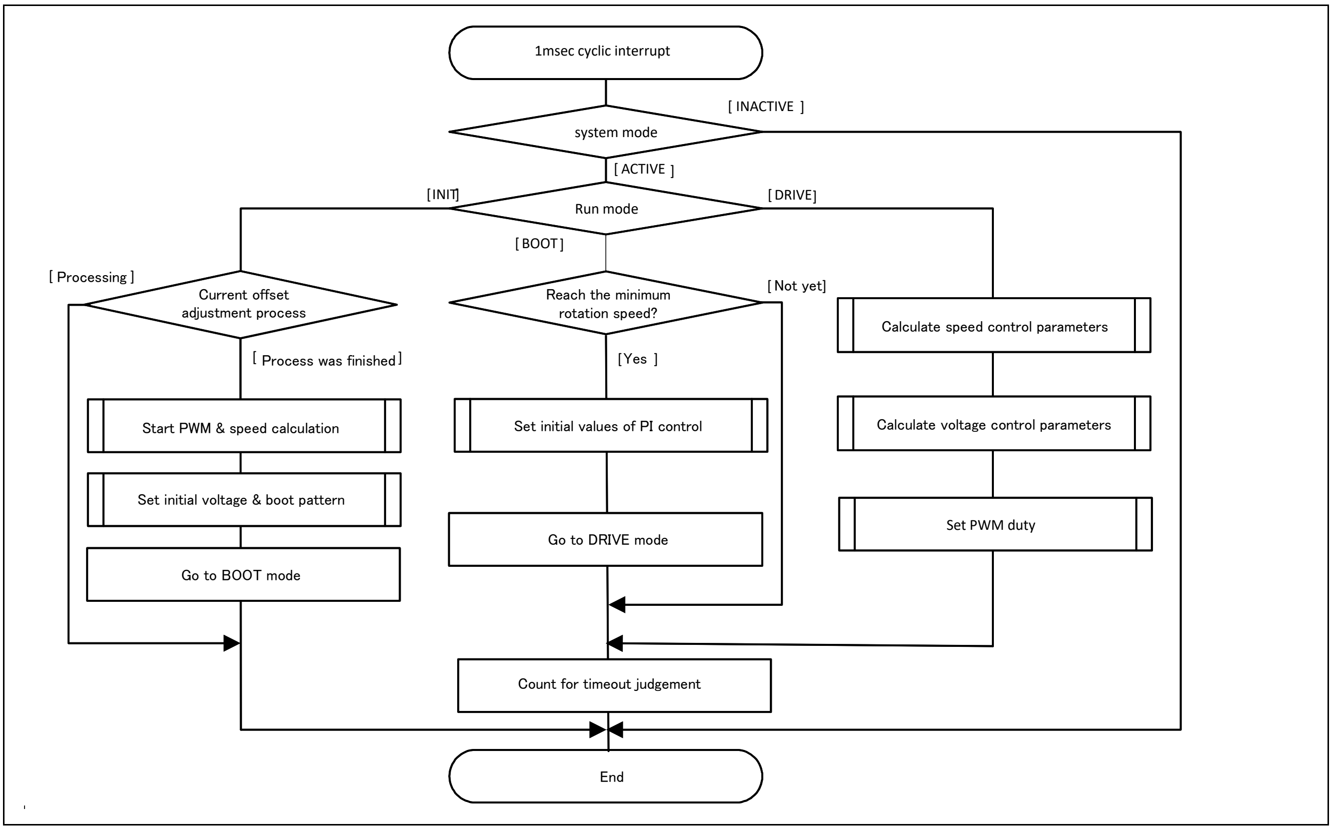


Figure 11-3 Speed Control Periodic Interrupt Process Flowchart

11.4 Hall signal interrupt process

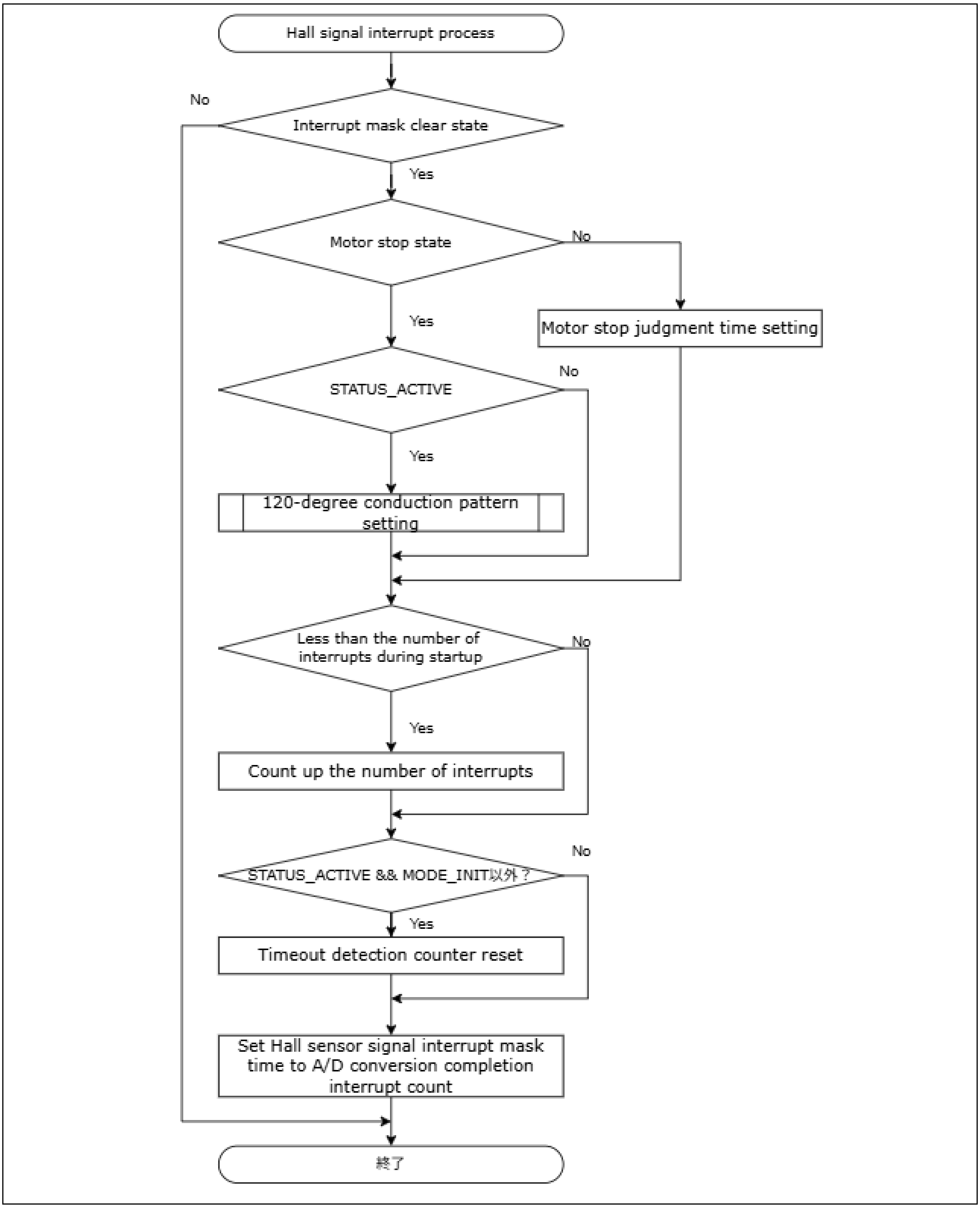


Figure 11-4 Hall signal interrupt process flowchart

11.5 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.

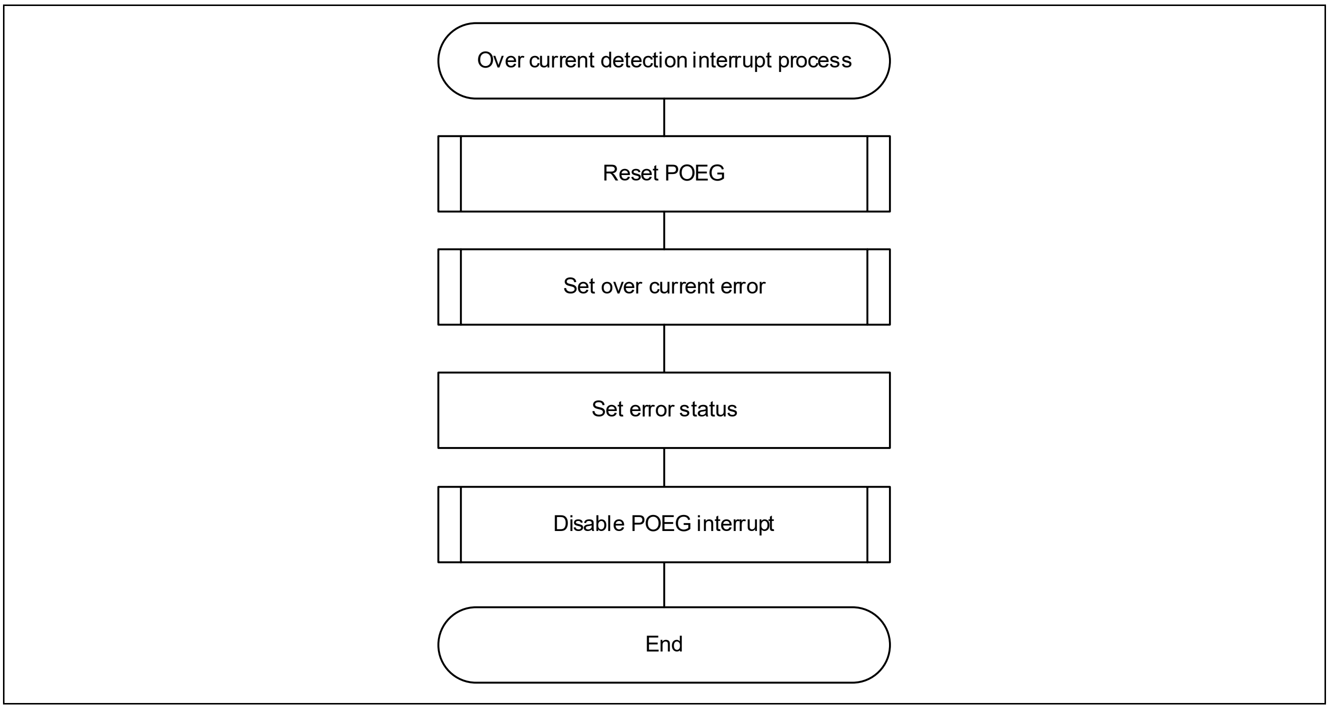


Figure 11-5 Over Current Detection Interrupt Process Flowchart

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

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/.