

RA2T1 Sensorless vector control for permanent magnetic synchronous motor

For Renesas Flexible Motor Control Series

Summary

This application note explains the sample programs for driving a permanent magnet synchronous motor in the sensorless vector method using the RA2T1 microcontroller. This note also explains how to use the motor control development support tool Renesas Motor Workbench (RMW).

These sample programs are intended to be used as references only, and Renesas Electronics Corporation does not guarantee their operation. Please use them after carrying out a thorough evaluation in a suitable environment. Working in a high voltage environment is dangerous, so please read the user's manual for each development environment carefully before using the product in consideration of safety. Renesas cannot be held responsible for any accidents or damages that may occur in the development environment listed in this application note.

Operation checking device

Operations of the sample programs have been checked using the following device.

- RA2T1(R7FA2T1074CFL)

Applicable sample programs

This application note regards the following sample programs.

<1 shunt current detection method>

- RA2T1_MCILV1_PM_LESS_FOC_1S_E2S_V101 (IDE: e²studio)

<3 shunt current detection method>

- RA2T1_MCILV1_PM_LESS_FOC_3S_E2S_V101 (IDE: e²studio)

References

- RA2T1 Group User's Manual : Hardware (R01UH1089)
- Renesas Motor Workbench User's Manual (R21UZ0004)
- RA2T1 Group MCK-RA2T1 User's Manual (R12UZ0152)
- RA2T1 Group MCB-RA2T1 User's Manual (R12UZ0145)
- MCI-LV-1 User's Manual (R12UZ0092)
- MC-COM User's Manual (R12UZ0093)

Contents

1. Overview3

2. System overview4

3. Explanation of Control Programs14

4. Usage of Motor Control Development Support Tool, Renesas Motor Workbench79

1. Overview

This application note explains how to implement the sensorless vector control sample programs of the permanent magnetic synchronous motor (PMSM) using the RA2T1 microcontroller, and how to use the motor control development support tool Renesas Motor Workbench.

1.1 Development environment

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

Table 1-1 Development Environment of the Sample Programs (Hardware)

Microcontroller	Evaluation board	Motor
RA2T1 (R7FA2T1074CFL)	RA2T1 CPU board (RTK0EMA810C00000BJ) ^(Note 1) Inverter board (RTK0EM0000B12020BJ) ^(Note 1) Communication board (RTK0EMXC90Z00000BJ) ^(Note 1)	MOONS' R42BLD30L3

Table 1-2 Development Environment of the Sample Programs (Software)

e²studio version	Build tool version	FSP	Debug tool
2026-04.2	gcc 13.2.1.arm-13-7	v6.5.0	Renesas Motor Workbench 3.3.1

For inquiries regarding purchases and technical support, please contact Renesas Electronics Corporation sales representatives and dealers.

Notes:

1. The RA2T1 CPU board (RTK0EMA810C00000BJ), Inverter board (RTK0EM0000B12020BJ) and Communication board (RTK0EMXC90Z00000BJ) are products of Renesas Electronics Corporation.

2. System overview

An overview of this system is provided below.

2.1 Hardware Specifications

MCK-RA2T1 consists of the Inverter board, the CPU board, and the Communication board. For the specifications of the kit and each board, please refer to the document below.

- MCK-RA2T1
RA2T1 Group MCK-RA2T1 User's Manual (R12UZ0152)
- CPU board
RA2T1 Group MCB-RA2T1 User's Manual (R12UZ0145)
- Inverter board
MCI-LV-1 User's Manual (R12UZ0092)
- Communication board
MC-COM User's Manual (R12UZ0093)

RA2T1 Sensorless vector control for permanent magnetic synchronous motor For Renesas Flexible Motor Control Series

2.1.1 Hardware configuration

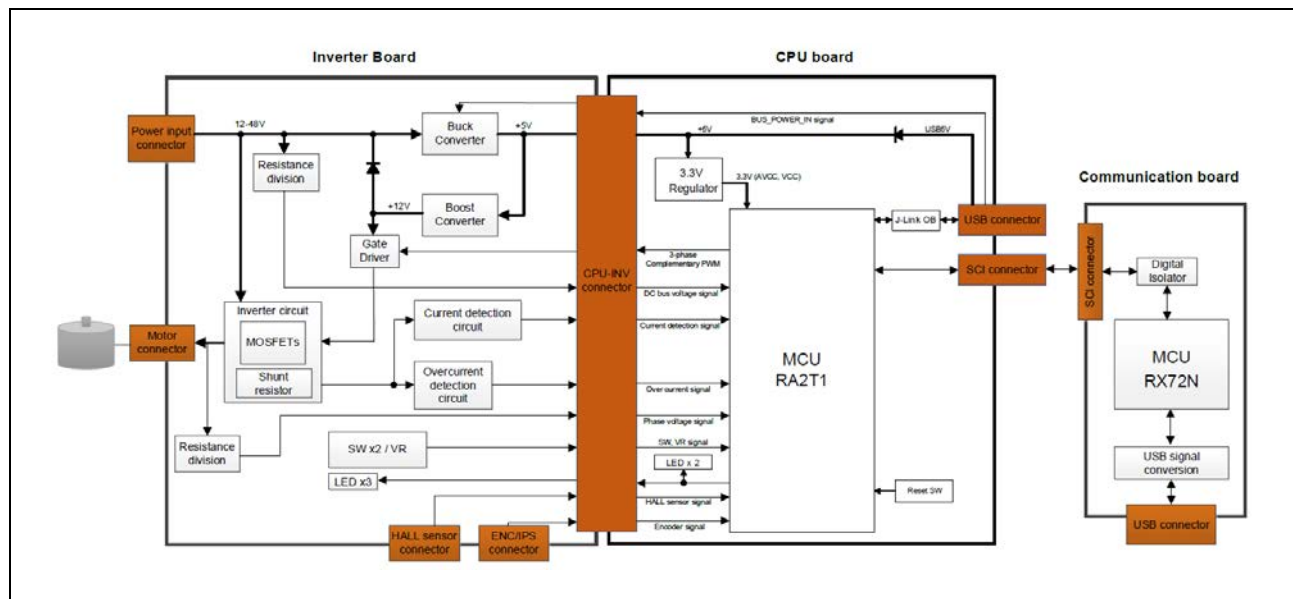


Figure 2-1 Hardware Configuration Diagram

2.1.2 Hardware setting

The hardware needs to be set up to use the sample software. The settings are shown below. The settings are the same as the initial settings for MCK-RA2T1.

(1) Inverter board

The settings and functions of the jumper pins (JP1~JP15) are as follows.

Table 2-1 Jumper pin setting of Inverter board

JP No.	Setting	Function
JP1	2-3pin short	1-2pin short: Disable 5V regulator 2-3pin short: Enable 5V regulator
JP2, JP3, JP4, JP6, JP12, JP13	2-3pin short	1-2pin short: Disable current detection amplifiers 2-3pin short: Enable current detection amplifiers
JP5, JP7, JP9, JP10, JP14, JP15	1-2pin short	1-2pin short: Current detection amplifier gain = 20 1-2pin open: Current detection amplifier gain = 10
JP8, JP11	2-3pin short	1-2pin short: 2-/3-shunt current detection 2-3pin short: 1-shunt current detection

(2) CPU board

The settings and functions of the jumper pins (JP3) are as follows.

Table 2-2 Jumper pin setting of CPU board

JP No.	Setting	Function
JP3	1-2pin open	1-2pin short: On board debugger invalid 1-2pin open: On board debugger valid

(3) Communication board

The settings and functions of the jumper pins (JP1~JP3) are as follows.

Table 2-3 Jumper pin setting of Communication board

JP No.	Setting	Function
JP1	1-2pin open	1-2pin short: Enable pull-up for MD port (Not available) 1-2pin open: Enable pull-up for MD port
JP2	1-2pin short	1-2pin short: Disable pull-up for GPIO(PC6) 1-2pin open: Enable pull-up for GPIO(PC6)
JP3	1-2pin short	1-2pin short: Disable pull-up for GPIO(PC5) 1-2pin open: Enable pull-up for GPIO(PC5)

2.1.3 User interface

The user interfaces of this system are listed below.

(1) Inverter board

Table 2-4 User interface of Inverter board

Item	Interface component	Function
VR1	Volume	Not used in this system
SW1	Toggle switch	Not used in this system
SW2	Push switch	Not used in this system
LED1	Orange LED	On: The motor is rotating. Off: The motor is stopped.
LED2	Orange LED	On: An error was detected. Off: The system is operating normally.
LED3	Orange LED	Not used in this system

Table 2-5 Connector interface of Inverter board

Item	Number of ports	Function
CN1	2	Power input connector
CN2	3	Motor connector
CN3	34	CPU board connector
CN4	34	CPU board connector
CN5	10	ENC/IPS connector
CN6	5	HALL sensor connector

(2) CPU board

Table 2-6 User interface of CPU board

Item	Interface component	Function
RESET	Tact switch (SW1)	MCU reset switch
LED1	Orange LED	On: The motor is rotating. Off: The motor is stopped.
LED2	Orange LED	On: An error was detected. Off: The system is operating normally.

Table 2-7 Connector interface of CPU board

Item	Number of ports	Function
CN1	6	Not mounted
CN2	34	Inverter board connector
CN3	34	Inverter board connector
CN4	4	SCI connector for Communication board
CN5	(USB Type-C)	USB connector for J-Link OB

(3) Communication board

Table 2-8 User interface of Communication board

Item	Interface component	Function
RESET	Tact switch (SW1)	MCU reset switch

Table 2-9 Connector interface of Communication board

Item	Number of ports	Function
CN1	4	Not mounted
CN2	(USB miniB)	USB connector (not available for users)
CN3	(USB Type-C)	USB Type-C connector for PC
CN4	4	Not mounted
CN5	4	SCI connector for CPU board

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 2-10 shows the pin functions of the RA2T1 microcontroller in this system.

Table 2-10 RA2T1 pin function list

Pin No.	RA2T1 pin function	Signal function
1	P400/GTIOC3A	PFCPWM1/ENCA
2	P401/GTIOC3B	PFCPWM2/ENCB
3	VCL1	System
4	P402	SW1
5	P403	INV CONNECTED
6	VSS1	GND
7	P213/XTAL	External clock
8	P212/EXTAL	External clock
9	VCC1	VCC
10	P409/IRQ6	HALL_V
11	P408/IRQ7	HALL_W
12	P407/GTIOC0B	UN
13	P915	LED3
14	P914	LED2
15	P913	LED1
16	P208/GTIOC0A	UP
17	P207	Relay
18	P206/IRQ0	HALL_U
19	RES#	Reset
20	P201/MD	ARM debugger
21	P200	SAFE LOCK
22	P302/GTETRGA	OC INV
23	P301/GTETRGB	OCPFC/ENCZ
24	P300/SWCLK	ARM debugger
25	P108/SWDIO	ARM debugger
26	P109/GTIOC2A	VP
27	P110/GTIOC2B	VN
28	P111/GTIOC1A	WP
29	P112/GTIOC1B	WN
30	VCC2	VCC
31	VSS2	GND
32	P104/IRQ1	SW2/Overtemp
33	P103/CTS0/AN019	VCOM/HVtemp
34	P102/AN020	VU
35	P101/TXD0	SCI
36	P100/RXD0	SCI
37	P500 VCOM	VCOM
38	P015/AN002	IW
39	P014/AN001	IV
40	P013/AN000	IU
41	P012/AN007	VV
42	AVCC0	AVCC
43	AVSS0	AGND
44	P011/AN006	VW
45	P010/AN005	IPFC2/VR
46	P002/AN010	IPFC1
47	P001/AN009	VPFC
48	P000/AN008	VPN

2.1.4 Peripheral functions

Table 2-11 List of Peripheral Functions Table 2-11 is a list of peripheral functions used in this system.

Table 2-11 List of Peripheral Functions

Peripheral Function	Usage
12-bit A/D converter (ADC12)	<ul style="list-style-type: none">• Inverter bus voltage detection• 3-phase current detection [Used in 3-shunt mode]• DC link current detection [Used in 1-shunt mode]
Low Power Asynchronous General Purpose Timer (AGTW channel 0)	<ul style="list-style-type: none">• 1 [ms] interval timer
General PWM Timer (GPT channel 0, 1, 2)	<ul style="list-style-type: none">• PWM output using complementary PWM mode• 50[us] timer
Port Output Enable for GPT (POEG)	<ul style="list-style-type: none">• PWM output forcibly shut off by trigger input
Watchdog Timer (WDT)	<ul style="list-style-type: none">• Program runaway detection

(1) 12-bit A/D converter (ADC12)

V-phase current (I_v), W-phase current (I_w), and inverter bus voltage (V_{dc}) are measured using the 12-bit A/D converter. [Used in 3-shunt mode]

DC link current and inverter bus voltage (V_{dc}) are measured using the 12-bit A/D converter. [Used in 1-shunt mode]

(2) Low Power Asynchronous General Purpose Timer (AGTW channel 0)

Uses 1 [ms] interval timer.

(3) General PWM Timer (GPT channel 0, 1, 2)

AD conversion can be executed at any timing using two AD conversion trigger compare registers.

Asymmetric waveforms can be output using complementary PWM mode 3.

(4) Port Output Enable for GPT (POEG)

The PWM output is forcibly shut off by the overcurrent signal (GTETRGA input) detected on the inverter board.

After detecting the cause for releasing the shutoff, the software releases the shutoff.

(5) Watchdog Timer (WDT)

Watchdog Timer is used as a program runaway detection.

2.2 Software specifications

2.2.1 Basic specifications

The basic software specifications of the system are listed below.

Table 2-12 Software Specifications

Item	Content	
Control method	Vector control	
Current detection method	Use the compile switch to switch between the method of detecting DC link current (1-shunt mode) and the method of detecting 3-phase current (3-shunt mode). Switch 1-shunt mode or 3-shunt mode with the compile switch (Default is 1-shunt mode)	
Motor rotation start/stop	Operation using the motor control development support tool ^{Note}	
Position detection of rotor magnetic pole (Sensorless)	Angle estimation using induced voltage Initial position detection using motor polarity and magnetic saturation characteristics	
Input voltage	DC 24 [V]	
Main clock frequency	CPU clock: ICLK 64 [MHz] GPT clock: PCLKB 32 [MHz]	
Carrier frequency (PWM)	16, 20 (default), 40 [kHz]	
Deadtime	2 [μ s]	
Control cycle	Current control / position, speed estimation(default): 100 [μ s] (1-shunt mode) 50 [μ s] (3-shunt mode) Speed control: 1 [ms]	
Rotational speed range	CW: 0 [rpm] - 4000 [rpm] CCW: 0 [rpm] - 4000 [rpm] However, driving is performed as an open loop at 800 [rpm] or less	
Optimal setting	Optimize more(-O2)	
ROM/RAM Size	ROM	1-shunt mode: 32476 B 3-shunt mode: 30476 B
	RAM	1-shunt mode: 3180 B 3-shunt mode: 3072 B
Processing stop for protection	- Disables the motor control signal output (six outputs), under any of the following conditions. <ol style="list-style-type: none"> 1. Detects a value exceeding the inverter bus voltage of 48 [V] 2. Detects a value less than the inverter bus voltage of 8 [V] 3. Detects a value exceeding the rotation speed of 8000 [rpm] 4. Detects a value where the current of each phase exceeds 2.26 [A] 5. Detects a value where the current of each phase exceeds 21.4 [A] (Forced shut-off by POEG) 	

[Note] For details, see "Usage of Motor Control Development Support Tool, Renesas Motor Workbench."

RA2T1 Sensorless vector control for permanent magnetic synchronous motor For Renesas Flexible Motor Control Series

2.2.2 Software file structure

The folder and file configurations of the sample programs are given in Table 2-12 below.

Table 2-13 Folder and File Configurations of the Sample Programs

Folder		File	Content
src/config		r_mtr_config.h	Configuration definition
		r_mtr_motor_parameter.h	Motor parameter definition
		r_mtr_control_parameter.h	Control parameter definition
		r_mtr_inverter_parameter.h	Inverter parameter definition
		r_mtr_scaling_parameter.h	Scaling parameter definition
src/ application	main	mtr_main.h mtr_main.c	Main function
		r_mtr_board.h r_mtr_board.c	Function definition for hardware UI Hardware UI
	user_interface/ ics	r_mtr_ics.h r_mtr_ics.c	Function definition for Analyzer ^(Note1) UI Analyzer ^(Note1) UI
		ICS2_RA2T1.h ICS2_RA2T1.o	Function declaration for RMW communication Library for RMW communication
		RA2T1_MCILV1_PM_LESS_F OC_1S_E2S_conv.map RA2T1_MCILV1_PM_LESS_F OC_1S_E2S.rmt	Map file (1-shunt mode) RMW initial setting file (1-shunt mode)
		RA2T1_MCILV1_PM_LESS_F OC_3S_E2S_conv.map RA2T1_MCILV1_PM_LESS_F OC_3S_E2S.rmt	Map file (3-shunt mode) RMW initial setting file (3-shunt mode)
src/driver		r_mtr_ra2t1.h r_mtr_ra2t1.c	Function definition for MCU control
		r_mtr_ADC.h r_mtr_ADC.c	Function definition for ADC driver ADC driver
		r_mtr_GPT_THREE_PHASE.h r_mtr_GPT_THREE_PHASE.c	Function definition for GPT driver GPT driver
		r_mtr_POEG.h r_mtr_POEG.c	Function definition for POEG driver POEG driver
		r_mtr_pin_data.h r_mtr_pin_data.c	Handling of unused pins
		r_mtr_common.h	Hardware common definition
src/middle		R_DSP_RA2_GCC_32BIT.h R_DSP_RA2_GCC_32BIT.o	DSP definition Arithmetic library for motor control
		rm_mtr_common.h	Common definition
		rm_mtr_parameter.h	Motor control parameter definition
		rm_mtr_ctrl_gain.h rm_mtr_ctrl_gain.o	Gain design function definition Gain design
		rm_mtr_driver.h rm_mtr_driver.c	Function definition for driver access Driver access
		rm_mtr_foc_less_speed.h rm_mtr_foc_less_speed.c	Sensorless vector control-related function definition Sensorless vector control-related
		rm_mtr_speed.c	Speed control interrupt processing
		rm_mtr_est_phase_err.h rm_mtr_est_phase_err.o	Phase error estimating function definition Phase error estimation
		rm_mtr_ol2cl_ctrl.h rm_mtr_ol2cl_ctrl.o	Open loop to Closed loop switch control function definition Open loop to Closed loop switch control
		rm_mtr_carrier.c	Carrier period interrupt processing
		rm_mtr_statemachine.h	Function definition for state machine

Note 1: Regarding the specification of the Analyzer function in the motor control development support tool Renesas Motor Workbench (RMW), please refer to Chapter 4. The identifier ics/ICS (ICS is the previous motor control development support tool, In Circuit Scope) is attached to the names of folders, files, functions, and variables related to Renesas Motor Workbench.

2.2.3 Module configuration

Figure 2-2 shows the module configuration of the sample programs.

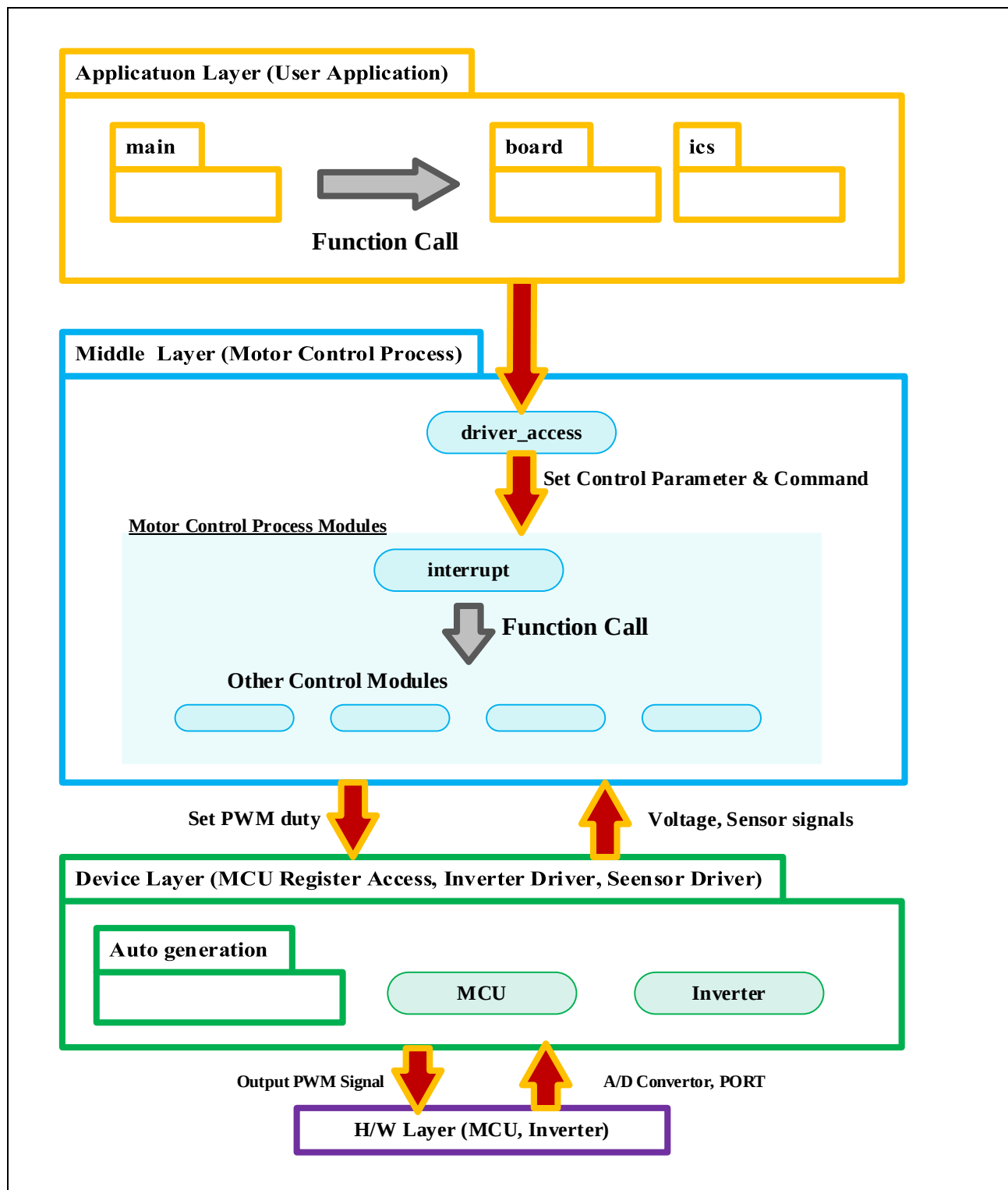


Figure 2-2 Module Configuration of the Sample Programs

3. Explanation of Control Programs

The sample programs to which this application note applies are explained here.

3.1 Contents of control

3.1.1 Motor start/stop

Starting and stopping of the motor are controlled by input from Renesas Motor Workbench.

3.1.2 A/D converter

(1) Inverter bus voltage

The inverter bus voltage is measured as given in Table 3-1. It is used for modulation factor calculation and over- and undervoltage detection. (When an abnormality is detected, PWM is stopped).

Table 3-1 Inverter Bus Voltage Conversion Ratio

Item	Conversion ratio (Inverter bus voltage: A/D conversion value)	Channel
Inverter bus voltage	0 [V] – 73.5 [V]: 0000H – 0FFFFH	AN008

(2) DC link shunt resistor current [Used in 1-shunt mode]

As shown in the table below, DC link shunt resistor current is measured and used for vector control.

Table 3-2 Conversion ratio of DC link shunt resistor current

Item	Conversion ratio (DC link shunt resistor currents: A/D conversion value)	Channel
DC link shunt resistor current	-8.25 [A] – 8.25 [A]: 0000H – 0FFFFH	AN000

(3) U-phase and W-phase current [Used in 3-shunt mode]

As shown in the table below, U-phase, V-phase and W-phase current are measured and used for vector control.

Table 3-3 Conversion Ratios of 3-Phase Currents

Item	Conversion ratio (3-phase currents: A/D conversion value)	Channel
U-phase, V-phase and W-phase current	-8.25 [A] – 8.25 [A]: 0000H – 0FFFFH	Iu: AN000 Iv: AN001 Iw: AN002

[Note] For more information about A/D conversion characteristics, see “RA2T1 User’s Manual - Hardware.”

3.1.3 POEG

(1) Overcurrent detection

It detects the low level of the overcurrent detection signal on the inverter board and forcibly cuts off the PWM output.

3.1.4 Voltage control by PWM

PWM control is used for controlling output voltage. PWM control is a control method that continuously adjusts the average voltage by varying the pulse duty, as shown in Figure 3-1.

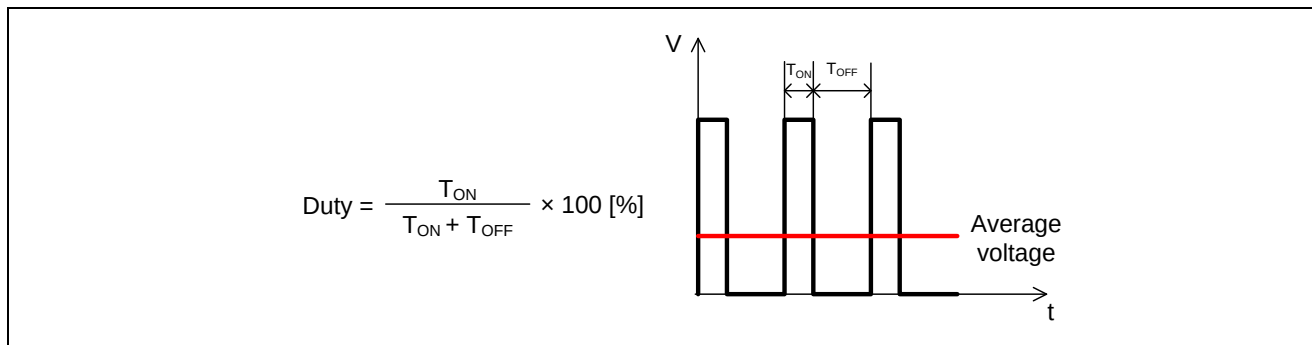


Figure 3-1 PWM Control

Here, the modulation factor m is defined as follows.

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

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

3.1.5 Modulation

The input voltage to the motor is signal generated by pulse-width modulation (below, PWM) and applied. This section explains the method of creating the PWM pulse width.

(1) Triangle Wave Comparison Method

The triangle wave comparison method is one of the method to output command value voltage. The pulse width of the output voltage is determined by comparing the carrier waveform (triangle wave) and the command voltage waveform. A sine wave-shaped command voltage output can be simulated by turning the switch on when the command voltage value is greater than the carrier wave voltage, and turning the switch off when it is smaller.

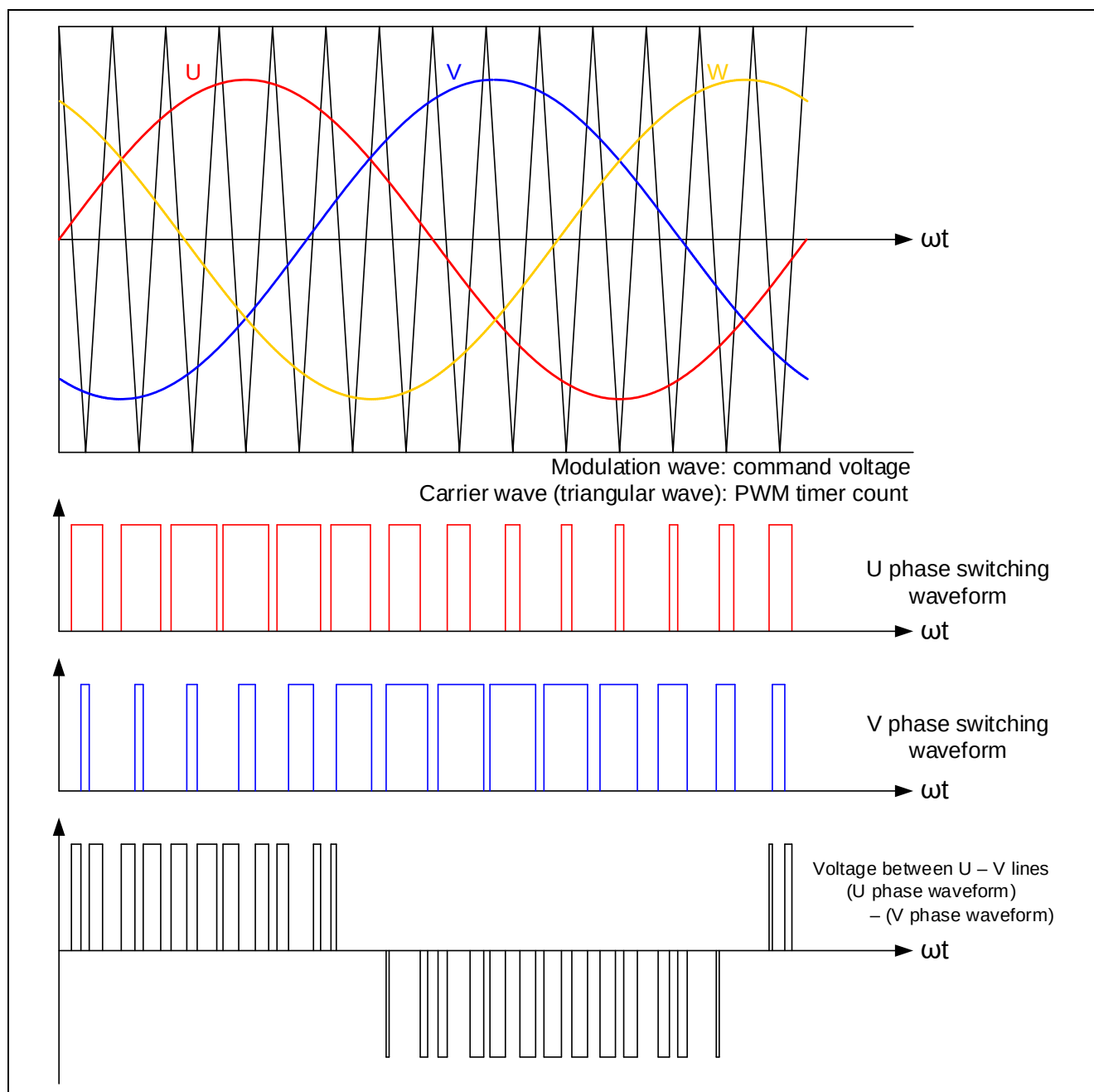


Figure 3-2 Conceptual Diagram of the Triangle Wave Comparison Method

(2) Third Harmonic Injection Method

In the triangle wave comparison method, only approximately 86.6% of the direct current voltage at which the line voltage amplitude is input can be used. There are many modulation methods for improving voltage utilization efficiency, but in this program the third harmonic imposition method can be used. By calculating the reference voltage as shown below, the reference voltage becomes the same as with third harmonic waves imposed.

$$v_o = \frac{\max(v_u^*, v_v^*, v_w^*) + \min(v_u^*, v_v^*, v_w^*)}{2}$$

$$\overline{v_u^*} = v_u^* - v_o$$

$$\overline{v_v^*} = v_v^* - v_o$$

$$\overline{v_w^*} = v_w^* - v_o$$

v_u^*, v_v^*, v_w^* : Original 3phase reference voltage

$\overline{v_u^*}, \overline{v_v^*}, \overline{v_w^*}$: 3-phase reference voltage of 3rd harmonic superimposition method

v_o : Average value of maximum reference voltage and minimum reference voltage

(3) 2-phase Modulation Method

2-phase modulation is a method in which an offset voltage is injected into the reference voltage of each phase so that it is controlled by switching only two of the three phases. In this program, the offset voltage is injected so that the reference voltage of the phase with the maximum reference voltage is equal to half the DC link voltage.

$$v_o = -\frac{V_{dc}}{2} + \max(v_u^*, v_v^*, v_w^*)$$

$$\overline{v_u^*} = v_u^* - v_o$$

$$\overline{v_v^*} = v_v^* - v_o$$

$$\overline{v_w^*} = v_w^* - v_o$$

v_u^*, v_v^*, v_w^* : Original UVW-phase reference voltage

$\overline{v_u^*}, \overline{v_v^*}, \overline{v_w^*}$: 3-phase reference voltage of 2-phase modulation

v_o : offset voltage

It is possible to change the above modulation method by setting the following values to MOD_METHOD in r_mtr_config.h, and compiling it.

MOD_3PH_SPWM	Triangle Wave Comparison Method	0
MOD_3PH_TOW	Third Harmonic Imposition Method	1: Default setting
MOD_2PH_BOT	2-phase Modulation Method	2

3.1.6 State transitions

The state transition diagram for this program is shown in Figure 3-3.

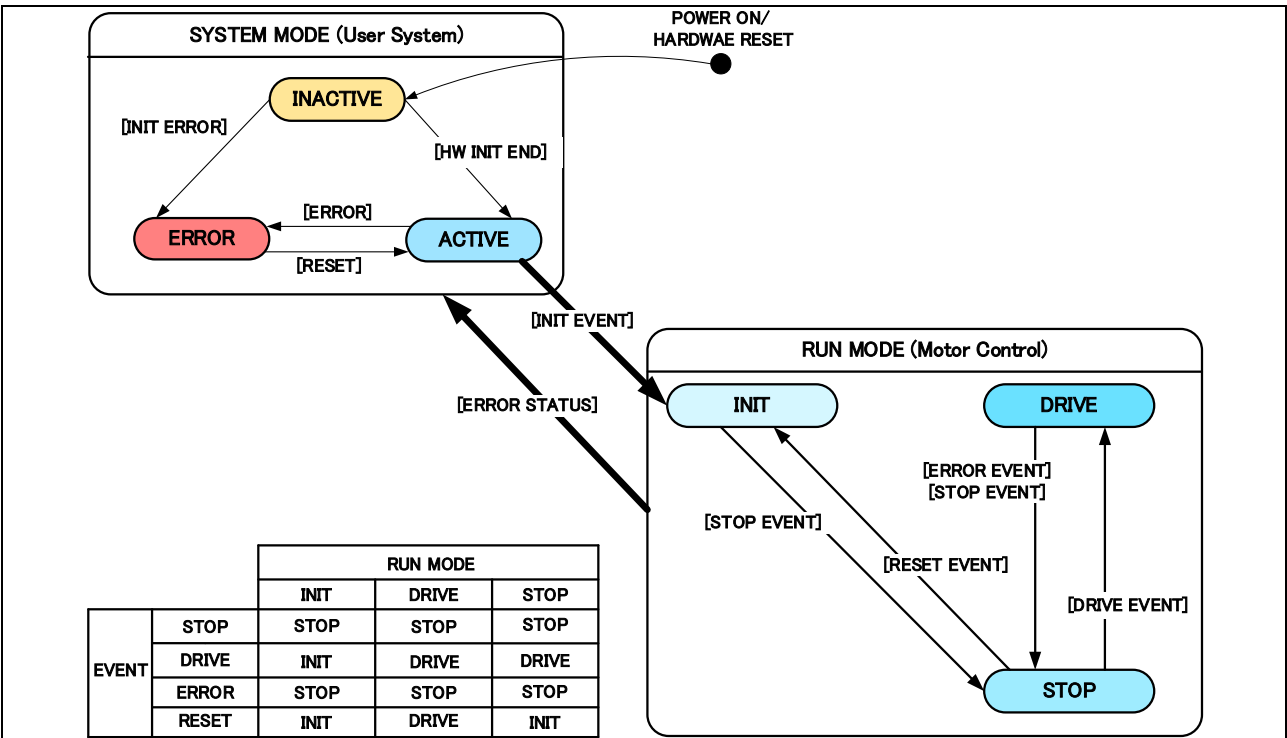


Figure 3-3 State Transition Diagram

(1) SYSTEM MODE

SYSTEM MODE indicates the operating state of the system. SYSTEM MODE has three states, which are the motor drive stop (INACTIVE), motor drive (ACTIVE), and abnormal condition (ERROR) states.

(2) RUN MODE

RUN MODE indicates the drive condition of the motor. The state is changed by the occurrence of an EVENT.

(3) EVENT

EVENT indicates a change in RUN MODE. When an EVENT occurs, the RUN MODE changes as shown in Figure 3-3. Each EVENT is caused by an occurrence as shown in Table 3-4.

Table 3-4 EVENT List

EVENT name	Occurrence factor
STOP	By user operation
DRIVE	By user operation
ERROR	When the system detects an error
RESET	By user operation

In the DRIVE event of RUN MODE, the DRIVE status changes from the table in Table 3-5 according to the drive status of the motor.

Table 3-5 DRIVE status List

Status name	Content
MTR_STOP	Motor drive stop
MTR_DRIVE	Motor drive
MTR_DRIVE_INIT	Motor drive initialization
MTR_DRIVE_START	Motor starting operation (open loop drive)
MTR_OFFSET_CALC_EXE	Execution of current offset detection process
MTR_DRIVE_END	Drive stop

3.1.7 Startup method

The description of startup control of the sensorless vector control software is shown in Figure 3-4. The mode is controlled by the states that control the reference values of the d-axis current, q-axis current, and speed.

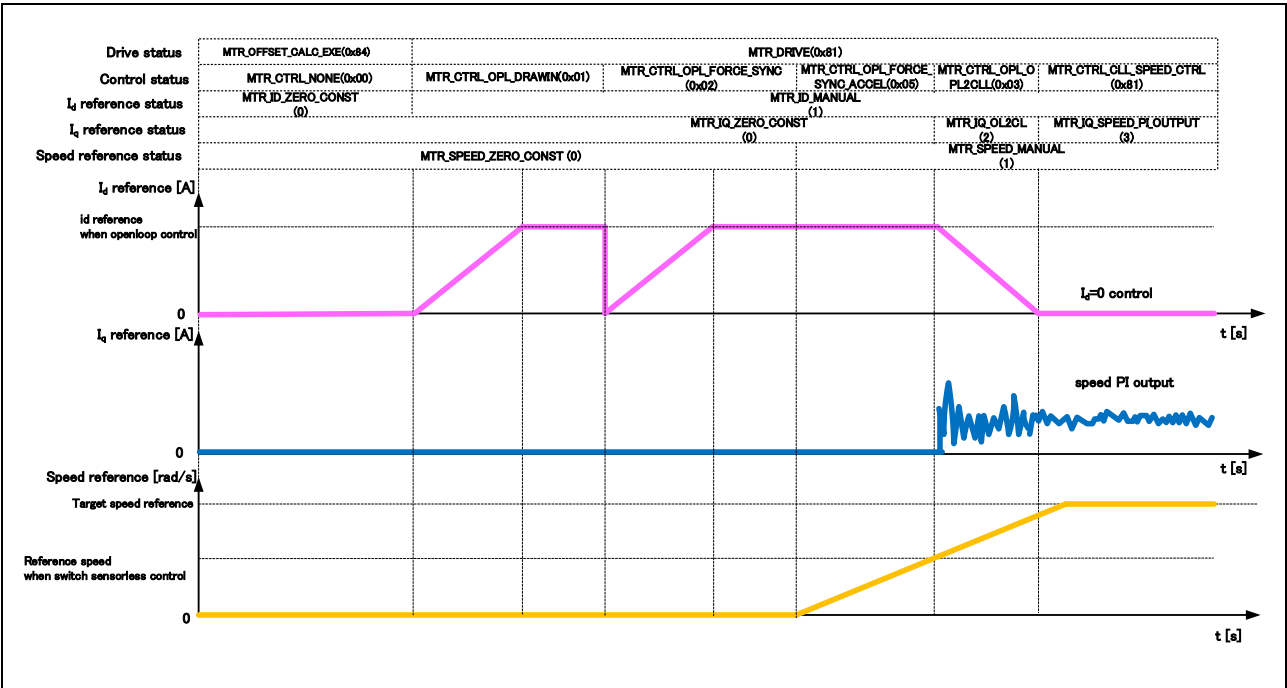


Figure 3-4 Description of Sensorless Speed Control Software Startup Control

3.1.8 Control method

The block diagram of the entire control system is shown in Figure 3-5. The control system is made up of a coordinate converter and decoupling controller, phase error estimator, PLL controller, Auto Speed Regulator (ASR), and Auto Current Regulator (ACR). Also, a primary LPF is set up for the estimated speed.

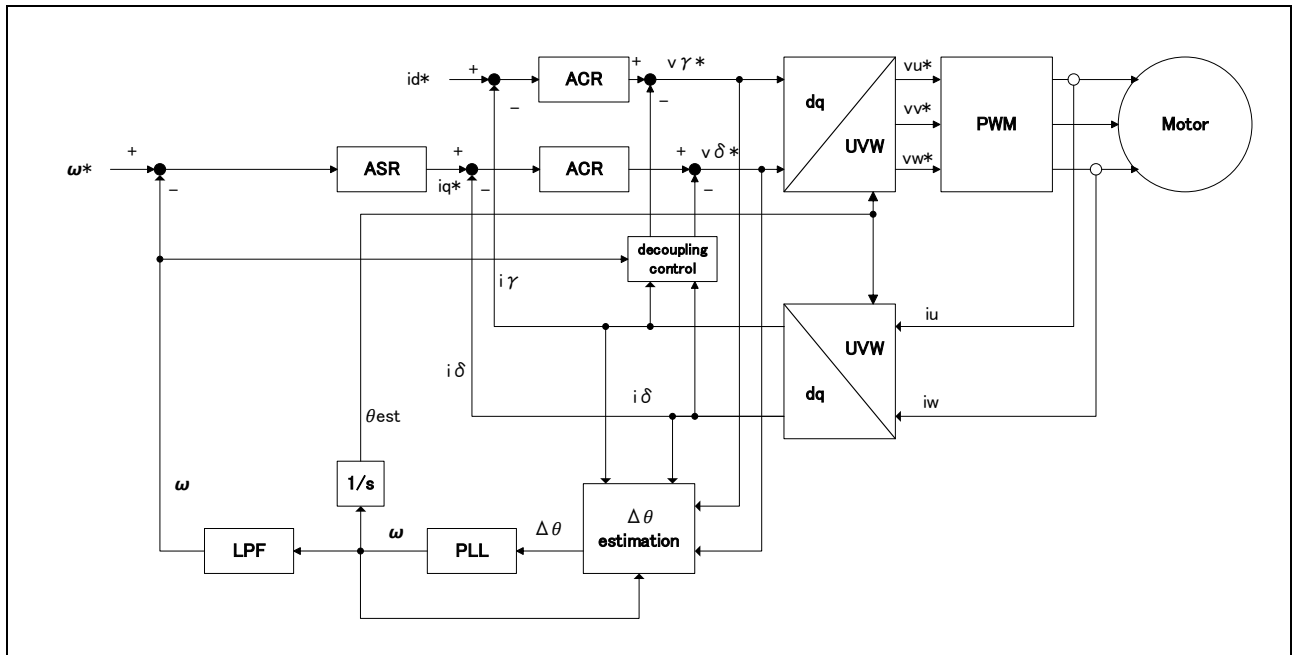


Figure 3-5 Control System Block Diagram

The phase error estimator estimates the phase error $\Delta\theta$ between the actual dq axis and the estimated $\gamma\delta$ axis. The voltage equation for the $\gamma\delta$ axis can be obtained by multiplying the rotation matrix in the formula shown below to both sides of the voltage equation for the dq axis in the following equation.

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R_a + pL_d & -\omega L_q \\ \omega L_d & R_a + pL_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega\psi_a \end{bmatrix}$$

v_d, v_q : d-axis voltage, q-axis voltage
 i_d, i_q : d-axis current, q-axis current
 R_a : resistance

ω : angular speed
 L_d, L_q : dq axis inductance
 ψ_a : Effective value of electronic interconnected magnetic flux due to permanent magnet

The following formula is the rotation matrix in which the above equation is multiplied.

$$\begin{bmatrix} \cos\Delta\theta & -\sin\Delta\theta \\ \sin\Delta\theta & \cos\Delta\theta \end{bmatrix}$$

The BEMF constituent elements e_γ and e_δ of the γ -axis and δ -axis are calculated, and the phase error $\Delta\theta$ is obtained using the following equation. When calculating the BEMF, the γ -axis voltage and the δ -axis voltage are each used in approximation with reference voltage values v_γ^* and v_δ^* . The estimated speed ω is determined by constructing a feedback loop so that this phase error reaches 0 (PLL controller).

$$\Delta\theta = \text{atan}\left(\frac{e_\gamma}{e_\delta}\right)$$

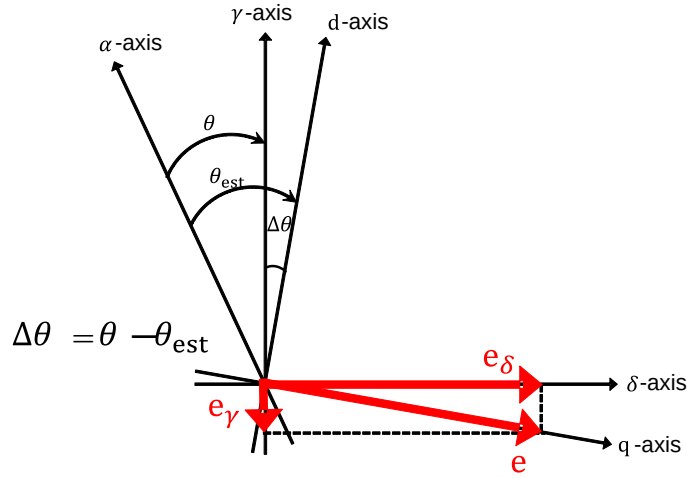


Figure 3-6 Phase Error and γ - and δ -axis Induced Voltage Constituent Elements

The ACR, ASR, and PLL controller are implemented by using the PI controller. Their gain requires suitable adjustment combined with the desired controls. The current PI control gain K_{pACR} and K_{iACR} , the speed PI control gain K_{pASR} and K_{iASR} , and the PLL control gain K_{pPLL} and K_{iPLL} are each defined as in the following formulas.

$$K_{pACR} = \omega_{CG}L, \quad K_{iACR} = \omega_{CG}R$$

ω_{CG} : Current PI control natural frequency

L : Inductance (L_d at d-axis, L_q at q-axis)

R : Resistance

$$K_{pASR} = \frac{\omega_{SG}J}{P_n^2\psi_a}, \quad K_{iASR} = \frac{\omega_{SG}D_1}{\alpha P_n^2\psi_a}$$

ω_{SG} : Speed PI control natural frequency

J : Inertia

D_1 : Kinetic friction coefficient

ψ_a : BEMF constant

P_n : Number of pole pairs

α : Adjustment parameters

$$K_{p_{PLL}} = \omega_{\Delta\theta}, \quad K_{i_{PLL}} = \frac{\omega_{\Delta\theta}^2}{5}$$

$\omega_{\Delta\theta}$: PLL control natural frequency

3.1.9 System protection function

This program has the following types of error states, and executes an emergency stop function in the event that any of the following errors occur. Refer to Table 3-6 for the settings of the system protection functions.

- Overcurrent error for hardware

When an emergency stop signal (overcurrent detection) from the external hardware is detected, voltage output is stopped.

- Overcurrent error

3-phase current are monitored in the overcurrent monitoring cycle. When overcurrent (value exceeding the overcurrent limit) is detected, an emergency stop occurs.

- Overvoltage error

The inverter bus voltage is monitored in the overvoltage monitoring cycle. When overvoltage (value exceeding the overvoltage limit) is detected, an emergency stop occurs. The overvoltage limit is set in consideration of the error of the resistance value of the detection circuit.

- Undervoltage error

The inverter bus voltage is monitored in the undervoltage monitoring cycle. When undervoltage is detected (when it goes below the undervoltage limit), an emergency stop occurs. The undervoltage limit is set in consideration of the error of the resistance value of the detection circuit.

- Rotational speed error

The speed is monitored in the rotational speed monitoring cycle. When the speed limit value is exceeded, an emergency stop occurs.

Table 3-6 System Protection Function Settings

Kinds of error	Threshold	
Overcurrent error for hardware	Overcurrent limit [A]	21.4
Overcurrent error	Overcurrent limit [A]	2.26
	Monitoring cycle	Current control / position, speed estimation
Overvoltage error	Overvoltage limit [V]	48
	Monitoring cycle	Control period for speed loop
Undervoltage error	Undervoltage limit [V]	8
	Monitoring cycle	Control period for speed loop
Rotational speed error	Speed limit [rpm]	8000
	Monitoring cycle	Control period for speed loop

3.1.10 Per-unit method (PU)

The dynamic range of motor control is determined during compiling using fixed point arithmetic. If there is a large difference between the actual motor characteristic and the hypothetical motor characteristic during design, problems such as overflow and rounding errors tend to occur due to differences in dynamic ranges. The program uses the per-unit method (PU: per-unit) in order to reduce the calculated dynamic range's dependency on the motor characteristics. The PU value of any physical quantity is its value relative to a base physical quantity, and can be derived as follows:

$$PU\ Value = \frac{Physical\ quantity}{Base\ Value}$$

All PU units used for control, such as physical quantity and gain, can be derived from the base current, base voltage, base frequency, and base angle. For example, base resistance can be calculated from the base voltage and base current:

$$Base\ Resistance = \frac{Base\ Voltage}{Base\ Current}$$

The effect of motor characteristics on calculated dynamic range is reduced, so it is necessary to set standard values for current, voltage, and angular frequency based on the motor characteristics (the method of deriving the standard value is not unique). In this program, rated current, voltage input to inverter, and maximum speed are set to standard values (PU units) for current, voltage, and angular frequency. The base value for each physical quantity is shown in Table 3-7. These values are defined in `r_mtr_scaling_parameter.h`.

Table 3-7 PU system base values

Category	Item	Definition	Unit
PU base physical quantity	Current	Rated current	[A]
	Voltage	Input voltage (inverter input)	[V]
	Angular frequency	$2\pi \times \text{maximum speed [rpm]} \times \text{number of pole pairs}/60$	[Hz]
	Angle	1	[rad]
Physical quantity	Time	Angle / Angular frequency	[s]
	Resistance	Voltage / Current	[Ω]
	Inductance	Resistance / Angular frequency	[H]
	BEMF constant	Voltage / Angular frequency	[V · s/rad]
	Inertia	$\text{BEMF constant} \times \text{current} \times (\text{number of pole pairs} / \text{angular frequency})^2$	[kgm ² /(rad ²)]
Current control	Kp	Resistance	[Ω]
	Kidt	Resistance	[Ω]
Speed control	Kp	Current / angular frequency	[A/(rad/s)]
	Kidt	Current / angular frequency	[A/(rad/s)]
PLL control	Kp	Angular frequency / angle	[Hz]
	Kidt	Angular frequency / angle	[Hz]

3.1.11 Current measurement method with 1 shunt resistor

Current measurement is performed using 1 shunt resistor.

3.1.11.1 Timing to measure the 1-shunt resistor current

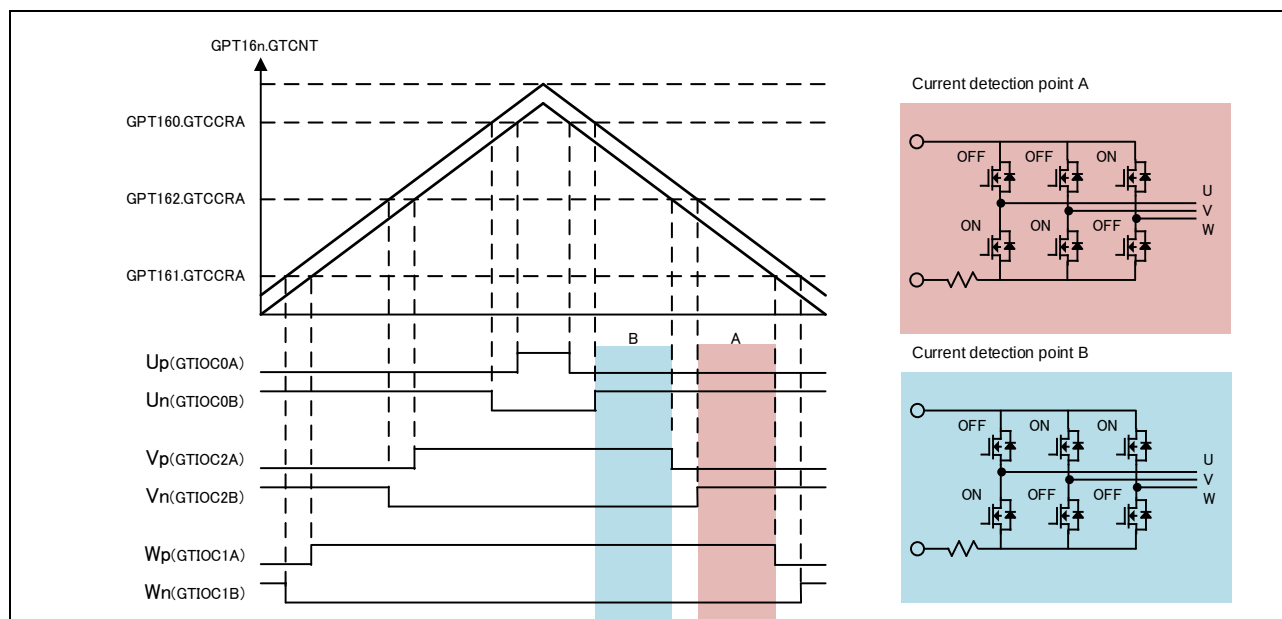


Figure 3-7 Complementary PWM waveform (For example, duty magnitude relation $W > V > U$).

In this program, the GPT complementary PWM mode is used to control by 3-phase PWM output with dead time. Figure 3-7 shows complementary PWM waveforms (For example, duty magnitude relation $W > V > U$). At point A in the figure, only the W-phase of the upper arm is ON as shown in the red frame on the upper right. In this case, the current flowing through the shunt resistor is the current flowing through the W-phase.

$$I_A = I_W$$

At point B in the figure, only the U-phase of the lower arm is ON as shown in the blue frame at the lower right. In this case, the current flowing through the shunt resistor is the current flowing through the U-phase. However, when it is used for control, the direction of the current flowing into the motor is the positive direction, so here, it is necessary to convert it to positive value.

$$I_B = -I_U$$

Since point C, which is the remaining one phase, is the current flowing in the V-phase, the combined current between the U-phase and W-phase can be obtained from Kirchhoff's first law.

$$I_C = -(I_A + I_B) = I_V$$

Therefore, if the current value flowing through 1-shunt resistor at points A and B can be obtained, the 3-phase current can be derived.

This is because the duty ratio is $W > V > U$, and the current that can be detected at points A and B is switched by the combination of six patterns of the duty ratio as the PWM changes. Since this magnitude relationship is known at the time of setting the duty, the phase of the detected current is determined accordingly.

3.1.11.2 How to measure 1-shunt resistor current

When current measurement by a 1-shunt resistor is performed as shown in 3.1.11.1, the conversion timing of the A/D converter must be adjusted according to the PWM duty setting. This program achieves this by using the complementary PWM mode function.

- General PWM Timer (GPT)
 - Complementary PWM mode 3 (Asymmetric 3-phase waveform output)
- A/D converter
 - Double trigger mode in single scan mode

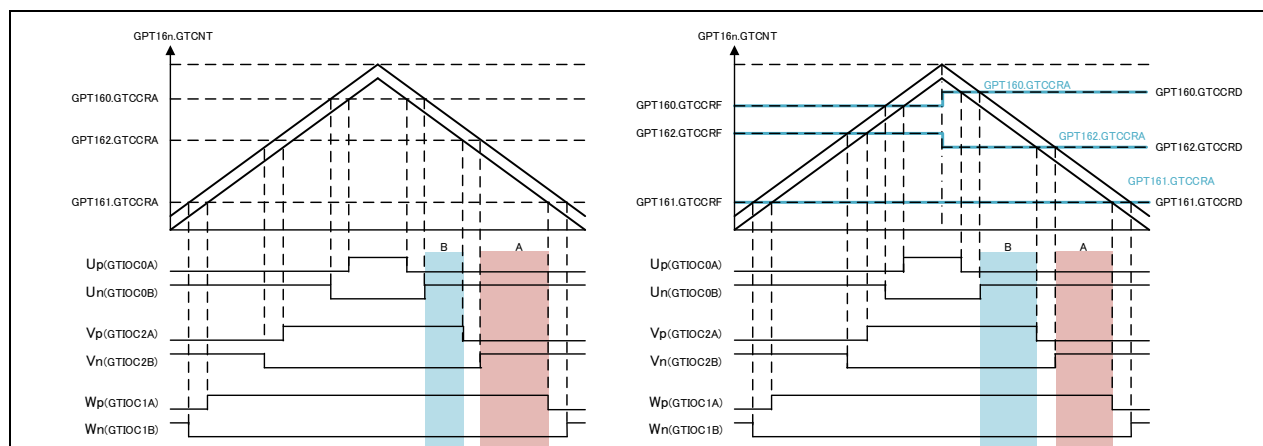


Figure 3-8 Timing of A/D Conversion

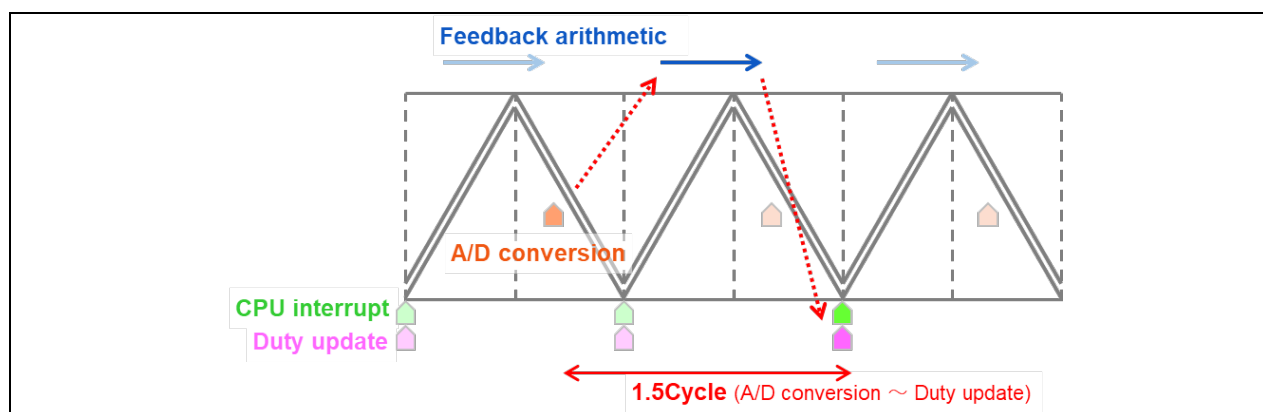


Figure 3-9 Timing of interrupt

The description in this section is based on Figure 3-8 and Figure 3-9 .

The GPT setting generates an interrupt at the valley of the carrier and calculates the next duty. The A/D conversion timing is set to the left edge of point B and the left edge of point A as GPT A/D conversion start request B and A/D conversion start request A, respectively.

The A/D converter performs A/D conversion using the GPT's A/D conversion start requests A and B as hardware triggers. The A/D conversion result is used for duty and the count value for the timings of A and B calculation at the next cycle, in which the calculated result is reflected at the valley of the next cycle (1.5 cycle later).

3-phase currents can be restored from the two A/D conversion results stored in the A/D data register and the A/D data duplication register (ADDBLDRn). If the PWM waveform cannot be kept constant during the two AD conversion times, the AD conversion time is secured by outputting an asymmetric 3-phase waveform in complementary PWM mode.

3.1.11.3 Duty adjustment

If the difference between the duty values of each phase is small, the A/D conversion may not occur in time. Therefore, to ensure the time required for A/D conversion, the duty is adjusted as shown in the figure below by outputting an asymmetrical waveform.

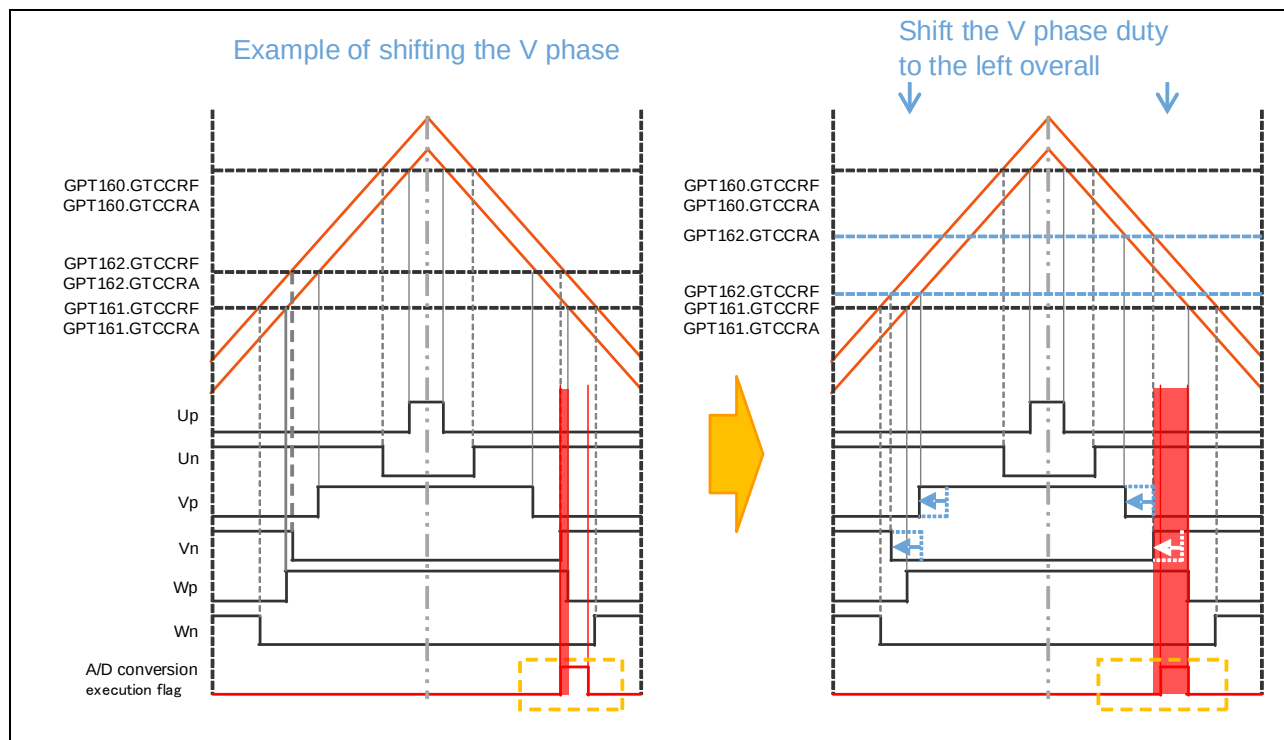


Figure 3-10 Duty adjustment (example)

3.1.12 Open loop Control

Position estimation is not possible because the BEMF is small in the low-speed range. Therefore, a rotating magnetic field is generated using the d-axis current to force the motor to be driven synchronously to a speed at which position estimation is possible (open loop control). At that time, the speed of the motor oscillates at a natural frequency that depends on the current and motor parameters. Therefore, by performing damping control as shown in the block diagram in Figure 3-11, the vibration of the motor during open loop control in the low-speed range is reduced.

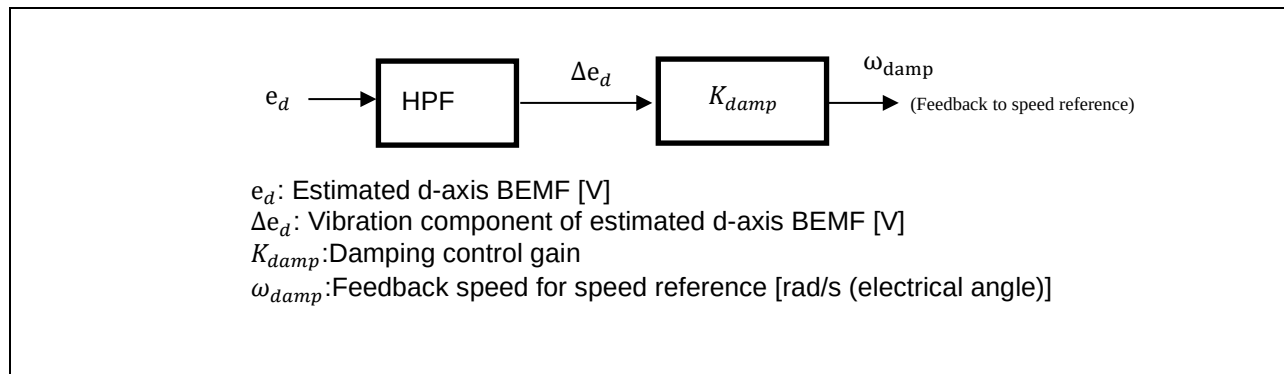


Figure 3-11 Block diagram of the open loop damping control

3.1.13 Open loop to closed loop switch control

When switching from open loop control to closed loop control, torque is generated due to the axial error between the d-axis and the γ -axis during open loop, and the motor rotates. Especially in the case of high load, etc., the shaft error is large, and hunting may occur in the current and estimated speed at the time of transition to closed loop control, resulting in unstable control. To reduce the phenomenon of hunting, the axial error is reduced to zero before shifting to speed PI control by adjusting the q-axis current.

3.2 Sensorless Vector Control Software Function Specification

A list of functions used in this control program is provided below.

Table 3-8 List of Functions in “mtr_main.c”

File	Function	Process overview
mtr_main.c	mtr_main Input: none Output: none	<ul style="list-style-type: none"> - Call hardware initialization function - Renesas Motor Workbench Communication Initialization - Call user interface initialization function - Call main processing use variable initialization function - Call state transition and event execution function - Call bus voltage stability waiting process - Main process <ul style="list-style-type: none"> ⇒ Call user interface process ⇒ Call watchdog timer clear function
	ics_ui Input: none Output: none	Uses Renesas Motor Workbench <ul style="list-style-type: none"> - Motor status change
	system_init Input: none Output: int :: Execution result	Initialization of system used for main process
	software_term Input: none Output: none	System shutdown processing
	software_init Input: none Output: none	Initialization of variable used for main process

Table 3-9 List of Functions in “r_mtr_ics.c”

File	Function	Process overview
r_mtr_ics.c	R_MTR_SetIcs Input:none Output:none	Preparation for RMW communication
	R_MTR_SetCOMVariables Input: none Output: none	Preprocess to set control variables <ul style="list-style-type: none"> - Control variable rewrite Variable value (com variable) - Input to control pass buffer variable (ICS variable) - input values of ICS variables to ICS buffer variables
	R_MTR_ICSVariablsInit Input: none Output: none	Initialization of com variables
	R_MTR_ICSIIntLevel Input: uint8_t u1_level :: Priority of interrupt Output: none	Set ICS interrupt priority
	mtr_limit (inline function) Input: int32_t s4_value :: target value int32_t s4_max :: maximum value (int32_t) s4_min :: minimum limit Output: int32_t :: limited value	Limit between maximum and minimum values

Table 3-10 List of Functions in "ICS2_RA2T1.o"

File	Function	Process overview
ICS2_RA2T1.o	ics2_watchpoint Input: none Output: none	Call transfer function Must be called at intervals of 300us or more.
	ics2_init Input: uint16_t addr :: DTC vector table start address uint16_t pin :: Pins used by SCI uint8_t level :: Interrupt level uint8_t num :: Top address of DTC structure uint8_t brr :: communication speed uint8_t mode :: Communication mode Output: none	Communication initialization
	ics2_version Input: none Output: uint32_t :: Version number	Getting the version number

Table 3-11 List of Functions in "r_mtr_board.c"

File	Function	Process overview
r_mtr_board.c	R_MTR_BoardLedContrl Input: uint8_t u1_motor_status :: motor status uint8_t u1_system_status :: system status Output: none	LED control

Table 3-12 List of Functions in "r_mtr_ADC.c"

File	Function	Process overview
r_mtr_ADC.c	R_MTR_ADC_Open Input: adc_ctrl_t *p_ctrl :: ADC information adc_cfg_t const *const p_cfg :: Config information Output: fsp_err_t :: Error number	ADC initialization
	R_MTR_ADC_WaitConversion Input: adc_ctrl_t *p_ctrl :: ADC information uint16_t timeout_count :: Timeout Counter Output: fsp_err_t :: Error number	AD conversion wait
	R_MTR_ADC_SetChannels Input: adc_ctrl_t *p_ctrl :: ADC information uint32_t scan_mask_a :: AD channel selection register A uint32_t scan_mask_b :: AD channel selection register B Output: fsp_err_t :: Error number	ADC Channel setting
	R_MTR_ADC_ReadDuplex Input: adc_ctrl_t *p_ctrl :: ADC information uint16_t *const p_data :: Read data Output: fsp_err_t :: Error number	Reading duplicated data
	R_MTR_ADC_SetSampleStateCount Input: adc_ctrl_t *p_ctrl :: ADC information Output: fsp_err_t :: Error number	Sampling state register setting

Table 3-13 List of Functions in “r_mtr_GPT_THREE_PHASE.c”

File	Function	Process overview
r_mtr_GPT_THREE_PHASE.c	R_MTR_GPT_THREE_PHASE_CompMode2_Open Input: three_phase_ctrl_t * const p_ctrl :: 3-phase control information three_phase_cfg_t const * const p_cfg :: Config information Output: fsp_err_t :: Error number	Initialize the 3-phase timer module in complementary PWM mode 2
	R_GPT_THREE_PHASE_CompMode2_DutyCycleSet Input: three_phase_ctrl_t * const p_ctrl :: 3-phase control information three_phase_duty_cycle_t * const p_duty_cycle :: Duty cycle information Output: fsp_err_t :: Error number	Duty cycle setting for 3-phase timer module in complementary PWM mode 2
	R_MTR_GPT_THREE_PHASE_CompMode3_Open Input: three_phase_ctrl_t * const p_ctrl :: 3-phase control information three_phase_cfg_t const * const p_cfg :: Config information Output: fsp_err_t :: Error number	Initialize the 3-phase timer module in complementary PWM mode 3
	R_MTR_GPT_THREE_PHASE_CompMode3_DutyCycleSet Input: three_phase_ctrl_t * const p_ctrl :: 3-phase control information three_phase_duty_cycle_t * const p_duty_cycle :: Duty cycle information Output: fsp_err_t :: Error number	Duty cycle setting for 3-phase timer module in complementary PWM mode 3
	R_MTR_GPT_THREE_PHASE_SetDeadTime Input: three_phase_ctrl_t * const p_ctrl :: 3-phase control information uint16_t up_count :: Up count uint16_t down_count :: Down count Output: fsp_err_t :: Error number	3-phase timer module dead time value setting
	R_MTR_GPT_THREE_PHASE_SetADCTrigger Input: three_phase_ctrl_t * const p_ctrl :: 3-phase control information uint16_t a_down_count :: A/D conversion start request timing A uint16_t b_down_count :: A/D conversion start request timing B Output: fsp_err_t :: Error number	ADC trigger setting for 3-phase timer module in complementary PWM mode 3
	R_MTR_GPT_THREE_PHASE_EnableOutputAll Input: three_phase_ctrl_t * const p_ctrl :: 3-phase control information Output: fsp_err_t :: Error number	Enables all outputs of the 3-phase timer module in complementary PWM mode 3
	R_MTR_GPT_THREE_PHASE_DisableOutputAll Input: three_phase_ctrl_t * const p_ctrl :: 3-phase control information Output: fsp_err_t :: Error number	Disable all outputs of the 3-phase timer module in complementary PWM mode 3

Table 3-14 List of Functions in “r_mtr_POEG.c”

File	Function	Process overview
r_mtr_POEG.c	R_MTR_POEG_SoftwareRelease Input: poeg_ctrl_t * const p_ctrl :: POEG control information uint16_t timeout_count :: Timeout Value Output: fsp_err_t :: Error number	Initializing the POEG module
	R_MTR_POEG_RecoverForcedShutdown Input: poeg_ctrl_t * const p_ctrl :: POEG control information uint16_t timeout_count :: Timeout Value uint8_t * p_is_shutdown :: Shutdown State Output: fsp_err_t :: Error number	Recovers the POEG module from a forced shutdown state

Table 3-15 List of Functions in “r_mtr_ra2t1.c”

File	Function	Process overview
r_mtr_ra2t1.c	R_MTR_InitUnusedPins Input: none Output: none	No processing

Table 3-16 List of Functions in "rm_mtr_ctrl_gain.o"

File	Function	Process overview
rm_mtr_ctrl_gain.o	RM_MTR_CtrlGain Input: st_mtr_ctrl_gain_t *st_gain_buf :: Control Gain structure pointer const st_mtr_design_parameter_t *st_ctrl_param :: Design parameter structure pointer Output: none	Gain design process

Table 3-17 List of Functions in "rm_mtr_driver.c" (1/2)

File	Function	Process overview
rm_mtr_driver.c	RM_MTR_ControllerInit Input: none Output: none	Initialization of motor control system - initialization of motor status - initialization of control variables
	RM_MTR_EventExec Input: uint8_t u1_event :: event Output: none	Change motor status and execute event process
	RM_MTR_CapacitorCharge Input: none Output: uint16_t :: timeout error	Waiting for stability of bus voltage
	RM_MTR_SpeedSet Input: int32_t s4_ref_speed_rpm :: target rotational speed int32_t s4_ref_min_speed_rpm :: minimum value of target rotational speed Output: uint8_t :: motor stop flag	Set speed command value
	RM_MTR_SpeedGet Input: none Output: int32_t :: rotational speed	Get speed
	RM_MTR_DirSet Input: int8_t s1_dir :: direction of rotation Output: none	Set direction of rotation
	RM_MTR_DirGet Input: none Output: int8_t :: direction of rotation	Get direction of rotation
	RM_MTR_StatusGet Input: none Output: uint8_t :: motor status	Get motor status
	RM_MTR_StatusErrorGet Input: none Output: uint16_t :: error status	Get error status
	RM_MTR_IcsBufferSet Input: none Output: none	Input values of ICS variables to ICS buffer variables
	RM_MTR_IcsVariablesSet Input: none Output: none	Input values of ICS buffer variables to control variables
	RM_MTR_InputBuffParamReset Input: none Output: none	Reset ICS buffer variables
	RM_MTR_UpdatePollingSet Input: none Output: none	Set control variables

Table 3-18 List of Functions in “rm_mtr_driver.c” (2/2)

File	Function	Process overview
rm_mtr_driver.c	mtr_statemachine_init Input: (st_mtr_statemachine_t) *p_state_machine :: Motor Status Structure Output: none	Motor status initialization
	mtr_statemachine_reset Input: (st_mtr_statemachine_t) *p_state_machine :: Motor Status Structure Output: none	Reset the motor status
	mtr_state_machine_event Input: (st_mtr_statemachine_t) *p_state_machine :: Motor Status Structure (void) *p_object :: Structure for control variables (uint8_t) u1_event :: Event Output: (uint8_t) p_state_machine->u2_error_status :: Error Status	Event Execution
	mtr_statemachine_get_status Input: (st_mtr_statemachine_t) *p_state_machine :: Motor Status Structure Output: (uint8_t) p_state_machine->u1_status :: Motor Status	Get motor status
	mtr_act_none Input: (st_mtr_statemachine_t) *st_stm :: Motor Status Structure (void) *p_param :: Structure for control variables Output: (uint8_t) action_ret :: Execution result (always 0)	No processing
	mtr_act_init Input: (st_mtr_statemachine_t) *st_stm :: Motor Status Structure (void) *p_param :: Structure for control variables Output: (uint8_t) action_ret :: Execution result (always 0)	Initialization of control variables
	mtr_act_error Input: (st_mtr_statemachine_t) *st_stm :: Motor Status Structure (void) *p_param :: Structure for control variables Output: (uint8_t) action_ret :: Execution result (always 0)	Post-processing when an error is detected
	mtr_act_drive Input: (st_mtr_statemachine_t) *st_stm :: Motor Status Structure (void) *p_param :: Structure for control variables Output: (uint8_t) action_ret :: Execution result (always 0)	Motor start
	mtr_act_stop Input: (st_mtr_statemachine_t) *st_stm :: Motor Status Structure (void) *p_param :: Structure for control variables Output: (uint8_t) action_ret :: Execution result (always 0)	Stopping the motor

Table 3-19 List of Functions in “rm_mtr_foc_less_speed.c”

File	Function	Process overview
rm_mtr_foc_less_speed.c	RM_MTR_FOCMotorDefaultInit Input: st_mtr_common_foc_t *p_st_foc_common :: Sensorless FOC main data structure pointer Output: none	Initialization of control variables
	RM_MTR_FOCMotorReset Input: st_mtr_common_foc_t *p_st_foc_common :: Sensorless FOC main data structure pointer Output: none	Reset control variables

Table 3-20 List of Functions in “rm_mtr_carrier.c” (1/4)

File	Function	Process overview
rm_mtr_carrier.c	mtr_carrier_cyclic Input: timer_callback_args_t *p_args :: interrupt handler argument pointer [Used in 3-shunt mode] adc_callback_args_t *p_args :: interrupt handler argument pointer [Used in 3-shunt mode] Output: none	Carrier interrupt handler Periodic timer interrupt (called by GPT) Period: Used in 1-shunt mode 100[μs] (1 thinning of 50[μs]) Used in 3-shunt mode 50[μs] • Call vector control function • Obtain inverter bus voltage • Call communication processing • Call vector control preparation function
	mtr_carrier_foc_sequence [inline function] Input: none Output: none	Vector Control Function
	mtr_carrier_vdc_get Input: int32_t *p_s4_vdc_ad :: VDC pointer st_mtr_mod_t *p_st_mod :: modulation structure pointer Output: none	Inverter bus voltage detection • Calculate inverter bus voltage from A/D conversion value
	mtr_carrier_current_ss_get [inline function] Input: st_mtr_sscs_t *p_st_sscs :: 1-shunt resistor current detection structure pointer st_coordinate28 *p_st_ad_i :: UVW phase detection current pointer st_coordinate28 *p_st_repro_i :: reproduced current coordinate system structure pointer Output: none	Current detection [Used in 1-shunt mode] • Calculate phase current from A/D conversion value • During 2-phase cross current compensation, the detected current is replaced by the reproduced current coordinate.
	mtr_carrier_current_ts_get [inline function] Input: st_mtr_tscs_t *p_st_tscs :: 3-shunt current detection structure pointer st_coordinate28 *p_st_ad_i :: UVW phase detection current pointer Output: none	Current detection [Used in 3-shunt mode] • Calculate phase current from A/D conversion value
	mtr_carrier_lim_sw_over_current [inline function] Input: st_coordinate28 *p_st_ad_i :: UVW phase detection current pointer Output: uint16_t :: error status	Overcurrent error check for SW
	mtr_carrier_uvw2dq [inline function] Input: st_coordinate28 *p_st_ad_i :: ADC current coordinate system structure pointer st_sincos28 *p_ad_i_angle :: angular struct pointer Output: none	Conversion of detected current (U,V,W-phase → d,q axis)
	mtr_carrier_est_phase_err [inline function] Input: st_coordinate28 *p_st_ad_i :: ADC current coordinate system structure pointer st_mtr_parameter_t *p_st_motor :: motor parameter structure pointer st_mtr_estimate_phe_t *p_st_phe :: phase error estimation structure pointer int32_t *p_s4_vdq :: dq axis command voltage pointer st_mtr_asr_t *p_st_asr :: ASR structure pointer Output: int32_t :: axis error estimate	Axis error estimate process
	mtr_carrier_est_speed_cll [inline function] Input: st_mtr_pll_t *p_st_pll :: PLL control structure pointer st_mtr_asr_t *p_st_asr :: ASR structure pointer int32_t *p_ad_i_angle_rad :: detected current angle info. int32_t *p_ref_v_angle_rad :: command voltage angle info. st_mtr_common_foc_t *p_st_common :: Sensorless FOC main data structure pointer Output: none	Closed-loop speed estimation

Table 3-21 List of Functions in "rm_mtr_carrier.c" (2/4)

File	Function	Process overview
rm_mtr_carrier.c	mtr_carrier_est_speed_opl [inline function] Input: st_mtr_estimate_phe_t *p_st_phe :: phase error estimate structure pointer st_mtr_asr_t *p_st_asr :: ASR structure pointer st_mtr_pll_t *p_st_pll :: PLL control structure pointer int32_t *p_ad_i_angle_rad :: detected current angle info. int32_t *p_ref_v_angle_rad :: reference voltage angle info. Output: none	Open loop speed estimation
	mtr_carrier_get_direction [inline function] Input: int32_t p_s4_direction :: rotation direction Output: int8_t :: rotation direction	Get the motor rotation direction CW:1, CCW:-1
	mtr_carrier_ctrl_pi [inline function] Input: int32_t *p_s4_ad_idq :: ADC current st_coordinate28 *p_st_ref_i :: reference current coordinate system structure pointer st_coordinate28 *p_st_ref_v :: reference voltage coordinate system structure pointer st_mtr_acr_t *p_st_acr :: ACR structure pointer Output: none	Current PI control
	mtr_carrier_ctrl_decoupling [inline function] Input: st_mtr_parameter_t *p_st_motor :: motor parameter structure pointer int32_t *p_s4_ad_idq :: ADC current st_coordinate28 *p_st_ref_v :: reference voltage coordinate system structure pointer st_mtr_asr_t *p_st_asr :: ASR structure pointer Output: none	Decoupling controller
	mtr_carrier_dq2uvw_voltage [inline function] Input: st_coordinate28 *p_st_ref_v :: reference voltage coordinate system structure pointer Output: none	Conversion of detected voltage (d,q axis → U,V,W-phase)
	mtr_carrier_set_pwm_ss [inline function] Input: st_coordinate28 *p_st_ref_v :: reference voltage coordinate system structure pointer st_mtr_mod_t *p_st_mod :: modulation structure pointer st_mtr_sscs_t *p_st_sscs :: 1-shunt resistor current detection structure pointer st_coordinate28 *p_st_ref_i :: reference current coordinate system structure pointer Output: none	Setting of PWM output [Used in 1-shunt mode]
	mtr_carrier_set_pwm_ts [inline function] Input: st_coordinate28 *p_st_ref_v :: reference voltage coordinate system structure pointer st_mtr_mod_t *p_st_mod :: modulation structure pointer st_mtr_tscs_t *p_st_tscs :: 3-phase current detection structure pointer st_coordinate28 *p_st_ref_i :: reference current coordinate system structure pointer Output: none	Setting of PWM output [Used in 3-shunt mode]

Table 3-22 List of Functions in "rm_mtr_carrier.c" (3/4)

File	Function	Process overview
rm_mtr_carrier.c	mtr_carrier_conv_ad_ss [inline function] Input: st_mtr_sscs_t *p_st_sscs :: 1-shunt resistor current detection structure pointer Output: none	A/D conversion timing calculation [Used in 1-shunt mode]
	mtr_carrier_2phase_duty_cross [inline function] Input: st_coordinate28 *p_st_ad_i :: ADC current coordinate system structure pointer st_coordinate28 *p_st_repro_i :: reproduction current coordinate system structure pointer st_mtr_lpf1_t *p_st_repro_id :: Id for reproduction current LPF pointer st_mtr_lpf1_t *p_st_repro_iq :: Iq for reproduction current LPF pointer Output: none	2-phase cross current compensation
	mtr_carrier_deadtime_comp [inline function] Input: st_coordinate28 *p_st_ref_i :: reference current coordinate system structure pointer st_coordinate28 *p_st_ref_v :: reference voltage coordinate system structure pointer st_mtr_s4_comp_t *p_st_s4_comp :: deadtime compensation parameter structure pointer Output: none	Deadtime compensation process
	mtr_carrier_mod_ss [inline function] Input: st_mtr_mod_t *p_st_mod :: modulation structure pointer st_coordinate28 *p_st_ref_v :: reference voltage coordinate system structure pointer uint8_t *p_u1_drv_pat :: PWM magnitude relationship pattern st_mtr_sscs_t *p_st_sscs :: 1-shunt resistor current detection structure pointer st_coordinate28 *p_st_ref_i :: reference current coordinate system structure pointer Output: none	Modulation process [Used in 1-shunt mode]
	mtr_carrier_mod_ts [inline function] Input: st_mtr_mod_t *p_st_mod :: modulation structure pointer st_coordinate28 *p_st_ref_v :: reference voltage coordinate system structure pointer st_coordinate28 *p_st_ref_i :: reference current coordinate system structure pointer Output: none	Modulation process [Used in 3-shunt mode]
	mtr_carrier_pwm_duty_ss [inline function] Input: st_mtr_sscs_t *p_st_sscs :: 1-shunt resistor current detection structure pointer Output: none	Duty calculation [Used in 1-shunt mode]
	mtr_carrier_pwm_duty_nonadc_ss [inline function] Input: st_mtr_sscs_t *p_st_sscs :: 1-shunt resistor current detection structure pointer Output: none	Duty calculation [Used in 1-shunt mode]
	mtr_carrier_pwm_duty_ts [inline function] Input: st_mtr_tscs_t *p_st_tscs :: 3-phase current detection structure pointer st_mtr_mod_t *p_st_mod :: modulation structure pointer Output: none	Duty calculation [Used in 3-shunt mode]
	mtr_carrier_set_duty_adj_ss [inline function] Input: st_mtr_sscs_t *p_st_sscs :: 1-shunt resistor current detection structure pointer Output: none	Duty calculation [Used in 1-shunt mode]
	mtr_carrier_lim_uvw_voltage [inline function] Input: int32_t *p_s4_ref_v_uvw :: UVW-phase voltage int32_t s4_voltage_limit :: voltage limit value Output: none	3-phase voltage limit process

Table 3-23 List of Functions in "rm_mtr_carrier.c" (4/4)

File	Function	Process overview
rm_mtr_carrier.c	mtr_carrier_startup_sequence [inline function] Input: none Output: none	Vector control preparation
	mtr_carrier_offset_ss [inline function] Input: st_mtr_sscs_t *p_st_sscs :: 1-shunt resistor current detection structure pointer Output: uint8_t :: drive mode status	Current offset detection [Used in 1-shunt mode]
	mtr_carrier_offset_ts [inline function] Input: st_mtr_tscs_t *p_st_tscs :: 3-phase current detection structure pointer Output: uint8_t :: drive mode status	Current offset detection [Used in 3-shunt mode]
	mtr_limit [inline function] Input: int32_t s4_value :: target value int32_t s4_max :: maximum value int32_t s4_min :: minimum value Output: int32_t :: limit value	Maximum and minimum limitation process
	mtr_limit_or_zero [inline function] Input: int16_t s2_value :: target value int16_t s2_max :: maximum value int16_t s2_min :: minimum value int16_t s2_zero_line :: 0 correction threshold Output: int16_t :: limit value	Maximum and minimum limitation process with 0 correction
	mtr_lpf1_run [inline function] Input: st_mtr_lpf1_t *p_st_lpf :: LPF structure pointer int32_t s4_input :: LPF input const uint8_t u1_q :: Q value of LPF Output: int32_t :: LPF output	Primary LPF process
	mtr_pi_run_acr [inline function] Input: st_mtr_pi_t *p_st_pi :: PI control structure pointer int32_t s4_err :: deviation const uint8_t u1_kp_q :: Q value of proportional gain const uint8_t u1_q_err_shift :: shift value of error number const uint8_t u1_q_kidt_shift :: Q value of integral gain const uint8_t u1_q_intg_shift :: shift value of integer part Output: int32_t :: PI output	Current PI control process
	mtr_abs [inline function] Input: int32_t s4_value :: input value Output: int32_t :: output value	Output to absolute value of input
	mtr_limit_abs [inline function] Input: int32_t s4_value :: input value int32_t s4_limit_value :: limit value Output: int32_t :: output value	limit input by absolute value
	mtr_carrier_pwm_duty_calc [inline function] Input: int32_t s4_value :: duty value Output: int16_t :: calculated value	Calculating the PWM Duty Value

Table 3-24 List of Functions in “rm_mtr_est_phase_err.o”

File	Function	Process overview
rm_mtr_est_phase_err.o	RM_MTR_ESTIMATE_AngleSpeedGet Input: st_mtr_est_phe_t *st_phe :: phase error estimation structure pointer int32_t s4_est_speed :: speed st_mtr_est_vect_t *p_st_est_vect :: phase error estimate structure pointer Output: int32_t :: phase error	Phase error estimating process

Table 3-25 List of Functions in “RM_DSP_RA2_GCC_32BIT.o”

File	Function	Process overview																		
RM_DSP_RA2_G CC_32BIT.o	R_motor_uw2dq_abs_FIX28 Input: st_coordinate28 *p_uw2dq :: Coordinate transformation structure pointer Input range: $-4\sqrt{2/3} \leq u, w \leq 4\sqrt{2/3}$ (FIX28) Output: none Output range: $-4 \leq d, q \leq 4$ (FIX28)	Composite transformation of Clark transformation and Park transformation (absolute transformation) Convert U-phase (FIX 28 u) and W-phase (FIX 28 w) values to d-axis and q-axis values and store them in (FIX 28 d) and (FIX 28 q), respectively.																		
	R_motor_dq2uvw_abs_FIX28 Input: st_coordinate28 *p_dq2uvw :: Coordinate transformation structure pointer Input range: $-4 \leq d, q \leq +4$ (FIX28) $-1 \leq \text{angle.sin}, \text{angle.cos} \leq 1$ (FIX28) Output: none Output range: $-4\sqrt{2/3} \leq U, V, W \leq 4\sqrt{2/3}$ (FIX28)	Composite transformation of inverse Clark transformation and inverse Park transformation (absolute transformation) Converts d-axis (FIX 28 d) and q-axis (FIX 28 q) values to UVW-phase values and stores in (FIX 28 u), (FIX 28 u) and (FIX 28 w), respectively.																		
	R_motor_sincos_FIX28 Input: st_sincos28 *p_sincos :: Angle structure pointer Input range: $-2\pi \leq \text{angle_rad} \leq 2\pi$ (FIX28) Output: none Output range: $-1 \leq \sin, \cos \leq 1$ (FIX28)	Calculate sine and cosine values from the angle (FIX 28 angle) and store them in (FIX 28 sin) and (FIX 28 cos), respectively.																		
	R_motor_atan2_FIX28 Input: FIX28 x :: input value x FIX28 y :: input value y Input range : $-4 \leq x, y \leq 4$ (FIX28) Output: FIX28 atan(y/x) Output range: $-\pi \leq \text{atan} \leq \pi$ (FIX28)	Calculate the principal value of arctangent to (y / x) from the input values x (FIX28 x) and y (FIX28 y) and output it as the (FIX28) type. Output when the input contains 0 <table><tr><td>x</td><td>y</td><td>atan</td></tr><tr><td>0</td><td>Positive</td><td>$\pi/2$</td></tr><tr><td>0</td><td>Negative</td><td>$-\pi/2$</td></tr><tr><td>0</td><td>0</td><td>$\pi/4$</td></tr><tr><td>Positive</td><td>0</td><td>0</td></tr><tr><td>Negative</td><td>0</td><td>$-\pi$</td></tr></table>	x	y	atan	0	Positive	$\pi/2$	0	Negative	$-\pi/2$	0	0	$\pi/4$	Positive	0	0	Negative	0	$-\pi$
	x	y	atan																	
	0	Positive	$\pi/2$																	
0	Negative	$-\pi/2$																		
0	0	$\pi/4$																		
Positive	0	0																		
Negative	0	$-\pi$																		
R_motor_sqrt_sum_FIX28 Input: FIX28 x :: input value x FIX28 y :: input value y Input range : $-4 \leq x, y \leq 4$ (FIX28) and $\sqrt{x^2 + y^2} \leq 4$ (FIX28) Output: FIX28 $\sqrt{x^2 + y^2}$ Output range: $0 \leq \text{sqrt} \leq 4\sqrt{2}$ (FIX28)	Calculate the value of $\sqrt{x^2 + y^2}$ from the input values x and y and output as the (FIX28) type.																			
R_motor_sqrt_dif_FIX28 Input: FIX28 x :: input value x FIX28 y :: input value y Input range : $-4 \leq x, y \leq 4$ (FIX28) Output: FIX28 $\sqrt{x^2 - y^2}$ Output range: $0 \leq \text{sqrt} \leq 4$ (FIX28)	Calculate the value of $\sqrt{x^2 - y^2}$ from the input values x and y and output as the (FIX28) type.																			

Table 3-26 List of Functions in "rm_mtr_speed.c" (1/2)

File	Function	Process overview
rm_mtr_speed.c	mtr_speed_cyclic Input: timer_callback_args_t *p_args :: Interrupt Handler Argument Pointer Output: none	Speed Control Interrupt Handler Cycle timer interrupt Cycle: 1 ms - Startup control - Field weakening control/maximum torque control - Call command value setting process for d-axis and q-axis current and rotational speed - Call speed PI control process - Call error monitoring process
	mtr_lpf1_run Input: st_mtr_lpf1_t *p_st_lpf :: LPF structure pointer int32_t s4_input :: LPF input const uint8_t u1_q :: Q value of LPF Output: int32_t :: LPF output	Primary LPF process
	mtr_ref_speed_get Input: st_mtr_foc_speed_t *p_st_speed :: FOC speed control data structure pointer Output: int32_t :: speed command value	Set command value for speed control
	mtr_pi_run_asr Input: st_mtr_pi_t *p_st_pi :: PI control structure pointer int32_t s4_err :: deviation const uint8_t u1_kp_q :: Q value of proportional gain const uint8_t u1_q_err_shift :: shift value of error number const uint8_t u1_q_kidt_shift :: Q value of integral gain const uint8_t u1_q_intg_shift :: shift value of integer part Output: int32_t :: PI output	Speed PI control process
	mtr_speed_drive_ctrl Input: st_mtr_common_foc_t *p_st_foc_common :: Sensorless FOC main data structure pointer Output: int8_t :: control status	Motor Speed Drive Control
	mtr_speed_process_opl_ctrl Input: st_mtr_common_foc_t *p_st_foc_common :: Sensorless FOC main data structure pointer uint8_t u1_state_ctrl :: control status int32_t s4_ref_temp_idq_ctrl[2] :: current component reference value int32_t *p_s4_ref_temp_speed_ctrl :: reference speed control value Output: int8_t :: control status	Motor speed control process in open loop control mode
	mtr_speed_process_cll_ctrl Input: st_mtr_common_foc_t *p_st_foc_common :: Sensorless FOC main data structure pointer uint8_t u1_state_ctrl :: control status int32_t s4_ref_temp_idq_ctrl[2] :: current component reference value int32_t *p_s4_ref_temp_speed_ctrl :: reference speed control value Output: int8_t :: control status	Motor speed control process in closed loop control mode

Table 3-27 List of Functions in "rm_mtr_speed.c" (2/2)

File	Function	Process overview
rm_mtr_speed.c	mtr_speed_state_ctrl_get Input: none Output: uint8_t :: drive control status	Get drive control status
	mtr_ref_iq_get Input: st_mtr_speed_t *p_st_speed :: FOC structure pointer st_mtr_foc_carrier_t *p_st_carrier :: speed control structure pointer Output: int32_t :: q-axis current command value	Set q-axis current command value
	mtr_ref_id_get Input: st_mtr_foc_speed_t *p_st_speed :: FOC speed control data structure pointer st_mtr_foc_carrier_t *p_st_carrier :: speed control structure pointer Output: int32_t :: d-axis current command value	Set d-axis current command value
	mtr_offset_calc_init Input: none Output: none	Initialize the offset calculation of the current detection
	mtr_offset_calc_status_get Input: none Output: uint8_t :: Calculation Processing Status	Get current detection offset calculation status
	mtr_error_check Input: st_mtr_foc_carrier_t *p_st_carrier :: speed control structure pointer Output: none	Error process - Overvoltage detection - Undervoltage detection - Excessive speed detection
	mtr_abs [inline function] Input: int32_t s4_value :: input value Output: int32_t :: output value	Output absolute value of input
	mtr_limit_abs [inline function] Input: int32_t s4_value :: input value int32_t s4_limit_value :: limit value Output: int32_t :: output value	Limit input by absolute value

Table 3-28 List of Functions in “rm_mtr_ol2cl_ctrl.o”

File	Function	Process overview
rm_mtr_ol2cl_ctrl.o	RM_MTR_OplOL2CLiqCalc Input: st_mtr_ol2cl_t *st_ol2cl :: Open loop to Closed loop switch control structure pointer int32_t s4_ref_id :: Id reference int8_t s1_dir :: direction of rotation Output: none	Torque current calculation for Open loop to Closed loop switch control
	RM_MTR_OplOL2CLRefIqGet Input: st_mtr_ol2cl_t *st_ol2cl :: Open loop to Closed loop switch control structure pointer int32_t s4_ref_id :: Id reference Output: int32_t :: Iq reference	Iq reference calculation for Open loop to Closed loop switch control
	RM_MTR_OpenloopDampCtrl Input: st_mtr_damp_t *st_damp :: axis error estimate structure pointer int32_t s4_ed :: d-axis Back-EMF voltage int32_t s4_ref_speed :: speed reference Output: int32_t :: damping speed reference	Damping control process
	mtr_hpf1_run [inline function] Input: st_mtr_hpf1_t *st_hpf :: HPF structure pointer int32_t s4_input :: HPF Input const uint8_t u1_q :: Q value of HPF Output: int32_t :: HPF Output	Primary HPF process
	mtr_limit_abs [inline function] Input: int32_t s4_value :: input value int32_t s4_limit_value :: limit value Output: int32_t :: output value	Input limitation by absolute value
	mtr_abs [inline function] Input: int32_t s4_value :: input value Output: int32_t :: output value	Output to absolute value of input

3.3 List of Sensorless Vector Control Software Function Variables

A list of variables used in this control program is provided below. However, note that the local variables are not listed. The control values in this control program are calculated after scaling each value. Regarding the variables to which the Q notation is applied, Qn in the scaling column indicates “n” bits, which is the fractional part of a value. However, the Q notation for some variables and structure members is calculated using definitions in `r_mtr_scaling_parameter.h`, so the default Q notation is written in the scale field in these cases. Variable/structure member units to which PU units are applied are written as [PU ([original unit])].

Table 3-29 List of Variables in “mtr_main.c”

Variable	Type	Qn	PU	Content	Remarks
g_u1_system_mode	static uint8_t	Q0	-	Mode system management	
g_u1_motor_status	static uint8_t	Q0	-	Motor status management	
g_u2_error_status	static uint16_t	Q0	-	Error status management	
g_u2_conf_hw	uint16_t	Q0	-	RMW configuration	
g_u2_conf_sw	uint16_t	Q0	-		
g_u2_conf_tool	uint16_t	Q0	-		
gui_u1_active_gui	uint8_t	Q0	-		
g_u2_conf_sw_ver	uint16_t	Q0	-		
com_u1_run_event	uint8_t	Q0	-	Change run mode	0 : MTR_EVENT_STOP 1 : MTR_EVENT_DRIVE 2 : MTR_EVENT_ERROR 3 : MTR_EVENT_RESET
g_u1_run_event	uint8_t	Q0	-		

Table 3-30 List of Variables in “r_mtr_ics.c”

Variable	Type	Qn	PU	Content	Remarks
st_ics	st_mtr_ctrl_input_t	-	-	Structure for ICS variable transfer	Structure
g_st_com	st_com_t	-	-	COM variable structure	Structure
g_st_com_init	st_com_t	-	-	Initial values for the COM variable structure	Structure const
g_st_design_param_macro	st_mtr_design_parameter_macro_t	-	-	Initial values of the control parameter structure	Structure const
g_s2_enable_write	int16_t	Q0	-	Variable to allow variable rewriting	
s_u1_cnt_ics	static uint8_t	Q0	-	Communication processing cycle delay variable	

Table 3-31 List of Variables in “rm_mtr_pin_data.c”

Variable	Type	Qn	PU	Content	Remarks
g_user_pin_cfg_data[]	ioport_pin_cfg_t	-	-	Pin config information	
g_user_pin_ctrl	ioport_instance_ctrl_t	-	-	Pin information	
g_user_pin_cfg	ioport_cfg_t	-	-	Pin config information	

Table 3-32 List of Variables in “rm_mtr_driver.c”

Variable	Type	Qn	PU	Content	Remarks
g_st_ctrl_params	st_mtr_design_parameter_t	-	-	Control parameters	
g_st_gain_buf	st_mtr_ctrl_gain_t	-	-	Input Gain Data	
g_u1_trig_enable_write	uint8_t	Q0	-	Transfer completion flag	
g_u1_stop_req	uint8_t	Q0	-	Motor stop flag	
g_s2_cnt	int16_t	Q0	-	Counter	
s_state_transition_table [MTR_SIZE_EVENT] [MTR_SIZE_STATE]	static uint8_t	Q0	-	Macro array for state transition	
s_action_table [MTR_SIZE_EVENT] [MTR_SIZE_STATE]	static mtr_action_t	Q0	-	Function array for state transition	

Table 3-33 List of Variables in “rm_mtr_carrier.c”

Variable	Type	Qn	PU	Content	Remarks
g_st_foc_common	st_mtr_common_foc_t	-	-	Current Control Structure	Structure

3.4 List of Sensorless Vector Control Software Structures

A list of structures used in this control program is provided below. Structures that are not used have been omitted.

Table 3-34 List of Variables in "r_mtr_parameter.h" / Structure: "st_mtr_parameter_t"

Variable	Type	Qn	PU	Content	Remarks
u2_mtr_pp	uint16_t	Q0	-	Number of pole pairs	
s4_mtr_r	int32_t	Q31	Resistance (voltage/current)	Resistance [PU]	
s4_mtr_ld	int32_t	Q32	Inductance (resistance/angular frequency)	d-axis inductance [PU]	
s4_mtr_lq	int32_t	Q32	Inductance (resistance/angular frequency)	q-axis inductance [PU]	
s4_mtr_ke	int32_t	Q28	BEMF constant (voltage/angular frequency)	BEMF constant [PU]	
s4_mtr_j	int32_t	Q25	Inertia (BEMF constant × current × (number of pole pairs/angular frequency) ^2)	Inertia [PU]	

Table 3-35 List of Variables in "rm_mtr_ctrl_gain.h" / Structure: "st_mtr_design_parameter_t"

Variable	Type	Qn	PU	Content	Remarks
st_macro	st_mtr_design_parameter_macro_t	-	-	Control parameter structure	
st_com	st_mtr_design_parameter_com_t	-	-	COM Variable Structure	
f4_i_repro_cof_hz	float	-	-	LPF cut-off frequency for current reproduction [Hz]	
f4_r	float	-	-	Resistance [Ω]	
f4_ld	float	-	-	d-axis inductance [H/rad]	
f4_lq	float	-	-	q-axis inductance [H/rad]	
f4_ke	float	-	-	BEMF constant [Vs/m]	
f4_j	float	-	-	Rotor inertia [$\text{kgm}^2/\text{rad}^2$]	
f4_ol_ref_id	float	-	-	Open loop reference Id [A]	
f4_ol2cl_speed	float	-	-	Switching speed from open loop [rpm]	
f4_ramp_limit_speed	float	-	-	Limit of acceleration [rpm/ms] (mechanical)	
f4_do_hpf_cof_hz	float	-	-	Disturbance suppression HPF cutoff frequency[Hz]	
f4_d0_div_pm	float	-	-	Static friction coefficient/(number of pole pairs × BEMF constant) [$\text{kg} \cdot \text{m}^2/(\text{rad} \cdot \text{s}^2)$]	
f4_d1_div_p2m	float	-	-	Kinetic friction coefficient/((number of pole pairs)^2 × BEMF constant) [$\text{kg} \cdot \text{m}^2/(\text{rad}^2 \cdot \text{s})$]	
f4_damp_hpf_cof_hz	float	-	-	Damping control HPF cutoff frequency [Hz]	
f4_damp_zeta	float	-	-	Damping control damping coefficient	
f4_pherr_lpf_cof_hz	float	-	-	Phase error LPF cutoff frequency for switching control [Hz]	
f4_ol2cl_switch_time	float	-	-	Time[s] to switch open loop to Sensorless	
f4_fw_pi_nf_hz	float	-	-	PI controller natural frequency for Field-Weakening Control [Hz]	
f4_fw_speed_err_lpf_cof_hz	float	-	-	Speed error LPF cutoff frequency [Hz]	
u2_mtr_pp	uint16_t	Q0	-	Number of pole pairs	

Table 3-36 List of Variables in “rm_mtr_ctrl_gain.h” / Structure: “st_mtr_design_parameter_macro_t”

Variable	Type	Qn	PU	Content	Remarks
f4_dt	float	-	-	Control period [sec]	
f4_dt_speed	float	-	-	Control period for speed loop [sec]	
f4_pu_sf_afreq	float	-	-	Frequency scale factor	
f4_i_lim_vd	float	-	-	Limit on the current PI integral term of vd	
f4_i_lim_vq	float	-	-	Limit on the current PI integral term of vq	
f4_speed_lim_rad	float	-	-	Speed limit exceeded [rad/s]	
u1_q_current	uint8_t	Q0	-	Q-format of current	
u1_q_acr_kp	uint8_t	Q0	-	Q-format of Q-axis current PI proportional gain	
u1_q_acr_kidt	uint8_t	Q0	-	Q-format of Q-axis current PI $k_i \times dt$	
u1_q_acr_deadband	uint8_t	Q0	-	Q-format of current PI deadband	
u1_q_asr_kp	uint8_t	Q0	-	Q-format of Speed current PI proportional gain	
u1_q_asr_kidt	uint8_t	Q0	-	Q-format of Speed current PI $k_i \times dt$	
u1_q_asr_deadband	uint8_t	Q0	-	Q-format of speed PI deadband	
u1_q_pll_kp	uint8_t	Q0	-	Q-format of D-axis current PI proportional gain	
u1_q_pll_kidt	uint8_t	Q0	-	Q-format of D-axis current PI $k_i \times dt$	
u1_q_pll_deadband	uint8_t	Q0	-	Q-format of PLL deadband	
u1_q_acr_lpf_k	uint8_t	Q0	-	Q-format of Current LPF gain	
u1_q_asr_lpf_k	uint8_t	Q0	-	Q-format of Speed LPF gain	
u1_q_i_repro_lpf_k	uint8_t	Q0	-	Q-format of LPF gain for current reproduction	
u1_q_do_lpf_k	uint8_t	Q0	-	Q-format of disturbance suppression LPF	
u1_q_do_hpf_k	uint8_t	Q0	-	Q-format of disturbance suppression HPF	
u1_q_j_div_tc_p2m	uint8_t	Q0	-	Q-format of Inertia / (time constant of LPF $\times$ (number of pole pairs) $^2 \times$ BEMF constant)	
u1_q_damp_k	uint8_t	Q0	-	Q-format of Damping control gain	
u1_q_damp_hpf_k	uint8_t	Q0	-	Q-format of Damping control HPF	
u1_q_pherr_lpf_k	uint8_t	Q0	-	Q-format of Phase error LPF gain	
u1_q_ol2cl_current_k1	uint8_t	Q0	-	Q-format of Open loop to Closed loop switch Control gain	
u1_q_coef_mtpa_current	uint8_t	Q0	-	Q-format of (Id-Iq) / k_e calculation	
u1_q_fw_lpf_k	uint8_t	Q0	-	Q-format of LPF for Field-Weakening Control	
u1_q_fw_kp	uint8_t	Q0	-	Q-format of D-axis FW PI proportional gain	
u1_q_fw_kidt	uint8_t	Q0	-	Q-format of D-axis FW PI $k_i \times dt$	
u1_q_d1_div_p2m	uint8_t	Q0	-	Q-format of Primary friction coefficient	
u1_q_pi_run_acr_err	uint8_t	Q0	-	Q-format of ACR PI error	
u1_q_pi_run_asr_err	uint8_t	Q0	-	Q-format of ASR PI error	
u1_q_pi_run_pll_err	uint8_t	Q0	-	Q-format of PLL PI error	
u1_q_afreq	uint8_t	Q0	-	Q-format of angular frequency	

Table 3-37 List of Variables in "rm_mtr_ctrl_gain.h" / Structure: "st_mtr_design_parameter_com_t"

Variable	Type	Qn	PU	Content	Remarks
f4_acr_nf_hz	float	-	-	Current PI control natural frequency [Hz]	
f4_acr_lpf_cof_hz	float	-	-	Current LPF cut-off frequency [Hz]	
f4_acr_deadband_lsb	float	-	-	Deadband of current PI calculation [LSB]	
f4_asr_nf_hz	float	-	-	Speed PI control natural frequency [Hz]	
f4_asr_lpf_cof_hz	float	-	-	Speed LPF cutoff frequency [Hz]	
f4_asr_deadband_lsb	float	-	-	Deadband of speed PI calculation [LSB]	
f4_asr_ki_aug	float	-	-	Augmentation rate for integral part of ASR	
f4_pll_nf_hz	float	-	-	PLL natural frequency [Hz]	
f4_pll_deadband_lsb	float	-	-	Deadband of speed estimator PI calculation [LSB]	

Table 3-38 List of Variables in "rm_mtr_ctrl_gain.h" / Structure: "st_mtr_ctrl_gain_t"

Variable	Type	Qn	PU	Content	Remarks
st_pi_acr_id	st_mtr_pi_t	-	-	d-axis current control structure	
st_pi_acr_iq	st_mtr_pi_t	-	-	q-axis current control structure	
st_pi_asr	st_mtr_pi_t	-	-	Speed control structure	
s4_k_lpf_in_asr	int32_t	Q31	-	Speed LPF input coefficient	
s4_k_lpf_in_acr	int32_t	Q31	-	Current LPF input coefficient	
s4_k_do_j_div_tc_p2m	int32_t	Q31	-	(Inertia) / (time constant of LPF × (number of pole pairs) ^2 × BEMF constant)	
s4_k_lpf_in_do	int32_t	Q31	-	Disturbance suppression LPF input coefficient	
s4_k_hpf_in_do	int32_t	Q31	-	Disturbance suppression HPF coefficient	
s4_coef_d0_div_pm	int32_t	Q28	-	Static friction coefficient / (number of pole pairs × BEMF constant)	
s4_coef_d1_div_p2m	int32_t	Q37	-	Kinetic friction coefficient / ((number of pole pairs) ^2 × BEMF constant)	
s4_k_lpf_in_i_repro	int32_t	Q31	-	LPF input coefficient for current reproduction	
st_pi_pll	st_mtr_pi_t	-	-	PLL control structure	
s4_k_damp	int32_t	Q28	Speed/ Voltage	Damping control gain	
s4_k_hpf_in_damp	int32_t	Q31	-	Damping control HPF coefficient	
s4_k0_ol2cl_theta2current	int32_t	Q28	-	Zero order component of the angle-to-current conversion factor	
s4_k1_ol2cl_theta2current	int32_t	Q42	-	First-order component of the angle-to-current conversion factor	
s4_k_lpf_in_pherr	int32_t	Q31	-	Phase error LPF input coefficient	
s4_coef_mtpa_current	int32_t	-	-	MTPA current coefficient (Id - Iq) / ke	
s4_kp_fw	int32_t	-	-	FW PI proportional gain	
s4_kidt_fw	int32_t	-	-	FW PI ki * dt	
s4_k_lpf_in_fw	int32_t	-	-	FW LPF input coefficient	
s4_lim_asr_err	int32_t	Q28	Angular frequency	Speed PI control maximum error limit [PU(rad/s)]	

Table 3-39 List of Variables in "RM_DSP_RA2_GCC_32BIT.h" / Structure: "st_sincos28"

Variable	Type	Qn	PU	Content	Remarks
sin	int32_t	Q28	-	Sine (FIX28)	
cos	int32_t	Q28	-	Cosine (FIX28)	
angle_rad	int32_t	Q28	1	Angle [rad] (FIX28)	

Table 3-40 List of Variables in "RM_DSP_RA2_GCC_32BIT.h" / Structure: "st_coordinate28"

Variable	Type	Qn	PU	Content	Remarks
ph_u	int32_t	Q28	-	U-phase (FIX28)	
ph_v	int32_t	Q28	-	V-phase (FIX28)	
ph_w	int32_t	Q28	-	W-phase (FIX28)	
ax_d	int32_t	Q28	-	d-axis (FIX28)	
ax_q	int32_t	Q28	-	q-axis (FIX28)	
st_angle	st_sincos28	-	-	Angle structure	

(FIX n) is a variable representation with a fixed minority and is defined as follows.

1 bit sign	31-n bit Integer part	n bit Decimal part
---------------	--------------------------	-----------------------

Table 3-41 List of Variables in "rm_mtr_driver.h" / Structure: "st_mtr_ctrl_input_t"

Variable	Type	Qn	PU	Content	Remarks
s1_ref_direction	int8_t	Q0	-	Rotation direction command value	
u2_cnt_offset_calc	uint16_t	Q0	-	Offset detection time	
s4_ref_speed	int32_t	Q28	Angular frequency	Reference rotational speed [PU]	Electric angle
s4_lim_accel	int32_t	Q28	Angular frequency	Limit of acceleration [PU]	Electric angle
s4_max_speed	int32_t	Q28	Angular frequency	Maximum speed [PU]	Electric angle
s4_th_speed_cl2ol	int32_t	Q28	Angular frequency	Switching speed from sensorless to open loop [PU]	Electric angle
s4_th_speed_ol2cl	int32_t	Q28	Angular frequency	Switching speed from open loop to sensorless [PU]	Electric angle
s4_ref_id_ol	int32_t	Q28	Current	Open loop d-axis reference current [PU]	
s4_cnt_wait_draw_in	int32_t	Q0	-	Draw-in wait time count value	
s4_coef_init_intg	int32_t	Q28	Current	ASR integral term initial value during sensorless transition	
s4_lim_current	int32_t	Q13	Current	Limit value for current rise [PU/ms]	
s4_mod_3ph2ph_speed_rad	int32_t	Q0	-	Switching speed from 3-phase modulation to 2-phase modulation [rpm]	[Used in 1-shunt mode and 2-phase modulation]
s4_mod_2ph3ph_speed_rad	int32_t	Q0	-	Switching speed from 2-phase modulation to 3-phase modulation [rpm]	
s4_ad_point_a_adj_cnt_3ph	int32_t	Q0	-	Adjustment value of A/D conversion timing A (3-phase modulation)	
s4_ad_point_b_adj_cnt_3ph	int32_t	Q0	-	Adjustment value of A/D conversion timing B (3-phase modulation)	
s4_ad_point_a_adj_cnt_2ph	int32_t	Q0	-	Adjustment value of A/D conversion timing A (2-phase modulation)	
s4_ad_point_b_adj_cnt_2ph	int32_t	Q0	-	Adjustment value of A/D conversion timing B (2-phase modulation)	
s2_ad_point_a_adj_cnt	int16_t	Q0	-	Adjustment for A/D delay counts for A point	[Used in 1-shunt mode]
s2_ad_point_b_adj_cnt	int16_t	Q0	-	Adjustment for A/D delay counts for B point	
s2_mod_2ph_bot_change_cnt	int16_t	Q0	-	2-phase modulation transition counter	[Used in 2-phase modulation]
s4_speed_limit_rate	int32_t	Q0	-	Damping control speed limit	[Open loop damping control]
st_motor	st_mtr_parameter_t	-	-	Structure for motor parameter	Structure

Table 3-42 List of Variables in "rm_mtr_statemachine.h" / Structure: "st_mtr_statemachine_t"

Variable	Type	Qn	PU	Content	Remarks
u1_status	uint8_t	Q0	-	Motor status	
u1_status_next	uint8_t	Q0	-	Next motor status	
u2_status_error	uint16_t	Q0	-	Error status	
u1_crnt_event	uint8_t	Q0	-	Execution event	

Table 3-43 List of Variables in "rm_mtr_est_phase_err.h" / Structure: "st_mtr_estimate_phe_t"

Variable	Type	Qn	PU	Content	Remarks
s4_est_ed	int32_t	Q28	Voltage	d-axis induced voltage	
s4_est_eq	int32_t	Q28	Voltage	q-axis induced voltage	
s4_est_e	int32_t	Q28	Voltage	Induced voltage $\sqrt{ed^2+eq^2}$	
s4_coef_reci_ke	int32_t	Q28	1/BEMF constant	Reciprocal of BEMF constant	

Table 3-44 List of Variables in "rm_mtr_est_phase_err.h" / Structure: "st_mtr_est_vect_t"

Variable	Type	Qn	PU	Content	Remarks
s4_ref_vd	int32_t	Q28	Voltage	vd	
s4_ref_vq	int32_t	Q28	Voltage	vq	
s4_est_r_id	int32_t	Q28	Voltage	$R \times id$	
s4_est_r_iq	int32_t	Q28	Voltage	$R \times iq$	
s4_est_speed_ld_id	int32_t	Q28	Voltage	$Speed \times Ld \times id$	
s4_est_speed_lq_iq	int32_t	Q28	Voltage	$Speed \times Lq \times iq$	

Table 3-45 List of Variables in "rm_mtr_common.h" / Structure: "st_mtr_lpf1_t"

Variable	Type	Qn	PU	Content	Remarks
s4_in_k	int32_t	Current : Q31 Speed : Q31 Phase : Q31	-	LPF input gain	
s4_out_k	int32_t	Current : Q31 Speed : Q31 Phase : Q31	-	LPF previous gain	
s4_pre_out	int32_t	Current : Q28 Speed : Q28 Phase : Q28	Current : Current Speed : Angular frequency Phase : Angle	Previous output value	

Table 3-46 List of Variables in "rm_mtr_common.h" / Structure: "st_mtr_hpf1_t"

Variable	Type	Qn	PU	Content	Remarks
s4_k	int32_t	Q31	-	HPF input gain	
s4_pre_in	int32_t	Q28	-	Previous input value	
s4_pre_out	int32_t	Q28	-	Previous output value	
u1_q_k	uint8_t	Q0	-	HPF input gain Q value	

Table 3-47 List of Variables in "rm_mtr_foc_less_speed.h" / Structure: "st_mtr_pi_t"

Variable	Type	Qn	PU	Content	Remarks
s4_kp	int32_t	Speed : Q29 Current : Q32 PLL : Q33	Current : Resistance Speed : Current/angular frequency PLL : Angular frequency/angle	Proportional gain	
s4_kidt	int32_t	Speed : Q38 [Used in 1-shunt mode] Current : Q36 PLL : Q40 [Used in 3-shunt mode] Current : Q37 PLL : Q41	Current : Resistance Speed : Current/angular frequency PLL : Angular frequency/angle	Integral gain x control period	
s4_max_intg	int32_t	Speed : Q22 [Used in 1-shunt mode] Current : Q26 PLL : Q20 [Used in 3-shunt mode] Current : Q25 PLL : Q19	Current : Current Speed : Angular frequency PLL : Angle	Integral upper limit	
s4_min_intg	int32_t	Speed : Q22 [Used in 1-shunt mode] Current : Q26 PLL : Q20 [Used in 3-shunt mode] Current : Q25 PLL : Q19	Current : Current Speed : Angular frequency PLL : Angle	Integral lower limit	
s4_decimal	int32_t	Speed : Q22 [Used in 1-shunt mode] Current : Q26 PLL : Q20 [Used in 3-shunt mode] Current : Q25 PLL : Q19	Current : Current Speed : Angular frequency PLL : Angle	Stored decimal part	

Table 3-48 List of Variables in "rm_mtr_foc_less_speed.h" / Structure: "st_mtr_acr_t"

Variable	Type	Qn	PU	Content	Remarks
u1_flg_err_zero	uint8_t	Q0	-	Current error zero flag 0 : No 2-phase cross compensation 1 : With 2-phase cross compensation	[Used in 1-shunt mode] Resets current error to zero during 2-phase cross compensation
s4_ref_id	int32_t	Q28	Current	d-axis reference current	
s4_ref_iq	int32_t	Q28	Current	q-axis reference current	
s4_ref_id_ol	int32_t	Q28	Current	Open loop d-axis current command value	
s4_lim_ramp_current	int32_t	Q28	Current	Limit value for current rise [PU/ms]	
s4_ref_vd_pre	int32_t	Q28	Voltage	Previous d-axis reference voltage	
s4_ref_vq_pre	int32_t	Q28	Voltage	Previous q-axis reference voltage	
s4_ref_vd_buffer	int32_t	Q28	Voltage	d-axis voltage buffer value	
s4_ref_vq_buffer	int32_t	Q28	Voltage	q-axis voltage buffer value	
st_ad_iq_lpf	st_mtr_lpf1_t	-	-	q-axis current LPF structure	Structure
st_ad_id_lpf	st_mtr_lpf1_t	-	-	d-axis current LPF structure	
st_pi_id	st_mtr_pi_t	-	-	d-axis current PI structure	
st_pi_iq	st_mtr_pi_t	-	-	q-axis current PI structure	

Table 3-49 List of Variables in "rm_mtr_foc_less_speed.h" / Structure: "st_mtr_pll_t"

Variable	Type	Qn	PU	Content	Remarks
s4_est_speed	int32_t	Q28	Angular frequency	Speed detection value	
st_pi	st_mtr_pi_t	-	-	PI structure	Structure

Table 3-50 List of Variables in "rm_mtr_foc_less_speed.h" / Structure: "st_mtr_s4_comp_t"

Variable	Type	Qn	PU	Content	Remarks
s4_comp_vu	int32_t	Q28	Voltage	U-phase voltage compensation value	
s4_comp_vv	int32_t	Q28	Voltage	V-phase voltage compensation value	
s4_comp_vw	int32_t	Q28	Voltage	W-phase voltage compensation value	

Table 3-51 List of Variables in "rm_mtr_foc_less_speed.h" / Structure: "st_mtr_asr_t"

Variable	Type	Qn	PU	Content	Remarks
s1_ref_dir	int8_t	-	-	Direction of rotation command	1 : CW -1 : CCW
s4_ref_speed	int32_t	Q28	Angular frequency	Reference speed	
s4_ref_speed_ctrl	int32_t	Q28	Angular frequency	Reference speed for speed PI control [PU(rad/s)]	
s4_speed	int32_t	Q28	Angular frequency	Speed operation value [PU(rad/s)]	
s4_lim_ramp_speed	int32_t	Q28	Angular frequency	Limit of acceleration	
s4_cnt_ramp_deci_sample	int32_t	Q0	Angular frequency	Number of decimation of acceleration limit value	
s4_max_speed	int32_t	Q28	Angular frequency	Maximum speed	
s4_init_intg	int32_t	Q28	Current	Integral term initial value during sensorless switching	
s4_th_cl2ol_speed	int32_t	Q28	Angular frequency	Switching speed from closed loop to open loop	
s4_th_ol2cl_speed	int32_t	Q28	Angular frequency	Switching speed from open loop to closed loop	
s4_coef_d0_div_pm	int32_t	Q28	-	Static friction coefficient/(number of pole pairs × BEMF constant)	
s4_coef_d1_div_p2m	int32_t	Q37	-	Kinetic friction coefficient/((number of pole pairs)^2 × BEMF constant)	
u2_cnt_cl2ol_judge_wait	uint16_t	Q0	-	Waiting counter for start to judge closed-loop to open-loop	
st_pi	st_mtr_pi_t	-	-	Speed PI structure	Structure
s4_lim_asr_err	int32_t	-	-	Limit of ASR error	
st_lpf	st_mtr_lpf1_t	-	-	Speed LPF structure	Structure
s4_ref_speed_damp_ctrl	int32_t	Q28	Angular frequency	Reference speed for Open loop Damping Control	

Table 3-52 List of Variables in "rm_mtr_foc_less_speed.h / Structure:" st_mtr_mod_t"

Variable	Type	Qn	PU	Content	Remarks
u1_flg_uvw_min_pattern	uint8_t	-	-	Phase current minimum pattern	
u1_flg_uvw_min_pattern_pre	uint8_t	-	-	Previous value of minimum phase current pattern	
s2_cnt_2ph_bot_change	int16_t	-	-	2-phase modulation transition counter	[Used in 2-phase modulation]
s4_v_diff_sum	int32_t	-	-	Integration of the middle to minimum 2-phase voltage difference	
s4_mod_u	int32_t	Q28	-	U-phase modulation factor	
s4_mod_v	int32_t	Q28	-	V-phase modulation factor	
s4_mod_w	int32_t	Q28	-	W-phase modulation factor	
s4_ad_reci_vdc	int32_t	Q28	1/voltage	Inverse of voltage	
s4_lim_vout	int32_t	Q28	Voltage	Voltage limit	

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 3-53 List of Variables in "rm_mtr_foc_less_speed.h / Structure:"st_mtr_tscs_t" [Used in 3-shunt mode]

Variable	Type	Qn	PU	Content	Remarks
u4_offset_idc_adc	uint32_t	Q0	-	DC link current offset value	
u4_offset_calc_cnt	uint32_t	Q0	-	Offset current measurement count	
u4_offset_sample_cnt	uint32_t	Q0	-	Offset current measurement sample count	
s4_offset_iu	int32_t	Q0	-	U-phase current offset value	
s4_offset_iv	int32_t	Q0	-	V-phase current offset value	
s4_offset_iw	int32_t	Q0	-	W-phase current offset value	
s2_duty_u	int16_t	Q0	-	U-phase duty (PWM register setting)	
s2_duty_v	int16_t	Q0	-	V-phase duty (PWM register setting)	
s2_duty_w	int16_t	Q0	-	W-phase duty (PWM register setting)	
u2_ad_current_uph	uint16_t	Q0	-	U-phase current A/D conversion value	
u2_ad_current_vph	uint16_t	Q0	-	V-phase current A/D conversion value	
u2_ad_current_wph	uint16_t	Q0	-	W-phase current A/D conversion value	

Table 3-54 List of Variables in "rm_mtr_foc_less_speed.h / Structure:"st_mtr_sscs_t" [Used in 1-shunt mode] (1/2)

Variable	Type	Qn	PU	Content	Remarks
u4_offset_calc_cnt	uint32_t	Q0	-	Offset current measurement count	
u4_offset_sample_cnt	uint32_t	Q0	-	Offset current measurement sample count	
u1_drv_pattern_get	uint8_t	Q0	-	PWM drive pattern to get	
u1_drv_pattern_set	uint8_t	Q0	-	PWM drive pattern to set	
u1_drv_pattern_put	uint8_t	Q0	-	PWM drive pattern to put	
u1_flag_mod_2ph	uint8_t	Q0	-	2-phase modulation flag	
s2_duty_max	int16_t	Q0	-	Maximum duty value	
s2_duty_mid	int16_t	Q0	-	Middle duty value	
s2_duty_min	int16_t	Q0	-	Minimum duty value	
s2_duty_u	int16_t	Q0	-	U-phase duty value	
s2_duty_v	int16_t	Q0	-	V-phase duty value	
s2_duty_w	int16_t	Q0	-	W-phase duty value	
s2_duty_max_adj	int16_t	Q0	-	Maximum duty adjustment value	
s2_duty_mid_adj	int16_t	Q0	-	Middle duty adjustment value	
s2_duty_min_adj	int16_t	Q0	-	Minimum duty adjustment value	
s2_duty_max_adj_comp	int16_t	Q0	-	Maximum duty adjustment compensation value	
s2_duty_mid_adj_comp	int16_t	Q0	-	Middle duty adjustment compensation value	
s2_duty_min_adj_comp	int16_t	Q0	-	Minimum duty adjustment compensation value	
s2_duty_u_adj	int16_t	Q0	-	U-phase duty adjustment value	
s2_duty_v_adj	int16_t	Q0	-	V-phase duty adjustment value	
s2_duty_w_adj	int16_t	Q0	-	W-phase duty adjustment value	

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 3-55 List of Variables in "rm_mtr_foc_less_speed.h / Structure:"st_mtr_sscs_t" [Used in 1-shunt mode] (2/2)

Variable	Type	Qn	PU	Content	Remarks
s2_duty_u_adj	int16_t	Q0	-	U-phase duty adjustment compensation value	
s2_duty_v_adj	int16_t	Q0	-	V-phase duty adjustment compensation value	
s2_duty_w_adj	int16_t	Q0	-	W-phase duty adjustment compensation value	
s2_ad_point_a_cnt	int16_t	Q0	-	U-phase duty adjustment compensation value	
s2_ad_point_b_cnt	int16_t	Q0	-	V-phase duty adjustment compensation value	
s4_mod_3ph2ph_speed_rad	int32_t	Q0	-	Switching speed from 3-phase modulation to 2-phase modulation	[Used in 1-shunt mode and 2-phase modulation]
s4_mod_2ph3ph_speed_rad	int32_t	Q0	-	Switching speed from 2-phase modulation to 3-phase modulation	
s2_ad_point_a_adj_cnt_2ph	int16_t	Q0	-	Adjustment value of A/D conversion point A count (2-phase modulation)	
s2_ad_point_b_adj_cnt_2ph	int16_t	Q0	-	Adjustment value of A/D conversion point B count (2-phase modulation)	
s2_ad_point_a_adj_cnt_3ph	int16_t	Q0	-	Adjustment value of A/D conversion point A count (3-phase modulation)	
s2_ad_point_b_adj_cnt_3ph	int16_t	Q0	-	Adjustment value of A/D conversion point B count (3-phase modulation)	
s2_ad_point_a_adj_cnt	int16_t	Q0	-	A/D conversion point A count adjustment value	
s2_ad_point_b_adj_cnt	int16_t	Q0	-	A/D conversion point B count adjustment value	
s2_ad_ss_a	int16_t	Q0	-	Point A A/D conversion result	
s2_ad_ss_b	int16_t	Q0	-	Point A A/D conversion result	
s4_offset_ia	int32_t	Q0	-	Point A Current offset value	
s4_offset_ib	int32_t	Q0	-	Point B current offset value	
u2_crnt_ad_a	uint16_t	Q0	-	A point current AD conversion value	
u2_crnt_ad_b	uint16_t	Q0	-	B point current AD conversion value	

Table 3-56 List of Variables in "rm_mtr_foc_less_speed.h / Structure:"st_mtr_common_foc_t"

Variable	Type	Qn	PU	Content	Remarks
u1_state_drive	uint8_t	-	-	Drive mode status	0x00: Stop 0x01: Stop driving 0x81: Motor driving 0x82: Initialize driving 0x83: Start driving 0x84: Offset is being removed
u1_state_ctrl	uint8_t	-	-	Control status	0x00: Motor control none 0x01: Open loop Draw-in 0x02: Force sync control 0x03: Open loop to closed loop control 0x04: Closed loop to open loop control 0x05: Open loop speed control 0x80: Open loop to closed loop Switch control 0x81: Closed loop speed control
s1_direction	int8_t	-	-	Current direction of rotation	1 : CW -1 : CCW
s4_ad_vdc	int32_t	Q28	Voltage	Power source voltage	
s4_phase_err_rad	int32_t	Q28	Angle	Phase error[rad]	
st_foc_carrier	st_mtr_foc_carrier_t	-	-	Structure for carrier control	Structure
st_foc_speed	st_foc_speed_t	-	-	Structure for speed control	Structure

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 3-57 List of Variables in "rm_mtr_foc_less_speed.h / Structure:" st_mtr_foc_carrier_t"

Variable	Type	Qn	PU	Content	Remarks
u1_flg_charge_cap	uint8_t	-	-	Power source charged flag	0: Unreached 1: Reached
u1_flg_offset_end	uint8_t	-	-	Offset calculate end flag	
u2_error_status	uint16_t	-	-	Error status	0x0000: No error 0x0001: Overcurrent error (hardware) 0x0002: Overvoltage error 0x0004: Rotational speed error 0x0008: Hall timeout error 0x0010: BEMF timeout error 0x0020: Hall pattern error 0x0040: BEMF pattern error 0x0080: Undervoltage error 0x0100: Overcurrent error (software) 0x0400: Charge capacitor timeout error 0xFFFF: Undefined error
st_stm	st_mtr_state_machine_t	-	-	Structure for state machine	
st_ad_i	st_coordinate28	-	-	ADC current coordinate system	
st_ref_v	st_coordinate28	-	-	Reference voltage coordinate system	
st_ref_i	st_coordinate28	-	-	Reference current coordinate system	
st_motor	st_mtr_parameter_t	-	-	Structure for motor parameter	
st_phe	st_mtr_estimate_phe_t	-	-	Structure for phase error estimate	
st_tscs	st_mtr_tscs_t	-	-	Structure for 3-phase current detection	[Used in 3-shunt mode]
st_sscs	st_mtr_sscs_t	-	-	Structure for 1-shunt resistor current detection	[Used in 1-shunt mode]
st_acr	st_mtr_acr_t	-	-	Structure for ACR	Current PI control
st_mod	st_mtr_mod_t	-	-	Structure for modulation	
st_pll	st_mtr_pll_t	-	-	Structure for PLL control	
st_i_repro	st_coordinate28	-	-	Reproduced current coordinate system	
st_iq_repro	st_mtr_lpf1_t	-	-	q-axis current LPF for current reproduction	
st_id_repro	st_mtr_lpf1_t	-	-	d-axis current LPF for current reproduction	

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 3-58 List of Variables in "rm_mtr_foc_less_speed.h / Structure:"st_mtr_foc_speed_t"

Variable	Type	Qn	PU	Content	Remarks
u1_state_ref_id	uint8_t	-	-	d-axis current command value generation status	0: d-axis current 0 1: d-axis current manual control 2: Field weakening control/maximum torque control
u1_state_ref_iq	uint8_t	-	-	q-axis current command value generation status	0: q-axis current 0 1: q-axis current manual control 2: Open loop to Closed loop switch Control 3: Speed PI output
u1_state_ref_speed	uint8_t	-	-	Speed command value generation status	0: Speed 0 1: Speed change
u2_cnt_draw_in_wait	uint16_t	Q0	-	Wait times of Id draw-in	
u2_cnt_draw_in_wait_buff	uint16_t	Q0	-	Wait times buffer of Id draw-in	
u2_cnt_draw_in_time_calc	uint16_t	Q0	-	Wait times counter of Id draw-in	
st_asr	st_mtr_asr_t	-	-	Structure for ASR	Speed PI control
st_ol2cl	st_mtr_ol2cl_t	-	-	Structure for Open loop to Closed loop switch Control	
st_phase_err_lpf	st_mtr_lpf1_t	-	-	Structure for phase error lpf	
s4_damp_speed	int32_t	Q28	Speed	Damping speed output	

Table 3-59 List of Variables in "rm_mtr_ol2cl_ctrl.h" / Structure:"st_mtr_damp_t"

Variable	Type	Qn	PU	Content	Remarks
s4_k	int32_t	Q28	Speed/Voltage	Damping control gain	
s4_speed_lim_rate	int32_t	Q28	Speed	Speed limit	
u1_q_damp_speed_calc	uint8_t	Q0	-	Damping control speed Q value	
u1_q_damp_speed_limit_calc	uint8_t	Q0	-	Damping control speed limit Q value	
st_hpf	st_mtr_hpf1_t	-	-	Structure for HPF	

Table 3-60 List of Variables in "rm_mtr_ol2cl_ctrl.h" / Structure:"st_mtr_ol2cl_t"

Variable	Type	Qn	PU	Content	Remarks
s4_lim_ramp_i	int32_t	Q28	Current	Current rise limit value during switching process [A/ms]	
s4_iq_ol2cl	int32_t	Q28	Current	q-axis current during switching process	
s4_k0_angle2current	int32_t	Q28	-	Reference current = $k_0 + k_1 \times (\text{phase error})$	
s4_k1_angle2current	int32_t	Q42	-	Reference current = $k_0 + k_1 \times (\text{phase error})$	
s4_th_phase_err_rad_cl	int32_t	Q28	Angle	Phase error during sensorless switching processing	
u2_cnt_switch	uint16_t	Q0	-	Switching time [cnt]	
u1_q_k1_angle2current	uint8_t	Q0	-	Q value for angular error of reference current	
u1_q_idelta_calc	uint8_t	Q0	-	Q value for calculating the q-axis component of current	
st_ph_err	st_sincos28	-	-	Structure for phase error angle	
st_damp	st_mtr_damp_t	-	-	Structure for damping control	

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 3-61 List of Variables in "r_mtr_ics.h" / Structure:" st_com_t"

Variable	Type	Qn	PU	Content	Remarks
st_com	st_mtr_design_parameter_com_t	-	-	Structure for COM parameters	
com_u1_direction	uint8_t	Q0	-	Direction of rotation	0 : CW 1 : CCW
com_f4_mtr_r	float	-	-	Resistance[Ω]	
com_f4_mtr_ld	float	-	-	d-axis inductance [H/rad]	
com_f4_mtr_lq	float	-	-	q-axis inductance [H/rad]	
com_f4_mtr_m	float			BEMF constant [Vs/rad]	
com_f4_mtr_j	float	-	-	Rotor inertia [kgm ² /rad ²]	
com_f4_mtr_d0	float	-	-	Coefficient of friction at rest[kgm ² /(rad · s ²)]	
com_f4_mtr_d1	float	-	-	Coefficient of kinetic friction [kgm ² /(rad ² · s)]	
com_u2_mtr_pp	uint16_t	Q0	-	Number of pole pairs	
com_u2_offset_calc_cnt	uint16_t	Q0	-	Current offset detection time	
com_f4_ref_speed_rpm	float	-	-	Reference rotational speed [rpm]	Mechanical angle
com_f4_ramp_limit_speed_rpm	float	-	-	Limit of acceleration [rpm/ms]	Mechanical angle
com_f4_max_speed_rpm	float	-	-	Maximum speed [rpm]	Mechanical angle
com_f4_cl2ol_speed_rpm	float	-	-	Switching speed from sensorless to open loop [rpm]	Mechanical angle
com_f4_ol2cl_speed_rpm	float	-	-	Switching speed from open loop to sensorless [rpm]	Mechanical angle
com_f4_ol_ref_id	float	-	-	Open loop d-axis command current [A]	
com_f4_draw_in_wait_time	float	-	-	Draw-in wait time count value [s]	
com_f4_init_asr_intg	float	-	-	ASR integral term initial value during sensorless transition	
com_f4_ramp_limit_current	float	-	-	Limit value for current rise [A/ms]	
com_f4_i_repro_cof_hz	float	-	-	LPF cut-off frequency for current reproduction [Hz]	[Used in 1-shunt mode]
com_s2_ad_point_a_adj_cnt	int16_t	Q0	-	Adjustment for A/D delay counts for A point	
com_s2_ad_point_b_adj_cnt	int16_t	Q0	-	Adjustment for A/D delay counts for B point	
com_f4_mod_3ph2ph_speed_rpm	float	-	-	Switching speed from 3-phase modulation to 2-phase modulation [rpm]	[Used in 1-shunt mode and 2-phase modulation]
com_f4_mod_2ph3ph_speed_rpm	float	-	-	Switching speed from 2-phase modulation to 3-phase modulation [rpm]	
com_s2_ad_point_a_adj_cnt_3ph	int16_t	Q0	-	Adjustment value of A/D conversion timing A (3-phase modulation)	
com_s2_ad_point_b_adj_cnt_3ph	int16_t	Q0	-	Adjustment value of A/D conversion timing B (3-phase modulation)	
com_s2_ad_point_a_adj_cnt_2ph	int16_t	Q0	-	Adjustment value of A/D conversion timing A (2-phase modulation)	
com_s2_ad_point_b_adj_cnt_2ph	int16_t	Q0	-	Adjustment value of A/D conversion timing B (2-phase modulation)	
com_s2_mod_2ph_bot_change_cnt	int16_t	Q0	-	Number of counts during 2-phase modulation transition	
com_f4_damp_hpf_cof_hz	float	-	-	HPF cutoff frequency for damping control [Hz]	[Open loop damping control]
com_f4_damp_zeta	float	-	-	Damping coefficient of damping control	
com_f4_damp_speed_limit_rate	float	-	-	Damping control speed limit	
com_f4_pherr_lpf_cof_hz	float	-	-	Phase error LPF cutoff frequency for Open loop to Closed loop switch Control	[Open loop to Closed loop switch Control]
com_f4_ol2cl_switch_time	float	-	-	Time[s] to switch open loop to sensorless	
com_s2_enable_write	int16_t	Q0	-	Variable to allow variable rewriting	

3.5 List of Sensorless Vector Control Software Macro Definitions

A list of macro definitions used in this control program is provided below.

Table 3-62 List of Macro Definitions in "r_mtr_config.h"

Macro	Definition value	Description	Remarks
IP_MCK	-	Select inverter board	
MP_R42BLD30L3_A_a1	-	Select motor parameters	
CP_R42BLD30L3_A_a1	-	Select control parameters	
SINGLE_SHUNT	0	1-shunt mode	
THREE_SHUNT	1	3-shunt mode	
CURRENT_SENS_METHOD	SINGLE_SHUNT THREE_SHUNT	Current detection method ^(Note)	
USE_DEADTIME_COMP	0 : 1	Select deadtime compensation process (1: Enable 0: Disable)	Default setting 1
USE_SPEED_LPF	0 : 1	Select speed LPF (1: Enable 0: Disable)	Default setting 1
USE_CURRENT_LPF_IQ	0 : 1	Select q-axis current LPF (1: Enable 0: Disable)	Default setting 0
USE_CURRENT_LPF_ID	0 : 1	Select d-axis current LPF (1: Enable 0: Disable)	Default setting 0
USE_MTPA_AND_FW	0 : 1	Select Field-Weakening control/maximum torque control (1: Enable 0: Disable)	Default setting 0
USE_OPENLOOP_DAMPING	0 : 1	Select Open loop damping Control (1: Enable 0: Disable)	Default setting 0
USE_OL2CL_CTRL	0 : 1	Select Open loop to Closed loop switch Control (1: Enable 0: Disable)	Default setting 1
USE_DISTURBANCE_SUPPRESSION	0 : 1	Select disturbance suppression function	Default setting 0
PS_IPD_SAL	0	Initial position detection for salient motor	Not implemented
PS_IPD_NON_SAL	1	Initial position detection for non-salient motor	Not implemented
PS_IPD_UNKNOWN	2	Initial position detection without saliency information	
PS_DRAW_IN	3	Draw-in	
PS_METHOD	PS_IPD_SAL PS_IPD_NON_SAL PS_IPD_UNKNOWN PS_DRAW_IN	preparation method for start-up	Default setting PS_DRAW_IN
CP_CARRIER_FREQ_20KHZ	0	Carrier frequency 20kHz	
CP_CARRIER_FREQ_40KHZ	1	Carrier frequency 40kHz	
CP_CARRIER_FREQ_16KHZ	2	Carrier frequency 16kHz	
CP_CARRIER_FREQ_SEL	CP_CARRIER_FREQ_20KHZ CP_CARRIER_FREQ_40KHZ CP_CARRIER_FREQ_16KHZ	Carrier frequency selects	Default setting CP_CARRIER_FREQ_20KHZ
MOD_3PH_SPWM	0	Sine wave modulation	Not implemented
MOD_3PH_TOW	1	Third harmonic calculation	
MOD_2PH_BOT	2	2-phase modulation	
MOD_METHOD	MOD_3PH_SPWM MOD_3PH_TOW MOD_2PH_BOT	Sine wave modulation	Default setting MOD_3PH_TOW

Note: When switching this macro, set the parameters related to current detection in the config folder as appropriate.

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 3-63 List of Macro Definitions in "r_mtr_motor_parameter.h"

Macro	Definition value	Description	Remarks
MP_POLE_PAIRS	4	Number of pole pairs	R42BLD30L3_24V
MP_RESISTANCE	1.32f	Resistance [Ω]	R42BLD30L3_24V
MP_D_INDUCTANCE	0.0021f	d-axis inductance [H/rad]	R42BLD30L3_24V
MP_Q_INDUCTANCE	0.0021f	q-axis inductance [H/rad]	R42BLD30L3_24V
MP_BEMF_CONSTANT	0.0115f	BEMF constant [V·s/m]	R42BLD30L3_24V
MP_ROTOR_INERTIA	0.0000032f	Inertia [$\text{kgm}^2/\text{rad}^2$]	R42BLD30L3_24V
MP_FRICTION_0TH_ORDER	0.003583013f	Static friction coefficient [$\text{kgm}^2/(\text{rad} \cdot \text{s}^2)$]	R42BLD30L3_24V
MP_FRICTION_1ST_ORDER	0.000001850949f	Kinetic friction coefficient [$\text{kgm}^2/(\text{rad}^2 \cdot \text{s})$]	R42BLD30L3_24V
MP_RATED_CURRENT	1.13f	Nominal current [A]	R42BLD30L3_24V
MP_RATED_SPEED	4000	Rated speed [rpm]	R42BLD30L3_24V
STAR	0	Star wiring	Default
DELTA	1	Delta wiring	
MP_MOTOR_WIRE_CONNECTION	STAR	Selection of motor wiring connection	Star / Delta

Table 3-64 List of Macro Definitions in "r_mtr_control_parameter.h" (1/2)

Macro	Definition value	Description	Remarks
CP_PWM_TIMER_FREQ	32.0f	PWM timer frequency [MHz]	
CP_INTVAL_TIMER_FREQ	32.0f	Interval timer frequency [MHz]	
CP_CARRIER_FREQ	20.0f	PWM carrier frequency	
CP_TRX_TIMER_FREQ	32.0f	Timer frequency [MHz]	
CP_INT_DECIMATION	1[1 shunt mode] 0[3 shunt mode]	Interrupt thinning count	
CP_V_PHASE_LEAD_COEF	1.0f[1shunt mode] 1.5f[3shunt mode]	Voltage leading phase coefficient	
CP_SPEED_CTRL_PERIOD	0.001f	Speed control period [seconds]	
CP_AD_CONVERSION_TIME	1.1375f	A/D conversion time	
CP_AD_RINGING_WAIT_CNT	32	Current ringing wait time at AD conversion	
CP_AD_RINGING_WAIT_2PH_CNT	42	Current ringing wait time at AD conversion for 2-phase modulation	
CP_ACR_NF_HZ	500.0f	Current PI control natural frequency [Hz]	
CP_ASR_NF_HZ	16.00000f	Speed PI control natural frequency [Hz]	
CP_PLL_NF_HZ	80.00000f	PLL control natural frequency [Hz]	
CP_ACR_DEADBAND_LSB	0.05000f	Current PI deadband [LSB]	
CP_ASR_DEADBAND_LSB	0.05000f	Speed PI deadband [LSB]	
CP_PLL_DEADBAND_LSB	0.05000f	PLL control deadband [LSB]	
CP_ASR_LPF_COF_HZ	200.00000f	Speed LPF cutoff frequency [Hz]	
CP_ACR_LPF_COF_HZ	2000.00000f	Current LPF cutoff frequency [Hz]	
CP_I_REPRO_COF_HZ	2000.00000f	Cutoff frequency of LPF for current reproduction [Hz]	
CP_ASR_KI_AUG	8.00000f	Augmentation rate for integral part of ASR	
CP_MAX_SPEED_RPM	4000	Maximum speed (mechanical angle) [rpm]	
CP_SPEED_LIMIT_RPM	8000	Speed limit (mechanical angle) [rpm]	
CP_OC_LIMIT	2.26000f	Overcurrent limit value [A]	
CP_OL_REF_ID	1.13000f	d-axis current command value [A]	

Table 3-65 List of Macro Definitions in "r_mtr_control_parameter.h" (2/2)

Macro	Definition value	Description	Remarks
CP_DRAW_IN_WAIT_TIME	0.20000f	Draw-in wait time count value	
CP_INIT_ASR_INTEG	0.07641f	q-axis current PI integral term PI initial value [A]	
CP_RAMP_LIMIT_CURRENT	0.0121680f	Limit value for current rise [PU/ms]	
CP_MOD_3PH2PH_SPEED_RPM	1600.0000000f	Switching speed from 3-phase to 2-phase modulation [rpm]	[Used in 1-shunt mode and 2-phase modulation]
CP_MOD_2PH3PH_SPEED_RPM	1400.0000000f	Switching speed from 2-phase to 3-phase modulation [rpm]	
CP_OL2CL_SPEED_RPM	800.0000000f	Speed to start decreasing id [rpm]	
CP_CL2OL_SPEED_RPM	600.0000000f	Speed to start increasing id [rpm]	
CP_CL2OL_JUDGE_WAIT_TIME	12.5000000f	Waiting time to judge down to open loop [ms]	
CP_RAMP_LIMIT_SPEED_RPM	6.8223750f	Limit of acceleration [rpm/ms]	
CP_RAMP_SPEED_CNT_DECIMATION	0	Number of decimations out of acceleration limit	
CP_OFFSET_CALC_CNT	512	Current offset value calculation time [ms]	
CP_OFFSET_CALC_ST_WAIT_CNT	100	Offset calculation stable waiting time	
CP_AD_POINT_A_ADJ_CNT	32	Adjustment for A/D delay counts for A point	[Used in 1-shunt mode]
CP_AD_POINT_B_ADJ_CNT	32	Adjustment for A/D delay counts for B point	
CP_AD_POINT_A_ADJ_2PH_CNT	42	Adjustment for A/D delay counts for A point	[Used in 1-shunt mode and 2-phase modulation]
CP_AD_POINT_B_ADJ_2PH_CNT	42	Adjustment for A/D delay counts for B point	
CP_MOD_2PH_BOT_CHANGE_CNT	200	2-phase modulation phase switching judgment counter	[Used in 2-phase modulation]
CP_DAMP_HPF_COF_HZ	5.0000000f	HPF cutoff frequency for damping control [Hz]	[Open loop damping]
CP_DAMP_ZETA	1.0000000f	Damping control damping coefficient	
CP_DAMP_SPEED_LIMIT_RATE	0.2000000f	Damping control speed limit	
CP_PHASE_ERR_LPF_COF_HZ	10.0000000f	Phase error LPF cutoff frequency for Open loop to Closed loop switch control [Hz]	[Open loop to Closed loop switch Control]
CP_OL2CL_SWITCH_TIME	0.1729000f	Time[s] to switch open loop to sensorless	
CP_OL2CL_SWITCH_ANGLE_MIN	5.0000000f	Minimum angle for switching to open loop [deg]	

Table 3-66 List of Macro Definitions in “r_mtr_inverter_parameter.h”

Macro	Definition value	Description	Remarks
IP_DEADTIME	2.0f	Deadtime	
IP_CURRENT_RANGE	16.5f	Current scaling range [A]	
IP_VDC_RANGE	73.5f	Voltage scaling range [V]	
IP_INPUT_V	24.0f	Input voltage [V]	
IP_CURRENT_LIMIT	21.4f	Current limit value [A]	
IP_OVERVOLTAGE_LIMIT	48.0f	Overvoltage limit [V]	
IP_UNDERVOLTAGE_LIMIT	8.0f	Undervoltage limit [V]	
IP_PHASE_SHUNT_RESISTANCE	0.010f	Phase current Shunt Resistance [ohm]	
IP_PHASE_AMPLIFICATION_GAIN	20.000f	Phase current Amplification Gain	
IP_DC_SHUNT_RESISTANCE	0.010f	DC Link Shunt Resistance [ohm]	
IP_DC_AMPLIFICATION_GAIN	20.0f	DC Link Current Amplification Gain	
IP_BSC_CHARGE_CNT	150	Period of charging bootstrap capacitor	
IP_CHARGE_CAP_WAIT_CNT	300	Charge time of DC capacitor	

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 3-67 List of Macro Definitions in "r_mtr_scaling_parameter.h" (1/2)

Macro	Definition value	Description	Remarks
FP_SF_VOLTAGE	822226302	Voltage PU conversion value $((IP_VDC_RANGE \times PU_SF_VOLTAGE) \times (1 < MTR_Q_VOLTAGE))$	
FP_SF_CURRENT	3919632764	Current PU conversion value $((IP_CURRENT_RANGE \times PU_SF_CURRENT) \times (1 < MTR_Q_CURRENT))$	
FP_SF_CURRENT_AD	FP_SF_CURRENT >> MTR_ADC_12BIT	Current PU conversion value corrects of ADC	
PU_BASE_CURRENT_A	MP_RATED_CURRENT	Current standard value [A]	
PU_BASE_VOLTAGE_V	IP_INPUT_V	Voltage standard value [A]	
PU_BASE_FREQ_Hz	$MTR_TWOPI \times CP_MAX_SPEED_RPM \times MP_POLE_PAIRS / 60$	Frequency standard value [Hz]	
PU_BASE_ANGLE_ROT	1.0f	Angle standard value [rad]	
PU_SF_CURRENT	$1.0f / PU_BASE_CURRENT_A$	Current scale [1/A]	
PU_SF_VOLTAGE	$1.0f / PU_BASE_VOLTAGE_V$	Voltage scale [1/V]	
PU_SF_AFREQ	$1.0f / PU_BASE_FREQ_Hz$	Angular frequency scale [s/rad]	
PU_SF_ANGLE	$1.0f / PU_BASE_ANGLE_ROT$	Angle scale [1/rad]	
PU_SF_TIME	$PU_SF_ANGLE / PU_SF_AFREQ$	Time scale [1/s]	
PU_SF_RES	$PU_SF_VOLTAGE / PU_SF_CURRENT$	Resistance scale [1/Ω]	
PU_SF_IND	PU_SF_RES / PU_SF_AFREQ	Inductance scale [rad/H]	
PU_SF_BEMF_CONST	$PU_SF_VOLTAGE / PU_SF_AFREQ$	BEMF constant scale [rad/(V·s)]	
PU_SF_INERTIA	$PU_SF_BEMF_CONST \times PU_SF_CURRENT / (MP_POLE_PAIRS \times MP_POLE_PAIRS \times PU_SF_AFREQ \times PU_SF_AFREQ)$	Inertia scale [rad ² /kg·m ²]	
PU_SF_D1_DIV_P2M	$(PU_SF_CURRENT / PU_SF_AFREQ)$	Scale pf kinetic friction coefficient/((number of pole pairs) ² × BEMF constant) [A·s/rad]	
PU_SF_RPM_RAD	$1.0f / CP_MAX_SPEED_RPM$	Scale of conversion from [rpm] to [rad/s]	
PU_SF_RAD_RPM	CP_MAX_SPEED_RPM	Scale of conversion from [rad/s] to [rpm]	
PU_SF_ACR_KP	PU_SF_RES	Current PI proportional gain scale	
PU_SF_ACR_KIDT	PU_SF_RES	Current PI integral gain scale	
PU_SF_ASR_KP	$PU_SF_CURRENT / PU_SF_AFREQ$	Speed PI proportional gain scale	
PU_SF_ASR_KIDT	$PU_SF_CURRENT / PU_SF_AFREQ$	Speed PI integral gain scale	
PU_SF_PLL_KP	$PU_SF_AFREQ / PU_SF_ANGLE$	PLL proportional gain scale	
PU_SF_PLL_KIDT	$PU_SF_AFREQ / PU_SF_ANGLE$	PLL integral gain scale	
MTR_Q_ANGLE	28	Q-format of angle	
MTR_Q_CURRENT	28	Q-format of current	
MTR_Q_VOLTAGE	28	Q-format of voltage	
MTR_Q_VMOD	28	Q-format of PWM modulation factor	
MTR_Q_AFREQ	28	Q-format of angular frequency	
MTR_Q_CTRL_TIME	33[1 shunt] 34[3 shunt]	Q-format of FOC control cycle	
MTR_Q_CTRL_TIME_SPEED	30	Q-format of speed control cycle	
MTR_Q_RESISTANCE	32	Q-format of resistance	
MTR_Q_INDUCTANCE	33	Q-format of inductance	
MTR_Q_BEMF_CONST	28	Q-format of BEMF constant	
MTR_Q_INERTIA	25	Q-format of inertia	
MTR_Q_D1_DIV_P2M	37	Q-format of kinetic friction coefficient/((number of pole pairs) ² × BEMF constant)	
MTR_Q_RECIV	28	Q-format of inverse voltage	
MTR_Q_RECIM	30	Q-format of reciprocal of BEMF constant	

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 3-68 List of Macro Definitions in “r_mtr_scaling_parameter.h” (2/2)

Macro	Definition value	Description	Remarks
MTR_Q_ACR_KP	32	Q-format of speed PI proportional gain	
MTR_Q_ACR_KIDT	36[1 shunt] 37[3 shunt]	Q-format of speed PI integral gain×control period	
MTR_Q_ASR_KP	29	Q-format of current PI proportional gain	
MTR_Q_ASR_KIDT	37	Q-format of current PI integral gain×control period	
MTR_Q_PLL_KP	32	Q-format of PLL proportional gain	
MTR_Q_PLL_KIDT	39[1 shunt] 40[3 shunt]	Q-format of PLL integral gain×control period	
MTR_Q_SPEED_LPF_CO	31	Q-format of speed LPF gain	
MTR_Q_CURRENT_LPF_CO	31	Q-format of current LPF gain	
MTR_Q_CURRENT_REPO_LPF_CO	31	Q-format of current reproduction LPF gain	
MTR_Q_DAMP_K	28	Q-format of damping control gain	
MTR_Q_DAMP_HPF_CO	31	Q-format of damping control HPF gain	
MTR_Q_DAMP_SL_RATE	33	Q-format of damping control speed limit	
MTR_Q_PHERR_LPF_CO	31	Q-format of Phase error LPF gain	
MTR_Q_OL2CL_K1	40	Q-format of Open loop to Closed loop switch Control gain	
MTR_Q_FW_I	28	Q-format of ((q-axis inductance- d-axis inductance))/BEMF-const	
MTR_Q_FW_I_SQ	26	Q-format of $i_q * i_q^*$ ((q-axis inductance- d-axis inductance))/BEMF-const calculation	
MTR_Q_KE_DIV_2LDQ	31	Q-format of Flux / (2*(q-axis inductance- d-axis inductance))	
MTR_Q_FW_KP	26	Q-format of PI proportional gain for Field-Weakening Control	
MTR_Q_FW_KIDT	42[1 shunt] 43[3 shunt]	Q-format of PI (integral gain) * (sampling period) for Field-Weakening Control	
MTR_Q_FW_LPF_CO	31	Q-format of speed error LPF coefficient for Field-Weakening Control	
MTR_Q_DO_LPF_CO	31	Q-format of filter coefficients for disturbance suppression	
MTR_Q_J_DIV_TC_P2M	34	Q-format of kinetic friction coefficient/ ((number of pole pairs)^2 × BEMF constant)	
MTR_Q_V_PHASE_LEAD_COEF	31[1 shunt] 30[3 shunt]	Q-format of voltage leading phase coefficient	
MTR_Q_DIV_DSP	32	Q-format of DSP function division	
MTR_Q_SIN_COS_DSP	28	Q-format of trigonometric function of DSP function	
MTR_Q_DUTY	12	Q-format of PWM duty	
MTR_Q_PI_RUN_ACR_ERR	25[1 shunt] 24[3 shunt]	Q-format of ACR PI error	
MTR_Q_PI_RUN_PLL_ERR	21[1 shunt] 20[3 shunt]	Q-format of PLL PI error	
MTR_Q_PI_RUN_ACR_INTG	12	Q-format of ACR integration	
MTR_Q_PI_RUN_ASR_ERR	23	Q-format of ASR PI error	
MTR_Q_PI_RUN_ASR_INTG	12	Q-format of ASR integration	

Table 3-69 List of Macro Definitions in “mtr_main.h”

Macro	Definition value	Description	Remarks
MODE_INACTIVE	0x00	Inactive mode	
MODE_ACTIVE	0x01	Active mode	
MODE_ERROR	0x02	Error mode	
SIZE_STATE	3	Number of modes	

Table 3-70 List of Macro Definitions in “r_mtr_ics.h”

Macro	Definition value	Description	Remarks
TS_300US	-	ICS communication speed	
ICS_BRR	4	ICS bit rate register selection	
MTR_ICS_DECIMATION	2[1 shunt] 5[3 shunt]	Number of pixels skipped in ICS processing	
ICS_ADDR	0xFE00	Address of ICS	
ICS_INT_LEVEL	3	ICS interrupt level setting	
ICS_NUM	0x40	Data size of ICS communication	
ICS_MODE	0	ICS interrupt mode setting	

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 3-71 List of Macro Definitions in "r_mtr_ra2t1.h" (1/2)

Macro	Definition value	Description	Remarks
USE_PWMOPA	MTR_SET	Select of overcurrent use by PWMOPA	
USE_ADC_FAST_MODE	MTR_SET	CPU main clock [MHz]	
USE_TIME_MEASURE	MTR_SET	Interrupt processing carrier pixel skipping	
USE_DUTY_SHIFT	MTR_CLR	TRD interrupt skipping count	
USE_U_PHASE_CURRENT	MTR_CLR	PWM timer frequency [kHz]	
TRD_SKIPPING_COUNT	CP_INT_DECIMATION	TRD interrupt skipping count	
MTR_PWM_TIMER_FREQ	CP_PWM_TIMER_FREQ	PWM timer frequency [kHz]	
MTR_INTVAL_TIMER_FREQ	CP_INTVAL_TIMER_FREQ	Interval timer frequency [kHz]	
MTR_CARRIER_FREQ	CP_CARRIER_FREQ	Carrier interrupt frequency [kHz]	
MTR_DEADTIME	IP_DEADTIME	Deadtime [μs]	
MTR_DEADTIME_CNT	$(\text{int16_t})(\text{MTR_DEADTIME} * \text{MTR_PWM_TIMER_FREQ})$	Deadtime settings	
MTR_CARRIER_CNT	$(\text{uint16_t})(\text{MTR_PWM_TIMER_FREQ} * 1000 / \text{MTR_CARRIER_FREQ} * 0.5f)$	Carrier settings	
MTR_HALF_CARRIER_CNT	$(\text{uint16_t})(\text{MTR_CARRIER_CNT} * 0.5f)$	Carrier settings (intermediate value)	
MTR_HALF_DEADTIME_CNT	$(\text{uint16_t})(\text{MTR_DEADTIME_CNT} * 0.5f)$	Half of the dead time setting value	
MTR_DUTY_RANGE_CNT	$(\text{uint16_t})(\text{MTR_CARRIER_CNT} - \text{MTR_DEADTIME_CNT}) * 0.5f$	Count value of modulation rate range	
MTR_CURRENT_AD_CONV_TIME	$\text{CP_AD_CONVERSION_TIME} * 2.0f$	Time [μs] taken for 2-phase current A/D conversion	[Used in 3 shunt]
MTR_ADC_SH_CNT	44	Count value of AD sample hold value	
MTR_ADC_RINGING_CNT	CP_AD_RINGING_WAIT_CNT	Count value of AD ringing	
MTR_ADC_TS_CNT	$(\text{MTR_ADC_SH_CNT} + \text{MTR_ADC_RINGING_CNT} + 1) / 2$	Count value of AD conversion time	
MTR_VOLTAGE_LIMIT_OFFSET	$(\text{int32_t})(((\text{MTR_CURRENT_AD_CONV_TIME} + \text{MTR_DEADTIME} * 2) / (1000 / \text{MTR_CARRIER_FREQ})) * 0.5f * (1 << \text{MTR_Q_VOLTAGE}))$	Voltage offset limit [PU (V)]	
MTR_CENTER_AMPLITUDE_CNT	$(\text{MTR_CARRIER_CNT} + \text{MTR_ADC_TS_CNT} + \text{MTR_DEADTIME_CNT} + 1) / 2$	PWM timer center amplitude	[Used in 3 shunt and 2-phase modulation]
MTR_OFFSET_CENTER_AMPLITUDE_CNT	MTR_CENTER_AMPLITUDE_CNT	PWM timer center amplitude	
MTR_CURRENT_AD_CONV_TIME	$\text{CP_AD_CONVERSION_TIME} * 2.0f$	A/D conversion time count value	
MTR_VOLTAGE_LIMIT_OFFSET	0	Minimum A/D conversion time	
MTR_CENTER_AMPLITUDE_CNT	$(\text{int16_t})(\text{MTR_HALF_CARRIER_CNT} + \text{MTR_DEADTIME_CNT})$	PWM timer center amplitude	
MTR_OFFSET_CENTER_AMPLITUDE_CNT	MTR_CENTER_AMPLITUDE_CNT	A/D conversion time count value	[Used in 1-shunt mode and 2-phase modulation]
MTR_VOLTAGE_LIMIT_OFFSET	$(\text{int16_t})(((\text{MTR_DEADTIME} * 2) / (1000 / \text{MTR_CARRIER_FREQ})) * 0.5f * (1 << \text{MTR_Q_VOLTAGE}))$	Voltage offset limit [PU (V)]	
MTR_VOLTAGE_LIMIT_OFFSET	0	Voltage offset limit [PU (V)]	

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 3-72 List of Macro Definitions in "r_mtr_ra2t1.h" (2/2)

Macro	Definition value	Description	Remarks
MTR_DEADTIME_RATIO	$(\text{MTR_DEADTIME} * 1.0f * \text{MTR_CARRIER_FREQ}) / 1000.0f$	Deadtime compensation coefficient	
MTR_DEADTIME_CURRENT_LIMIT	$(\text{float})(\text{IP_CURRENT_RANGE}) / (1 \ll \text{MTR_ADC_12_BIT})$	Current limit value	
MTR_CTRL_PERIOD	$(\text{MTR_INT_DECIMATION} + 1) / (\text{MTR_CARRIER_FREQ} * 1000)$	Current control cycle	
MTR_SPEED_CTRL_PERIOD	CP_SPEED_CTRL_PERIOD	Speed control cycle	
MTR_ADC_DATA_SHIFT	0	A/D conversion value shift amount	
MTR_ADC_OFFSET	0x7FF	A/D conversion value offset	
MTR_CHARGE_CAP_WAIT_CNT	IP_CHARGE_CAP_WAIT_CNT	Charge time of DC capacitor	
MTR_V_PHASE_LEAD_COEF	$\text{FIX_fromfloat}(\text{CP_V_PHASE_LEAD_COEF}, \text{MTR_Q_V_PHASE_LEAD_COEF})$	Overcurrent detection reference value	
MTR_LOWEST_DUTY_CNT	$\text{MTR_CARRIER_CNT} + \text{MTR_DEADTIME_CNT}$	Overcurrent detection offset value	
MTR_MAX_DUTY_CNT	$\text{MTR_CARRIER_CNT} - 1$	Voltage leading phase coefficient	
MTR_MIN_DUTY_CNT	1	Timer count: lowest	
POEG_RECOVER_TIMEOUT	3	Count of POEG timeout	
ADC_CONVERT_TIMEOUT	0xFFFF	Count of ADC timeout	

Table 3-73 List of Macro Definitions in "r_mtr_common.h"

Macro	Definition value	Description	Remarks
R_MTR_ADCR_IDC	ADC_CHANNEL_0	Select ADCR register of point A	[Used in 1 shunt]
R_MTR_ADCR_IDCDUP	ADC_CHANNEL_DUPLEX	Select ADCR register of point AB	
R_MTR_ADCR_VDC	ADC_CHANNEL_8	Select ADCR register of inverter bus voltage	
R_MTR_ADC_SCAN_MASK_VW	0x00000006U	Scan mask of VW current channels	
R_MTR_ADC_SCAN_MASK_UVW	0x00000007U	Scan mask of UVW current channels	
R_MTR_ADC_SCAN_MASK_VW_IDC	0x00000007U	Scan mask of VW current and DC current channels	
R_MTR_ADC_SCAN_MASK_VDC	0x00000100U	Scan mask of DC voltage	
R_MTR_ADCR_IU	ADC_CHANNEL_0	Select ADCR register of Iu	[Used in 3 shunt]
R_MTR_ADCR_IV	ADC_CHANNEL_1	Select ADCR register of Iv	
R_MTR_ADCR_IW	ADC_CHANNEL_2	Select ADCR register of Iw	
R_MTR_ADS_CHANNEL_IDC	ADC_CHANNEL_0	Select ADS register of DC current	
R_MTR_ADS_CHANNEL_VDC	ADC_CHANNEL_8	Select ADS register of inverter bus voltage	
R_MTR_ADC_SCAN_MASK_VW	0x00000006U	Scan mask of VW current channels	
R_MTR_ADC_SCAN_MASK_UVW	0x00000007U	Scan mask of UVW current channels	
R_MTR_ADC_SCAN_MASK_VW_IDC	0x00000007U	Scan mask of VW current and DC current channels	
R_MTR_ADC_SCAN_MASK_VDC	0x00000100U	Scan mask of DC voltage	

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 3-74 List of Macro Definitions in "rm_mtr_common.h"

Macro	Definition value	Description	Remarks
MTR_TWOPi	$2 * 3.14159265f$	2π	
MTR_SQRT_3	$1.7320508f$	$\sqrt{3}$	
MTR_CW	1	CW	
MTR_CCW	-1	CCW	
MTR_ON	0	ON	
MTR_OFF	1	OFF	
MTR_CLR	0	Flag clear	
MTR_SET	1	Flag set	
MTR_OPL	0	Open loop	
MTR_CLL	1	Closed loop	
MTR_UNREACHED	0	Unreached	
MTR_REACHED	1	Reached	
MTR_WAIT_CNT_300US	1500	Wait for 300us	
MTR_ADC_12BIT	12	12bit ADC	

Table 3-75 List of Macro Definitions in "rm_mtr_parameter.h"

Macro	Definition value	Description	Remarks
MTR_PWM_DUTY_RANGE	$(1L << MTR_Q_VMOD)-1$	Duty range	
MTR_INPUT_V	IP_INPUT_V	Input voltage	
MTR_HALF_VDC	$MTR_INPUT_V * 0.5f$	50% of voltage	
MTR_MCU_ON_V	$MTR_INPUT_V * 0.8f$	80% of voltage	
MTR_OVERVOLTAGE_LIMIT	IP_OVERVOLTAGE_LIMIT	Overvoltage limit value	
MTR_UNDERVOLTAGE_LIMIT	IP_UNDERVOLTAGE_LIMIT	Undervoltage limit value	
MTR_RECIV_COEF	$(int32_t)(2 * PU_BASE_VOLTAGE_V / IP_VDC_RANGE * (1 << MTR_Q_RECIV))$	Angle range 2π	
MTR_ANGLE_RANGE	$(int32_t)(MTR_TWOPi * PU_SF_ANGLE * (1 << MTR_Q_ANGLE))$	Angle range π	
MTR_ANGLE_QUAT_RANGE	$(int32_t)(MTR_ANGLE_RANGE / 4)$	Angle range $\pi/2$	
MTR_OVERCURRENT_LIMIT	CP_OC_LIMIT	Current limit value	
MTR_I_LIMIT_VD	$1.5f * IP_INPUT_V * 0.5f$	Vd current PI limit	
MTR_I_LIMIT_VQ	$1.5f * IP_INPUT_V * 0.5f$	Vq current PI limit	
MTR_RPM_RAD	$(MP_POLE_PAIRS * MTR_TWOPi) / 60.0f$	Conversion from [rpm] to [rad/s]	
MTR_SPEED_LIMIT_RAD	$CP_SPEED_LIMIT_RPM * MTR_RPM_RAD$	Speed limit value [rad/s]	
MTR_MAX_SPEED_RAD	$CP_MAX_SPEED_RPM * MTR_RPM_RAD$	Maximum speed [rad/s]	
MTR_LIMIT_IQ	$MP_RATED_CURRENT * MTR_SQRT_3$	Speed PI output limit value	
MTR_I_LIMIT_CURRENT	$MP_RATED_CURRENT * MTR_SQRT_3$	Limit value for speed PI integral term output	
MTR_CL2OL_SPEED_RAD	$CP_CL2OL_SPEED_RPM * MTR_RPM_RAD$	Switching speed from sensorless to open loop [rad/s]	
MTR_OL2CL_SPEED_RAD	$CP_OL2CL_SPEED_RPM * MTR_RPM_RAD$	Switching speed from open loop to sensorless [rad/s]	
MTR_RECIM	$1.0f/(MP_BEMF_CONSTANT * PU_SF_BEMF_CONST)$	1/ BEMF constant [PU]	
MTR_DRAW_IN_WAIT_CNT	$CP_DRAW_IN_WAIT_TIME / MTR_SPEED_CTRL_PERIOD$	Draw-in wait time count value	
MTR_SWITCH_COUNT	$(uint16_t)(CP_OL2CL_SWITCH_TIME / MTR_SPEED_CTRL_PERIOD)$	Time[cnt] to switch open loop to closed loop	
MTR_CL2OL_JUDGE_WAIT_CNT	$(uint16_t)(CP_CL2OL_JUDGE_WAIT_TIME / (CP_SPEED_CTRL_PERIOD * 1000))$	Wait counts for open loop judgement	

Table 3-76 List of Macro Definitions in "rm_mtr_statemachine.h"

Macro	Definition value	Description	Remarks
MTR_MODE_INIT	0x00	Initialization mode	
MTR_MODE_DRIVE	0x01	Drive mode	
MTR_MODE_STOP	0x02	Stop mode	
MTR_SIZE_STATE	3	Number of states	
MTR_EVENT_STOP	0x00	Stop event	
MTR_EVENT_DRIVE	0x01	Run event	
MTR_EVENT_ERROR	0x02	Error event	
MTR_EVENT_RESET	0x03	Reset event	
MTR_SIZE_EVENT	4	Number of events	
MTR_STATEMACHINE_ERROR_NONE	0x00	No state machine error	
MTR_STATEMACHINE_ERROR_EVENTOUTBOUND	0x01	Event index out of range	
MTR_STATEMACHINE_ERROR_STATEOUTBOUND	0x02	State index is out of range	
MTR_STATEMACHINE_ERROR_ACTIONEXCEPTION	0x04	Action failure	

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 3-77 List of Macro Definitions in “r_mtr_foc_less_speed.h” (1/2)

Macro	Definition value	Description	Remarks
MTR_ERROR_NONE	0x0000	No error	
MTR_ERROR_OVER_CURRENT_HW	0x0001	Overcurrent error	
MTR_ERROR_OVER_VOLTAGE	0x0002	Overvoltage error	
MTR_ERROR_OVER_SPEED	0x0004	Excessive speed error	
MTR_ERROR_HALL_TIMEOUT	0x0008	Timeout error of Hall sensor detection Mismatch error of Hall sensor signal	
MTR_ERROR_BEMF_TIMEOUT	0x0010	BEMF timeout error	
MTR_ERROR_HALL_PATTERN	0x0020	Hall pattern error	
MTR_ERROR_BEMF_PATTERN	0x0040	BEMF pattern error	
MTR_ERROR_UNDER_VOLTAGE	0x0080	Undervoltage error	
MTR_ERROR_OVERCURRENT_SW	0x0100	Overcurrent error for SW	
MTR_ERROR_CHARGE_CAP_TIMEOUT	0x0400		
MTR_ERROR_UNKNOWN	0xffff	Undefined error	
MTR_ID_ZERO_CONST	0	d-axis current 0 control	
MTR_ID_MANUAL	1	d-axis current manual control	
MTR_ID_MTPA2FW	2	d-axis current MTPA and Field weakening control	
MTR_IQ_ZERO_CONST	0	q-axis current 0 control	
MTR_IQ_MANUAL	1	q-axis current manual control	
MTR_IQ_OL2CL	2	q-axis current for Open loop to Closed loop switch Control	
MTR_IQ_SPEED_PI_OUTPUT	3	Speed PI control output	
MTR_SPEED_ZERO_CONST	0	Speed 0 control	
MTR_SPEED_MANUAL	1	Speed manual control	
MTR_STOP	0x00	Drive stop	
MTR_DRIVE	0x81	Motor driving	
MTR_DRIVE_INIT	0x82	Initialize driving	
MTR_DRIVE_START	0x83	Start driving	
MTR_OFFSET_CALC_EXE	0x84	Offset is being removed	
MTR_DRIVE_END	0x01	Drive stop	
MTR_MASK_RUN_MODE	0x80	Mask run mode	
MTR_CTRL_NONE	0x00	Motor control none	
MTR_CTRL_OPL_DRAWIN	0x01	Open loop draw-in control	
MTR_CTRL_OPL_FORCE_SYNC	0x02	Open loop forced synchronize motor control	
MTR_CTRL_OPL_OPL2CLL	0x03	Open loop to closed loop control	
MTR_CTRL_OPL_CLL2OPL	0x04	Closed loop to open loop control	
MTR_CTRL_OPL_FORCE_SYNC_ACCEL	0x05	Open loop accelerate speed control	
MTR_CTRL_CLL_OPL2CLL	0x80	Open loop to closed loop control	
MTR_CTRL_CLL_SPEED_CTRL	0x81	Closed loop speed control	
MTR_DRIVE_OPL	0x00	Open loop speed control	
MTR_DRIVE_CLL	0x80	Closed loop speed control	
MTR_PHASE_U	0	U-phase	
MTR_PHASE_V	1	V-phase	
MTR_PHASE_W	2	W-phase	

RA2T1 Sensorless vector control for permanent magnetic synchronous motor
For Renesas Flexible Motor Control Series

Table 3-78 List of Macro Definitions in “r_mtr_foc_less_speed.h” (2/2)

Macro	Definition value	Description	Remarks
MTR_DRV_UVW	1	Duty size relationship U>V>W	[Used in 1-shunt mode]
MTR_DRV_UWV	2	Duty size relationship U>W>V	
MTR_DRV_VUW	3	Duty size relationship V>U>W	
MTR_DRV_VWU	4	Duty size relationship V>W>U	
MTR_DRV_WUV	5	Duty size relationship W>U>V	
MTR_DRV_WVU	6	Duty size relationship W>V>U	
MTR_DRV_MASK	0x0F	Mask of modulation drive pattern	
MTR_DRV_CURRENT_UNAVAILABLE	0x10	Flag of reproduction current	
MTR_DRV_DUTY_SHIFT_RUN	0x20	Duty shift run	
MTR_DRV_PWM_SHIFT_RUN	0x40	PWM shift run	
MTR_CROSS_MID_MIN	1	Duty cross for middle and minimum	
MTR_CROSS_MAX_MID	2	Duty cross for maximum and middle	
MTR_PWM_SHIFT_TIMER_HALF	(uint16_t)(MTR_CARRIER_CNT)	Half of PWM timer count	
MTR_PWM_SHIFT_DEADTIME	(uint16_t)(MTR_DEADTIME_CNT)	Count of deadtime	
MTR_PWM_SHIFT_AD_CONVERT_CNT	(uint16_t)(MTR_PWM_TIMER_FREQ * CP_AD_CONVERSION_TIME)	Count of AD convert	
MTR_PWM_SHIFT_TIMER_SET_100	(uint16_t)(0)	Count of 100% Duty (without deadtime)	
MTR_PWM_SHIFT_TIMER_NEAR_100	(uint16_t)(MTR_DEADTIME_CNT)	Count of 100% Duty (with deadtime)	
MTR_PWM_SHIFT_TIMER_NEAR_0	(uint16_t)(MTR_CARRIER_CNT)	Count of 0% Duty (without deadtime)	
MTR_PWM_SHIFT_TIMER_SET_0	(uint16_t)(MTR_CARRIER_CNT + MTR_DEADTIME_CNT)	Count of 0% Duty (with deadtime)	

3.6 Interrupt Processing Specifications

This section describes interrupt processing for the sample code.

The interrupt processing consists of two cycle interrupts: a carrier cycle interrupt and a 1[ms] cycle interrupt. The carrier cycle interrupt in 3-shunt mode occurs every 50[μs], and the carrier cycle interrupt in 1-shunt mode occurs every 100[μs] (1-time decimations of 50[μs]).

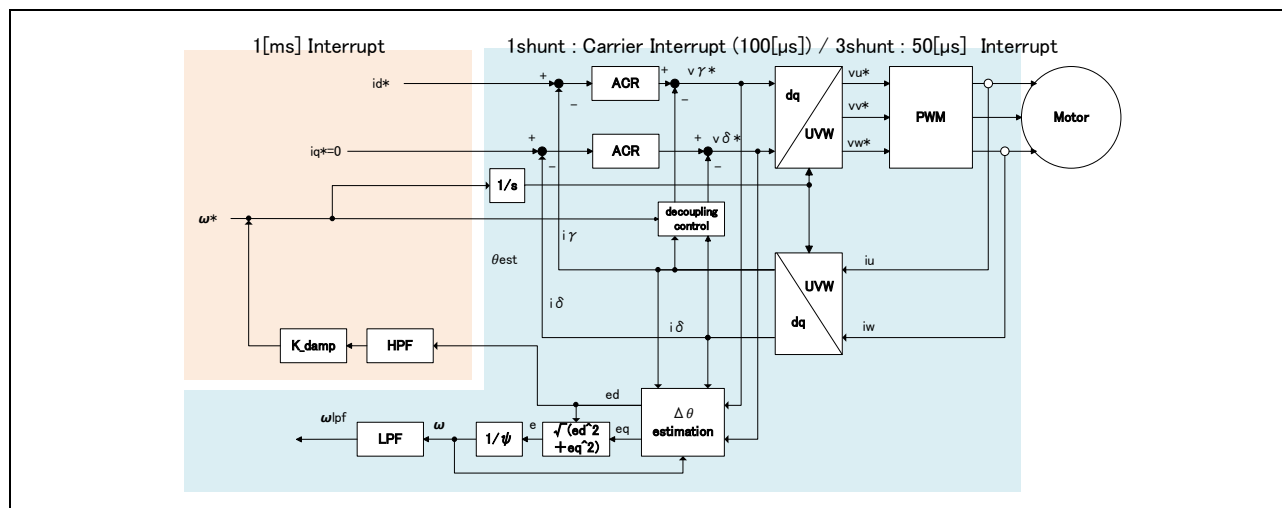


Figure 3-12 Interrupt processing inside control block (during open loop control)

Here is an outline of driving force in an open loop. The d-axis reference current is allocated, the speed is ramped up from 0 to the reference speed, and the angle is updated by using the angle information obtained by integrating the reference speed. The estimated speed is predicted by using the BEMF value output from the phase error estimator.

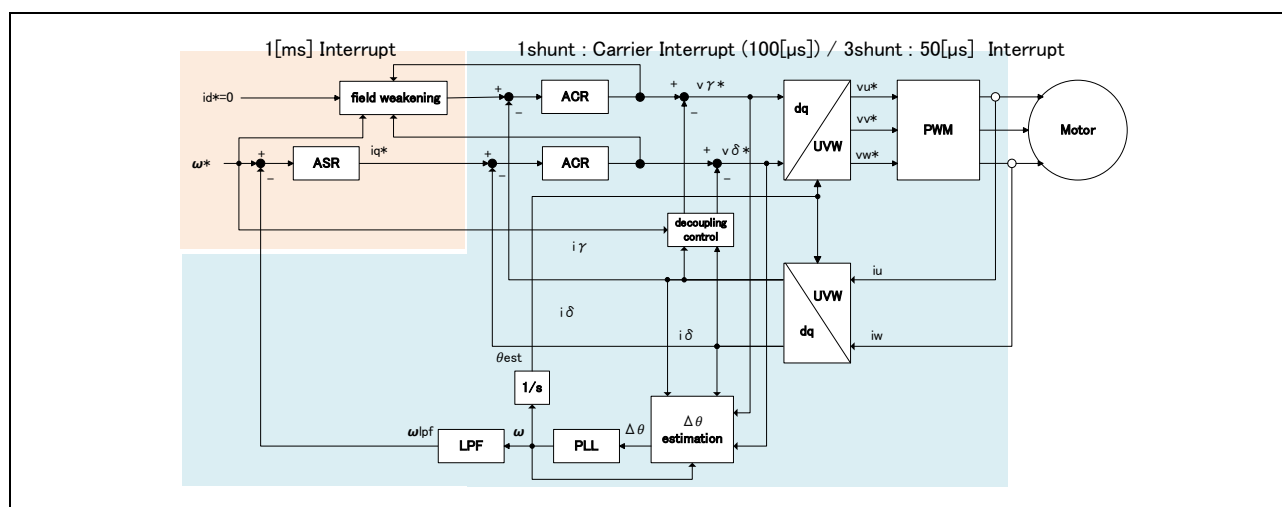


Figure 3-13 Interrupt processing inside control block (during closed loop control)

Here is an outline of the driving force in a closed loop. Suppose the d-axis current is set to zero, the phase error derived from the reference speed value and phase error estimator is input into the speed controller (ASR), in which the output is obtained as the reference value for q-axis current. The angle is updated by using the angle information obtained from integrating the estimation speed.

To keep the interrupt processing occupancy rate at around 60%, one of every two carrier cycle interrupts in 1 shunt mode is skipped. Therefore, the control cycle of the current control system becomes 100[μs], because one of 50[μs] is skipped.

The process of speed control system and dq-axis current reference values output are carried out in the 1[ms] cycle interrupt. The control cycle for the speed control system is 1[ms]. The interrupt timing for 1-shunt mode and 3-shunt mode is the same. The carrier cycle interrupts occur at valleys of 100[μs] interval (100[μs]: 1-time decimations of 50[μs]), and the PWM duty updated at valleys.

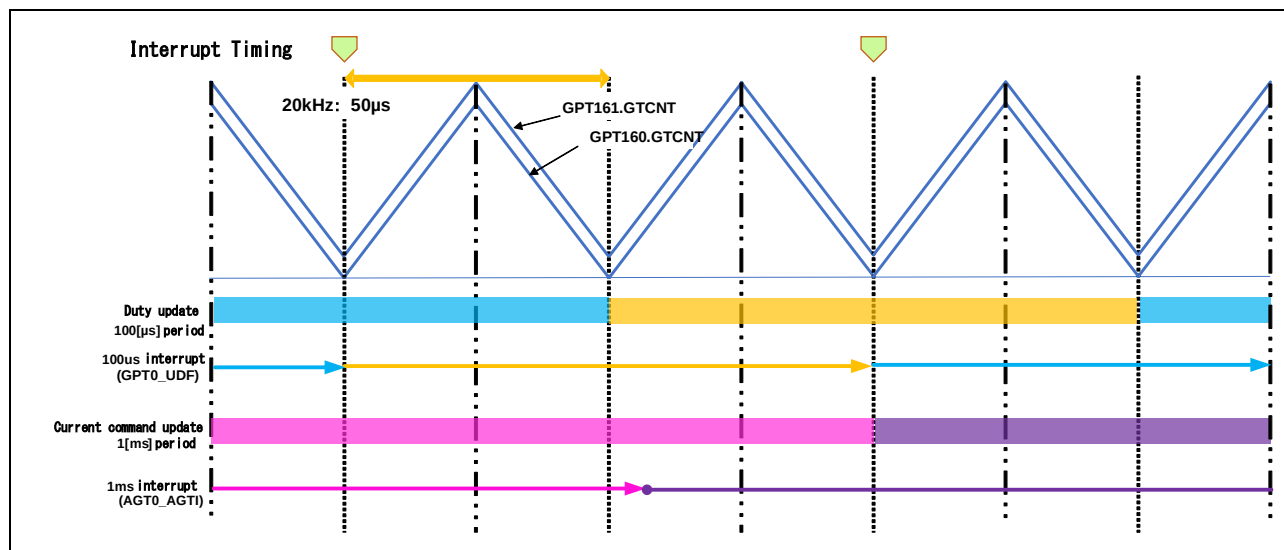


Figure 3-14 Image of interrupt generation and update of output value

3.7 Control flows (flowcharts)

3.7.1 Main process

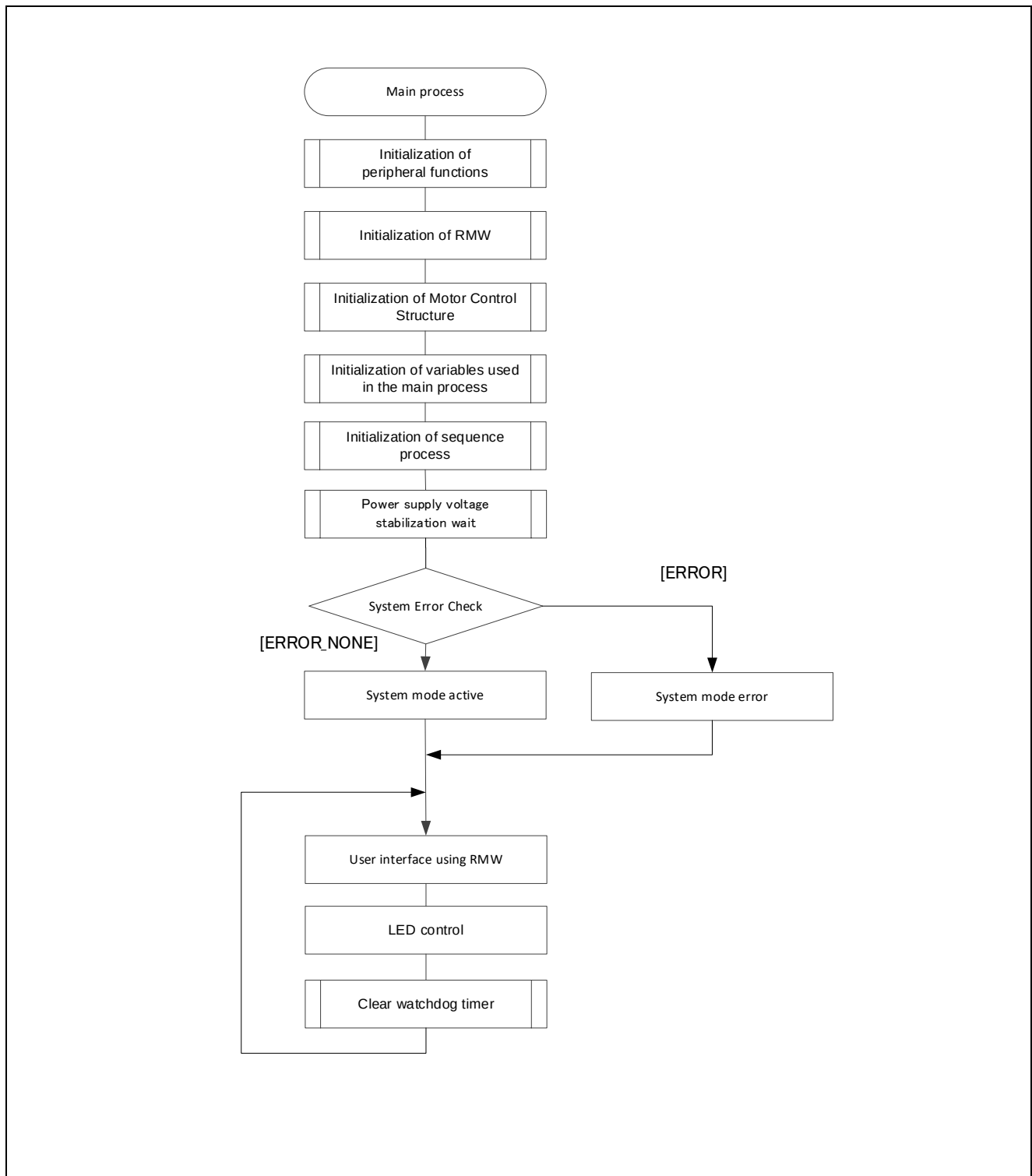


Figure 3-15 Main Process Flowchart

3.7.2 Carrier interrupt handling (Used in 1-shunt mode)

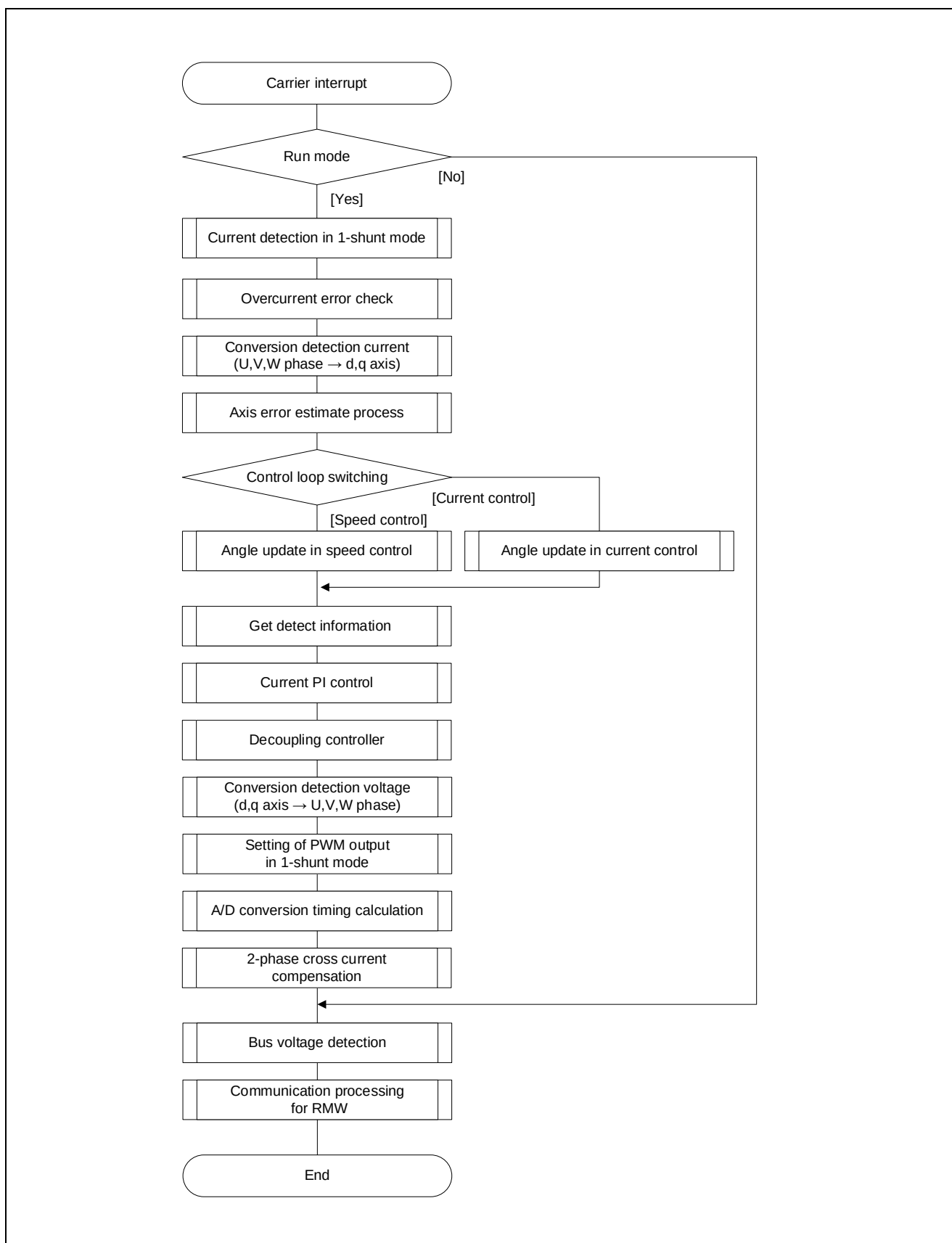


Figure 3-16 Carrier Cycle Interrupt Handling Flowchart

3.7.3 Carrier interrupt handling (Used in 3-shunt mode)

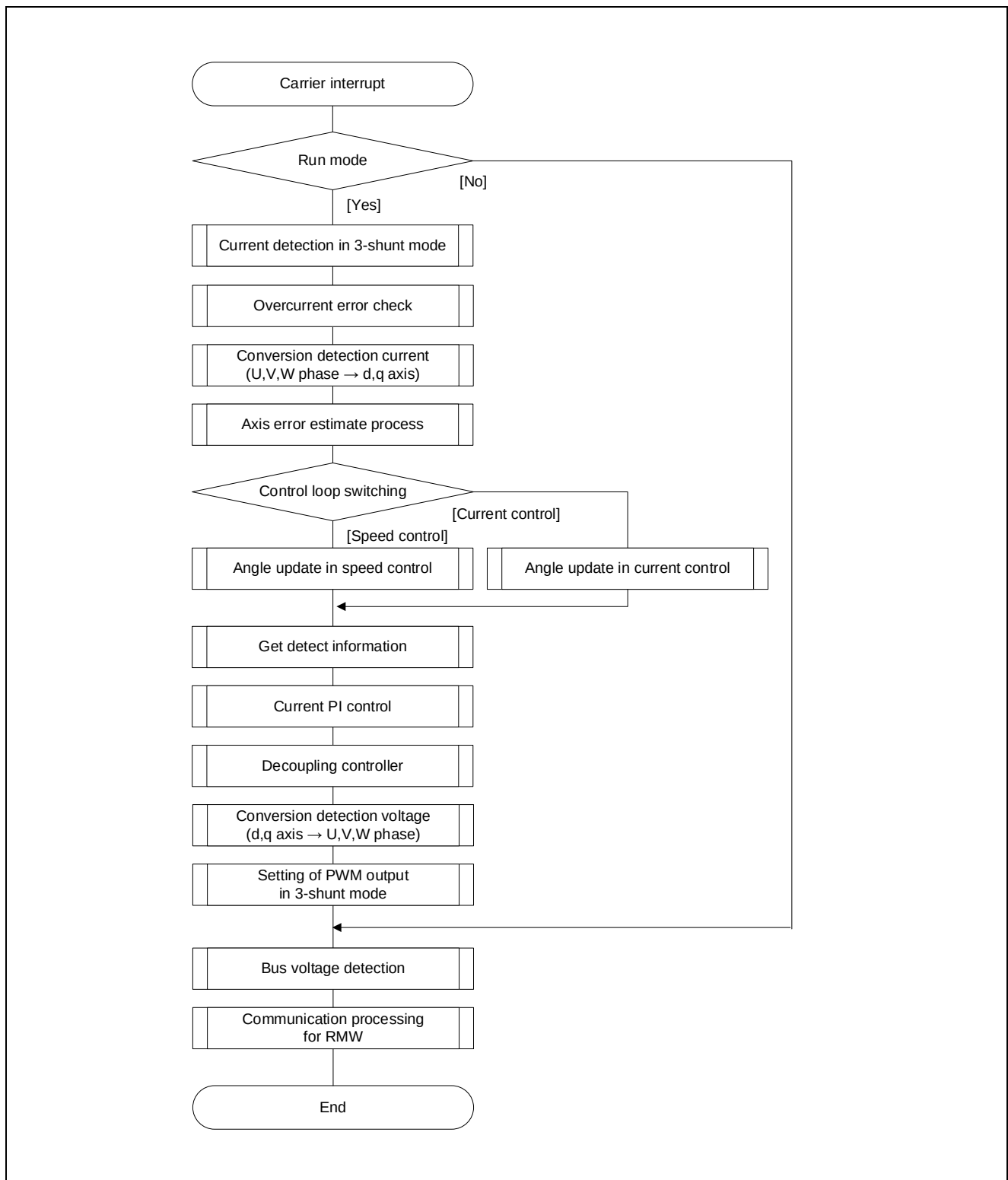


Figure 3-17 Carrier Cycle Interrupt Handling Flowchart

3.7.4 1-ms interrupt handling

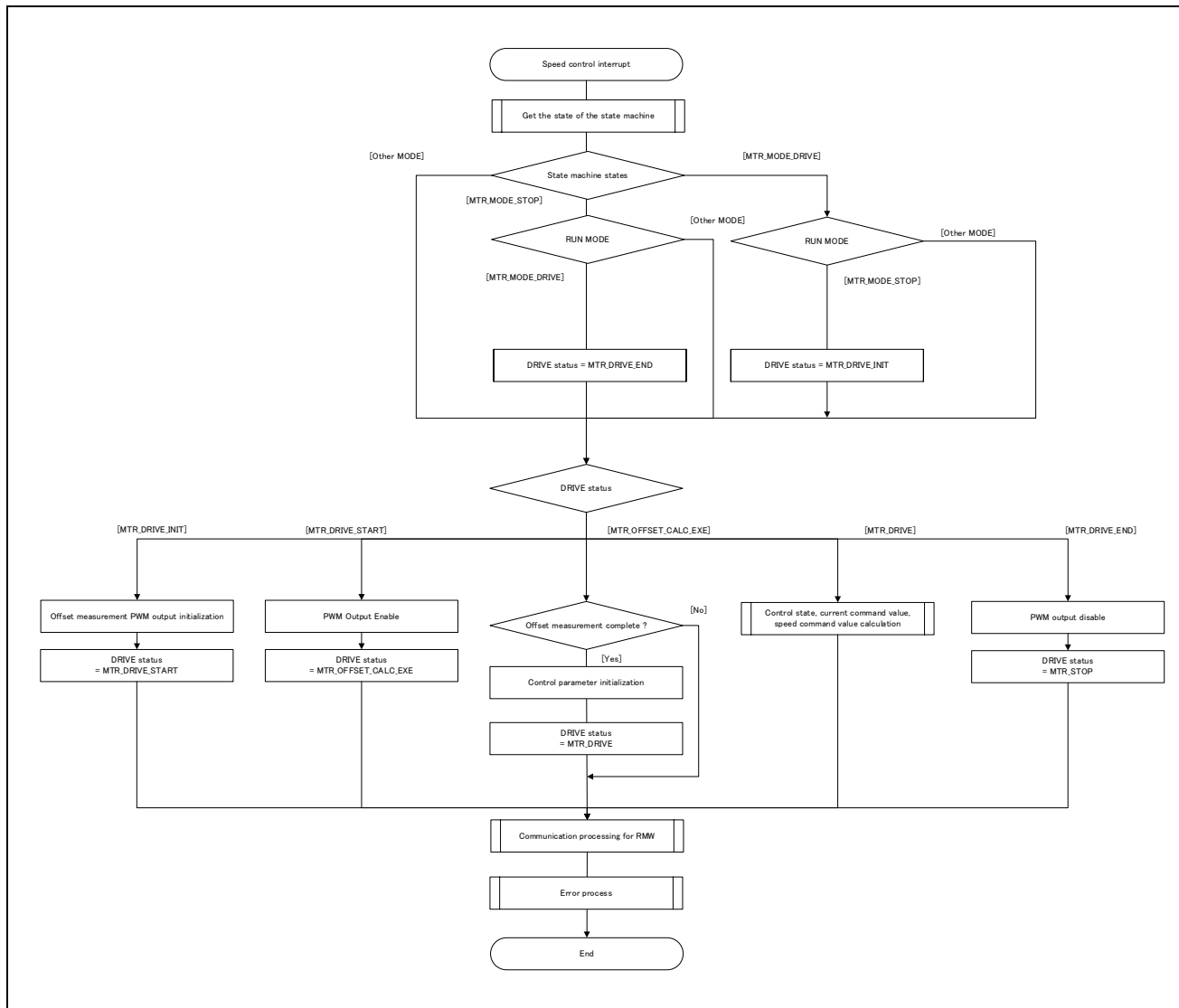


Figure 3-18 1-ms Interrupt Handling Flowchart

4. Usage of Motor Control Development Support Tool, Renesas Motor Workbench

4.1 Overview

In the target sample programs described in this application note, you can use user interfaces (rotation/stop command, rotational speed command, etc.) based on the motor control development support tool Renesas Motor Workbench. Please refer to the 'Renesas Motor Workbench User's Manual' for usage and more details. You can find the 'Renesas Motor Workbench' on Renesas Electronics Corporation's website.

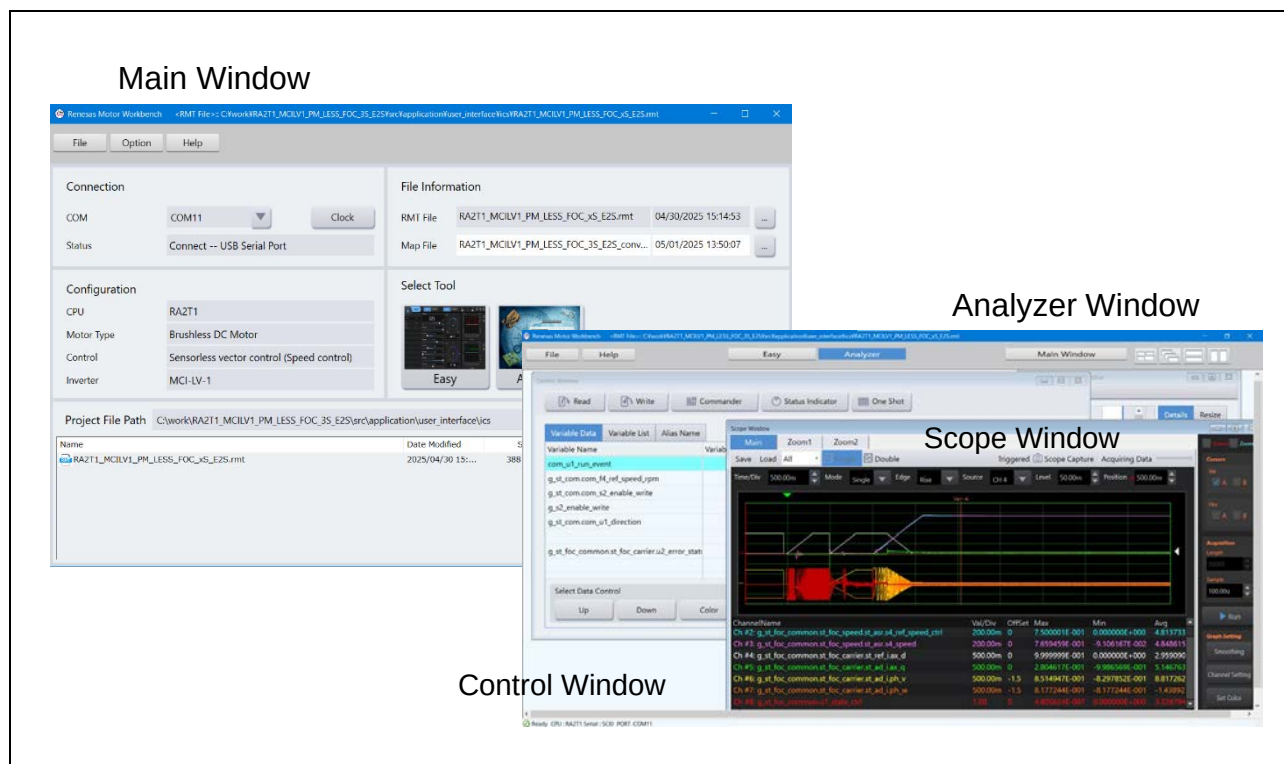
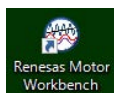


Figure 4-1 Screenshots of Renesas Motor Workbench

How to use the motor control development support tool, Renesas Motor Workbench



- (1) Start Renesas Motor Workbench by clicking this icon.
- (2) From the menu bar in the main window, select [File] -> [Open RMT File(O)].
Select RMT file in the '[Project Folder]/application/user_interface/ics/'.
- (3) Use the 'Connection' COM select menu to choose the COM port.
- (4) Click the 'Analyzer' icon on the right side of the Main Window.
(The Analyzer Window will be displayed.)
- (5) Please refer to '4.3 Operation Example for Analyzer' for the motor driving operation.

4.2 List of variables for Analyzer

Table 4-1 is a list of variables for the Analyzer. These variable values are reflected to the protect variables when the same values as g_s2_enable_write are written to g_st_com.s2_enable_write. However, note that variables with (*) do not depend on g_st_com.s2_enable_write.

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

Variable	Type	Content	Remarks ([]: reflection variable name)
com_u1_run_event (*)	uint8_t	Change run mode 0: Stop event 1: Drive event 2: Error event 3: Reset event	[g_u1_run_event]
g_st_com.st_com.f4_acr_nf_hz	float	Direction of rotation 0: CW 1: CCW	[g_st_ctrl_params.st_com.f4_acr_nf_hz]
g_st_com.st_com.f4_acr_lpf_cof_hz	float	Resistance [Ω]	[g_st_ctrl_params.st_com.f4_acr_lpf_cof_hz]
g_st_com.st_com.f4_acr_deadband_lsb	float	d-axis inductance [H]	[g_st_ctrl_params.st_com.f4_acr_deadband_lsb]
g_st_com.st_com.f4_asr_nf_hz	float	q-axis inductance [H]	[g_st_ctrl_params.st_com.f4_asr_nf_hz]
g_st_com.st_com.f4_asr_lpf_cof_hz	float	BEMF constant [Vs/rad]	[g_st_ctrl_params.st_com.f4_asr_lpf_cof_hz]
g_st_com.st_com.f4_asr_deadband_lsb	float	Inertia [$\text{kgm}^2/(\text{rad} \cdot \text{s}^2)$]	[g_st_ctrl_params.st_com.f4_asr_deadband_lsb]
g_st_com.st_com.f4_asr_ki_aug	float	Static friction coefficient [$\text{kgm}^2/(\text{rads}^2)$]	[g_st_ctrl_params.st_com.f4_asr_ki_aug]
g_st_com.st_com.f4_pll_nf_hz	float	Kinetic friction coefficient [$\text{kgm}^2/(\text{rads}^2)$]	[g_st_ctrl_params.st_com.f4_pll_nf_hz]
g_st_com.st_com.f4_pll_deadband_lsb	float	Number of pole pairs	[g_st_ctrl_params.st_com.f4_pll_deadband_lsb]
g_st_com.com_u1_direction	uint8_t	Current offset detection time	[g_st_foc_common.st_foc_speed.st_asr.s1_ref_dir]
g_st_com.com_f4_mtr_r	float	Command rotational speed [rpm]	[g_st_foc_common.st_foc_carrier.st_motor.s4_mtr_r]
g_st_com.com_f4_mtr_ld	float	Limit of acceleration [rpm/ms]	[g_st_foc_common.st_foc_carrier.st_motor.s4_mtr_ld]
g_st_com.com_f4_mtr_lq	float	Maximum speed [rpm]	[g_st_foc_common.st_foc_carrier.st_motor.s4_mtr_lq]
g_st_com.com_f4_mtr_m	float	Current PI control natural frequency [Hz]	[g_st_foc_common.st_foc_carrier.st_motor.s4_mtr_ke] [g_st_foc_common.st_foc_carrier.st_phe.s4_coef_reci_ke]
g_st_com.com_f4_mtr_j	float	Speed PI control natural frequency [Hz]	[g_st_foc_common.st_foc_carrier.st_motor.s4_mtr_jj]
g_st_com.com_f4_mtr_d0	float	ASR LPF natural frequency [Hz]	[g_st_ctrl_params.f4_d0_div_pm]
g_st_com.com_f4_mtr_d1	float	ACR LPF natural frequency [Hz]	[g_st_ctrl_params.f4_d1_div_p2m]
g_st_com.com_u2_mtr_pp	uint16_t	PLL natural frequency [Hz]	[g_st_foc_common.st_foc_carrier.st_motor.u2_mtr_pp]
g_st_com.com_u2_offset_calc_cnt	uint16_t	Deadband of current PI calculation [LSB]	[g_st_foc_common.st_foc_carrier.st_tscs.u4_offset_calc_cnt]
g_st_com.com_f4_ref_speed_rpm	float	Deadband of speed PI calculation [LSB]	[g_st_foc_common.st_foc_speed.st_asr.s4_ref_speed]
g_st_com.com_f4_ramp_limit_speed_rpm	float	Deadband of phase locked loop calculation [LSB]	[g_st_foc_common.st_foc_speed.st_asr.s4_lim_ramp_speed]
g_st_com.com_f4_max_speed_rpm	float	Augmentation rate for integral part of ASR	[g_st_foc_common.st_foc_speed.st_asr.s4_max_speed]
g_st_com.com_f4_cl2ol_speed_rpm	float	Switching speed from sensorless to open loop [rpm]	[g_st_foc_common.st_foc_speed.st_asr.s4_th_cl2ol_speed]
g_st_com.com_f4_ol2cl_speed_rpm	float	Switching speed from open loop to sensorless [rpm]	[g_st_foc_common.st_foc_speed.st_asr.s4_th_ol2cl_speed]

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

Variable	Type	Content	Remarks ([]: reflection variable name)
g_st_com.com_f4_ol_ref_id	float	Open loop d-axis command current [A]	[g_st_foc_common.st_foc_carrier.st_acr.s4_ref_id_ol]
g_st_com.com_f4_draw_in_wait_time	float	Draw-in wait time count value [s]	[g_st_foc_common.st_foc_speed.u2_cnt_draw_in_wait_buff]
g_st_com.com_f4_init_asr_intg	float	ASR integral term initial value during sensorless transition	[g_st_foc_common.st_foc_speed.st_asr.s4_init_intg]
g_st_com.com_f4_ramp_limit_current	float	Limit value for current rise [A/ms]	[g_st_foc_common.st_foc_carrier.st_acr.s4_lim_ramp_current]
g_st_com.com_f4_i_repro_cof_hz	float	LPF cut-off frequency for current reproduction [Hz]	[g_st_ctrl_params.f4_i_repro_cof_hz]
g_st_com.com_f4_mod_3ph2ph_speed_rpm	int32_t	Switching speed from 3-phase modulation to 2-phase modulation [rpm]	[g_st_foc_common.st_foc_carrier.st_ssc.s.s4_mod_3ph2ph_speed_rad]
g_st_com.com_f4_mod_2ph3ph_speed_rpm	int32_t	Switching speed from 2-phase modulation to 3-phase modulation [rpm]	[g_st_foc_common.st_foc_carrier.st_ssc.s.s4_mod_2ph3ph_speed_rad]
g_st_com.com_s2_ad_point_a_adj_cnt_3ph	int16_t	Adjustment value of A/D conversion timing A (3-phase modulation)	[g_st_foc_common.st_foc_carrier.st_ssc.s.s2_ad_point_a_adj_cnt_3ph]
g_st_com.com_s2_ad_point_b_adj_cnt_3ph	int16_t	Adjustment value of A/D conversion timing B (3-phase modulation)	[g_st_foc_common.st_foc_carrier.st_ssc.s.s2_ad_point_b_adj_cnt_3ph]
g_st_com.com_s2_ad_point_a_adj_cnt_2ph	int16_t	Adjustment value of A/D conversion timing A (2-phase modulation)	[g_st_foc_common.st_foc_carrier.st_ssc.s.s2_ad_point_a_adj_cnt_2ph]
g_st_com.com_s2_ad_point_b_adj_cnt_2ph	int16_t	Adjustment value of A/D conversion timing B (2-phase modulation)	[g_st_foc_common.st_foc_carrier.st_ssc.s.s2_ad_point_b_adj_cnt_2ph]
g_st_com.com_s2_ad_point_a_adj_cnt	int16_t	Adjustment for A/D delay counts for A point	[g_st_foc_common.st_foc_carrier.st_ssc.s.s2_ad_point_a_adj_cnt]
g_st_com.com_s2_ad_point_b_adj_cnt	int16_t	Adjustment for A/D delay counts for B point	[g_st_foc_common.st_foc_carrier.st_ssc.s.s2_ad_point_b_adj_cnt]
g_st_com.com_s2_mod_2ph_bot_change_cnt	int16_t	Number of counts during 2-phase modulation transition	[g_st_foc_common.st_foc_carrier.st_mod.s2_cnt_2ph_bot_change]
g_st_com.com_f4_damp_hpf_cof_hz	float	HPF cutoff frequency for damping control [Hz]	[g_st_ctrl_params.f4_damp_hpf_cof_hz]
g_st_com.com_f4_damp_zeta	float	Damping coefficient of damping control	[g_st_ctrl_params.f4_damp_zeta]
g_st_com.com_f4_damp_speed_limit_rate	float	Damping control speed limit	[g_st_foc_common.st_foc_speed.st_ol2cl.st_damp.s4_speed_lim_rate]
g_st_com.com_f4_pherr_lpf_cof_hz	float	Phase error LPF cutoff frequency for Open loop to Closed loop switch Control	[g_st_ctrl_params.f4_pherr_lpf_cof_hz]
g_st_com.com_f4_ol2cl_switch_time	float	Time[s] to switch open loop to sensor-less	[g_st_foc_common.st_foc_speed.st_ol2cl.u2_cnt_switch]
g_st_com.com_s2_enable_write	int16_t	Variable to allow to variable writing	[g_s2_enable_write]

4.3 Operation Example for Analyzer

An example of a motor driving operation using Analyzer is shown below. For the operation, the "Control Window" shown in Figure 4-1 is used. Refer to the 'Renesas Motor Workbench User's Manual' for details about the "Control Window".

- Driving the motor
 - ① Confirm that the [W?] check boxes contain checkmarks for "com_u1_run_event", "g_st_com.f4_ref_speed_rpm", and "g_st_com.s2_enable_write".
 - ② Input a reference rotational speed value in the [Write] box of "g_st_com.f4_ref_speed_rpm".
 - ③ Click the "Write" button.
 - ④ Click the "Read" button. Confirm the [Read] box of "g_st_com.f4_ref_speed_rpm" and "g_st_com.s2_enable_write".
 - ⑤ Input the value in the [Read] box of "g_st_com.s2_enable_write", confirmed in step (4), in the [Write] box of "g_st_com.s2_enable_write".
 - ⑥ Input a value of "1" in the [Write] box of "com_u1_run_event".
 - ⑦ Click the "Write" button.

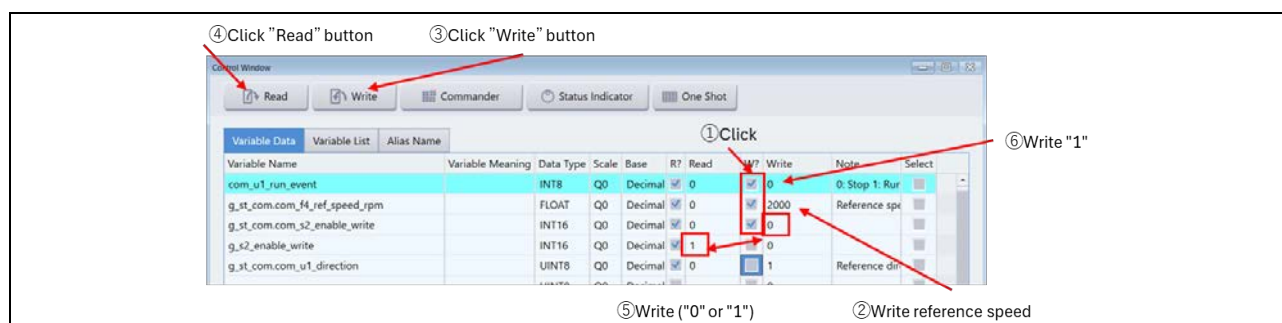


Figure 4-2 Procedure - Driving the motor

- Stop the motor
 - ① Input a value of "0" in the [Write] box of "com_u1_run_event".
 - ② Click the "Write" button.

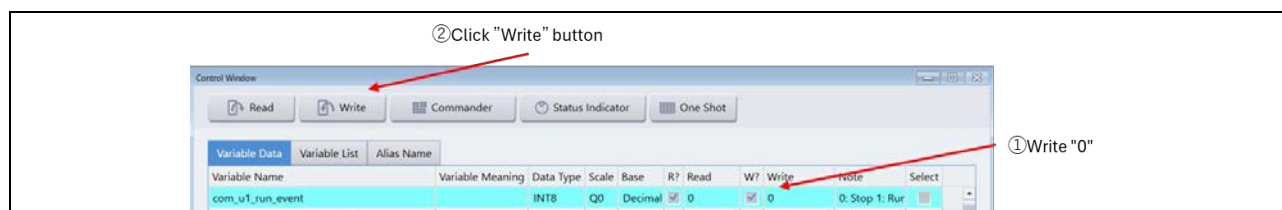


Figure 4-3 Procedure - Stop the motor

- Error cancel operation
 - ① Input a value of "3" in the [Write] box of "com_u1_run_event".
 - ② Click the "Write" button.

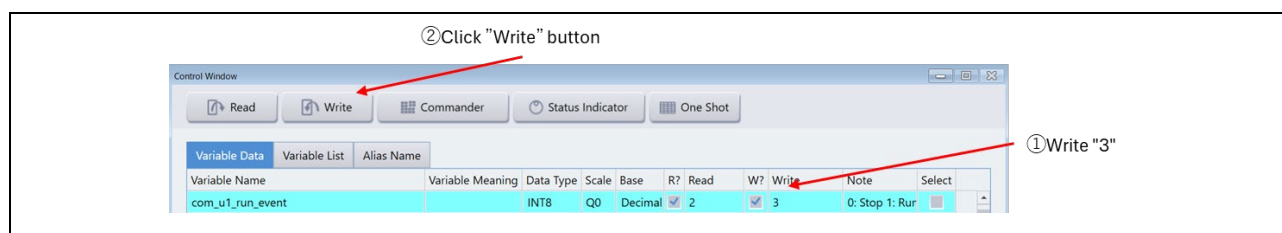


Figure 4-4 Procedure - Error cancel operation

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	July 9, 2025	-	First edition
1.01	June 30, 2026	-	<ul style="list-style-type: none">- Update of the development environment- Change of motor parameters

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/.