

RZ/T2N

Encoder Vector Control of Permanent Magnet Synchronous Motor (Serial Encoder) - 4 motor

Introduction

This document is an application note on a vector control sample program for a permanent magnet synchronous motor using a serial encoder with the RZ/T2N Evaluation Board equipped with the RZ/T2N MPU from Renesas Electronics. The supported serial encoder communication interface is FA-CODER®.

The target applications are the motor control units of servo amplifiers, robots, machine tools, and NC/CNC devices. The sample program supports 4-axes position control, velocity control, and torque control modes, which can be switched according to the application.

The supported algorithms are decoupling control, flux-weakening control, maximum torque per current control, voltage phase lead compensation, voltage error compensation, space vector modulation and sinusoidal modulation. The position control function supports position profile with triangular and trapezoidal speed control.

Figure 1-1 shows the waveforms for output U-phase current, d-axis and q-axis command current and actual current, speed command and actual speed, and position during position profile with trapezoidal speed control.

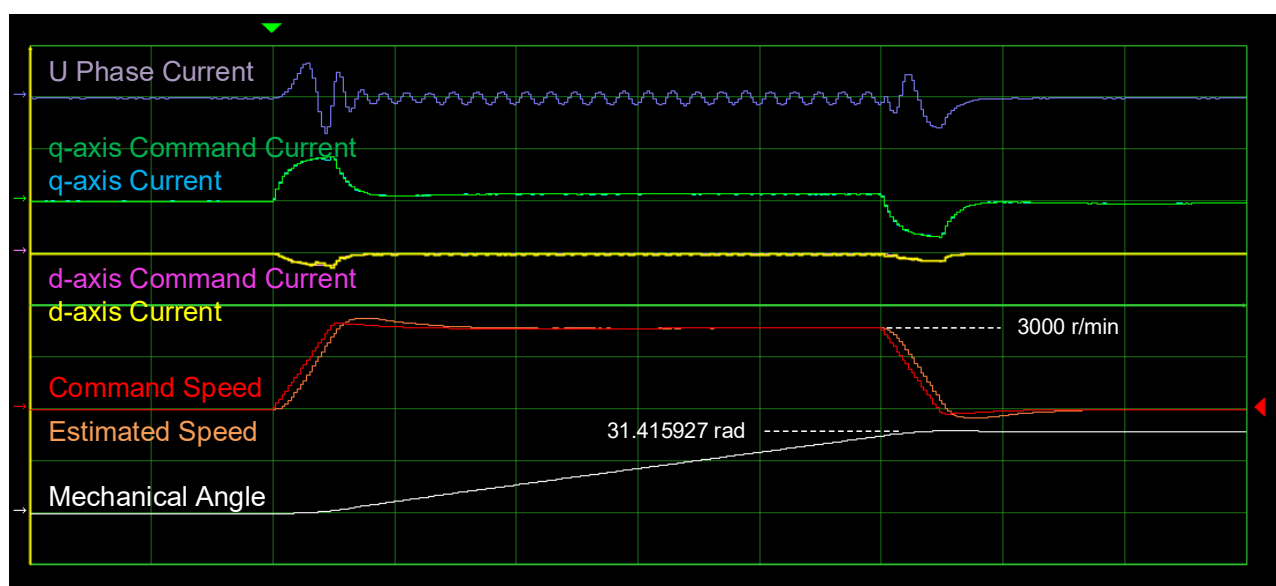


Figure 1-1 Trapezoidal speed control.

Target device

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

- RZ/T2N (R9A07G076M48GBG)

FA-CODER® is a trademark of TAMAGAWA SEIKI CO., LTD.

Table of Contents

1. Introduction	6
2. Glossary	8
3. Hardware and software.....	9
3.1 Hardware	9
3.1.1 Motor specification	9
3.1.2 Inverter board specification	10
3.1.3 Serial encoder (FA-CODER) specification	10
3.2 Software	10
4. Building a Hardware Environment.....	11
4.1 Hardware Environment Overview.....	11
4.2 Power Supply	11
4.3 Motor	11
4.4 Load.....	12
4.5 Serial encoder	12
4.6 Inverter board	13
4.6.1 Jumper pin setting	13
4.6.2 Volume setting.....	13
4.6.3 Motor and encoder connections	14
4.7 CPU board.....	15
4.7.1 Switch setting	15
4.7.2 Jumper pin setting	18
4.7.3 Power supply connection	20
4.7.4 Inverter board connection.....	20
4.7.5 Communication board connection.....	20
4.7.6 Connect to J-Link OB	20
5. Software building environment.....	21
5.1 Installing e ² Studio	21
5.2 Installing FSP	21
5.3 Introducing Renesas Motor Workbench.....	21
6. Driving the Motor	22
6.1 Precautions before Driving the Motor.....	22
6.2 Operation Preparation Procedure	23
6.3 Importing sample program	24
6.4 Building sample program.....	24
6.5 Writing sample program	24
6.6 Connecting with RMW	25
6.7 Registering and Updating the MAP File	26
6.8 Controlling motor using RMW	27
6.8.1 Variables Used for Working with RMW	27
6.8.2 Position control.....	29

6.8.3	Speed control	30
6.8.4	Torque control	31
6.8.5	Open loop control	32
6.8.6	Confirm error	33
6.8.7	Reset error.....	33
6.8.8	Current offset adjustment	34
6.8.9	Position offset adjustment	35
7.	Motor Control Algorithm	36
7.1	Overview.....	36
7.2	Control Block Diagram	36
7.3	Control timing	38
7.3.1	Overview.....	38
7.3.2	Starting transmission to serial encoder	38
7.3.3	Timer counter synchronized start between GPT units	39
7.4	Current sensing using delta-sigma modulator.....	40
7.4.1	Phase current sensing circuit	40
7.4.2	Current sensing timing and sinc filter settings.....	41
7.4.3	Conversion from DSMIF values to current values.....	42
7.5	Current Control Function	43
7.5.1	Decoupling Control.....	43
7.5.2	Voltage Phase Lead Compensation.....	44
7.5.3	Voltage Error Compensation	45
7.5.4	Pulse Width Modulation (PWM) Mode	46
7.6	Speed Control Function.....	47
7.6.1	Flux-weakening Control.....	47
7.6.2	Maximum Torque per Current Control (MTPA)	48
7.7	Position control	49
7.7.1	Position profile	49
7.8	Encoder vector control	50
7.8.1	Encoder communication time compensation	50
7.8.2	Speed detection.....	51
7.8.3	Multi-turn position detection	52
7.8.4	Position offset.....	53
8.	Software Specifications & Configurations	54
8.1	Software Specifications	54
8.2	Overall Configuration of the Software	55
8.3	Interrupts	56
8.4	Files and Folders	57
8.5	Application Layer	59
8.5.1	Functions	59
8.5.2	Configuration	59
8.5.3	Structure and Variable Information.....	60
8.5.4	Macro Definitions.....	62
8.5.5	Adjustment and configuration of parameters	62
8.6	Manager Modules and Motor Control Modules	63

8.6.1	Functions	63
8.6.2	Module Configuration	64
8.6.3	Mode Management	65
8.6.4	Protection Function	66
8.6.5	Wait process after transitioning to RUN state	66
8.6.6	API	67
8.6.7	Structure and Variable Information.....	70
8.6.8	Macro Definitions.....	78
8.7	Driver Modules	80
8.7.1	Functions	80
8.7.2	API	80
8.7.3	Configurations	81
8.7.4	Structure and Variable Information.....	81
8.7.5	Adjustment and Configuration of Parameters	82
8.8	Serial encoder driver module	83
8.8.1	Functions	83
8.8.2	API	83
8.8.3	Macro definition	84
9.	Setting Parameters	85
9.1	Overview.....	85
9.2	MPU-related Parameters.....	85
9.3	List of Control Function Setting Parameters	86
9.4	Protection-related Parameters	89
9.5	Setting the Pulse Modulation Method.....	90
9.6	Inverter Parameters.....	91
9.6.1	Overview.....	91
9.6.2	Current Detection Gain.....	92
9.6.3	Voltage Detection Gain	93
9.6.4	Voltage Error Compensation Parameters	93
9.7	Motor Parameters.....	94
9.8	Current Control Parameters	97
9.9	Speed Control Parameters.....	98
9.10	Position control parameters.....	99
9.11	Voltage Phase Lead Compensation Parameters	100
10.	FSP Settings.....	101
10.1	Clock settings	101
10.2	Interrupt settings.....	101
10.3	ELC settings	102
10.4	DSMIF settings	102
10.5	POEG settings.....	103
10.6	GPT settings.....	104
10.7	ADC settings.....	105
10.8	WDT settings	106
10.9	Pin settings	106

11. Evaluation results	109
11.1 Acceleration and deceleration characteristics	109
11.2 Load characteristics.....	110
11.3 CPU Utilization	110
11.4 Program size and RAM usage	110
12. FAQ.....	111
Revision History	112

1. Introduction

This document is an application note on a vector control sample program for a permanent magnet synchronous motor using a serial encoder with the RZ/T2N Evaluation Board equipped with the RZ/T2N MPU from Renesas Electronics. The communication interface of the serial encoder supported by the sample program in this application note is FA-CODER®. In addition, this sample program supports Renesas Motor Workbench (RMW), a motor control development support tool, which can be used as a user interface for checking the MPU internal data and controlling motors. You can use the sample program for reference purposes when selecting the MPU to be used or developing software by checking how MPU functions are allocated, how control is loaded by interruptions, and other information in the sample program.

The sample program described in this application note was developed and evaluated in the motor and inverter environment as described in this document and is not guaranteed to work in your motor or inverter environment. Control performance may be limited by factors such as mounting errors of the current sensor or encoder; PCB design pattern of the signal path; specifications of sampling/resolution/filter; magnetic saturation characteristics of the motor and variations between individual motors. Under the responsibility of the user, algorithm refinement and parameter tuning are required.

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

Main hardware and devices used for evaluation.

- CPU Board: RZ/T2N Evaluation Board (RTK9RZT2N1C00010BJ)
- Inverter Board: RZ/T Series Inverter Board (RTK0EM0000B15010BJ)
- Motor: TSM3101N2001E020 (TAMAGAWA SEIKI)

Target software

The following shows the target software for this application:

- RZT2N_INVBLV_4PM_ENCD_FOC_E2S_V100

Reference materials

- RZ/T2N Group User's Manual: Hardware (R01UH1084)
- RZ/T2N Group RZ/T2N Evaluation Board User's Manual (R20UT5574)
- RZ/T2N Group FA-CODER Sample Program (R11AN1031)
- RZ/T, RZ/N Series Getting Started with RZ Flexible Software Package for RZ/T FSP (R01AN6434)
- RZ/T Series Inverter Board/Kit User's Manual (R12UZ0155)
- Motor Control Development Support Tool Renesas Motor Workbench User's Manual (R21UZ0004)
- MC-COM User's Manual (R12UZ0093)

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 contents to be checked and corresponding chapters.

Items to be checked	Corresponding chapter
Check the equipment	3.1
Prepare the power supply	4.2
Prepare the motor	4.3, 4.5
Prepare the inverter board	4.6
Check the wiring	4
Install development environment	5.1, 5.2
Write a sample program to MPU	6.5
Install RMW into PC	5.3
Update the Map file after changing the sample program	6.7
Check the internal information of the inverter on a PC	6.8.1
Driving the motor	6.8
Examining Motor Control Algorithms	7
Examine the structure of the sample program	8
Check the inverter parameters	9.6
Check the motor parameters	9.7
Check MPU settings	10
Check FAQ	12
Check what to do if a problem occurs	

2. Glossary

Table 2-1 shows the main terms used in this document and their explanations.

Table 2-1 Glossary

Term	Description
APP	Application layer
rms	root mean square
RMW	Renesas Motor Workbench
PMSM	Permanent Magnet Synchronous Motor
Serial encoder	Encoder that outputs encoder counts as serial data
ENCIF	Peripheral that communicates with serial encoders
Delta Sigma Modulator	AD converter that outputs sampled analog signals as pulse density modulated signals
MCLK	Clock signal output by a delta sigma modulator
MDAT	Data signals output by delta-sigma modulators.
DSMIF	Peripheral to process MCLK and MDAT
sinc filter	Digital filter used to obtain DSMIF conversion values from pulse density modulated signals
Decimation	Down sampling pulse density modulated signals and converting frequencies
GPTm_n_UDF	Events that occur when the timer counter (GTCNT) of the GPTm_n (m: unit number, n: channel number) underflows
GPTm_n_OVF	Events that occur when the timer counter (GTCNT) of the GPTm_n (m: unit number, n: channel number) overflows
Crest	The time when GPTm_n_OVF occurs
Trough	The time when GPTm_n_UDF occurs

3. Hardware and software

3.1 Hardware

Table 3-1 Hardware

Hardware	Name	Manufacturer
MPU	RZ/T2N (R9A07G076M48GBG)	Renesas Electronics
CPU board	RZ/T2N Evaluation Board (RTK9RZT2N1C00000BE)	Renesas Electronics
Motor (*1)	TSM3101N2001E020	TAMAGAWA SEIKI CO., LTD
Inverter board	RZ/T Series Inverter Board (RTK0EM0000B15010BJ)	Renesas Electronics
Communication board	MC-COM (RTK0EMXC90S00000BJ)	Renesas Electronics
Debugger	J-Link OB	SEGGER

*1 A serial encoder is installed as standard. The communication interface is FA-CODER.

3.1.1 Motor specification

Table 3-2 shows motor specification. The resistance values, d-axis inductance, and q-axis inductance values are measured independently by our company, and the parameters obtained may vary depending on the measurement conditions and individual differences.

Table 3-2 Motor specification

Name	TSM3101N2001E020
Rated speed	3000 rpm
Maximum speed	6000 rpm
Rated frequency	250 Hz
Rated voltage	DC24V
Rated output	30 W
Rated current	3.5 Arms
Instantaneous maximum current	10.6 Arms
Rated torque	0.095 Nm
Instantaneous maximum torque	0.34 Nm
Resistance	0.626 Ω
d-axis inductance	0.000574 H
q-axis inductance	0.000813 H
Magnetic flux	0.003684 Wb
Rotor inertia	0.0000023 kg · m ²
Number of pole pairs	5 pole pairs

3.1.2 Inverter board specification

Table 3-3 Inverter board specification

Name	RZ/T Series Inverter Board (RTK0EM0000B15010BJ)
Rated voltage	DC24V
Rated current	4 Arms
Rated maximum current	12 Arms (300% rated output)

3.1.3 Serial encoder (FA-CODER) specification

Table 3-4 shows the specification of FA-CODER serial encoder. The transmission time is the time it takes to acquire single-turn encoder count for one axis when using this sample program.

Table 3-4 Serial encoder(FA-CODER) specification

Communication interface	FA-CODER
Single rotate resolution	17 bits
Transmission rate	2.5 Mbps
Transmission time	Approximately 34μs

3.2 Software

Table 3-5 Software

Software	Name	Version	Manufacturer
IDE	e ² studio	2026-07	Renesas Electronics
FSP	RZ Flexible Software Package	4.2.0	Renesas Electronics
Toolchain	GNU ARM Embedded	13.3.Rel1 (13.3.1.arm-13-24)	ARM
Development tool	Renesas Motor Workbench	3.3.1	Renesas Electronics

4. Building a Hardware Environment

4.1 Hardware Environment Overview

This section describes a hardware environment in which a motor is operated using this sample program. Since this sample program performs 4-axis control, prepare four sets of inverter boards and motors with encoders.

Figure 4-1 shows an example of hardware configuration. In the sections that follow, the power supply (4.2), the motor (4.3) and load system (4.4), the inverter board (RZ/T Series Inverter board) (4.6), CPU board (RZ/T2N Evaluation Board) (4.7), communication board (MC-COM) (4.7.5), and debugger (J-Link) (4.7.6) are described in detail.

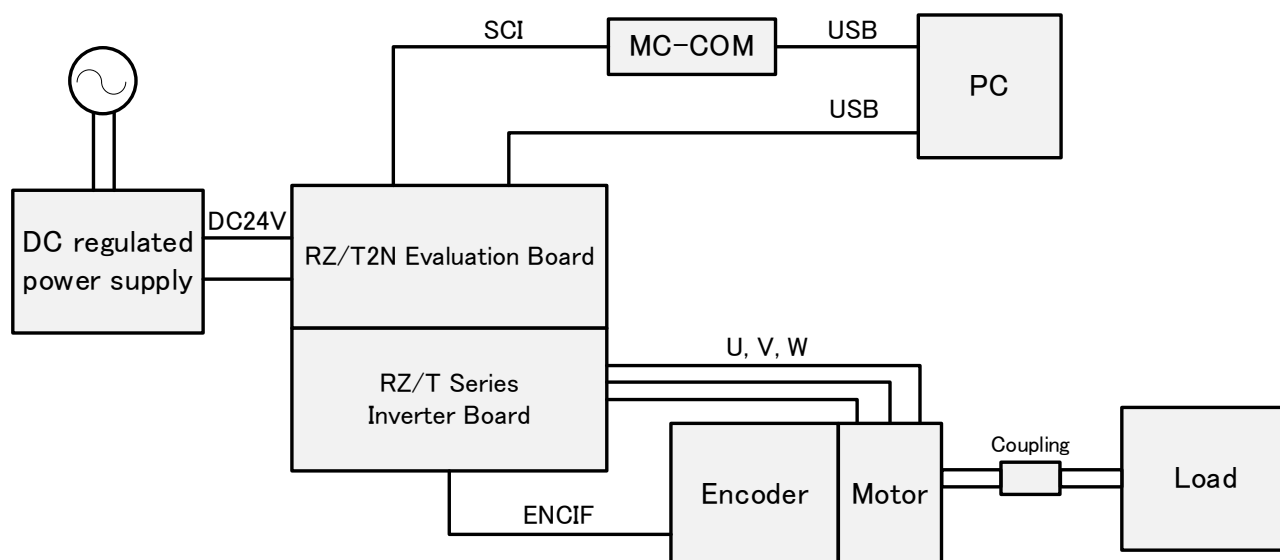


Figure 4-1 Hardware configuration example.

4.2 Power Supply

In this sample program, a voltage of 24 VDC is supplied to the 3-phase inverter RZ/T Series Inverter Board via the RZ/T2N Evaluation Board by using a DC regulated power supply, AC adapter. USB power supply is not recommended because in this sample program, a higher rated power supply capacity for low-speed high torque is required.

The voltage supplied to the inverter board varies depending on the inductive voltage, rating conditions, and maximum load conditions of the motor to be used. Select an appropriate type of power supply based on your experimental environment and restrictions and conditions of the power supply to be used. The inverter board introduced here has a rated output current of 4A, and if the motor is changed, the rated operation may not be possible depending on the rating conditions of the motor. Please note.

4.3 Motor

The motor is connected to the inverter board using a motor connector (TE Connectivity 4-pin connector).

Figure 4-2 shows connection between motor connector and motor.

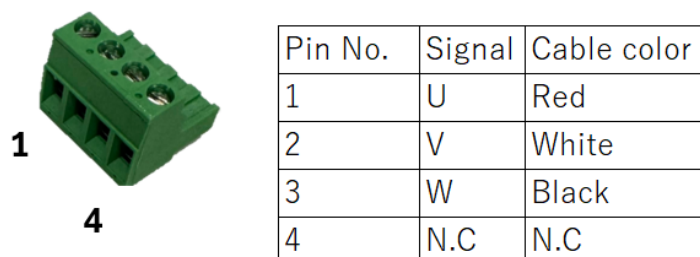


Figure 4-2 Connection between motor connector (TE Connectivity 4 pin connector) and motor

4.4 Load

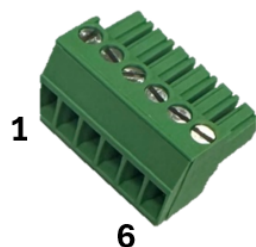
Select the appropriate load according to the power of the motor. Table 4-1 shows loading equipment used to evaluate the sample program.

Table 4-1 Load system

Hardware	Name	Manufacturer
Torque meter	TM303	MAGTROL
Hysteresis brake	HB-50M-2	MAGTROL
Dynamometer Controller	MODEL DSP6001	MAGTROL

4.5 Serial encoder

The serial encoder is connected to the inverter board using an encoder connector (TE Connectivity 6-pin connector). Figure 4-3 shows connection between encoder connector and serial encoder.



Pin No.	Signal	Cable color
1	VDD1	Red
2	GND	Black
3	D0+	Blue
4	D0-	Blue/Black
5	N.C	N.C
6	N.C	N.C

Figure 4-3 connection between encoder connector (TE Connectivity 6 pin connector) and serial encoder

4.6 Inverter board

4.6.1 Jumper pin setting

Make sure that the jumper pins on the inverter board are set to default positions as shown in Table 4-2. The power supply voltage of the serial encoder used in this sample program is 5V. To supply 5V from the inverter board to the serial encoder, make JP10 2-3 pin short.

Table 4-2 Jumper pin settings of inverter board

Jumper pin	Default	Setting
JP7, JP8, JP9	1-2pin Short	Incremental encoder input selection 1-2pin short: RX485 RX Receiver (differential input)
JP10	2-3pin Short	Absolute encoder I/O power supply selection 2-3pin short: +5.0V
JP11	2-3pin Short	Incremental encoder I/O power supply selection 2-3pin short: +5.0V

For more information about the RZ/T Series Inverter Board, refer to the RZ/T Series Inverter Board/Kit User's Manual (R12UZ0155).

Figure 4-4 shows jumper pin layout of inverter board.

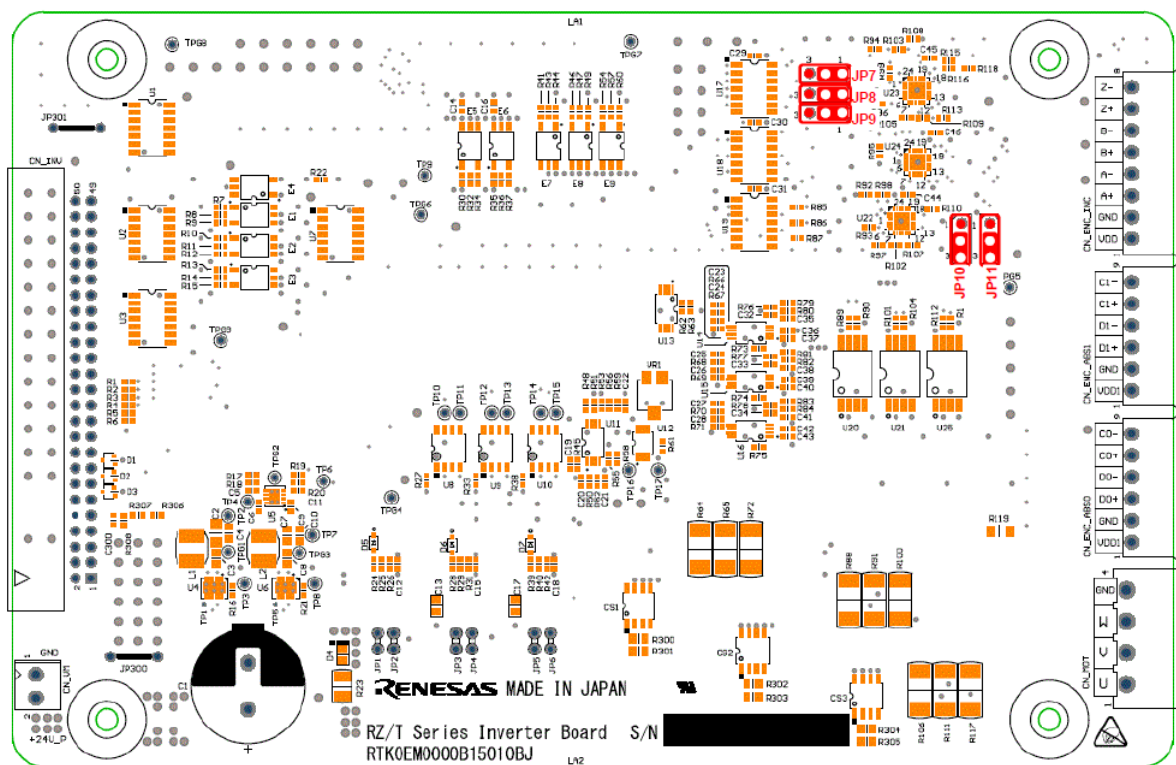


Figure 4-4 Jumper pin layout of inverter board.

4.6.2 Volume setting

Make sure the volume VR1 on the inverter board is fully rotated clockwise.

4.6.3 Motor and encoder connections

Connect the motor connector to the CN_MOT terminal on the inverter board. Connect the encoder connector to the CN_ENC_ABS0 terminal.

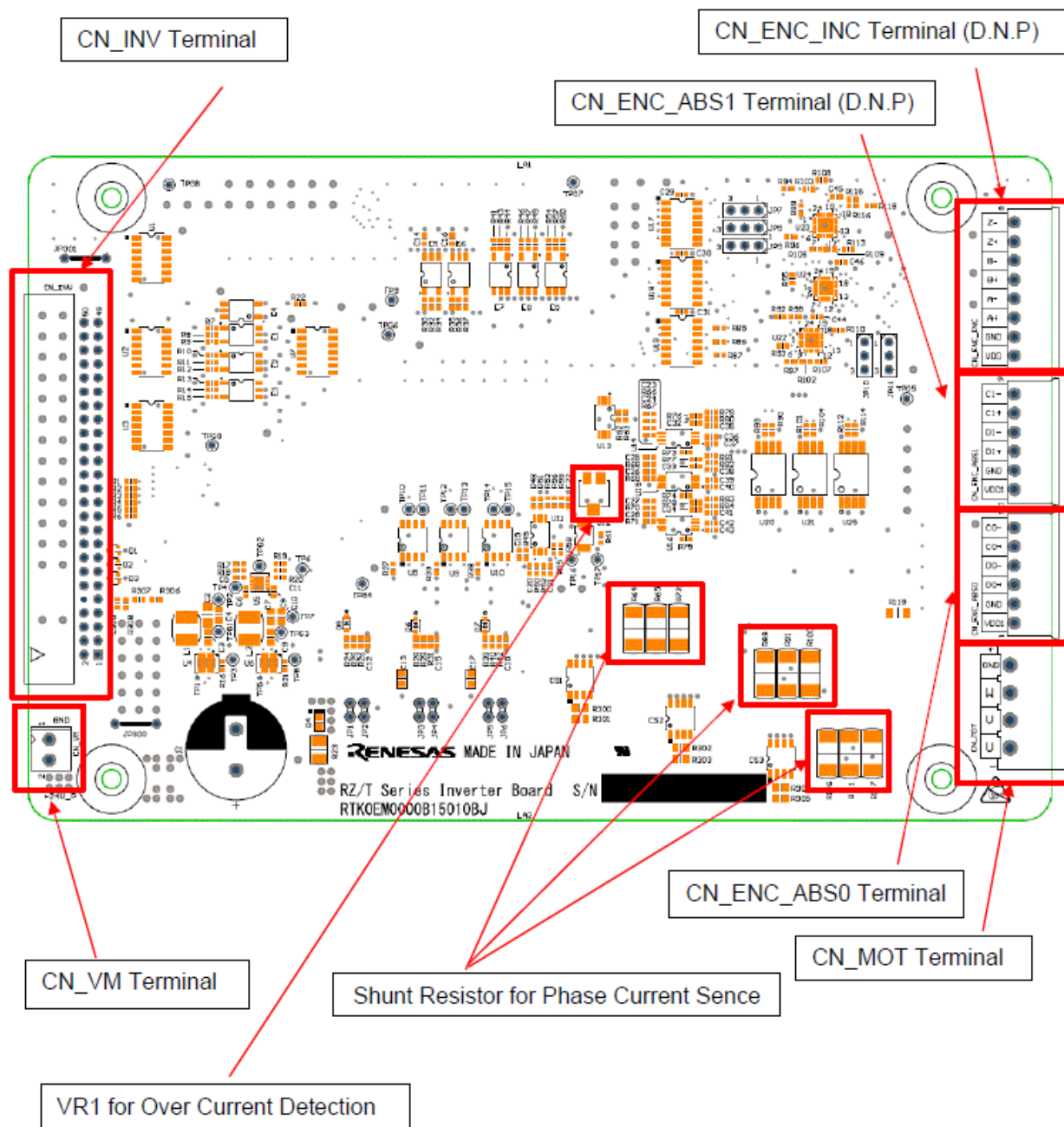


Figure 4-5 Connector layout of inverter board.

4.7 CPU board

4.7.1 Switch setting

Table 4-3 to Table 4-10 show switch setting of CPU board. *1 indicates the parts that need to be changed from the default.

Table 4-3 Settings of mode setting switch SW8

Switch	ON/OFF	Setting
SW8_1	ON	xSPI0 boot mode(x1 boot serial flash)
SW8_2	ON*1	
SW8_3	ON	
SW8_4	OFF	Unused

Table 4-4 Settings of mode setting switch SW10

Switch	ON/OFF	Setting
SW10_1	OFF	Unused
SW10_2	OFF*1	VCC1833_1 = 3.3V (J15 : 2-3 short)
SW10_3	OFF*1	VCC1833_2 = 3.3V (J18 : 2-3 short)
SW10_4	OFF	Unused
SW10_5	OFF	Unused
SW10_6	ON	VCC1833_5 = 1.8V (J39 : 1-2 short)
SW10_7	OFF	Cortex-R52 CPU0 ATCM wait cycle = 1 wait
SW10_8	OFF	Cortex-R52 CPU1 ATCM wait cycle = 1 wait
SW10_9	ON	JTAG mode = Normal mode
SW10_10	ON*1	2nd Boot CPU = Cortex-R52 CPU0

Table 4-5 Settings of signal function selection switch SW7

Switch	ON/OFF	Setting
SW7_1	OFF*1	P03_6, P03_7, P04_0 to P04_3, and P15_0 are used as control signals for inverter board INV3. (P15_0 is also used as a control signal for BSC2.)
SW7_2	OFF*1	P23_1, P23_2, P23_4 to P23_7, P24_0 to P24_4, and P25_0 are used as control signals for inverter board INV2, INV3, INV4, BSC1, and GPIO connector.
SW7_3	OFF*1	P00_0 to P00_7, P01_0 to P01_5, P16_1, and P16_7 are used as control signals for inverter board INV1, INV4, BSC1, SPI expansion connector, and PMOD1.
SW7_4	OFF*1	P01_6, P01_7, P02_0 to P02_7, P03_0, P03_1, P03_4, P03_5, and P15_1 are used as control signals for inverter board INV1, INV2, INV3, INV4, BSC1, BSC2, PMOD1, and PMOD2.
SW7_5	OFF*1	P15_7 and P16_2 to P16_5 are used as control signals for inverter board INV2, INV4, and PMOD1.
SW7_6	OFF*1	P17_5 and P18_3 to P18_6 are used as control signals for inverter board INV1, INV3, and INV4.
SW7_7	ON	P04_4 to P04_7 and P05_0 to P05_6 are used as control signals for HSPI, inverter board INV3, and PMOD2.
SW7_8	ON	

Table 4-6 Settings of signal function selection switch SW9

Switch	ON/OFF	Setting
SW9_1	OFF	Unused
SW9_2	OFF	Unused
SW9_3	OFF	P08_5 to P08_7 and P09_3 are used as control signals for inverter board INV4, BSC1, and SPI expansion connector.
SW9_4	OFF	Unused
SW9_5	OFF	P00_6, P00_7, and P01_0 are used as ENCIF signals for inverter board INV1 when SW7-3 = OFF.
SW9_6	OFF	P02_4 to P02_6 are used as ENCIF signals for inverter board INV2 when SW7-4 = OFF.
SW9_7	OFF	P15_2, and P15_0 and P15_1 when SW7-1, -4 = OFF, OFF are used as ENCIF signals for inverter board INV3.
SW9_8	OFF	P16_6, P16_7 when SW7-3 = OFF, and P17_0 when SW13-1 = ON are used as ENCIF signals for inverter board INV4 when SW12-5 = ON.

Table 4-7 Settings of signal function selection switch SW11

Switch	ON/OFF	Setting
SW11_1	ON	P11_0 is used as VBUSEN for USB host.
SW11_2	OFF	
SW11_3	OFF	P11_7 is used as control signal for inverter board INV2 or SHOST.
SW11_4	ON	
SW11_5	OFF	P12_4 is used as user push switch SW4 input.
SW11_6	OFF	
SW11_7	ON	
SW11_8	ON	P14_3 is used as EXICEN for USB OTG.
SW11_9	OFF	
SW11_10	OFF	Unused

Table 4-8 Settings of signal function selection switch SW12

Switch	ON/OFF	Setting
SW12_1	OFF	Unused
SW12_2	OFF	Unused
SW12_3	ON	P20_4, P20_5, and P07_0 are used as MDC, MDIO, and NRESET for Ethernet Port 0.
SW12_4	ON	P20_4, P20_5, and P07_0 are used as MDC, MDIO, and NRESET for Ethernet Port 2.
SW12_5	ON*1	P16_6, P16_7 when SW7-3 = OFF, and P17_0 when SW13-1 = ON are used as control signals for inverter board INV4.
SW12_6	OFF	PCIe is used as an endpoint.
SW12_7	OFF	The 12V power supply of the PCIe connector J6 is OFF.
SW12_8	OFF	The 3.3V power supply of the PCIe connector J6 is OFF.

Table 4-9 Settings of signal function selection switch SW13

Switch	ON/OFF	Setting
SW13_1	ON	P17_0 is used as a control signal for SPI expansion connector and inverter board INV4.
SW13_2	OFF	
SW13_3	ON	P15_4 is used as SCL for EEPROM.
SW13_4	OFF	
SW13_5	OFF	
SW13_6	ON	P17_1 is used as a control signal for SPI expansion connector and inverter board INV4.
SW13_7	OFF	
SW13_8	ON	P15_5 is used as SDA for EEPROM
SW13_9	OFF	
SW13_10	OFF	

Table 4-10 Settings of signal function selection switch SW14

Switch	ON/OFF	Setting
SW14_1	ON	AN000 is connected to the potentiometer.
SW14_2	OFF	

4.7.2 Jumper pin setting

Table 4-11 to Table 4-15 show jumper pin setting of CPU board. *1 indicates the parts that need to be changed from the default.

Table 4-11 Settings of I/O power selection jumper

Jumper pin	Open / Short	Setting
J15	2-3 Short*1	3.3V power is supplied to VCC1833_1. The setting for: - using domain signals of VCC1833_1 for other than Ethernet Port 2 when SW7-2 = OFF.
J18	2-3 Short*1	3.3V power is supplied to VCC1833_2. The setting for: - using domain signals of VCC1833_2 for Ethernet Port 3 and 4, when SW7-3, -4 = ON, ON and RMII is used at Ethernet Port 3 and 4. or - using domain signals of VCC1833_2 for other than Ethernet Port 3 and 4 when SW7-3, -4 = OFF, OFF.
J25	1-2 Short*1	3.3V power is supplied to VCC1833_6. The setting for: - using domain signals of VCC1833_6 for other than microSD when SW7-1 = OFF.
J39	1-2 Short	1.8V power is supplied to VCC1833_5. The setting for: - using domain signals of VCC1833_5 for eMMC when SW7-7, -8 = ON, ON.
J41	2-3 Short	1.8V power is supplied to the VDDMDIO of PHY (U54) for Ethernet Port 0. The setting for: - P20_4, P20_5, and P07_0 are used as MDC, MDIO, and NRESET for Ethernet Port 0 when SW12-3 = ON.

Table 4-12 Setting of debug function selection jumper

Jumper pin	Open / Short	Setting
J29	Open	The on-board debugging function J-Link™ OB is enabled.

Table 4-13 Setting of PCIe reset signal control jumper

Jumper pin	Open / Short	Setting
J1	Open	A PCIe reset is not included in the system reset factors.

Table 4-14 Settings of current measurement jumper

Jumper pin	Open / Short	Setting
J19	Open	The jumper blocks are open-circuit at the time of shipment. To measure the current, remove R75 and insert an ammeter between pins 1,3 and 2,4.
J13	Short	The jumper blocks are short-circuit at the time of shipment. To measure the current, make the relevant jumper block open-circuit and insert an ammeter between pins 1-2.
J12	Short	
J16	Short	
J11	Short	
J14	Short	
J17	Short	
J20	Short	
J22	Short	
J23	Short	
J24	Short	

Table 4-15 Setting of inverter board 24V power supply GND connection jumper

Jumper pin	Open / Short	Setting
J48	Short	Inverter board 24V power supply GND (GND_24V) is connected to GND. Connect pins 1-2, 3-4, and 5-6 respectively. The setting for supplying 24V to this board from J10 and supplying 24V to inverter boards from this board.

4.7.3 Power supply connection

Connect the output of the DC regulated power supply to J10(Terminal Blocks 24V IN) on the CPU board.

4.7.4 Inverter board connection

Connect CN_INV of inverter board to J49(INV1), J50(INV2), J51(INV3) and J52(INV4) of the CPU board.

4.7.5 Communication board connection

Connect SCI connector of communication board to J40(RMW connector) of the CPU board.

4.7.6 Connect to J-Link OB

Connect the USB cable connected to the PC to J32 on the CPU board.

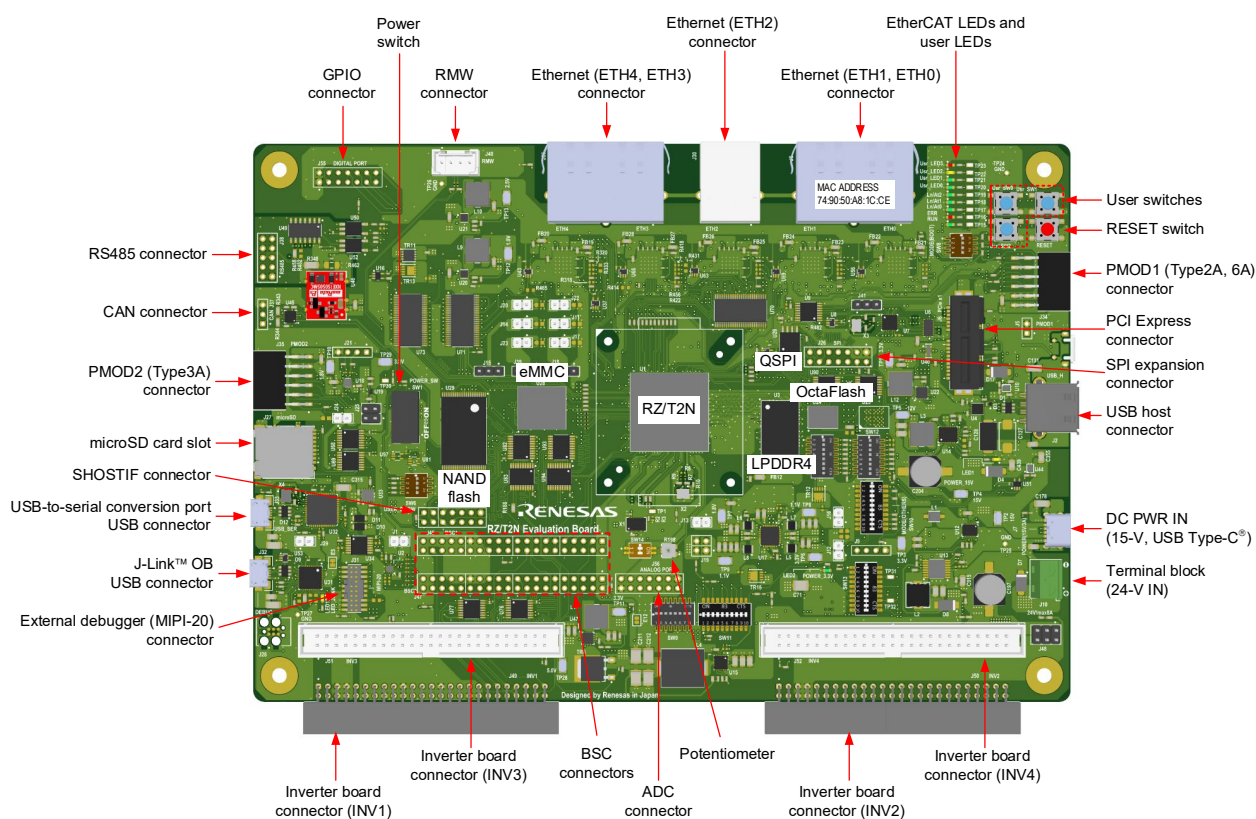


Figure 4-6 Connector layout of CPU Board

5. Software building environment

5.1 Installing e² Studio

Download e² studio from the following URL:

<https://www.renesas.com/en/software-tool/e2-studio>

When installing e² studio, make sure GNU ARM Embedded 13.3-Rel1 (13.3.1.arm-13-24) is selected on the additional software selection. Figure 5-1 shows the additional software selection.

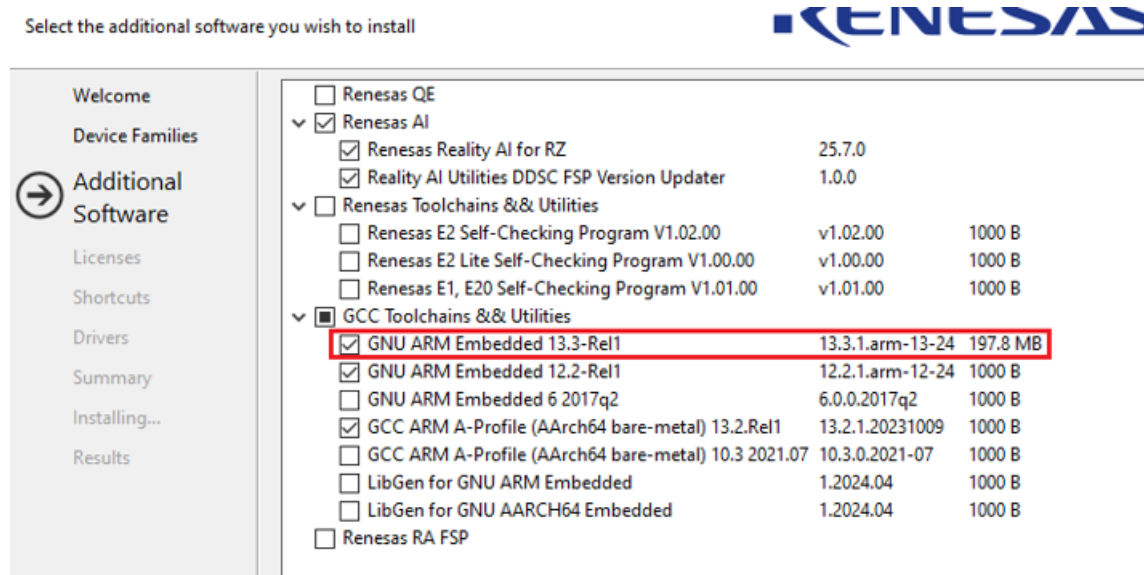


Figure 5-1 Additional software selection.

5.2 Installing FSP

Download the installer (RZ_FSP_Packs_v4.2.0.exe) from the following URL and install FSP Version 4.2.0. Select the path where e² studio is installed.

<https://github.com/renesas/rz-fsp/releases>

5.3 Introducing Renesas Motor Workbench

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

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

6. Driving the Motor

6.1 Precautions before Driving the Motor

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

- Do not control the motor under conditions where the tracing and breakpoints are set while using debugger. 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 evaluation and experiment environment with the consideration that the motor can be stopped in an emergency.
- Beware of disconnection of the encoder wire. If the encoder wire is broken or has poor connection, the motor cannot be controlled normally.
- Because the motor shaft rotates, be sure to install the cover as a guard. Especially, parts such as couplings may scatter outside if damaged during rotation.
- When the inverter is stopped but the motor is still rotating, the motor generates an inductive voltage, thus applying voltage to the U/V/W three-phase wiring. Touching exposed conductive parts may cause electric shock.

6.2 Operation Preparation Procedure

Preparation procedures for operation are as follows:

Table 6-1 Operation preparation procedure

Step	Description	Related chapters
1	Confirm switch and jumper pin setting of CPU board	4.7.1 4.7.2
2	Confirm jumper pin setting and volume setting of inverter board	4.6.1 4.6.2
3	Import sample program	6.3
4	Build sample program	6.4
5	Connect power supply to CPU board	4.7.3
6	Connect to J-Link OB	4.7.6
7	Power on	-
8	Writing sample program	6.5
9	Power off	-
10	Remove USB cable from CPU board	4.7.6
11	Connect inverter board to CPU board	4.7.4
12	Connect motor and encoder to inverter board	4.6.3
13	Connect communication board to CPU board	4.7.5
14	Power on	-
15	Connect with RMW	6.6
16	Control motor using RMW	Position control(6.8.2) Speed control(6.8.3) Torque control(6.8.4) Open loop control(6.8.5)

6.3 Importing sample program

- (1) Launch e² studio and select File - Import from the menu bar to open the Import dialog box.
- (2) Select General—Existing Projects into Workspace from the menu bar.
- (3) Select “Browse” button and select the directory where the sample project is stored.
- (4) Select “Finish” button.

6.4 Building sample program

- (1) Select Project - Build All from the menu bar in e²studio.
- (2) If the building is successful, the console will display "Build Finished".

6.5 Writing sample program

- (1) Select Run - Debug from the menu bar in e²studio and write the program.
- (2) When the program writing is complete, the console will display “Finished download.”
- (3) Select Run - Terminate from the menu bar in e²studio and finish debugging.

6.6 Connecting with RMW

- (1) Select RZT2N_INVBLV_4PM_ENCD_FOC_E2S_V100\src\application\user_interface\ics\RZT2N_INVBLV_4PM_ENCD_FOC_E2S_V100.rmt to start RMW.

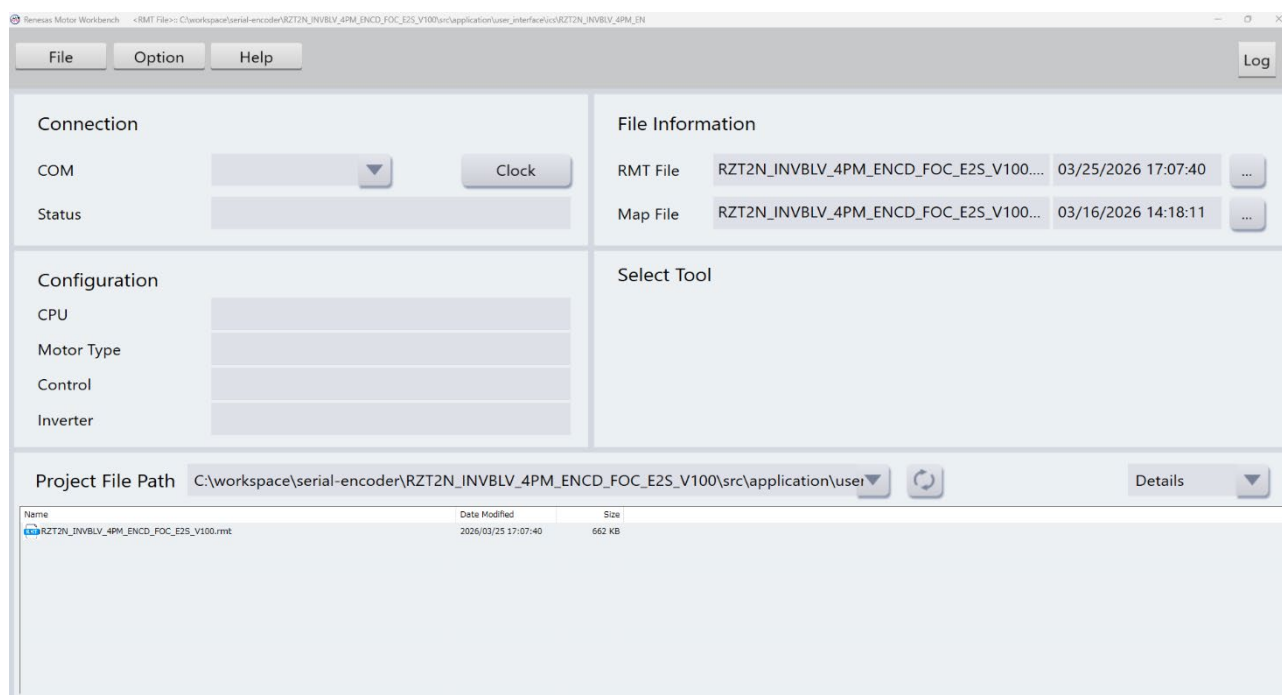


Figure 6-1 RMW after startup.

- (2) Select COM port to use connect with RMW.

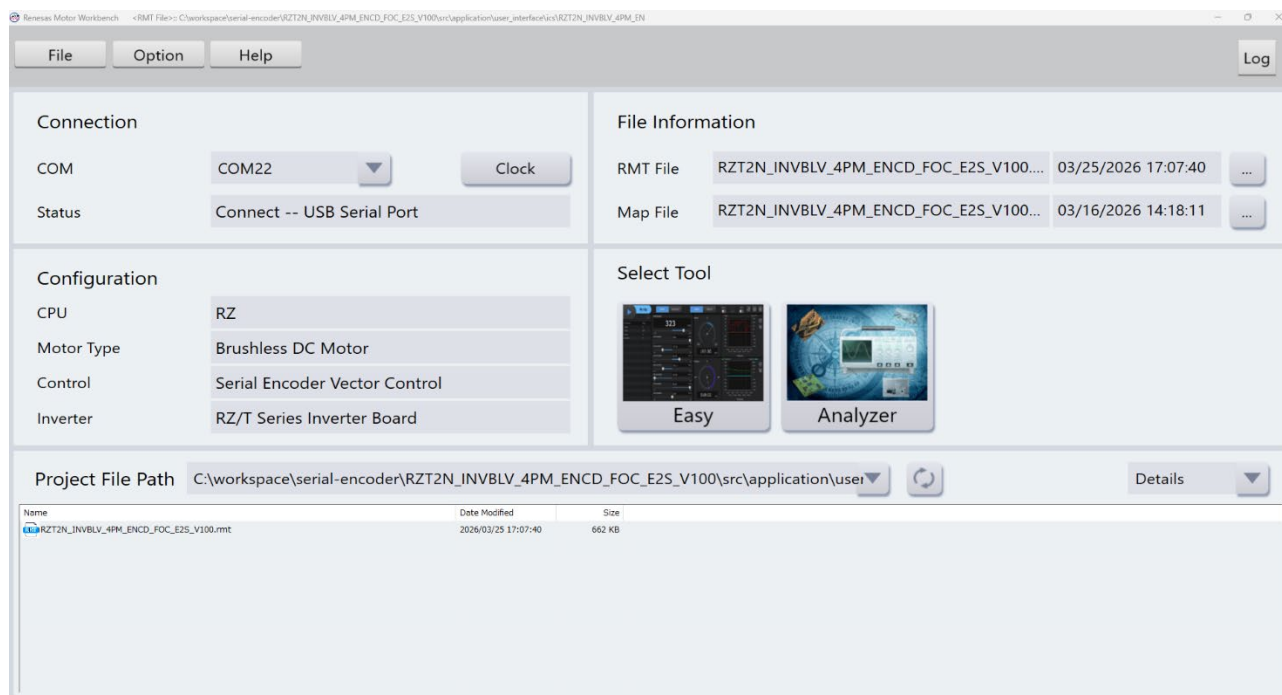


Figure 6-2 RMW after connection.

6.7 Registering and Updating the MAP File

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

When updating the map file, perform the following steps.

- (1) Click the “...” button in File Information – Map File.
- (2) Select
RZT2N_INVBLV_4PM_ENCD_FOC_E2S_V100\src\application\user_interface\ics\RZT2N_INVBLV_4PM_ENCD_FOC_E2S_V100_conv.map.
- (3) Click “Set” in “User Setting Form”.

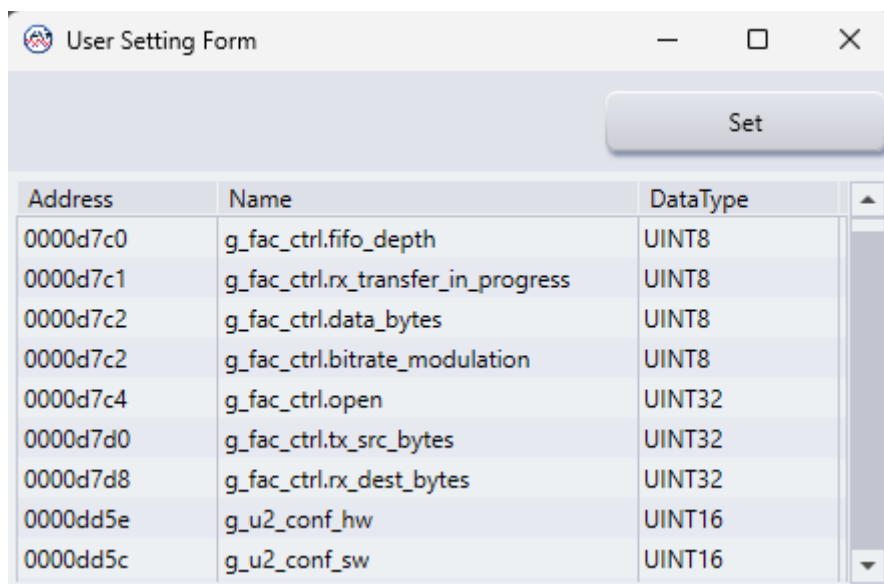
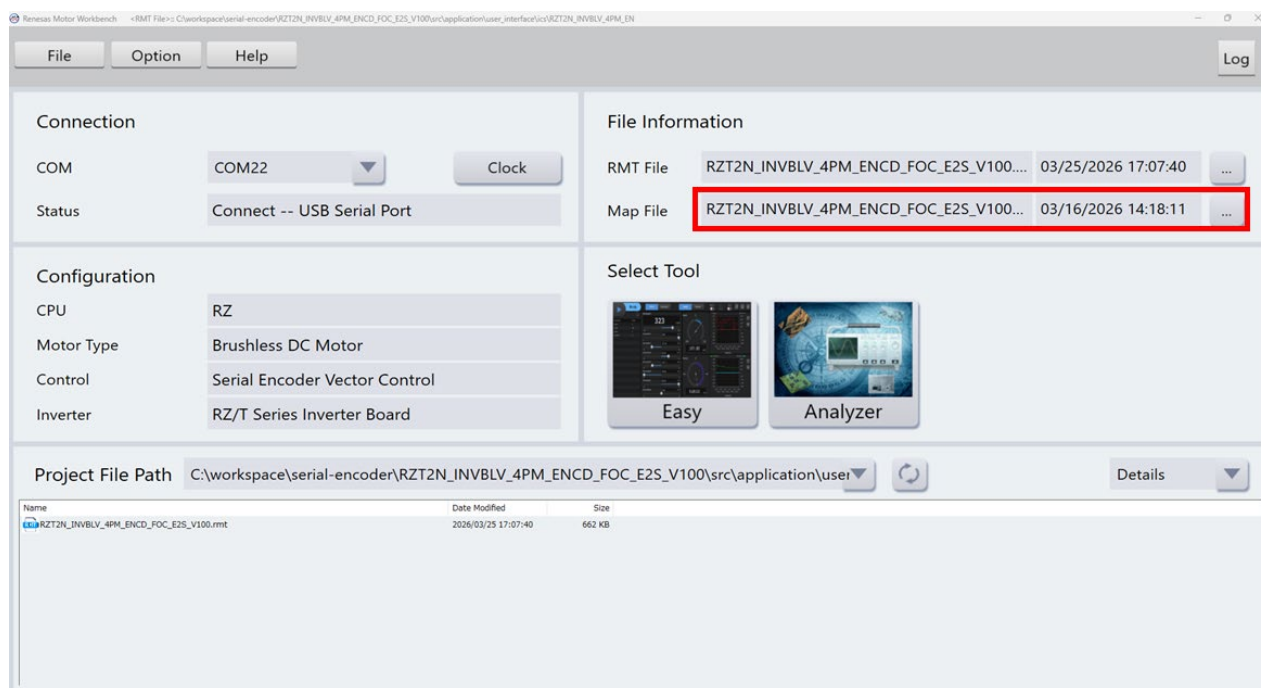


Figure 6-3 Map file registration setup window (upper) and User Setting Form (lower) in RMW.

6.8 Controlling motor using RMW

6.8.1 Variables Used for Working with RMW

In this sample program, the motor is controlled using RMW. Table 6-2 lists the data input variables that are used when 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 `comX.u1_enable_write` variable is the same as the value of the `g_u1_enable_write` variable. However, note that the variables indicated by an asterisk (*) do not depend on the value of the `comX.u1_enable_write` variable.

Some parameters used for motor control can be changed while the motor is stopped. For details, see Table 8-7.

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

Table 6-2 List of main input variables for Analyzer function (X = 0, 1, 2, 3)

Variable name	Type	Description
<code>comX.u1_system_mode (*)</code>	<code>uint8_t</code>	System mode 0 : Stop 1 : Position control 2 : Speed control 3 : Torque control 4 : Open loop control 5 : Reset error 8 : Current offset adjustment 9 : Position offset adjustment
<code>comX.u1_enable_write</code>	<code>uint8_t</code>	Variable to enable rewrite variable. Input data reflected by matching <code>comX.g_u1_enable_write</code>
<code>comX.f4_torque_ref_nm (*)</code>	<code>float</code>	Torque command value [Nm]
<code>comX.f4_ref_speed_rpm (*)</code>	<code>float</code>	Speed command value (mechanical angle) [rpm]
<code>comX.f4_ref_position_deg (*)</code>	<code>float</code>	Position command value (mechanical angle) [deg]
<code>comX.f4_ol_crnt_ref</code>	<code>float</code>	d-axis command current (Open loop control) [A]

Table 6-3 lists main structure variables that are often monitored when evaluating motor control. Use this table for reference when the waveform is to be displayed, or the values of variables are to be loaded with an Analyzer function.

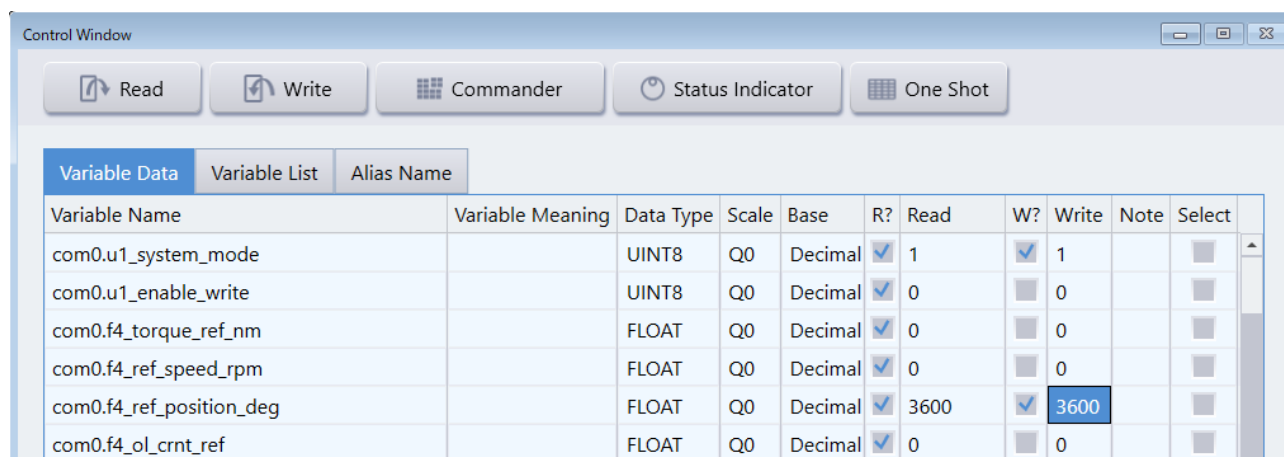
Table 6-3 List of main variables for motor control (X = 0, 1, 2, 3)

Variable	Type	Description
g_st_serial_encoder_vectorX.u2_error_status	uint16_t	An error status For details, see Table 6-4.
g_st_ccX.f4_vdc_ad	float	Inverter bus voltage [V]
g_st_ccX.f4_id_ref	float	d-axis current command value [A]
g_st_ccX.f4_id_ad	float	d-axis current detection value [A]
g_st_ccX.f4_iq_ref	float	q-axis current command value [A]
g_st_ccX.f4_iq_ad	float	q-axis current detection value [A]
g_st_ccX.f4_iu_ad	float	U-phase current detection value [A]
g_st_ccX.f4_iv_ad	float	V-phase current detection value [A]
g_st_ccX.f4_iw_ad	float	W-phase current detection value [A]
g_st_ccX.f4_vd_ref	float	d-axis voltage command value [V]
g_st_ccX.f4_vq_ref	float	q-axis voltage command value [V]
g_st_ccX.f4_refu	float	U-phase voltage command value [V]
g_st_ccX.f4_refv	float	V-phase voltage command value [V]
g_st_ccX.f4_refw	float	W-phase voltage command value [V]
g_st_ccX.st_rotor_angle.f4_rotor_angle_rad	float	Magnetic pole position (electrical angle) [rad]
g_st_serial_encoder_vectorX.f4_rotor_angle_offset_rad	float	Position offset (electrical angle) [rad]
g_st_scX.f4_ref_speed_rad_ctrl	float	Speed command value (mechanical angle) [rad/s]
g_st_scX.f4_speed_rad	float	Speed detection value (mechanical angle) [rad/s]
g_st_pcX.f4_ref_pos_rad_ctrl	float	Position command value (mechanical angle) [rad]
g_st_pcX.f4_pos_rad	float	Position detection value (mechanical angle) [rad]

6.8.2 Position control

The following operation with RMW rotates the motor 3600 deg in the CW direction.

- (1) Select “Select Tool” – Analyzer mode after RMW connection.
- (2) Check the box in “W?” column of comX.u1_system_mode and comX.f4_ref_position_deg.
- (3) Enter 1 in “Write” column of comX.u1_system_mode.
- (4) Click the Write button.
- (5) Enter 3600 in “Write” column of comX.f4_ref_position_deg.
- (6) Click the Write button.
- (7) The motor rotates 3600 deg in the CW direction.
- (8) Enter 0 in “Write” column of comX.u1_system_mode.
- (9) Click the Write button.



The screenshot shows the 'Control Window' application. At the top, there are buttons for 'Read', 'Write', 'Commander', 'Status Indicator', and 'One Shot'. Below these is a table titled 'Variable Data' with columns: Variable Name, Variable Meaning, Data Type, Scale, Base, R?, Read, W?, Write, Note, and Select. The table lists six variables. The 'W?' column has checkboxes, and the 'Write' column has input fields. The variable 'com0.f4_ref_position_deg' has '3600' entered in its 'Write' field.

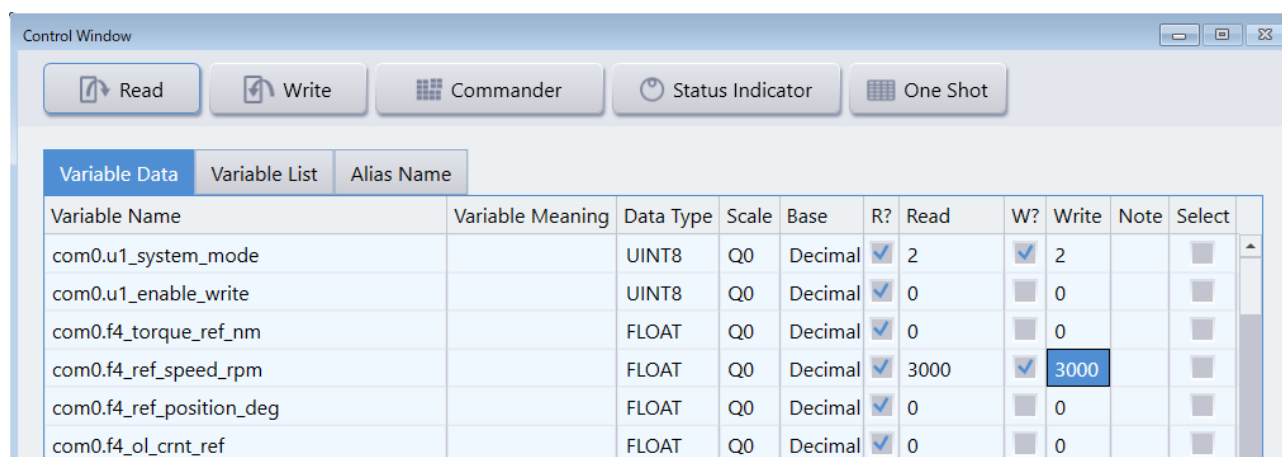
Variable Name	Variable Meaning	Data Type	Scale	Base	R?	Read	W?	Write	Note	Select
com0.u1_system_mode		UINT8	Q0	Decimal	☒	1	☒	1		☐
com0.u1_enable_write		UINT8	Q0	Decimal	☒	0	☐	0		☐
com0.f4_torque_ref_nm		FLOAT	Q0	Decimal	☒	0	☐	0		☐
com0.f4_ref_speed_rpm		FLOAT	Q0	Decimal	☒	0	☐	0		☐
com0.f4_ref_position_deg		FLOAT	Q0	Decimal	☒	3600	☒	3600		☐
com0.f4_ol_crnt_ref		FLOAT	Q0	Decimal	☒	0	☐	0		☐

Figure 6-4 Position control.

6.8.3 Speed control

The following operation with RMW rotates the motor at 3000 rpm in the CW direction.

- (1) Select "Select Tool" – Analyzer mode after RMW connection.
- (2) Check the box in "W?" column of comX.u1_system_mode and comX.f4_ref_speed_rpm.
- (3) Enter 2 in "Write" column of comX.u1_system_mode.
- (4) Click the Write button.
- (5) Enter 3000 in "Write" column of comX.f4_ref_speed_rpm.
- (6) Click the Write button.
- (7) The motor rotates at 3000rpm in the CW direction.
- (8) Enter 0 in "Write" column of comX.f4_ref_speed_rpm.
- (9) Click the Write button.
- (10) Enter 0 in "Write" column of comX.u1_system_mode.
- (11) Click the Write button.



The screenshot shows the 'Control Window' application. At the top, there are buttons for 'Read', 'Write', 'Commander', 'Status Indicator', and 'One Shot'. Below these is a table titled 'Variable Data' with columns: Variable Name, Variable Meaning, Data Type, Scale, Base, R?, Read, W?, Write, Note, and Select. The table lists several variables, with 'com0.f4_ref_speed_rpm' highlighted, showing a value of 3000 in the 'Write' column.

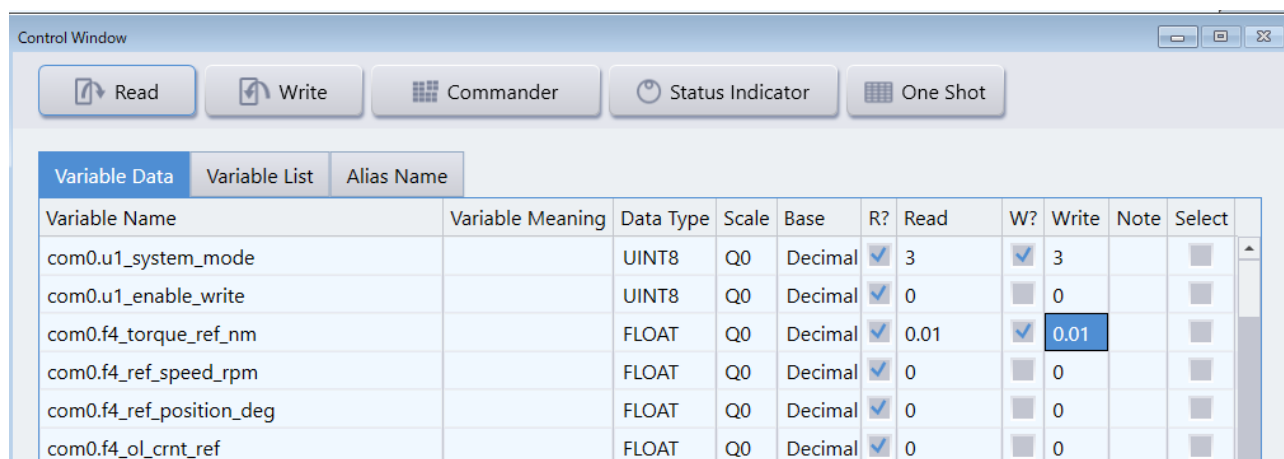
Variable Name	Variable Meaning	Data Type	Scale	Base	R?	Read	W?	Write	Note	Select
com0.u1_system_mode		UINT8	Q0	Decimal	☒	2	☒	2		☐
com0.u1_enable_write		UINT8	Q0	Decimal	☒	0	☐	0		☐
com0.f4_torque_ref_nm		FLOAT	Q0	Decimal	☒	0	☐	0		☐
com0.f4_ref_speed_rpm		FLOAT	Q0	Decimal	☒	3000	☒	3000		☐
com0.f4_ref_position_deg		FLOAT	Q0	Decimal	☒	0	☐	0		☐
com0.f4_ol_crnt_ref		FLOAT	Q0	Decimal	☒	0	☐	0		☐

Figure 6-5 Speed control.

6.8.4 Torque control

The following operation with RMW rotates the motor in the CW direction in torque control mode.

- (1) Select "Select Tool" – Analyzer mode after RMW connection.
- (2) Check the box in "W?" column of comX.u1_system_mode and comX.f4_torque_ref_nm.
- (3) Enter 3 in "Write" column of comX.u1_system_mode.
- (4) Click the Write button.
- (5) Enter 0.01 in "Write" column of comX.f4_torque_ref_nm.
- (6) Click the Write button.
- (7) The motor rotates in the CW direction.
- (8) Enter 0 in "Write" column of comX.f4_torque_ref_nm.
- (9) Click the Write button.
- (10) Enter 0 in "Write" column of comX.u1_system_mode.
- (11) Click the Write button.



The screenshot shows the 'Control Window' application. At the top, there are buttons for 'Read', 'Write', 'Commander', 'Status Indicator', and 'One Shot'. Below these is a table with columns: Variable Data, Variable List, and Alias Name. The main table has columns: Variable Name, Variable Meaning, Data Type, Scale, Base, R?, Read, W?, Write, Note, and Select. The table lists several variables, with 'com0.f4_torque_ref_nm' highlighted, showing a value of 0.01 in the 'Write' column.

Variable Name	Variable Meaning	Data Type	Scale	Base	R?	Read	W?	Write	Note	Select
com0.u1_system_mode		UINT8	Q0	Decimal	☒	3	☒	3		☐
com0.u1_enable_write		UINT8	Q0	Decimal	☒	0	☐	0		☐
com0.f4_torque_ref_nm		FLOAT	Q0	Decimal	☒	0.01	☒	0.01		☐
com0.f4_ref_speed_rpm		FLOAT	Q0	Decimal	☒	0	☐	0		☐
com0.f4_ref_position_deg		FLOAT	Q0	Decimal	☒	0	☐	0		☐
com0.f4_ol_crnt_ref		FLOAT	Q0	Decimal	☒	0	☐	0		☐

Figure 6-6 Torque control.

6.8.5 Open loop control

The following operation with RMW rotates the motor at about 500 rpm in the CW direction in open loop control mode.

- (1) Select "Select Tool" – Analyzer mode after RMW connection.
- (2) Check the box in "W?" column of comX.u1_system_mode, comX.u1_enable_write, comX.f4_ref_speed_rpm and comX.f4_ol_crnt_ref.
- (3) Enter 4 in "Write" column of comX.u1_system_mode.
- (4) Click the Write button.
- (5) Enter 500 in "Write" column of comX.f4_ref_speed_rpm and 1 in "Write" column of comX.f4_ol_crnt_ref.
- (6) Click the Write button.
- (7) Enter 1 in "Write" column of comX.u1_enable_write.
- (8) Click the Write button.
- (9) The motor rotates in CW direction.
- (10) Enter 0 in "Write" column of comX.f4_ref_speed_rpm and comX.f4_ol_crnt_ref.
- (11) Click the Write button.
- (12) Enter 0 in "Write" column of comX.u1_enable_write.
- (13) Click the Write button.

Variable Name	Variable Meaning	Data Type	Scale	Base	R?	Read	W?	Write	Note	Select
com0.u1_system_mode		UINT8	Q0	Decimal	☒	4	☒	4		☐
com0.u1_enable_write		UINT8	Q0	Decimal	☒	1	☒	1		☐
com0.f4_torque_ref_nm		FLOAT	Q0	Decimal	☒	0	☐	0		☐
com0.f4_ref_speed_rpm		FLOAT	Q0	Decimal	☒	500	☒	500		☐
com0.f4_ref_position_deg		FLOAT	Q0	Decimal	☒	0	☐	0		☐
com0.f4_ol_crnt_ref		FLOAT	Q0	Decimal	☒	1	☒	1		☐

Figure 6-7 Open loop control.

6.8.6 Confirm error

When an error occurs, LED11 (red) on the CPU board will turn on.

- (1) Check the box in "R?" column of g_st_serial_encoder_vectorX.u2_error_status.
- (2) Click the Read button.
- (3) Confirm the value displayed in "Read" column of g_st_serial_encoder_vectorX.u2_error_status.

Table 6-4 Error status description

Value	Error Description	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
0x0080	Low-voltage error	MOTOR_ERROR_LOW_VOLTAGE
0x0100	Software overcurrent error	MOTOR_ERROR_OVER_CURRENT_SW
0xffff	Undefined error	MOTOR_ERROR_UNKNOWN

6.8.7 Reset error

- (1) Enter 5 in "Write" column of comX.u1_system_mode.
- (2) Click the Write button.

When the error is cleared, LED11 (red) on the CPU board will turn off.

6.8.8 Current offset adjustment

The current offset adjustment starts automatically 0.5 seconds after power-on. LED 8 will light up during the operation. You can also manually recalibrate it by doing the following: Normally, there is no need to manually recalibrate.

- (1) Select "Select Tool" – Analyzer mode after RMW connection.
- (2) Check the box in "W?" column of comX.u1_system_mode.
- (3) Enter 8 in "Write" column of comX.u1_system_mode.
- (4) Click the Write button.
- (5) Click the Read button and make sure that "Read" column of comX.u1_system_mode is 0.

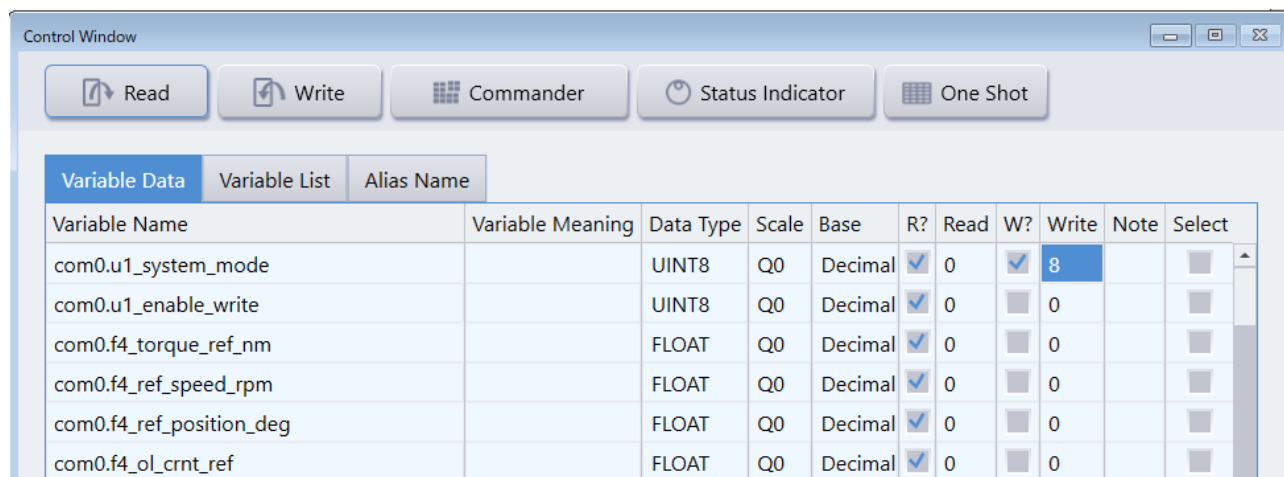


Figure 6-8 Current offset adjustment.

6.8.9 Position offset adjustment

In this sample program, the position offset when using the motor (TSM3101N2001E020) is set as the initial value for the member variable `f4_rotor_angle_offset_rad` within `g_st_serial_encoder_vectorX` ($X = 0, 1, 2, 3$). Therefore, manual position offset adjustment is typically unnecessary. If manual position offset adjustment is required, follow the steps below.

- (1) Select “Select Tool” – Analyzer mode after RMW connection.
- (2) Check the box in “W?” column of `comX.u1_system_mode`.
- (3) Enter 9 in “Write” column of `comX.u1_system_mode`.
- (4) Click the Write button.
- (5) Click the Read button and make sure that “Read” column of `comX.u1_system_mode` is 0.

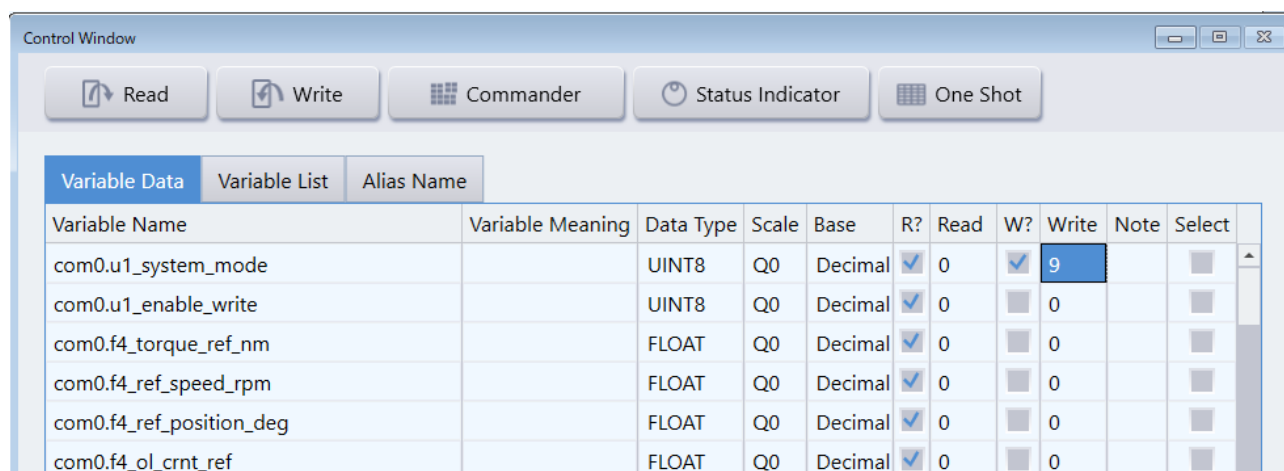


Figure 6-9 Position offset adjustment.

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

Control method	Encoder vector control
Pulse Width Modulation method	Space vector modulation or Sinusoidal modulation
Position and speed detection methods	Serial encoder
Control Mode	Position control, speed control, torque control and open loop control
Control and compensation functions	- Decoupling control - Voltage phase lead compensation - Voltage error compensation - PWM modulation - Flux-weakening control - Maximum Torque per Current control (MTPA) - Position profile - Encoder communication time compensation

7.2 Control Block Diagram

Figure 7-1 shows current control block diagram. Figure 7-2 shows position and speed control block diagram.

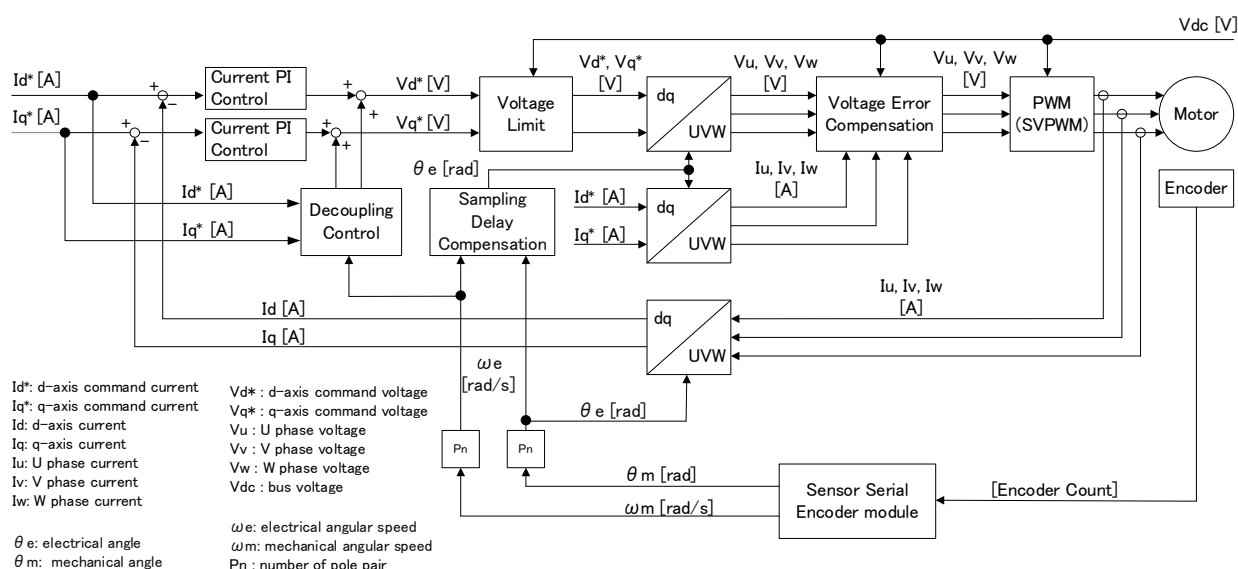


Figure 7-1 Current control block diagram.

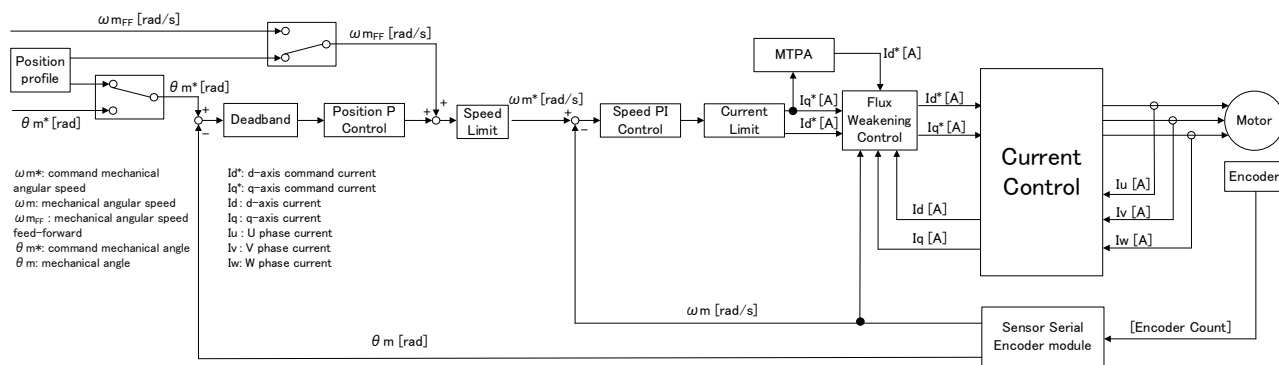


Figure 7-2 Position and speed control block diagram.

7.3 Control timing

7.3.1 Overview

Figure 7-3 shows control timing. For details of interruption process in Figure 7-3, see 8.3.

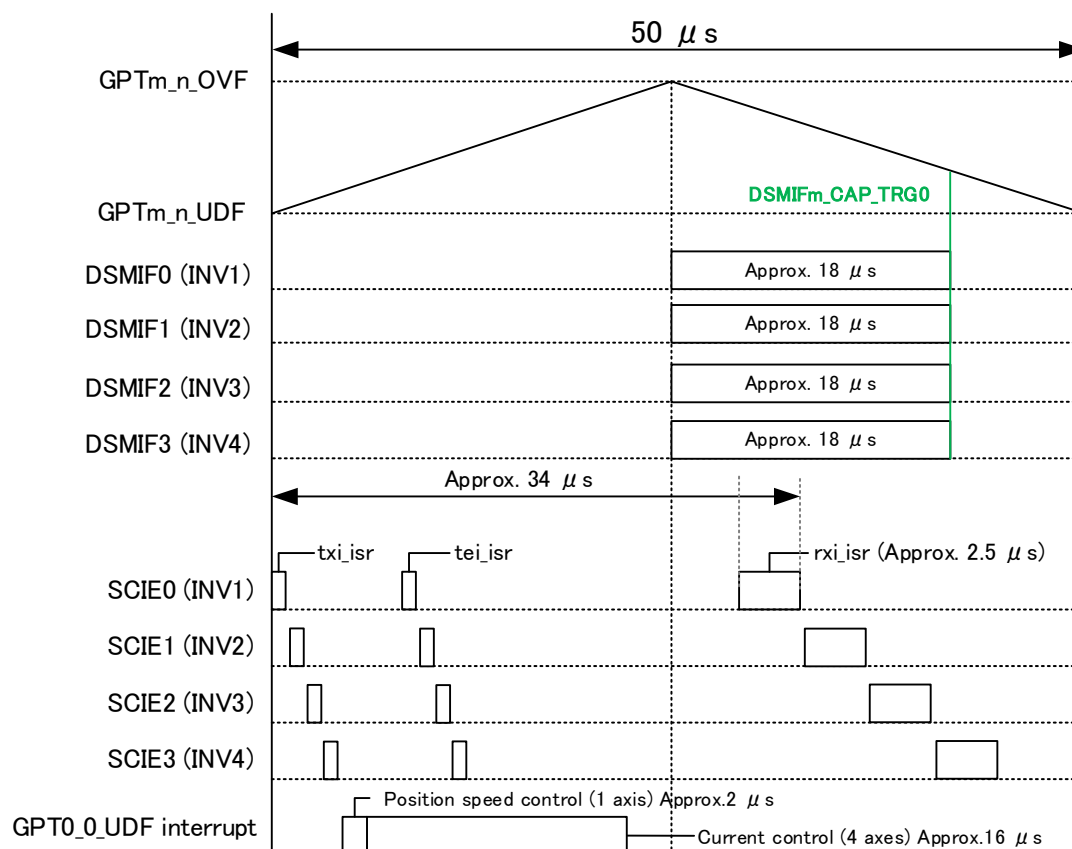


Figure 7-3 Control timing

7.3.2 Starting transmission to serial encoder

This sample program uses DMAC to synchronize the start of communication between GPTm_n_UDF and the encoder. When GPTm_n_UDF occurs, DMAC is triggered to transfer data to SCIE's Common Control Register 0 (CCR0), setting the TIE bit and TE bit to 1, thereby generating the FA-CODER transmission interrupt. Figure 7-4 shows the signal connections between peripherals, using GPT Unit 0, Channel 0 as an example.

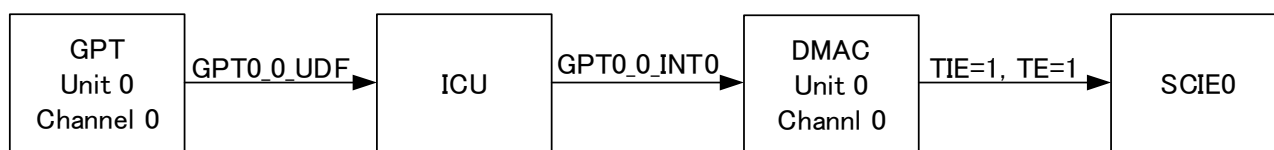


Figure 7-4 Signal connection between GPT, ICU, DMAC, and SCIE

7.3.3 Timer counter synchronized start between GPT units

This sample program first starts with the timer counters GPT0_0, GPT0_1, and GPT0_2 used for controlling the first axis motor. Using ELC, when GPT0_0_UDF occurs in the next cycle, it starts the timer counters for the second, third, and fourth axis GPTs, synchronizing the start of the timer counters for all four axes. Figure 7-5 shows the control timing when the GPTs for each unit are synchronized.

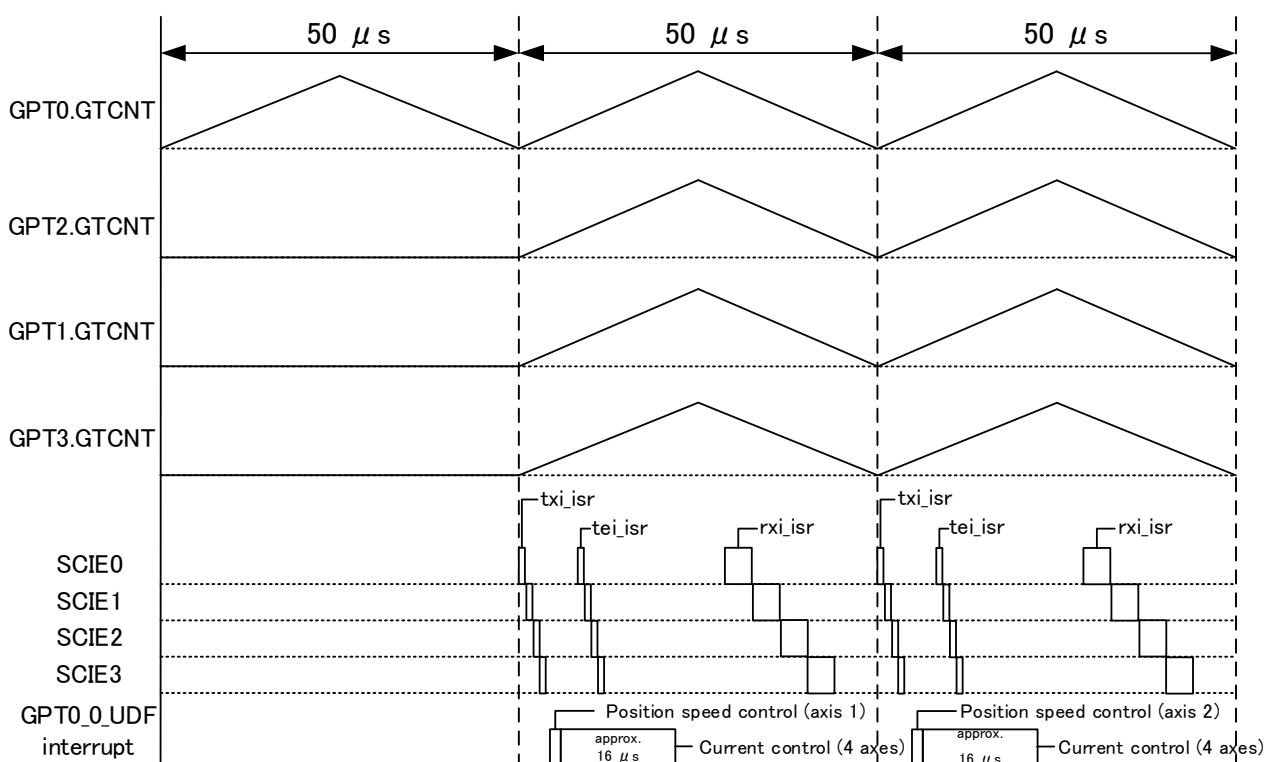


Figure 7-5 Control timing when each GPT unit is synchronized

7.4 Current sensing using delta-sigma modulator

7.4.1 Phase current sensing circuit

RZ/T Series Inverter Board has the delta-sigma modulator RV1S9355A for sensing phase current. Figure 7-6 shows the block diagram of the phase current sensing circuit on RZ/T Series Inverter Board.

The delta-sigma modulator outputs a pulse density modulated signal (MDAT) and a clock signal (MCLK, 20MHz) to the MPU (RZ/T2N) based on the voltage between the VIN+ and VIN- terminals. The MPU processes the signal output by the delta-sigma modulator using the sinc filter of the DSMIF (Delta-Sigma Modulator Interface) to obtain the DSMIF conversion value.

RZ/T Series Inverter Board defines the current flowing from the inverter board to the motor as positive, and the current flowing from the motor to the inverter board as negative.

The GND terminal of the regulator supplying +5V to the delta-sigma modulator and the GND terminal on the +5V side of the delta-sigma modulator are connected to the motor terminals for each phase. Therefore, when no PWM signal is input to the inverter board, the delta-sigma modulator does not output the MDAT signal and the MCLK signal. This sample program outputs a PWM signal to the inverter board, then waits until the signal from the delta-sigma modulator is output and current detection becomes possible. For details on the wait process, see 8.6.5.

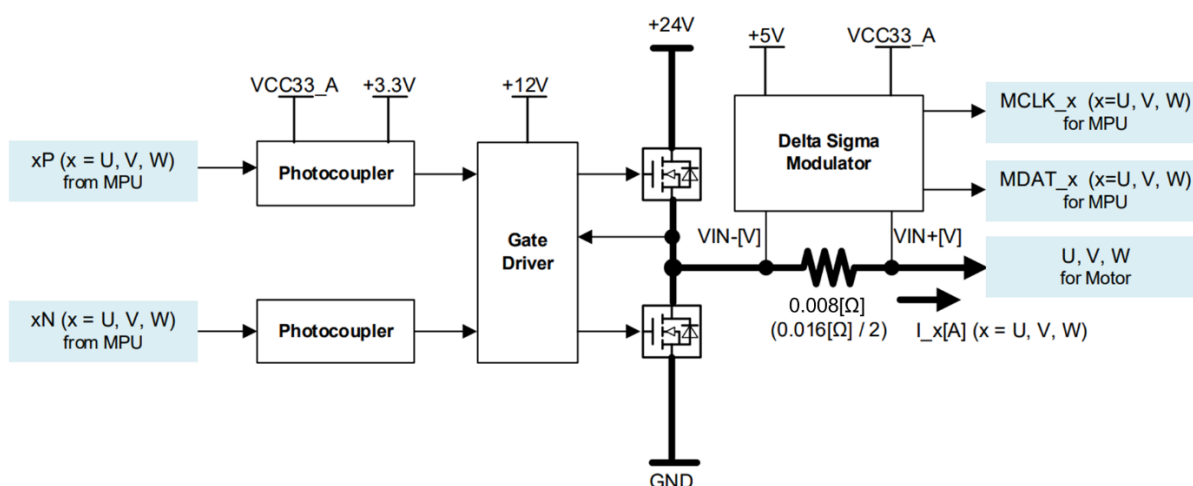


Figure 7-6 Block diagram of phase current sensing circuit

7.4.2 Current sensing timing and sinc filter settings

To sense current that does not include ripple, detect the current at the carrier crest and valley timing (GPT1_OVF and GPT1_UDF). When sensing current using a delta-sigma modulator, the delay time of the sinc filter must be considered. As shown in Figure 7-7, acquiring the DSMIF conversion value after the sinc filter's delay time has elapsed from the carrier crest allows sensing of current without ripple.

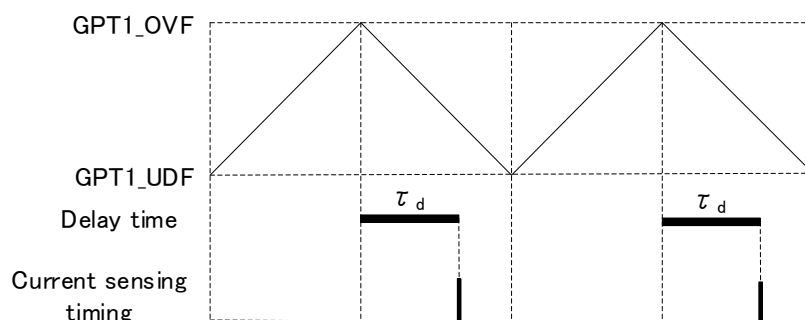


Figure 7-7 Delay time and current sensing timing

The delay time τ_d of the sinc filter can be expressed by the following equation, where N is the order of the sinc filter, DR is the decimation ratio, and τ_M is the output clock period of the delta-sigma modulator.

$$\tau_d = \frac{\tau_M(N \times DR - 2)}{2}$$

The output clock of the RV1S9355A is 20 MHz, so the output clock period τ_M is 0.05 μ s.

$$\tau_d(\mu s) = \frac{0.05 \times (N \times DR - 2)}{2}$$

Position and speed control, as well as current control, start after the FA-CODER transmission interrupt triggered at the carrier valley (GPTm_n_UDF) has completed (see Figure 7-3). When detecting the current at the carrier crest of the previous cycle and performing current control during the GPTm_n_UDF interrupt of the next cycle, the following inequality for the delay time τ_d is obtained.

$$\tau_d(\mu s) < 25.0$$

Determine the filter order N and decimation ratio DR to satisfy this inequality.

The DSMIF of RZ/T2N includes first-, second-, and third-order sinc filters. Since higher filter order N outputs higher resolution, a third-order sinc filter is used.

The DSMIF of the RZ/T2N can be selected from 4 to 256 as the decimation ratio. Table 7-2 shows the decimation ratio DR and delay time when the filter order N is 3. Since the output frequency of the delta-sigma modulator RV1S9355 is 20 MHz, the post-decimation frequency is 20 MHz / DR . Since the post-decimation frequency should ideally be an integer multiple of the carrier frequency, Table 7-2 only shows decimation ratios where the post-decimation output frequency is an integer multiple of the carrier frequency. The maximum decimation ratio DR satisfying the delay time τ_d inequality is 250, and the delay time τ_d is approximately 18.7 μ s.

Table 7-2 Decimation Ratio (DR) and delay time τ_d (the filter order N is 3)

Decimation Ratio (DR)	Delay time τ_d (μ s)
25	1.83
50	3.70
100	7.45
125	9.33
250	18.70

7.4.3 Conversion from DSMIF values to current values

(1) Voltage between VIN+ and VIN- of the delta-sigma modulator and the DSMIF conversion value

The voltage between VIN+ and VIN- of the delta-sigma modulator RV1S9355 on the RZ/T Series Inverter Board is a maximum of 0.32V and a minimum of -0.32V. Since the DSMIF conversion value is 12 bits, the voltage per count is $(0.64 / 4095)$ [V/count].

(2) Shunt register value

Since two-0.016 [ohm] resistors are connected in parallel for each phase, the shunt resistor value is 0.008 [ohm].

(3) Polarity of the current value

RZ/T Series Inverter Board defines the current flowing from the inverter board to the motor as positive, and the current flowing from the motor to the inverter as negative. When current flows from the inverter board to the motor, i.e., when positive current flows, the voltage drops across the shunt resistor causes the VIN- voltage to be higher than the VIN+ voltage. At this time, the voltage value between VIN+ and VIN- (VIN+ - VIN-) is negative. Since the DSMIF conversion value varies around the DSMIF conversion value (2047) when the voltage between VIN+ and VIN- is 0V, when calculating the current value from the value after subtracting the offset, it is necessary to multiply by (-1) to reverse the polarity.

The formula for calculating the current value from the DSMIF conversion value of the delta-sigma modulator RV1S9355 is shown below.

$$\text{Current value [A]} = (-1) \times (\text{DSMIF conversion value} - 2047) [\text{count}] \times 0.640[\text{V}] / 4095[\text{count}] \times (1 / 0.008) [1 / \text{ohm}]$$

In this sample program, since the center value of the DSMIF conversion output from the delta-sigma modulator on the RZ/T Series Inverter Board does not correspond to 2047, we use 1915 as the value to subtract from the DSMIF conversion result.

$$\text{Current value [A]} = (-1) \times (\text{DSMIF conversion value} - 1915) [\text{count}] \times 0.640[\text{V}] / 4095[\text{count}] \times (1 / 0.008) [1 / \text{ohm}]$$

Rewriting this expression using the macros defined in r_motor_inverter_cfg.h and r_motor_module_cfg.h yields the following expression:

$$\text{Current value [A]} = (-1) \times (\text{DSMIF conversion value} - \text{MOTOR_MCU_CFG_ADC_OFFSET}) [\text{count}] \times \text{INVERTER_CFG_CURRENT_ADC_REF_VOLTAGE [V]} / \text{MOTOR_MCU_CFG_AD12BIT_DATA} [\text{count}] \times (1 / \text{INVERTER_CFG_SHUNT_RESIST}) [1 / \text{ohm}]$$

For MOTOR_MCU_CFG_ADC_OFFSET and MOTOR_MCU_CFG_AD12BIT_DATA, see Table 9-2. For INVERTER_CFG_CURRENT_ADC_REF_VOLTAGE and INVERTER_CFG_SHUNT_RESIST, see 9.6.2.

7.5 Current Control Function

The current control function uses the value of the incoming current to perform coordinate transformation and feedback control that are necessary for vector control and then calculates the voltage of the PWM output. This module also controls submodules including the decoupling, voltage phase lead compensation, voltage error compensation, and mechanical angular speed / electrical angle detection. Figure 7-8 shows the module configuration.

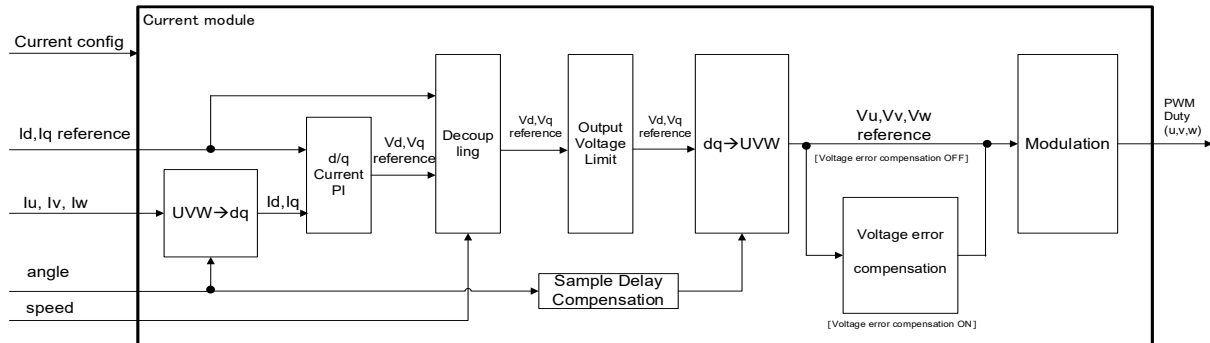


Figure 7-8 Functional block diagram of current control.

7.5.1 Decoupling Control

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

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

i_d^*, i_q^* : Current command value [A], ω : Rotational speed (electrical angle) [rad/s],
 R: Primary resistance of the motor [Ω], L_d, L_q : Inductance of the motor [H],
 Ψ : Magnetic flux linkage of the motor [Wb]

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

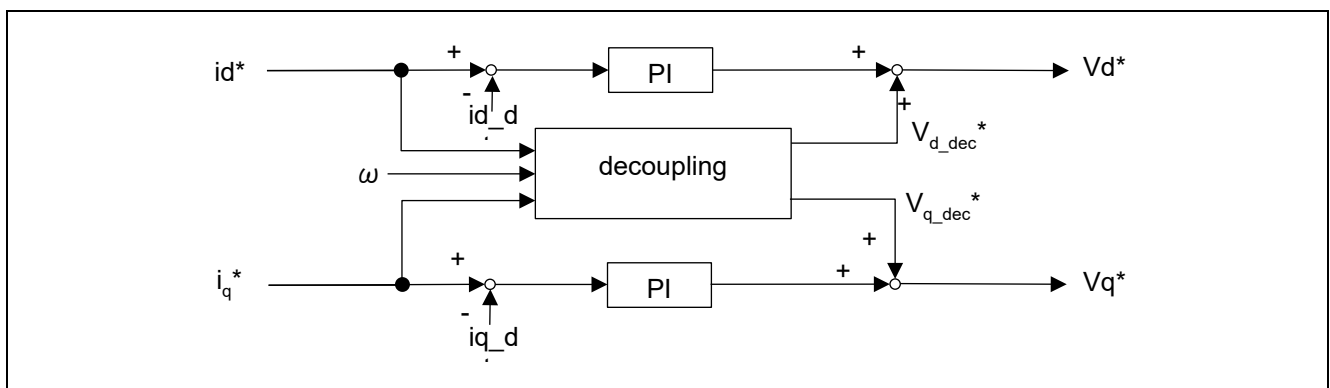


Figure 7-9 Functional block diagram of decoupling control.

7.5.2 Voltage Phase Lead Compensation

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

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

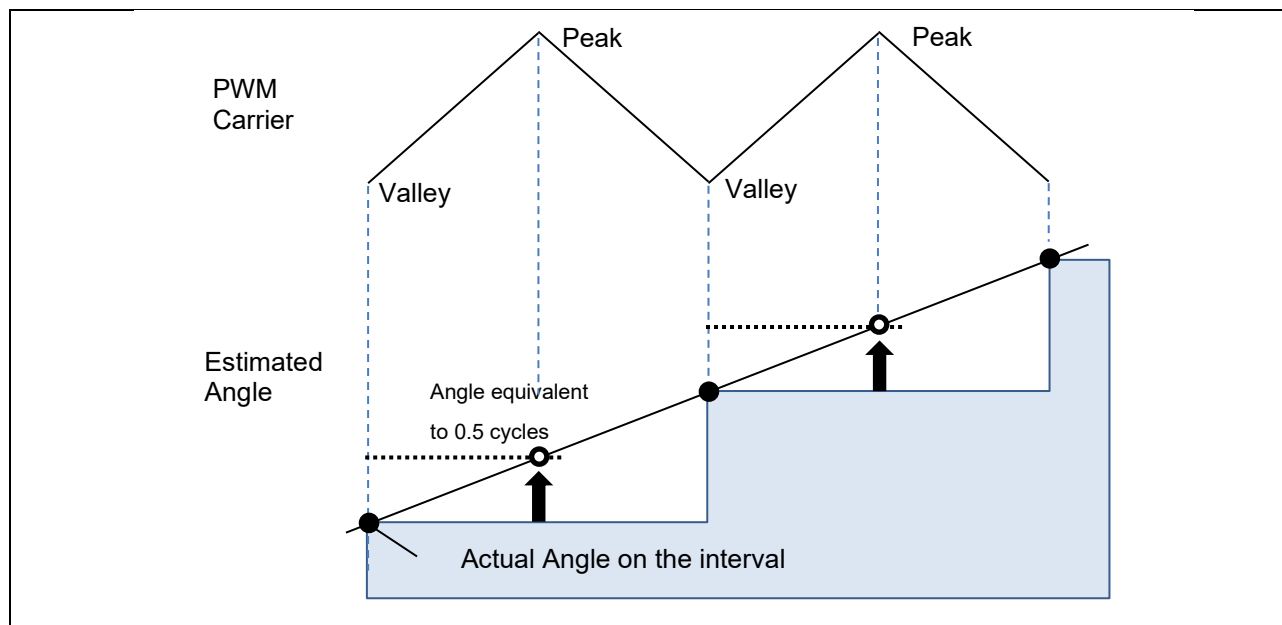


Figure 7-10 Example of the amount of angular advanced in a PWM carrier cycle.

7.5.3 Voltage Error Compensation

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

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

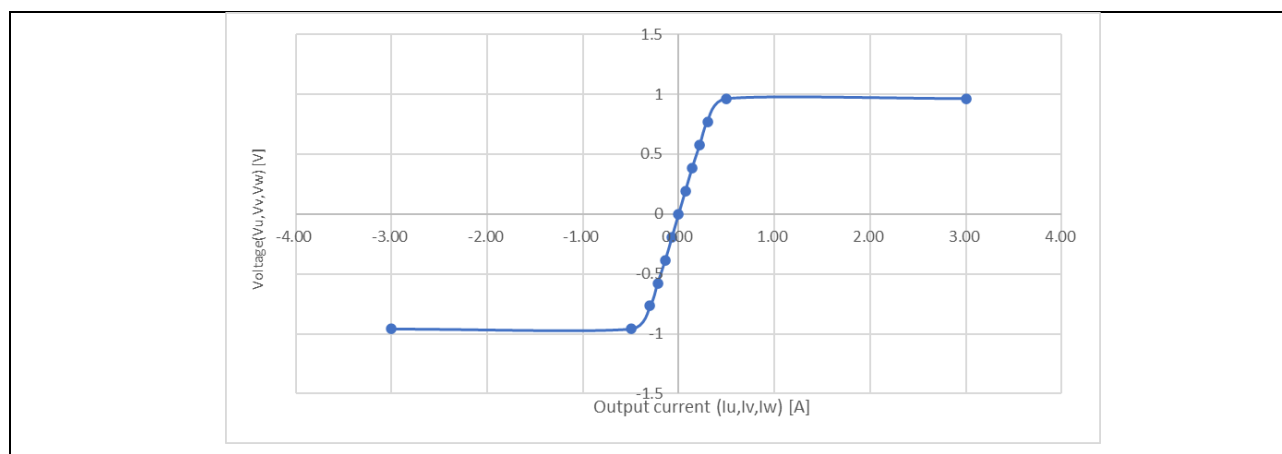


Figure 7-11 Current dependency of voltage error (example)

7.5.4 Pulse Width Modulation (PWM) Mode

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

(a) Sinusoidal modulation (MOD_METHOD_SPWM)

The modulation rate m is defined as follows:

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

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

(b) Space Vector Modulation (MOD_METHOD_SVPWM)

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

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

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

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

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

The modulation rate m is defined as follows:

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

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

7.6 Speed Control Function

The speed control function performs PI control so that the motor follows the speed command. When receiving a speed command value, the internal speed regulator outputs a current command value based on the deviation from the estimated speed value. This function also controls the submodules including flux-weakening control and MTPA control.

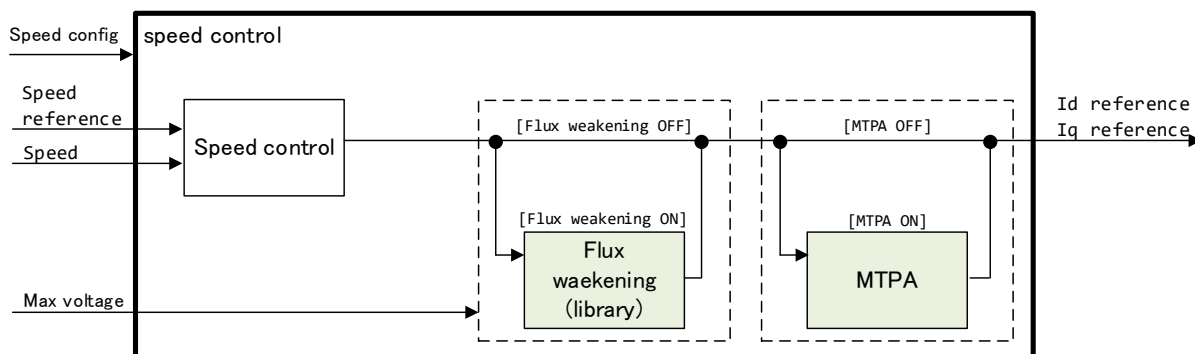


Figure 7-12 Functional block diagram for speed control.

7.6.1 Flux-weakening Control

The flux-weakening control is a method that even under the condition where the inductive voltage ($=\omega\Psi$) generated proportional to the PM motor rotation exceeds the maximum output of the bus voltage, it controls the d-axis current in the negative direction and increases the d-axis current command value by the negative value to cancel it (Figure 7-13). By canceling the voltage saturation, it is possible to increase the q-axis current command value required for acceleration, realizing high-speed rotation and improvement of output in the high-speed rotation range.

The flux-weakening control automatically detects the state in which the motor rotation speed is high for the current voltage and the voltage margin is severe, then increases the value of I_d^* in the negative direction and cancels the inductive voltage according to the voltage equation of the PM motor.

For this control, first, calculate the inductive voltage limit value from the following formula, where R is the resistance value of the motor, and I_a is the sum of squares of the I_d and I_q detection values ($\sqrt{I_d^2 + I_q^2}$). For V_{max} , use the maximum value of the voltage vector that has been pre-calculated by voltage error compensation or modulation processing.

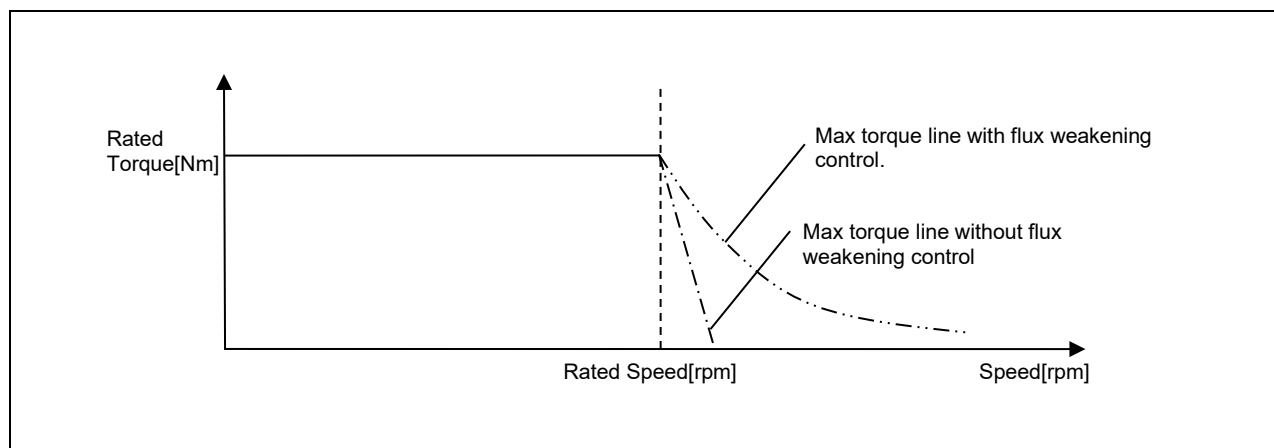


Figure 7-13 Example of the relationship between output torque and speed.

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

V_{om} : Inductive voltage limit [V], V_{amax} : Maximum value of voltage vector [V]

I_a : Magnitude of the current vector [A]

Figure 7-14 Formula for calculating the inductive voltage limit.

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

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

V_{om} : Inductive voltage limit [V], V_{amax} : Maximum value of voltage vector [V],
 I_a : Magnitude of current vector [A]

Figure 7-15 Formula for calculating the d-axis current command value in flux-weakening control.

7.6.2 Maximum Torque per Current Control (MTPA)

For a PM motor having saliency like an IPM motor, maximum torque per current control (MTPA) can be applied. MTPA uses reluctant torque, which is not used in control with $I_d = 0$, and is therefore capable of adjusting the torque per unit current to the maximum. Note that the reluctant torque for an SPM motor, which does not have saliency, is difficult to use due to the motor's structure and therefore cannot be used with MTPA. When the motor in use is an SPM motor, the L_d and L_q values of which are about the same, be sure to disable MTPA. Otherwise, this sample program will not work correctly. The sample program does not automatically determine whether to enable MTPA by monitoring the L_d and L_q values.

The equation used is shown below. The d-axis current command value can be obtained using the q-axis current command value I_q^* output by the speed regulator as input.

$$I_d^* = \frac{\Psi}{2(L_q - L_d)} - \sqrt{\left(\frac{\Psi}{2(L_q - L_d)}\right)^2 + I_q^{*2}}$$

Ψ : Magnetic flux linkage (Wb), L_d , L_q : d-axis inductance and q-axis inductance of the motor (H)

7.7 Position control

The position control function performs P control so that the motor follows the position command. After receiving the input of the position command value, the internal position controller outputs the speed command value based on the deviation from the current position.

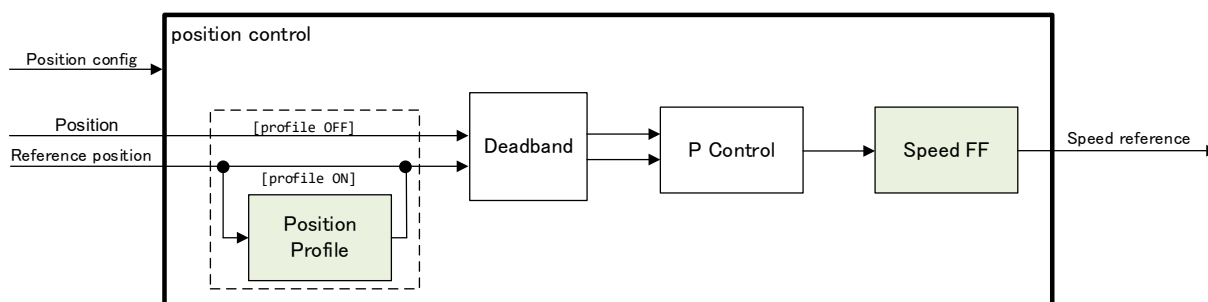


Figure 7-16 Functional block diagram for position control.

7.7.1 Position profile

The position profile has a function that controls the speed command value by recalculating the speed command value at each control interval based on the position command value that is preset, the acceleration/deceleration time, and the maximum speed (moving-average acceleration/deceleration algorithm). Figure 7-17 shows an overview of this algorithm. If the speed obtained from the position deviation and acceleration time exceeds the maximum speed, a position command value is created so that the speed command value forms a trapezoidal shape.

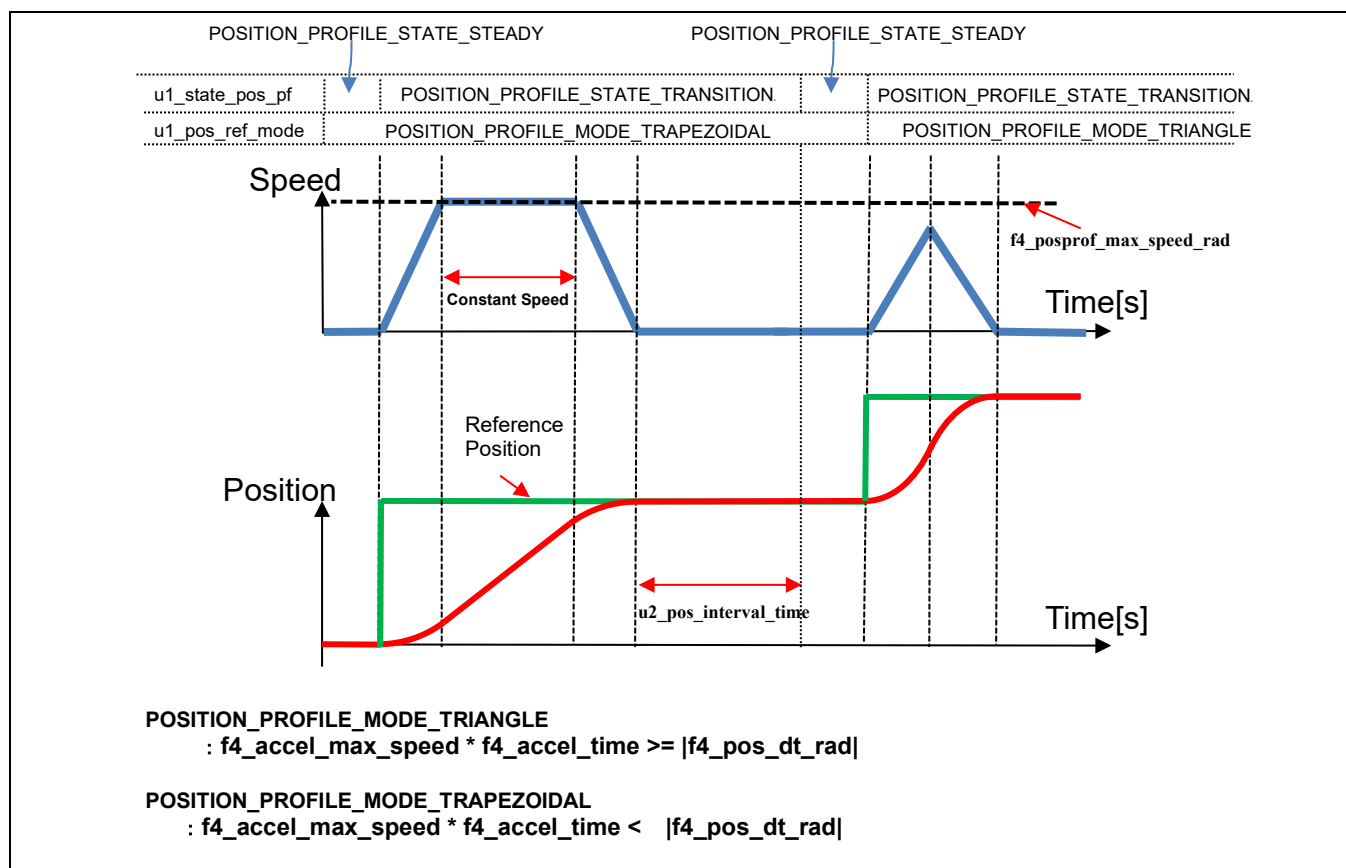


Figure 7-17 Position profile.

7.8 Encoder vector control

7.8.1 Encoder communication time compensation

The time when the position is determined by communication with the serial encoder and the time when the current is controlled are different. Use the following formula to compensate for the mechanical angle obtained using the encoder from the rotational speed [rad/s] and the compensation time [s]. The compensation time refers to the time between when the encoder count is finalized and when current control begins.

mechanical angle after compensation [rad]

$$= \text{mechanical angle calculated from encoder count[rad]} + \text{speed} \left[\frac{\text{rad}}{\text{s}} \right] \times \text{compensation time[s]}$$

In this sample program, it is assumed that the encoder value becomes fixed when the FA-CODER transmission completion interrupt finishes (approximately 5 μs after the GPTm_n_UDF event). Current control starts approximately 2.5 μs after the GPTm_n_UDF event of the next cycle. The transmission time to be compensated is 47.5 μs (= 50 μs + 2.5 μs - 5 μs).

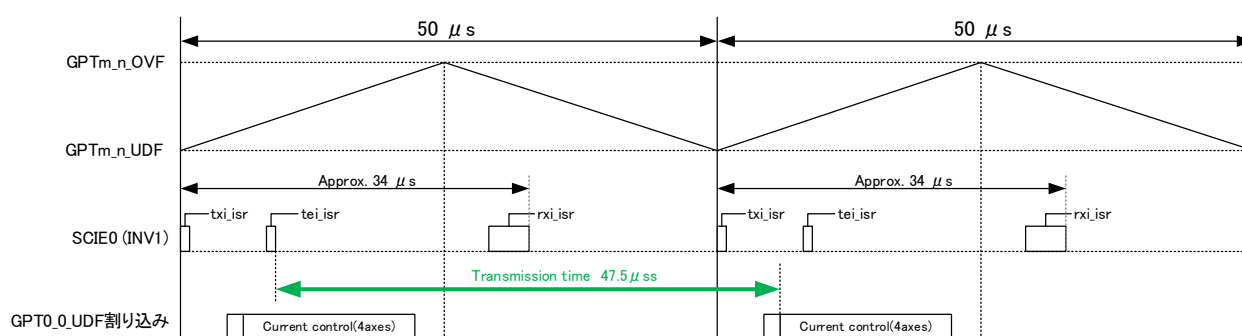


Figure 7-18 Compensation time of FA-CODER

7.8.2 Speed detection

This process detects the speed from the difference between the encoder count of the previous cycle and the current cycle. Figure 7-19 shows the flowchart of speed detection process.

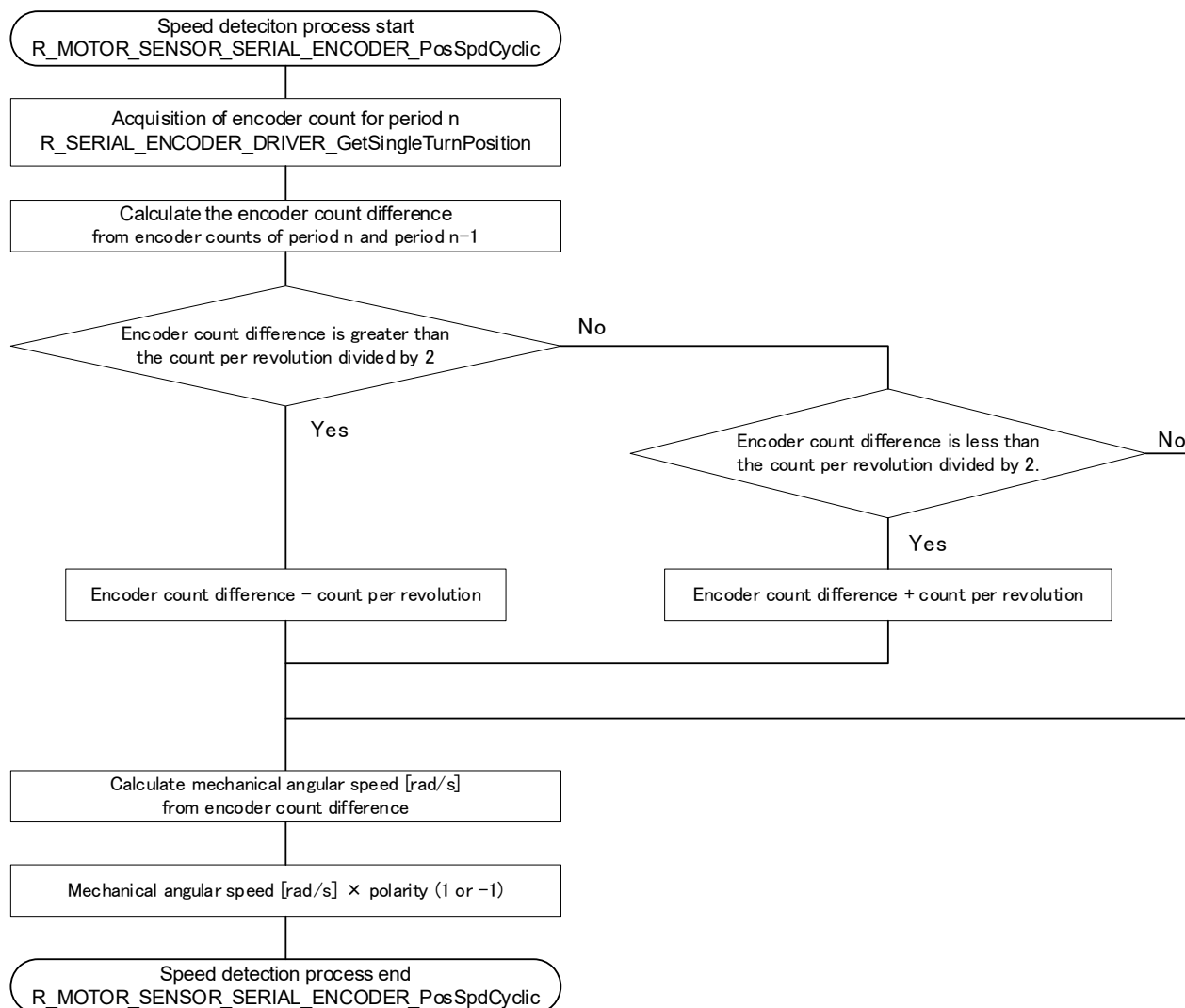


Figure 7-19 Speed detection process.

7.8.3 Multi-turn position detection

This process detects that the motor has rotated one turn from the difference between the encoder count of the previous cycle and the current cycle and detects multiple rotation angles. Figure 7-20 shows multiturn position detection process.

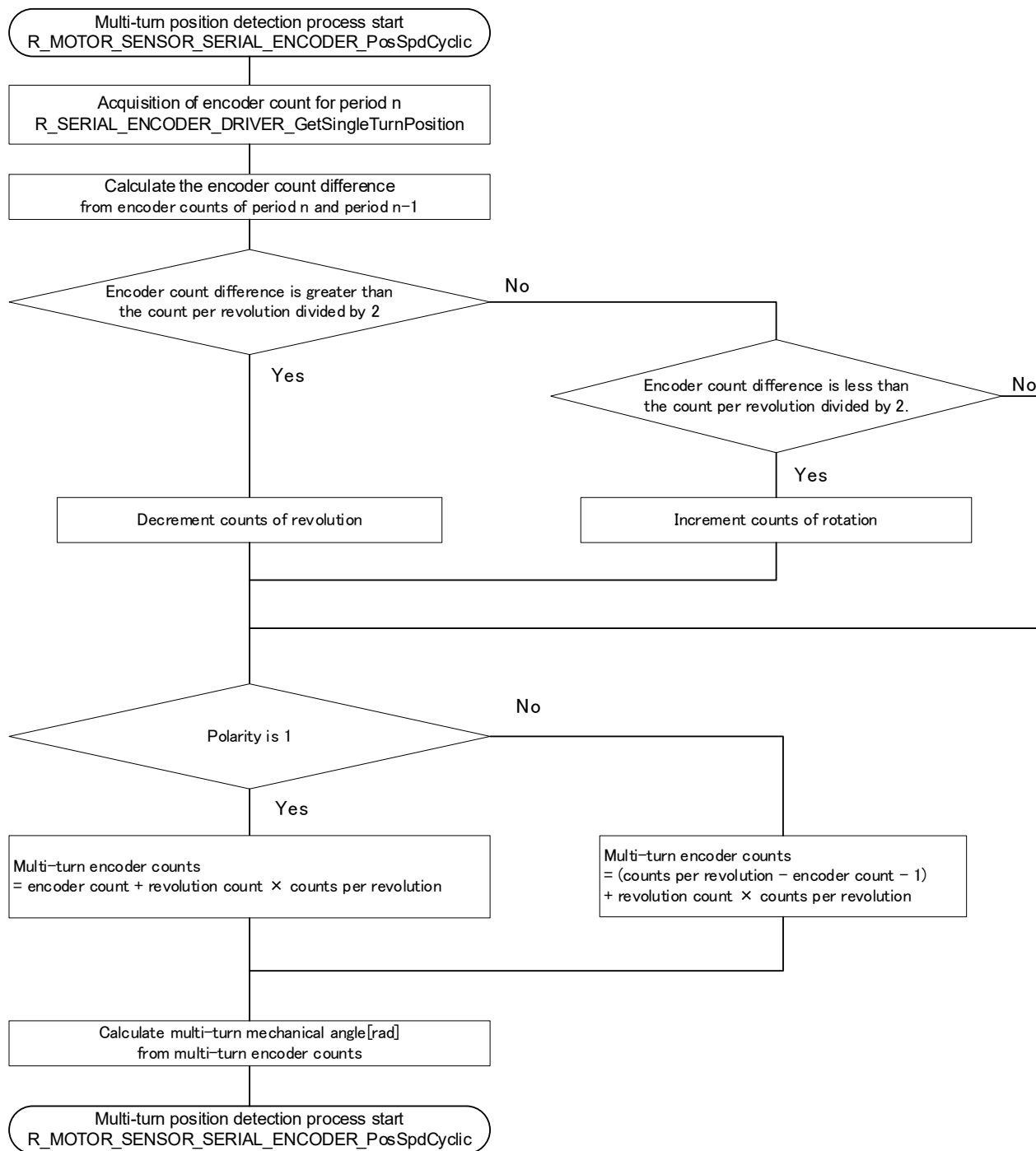


Figure 7-20 Multiturn angle detection process.

7.8.4 Position offset

The magnetic pole position must be aligned with the reference angle of the sensor. This system determines the initial magnetic pole position in the sequence shown in Figure 7-21 so that the directions of the d axis and current vector match. The angle of this adjustment is held as an offset value and is subtracted from the detected angle during motor control. Figure 7-22 shows the startup sequence that applies in this case.

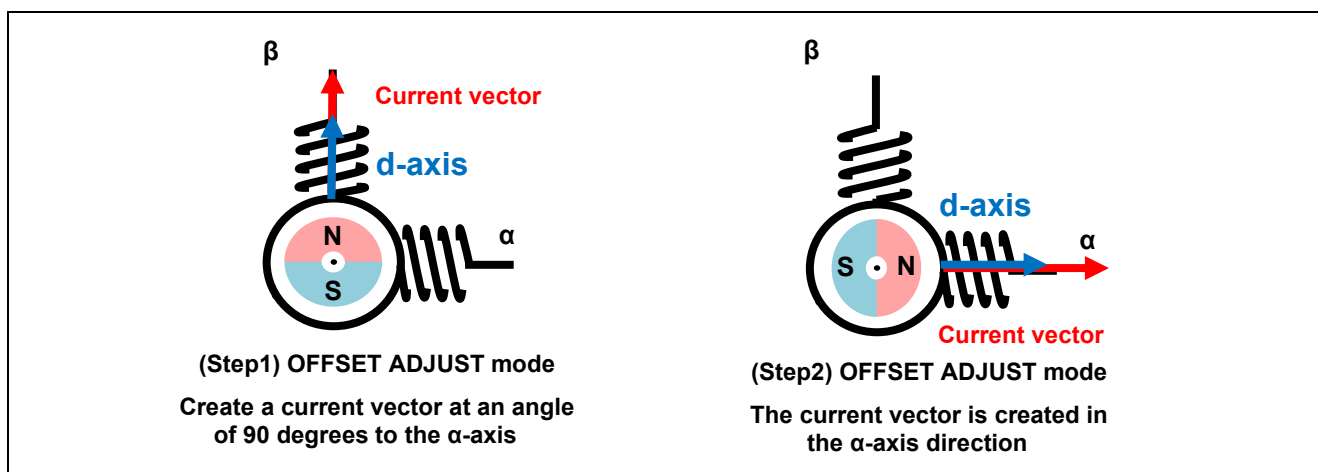


Figure 7-21 Determining the position of a permanent magnet.

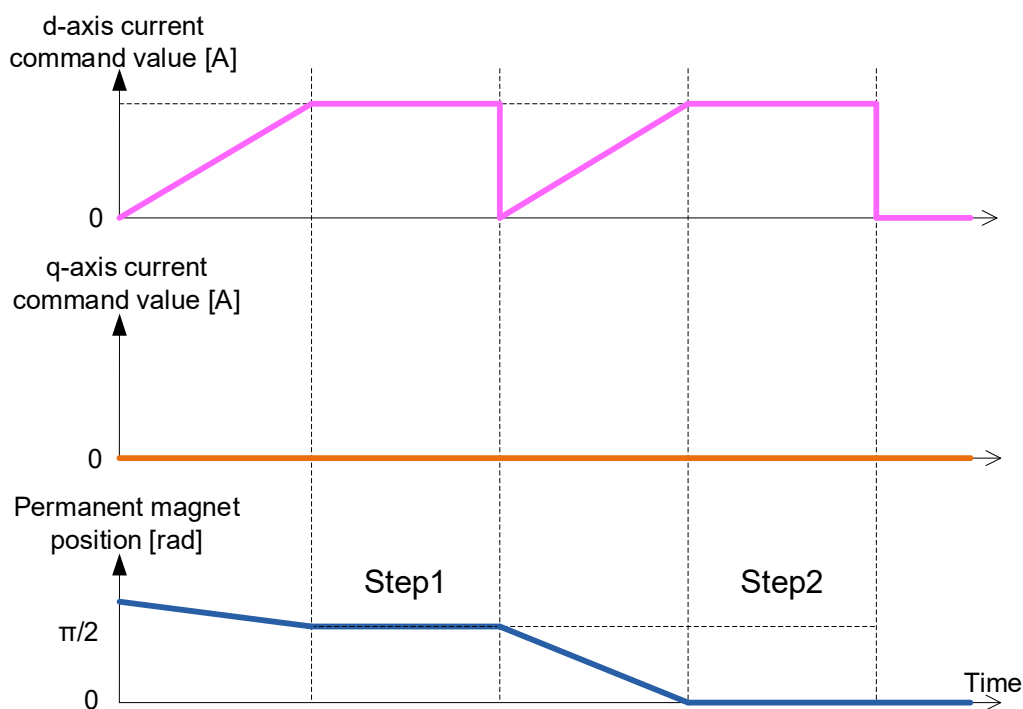


Figure 7-22 Q-axis current command value, D-axis current command value, rotor position (example)

8. Software Specifications & Configurations

8.1 Software Specifications

Table 8-1 shows software specifications.

Table 8-1 Software specifications

Item	Description
Control method	Encoder vector control
Motor control start / stop	RMW
Control mode	Position control, Speed control, Torque control, Open loop control
Current sensing	Current sensing using RV1S9355, the delta sigma modulator Filter setting: Decimation ratio is 250, the order of sinc filter is 3
Rotor position detection	17bit serial encoder (FA-CODER)
Carrier frequency	20 kHz
Dead time	1 μ s
PWM modulation	Space vector modulation, Sinusoidal modulation
Current control period	50 μ s
Speed control period	200 μ s
Position control period	200 μ s
Speed command range	0 rpm ~ 6000 rpm (CW, CCW)
Position command range	-360000 deg ~ 360000 deg
Position dead band	FA-CODER: +/-3 counts (+/- 0.008 deg)
Natural frequencies	Current control: 750 Hz Speed control: 30 Hz Position control: 7 Hz
Protection	Overcurrent error (S/W detection, monitoring period: 50 μ s) Overvoltage error (monitoring period: 50 μ s) Undervoltage error (monitoring period: 50 μ s) Overspeed error (monitoring period: 50 μ s) Overcurrent error (H/W detection, Transition the PWM output pin to a high-impedance state upon OC signal detection)
Compiler optimization	Optimize for debug (-Og)

8.2 Overall Configuration of the Software

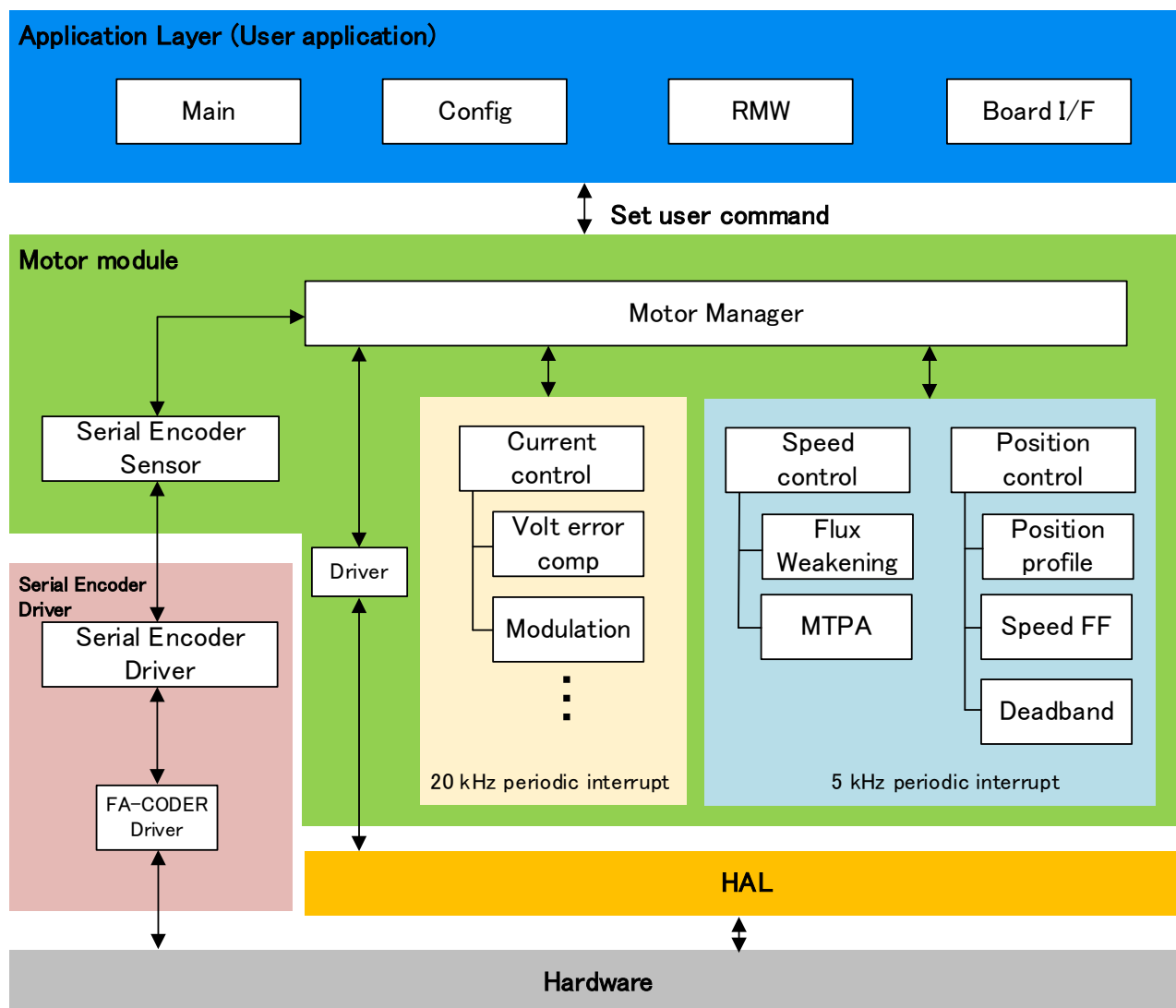


Figure 8-1 Overall configuration of motor control software.

8.3 Interrupts

Table 8-2 Interrupts

Interrupt	Peripheral	Priority (*1)	Period	Multiplex interrupt
HW over current interrupt	POEG	0	-	Disable
FA-CODER transmit interrupt (txi_isr)	SCIE	2	-	Disable
FA-CODER transmit end interrupt (tei_isr)	SCIE	2	-	Disable
FA-CODER receive interrupt (rx_i_isr)	SCIE	2	-	Disable
GPTm_n_UDF interrupt (m: unit number, n: channel number)	GPT	4	50 μ s	Enable

*1 If the number is small, the priority is high, and if the number is large, the priority is low.

- HW over current interrupt
When an overcurrent fault occurs on the inverter board, the POEG interrupt triggers the execution of the overcurrent fault handling process.
- FA-CODER transmit interrupt (txi_isr)
After a GPTm_n_UDF (underflow of GPTm_n timer count) occurs, the TIE and TE bits in the Common Control Register 0 (CCR0) are set to 1, and the FA-CODER transmission interrupt starts. Since the FA-CODER transmission interrupt has a higher priority than the GPTm_n_UDF interrupt, when GPTm_n_UDF occurs, the FA-CODER transmission interrupt is processed before the GPTm_n_UDF interrupt.
- FA-CODER transmit end interrupt (tei_isr)
The FA-CODER transmission completion interrupt starts approximately 5 μ s after the FA-CODER transmission interrupt occurs. This interrupt also has a higher priority than the GPTm_n_UDF interrupt. Because GPTm_n_UDF allows nested interrupts, the FA-CODER transmission completion interrupt can interrupt the processing of the GPTm_n_UDF interrupt.
- FA-CODER receive interrupt (rx_i_isr)
The FA-CODER receive interrupt starts approximately 27 μ s after the FA-CODER transmission completion interrupt occurs. This interrupt handles the acquisition of the received data.
- GPTm_n_UDF interrupt
The GPTm_n_UDF interrupt starts after the FA-CODER transmission interrupt has completed. In this interrupt, position and speed control for one axis (control cycle: 200 μ s) is performed sequentially for Axis 1 (INV1), Axis 2 (INV2), Axis 3 (INV3), and Axis 4 (INV4). Additionally, communication with RMW occurs once every 250 μ s. Current control for four axes (control cycle: 50 μ s) is also executed.

8.4 Files and Folders

Table 8-3 Files and folders

Files and folders				Description
rz	arm	CMSIS_6	-	Arm-core related files
	board	custom	-	CPU board related files
	fsp	inc	api	FSP related structures, enums, and functions
			instances	
			fsp_features.h	
			fsp_version.h	
		src	bsp	BSP
			r_adc	ADC
			r_dmac	DMAC
			r_dsmif	DSMIF
			r_elc	ELC
			r_gpt	GPT
			r_gpt_three_phase	GPT
			r_ioport	IO Port
			r_poeg	POEG
r_sci_uart	SCI			
r_wdt	WDT			
rz_gen	-			Files generated by FSP
src	application	main	mtr_interrupt.c / .h	Interrupts
			mtr_main.c / .h	Main process
		mcu/rzt2n	r_app_mcu_callback.c	FSP defined interruption function
			r_app_mcu.c / .h	Application process
			r_motor_driver_fsp.c	Wrapper functions for FSP
			r_motor_driver_hal.h	
		motor_module	cfg	motor module configuration file
			current	Current control module
			driver	Driver module
			general	Common functions
			position	Position control module
			sensor	Serial encoder sensor module
			serial_encoder_vector	Motor manager
			speed	Speed control module

		user_interface	board_ui	LED control
			ics	RMW related files
	serial_encoder_driver	mcu/rzt2n	FA-CODER	FA-CODER driver
		r_fac_if.h	-	FA-CODER header file
		r_serial_encoder_cfg.h	-	Define macros for serial encoder
		r_serial_encoder_driver.c/.h	-	Serial encoder driver
	hal_entry.c			hal_entry
Debug	-			Build outputs
rz_cfg	-			FSP module configuration file
script	-			Linker script

8.5 Application Layer

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

8.5.1 Functions

Table 8-4 lists the functions that are configured in the application layer.

Table 8-4 Functions available in the application layer

Function	Description
Main processing	Enables or disables each user command in the system.
UI Processing	Manages processes related to RMW.
RMW UI Processing	Acquires and set parameters (including command values).

8.5.2 Configuration

Table 8-5 shows the configurations used in the application layer.

Table 8-5 List of configurations

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

Table 8-6 List of initial values for configurations

Macro name	Set value
APP_CFG_FREQ_BAND_LIMIT	3.0f
APP_CFG_MAX_CURRENT_OMEGA	1500.f
APP_CFG_MIN_OMEGA	1.0f
APP_CFG_SCI_CH_SELECT	ICS_SCI4_P067_P066

8.5.3 Structure and Variable Information

Table 8-7 lists the member variables of `st_rmw_com_t` structure that can be used by users in the application layer. Table 8-8 lists the members of the structure provided for updating the motor module parameters by using RMW.

When you use RMW to set a value to a variable shown in Table 8-7, the application layer reflects the updated value in the variable in each control module using the Update function of each module via the structure shown in Table 8-8.

Table 8-7 List of member variables of `st_rmw_com_t` structure

Variable	Description
<code>u1_system_mode</code>	System mode 0 : Stop 1 : Position control 2 : Speed control 3 : Torque control 4 : Open loop control 5 : Reset error 8 : Current offset adjustment 9 : Position offset adjustment
<code>u1_enable_write</code>	Whether to enable rewrite of variables for user entry
<code>u2_offset_calc_time</code>	Current offset value calculation time setting
<code>f4_offset_rotor_rad</code>	Position offset [rad]
<code>u2_mtr_pp</code>	Number of pole pairs of the motor to be driven
<code>f4_mtr_r</code>	Resistance of the motor to be driven [Ω]
<code>f4_mtr_ld</code>	d-axis inductance of the motor to be driven [H]
<code>f4_mtr_lq</code>	q-axis inductance of the motor to be driven [H]
<code>f4_mtr_m</code>	Magnetic flux of the motor to be driven [Wb]
<code>f4_mtr_j</code>	Rotor inertia of the motor to be driven [kgm^2]
<code>f4_nominal_current_rms</code>	Rated current of the motor to be driven [Arms]
<code>f4_max_speed_rpm</code>	Maximum speed (mechanical angle) of the motor to be driven [rpm]
<code>f4_current_omega_hz</code>	Natural frequency for current control system [Hz]
<code>f4_current_zeta</code>	Attenuation coefficient for the current control system
<code>f4_speed_omega_hz</code>	Natural frequency for speed control system [Hz]
<code>f4_speed_zeta</code>	Attenuation coefficient for the speed control system
<code>f4_speed_rate_limit_rpm</code>	Maximum increment/decrement width for the speed command [rpm/s] (used when speed control is enabled, mechanical angle)
<code>f4_overspeed_limit_rpm</code>	Speed limit value (mechanical angle) [rpm]
<code>u1_pos_cmd_mode</code>	Position control command mode. 0: Position command value is 0 [rad] 1: Step 2: Trapezoidal speed
<code>u2_pos_interval_time</code>	Positioning wait time

Variable	Description
u2_pos_dead_band	dead band width [encoder count]
u2_pos_band_limit	in-position width [encoder count]
f4_pos_omega_hz	Natural frequency for position control system [Hz]
f4_pos_ff_ratio	Speed feed forward ratio
f4_accel_time	Acceleration / Deceleration time [s]
f4_posprof_max_speed_rpm	Maximum speed during trapezoidal speed control [rpm]
u1_flag_volt_err_comp_use	Flag for voltage error compensation setting 0: Disable, 1: Enable
u1_flag_fluxwkn_use	Flag when flux weakening is used. 0: Disable, 1: Enable
u1_flag_mtpa_use	Flag when MTPA is used. 0: Disable, 1: Enable
f4_torque_ref_nm	Torque command value [Nm]
f4_ref_speed_rpm	Speed command value [rpm]
f4_ref_position_deg	Position command value (mechanical angle) [deg]
f4_ol_crnt_ref	d-axis current command value during open-loop control [A]

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

Structure	Variable	Description
st_rmw_param_buffer_t Structure for updating RMW variables	u2_offset_calc_time	Current offset detection time setting
	f4_offset_rotor_rad	Magnetic pole position offset [rad]
	st_motor	Structure for motor parameters
	f4_max_speed_rpm	Maximum speed (mechanical angle) [rpm]
	f4_current_omega_hz	Natural frequency for current control system [Hz]
	f4_current_zeta	Attenuation coefficient for the current control system
	f4_speed_omega_hz	Natural frequency for the speed control system [Hz]
	f4_speed_zeta	Attenuation coefficient for the speed control system
	f4_speed_rate_limit_rpm	Speed variation limit (mechanical angle) [rpm/s]
	f4_overspeed_limit_rpm	Speed limit value (mechanical angle) [rpm]
	u1_pos_cmd_mode	Position control command mode. 0: Position command value is 0 [rad] 1: Step 2: Trapezoidal speed
	u2_pos_interval_time	Positioning wait time
	u2_pos_dead_band	dead band width [encoder count]
	u2_pos_band_limit	in-position width [encoder count]

Structure	Variable	Description
	f4_pos_omega_hz	Natural frequency for position control system [Hz]
	f4_pos_ff_ratio	Speed feed forward ratio
	f4_accel_time	Acceleration / Deceleration time [s]
	f4_posprof_max_speed_rpm	Maximum speed during trapezoidal speed control [rpm]
	u1_flag_volt_err_comp_use	Flag for whether to use voltage error compensation 0: Disable, 1: Enable
	u1_flag_fluxwkn_use	Flag for whether to use flux-weakening 0: Disable, 1: Enable
	u1_flag_mtpa_use	Flag for whether to use MTPA 0: Disable, 1: Enable
	f4_torque_ref_nm	Torque command value [Nm]
	f4_ref_speed_rpm	Speed command value (mechanical angle) [rpm]
	f4_ref_position_deg	Position command value [deg]
	f4_ol_crnt_ref	Current command value for open-loop [A]

8.5.4 Macro Definitions

Table 8-9 List of macros

File name	Macro name	Value	Description
r_app_mcu.h	POSITION_SPEED_CONTROL_DECIMATION	3	Number of times to skip position-speed control process when a GTCNT underflow interrupt occurs
	ICS_DECIMATION	4	RMW watchpoint skip count
	ICS_BRR	15	Communication baud rate for RMW
	ICS_INT_MODE	1	Communication mode selection for RMW

8.5.5 Adjustment and configuration of parameters

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

For the variables listed in Table 8-7, perform adjustment and configuration from RMW. For details about how to use RMW, see 6.8 and Renesas Motor Workbench User's Manual (R21UZ0004).

8.6 Manager Modules and Motor Control Modules

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

8.6.1 Functions

Table 8-10 lists the functions of the manager module. Table 8-11, Table 8-12 and Table 8-13 list the functions of the motor control module

Table 8-10 List manager module functions

Function	Description
Mode Management	Switches the operation mode of the system in response to the user command to control the motor.
Protection function	Handles errors by using the system protection function.
Control method management	Acquires and sets the states of speed control and current control.
Speed and position information acquisition	Acquires the speed and position information from the speed control module and current control module.
Control module command value setting	Selects the command values to be entered to the current control module and speed control module based on the control states.
Interrupt processing	Assign processing to appropriate modules in response to interruptions set in FSP.
Wait process after transitioning to RUN state	When transitioning to RUN state, a wait process is performed until the signal from the delta-sigma modulator is output and current detection becomes possible.

Table 8-11 List of functions of the position control module

Function	Description
Position control	Calculating and outputs a speed command value that allows the position to follow the position command value.
Speed feed forward	Feed forwards the command speed to the output of the position controller.
Position profile	The position command value and the drive method (triangular and trapezoidal drive) are controlled from the difference in position command value.
Dead band	Detect that the rotor position has entered the dead zone and set the position deviation to 0 [rad/s].

Table 8-12 List of functions of the speed control module

Function	Description
Speed control	Calculates and outputs a current command value that allows the speed to follow the speed command value.
Speed command setting	Sets a speed command value in the speed control module.
Flux-weakening control	Controls the d-axis current to enable the operation over the rated rotational speed.
MTPA control	By using a reluctant torque, it is controlled to maximize torque.

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

Function	Description
Current control	Performs calculation according to the current command value to set the PWM output value.
Current offset adjustment	Calculates the offset value of the current value detected by AD.
Voltage error compensation	Compensates for the effects of output voltage dead time.
Forward and inverse transformation	Performs coordinate transformation for the current value detected to perform vector control. This function also performs inverse transformation of the coordinate for the calculation results to restore the original coordinate axis.
Modulation	Improves the efficiency by modulation to a PWM signal.
Decoupling control	Calculates interference cancellation to prevent interference between the d and q axes.
Sample delay compensation	Compensates for sample delay for current control cycles when generating a three-phase voltage command value.

Table 8-14 List of functions of the serial encoder sensor module

Function	Description
Encoder counts	Use the serial encoder driver to obtain encoder counts.
Mechanical angle	Converts encoder counts to mechanical angle [rad].
Encoder communication time compensation	Compensates for mechanical angle based on speed and compensation time.
Mechanical angular speed	Calculate the mechanical angular speed [rad/s] from the difference in encoder counts.
Multiturn mechanical angle	Calculate the multi-turn mechanical angle [rad] from the number of revolutions and the encoder count.

8.6.2 Module Configuration

Figure 8-2 shows module configuration.

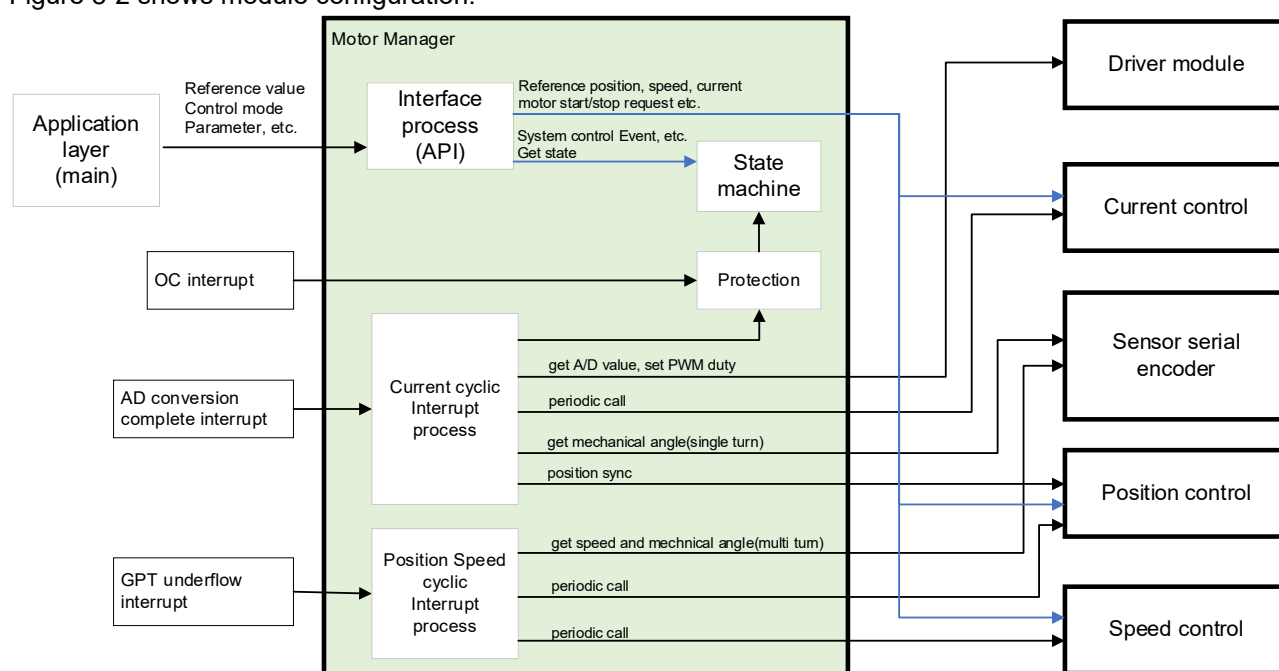


Figure 8-2 Module configuration.

8.6.3 Mode Management

Figure 8-3 shows the state transition diagram in this sample program. In this sample program, the states are managed by using two types of modes: "SYSTEM MODE" and "RUN MODE".

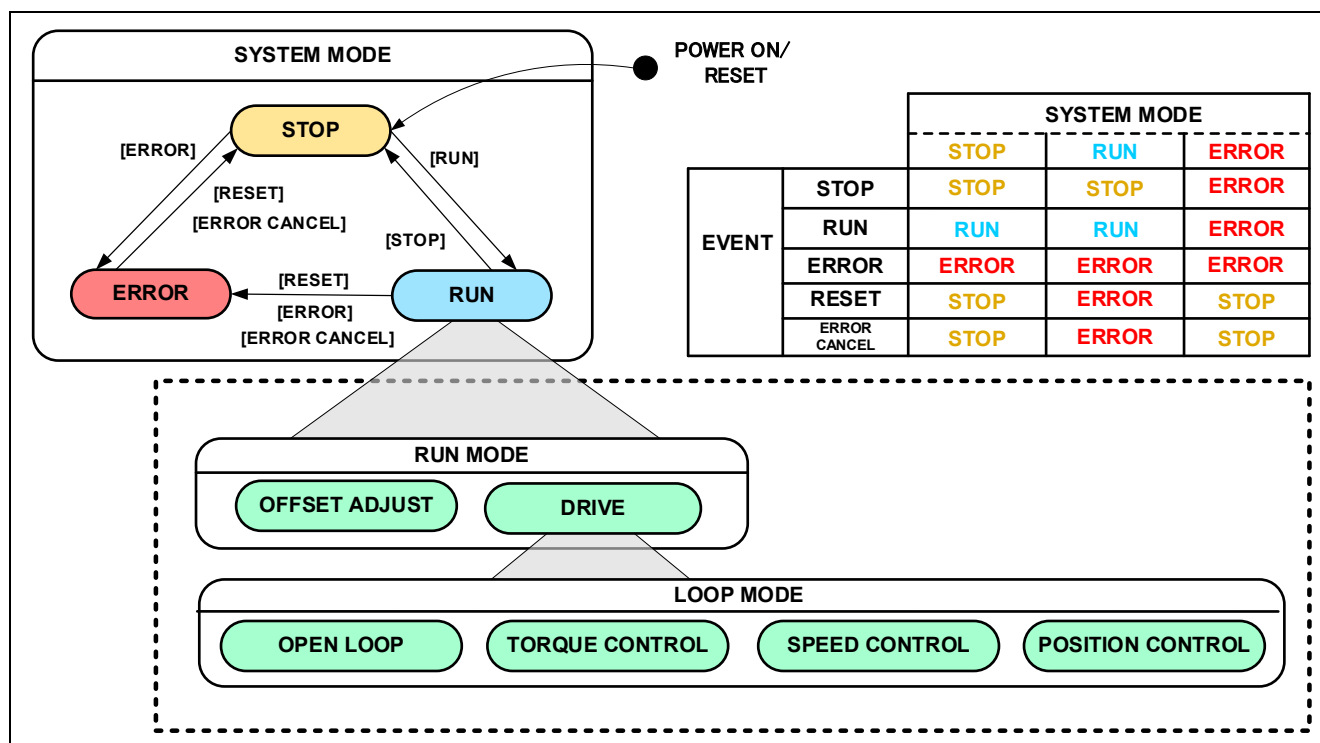


Figure 8-3 State transition diagram for encoder vector control software.

(1) SYSTEM MODE

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

(2) RUN MODE & LOOP MODE

These modes are used to indicate the state of motor control. When the system becomes ACTIVE mode, the motor is driven or calibrated according to the configuration settings shown in Figure 8-3.

(3) EVENT

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

Table 8-15 EVENT LIST

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

8.6.4 Protection Function

This control program provides the following error states and implements an emergency stop function in each error case. For details about the values that can be specified for the settings of the system protection function, see Table 8-16.

- 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).

The overcurrent limit value is automatically calculated from the rated current of the motor (MOTOR_CFG_NOMINAL_CURRENT_RMS).

- Overvoltage error

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

- Low-voltage error

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

- Overspeed 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 8-16 Operating conditions and setting values for the system protection functions.

Overcurrent error (S/W detection)	Overcurrent limit value [A]	12 A
	Monitoring interval [μ s]	50 μ s (Current control period)
Overcurrent error (H/W detection)	Overcurrent limit value [A]	25 A
	Overvoltage limit value [V]	28 V
Overvoltage error	Monitoring interval [μ s]	50 μ s (Current control period)
	Low-voltage limit value [V]	20 V
Low-voltage error	Monitoring interval [μ s]	50 μ s (Current control period)
	Speed limit value [rpm]	7200 rpm
Overspeed error	Monitoring interval [μ s]	50 μ s (Current control period)

8.6.5 Wait process after transitioning to RUN state

The delta-sigma modulator on the RZ/T Series Inverter Board does not output the MCLK signal or MDAT signal when no PWM signal is input to the inverter board. See 7.4.1 for details on the phase current sensing circuit.

The motor manager performs a wait operation between the transition to the RUN state (where PWM output is permitted) and the point when signals are output from the delta-sigma modulator, enabling current sensing. We experimentally verified the time required for the MDAT signal and MCLK signal to be output from the delta-sigma modulator and set the wait time to 25 ms. The wait time is configured using INVERTER_CFG_DSM_POWER_ON_DELAY (Table 9-12).

8.6.6 API

Table 8-17 List of API functions

API	Description
R_MOTOR_SERIAL_ENCODER_VECTOR_Open	Generates an instance of the manager module. This function also uses the Open function of the control module to generate an instance.
R_MOTOR_SERIAL_ENCODER_VECTOR_Create	Receive the pointer to the instance of the module used with the motor manager and initialize the following modules. ▪ st_serial_encoder_vector_control_t (Motor manager) ▪ st_current_control_t (Current control module) ▪ st_speed_control_t (Speed control module) ▪ st_position_control_t (Position control module) ▪ st_motor_driver (Driver module) ▪ st_sensor_serial_encoder_t (Serial encoder sensor module)
R_MOTOR_SERIAL_ENCODER_VECTOR_Close	Places the modules, including the manager module, in a reset state.
R_MOTOR_SERIAL_ENCODER_VECTOR_Reset	Initializes the modules, including the manager module.
R_MOTOR_SERIAL_ENCODER_VECTOR_ParameterUpdate	Updates the control parameters of the manager module. This function also updates the control parameters for the related modules.
R_MOTOR_SERIAL_ENCODER_VECTOR_MotorStart	Places the motor in the running state.
R_MOTOR_SERIAL_ENCODER_VECTOR_MotorStop	Places the motor in the stopped state.
R_MOTOR_SERIAL_ENCODER_VECTOR_MotorReset	Releases the system from the error state. Places each module in a software reset state.
R_MOTOR_SERIAL_ENCODER_VECTOR_MotorErrorCancel	Releases the system from the error state. Each module will be initialized.
R_MOTOR_SERIAL_ENCODER_VECTOR_ErrorSet	Places the system in an error state.
R_MOTOR_SERIAL_ENCODER_VECTOR_PositionSet	Sets the position command value. This function is enabled when position control is being performed.
R_MOTOR_SERIAL_ENCODER_VECTOR_PositionGet	Acquires the position information.
R_MOTOR_SERIAL_ENCODER_VECTOR_SpeedSet	Sets the speed command value. This function is enabled when speed control is being performed.
R_MOTOR_SERIAL_ENCODER_VECTOR_SpeedGet	Acquires speed information.
R_MOTOR_SERIAL_ENCODER_VECTOR_StatusGet	Acquires status from the state machine.
R_MOTOR_SERIAL_ENCODER_VECTOR_ErrorStatusGet	Acquires the error state.
R_MOTOR_SERIAL_ENCODER_VECTOR_CtrlTypeSet	Sets each mode.
R_MOTOR_SERIAL_ENCODER_VECTOR_LoopModeStatusGet	Acquires the loop mode.
R_MOTOR_SERIAL_ENCODER_VECTOR_PositionCommandModeSet	Allows the user to select the generation mode of position commands for position control. To perform position control, set 1 or 2. 0: The position is always 0. 1: Step wave response operation 2: Trapezoidal driving operation
R_MOTOR_SERIAL_ENCODER_VECTOR_InPositionFlagGet	Acquires the position control completion state. This function is enabled when position control is being performed. 0: Position control is not completed yet.

API	Description
	1: Position control has been completed.
R_MOTOR_SERIAL_ENCODER_VECTOR_RotorAngleOffsetSet	Sets the magnetic pole position offset.
R_MOTOR_SERIAL_ENCODER_VECTOR_CurrentOffsetSet	Sets the current offset.
R_MOTOR_SERIAL_ENCODER_VECTOR_CurrentOpenLoopSet	Sets the current open loop command.
R_MOTOR_SERIAL_ENCODER_VECTOR_TorqueSet	Sets the torque command.
R_MOTOR_SERIAL_ENCODER_VECTOR_OffsetCalibrationStatusGet	Acquires the offset measurement state. 0: Measurement is in progress 1: Measurement is complete.
R_MOTOR_SERIAL_ENCODER_VECTOR_PosSpeedInterrupt	Performs interrupt processing for position control and speed control.
R_MOTOR_SERIAL_ENCODER_VECTOR_CurrentInterrupt	Performs interrupt processing for current control.
R_MOTOR_SERIAL_ENCODER_VECTOR_OverCurrentInterrupt	Performs interrupt processing when an overcurrent occurs.

Table 8-18 List of API functions for the current control modules

API	Description
R_MOTOR_CURRENT_Open	Generates an instance of the current control module.
R_MOTOR_CURRENT_Create	Receive the pointer to the instance of the current control module and initialize the instance.
R_MOTOR_CURRENT_Close	Places the current control module in a reset state.
R_MOTOR_CURRENT_Reset	Initializes the current control module.
R_MOTOR_CURRENT_Run	Activates the current control module.
R_MOTOR_CURRENT_ParameterSet	Inputs the variable information that is used for current control.
R_MOTOR_CURRENT_ParameterGet	Acquires the current control results that are output.
R_MOTOR_CURRENT_ParameterUpdate	Updates the control parameters of the current control module.
R_MOTOR_CURRENT_CurrentCyclic	Performs current control.
R_MOTOR_CURRENT_OffsetCalibration	Performs offset adjustment of current detection.
R_MOTOR_CURRENT_CurrentOffsetRemove	Returns the value with the current detection offset value excluded.
R_MOTOR_CURRENT_VoltErrCompParamSet	Set the voltage error compensation parameters.
R_MOTOR_CURRENT_CurrentOffsetSet	Set the AD current offset.

Table 8-19 List of API functions for the speed control modules

API	Description
R_MOTOR_SPEED_Open	Generates an instance of the speed control module.
R_MOTOR_SPEED_Create	Receive the pointer to the instance of the speed control module and initialize the instance.
R_MOTOR_SPEED_Close	Place the module in a reset state.

R_MOTOR_SPEED_Reset	Initializes the module.
R_MOTOR_SPEED_Run	Activates the module.
R_MOTOR_SPEED_ParameterSet	Inputs the variable information that is used for speed control.
R_MOTOR_SPEED_ParameterGet	Acquires speed control results that are output.
R_MOTOR_SPEED_ParameterUpdate	Updates the control parameters of the module.
R_MOTOR_SPEED_SpdRefSet	Sets the speed command value.
R_MOTOR_SPEED_SpeedCyclic	Performs speed control.

Table 8-20 List of API functions for the position control modules

API	Description
R_MOTOR_POSITION_Open	Generates an instance of the position control module.
R_MOTOR_POSITION_Create	Receive the pointer to the instance of the position control module and initialize the instance.
R_MOTOR_POSITION_Close	Places the position control module in a reset state.
R_MOTOR_POSITION_Reset	Initialize the position control module.
R_MOTOR_POSITION_Run	Activates the position control module.
R_MOTOR_POSITION_ParameterSet	Sets the parameters that are used for the position control loop.
R_MOTOR_POSITION_ParameterGet	Acquires the variable information of the position control module.
R_MOTOR_POSITION_ParameterUpdate	Updates the control parameters of the position control module.
R_MOTOR_POSITION_PositionCyclic	Performs the position control loop processing.
R_MOTOR_POSITION_CommandModeSet	Sets the position control command mode.
R_MOTOR_POSITION_PosRefSet	Sets the position command.
R_MOTOR_POSITION_Sync	Changes the position information. For example, this item is used to resume the processing from the position at which the processing was stopped previously.

Table 8-21 List of API functions for the serial encoder sensor module

API	Description
R_MOTOR_SENSOR_SERIAL_ENCODER_Open	Generate an instance of the serial encoder sensor module
R_MOTOR_SENSOR_SERIAL_ENCODER_Create	Receive the pointer to the instance of the serial encoder sensor module and initialize the instance.
R_MOTOR_SENSOR_SERIAL_ENCODER_CurrentCyclic	Get the encoder count and convert it to single turn mechanical angle [rad]
R_MOTOR_SENSOR_SERIAL_ENCODER_PosSpdCyclic	Get the encoder count and convert it to mechanical angular speed [rad/s] and multiturn mechanical angle [rad]
R_MOTOR_SENSOR_SERIAL_ENCODER_ParameterGet	Get parameters of serial encoder sensor module instance
R_MOTOR_SENSOR_SERIAL_ENCODER_ParameterUpdate	Update parameters of serial encoder sensor module
R_MOTOR_SENSOR_SERIAL_ENCODER_Close	Reset serial encoder sensor module instances

8.6.7 Structure and Variable Information

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

Structure	Variable	Description
st_serial_encoder_vector_t Structure for the manager module	u1_state_id_ref	State of the d-axis current command value: 0: Zero command 1: Id command input 2: Manual command input
	u1_state_iq_ref	State of the q-axis current command value: 0: Zero command 1: Iq command input 2: Manual command input
	u1_state_speed_ref	State of the speed command value: 0: Zero command 1: Speed command input 2: Manual command input
	u1_state_open_loop	Open loop state: 0: Calculates the control phase of current control based on the position from the position sensor (closed-loop) 1: Calculates the control phase of current control based on the instructed speed (open-loop)
	u1_direction	Rotation direction 0: CW 1: CCW
	u1_ctrl_loop_mode	Control loop mode: 0: d-axis current control 1: q-axis current control 2: Speed control 3: Position control
	u1_offset_adjust_mode	Offset removal mode: 0: Current offset 1: Position offset 2: Completion
	u1_flag_offset_calc	Offset calculation flag
	u2_error_status	Error state
	u2_error_status_mask	Error status bit mask, the error that the corresponding bit is set 0 will be ignored
	u2_run_mode	Motor driving mode: 0: Offset removal 1: Motor driving
	u1_flag_charge_cap	Voltage stability flag
	f4_time_elapsed_power_on	Time elapsed since power-on [s]
	f4_time_elapsed_pwm_output	Time elapsed since PWM output [s]
	f4_vdc_ad	Power supply voltage [V]
	f4_iu_ad	U-phase current [A]
	f4_iv_ad	V-phase current [A]
	f4_iw_ad	W-phase current [A]
	f4_overcurrent_limit	Overcurrent limit value [A]
	f4_overvoltage_limit	Overvoltage limit value [V]
	f4_undervoltage_limit	Low-voltage limit value [V]
	f4_overspeed_limit_rad	Overspeed limit value [rad]
	f4_user_open_loop_current	User open loop mode current command [A]
	f4_user_torque_nm	User torque command [Nm]

Structure	Variable	Description
	f4_open_loop_angle_rad	Current control phase for open loop control [rad]
	f4_rotor_angle_rad	Magnetic pole position (electrical angle) [rad]
	st_vdc_lpf	Low pass filter structure for bus voltage of inverter board
	f4_rotor_angle_offset_time	Position offset acquisition time [s]
	f4_rotor_angle_offset_rad	Position offset [rad]
	st_current_output	Output structure for the current control module
	st_speed_output	Output structure for the speed control module
	st_position_output	Output structure for the position control module
	st_sensor_output	Output structure for the sensor module
	st_stm	State machine structure
	st_motor	Motor parameter structure
	p_st_driver	Pointer to the structure for the driver module
	p_st_cc	Pointer to the structure for the current control module
	p_st_sc	Pointer to the structure for the speed control module
	p_st_pc	Pointer to the structure for the position control module
	p_st_sensor_serial_encoder	Pointer to the structure for serial encoder sensor control module
st_serial_encoder_vector_cfg_t Structure for setting the manager module control parameters	f4_overspeed_limit_rpm	Limit value for the speed error occurrence speed
	st_motor	Motor parameter structure

Table 8-23 List of structures and variables for current control modules

Structure	Variable	Description
st_current_control_t Structure for current control module	u1_active	The active state of the current control module
	u1_flag_volt_err_comp_use	Enables/disables voltage error compensation function
	u1_state_id_ref	Status of the d-axis input
	u1_state_iq_ref	Status of the q-axis input
	u1_flag_offset_calc	Flags for current offset calculation
	u2_offset_calc_time	Measurement time setting in current offset adjustment
	u2_offset_calc_wait	Current offset adjustment timing counter
	u2_crnt_offset_cnt	Measurement count in current offset adjustment
	f4_ctrl_period	Current control interval (period) [s]
	f4_refu	U-phase command voltage [V]
	f4_refv	V-phase command voltage [V]
	f4_refw	W-phase command voltage [V]
	f4_vd_ref	d-axis voltage command value [V]
	f4_vq_ref	q-axis voltage command value [V]
	f4_id_ref	d-axis current command value [A]
	f4_iq_ref	q-axis current command value [A]
	f4_id_ad	d-axis current value [A]
	f4_iq_ad	q-axis current value [A]
	f4_lim_iq	q-axis current limit [A]
	f4_offset_iu	U-phase offset current value [A]
	f4_offset_iv	V-phase offset current value [A]
	f4_offset_iw	W-phase offset current value [A]
	f4_sum_iu_ad	U-phase total current value [A]
	f4_sum_iv_ad	V-phase total current value [A]
	f4_sum_iw_ad	W-phase total current value [A]
	f4_vdc_ad	Bus voltage value [V]
	f4_iu_ad	U-phase current value [A]
	f4_iv_ad	V-phase current value [A]
	f4_iw_ad	W-phase current value [A]
	f4_modu	U-phase duty cycle
	f4_modv	V-phase duty cycle
	f4_modw	W-phase duty cycle

Structure	Variable	Description
	f4_speed_rad	Speed [rad/s]
	f4_rotor_angle_input_rad	Rotor angle [rad]
	f4_id_ref_manual	d-axis current status: d-axis current command value in d-axis fixed-command mode [A]
	f4_iq_ref_manual	q-axis current status: q-axis current command value in q-axis fixed-command mode [A]
	f4_ref_id_ctrl	d-axis current command value [A]
	f4_ref_iq_ctrl	q-axis current command value [A]
	f4_current_rate_limit	Current increase limit value
	f4_va_max	Maximum voltage on the d- and q- axes [V]
	st_mod	Structure for modulation
	st_volt_comp	Structure for the voltage error compensation
	st_pi_id	Structure for the d-axis PI control
	st_pi_iq	Structure for the q-axis PI control
	st_rotor_angle	Rotor information structure
	st_rotor_angle_phasecomp	Rotor Information structure (lead compensation)
	st_motor	Structure for motor parameters
st_current_cfg_t Structure for setting current control module control parameters	u2_offset_calc_time	Offset calculation time setting
	f4_ctrl_period	Control interval [s]
	f4_current_omega_hz	Natural frequency for current control system [Hz]
	f4_current_zeta	Attenuation coefficient for current control system
	u1_flag_volt_err_comp_use	Enables/disables voltage error compensation
	st_motor	Structure for motor parameters
st_current_output_t Structure for the output of the current control module	u1_flag_offset_calc	Current offset flag
	f4_modu	U-phase duty cycle
	f4_modv	V-Phase duty cycle
	f4_modw	W-phase duty cycle
	f4_neutral_duty	Duty ratio in offset measurement
	f4_va_max	Maximum voltage on the d- and q- axes [V]
st_current_input_t Structure for current control module input	u1_state_id_ref	Status of the d-axis
	u1_state_iq_ref	Status of the q-axis
	f4_rotor_angle_rad	Magnetic pole position (electrical angle) [rad]
	f4_iu_ad	U-phase current value [A]

Structure	Variable	Description
	f4_iv_ad	V-phase current value [A]
	f4_iw_ad	W-phase current value [A]
	f4_vdc_ad	Bus voltage value [V]
	f4_speed_rad	Speed [rad/s]
	f4_id_ref	d-axis current command value [A]
	f4_iq_ref	q-axis current command value [A]
	f4_id_ref_manual	d-axis current status: d-axis current command value in d-axis fixed-command mode [A]
	f4_iq_ref_manual	q-axis current status: q-axis current command value in q-axis fixed-command mode [A]

Table 8-24 List of structures and variables for the speed control module 1

Structure	Variable	Description
st_speed_control_t Structure for speed module	u1_active	Selects whether to enable the module
	u1_state_speed_ref	The variable for managing the states that determine the speed command value. It manages the states as shown in "Macro definition" below.
	u1_flag_fluxwkn_use	Flag for whether to use magnetic flux-weakening control
	u1_flag_mtpa_use	Flag for whether to use MTPA control
	f4_speed_ctrl_period	Speed loop control interval [s]
	f4_ref_speed_rad_ctrl	Speed command value for control [rad/s]
	f4_ref_speed_rad	Speed command value output by the speed control module during speed control [rad/s]
	f4_ref_speed_rad_manual	Speed command value set by the user during speed control [rad/s]
	f4_speed_rad_ctrl	Speed calculated by the speed control module [rad/s]
	f4_speed_rad	Speed that is input [rad/s]
	f4_max_speed_rad	Maximum speed (mechanical angle) [rad/s]
	f4_speed_rate_limit_rad	Speed variation limit value [rad/s]
	f4_id_ref_output	d-axis current command value [A]
	f4_iq_ref_output	q-axis current command value [A]
	f4_va_max	Maximum voltage on the d- and q- axes [V]
	f4_id_ad	d-axis current value [A]
	f4_iq_ad	q-axis current value [A]
	st_motor	Structure for motor parameters
	st_pi_speed	Structure for PI control

	st_fluxwkn	Structure for magnetic flux-weakening control
	st_mtpa	Structure for MTPA control
st_speed_cfg_t Structure for setting speed module control parameters	st_motor	Structure for motor parameters
	u1_flag_fluxwkn_use	Flag for whether to use magnetic flux-weakening control
	u1_flag_mtpa_use	Flag for whether to use MTPA control
	f4_max_speed_rpm	Maximum speed (mechanical angle) [rpm]
	f4_speed_ctrl_period	Speed control period [s]
	f4_speed_rate_limit_rpm	Speed variation limit value (mechanical angle) [rpm]
	f4_speed_omega_hz	Natural frequency for speed control system [Hz]
	f4_speed_zeta	Attenuation coefficient for the speed control system
st_speed_input_t Structure for speed module input	u1_state_speed_ref	Speed command status
	f4_ref_speed_rad	Speed command value [rad/s].
	f4_speed_rad	Input speed [rad/s]
	f4_va_max	Maximum voltage on the d- and q- axes [V]
st_speed_output_t Structure for speed module output	f4_id_ref	d-axis current command value [A]
	f4_iq_ref	q-axis current command value [A]
	f4_ref_speed_rad_ctrl	Speed used for PI control [rad/s]

Table 8-25 List of structures and variables for the position control module

Structure	Variable	Description
st_motor_position_t Structure for the position control module	u1_is_in_position	Positioning completion flag
	u1_active	Flag indicating whether the module is active
	u1_pos_command_mode	Position command value creation mode
	u1_ctrl_method_mode	IPD/P control mode switching
	u2_pos_dead_band	Dead band (number of position sensor pulses)
	u2_pos_band_limit	Positioning completion width (number of position sensor pulses)
	f4_pos_kp	Position P control gain coefficient
	f4_pos_err_rad	Position deviation [rad]
	f4_pos_rad	Current position [rad]
	f4_ref_pos_rad	Position command value [rad]
	f4_ref_pos_pre_rad	Previous position command value [rad]
	f4_ref_pos_rad_ctrl	Position command after the position profile is processed [rad]
	f4_speed_ff_rad	Speed feedforward value [rad/s]
	f4_speed_ff_ratio	Speed feedforward proportionality coefficient
	f4_ref_speed_rad_output	Speed command value [rad/s]
	f4_max_speed_rad	Maximum speed (mechanical angle) [rad/s]
	f4_ctrl_period	Control interval [s]
	f4_mech_angle_per_sensor_cnt	Angle per position sensor count [rad]

Structure	Variable	Description
	st_ppf	Structure for the position profile
	st_motor	Structure for motor constants
st_position_cfg_t Structure for setting parameters for controlling the position control module	st_motor	Motor parameter structure
	u2_dead_band	Dead band (number of position sensor pulses)
	u2_band_limit	Positioning completion width (number of position sensor pulses)
	u2_pos_interval_time	Time to wait for stability (number of control intervals)
	f4_feedforward_ratio	Speed feedforward proportionality coefficient
	f4_position_omega_hz	Frequency for position control [Hz]
	f4_ctrl_period	Control interval [s]
	f4_mech_angle_per_sensor_cnt	Angle per position sensor pulse [rad]
	f4_max_speed_rad	Maximum speed (mechanical angle) [rad/s]
	f4_accel_time	Acceleration time [s]
	f4_posprof_max_speed_rad	Maximum speed for the position profile [rad/s]
st_position_input_t Structure for position control module input	f4_position_rad	Current position [rad]
st_position_output_t Structure for position control module output	f4_speed_ref	Speed command output value [rad/s]
	f4_position_err	Position deviation value [rad] Use this variable when you want to perform deviation decision externally in cases such as when automatic adjustment is used.
	u1_in_position	Positioning completion flag

Table 8-26 List of structures and variables for the serial encoder sensor module

Structure	Variable	Description
st_sensor_serial_encoder Structure for serial encoder sensor module	s4_encd_cnt	Encoder count for current period
	s4_encd_cnt_pre	Encoder count for previous period
	s4_encd_cnt_max	Maximum encoder count FA-CODER (17bit): 131071
	s4_encd_cnt_full_scale	Full scale value of encoder count FA-CODER (17bit): 131072
	s4_rotate_threshold	Threshold for determining whether or not one rotation has been made FA-CODER (17bit):65536
	s4_polarity	Rotational polarity Encoder count is incremented when flowing positive q-axis current: +1 Encoder count is decremented when flowing positive q-axis current: -1
	s4_rotate_cnt	Number of Rotations
	s4_encd_cnt_multiturn	Multiturn encoder counts
	u1_serial_encoder_type	Type of serial encoder
	f4_speed_ctrl_period	Speed control period [s]

	f4_output_angle_rad	Single turn mechanical angle [rad]
	f4_output_speed_rad	Mechanical angular speed [rad/s]
	f4_output_pos_rad	Multiturn mechanical angle [rad]
st_sensor_serial_encoder_output_t Structure for serial encoder sensor module output	f4_angle_rad	Single turn mechanical angle [rad]
	f4_speed_rad	Mechanical angular speed [rad/s]
	f4_pos_rad	Multiturn mechanical angle [rad]
	f4_speed_lpf_rad	Mechanical angular speed [rad/s]

8.6.8 Macro Definitions

Table 8-27 List of macros

File name	Macro name	Defined value	Remarks
r_motor_serial_encoder_vector_api.h	MOTOR_LOOP_CURRENT_ID	0	Magnetic flux control mode
	MOTOR_LOOP_CURRENT_IQ	1	Torque control mode
	MOTOR_LOOP_SPEED	2	Speed control mode
	MOTOR_LOOP_POSITION	3	Position control mode.
	MOTOR_ERROR_NONE	(0x0000)	Error status: There is no error.
	MOTOR_ERROR_OVER_CURRENT_HW	(0x0001)	Error status: A hardware overcurrent error has occurred.
	MOTOR_ERROR_OVER_VOLTAGE	(0x0002)	Error status: An overvoltage error has occurred.
	MOTOR_ERROR_OVER_SPEED	(0x0004)	Error status: An overspeed error has occurred.
	MOTOR_ERROR_UNDER_VOLTAGE	(0x0080)	Error status: An under-voltage error has occurred.
	MOTOR_ERROR_OVER_CURRENT_SW	(0x0100)	Error status: A software overcurrent error has occurred.
	MOTOR_ERROR_OVER_TEMPERATURE	(0x0200)	Error status: An inverter over temperature error has occurred.
	MOTOR_ERROR_UNKNOWN	(0xffff)	Error status: An error whose error code is unknown has occurred.
r_motor_serial_encoder_vector_manager.h	MOTOR_SERIAL_ENCODER_VECTOR_MODE_OFFSET_ADJUST	(0x00)	The operation mode for initialization
	MOTOR_SERIAL_ENCODER_VECTOR_MODE_DRIVE	(0x01)	The operation mode with the motor driven
	MOTOR_SERIAL_ENCODER_VECTOR_OFAJ_CURRENT_OFFSET	(0x00)	The operation mode for current offset
	MOTOR_SERIAL_ENCODER_VECTOR_OFAJ_POSITION_OFFSET	(0x01)	The operation mode for position offset
	MOTOR_SERIAL_ENCODER_VECTOR_OFAJ_FINISH	(0x02)	Offset adjustment completed
r_motor_serial_encoder_vector_api.h	MOTOR_CTRL_TYPE_POS	0	Macro for switching the control method. Position control mode.
	MOTOR_CTRL_TYPE_SPEED	1	Macro for switching the control method. Speed control mode.
	MOTOR_CTRL_TYPE_TORQUE	2	Macro for switching the control method. Torque control mode.
	MOTOR_CTRL_TYPE_VOLTAGE	3	Not used
	MOTOR_CTRL_TYPE_CURRENT	4	Macro for switching the control method. Open loop current control.
	MOTOR_CTRL_TYPE_ADJ_CRNT_OFFSET	5	Macro for switching the control method. Current offset adjustment mode.

	MOTOR_CTRL_TYPE_A DJ_POS_OFFSET	6	Macro for switching the control method. Position offset adjustment mode.
--	------------------------------------	---	---

8.7 Driver Modules

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

8.7.1 Functions

Table 8-28 lists the functions of the driver module.

Table 8-28 List of functions of the driver module

Function	Description
Acquisition of the A/D conversion value	Acquires DSMIF conversion values such as the phase current and inverter board bus voltage as AD conversion value.
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 PWM output

8.7.2 API

Table 8-29 lists and describes the API functions for the driver module.

Table 8-29 List of API for driver modules

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

8.7.3 Configurations

Table 8-30 lists the configuration for the driver module. Set up the functions to be used and the necessary parameters.

Table 8-30 List of configuration information

File name	Macro name	Set value	Description
r_motor_module_cfg.h	DRIVER_CFG_FUNC_PWM_OUTPUT_START	R_Config_MOTOR_StartTimerCtrl	Sets the function that enables PWM output.
	DRIVER_CFG_FUNC_PWM_OUTPUT_STOP	R_Config_MOTOR_StopTimerCtrl	Sets the function that disables PWM output.
	DRIVER_CFG_FUNC_ADC_DATA_GET	R_Config_MOTOR_AdcGetConvVal	Sets the function that acquires the A/D conversion results.
	DRIVER_CFG_FUNC_DUTY_SET	R_Config_MOTOR_UpdDuty	Sets the function that sets the duty cycle.
r_motor_inverter_cfg.h	INVERTER_CFG_ADC_REF_VOLTAGE	1.8	Reference voltage [V] for voltage AD conversion
	INVERTER_CFG_CURRENT_AD_DC_REF_VOLTAGE	0.640	Reference voltage [V] for current AD conversion
r_motor_module_cfg.h	MOTOR_MCU_CFG_ADC_OFFSET	1915	Offset value of DSMIF conversion value

8.7.4 Structure and Variable Information

Table 8-31 lists the structures that are used for the driver module. In the driver module, the structure for the driver module (g_st_driver) is defined by securing an instance of the module from the API.

Table 8-31 Structure list of variables

Structure	Variable	Description
st_motor_driver_t Structure for the driver module	*ADCDatGet	Pointer to AD conversion result acquisition function
	*BLDCDutySet	Pointer to PWM output enable function
	*PWMOutputStop	Pointer to PWM output disable function
	*PWMOutputStart	Pointer to Duty Cycle Setting Function
	f4_ad_crnt_per_digit	Scale for A/D conversion of the current
	f4_ad_vdc_per_digit	Scale for A/D conversion of the voltage

st_motor_driver_cfg_t Structure for setting the parameters for controlling driver module	f4_pwm_period_cnt	Count value for one interval of the PWM counter (information for the duty setting)
	f4_pwm_dead_time_cnt	Count value for the dead time (information for the duty setting)
	*ADCDDataGet	Pointer to AD conversion result acquisition function
	*BLDCDutySet	Pointer to PWM output enable function
	*PWMOutputStop	Pointer to PWM output disable function
	*PWMOutputStart	Pointer to Duty Cycle Setting Function
	f4_shunt_ohm	Shunt resistance value [ohm] (for calculation of f4_ad_crnt_per_digit)
	f4_volt_gain	Voltage conversion gain coefficient (for calculation of f4_ad_vdc_per_digit)
	f4_crnt_amp_gain	Current conversion gain coefficient (for calculation of f4_ad_crnt_per_digit)
	f4_pwm_period_cnt	Count value for one interval of the PWM counter (information for the duty setting)
	f4_pwm_dead_time_cnt	Count value for the dead time (information for the duty setting)

8.7.5 Adjustment and Configuration of Parameters

In the driver module, parameters that are input from the control parameter configuration (R_MOTOR_DRIVER_ParameterUpdate) are used to associate the motor module and FSP and to convert data. The parameters are input by using st_speed_config_t (the structure for setting the parameters for controlling the drive module). In the sample program, the information that is defined as configurations is used as the parameter settings. Table 8-32 shows the settings.

Table 8-32 Example of settings specified in the sample program.

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

8.8 Serial encoder driver module

8.8.1 Functions

Table 8-33 shows functions of serial encoder driver module.

Table 8-33 Functions of serial encoder driver module

Function	Description
Start / Stop	Start and stop serial encoder
Encoder count acquisition	Acquire encoder count from serial encoder

8.8.2 API

Table 8-34 shows API of serial encoder driver module.

Table 8-34 API of serial encoder driver module

API	Description
R_SERIAL_ENCODER_DRIVER_Open	Initialize serial encoder
R_SERIAL_ENCODER_DRIVER_Start	Start communication with serial encoder via ELC
R_SERIAL_ENCODER_DRIVER_GetSingleTurnPosition	Acquire encoder count
R_SERIAL_ENCODER_DRIVER_GetDelay	Acquire compensation time used for encoder communication time compensation
R_SERIAL_ENCODER_DRIVER_Stop	Stop communication with serial encoder via ELC
R_SERIAL_ENCODER_DRIVER_Close	Stop serial encoder

8.8.3 Macro definition

Table 8-35 shows macros of serial encoder driver module.

Table 8-35 Macros of serial encoder driver module

Macro	Setting value	Description
SERIAL_ENCODER_TYPE	FACODER	Communication interface of serial encoder
SERIAL_ENCODER_MODEL	TSM3101N2001E020	Part number of motor
SERIAL_ENCODER_CFG_RESOLUTION_BIT	17	Resolution bit of serial encoder
SERIAL_ENCODER_CFG_POLARITY	1	Rotational polarity Encoder count is incremented when flowing positive q-axis current: +1. Encoder count is decremented when flowing positive q-axis current: -1
SERIAL_ENCODER_CFG_CH	FAC_ID_0	Channel number
SERIAL_ENCODER_CFG_IDX	FAC0_INDEX	ID number
SERIAL_ENCODER_CFG_DELAY_COMP_TIMER_CNT_DOWN	0.0f	Compensation time (PWM count decrements)
SERIAL_ENCODER_CFG_DELAY_COMP_TIMER_CNT_UP	0.0000475f	Compensation time (PWM count increments)
SERIAL_ENCODER_CFG_DEAD_BAND	3.0f	Dead band width [encoder count]

9. Setting Parameters

9.1 Overview

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

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

Table 9-1 List of parameters setting files.

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

9.2 MPU-related Parameters

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

Table 9-2 List of MPU-related parameters

File name	Macro name	Setting value	Description
r_motor_module_cfg.h	MOTOR_MCU_CFG_PWM_TIMER_FREQ	400.0f	PWM timer frequency [MHz]
	MOTOR_MCU_CFG_CARRIER_FREQ	20.0f	Carrier wave frequency [kHz]
	MOTOR_MCU_CFG_INTR_DECOMPOSITION	0	Skipping count for carrier wave interrupts
	MOTOR_MCU_CFG_AD_FREQ	25.0f	ADC operating frequency [MHz]
	MOTOR_MCU_CFG_AD_SAMPLING_CYCLE	42.0f	ADC sampling cycle [cycle]
	MOTOR_MCU_CFG_AD12BIT_DATA	4095.0f	ADC resolution
	MOTOR_MCU_CFG_ADC_OFFSET	1915	ADC offset value
	MOTOR_MCU_CFG_TFU_OPTIMIZE	MTR_ENABLE	TFU(Trigonometric function unit) MTR_ENABLE: enable TFU MTR_DISABLE: disable TFU

9.3 List of Control Function Setting Parameters

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

Table 9-3 List of operational parameters (general)

File name	Macro name	Setting value	Description
r_motor_module_cfg.h	MOTOR_TYPE_BLDC	MOTOR_TY E_BLDC	Use as default.
	MOTOR_COMMON_CFG_LOOP_MODE	MOTOR_LOOP_SPEED	Default control mode The following control modes can be set. MOTOR_LOOP_CURRENT_IQ MOTOR_LOOP_SPEED MOTOR_LOOP_POSITION
	MOTOR_COMMON_CFG_OVERCURRENT_MARGIN_MULT	2.0f	Limit coefficient for overcurrent
	MOTOR_COMMON_CFG_IAX_CALC_MULT	MTR_SQRT_3	Coefficient for calculating the overcurrent limit value Set to $\sqrt{3}$.
	MOTOR_COMMON_CFG_AUTO_OFFSET_ADJUST_DELAY	(0.5f)	After power on, the delay [s] before starting to perform current offset adjustment.

Table 9-4 List of operational parameters (current control)

File name	Macro name	Setting value	Description
r_motor_module_cfg.h	CURRENT_CFG_VOLT_ERR_COMP	MTR_ENABLE	MTR_ENABLE: enable voltage error compensation MTR_DISABLE: disable voltage error compensation
	CURRENT_CFG_MODULATION_METHOD	MOD_METHOD_SVPWM	MOD_METHOD_SVPWM: Space vector modulation MOD_METHOD_SPWM: Sinusoidal modulation
	CURRENT_CFG_OFFSET_CALC_TIME	512.0f	Current offset measurement time.
	CURRENT_CFG_OFFSET_CALC_WAIT	4000	Current offset detection timing
	CURRENT_CFG_PERIOD_MARG_VALUE	0.5f	Coefficient of voltage phase lead compensation
	CURRENT_CFG_PI_INTEGRAL_LIMIT_VD	INVERTER_CFG_INPUT_V * 0.5f (12.0f)	Output limit of current controller
	CURRENT_CFG_PI_INTEGRAL_LIMIT_VQ	INVERTER_CFG_INPUT_V * 0.5f (12.0f)	Output limit of current controller
	CURRENT_CFG_OMEGA	750.0f	Natural frequency of current control [Hz]
	CURRENT_CFG_ZETA	1.0f	Damping coefficient of current control

	CURRENT_CFG_RATE_LIMIT	0.0004f	Rate of change of command current [A / current control cycle]
--	------------------------	---------	---

Table 9-5 List of operational parameters (speed control)

File name	Macro name	Setting value	Description
r_motor_module_cfg.h	SPEED_CFG_FLUX_WEAKENING	MTR_ENABLE	MTR_ENABLE: enable flux-weakening control MTR_DISABLE: disable flux-weakening control
	SPEED_CFG_MTPA	MTR_ENABLE	MTR_ENABLE: enable MTPA control MTR_DISABLE: disable MTPA control
	SPEED_CFG_CTRL_PERIOD	0.000200f	Speed control period [s]
	SPEED_CFG_OMEGA	30.0f	Natural frequency of speed control [Hz]
	SPEED_CFG_ZETA	1.0f	Damping coefficient of speed control
	SPEED_CFG_SPEED_LIMIT_RPM	7200.0f	Speed limit value [rpm]
	SPEED_CFG_RATE_LIMIT_RPM	3000.0f	Rate of change of command speed [rpm/s]

Table 9-6 List of operational parameters (position control)

File name	Macro name	Setting value	Description
r_motor_module_cfg.h	POSITION_CFG_CTRL_PERIOD	0.000200f	Position control period [s]
	POSITION_CFG_SPEED_FF_RATIO	1.0f	Speed feed forward coefficient
	POSITION_CFG_DEAD_BAND	3.0f	Dead band width [encoder count]
	POSITION_CFG_INTERVAL_TIME	400	Number of times to make a positioning judge
	POSITION_CFG_OMEGA	7.0f	Natural frequency of position control [Hz]
	POSITION_CFG_BAND_LIMIT	100.0f	Positioning width [encoder count]
	POSITION_CFG_ACCEL_TIME	0.5f	Acceleration and deceleration time [s]

Table 9-7 List of operational parameters (position offset)

File name	Macro name	Setting value	Description
r_motor_module_cfg.h	ANGLE_ADJ_RAMP_UP_TIME	1.0f	Ramp up time of d-axis command current
	ANGLE_ADJ_RAMP_DOWN_TIME	0.5f	Ramp down time of d-axis command current
	ANGLE_ADJ_90DEG_CURRENT_RAMP_UP	1.0f	Time of excitation in the direction of 90 deg (electrical angle)
	ANGLE_ADJ_90DEG_CURRENT_RAMP_DOWN	1.5f	Time to finish excitation in the direction of 90 deg (electrical angle)
	ANGLE_ADJ_0DEG_CURRENT_RAMP_UP	2.5f	Time of excitation in the direction of 0 deg (electrical angle)
	ANGLE_ADJ_0DEG_CURRENT_RAMP_DOWN	3.0f	Time to finish excitation in the direction of 0 deg (electrical angle)
	ANGLE_ADJ_CURRENT_RAMP_UP_RATE	0.00004375f	Rate of change of d-axis command current during ramp up [A / current control cycle]
	ANGLE_ADJ_CURRENT_RAMP_DOWN_RATE	0.0004375f	Rate of change of d-axis command current during ramp down [A / current control cycle]
	SENSOR_CFG_OFFSET_ROT_OR_RAD	3.141592f (0.5f * MTR_TWOPI)	Initial value of position offset [rad]

9.4 Protection-related Parameters

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

Table 9-8 Motor and inverter parameter settings

File name	Macro name	Set value	Description
r_motor_inverter_cfg.h	INVERTER_CFG_CURRENT_LIMIT	12.0f	Overcurrent limit value [A]
	INVERTER_CFG_OVERVOLTAGE_LIMIT	28.0f	Overvoltage limit [V]
	INVERTER_CFG_UNDERVOLTAGE_LIMIT	20.0f	Low-voltage limit [V]

INVERTER_CFG_CURRENT_LIMIT

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

INVERTER_CFG_OVERVOLTAGE_LIMIT

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

INVERTER_CFG_UNDERVOLTAGE_LIMIT

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

9.5 Setting the Pulse Modulation Method

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

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

Table 9-9 List of configurations

File name	Macro name	Setting value
r_motor_module_cfg.h	CURRENT_CFG_MODULATION_METHOD	(MOD_METHOD_SVPWM)

Table 9-10 Setting items for the pulse-width modulation drive mode.

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

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

Table 9-11 List of macros

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

9.6 Inverter Parameters

9.6.1 Overview

When you use the sample program, you need to correctly set the inverter information. Table 9-12 shows the inverter parameters set in the sample program.

Table 9-12 Inverter parameter setting.

File name	Macro name	Set value	Description
r_motor_inverter_cfg.h	INVERTER_CFG_SHUNT_RESIST	0.008f	Shunt resistance value [ohm]
	INVERTER_CFG_DEADTIME	1.0f	Dead time [μ s]
	INVERTER_CFG_VOLTAGE_GAIN	22.702127f	Coefficient for voltage detection
	INVERTER_CFG_CURRENT_AMP_GAIN	1.0f	Gain of the amplifier for current detection
	INVERTER_CFG_CURRENT_LIMIT	12.0f	Overcurrent limit value [A]
	INVERTER_CFG_OVERVOLTAGE_LIMIT	28.0f	Overvoltage limit value [V]
	INVERTER_CFG_UNDERVOLTAGE_LIMIT	20.0f	Undervoltage limit value [V]
	INVERTER_CFG_INPUT_V	24.0f	Input voltage [V]
	INVERTER_CFG_VDC_ADC_REF_VOLTAGE	1.8f	AD conversion reference voltage [V]
	INVERTER_CFG_CURRENT_ADC_REF_VOLTAGE	0.640f	Input voltage range of delta-sigma modulator [V]
	INVERTER_CFG_DSM_POWER_ON_DELAY	0.025f	Wait time after PWM signal input until current detection becomes possible [s]
	INVERTER_CFG_VDC_LPF_OMEGA	10.0f	Low-pass filter bandwidth [Hz]
	INVERTER_CFG_COMP_V0	0.096f	Coefficient for compensation of the voltage error [V]
	INVERTER_CFG_COMP_V1	0.192f	
	INVERTER_CFG_COMP_V2	0.288f	
	INVERTER_CFG_COMP_V3	0.384f	
	INVERTER_CFG_COMP_V4	0.480f	
	INVERTER_CFG_COMP_I0	0.028f	Coefficient for compensation of the voltage error [A]
	INVERTER_CFG_COMP_I1	0.056f	
	INVERTER_CFG_COMP_I2	0.088f	
	INVERTER_CFG_COMP_I3	0.120f	
	INVERTER_CFG_COMP_I4	0.200f	

9.6.2 Current Detection Gain

The RZ/T Series Inverter board specifies the input voltage value of the delta-sigma modulator and DSMIF conversion value, as shown in Table 9-13.

Table 9-13 Current signal specifications for RZ/T Series Inverter board

3-phase output current	Input voltage value of the delta-sigma modulator	DSMIF conversion values
+40.0A	0.32 V	4095
0A	0.0 V	2047
-40.0A	-0.32 V	0

In this sample program. This sample program uses the following macros to set the current sensing gain.

INVERTER_CFG_CURRENT_AMP_GAIN

This sample program sets *INVERTER_CFG_CURRENT_AMP_GAIN* as 1.0.

INVERTER_CFG_SHUNT_RESIST

Set the resistance of shunt register. The RZ/T Series Inverter Board has a shunt resistance value of 0.008 [ohm].

INVERTER_CFG_ADC_REF_VOLTAGE

Set the input voltage range for the delta sigma modulator (RV1S9355) installed on the RZ/T Series Inverter board. The *INVERTER_CFG_CURRENT_ADC_REF_VOLTAGE* has an input full-scale voltage range of 0.640 [V] for the delta-sigma modulator (RV1S9355).

f4_ad_crnt_per_digit: member variables of *st_motor_driver_t* structures

f4_ad_crnt_per_digit is the current value per DSMIF conversion count. The calculation formula for *f4_ad_crnt_per_digit* is shown below.

$$\begin{aligned}
 f4_ad_crnt_per_digit \left[\frac{A}{digit} \right] &= \frac{INVERTER_CFG_CURRENT_ADC_REF_VOLATGE [V]}{MOTOR_MCU_CFG_AD12BIT_DATA [digit]} \\
 &\times \frac{INVERTER_CFG_CURRENT_AMP_GAIN(1.0) \times INVERTER_CFG_SHUNT_RESISIT[\Omega]}{1} \\
 &= 0.01953692 [A/digit]
 \end{aligned}$$

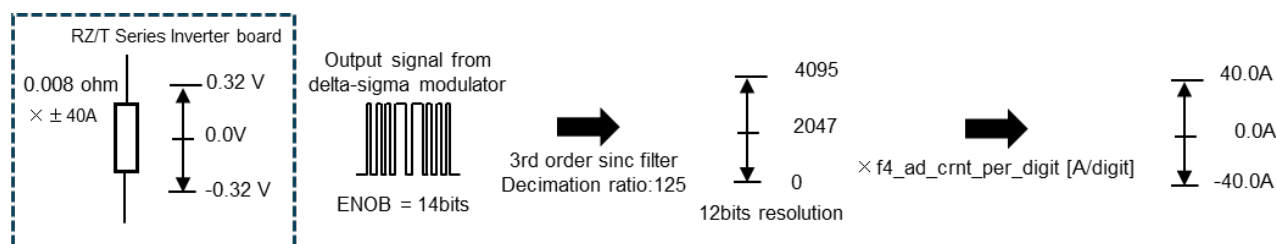


Figure 9-1 Current detection calculation flow.

9.6.3 Voltage Detection Gain

The voltage detection gain is set by INVERTER_CFG_VOLTAGE_GAIN.

Set a coefficient to determine how much inverter bus voltage (V) is equivalent per 1 V of voltage input by the ADC. If the voltage of 1.8 V input to ADC is equivalent to 40.86383 V, $40.86383 / 1.8 = 22.702127$.

Therefore, set 22.702127 to INVERTER_CFG_VOLTAGE_GAIN.

$$\text{INVERTER_CFG_VOLTAGE_GAIN} = \frac{\text{Reference of Inverter bus voltage}}{\text{Reference of ADC input voltage}} = \frac{40.86383}{1.8} = 22.702127$$

Table 9-14 Inverter bus voltage signal specifications for RZ/T Series Inverter board

Inverter bus voltage value	ADC input voltage value	ADC conversion value
0V	0V	0
40.86383V	1.8V	4095

9.6.4 Voltage Error Compensation Parameters

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

① Selecting a dead time value

The dead time value is determined by the characteristics of the power semiconductors used in the inverter. In this sample software, it is set to 1.0μs according to the characteristics of RZ/T Series Inverter board. Reflect the selected dead time value in an input location provided in the FSP settings.

② Setting the flag for whether to enable the voltage error compensation function

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

③ Setting the voltage compensation table

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

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

$$\text{Compensation voltage limit} = (\text{carrier frequency [kHz]} \times \text{dead time [\mu s]} \div 1000) \times \text{bus-bar voltage value}$$

If the relationship between I_u and V_u (I_v - V_v , I_w - V_w) cannot be obtained experimentally for the slope near the zero cross, it is necessary to calculate the table by considering the above equation and the characteristics of the main circuit.

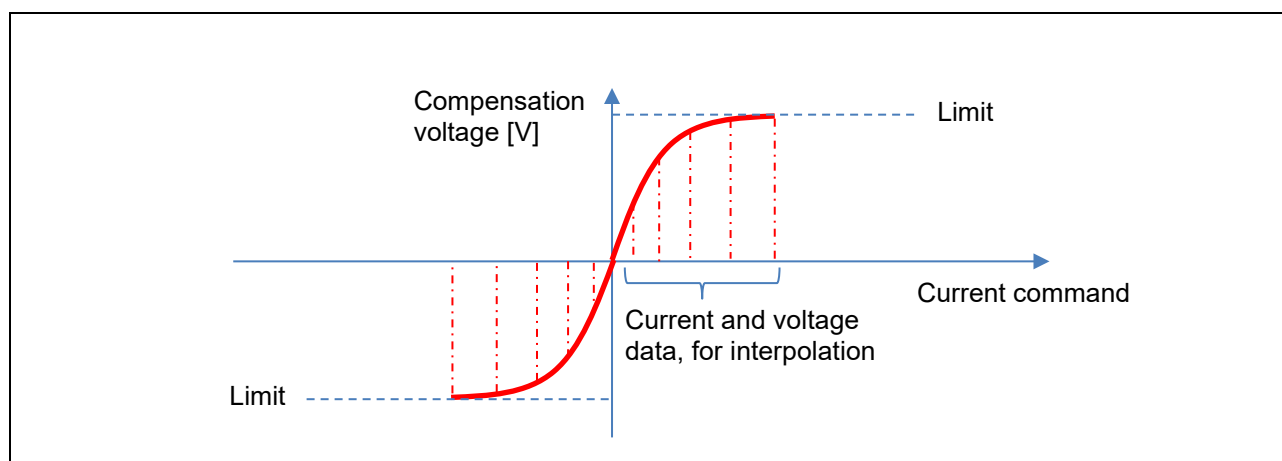


Figure 9-2 Relationship between the compensation voltage value, limit, and current command value.

Table 9-15 Interpolation current and voltage data for different carrier cycles

Carrier cycle		20kHz
	I_u	ΔV_u
0	0.00	0.00
1	0.028	0.096
2	0.056	0.192
3	0.088	0.288
4	0.120	0.384
5	0.200	0.480

9.7 Motor Parameters

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

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

When you use the sample program, you need to correctly set the inverter information and the information about the motor to be used. Table 9-16 shows the values set in the sample program.

Table 9-16 Motor parameter setting.

File name	Macro name	Set value	Description
r_motor_targetmotor_cfg.h	MOTOR_CFG_POLE_PAIRS	5	Number of pole pairs
	MOTOR_CFG_MAGNETIC_FLUX	0.003684f	Magnetic flux [Wb]
	MOTOR_CFG_RESISTANCE	0.626f	Resistance [ohm]
	MOTOR_CFG_D_INDUCTANCE	0.000574f	d-axis inductance [H]
	MOTOR_CFG_Q_INDUCTANCE	0.000813f	q-axis inductance [H]
	MOTOR_CFG_ROTOR_INERTIA	0.0000023f	Rotor inertia [kg m ²]
	MOTOR_CFG_NOMINAL_CURRENT_RMS	3.5f	Rated current [A]
	MOTOR_CFG_MAX_SPEED_RPM	6000.0f	Maximum speed [rpm]

MOTOR_CFG_POLE_PAIRS

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

MOTOR_CFG_RESISTANCE

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

MOTOR_CFG_D_INDUCTANCE, MOTOR_CFG_Q_INDUCTANCE

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

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

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

MOTOR_CFG_ROTOR_INERTIA

Specify the inertia (moment of inertia) of the motor's rotor and shaft. The unit is kg m². Usually, you can find a description in the documentation provided with the motor. If a load is installed, inertia on the load side should also be added to the setting.

MOTOR_CFG_NOMINAL_CURRENT_RMS

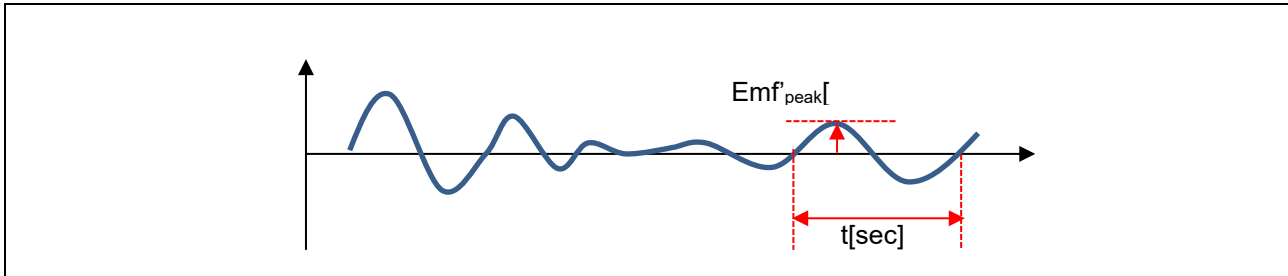
Specify the motor's rated current (RMS). The unit is ampere. It is indicated on the nameplate of the motor or in the accompanying documentation.

MOTOR_CFG_MAGNETIC_FLUX

Select two of the motor's three-phase output wires U, V, and W and connect them to the oscilloscope. For example, connect the oscilloscope probes to the U and V phases so that the voltages can be measured. The U-V phase line voltage value can be obtained by connecting a motor that can rotate at the rated speed to the

end of the motor shaft and rotating it at the rated speed. Dividing the line voltage value by $\sqrt{3}$ 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 $2\pi f$ 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 $\sqrt{2}$ 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 PM 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 PM motor rotates at its rated speed can be determined simply. To determine the inductive voltage, it must be measured by rotating the motor shaft at the rated speed using a load test device.

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

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

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

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

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

9.8 Current Control Parameters

Table 9-17 shows the current control parameters. Current control parameters are calculated based on the motor parameters, PWM carrier frequency, and desired current response performance.

The macros for the current control parameters shown in Table 9-17 are set and reflected in the internal variables at start-up, but if adjustments are needed after start-up, the following two parameters can be adjusted from the RMW. See Table 8-7. Note that not all parameters can be changed.

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

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

Table 9-17 List of current control parameters

File name	Macro name	Set value	Description
r_motor_targetmotor_cfg.h	CURRENT_CFG_OFFSET_CALC_TIME	512.0f	Current offset measurement time setting
	CURRENT_CFG_OMEGA	750.0 f	Natural frequency for the current control system [Hz]
	CURRENT_CFG_ZETA	1.0f	Attenuation coefficient for the current control system
	CURRENT_CFG_RATE_LIMIT	0.0004f	Limit for rate of change in current command [A/period]

CURRENT_CFG_OFFSET_CALC_TIME

Specify the number of times to measure the offset value when measuring the offset of current detection at start-up. Normally, use the default value.

CURRENT_CFG_OMEGA, CURRENT_CFG_ZETA

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

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

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

CURRENT_CFG_RATE_LIMIT

During torque control, the rate of change in current command value is limited so that it does not increase rapidly. Specify the current value [A] that can change per current control cycle. By default, the current control cycle is 50 μ s, so a value of 0.0004 [A] allows up to 16 A change per second.

9.9 Speed Control Parameters

Table 9-18 shows the speed control parameters. The values set in this file are applied as initial values at system startup. The macros for the speed control parameters shown in Table 9-18 are set and reflected in the internal variables at start-up, but if adjustments are needed after start-up, the following three parameters are available that can be adjusted from the RMW. See Table 8-7. Note that not all speed control parameters can be changed on RMW.

Natural frequency for speed control system: `com_f4_speed_omega_hz`

Attenuation coefficient for the speed control system: `com_f4_speed_zeta`

Table 9-18 List of speed control parameters

File name	Macro name	Set value	Description
r_motor_target motor_cfg.h	SPEED_CFG_CTRL_PERIOD	0.000200f	Control interval setting [s]
	SPEED_CFG_OMEGA	30.0f	Natural frequency for the speed control system [Hz]
	SPEED_CFG_ZETA	1.0f	Attenuation coefficient for the speed control system
	SPEED_CFG_SPEED_LIMIT_RPM	7200.0f	Speed limit value (mechanical angle) [rpm]
	SPEED_CFG_RATE_LIMIT_RPM	3000.0f	Acceleration limit [rpm/s]

SPEED_CFG_CTRL_PERIOD

The speed control period of this sample program is 200 μ s. Set 0.000200 for `SPEED_CFG_CTRL_PERIOD`.

SPEED_CFG_OMEGA, SPEED_CFG_ZETA

In the speed control module, the control gain is adjusted by tuning the natural frequency for the speed control system and the attenuation coefficient for the speed control system. Increasing the natural frequency for the speed control system improves responsiveness, expanding the following capability of the speed to the commanded speed. The maximum settable natural frequency for speed control is limited to 1/3 of the maximum settable natural frequency for current control to prevent interference with current control. If the natural frequency of the current control system is 500 Hz, then $500 \text{ Hz}/3 = 166 \text{ Hz}$.

For the attenuation coefficient for the speed control system, a value in the range from 0.7 to 1.0 is typically set. Setting a value nearer to 1.0 makes the response more stable and moderate. Make adjustment while checking the speed responsiveness.

SPEED_CFG_RATE_LIMIT_RPM

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

9.10 Position control parameters

Table 9-19 shows parameters for position control.

Table 9-19 Position control parameters

File name	Macro name	Setting value	Description
r_motor_module_cfg.h	POSITION_CFG_CTRL_PERIOD	0.000200f	Position control period [s]
	POSITION_CFG_SPEED_FF_RATIO	1.0f	Speed feed forward coefficient
	POSITION_CFG_DEAD_BAND	3.0f	Dead band width [encoder count]
	POSITION_CFG_OMEGA	7.0f	Natural frequency of position control [Hz]
	POSITION_CFG_ACCEL_TIME	0.5f	Acceleration and deceleration time [s]

POSITION_CFG_CTRL_PERIOD

The position control period of this sample program is 200 μ s. Set 0.000200 to POSITION_CFG_CTRL_PERIOD.

POSITION_CFG_SPEED_FF_RATIO

Set 1.0 to feed forward 100% of the speed directive values.

POSITION_CFG_DEAD_BAND

This sample program sets the POSITION_CFG_DEAD_BAND value to achieve a deadband width of 0.008 degrees when using the FA-CODER.

POSITION_CFG_OMEGA

In this sample program, the natural frequency of the position control system is set to be 1/5 of the natural frequency of the speed control system.

POSITION_CFG_ACCEL_TIME

The default value of the acceleration and deceleration time of the trapezoidal speed control is 0.5 s. Consider the inertia and current value when setting the settings.

The com variable to be used when making parameter adjustments from RMW is shown in Table 9-20.

Table 9-20 com variables related position control (X = 0, 1, 2, 3)

Variable	Description
comX.u1_pos_cmd_mode	Position control command mode 0: Position command value is 0 [rad] 1: Step 2: Trapezoidal speed
comX.u2_pos_interval_time	Positioning wait time
comX.u2_pos_dead_band	dead band width [encoder count]
comX.u2_pos_band_limit	in-position width [encoder count]
comX.f4_pos_omega_hz	Natural frequency for position control [Hz]

comX.f4_pos_ff_ratio	Speed feed forward ratio
comX.f4_accel_time	Acceleration and deceleration time [s]
comX.f4_posprof_max_speed_rpm	Maximum speed during trapezoidal speed control [rpm]

9.11 Voltage Phase Lead Compensation Parameters

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

Table 9-21 List of configurations

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

10. FSP Settings

10.1 Clock settings

Table 10-1 Clock settings

Clock	Value
CR52C0CLK	800 MHz
PCLKH	200 MHz
PCLKM	100 MHz
PCLKL	50 MHz
PCLKGPTL	400 MHz
PCLKSCIEn	96 MHz

10.2 Interrupt settings

Table 10-2 Interrupt settings

Event	ISR
GPT0_0_INT0 (GPT0_0 selected interrupt 0)	gpt_int_select_isr
POEG0_GROUP0 (POEG group A interrupt for channels in LLPP)	poeg_event_isr
POEG0_GROUP1 (POEG group B interrupt for channels in LLPP)	poeg_event_isr
POEG0_GROUP2 (POEG group C interrupt for channels in LLPP)	poeg_event_isr
POEG0_GROUP3 (POEG group D interrupt for channels in LLPP)	poeg_event_isr
SCI4_ERI (SCI4 Receive error)	sci_uart_eri_isr
SCI4_RXI (SCI4 Receive data full)	sci_uart_rxi_isr
SCI4_TXI (SCI4 Transmit data empty)	sci_uart_txi_isr
SCI4_TEI (SCI4 Transmit end)	sci_uart_tei_isr
ENCIF00_INT0 (ENCIF00 combined interrupt 0)	fac0_eri_isr
ENCIF00_INT1 (ENCIF00 combined interrupt 1)	fac0_rxi_isr
ENCIF00_INT2 (ENCIF00 combined interrupt 2)	fac0_txi_isr
ENCIF00_INT3 (ENCIF00 combined interrupt 3)	fac0_tei_isr
ENCIF01_INT0 (ENCIF01 combined interrupt 0)	fac1_eri_isr
ENCIF01_INT1 (ENCIF01 combined interrupt 1)	fac1_rxi_isr
ENCIF01_INT2 (ENCIF01 combined interrupt 2)	fac1_txi_isr
ENCIF01_INT3 (ENCIF01 combined interrupt 3)	fac1_tei_isr
ENCIF02_INT0 (ENCIF02 combined interrupt 0)	fac2_eri_isr
ENCIF02_INT1 (ENCIF02 combined interrupt 1)	fac2_rxi_isr
ENCIF02_INT2 (ENCIF02 combined interrupt 2)	fac2_txi_isr
ENCIF02_INT3 (ENCIF02 combined interrupt 3)	fac2_tei_isr

ENCIF03_INT0 (ENCIF03 combined interrupt 0)	fac3_eri_isr
ENCIF03_INT1 (ENCIF03 combined interrupt 1)	fac3_rxi_isr
ENCIF03_INT2 (ENCIF03 combined interrupt 2)	fac3_txi_isr
ENCIF03_INT3 (ENCIF03 combined interrupt 3)	fac3_tei_isr

10.3 ELC settings

Table 10-3 ELC settings

Peripheral	Event
DSMIF0 CAP_TRG0	GPT0_0_ADTRGB (GPT0_0 GTADTRB compare match)
DSMIF1 CAP_TRG0	GPT2_0_ADTRGB (GPT2_0 GTADTRB compare match)
DSMIF2 CAP_TRG0	GPT1_0_ADTRGB (GPT1_0 GTADTRB compare match)
DSMIF3 CAP_TRG0	GPT3_0_ADTRGB (GPT3_0 GTADTRB compare match)
GPT1 event A	GPT0_0_ADTRGA (GPT0_0 GTADTRA compare match)
GPT2 event A	GPT0_0_ADTRGA (GPT0_0 GTADTRA compare match)
GPT3 event A	GPT0_0_ADTRGA (GPT0_0 GTADTRA compare match)
ADC2 A	GPT0_0_INT0 (GPT0_0 selected interrupt 0)

10.4 DSMIF settings

Table 10-4 DSMIF settings(Unit setting)

Property	Value
General – Name	g_dsmifX (X = 0, 1, 2, 3)
General – Unit	0, 1, 2, 3
Extra - Channel Synchronization - Counter Reset - Division counter initialization trigger	No Trigger
Extra - Channel Synchronization - Counter Reset - Edge	Negative Edge
Extra - Channel Synchronization - Capture Trigger A	Trigger 0
Extra - Channel Synchronization - Capture Trigger B	No Trigger

Table 10-5 DSMIF settings(Channel setting)

Property	Value
A/D Convert - Clock - Clock Mode	Slave
A/D Convert - Sinc Filter - Filter Order	Filter Order 3rd
A/D Convert - Sinc Filter - Decimation Ratio	250

Note: The output clock frequency of the delta-sigma modulator RV1S9355 on the RZ/T Series Inverter Board is 20MHz. In this sample program, the DSMIF operates based on the MCLK signal input from the delta-sigma modulator.

Note: For details on the filter order, decimation ratio, and delay time of the sinc filter, see 7.4.2.

10.5 POEG settings

Table 10-6 POEG settings

Property	Value
General - Trigger	GTETRG Pin GPT Output level
General - Name	g_poegX (X = 0, 1, 2, 3)
General - Channel	0
General - Group	A, B, C, D
Input - GTETRG Polarity	Active High
Input - GTETRG Noise Filter (Base clock CH0=PCLKH, Other=PCLKL)	DIV128
Interrupts - Callback	callback_poegX_overcurrent (X = 0, 1, 2, 3)
Interrupts - Interrupt Priority	Priority 0(highest)

10.6 GPT settings

Table 10-7 GPT settings (Unit Settings)

Property	Value
General - Name	g_three_phaseX (X = 0, 1, 2, 3)
General - Mode	Triangle-Wave Symmetric PWM
General - Period	50
General - Period Unit	Microseconds
General - GPT U-Unit	0, 2, 1, 3
General - GPT U-Channel	0
General - GPT V-Unit	0, 2, 1, 3
General - GPT V-Channel	1
General - GPT W-Unit	0, 2, 1, 3
General - GPT W-Channel	2
General - Callback Channel	U-Channel
General - Buffer Mode	Single Buffer
General - GTIOCA Stop Level	Pin Level Low
General - GTIOCB Stop Level	Pin Level High
Extra Features - Dead Time - Dead Time Count Up (Raw Counts)	400
Extra Features - Dead Time - Dead Time Count Down (Raw Counts)	400

Table 10-8 GPT settings (Channel setting)

Property	Value
Output - Duty Cycle Percent	50
Input – Start Source	None (Only GPT0_0, GPT0_1, GPT0_2) ELC GPTA (Except for GPT0_0, GPT0_1, GPT0_2)
Interrupts(GPT_SEL) - Callback	callback_gpt_udf_cyclic (Only GPT0_0) NULL (Except for GPT0_0)
Interrupts(GPT_SEL) – Underflow/Trough Interrupt Source	INT0 (Only GPT0_0, GPT1_0, GPT2_0, GPT3_0) None (Except GPT0_0, GPT1_0, GPT2_0, GPT3_0)
Interrupts(GPT_SEL) – INT0 Interrupt Priority	Priority 4 (Only GPT0_0) Disabled (Except GPT0_0)
Extra Features - Output Disable - POEG Link	POEG Channel 0, 1, 2, 3
Extra Features - Output Disable - GTIOCA Disable Setting	Set Hi Z

Extra Features - Output Disable - GTIOCB Disable Setting	Set Hi Z
Extra Features - ADC Trigger - Start Event Trigger	Trigger Event A/D Converter Start Request A During Down Counting Trigger Event A/D Converter Start Request B During Down Counting (Only GPT0_0, GPT1_0, GPT2_0, GPT3_0) None (Except for GPT0_0, GPT2_0, GPT2_0, GPT3_0)
Extra Features - ADC Trigger - ADC A Compare Match(Raw Counts)	1 (Only GPT0_0, GPT1_0, GPT2_0, GPT3_0) 0 (Except for GPT0_0, GPT1_0, GPT2_0, GPT3_0)
Extra Features - ADC Trigger - ADC B Compare Match(Raw Counts)	2500 (Only GPT0_0, GPT1_0, GPT2_0, GPT3_0) 0 (Except for GPT0_0, GPT1_0, GPT2_0, GPT3_0)

10.7 ADC settings

Table 10-9 ADC settings

Property	Setting value
General - Name	g_adc2
General - Unit	2
General - Clear after read	Off
General - Mode	Single Scan
General - Double-trigger	Disabled
Input - Channel Scan Mask	Channel 4, 5, 6, 7
Input - Start Trigger - Group A	ELC Trigger Group A
Interrupts - Normal/Group A Trigger	GPT0_0_INT0 (GPT0_0 selected interrupt 0)
Interrupts - Callback	NULL
ELC – ADC2 event B – Trigger Source	GPT0_0_INT0 (GPT0_0 selected interrupt 0)

10.8 WDT settings

Table 10-10 WDT settings

Property	Value
Timeout	1,024 Cycles
Clock Division Ratio	PCLK/4
Window Start Position	100
Window End Position	0
WDT Callback	NULL

10.9 Pin settings

Table 10-11 Inverter Board Interface Connector INV1 (J49) Pin settings

Pin Name	Pin Setting
INV1_AN204	AN204
MCLK00	P01_2
MDAT00	P06_1
MCLK01	P06_2
MDAT01	P01_5
MCLK02	P01_6
MDAT02	P01_7
GTIOC00_0A	P00_0
GTIOC00_0B	P00_1
GTIOC00_1A	P00_3
GTIOC00_1B	P00_2
GTIOC00_2A	P00_4
GTIOC00_2B	P00_5
GTETRG	P11_5
ENCIFOE00	P00_7
ENCIFDO00	P01_0
ENCIFDI00	P01_1

Table 10-12 Inverter Board Interface Connector INV2 (J50) Pin settings

Pin Name	Pin Setting
INV2_AN205	AN205
MCLK10	P03_0
MDAT10	P03_1
MCLK11	P24_1
MDAT11	P24_2

MCLK12	P24_4
MDAT12	P24_3
GTIOC02_0A	P14_1
GTIOC02_0B	P14_2
GTIOC02_1A	P14_4
GTIOC02_1B	P14_5
GTIOC02_2A	P14_6
GTIOC02_2B	P14_7
GTETRGB	P11_7
ENCIFOE01	P02_5
ENCIFDO01	P02_6
ENCIFDI01	P02_7

Table 10-13 Inverter Board Interface Connector INV3 (J51) Pin settings

Pin Name	Pin Setting
INV3_AN206	AN206
MCLK20	P18_5
MDAT20	P18_6
MCLK21	P23_1
MDAT21	P23_2
MCLK22	P25_0
MDAT22	P24_5
GTIOC01_0A	P03_6
GTIOC01_0B	P03_7
GTIOC01_1A	P04_1
GTIOC01_1B	P04_0
GTIOC01_2A	P04_2
GTIOC01_2B	P04_3
GTETRGC	P07_7
ENCIFOE02	P15_1
ENCIFDO02	P15_2
ENCIFDI02	P15_3

Table 10-14 Inverter Board Interface Connector INV4 (J52) Pin settings

Pin Name	Pin Setting
INV4_AN207	AN207
MCLK30	P23_5

MDAT30	P23_6
MCLK31	P09_3
MDAT31	P09_4
MCLK32	P23_7
MDAT32	P24_0
GTIOC03_0A	P16_0
GTIOC03_0B	P16_1
GTIOC03_1A	P16_3
GTIOC03_1B	P16_2
GTIOC03_2A	P16_4
GTIOC03_2B	P16_5
GTETRGD	P02_0
ENCIFOE03	P16_7
ENCIFDO03	P17_0
ENCIFDI03	P17_1

Table 10-15 RMW Interface Connector J40 Pin settings

Pin Name	Pin Setting
RMW_RXD4	P06_6
RMW_TXD4	P06_7

Table 10-16 LED Pin settings

Pin Name	Pin Setting
LED8(Green , Lights up when the motor is controlled.)	P13_3
LED11(Red , Lights up when an error occurs)	P10_3

11. Evaluation results

11.1 Acceleration and deceleration characteristics

Figure 11-1 shows the waveform when trapezoidal speed control is performed with an acceleration and deceleration time of 10ms.



Figure 11-1 Acceleration and deceleration characteristics.

11.2 Load characteristics

Figure 11-2 shows the response of speed, q-axis current and d-axis current when the rated torque of 0.095 Nm is applied when rotating at 3000rpm.

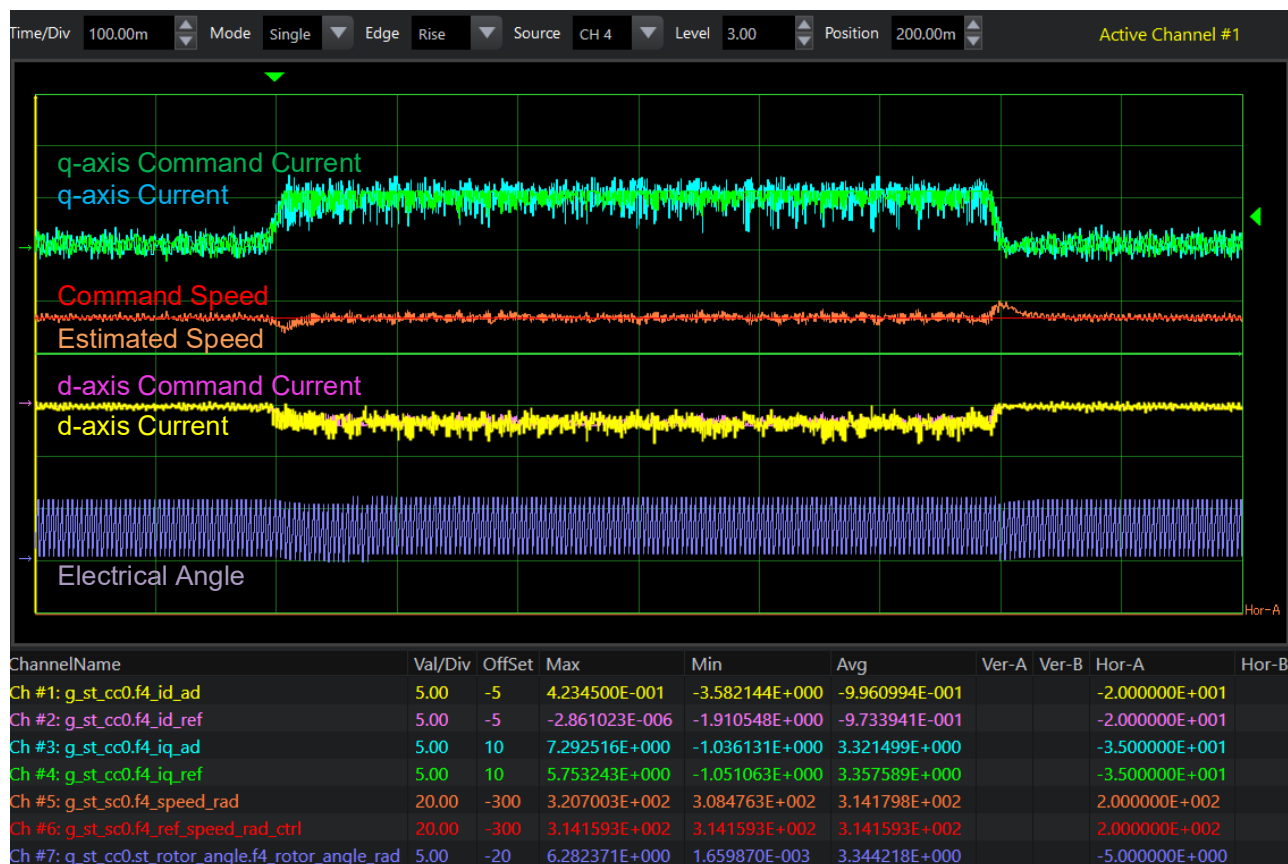


Figure 11-2 Load characteristics (3000rpm)

11.3 CPU Utilization

Table 11-1 shows the CPU processing time and loading rate for each control cycle.

Table 11-1 Loop and CPU loading rate.

Control loop type	Control period	Processing time	CPU loading rate
Current control loop(for 4-axis)	50 μ s	16.7 μ s	37.7%
Speed control loop(for 1-axis)	200 μ s	2.13 μ s	

11.4 Program size and RAM usage

Table 11-2 shows the program size and RAM usage when compiling this sample program at the optimization level is Optimize for debug(-Og).

Table 11-2 Program size and RAM usage

Program size	131,144 [Bytes]
RAM usage	91,880 [Bytes]

12. FAQ

Table 12-1 FAQ

Phenomenon	Solution
The motor does not run. An overcurrent error occurs.	- Check whether there is a break or poor contact of U, V, and W. - Be careful not to break the encoder sensor wires. If the encoder sensor wire is broken or has poor contact, the motor cannot be controlled normally. - Check if the shaft is locked or overloaded.
The bus voltage drops during motor operation.	The output performance of the power supply may be insufficient for the load of the motor, or there may be a short circuit on the motor side.
The motor cannot rotate after stopping due to an error.	The error must be cleared. See 6.8.7.
The motor stops with an error even after starting.	- See 6.8.6 for the cause of the error. - Verify that there is no overvoltage or undervoltage in the inverter bus voltage and that the output is not short-circuited. - Verify that sensor-related signal settings are correct. - Verify that the POE settings, motor settings (GPT terminal and active level settings), etc. in the FSP settings are properly set according to the inverter circuit specifications.
The value of the variable displayed on the RMW is abnormal.	After changing the sample program, you need to register and update the MAP file. For instructions on updating MAP file registration, see 6.7.
Values set from RMW are not reflected.	Manipulate variables in comX.u1_enable_write to rewrite the parameters. When the timing for writing values to comX.u1_enable_write is prior to writing parameters, the internal reflection process operates first. Address as follows: - Put the comX.u1_enable_write on the last line - Write the comX.u1_enable_write twice (with 1 then 0) or toggle write

Revision History

Rev.	Date of issue	Amendments	
		Page	Point
1.00	Nov.4.26	-	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/.