RX66T Group

Sensorless vector control for dual permanent magnetic synchronous motor

Abstract

This application note aims at explaining sensorless vector control software for two permanent magnet synchronous motors, by using functions of RX66T. The explanation includes, how to use the library of 'Renesas Motor Workbench' tool, that is support tool for motor control development.

This application note describes the MTU version that drives two motors with MTU, the GPT version that drives with GPT, and the hardware configuration that operates them.

The only difference between the MTU version and GPT version software is the timer used, and the specifications are the same.

For details on the differences and commonalities between the MTU and GPT versions of the hardware, refer to "2.3.2 Peripheral Functions".

After that, unless otherwise specified, the specifications are common to the MTU version and GPT version.

The target software of this application note is only to be used as reference and Renesas Electronics Corporation does not guarantee the operations. Please use them after carrying out a thorough evaluation in a suitable environment.

Operation Checking Device

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

- RX66T (R5F566TEADFP)

Target Software

The target software of this application note is as follows.

MTU version

- RX66T_MRSSK2_SPM_LESS_FOC_MTU_CSP_RV100 (IDE: CS+)
- RX66T_MRSSK2_SPM_LESS_FOC_MTU_E2S_RV100 (IDE: e²studio)

GPT version

- RX66T_MRSSK2_SPM_LESS_FOC_GPT_CSP_RV100 (IDE: CS+)
- RX66T_MRSSK2_SPM_LESS_FOC_GPT_E2S_RV100 (IDE: e²studio)

- RX66T Sensorless vector control software for 'Evaluation System for BLDC Motor' and 'RX66T CPU Card'

Contents

1. Overview

This application note explains how to implement two sensorless vector control software of permanent magnet synchronous motor (PMSM)[1] using the RX66T microcontroller. The explanation includes, how to use the library of 'Renesas Motor Workbench' tool, that is support tool for motor control development.

Note that the software uses the algorithm described in the application note 'Sensorless vector control for permanent magnet synchronous motor (Algorithm)' (R01AN3786), so please refer to that for the details of the algorithm.

In this document, motors are called motor A and motor B.

Note: 1. PMSM is also known as brushless DC motor (BLDC).

1.1 Development Environment

Table 1-1 and Table 1-2 show development environment of the software explained in this application note.

Table 1-1   Hardware Development Environment

Microcontroller	Evaluation board	Motor[3]
RX66T (R5F566TEADFP)	48V inverter board (2 unit) RX66T CPU Card 2nd motor Extender Board	TG-55L KA,24V (2 piece)

Table 1-2   Software Development Environment

IDE version	Smart Configurator for RX	Toolchain version[4]
CS+ V8.06.00	Standalone Version 2.10.0	CC-RX: V3.03.00
e² studio version 2021-07	Bundled with e² studio as plug-in	

For purchase and technical support, contact sales representatives and dealers of Renesas Electronics Corporation.

Notes: 1. 48V 5A Inverter Board for BLDC Motor (RTK0EM0000B10020BJ) and RX66T CPU Card (RTK0EMX870C00000BJ) are products of Renesas Electronics Corporation. 48V 5A Inverter Board for BLDC Motor is included in Evaluation System for BLDC Motor (RTK0EMX270S00020BJ).
2. TG-55L-KA is a product of TSUKASA ELECTRIC.
TSUKASA ELECTRIC (https://www.tsukasa-d.co.jp/)
3. Motors conforming to the inverter specifications listed in chapter 2 of Evaluation System for BLDC Motor User's Manual (R12UZ0062) can be connected to the product. When using motors other than the one included with the product, make sure to check the motor specifications carefully.
4. If the same version of the toolchain (C compiler) specified in the project is not in the import destination, the toolchain will not be selected and an error will occur.
Check the selected status of the toolchain on the project configuration dialog.
For the setting method, refer to FAQ 3000404.
(https://en-support.renesas.com/knowledgeBase/18398339)

2. System Overview

Overview of this system is explained below.

This APN describes two type hardware configurations.

1. Not using the extender board:
   The CPU card and the inverter board need to be modified, and you need to make the cable yourself.
2. Using the extender board:
   The CPU card needs to be modified, and needs to prepare the cable commercially available.

2.1 Hardware Configuration

The hardware configuration is shown below.

(1). Overall configuration (Not using the extender board)
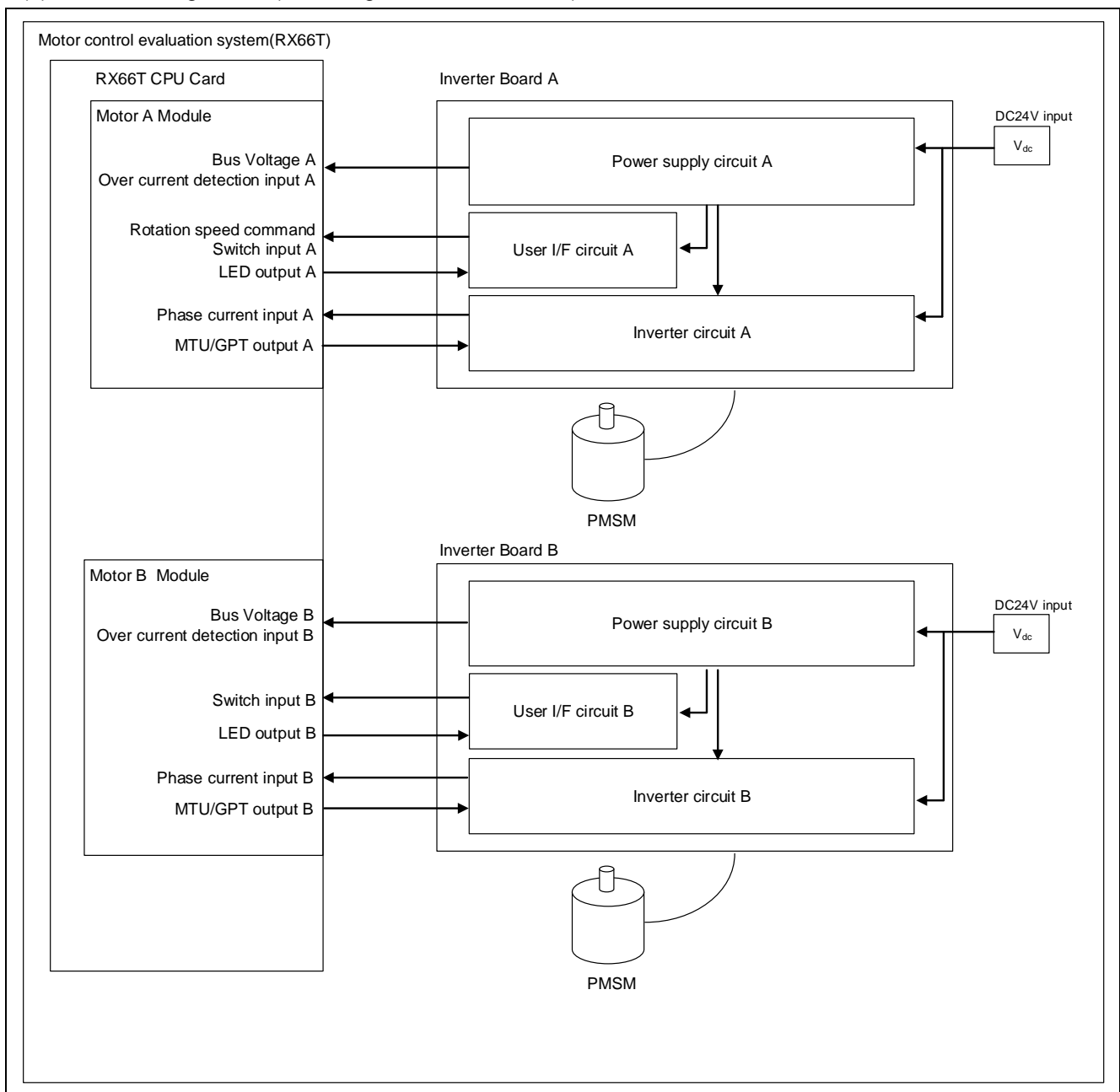
Figure 2-1 Hardware Configuration Diagram (Not using the extender board)

(2). Overall configuration (Using the extender board)
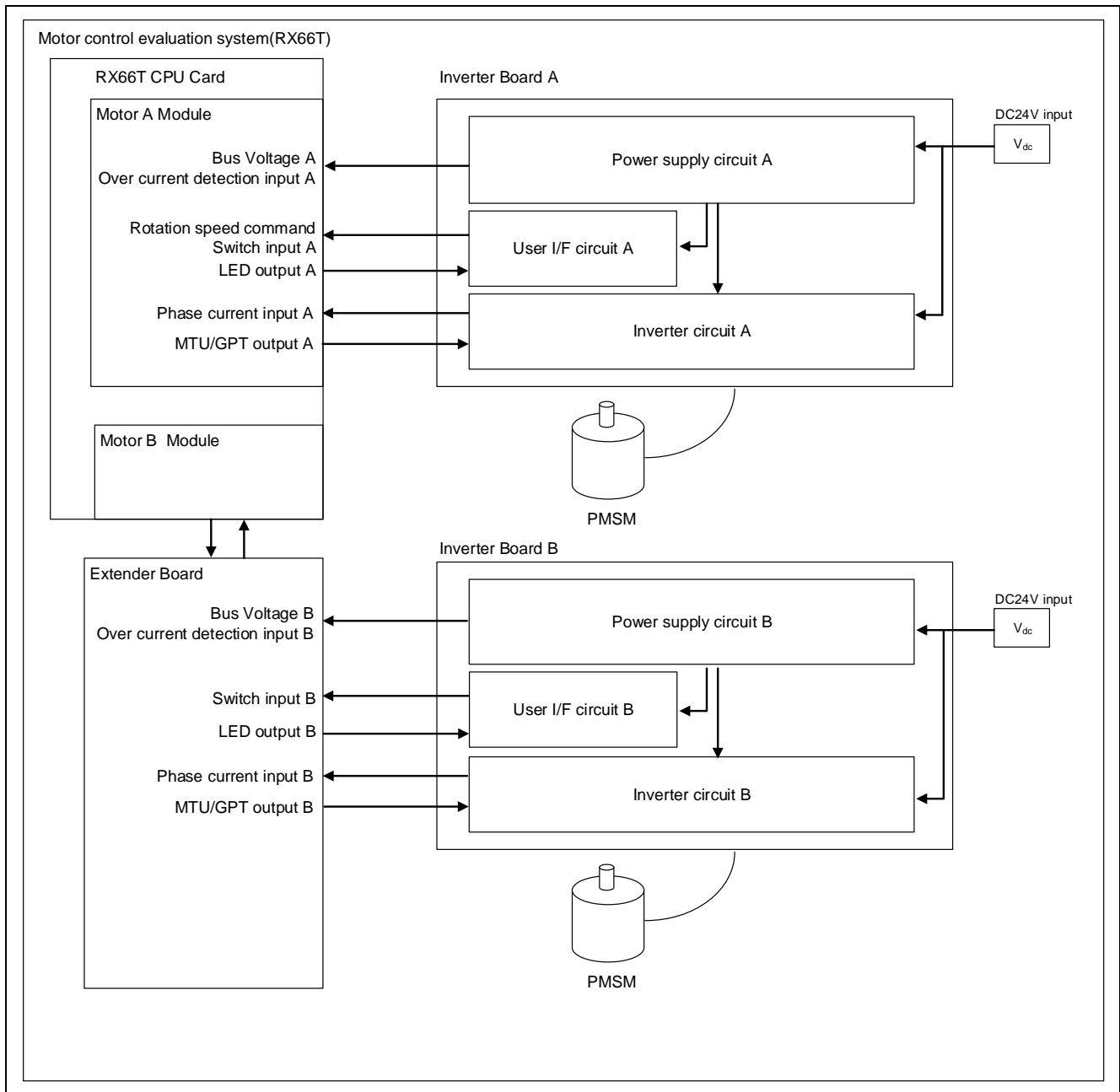
Figure 2-2 Hardware Configuration Diagram (Using the extender board)

(3). Motor A module configuration
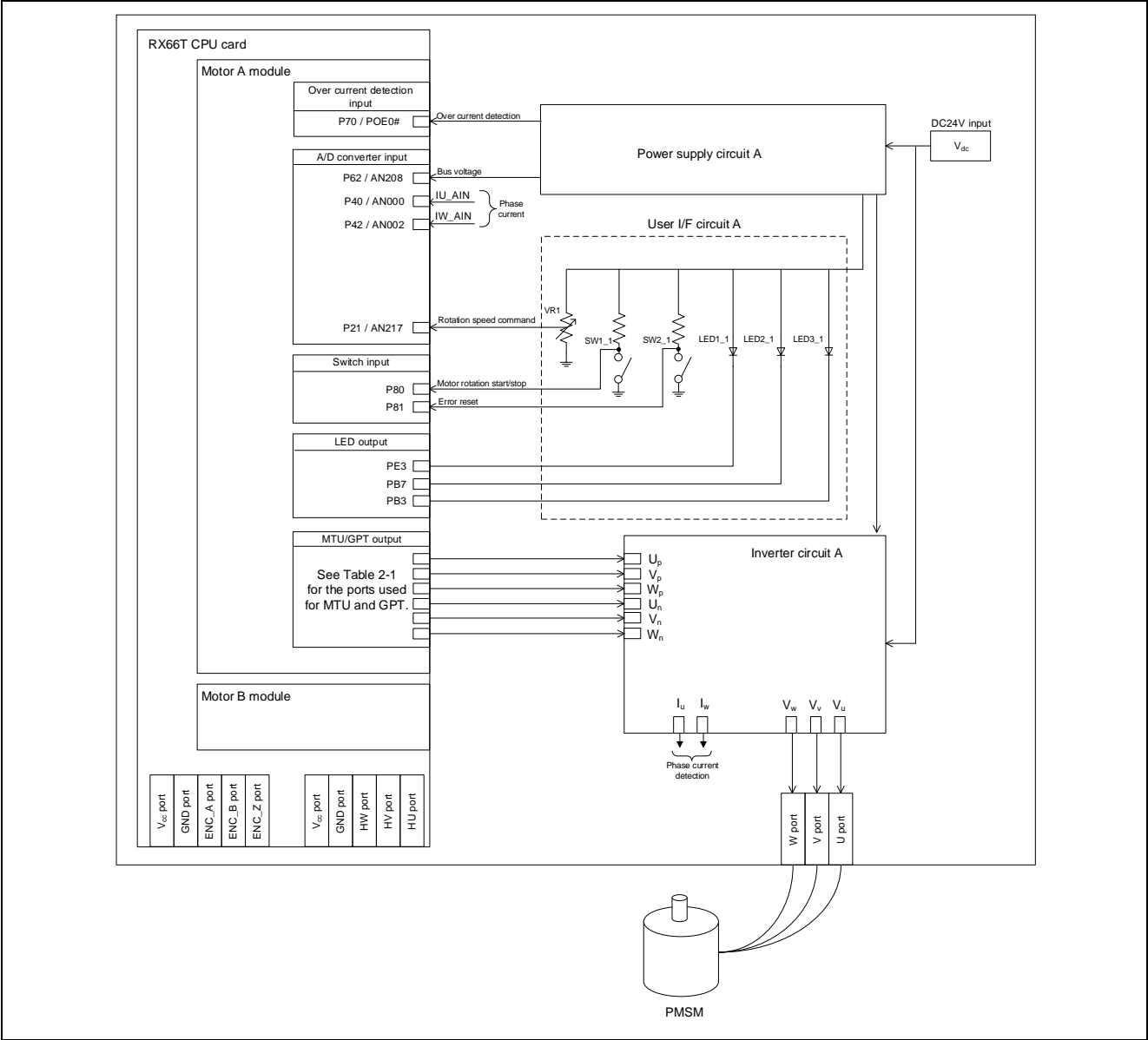
Figure 2-3 Hardware Configuration Diagram (Motor A module)

The ports used by the MTU version and the GPT version are common, and only the pin function of the PWM output is different. The pin functions of the PWM output used for each of the MTU version and GPT version are shown below.

Table 2-1 The pin functions of the PWM output for MTU version/GPT version

Port	MTU pin functions	GPT pin name functions
P71	MTIOC3B	GTIOC4A
P72	MTIOC4A	GTIOC5A
P73	MTIOC4B	GTIOC6A
P74	MTIOC3D	GTIOC4B
P75	MTIOC4C	GTIOC5B
P76	MTIOC4D	GTIOC6B

(4). Motor B module configuration
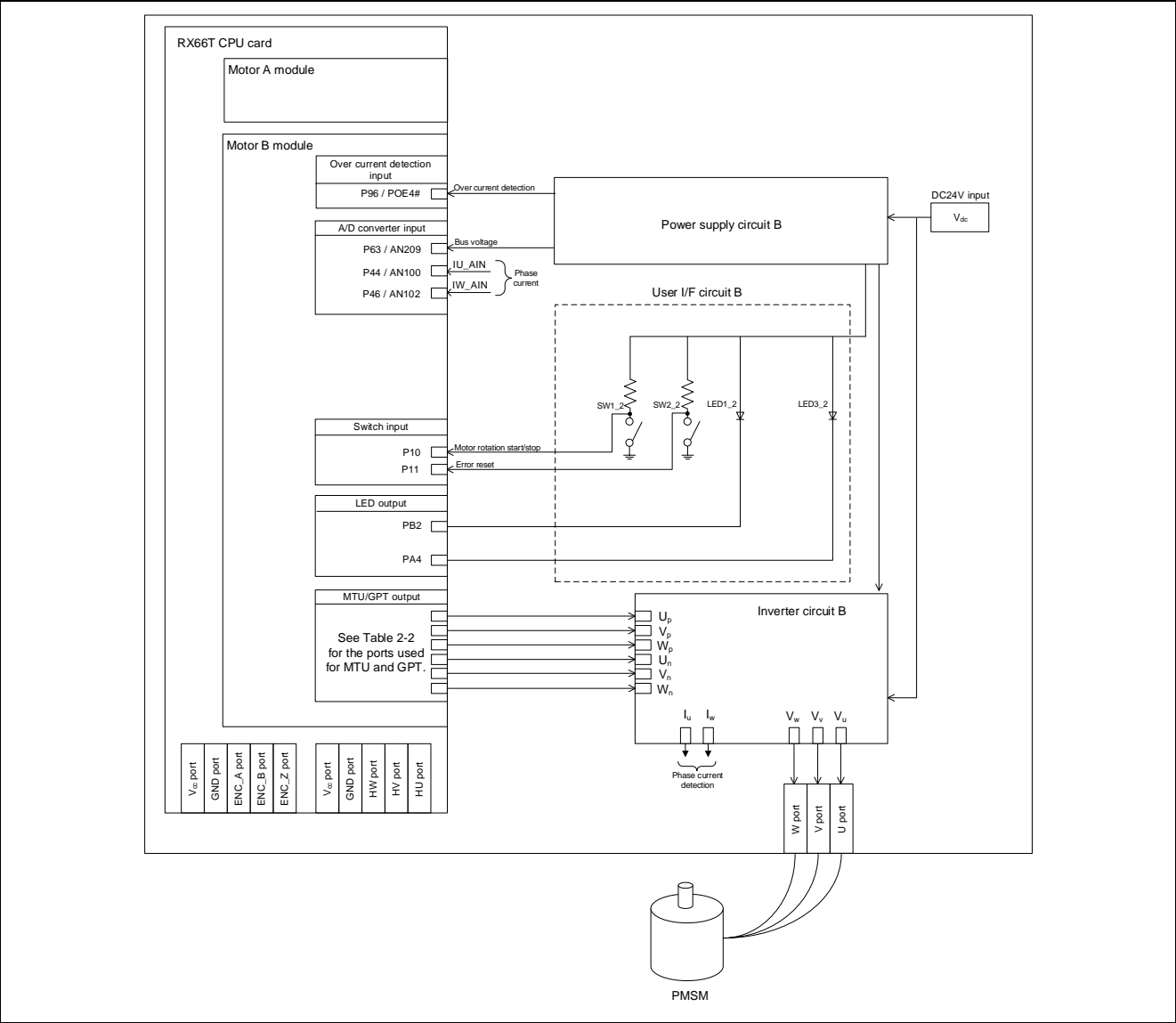
Figure 2-4 Hardware Configuration Diagram (Motor B module)

The ports used by the MTU version and the GPT version are common, and only the pin function of the PWM output is different. The pin functions of the PWM output used for each of the MTU version and GPT version are shown below.

Table 2-2 The pin functions of the PWM output for MTU version/GPT version

Port	MTU pin functions	GPT pin functions
P95	MTIOC6B	GTIOC7A
P94	MTIOC7A	GTIOC8A
P93	MTIOC7B	GTIOC9A
P92	MTIOC6D	GTIOC7B
P91	MTIOC7C	GTIOC8B
P90	MTIOC7D	GTIOC9B

2.2 Hardware modification details

The hardware modification is required to use this system. Here, it introduce each modification method mentioned above (1. Not using the extender board, 2. Using the extender board). The parts introduced in this chapter are just examples. Be sure to check the parts to be used by yourself.

2.2.1 Not using the extender board

(1) Mount the pin header on the "CNE" of the inverter board B.

(ex.)Molex 901310945
2.54mm Pitch C-Grid III
Modular Crimp Housing,
Dual Row, 50 Circuits, Black
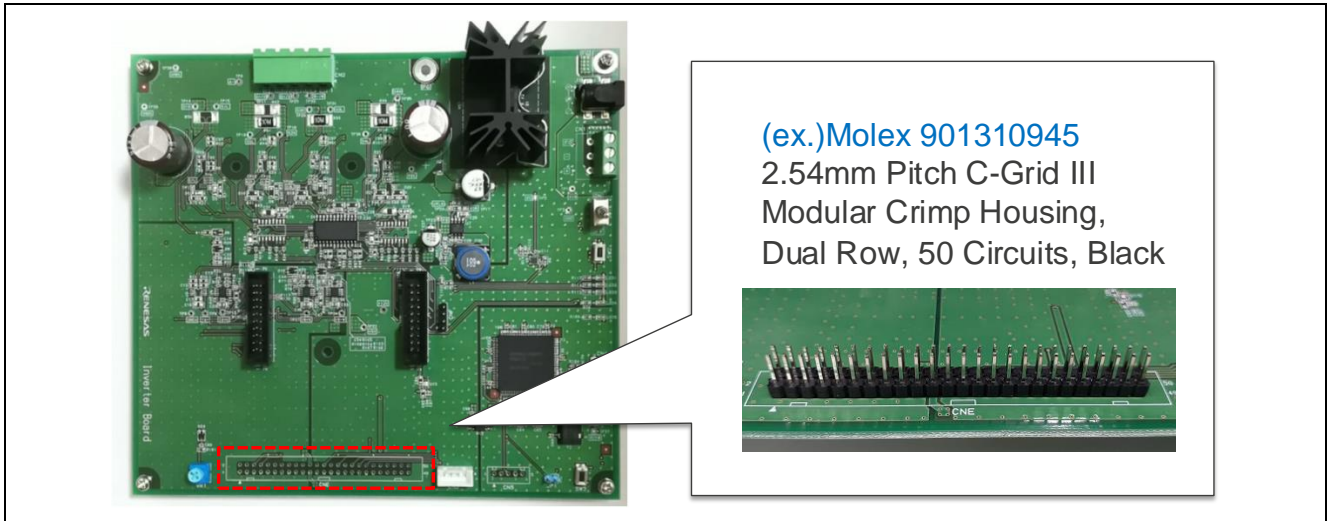
Figure 2-5 Mounting the pin header to the inverter board B

(2) Mount the 0Ω resistor(size : 0603(mm)) at the R140, R142, R144, R145, R146, R147, R148, R149, R150, R152, R153, R154, R155, R156 on the inverter board 2.

Inverter board B front side

Inverter board B back side
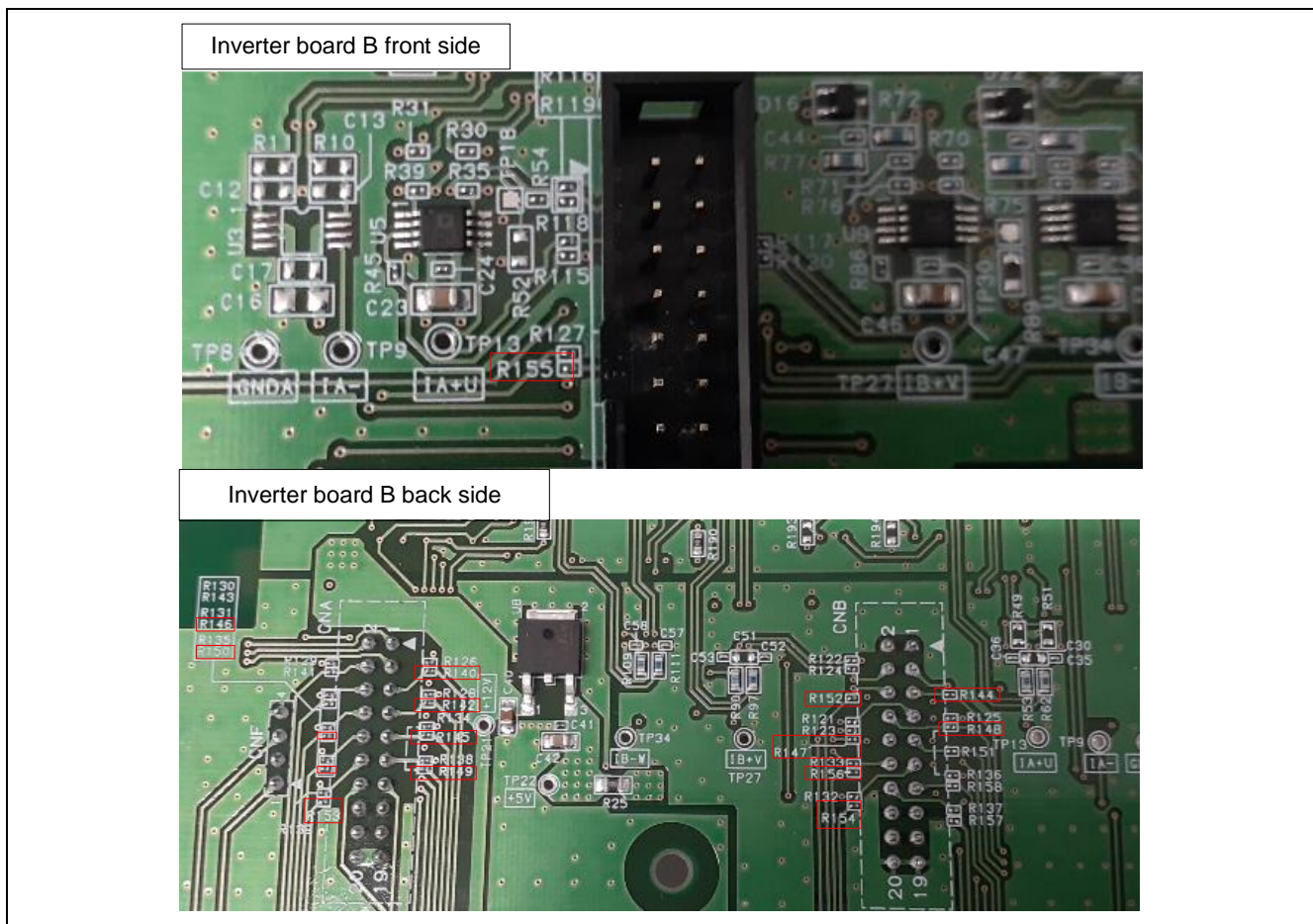
Figure 2-6 Mounting any resistors to the inverter board B

(3) Mount the pin header on the "CNC" of the CPU card.

CNC on RX66T CPU Card
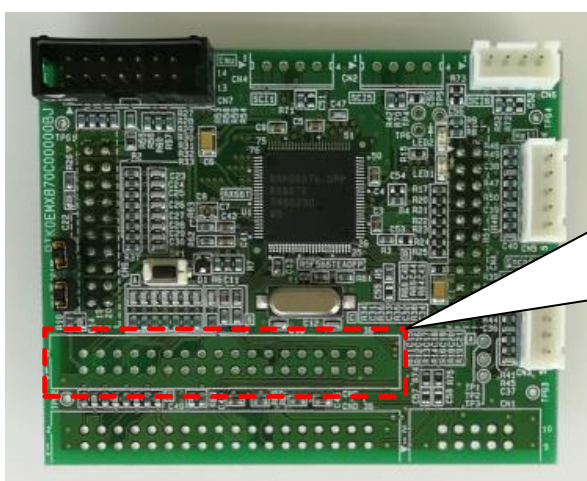
(ex.)A1-30PA-2.54DSA
2.54mm Pitch,
Dual Row, 30 Circuits
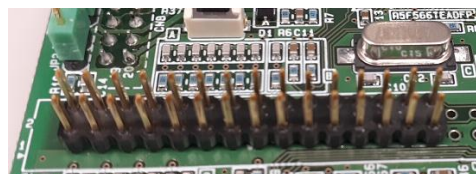
Figure 2-7 Mounting the pin header to the CPU card

(4) Connect the corresponding signal between the inverter board and the CPU card.

Inverter board B

(ex.)Molex 901430050
2.54mm Pitch C-Grid III Modular
Crimp Housing, Dual Row, 50
Circuits, Black

Inverter board B CNE

(ex.)Molex 797582022
Pre-Crimped Lead C-Grid III
Female-to-C-Grid III Female,
150.00mm Length, 22 AWG, Black

(ex.)Molex 901430030
2.54mm Pitch C-Grid III Modular
Crimp Housing, Dual Row, 30
Circuits, Black

CPU Card CNC

CNE

CPU-INV cable

CNC

CPU Card

Inverter board A

Image of the entire system

Figure 2-8 Connection between the inverter board and the CPU card

Table 2-3 Pin assignment correspondence table

INV (CNE)		RX66T(CNC)	
Pin#	Pin name	Pin#	Pin name
3	GND_A	1	GND_A
12	VDC_EX	2	VDC_EX
16	I_U/A+_EX	3	I_U/A+_EX
18	I_V/B+_EX	4	I_V/B+_EX
20	I_W/B-_EX	5	I_W/B-_EX
4	V_U/A+_EX	7	V_U/A+_EX
6	V_V/B+_EX	8	V_V/B+_EX
8	V_W/B-_EX	9	V_W/B-_EX
23	GND_D	11	GND_D
19	GND_A	12	GND_A
41	LED1	13	LED1
44	LED2	14	LED2
43	LED3	15	LED3
40	FO#_EX	16	FO#_EX
24	PWM_WL/B-L_EX	17	PWM_WL/B-L_EX
26	PWM_VL/B+L_EX	18	PWM_VL/B+L_EX
28	PWM_UL/A+L_EX	19	PWM_UL/A+L_EX
30	PWM_WH/B-_EX	20	PWM_WH/B-_EX
32	PWM_VH/B+H_EX	21	PWM_VH/B+H_EX
34	PWM_UH/A+H_EX	22	PWM_UH/A+H_EX
46	SW1	23	SW1
45	SW2	24	SW2
25	GND_D	25	GND_D
27	GND_D	26	GND_D

2.2.2   Using an extender board

If use the extender board that fits the connector accommodation, there is little board modification and it can be realized easy connection with commercially available cable. For the extender board, refer to the information (CAD data, circuit diagram, BOMList) included in the APN for development.

(1) Mount the pin header on the "CNC" of the CPU card.

CNC on RX66T CPU Card

(ex.)A1-30PA-2.54DSA
2.54mm Pitch,
Dual Row, 30 Circuits
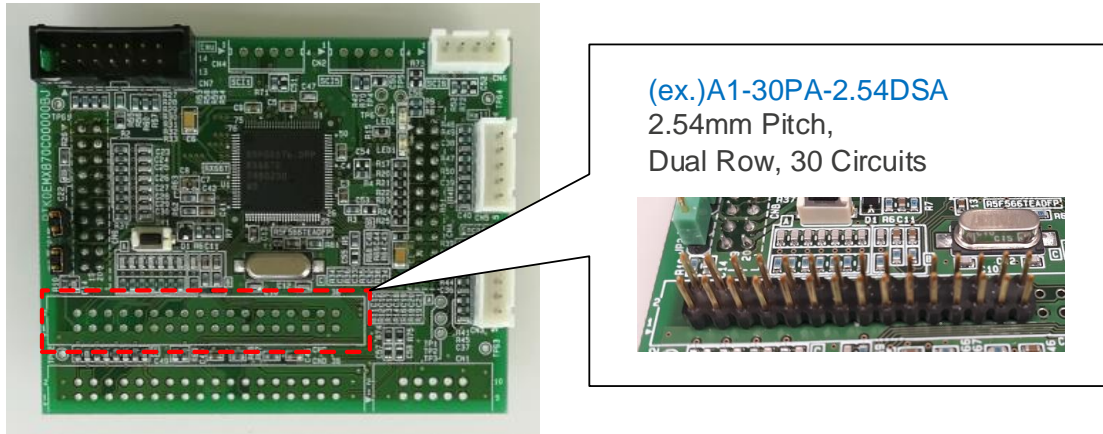
Figure 2-9 Mounting a pin header to the CPU card

(2) Connect the corresponding signal between the extender board and the CPU card.

Since the CNC of the extender board is 34pin, mount the connector without envelope to connect with the CNC of 36pin RX66T. When connecting with the cable, make sure that the 1PIN side of each board is aligned.

Inverter board B

Extender board

CNC

1PIN

CPU-INV cable

1PIN

CNC

Inverter board A

CPU card

Image of the entire system
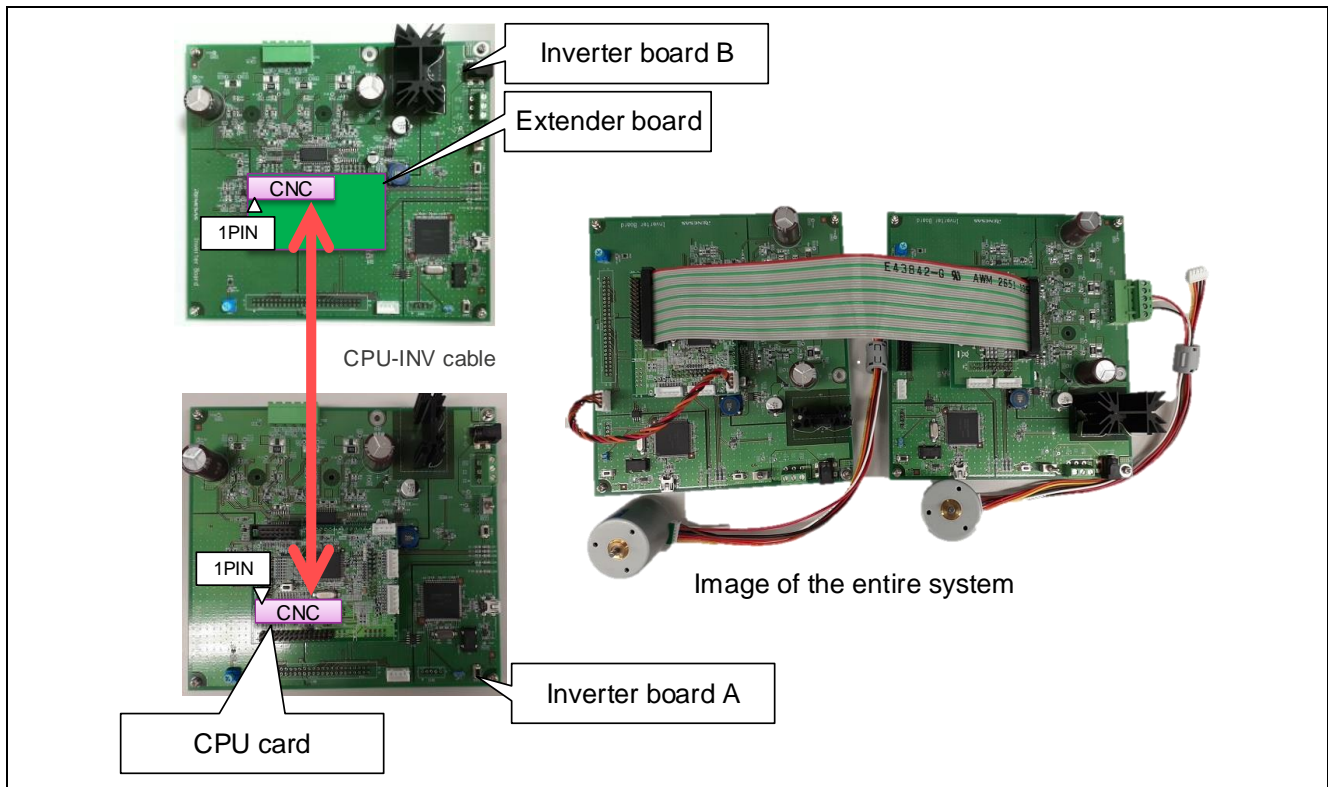
Figure 2-10 Connection between the extender board and the CPU card

Table 2-4 Pin assignment correspondence table

Extender board (CNC)		RX66T(CNC)	
Pin#	Pin name	Pin#	Pin name
1	PGAVSS_2	1	PGAVSS
2	VPN_2	2	VPN
3	IU_2	3	IU
4	IV_2	4	IV
5	IW_2	5	IW
6	TEMP_2	6	NC
7	VU_2	7	VU
8	VV_2	8	VV
9	VW_2	9	VW
10	VR_2	10	NC
11	GND	11	VSS
12	GND	12	VSS
13	LED1#_2	13	LED1
14	LED2#_2	14	LED2
15	LED3#_2	15	LED3
16	FO#_2	16	OC#
17	WN_2	17	WN
18	VN_2	18	VN
19	UN_2	19	UN
20	WP_2	20	WP
21	VP_2	21	VP
22	UP_2	22	UP
23	SW1#_2	23	SW1
24	SW2#_2	24	SW2
25	GND	25	VSS
26	GND	26	VSS
27	ENC_A_2	27	ENC_A
28	ENC_B_2	28	ENC_B
29	ENC_Z_2	29	ENC_Z
30	GND	30	VSS
31	GND	31	VSS
32	HALL_U_2	32	HALL_U
33	HALL_V_2	33	HALL_V
34	HALL_W_2	34	HALL_W
-	-	35	VRL2
-	-	36	NC

2.3   Hardware Specifications

2.3.1   User Interface

List of user interfaces of this system is in Table 2-5.

Table 2-5   User Interfaces

Item	Interface component	Function
Rotation speed	Variable resistor (VR1) for motor A	Reference value of rotation speed input (analog value)
START/STOP1	Toggle switch (SW1_1) for motor A	Command of motor A rotation start/stop
ERROR RESET1	Push switch (SW2_1) for motor A	Command of recovery from error status
LED1_1	Orange LED for motor A	- At the time of motor A rotation: ON - At the time of stop: OFF
LED3_1	Orange LED for motor A	- At the time of error detection: ON - At the time of normal operation: OFF
START/STOP2	Toggle switch (SW1_2) for motor B	Command of motor B rotation start/stop
ERROR RESET2	Push switch (SW2_2) for motor B	Command of recovery from error status
LED1_2	Orange LED for motor B	- At the time of motor B rotation: ON - At the time of stop: OFF
LED3_2	Orange LED for motor B	- At the time of error detection: ON - At the time of normal operation: OFF
RESET	Push switch (RESET1)	System reset

List of port interfaces of this system is given in Table 2-6, Table 2-7.

Table 2-6   Port Interfaces (For motor A)

R5F566TEADFP pin functions		
MTU version	**GPT version**	**Function**
P62 / AN208		Inverter bus voltage measurement
P21 / AN217		For rotation speed command value input (analog value)
P80		START/STOP toggle switch (SW1_1)
P81		ERROR RESET push switch (SW2_1)
PE3		LED1_1 ON/OFF control
PB7		LED3_1 ON/OFF control
P40 / AN000		U1 phase current measurement
P42 / AN002		W1 phase current measurement
P71 / MTIOC3B	P71 / GTIOC4A	PWM output (U_p1) / Low active
P72 / MTIOC4A	P72 / GTIOC5A	PWM output (V_p1) / Low active
P73 / MTIOC4B	P73 / GTIOC6A	PWM output (W_p1) / Low active
P74 / MTIOC3D	P74 / GTIOC4B	PWM output (U_n1) / High active
P75 / MTIOC4C	P75 / GTIOC5B	PWM output (V_n1) / High active
P76 / MTIOC4D	P76 / GTIOC6B	PWM output (W_n1) / High active
P70 / POE0#		PWM emergency stop input at the time of over-current detection

Table 2-7   Port Interfaces (For motor B)

R5F566TEADFP pin functions		
MTU version	**GPT version**	**Function**
P63 / AN209		Inverter bus voltage measurement
P10		START/STOP toggle switch (SW1_2)
P11		ERROR RESET push switch (SW2_2)
PB2		LED1_2 ON/OFF control
PA4		LED3_2 ON/OFF control
P44 / AN100		U2 phase current measurement
P46 / AN102		W2 phase current measurement
P95 / MTIOC6B	P95 / GTIOC7A	PWM output (U_p2) / Low active
P94 / MTIOC7A	P94 / GTIOC8A	PWM output (V_p2) / Low active
P93 / MTIOC7B	P93 / GTIOC9A	PWM output (W_p2) / Low active
P92 / MTIOC6D	P92 / GTIOC7B	PWM output (U_n2) / High active
P91 / MTIOC7C	P91 / GTIOC8B	PWM output (V_n2) / High active
P90 / MTIOC7D	P90 / GTIOC9B	PWM output (W_n2) / High active
P96 / POE4#		PWM emergency stop input at the time of over-current detection

2.3.2 Peripheral Functions

Of the peripheral functions used in this system, Table 2-8 shows the list of the peripheral functions that differ between MTU version and GPT version, and Table 2-9 shows the list of the common peripehral functions.

Table 2-8   List of the different Peripheral Functions

Peripheral functions	Resource	Use
MTU3	Motor A: CH3, CH4 Motor B: CH6, CH7	Complementary PWM output
GPTW	Motor A: CH4, CH5, CH6 Motor B: CH7, CH8, CH9	Complementary PWM output

Table 2-9   List of the Common Peripheral Functions

Peripheral functions	Resource	Use
12-bit A/D Converter *	Motor A: AN000, AN002, AN208, AN217 Motor B: AN100, AN102, AN209	• Rotation speed command value input • Current of each phase U and W measurement • Inverter bus voltage measurement
CMT	CH0	1 [ms] interval timer
POE3B	Motor A: POE0# Motor B: POE4#	Set PWM output ports to high impedance state to stop the PWM output.

*: Resources are common to the MTU / GPT version, but the usage is different because the specifications of each timer are different. Please refer to (3) and (4) below for details.

(1) Multi-Function Timer Pulse Unit 3 (MTU3)

On the channel 3, 4 (for motor A), and 6, 7 (for motor B) output (p-side active level: low, n-side active level: high) with dead time is performed by using the complementary PWM mode.

(2) General PWM Timer (GPTW)

On the channel 4, 5, 6 (for motor A), and 7, 8, 9 (for motor B) output with dead time (p−side is low active, n−side is high active) is performed by using the complementary PWM Output Operating Mode.

(3) 12-Bit A/D Converter (S12ADH) (MTU version)

U-phase current (Iu1) and W-phase current (Iw1), U-phase current (Iu2) and W-phase current (Iw2), inverter bus voltage1 (Vdc1) and rotation speed command (VR1), inverter bus boltage2 (Vdc2) are measured in "Single scan mode" (use a hardware trigger).
Detection of the U-phase current(Iu1, Iu2) and W-phase current(Iw1, Iw2) uses sample & hold function.
A/D conversion is implemented to be synchronized with carrier synchronized interrupt (refer to 3.1.7 Carrier synchronized interrupt) by using A/D Converter Start Request Delaying Function.

(4) 12-Bit A/D Converter (S12ADH) (GPT version)

U-phase current (Iu1) and W-phase current (Iw1), U-phase current (Iu2) and W-phase current (Iw2), inverter bus voltage1 (Vdc1) and rotation speed command (VR1), inverter bus boltage2 (Vdc2) are measured in "Group scan mode" (use a hardware trigger).
- Group A (for motor A) : Iu1, Iw1, Vdc1, VR1
- Group B (for motor B) : Iu2, Iw2, Vdc2
Detection of the U-phase current(Iu1, Iu2) and W-phase current(Iw1, Iw2) uses sample & hold function.
A/D conversion is implemented to be synchronized with carrier synchronized interrupt (refer to 3.1.7 Carrier synchronized interrupt) by using Event Link Function.

(5) Compare Match Timer (CMT)

The compare match timer is used as 1 [ms] interval timer.

(6) Port Output Enable 3 (POE3B)

PWM output ports are set to high impedance state when an over-current is detected (when a falling edge of the POE0#, POE4# port is detected) or when an output short circuit is detected.

2.4 Software Configuration

2.4.1 Software File Configuration

Folder and file configuration of the software are given below.
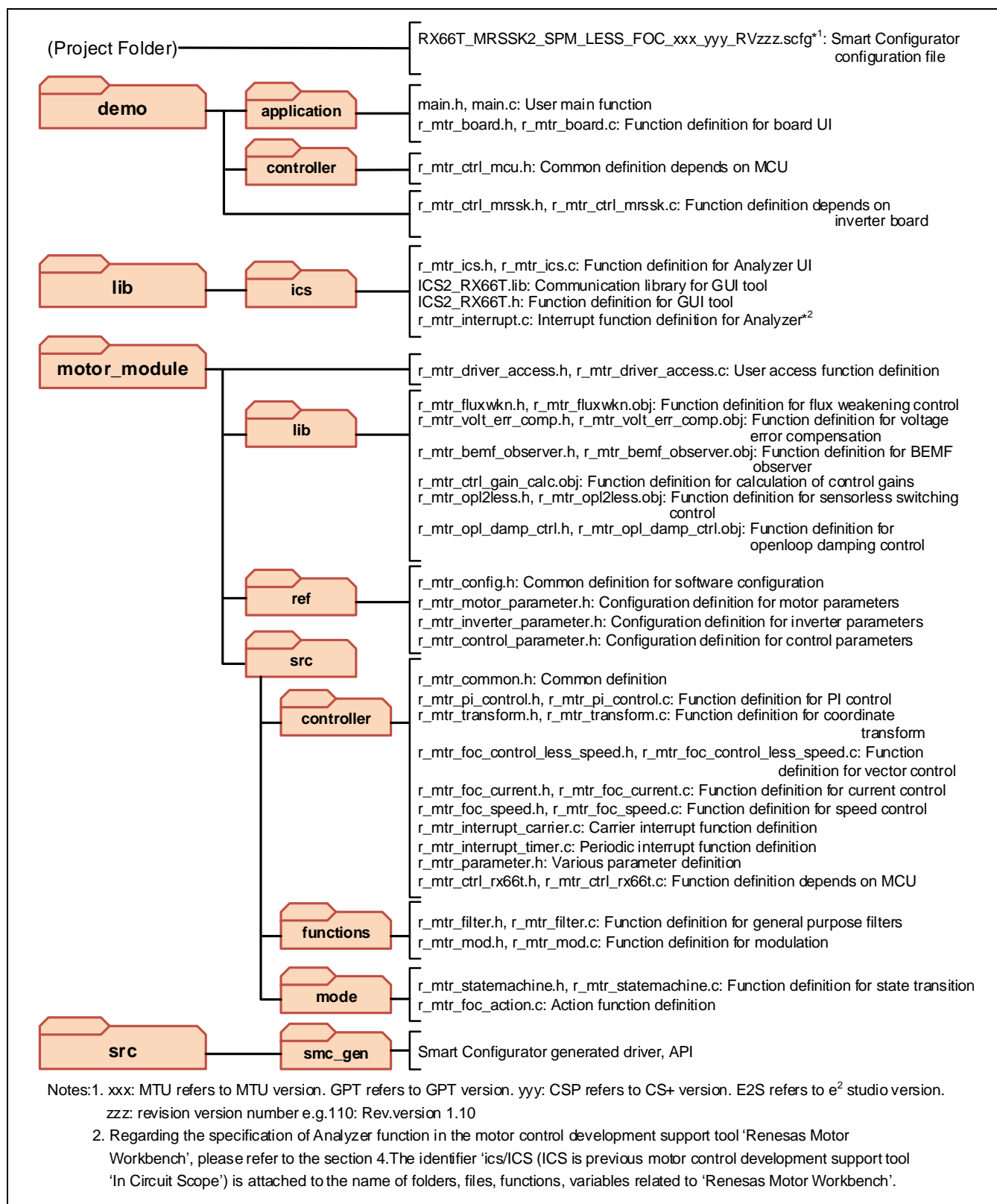
Figure 2-11   Folder and File Configuration

2.4.2 Smart Configurator File Configuration

Peripheral drivers are configured easily by using Smart Configurator (Hereinafter referred to as SC). Also, by using the components dedicated to the motor, it is possible to generate drivers for the multifunction timer pulse unit, general PWM timer and 12-bit A / D converter.

SC saves information such as the target MCU, peripheral, clock and pin functions setting for the project in *.scfg file. To check the peripheral function settings of this software, refer to the following file.

- "RX66T_MRSSK2_SPM_LESS_FOC_xxx_yyy_RVzzz.scfg"
- (xxx: MTU refers to MTU version. GPT refers to GPT version. yyy: CSP refers to CS+ version. E2S refers to e^2 studio version. zzz: revision version number)

Folder and file configuration of SC generated for MTU version and GPT version output are shown below.

Figure 2-12 MTU Version Smart Configurator Folder and File Configurations

(Smart Configurator Output Folder)

src

smc_gen

Config_PORT
Peripherals driver for ports. Unused ports are also configured here.
Config_PORT.h, Config_PORT.c: Driver and API for Port
Config_PORT_user.c: User function for Port

Config_S12AD0
Peripheral driver for 12-bit A/D conversion for I_u1, I_w1
Config_S12AD0.h, Config_S12AD0.c: Driver and API for AD port
Config_S12AD0_user.c: User function for AD function of Board UI

Config_S12AD1
Peripheral driver for 12-bit A/D conversion for I_u2, I_w2
Config_S12AD1.h, Config_S12AD1.c: Driver and API for AD port
Config_S12AD1_user.c: User function for AD function of Board UI

Config_S12AD2
Peripheral driver for 12-bit A/D conversion for VR1
Config_S12AD2.h, Config_S12AD2.c: Driver and API for AD port
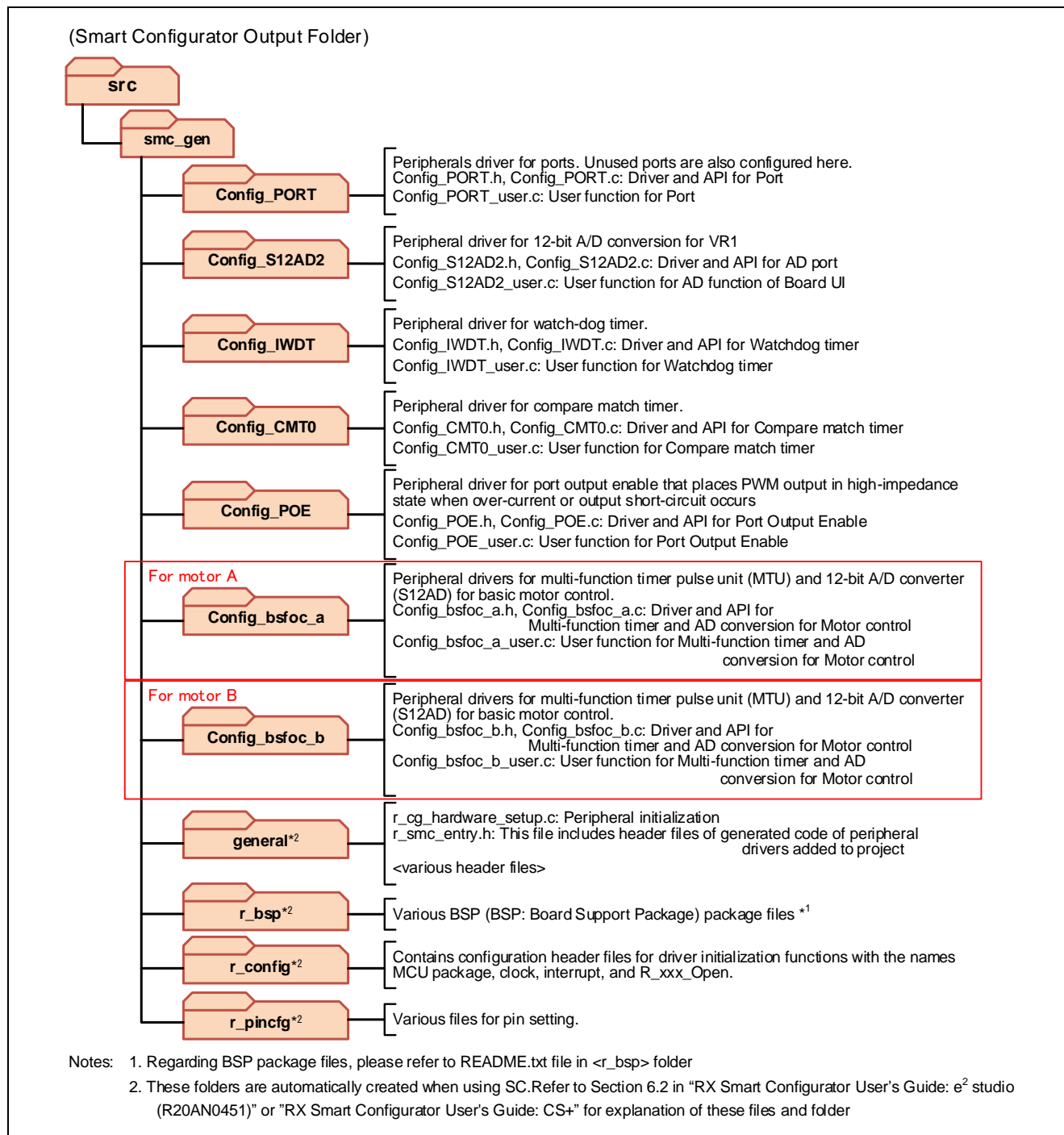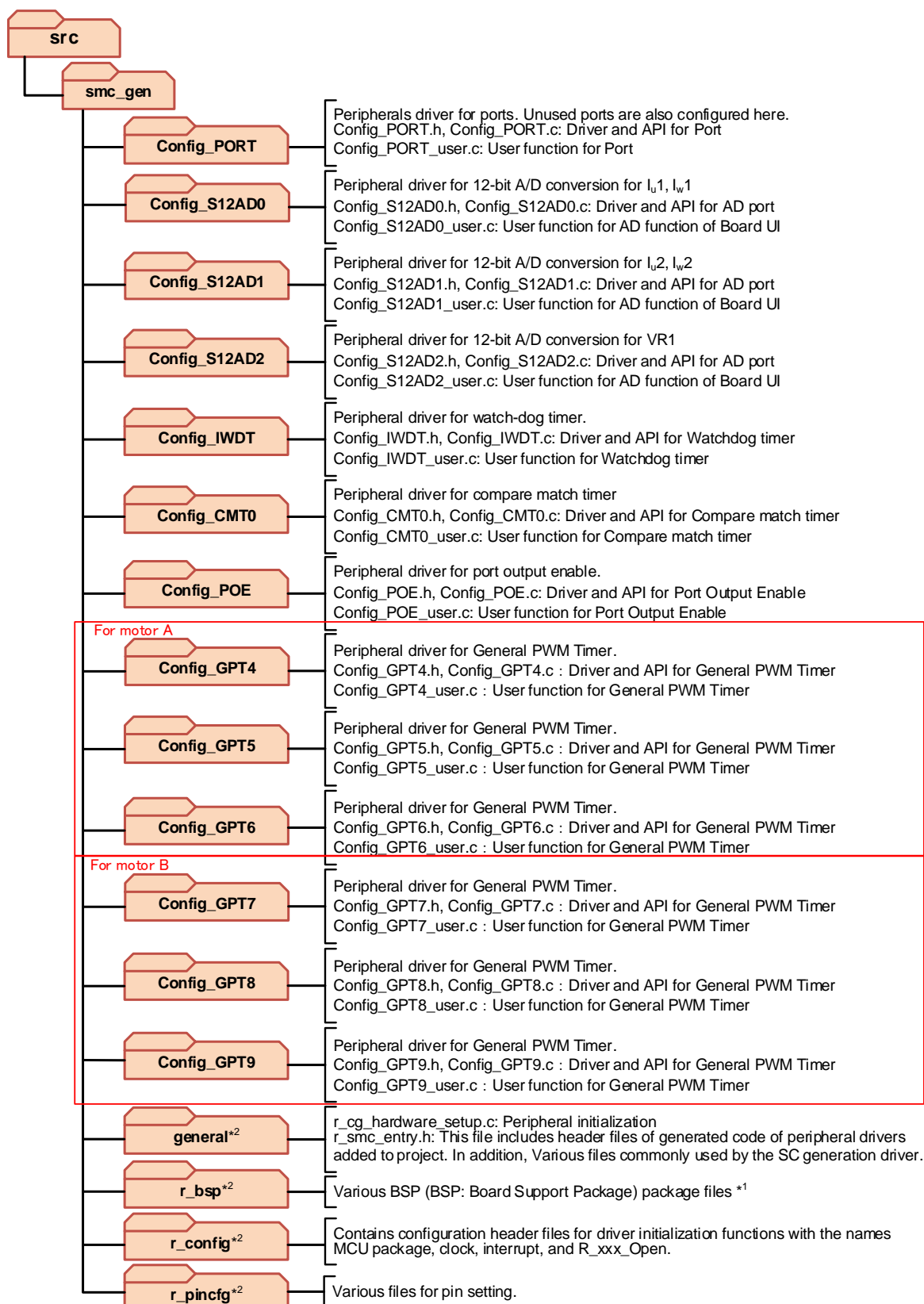Config_S12AD2_user.c: User function for AD function of Board UI

Config_IWDT
Peripheral driver for watch-dog timer.
Config_IWDT.h, Config_IWDT.c: Driver and API for Watchdog timer
Config_IWDT_user.c: User function for Watchdog timer

Config_CMT0
Peripheral driver for compare match timer
Config_CMT0.h, Config_CMT0.c: Driver and API for Compare match timer
Config_CMT0_user.c: User function for Compare match timer

Config_POE
Peripheral driver for port output enable.
Config_POE.h, Config_POE.c: Driver and API for Port Output Enable
Config_POE_user.c: User function for Port Output Enable

For motor A

Config_GPT4
Peripheral driver for General PWM Timer.
Config_GPT4.h, Config_GPT4.c : Driver and API for General PWM Timer
Config_GPT4_user.c : User function for General PWM Timer

Config_GPT5
Peripheral driver for General PWM Timer.
Config_GPT5.h, Config_GPT5.c : Driver and API for General PWM Timer
Config_GPT5_user.c : User function for General PWM Timer

Config_GPT6
Peripheral driver for General PWM Timer.
Config_GPT6.h, Config_GPT6.c : Driver and API for General PWM Timer
Config_GPT6_user.c : User function for General PWM Timer

For motor B

Config_GPT7
Peripheral driver for General PWM Timer.
Config_GPT7.h, Config_GPT7.c : Driver and API for General PWM Timer
Config_GPT7_user.c : User function for General PWM Timer

Config_GPT8
Peripheral driver for General PWM Timer.
Config_GPT8.h, Config_GPT8.c : Driver and API for General PWM Timer
Config_GPT8_user.c : User function for General PWM Timer

Config_GPT9
Peripheral driver for General PWM Timer.
Config_GPT9.h, Config_GPT9.c : Driver and API for General PWM Timer
Config_GPT9_user.c : User function for General PWM Timer

general[2]
r_cg_hardware_setup.c: Peripheral initialization
r_smc_entry.h: This file includes header files of generated code of peripheral drivers
added to project. In addition, Various files commonly used by the SC generation driver.

r_bsp[2]
Various BSP (BSP: Board Support Package) package files [1]

r_config[2]
Contains configuration header files for driver initialization functions with the names
MCU package, clock, interrupt, and R_xxx_Open.

r_pincfg[2]
Various files for pin setting.

Notes: 1. Regarding BSP package files, please refer to README.txt file in <r_bsp> folder

2. These folders are automatically created when using SC.Refer to Section 6.2 in "RX Smart Configurator User's Guide:
e[2] studio (R20AN0451)" or "RX Smart Configurator User's Guide: CS+" for explanation of these files and folder

Figure 2-13 GPT Version Smart Configurator Folder and File Configurations

SC Motor component configuration name in this software is named according to the following format.

Config_<Type of motor><Type of sensor><Motor drive method>_<Motor>

The following table shows definitions for configuring the above format.

Table 2-10   Smart Configurator Motor configuration name format

Type of motor	Type of sensor	Motor drive method	Motor
s: stepping motor b: brushless DC Motor (BLDC) i: induction motor	r: resolver e: encoder m: magnetic encoder s: sensor-less h: hall sensor	foc: field-oriented control 120: 120-degree conduction control	a: Motor A b: Motor B

In this project, the type of motor used is BLDC motor and driven with sensorless field-oriented control. Therefore, the configuration name is Config_bsfoc.

Tips:

The application-specific SC Motor component is presented in a simple and easy to understand GUI that consolidates several peripherals to configure peripherals required for basic motor drive in one interface. These peripherals include the multi-function timer pulse unit, general PWM timer and AD converter.

While benefiting from the ease of configuring Motor driver related peripherals in single interface, it is important to note that the Motor component set-up the same registers that could have been set-up by other components, (eg. AD converter) and vice-versa. This will cause overwriting of registers that are commonly set-up by both the Motor or AD converter component. This is expected and user must pay attention to these circumstances and to take appropriate countermeasure. User can make use of the generated <Configuration_name>_user.c of affected component to perform the countermeasure.

2.4.3   Module Configuration

Module configuration of the software is described below.

The source code of the MTU version and GPT version of Application Layer and Middle Layer is the same. Differences depending on the timer used (initial setting, PWM start / stop, duty update, etc.) are absorbed by the Device Layer (driver generated by SC).
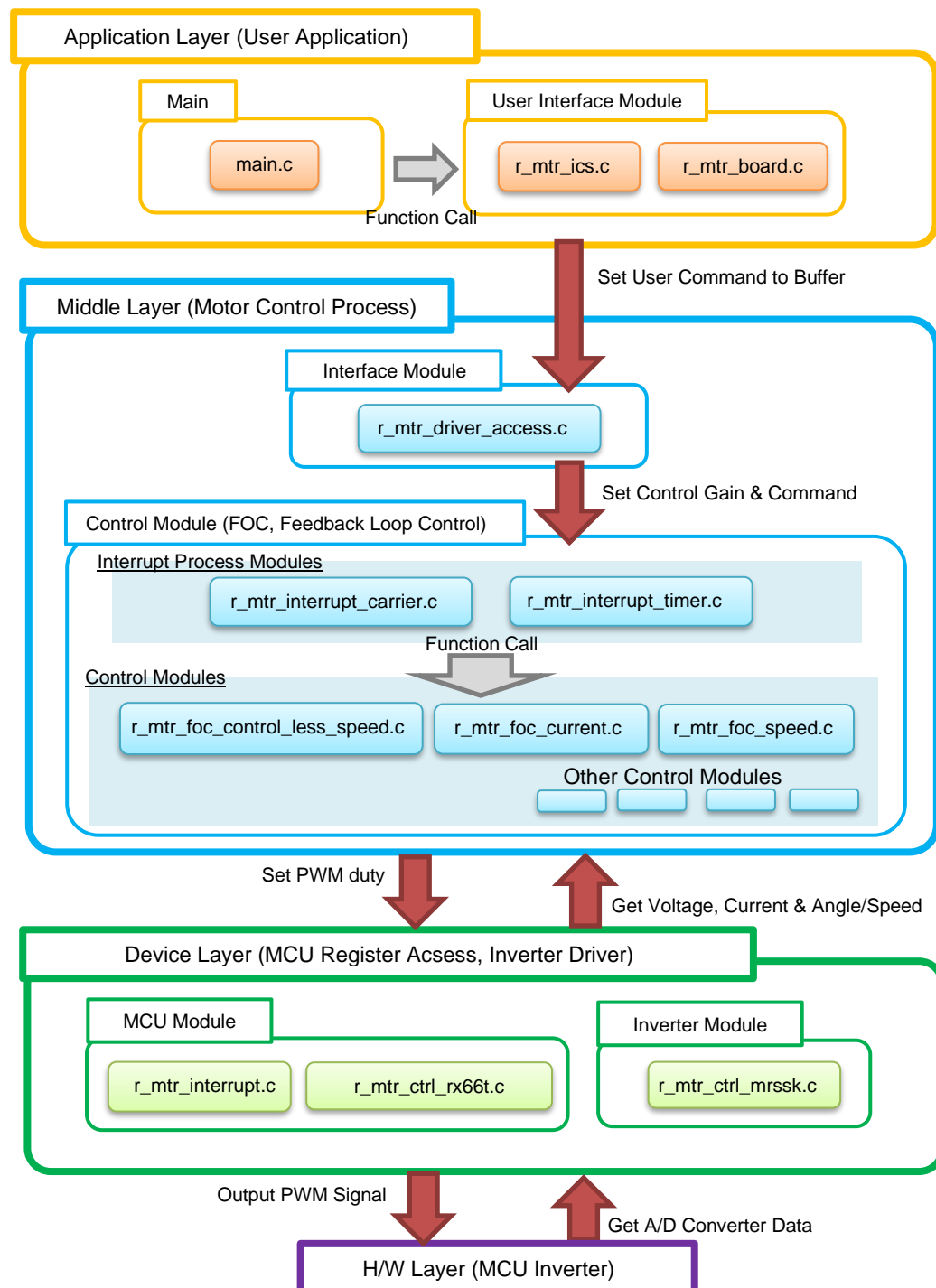

Figure 2-14   Module Configuration

2.5 Software Specifications

Table 2-11 shows basic software specification of this system. For details of the sensorless vector control, refer to the application note 'Sensorless vector control for permanent magnet synchronous motor (Algorithm)' (R01AN3786).

Specifications are the same for motor A and B, except for hardware interface.

Table 2-11   Basic Specifications of Sensorless Vector Control Software (Common to MTU / GPT)

Item	Content	
Control method	Vector control	
Position detection method	Sensorless	
Motor rotation start/stop	Determined depending on the level of SW1_1(for motor A) or SW1_2(for motor B) ('Low': rotation start, 'High': stop) or input from Renesas Motor Workbench	
Input voltage	DC 24 [V]	
Carrier frequency (PWM)	20 [kHz] (Carrier period: 50 [µs])	
Dead time	2 [µs]	
Control period	Current control / Position and speed estimation: 100 [µs] Speed control:1 [ms]	
Rotation speed control range	CW: 0 [rpm] to 2650 [rpm] CCW: 0 [rpm] to 2650 [rpm] When less than 600 [rpm], the motor is driven under Speed Open-loop control.*	
Natural frequency of each control system	Current control system: 300 [Hz] Speed control system: 3 [Hz] BEMF estimation system: 1000 [Hz] Position estimation system: 20 [Hz]	
Optimization setting of compiler	Optimization level	2 (-optimize = 2) (default setting)
	Optimization method	Size priority (-size) (default setting)
ROM/RAM size	ROM: 28.3KB RAM: 11.0KB	

Processing stop for protection	[Motor A]
	 Disables the motor A control signal output (six outputs), under any of the following conditions.
	   1. Current of each phase exceeds 0.89 [A] (monitored every 100 [µs])
	   2. Inverter bus voltage exceeds 28 [V] (monitored every 100 [µs])
	   3. Inverter bus voltage is less than 14 [V] (monitored every 100 [µs])
	   4. Rotation speed exceeds 3000 [rpm] (monitored every 100 [µs])
	
	 When an external over-current signal is detected (when a falling edge of the POE0# port is detected) or when the output short circuit is detected, the PWM output ports are set to high impedance state.
		[Motor B]
		 Disables the motor B control signal output (six outputs), under any of the following conditions.
		   1. Current of each phase exceeds 0.89 [A] (monitored every 100 [µs])
		   2. Inverter bus voltage exceeds 28 [V] (monitored every 100 [µs])
		   3. Inverter bus voltage is less than 14 [V] (monitored every 100 [µs])
		   4. Rotation speed exceeds 3000 [rpm] (monitored every 100 [µs])
		
		 When an external over-current signal is detected (when a falling edge of the POE4# port is detected) or when the output short circuit is detected, the PWM output ports are set to high impedance state.

Note: *   Set rotation speed command value higher than 600 [rpm] in order to rotate motor by Sensorless vector control.

2.6 Interrupt Priority

The interrupts and priorities used in this system are shown below.

Table 2-12 Interrupt priority

Interrupt level	Priority	function
0	(Interrupt Disable)	
1	Min	
2		
3		
4		
5		
6		RWM communication interrupt
7		
8		
9		
10		
11		1 [msec] Interrupt handling
12		Carrier synchronized interrupt (Motor A, B are same priority)
13		
14		
15	Max	Over current error interrupt (Motor A, B are same priority)

3. Descriptions of the Control Program

The target software of this application note is explained here.

3.1 Contents of Control

3.1.1 Motor Start/Stop

The start and stop of the motor are controlled by input from 'Renesas Motor Workbench' or SW1_1, SW1_2 switch of inverter board.

A general-purpose port is assigned to SW1_1, SW1_2. When the port is at a 'Low' level, the software determines that the motor should be started. Conversely, when the level is switched to 'High' level, the software determines that the motor should be stopped.

3.1.2 A/D Converter

(1) Motor Rotation Speed Reference

The motor rotation speed reference can be set by Renesas Motor Workbench input or A/D conversion of the VR1 output value (analog value). VR1 of inverter board A is used to determine the rotation speed of both motor A and motor B. The A/D converted VR1 value is used as rotation speed command value, as shown below.

Table 3-1 Conversion Ratio of the Rotation Speed Reference

Item		Conversion ratio (Reference: A/D conversion value)	Channel
Rotation speed reference	CW	0 rpm to 2650 rpm: 07FFH to 0000H	AN217
	CCW	0 rpm to 2650 rpm: 0800H to 0FFFH	

(2) Inverter Bus Voltage

Inverter bus voltage is measured as shown in the table below. It is used for modulation factor calculation and detection of over-voltage / under-voltage (when an abnormality is detected, PWM is stopped).

Table 3-2 Inverter Bus Voltage Conversion Ratio

Item	Conversion ratio (Inverter bus voltage: A/D conversion value)	Channel
Inverter bus voltage	0 [V] to 111 [V]: 0000H to 0FFFH	[Motor A] AN208 [Motor B] AN209

(3) U, W Phase Current

The U and W phase currents are measured as shown below and used for vector control.

Table 3-3 Conversion Ratio of U and W Phase Current

Item	Conversion ratio (U, W phase current: A/D conversion value)	Channel
U, W phase current	-12.5 [A] to 12.5 [A]: 0000H to 0E8BH* Current value = (5V-2.5V)/(0.01Ohm*20) = 12.5A In this system, the current detection circuit is level shifted from 5V to 3V, so 0E8BH is the upper limit of A/D conversion.	[Motor A] Iu: AN000 Iw: AN002 [Motor B] Iu: AN100 Iw: AN102

Note: * For more details of A/D conversion characteristics, refer to 'RX66T Group User's Manual:Hardware'

3.1.3 Modulation

The target software of this application note uses pulse width modulation (hereinafter called PWM) to generate the input voltage to the motor. And the PWM waveform is generated by the triangular wave comparison method.

(1) Triangular Wave Comparison Method

The triangular wave comparison method is used to output the voltage command value. By this method, the pulse width of the output voltage can be determined by comparing the carrier waveform (triangular wave) and voltage command value waveform. The voltage command value of the pseudo sinusoidal wave can be output by turning the switch on or off when the voltage command value is larger or smaller than the carrier wave respectively.
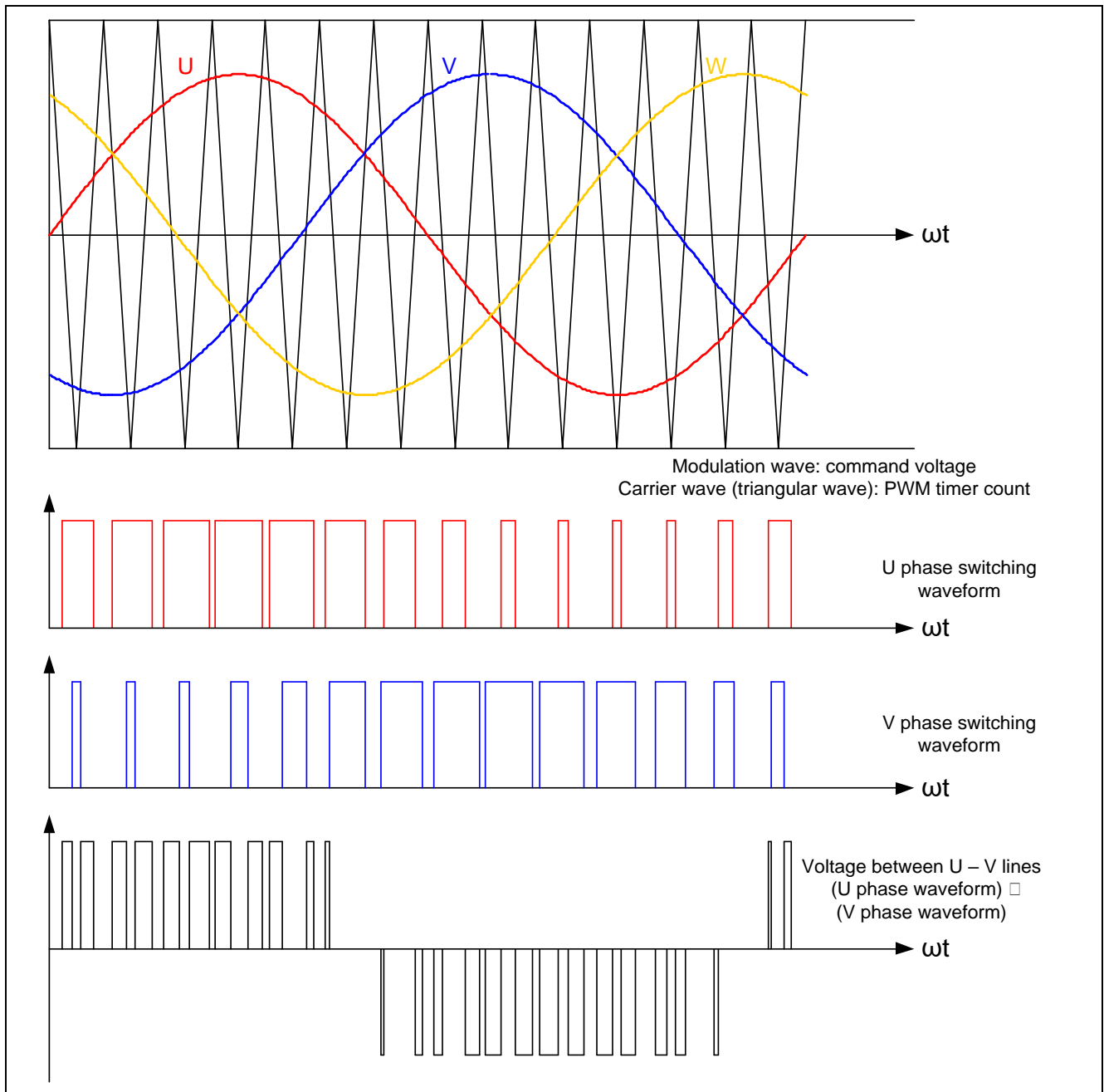

Figure 3-1   Conceptual Diagram of the Triangular Wave Comparison Method

As shown in Figure 3-2, ratio of the output voltage pulse to the carrier wave cycle is called duty.
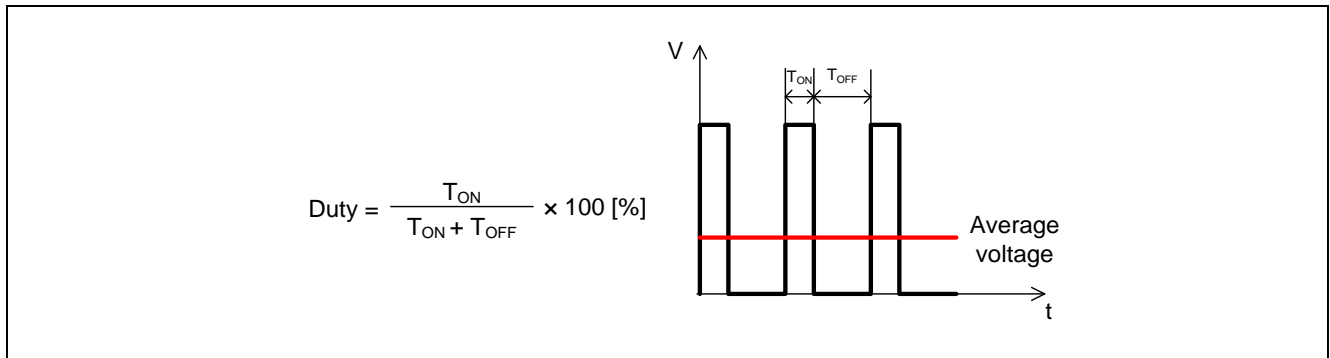

Figure 3-2   Definition of Duty

Modulation factor m is defined as follows.

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

m: Modulation factor V: Voltage command value E: Inverter bus voltage

The voltage command can be generated by setting PWM compare register properly to obtain the desired duty.

3.1.4 State Transition

Figure 3-3 is a state transition diagram of the sensorless vector control software. In the target software of this application note, the software state is managed by 'SYSTEM MODE' and 'RUN MODE'. And 'Control Config' shows the active control system in the software. Motor A and B are controlled in the same method.
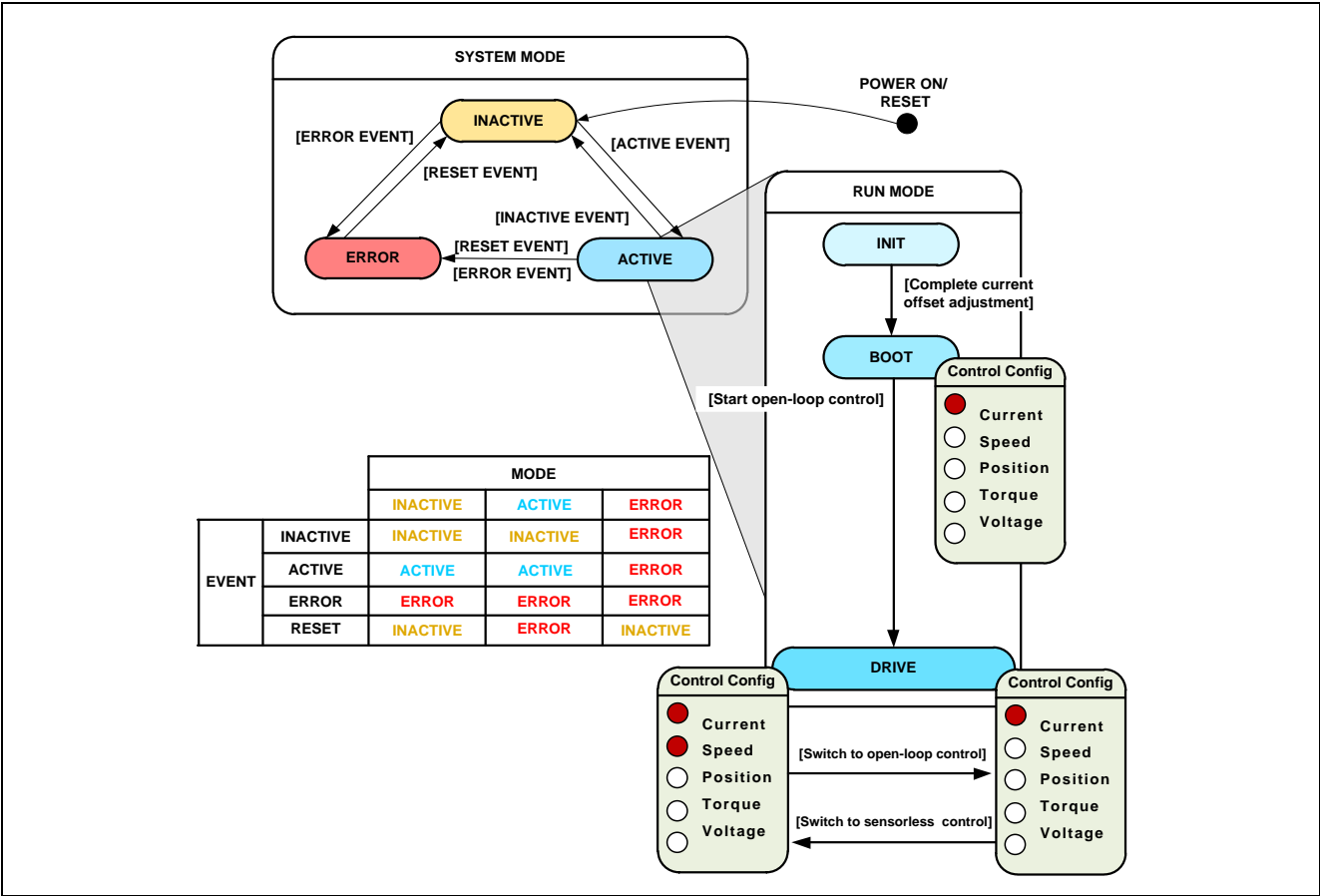

Figure 3-3   State Transition Diagram of Sensorless Vector Control Software

(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 (INACTIVE), motor drive (ACTIVE), and abnormal condition (ERROR).

(2) RUN MODE

'RUN MODE' indicates the condition of the motor control. 'RUN MODE' transits sequentially as shown in Figure 3-3 when 'SYSTEM MODE' is 'ACTIVE'.

(3) EVENT

When 'EVENT' occurs in each 'SYSTEM MODE', 'SYSTEM MODE' changes as shown in the table of Figure 3-3, according to that 'EVENT'.

Table 3-4 List of EVENT

EVENT name	Occurrence factor
INACTIVE	By user operation
ACTIVE	By user operation
ERROR	When the system detects an error
RESET	By user operation

3.1.5 Startup Method

Figure 3-4 shows startup control of sensorless vector control software. Each reference value setting of d-axis current, q-axis current and speed is managed by respective status. Motor A and B are controlled in the same method.
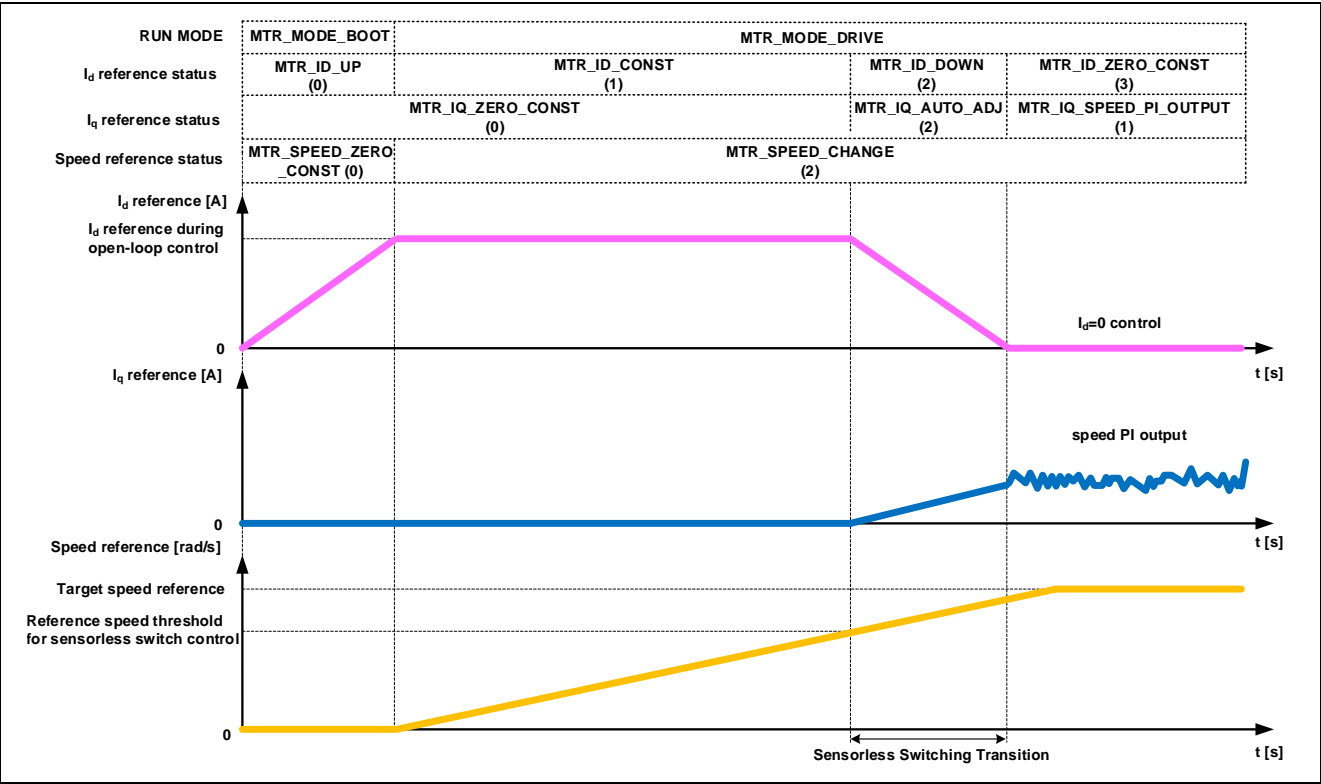

Figure 3-4   Startup Control of Sensorless Vector Control Software

3.1.6 System Protection Function

This control software has the following error status and executes emergency stop functions in case of occurrence of respective errors. Table 3-5 shows each setting value for the system protection function.

- Over-current error
  The over-current detection is performed by both hardware detection method and software detection method.
  The PWM output ports are set to high impedance state in response to an emergency stop signal (over current detection) from the hardware. In addition, U, V, and W phase currents are monitored in over-current monitoring cycle. When an over current (when the current exceeds the over current limit) is detected, the CPU executes emergency stop (software detection). When this error occurs, the CPU performs emergency stop the motor which the error occurred. The over-current limit value is calculated from the nominal current of the motor [MP_NOMINAL_CURRENT_RMS].

- Over-voltage error
- The inverter bus voltage is monitored in over voltage monitoring cycle. When an over-voltage is detected (when the voltage exceeds the over voltage limit), the CPU performs emergency stop. Here, the over-voltage limit is set in consideration of the error of resistance value of the detect circuit. When this error occurs, the CPU performs emergency stop the motor which the error occurred.

- Under-voltage error
  The inverter bus voltage is monitored in under-voltage monitoring cycle. The CPU performs emergency stop when under-voltage is detected (when voltage falls below the limit). Here, the under-voltage limit value is set in consideration of the error of resistance value of the detect circuit. When this error occurs, the CPU performs emergency stop the motor which the error occurred.

- Over-speed error
- The rotation speed is monitored in rotation speed monitoring cycle. The CPU performs emergency stop when the speed is over the limit. When this error occurs, the CPU performs emergency stop the motor which the error occurred.

Table 3-5   Setting Values of the System Protection Function

Error name	Threshold	
Over-current error	Over-current limit value [A]	0.89
	Monitoring cycle [μs]	100
Over-voltage error	Over-voltage limit value [V]	28
	Monitoring cycle [μs]	100
Under-voltage error	Under-voltage limit value [V]	14
	Monitoring cycle [μs]	100
Over-speed error	Speed limit value [rpm]	3000
	Monitoring cycle [μs]	100

3.1.7 Carrier synchronized interrupt

In the case of the two motors control, if the process is implemented in one interrupt, each control of the motors shall be sequential. Therefore, there will be a difference in the control of each motor A and motor B. To decrease the difference, the carrier synchronized interrupt will be generated in different timing for the motor A and motor B. In addition, by using the interrupt skipping function and generating an interrupt once every two cycles (100μs), motor A and motor B can perform interrupt processing alternately.

In this control program, it has been achieved by adjusting carrier cycle for the motor B.
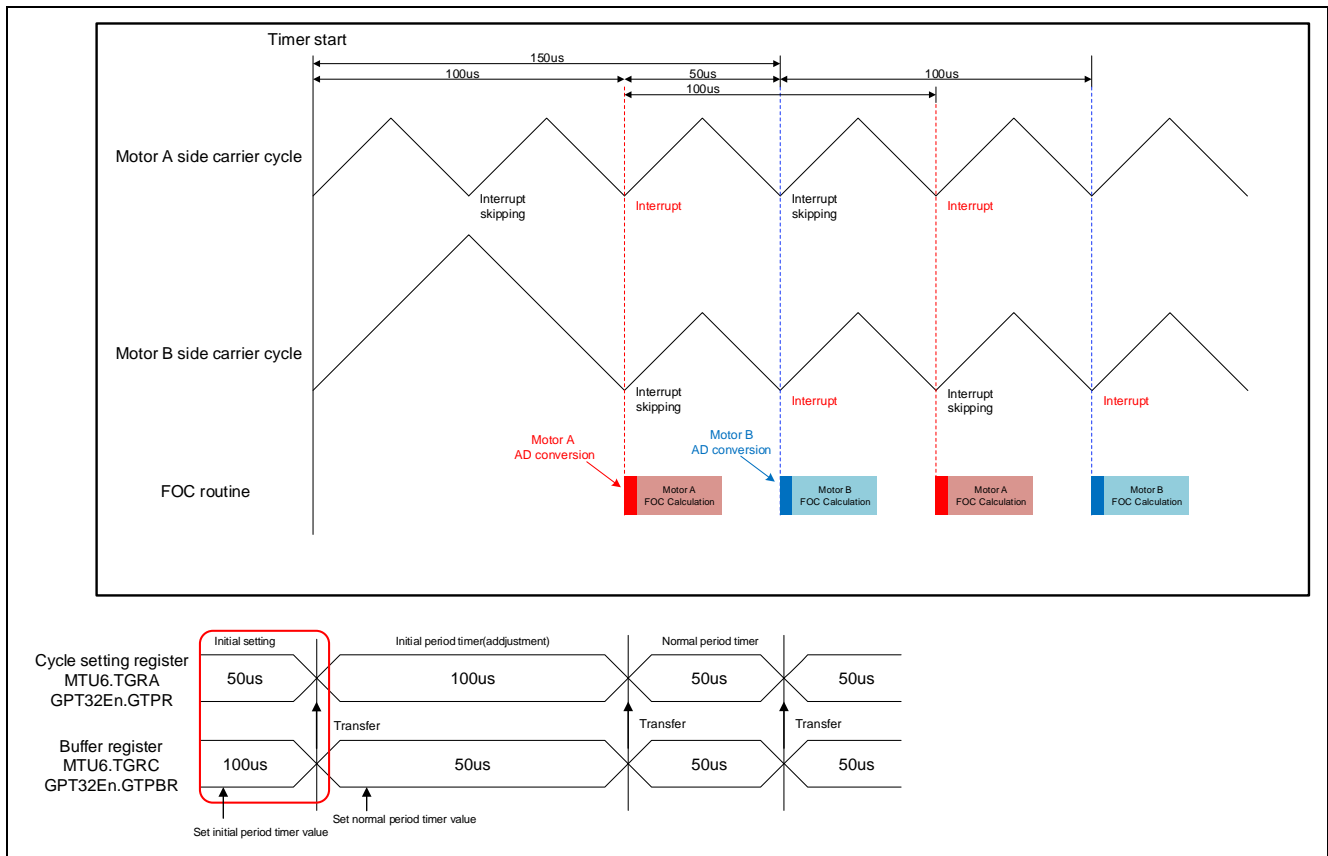

Figure 3-5 Start carrier output

By this setting, after carrier period 100us for the motor B has been completed at first and later the period of interrupt becomes 50us.Therefore, carrier synchronized interrupt will not occur at the same time. Carrier synchronous interrupts of motor A and motor B occur alternately at periods of 50us. Therefore, it is necessary to implement the processing time of interrupt (including processing time at error occurrence) within 50us.

3.2 Function Specifications of Sensorless Vector Control Software

The control process of the target software of this application note is mainly consisted of 100 [μs] period interrupt (carrier interrupt) and 1 [ms] period interrupt. In Figure 3-6 and Figure 3-7, the control process in the red broken line part is executed every 100 [μs] period, and the control process in the blue broken line part is executed every 1 [ms] period.
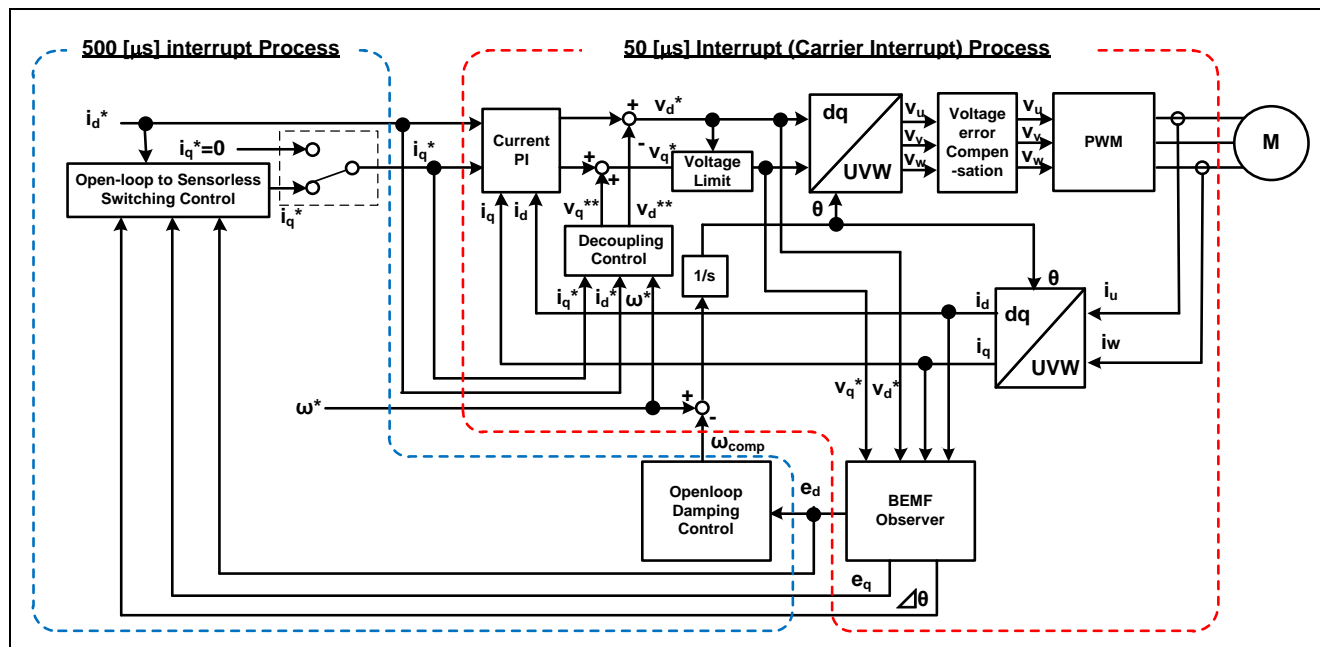

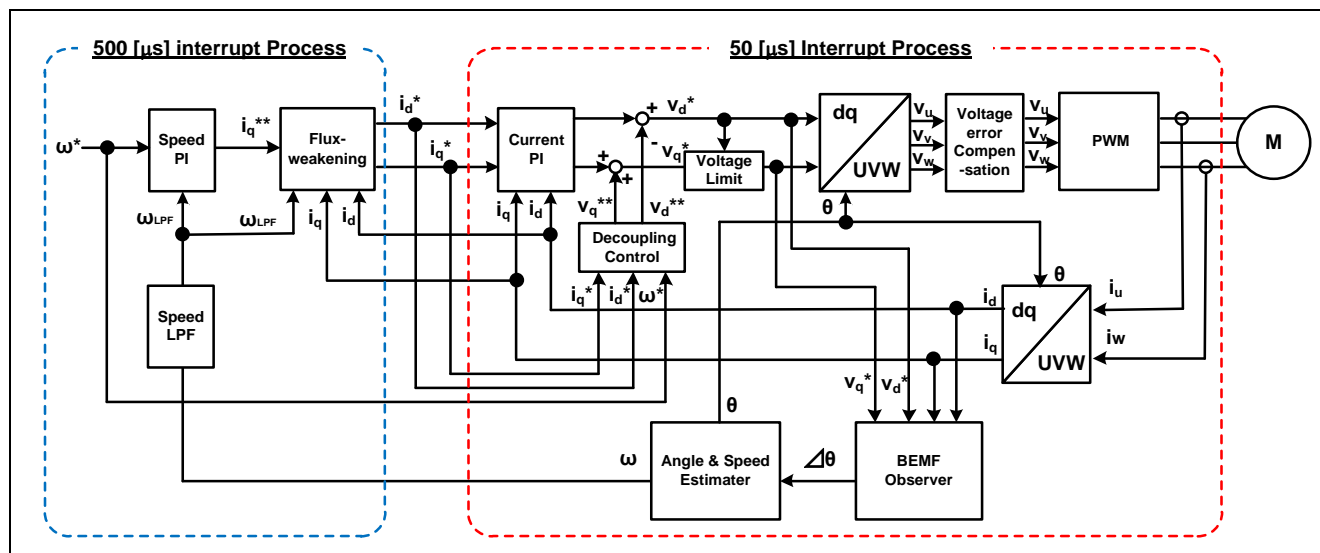**Figure 3-6   Block Diagram of Sensorless Vector Control (Open-loop Control)**

Figure 3-7   Block Diagram of Sensorless Vector Control (Sensorless Control)

This chapter shows the specification of 2 interrupt functions and functions executed in each interrupt period. In the following tables, only primary functions of the sensorless vector control are listed. Regarding the specification of functions not listed in the following tables, refer to source codes.

In a 100 [µs] period interrupts, different interrupt functions are called for each interrupt trigger of motor A and motor B, and mtr_foc_interrupt_carrier is called in each interrupt function. The pointer of the structure for motor A (st_foc_a) and motor B (st_foc_b) is passed as an argument when calling.

1 [ms] Period interrupt is common to motor A and motor B, and both motor A and motor B are processed by a single interrupt function.

Table 3-6   List of Interrupt Functions

File name	Function overview	Process overview
r_mtr_interrupt_ carrier.c	mtr_foc_interrupt_carrier Input: (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Calling every 100 [µs] • Current and Vdc monitoring • Current PI control • Speed/position estimation • PWM duty setting
r_mtr_interrupt_timer.c	mtr_foc_interrupt_1ms Input: (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Calling every 1 [ms] • Startup control • d-axis/q-axis current and speed reference setting • Speed PI control

Table 3-7   List of Functions Executed in 100 [μs] Period Interrupt (Carrier Interrupt) (1/2)

File name	Function overview	Process overview

r_mtr_ctrl_mrssk.c	mtr_get_current_iuiw
	Input: (float*) f4_iu_ad / U phase current A/D conversion value pointer
	        (float*) f4_iw_ad / W phase current A/D conversion value pointer
	        (uint8_t) u1_id / Motor ID
	Output: None
		Obtaining the U/W phase current

	mtr_get_vdc Input: (uint8_t) u1_id / Motor ID Output: (float) f4_temp_vdc / Vdc value	Obtaining the Vdc
r_mtr_foc_control_ less_speed.c	mtr_error_check Input: (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Error monitoring
	mtr_current_offset_adjustment Input: (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Cancel current offset
	mtr_calib_current_offset Input: (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Calculation of current offset
	mtr_angle_speed Input: (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Position and speed estimation
	mtr_foc_voltage_limit Input: (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Limiting voltage reference
r_mtr_foc_current.c	mtr_current_pi_control Input: (mtr_current_control_t *) st_cc / Structure pointer for current control Output: None	Current PI control

	mtr_foc_current_decoupling
	Input: (mtr_current_control_t *) st_cc / Structure pointer for current control
	        (float) f4_speed_rad / Rotation speed
	        (const mtr_parameter_t *) p_mtr / Structure pointer for motor parameter
	Output: None
		Decoupling control

r_mtr_transform.c	mtr_transform_uvw_dq_abs
	Input: (const mtr_rotor_angle_t *) p_angle / Structure pointer for phase management
	        (const float*) f4_uvw / UVV phase pointer
	        (float*) f4_dq / dq-axis pointer
	Output: None
		Coordinate transform UVW to dq

	mtr_transform_dq_uvw_abs
	Input: (const mtr_rotor_angle_t *) p_angle / Structure pointer for phase management
	        (const float*) f4_dq / dq-axis pointer
	        (float*) f4_uvw / UVW phase pointer
	Output: None
		Coordinate transform dq to UVW

Table 3-7   List of Functions Executed in 100 [μs] Period Interrupt (Carrier Interrupt) (2/2)

File name	Function overview	Process overview

r_mtr_volt_err_comp.obj	mtr_volt_err_comp_main
	Input: (mtr_volt_comp_t *) st_volt_comp / Structure pointer for voltage
	     error compensation
	     (float*) p_f4_v_array / Array pointer for 3-phase voltage
	     compensation amount
	     (float*) p_f4_i_array / Array pointer for 3 phase current
	     (float) f4_vdc / Inverter bus voltage
	Output: None
		Voltage error compensation

r_mtr_ctrl_rx66t.c	mtr_inv_set_uvw
	Input: (float) f4_duty_u / U phase modulation factor
	     (float) f4_duty_v / V phase modulation factor
	     (float) f4_duty_w / W phase modulation factor
	     (uint8_t) u1_id / Motor ID
	Output: None
		PWM output setting

r_mtr_bemf_observer.obj	mtr_bemf_observer
	Input: (mtr_bemf_observer_t *) st_bemf_obs / Structure pointer for
	     BEMF observer
	     (float) f4_vd_ref / d-axis voltage reference
	     (float) f4_vq_ref / q-axis voltage reference
	     (float) f4_id / d-axis current
	     (float) f4_iq / q-axis current
	Output: None
		Calculation for BEMF observer

	mtr_bemf_calc_d
	Input: (mtr_bemf_observer_t *) st_bemf_obs / Structure pointer for
	     BEMF observer
	     (float) f4_speed_rad / Estimated speed
	     (float) f4_iq / q-axis current
	Output: (float) f4_temp / Estimated d-axis BEMF
		Calculation for estimated d-axis BEMF

	mtr_bemf_calc_q
	Input: (mtr_bemf_observer_t *) st_bemf_obs / Structure pointer for
	     BEMF observer
	     (float) f4_speed_rad / Estimated speed
	     (float) f4_id / d-axis current
	Output: (float) f4_temp / Estimated q-axis BEMF
		Calculation for estimated q-axis BEMF

	mtr_angle_speed_pll
	Input: (mtr_pll_est_t *) st_pll_est / Structure pointer for position and
	     speed estimation
	     (float) f4_phase_err / Phase error
	     (float*) f4_speed / Estimated speed pointer
	Output: None
		Calculation for position and speed estimation

r_mtr_opl_damp_ctrl.obj	mtr_opl_damp_ctrl
	Input: (mtr_opl_damp_t *) st_opl_damp / Structure pointer for open-loop
	     damping control
	     (float) f4_ed / Estimated d-axis BEMF
	     (float) speed_ref / Speed reference
	Output: (float) f4_temp_damp_comp_speed / Feedback value for speed
	     reference
		Open-loop damping control

Table 3-8   List of Functions Executed in 1 [ms] Period Interrupt

File name	Function overview	Process overview
r_mtr_foc_control_less_speed.c	mtr_set_speed_ref Input: (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: (float) f4_speed_rad_ref_buff / Speed reference	Speed reference setting
	mtr_set_iq_ref Input: (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: (float) f4_iq_ref_buff / q-axis current reference	q-axis current reference setting
	mtr_set_id_ref Input: (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: (float) f4_id_ref_buff / d-axis current reference	d-axis current reference setting
r_mtr_foc_speed.c	mtr_speed_pi_control Input: (mtr_speed_control_t *) st_sc / Structure pointer for speed control (float) f4_speed_rad / Rotation speed Output: (float) f4_iq_ref_calc / q-axis current reference	Speed PI control

r_mtr_opl2less.obj	mtr_opl2less_iq_calc
	input: (float) f4_ed / Estimated d-axis BEMF
	    (float) f4_eq / Estimated q-axis BEMF
	    (float) f4_id / d-axis current reference when open-loop
	    (float) f4_torque_current / Torque current when open-loop control
	    (float) f4_phase_err / Phase error
	Output: (float) f4_temp_iq_ref / q-axis current reference
		Generating q-axis current reference for sensorless switching control

r_mtr_fluxwkn.obj	R_FLUXWKN_Run
	Input: (fluxwkn_t *) p_fluxwkn / Structure pointer for flux weakening control
	    (float) f4_speed_rad / Rotation speed
	    (const float*) p_f4_idq / dq-axis current pointer
	    (float*) p_f4_idq_ref / dq-axis current reference pointer
	Output: (uint16_t) u2_fw_status / Status of flux-weakening control
		Flux-weakening control

3.3 Macro Definition of Sensorless Vector Control Software

The macro definitions in the target software of this application note are listed below. In the following tables, only definitions set the software configuration are listed. Regarding the macro definitions not listed in the following tables, refer to source codes.

Macro with '_A' in Macro name are for motor A control. Macros with "_B" in the macro name are for motor B control, so use the appropriate macro.

Table 3-9   List of Macro Definitions 'r_mtr_motor_parameter.h'

File name	Macro name	Definition value	Remarks
r_motor_parameter.h	MP_POLE_PAIRS_A	2	Number of pole pairs
	MP_MAGNETIC_FLUX_A	0.02159f	Flux [Wb]
	MP_RESISTANCE_A	8.5f	Resistance [Ω]
	MP_D_INDUCTANCE_A	0.0045f	d-axis inductance [H]
	MP_Q_INDUCTANCE_A	0.0045f	q-axis inductance [H]
	MP_ROTOR_INERTIA_A	0.0000028f	Rotor inertia [kgm^2]
	MP_NOMINAL_CURRENT_RMS_A	0.42f	Nominal current [A(rms)]

Table 3-10   List of Macro Definitions 'r_mtr_control_parameter.h'

File name	Macro name	Definition value	Remarks
r_mtr_control_parameter.h	CP_CURRENT_OMEGA_A	300.0f	Natural frequency of current control system [Hz]
	CP_CURRENT_ZETA_A	1.0f	Damping ratio of current control system
	CP_SPEED_OMEGA_A	1.5f	Natural frequency of speed control system [Hz]
	CP_SPEED_ZETA_A	1.0f	Damping ratio of speed control system
	CP_E_OBS_OMEGA_A	1000.0f	Natural frequency of BEMF estimation system [Hz]
	CP_E_OBS_ZETA_A	1.0f	Damping ratio of BEMF estimation system
	CP_PLL_EST_OMEGA_A	20.0f	Natural frequency of position estimation system [Hz]
	CP_PLL_EST_ZETA_A	1.0f	Damping ratio of position estimation system
	CP_ID_DOWN_SPEED_RPM_A	600	Speed (mechanical) when start decreasing d-axis current reference [rpm]
	CP_ID_UP_SPEED_RPM_A	300	Speed (mechanical) when start increasing d-axis current reference [rpm]
	CP_MAX_SPEED_RPM_A	2650	Maximum speed (mechanical) [rpm]
	CP_OVERSPEED_LIMIT_RPM_A	3000	Speed limit value (mechanical) [rpm]
	CP_OL_ID_REF_A	0.3f	d-axis current reference in open-loop mode [A]

Table 3-11 List of Macro Definitions 'r_mtr_inverter_parameter.h'

File name	Macro name	Definition value	Remarks
r_mtr_inverter_parameter.h	IP_DEADTIME	2.0f	Dead time [μs]
	IP_CURRENT_RANGE	25.0f	Current A/D conversion range [A] (-12.5 [A] ~ 12.5 [A])
	IP_VDC_RANGE	111.0f	Vdc A/D conversion range [V] (0 [V] ~ 111 [V])
	IP_INPUT_V	24.0f	Input DC voltage [V]
	IP_CURRENT_LIMIT	10.0f	Over-current limit [A]*
	IP_OVERVOLTAGE_LIMIT	28.0f	High voltage limit [V]
	IP_UNDERVOLTAGE_LIMIT	14.0f	Low voltage limit [V]

Note: * This value is calculated from the rated power of the shunt resistance.

Table 3-12   List of Macro Definitions 'r_mtr_config.h'

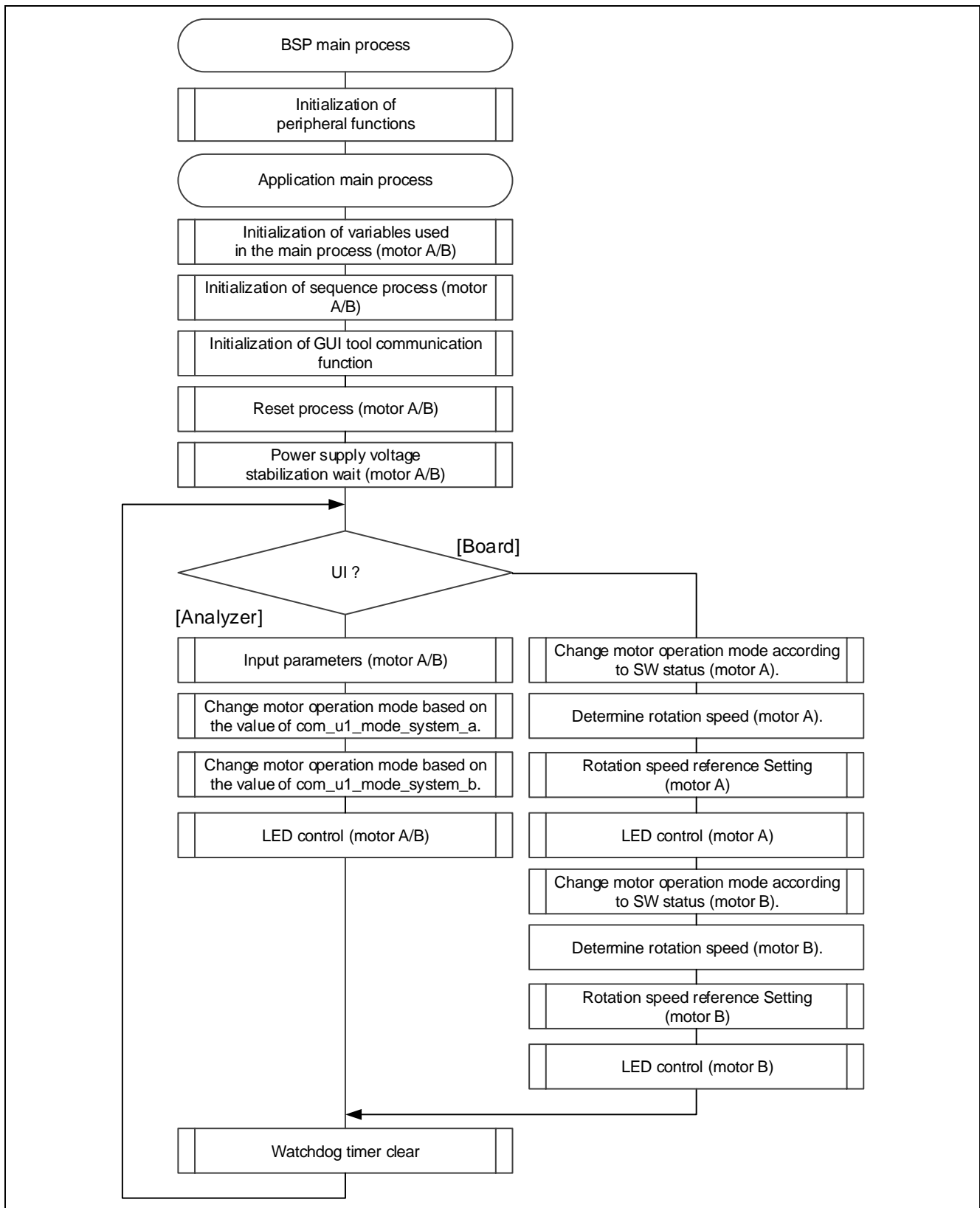| File name	Macro name	Definition value	Remarks
r_mtr_config.h	RX66T_MRSSK	—	MCU select macro
	IP_MRSSK	—	Inverter select macro
	MP_TG55L	—	Motor select macro
	CP_TG55L	—	
	CONFIG_DEFAULT_UI	ICS_UI	Default UI selection ICS_UI: Use Analyzer UI BOARD_UI: Board UI
	FUNC_ON	1	Enable
	FUNC_OFF	0	Disable
	DEFAULT_LESS_SWITCH_A	1	Sensorless switching control 0: Disable 1: Enable
	DEFAULT_FLUX_WEAKENING_A	0	Flux weakening control 0: Disable 1: Enable
	DEFAULT_VOLT_ERR_COMP_A	1	Voltage error compensation 0: Disable 1: Enable
	DEFAULT_OPENLOOP_DAMPING_A	1	Open-loop damping control 0: Disable 1: Enable
	GAIN_MODE_A	MTR_GAIN_DESIGN_MODE	Gain mode MTR_GAIN_DESIGN_MOD: PI gain design mode MTR_GAIN_DIRECT_MODE: PI gain direct input mode
	MOD_METHOD_A	MOD_METHOD_SVPWM	Modulation method MOD_METHOD_SPWM: Sinusoidal PWM MOD_METHOD_SVPWM: Space Vector PWM

3.4 Control Flowcharts

3.4.1 Main Process

```
                        ┌─────────────────────┐
                        │   BSP main process  │
                        └─────────────────────┘
                        ┌─────────────────────┐
                        │   Initialization of │
                        │ peripheral functions│
                        └─────────────────────┘
                        ┌─────────────────────┐
                        │ Application main process │
                        └─────────────────────┘
                        ┌─────────────────────┐
                        │ Initialization of variables used │
                        │ in the main process (motor A/B) │
                        └─────────────────────┘
                        ┌─────────────────────┐
                        │ Initialization of sequence process (motor A/B) │
                        └─────────────────────┘
                        ┌─────────────────────┐
                        │ Initialization of GUI tool communication function │
                        └─────────────────────┘
                        ┌─────────────────────┐
                        │ Reset process (motor A/B) │
                        └─────────────────────┘
                        ┌─────────────────────┐
                        │ Power supply voltage stabilization wait (motor A/B) │
                        └─────────────────────┘
```

BSP main process
- Initialization of peripheral functions

Application main process
- Initialization of variables used in the main process (motor A/B)
- Initialization of sequence process (motor A/B)
- Initialization of GUI tool communication function
- Reset process (motor A/B)
- Power supply voltage stabilization wait (motor A/B)

UI ?

[Board]
- Change motor operation mode according to SW status (motor A).
- Determine rotation speed (motor A).
- Rotation speed reference Setting (motor A)
- LED control (motor A)
- Change motor operation mode according to SW status (motor B).
- Determine rotation speed (motor B).
- Rotation speed reference Setting (motor B)
- LED control (motor B)

[Analyzer]
- Input parameters (motor A/B)
- Change motor operation mode based on the value of com_u1_mode_system_a.
- Change motor operation mode based on the value of com_u1_mode_system_b.
- LED control (motor A/B)

Watchdog timer clear

Figure 3-8   Main Process Flowchart

3.4.2 100 [µs] Period Interrupt (Carrier synchronized interrupt) Process

Motor A and motor B have different interrupt functions, but since the common function is called from each interrupt function, the processing flow is the same.

The parameter difference between motor A and motor B is commonalized by passing it to interrupt processing as an argument.
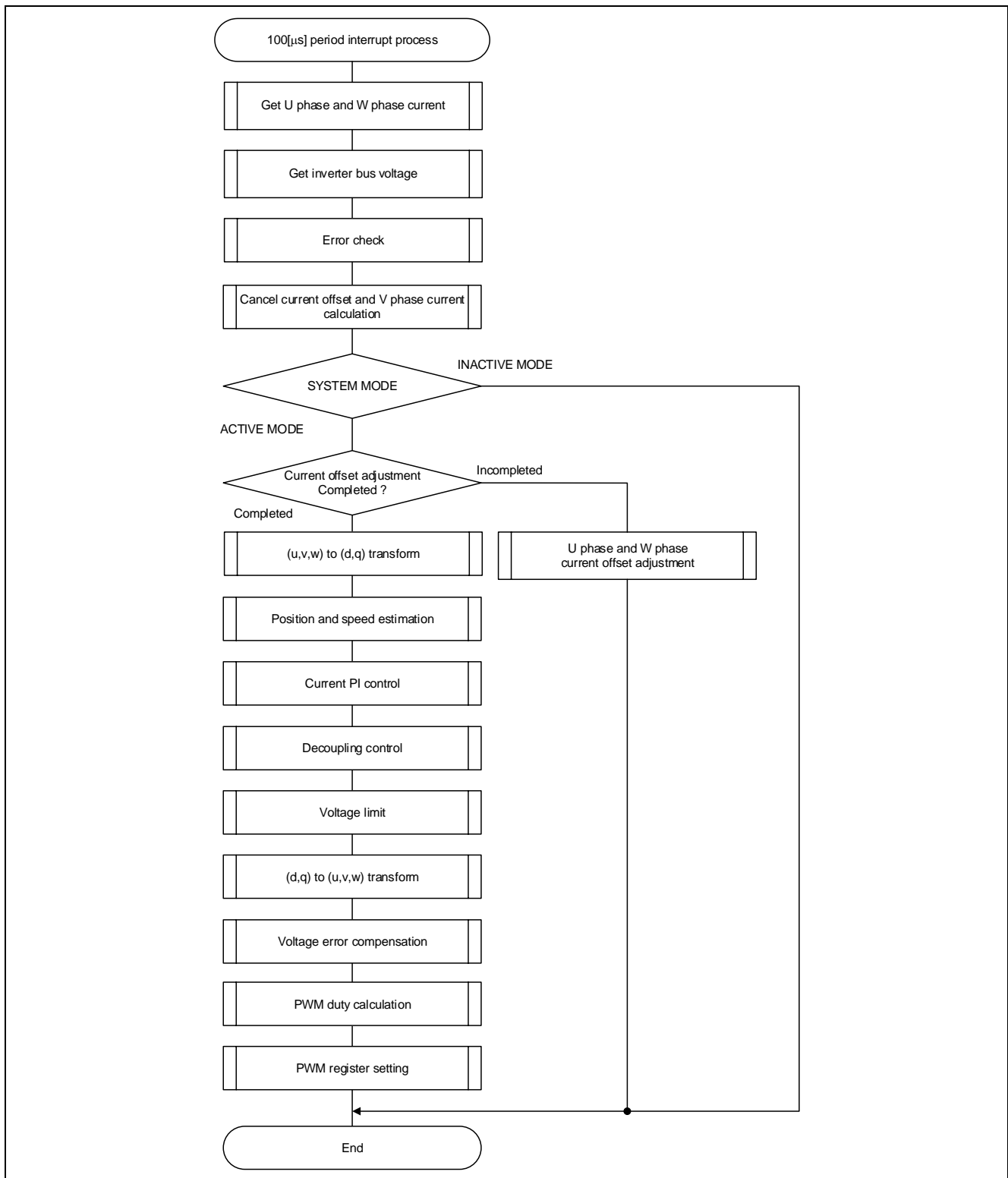

Figure 3-9 100 [µs] Period Interrupt Process Flowchart
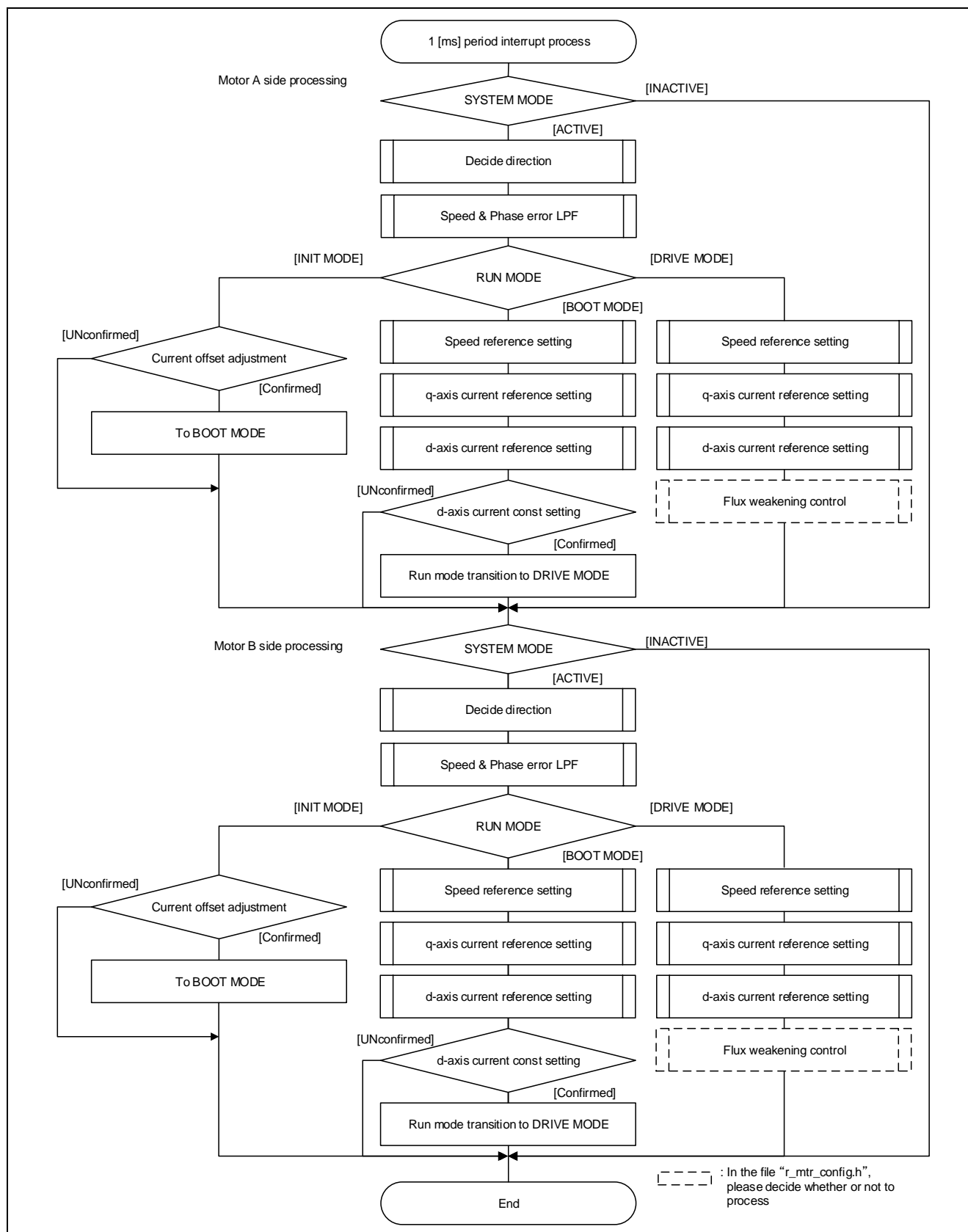
3.4.3　1 [ms] Period Interrupt Process

Figure 3-10 1[ms] Period Interrupt Process Flowchart

3.4.4 Over-Current Detection Interrupt Process

The over-current detection interrupt occurs when POE0#, POE4# pin detects falling-edge or when output levels of the MTU complementary PWM output pins are compared and simultaneous active-level output continues for one cycle or more. Therefore, when this interrupt process is executed, PWM output pins are already in high-impedance state and the output to the motor is stopped.

This interrupt processing is performed on the motor that generated the interrupt.
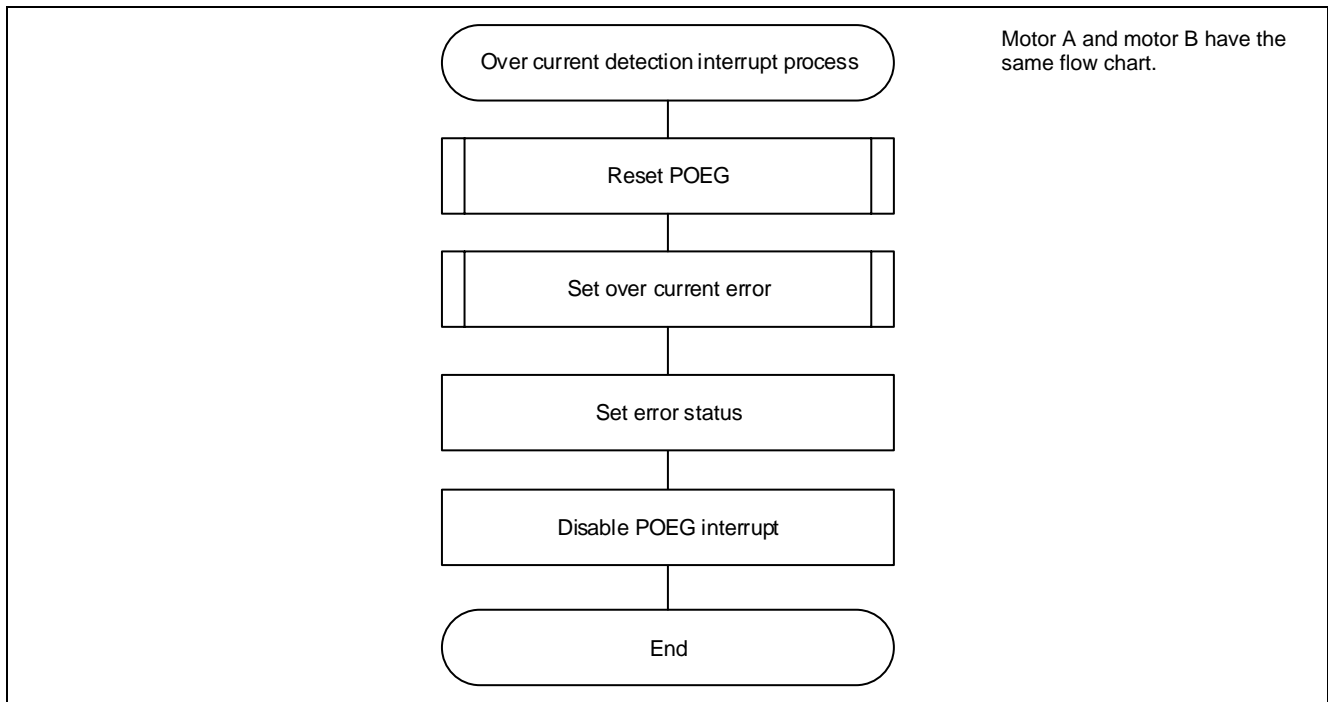
Figure 3-11 Over-Current Detection Interrupt Process Flowchart

4. How to build Evaluation environment

4.1 Importing the Demo Project

The sample application provided with this document may be imported into e²studio using the steps in this section. This operation is not required for the CS+ version.

1. Select File → Import.
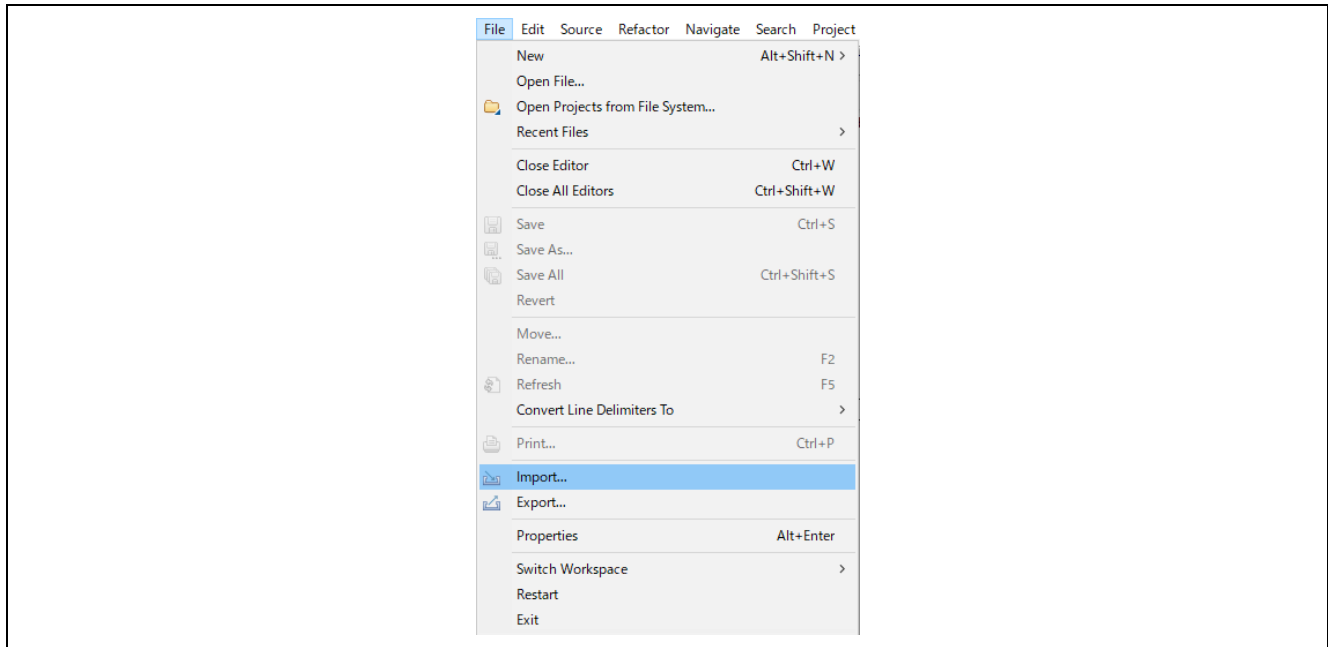
Figure 4-1 File Menu

2. Select "Existing Projects into Workspace".
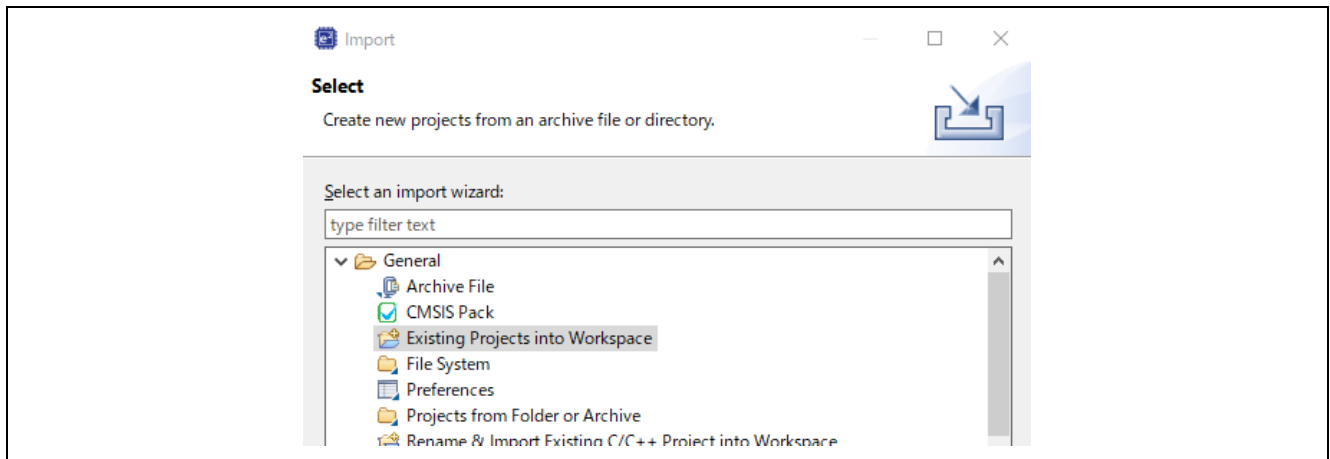
Figure 4-2 Import Wizard Selection

3. Click "Browse…" button and select the demo project. Click Finish button and the demo project is imported.
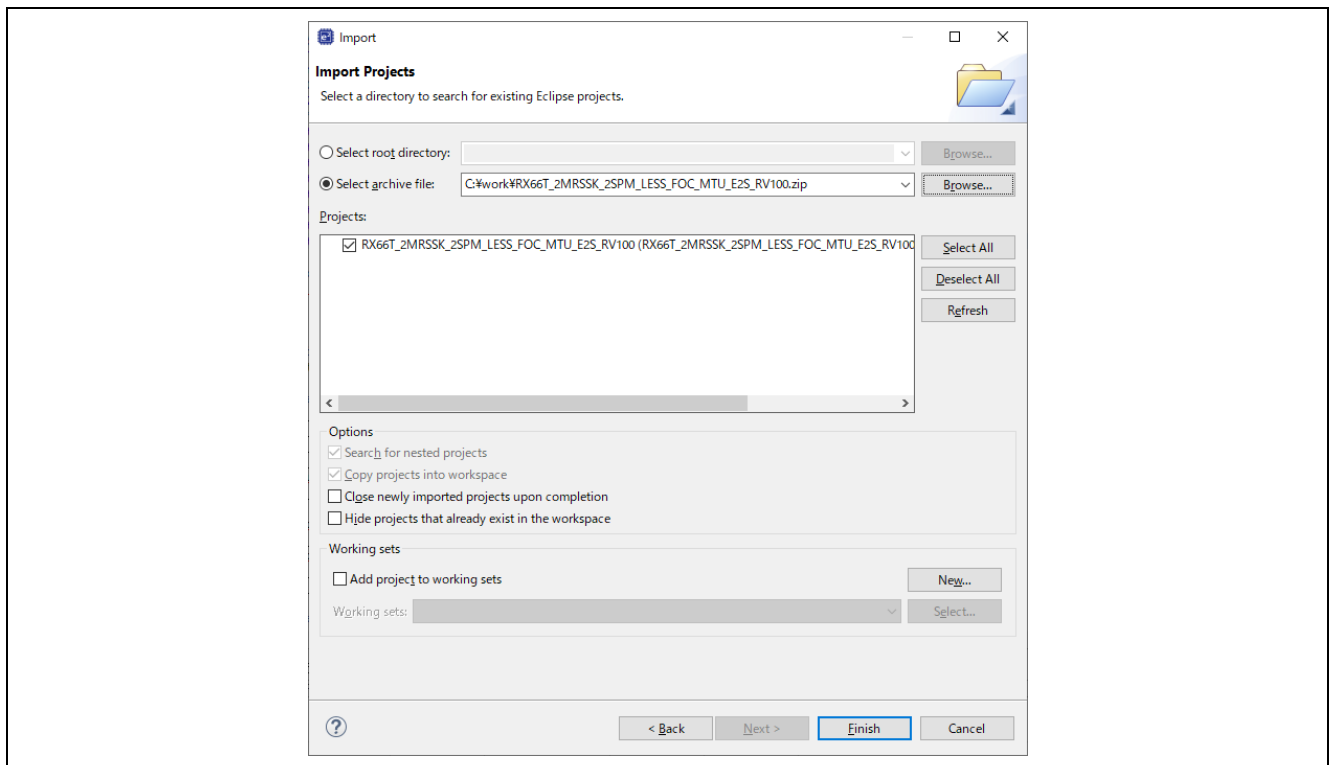
Figure 4-3 Import Projects

4.2   Building and Debugging

For the e2 studio version,   Refer to " e^{2}studio Getting Started Guide (R20UT4204)", for the CS + version,
Refer to "CS+ V8.06.00 Integrated Development Environment User's Manual: RX Debug Tool (R20UT4977)"
"CS+ Integrated Development Environment User's Manual: CC-RX Build Tool Operation (R20UT3478)".

4.3   Motor Control Development Support Tool 'Renesas Motor Workbench'

4.3.1   Overview

In the target software of this application note, the motor control development support tool "Renesas Motor Workbench" is used as an user interface (rotating/stop motor, set rotation speed reference, etc). Please refer to 'Renesas Motor Workbench User's Manual' for usage and more details.

You can find 'Renesas Motor Workbench' on Renesas Electronics Corporation website.
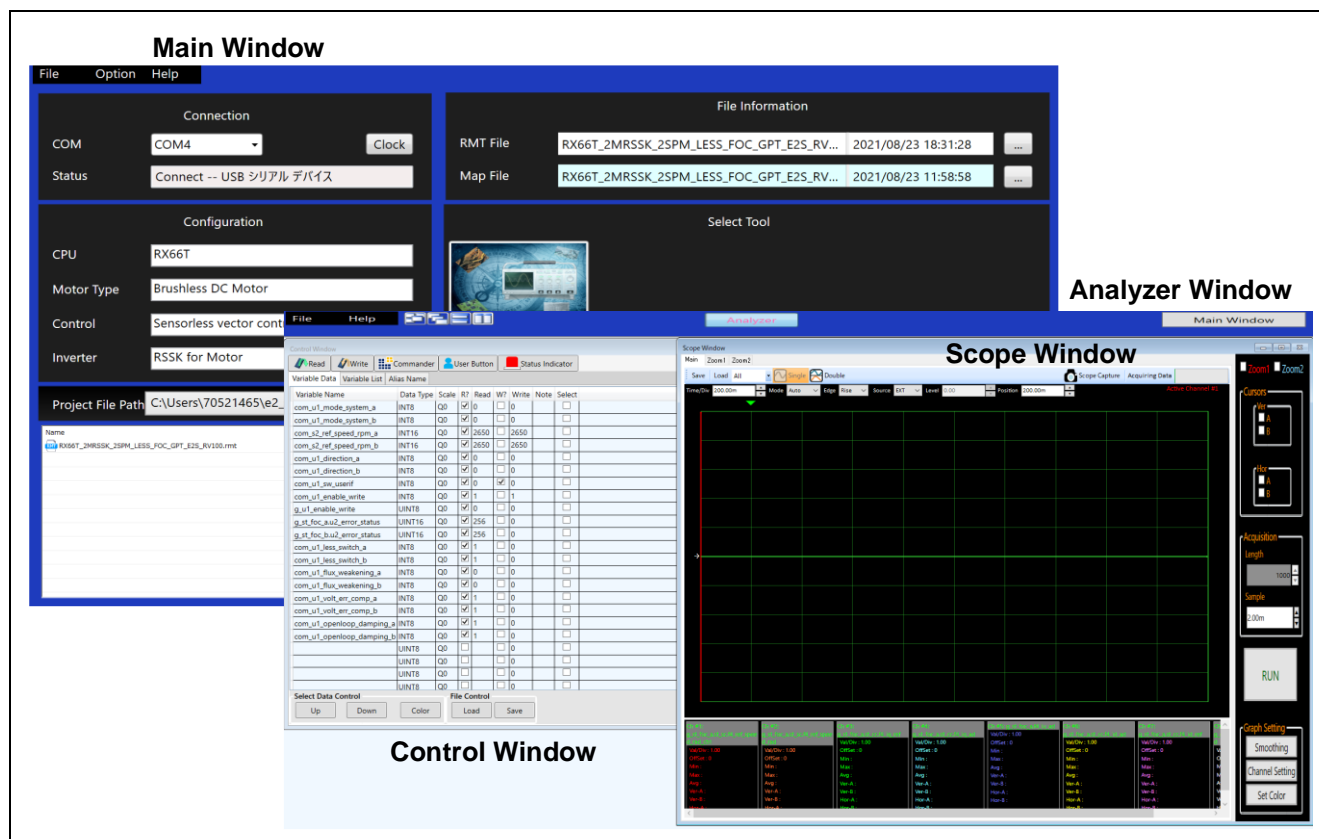
Figure 4-4   Renesas Motor Workbench – Appearance

Set up for 'Renesas Motor Workbench'

(1) Start 'Renesas Motor Workbench' by clicking this icon.

(2) Drop down menu [File] → [Open RMT File(O)].

   And select RMT file in '[Project Folder]/src/application/user_interface/ics/'.

(3) Use the 'Connection' [COM] select menu to choose the COM port.

(4) Click the Analyzer button of Select Tool to activate Analyzer function.

(5) Please refer to '4.3.3 Operation Example for Analyzer' for motor driving operation.

4.3.2 List of Variables for Analyzer function

Table 4-1 is a list of variables for Analyzer. These variables are reflected to the corresponding variables in 'Middle Layer' when the same value as of g_u1_enable_write is written to com_u1_enable_write. However, note that variables with (*) do not depend on com_u1_enable_write.

Variables with '_a' in variable name are for motor A control. When controlling motor B, use variables with '_b' in variable name. In Table 4-1, only variables for motor A control are shown.

Table 4-1 List of Variables for Analyzer (1/2)

Variable name	Type	Content
com_u1_sw_userif (*)	uint8_t	User interface switch 0: ICS user interface use (default) 1: Board user interface use
com_u1_mode_system_a (*)	uint8_t	State management 0: Stop mode, 1: Run mode, 3: Reset
com_u1_direction_a	uint8_t	Rotation direction 0: CW 1: CCW
com_s2_ref_speed_rpm_a	int16_t	Speed reference (Mechanical) [rpm]
com_u2_mtr_pp_a	uint16_t	Number of pole pairs
com_f4_mtr_r_a_a	float	Resistance [Ω]
com_f4_mtr_ld_a	float	d-axis Inductance [H]
com_f4_mtr_lq_a	float	q-axis Inductance [H]
com_f4_mtr_m_a	float	Flux [Wb]
com_f4_mtr_j_a	float	Inertia [kgm^2]
com_u2_offset_calc_time_a	uint16_t	Current offset value calculation time [ms]
com_f4_limit_speed_change_a	float	Speed limit change rate (Electrical) [rpm]
com_u2_max_speed_rpm_a	uint16_t	Maximum speed value (Mechanical) [rpm]
com_u2_id_up_speed_rpm_a	uint16_t	Speed when start increasing d-axis current reference (Mechanical) [rpm]
com_f4_id_up_time_a	float	Ramping up time of d-axis current reference [ms]
com_f4_ref_id_a	float	d-axis current reference in open loop mode [A]
com_u2_id_down_speed_rpm_a	uint16_t	Speed when start decreasing d-axis current reference (Mechanical) [rpm]
com_f4_id_down_time_a	float	Decreasing time of d-axis current reference [ms]
com_f4_speed_omega_a	float	Natural frequency of speed control system [Hz]
com_f4_speed_zeta_a	float	Damping ratio of speed control system
com_f4_current_omega_a	float	Natural frequency of current control system [Hz]
com_f4_current_zeta_a	float	Damping ratio of current control system
com_f4_e_obs_omega_a	float	Natural frequency of BEMF estimation system [Hz]
com_f4_e_obs_zeta_a	float	Damping ratio of BEMF estimation system
com_f4_pll_est_omega_a	float	Natural frequency of position estimation system [Hz]
com_f4_pll_est_zeta_a	float	Damping ratio of position estimation system
com_f4_id_kp_a	float	d-axis current PI control proportional gain
com_f4_id_ki_a	float	d-axis current PI control Integral gain
com_f4_iq_kp_a	float	q-axis current PI control proportional gain
com_f4_iq_ki_a	float	q-axis current PI control Integral gain
com_f4_speed_kp_a	float	Speed PI control proportional gain
com_f4_speed_ki_a	float	Speed PI control Integral gain
com_u2_overspeed_limit_rpm_a	uint16_t	Over-speed limit value (Mechanical) [rpm]
com_f4_nominal_current_rms_a	float	Nominal current [A(rpm)]
com_f4_switch_phase_err_deg_a	float	Phase error enabled switching to sensorless control (Electrical) [deg]
com_f4_opl2less_sw_time_a	float	Process time of sensorless switching control [s]

Table 4-1   List of Variables for Analyzer (2/2)

Variable name	Type	Content
com_f4_ed_hpf_omega_a	float	d-axis BEMF HPF cut-off frequency [Hz]
com_f4_ol_damping_zeta_a	float	Damping ratio of open-loop damping control
com_f4_ol_damp_fb_limit_rate_a	float	Feedback limit of open-loop damping control
com_f4_phase_err_lpf_cut_freq_a	float	Phase error LPF cut-off frequency [Hz]
com_u1_less_switch_a	uint8_t	Sensorless switching control 0: ON 1: OFF
com_u1_flux_weakening_a	uint8_t	Flux weakening control 0: ON 1: OFF
com_u1_volt_err_comp_a	uint8_t	Voltage error compensation 0: ON 1: OFF
com_u1_openloop_damping_a	uint8_t	Open-loop damping control 0: ON 1: OFF
com_u1_enable_write	uint8_t	Enable to rewriting variables (when the same values as of g_u1_enable_write is written)

Next, the primary variables that are frequently observed during the motor driving evaluation are listed in Table 4-2. Please refer when using Analyzer function. Regarding variables not listed in Table 4-2, refer to source codes.

Variables with 'A' in variable name are for motor A control. When controlling motor B, use variables with 'B' in variable name. InTable 4-2, only variables for motor A control are shown.

Table 4-2   List of Primary Variables for Sensorless Vector Control

Name of primary variables	Type	Content
g_st_foc_a.st_cc.f4_id_ref	float	d-axis current reference [A]
g_st_foc_a.st_cc.f4_id_ad	float	d-axis current [A]
g_st_foc_a.st_cc.f4_iq_ref	float	q-axis current reference [A]
g_st_foc_a.st_cc.f4_iq_ad	float	q-axis current [A]
g_st_foc_a.f4_iu_ad	float	U phase current A/D conversion value [A]
g_st_foc_a.f4_iv_ad	float	V phase current A/D conversion value [A]
g_st_foc_a.f4_iw_ad	float	W phase current A/D conversion value [A]
g_st_foc_a.st_cc.f4_vd_ref	float	d-axis output voltage reference [V]
g_st_foc_a.st_cc.f4_vq_ref	float	q-axis output voltage reference [V]
g_st_foc_a.f4_refu	float	U phase voltage reference [V]
g_st_foc_a.f4_refv	float	V phase voltage reference [V]
g_st_foc_a.f4_refw	float	W phase voltage reference [V]
g_st_foc_a.f4_modu	float	U phase modulation factor
g_st_foc_a.f4_modv	float	V phase modulation factor
g_st_foc_a.f4_modw	float	W phase modulation factor
g_st_foc_a.f4_ed	float	Estimated d-axis BEMF [V]
g_st_foc_a.f4_eq	float	Estimated q-axis BEMF [V]
g_st_foc_a.st_rotor_angle.f4_rotor_angle_rad	float	Estimated position (Electrical) [rad]
g_st_foc_a.st_sc.f4_ref_speed_rad_ctrl	float	Speed reference (Electrical) [rad/s]
g_st_foc_a.st_sc.f4_speed_rad	float	Estimated speed (Electrical) [rad/s]
g_st_foc_a.f4_phase_err_rad	float	Phase error (Electrical) [rad]
g_st_foc_a.u2_error_status	uint16_t	Error status

4.3.3 Operation Example for Analyzer

The section shows an example below for motor driving operation using Analyzer. Operation is using 'Control Window' of Analyzer. Regarding specification of 'Control Window', refer to 'Renesas Motor Workbench User's Manual'.

- Driving the motor
  (1) Confirm the check-boxes of column [W?] for 'com_u1_mode_system_a', 'com_s2_ref_speed_rpm_a', 'com_u1_enable_write' marks.
  (2) Input a reference speed value in the [Write] box of 'com_s2_ref_speed_rpm_a'.
  (3) Click the 'Write' button.
  (4) Click the 'Read' button. Confirm the [Read] box of 'com_s2_ref_speed_rpm_a', 'g_u1_enable_write'.
  (5) Set a same value of 'g_u1_enable_write' in the [Write] box of 'com_u1_enable_write'.
  (6) Write '1' in the [Write] box of 'com_u1_mode_system_a'.
  (7) Click the 'Write' button.
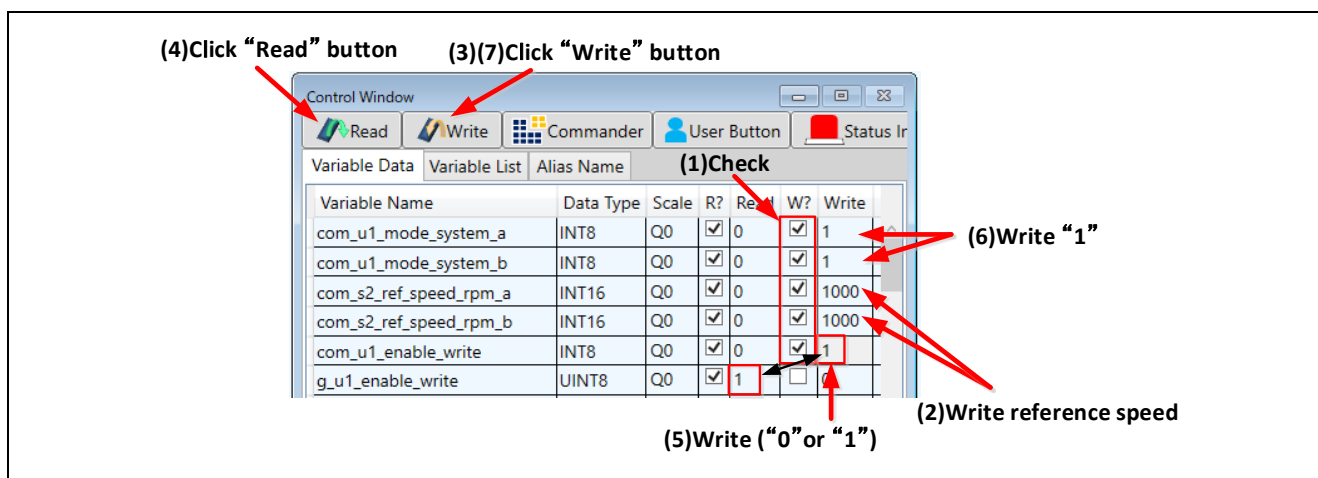
Figure 4-5   Procedure – Driving the Motor

- Stop the motor
  (1) Write '0' in the [Write] box of 'com_u1_mode_system_a'
  (2) Click the 'Write' button.
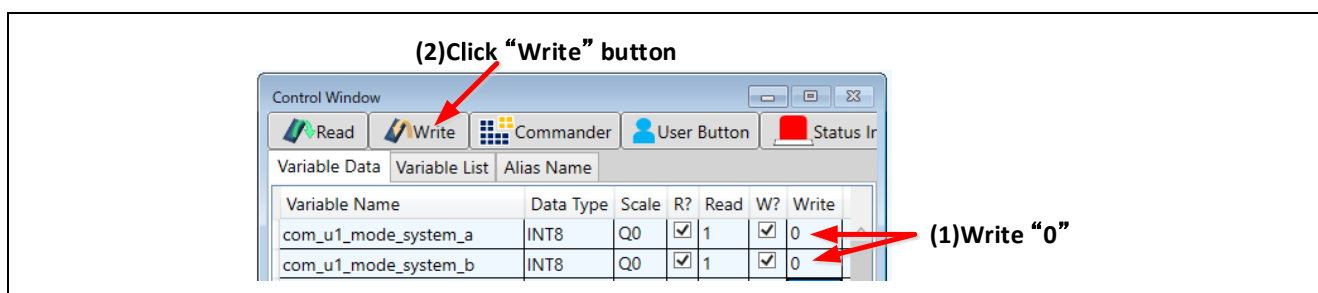
Figure 4-6   Procedure – Stop the Motor

- Error cancel operation
  - (1) Write '3' in the [Write] box of 'com_u1_mode_system_a'
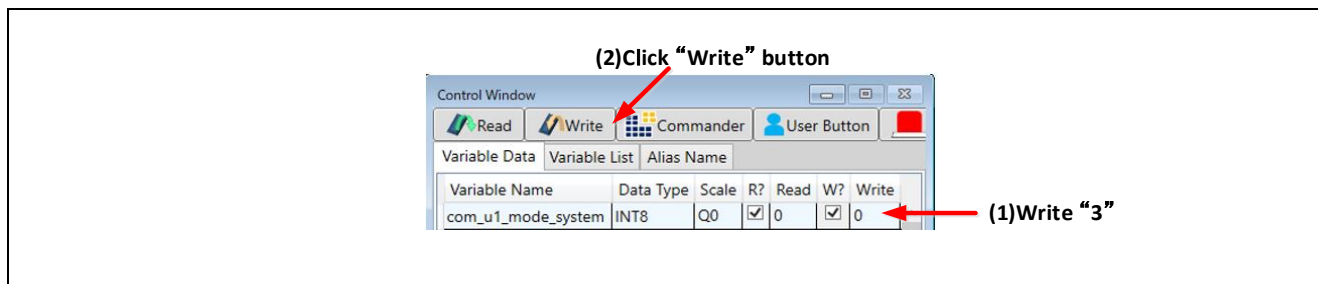 - (2) Click the 'Write' button.

Figure 4-7   Procedure – Error Cancel Operation

4.3.4   Operation Example for User Button

The section shows an example below for motor driving operation using User Button.

- Driving or Stop the motor
  By setting as shown in Figure 4-8, driving and stopping change each time the button is pressed.
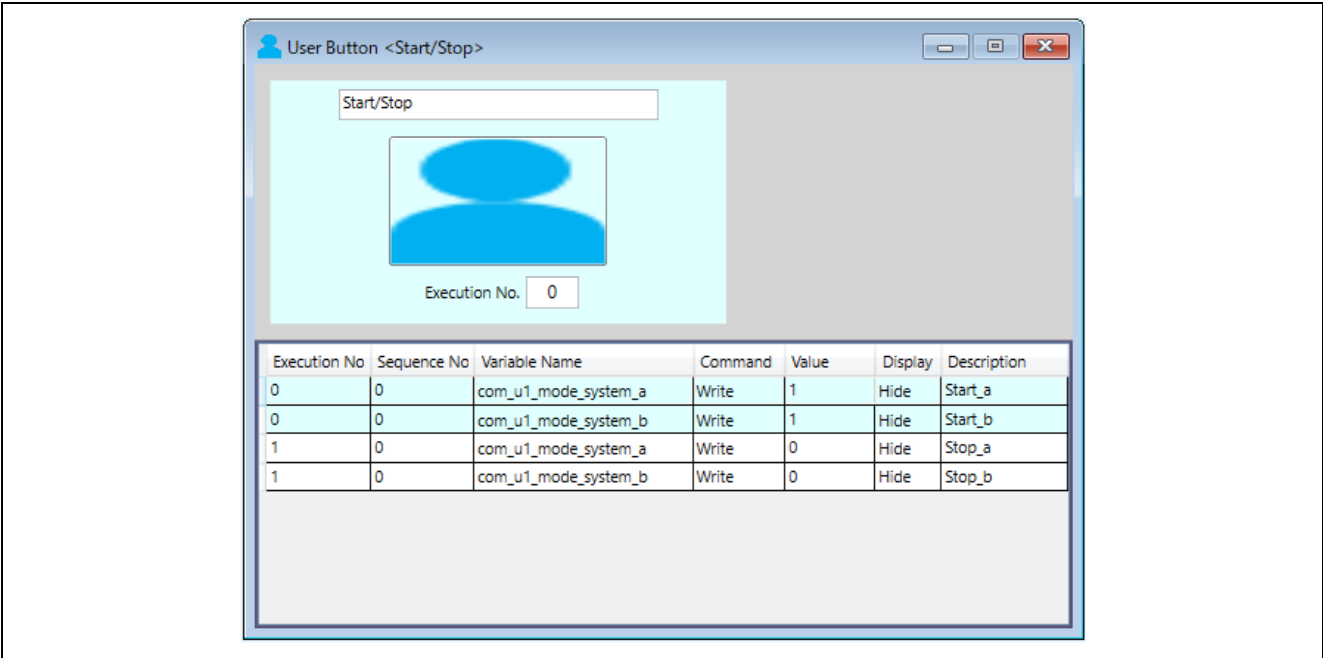

Figure 4-8   Driving or Stop the Motor

- Change speed
  By setting as shown in Figure 4-9, enter the command speed and press the button to change the speed.
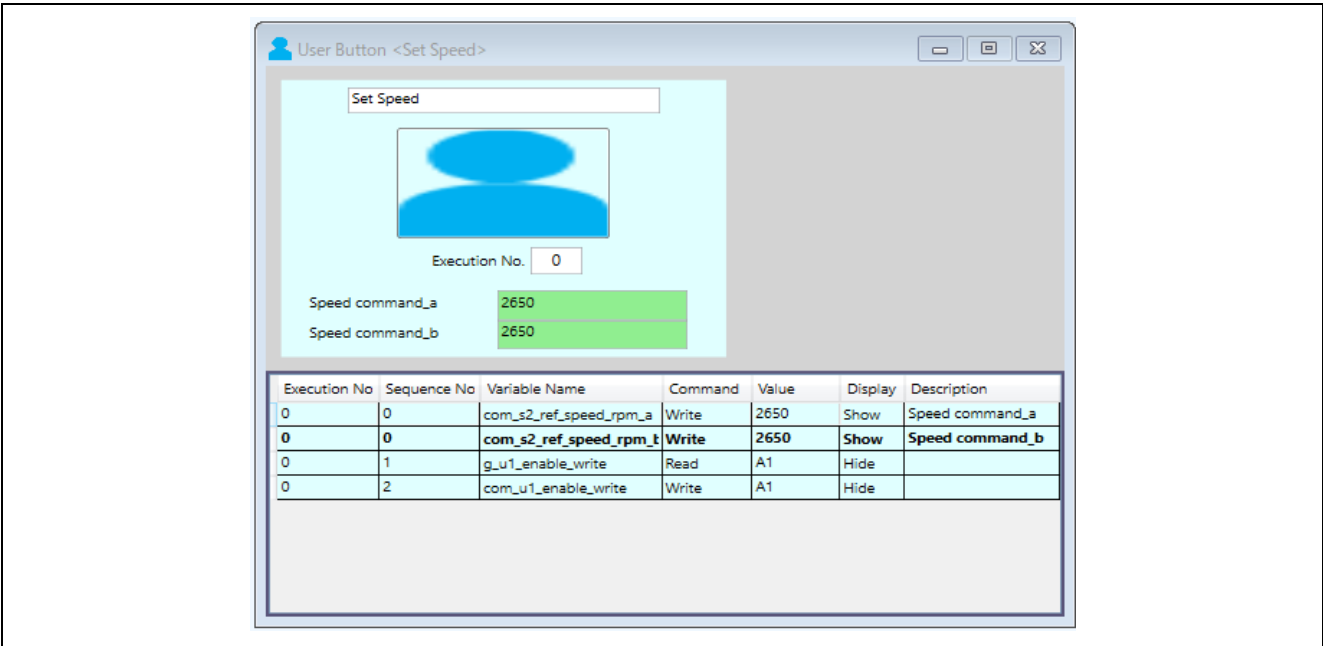
Figure 4-9   Change speed

5. Reference Documents

- RX66T Group User's Manual: Hardware (R01UH0749)
- Application note: 'Sensorless vector control for permanent magnet synchronous motor (Algorithm)' (R01AN3786)
- Renesas Motor Workbench User's Manual (R21UZ0004)
- Evaluation System for BLDC Motor User's Manual (R12UZ0062)
- RX66T CPU Card User's Manual (R12UZ0028)
- Smart Configurator User's Manual: RX API Reference (R20UT4360)
- RX Smart Configurator User's Guide: CS+ (R20AN0470)
- RX Smart Configurator User's Guide: e^2 studio (R20AN0451)

Revision History

Rev.	Date	Description	
		Page	**Summary**
1.00	Sep.17.21	—	First edition issued

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.

3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.

5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.

6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.

8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.

12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1   October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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