RA2T1 120-degree conducting control of permanent magnetic synchronous motor

For Renesas Flexible Motor Control Series

Summary

This application note explains the sample programs driving a permanent magnetic synchronous motor in the 120-degree conducting 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 by using the following device.

- RA2T1(R7FA2T1074CFL)

Target sample programs

The target sample programs of this application note are as follows.

- RA2T1_MCILV1_PM_120_E2S_V101 (IDE: e^2 studio)

Reference

- 120-degree conducting control of permanent magnetic synchronous motor Algorithm (R01AN2657)
- 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. Overview 3

2. System overview 4

3. Descriptions of the control program 15

4. Usage of Motor Control Development Support Tool, ‘Renesas Motor Workbench’ 76

1. Overview

This application note explains how to implement the 120-degree conducting control sample programs of permanent magnetic synchronous motor (PMSM) using the RA2T1 microcontroller and how to use the motor control development support tool, 'Renesas Motor Workbench'. Note that this sample programs use the algorithm described in the application note '120-degree conducting control of permanent magnetic synchronous motor: algorithm'.

1.1 Development environment

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

Table 1-1 Development Environment of the Sample Programs (H/W)

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 (S/W)

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

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)

2.2 Hardware configuration

The hardware configuration is shown below.

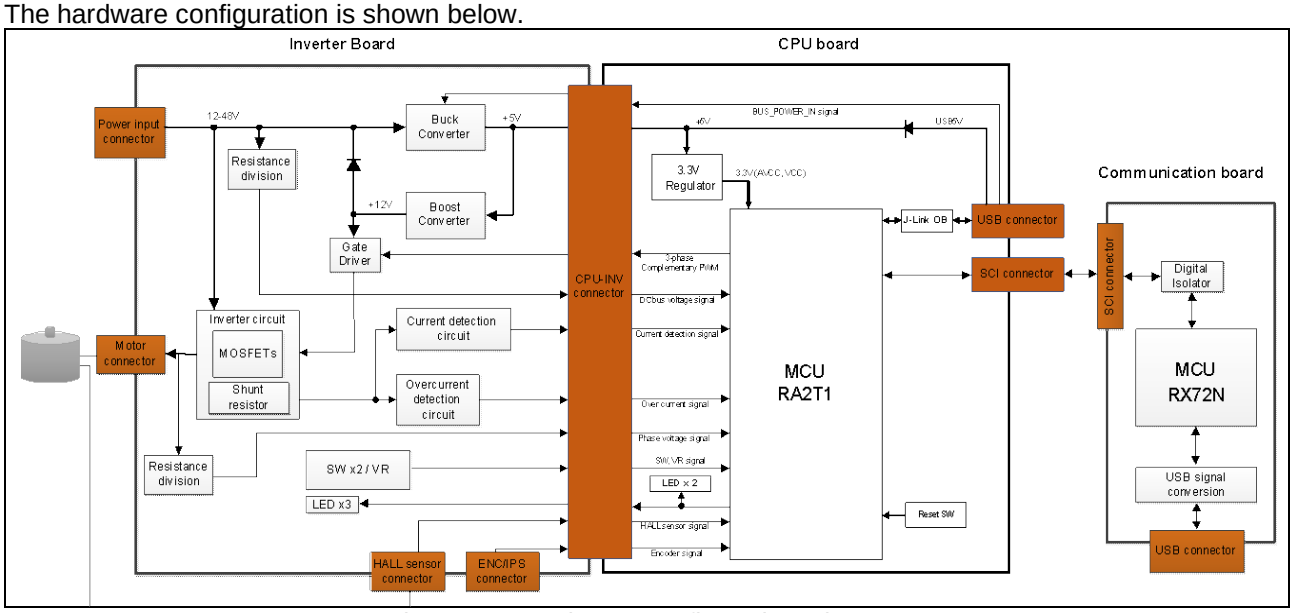


Figure 2-1 Hardware Configuration Diagram

2.3 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)

2.3.1 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

In this sample program, connecting the Hall sensor signal lines to MCI-LV-1 is required to use a Hall sensor. The correspondence table between MCI-LV-1 and MOONS's R42BLD30L3 is shown below. For R42BLD30L3 connectors, MOLEX 43025-0600 is used. Mating parts are 43020-0600 and 43045-0600. The wire is AWG24 UL1061. If it is difficult to prepare the terminals, disconnect the connector from the Hall sensor wire, strip the coating, insert and secure it to the terminal block of MCI-LV-1.

Table 2-6 Correspondence between Hall sensor and inverter sensor input terminals

Signal line specifications	MCI-LV-1	MOON’S R42BLD30L3
DGND	CN6-1	5-BLACK
+5V	CN6-2	1-RED
HALL_W	CN6-3	4-BLUE
HALL_V	CN6-4	3-YELLOW/WHITE
HALL_U	CN6-5	6-ORANGE/YELLOW

(2) CPU board

Table 2-7 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 2-8 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 2-9 User interface of Communication board

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

Table 2-10 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

Table 2-11 shows the pin functions of the RA2T1 microcontroller in this system.

Table 2-11 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

2.3.2 Peripheral functions

Table 2-12 is a list of peripheral functions used in this system.

Table 2-12 List of Peripheral Functions

Peripheral Function	Usage
12-bit A/D converter (ADC12)	- Inverter bus voltage measurement - DC-link current detection - U/V/W phase voltages measurement [Used in Sensorless mode]
Low Power Asynchronous General Purpose Timer (AGTW 0, 1)	- Free-running timer for rotational speed measurement (AGT0) - Delay timer for changing conducting pattern (AGT1) [Sensorless mode]
General PWM Timer (GPT 0 ,1 ,2, 3)	- PWM output using complementary PWM mode - 50[us] interval timer(GPT0,1,2) - 1[ms] interval timer(GPT3)
External Interrupt (IRQ 0, 6, 7)	- Input signal from Hall effect sensor [Hall effect sensor mode]
Port Output Enable for GPT (POEG)	- PWM output forcibly shut off by trigger input
Watch dog timer (WDT)	- Program runaway detection

(1) 12-bit A/D converter (ADC12)
Inverter bus voltage (Vdc), DC link current (Idc), UVW phase voltage[Sensorless mode] are measured by using the '12-bit A/D converter'.

(2) Low Power Asynchronous General Purpose Timer (AGTW 0,1)
Uses rotational speed measurement and changing conducting pattern [Sensorless mode].

(3) General PWM Timer (GPT 0, 1, 2, 3)
AD conversion can be executed at any timing using A/D Conversion Start Request Timing Register.
Asymmetric waveforms can be output using complementary PWM mode 3.

(4) External Interrupt (IRQ 0, 6, 7)
Inputs the motor's magnetic pole position detection signal (Hall sensor output signal). Using interrupt mode, it captures the Hall sensor signal (detects position information), changes the current conduction pattern, and measures the rotation speed.

(5) 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.

(6) Watchdog Timer (WDT)
Watchdog Timer is used as a program runaway detection.

2.4 Software structure

2.4.1 Software specifications

The basic specifications of the software for this system are shown below. For details of 120-degree conducting control, refer to the application note '120-degree conducting control of permanent magnetic synchronous motor: algorithm'.

Table 2-13 Basic Specifications of Software[1/2]

Item	Content	
Control method	120-degree conducting method (high-side chopping)	
Current detection method	Use the method of detecting DC link current.	
Motor rotation start/stop	Operation using the motor control development support tool[Note]	
Position detection of rotor magnetic pole	Hall effect sensor: The positioning of the rotor is obtained based on interruption of signals from Hall effect sensors. (every 60 degrees) Sensorless: The position of the rotor is obtained based on the zero-crossing of the BEMF voltage, which is scaled at the ADC input. (every 60 degrees) - At the change of voltage pattern, PWM duty is updated, and the motor windings are energized accordingly.	
Control mode	There are 4 control modes in this sample software. - Sensorless current mode - Sensorless voltage mode - Hall effect sensor-based current mode - Hall effect sensor-based voltage mode For details, please refer to 4.4.	
Input voltage	DC24[V]	
Main clock frequency	CPU clock: ICLK 64[MHz] TRD clock: PCLKB 32[MHz]	
Carrier frequency (PWM)	20 [kHz]	
Dead time	1 [µs]	
Control cycle	Speed PI control: 1 [ms] Current PI control: 50 [µs]	
Range of rotational speed control	CW:0 [rpm] ~ 2400[rpm] CCW:0 [rpm] ~ 2400 [rpm] Note that the motor is driven at open-loop mode below 500 [rpm]	
Optimization setting	Optimize more(-O2)	
ROM/RAM Size	ROM	28.756KB [Sensorless current mode] 28.892KB [Sensorless voltage mode] 27.036KB [Hall effect sensor-based current mode] 26.624KB [Hall effect sensor-based voltage mode]
	RAM	2.752KB [Sensorless current mode] 2.692KB [Sensorless voltage mode] 2.644KB [Hall effect sensor-based current mode] 2.584KB [Hall effect sensor-based voltage mode]

Table 2-14 Basic Specifications of Software[2/2]

Item	Content
Processing stop for protection	- The motor control signal output (six outputs) are disabled under any of the following conditions. 1. Inverter bus voltage exceeds 48 V (monitored per 1 [ms]) 2. Inverter bus voltage is less than 8 V (monitored per 1 [ms]) 3. Rotational speed exceeds 3262.5 rpm (monitored per 1 [ms]) 4. Hall effect sensor signals or zero-crossing are not detected for a constant period. 5. Detection of an unexpected output voltage pattern 6. The current of each phase exceeds 3.955 A 7. When an external overcurrent detection signal is detected, the PWM output terminal goes low.

[Note]

1. For details, please refer to “Renesas Motor Workbench”
2. Please refrain from operating the motor for extended period at speed above rated rotational speed.

2.4.2 Software file structure

The folder and file configurations of the sample programs are shown in Table 2-15.

Table 2-15 Folder and File Configuration of the Sample Program

Folder		File	Content
src/application	board	r_mtr_board.c	LED Control
		r_mtr_board.h	
	user_interface/ics	ICS2_RA2T1.o	communication Library for RMW
		ICS2_RA2T1.h	
		r_mtr_ics.c	Analyzer(Note1) UI
		r_mtr_ics.h	Function definition for Analyzer(Note1) UI
		RA2T1_MCILV1_PM_120_E2S_V101_conv.map	Map file
		RA2T1_MCILV1_PM_120_E2S_V101.rmt	RMW initial setting file
		ElfMapConverter.exe	Map file converter
		convert.bat	
	main	mtr_main.c	Main function
		mtr_main.h	
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/driver		r_mtr_common.h	common definition
		r_mtr_ADC.c	ADC driver
		r_mtr_ADC.h	Function definition for ADC driver
		r_mtr_AGT.c	AGT driver
		r_mtr_AGT.h	Function definition for AGT driver
		r_mtr_GPT_THREE_PHASE.c	GPT driver
		r_mtr_GPT_THREE_PHASE.h	Function definition for GPT driver
		r_mtr_pin_data.c	I/O port driver
		r_mtr_pin_data.h	Function definition for I/O port driver
		r_mtr_POEG.c	POEG driver
		r_mtr_POEG.h	Function definition for POEG driver
		r_mtr_ra2t1.c	MCU control
		r_mtr_ra2t1.h	Function definition MCU control
src/middle		R_DSP_RA2_GCC_32BIT.h	DSP definition
		rm_mtr_carrier.c	Carrier interrupt
		rm_mtr_common.h	Common definition
		rm_mtr_ctrl_gain.o	Gain control
		rm_mtr_ctrl_gain.h	Function definition for gain control
		rm_mtr_120.c	120-degree conducting control
		rm_mtr_120.h	Function definition for 120-degree conducting control
		rm_mtr_ol2cl_ctrl.c	Open loop to Closed loop switch control
		rm_mtr_ol2cl_ctrl.h	Function definition for Open loop to Closed loop switch control
		rm_mtr_parameter.h	Motor control parameter definition
		rm_mtr_speed.c	Speed control interrupt processing
		rm_mtr_statemachine.h	State transition processing related definitions
		rm_mtr_driver.c	Driver access
		rm_mtr_driver.h	Function definition for driver access

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.

2.4.3 Module configuration

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

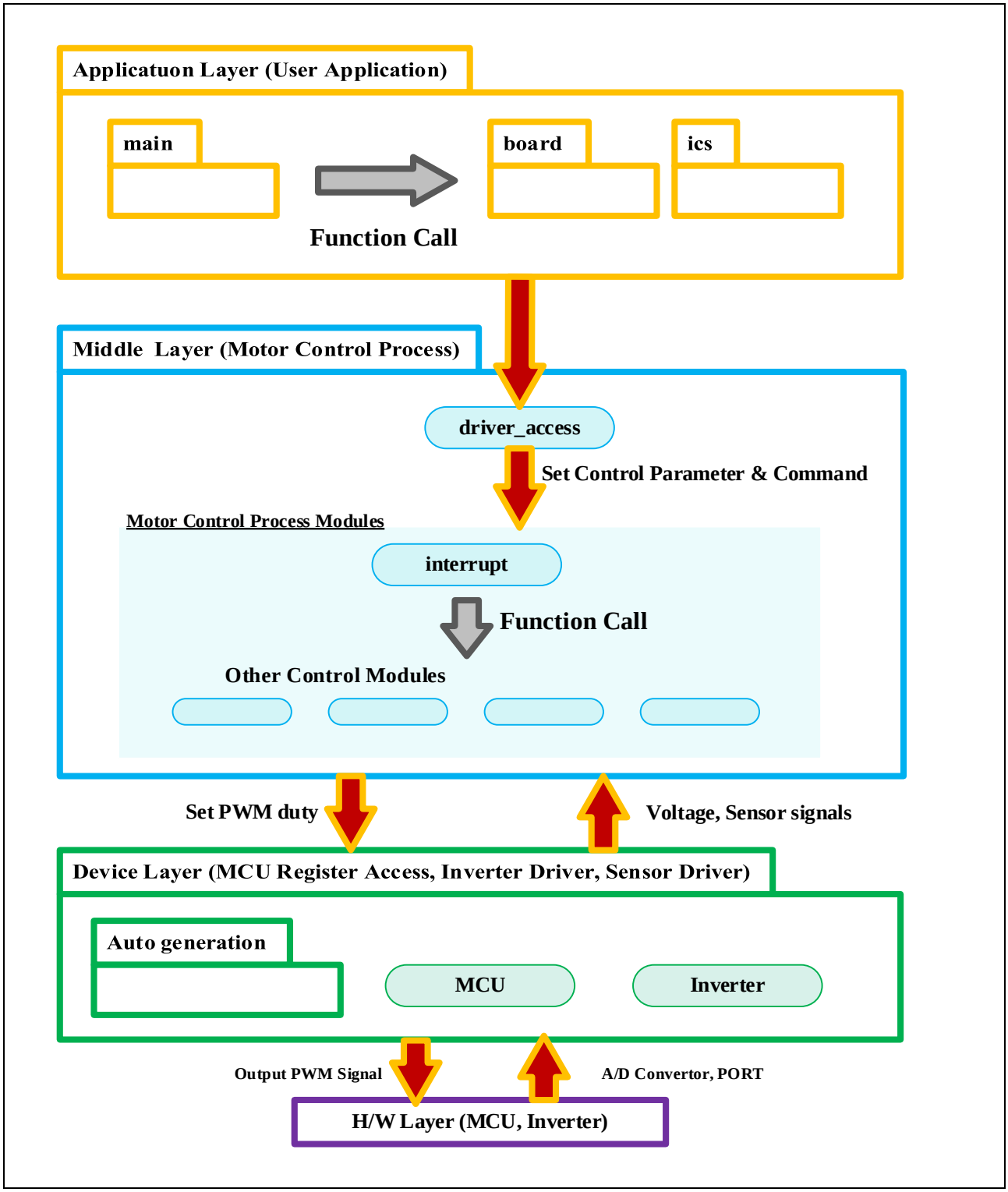


Figure 2-2 Module Configuration of the Sample Programs

3. Descriptions of the control program

The target sample programs of this application note are explained here.

3.1 Contents of control

3.1.1 Motor start/stop

The start and stop operations of motor are controlled by inputs from Renesas Motor Workbench.

3.1.2 A/D Converter

(1) Inverter bus voltage

As shown in Table 3-1 below, the inverter bus voltage is measured and used for modulation factor calculation and over-voltage and under-voltage 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] to 73.5 [V] : 0000H to 0FFFH	AN008

(2) DC link shunt resistor current

The DC link shunt resistor current is measured as shown in the Table 3-2 and used for vector control.

Table 3-2 Conversion Ratio of Offset current

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

(3) U phase, V phase, and W phase voltage

As shown in Table 3-3 below, the U, V and W phase voltages are measured and used for determining zero-crossing of BEMF.

Table 3-3 Conversion Ratio of U, V, and W Phase Voltage

Item	Conversion ratio (U, V, and W phase voltage: A/D conversion value)	Channel
U, V, W phase voltages	0 [V]～73.5 [V]：0000H～0FFFH	AN020,AN007,AN006

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

3.1.3 POEG

(1) Overcurrent detection

When the inverter board's overcurrent detection comparator output is detected as low, the PWM output is forcibly shut off.

3.1.4 Voltage control by PWM

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

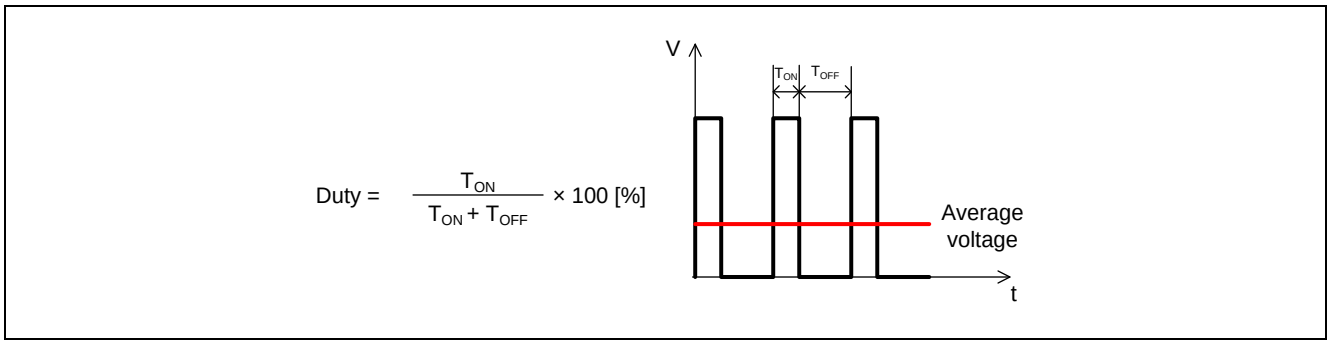


Figure 3-1 PWM Control

Here, modulation factor “m” is defined as follows.

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

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

This modulation factor is set to registers for PWM duty in GPT.

In the target software of this application note, upper arm chopping is used to control the output voltage and speed. Figure 3-2 and Figure 3-3 show an example of output waveforms at upper arm chopping. Non-complementary / complementary PWM can be switched by setting the configuration definition file “r_mtr_config.h”.

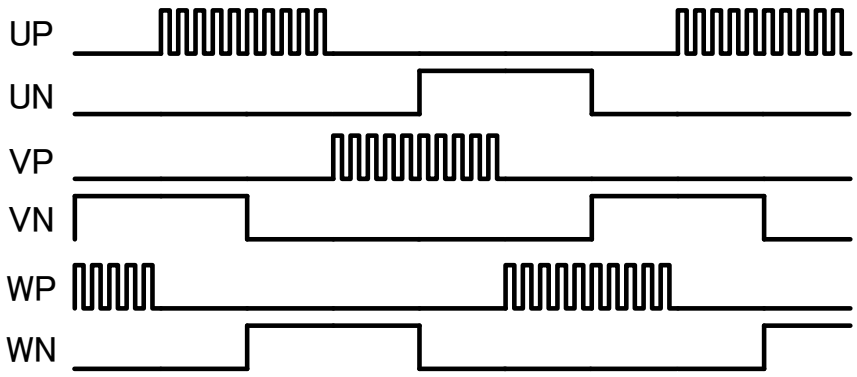


Figure 3-2 Upper Arm Chopping (Non-complementary PWM)

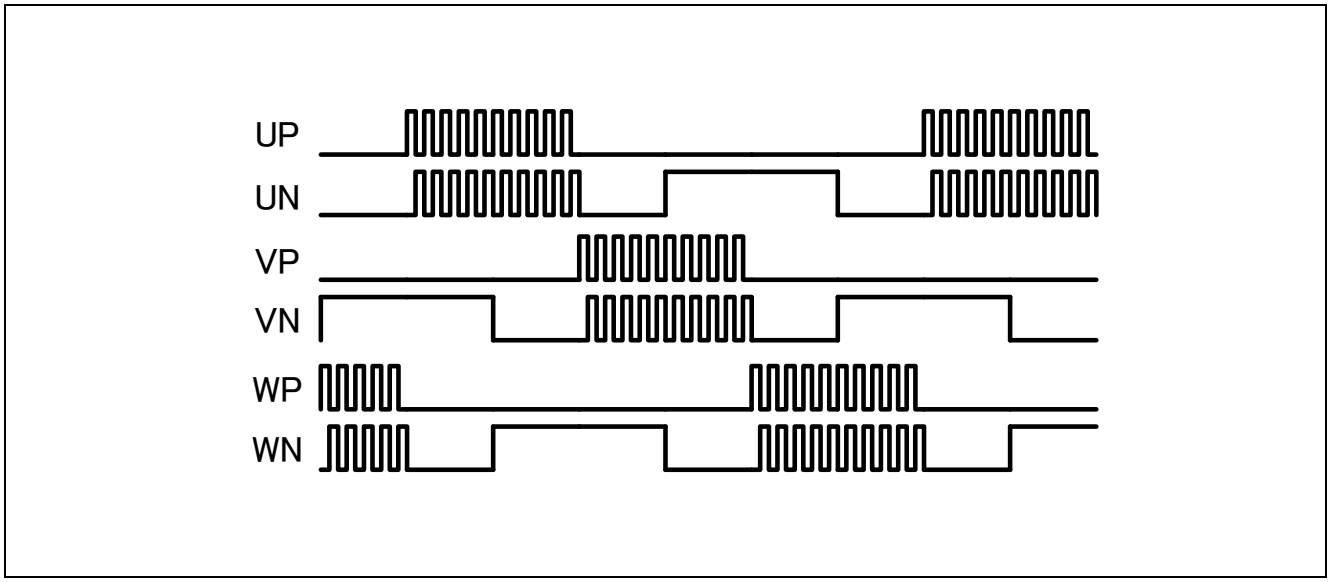
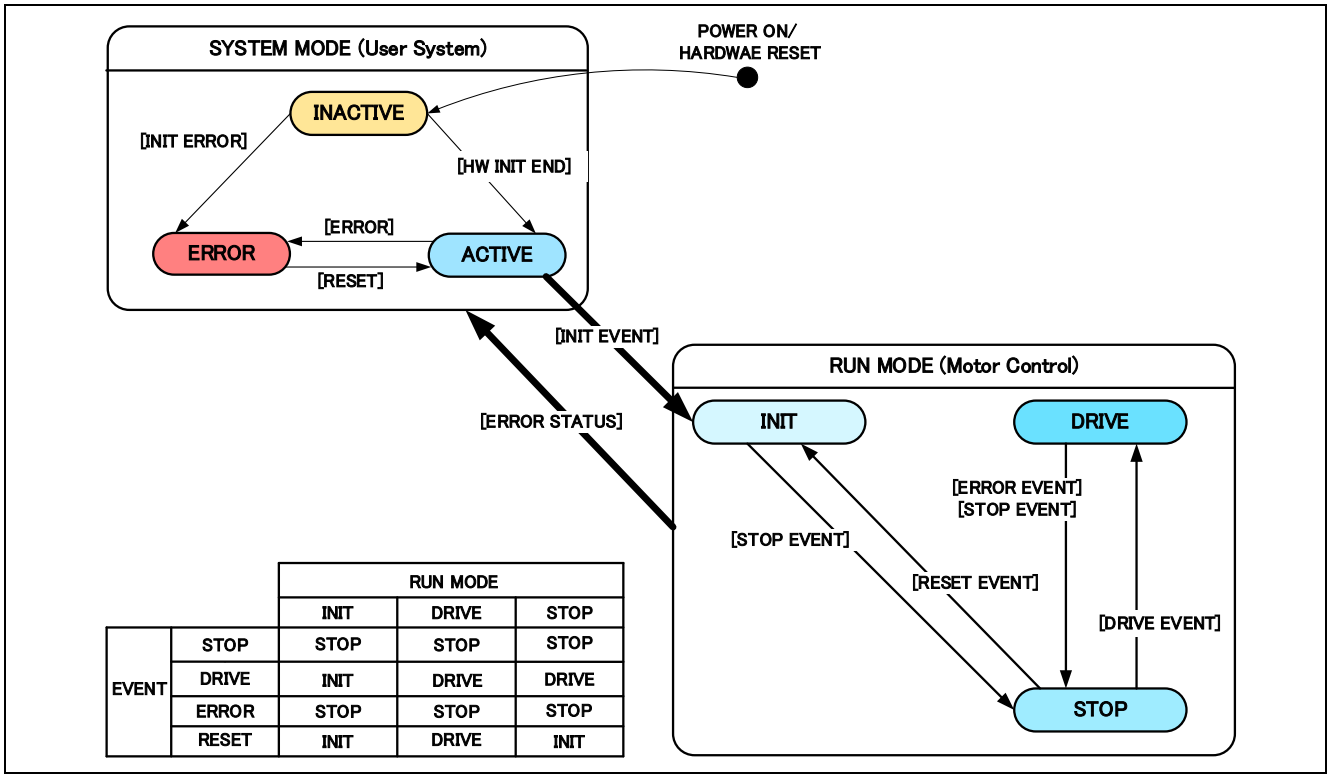


Figure 3-3 Upper Arm Chopping (Complementary PWM)

3.1.5 State transitions

Figure 3-4 shows state transition diagrams of 120-degree conducting control software.



		RUN MODE		
		INIT	DRIVE	STOP
EVENT	STOP	STOP	STOP	STOP
	DRIVE	INIT	DRIVE	DRIVE
	ERROR	STOP	STOP	STOP
	RESET	INIT	DRIVE	INIT

Figure 3-4 State Transition Diagram 120-degree Conducting Control Software

(1) SYSTEM MODE
"SYSTEM MODE" indicates the operating states of the system. "SYSTEM MODE" has 3 states that are motor drive stop (INACTIVE), motor drive (ACTIVE), and abnormal condition (ERROR).
(2) RUN MODE
"RUN MODE" indicates the condition of the motor control. The state is changed by occurrence of "EVENT".
(3) EVENT
"EVENT" indicates the change of "RUN MODE". When "EVENT" occurs, "RUN MODE" changes as shown table in Figure 3-4. Each "Event" is caused by occurrence as shown in Table 3-4.

Table 3-4 List of "EVENT"

"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 DRIVE status list 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 drive start
MTR_OFFSET_CALC_EXE	Execution of current offset detection process
MTR_DRIVE_END	Stopping the drive

3.1.6 Startup method

(1) Hall effect sensor mode

In the Hall effect sensor mode, the output pattern is selected from the initial Hall effect sensor signal. Then, voltage is applied, and state is changed to PI control state. The rotational speed is calculated after second hall effect sensor interruption. The start-up sequence for current control mode and voltage control mode is shown in Figure 3-5 and Figure 3-6 respectively.

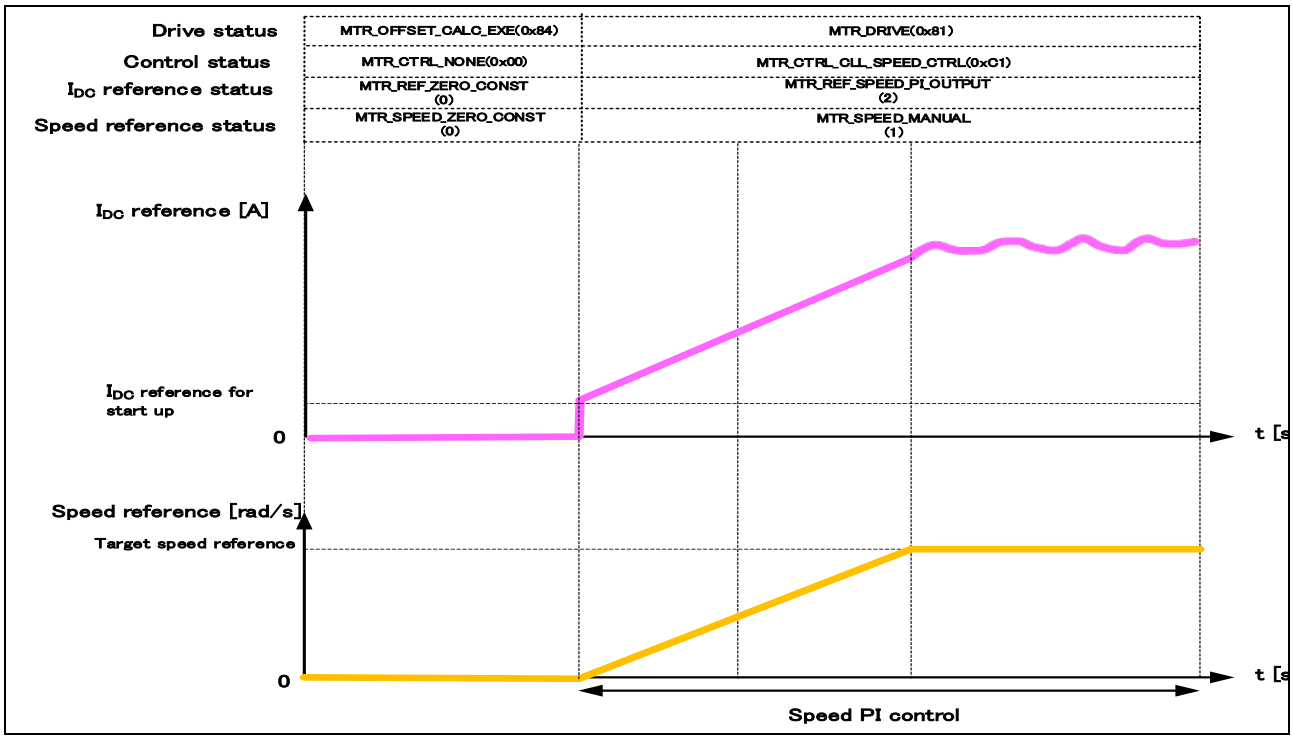


Figure 3-5 Start-up sequence for Hall effect sensor-based current control mode

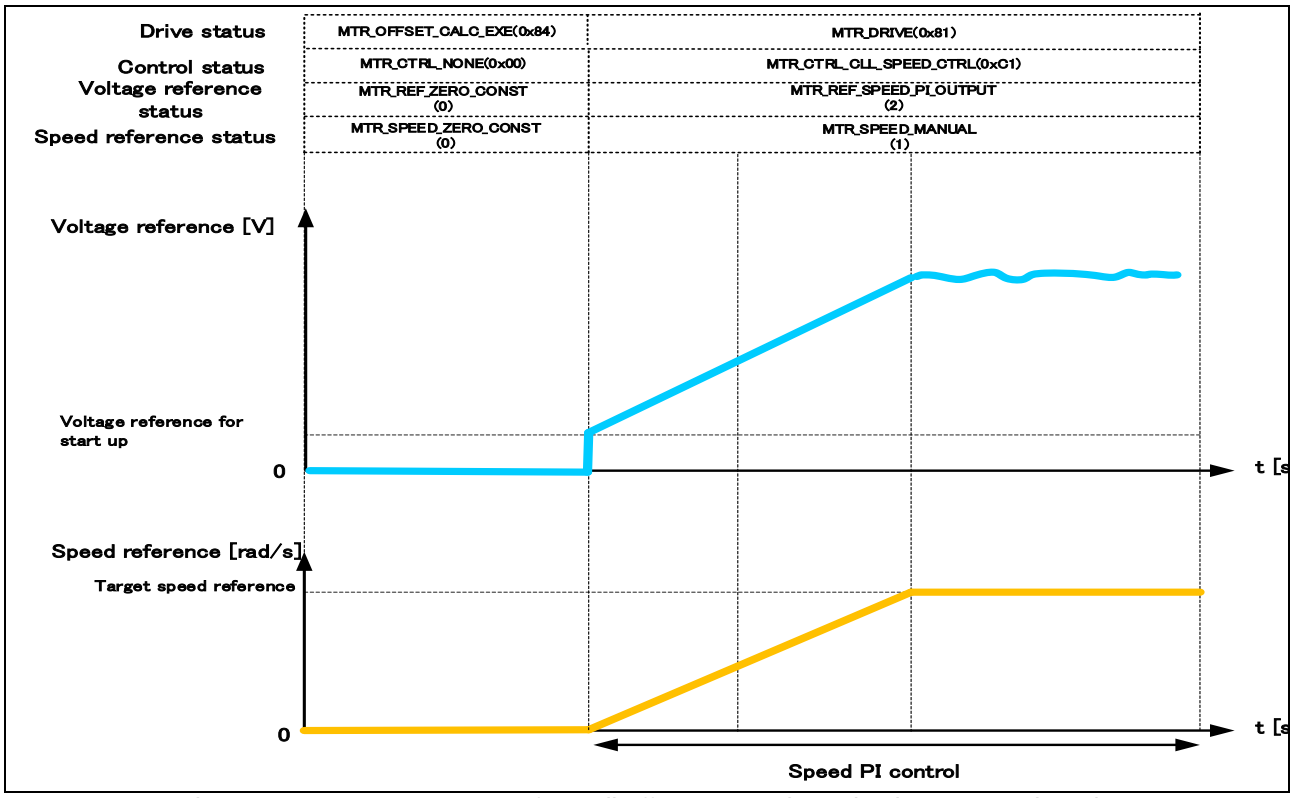


Figure 3-6 Start-up sequence for Hall effect sensor-based voltage control mode

(2) Sensorless mode

In sensorless mode, the position of the magnetic poles is estimated every 60 degrees from induced voltage that is generated from the variation of magnetic flux due to the rotation of the permanent magnet(rotor). However, since the induced voltage is generated by the rotation, at low speed it is not possible to estimate the position of the rotor.

Therefore, the method to generate a rotating magnetic field by forcibly switching conducting pattern in the synchronous speed regardless of the position of rotor, is often used.

Figure 3-7 shows the start-up sequence for sensorless current control mode. In “MTR_DRIVE” the rotor is drawn in. Then, it is driven in open-loop drive and transit to PI control state when the speed of rotation reaches the reference speed to switch to sensorless control.

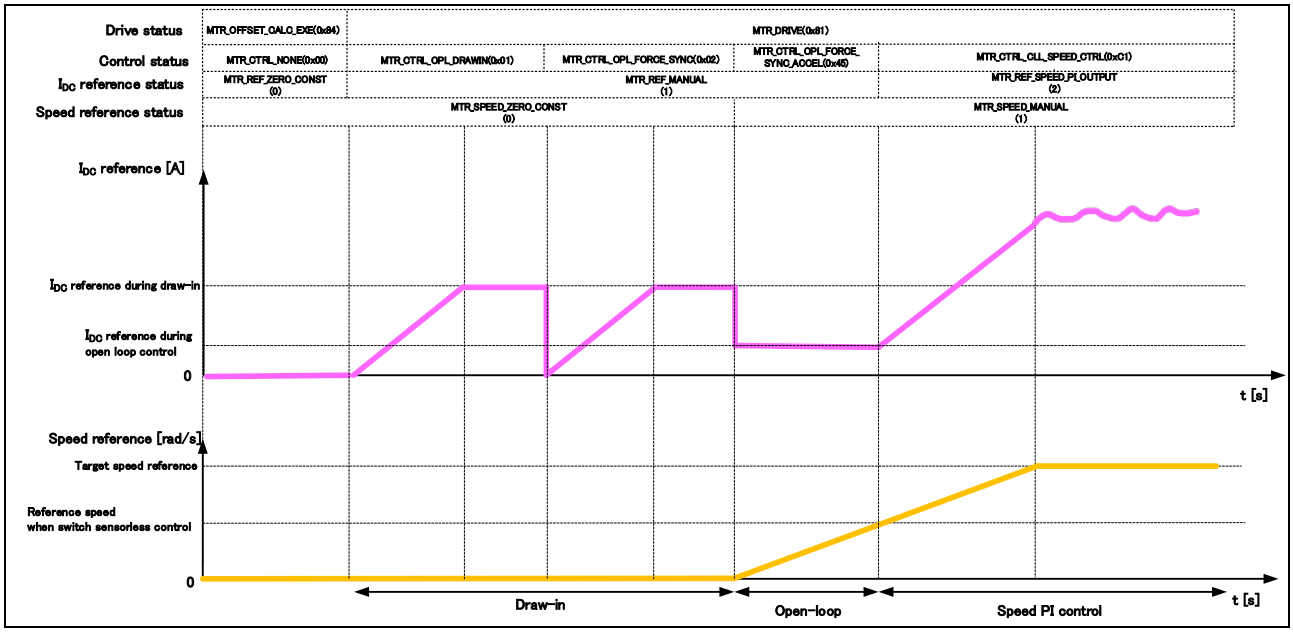


Figure 3-7 Start-up sequence for sensorless current control mode (Start-up with draw-in)

Figure 3-8 shows the start-up sequence for sensorless voltage control mode. In “MTR_DRIVE” the rotor is drawn in. Then, it is driven in open-loop drive, and transit smoothly to PI control state after the induced voltage zero crossing signal is detected 3 times.

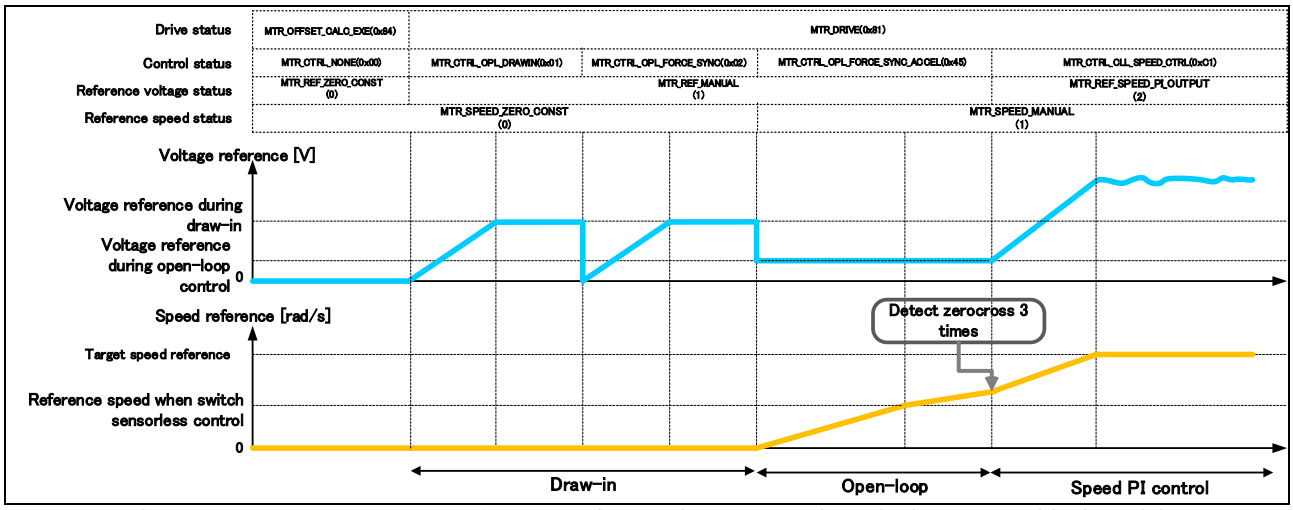
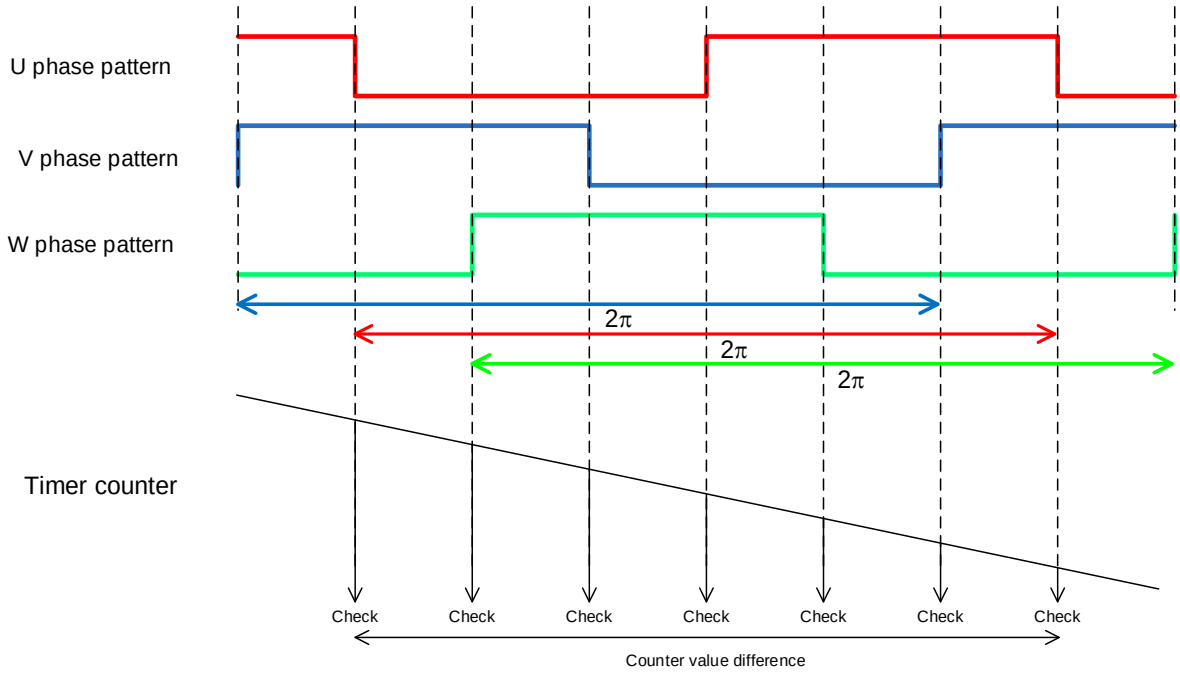


Figure 3-8 Start-up sequence for sensorless voltage control mode (Start-up with draw-in)

3.1.7 Speed control

In this system, a free-running timer counter is used, and rotational speed is calculated by using the difference between the current timer counter value and the timer counter value 2π [rad] earlier. The counter values are obtained when Hall effect sensor interruption is executed in Hall effect sensor control mode, or when zero-crossing of BEMF voltage is detected in sensorless mode.



Motor rotational speed [rad/s] = (2π × frequency of timer) / (difference of timer counts)

Figure 3-9 Method of Calculation for Rotational Speed

3.1.8 Control method

This sample software uses different control systems for current control mode and voltage control mode. In current control mode, the control system is made up of an Auto Speed Regulator (ASR) and an Auto Current Regulator (ACR). In voltage control mode, the control system is only made up of an Auto Speed Regulator (ASR). These controllers are implemented using PI controllers, and the gains are required to be adjusted according to the type of controllers.

3.1.8.1 Control method for current control mode

The block diagram of the entire control system for current control mode is shown in Figure 3-10.

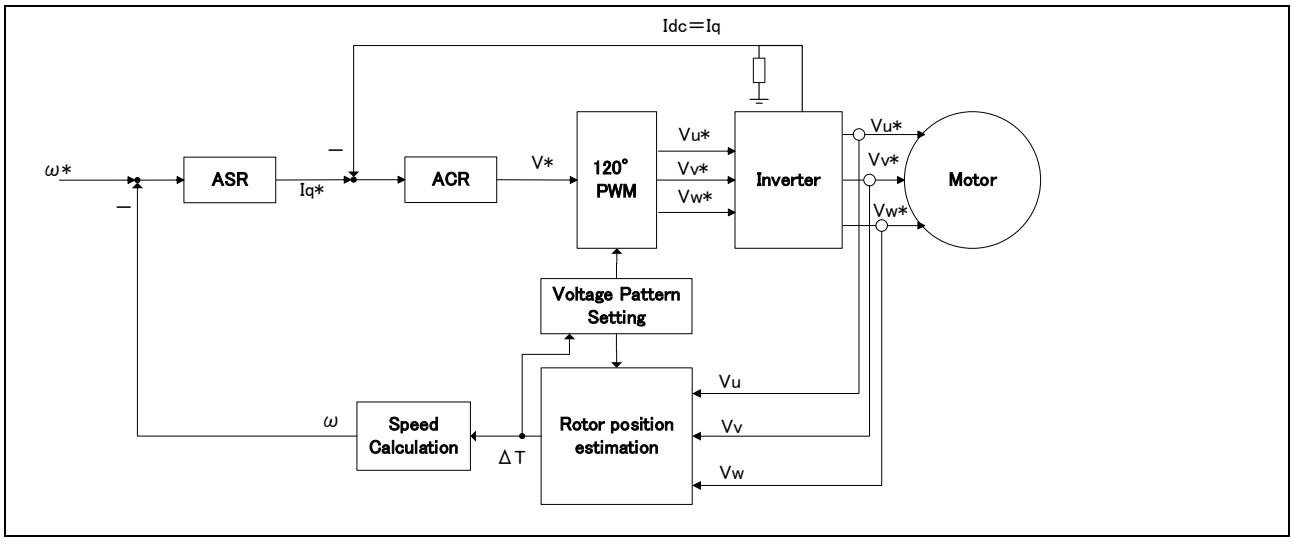


Figure 3-10 Control System Block Diagram (Current control mode)

The speed PI controller and current PI controller are shown as below. The speed PI controller outputs reference current value whereas the current PI controller outputs reference voltage value.

$$i^* = (K_{pASR} + \frac{K_{iASR}}{s})(\omega^* - \omega) \quad \cdots ①$$

i^*: Reference current ω^*: Reference rotational speed ω: Rotational speed

K_{pASR}: Speed PI proportional gain K_{iASR}: Speed PI integral gain s: Laplace operator

$$v^* = (K_{pACR} + \frac{K_{iACR}}{s})(i^* - i) \quad \cdots ②$$

v^*: Reference voltage i^*: Reference current i: Current

K_{pACR}: Current PI proportional gain K_{iACR}: Current PI integral gain s: Laplace operator

3.1.8.2 Control method for voltage control mode

The block diagram of the entire control system for current voltage control mode is shown in Figure 3-11.

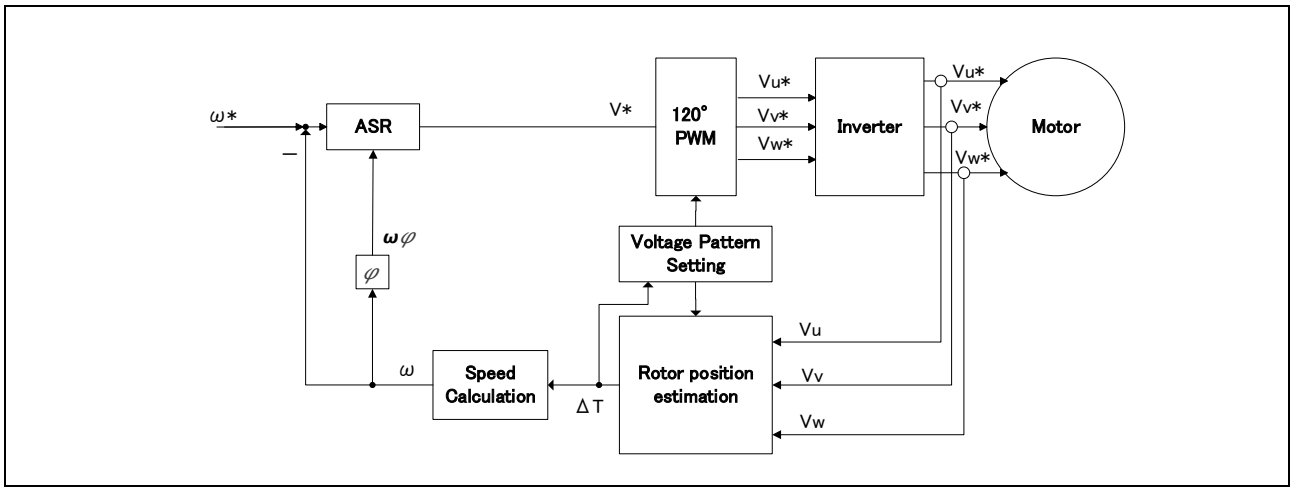


Figure 3-11 Control System Block Diagram (Voltage control mode)

The speed PI controller and current PI controller are shown as below. The speed PI controller outputs reference voltage value.

$$v^* = (K_{pASR} + \frac{K_{iASR}}{s})(\omega^* - \omega) \quad \cdots ③$$

v^*: Reference voltage ω^*: Reference rotational speed ω: Rotational speed
K_{pASR}: Speed PI proportional gain K_{iASR}: Speed PI integral gain s: Laplace operator

3.1.9 Interrupt Processing Specifications

The interrupt processing of this sample software is composed of two cycle interrupts: a carrier cycle interrupt (50[µs]) and a 1-ms cycle interrupt, applicable in both Hall effect sensor mode and sensorless mode. Since the current control system is carried out in the carrier cycle interrupt, the control cycle for the current control system is 50[µs]. On the other hand, since the speed control system is carried out in the 1-ms cycle interrupt, the control cycle for the speed control system is 1[ms].

3.1.9.1 Interrupt processing for current control mode

In current control mode, reference voltage is updated in a 50[µs] interval as the output of current PI controller.

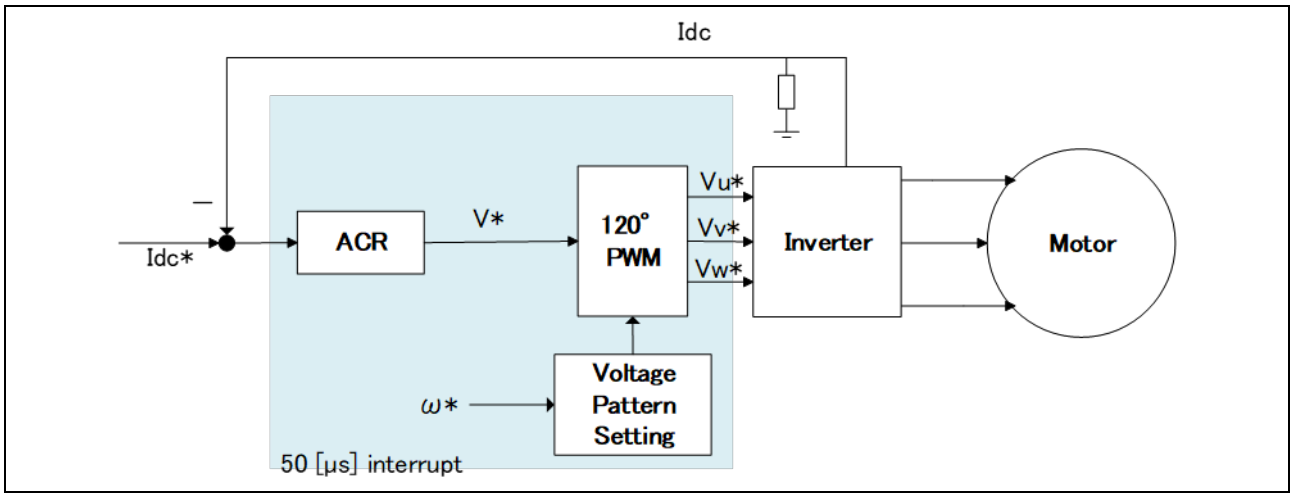


Figure 3-12 Interrupt processing inside control block (open-loop control)

Here is an outline for open-loop drive for current control mode. In open-loop control, only current PI controller is used. Based on the reference voltage obtained as the output of the current PI controller, voltage is applied to the rotor according to the commutation pattern. Each commutation pattern outputs for an electrical angle of 60 degrees, and the commutation patterns are sequentially switched according to the reference speed.

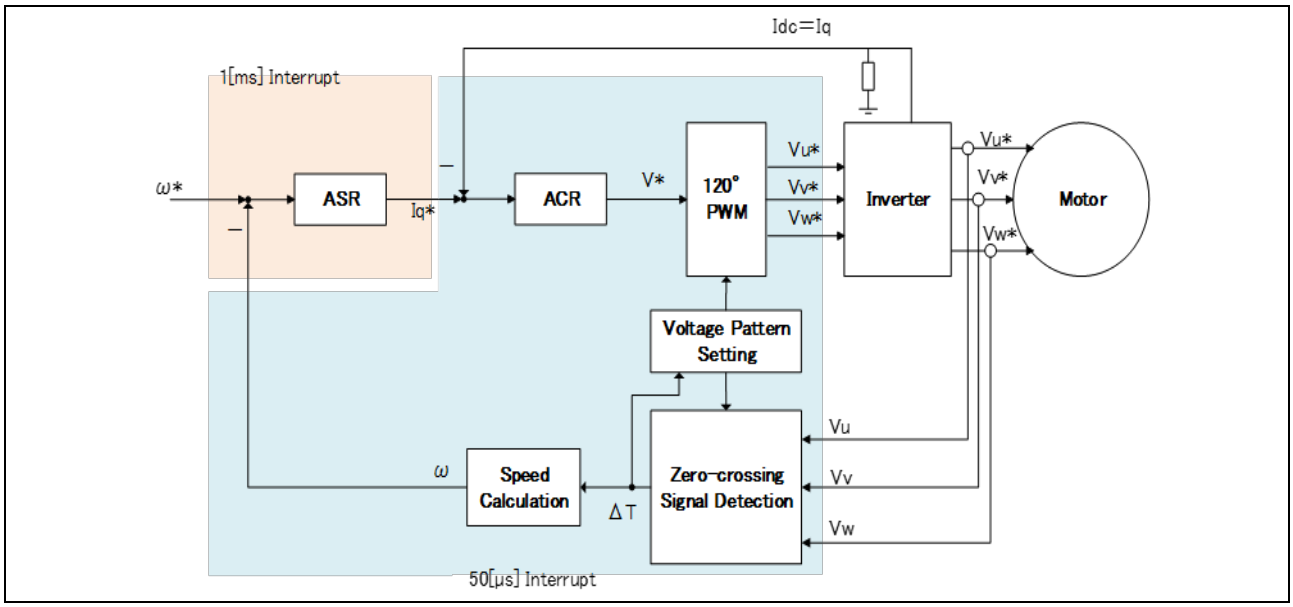


Figure 3-13 Interrupt processing inside control block (sensorless closed loop control)

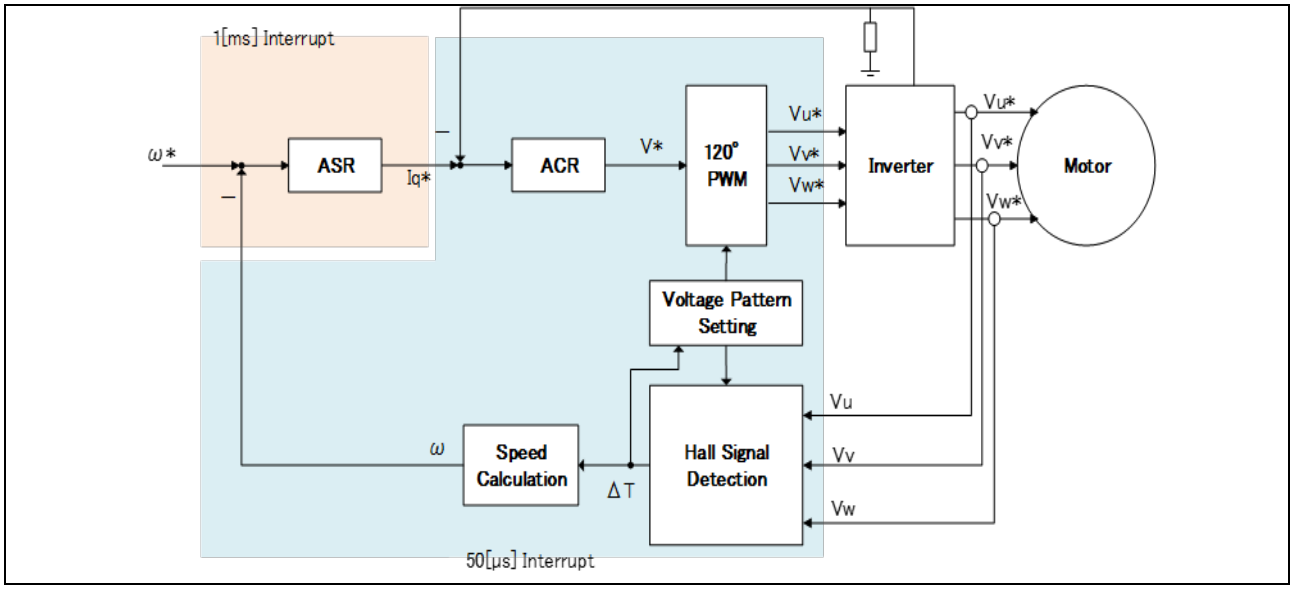


Figure 3-14 Interrupt processing inside control block (Hall effect sensor-based closed loop control)

Here is an outline for closed loop drive for current control mode. In closed loop control, the rotational speed is calculated based on the rotor position, where it is detected using the zero-crossing detection signal in sensorless mode and Hall effect sensor signal in Hall effect sensor mode. The speed error is input into the speed PI controller and the output is reference current. The current PI controller outputs reference voltage which is then applied to the rotor according to the commutation pattern.

3.1.9.2 Interrupt processing for voltage control mode

In voltage control mode, reference voltage is updated in a 1[ms] interval as the output of speed PI controller.

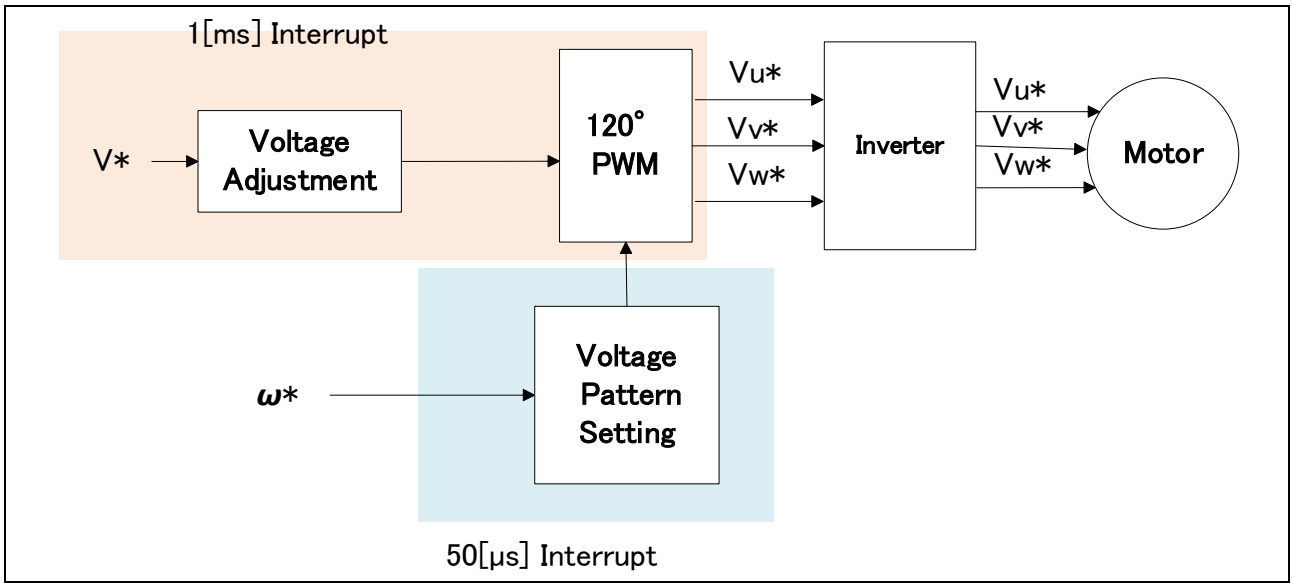


Figure 3-15 Interrupt processing inside control block (open-loop control)

Here is an outline for open-loop drive for voltage control mode. In open-loop control, speed PI controller is not used. Therefore, constant voltage is applied to the rotor according to the commutation pattern. Each commutation pattern outputs for an electrical angle of 60 degrees, and the commutation patterns are sequentially switched according to the reference speed.

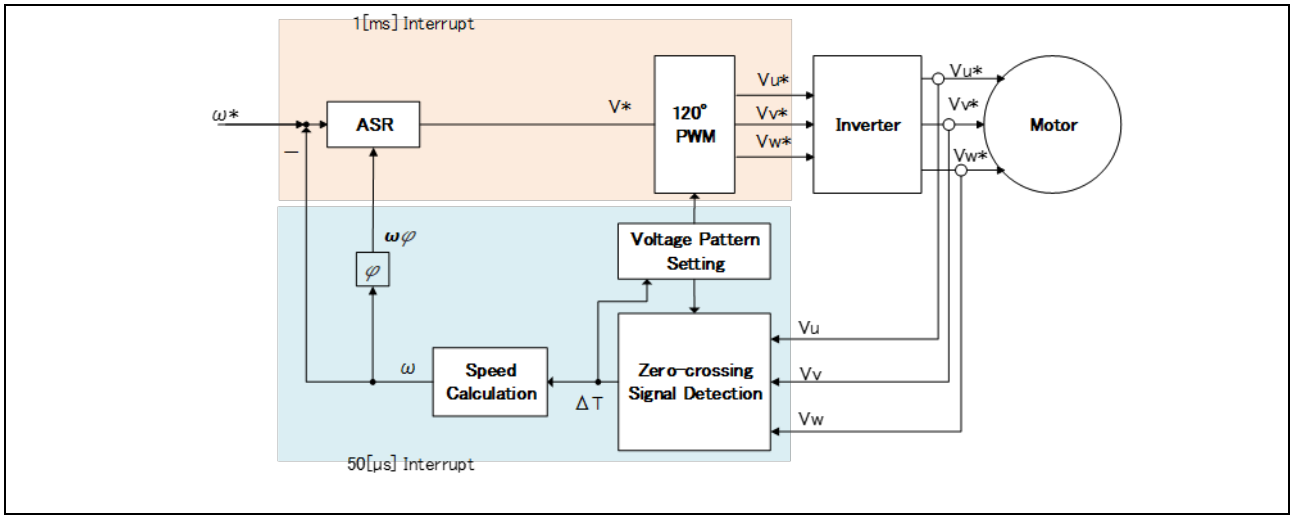


Figure 3-16 Interrupt processing inside control block (sensorless closed loop control)

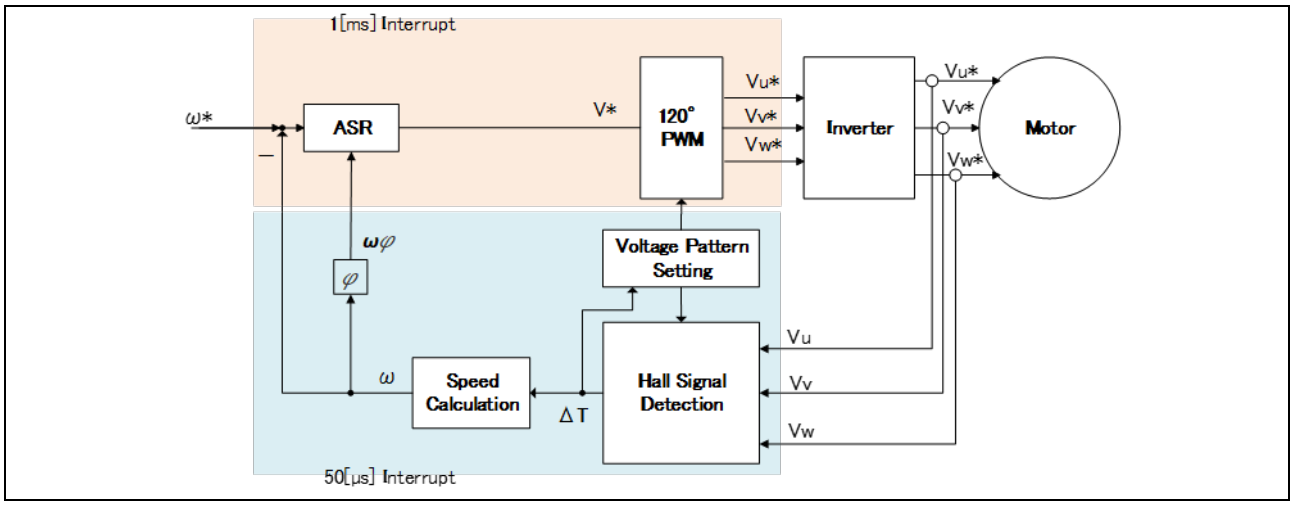


Figure 3-17 Interrupt processing inside control block (Hall effect sensor-based closed loop control)

Here is an outline for closed loop drive for voltage control mode. In closed loop control, the rotational speed is calculated based on the rotor position, where it is detected using the zero-crossing detection signal in sensorless mode and Hall effect sensor signal in Hall effect sensor mode. The speed error is input into the speed PI controller and the output is reference voltage, which is then applied to the rotor according to the commutation pattern.

3.1.10 DC link current, bus voltage and phase voltage measurement method

This section explains the method used in this sample program to measure DC link current, bus voltage and phase voltage

3.1.10.1 Timing to measure DC link current, bus voltage and phase voltage

In this sample program, the following functions of RA2T1 is used to measure DC link current, bus voltage and phase voltage.

- GPT
Complementary PWM mode 2 and A/D converter start request A
- A/D converter
Group Scan Mode

In this program, the DC link current, bus voltage, and phase voltage are measured at the peak of the carrier cycle, and motor control calculations are performed at the underflow interrupt of the carrier cycle. Figure 3-18 shows the A/D conversion timing and carrier interrupt timing.

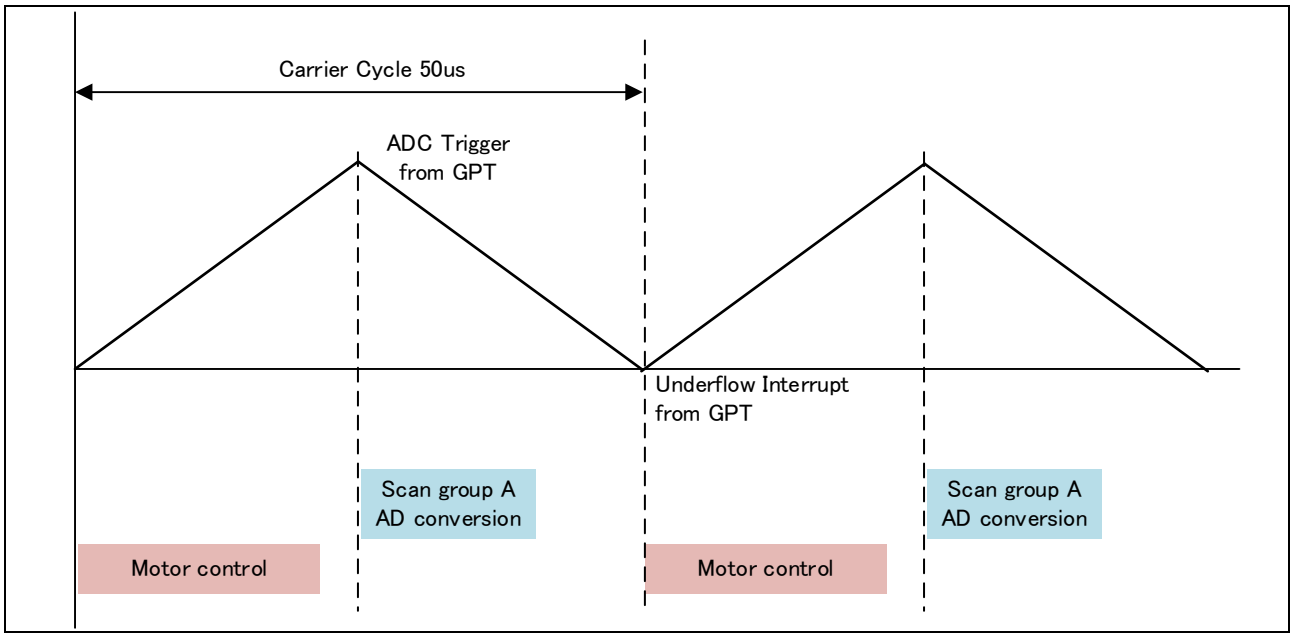


Figure 3-18 Measurement timing of bus voltage and phase voltage

3.1.11 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.

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

- Software 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.

- Timeout error
The timeout counter is monitored at the timeout monitoring cycle. When pattern switching by Hall effect sensor interruption in Hall effect sensor control mode or zero-crossing of induced voltage in sensorless control mode don’t happen for a timeout period, voltage output is stopped.

- Pattern error
The output voltage pattern is monitored at the pattern monitoring cycle. When unexpected pattern is detected in voltage pattern set from Hall effect sensor in Hall effect sensor control mode or induced voltage in sensorless control mode, voltage output is stopped.

Table 3-6 System Protection Function Settings

Types of error	Threshold	
Hardware overcurrent error	Overcurrent limit [A]	21.4
Software overcurrent error	Overcurrent limit [A]	3.955
	Monitoring cycle [μs]	50
Overvoltage error	Overvoltage limit [V]	48
	Monitoring cycle [μs]	1000
Undervoltage error	Undervoltage limit [V]	8
	Monitoring cycle [μs]	1000
Rotational speed error	Speed limit [rpm]	3262.5
	Monitoring cycle [μs]	1000
Timeout error	Timeout value [ms]	226 [Sensorless mode] 113 [Hall-effect sensor mode]

3.1.12 PU system

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	2π× maximum speed [rpm] × 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/rad]
	BEMF constant	Voltage / Angular frequency	[V·s/rad]
	Inertia	BEMF constant × current × (number of pole pairs / angular frequency)^2	[kgm^2/(rad^2)]
Current control	Kp	Resistance	[Ω]
	Kidt	Resistance	[Ω]
Speed control	Kp	Current / angular frequency [Current control mode] Voltage / angular frequency [Voltage control mode]	[A/(rad/s)] [V/(rad/s)]
	Kidt	Current / angular frequency [Current control mode] Voltage / angular frequency [Voltage control mode]	[A/(rad/s)] [V/(rad/s)]

3.2 Function specifications of 120-degree conducting control software

Lists of functions used in this control program are shown below. Functions not used in this system are not listed.

Table 3-8 List of Functions “r_mtr_board.c”

File	Function	Process
r_mtr_board.c	RM_MTR_BoardLedContrl input: uint8_t u1_motor_status :: A motor status uint8_t u1_system_status :: A system status output: none	Control LEDs

Table 3-9 List of Functions “ICS2_RA2T1.o”

File	Function	Process
ICS2_RA2T1.o	ics2_watchpoint input: none output: none	Call a function to transfer data Must be called at intervals of 300us or more
	ics2_init input: uint16_t port :: ports used by SCI uint8_t level :: an interrupt level uint8_t speed :: a communication speed uint8_t mode :: a communication mode output: none	Initialize communications
	ics2_version input: none output: uint32_t :: a version number	Get a version number

Table 3-10 List of Functions “r_mtr_ics.c”

File	Function	Process
r_mtr_ics.c	R_MTR_SetIcs input: none output: none	Prepare RMW communication
	R_MTR_SetCOMVariables input: none output: none	Pre-process to set control variables - input values of command variables to RMW variables - input values of RMW variables to RMW buffer variables
	R_MTR_ICSVariablesInit input: none output: none	Initialize com variables
	mtr_limit_float [inline function] input: float f4_value :: a target value int32_t s4_max :: a maximum limit int32_t s4_min :: a minimum limit output: float :: a limited target value	Limit a value between a maximum and a minimum

Table 3-11 List of Functions “mtr_main.c”

File	Function	Process
mtr_main.c	mtr_main Input: none Output: none	- Call a function to initialize hardware - Initialize Renesas Motor Workbench Communication - Call a function to initialize a motor control system - Call a function to transition states and execute events - Call a process bus to wait voltage stability - Main process ⇒Call a process for user interface ⇒Call a process to control LEDs ⇒Call a function to clear a watchdog timer
	ics_ui input: none output: none	User interface that uses Renesas Motor Workbench - Input values of command variables to RMW variables - Change motor status depending on an input event
	system_init input: none output: int :: an execution result	Initialize the system for the main process
	system_term input: none output: none	Process to shutdown the system
	software_init input: none output: none	Initialize variables that be used in the main process

Table 3-12 List of Functions “r_mtr_ADC.c”

File	Function	Process
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 :: an error number	Initialize ADC with fast-mode
	R_MTR_ADC_WaitConversion input: adc_ctrl_t * p_ctrl :: ADC information uint16_t timeout_count :: a time-out count output: fsp_err_t :: an error number	Wait for an A/D conversion
	R_MTR_ADC_SetChannels input: adc_ctrl_t * p_ctrl :: ADC information uint32_t scan_mask_a :: a bit-mask for A/D channel select register A uint32_t scan_mask_b :: a bit-mask for A/D channel select register B output: fsp_err_t :: an error number	[Hall effect sensor control mode] Set ADC channels

Table 3-13 List of Functions “r_mtr_AGT.c”

File	Function	Process
r_mtr_ AGT.c	R_MTR_AGT_ReadTimerCount_32Bit input: timer_ctrl_t *p_ctrl :: Timer information uint32_t * p_value :: a pointer to store the counter value output: fsp_err_t :: an error number	Get AGT counter value

Table 3-14 List of Functions “r_mtr_GPT_THREE_PHASE.c”

File	Function	Process
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 :: an error number	Initialize a 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 :: An error number	Set duty cycles for a 3-phase timer module in complementary PWM mode 2
	R_MTR_GPT_THREE_PHASE_EnableOutputAll input: three_phase_ctrl_t * const p_ctrl :: 3 Phase Control information output: fsp_err_t :: an error number	Enables all outputs of a 3-phase timer module in complementary PWM
	R_MTR_GPT_THREE_PHASE_DisableOutputAll input: three_phase_ctrl_t * const p_ctrl :: 3 Phase Control information output: fsp_err_t :: an error number	Disables all outputs of a 3-phase timer module in complementary PWM
	R_MTR_GPT_THREE_PHASE_PatternOutput input: three_phase_ctrl_t * const p_ctrl :: 3 Phase Control information uint16_t u2_pattern :: voltage pattern uint8_t *p_u1_err_flg :: a pointer to store whether a pattern error exists output: fsp_err_t :: an error number	Set registers to enable or disable output PWMs

Table 3-15 List of Functions “r_mtr_POEG.c”

File	Function	Process
r_mtr_POEG.c	R_MTR_POEG_SoftwareRelease input: poeg_ctrl_t * const p_ctrl :: POEG control information uint16_t timeout_count :: a time-out count output: fsp_err_t :: an error number	Initialize a POEG module
	R_MTR_POEG_RecoverForcedShutdown input: poeg_ctrl_t * const p_ctrl :: POEG control information uint16_t timeout_count :: a time-out count uint8_t * p_is_shutdown :: a pointer to store a shutdown state output: fsp_err_t :: an error number	Recover a POEG module from a forced shutdown state

Table 3-16 List of Functions “rm_mtr_carrier.c” (1/3)

File	Function	Process
rm_mtr_carrier.c	mtr_carrier_cyclic input: timer_callback_args_t * p_args :: a pointer to interrupt handler arguments output: none	Carrier interrupt handler Periodic timer interrupt (called by GPT) Period: 50[µs] - Obtain an inverter bus voltage - Call a function to process a carrier interrupt - Call a process for communication
	mtr_carrier_vdc_get input: int32_t * p_s4_ad_vdc :: a pointer to store VDC int16_t * p_s2_ad_vdc_bit :: a pointer to store an A/D conversion value output: none	Detect an inverter bus voltage (VDC) - Calculate an inverter bus voltage from an A/D conversion value
	mtr_carrier_sequence [inline function] input: none output: none	Function to process a carrier interrupt
	mtr_carrier_idc_get input: int32_t * p_s4_ad_idc :: a pointer to store IDC int16_t * p_s2_ad_idc_bit :: a pointer to store an A/D conversion value uint32_t u4_ad_offset_idc :: an offset of IDC output: none	Detect a DC link current (IDC) - Calculate IDC from an A/D conversion value
	mtr_carrier_lim_sw_over_current [inline function] input: int32_t s4_lim_over_current :: a threshold of over-current int32_t s4_ad_idc :: IDC output: uint16_t :: an error status	Error check for software over-current
	mtr_current_pi_ctrl [inline function] input: st_mtr_acr_t * p_st_acr :: a pointer to an ACR structure int32_t s4_idc_err :: an error of the current int32_t * p_s4_ref_v_ctrl :: a pointer to a reference voltage int32_t s4_lim_v :: a minimum limit of the voltage output: none	PI control process
	mtr_pi_run_acr [inline function] input: st_mtr_pi_t * p_st_pi :: a pointer to a PI control sturcture int32_t s4_err :: an error const uint8_t u1_kp_q :: Q value of a proportional gain const uint8_t u1_q_err_shift :: a shift value of a error const uint8_t u1_q_kidt_shift :: Q value of a integral gain const uint8_t u1_q_intg_shift :: a shift value of an integer part output: int32_t :: PI output	Current PI control process
	mtr_carrier_set_pwm [inline function] input: uint16_t u2_pattern :: a voltage pattern int8_t s1_dir :: a direction of rotation uint16_t * p_u2_pwm_duty :: a pointer to duty values int32_t s4_ref_v_ctrl :: a reference voltage int16_t s2_ad_vdc_bit :: an A/D conversion value of VDC output: none	Set up PWM output

Table 3-17 List of Functions “rm_mtr_carrier.c” (2/3)

File	Function	Process
rm_mtr_carrier.c	mtr_duty_calc [inline function] input: int32_t s4_ref_v :: a reference voltage int16_t s2_ad_vdc_bit :: an A/D conversion value of VDC output: int16_t :: a duty value	Calculate a PWM duty value
	mtr_carrier_startup_sequence [inline function] input: none output: none	Prepare to process carrier interrupts
	mtr_calib_current_offset_dc [inline function] input: st_mtr_120_control_t * p_st_120 :: a pointer to a structure of motor control main data output: none	Calibrate an offset of a DC link current
	mtr_abs [inline function] input: int32_t s4_value :: an input value output: int32_t :: an output value	Calculate an absolute value
	mtr_limit [inline function] input: int32_t s4_value :: an input value int32_t s4_max :: a maximum value int32_t s4_min :: a minimum value output: int32_t :: an output value	[Current control mode] Limit a value
	mtr_limit_abs [inline function] input: int32_t s4_value :: an input value int32_t s4_limit_value :: a limit value output: int32_t :: an output value	Limit a value with an absolute value
	mtr_delay_interrupt input: timer_callback_args_t * p_args :: a pointer to interrupt hander arguments output: none	Delay Timer interrupt - Set a voltage pattern
	mtr_carrier_voltage_get input: int16_t * p_s2_ad_vu :: a pointer to U phase voltage int16_t * p_s2_ad_vv :: a pointer to V phase voltage int16_t * p_s2_ad_vw :: a pointer to W phase voltage int16_t * p_s2_ad_vn :: a pointer to neutral voltage output: none	[Sensorless mode] Get the voltage
	mtr_detect_zerocross [inline function] input:st_mtr_sensorless_carrier_t * p_st_less :: a pointer to a structure for sensorless control variables st_mtr_ol2cl_t * p_st_ol2cl :: a pointer to a structure for sensorless control transition uint32_t * p_u4_cnt_timeout :: a pointer to a time-out counter int8_t s1_dir :: a rotational direction uint16_t u2_v_pattern :: a voltage pattern output: uint8_t :: a BEMF signal	[Sensorless mode] Estimate a position of a rotor from zero-crossing of an induced voltage
	mtr_drive_openloop_less [inline function] input: none output: none	[Sensorless mode] Process for an open-loop control
	mtr_set_angle_shift [inline function] input: none output: none	[Sensorless mode] - Calculatie a delay count after zero-crossing - Calculate a zero-crossing filter count

Table 3-18 List of Functions “rm_mtr_carrier.c” (3/3)

File	Function	Process
rm_mtr_carrier.c	mtr_start_delay_timer [inline function] input: uint32_t u4_delay_count :: a delay count output: none	[Sensorless mode] Start Delay Timer
	mtr_openloop_pattern_set [inline function] input: none output: uint8_t :: a voltage pattern	[Sensorless mode] Set a voltage pattern for open-loop
	mtr_hall_u_interrupt input: external_irq_callback_args_t * p_args :: a pointer to interrupt handler arguments output: none	[Hall effect sensor mode] Process for Hall effect sensor interrupt - Call a function for Hall effect sensor interrupt - Set a voltage pattern depending on a signal from Hall effect sensors - Calculate a speed
	mtr_hall_v_interrupt input: external_irq_callback_args_t * p_args :: a pointer to interrupt handler arguments output: none	Hall effect sensor mode] Process for Hall effect sensor interrupt - Call a function for Hall effect sensor interrupt - Set a voltage pattern depending on a signal from Hall effect sensors - Calculate a speed
	mtr_hall_w_interrupt input: external_irq_callback_args_t * p_args :: a pointer to interrupt handler arguments output: none	[Hall effect sensor mode] Process for Hall effect sensor interrupt - Call a function for Hall effect sensor interrupt - Set a voltage pattern depending on a signal from Hall effect sensors - Calculate a speed
	mtr_hall_signal_set [inline function] input: none output: none	[Hall effect sensor mode] Set a voltage pattern depending on a signal from Hall effect sensors
	mtr_hall_signal_process input: none output: none	[Hall effect sensor mode] Process for Hall effect sensor interrupt - Call a function for Hall effect sensor interrupt - Set a voltage pattern depending on a signal from Hall effect sensors - Calculate a rotational speed
	mtr_speed_calc [inline function] input: none output: none	Calculate of an rotational speed
	mtr_set_chopping_pattern [inline function] input: uint16_t u2_pattern :: a voltage pattern output: none	Set a chopping pattern

Table 3-19 List of Functions “rm_mtr_ctrl_gain.o”

File	Function	Process
rm_mtr_ctrl_gain.o	R_MTR_CtrlGainVoltageMode input: st_mtr_ctrl_gain_voltage_mode_t * p_st_gain_v_buf :: A pointer to a voltage control gain structure const st_mtr_design_parameter_t * p_st_ctrl_param :: A pointer to a design parameter structure output: none	[Voltage control mode] Process to design control gains
	R_MTR_CtrlGainCurrentMode input: st_mtr_ctrl_gain_current_mode_t * p_st_gain_i_buf :: A pointer to a current control gain structure const st_mtr_design_parameter_t * p_st_ctrl_param :: A pointer to a design parameter structure output: none	[Current control mode] Process to design control gains

Table 3-20 List of Functions “rm_mtr_120.c”

File	Function	Process
rm_mtr_120.c	R_MTR_120MotorDefaultInit input: st_mtr_120_control_t * p_st_120 :: a pointer to a structure of motor control main data output: none	Initialize control variables
	R_MTR_120MotorReset input: st_mtr_120_control_t * p_st_120 :: a pointer to a structure of motor control main data output:none	Reset control values

Table 3-21 List of Functions “rm_mtr_ol2cl_ctrl.c”

File	Function	Process
rm_mtr_ol2cl_ctrl.c	R_MTR_OL2CLCtrl input: st_mtr_ol2cl_t * p_st_ol2cl :: a pointer to a structure for sensorless control transition int16_t *s2_v_uvwn_ad :: a pointer to U phase voltage int8_t s1_dir :: a rotational direction uint32_t u4_cnt_ol_speed :: a count for pattern changes in open-loop control mode uint8_t u1_flag_pattern_change :: a flag for zero-crossing detection uint32_t u4_ol_pattern_period :: a period for pattern changes ino pen-loop uint8_t u2_ol_v_pattern :: a voltage pattern in open-loop output: none	[Sensorless mode] Transition from open-loop control mode tosensorless control one - Compare 3-phase voltages with a threshold voltage - Count for zero-crossing detections in open-loop control mode - Detect the rotor phase (lead/lag)
	mtr_get_bemf_voltage [inline function] input: int8_t s1_dir :: a rotational direction uint8_t u2_ol_v_pattern :: a voltage pattern in open-loop control mode int16_t * s2_v_uvwn_ad :: a pointer to U phase voltage output: int16_t :: an induced voltage	Get an induced voltage
	mtr_openloop_phase_ctrl [inline function] input: st_mtr_ol2cl_t * p_st_ol2cl :: a pointer to a structure for sensorless control transition int8_t s1_dir :: an rotational direction uint8_t u2_ol_v_pattern :: a voltage pattern in open-loop control mode int16_t *s2_v_uvwn_ad :: a pointer to U phase voltage output:none	[Sensorless mode] Detect the rotor phase (lead/lag)
	mtr_abs [inline function] input:int16_t s2_value :: an input value output:int16_t :: an output value	[Sensorless mode] Calculate an absolute value
	R_MTR_BemfSignalFilter input: st_mtr_ol2cl_t * p_st_ol2cl :: a pointer to a structure for sensorless control transition uint32_t u4_cnt_ol_speed :: a count for pattern changes in open-loop control mode uint8_t * p_u1_pre_bemf_signal :: a pointer to the previous estimated hall pattern signal uint16_t u2_temp_signal :: estimated hall pattern signal uint8_t u1_flag_ol_pattern_change :: : flag for a pattern change in open-loop control mode output: int16_t :: estimated hall pattern signal	[Sensorless mode] Filter noise signals during a voltage pattern change to estimate hall pattern signals
	R_MTR_SetSpeedRefOL2CL input: st_mtr_ol2cl_t * p_st_ol2cl :: a pointer to a structure for sensorless control transition int8_t s1_dir :: a rotational direction int32_t * s2_ref_speed_rad_ctrl :: a pointer to areference rotational speed output: none	[Sensorless mode] Set a reference rotational speed for open-loop

Table 3-22 List of Functions “rm_mtr_speed.c” (1/2)

File	Function	Process
rm_mtr_speed.c	mtr_speed_cyclic input: timer_callback_args_t *p_args :: a pointer to interrupt handler arguments output: none	Speed Control Interrupt Handler Cycle timer interrupt (called by GPT) Cycle: 1 [ms] - Set the drive mode stauts - Call a process for speed PI control - Call a pre-process to calibrate a current offset - Call a function to set a voltage pattern in a draw-in state - Call a process to set command values - Call a process to monitor erros
	mtr_speed_drive_ctrl_less input: st_mtr_120_control_t * p_st_120 :: a pointer to a structure of motor control main data output: int8_t :: a control status	[Sensorless mode] Control the motor speed drive
	mtr_speed_process_opl_ctrl_less input: st_mtr_120_control_t * p_st_120 :: a pointer to a structure of motor control main data uint8_t u1_state_ctrl_mode :: a control status output: int8_t :: a control status	[Sensorless mode] Control the motor speed in open-loop control mode
	mtr_speed_process_cll_ctrl_less input: st_mtr_120_control_t * p_st_120 :: a pointer to a structure of motor control main data uint8_t u1_state_ctrl_mode :: a control status output: int8_t :: a control status	[Sensorless mode] Control the motor speed in closed-loop control mode
	mtr_speed_drive_ctrl_hall input: st_mtr_120_control_t * p_st_120 :: a pointer to a structure of motor control main data output: int8_t :: a control status	[Hall effect sensor mode] Control the motor speed drive
	mtr_speed_process_opl_ctrl_hall input: st_mtr_120_control_t * p_st_120 :: a pointer to a structure of motor control main data uint8_t u1_state_ctrl_mode :: a control status output: int8_t :: a control status	[Hall effect sensor mode] Control the motor speed in open-loop control mode
	mtr_speed_process_cll_ctrl_hall input: st_mtr_120_control_t * p_st_120 :: a pointer to a structure of motor control main data uint8_t u1_state_ctrl_mode :: a control status output: int8_t :: a control status	[Hall effect sensor mode] Control the motor speed in closed-loop control mode
	mtr_offset_calc_init input: none output: none	Initialize the offset calculation of the current detection
	mtr_set_speed_ref input: int8_t s1_ref_dir :: a reference rotational direction int8_t s1_dir :: a rotational direction output: none	Set a reference speed
	mtr_set_voltage_ref input: st_mtr_120_control_t * p_st_120 :: a pointer to a structure of motor control main data output:none	[Voltage control mode] Set a reference voltage
	mtr_set_current_ref input: st_mtr_120_control_t * p_st_120 :: a pointer to a structure of motor control main data output:none	[Current control mode] Set a reference current

Table 3-23 List of Functions “rm_mtr_speed.c” (2/2)

File	Function	Process
rm_mtr_speed.c	mtr_pi_run_asr input: st_mtr_pi_t * p_st_pi :: a pointer to PI control structure int32_t s4_err :: An error const uint8_t u1_kp_q :: Q value of proportional gain const uint8_t u1_q_err_shift :: a shift value of an error const uint8_t u1_q_kidt_shift :: Q value of itegral gain const uint8_t u1_q_intg_shift :: a shift value of an iteger part output:int32_t :: PI output	Process for speed PI control
	mtr_duty_calc [inline function] input: int32_t s4_ref_v :: a reference value int16_t s2_ad_vdc_bit :: an A/D conversion value of VDC output: int16_t :: a duty value	Calculate a duty value
	mtr_abs [inline function] input: int32_t s4_value :: an input value output: int32_t :: an output value	Calculate an absolute value
	mtr_limit [inline function] input: int32_t s4_value :: an input value int32_t s4_max :: a maximum value int32_t s4_min :: a minimum value output: int32_t :: an output value	[Current control mode] Limit a value
	mtr_limit_abs [inline function] input: int32_t s4_value :: an input value int32_t s4_limit_value :: a limit value output: int32_t :: an output value	Limit a value with an absolute value
	mtr_error_check input: st_mtr_foc_carrier_t *p_st_carrier :: A pointer to a structure for speed control output: none	Process errors - Detect over-voltage - Detect under-voltage - Detect over-speed - Detect time-out - Detect a pattern error
	mtr_hall_signal_set [inline function] input: none output: none	[Hall effect sensor mode] Set a voltage pattern depending on a signal from Hall effect sensors
	mtr_set_chopping_pattern [inline function] input: uint16_t u2_pattern :: a voltage pattern output: none	Set a chopping pattern
	mtr_draw_in_pattern_set input: uint8_t u1_state_ctrl_mode :: a control status int8_t s1_dir :: a rotational direction output: none	[Sensorless mode] Set a voltage pattern in a draw-in state
	mtr_start_delay_timer [inline function] input: uint32_t u4_delay_count :: a delay count output: none	[Sensorless mode] Start Delay Timer
	mtr_openloop_pattern_set [inline function] input: none output: uint8_t :: a voltage pattern	[Sensorless mode] Set a voltage pattern in open-loop control mode

Table 3-24 List of Functions “rm_mtr_driver.c” (1/2)

File	Function	Process
rm_mtr_driver.c	RM_MTR_ControlInit input: none output: none	Initialize the motor control system - initialize the motor status - initialize control variables
	RM_MTR_EventExec input: uint8_t u1_event :: an event output: none	Change the motor status and execute an event process
	RM_MTR_CapacitorCharge input: none output: uint16_t :: a time-out error	Wait for stability of the bus voltage
	RM_MTR_StatusGet input: none output: uint8_t :: the motor status	Get the motor status
	RM_MTR_StatusErrorGet input: none output: uint16_t :: the error status	Get the error status
	RM_MTR_IcsBufferSet input: none output: none	Input values of ICS variables to ICS buffer variables
	RM_MTR_VariablesSet 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 3-25 List of Functions “rm_mtr_driver.c” (2/2)

File	Function	Process
rm_mtr_driver.c	mtr_statemachine_init input: st_mtr_statemachine_t * p_state_machine :: a pointer to a motor status structure output:none	Initialize the motor status
	mtr_statemacine_reset input: st_mtr_statemachine_t * p_state_machine :: a pointer to a motor status structure output: none	Reset the motor status
	mtr_state_machine_event input: st_mtr_statemachine_t * p_state_machine :: a pointer to a motor status structure void * p_object :: a pointer to a structure of control variables uint8_t u1_event :: an event output:uint8_t :: an error status	Execute an event process
	mtr_statemachine_get_status input: st_mtr_statemachine_t * p_state_machine :: a pointer to a motor status structure output: uint8_t :: the motor status	Get the motor status
	mtr_act_none input: st_mtr_statemachine_t * p_state_machine :: a pointer to a motor status structure void * p_param :: a pointer to a structure of control variables output: uint8_t :: an execution result (always 0)	No process
	mtr_act_init input: st_mtr_statemachine_t * p_state_machine :: a pointer to a motor status structure void * p_param :: a pointer to a structure of control variables output: uint8_t :: an execution result (always 0)	Initialize control variables
	mtr_act_error input: st_mtr_statemachine_t * p_state_machine :: a pointer to a motor status structure void *p_param :: a pointer to a structure of control variables output: uint8_t :: an execution result (always 0)	Post-process when an error is detected
	mtr_act_drive input: st_mtr_statemachine_t * p_state_machine :: a pointer to a motor status structure void *p_param :: a pointer to a structure of control variables output: uint8_t :: an execution result (always 0)	Start the motor
	mtr_act_stop input: st_mtr_statemachine_t * p_state_machine :: a pointer to a motor status structure void * p_param :: a pointer to a structure of control variables output: uint8_t :: an execution result (always 0)	Stop the motor

3.3 Lists of variables of sensorless 120-degree conducting control software

Lists of variables used in this control program are shown below. However, note that the local variables are not mentioned.

In the sample programs, fixed-point number is used for calculation. Therefore, in advance, some control variables are set in fixed-point number. Bits number in fractional part of fixed-point number is expressed in the Q format. “Qn” means n bits left shift.

Table 3-26 List of variables “mtr_main.c”

Variable	Type	Qn	PU	Content	Remarks
g_u2_conf_hw	uint16_t	Q0	-	RMW configuration variables	
g_u2_conf_sw	uint16_t	Q0	-	RMW configuration variables	
g_u2_conf_tool	uint16_t	Q0	-	RMW configuration variables	
gui_u1_active_gui	uint8_t	Q0	-	RMW configuration variables	
g_u2_conf_sw_ver	uint16_t	Q0	-	RMW configuration variables	
com_u1_run_event	uint8_t	Q0	-	Input event and change run mode	
g_u1_run_event	uint8_t	Q0	-	Input event and change run mode	
g_u1_conf_motor_type_len	uint8_t	Q0	-	Motor Type String Length	
g_u1_conf_control_len	uint8_t	Q0	-	Control String Length	
g_u1_conf_inverter_len	uint8_t	Q0	-	Inverter String Length	
g_u1_conf_motor_type[CONF_MOTOR_TYPE_LEN]	uint8_t	Q0	-	Motor Type String	
g_u1_conf_control[CONF_CONTROL_LEN]	uint8_t	Q0	-	Control String	
g_u1_conf_inverter[CONF_INVERTER_LEN]	uint8_t	Q0	-	Inverter String	

Table 3-27 List of variables “r_mtr_ics.c”

Variable	Type	Qn	PU	Content	Remarks
st_ics	st_mtr_ctrl_input_t	-	-	Structure for ICS variable transfer	
g_st_com_init	const st_com_t	-	-	Initial values for the COM variable structure	
g_st_com	st_com_t	-	-	COM Variable Structure	
g_st_design_param_macro	const st_mtr_design_parameter_macro _t	-	-	Initial values of the control parameter structure	
g_s2_enable_write	int16_t	Q0	-	Variable to allow variable rewriting	

Table 3-28 List of variables “rm_mtr_pin_data.c”

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

Table 3-29 List of variables “rm_mtr_120.c”

Variable	Type	Qn	PU	Content	Remarks
g_u1_cnt_ics	volatile uint8_t	Q0	-	Counter for period of RMW communication	

Table 3-30 List of variables “rm_mtr_driver.c”

Variable	Type	Qn	PU	Content	Remarks
g_u1_trig_enable_write	uint8_t	Q0	-	Transfer completion flag	
g_st_ctrl_params	st_mtr_design_parameter_t	-	-	Control parameters	
g_st_gain_v_buf	st_mtr_ctrl_gain_voltage_mode_t	-	-	Input Gain Data	[Voltage control mode]
g_st_gain_i_buf	st_mtr_ctrl_gain_current_mode_t	-	-	Input Gain Data	[Current control mode]

Table 3-31 List of variables “r_mtr_carrier.c”

Variable	Type	Qn	PU	Content	Remarks
g_st_120	st_mtr_120_control_t	-	-	Structure for 120 conducting control	structure
g_u2_ol_v_pattern_table_CW[7]	const uint16_t	Q0	-	Array for voltage pattern at open-loop drive (CW)	
g_u2_ol_v_pattern_table_CCW[7]	const uint16_t	Q0	-	Array for voltage pattern at open-loop drive (CCW)	
g_u2_chopping_pattern_table_CW[6]	const uint16_t	Q0	-	Array for chopping pattern (CW)	[Non Complementary mode]
g_u2_chopping_pattern_table_CCW[6]	const uint16_t	Q0	-	Array for chopping pattern (CCW)	[Non Complementary mode]
g_u2_chopping_pattern_table_CW[6]	const uint16_t	Q0	-	Array for chopping pattern (CW)	[Complementary mode]
g_u2_chopping_pattern_table_CCW[6]	const uint16_t	Q0	-	Array for chopping pattern (CCW)	[Complementary mode]
g_u1_flag_delay	static uint8_t	Q0	-	Delay interrupt flag	

Table 3-32 List of variables “rm_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	Q30	BEMF constant (voltage/angular frequency)	Induced voltage constant [PU]	
s4_mtr_j	int32_t	Q25	Inertia (BEMF constant ×current×(number of pole pairs/angular frequency) ^2)	Rotor inertia [PU]	

Table 3-33 List of variables “rm_mtr_ctrl_gain.h / Structure : st_mtr_design_parameter_macro_t”

Variable	Type	Qn	PU	Content	Remarks
f4_dt	float	-	-	Control period for current [s]	
f4_dt_speed	float	-	-	Control period for speed loop [sec]	
f4_pu_sf_afreq	float	-	-	Frequency scale factor	
f4_speed_limit_rpm	float	-	-	Speed limit exceeded	
u1_q_current	uint8_t	Q0	-	Q-format of current	
u1_q_voltage	uint8_t	Q0	-	Q-format of voltage	
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 ki × dt	
u1_q_asr_err	uint8_t	Q0	-	Q-format of ASR PI error	
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 ki× dt	
u1_q_acr_err	uint8_t	Q0	-	Q-format of ACR PI error	

Table 3-34 List of variables “rm_mtr_ctrl_gain.h / Structure : st_mtr_design_parameter_com_t”

Variable	Type	Qn	PU	Content	Remarks
f4_speed_omega	float	-	-	Speed PI controller natural frequency [Hz]	
f4_asr_zeta	float	-	-	Speed PI controller damping factor	
f4_acr_nf_hz	float	-	-	Current PI control natural frequency [Hz]	
f4_asr_nf_hz	float	-	-	Speed PI control natural frequency [Hz]	
f4_asr_ki_aug	float	-	-	Augmentation rate for integral part of ASR	

Table 3-35 List of variables “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	
u2_mtr_pp	uint16_t	Q0	-	Number of pole pairs	
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 [kgm^2/rad^2]	
f4_d0_div_pm	float	-	-	Static friction coefficient/(number of pole pairs × BEMF constant)	
f4_d1_div_p2m	float	-	-	Kinetic friction coefficient/((number of pole pairs)^2 × BEMF constant)	

Table 3-36 List of variables “rm_mtr_ctrl_gain.h / Structure : st_mtr_ctrl_gain_voltage_mode_t”

Variable	Type	Qn	PU	Content	Remarks
s4_asr_kp	int32_t	[Hall effect sensor mode] Q31 [Sensorless mode] Q32	Voltage/Angular frequency	Speed PI proportional gain [PU]	
s4_asr_kidt	int32_t	[Hall effect sensor mode] Q35 [Sensorless mode] Q35	Voltage/Angular frequency	Speed PI integral gain * dt [PU]	
s4_asr_max_intg	int32_t	[Hall effect sensor mode] Q26 [Sensorless mode] Q26	Angular frequency	Integral upper limit	
s4_lim_asr_err	int32_t	[Hall effect sensor mode] Q26 [Sensorless mode] Q26	Angular frequency	Limit of ASR error	

Table 3-37 List of variables “rm_mtr_ctrl_gain.h / Structure : st_mtr_ctrl_gain_current_mode_t”

Variable	Type	Qn	PU	Content	Remarks
s4_asr_kp	int32_t	[Hall effect sensor mode] Q30 [Sensorless mode] Q29	Current/ Angular frequency	Speed PI proportional gain [PU]	
s4_asr_kidt	int32_t	[Hall effect sensor mode] Q38 [Sensorless mode] Q38	Current/ Angular freqeuncy	Speed PI integral gain * dt [PU]	
s4_asr_max_intg	int32_t	[Hall effect sensor mode] Q22 [Sensorless mode] Q22	Angular frequency	Integral upper limit	
s4_acr_kp	int32_t	[Hall effect sensor mode] Q32 [Sensorless mode] Q32	Resistance	DC link current control proportional gain [PU]	
s4_acr_kidt	int32_t	[Hall effect sensor mode] Q37 [Sensorless mode] Q37	Resistance	DC link current control integral gain* dt [PU]	
s4_acr_max_intg	int32_t	[Hall effect sensor mode] Q24 [Sensorless mode] Q23	Current	Integral upper limit	
s4_d0_div_pm	int32_t	Q28	-	Static friction coefficient/(number of pole pairs × BEMF constant)	
s4_d1_div_p2m	int32_t	Q37	-	Kinetic friction coefficient/((number of pole pairs) ^2 × BEMF constant)	
s4_lim_asr_err	int32_t	[Hall effect sensor mode] Q22 [Sensorless mode] Q22	Angular frequency	Limit of ASR error	

Table 3-38 List of variables “r_mtr_driver.h / Structure : st_mtr_ctrl_input_t”

Variable	Type	Qn	PU	Content	Remarks
s1_ref_direction	int8_t	Q0	-	Direction of rotation	
s4_ref_speed	int32_t	Q28	Angular frequency	Command rotational speed [PU]	electric angle
s4_lim_accel	int32_t	Q28	Angular frequency	Limit of acceleration [PU]	electric angle
u2_cnt_offset_calc	uint16_t	Q0	-	Current off set detection time	
s4_ref_lim_v	int32_t	Q28	Voltage	Lower limit of voltage reference [PU]	
s4_ramp_lim_v	int32_t	Q28	Voltage	Limit of variation of voltage [PU]	[Voltage control mode]
s4_lim_current	int32_t	Q28	Current	Limit of variation of current [PU]	[Current control mode]
s4_ref_v_start	int32_t	Q28	Voltage	Initial reference voltage at start-up	[Hall effect sensor mode] [Voltage control mode]
s4_ref_idc_start	int32_t	Q28	Current	Initial reference current at start-up	[Hall effect sensor mode] [Current control mode]
s4_draw_in_ref_v	int32_t	Q28	Voltage	Reference voltage at draw-in [PU]	[Voltage control mode]
s4_ol_ref_v_buff	int32_t	Q28	Voltage	Reference voltage at open-loop drive buffer [PU]	[Voltage control mode]
s4_draw_in_ref_idc	int32_t	Q28	Current	DC link current reference at draw-in [PU]	[Current control mode]
s4_ol_ref_idc_buff	int32_t	Q28	Current	Openloop start reference DC link current buffer [PU]	[Current control mode]
s4_th_speed_ol2cl	int32_t	Q28	Angular frequency	Speed allowed to transition to PI control [PU]	electric angle [Sensorless mode]
s4_ol2cl_ramp_speed	int32_t	Q28	Angular frequency	Acceleration at transition to PI control [PU]	electric angle [Sensorless mode]
s4_th_speed_cl2ol	int32_t	Q28	Angular frequency	adjust delay counts	[Sensorless mode]
s4_angle_shift_adjust	int32_t	Q0	-	Speed to transition to open-loop drive [PU]	electric angle [Sensorless mode]
st_motor	st_mtr_parameter_t	-	-	Structure for motor parameter	

Table 3-39 List of variables “r_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	
u2_crnt_event	uint8_t	Q0	-	Execution event	

Table 3-40 List of variables “rm_mtr_120.h / Structure : st_mtr_ol2cl_t”

Variable	Type	Qn	PU	Content	Remarks
u1_flg_change	uint8_t	-	-	Flag to allow to transition to PI control	[Sensorless mode]
u1_flg_change_speed	uint8_t	-	-	Flag for exceeding change speed	
u1_flg_zc	uint8_t	-	-	Flag for zero-crossing detection avoiding commutation	
u1_rotor_pos	uint8_t	-	-	Phase of rotor position	
u2_zc_cnt	uint16_t	-	-	Counter for zero-crossing detection	
s4_change_speed	int32_t	Q28	Angular frequency	Speed allowed to transition to PI control [PU]	
s4_ref_speed_buff	int32_t	Q28	Angular frequency	Buff for reference rotational speed [PU]	
s4_ramp_speed	int32_t	Q28	Angular frequency	Acceleration at transition to PI control [PU]	

Table 3-41 List of variables “rm_mtr_120.h / Structure : st_mtr_pi_t”

Variable	Type	Qn	PU	Content	Remarks
s4_kp	int32_t	[Hall effect sensor mode] Speed : Q30 Current : Q32 Voltage : Q31 [Sensorless mode] Speed : Q29 Current : Q32 Voltage : Q32	Speed : Current/angular frequency Current : Resistance Voltage : Voltage/angular frequency	Proportional gain	
s4_kidt	int32_t	[Hall effect sensor mode] Speed : Q38 Current : Q37 Voltage : Q35 [Sensorless mode] Speed : Q38 Current : Q37 Voltage : Q35	Speed : Current/angular frequency Current : Resistance Voltage : Voltage/angular frequency	Integral gain x control period	
s4_max_intg	int32_t	[Hall effect sensor mode] Speed : Q22 Current : Q24 Voltage : Q26 [Sensorless mode] Speed : Q22 Current : Q23 Voltage : Q26	Speed : Angular frequency Current : Current Voltage : Angular frequency	Integral upper limit	
s4_min_intg	int32_t	[Hall effect sensor mode] Speed : Q22 Current : Q24 Voltage : Q26 [Sensorless mode] Speed : Q22 Current : Q23 Voltage : Q26	Speed : Angular frequency Current : Current Voltage : Angular frequency	Integral lower limit	
s4_decimal	int32_t	[Hall effect sensor mode] Speed : Q22 Current : Q24 Voltage : Q26 [Sensorless mode] Speed : Q22 Current : Q23 Voltage : Q26	Speed : Angular frequency Current : Current Voltage : Angular frequency	Stored decimal part	

Table 3-42 List of variables “rm_mtr_120.h / Structure : st_mtr_voltage_asr_t” [Voltage control mode]

Variable	Type	Qn	PU	Content	Remarks
s4_pre_speed	int32_t	Q28	Angular frequency	Previous rotational speed [PU]	
st_pi	st_mtr_pi_t	-	-	Speed PI structure	
s4_lim_asr_err	int32_t	Q28	Angular frequency	Limit of ASR error	

Table 3-43 List of variables “rm_mtr_120.h / Structure : st_mtr_current_acr_t” [Current control mode]

Variable	Type	Qn	PU	Content	Remarks
s4_lim_idc	int32_t	Q28	Current	DC link current limit	
s4_lim_ramp_current	int32_t	Q28	Current	Limit value for current rise [PU/ms]	
st_pi	st_mtr_pi_t	-	-	Current PI structure	

Table 3-44 List of variables “rm_mtr_120.h / Structure : st_mtr_current_asr_t” [Current control mode]

Variable	Type	Qn	PU	Content	Remarks
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)	
st_pi	st_mtr_pi_t	-	-	Speed PI structure	
s4_lim_asr_err	int32_t	Q28	Angular frequency	Limit of ASR error	

Table 3-45 List of variables “rm_mtr_120.h / Structure : st_mtr_hall_speed” [Hall effect sensor mode]

Variable	Type	Qn	PU	Content	Remarks
u2_hall_signal	uint16_t	-	-	Hall signal value	
u1_flag_1st_interrupt	uint8_t	-	-	First hole interrupt flag	
s4_ref_v_start	int32_t	Q28	Voltage	Initial reference voltage at start-up	[Voltage control mode]
s4_ref_idc_start	int32_t	Q28	Current	Initial reference current at start-up	[Current control mode]

Table 3-46 List of variables “rm_mtr_120.h / Structure : st_mtr_sensorless_control_t”

Variable	Type	Qn	PU	Content	Remarks
s2_ad_vn_offset	int16_t	Q0	-	Virtual neutral voltage offset	
s4_angle_shift_adjust	int32_t	Q0	-	Virtual hall pattern delay adjustment value	
st_ol2cl	st_mtr_ol2cl_t	-	-	Sensorless switching process structure	

Table 3-47 List of variables “rm_mtr_120.h / Structure : st_mtr_sensorless_carrier_t” [Sensorless mode]

Variable	Type	Qn	PU	Content	Remarks
u1_flg_pattern_change	uint8_t	Q0	-	Flag for zero-crossing detection	
u1_bemf_signal	uint8_t	Q0	-	Estimated Hall pattern	
u1_pre_bemf_signal	uint8_t	Q0	-	Previous estimated Hall pattern	
u1_flg_pattern_change_ol	uint8_t	Q0	-	Flag for voltage pattern change at open-loop drive	
u4_speed_cnt_ol	uint32_t	Q0	-	Counter for patten change at open-loop drive	
u4_pattern_period_ol	uint32_t	Q0	-	Period for pattern change at open-loop drive	
u2_v_pattern_ol	uint16_t	Q0	-	Voltage pattern at open-loop drive	
u2_v_pattern_num_ol	uint16_t	Q0	-	Ring buffer for voltage pattern at open-loop drive	
u4_bemf_delay	uint32_t	Q0	-	Delay counts	
u4_zerocross_filter_cnt	uint32_t	Q0	-	Zerocross filter count	
s2_ad_vu	int16_t	Q0	-	U phase voltage	
s2_ad_vv	int16_t	Q0	-	V phase voltage	
s2_ad_vw	int16_t	Q0	-	W phase voltage	
s2_ad_vn	int16_t	Q0	-	Virtual neutral voltage	
s2_ad_vn_ctrl	int16_t	Q0	-	Virtual neutral voltage for control	

Table 3-48 List of variables “rm_mtr_120.h / Structure : st_mtr_sensorless_speed_t”

Variable	Type	Qn	PU	Content	Remarks
u4_cnt_draw_in	uint32_t	Q0	-	Counter for pattern change at draw-in	
s4_lim_ramp_v_ol	int32_t	Q28	Voltage	Limit of rate of voltage change during open-loop control	
s4_speed_cl2ol_th	int32_t	Q28	Angular frequency	Transition speed from sensorless control to open-loop [PU]	
u4_v_const_period	uint32_t	Q0	-	Period for pattern change at draw-in	[Voltage control mode]
s4_ref_v_draw_in	int32_t	Q28	Voltage	Draw-in command voltage [PU]	[Voltage control mode]
s4_ref_v_ol	int32_t	Q28	Voltage	Startup voltage [PU]	[Voltage control mode]
s4_ref_v_buff_ol	int32_t	Q28	Voltage	Startup voltage buffer [PU]	[Voltage control mode]
u4_idc_const_period	uint32_t	Q0	-	Period for pattern change at draw-in	[Current control mode]
s4_ref_idc_draw_in	int32_t	Q28	Current	Draw-in command current [PU]	[Current control mode]
s4_ref_idc_ol	int32_t	Q28	Current	Startup current [PU]	[Current control mode]
s4_ref_idc_buff_ol	int32_t	Q28	Current	Startup current buffer [PU]	[Current control mode]

Table 3-49 List of variables “r_mtr_120.h / Structure : st_mtr_120_carrier_t”

Variable	Type	Qn	PU	Content	Remarks
s4_ad_idc	int32_t	Q28	Current	DC link current [PU]	
s2_ad_vdc_bit	int16_t	Q0	-	A/D conversion result of bus voltage	
s2_ad_idc_bit	int16_t	Q0	-	A/D conversion result of DC link current	
u4_ad_offset_idc_sum	uint32_t	Q0	-	DC link current offset value integral value	
u4_ad_offset_idc	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_lim_over_current	int32_t	Q28	Current	Overcurrent limit value	
st_less_carrier	st_mtr_sensorless_carrier_t	-	-	Structure for sensorless mode carrier control	[Sensorless mode]
st_motor	st_mtr_parameter_t	-	-	Structure for motor parameter	

Table 3-50 List of variables “r_mtr_120.h / Structure : st_mtr_120_speed_t”

Variable	Type	Qn	PU	Content	Remarks
s4_ref_speed	int32_t	Q28	Angular frequency	Reference rotational speed [PU]	
s4_lim_speed	int32_t	Q28	Angular frequency	Overspeed limit value	
s4_ramp_lim_speed	int32_t	Q28	Angular frequency	Limit of acceleration [PU]	
s4_ref_v	int32_t	Q28	Voltage	Target voltage [PU]	[Voltage control mode]
s4_ramp_lim_v	int32_t	Q28	Voltage	Voltage change range limit value [PU]	[Voltage control mode]
st_voltage_asr	st_mtr_voltage_asr_t	-	-	ASR structure	[Voltage control mode]
s4_ref_idc	int32_t	Q28	Current	Target current [PU]	[Current control mode]
st_current_asr	st_mtr_current_asr_t	-	-	ASR structure	[Current control mode]
st_less_speed	st_mtr_sensorless_speed_t	-	-	Structure for hall effect sensor mode speed control	[Current control mode]
st_hall_speed	st_mtr_hall_speed_t	-	-	Structure for sensorless mode speed control	[Sensorless mode]
st_stm	st_mtr_statemachine_t	-	-	Structure for state machine	[Hall effect sensor mode]

Table 3-51 List of variables “rm_mtr_120.h / Structure : st_mtr_120_control_t” [1/2]

Variable	Type	Qn	PU	Content	Remarks
u1_state_drive	uint8_t	Q0	-	Control status	0x00: Motor control none 0x01: Open loop Draw-in 0x02: Force sync control 0x05: Open loop speed control 0x81: Closed loop speed control
u1_state_ctrl_mode	uint8_t	Q0	-	Direction of rotation	1 : CW -1 : CCW
s1_ref_dir	int8_t	Q0	-	Reference direction of rotation	1 : CW -1 : CCW
s1_dir	int8_t	Q0	-	Current direction of rotation	0: Unreache 1: Reached
u1_flg_charge_cap	uint8_t	Q0	-	Power source charged flag	
u1_flg_offset_end	uint8_t	Q0	-	Offset calculate end flag	
u1_flg_pattern_error	uint8_t	Q0	-	Flag for voltage pattern error	
u1_first_rotation_cnt	uint8_t	Q0	-	Count first 6 pattern changes for speed calculation	
u1_v_pattern_ring_buff	uint8_t	Q0	-	Ring buffer for timer counter for speed calculation	
u1_state_ref_speed	uint8_t	Q0	-	Speed command value generation status	0: Speed 0 1: Speed change
u4_speed_timer_cnt	uint32_t	-	-	Value of timer counter	
u4_pre_speed_timer_cnt	uint32_t	-	-	Previous value of timer counter	
u4_timer_cnt_buff[6]	uint32_t	-	-	Timer counts for 60 degree	
u4_timer_cnt_sum	uint32_t	-	-	Total counter value for speed calculation	
u2_v_pattern	uint16_t	-	-	Voltage pattern	
u2_pwm_duty	uint16_t	-	-	PWM duty	
u4_cnt_timeout	uint32_t	-	-	Counter for timeout error	
s4_speed	int32_t	Q28	Angular frequency	Rotational speed [PU]	
s4_ref_speed_ctrl	int32_t	Q28	Angular frequency	Reference rotational speed for control [PU]	
s4_ad_vdc	int32_t	Q28	Voltage	Bus voltage [PU]	
u2_error_status	uint16_t	Q0	-	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

Table 3-52 List of variables “rm_mtr_120.h / Structure : st_mtr_120_control_t” [2/2]

Variable	Type	Qn	PU	Content	Remarks
s4_lim_v	int32_t	Q28	Voltage	Lower limit of voltage	
s4_ref_lim_v	int32_t	Q28	Voltage	Lower limit of voltage reference	
s4_ref_v_ctrl	int32_t	Q28	Voltage	Reference voltage [PU]	
u2_state_ref_voltage	uint16_t	-	-	Reference voltage mode	[Voltage control mode] 0: voltage 0 1: voltage manual control 2: Speed PI output
u2_state_ref_current	uint16_t	-	-	Reference current mode	[Current control mode] 0: current 0 1: current manual control 2: Speed PI output
s4_ref_idc_ctrl	int32_t	Q28	Current	Reference current [PU]	[Current control mode]
st_acr	st_mtr_acr_t	-	-	ACR structure	Structure
st_carrier	st_mtr_foc_carrier_t	-	-	Structure for carrier control	Structure
st_speed	st_mtr_foc_speed_t	-	-	Structure for speed control	Structure
st_less_control	st_mtr_sensorless_control_t	-	-	Structure for sensorless control	Structure

Table 3-53 List of variables “rm_mtr_ics.h / Structure : st_com_t”

Variable	Type	Qn	PU	Content	Remarks
st_com	st_mtr_design_parameter_com_t	-	-	Com variable structure	
com_u1_direction	uint8_t	Q0	-	Direction of rotation	0：CW 1：CCW
com_u2_mtr_pp	uint16_t	Q0	-	Number of pole pairs	
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_ke	float	-	-	Induced voltage constant [V･s/rad]	
com_f4_mtr_j	float	-	-	Rotor inertia [kg･m^2/(rad^2)]	
com_f4_ref_lim_v	float	-	-	Lower limit of voltage reference [V]	
com_f4_ramp_limit_speed_rpm	float	-	-	Limit of acceleration [rpm/ms]	Mechanical angle
com_f4_ref_speed_rpm	float	-	-	Command rotational speed [rpm]	Mechanical angle
com_u2_offset_calc_cnt	uint16_t	Q0	-	Current offset detection time	
com_f4_ramp_lim_v	float	-	-	Limit of variation of voltage [V/ms]	[Voltage control mode]
com_f4_ramp_lim_current	float	-	-	Limit of variation of current [A/ms]	[Current control mode]
com_f4_mtr_d0	float	-	-	Coefficient of static friction [kg･m^2/(rad･s^2)]	[Current control mode]
com_f4_mtr_d1	float	-	-	Coefficient of kinetic friction [kg･m^2/(rad^2･s)]	[Current control mode]
com_f4_ref_v_start	float	-	-	Initial reference voltage at start-up	[Hall effect sensor-based voltage control mode]
com_f4_ref_idc_start	float	-	-	Initial reference current at start-up	[Hall effect sensor-based current control mode]
com_f4_ref_v_draw_in	float	-	-	Reference voltage at draw-in [V]	[Sensorless voltage control mode]
com_f4_ref_v_ol	float	-	-	Reference voltage at open-loop drive[V]	[Sensorless voltage control mode]
com_f4_ref_idc_draw_in	float	-	-	Command current at draw-in [V]	[Sensorless current control mode]
com_f4_ref_idc_ol	float	-	-	Command current at open-loop drive[V]	[Sensorless current control mode]
com_f4_ol2cl_speed_rpm	float	-	-	Speed allowed to transit to PI control[rpm]	Mechanical angle [Sensorless mode]
com_f4_ol2cl_ramp_speed_rpm	float	-	-	Acceleration at transition to PI control [rpm/ms]	Mechanical angle [Sensorless mode]
com_s4_angle_shift_adjust	uint32_t	Q0	-	Adjust delay counts	[Sensorless mode]
com_s2_enable_write	int16_t	Q0	-	Variable to allow to input RMW structure	

3.4 Macro definitions of sensorless 120-degree conducting control software

Lists of macro definitions used in this control program are shown below.

Table 3-54 List of Macro definitions “r_mtr_config.h”

Macro	Definition value	Description	Remarks
IP_MCK	-	Selection of inverter board	
MP_R42BLD30L3_24V	-	Selection of motor parameters	
CP_R42BLD30L3_24V	-	Selection of control parameters	
SENSORLESS_CURRENT	0	Sensorless and Current control mode	
SENSORLESS_VOLTAGE	1	Sensorless and Voltage control mode	
HALL_CURRENT	2	Hall effect sensor and Current control mode	
HALL_VOLTAGE	3	Hall effect sensor and Voltage control mode	
MTRCONF_MODE	SENSORLESS_CURRENT SENSORLESS_VOLTAGE HALL_CURRENT HALL_VOLTAGE	Selection of control mode	Default setting SENSORLESS_CURRENT
NON_COMPLEMENTARY	0	Non-complementary PWM mode	
COMPLEMENTARY	1	Complementary PWM mode	
MTRCONF_PWM_MODE	NON_COMPLEMENTARY COMPLEMENTARY	Selection of PWM mode	Default setting COMPLEMENTARY
PS_DRAW_IN	3	Draw-in method	
PS_METHOD	PS_DRAW_IN	Preparation method for start-up	Default setting PS_DRAW_IN
HALL	0	Hall effect sensor mode	
LESS	1	Sensorless mode	
VOLTAGE	0	Voltage control mode	
CURRENT	1	Current control mode	
MTRCONF_SENSOR_MODE	HALL LESS	Selection of sensor mode	Default setting LESS
MTRCONF_CONTROL_MODE	VOLTAGE CURERNT	Selection of control mode	Default setting CURRENT

Table 3-55 List of Macro definitions “r_mtr_motor_parameter.h”

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

Table 3-56 **L**ist of Macro definitions “r_mtr_inverter_parameter.h”

Macro	Definition value	Description	Remarks
IP_DEADTIME	1	Deadtime	
IP_CURRENT_RANGE	16.5	Current scaling range [A]	
IP_VDC_RANGE	73.51276f	Voltage scaling range [V]	
IP_INPUT_V	24	Input voltage [V]	
IP_CURRENT_LIMIT	4	Current limit value [A]	
IP_OVERVOLTAGE_LIMIT	28	Overvoltage limit [V]	
IP_UNDERVOLTAGE_LIMIT	8	Undervoltage limit [V]	
IP_DC_SHUNT_RESISTANCE	0.01f	DC Link Shunt Resistance [ohm]	
IP_DC_AMPLIFICATION_GAIN	20	DC Link Current Amplification Gain	
IP_BSC_CHARGE_CNT	150	Period of charging bootstrap capacitor	
IP_CHARGE_CAP_WAIT_CNT	350	Charge time of DC capacitor	

Table 3-57 List of Macro definitions “r_mtr_control_parameter.h” (1/2)

Macro	Definition value	Description	Remarks
CP_PWM_TIMER_FREQ	32.0f	PWM timer frequency [MHz]	
CP_TRG_TIMER_FREQ	32.0f	1[ms] interrupt timer frequency [MHz]	
CP_CARRIER_FREQ	20.0f	PWM carrier frequency [kHz]	
CP_SPEED_CALC_TIMER_FREQ	32000.0f	Frequency of a timer to estimate a rotational speed [kHz]	
CP_SPEED_CTRL_PERIOD	0.001f	Speed control cycle [sec]	
CP_AD_TRG_DEFAULT_CNT	32.0f	A/D trigger count default value	
CP_MAX_SPEED_RPM	2900	Maximum limit of command rotational speed [rpm]	Mechanical angle
CP_SPEED_LIMIT_RPM	3262.5f	Maximum limit of estimated rotational speed [rpm]	Mechanical angle
CP_OC_LIMIT	3.955f	Overcurrent limit [A]	
CP_OFFSET_CALC_CNT	512	Number of measurements of DC link current offset	
CP_OFFSET_CALC_ST_WAIT_CNT	100	Offset calculation stable waiting time	
CP_HALL2OL_REV_SPEED_RPM	500	Transition speed to PI control when direction of rotation is reversed [rpm]	[Hall effect sensor mode] Mechanical angle
CP_MIN_SPEED_RPM	100 [Hall effect sensor and Current control mode] 200 [Hall effect sensor and Voltage control mode] 300 [Sensorless mode]	Minimum limit of command rotational speed [rpm]	Mechanical angle
CP_LIMIT_V	0.29f [Hall effect sensor and Current control mode] 1.0f [Hall effect sensor and Voltage control mode] 1.2f [Sensorless and Current control mode] 1.56f [Sensorless and Voltage control mode]	Lower limit of voltage [V]	
CP_RAMP_LIMIT_SPEED_RPM_HALL	9.306984f [Current control mode] 9.306984f [Voltage control mode]	Limit of acceleration [rpm/ms]	[Hall effect sensor mode] Mechanical angle
CP_ACR_NF_HZ	500 [Hall effect sensor mode] 300 [Sensorless mode]	Current PI control natural frequency [Hz]	[Current control mode]
CP_ASR_NF_HZ	9.99f [Hall effect sensor mode] 10.67f [Sensorless mode]	Speed PI control natural frequency [Hz]	[Current control mode]

Table 3-58 List of Macro definitions “r_mtr_control_parameter.h” (2/2)

Macro	Definition value	Description	Remarks
CP_ASR_KI_AUG	8 [Hall effect sensor mode] 8 [Sensorless mode]	Augmentation rate for integral part of ASR	[Current control mode]
CP_RAMP_LIMIT_CURRENT	0.043821f [Hall effect sensor mode] 0.043821f [Sensorless mode]	Limit for variation of current [A/ms]	[Current control mode] Mechanical angle
CP_START_REF_IDC	0.05f	Reference current for start-up [A]	[Hall effect sensor mode] [Current control mode]
CP_ASR_OMEGA_HZ	20 [Hall effect sensor mode] 20 [Sensorless mode]	Frequency for speed PI control [Hz]	[Voltage control mode]
CP_ASR_ZETA	0.5f [Hall effect sensor mode] 0.4f [Sensorless mode]	Damping factor for speed PI control	[Voltage control mode]
CP_RAMP_LIMIT_V_HALL	0.29f	Limit for variation of voltage [V]	[Hall effect sensor mode] [Voltage control mode]
CP_START_REF_V	0.29f	Reference voltage for start-up [V]	[Hall effect sensor mode] [Voltage control mode]
CP_OL2CL_SPEED_RPM	500	Speed allowed to transition to PI control [rpm]	[Sensorless mode] Mechanical angle
CP_OL2CL_SPEED_RAMP_RPM	2	Acceleration allowed to transition to PI control [rpm/ms]	[Sensorless mode] Mechanical angle
CP_OL2CL_SPEED_MARGIN	80	Rotational speed margin for transition of control mode [rpm]	[Sensorless mode] Mechanical angle
CP_OL2CL_ZC_CNT	3	Number of zero-crossings for transition to PI control	[Sensorless mode]
CP_SKIP_SPIKE_NOSE	8	Carrier counts for avoiding spike noise	[Sensorless mode]
CP_RAMP_LIMIT_SPEED_RPM_LESS	9.306984f [Current control mode] 9.306984f [Voltage control mode]	Limit of acceleration [rpm/ms]	[Sensorless mode] Mechanical angle
CP_DRAW_IN_REF_IDC	1.13f	Reference current in draw-in state[A]	[Sensorless mode] [Current control mode]
CP_OL_REF_IDC	0.07f	Reference current in open-loop [A]	[Sensorless mode] [Current control mode]
CP_LAMP_LIMIT_V_LESS	0.13f	Limit value for voltage rise [V]	[Sensorless mode] [Voltage control mode]
CP_DRAW_IN_REF_V	1.49f	Reference voltage in draw-in state[V]	[Sensorless mode] [Voltage control mode]
CP_OL_REF_V	0.3f	Reference Voltage in open-loop [V]	[Sensorless mode] [Voltage control mode]

Table 3-59 List of Macro definitions “r_mtr_scaling_parameter.h” (1/2)

Macro	Definition value	Description	Remarks
FP_SF_VOLTAGE	822226302	Voltage PU conversion value (IP_VDC_RANGE*PU_SF_VOLTAGE) * (1<<MTR_Q_VOLTAGE)	
FP_SF_CURRENT	3919632764	Current PU conversion value (IP_CURRENT_RANGE * PU_SF_CURRENT) * (1<<MTR_Q_CURRENT)	
PU_BASE_CURRENT_A	MP_RATED_CURRENT	Based quantity for current [A]	
PU_BASE_VOLTAGE_V	IP_INPUT_V	Based quantity for voltage [V]	
PU_BASE_FREQ_Hz	MTR_TWOPI * CP_MAX_SPEED_RPM * MP_POLE_PAIRS/60	Based quantity for frequency [Hz]	
PU_BASE_ANGLE_Rad	1.0f	Based angle for angle [rad]	
PU_SF_CURRENT	1.0f / PU_BASE_CURRENT_A	Current scale factor [PU/A]	
PU_SF_VOLTAGE	1.0f / PU_BASE_VOLTAGE_V	Voltage scale factor [PU/V]	
PU_SF_AFREQ	1.0f / PU_BASE_FREQ_Hz	Angular frequency scale factor [PU/(rad/s)]	
PU_SF_ANGLE	1.0f / PU_BASE_ANGLE_Rad	Angle scale factor [1/rad]	
PU_SF_TIME	PU_SF_ANGLE / PU_SF_AFREQ	Time scale factor [PU/s]	
PU_SF_RES	PU_SF_VOLTAGE / PU_SF_CURRENT	Impedance (resistance) scale factor [PU/ohm]	
PU_SF_IND	PU_SF_RES / PU_SF_AFREQ	Inductance scale factor [PU/H]	
PU_SF_BEMF_CONST	PU_SF_VOLTAGE / PU_SF_AFREQ	Induced voltage constant scale [PU/Wb]	
PU_SF_INERTIA	PU_SF_BEMF_CONST * PU_SF_CURRENT / (MP_POLE_PAIRS * MP_POLE_PAIRS * PU_SF_AFREQ * PU_SF_AFREQ)	Inertia scale factor [PU/(rad/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) [kg･m^2/(V･s^2)	
PU_SF_RPM_RAD	1.0f / CP_MAX_SPEED_RPM	Speed scale factor to convert from [rpm] to [rad/s]	
PU_SF_RAD_RPM	CP_MAX_SPEED_RPM	Speed scale factor to convert from [rad/s] to [rpm]	
PU_SF_ACR_KP	PU_SF_RES	Current PI proportional gain scale factor	
PU_SF_ACR_KIDT	PU_SF_RES	Current PI integral gain scale factor	
PU_SF_ASR_KP	PU_SF_CURRENT / PU_SF_AFREQ [Current control mode] PU_SF_VOLTAGE / PU_SF_AFREQ [Voltage control mode]	Speed PI proportional gain scale factor	
PU_SF_ASR_KIDT	PU_SF_CURRENT / PU_SF_AFREQ [Current control mode] PU_SF_VOLTAGE / PU_SF_AFREQ [Voltage control mode]	Speed PI integral gain scale factor	

Table 3-60 **L**ist of Macro definitions “r_mtr_scaling_parameter.h” (2/2)

Macro	Definition value	Description	Remarks
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_AFREQ	28	Q-format of angular frequency	
MTR_Q_CTRL_TIME_SPEED	30	Q-format of speed control cycle	
MTR_Q_RESISTANCE	33	Q-format of resistance	
MTR_Q_INDUCTANCE	34	Q-format of inductance	
MTR_Q_BEMF_CONST	31	Q-format of induced voltage constant	
MTR_Q_INERTIA	27	Q-format of inertia	
MTR_Q_D1_DIV_P2M	37	Q-format of kinetic friction coefficient /((number of pole pairs)^2 x BEMF constant)	
MTR_Q_ACR_KP	32 [Hall effect sensor mode] 33 [Sensorless mode]	Q-format of current PI proportional gain	[Current control mode]
MTR_Q_ACR_KIDT	37 [Hall effect sensor mode] 38 [Sensorless mode]	Q-format of current PI integral gain x control period	[Current control mode]
MTR_Q_ACR_ERR	23 [Hall effect sensor mode] 22 [Sensorless mode]	Q-format of current PI error	[Current control mode]
MTR_Q_ACR_INTG	12 [Hall effect sensor mode] 12 [Sensorless mode]	Q-format of current PI integer part	[Current control mode]
MTR_Q_ASR_KP	30 [Hall effect sensor and Current control mode] 33 [Hall effect sensor and Voltage control mode] 30 [Sensorless and Current control mode] 34 [Sensorless and Voltage control mode]	Q-format of speed PI proportional gain	
MTR_Q_ASR_KIDT	38 [Hall effect sensor and Current control mode] 36 [Hall effect sensor and Voltage control mode] 38 [Sensorless and Current control mode] 36 [Sensorless and Voltage control mode]	Q-format of speed PI integral gain x control period	
MTR_Q_ASR_ERR	22 [Hall effect sensor and Current control mode] 25 [Hall effect sensor and Voltage control mode] 22 [Sensorless and Current control mode] 25 [Sensorless and Voltage control mode]	Q-format of speed PI error	
MTR_Q_ASR_INTG	12 [Hall effect sensor and Current control mode] 12 [Hall effect sensor and Voltage control mode] 12 [Sensorless and Current control mode] 12 [Sensorless and Voltage control mode]	Q-format of speed PI integer part	
RSFT_AFREQ_KP_2VOLTAGE	MTR_Q_ASR_KP + MTR_Q_AFREQ - MTR_Q_VOLTAGE [Hall effect sensor mode] MTR_Q_ASR_KP + MTR_Q_AFREQ - MTR_Q_VOLTAGE [Sensorless mode]	Right shift, (proportional gan * induced voltage) to voltage	
RSFT_AFREQ_KIDT_2VOLTAGE	MTR_Q_ASR_KIDT + MTR_Q_AFREQ - MTR_Q_VOLTAGE [Hall effect sensor mode] MTR_Q_ASR_KIDT + MTR_Q_AFREQ - MTR_Q_VOLTAGE [Sensorless mode]	Right shift, (integral gan * induced voltage) to voltage	
RSFT_AFREQ_BEMF_CONST_2VOLTAGE	MTR_Q_BEMF_CONST + MTR_Q_AFREQ - MTR_Q_VOLTAGE [Hall effect sensor mode] MTR_Q_BEMF_CONST + MTR_Q_AFREQ - MTR_Q_VOLTAGE [Sensorless mode]	Right shift, (speed * induced voltage) to voltage	
MTR_SPEED_CALC_BASE_SHIFT	13	Right shift amount of the speed calculation coefficient	

Table 3-61 **L**ist of Macro definitions “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 3-62 **L**ist of Macro definitions “ICS2_RA2T1.h”

Macro	Definition value	Description	Remarks
ICS_SCI0_P101_P100	0x00	SCI port settings	

Table 3-63 **L**ist of Macro definitions “r_mtr_ics.h”

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

Table 3-64 List of Macro definitions “r_mtr_ra2t1.h” (1/2)

Macro	Definition value	Description	Remarks
USE_POEG	MTR_SET	Selection of overcurrent usage by POEG	
MTR_CARRIER_FREQ	CP_CARRIER_FREQ	Frequency of carrier [kHz]	
MTR_SPEED_CALC_TIMER_FREQ	CP_SPEED_CALC_TIMER_FREQ	Frequency of the timer for rotational speed [kHz]	
MTR_PWM_TIMER_FREQ	CP_PWM_TIMER_FREQ	Frequency of PWM timer [MHz]	
MTR_SPEED_CTRL_PERIOD	CP_SPEED_CTRL_PERIOD	Control period for speed loop [s]	
MTR_CTRL_PERIOD	1.0f / (MTR_CARRIER_FREQ * 1000)	Control period for current loop [s]	
MTR_OL2CL_BEMF_THRESH	3	Threshold for measuring induced voltage	
MTR_ADC_OFFSET	0x7FF	A/D conversion offset	
MTR_DEADTIME	IP_DEADTIME	Dead-time [us]	
MTR_DEADTIME_CNT	MTR_DEADTIME * MTR_PWM_TIMER_FREQ	Deadtime settings	
MTR_NOT_CARRIER_SET	MTR_PWM_TIMER_FREQ * 1000 / MTR_CARRIER_FREQ / 2	Carrier count of PWM without dead-time	[Complementary PWM mode]
MTR_CARRIER_SET	MTR_NOT_CARRIER_CNT [Complementary PWM mode] MTR_PWM_TIMER_FREQ * 1000 / MTR_CARRIER_FREQ / 2 [Non-Complementary PWM mode]	Carrier count of PWM	
MTR_MAX_DUTY_CNT	MTR_NOT_CARRIER_CNT [Complementary PWM mode] MTR_ CARRIER_SET [Non-Complementary PWM mode]	Maximum value of duty	
MTR_HALF_CARRIER_CNT	(MTR_CARRIER_CNT + MTR_DEADTIME_CNT) / 2 [Complementary PWM mode] (MTR_CARRIER_CNT + MTR_DEADTIME_CNT) / 2 [Non-Complementary PWM mode]	Half carrier count of PWM	
MTR_LOWEST_DUTY_CNT	MTR_CARRIER_CNT + MTR_DEADTIME_CNT	Carrier count of zero percent PWM	
MTR_AD_TRG_DEFAULT_CNT	CP_AD_TRG_DEFAULT_CNT	A/D trigger default count	
MTR_OFFSET_CALC_ST_WAIT_CNT	CP_OFFSET_CALC_ST_WAIT_CNT	Stable waiting time before acquiring current offset	
MTR_CHARGE_CAP_WAIT_CNT	IP_CHARGE_CAP_WAIT_CNT	Charge time of DC capacitor	
MTR_PATTERN_ERROR	0	Voltage pattern (Error)	
MTR_UP_PWM_VN_ON	1	Voltage pattern	
MTR_UP_PWM_WN_ON	2	Voltage pattern	
MTR_VP_PWM_UN_ON	3	Voltage pattern	
MTR_VP_PWM_WN_ON	4	Voltage pattern	
MTR_WP_PWM_UN_ON	5	Voltage pattern	
MTR_WP_PWM_VN_ON	6	Voltage pattern	
MTR_UP_ON_VN_PWM	7	Voltage pattern	
MTR_UP_ON_WN_PWM	8	Voltage pattern	
MTR_VP_ON_UN_PWM	9	Voltage pattern	
MTR_VP_ON_WN_PWM	10	Voltage pattern	
MTR_WP_ON_UN_PWM	11	Voltage pattern	
MTR_WP_ON_VN_PWM	12	Voltage pattern	

Table 3-65 List of Macro definitions “r_mtr_ra2t1.h” (2/2)

Macro	Definition value	Description	Remarks
MTR_U_PWM_VN_ON	13	Voltage pattern	
MTR_U_PWM_WN_ON	14	Voltage pattern	
MTR_V_PWM_UN_ON	15	Voltage pattern	
MTR_V_PWM_WN_ON	16	Voltage pattern	
MTR_W_PWM_UN_ON	17	Voltage pattern	
MTR_W_PWM_VN_ON	18	Voltage pattern	
MTR_UP_ON_V_PWM	19	Voltage pattern	
MTR_UP_ON_W_PWM	20	Voltage pattern	
MTR_VP_ON_U_PWM	21	Voltage pattern	
MTR_VP_ON_W_PWM	22	Voltage pattern	
MTR_WP_ON_U_PWM	23	Voltage pattern	
MTR_WP_ON_V_PWM	24	Voltage pattern	
POEG_RECOVER_TIMEOUT	3	Count of POEG time-out	
ADC_CONVERT_TIMEOUT	0xFFFF	Count of ADC time-out	
MTR_SPEED_CNT_MAX	0x1FFFFFFF	Maximum of a count to measure a rotational speed	
MTR_PORT_PIN_HALL_U	BSP_IO_PORT_02_PIN_06	U phase Hall effect sensor input port	[Hall effect sensor mode]
MTR_PORT_PIN_HALL_V	BSP_IO_PORT_04_PIN_09	V phase Hall effect sensor input port	[Hall effect sensor mode]
MTR_PORT_PIN_HALL_W	BSP_IO_PORT_04_PIN_08	W phase Hall effect sensor input port	[Hall effect sensor mode]
MTR_HALL_SIGNAL_L_POLARITY	1	Low signal for Hall effect sensor	[Hall effect sensor mode]
MTR_HALL_SIGNAL_H_POLARITY	0	High signal for Hall effect sensor	[Hall effect sensor mode]
ERROR_NONE	0x00	No error	
ERROR_CHANGE_CLK_TIMEOUT	0x01	Timeout error of change of register for clock	
ERROR_CHARGE_CAP_TIMEOUT	0x02	Timeout error of capacitor charge	
ERROR_DRIVING	0x03	ERROR at driving mode	

Table 3-66 **L**ist of Macro definitions “r_mtr_ADC.h”

Macro	Definition value	Description	Remarks
ADC_PRV_ADCSR_ADST	0x1U << 15	Start A/D conversion	

Table 3-67 **L**ist of Macro definitions “r_mtr_GPT_TRHEE_PHASE.h”

Macro	Definition value	Description	Remarks
GPT_THREE_PHASE_OPEN	0x47505433ULL	Flag of GPT opened	
GPT_THREE_PHASE_PRV_GTWP_RESET_VALUE	0xA500U	Preservation of resetting to GPT registers	
GPT_THREE_PHASE_PRV_GTWP_WRITE_PROTECT	0xA501U	Preservation of writing to GPT registers	
GPT_THREE_PHASE_PRV_GTCR_MODE_MASK	0x000F0000U	Mask for GPT mode selection	
GPT_THREE_PHASE_PRV_GTCR_COMP2_MODE	0x000D0000U	Selection of GPT complementary PWM mode 2	
GPT_THREE_PHASE_PRV_GTIOR_OUTPUT_DISABLE	0x01000100U	Disable GPT outputs	
GPT_PRV_GTCCRA	0U	GPT compare match register A	
GPT_PRV_GTCCRB	1U	GPT compare match register B	
GPT_PRV_GTCCRC	2U	GPT compare match register C	
GPT_PRV_GTCCRE	3U	GPT compare match register E	
GPT_PRV_GTCCRD	4U	GPT compare match register D	
GPT_PRV_GTCCRF	5U	GPT compare match register F	

Table 3-68 List of Macro definitions “r_mtr_common.h”

Macro	Definition value	Description	Remarks
R_MTR_ADC_SCAN_MASK_VU	0x00100000U	Scan mask for U phase voltage	[Hall effect sensor mode]
R_MTR_ADC_SCAN_MASK_VV	0x00000080U	Scan mask for V phase voltage	[Hall effect sensor mode]
R_MTR_ADC_SCAN_MASK_VW	0x00000040U	Scan mask for W phase voltage	[Hall effect sensor mode]
R_MTR_ADC_SCAN_MASK_A_HALL	~(R_MTR_ADC_SCAN_MASK_VU | R_MTR_ADC_SCAN_MASK_VV | R_MTR_ADC_SCAN_MASK_VW)	Scan mask for A/D conversion group A	[Hall effect sensor mode]
R_MTR_ADC_SCAN_MASK_B_HALL	0x00000000U	Scan mask for A/D conversion group B	[Hall effect sensor mode]
R_MTR_ADCR_IDC	ADC_CHANNEL_0	A/D conversion channel for DC link current	
R_MTR_ADCR_VDC	ADC_CHANNEL_8	A/D conversion channel for inverter bus voltage	
R_MTR_ADCR_VU	ADC_CHANNEL_20	A/D conversion channel for U phase voltage	[Sensorless mode]
R_MTR_ADCR_VV	ADC_CHANNEL_7	A/D conversion channel for V phase voltage	[Sensorless mode]
R_MTR_ADCR_VW	ADC_CHANNEL_6	A/D conversion channel for W phase voltage	[Sensorless mode]
MTR_CLR	0	Clear a flag	
MTR_SET	1	Set a flag	

Table 3-69 List of Macro definitions “rm_mtr_common.h”

Macro	Definition value	Description	Remarks
MTR_TWOPI	2 * 3.14159265359f	2･pi	
MTR_TWOPI_60	MTR_TWOPI / 60	2･pi / 60	
MTR_SQRT_3	1.7320508f	Square root of 3	
MTR_SQRT_2_3	0.8164966f	Square root of 2/3	
MTR_CW	1	CW	
MTR_CCW	-1	CCW	
MTR_WAIT_CNT_3US	20	Wait for 3us	
MTR_WAIT_CNT_5US	30	Wait for 5us	
MTR_WAIT_CNT_20US	80	Wait for 20us	
MTR_WAIT_CNT_300US	1500	Wait for 300us	
MTR_ADC_12BIT	12	12bit ADC	

Table 3-70 List of Macro definitions “rm_mtr_parameter.h” (1/2)

Macro	Definition value	Description	Remarks
MTR_SPEED_PI_LIMIT_V	IP_INPUT_V	Output voltage limit at PI control	
MTR_MCU_ON_V	IP_INPUT_V * 0.8f	MCU stable supply voltage	
MTR_US_MAX_DRIVE_V	IP_INPUT_V * 0.96f	Maximum output voltage	
MTR_US_MIN_DRIVE_V	IP_INPUT_V * 0.05f [Sensorless and Current control mode] IP_INPUT_V * 0.0f [other modes]	Minimum output voltage	
MTR_MAX_DRIVE_V	FIX_fromfloat(MTR_US_MAX_DRIVE_V * PU_SF_VOLTAGE, MTR_Q_VOLTAGE)	Maximum output voltage	
MTR_MIN_DRIVE_V	FIX_fromfloat(MTR_US_MIN_DRIVE_V * PU_SF_VOLTAGE, MTR_Q_VOLTAGE)	Minimum output voltage	
MTR_RECIV_COEF	(int32_t) (2 * PU_BASE_VOLTAGE_V / IP_VDC_RANGE * (1 << MTR_Q_VOLTAGE))	Inverse of based voltage	
MTR_US_SPEED_CALC_BASE	MTR_SPEED_CALC_TIMER_FREQ * 1000 * MTR_TWOPI	Calculation parameter to convert the timer counter for rotational speed	

Tables 3-71 List of Macro definitions “rm_mtr_parameter.h” (2/2)

Macro	Definition value	Description	Remarks
MTR_SPEED_CALC_BASE	FIX_fromfloat(MTR_US_SPEED_CALC_BASE * PU_SF_AFREQ, MTR_SPEED_CALC_BASE_SHIFT)	Calculation parameter to convert the timer counter for rotational speed	
MTR_US_OL_SPEED_CALC_BASE	MTR_CARRIER_FREQ * 1000 * MTR_TWOPI / MTR_PATTERN_NUM	Calculation parameter to convert rotational speed to the timer counter in open-loop control mode	
MTR_OL_SPEED_CALC_BASE	FIX_fromfloat(MTR_US_OL_SPEED_CALC_BASE * PU_SF_AFREQ, MTR_SPEED_CALC_BASE_SHIFT)	Calculation parameter to convert rotational speed to the timer counter in open-loop control mode	
MTR_SPEED_CALC_BASE_1ST	MTR_SPEED_CALC_BASE / 6	Calculation parameter to convert the timer counter for rotational speed at the first speed calculation	
MTR_SPEED_CALC_BASE_2ND	MTR_SPEED_CALC_BASE / 3	Calculation parameter to convert the timer counter for rotational speed at the second speed calculation	
MTR_SPEED_CALC_BASE_3RD	MTR_SPEED_CALC_BASE / 2	Calculation parameter to convert the timer counter for rotational speed at the thrid speed calculation	
MTR_SPEED_CALC_BASE_4TH	FIX_fromfloat(MTR_US_SPEED_CALC_BASE * 2 / 3 * PU_SF_AFREQ, MTR_Q_AFREQ)	Calculation parameter to convert the timer counter for rotational speed at the fourth speed calculation	
MTR_SPEED_CALC_BASE_5TH	FIX_fromfloat(MTR_US_SPEED_CALC_BASE * 5 / 6 * PU_SF_AFREQ, MTR_Q_AFREQ)	Calculation parameter to convert the timer counter for rotational speed at the fifth speed calculation	
MTR_BIT_SFT_NUM_FOR_DUTY_CALC	19	Number of bits shift for duty calculation	
MTR_BIT_SFT_FOR_DUTY_CALC	((uint32_t) (1)) << MTR_BIT_SFT_NUM_FOR_DUTY_CALC	Bits shifted to the left to improve accuracy of diuty calculation	
MTR_LIMIT_IDC	MP_RATED_CURRENT * MTR_SQRT_3	Speed PI output limit value	[Current control mode]
MTR_I_LIMIT_CURRENT	MP_RATED_CURRENT * MTR_SQRT_3	Limit value for speed PI integral term output	[Current control mode]
MTR_MIN_IDC_REF	0	Minimum value for reference current	[Current control mode]
MTR_I_LIMIT_VQ	IP_INPUT_V * 0.9f	Current PI Integral term limit	

Table 3-72 **L**ist of Macro definitions “r_mtr_ol2cl_ctrl.h”

Macro	Definition value	Description	Remarks
MTR_PHASE_LEAD	0	Leading phase of rotor	[Sensorless mode]
MTR_PHASE_LAG	1	Lagging phase of rotor	[Sensorless mode]
MTR_1ST_ZC_CNT	1	First zero-crossing detected during transition	[Sensorless mode] [Voltage control mode]

Table 3-73 List of Macro definitions “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	

Table 3-74 List of Macro definitions “r_mtr_120.h” (1/2)

Macro	Definition value	Description	Remarks
MTR_TIMEOUT_CNT	(uint16_t) (1.0f / ((float)MP_POLE_PAIRS * CP_MIN_SPEED_RPM / 60) * 1000 * 2)	Counts for timeout	
MTR_PATTERN_CW_V_U	2	Voltage patten at CW rotation	
MTR_PATTERN_CW_W_U	3	Voltage patten at CW rotation	
MTR_PATTERN_CW_W_V	1	Voltage patten at CW rotation	
MTR_PATTERN_CW_U_V	5	Voltage patten at CW rotation	
MTR_PATTERN_CW_U_W	4	Voltage patten at CW rotation	
MTR_PATTERN_CW_V_W	6	Voltage patten at CW rotation	
MTR_PATTERN_NUM	6	Number of voltage patterns	
MTR_TIMER_CNT_BUFFER_SIZE	6	Size of a buffer for timer counts	
MTR_HALL2OL_REV_SPEED_RAD	FIX_fromfloat(CP_HALL2OL_REV_SPEED_RPM * PU_SF_RPM_RAD, MTR_Q_AFREQ)	Transition speed when rotor reverses	[Hall effect sensor mode]
MTR_PATTERN_CCW_V_U	5 [Hall effect sensor mode] 3 [Sensorless mode]		
MTR_PATTERN_CCW_V_W	1 [Hall effect sensor mode] 2 [Sensorless mode]		
MTR_PATTERN_CCW_U_W	3 [Hall effect sensor mode] 6 [Sensorless mode]		
MTR_PATTERN_CCW_U_V	2 [Hall effect sensor mode] 4 [Sensorless mode]		
MTR_PATTERN_CCW_W_V	6 [Hall effect sensor mode] 5 [Sensorless mode]		
MTR_PATTERN_CCW_W_U	4 [Hall effect sensor mode] 1 [Sensorless mode]		
MTR_DRAW_IN_1ST_PATTERN	1	Voltage pattern at the first draw-in	[Sensorless mode]
MTR_DRAW_IN_2ND_PATTERN	2	Voltage pattern at the second draw-in	[Sensorless mode]
MTR_ERROR_NONE	0x0000	No error	
MTR_ERROR_OVER_CURRENT_HW	0x0001	Hardware over current error	
MTR_ERROR_OVER_VOLTAGE	0x0002	Over voltage error	
MTR_ERROR_OVER_SPEED	0x0004	Over speed error	
MTR_ERROR_HALL_TIMEOUT	0x0008	Timeout error for Hall effect sensor mode	
MTR_ERROR_BEMF_TIMEOUT	0x0010	Timeout error for sensorless control mode	

Table 3-75 List of Macro definitions “r_mtr_120.h” (2/2)

Macro	Definition value	Description	Remarks
MTR_ERROR_HALL_PATTERN	0x0020	Hall pattern error	
MTR_ERROR_BEMF_PATTERN	0x0040	BEMF pattern error	
MTR_ERROR_UNDER_VOLTAGE	0x0080	Under voltage error	
MTR_ERROR_OVERCURRENT_SW	0x0100	Software over current error	
MTR_ERROR_CHARGE_CAP_TIMEOUT	0x0400	Time-out error during charging capacitor	
MTR_ERROR_UNKNOWN	0xffff	Undefined error	
MTR_REF_ZERO_CONST	0	Reference value zero mode	
MTR_REF_MANUAL	1	Reference value manual input mode	
MTR_REF_SPEED_PI_OUTPUT	2	Reference value PI output mode	
MTR_SPEED_ZERO_CONST	0	Reference speed zero mode	
MTR_SPEED_MANUAL	1	Reference speed manual input mode	
MTR_STOP	0x00	Motor drive stop	
MTR_DRIVE	0x81	Motor drive	
MTR_DRIVE_INIT	0x82	Motor drive initialization	
MTR_DRIVE_START	0x83	Motor drive start	
MTR_OFFSET_CALC_EXE	0x84	Execution of current offset detection process	
MTR_DRIVE_END	0x01	Stopping the drive	
MTR_MASK_RUN_MODE	0x80	Mask for motor drive	
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_FORCE_SYNC_ACCEL	0x45	Open-loop accelerate speed control	
MTR_CTRL_CLL_OPL2CLL	0xC0	Open-loop to closed-loop control	
MTR_CTRL_CLL_SPEED_CTRL	0xC1	Closed-loop speed control	
MTR_CTRL_DRAWIN	0x00	Draw-in control	
MTR_CTRL_DRVE	0x40	Drive control	
MTR_DRIVE_OPL	0x00	Open-loop speed control	
MTR_DRIVE_CLL	0x80	Closed-loop speed control	

Table 3-76 List of Macro definitions “RM_DSP_RA2_GCC_32BIT.h”

Macro	Definition value	Description	Remarks
FIX_fromfloat(x, n)	(int32_t)ldexpf((float)(x),(n))	Convert float type value x to fixed-point value y in Qn format	
FIX_mul_rs(a, b, sar)	(int32_t) (((int32_t) (a) >> ((sar) >> 1)) * ((int32_t) (b) >> ((sar) - ((sar) >> 1))))	Multiply and rghit shift for fix-point values in 16 bits	
FIX_mul_rs_sign(a, b, sar)	(int32_t) (((b) >= 0) ? ((int32_t) (((int32_t) (a) >> ((sar)>>1)) * ((int32_t) (b) >> ((sar) - ((sar) >> 1))))) : ~((int32_t) (((int32_t) (a) >> ((sar) >> 1)) * ((int32_t) (~(b) + 1) >> ((sar) - ((sar) >>1)))) + 1))	Multiply and rghit shift for fix-point values in 16 bits (signed)	

3.5 Control flows (flow charts)

3.5.1 Main process

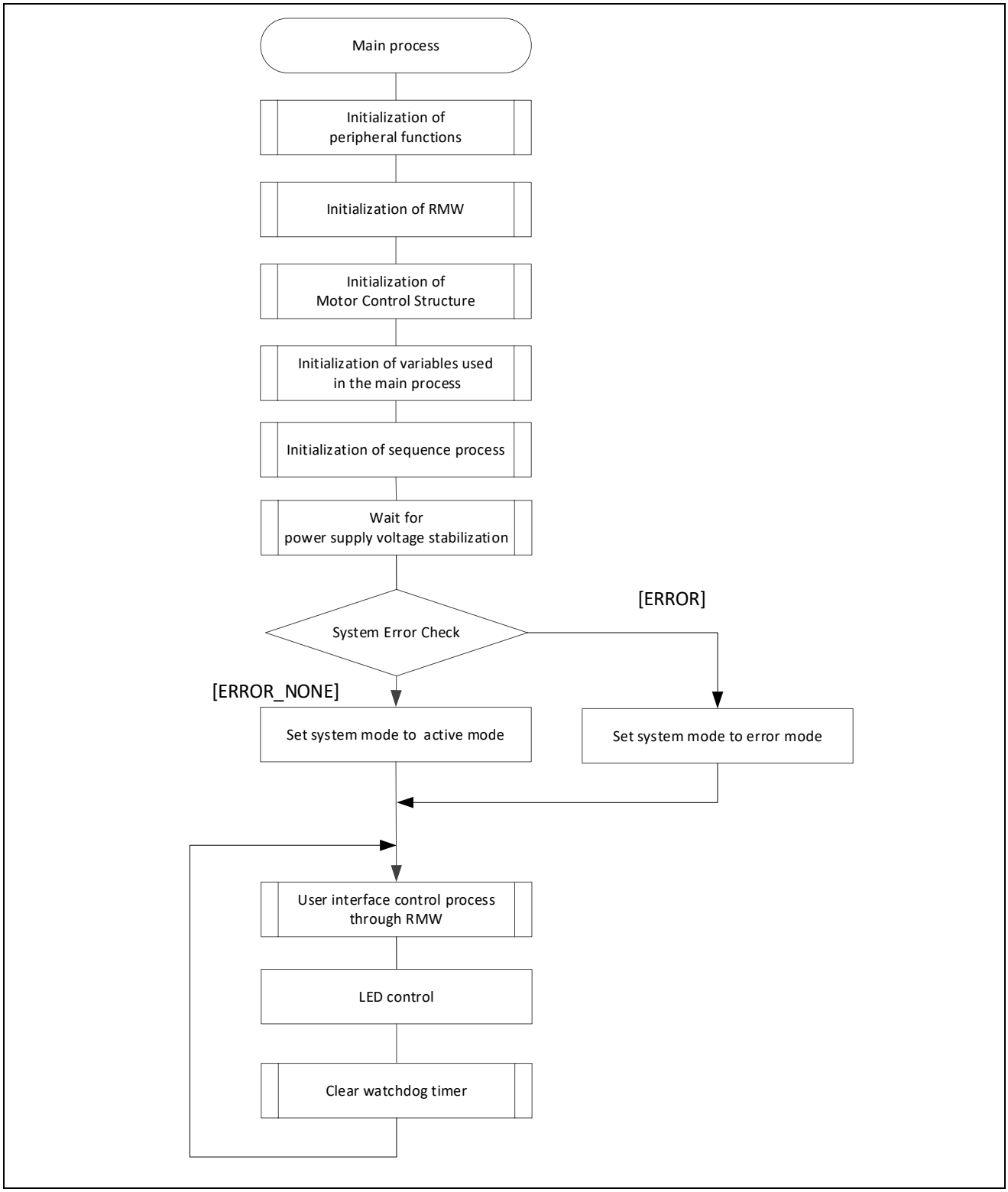


Figure 3-19 Main process flow chart

3.5.2 Carrier cycle interrupt (Sensorless mode)

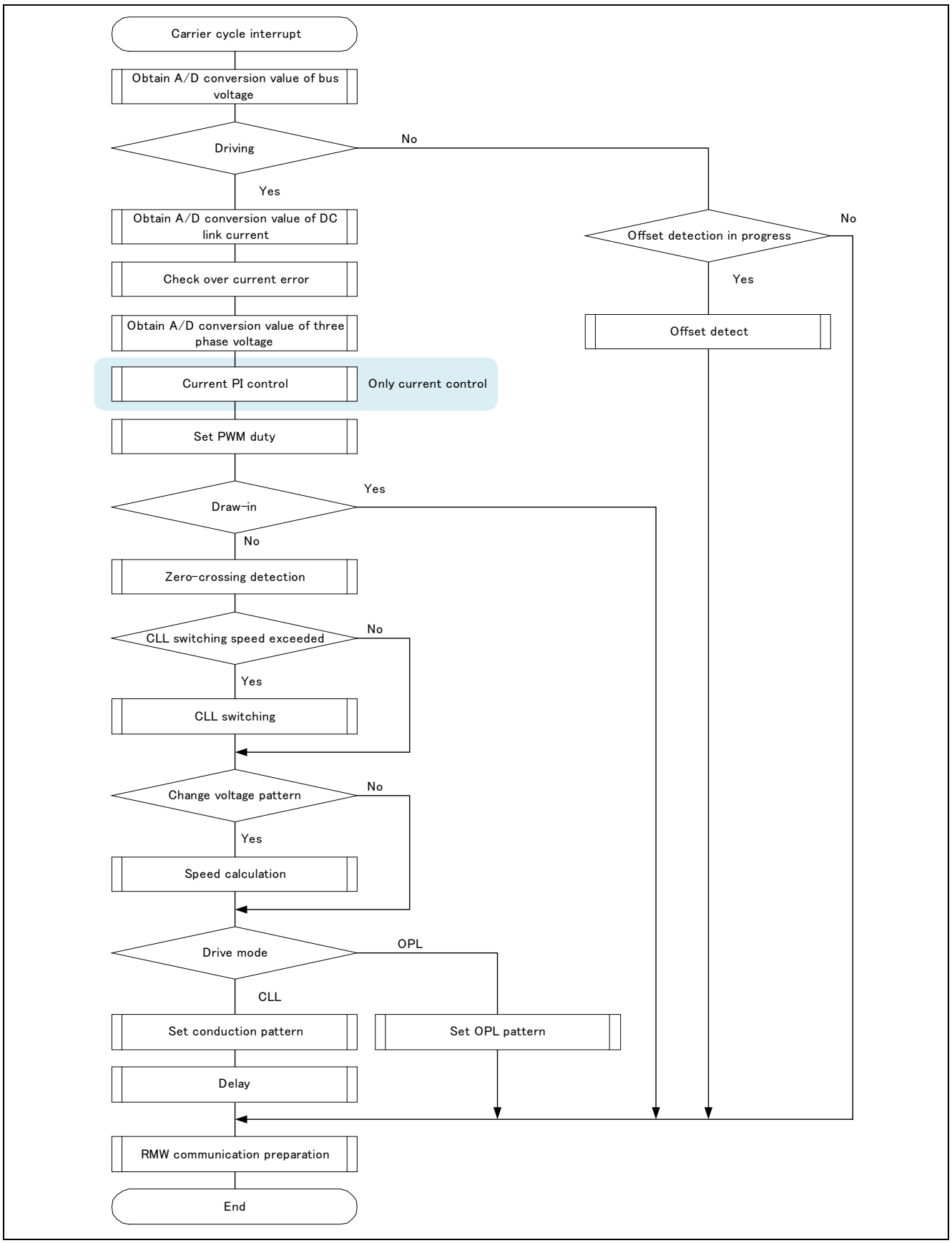


Figure 3-20 Carrier cycle interrupt (Sensorless mode) flow chart

3.5.3 Speed control interrupt

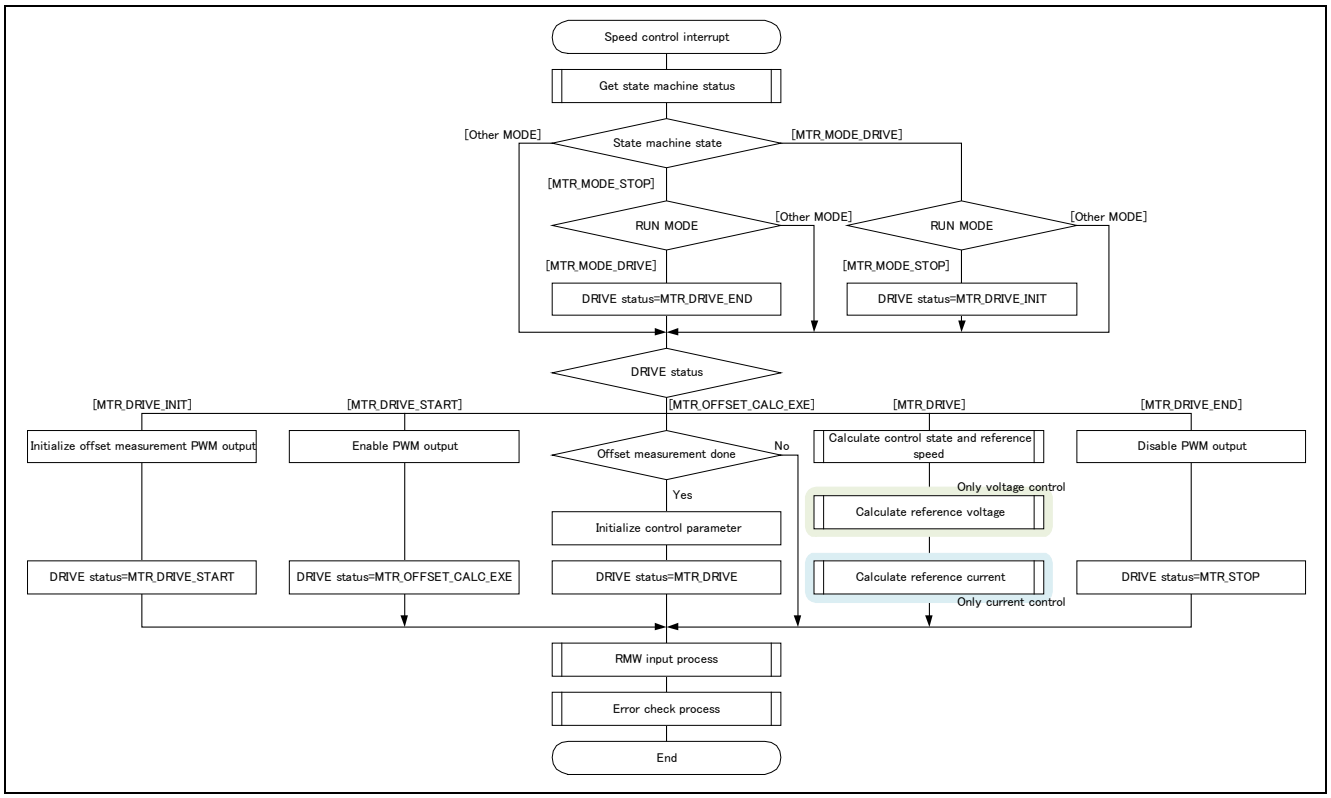


Figure 3-21 Speed control interrupt flow chart

3.5.4 Set control status and calculate reference speed (sensorless mode)

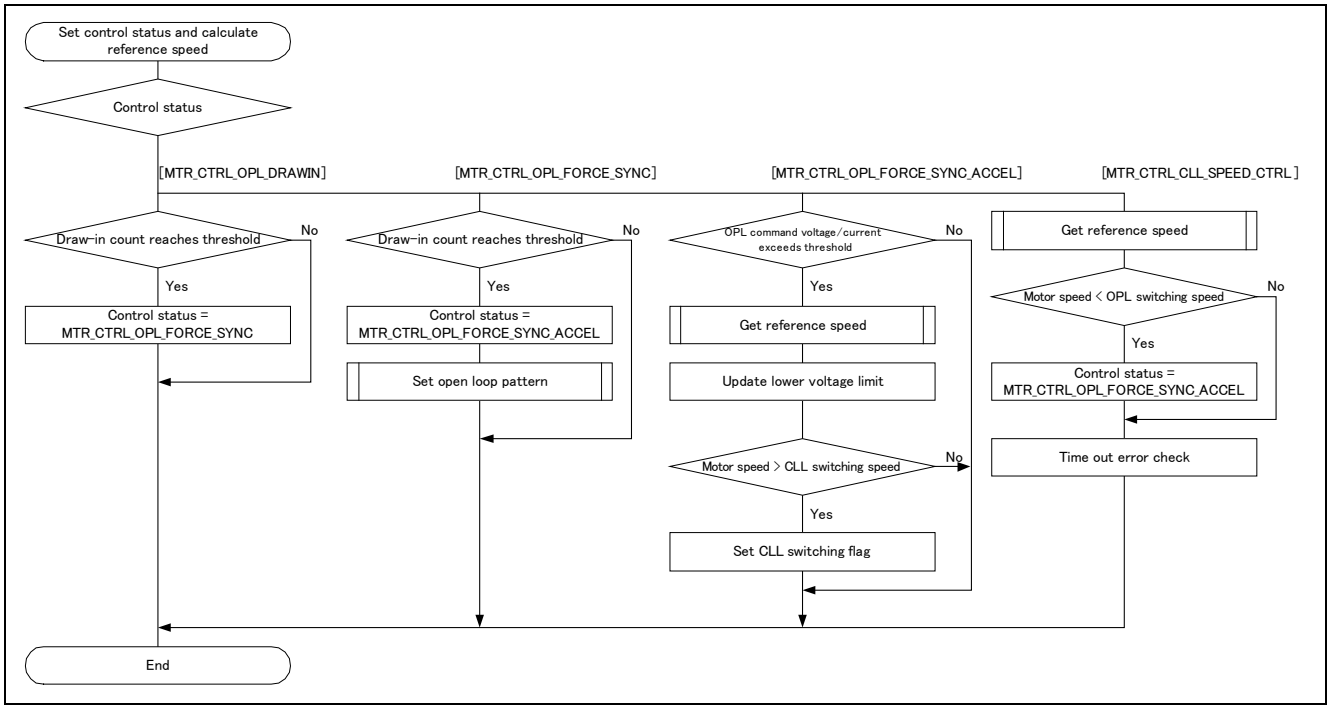


Figure 3-22 Set control status and calculate reference speed **(sensorless mode)** flow chart

3.5.5 Delay timer interrupt

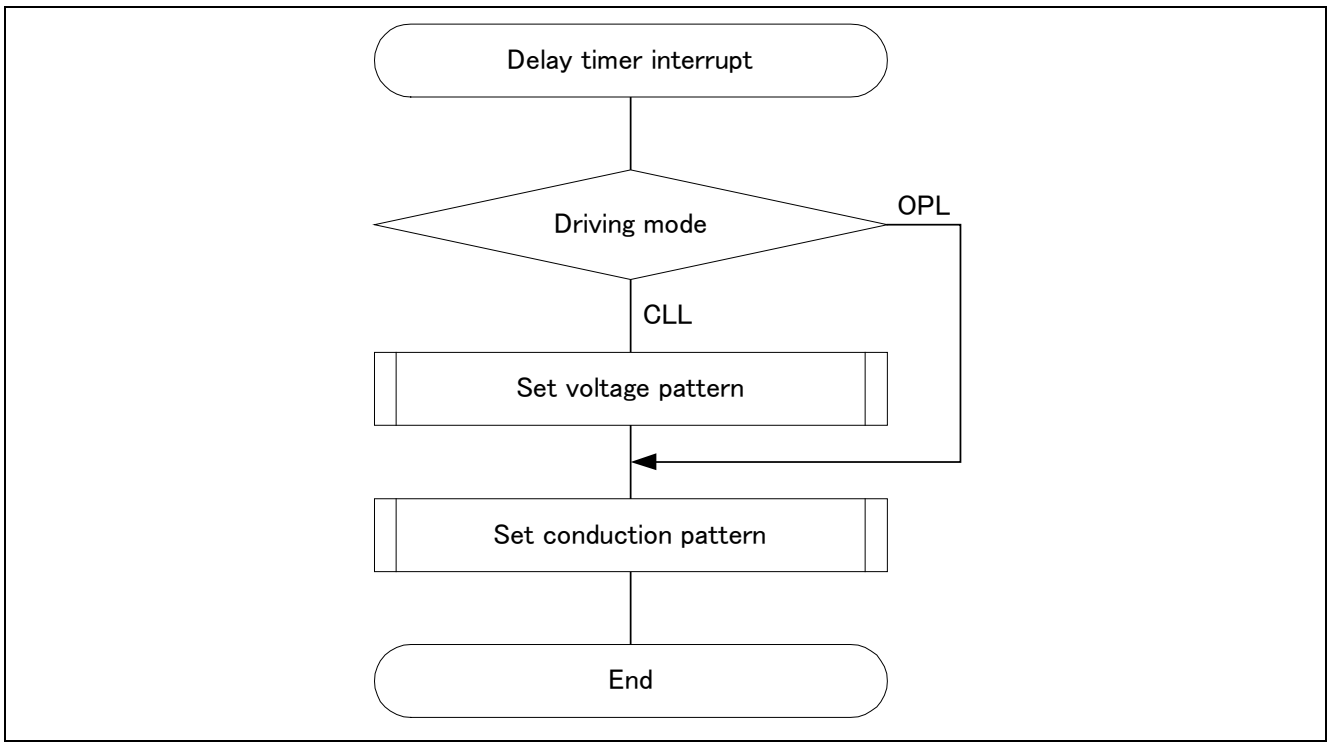


Figure 3-23 Delay timer interrupt flow chart

3.5.6 Carrier cycle interrupt (Hall effect sensor mode)

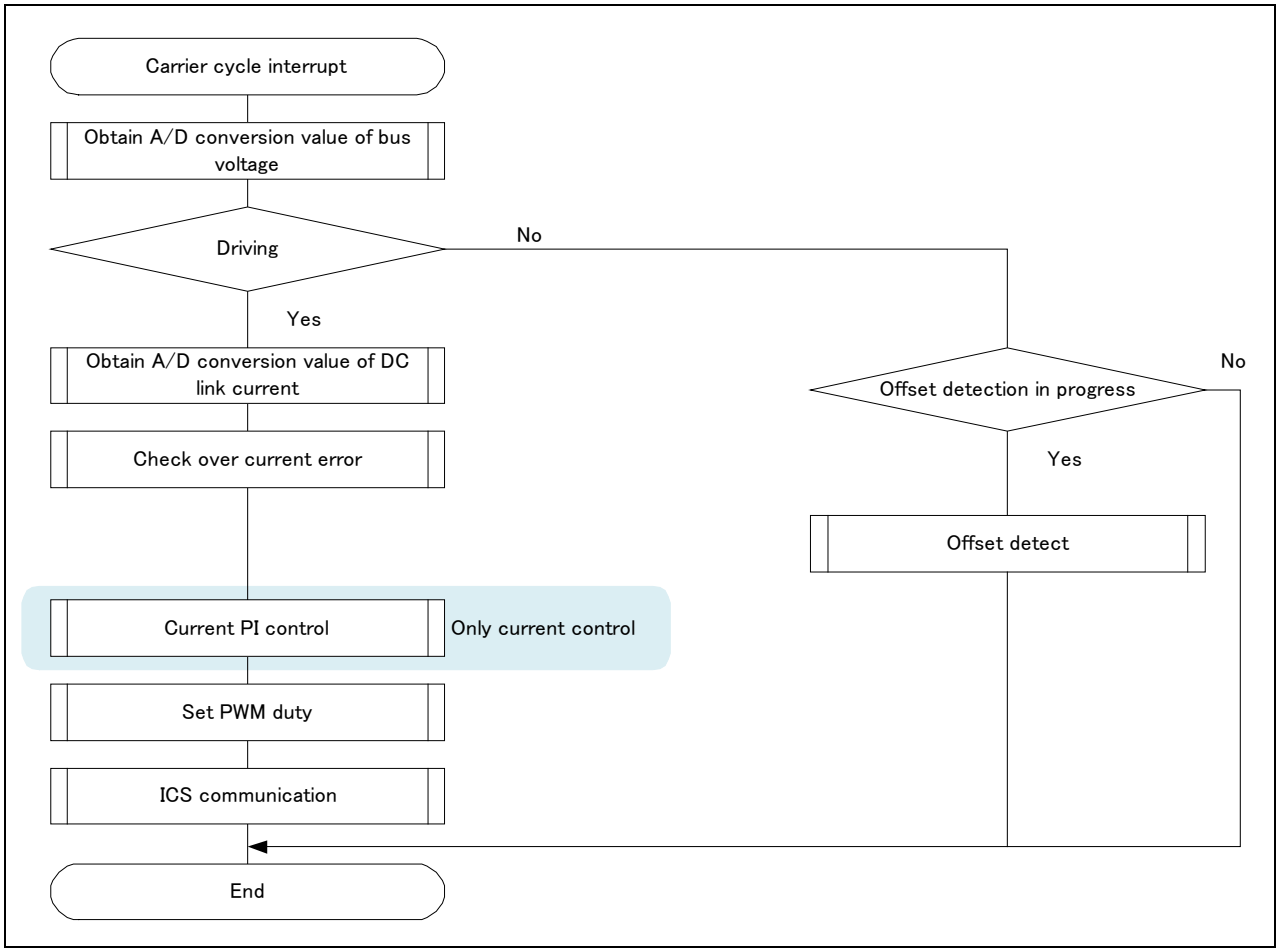


Figure 3-24 Carrier cycle interrupt (Hall effect sensor mode) flow chart

3.5.7 Set control status and calculate reference speed (Hall effect sensor mode)

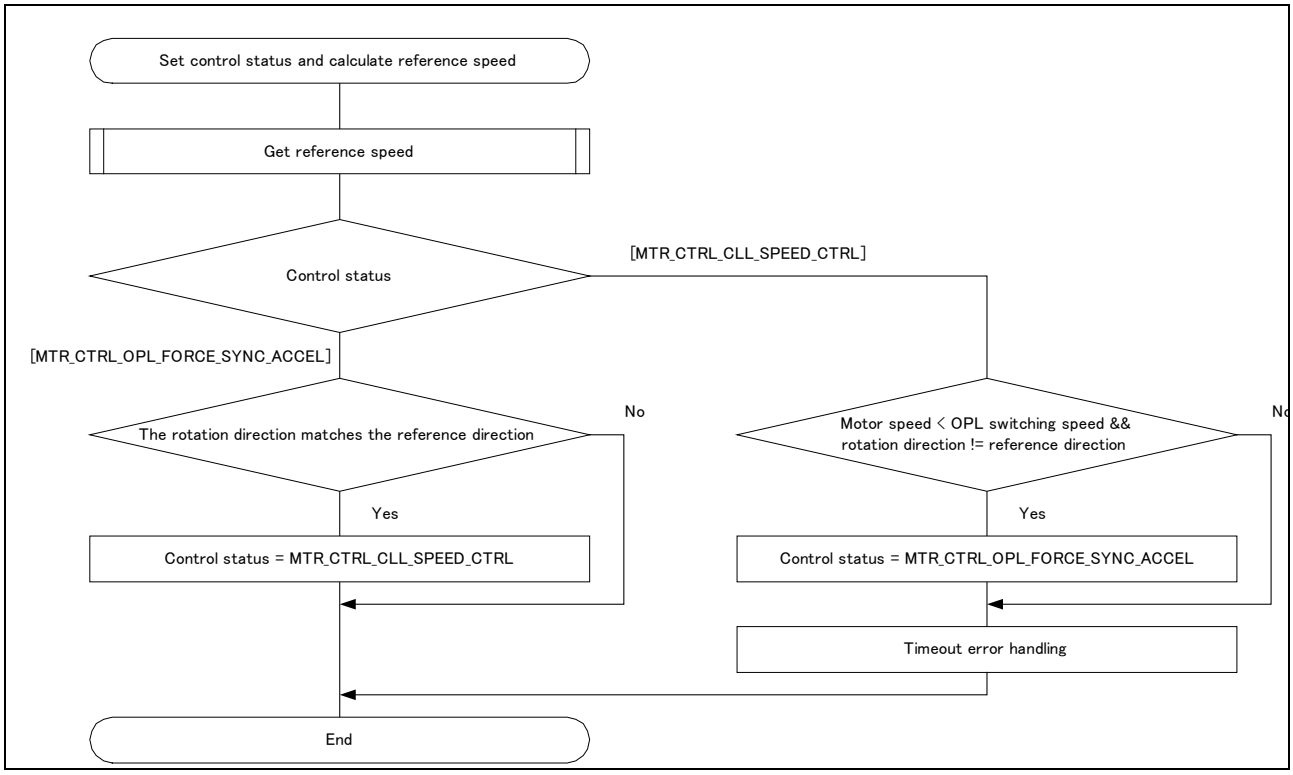


Figure 3-25 Set control status and calculate reference speed (Hall effect sensor mode) flow chart

3.5.8 Hall effect sensor interrupt

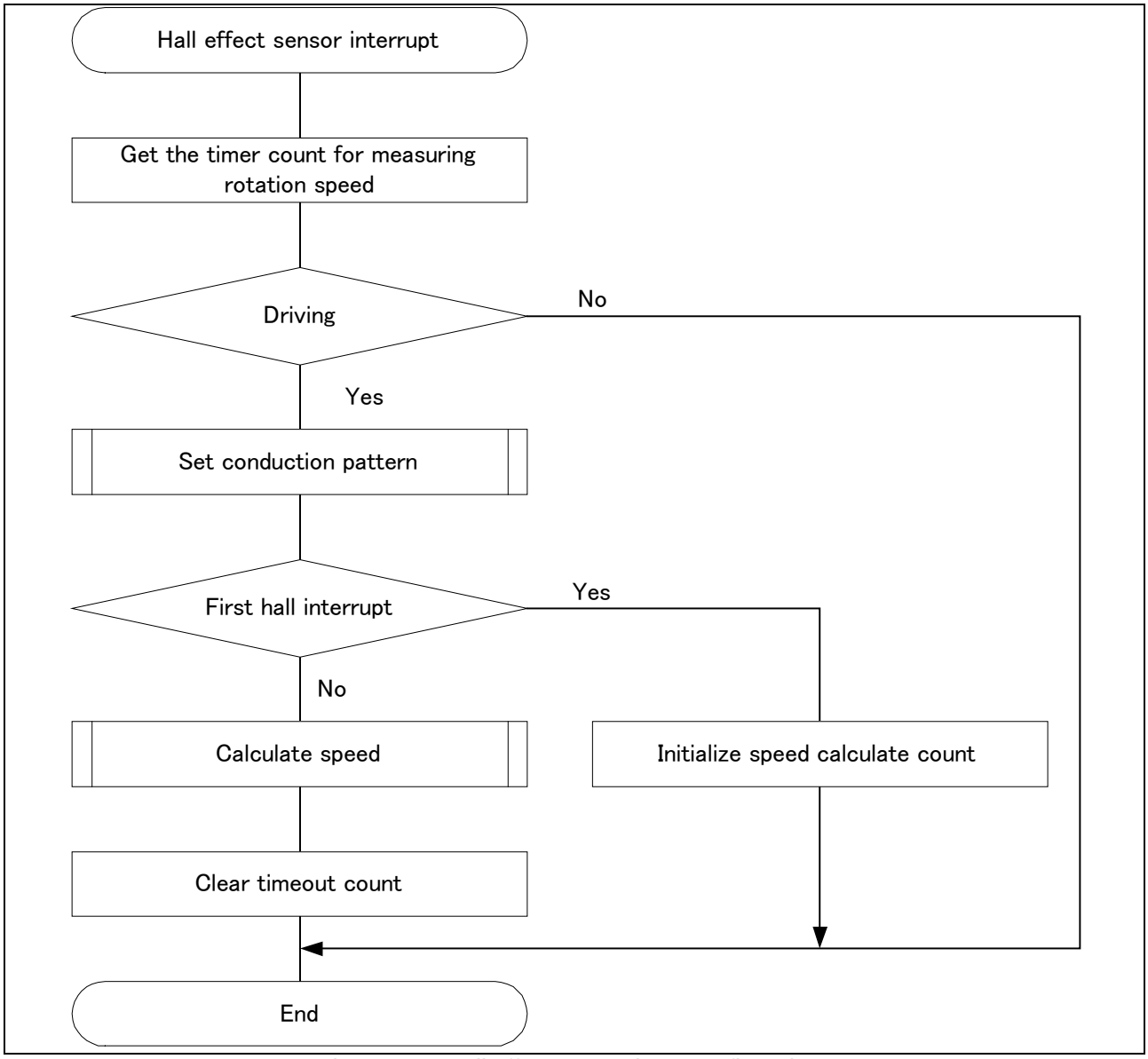


Figure 3-26 Hall effect sensor interrupt flow chart

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.

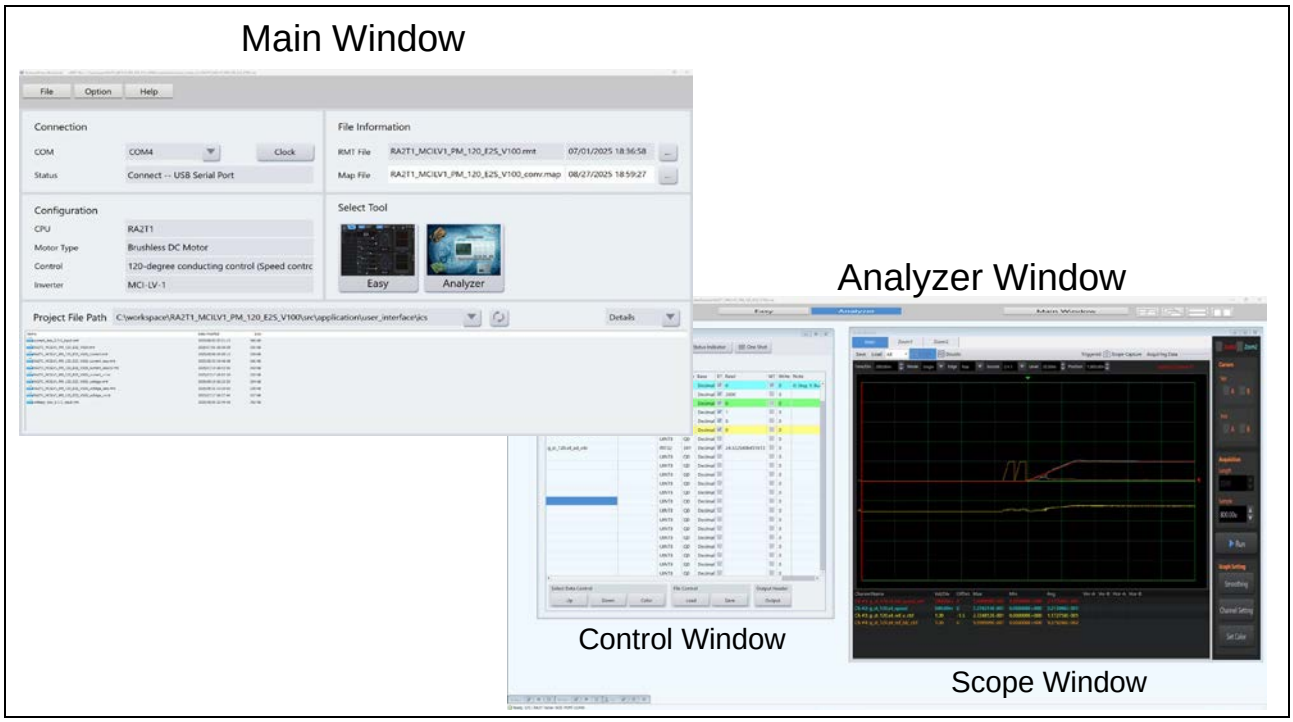


Figure 4-1 Renesas Motor Workbench– Appearance

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 ‘[Project Folder]/application/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 and Table 4-2 show the 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.com_s2_enable_write. However, note that variables with (*) do not depend on g_st_com.com_s2_enable_write.

Table 4-1 List of Variables for Analyzer [1/2]

Variable	Type	Content	Remarks ([ ]: Corresponding variable name)
com_u1_run_event (*)	uint8_t	Input event and change run mode 0: Stop event 1: Drive event 2: Error event 3: Reset event	[g_u1_run_event]
g_st_com.com_u1_direction	int8_t	Rotational direction 0: CW 1: CCW	[g_st_120.s1_ref_dir]
g_st_com.com_u2_mtr_pp	uint16_t	Number of pole pairs	[g_st_120.st_carrier.st_motor.u2_mtr_pp]
g_st_com.com_f4_mtr_r	float	Resistance [ohm]	[g_st_120.st_carrier.st_motor.s4_mtr_r]
g_st_com.com_f4_mtr_ld	float	D-axis inductance[H/rad]	[g_st_120.st_carrier.st_motor.s4_mtr_ld]
g_st_com.com_f4_mtr_lq	float	Q-axis inductance[H/rad]	[g_st_120 .st_carrier.st_motor.s4_mtr_lq]
g_st_com.com_f4_mtr_ke	float	BEMF voltage constant [V･s/rad]	[g_st_120.st_carrier.st_motor.s4_mtr_ke]
g_st_com.com_f4_mtr_j	float	Rotor inertia [kg･m^2/(rad･s^2)]	[g_st_120.st_carrier.st_motor.s4_mtr_j]
g_st_com.com_f4_ref_lim_v	float	Lower limit of voltage reference [V]	[g_st_120.s4_ref_lim_v]
g_st_com.com_f4_ramp_lim_speed_rpm	float	Limit of acceleration [rpm/ms]	[g_st_120.st_speed.s4_ramp_lim_speed]
g_st_com.com_f4_ref_speed_rpm	float	Reference rotational speed [rpm]	[g_st_120.st_speed.s4_ref_speed]
g_st_com.com_u2_offset_calc_cnt	uint16_t	Current offset detection time	[g_st_120.st_carrier.u4_offset_calc_cnt]
g_st_com.com_f4_ramp_lim_v	float	Limit of variation of voltage [V/ms]	[g_st_120.st_speed.s4_ramp_lim_v]
g_st_com.st_com.f4_speed_omega	float	Frequency for speed PI control [Hz]	[g_st_ctrl_params.st_com.f4_speed_omega]
g_st_com.st_com.f4_asr_zeta	float	Damping factor for speed PI control	[g_st_ctrl_params.st_com.f4_asr_zeta]
g_st_com.com_f4_ramp_lim_current	float	Limit of variation of current [A/ms]	[g_st_120.st_acr.s4_ramp_lim_current]
g_st_com.com_f4_mtr_d0	float	Static friction coefficient [kg･m^2/(rads^2)]	[g_st_120.st_speed.st_current_asr.s4_d0_div_pm]
g_st_com.com_f4_mtr_d1	float	Kinetic friction coefficient [kg･m^2/(rad^2･s)]	[g_st_120.st_speed.st_current_asr.s4_d1_div_p2m]
g_st_com.st_com.f4_acr_nf_hz	float	Current PI control natural frequency [Hz]	[g_st_ctrl_params.st_com.f4_acr_nf_hz]
g_st_com.st_com.f4_asr_nf_hz	float	Speed PI control natural frequency [Hz]	[g_st_ctrl_params.st_com.f4_asr_nf_hz]
g_st_com.st_com.f4_asr_ki_aug	float	Augmentation rate for integral part of ASR	[g_st_ctrl_params.st_com.f4_asr_ki_aug]
g_st_com.com_f4_ref_v _start	float	Reference voltage at start-up [V]	[g_st_120.st_speed.st_hall_speed.s4_ref_v_start]
g_st_com.com_f4_ref_idc_start	float	Reference current at start-up [A]	[g_st_120.st_speed.st_hall_speed.s4_ref_idc_start]
g_st_com.com_f4_ref_v_draw_in	float	Reference voltage at draw-in [V]	[g_st_120.st_speed.st_less_speed.s4_ref_v_draw_in]

Table 4-2 List of Variables for Analyzer [1/2]

Variable	Type	Content	Remarks ([ ]: Corresponding variable name)
g_st_com.com_f4_ref_v_ol	float	Reference voltage at open-loop drive [V]	[g_st_120.st_speed.st_less_speed.s4_ref_v_buff_ol]
g_st_com.com_f4_ref_idc_draw_in	float	Reference current at draw-in [A]	[g_st_120.st_speed.st_less_speed.s4_ref_idc_draw_in]
g_st_com.com_f4_ref_idc_ol	float	Reference current at open-loop drive [A]	[g_st_120.st_speed.st_less_speed.s4_ref_idc_buff_ol]
g_st_com.com_f4_ol2cl_speed_rpm	float	Speed allowed to transition to PI control[rpm]	[g_st_120.st_less_control.st_ol2cl.s4_change_speed]
g_st_com.com_f4_ol2cl_ramp_speed_rpm	float	Acceleration at transition to PI control [rpm/ms]	[g_st_120.st_less_control.st_ol2cl.s4_ramp_speed]
g_st_com.com_s4_angle_shift_adjust	int32_t	Adjust delay counts	[g_st_120.st_less_control.s4_angle_shift_adjust]
g_st_com.com_s2_enable_write	int16_t	Variable to allow to input RMW structure	[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, use ‘Control Window’ shown in Figure 4-2. 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.com_f4_ref_speed_rpm”, and “g_st_com.com_s2_enable_write.”
  - ② Input a reference rotational speed value in the [Write] box of “g_st_com.com_f4_ref_speed_rpm.”
  - ③ Click the “Write” button.
  - ④ Click the “Read” button. Confirm the [Read] box of “g_st_com.com_f4_ref_speed_rpm” and “g_s2_enable_write.”
  - ⑤ Input the value in the [Read] box of “g_s2_enable_write”, confirmed in step (4), in the [Write] box of “g_st_com.com_s2_enable_write.”
  - ⑥ Input a value of “1” in the [Write] box of “com_u1_run_event.”
  - ⑦ Click the “Write” button.

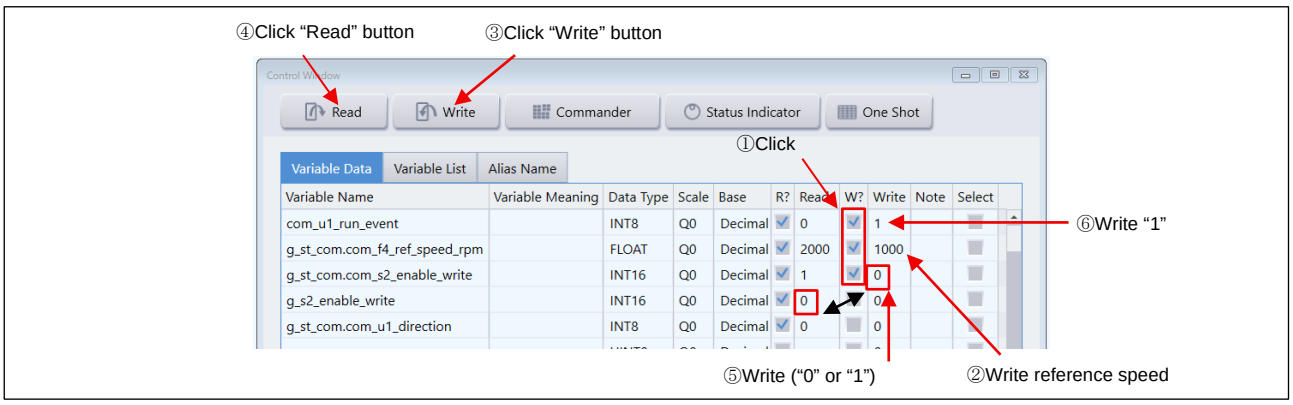


Figure 4-2 Procedure - Driving the motor

- 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.

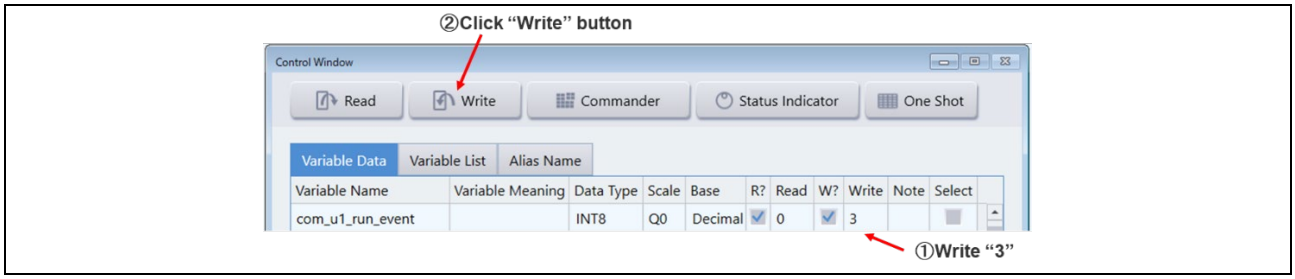


Figure 4-4 Procedure - Error cancel operation

4.4 Method to switch between control modes

In this sample software, there are four control modes provided. By changing the definition value of “MTRCONF_MODE” in the “r_mtr_config.h” file, you can switch between the four control modes as shown in the following table.

Table 4-3 Definition of control mode

Macro	Content	Remarks
SENSORLESS_CURRENT	Sensorless current mode	Default setting
SENSORLESS_VOLTAGAE	Sensorless voltage mode	
HALL_CURRENT	Hall effect sensor-based current mode	
HALL_VOLTAGE	Hall effect sensor-based voltage mode	

You can program the device to operate in the specified control mode by changing the definition value and recompiling in the IDE.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	29.08.2025	—	First edition issued
1.01	30.06.2026	—	- 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/.

© 2026 Renesas Electronics Corporation. All rights reserved.