RL78/G24

120-degree conducting control for permanent magnetic synchronous motor

Summary

This application note explains the sample programs driving a permanent magnetic synchronous motor in the 120-degree conducting method using the RL78/G24 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.

- RL78/G24(R7F101GLG2DFB)

Target sample programs

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

- RL78G24_MCEK_120_CSP_CC_V100 (IDE: CS+ for CC)
- RL78G24_MCEK_120_E2S_CC_V100 (IDE: e^{2}studio)

Reference

- Application note: ‘120-degree conducting control of permanent magnetic synchronous motor: algorithm’ (R01AN2657EJ0120)
- RL78/G24 Group User’s Manual: Hardware (R01UH0961EJ0110)
- Renesas Motor Workbench 3.1.2 User’s Manual (R21UZ0004EJ0402)
- RL78/G24 Motor Control Evaluation Kit User’s Manual (R12UT0021EJ0100)

Contents

1. Overview

This application note explains how to implement the 120-degree conducting control sample programs of permanent magnetic synchronous motor (PMSM) using the RL78/G24 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
RL78/G24 (R7F101GLGFB)	RL78/G24 CPU Card (RTK0EMG240C00000BJ) (Note 1) Inverter board (RTK0EMGPLVB00000BJ) (Note 1) Communication board (RTK0EMXC90Z00000BJ) (Note 1)	TSUKASA (Note 2) TG-55L

Table 1-2 Development Environment of the Sample Programs (S/W)

CS+ version	e2studio version	Build tool version	Smart Configurator	Debug tool
V9.11.00	2024-04	CC-RL V1.12.01	V1.9.0	-

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

Notes:

1. The RL78 / G24 CPU CARD (RTK0EMG240C00000BJ), Inverter board (RTK0EMGPLVB00000BJ) and communication board (RTK0EMXC90Z00000BJ) are products of Renesas Electronics Corporation.
2. TG-55L is a product of TSUKASA ELECTRIC.
   TSUKASA ELECTRIC. (https://www.tsukasa-d.co.jp/en/)

2. System overview

An overview of this system is provided below.

2.1 Hardware Specifications

RL78/G24 Motor Control Evaluation Kit (RTK0EMG24SS00000BJ) consists of Inverter board, CPU board, and communication board. Each specification is shown below.

Table 2-1 Specifications of RL78/G24 Motor Control Evaluation Kit (RTK0EMG24SS00000BJ)

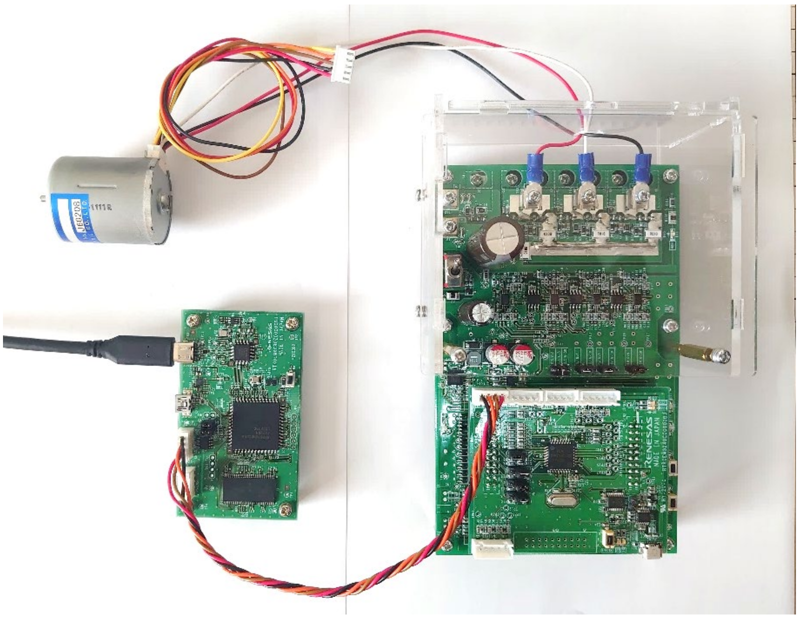
Item	Specification	
Product name	RL78/G24 Motor Control Evaluation Kit	
Kit model name	RTK0EMG24SS00000BJ	
Kit configuration	Low voltage Inverter Board LV400	RTK0EMGPLVB00000BJ
	RL78/G24 CPU Card	RTK0EMG240C00000BJ
	Communication Board MC-COM	RTK0EMXC90Z00000BJ
	PMSM	TG-55L-KA (TSUKASA Co., Ltd.)
Isolation	Inverter board - CPU card: Non-isolated Communication board - CPU board: Isolated	
External view	 Note: The actual product may differ from this photo.	
Board size	Inverter board : 100mm(W)×160mm(H) CPU card : 79mm(W)×66mm(H) Communication board : 89mm(W)×52mm(H)	
Heat dissipation measures	Natural air cooling with heat sinks	
Operating temperature	Room temperature	
Operating humidity	No condensation allowed	

Table 2-2 Specifications of Low voltage Inverter board (RTK0EMGPLVB00000BJ)

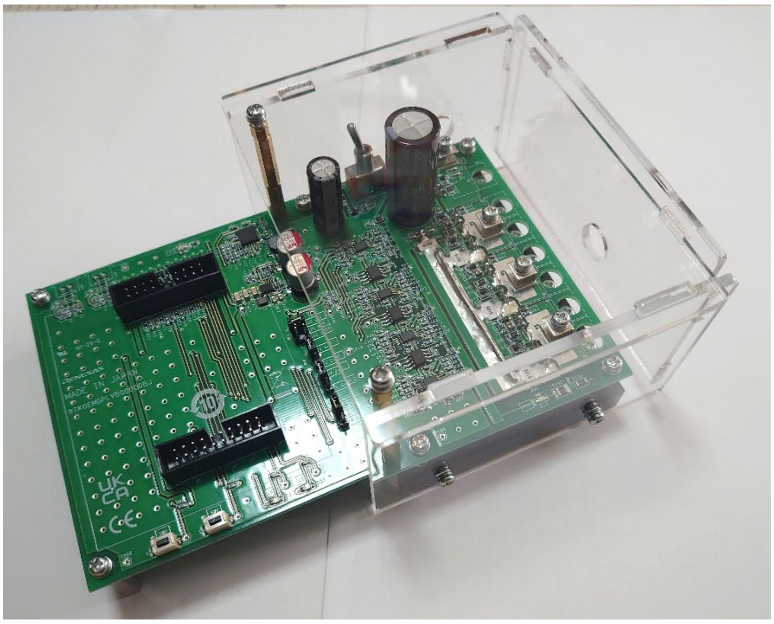
Item	Specification
Product name	Low voltage Inverter Board for Motor Control Evaluation Kit - LVI400
Board model name	RTK0EMGPLVB00000BJ
Operating input voltage	12V to 50V
Maximum output current	30A (peak current for each phase)
Driving motor	3-phase permanent magnet synchronous motor
Current detection method	Detects current using a shunt resistor for 3-phase and DC link
DC bus voltage detection	Detects using a resistance divider circuit
3-phase output voltage detection	Detects using a resistance divider circuit
PWM logic	Positive for both upper and lower arms
Overcurrent detection	Available by specifying the RL78/G24 MCU setting (PWMOPA) (No detection circuit is provided.)
Dead time	1 μs (Recommended value)
Switch	● Two tact switches ● Inverter power toggle switch
LEDs	● Two LEDs ● Inverter Power LED ● +5V power LED
Connectors	● CPU card connector: CNA, CNB ● Two analog signal input connectors: CN1 and CN2
Jumper pins	Current Amplifier Switching: magnification is 1/5/50: JP7 Current Amplifier Switching: magnification is 5/50 times: JP8, JP9, JP10
Screw terminals	● Power input: P, N ● Motor output: U, V, W
External view	 Note: The actual product may differ from this photo.

Table 2-3 Specifications of RL78/G24 CPU Card (RTK0EMG240C00000BJ)

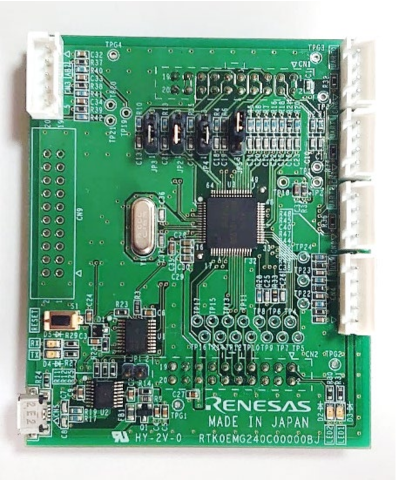
Item		Specification
Product name		RL78/G24 CPU Card for Motor Control Evaluation Kit
Board model name		RTK0EMG240C00000BJ
Mounted MCU	Product name	R7F101GLG2DFB
	CPU maximum operating frequency	48 MHz
	Bit number	16 bit
	Package / Pin number	LQFP / 64pin
	ROM	128 KB
	RAM	12 KB
Input power supply voltage		DC 5V Automatically select one of the following • Power supply from compatible inverter board • Power supply from USB connector
Switch		MCU reset switch
LED		• Two LEDs • Two USB communication LED
Connectors		• Inverter board connector:CN1, CN2 • ABZ encoder input connector:CN3 • UVW Hall signal input connector:CN4 • COM Port Micro USB connector for debugging:CN5 • Three serial communication connectors: CN6, CN7, CN8 • Unused pin connectors:CN9
Jumper pins		• V-phase voltage / Encoder A-phase switching: JP2 • U-phase voltage / Encoder Z-phase switching: JP3 • W phase voltage / Encoder B-phase switching:JP4 • W-phase current / GND switching: JP5
External view		 Note: The actual product may differ from this photo.

Table 2-4 Specifications of Communication board (RTK0EMXC90Z00000BJ)

Item		Specification
Product name		MC-COM Renesas Flexible Motor Control Communication Board
Board model name		RTK0EMXC90Z00000BJ
External view		Note: The actual product may differ from this photo.
Mounted MCU	Product group	RX72N group
	Product name	R5F572NNDDFB
	CPU maximum operating frequency	240MHz
	Bit number	32 bit
	Package / Pin number	LFQFP / 144 pin
	RAM	1M byte
MCU input clock		20MHz (Generate with external crystal oscillator)
Input power supply voltage		• DC 5V • Power is supplied from USB connector
Connector		• USB Type-C connector • USB miniB connector (not available for users) • SCI connector for Renesas Motor Workbench communication
Isolation		• Between SCI connector and MCU • Isolation device Si8622Bc-B-IS(Skyworks Solutions Inc.) Or ISO7421FED (Texas Instruments)
Switch		MCU reset switch

2.2 Hardware configuration

The hardware configuration is shown below.

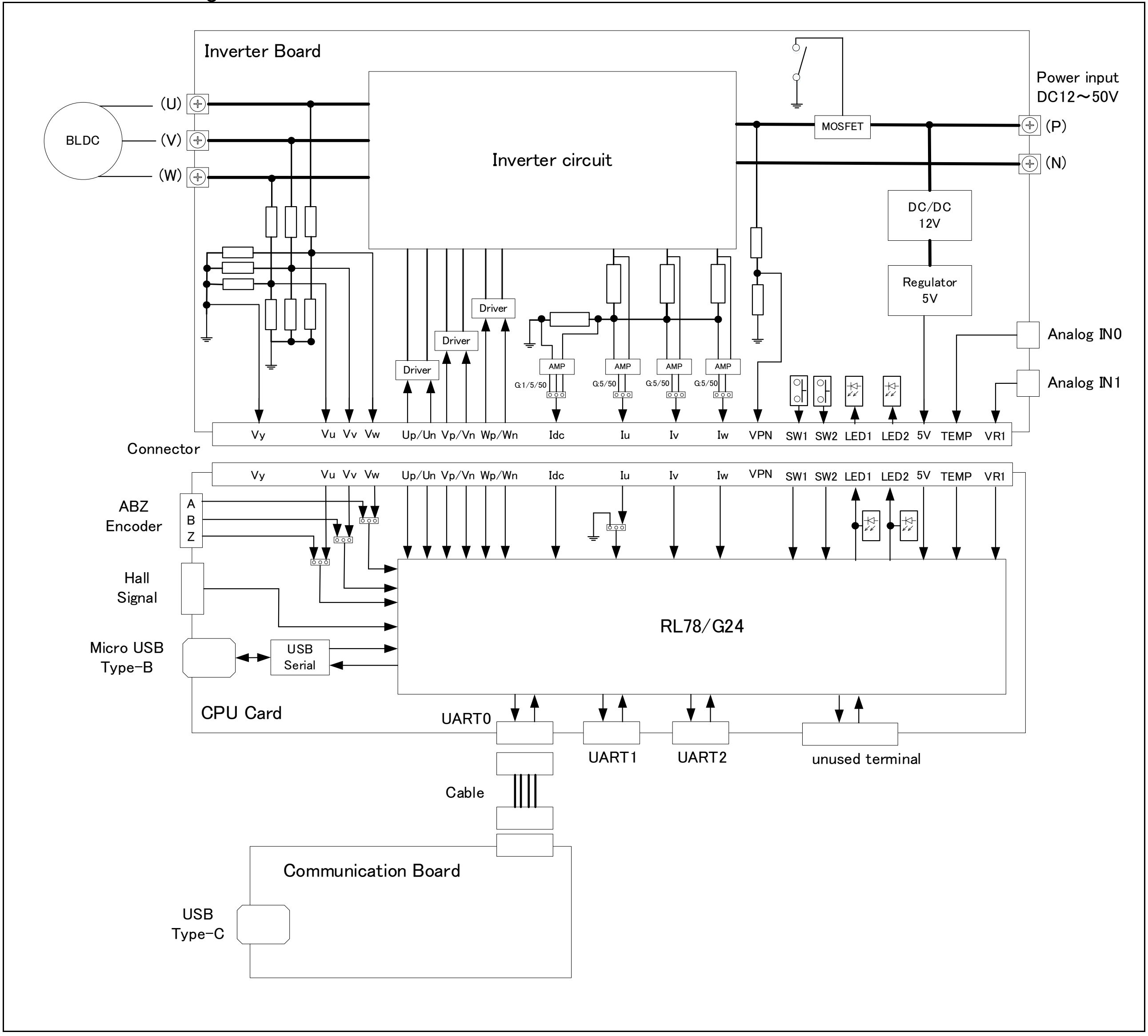


Figure 2-1 Hardware Configuration Diagram

2.3 Hardware specifications

2.3.1 User interface

Table 2-5 is a list of user interfaces of this system.

Table 2-5 User Interfaces

<table>
<tr><th>Board</th><th>Item</th><th>Interface component</th><th>Function</th></tr>
<tr><td rowspan="3">CPU Card</td><td>RESET</td><td>Tact switch (RESET)</td><td>System reset</td></tr>
<tr><td>LED1</td><td>Green LED</td><td rowspan="2"><table><tr><th></th><th>LED1</th><th>LED2</th></tr><tr><td>Stop</td><td>turn on</td><td>turn on</td></tr><tr><td>Run</td><td>turn on</td><td>turn off</td></tr><tr><td>Error</td><td>turn off</td><td>turn on</td></tr></table></td></tr>
<tr><td>LED2</td><td>Green LED</td></tr>
<tr><td>Inverter board</td><td>S1</td><td>Shut down switch of power supply current</td><td>Power supply current of Inverter board
・ON : Supply current
・OFF: Shut down supply current</td></tr>
</table>

The connector interfaces of this system are listed in Table 2-6.

Table 2-6 CPU Card Connector Interfaces

Board	Item	Number of ports	Function
CPU Card	CN1	20	Inverter board connectors
	CN2	20	Inverter board connectors
	CN3	5	Encoder signal input (not used in this system)
	CN4	5	Hall effect sensor signal input
	CN5	5	Emulator connector
	CN6	4	Serial communication (UART0) : Communication for RMW
	CN7	4	Serial communication (UART1) (not used in this system)
	CN8	4	Serial communication (UART2) (not used in this system)
	CN9	20	Universal area through holes (not used in this system)

This jumper setting of this system is listed in Table 2-7.

Table 2-7 Jumper settings

Board	Item	Terminal condition	Function
CPU Card	JP1	1-2 Open	Emulator connection enabled
	JP2	1-2 Short	Encoder not used (not used in this system)
	JP3	1-2 Short	Encoder not used (not used in this system)
	JP4	1-2 Short	Encoder not used (not used in this system)
	JP5	1-2 Short	[Not used in this system]
Inverter board	JP7	5-6 Short	DC link current gain = 50x
	JP8	2-3 Short	[Not used in this system]
	JP9	2-3 Short	[Not used in this system]
	JP10	2-3 Short	[Not used in this system]

Table 2-8 is a list of port interfaces of RL78/G24 microcontroller of this system.

Table 2-8 Port Interface

R7F101GLGFB port name	Function
P26 / ANI6	Inverter bus voltage detection
P147 / ANI18	DC link current detection
P120 / ANI19 / IVCMP0	U Phase voltage measurement (A/D)
P00 / ANI29 / IVCMP1	V Phase voltage measurement (A/D)
P01 / ANI30 / IVCMP2	W Phase voltage measurement (A/D)
P16 / ANI26 / IVREF0	Virtual neutral voltage input
P42	LED1 ON / OFF control
P43	LED2 ON / OFF control
P70 / TRDIOB0	PORT output / PWM output (Up)
P72 / TRDIOA1	PORT output / PWM output (Vp)
P74 / TRDIOB1	PORT output / PWM output (Wp)
P71 / TRDIOD0	PORT output / PWM output (Un)
P73 / TRDIOC1	PORT output / PWM output (Vn)
P75 / TRDIOD1	PORT output / PWM output (Wn)
P12 / TxD0	Serial communication (UART0): Communication for RMW
P11 / RxD0	Serial communication (UART0): Communication for RMW
P02 / TxD1	Serial communication (UART1) (not used in this system)
P03 / RxD1	Serial communication (UART1) (not used in this system)
P77 / TxD2	Serial communication (UART2) (not used in this system)
P76 / RxD2	Serial communication (UART2) (not used in this system)
P00 / TRGCLKA	Encoder A-phase input (not used in this system)
P01 / TRGCLKB	Encoder B-phase input (not used in this system)
P120 / TRGIDZ	Encoder Z-phase input (not used in this system)
P52 / INTP1	Hall effect sensor input (HU)
P53 / INTP2	Hall effect sensor input (HV)
P54 / INTP3	Hall effect sensor input (HW)
P40 / TOOL0	Data I/O for debugger
P50 / TOOLRxD	Data input for debugger
P51 / TOOLTxD	Data output for debugger
RESET	System reset input
VSS	Ground potential of the port
VDD	Positive power supply of the port
REGC	Regulator output stabilization capacitance connection for internal operation

Note: For pins other than those listed above, the R_MTR_InitUnusedPins function in r_mtr_rl78g24.c handles the pins as unused. Please make appropriate changes when changing the terminal arrangement.

2.3.2 Peripheral functions

Table 2-9 is a list of peripheral functions used in this system.

Table 2-9 List of Peripheral Functions

Peripheral Function	Usage
12-bit A/D converter (A/D)	- Inverter bus voltage measurement - U/V/W phase voltages measurement - DC-link current detection
Timer Array Unit (TAU)	- Free-running timer for rotational speed measurement (TAU1) - Delay timer for changing conducting pattern (TAU3) [Sensorless mode]
Timer RD2 (TRD2)	- PWM output using extended complementary PWM mode - 50[us] timer
Timer RG (TRG)	- 1[ms] interval timer
Timer RJ (TRJ)	- 50[us] timer for ICS communication [Initial Position Detection]
Timer RX(TRX)	- Current rising period measurement [Initial Position Detection]
Comparator (CMP3)	- Over current detection - Judgement for reaching threshold current [Initial Position Detection]
10-bit D/A converter (DA)	
External Interrupt (INTP1, INTP2, INTP3)	- Input signal from Hall effect sensor [Hall effect sensor mode]
PWM Option Unit A (PWMOPA)	- Forced shut-off of PWM output depending on CMP3 output
Watch dog timer (WDT)	- Program runaway detection

(1) 12-bit A/D converter (A/D)
The U phase voltage (Vu), V phase voltage (Vv), W phase voltage (Vw), inverter bus voltage (Vdc), DC link current (Idc) are measuled by using the '12-bit A/D converter'. [Sensorless mode]
Inverter bus voltage (Vdc), DC link current (Idc) are measured by using the '12-bit A/D converter'. [Hall effect sensor control mode]
A/D conversion mode is set to the advanced mode and the conversion operation mode to One-shot Conversion mode.

(2) Timer Array Unit (TAU)
a. Free-running timer for rotational speed measurement (TAU1)
This channel 1 of TAU is used as free-running counter for rotational speed calculation.
b. Delay timer for changing conducting pattern (TAU3)
The channel 3 of TAU is used as delay timer for changing conducting pattern with π/6 phase from the zero-crossing point.

(3) Timer RD (TRD)
AD conversion can be executed at any timing using two AD conversion trigger compare registers.
Three-phase PWM output of upper arm chopping with dead time (complementary) is performed using the Extended Complementary PWM Mode.

(4) Timer RG (TRG)
Timer RG (TRG) is used as 1[ms] interval timer for speed control.

(5) Timer RJ (TRJ)
Timer RJ (TRJ) is used as timer for ICS communication during initial position detection.

(6) Timer RX (TRX)
Timer RX (TRX) is used as timer for measuring period of reaching threshold current during initial position detection.

(7) Comparator (CMP3)
CMP3 is used for overcurrent or initial position detection. The output of D/A converter is selected for the reference value of CMP3, which is set as the threshold value to be compared with the corresponding current.

(8) 10-bit D/A Convertor (D/A)
The result of D/A converter is used as internal reference value for over current or initial position detection.

(9) External interrupt (INTP1. INTP2, INTP3)
Signals from Hall effect sensors are obtained for detection of rotor position. Both rising and falling edges are selected for detection. When the interruption occurs, input signals from Hall effect sensor are obtained (detection of rotor position), conducting pattern is changed, and calculation of rotational speed is performed.

(10) PWM Option Unit A (PWMOPA)
Force the PWM output to be cut off from the overcurrent signal detected in CMP3.
After detecting the cause of the cut-off release, the forced shut-off of the output is released from the software.
The output state at the time of interruption is Low-level output.

(11) Watch Dog Timer (WDT)
Watch Dog Timer is used as a runaway detection program.

2.4 Software structure

2.4.1 Software file structure

The folder and file configurations of the sample programs are given below.

Table 2-10 Folder and File Configuration of the Sample Program

Folder		File	Content
config		r_mtr_config.h	Common definition for software configuration
		r_mtr_motor_parameter.h	Configuration definition for motor parameters
		r_mtr_control_parameter.h	Configuration definition for control parameters
		r_mtr_inverter_parameter.h	Configuration definition for inverter parameters
		r_mtr_scaling_parameter.h	Configuration definition for scaling parameters
application	main	main.h main.c	Main function
	board	r_mtr_board.h r_mtr_board.c	Function definition for board UI
	ics	r_mtr_ics.h r_mtr_ics.c	Function definition for Analyzer(Note1) UI
		RL78_vector.h RL78_vector.c	Interrupt vector function definition for RMW CPU definition for RMW
		ICS2_RL78G24.h	Function declaration for RMW communication
		ICS2_RL78G24.lib	Library for RMW communication
driver		r_mtr_rl78g24.h, r_mtr_rl78g24.c	Function definition for MCU control
middle		r_dsp_cc_s.h R_DSP_RL78_CC_S.lib	DSP definition Arithmetic library for motor control
		r_mtr_common.h	Common definition
		r_mtr_parameter.h	Various parameter definition
		r_mtr_driver_access.h, r_mtr_driver_access.c	Function definition for driver access Driver access
		r_mtr_statemachine.h, r_mtr_statemachine.c	Function definition for state machine State machine
		r_mtr_120.h, r_mtr_120.c	Function definition for 120-degree conducting control 120-degree conducting control
		r_mtr_interrupt.c	Interrupt function definition
		r_mtr_ctrl_gain .h r_mtr_ctrl_gain.obj	Gain design function definition Gain design
		r_mtr_ipd.h r_mtr_ipd.c	Initial position detection function definition Initial position detection
		r_mtr_ol2cl_ctrl.h r_mtr_ol2cl_ctrl.c	Open loop to Closed loop switch control function definition Open loop to Closed loop switch control
		r_mtr_aa_ctrl.h r_mtr_aa_ctrl.obj	Advanced angle control function definition Advanced angle control
smc_gen			Smart Configurator generate file

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.

2.4.2 Module configuration

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

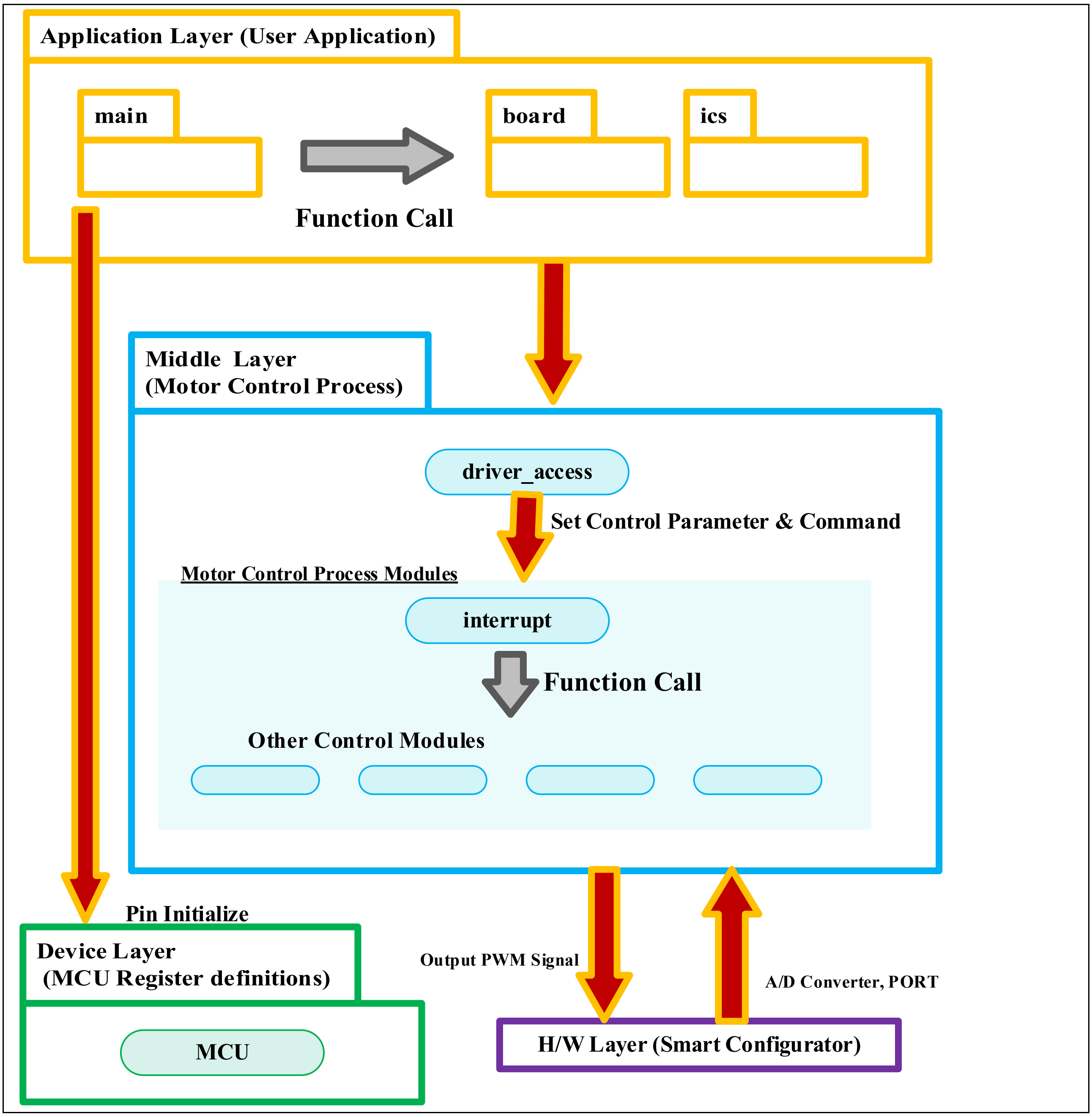


Figure 2-2 Module Configuration of the Sample Programs

2.5 Software specifications

Table 2-5 shows the basic specifications of target software of this application note. For details of 120-degree conducting control, refer to the application note ‘120-degree conducting control of permanent magnetic synchronous motor: algorithm’.

Table 2-11 Basic Specifications of Software[1/2]

Item	Content	
Control method	120-degree conducting method (high-side chopping)	
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. Initial Position Detection: The position of rotor is estimated using TRX and CMP3. - Angle detection within 180-degree is obtained by using the saliency characteristics of the rotor. - Angle detection within 360-degree is obtained by performing polarity detection using the magnetic saturation characteristics of the rotor.	
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: f_{CLK} 48[MHz] TRD clock: f_{PLL} 96[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] ~ 3975[rpm] CCW:0 [rpm] ~ 3975 [rpm] Note that the motor is driven at open-loop mode below 530 [rpm]	
Optimization setting	-Olite	
ROM/RAM Size	ROM	19.512KB [Sensorless current mode] 19.008KB [Sensorless voltage mode] 15.495KB [Hall effect sensor-based current mode] 14.865KB [Hall effect sensor-based voltage mode]
	RAM	1.130KB [Sensorless current mode] 1.078KB [Sensorless voltage mode] 0.896KB [Hall effect sensor-based current mode] 0.844KB [Hall effect sensor-based voltage mode]

Table 2-12 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 28 V (monitored per 1 [ms]) 2. Inverter bus voltage is less than 15 V (monitored per 1 [ms]) 3. Rotational speed exceeds 5590 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. TRX overflow during initial position detection 7. Detection of overcurrent - When an overcurrent detection signal (CMP3) is detected, the PWM output port is set to low-level (using PWMOPA).

[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.

2.6 User option bytes

The settings of the user option byte area of the RL78/G24 flash memory are shown below.

Table 2-13 User option byte settings

Setting	Address	Value	Description
783AEA	000C0H /040C0H	01111000B	- Uses watchdog timer interval interrupt: does not use interval interrupt - Period when watchdog timer window is open: 100 [%] - Watchdog timer counter operation control: Counter operation possible (After reset is canceled, count begins) - Watchdog timer overflow time: 100 [ms] - Watchdog timer counter operation control: In HALT/STOP mode, counter operation stops
	000C1H /040C1H	00111010B	- LVD0 off setting
	000C2H /040C2H	11101010B	- Flash operation mode setting: HS (high-speed main) mode - High-speed on-chip oscillator/block frequency fHOCO: 8 [MHz] fIH: 8 [MHz]

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 3-2 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 111 [V] : 0000H to 0FFFH	ANI6

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

As shown in Table 3-2 below, the U, V and W phase voltages are measured and used for determining zero-crossing of BEMF.

Table 3-2 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] to 111 [V] : 0000H to 0FFFH	ANI19, ANI29, ANI30

(3) DC link shunt resistor current

As shown in Table 3-3 below, the DC link shunt resistor current is measured and used for setting threshold current for CMP3.

Table 3-3 Conversion Ration of Offset current

Item	Conversion ratio (DC link shunt resistor current: A/D conversion value)	Channel
DC link shunt resistor current	-5 [A] to 5 [A] : 0000H to 0FFFH	ANI18

[Note] For more information about A/D conversion characteristics, see “RL78/G24 User’s Manual - Hardware.

3.1.3 Comparator

(1) Overcurrent detection and initial position detection (CMP3)

The output of the A/D is compared with the reference value of the internal D/A converter and used for the overcurrent detection and judgment of the current threshold value for initial position detection.

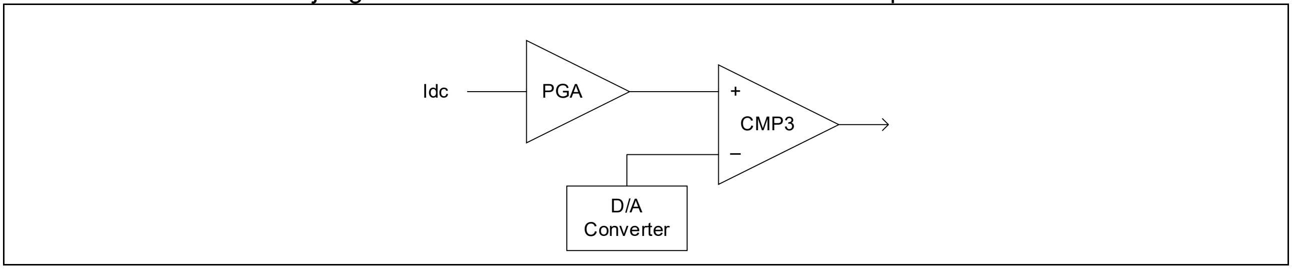


Figure 3-1 Overcurrent Detection Using CMP3

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 3-2.

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

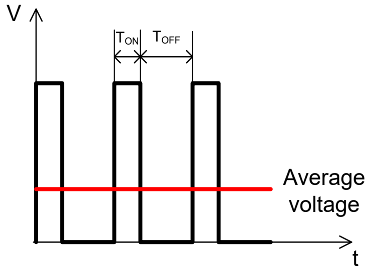


Figure 3-2 PWM Control

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 resisters for PWM duty in TRD.

In the target software of this application note, upper arm chopping is used to control the output voltage and speed. Figure 3-3 and Figure 3-4 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”.

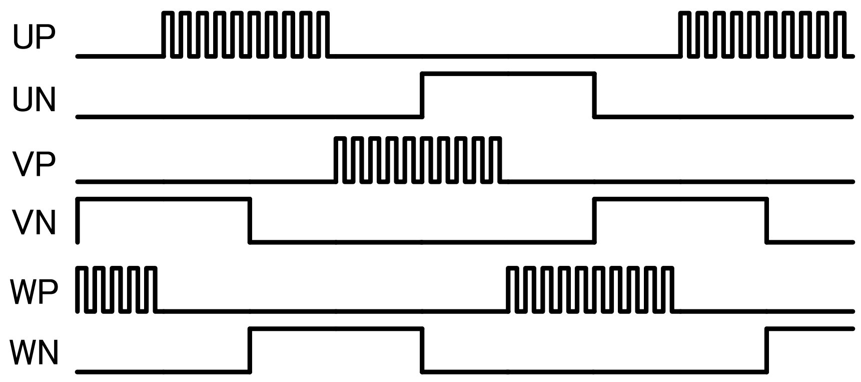


Figure 3-3 Upper Arm Chopping (Non-complementary PWM)

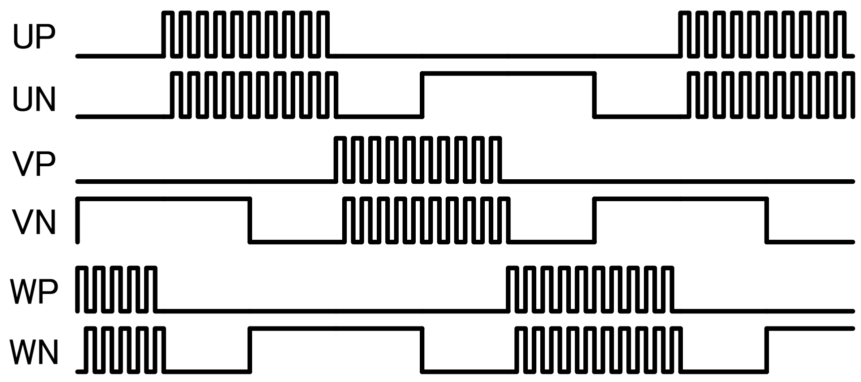
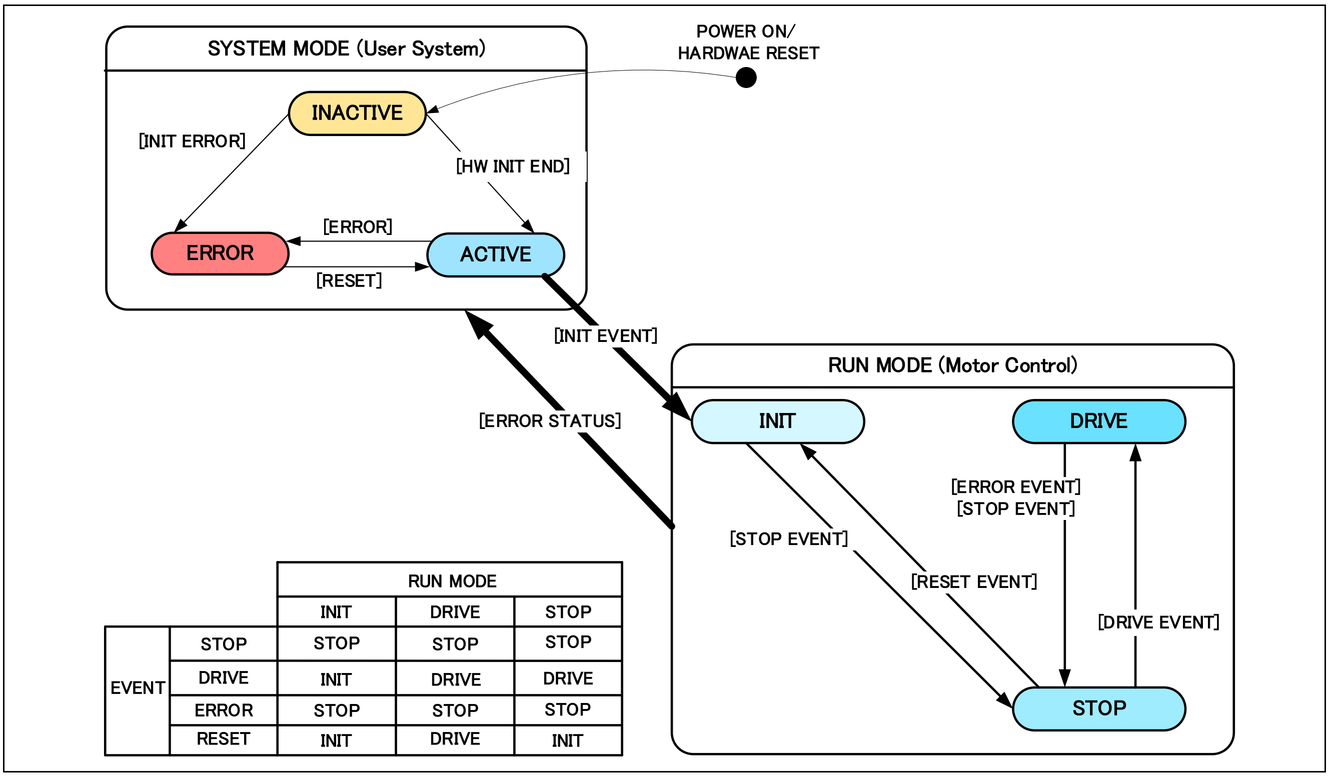


Figure 3-4 Upper Arm Chopping (Complementary PWM)

3.1.5 State transitions

Figure 3-5 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 3-5 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 3-5. 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 according to the drive status of the motor.

Table 3-5 DRIVE status list

Status name	Content
MTR_OFFSET_CALIB_EXE	Execution of current offset detection process
MTE_OFFSET_CALIB_END	Completion of current offset detection process
MTR_IPD_EXE	Execution of initial position detection process
MTR_IPD_END	Completion of initial position detection process
MTR_DRIVE_OL	Motor starting operation (open loop drive)
MTR_DRIVE_ASR_CL	Closed loop drive
MTR_DRIVE_BRAKE	Brake (Unimplemented)
MTR_DRIVE_END	Drive stop

3.1.6 Startup method

(1) Hall effect sensor mode

In the Hall effect sensor mode, after changing to "MTR_MODE_DRIVE", 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 3-6 and Figure 3-7 respectively.

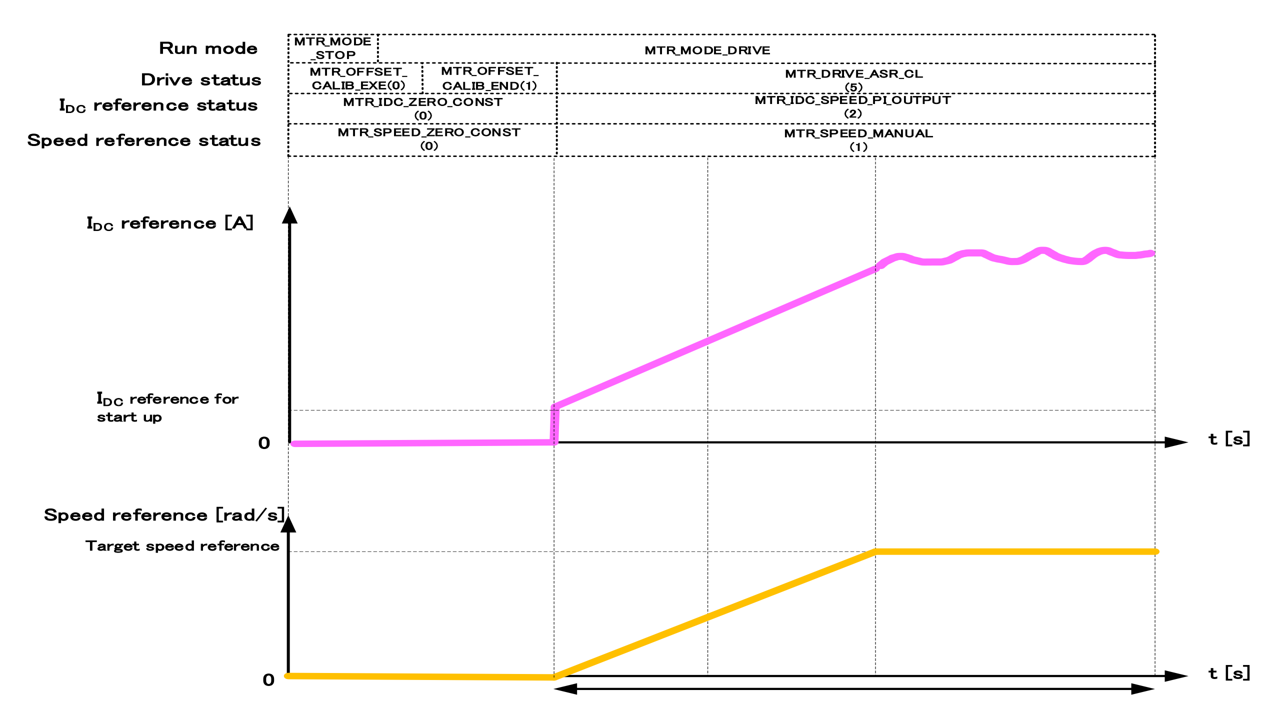


Figure 3-6 Start-up sequence for Hall effect sensor-based current control mode

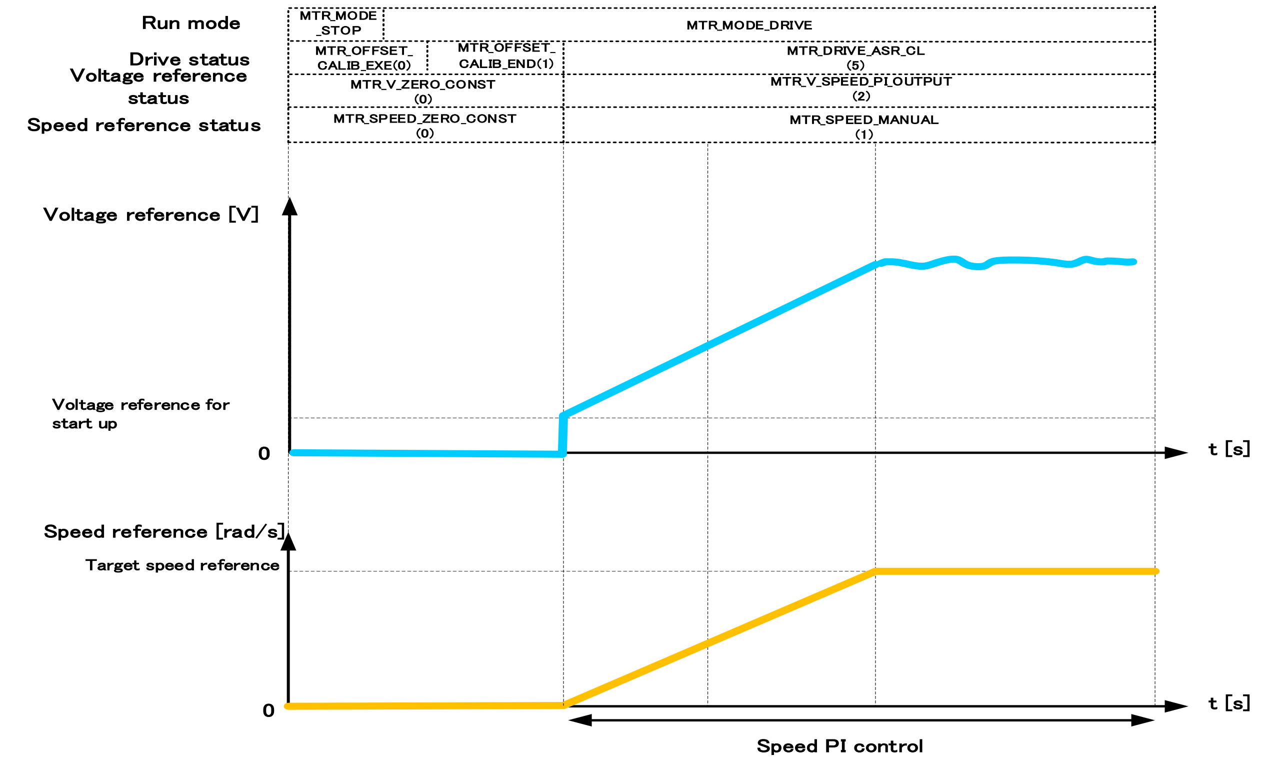


Figure 3-7 Start-up sequence for Hall effect sensor-based voltage control mode

(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 3-8 shows the start-up sequence for sensorless current control mode. In “MTR_MODE_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.

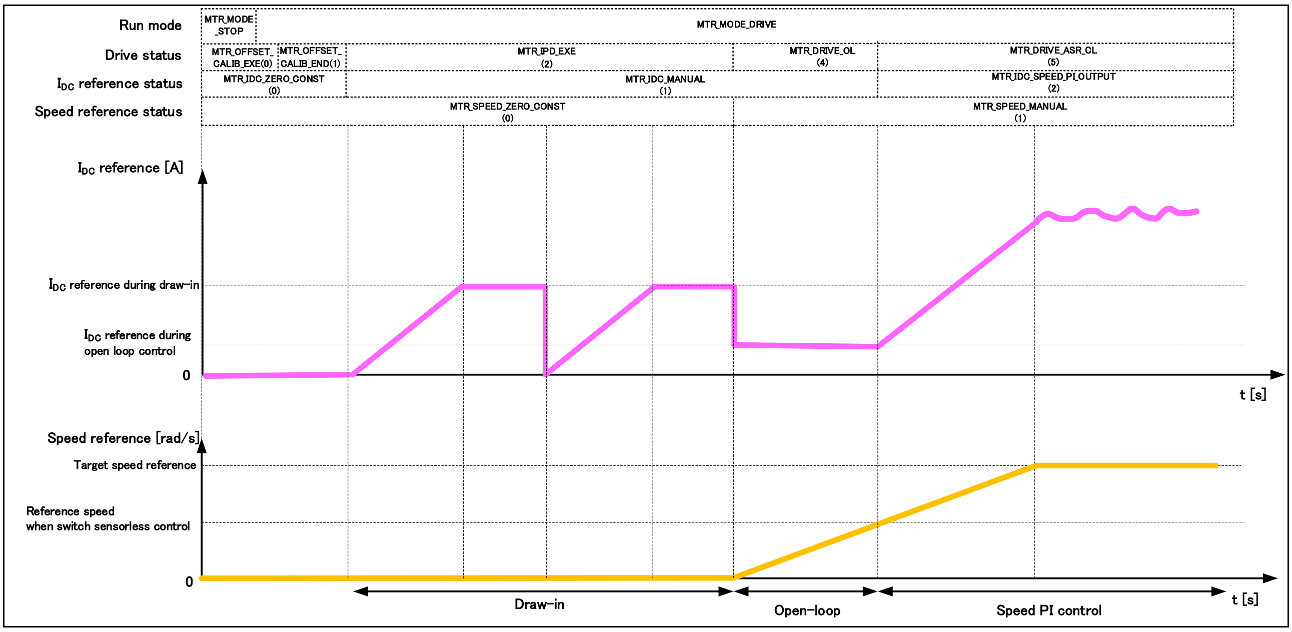


Figure 3-8 Start-up sequence for sensorless current control mode (Start-up with draw-in)

The start-up sequence for sensorless current control mode using the initial position detection function is shown in Figure 3-9. Instead of using draw-in to align the position of the rotor, initial position detection process is performed, after which open-loop process begins.

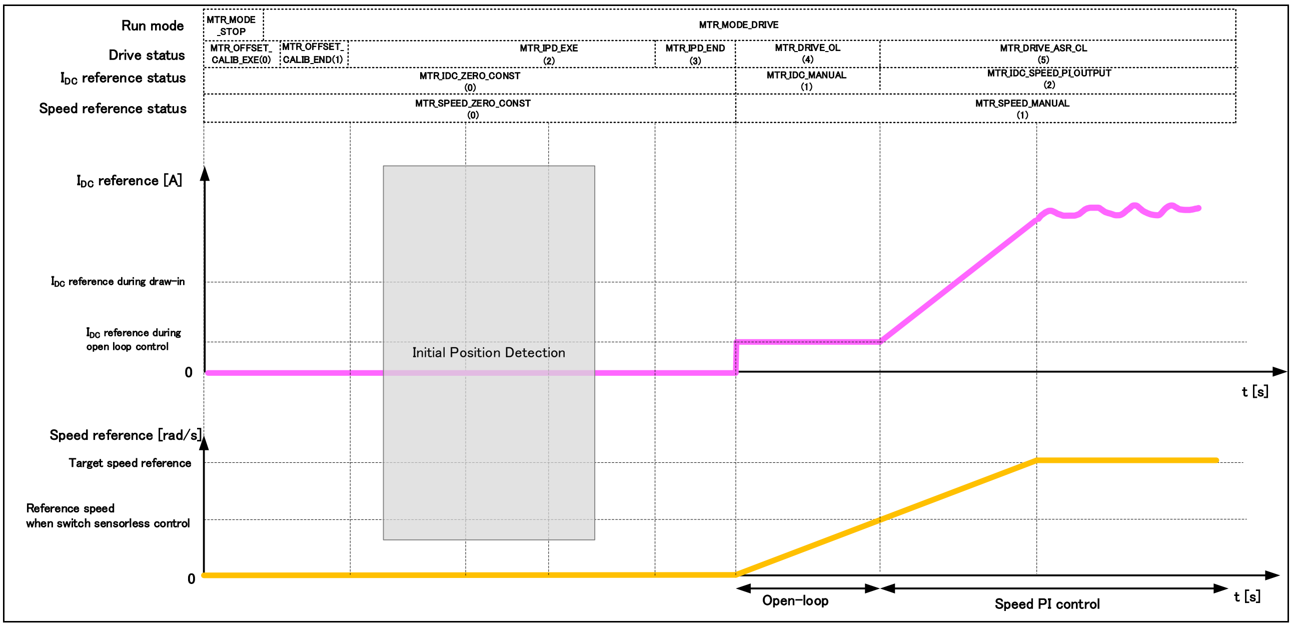


Figure 3-9 Start-up sequence for sensorless current control mode, start-up with Initial Position Detection)

Figure 3-10 shows the start-up sequence for sensorless voltage control mode. In “MTR_MODE_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.

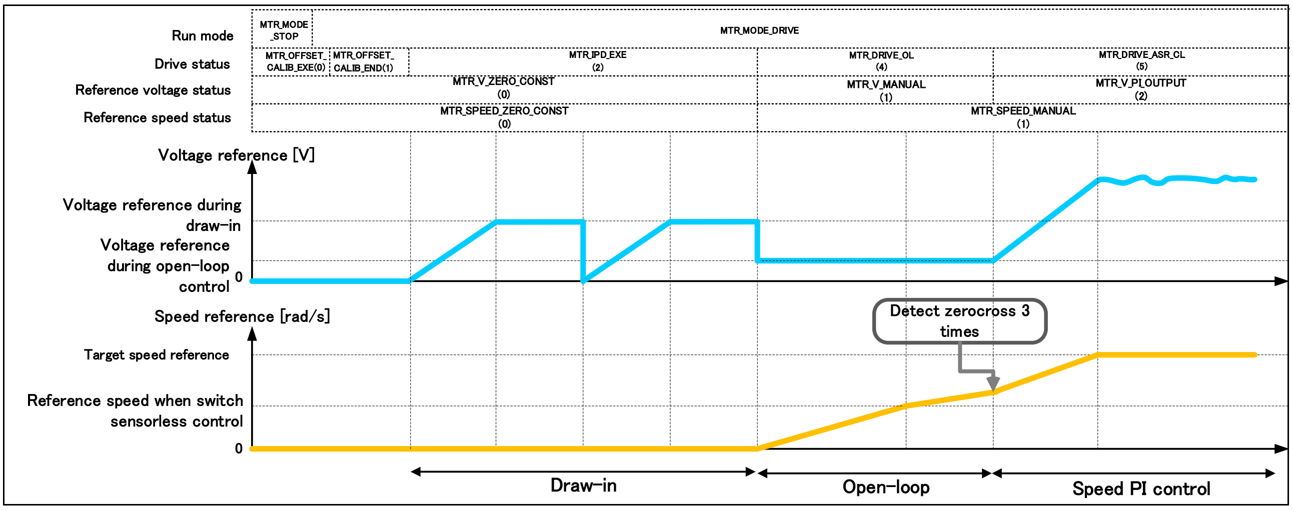


Figure 3-10 Start-up sequence for sensorless voltage control mode (Start-up with draw-in)

Figure 3-11 shows the start-up method for sensorless voltage control mode using initial position detection function.

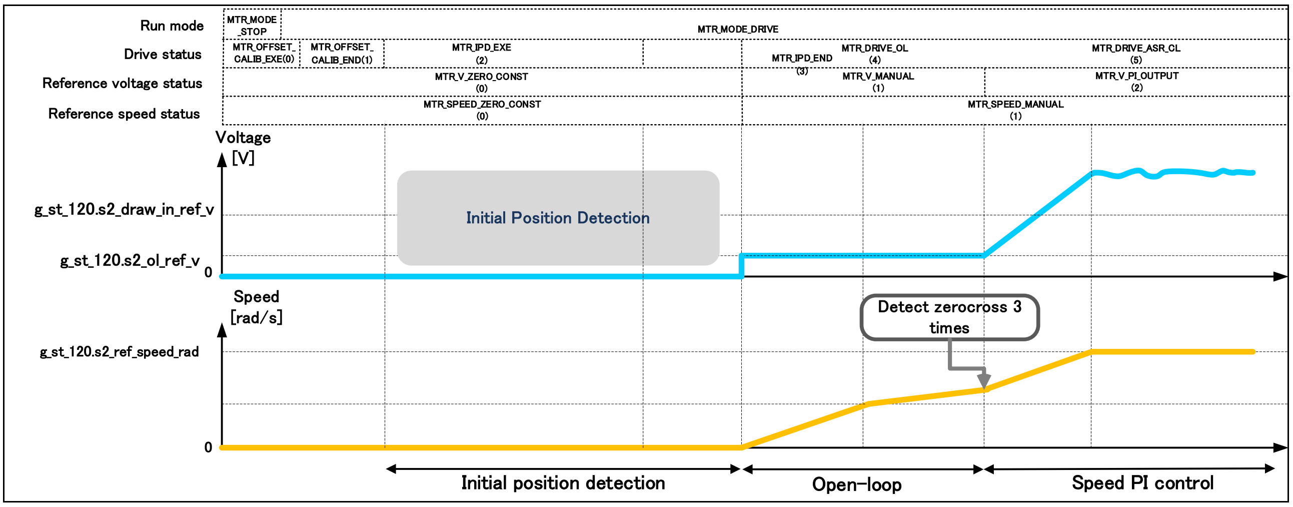
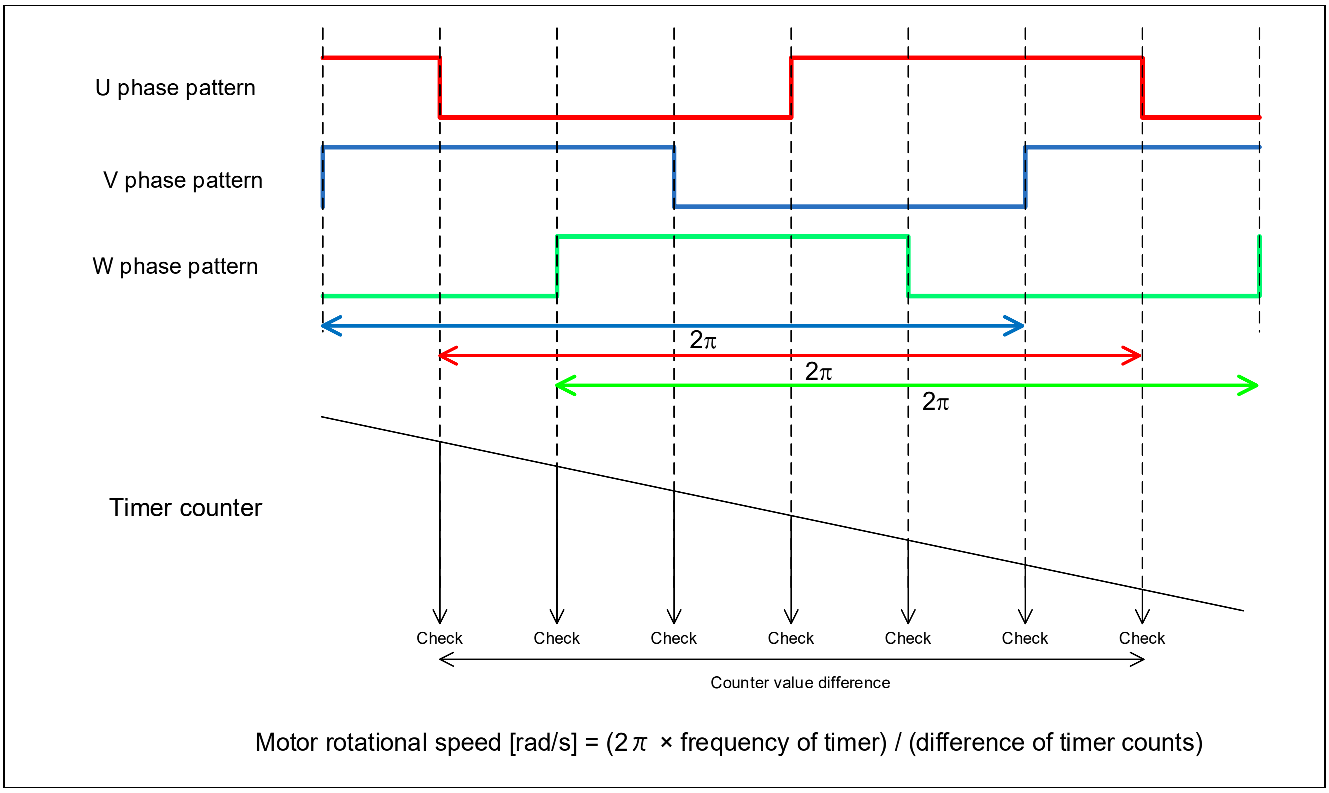


Figure 3-11 Start-up sequence for sensorless voltage control mode, start-up with Initial Position Detection)

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 3-12 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 3-13.

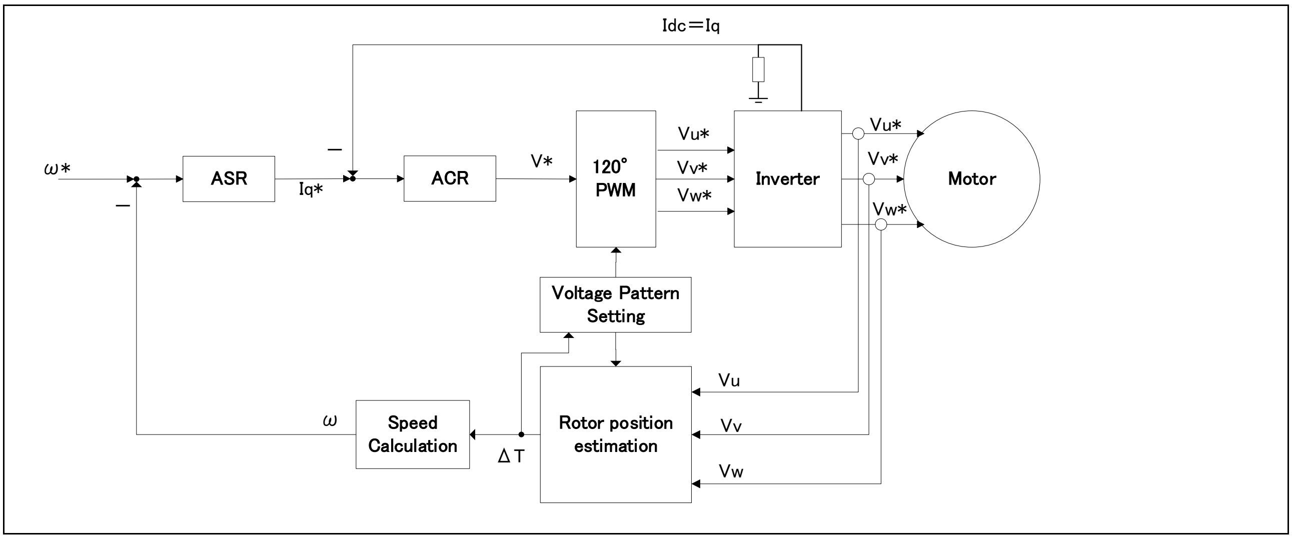


Figure 3-13 Control System Block Diagram (Current control mode)

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.

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

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

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

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 3-14.

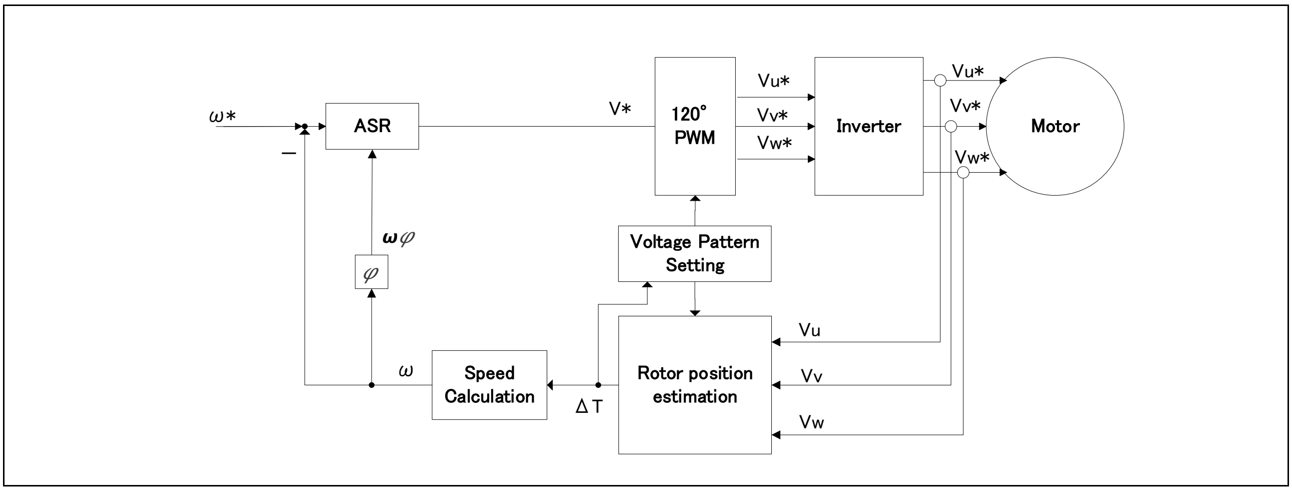


Figure 3-14 Control System Block Diagram (Voltage control mode)

The speed PI controller is shown as below. The speed PI controller outputs reference voltage value.

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

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.

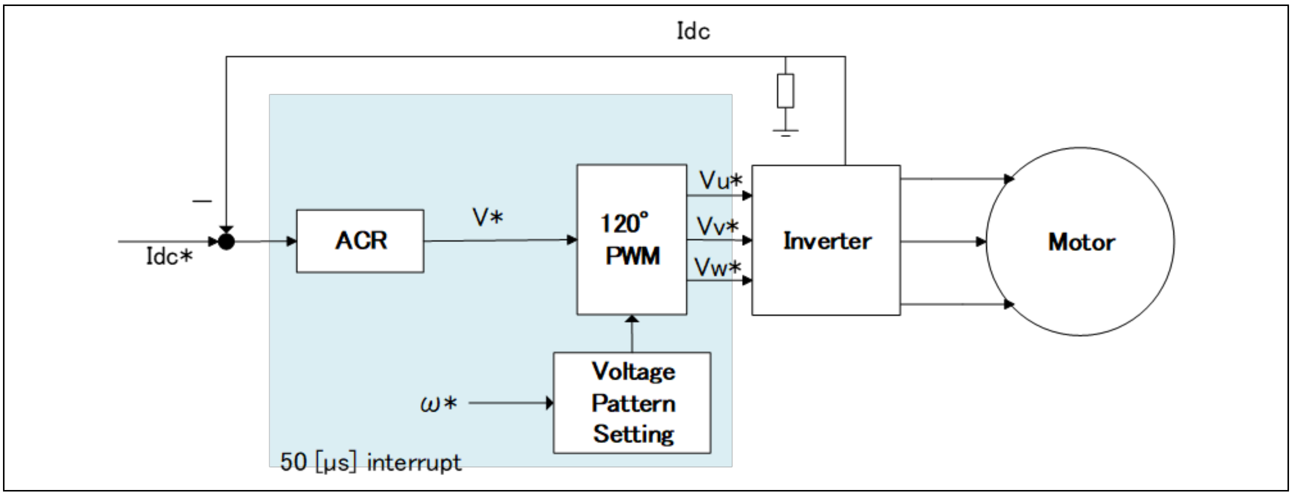


Figure 3-15 Interrupt processing inside control block (open-loop control)

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.

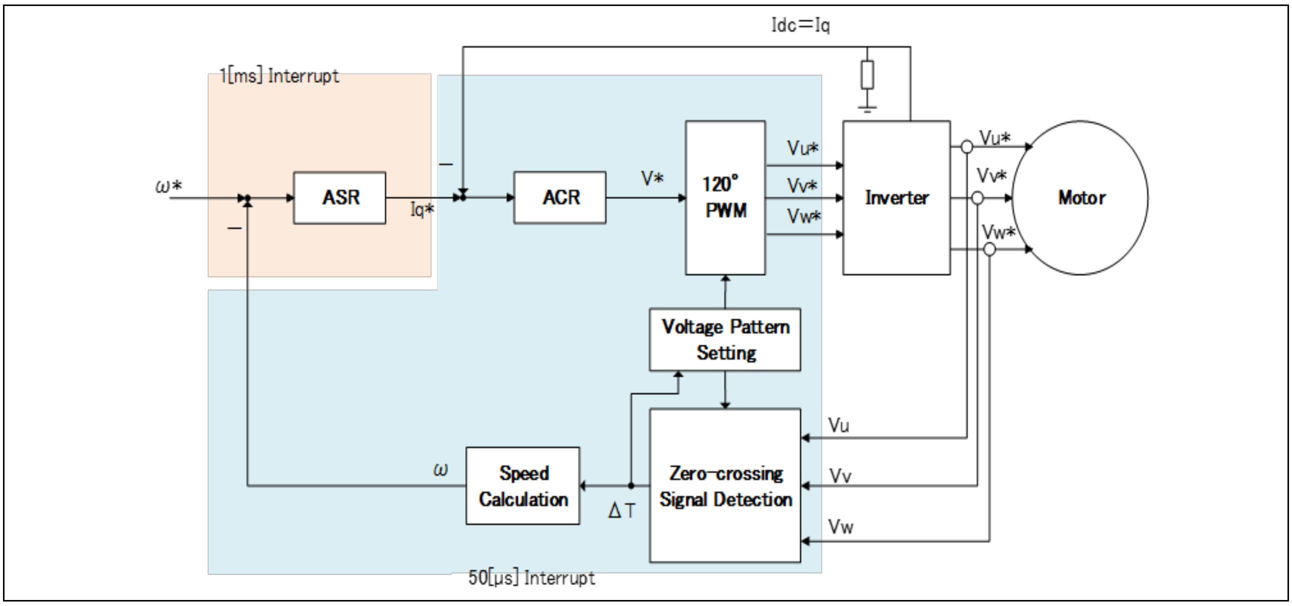


Figure 3-16 Interrupt processing inside control block (sensorless closed loop control)

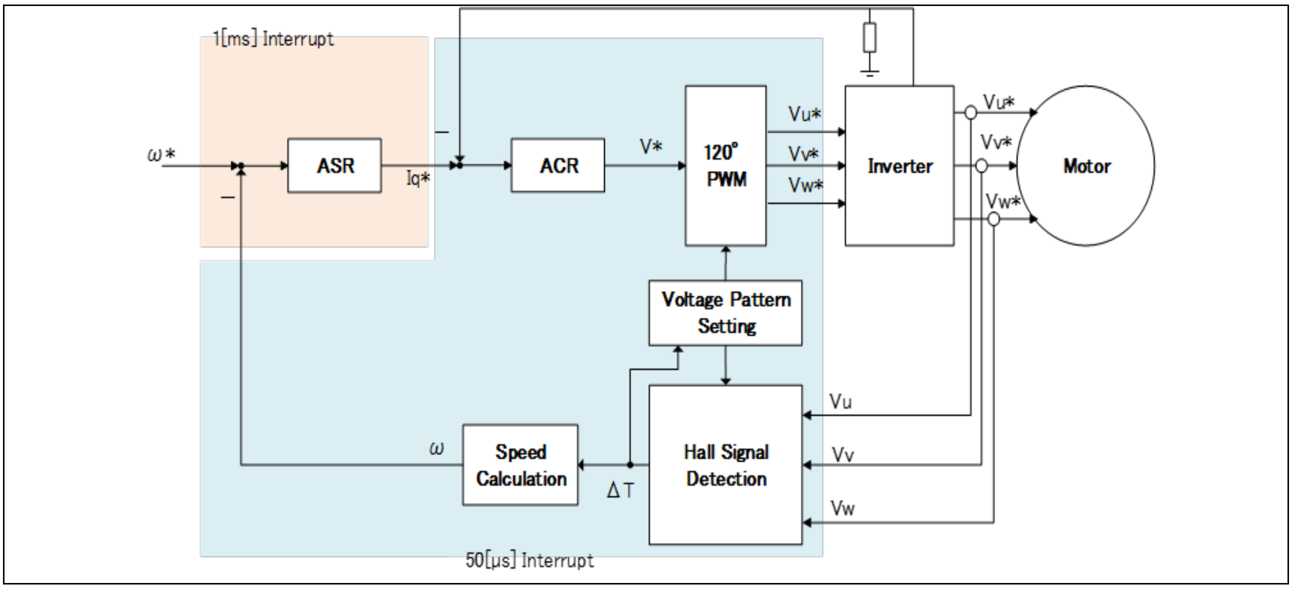


Figure 3-17 Interrupt processing inside control block (Hall effect sensor-based closed loop control)

Here is an outline for closed loop drive for 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.

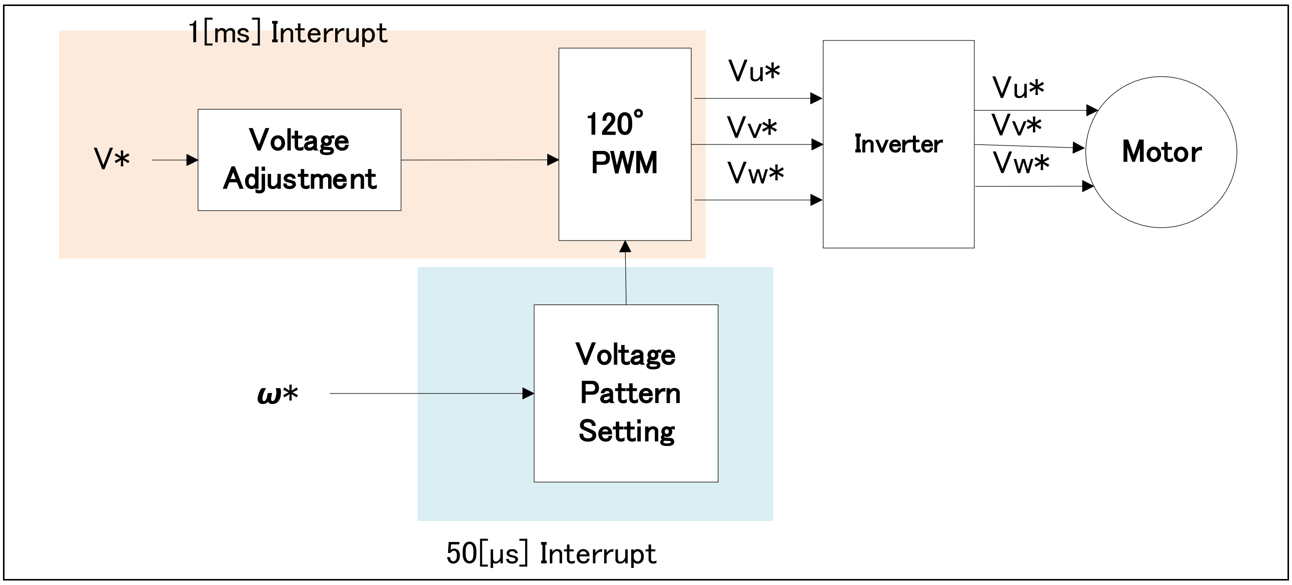


Figure 3-18 Interrupt processing inside control block (open-loop control)

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.

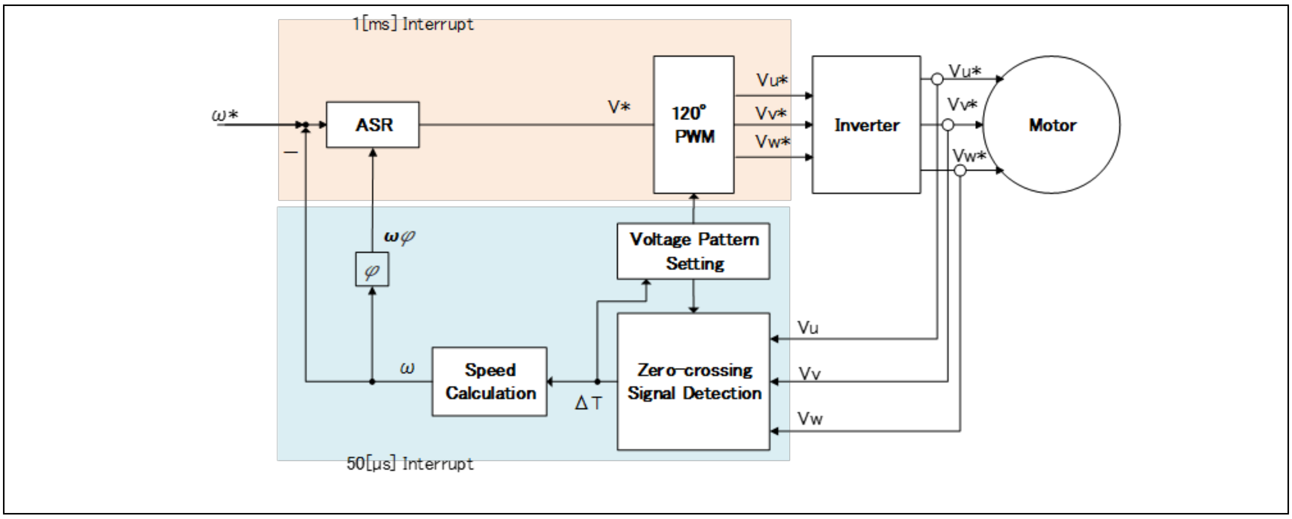


Figure 3-19 Interrupt processing inside control block (sensorless closed loop control)

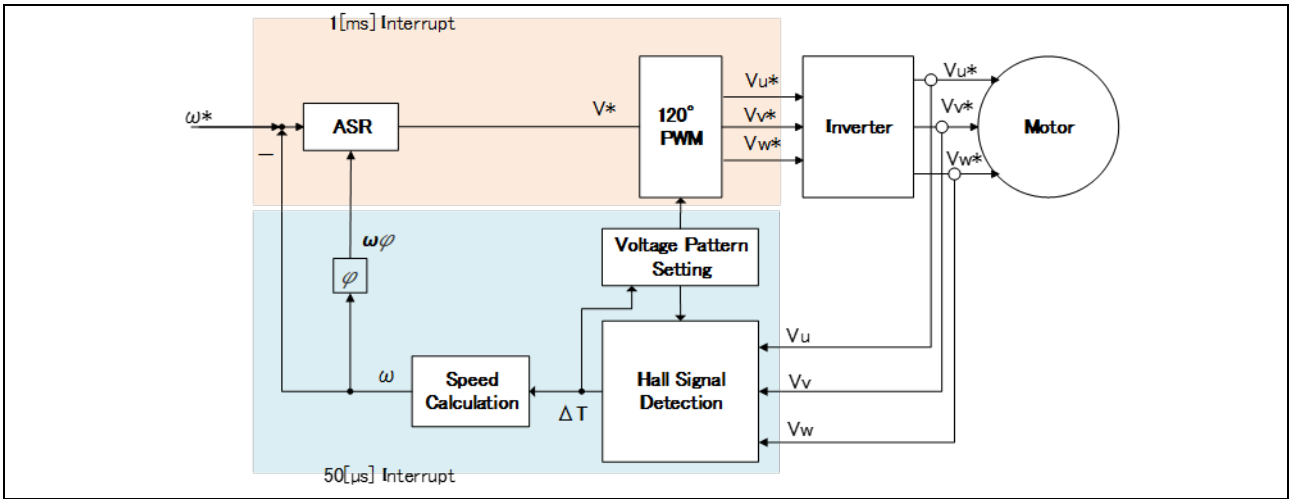


Figure 3-20 Interrupt processing inside control block (Hall effect sensor-based closed loop control)

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 RL78/G24 is used to measure DC link current, bus voltage and phase voltage.

- Timer RD2
Extended complementary PWM mode and A/D conversion trigger 0,1
- A/D converter
Advanced mode and hardware trigger mode

In this sample program, DC link current, bus voltage and phase voltage are measured over two carrier cycles. Figure 3-21 shows an example of extended complementary PWM waveforms.

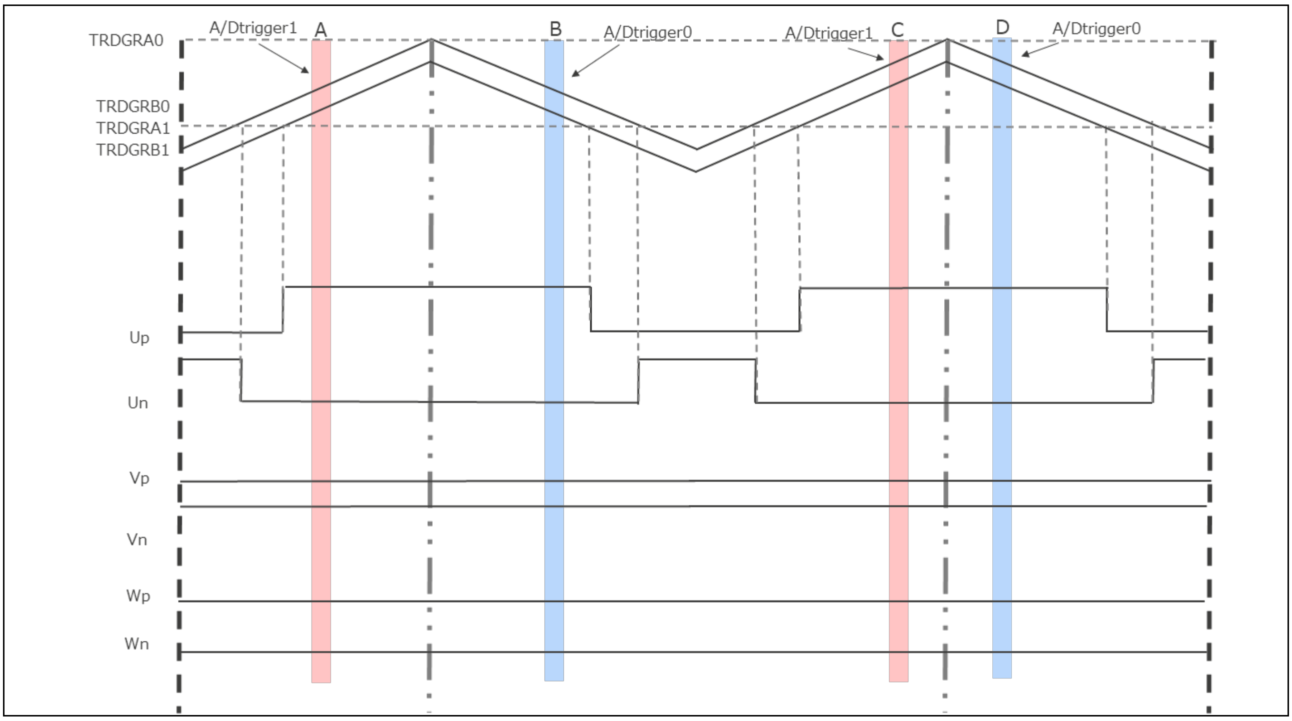


Figure 3-21 Extended complementary PWM waveforms
(Example: U phase: upper arm ON. V phase: lower arm ON, W phase: non-energized phase)

In the first carrier cycle, the DC link current and the non-energized phase voltage (W phase) are measured at point A in Figure 3-21. At point B, the non-energized phase voltage (W phase) and the phase voltage where the upper arm is turned on (U phase) are measured. In the second carrier cycle, the bus voltage and the non-energized phase voltage (W phase) are measured at point C. At point D, the non-energized phase voltage (W phase) and the phase voltage where the upper arm that is turned on (U phase) are measured. By shifting the position of the A/D conversion trigger closer to the peak of the carrier in the second cycle, non-energized phase voltage measurements are taken at four different points. The average of the measurement result is used as the non-energized phase voltage for control.

3.1.10.2 A/D conversion trigger timing adjustment

In this sample program, the induced voltage of the non-energized phase voltage is measured when the upper arm of the PWM is turned on. The induced voltage waveform typically appears as a damped oscillation after the upper arm is turned on. Since the measurement values vary depending on the A/D conversion trigger timing, the appropriate induced voltage is measured as follow. Two A/D conversion are performed within the period where the upper arm is turned on, and the average of the values over two carrier cycles is used. Therefore, it is necessary to adjust the A/D conversion trigger timing according to the duty value.

A threshold for duty value is set, and the A/D conversion trigger timing is switched based on whether the duty value is above or below the threshold. This prevents the A/D conversion process from overlapping with the timing of valley of carrier cycle where the setting of trigger is set. The threshold for the duty value is a tunable parameter.

When the duty value is below the threshold, A/D conversion trigger is set as shown in Figure 3-22. The A/D conversion trigger timing (①) from the ON/OFF edge of the upper arm, and the interval (②) between the A/D conversion trigger in first and second carrier are tunable parameters.

When the duty value exceeds the threshold, A/D conversion trigger 1 is fixed at the trigger position for the minimum duty value, as shown in Figure 3-23.

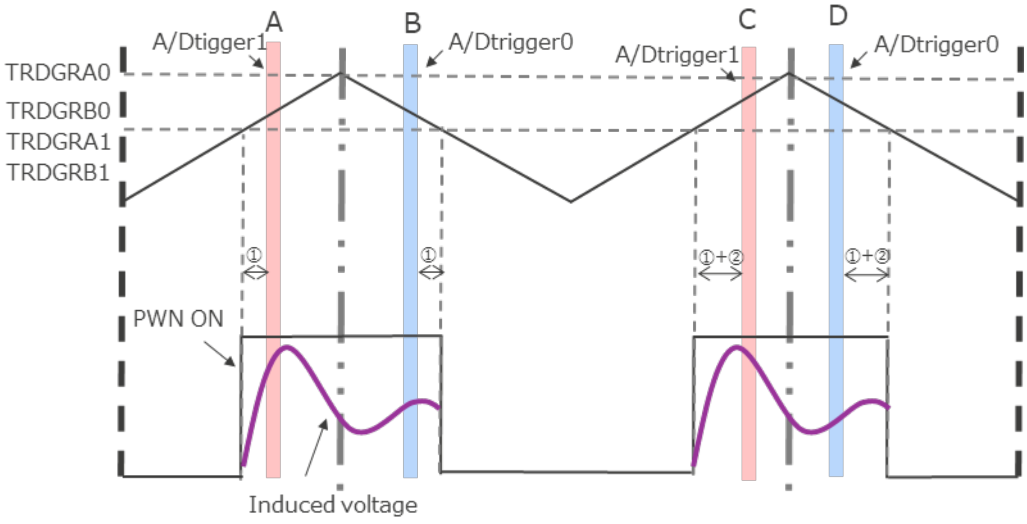


Figure 3-22 A/D conversion timing (Duty value is below threshold)

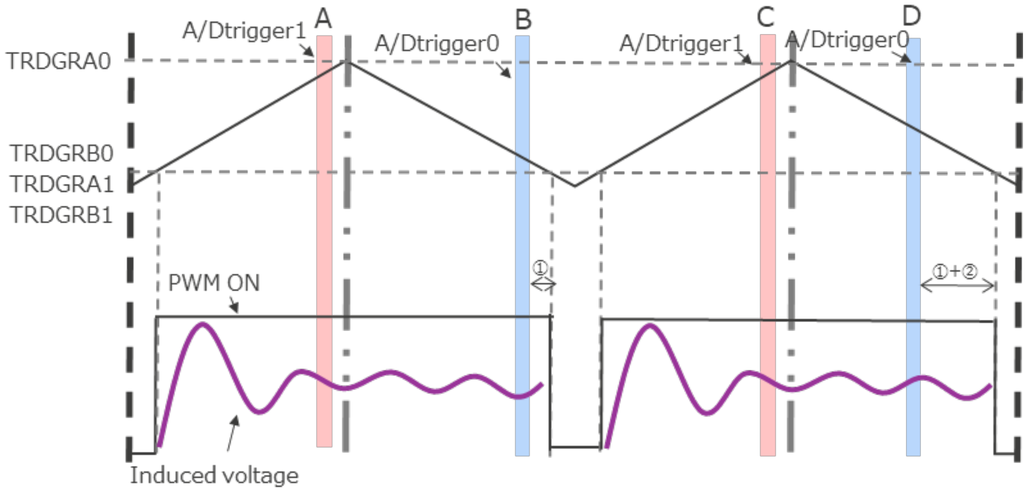


Figure 3-23 A/D conversion timing (Duty exceeds threshold value)

The following Table 3-6 shows the parameters and variables corresponding to the duty value threshold, the A/D conversion trigger timing (①) from the ON/OFF edge of the upper arm, and the interval (②) between the A/D conversion trigger in first and second carrier as shown in Figure 3-22 and Figure 3-23.

Table 3-6 Tunable parameters for A/D conversion timing

	Parameter name	Corresponding variable name
	CP_AD_TRG_DUTY_THRESHOLD_CNT (Duty value threshold)	g_st_120.st_less.s2_duty_th
①	CP_AD_TRG_ADJUST_UP_CNT (A/D conversion timing adjustment count value when the gradient of the non-energized phase voltage is rising)	g_st_120.st_less.s2_ad_trg_offset_cnt
	CP_AD_TRG_ADJUST_DOWN_CNT (A/D conversion timing adjustment count value when the gradient of the non-energized phase voltage is falling)	
②	CP_AD_TRG_DIFF_CNT (A/D conversion timing adjustment count value)	g_st_120.st_less.s2_ad_trg_offset_cnt

3.1.11 Initial position detection of a stationary rotor using saliency characteristic

A motor is said to have saliency if the magnetic resistance varies according to the position of the rotor. If the magnetic resistance changes sinusoidally, then the inductance will also change sinusoidally. As shown in Figure 3-24, inductance changes at twice the period of the rotor. In this case, when voltage is applied so that current flows from U→V, V→W, and W→U, the time it takes for the current flowing through the shunt to reach the threshold current value changes according to the position of the rotor. An example of this is shown in Figure 3-25. It therefore takes longer when voltage is applied in the V→W direction than when voltage is applied in the W→U direction.

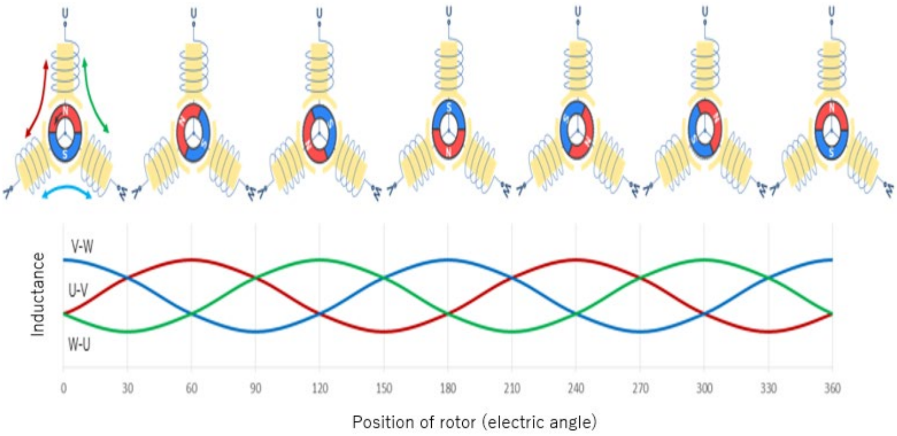


Figure 3-24 Changes in inductance according to the position of the rotor

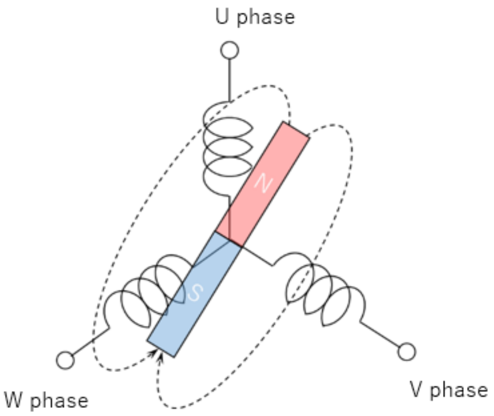


Figure 3-25 Relationship between rotor position and each phase

Here is a description of the rotor position detection method using this phenomenon. A diagram of the angle detection for salient motor used in this system is shown in Figure 3-26. It is distinguished by applying 3 patterns of voltage, measuring the time taken for the current that flows through the shunt resistor to reach the threshold current, and comparing these to detect the direction of the rotor facing in every 60 degrees within the 180 degrees of electrical angle.

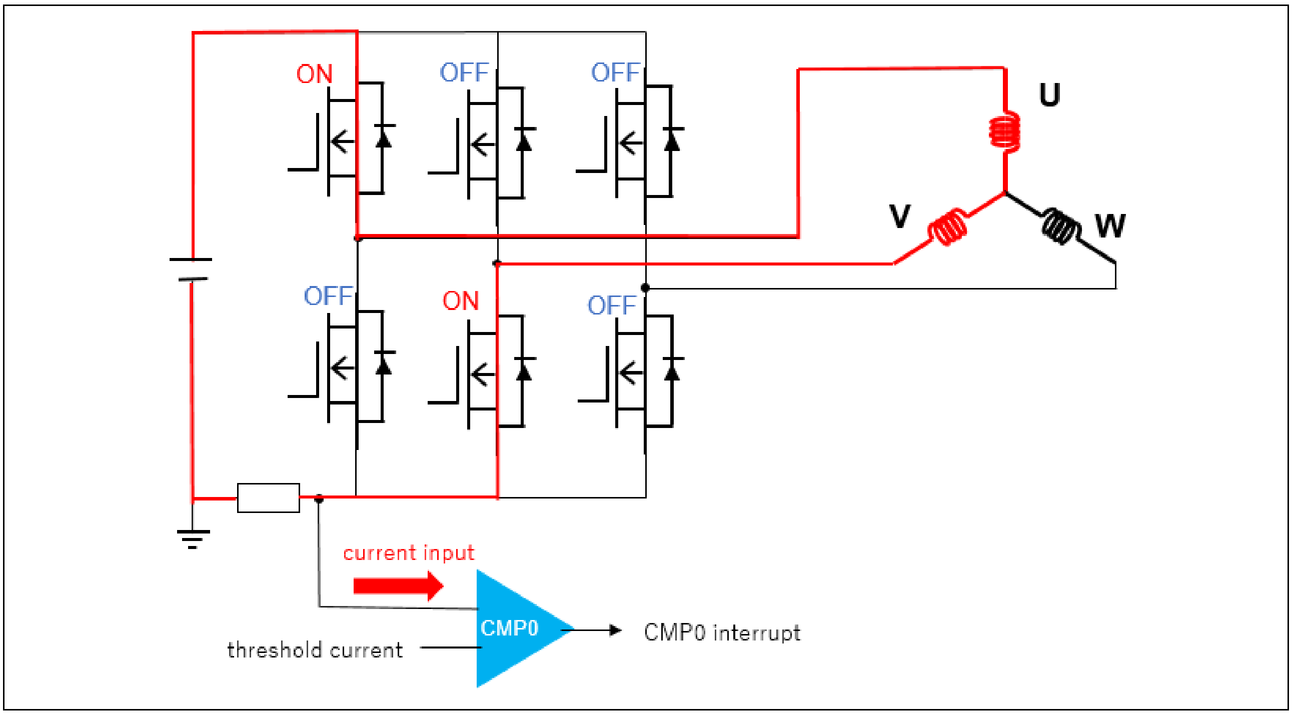


Figure 3-26 Angle detection diagram

The algorithm used in this system detects the time taken to reach the Internal reference current value using an RL78/G24 timer RX (TRX) and a comparator 3 (CMP3). It uses the TRD2 extended complementary PWM mode to apply pulse-shaped voltage to each phase. The count of the TRX starts at the rising edge synchronization of the TRD. The CMP3 generates an interrupt when it detects that the current flowing through the shunt resistor has reached the threshold current, and the current rise time required is measured.

Angle detection by saliency is performed every 60 degrees within the 180 degrees of electrical angle by comparing the cumulative time measured at each phase. Measurement ends when the cumulative time difference between the maximum phase and the minimum phase is greater than or equals to the threshold value. However, if the difference does not reach the threshold after the maximum number of measurements, angle detection by saliency is judged as failure.

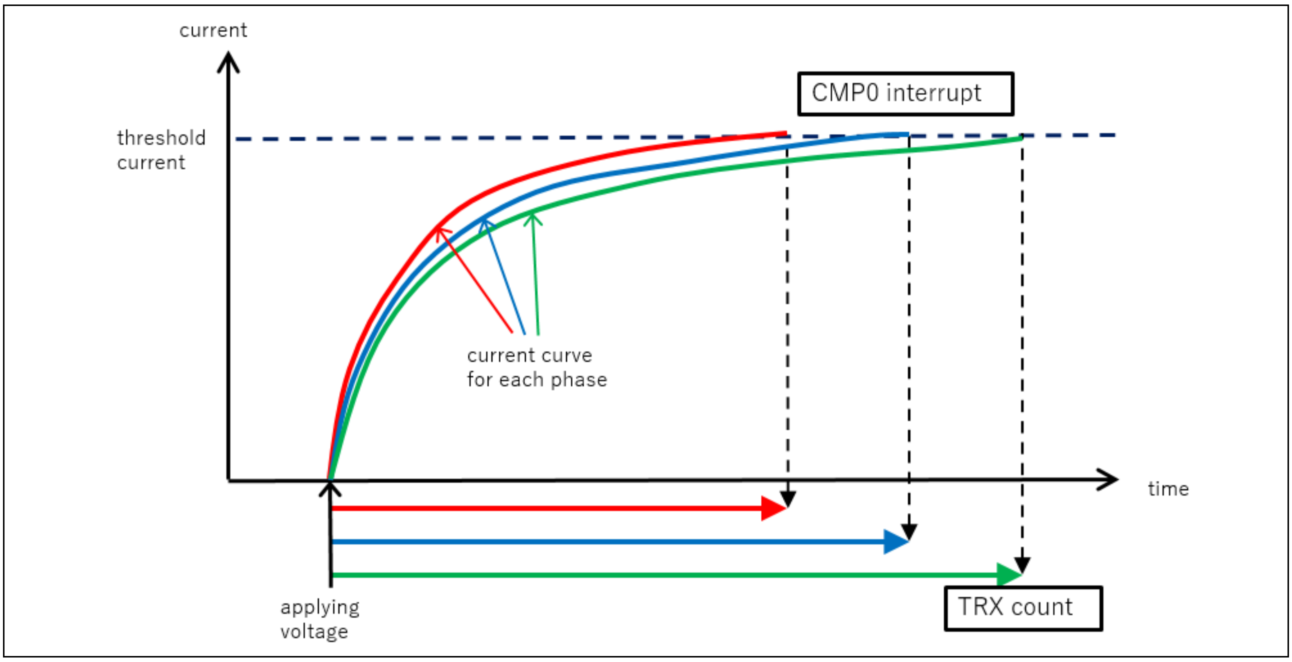


Figure 3-27 Current detection time differential among the 3-Phases

To confirm that the motor rotor has sufficient saliency to estimate its initial position, saliency judgement is performed. TRX count increases with the current rise during the measurement for 3-phases. The difference between maximum value and median value of the TRX count, and the difference between median value and minimum value of the TRX count, is compared and the phase with maximum difference is identified as the phase with maximum value or the phase with minimum value. Next, it applies voltage in the direction opposite to the identified phase and measures the time required for the current to rise. In this case, the mean value of the TRX count for the 2-phases is compared to the TRX count for the phase with the energization direction reversed. If the TRX count of the reversed phase has the same magnitude relation with the identified phase, saliency is judged to be sufficient, and if it does not, saliency is judged to be too low.

For example, as shown in Figure 3-28 if the rotor is oriented in the 120-degree direction, the phase identified will be the W-U phase because the difference between the maximum value and the median value is greater. By reversing the energization direction from the W-U phase, voltage is applied to the U-W phase, and the time take for current rise is measured. Compare the median value of the TRX counts of the U-V, V-W, and W-U phase, with the TRX count of U-W phase. If the U-W phase TRX count is greater, initial position detection using saliency is judged to be possible, but if it is lower, it is judged not to be possible.

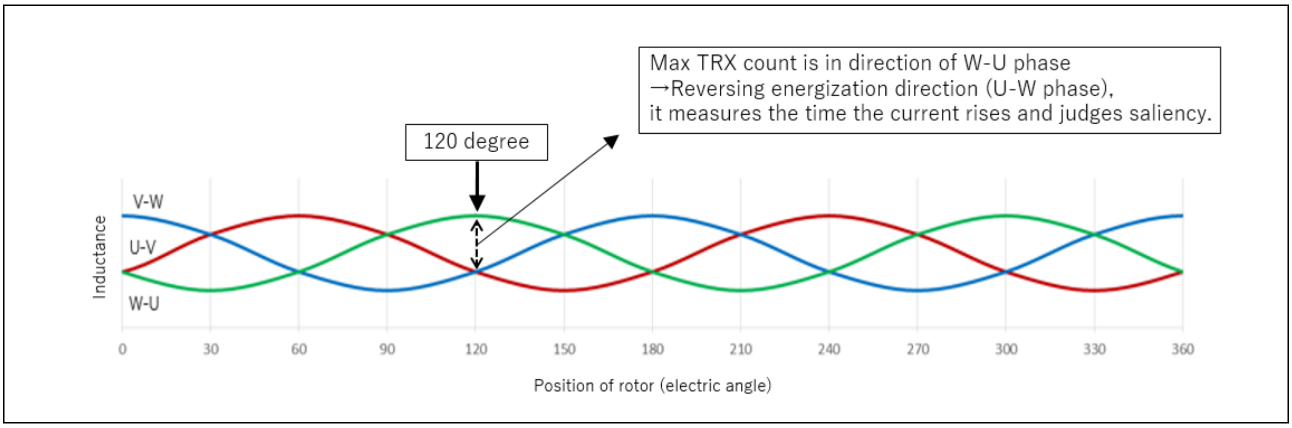


Figure 3-28 Saliency confirmation method

3.1.12 Initial position detection by saturation

Since the method described above uses the change in inductance due to saliency to estimate the position, it is not possible to determine the polarity (for example, there is no distinction between 60 degrees and 240 degrees). Also, it cannot be applied when a non-salient motor is used. Here, the magnetic saturation characteristics of the motor are used for polarity detection and angle detection with a non-salient rotor.

Due to the limited amount of magnetization that a magnetic material can have, if current is applied to a coil to generate an external magnetic field around the core of the coil, the core goes into a state of saturated magnetization when the external magnetic field exceeds a certain value. If the direction of the external magnetic field through the core is the same as the orientation of the magnetic field generated by the current flowing into the coil, the inductance becomes smaller because magnetization is more saturated than if the directions were opposite. These characteristics are used to judge the orientation of the magnetic pole.

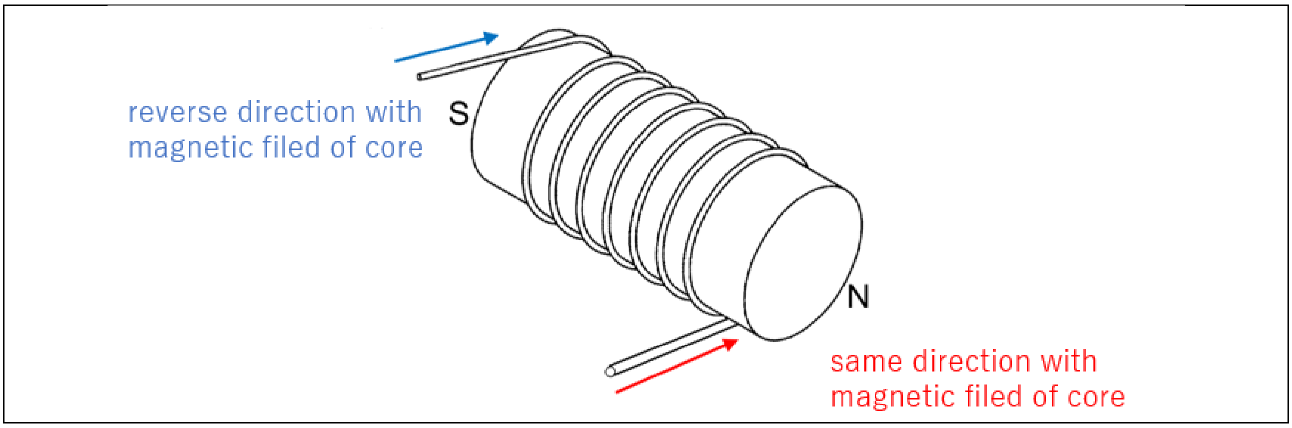


Figure 3-29 Example of magnetic pole wound with coil

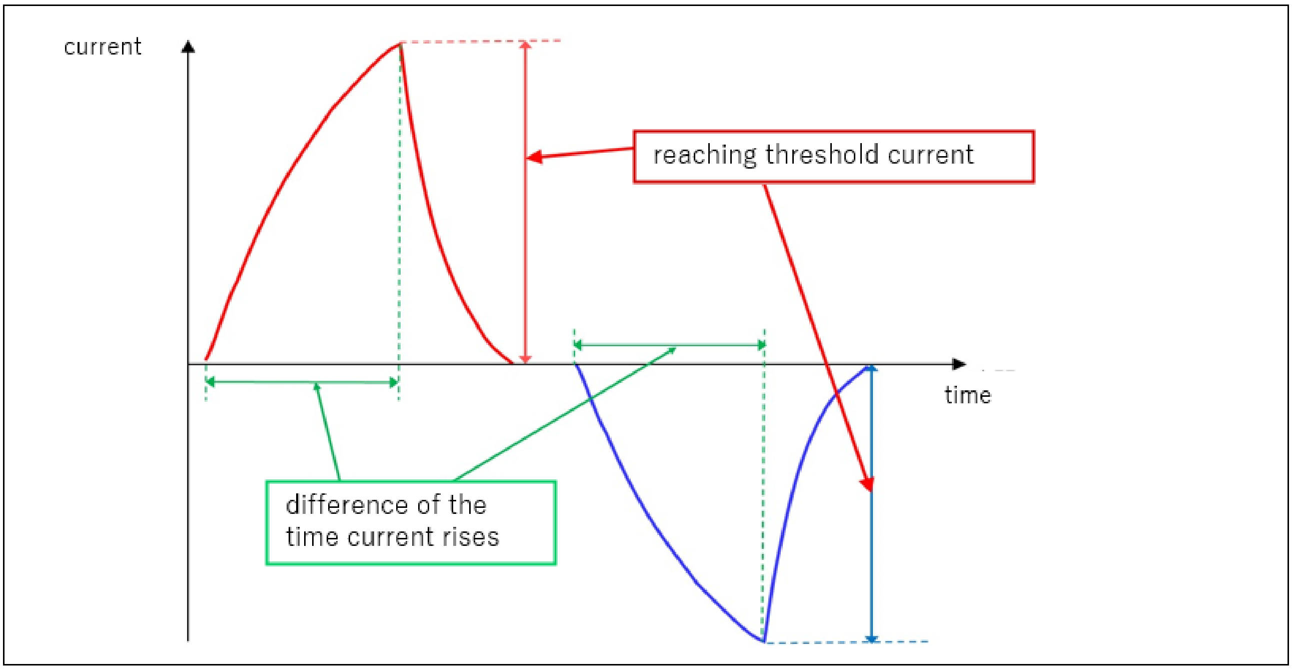


Figure 3-30 Current differential according to direction of applied current

Voltage is applied to the motor as shown in Figure 3-30 and the time required for the current flowing in the shunt resistor to rise is measured by TRX in the same way as it is measured when using saliency. For measurements using saturation characteristics, the TRX count is lowest when the direction of the voltage applied matches the direction of rotation, so this tendency is used to estimate the orientation of the rotor.

In the case of salient motor, to detect polarity by saturation, voltage is applied forward and backward based on the result of angle detection by saliency. The polarity of the rotor is determined by comparing the magnitude relation of current rise time. In the case of non-salient motor, the current rise time is measured by applying voltage in 6 directions, and the angle detection by saturation is performed by assuming the rotor is facing the phase with the minimum TRX count value.

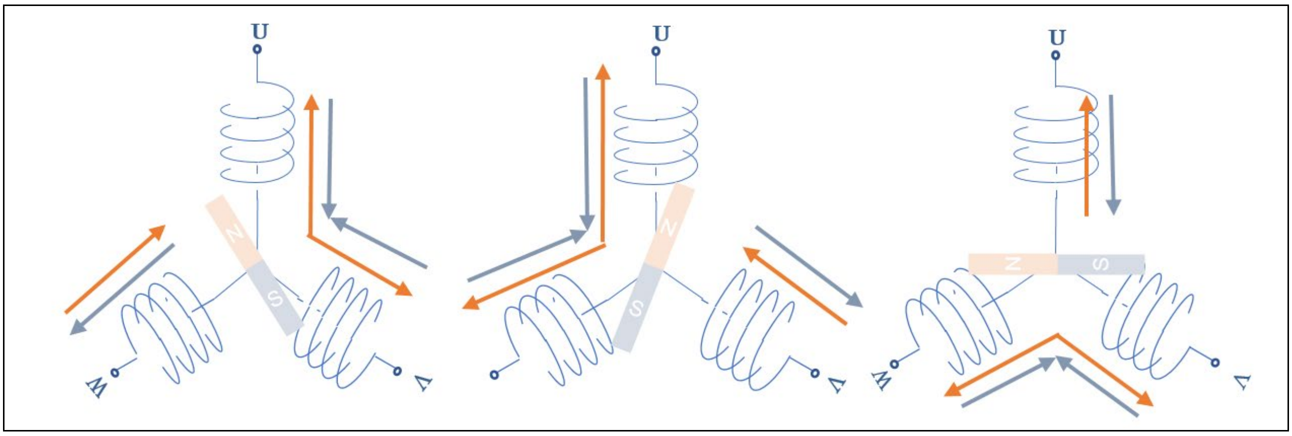


Figure 3-31 Example of applied voltage pattern due to initial position detection using magnetic saturation

Measurement stops when the TRX count difference for each directions exceed the threshold value. However, when the cumulative value of the TRX count difference does not reach the threshold value even after the maximum number of measurements, initial position detection is judged as success if the cumulative value is above the percentage of threshold set, whereas initial position detection is judged as failure if the cumulative value is less than percentage of threshold set.

The initial position detection process to be performed changes depending on the value of the preparation method for start-up (PS_METHOD). Table 3-7 shows the processes executed in each mode.

Table 3-7 Processing to be executed by the preparation method for start-up (PS_METHOD)

Preparation method for start-up (PS_METHOD)	Angle detection by saliency	Judgement of saliency	Polarity detection by saturation	Angle detection by saturation
PS_IPD_SAL	○	×	○	×
PS_IPD_NON_SAL	×	×	×	○
PS_IPD_UNKNOWN	○	○	○	○
PS_DRAW_IN	×	×	×	×

3.1.13 Advanced angle control

As the rotational speed increases, the phase of the current tends to lag the induced voltage of each phase voltage. When this phase lag occurs, the torque cannot be increased. Therefore, a method called advanced angle control is used, which the phase of the applied voltage is advanced such that the phase of the induced voltage and current coincide. In this sample program, by decreasing the angle delayed between BEMF zero-crossing detection and step commutation, angle can be advanced up to 30 degrees. In order to advance angle above 30 degrees, it is achieved by adjusting the value of the comparison voltage for BEMF zero-crossing detection (the midpoint voltage used for comparison with each phase's induced voltage).

3.1.14 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 3-8 for the settings of the system protection functions.

- Hardware overcurrent error

When the CMP3 interruption (overcurrent detection) is generated, voltage output is stopped.

- Software overcurrent error

3-phase current are monitored in the overvoltage monitoring cycle. When overvoltage (value exceeding the overvoltage 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.

- TRX overflow error

When TRX counter overflows at measuring period of current rising in initial position detection process, voltage output is stopped.

Table 3-8 System Protection Function Settings

Types of error	Threshold	
Hardware overcurrent error	Overcurrent limit [A]	4.5
Software overcurrent error	Overcurrent limit [A]	1.47
	Monitoring cycle [μs]	50
Overvoltage error	Overvoltage limit [V]	28
	Monitoring cycle [μs]	1000
Undervoltage error	Undervoltage limit [V]	12
	Monitoring cycle [μs]	1000
Rotational speed error	Speed limit [rpm]	5590
	Monitoring cycle [μs]	1000
Timeout error	Timeout value [ms]	226 [Sensorless mode] 113 [Hall-effect sensor mode]

3.1.15 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 3-9. These values are defined in r_mtr_scaling_parameter.h.

Table 3-9 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 3-10 List of Functions “main.c”

File	Function	Process
main.c	main Input: none Output: none	Initialization and main loop ・Initialization ⇒Initialization of hardware ⇒Initialization of RMW communication ⇒Initialization of system variables ⇒Initialization of control system ⇒Reset process ⇒Waiting for stability of bus voltage ・main loop ⇒System control depending on input from UI ⇒LED control ⇒Clear watch dog timer
	ics_ui Input: none Output: none	Process for RMW UI (GUI) ・Input values of command variables to RMW variables ・Change motor status depending on input event ・Initialization of system variables when reset event occurs
	software_init Input: none Output: none	Initialization of system variables ・Initialization of variables for main process ・Initialization of RMW variables

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

File	Function	Process
r_mtr_ics.c	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	Initialization of command variables
	mtr_limit [inline function] Input: (int16_t) s2_value :: target value (int16_t) s2_max :: maximum limit (int16_t) s2_min :: minimum limit Output: (int16_t) :: limited value	Limit between maximum and minimum values
	R_MTR_ICSIntLevel Input: (uint8_t) u1_level :: Priority of interrupt Output: none	Set ICS interrupt priority level

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

File	Function	Process
r_mtr_board.c	R_MTR_BoardLedContrl Input: (uint8_t) u1_motor_status :: motor status (uint8_t) u1_system_status :: system status Output: none	LED control depending on motor status and system mode

Table 3-13 List of Functions “ICS2_RL78G24.lib”

File	Function	Process
ICS1_RL78G24.lib	ics2_init Input: (uint16_t) addr :: DTC vector table start address (uint8_t) level :: Interrupt level (uint8_t) num :: Top address of DTC structure (uint8_t) brr :: Communication speed (uint8_t) mode :: Communication mode Output : None	Communication initialization
	ics2_watchpoint Input :: None Output :: None	Call transfer function Must be called at intervals off 300us or more.

Table 3-14 List of Functions “R_DSP_RL78_CC_S.lib”

File	Function	Process
R_DSP_RL78_CC_S.lib	R_motor_sincos_pu_FIX12 input: st_sincos12 *p_sincos12 :: Angle structure pointer Input range: -2π ≦theta ≦2π(FIX12) Output: none Output range: -1 ≦sin, cos ≦1 (FIX14)	Calculate sine and cosine values from the angle (FIX 12 theta) and store them in (FIX 14 sin) and (FIX 14 cos), respectively.

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

File	Function	Process
r_mtr_driver_access.c	R_MTR_InitControl Input: none Output: none	Initialization of control system ・Initialization of motor status ・Initialization of control variables
	R_MTR_IcsInput Input: (st_mtr_ctrl_input_t) *st_ics_input :: ICS structure Output: none	Input values of ICS variables to RMW buffer variables
	R_MTR_SetVariables [inline function] Input: none Output: none	Input values of ICS buffer variables to control variables
	R_MTR_InputBuffParamReset Input: none Output: none	Reset ICS buffer variables
	R_MTR_ExecEvent Input: (uint8_t) u1_event :: event Output: none	Change motor status and execute event process
	R_MTR_GetStatus Input: none Output: (uint8_t) :: motor status	Get motor status
	R_MTR_GetErrorStatus Input: none Output: (uint16_t) :: error status	Get error status
	R_MTR_GetDir Input: none Output: (uint8_t) :: direction of rotation	Get direction of rotation
	R_MTR_SetSpeed Input: (int16_t) s2_ref_speed_rpm :: command rotational speed Output: (uint8_t) u1_stop_req :: motor stop request	Set command rotational speed
	R_MTR_ChargeCapacitor Input: none Output: (uint16_t) :: timeout error	Waiting for stability of bus voltage
	R_MTR_UpdatePolling Input: none Output: none	Set control variables

Table 3-16 List of Functions “r_mtr_statemachine.c”

File	Function	Process
r_mtr_statemachine.c	mtr_statemachine_init Input: (st_mtr_statemachine_t) *p_state_machine :: structure for motor status Output: none	Initialization of motor status
	mtr_statemacine_reset Input: (st_mtr_statemachine_t) *p_state_machine :: structure for motor status Output: none	Reset motor status
	mtr_state_machine_event Input: (st_mtr_statemachine_t) *p_state_machine :: structure for motor status (void) *p_object :: structure for control variables (uint8_t) u1_event :: event Output: none	Execute event process
	mtr_statemachine_get_status Input: (st_mtr_statemachine_t) *p_state_machine :: structure for motor status Output: (uint8_t) p_state_machine->u1_status :: motor status	Get motor status
	mtr_act_none Input: (st_mtr_statemachine_t) *st_stm :: structure for motor status (void) *p_param :: structure for control variables Output: none	No process is performed
	mtr_act_init Input: (st_mtr_statemachine_t) *st_stm :: structure for motor status (void) *p_param :: structure for control variables Output: none	Initialization of control variables
	mtr_act_error Input: (st_mtr_statemachine_t) *st_stm :: structure for motor status (void) *p_param / structure for control variables Output: none	Stop motor control
	mtr_act_drive Input: (st_mtr_statemachine_t) *st_stm :: structure for motor status (void) *p_param :: structure for control variables Output: none	Reset control variables
	mtr_act_stop Input: (st_mtr_statemachine_t) *st_stm :: structure for motor status (void) *p_param :: structure for control variables Output: none	Stop motor control

Table 3-17 List of Functions “r_mtr_120.c”

File	Function	Process
r_mtr_120.c	R_MTR_120MotorDefaultInit Input: (st_mtr_120_control_t) *st_120 :: structure for control variables Output: none	Initialization of control variables
	mtr_120_motor_reset Input: (st_mtr_120_control_t) *st_120 :: structure for control variables Output: none	Reset control variables
	R_MTR_CtrlStop Input: none Output: none	Motor stop process ・Disable TRD2 output ・Disable hall signal interrupt ・Clear delay timer settings

Table 3-18 List of Functions “r_mtr_ctrl_gain_calc.obj”

File	Function	Process
r_mtr_ctrl_gain_calc.obj	R_MTR_CtrlGainVoltageMode Input: (st_mtr_ctrl_gain_t) *st_gain_v_buf :: structure for PI control (st_mtr_design_parameter_t)*st_ctrl_param:: structure for PI gain calculation Output: none	[Voltage control mode] PI gain calculation
	R_MTR_CtrlGainCurrentMode Input: (st_mtr_ctrl_gain_t) *st_gain_i_buf :: structure for PI control (st_mtr_design_parameter_t) *st_ctrl_param :: structure for PI gain calculation Output: none	[Current control mode] PI gain calculation

Table 3-19 List of Functions “r_mtr_interrupt.c” [1/3]

File	Function	Process
r_mtr_interrupt.c	R_MTR_CMP3Interrupt Input: none Output: none	・Hardware overcurrent detection process ・Get TRX count during initial position detection
	R_MTR_CarrierHallInterrupt Input: none Output: none	[Hall effect sensor control mode] Carrier interruption (50 μs) ・Obtain bus voltage and DC link current ・A/D conversion channel switching process ・Current PI control process, duty setting process [Current control mode] ・Check error ・Current offset calibration process ・ICS communication process
	R_MTR_1msHallInterrupt Input: none Output: none	[Hall effect sensor control mode]Interval timer interrupt (Called by INTTRG) Cycle : 1[ms] ・Change of motor drive status ・Set reference speed ・Set reference voltage, duty setting [Voltage control mode] ・Set reference current [Current control mode] ・Check error
	R_MTR_HallUInterrupt Input: none Output : none	[Hall effect sensor mode] Process for hall effect sensor interruption ・Set voltage pattern depending on signal from Hall effect sensor ・Speed calculation
	R_MTR_HallVInterrupt Input: none Output : none	[Hall effect sensor mode] Process for hall effect sensor interruption ・Set voltage pattern depending on signal from Hall effect sensor ・Speed calculation
	R_MTR_HallWInterrupt Input: none Output : none	[Hall effect sensor mode] Process for hall effect sensor interruption ・Set voltage pattern depending on signal from Hall effect sensor ・Speed calculation
	mtr_hall_signal_set [inline function] Input: none Output: none	[Hall effect sensor control mode] Set voltage pattern based on signal from Hall effect sensor
	mtr_hall_signal_process [inline function] Input: none Output: none	[Hall effect sensor control mode] Process for hall effect sensor interruption ・Set voltage pattern depending in signal from Hall effect sensor ・Rotational speed calculation
	R_MTR_CarrierLessInterrupt Input: none：none Output name: none	[Sensorless mode] Carrier interruption (50 μs) ・Obtain bus and UVW voltages and DC link current ・A/D conversion channel switching process ・Current PI control process, duty setting process [Current control mode] ・Current offset calibration process ・Initial position Detection ・Draw-in process ・Detection of zerocross ・Transition from open-loop drive to sensorless control ・Rotational Speed calculation ・Open-loop drive process ・Set delay timer ・CMP setting process ・ICS communication process

Table 3-20 List of Functions “r_mtr_interrupt.c” [2/3]

File	Function	Process
r_mtr_interrupt.c	mtr_calib_current_offset_dc [inline function] Input: (st_mtr_120_control_t) *st_120 :: structure for control variables Output: none	DC link current offset calibration
	mtr_speed_calc [inline function] Input: none Output: none	Calculation of rotational speed
	mtr_set_chopping_pattern [inline function] Input: (uint16_t) u2_pattern :: voltage pattern Output: none	Set chopping pattern
	mtr_set_speed_ref [inline function] Input: (int8_t) s1_ref_dir :: reference direction of rotation (int8_t) s2_dir :: direction of rotation Output: none	Set reference speed
	mtr_set_voltage_ref [inline function] Input: none Output: none	[Voltage control mode] Set reference voltage
	mtr_pi_ctrl [inline function] Input: (st_mtr_pi_control_t) *pi_ctrl :: PI control structure Output: (int16_t) :: variation of output voltage	[Voltage control mode] PI control process (velocity form)
	mtr_set_current_ref[inline function] Input: (st_mtr_120_control_t) *st_120 :: Structure for control variables Output: (int16_t) :: DC link current reference value	[Current control mode] Set reference DC link current ・PI control process (velocity form)
	mtr_current_pi_ctrl[inline function] Input:(st_mtr_acr_t) *st_acr :: current PI control variables structure pointer (int16_t) *s2_idc_ad :: DC link current pointer (int16_t) *s2_ref_idc_ctrl :: reference DC link current pointer (int16_t) *s2_ref_v_ctrl :: reference voltage pointer (uint8_t) *u1_state_drive :: state of drive pointer (int16_t) *s2_limit_v :: Voltage limit pointer. Output: none	[Current control mode] Set reference voltage ・PI control process((current form)
	mtr_pi_run [inine function] 入力：(st_mtr_pi_t) *st_pi :: PI control structure pointer (int16_t) s2_err :: error between reference and estimated speed of rotation (const uint8_t) u1_kp_q :: proportional gain shift value (const uint8_t) u1_kidt_q :: integral gain shift value Output：(int16_t) :: PI output	[Current control mode] PI control process
	mtr_duty_calc [inline function] Input: (int16_t) s2_ref_v :: reference voltage (int16_t) s2_vdc_ad :: bus voltage Output: (uint16_t) :: duty	Duty calculation
	mtr_abs [inline function] Input: (int16_t) s2_value :: input value Output: (int16_t) :: conversion value	Conversion to absolute value
	mtr_limit_value [inline function] Input: (int16_t) s2_value :: input value (int16_t) s2_limit_value :: limit value Output: (int16_t) :: conversion value	Limit process
	mtr_error_check [inline function] Input: none Output: none	Error check

Table 3-21 List of Functions “r_mtr_interrupt.c” [3/3]

File	Function	Process
r_mtr_interrupt.c	R_MTR_1msLessInterrupt Input: none Output: none	[Sensorless mode] Interval timer interrupt (called by INTTRG) Cycle : 1[ms] ・Change motor drive status ・Set reference speed ・Set reference voltage and duty [Voltage control mode] ・Set reference current [Current control mode] ・Call advanced angle control function [Advanced angle control Mode] ・Set time window ・check error
	R_MTR_DelayInterrupt Input: none Output: none	[Sensorless mode] Delay timer interruption ・Stop delay timer ・Set voltage pattern
	mtr_draw_in_pattern_set [inline function] Input: none Output: none	Set voltage pattern in draw-in state
	mtr_detect_zerocross [inline function] Input: (st_mtr_sensorless_control_t) *st_less :: structure for sensorless control variables (uint16_t) *u2_cnt_timeout :: timeout counter (uint16_t) u2_v_pattern :: voltage pattern Output: (uint16_t) :: voltage pattern	Estimate position of rotor from zero-crossing of induced voltage
	mtr_drive_openloop [inline function] Input: none Output: none	・Open-loop drive process ・Calculation of lower limit voltage reference
	mtr_set_angle_shift [inline function] Input: none Output: none	・Calculation of delay count after zero-crossing ・Calculation of zero-crossing filter count
	mtr_start_delay_timer [inline function] Input: (uint16_t) u2_delay_count :: delay count Output: none	Start delay timer
	mtr_stop_delay_timer [inline function] Input: none Output: none	Stop delay timer
	mtr_openloop_pattern_set [inline function] Input: none Output: (uint8_t) u2_pattern :: voltage pattern	Set voltage pattern at open-loop drive
	mtr_set_cmp3_overcurrent [inline function] Input: none Output: none	Set CMP3 for overcurrent
	mtr_set_ics[inline function] Input: none Output: none	ICS communication process
	R_MTR_TRJInterrupt Input: none Output: none	Interval timer interrupt (called by INTTRJ0) Cycle: 50[μs] ・ICS communication during initial position detection process
	mtr_limit [inline function] Input: (int16_t) s2_value :: target value (int16_t) s2_max :: maximum limit (int16_t) s2_min :: minimum limit Output: (int16_t) :: limited value	Limit between maximum and minimum values

Table 3-22 List of Functions “r_mtr_ipd.h”

File	Function	Process
r_mtr_ipd.h	R_MTR_SetCuttoffSource() Input: OPCTL0 &= ~_08_TRD_CUTOFF_SOURCE_COMP0, OPCTL0 |= _08_TRD_CUTOFF_SOURCE_COMP0 Output: None	Set cutoff source of PWMOPA
	R_MTR_ClearCutoffSource() Input: OPCTL0 &= ~_08_TRD_CUTOFF_SOURCE_COMP0 Output: None	Clear cutoff source of PWMOPA
	R_MTR_StartTAU() Input: None Output: None	Start TAU
	R_MTR_StopTAU() Input: None Output: None	Stop TAU

Table 3-23 List of Functions “r_mtr_ipd.c”

File	Function	Process
r_mtr_ipd.c	R_MTR_IpdProcess Input: (st_mtr_ipd_t) *st_ipd :: pointer to IPD structure (uint16_t) u2_current_offset :: current offset (uint16_t) *u2_error_status :: error status pointer Output: none	[Initial Position Detection] Initial position detection process
	R_MTR_ResetForDrive Input: (st_mtr_ipd_t) *st_ipd :: pointer to IPD structure (uint16_t) u2_current_offset :: current offset (int8_t) *s1_dir :: pointer to direction of rotation (uint16_t) *u2_ol_v_pattern_num :: pointer to ring buffer for voltage pattern at open-loop drive Output: none	[Initial Position Detection] Reset peripheral function settings after initial position detection process
	mtr_measure_inductance_effect[inline function] Input:(uint8_t) u1_energized_phase :: number of energized phase (uint8_t) u1_v_pattern :: voltage pattern (st_mtr_ipd_t) *p_st_ipd :: pointer to IPD structure (uint16_t) *u2_error_status :: pointer to error satus Output: (uint16_t) :: TRX count	Measure the time taken to reach threshold current during initial position detection
	mtr_salient_detect_angle [inline function] Input: (st_mtr_ipd_t) *st_ipd :: pointer to IPD structure (uint16_t) u2_current_offset :: current offset (uint16_t) *u2_error_status :: pointer to error status Output: none	Angle detection process for initial position detection with a salient motor
	mtr_salient_detect_polarity [inline function] Input: (st_mtr_ipd_t) *st_ipd :: pointer to IPD structure (uint16_t) u2_current_offset :: current offset (uint16_t) *u2_error_status :: pointer to error status Output: none	Polarity detection process for initial position detection with a salient motor
	mtr_non_salient_detect_angle [inline function] Input: (st_mtr_ipd_t) *st_ipd :: pointer to IPD structure (uint16_t) u2_current_offset :: current offset (uint16_t) *u2_error_status :: pointer to error status Output: none	Angle detection process for initial position detection with a non-salient motor
	mtr_set_initial_position[inline function] Input: (uint8_t) u1_position :: result of initial position detection (int8_t) s1_dir :: direction of rotation (uint16_t) *u2_ol_v_pattern_num :: pointer to ring buffer for voltage pattern at open-loop drive Output: none	Set first open-loop pattern depending on the result of initial position detection
	mtr_lower_arm_on[inline function] Input: (uint16_t) u2_low_on_period :: period for lower arms on Output: none	Set all lower arms on
	mtr_prepare_energize_phase[inline function] Input: (uint8_t) u1_energized_phase :: number of energized phases (uint8_t) u1_v_pattern ::voltage pattern Output: none	Set voltage pattern for initial position detection
	mtr_output_stop[inline function] Input: none Output: none	Stop voltage output
	mtr_reset_timer[inline function] Input: none Output: none	Reset TRD2 and TRJ
	R_MTR_TRDIntLevel Input: (uint8_t) u1_level :: priority of interrupt Output: none	Set TRD2 interrupt priority level

Table 3-1 List of Functions “r_mtr_ol2cl_ctrl.c”

File	Function	Process
r_mtr_ol2cl_ctrl.c	R_MTR_OL2CLCtrl Input: (uint16_t) u2_cnt_ol_speed :: Counter for pattern change at open-loop drive (uint8_t) u1_flag_pattern_change :: flag for zero-crossing detection (uint16_t) u2_ol_pattern_period :: period for pattern change at open-loop drive (int16_t) *s2_v_uvwn_ad :: pointer to U-phase voltage (st_mtr_ol2cl_t) *st_ol2cl :: pointer to sensorless control transition structure (int8_t) s1_dir :: direction of rotation (uint8_t) u2_ol_v_pattern :: voltage pattern at open-loop drive Output: none	Transition process from open-loop drive to sensorless control ・Compare 3-phase voltage to threshold voltage ・Number of counts for zero-crossing detection during open-loop drive ・Detect rotor phase lead/lag
	mtr_get_bemf_voltage [inline function] Input: (int8_t) s1_dir :: Direction of rotation (uint8_t) u2_ol_v_pattern :: voltage pattern at open-loop drive (int16_t) *s2_v_uvwn_ad :: pointer to U-phase voltage Output: (int16_t) :: induced voltage	Get induced voltage
	mtr_openloop_phase_ctrl [inline function] Input: (int8_t) s1_dir :: direction of rotation (uint8_t) u2_ol_v_pattern :: voltage pattern at open-loop drive (int16_t) *s2_v_uvwn_ad :: pointer to U-phase voltage (st_mtr_ol2cl_t) *st_ol2cl :: pointer to sensorless control transition structure Output:なし	Detect rotor phase lead/lag
	mtr_abs [inline function] Input: (int16_t) s2_value :: input value Output: (int16_t) :: conversion value	Conversion to absolute value
	R_MTR_BemfSignalFilter Input: (uint16_t) u2_cnt_ol_speed :: counter for pattern change at open-loop drive (uint8_t) *u1_pre_bemf_signal :: pointer to previous estimated hall pattern signal (st_mtr_ol2cl_t) *st_ol2cl :: pointer to sensorless control transition structure (uint16_t) u2_temp_signal :: estimated hall pattern signal (uint8_t) u1_flag_ol_pattern_change :: flag for c change of voltage pattern during open-loop drive Output: (int16_t) :: estimated hall pattern signal	Filter noise signal during voltage pattern change for estimated hall pattern signals
	R_MTR_SetSpeedRefOL2CL Input: (st_mtr_ol2cl_t) *st_ol2cl :: pointer to sensorless control transition structure (int8_t) s1_dir :: direction of rotation (int16_t) *s2_ref_speed_rad_ctrl :: pointer to reference speed Output: none	Set reference speed during open-loop drive

Table 3-24 List of Functions “r_dsp.h”

File	Function	Process
r_dsp.h	FIX_fromfloat Input: (float) x / input value n / Q-format Output: (int16_t) y / conversion value	Convert float type value x to fixed-point value y in Qn format
	FIX32_fromfloat Input: (float) x / input value n / Q-format Output: (int32_t) y / conversion value	Convert float type value x to fixed-point value y in Qn format

Table 3-25 List of Functions “r_mtr_aa_ctrl.obj”

File	Function	Process
r_mtr_aa_ctrl.obj	R_MTR_AACtrlVoltageMode Input：(st_mtr_aa_t) *st_aa :: pointer to advanced angle control structure ：(int16_t) s2_speed_rad :: rotational speed ：(s2_ref_v_ctrl) :: reference voltage ：(int16_t) *s2_vn_offset_ad :: pointer to neutral voltage offset Output：none	[Voltage control mode] [Advanced angle control mode] Advanced angle control process
	R_MTR_AACtrlCurrentMode 入力：(st_mtr_aa_t) *st_aa :: pointer to advanced angle control structure ：(int16_t) s2_speed_rad :: rotational speed ：(s2_ref_v_ctrl) :: reference voltage ：(int16_t) *s2_vn_offset_ad :: pointer to neutral voltage offset ：(int16_t) *s2_asr_intg :: integral term of ASR Output：none	[Current control mode] [Advanced angle control mode] Advanced angle control process

Table 3-26 List of Functions “r_mtr_rl78g24.c”

File	Function	Process
r_mtr_rl78g24.c	R_MTR_InitUnusedPins Input: none Output: none	Initialization of unused pins

Table 3-27 List of Functions “Config_COMP3_user.c”

File	Function	Process
Config_COMP3_user.c	R_MTR_EnableCMP3Intr Input: None Output: None	Enable the interrupt of CMP3
	R_MTR_DisableCMP3Intr Input: None Output: None	Disable the interrupt of CMP3

Table 3-28 List of Functions “Config_TRG.h”

File	Function	Process
Config_TRG.h	R_MTR_ ClearTrgImfa() Input: TRGIMFA = 0 Output: None	Clear IMFA flag for TRG

Table 3-29 List of Functions “Config_TAU0_3_user.h”

File	Function	Process
Config_TAU0_3.h	R_MTR_SetTDR03(cnt) Input: TDR03 = cnt Output: None	[Sensorless mode] Set delay timer count
	R_MTR_StartDelayCnt() Input: TS0L_bit.no3 = 1 Output: None	[Sensorless mode] Start delay timer
	R_MTR_StopDelayCnt() Input: TT0L_bit.no3 = 0 Output: None	[Sensorless mode] Stop delay timer
	R_MTR_ClearINTTM03() Input: TMIF03 = 0 Output: None	[Sensorless mode] Clear interruption flag of delay timer

Table 3-30 List of Functions “Config_INTC_user.c”

File	Function	Process
Config_INTC_user.c	R_MTR_EnableHallIntr Input: None Output: None	[Hall effect sensor mode] Enable the interruption of Hall effect sensor
	R_MTR_DisableHallIntr Input: None Output: None	[Hall effect sensor mode] Disable the interruption of Hall effect sensor

Table 3-31 List of Functions “Config_PWMOPA_user.c”

File	Function	Process
Config_PWMOPA_user.c	R_MTR_RecoverForcedShutdown Input: None Output: (uint8_t) u1_hw_current_error_flag :: Hardware current error flag	Recover from forced shutdown process

Table 3-32 List of Functions “Config_TRD0_TRD1.h”

File	Function	Process
Config_TRD0_TRD1.h	R_MTR_ ClearTRD0IMFA() Input: TRDSR0_bit.no0 = 0 Output: None	Clear IMFA flag of TRD

	R_MTR_SetTRDGRD0(duty)
	Input: TRDGRD0 = ((uint16_t)MTR_CARRIER_SET - duty);
	TRDGRC1 = ((uint16_t)MTR_CARRIER_SET - duty);
	TRDGRD1 = ((uint16_t)MTR_CARRIER_SET - duty);
	Output: None
		Set TRDgeneral register D0, C1, D1 for PWM duty

	R_MTR_SetTRD2ReloadFlag() Input: TRDRDT1 |= _01_TRD_TRDRDT1_RELOAD_REGISTER Output: None	Enable TRD simultaneous update trigger register
	R_MTR_ClearTRD0UDF() Input: TRDSR1_bit.no5 = 0 Output: None	Clear underflow flag for TRD
	R_MTR_SetDefaultTRDADTB0() Input: TRDADTB0 = ((uint16_t)MTR_CARRIER_SET - (uint16_t)MTR_AD_TRG_DEFAULT_CNT) Output: None	Set TRD A/D conversion trigger buffer register 0 to default count
	R_MTR_SetDefaultTRDADTB1() Input: TRDADTB1 = ((uint16_t)MTR_CARRIER_SET - (uint16_t)MTR_AD_TRG_DEFAULT_CNT) Output: None	Set TRD A/D conversion trigger buffer register 0 to default count

Table 3-33 List of Functions “Config_TRD0_TRD1.c”

File	Function	Process
Config_TRD0_TRD1.c	R_MTR_PatternOutput Input: (uint16_t) u2_pattern :: Conducting pattern (uint16_t) u2_pwm_duty : Duty Output: (uint8_t) u1_temp_error_flag :: Error flag	Set TRD registers to output PWM ・select output ports ・set PWM duty
	R_MTR_DisableTRDOutput Input: None Output: None	Disable TRD PWM output
	R_MTR_PortOutput Input: None Output: None	Stop output of voltage
	R_MTR_StartTRD Input: None Output: None	Start TRD
	R_MTR_StopTRD Input: None Output: None	Stop TRD
	R_MTR_SetTRDADTB0 Input: (int16_t) s2_offset :: offset Input: None Output: None	[Sensorless mode] Set TRD A/D conversion trigger buffer register 0
	R_MTR_SetTRDADTB1 Input: (int16_t) s2_offset :: offset Input: None Output: None	[Sensorless mode] Set TRD A/D conversion trigger buffer register 1

Table 3-34 List of Functions “Config_ADC_user.c”

File	Function	Process
Config_ADC_user.c	R_MTR_GetIdcVdcAdc Input: (int16_t) *s2_idc_ad :: The pointer to DC link current (int16_t) *s2_v_dc :: The pointer to bus voltage (uint8_t) u1_flag_ad_trg1_vi :: A/D conversion channel switching flag Output: None	Get the results of A/D conversion of DC link current and bus voltage
	R_MTR_SwitchChADC Input: (uint16_t) u2_pattern_adc :: Voltage pattern (int8_t) s1_dir :: Rotational direction (int16_t) *s2_ad_trg_adjust_cnt :: The pointer to AD trigger 0 adjustment count (uint8_t) *u1_flag_ad_trg1_vi :: The pointer to AD conversion channel switching flag (int16_t) *s2_ad_trg_offset_cnt :: The pointer to AD trigger offset (int16_t) *s2_ad_trg_diff_cnt :: The pointer to the difference in AD trigger count between previous and current carrier Output: None	[Sensorless mode] ・Switch A/D conversion channel for DC link current and bus voltage ・Call A/D conversion channels selection for U, V and W voltage function
	mtr_select_vuvw_adc_channel [inline function] Input: (uint16_t) u2_pattern_adc :: Voltage pattern (int8_t) s1_dir :: Rotational direction (int16_t) *s2_ad_trg_adjust_cnt :: The pointer to AD trigger 0 adjustment count Output: None	[Sensorless mode] Select A/D conversion channels for U, V and W voltage
	R_MTR_GetVuvwADC Input: (uint16_t) u2_pattern_ad :: Voltage pattern (int8_t) s1_dir :: Rotational direction (int16_t) *s2_v_uvw :: The pointer to U-phase voltage (int16_t) *s2_ad_trg_vn_buff :: The pointer to A/D conversion result of non-energized phase voltage (uint8_t) *u1_flag_duty_th :: The pointer to duty threshold exceed flag Output: None	[Sensorless mode] Get the results of A/D conversion of U, V and W voltage
	R_MTR_SwitchChVI Input: (uint8_t) *u1_flag_ad_trg1_vi :: The pointer to AD conversion channel switching flag Output: None	[Hall effect sensor mode] Switch A/D conversion channel for DC link current and bus voltage

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 3-35 List of variables “main.c”

Variable	Type	Qn	PU	Content	Remarks
g_u1_system_mode	static uint8_t	Q0	-	Mode system management	
g_u1_motor_status	static uint8_t	Q0	-	Motor status management	
g_u1_error_status	static uint8_t	Q0	-	Error status management	
g_u2_conf_hw	uint16_t	Q0	-	RMW configuration variables	
g_u2_conf_sw	uint16_t	Q0	-		
g_u2_conf_tool	uint16_t	Q0	-		
gui_u1_active_gui	uint8_t	Q0	-		
g_u2_conf_sw_ver	uint16_t	Q0	-		
com_u1_run_event	uint8_t	Q0	-	Input event and change run mode	0: MTR_EVENT_STOP 1: MTR_EVENT_DRIVE 2: MTR_EVENT_ERROR 3: MTR_EVENT_RESET
g_u1_run_event	uint8_t	Q0	-		
g_u2_system_error	uint16_t	Q0	-	System error management	

Table 3-36 List of variables “r_mtr_ics.c”[1/2]

Variable	Type	Qn	PU	Content	Remarks
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_m	float	-	-	Induced voltage constant [V･s/rad]	
com_f4_mtr_j	float	-	-	Rotor inertia [kg･m^2/(rad^2)]	
com_f4_limit_v_ref	float	Q0	-	Lower limit of voltage reference [V]	
com_f4_ramp_limit_speed_rpm	float	Q0	-	Limit of acceleration [rpm/ms]	Mechanical angle
com_s2_ref_speed_rpm	int16_t	Q0	-	Command rotational speed [rpm]	Mechanical angle
com_u2_offset_calc_cnt	uint16_t	Q0	-	Current offset detection time	
com_f4_ramp_limit_v	float	Q0	-	Limit of variation of voltage [V/ms]	[Voltage control mode]
com_f4_asr_omega_hz	float	-	-	Frequency for speed PI control [Hz]	
com_f4_asr_zeta	float	-	-	Damping factor for speed PI control	
com_f4_ramp_limit_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)]	
com_f4_mtr_d1	float	-	-	Coefficient of kinetic friction [kg･m^2/(rad^2･s)]	
com_f4_acr_nf_hz	float	-	-	Current PI control natural frequency [Hz]	
com_f4_asr_nf_hz	float	-	-	Speed PI control natural frequency [Hz]	
com_f4_asr_ki_aug	float	-	-	Augmentation rate for integral part of ASR	
com_f4_acr_deadband_lsb	float	-	-	Deadband of current PI calculation [LSB]	
com_f4_asr_deadband_lsb	float	-	-	Deadband of speed PI calculation [LSB]	
com_f4_start_ref_v	float	Q0	-	Initial reference voltage at start-up	[Hall effect sensor-based voltage control mode]
com_f4_start_ref_idc	float	Q0	-	Initial reference current at start-up	[Hall effect sensor-based current control mode]
com_f4_draw_in_ref_v	float	Q0	-	Reference voltage at draw-in [V]	[Sensorless voltage control mode]
com_f4_ol_ref_v	float	Q0	-	Reference voltage at open-loop drive[V]	
com_f4_draw_in_ref_idc	float	Q0	-	Command current at draw-in [V]	[Sensorless current control mode]
com_f4_ol_ref_idc	float	Q0	-	Command current at open-loop drive[V]	
com_s2_ol2lcl_speed_rpm	int16_t	Q0	-	Speed allowed to transit to PI control[rpm]	Mechanical angle [Sensorless mode]
com_s2_ol2cl_ramp_speed_rpm	int16_t	Q0	-	Acceleration at transition to PI control [rpm/ms]	Mechanical angle [Sensorless mode]
com_s2_angle_shift_adjust	int16_t	Q0	-	Adjust delay counts	[Sensorless mode]

Table 3-37 List of variables “r_mtr_ics.c”[2/2]

Variable	Type	Qn	PU	Content	Remarks
com_s2_ad_trg_adjust_up_cnt	int16_t	-	-	Adjustment count of A/D trigger for ascending non-energized phase voltage	[Sensorless mode]
com_s2_ad_trg_adjust_down_cnt	int16_t	-	-	Adjustment count of A/D trigger for descending non-energized phase voltage	
com_s2_ad_trg_diff_cnt	int16_t	-	-	Count difference of A/D between previous carrier and current carrier	
com_f4_sal_angle_current	float	-	-	Threshold current of angle detection for salient motor	[Initial Position Detection]
com_u4_sal_angle_th	uint32_t	Q0	-	TRX count value differential of angle detection for salient motor	
com_u2_sal_angle_discharge	uint16_t	Q0	-	Discharging time for angle detection for salient motor	
com_f4_sal_polarity_current	float	-	-	Threshold current of polarity detection for salient motor	
com_u4_sal_polarity_th	uint32_t	Q0	-	TRX count value differential of polarity detection for salient motor	
com_u2_sal_polarity_discharge	uint16_t	Q0	-	Discharging time for polarity detection for salient motor	
com_f4_non_sal_current	float	-	-	Threshold current of angle detection for non-salient rotor	
com_u4_non_sal_th	uint32_t	Q0	-	TRX count value differential of angle detection for non-salient rotor	
com_u2_non_sal_discharge	uint16_t	Q0	-	Discharging time for angle detection for non-salient motor	
com_s2_aa_speed_err_th_rpm	int16_t	Q0	-	Speed error threshold for Advanced Angle Control [rpm]	[Advanced Angle control mode]
com_f4_aa_pi_nf_hz	float	Q0	-	PI controller natural frequency for Advanced Angle Control [Hz]	
com_f4_aa_speed_err_lpf_cof_hz	float	Q0	-	Speed error LPF cutoff frequency [Hz]	
com_f4_aa_ramp_limit_v	float	Q0	-	Limit of rate of neutral voltage change for Advanced Angle Control [V/ms]	
com_f4_aa_ramp_limit_angle	float	Q0	-	Limit of rate of angle offset change for Advanced Angle Control [rad/ms]	
com_f4_aa_vlim	float	Q0	-	Voltage limit [V]	
com_s2_enable_write	int16_t	Q0	-	Variable to allow to input RMW structure	
g_s2_enable_write	int16_t	Q0	-	Variable to allow to input RMW structure	
st_ics_input	mtr_ctrl_input_t	Q0		Structure for RMW input	structure

Table 3-38 List of variables “r_mtr_parameter.h / Structure : st_mtr_parameter_t”

Variable	Type	Qn	PU	Content	Remarks
u2_mtr_pp	uint16_t	Q0	-	Number of pole pairs	
s2_mtr_r	int16_t	Q17	Resistance (voltage/current)	Resistance [PU]	
s2_mtr_ld	int16_t	Q19	Inductance (resistance/angular frequency)	D-axis inductance [PU]	
s2_mtr_lq	int16_t	Q19	Inductance (resistance/angular frequency)	Q-axis inductance [PU]	
s2_mtr_m	int16_t	Q16	BEMF constant (voltage/angular frequency)	Induced voltage constant [PU]	
s2_mtr_j	int16_t	Q12	Inertia (BEMF constant ×current×(number of pole pairs/angular frequency) ^2)	Rotor inertia [PU]	

Table 3-39 List of variables “r_mtr_driver_access.h / Structure : mtr_ctrl_input_t”[1/2]

Variable	Type	Qn	PU	Content	Remarks
s1_direction	int8_t	Q0	-	Direction of rotation	
s2_ref_speed_rad	int16_t	Q14	Angular frequency	Command rotational speed [PU]	electric angle
s2_ramp_limit_speed_rad	int16_t	Q14	Angular frequency	Limit of acceleration [PU]	electric angle
u2_offset_calc_cnt	uint16_t	Q0	-	Current off set detection time	
s2_limit_v_ref	int16_t	Q13	Voltage	Lower limit of voltage reference [PU]	
s2_ramp_limit_v	int16_t	Q13	Voltage	Limit of variation of voltage [PU]	[Voltage control mode]
s2_ramp_limit_current	int16_t	Q13	Current	Limit of variation of current [PU]	[Current control mode]
s2_start_ref_v	int16_t	Q13	Voltage	Initial reference voltage at start-up	[Hall effect sensor mode] [Voltage control mode]
s2_start_ref_idc	int16_t	Q13	Current	Initial reference current at start-up	[Hall effect sensor mode] [Current control mode]
s2_draw_in_ref_v	int16_t	Q13	Voltage	Reference voltage at draw-in [PU]	[Voltage control mode]
s2_ol_ref_v_buff	int16_t	Q13	Voltage	Reference voltage at open-loop drive buffer [PU]	[Voltage control mode]
s2_draw_in_ref_idc	int16_t	Q13	Current	DC link current reference at draw-in [PU]	[Current control mode]
s2_ol_ref_idc_buff	int16_t	Q13	Current	Openloop start reference DC link current buffer [PU]	[Current control mode]
s2_ol2cl_speed_rad	int16_t	Q14	Angular frequency	Speed allowed to transition to PI control [PU]	electric angle [Sensorless mode]
s2_ol2cl_ramp_speed_rad	int16_t	Q14	Angular frequency	Acceleration at transition to PI control [PU]	electric angle [Sensorless mode]
s2_cl2ol_speed_rad	int16_t	Q14	Angular frequency	Speed to transition to open-loop drive [PU]	electric angle [Sensorless mode]
s2_angle_shift_adjust	int16_t	Q0	-	adjust delay counts	[Sensorless mode]
s2_ad_trg_adjust_up_cnt	int16_t	Q0	-	Adjustment count of A/D trigger for ascending non-energized phase voltage	
s2_ad_trg_adjust_down_cnt	int16_t	Q0	-	Adjustment count of A/D trigger for descending non-energized phase voltage	
s2_ad_trg_diff_cnt	int16_t	Q0	-	Count difference of A/D between previous carrier and current carrier	

Table 3-40 List of variables “r_mtr_driver_access.h / Structure : mtr_ctrl_input_t”[2/2]

Variable	Type	Qn	PU	Content	Remarks
u2_sal_angle_current	uint16_t	Q0	-	Threshold current of angle detection for salient motor	[Initial Position Detection]
u4_sal_angle_th	uint32_t	Q0	-	TRX count value differential of angle detection for salient motor	
u2_sal_angle_discharge	uint16_t	Q0	-	Discharging time for angle detection for salient motor	
u2_sal_polarity_current	uint16_t	Q0	-	Threshold current of polarity detection for salient motor	
u4_sal_polarity_th	uint32_t	Q0	-	TRX count value differential of polarity detection for salient motor	
u2_sal_polarity_discharge	uint16_t	Q0	-	Discharging time for polarity detection for salient motor	
u2_non_sal_current	uint16_t	Q0	-	Threshold current of angle detection for non-salient rotor	
u4_non_sal_th	uint32_t	Q0	-	TRX count value differential of angle detection for non-salient rotor	
u2_non_sal_discharge	uint16_t	Q0	-	Discharging time for angle detection for non-salient motor	
s2_aa_speed_err_th_rad	int16_t	Q14	Angular frequency	Speed error threshold for Advanced Angle Control [PU]	[Advanced Angle control mode]
s2_aa_ramp_limit_v	int16_t	Q13	Voltage	Limit of rate of neutral voltage change for Advanced Angle Control [PU]	
s2_aa_ramp_limit_angle	int16_t	Q12	Angle	Limit of rate of angle offset change for Advanced Angle Control [PU]	
s2_aa_vlim	int16_t	Q13	Voltage	Voltage limit [PU]	
st_gain_v_buf	st_mtr_ctrl_gain_voltage_mode_t	-	-	Structure for PI control	[Voltage control mode] structure
st_gain_i_buf	st_mtr_ctrl_gain_current_mode_t	-	-	Structure for PI control	[Current control mode] structure
st_ctrl_params	st_mtr_design_parameter_h	-	-	Structure for PI gain calculation	structure
st_motor	st_mtr_parameter_t	-	-	Structure for motor parameter	structure

Table 3-41 List of variables “r_mtr_driver_access.c”

Variable	Type	Qn	PU	Content	Remarks
st_ics_input_buff	st_mtr_ctrl_input_t	-	-	Buffer for RMW input	structure
g_u1_trig_enable_write	uint8_t	Q0	-	Flag to allow to input RMW values	

Table 3-42 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	
u1_current_event	uint8_t	Q0	-	Execution event	

Table 3-43 List of variables “r_mtr_statemachine.c”

Variable	Type	Qn	PU	Content	Remarks
state_transition_table [MTR_SIZE_EVENT] [MTR_SIZE_STATE]	static uint8_t	Q0	-	Macro array for state transition	
action_table [MTR_SIZE_EVENT] [MTR_SIZE_STATE]	static mtr_action_t	Q0	-	Function array for state transition	

Table 3-44 List of variables “r_mtr_120.h / Structure : st_mtr_v_pi_t”

Variable	Type	Qn	PU	Content	Remarks
s2_pre_speed_rad	int16_t	Q14	Angular frequency	Previous rotational speed [PU]	[Voltage control mode]
st_pi	st_mtr_pi_t	-	-	Speed PI structure	

Table 3-45 List of variables “r_mtr_120.h / Structure : st_mtr_pi_t”

Variable	Type	Qn	PU	Content
s2_kp	int16_t	Current: Q17 Speed: Q14 [Current control mode] Q13 [Voltage control mode]	Current: Resistance Speed: Current/Angular frequency [Current control mode] Voltage/Angular frequency [Voltage control mode]	Proportional gain for speed PI control
s2_kidt	int16_t	Current: Q21 Speed: Q21 [Current control mode] Q15 [Voltage control mode]	Current: Resistance Speed: Current/Angular frequency [Current control mode] Voltage/Angular frequency [Voltage control mode]	Integral gain for speed PI control
s2_intg	int16_	Current: Q13 Speed: Q13 [Current control mode] Q13 [Voltage control mode]	Current: Voltage Speed: Current [Current control mode] Voltage [Voltage Control mode]	Integral term
s2_ilimit	int16_t	Current: Q13 Speed: Q13 [Current control mode] Q13 [Voltage control mode]	Current: Voltage Speed: Current [Current control mode] Voltage [Voltage Control mode]	Integral limit (up/down symmetry)
s2_deadband	int16_t	Current: Q13 Speed: Q14	Current: Current Speed: Angular frequency	Deadbandfor integral part

Table 3-46 List of variables “r_mtr_120.h / Structure : st_mtr_acr_t”

Variable	Type	Qn	PU	Content	Remarks
s2_limit_idc	int16_t	Q13	Current	DC link current limit	[Current control mode]
s2_ramp_limit_current	int16_t	Q13	Current	Limit value for current rise [PU/ms]	
st_pi	st_mtr_pi_t	-	-	Current PI structure	

Table 3-47 List of variables “r_mtr_120.h / Structure : st_mtr_asr_t”

Variable	Type	Qn	PU	Content	Remarks
s2_d0_div_pm	int16_t	Q13	-	Static friction coefficient/(number of pole pairs × BEMF constant)	[Current control mode]
s2_d1_div_p2m	int16_t	Q19	-	Kinetic friction coefficient/((number of pole pairs)^2 × BEMF constant)	
st_pi	st_mtr_pi_t	-	-	Speed PI structure	

Table 3-48 List of variables “r_mtr_120.h / Structure : st_mtr_hall_control_t”

Variable	Type	Qn	PU	Content	Remarks
u2_hall_signal	uint16_t	-	-	Signal from hall effect sensor	[Hall effect sensor mode]
u2_flag_1st_interrupt	uint16_t	-	-	Flag for first hall effect sensor interruption	
s2_start_ref_v	int16_t	Q13	Voltage	Initial reference voltage for start-up	[Hall effect sensor-based voltage control mode]
s2_start_ref_idc	int16_t	Q13	Current	Initial reference current for start-up	[Hall effect sensor-based current control mode]

Table 3-49 List of variables “r_mtr_120.h / Structure : st_mtr_sensorless_control_t”[1/2]

Variable	Type	Qn	PU	Content	Remarks
u1_state_draw_in	uint8_t	Q0	-	Draw-in state management	[Sensorless mode]
u1_flag_pattern_change	uint8_t	Q0	-	Flag for zero-crossing detection	
u1_bemf_signal	uint8_t t	Q0	-	Estimated Hall pattern	
u1_pre_bemf_signal	uint8_t	Q0	-	Previous estimated Hall pattern	
u1_zc_detect_method	uint8_t	Q0	-	Method of zero-crossing detection	
u1_state_control_mode	uint8_t	Q0	-	State management for control mode	
u1_state_driving	uint8_t	Q0	-	State management for driving speed	
u1_flag_ol_pattern_change	uint8_t	Q0	-	Flag for voltage pattern change at open-loop drive	
u2_cnt_ol_speed	uint16_t	Q0	-	Counter for patten change at open-loop drive	
u2_ol_pattern_period	uint16_t	Q0	-	Period for pattern change at open-loop drive	
u2_cnt_draw_in	uint16_t	Q0	-	Counter for pattern change at draw-in	
u2_ol_v_pattern	uint16_t	Q0	-	Voltage pattern at open-loop drive	
u2_ol_v_pattern_num	uint1_t	Q0	-	Ring buffer for voltage pattern at open-loop drive	
u2_bemf_delay	uint16_t	Q0	-	Delay counts	
u2_zerocross_filter_cnt	uint16_t	Q0	-	Zerocross filter count	
s2_ol_ramp_limit_v	int16_t	Q13	Voltage	Limit of rate of voltage change during open-loop control	
s2_cl2ol_speed_rad	int16_t	Q13	Angular frequency	Transition speed from sensorless control to open-loop [PU]	
s2_vu_ad	int16_t	Q0	-	U phase voltage	
s2_vv_ad	int16_t	Q0	-	V phase voltage	

Table 3-50 List of variables “r_mtr_120.h / Structure : st_mtr_sensorless_control_t”[2/2]

Variable	Type	Qn	PU	Content	Remarks
s2_vw_ad	int16_t	Q0	-	W phase voltage	
s2_vn_ad	int16_t	Q0	-	Virtual neutral voltage	
s2_vn_ctrl_ad	int16_t	Q0	-	Virtual neutral voltage for control	
s2_vn_offset_ad	int16_t	Q0	-	Virtual neutral voltage offset	
s2_angle_shift_adjust	int16_t	Q0	-	Virtual hall pattern delay adjustment value	
u2_pattern_adc	uint16_t	Q0	-	Voltage pattern for A/D conversion channel selection	
s2_ad_trg_adjust_cnt	int16_t	Q0	-	Adjustment count for A/D trigger	
s2_ad_trg_adjust_up_cnt	int16_t	Q0	-	Adjustment count of A/D trigger for ascending non-energized phase voltage	
s2_ad_trg_adjust_down_cnt	int16_t	Q0	-	Adjustment count of A/D trigger for descending non-energized phase voltage	
s2_ad_trg_diff_cnt	int16_t	Q0	-	Count difference of A/D between previous carrier and current carrier	
s2_ad_trg_offset_cnt	int16_t	Q0	-	A/D trigger offset	
s2_ad_trg_vn_buff[2]	int16_t	Q0	-	A/D conversion result of non-energized phase voltage [0]: Previous non-energized phase voltage value [1]: Non-energized phase voltage value	
u1_flag_duty_th	uint8_t	Q0	-	Exceed PWM duty threshold flag	
s2_duty_th	int16_t	Q0	-	PWM duty threshold	
u2_v_const_period	uint16_t	Q0	-	Period for pattern change at draw-in	[Sensorless voltage control mode]
s2_draw_in_ref_v	int16_t	Q13	Voltage	Draw-in command voltage [PU]	
s2_ol_ref_v	int16_t	Q13	Voltage	Startup voltage [PU]	
s2_ol_ref_v_buff	int16_t	Q13	Voltage	Startup voltage buffer [PU]	
u2_idc_const_period	uint16_t	Q0	-	Period for pattern change at draw-in	[Sensorless current control mode]
s2_draw_in_ref_idc	int16_t	Q13	Current	Draw-in command current [PU]	
s2_ol_ref_idc	int16_t	Q13	Current	Startup current [PU]	
s2_ol_ref_idc_buff	int16_t	Q13	Current	Startup current buffer [PU]	
st_ol2cl	st_mtr_ol2cl_t	-	-	Sensorless switching process structure	

Table 3-51 List of variables “r_mtr_120.h / Structure : st_mtr_120_control_t” [1/2]

Variable	Type	Qn	PU	Content	Remarks
u1_state_drive	uint8_t	-	-	Drive mode status	0: Current offset measurement for calibration executing 1: Current offset measurement for calibration finished 2: IPD process executing 3: IPD process finished 4: Open-loop driving 5: Speed PI drive 6: Brake mode 7: Driving finished
s1_ref_dir	int8_t	Q0	-	Reference direction of rotation	
s1_dir	int8_t	Q0	-	Direction of rotation	
u1_flag_charge_cap	uint8_t	Q0	-	Flag for charge capacitor	
u1_flag_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	
u2_run_mode	uint16_t	Q0	-	Run mode	
u2_state_speed_ref	uint16_t	Q0	-	Speed reference mode	
u2_speed_timer_cnt	uint16_t	Q0	-	Value of timer counter	
u2_pre_speed_timer_cnt	uint16_t	Q0	-	Previous value of timer counter	
u2_timer_cnt_buff[6]	uint16_t	Q0	-	Timer counts for 60 degree	
u2_timer_cnt_sum	uint16_t	Q0	-	Total counter value for speed calculation	
u2_v_pattern	uint16_t	Q0	-	Voltage pattern	
u2_pwm_duty	uint16_t	Q0	-	PWM duty	
u2_cnt_timeout	uint16_t	Q0	-	Counter for timeout error	
s2_speed_rad	int16_t	Q14	Angular frequency	Rotational speed [PU]	
s2_ref_speed_rad	int16_t	Q14	Angular frequency	Reference rotational speed [PU]	
s2_ref_speed_rad_ctrl	int16_t	Q14	Angular frequency	Reference rotational speed for control [PU]	
s2_ramp_limit_speed_rad	int16_t	Q14	Angular frequency	Limit of acceleration [PU]	
s2_vdc_ad_bit	int16_t	Q0	-	A/D conversion result of bus voltage	
s2_idc_ad_bit	int16_t	Q0	-	A/D conversion result of DC link current	
s2_vdc_ad	int16_t	Q13	Voltage	Bus voltage [PU]	
s2_idc_ad	int16_t	Q13	Current	DC link current [PU]	
u4_error_status	uint32_t	Q0	-	Error status	
u4_offset_idc_ad_sum	uint32_t	Q0	-	DC link current offset value integral value	
u1_use_cmp3	uint8_t	Q0	-	Selection of function of CMP3	
u1_state_current_offset	uint8_t	Q0	-	State management for measurement of U phase current offset	
u2_offset_idc_adc	uint16_t	Q0	-	DC link current offset value	

Table 3-52 List of variables “r_mtr_120.h / Structure : st_mtr_120_control_t” [1/2]

Variable	Type	Qn	PU	Content	Remarks
u2_offset_calc_cnt	uint16_t	Q0	-	Offset current measurement count	
s2_limit_over_voltage	int16_t	Q13	Voltage	Overvoltage limit value	
s2_limit_under_voltage	int16_t	Q13	Voltage	Undervoltage limit value	
s2_limit_speed_rad	int16_t	Q14	Angular frequency	Overspeed limit value	
s2_limit_over_current	int16_t	Q13	Current	Overcurrent limit value	
s2_limit_v	int16_t	Q13	Voltage	Lower limit of voltage	
s2_limit_v_ref	int16_t	Q13	Voltage	Lower limit of voltage reference	
s2_ref_v_ctrl	int16_t	Q13	Voltage	Reference voltage [PU]	
s2_ref_v	int16_t	Q13	Voltage	Target voltage [PU]	[Voltage control mode]
u2_state_voltage_ref	uint16_t	Q0	-	Reference voltage mode	
s2_ramp_limit_v	int16_t	Q13	Voltage	Voltage change range limit value [PU]	
st_asr	st_mtr_v_pi_t	-	-	ASR structure	
u2_state_current_ref	uint16_t	Q0	-	Reference current mode	[Current control mode]
s2_ref_idc	int16_t	Q13	Current	Target current [PU]	
s2_ref_idc_ctrl	int16_t	Q13	Current	Reference current [PU]	
st_asr	st_mtr_asr_t			ASR structure	
st_acr	st_mtr_acr_t			ACR structure	
st_ipd	st_mtr_ipd_t	-	-	Structure for initial position detection	
st_aa	st_mtr_aa_t	-	-	Structure for advanced angle control	
st_stm	st_mtr_statemachine_t	-	-	Structure for state machine	
st_motor	st_mtr_parameter_t	-	-	Structure for motor parameter	
st_hall	st_mtr_hall_control_t			Structure for hall effect sensor control	
st_less	st_mtr_sensorless_control_t			Structure for sensorless control	

Table 3-53 List of variables “r_mtr_120.c”

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

Table 3-54 List of variables “r_mtr_interrupt.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]	uint16_t	Q0	-	Array for voltage pattern at open-loop drive (CW)	
g_u2_ol_v_pattern_table_CCW[7]	uint16_t	Q0	-	Array for voltage pattern at open-loop drive (CCW)	
g_u2_chopping_pattern_table_CW[6]	uint16_t	Q0	-	Array for chopping pattern (CW)	
g_u2_chopping_pattern_table_CCW[6]	uint16_t	Q0	-	Array for chopping pattern (CCW)	
g_u1_cnt_ics	static uint8_t	Q0	-	Decimation counter for RMW process	
g_u1_flag_ad_trg1_vi	static uint8_t	Q0		A/D conversion channel switching flag	
g_u1_flag_delay	static uint8_t	Q0	-	Delay interrupt flag	

Table 3-55 List of variables “r_mtr_ctrl_gain.h / Structure : st_mtr_ctrl_gain_voltage_mode_t”

Variable	Type	Qn	PU	Content	Remarks
s2_asr_kp	int16_t	Q13	Voltage/Angular frequency	Speed PI proportional gain [PU]	
s2_asr_kidt	int16_t	Q15	Voltage/Angular frequency	Speed PI integral gain * dt [PU]	
s2_asr_deadband	int16_t	Q14	Angular frequency	Integral part deadband	
s2_aa_kp	int16_t	Q21	-	Advanced angle control PI proportional gain	
s2_aa_kidt	int16_t	Q21	-	Advanced angle control PI ki * dt	
s2_aa_lpf_in_k	int16_t	Q15	-	Advanced angle control LPF numerator	

Table 3-56 List of variables “r_mtr_ctrl_gain.h / Structure : st_mtr_ctrl_gain_current_mode_t”

Variable	Type	Qn	PU	Content	Remarks
s2_asr_kp	int16_t	Q14	Current/ Angular frequency	Speed PI proportional gain [PU]	
s2_asr_kidt	int16_t	Q21	Current/ Angular freqeuncy	Speed PI integral gain * dt [PU]	
s2_asr_deadband	int16_t	Q14	Angular frequency	Speed integral part deadband	
s2_acr_kp	int16_t	Q17	Resistance	DC link current control proportional gain [PU]	
s2_acr_kidt	int16_t	Q21	Resistance	DC link current control integral gain* dt [PU]	
s2_acr_deadband	int16_t	Q13	Current	Current integral part deadband	
s2_d0_div_pm	int16_t	Q13	-	Static friction coefficient/(number of pole pairs × BEMF constant)	
s2_d1_div_p2m	int16_t	Q19	-	Kinetic friction coefficient/((number of pole pairs) ^2 × BEMF constant)	
s2_aa_kp	int16_t	Q21	-	Advanced angle control PI proportional gain	
s2_aa_kidt	int16_t	Q21	-	Advanced angle control PI ki * dt	
s2_aa_lpf_in_k	int16_t	Q15	-	Advanced angle control LPF numerator	

Table 3-57 List of variables “r_mtr_ctrl_gain.h / Structure : st_design_parameter_t”

Variable	Type	Qn	PU	Content	Remarks
u2_pp	uint16_t	Q0	-	Number of pole pairs	
f4_r	float	Q0	-	Resistance [PU]	
f4 _ld	float	Q0	-	d-axis inductance [PU]	
f4 _lq	float	Q0	-	q-axis inductance [PU]	
f4 _m	float	Q0	-	BEMF constant [PU]	
f4 _j	float	Q0	-	Inertia [PU]	
f4_speed_omega	float	Q0	-	Speed PI controller natural frequency [Hz]	
f4_asr_zeta	float	Q0	-	Speed PI controller damping factor	
f4_dt	float	Q0	-	Control period for current [s]	
f4_pu_sf_asr_kp	float	Q0	-	proportional gain scale factor of speed controller	
f4_pu_sf_asr_kidt	float	Q0	-	integral gain scale factor of speed controller	
u1_q_voltage	uint8_t	Q0	-	Q-format of voltage	
u1_q_afreq	uint8_t	Q0	-	Q-format of angular frequency	
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	
f4_asr_ki_aug	float	Q0	-	Augmentation rate for integral part of ASR	
f4_dt_speed	float	Q0	-	Control period for speed loop [sec]	
f4_acr_nf_hz	float	Q0	-	Current PI control natural frequency [Hz]	
f4_asr_nf_hz	float	Q0	-	Speed PI control natural frequency [Hz]	
f4_acr_deadband_lsb	float	Q0	-	Deadband of current PI calculation [LSB]	
f4_asr_deadband_lsb	float	Q0	-	Deadband of speed PI calculation [LSB]	
f4_d0_div_pm	float	Q0	-	Static friction coefficient/(number of pole pairs × BEMF constant) [kg･m^2/(V･s^3/rad)]	
f4_d1_div_p2m	float	Q0	-	Kinetic friction coefficient/((number of pole pairs)^2 × BEMF constant) [kg･m^2/(V･s^2)]	
f4_pu_sf_afreq	float	Q0	-	Frequency scale factor	
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_deadband	uint8_t	Q0	-	Q-format of current PI deadband	
u1_q_asr_deadband	uint8_t	Q0	-	Q-format of speed PI deadband	
f4_aa_pi_nf_hz	float	Q0	-	Advanced angle controller natural frequency	
f4_aa_speed_err_lpf_cof_hz	float	Q0	-	Speed error LPF cutoff frequency	
u1_q_aa_lpf_k	uint8_t	Q0	-	Q-format of LPF numerator for Advanced Angle Control	
u1_q_aa_kp	uint8_t	Q0	-	Q-format of AA PI proportional gain	
u1_q_aa_kidt	uint8_t	Q0	-	Q-format of AA PI ki * dt	

Table 3-58 List of variables “r_mtr_ipd.h / Structure : st_mtr_ipd_t”

Variable	Type	Qn	PU	Content	Remarks
u1_state_ipd	uint8_t	Q0	-	State of initial position detection	[IPD]
u1_judge_sal	uint8_t	Q0	-	Result of salient jdgement	
u1_flag_cmp3_intr	uint8_t	Q0	-	Flag for CMP3 interrupt	
u2_sal_angle_current	uint16_t	Q0	-	Threshold current of angle detection for salient rotor	
u1_sal_angle_cnt	uint8_t	Q0	-	Number of times salient rotor angle detection is measured	
u2_sal_polarity_current	uint16_t	Q0	-	Threshold current of polarity detection of angle detection for salient rotor	
u1_sal_polarity_cnt	uint8_t	Q0	-	Number of times salient rotor polarity detection is measured	
u1_sal_polarity	uint8_t	Q0	-	Result of polarity detection for salient rotor	
u2_non_sal_current	uint16_t	Q0	-	Threshold current of angle detection for non-salient rotor	
u1_non_sal_cnt	uint8_t	Q0	-	Number of times non-salient rotor angle detection is measured	
u1_init_position	uint8_t	Q0	-	Initial position detection judgment result	
u2_temp_trx_cnt	uint16_t	Q0	-	TRX count value acquisition	
u2_sal_angle_discharge	uint8_t	Q0	-	Discharge period of angle detection for salient rotor	
u2_sal_polarity_discharge	uint8_t	Q0	-	Discharge period of polarity detection for salient rotor	
u2_non_sal_discharge	uint8_t	Q0	-	Discharge period of angle detection for non-salient rotor	
u4_sal_angle_trx_sum[3]	uint32_t	Q0	-	TRX count value of angle detection for salient rotor	
u4_sal_check_trx_sum	uint32_t	Q0	-	TRX count value during polarity detection	
u4_sal_polarity_trx_sum[2]	uint32_t	Q0	-	TRX count value of polarity detection for salient rotor	
u4_non_sal_trx_sum[6];	uint32_t	Q0	-	TRX count value of angle detection for non-salient rotor	
u4_sal_angle_trx_diff	uint32_t	Q0	-	TRX count value differential of angle detection for salient rotor	
u4_sal_polarity_trx_diff	uint32_t	Q0	-	TRX count value differential of polarity detection for salient rotor	
u4_non_sal_trx_diff	uint32_t	Q0	-	TRX count value differential in angle detection for non-salient rotor	
u4_sal_angle_th	uint32_t	Q0	-	TRX count value differential threshold of angle detection for salient rotor	
u4_sal_angle_per	uint32_t	Q0	-	Percentage of TRX count value differential threshold of angle detection for salient rotor	
u4_sal_polarity_th	uint32_t	Q0	-	TRX count value differential threshold of polarity detection for salient rotor	
u4_sal_polarity_per	uint32_t	Q0	-	Percentage of TRX count value differential threshold of polarity detection for salient rotor	
u4_non_sal_th	uint32_t	Q0	-	TRX count value differential threshold of angle detection for non-salient rotor	
u4_non_sal_per	uint32_t	Q0	-	Percentage of TRX count value differential threshold of angle detection for salient rotor	

Table 3-59 List of variables “r_mtr_120.h / Structure : st_mtr_ol2cl_t”

Variable	Type	Qn	PU	Content	Remarks
u1_flag_change	uint8_t	Q0	-	Flag to allow to transition to PI control	[Sensorless mode]
u1_flag_change_speed	uint8_t	Q0	-	Flag for exceeding change speed	
u1_zc_flag	uint8_t	Q0	-	Flag for zero-crossing detection avoiding commutation	
u1_rotor_pos	uint8_t	Q0	-	Phase of rotor position	
u2_zc_cnt	uint16_t	Q0	-	Counter for zero-crossing detection	
s2_change_speed_rad	int16_t	Q14	Angular frequenc y	Speed allowed to transition to PI control [PU]	
s2_ref_speed_rad_buf	int16_t	Q14	Angular frequenc y	Buff for reference rotational speed [PU]	
s2_ramp_speed_rad	int16_t	Q14	Angular frequenc y	Acceleration at transition to PI control [PU]	

Table 3-60 List of variables “r_mtr_aa_ctrl.h / Structure : st_mtr_aa_pi_t”

Variable	Type	Qn	PU	Content	Remarks
s2_kp	int16_t	Q21	-	Proportional gain	[Advanced angle control mode]
s2_kidt	int16_t	Q21	-	Integral gain multiply sampling period	
s2_intg	int16_t	Q12	-	Stored integral part	
s2_ilimit	int16_t	Q12	-	Integral part saturation limit (symmetric)	
s2_deadband	int16_t	Q13	Voltage	Dead band of integral part	
s2_pre_err	int16_t	Q13	Voltage	Previous error between reference voltage and voltage limit	
s2_pi_out_pre	int16_t	Q12	Angle	Previous output of PI calculation	

Table 3-61 List of variables “r_mtr_aa_ctrl.h / Structure : st_mtr_aa_lpf1_t”

Variable	Type	Qn	PU	Content	Remarks
s2_in_k	int16_t	Q15	-	LPF input gain	[Advanced angle control mode]
s2_out_k	int16_t	Q15	-	LPF previous output gain	
s2_pre_out	int16_t	Q14	Angular frequency	Previous output value	

Table 3-62 List of variables “r_mtr_aa_ctrl.h / Structure : st_mtr_aa_t”

Variable	Type	Qn	PU	Content	Remarks
u1_pi_on_flag	uint16_t	-	-	PI ON flag	[Advanced angle control mode]
u1_hys_flag	uint16_t	-	-	Hysteresis flag	
u1_q_aa_kp	uint16_t	-	-	Q-format of AA PI proportional gain	
u1_q_aa_kidt	uint16_t	-	-	Q-format of AA PI ki * dt	
u1_q_aa_lpf_co	uint8_t	-	-	Q-format of speed error LPF coefficient	
u1_q_bemf_const	uint8_t	-	-	Q-format of BEMF constant	
u1_control_mode	uint8_t	-	-	Control mode	
u2_fp_sf_voltage	uint16_t	-	-	Voltage fixed point scale factor	
s2_m	int16_t	Q15	BEMF constant	BEMF const for calculation	
s2_speed_err_th_rad	int16_t	Q14	Angular frequency	Speed error threshold	
s2_speed_err_rad	int16_t	Q14	Angular frequency	Speed error	
s2_speed_err_rad_lpf	int16_t	Q14	Angular frequency	Speed error LPF	
s2_angle_offset	int16_t	Q12	Angle	Angle offset	
s2_angle_offset_pre	int16_t	Q12	Angle	Previous value of angle offset	
s2_angle_max	int16_t	Q12	Angle	Maximum value of angle offset	
s2_angle_min	int16_t	Q12	Angle	Minimum value of angle offset	
s2_ramp_limit_angle	int16_t	Q12	Angle	Limit of rate of angle offset change	
s2_ramp_limit_voltage	int16_t	Q13	Voltage	Limit of rate of neutral voltage offset change	
s2_vmin	int16_t	Q13	Voltage	Minimum value of neutral voltage offset	
s2_vmax	int16_t	Q13	Voltage	Maximum value of neutral voltage offset	
s2_vlim	int16_t	Q13	Voltage	Voltage limit	
s2_vn_offset	int16_t	Q13	Voltage	Neutral voltage offset	
s2_vn_offset_pre	int16_t	Q13	Voltage	Previous value of neutral voltage offset	
s2_vn_offset_pre_ad	int16_t	-	-	Previous value of neutral voltage offset [bit]	
s2_aa_limit_current_ref	int16_t	Q13	Current	Limit of current value	
st_pi	st_mtr_aa_pi_t	-	-	PI controller parameter structure	
st_lpf	st_mtr_aa_lpf1_t	-	-	1st order LPF structure	
st_angle	st_sincos12	-	-	Angle 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 3-63 List of Macro definitions “r_mtr_config.h”

Macro	Definition value	Description	Remarks
IP_MCEK_1SHUNT	-	Select inverter board	
MP_TG_55L_KA	-	Select motor parameters	
CP_TG_55L_KA	-	Select control parameters	
SENSORLESS_CURRENT	0	Sensroless current control mode	
SENSORLESS_VOLTAGE	1	Sensorless voltage control mode	
HALL_CURRENT	2	Hall effect sensor-based current control mode	
HALL_VOLTAGE	3	Hall effect sensor-based 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	
COMPLEMENTARY	1	Complementary PWM	
MTRCONF_PWM_MODE	NON_COMPLEMENTARY COMPLEMENTARY	Selection of PWM mode	Default setting COMPLEMENTARY
USE_AA	0/1	Advanced angle control mode	Default setting 1
PS_IPD_SAL	0	Initial position detection for salient motor	
PS_IPD_NON_SAL	1	Initial position detection for non-salient motor	
PS_IPD_UNKNOWN	2	Initial position detection without saliency information	
PS_DRAW_IN	3	Draw-in	
PS_METHOD	PS_IPD_SAL PS_IPD_NON_SAL PS_IPD_UNKNOWN PS_DRAW_IN	Preparation method for start-up	Default setting PS_IPD_SAL
HALL	0	Hall effect sensor	
LESS	1	Sensorless	
MTRCONF_SENSOR_MODE	HALL LESS	Selection of sensor mode	
VOLTAGE	0	Voltage control	
CURRENT	1	Current control	
MTRCONF_CONTROL_MODE	VOLTAGE CURRENT	Selection of control mode	Default setting CURRENT

Table 3-64 List of Macro definitions “r_mtr_motor_parameter.h”

Macro	Definition value	Description	Remarks
MP_POLE_PAIRS	2	Number of pole pairs	
MP_RESISTANCE	9.125f	Resistance [Ω]	
MP_D_INDUCTANCE	0.003844f	d-axis inductance [H/rad]	
MP_Q_INDUCTANCE	0.004315f	q-axis inductance [H/rad]	
MP_BEMF_CONSTANT	0.02144f	BEMF constant [V･s/rad]	
MP_ROTOR_INERTIA	0.00000205f	Inertia [kg･m^2/(rad^2)]	
MP_FRICTION_0TH_ORDER	0.002748f	Static friction coefficient [kg･m^2/(rad･s^2)]	
MP_FRICTION_1ST_ORDER	0.000001873f	Kinetic friction coefficient [kg･m^2/(rad^2･s)]	
MP_RATED_CURRENT	0.42f	Nominal current [A]	
MP_RATED_SPEED	2650	Rated speed [rpm]	
STAR	0	Star wiring	Default
DELTA	1	Delta wiring	
MP_MOTOR_WIRE_CONNECTION	STAR DELTA	Selection of motor wiring connection	Star / Delta

Table 3-65 List of Macro definitions “r_mtr_inverter_parameter.h”

Macro	Definition value	Description	Remarks
IP_DEADTIME	1.0f	Deadtime	
IP_CURRENT_RANGE	10	Current scaling range [A]	
IP_VDC_RANGE	111	Voltage scaling range [V]	
IP_INPUT_V	24	Input voltage [V]	
IP_CURRENT_LIMIT	4.5f	Current limit value [A]	
IP_OVERVOLTAGE_LIMIT	28	Overvoltage limit [V]	
IP_UNDERVOLTAGE_LIMIT	12	Undervoltage limit [V]	
IP_DC_SHUNT_RESISTANCE	0.01f	DC Link Shunt Resistance [ohm]	
IP_DC_AMPLIFICATION_GAIN	50	DC Link Current Amplification Gain	
IP_BSC_CHARGE_TIME	150	Period of charging bootstrap capacitor	
IP_CHARGE_CAP_WAIT_CNT	350	Charge time of DC capacitor	

Table 3-66 List of Macro definitions “r_mtr_control_parameter.h”[1/2]

Macro	Definition value	Description	Remarks
CP_PWM_TIMER_FREQ	96.0f	PWM timer frequency [MHz]	
CP_TRG_TIMER_FREQ	48.0f	TRG timer frequency [MHz]	
CP_CARRIER_FREQ	20.0f	PWM carrier frequency [kHz]	
CP_TRX_TIMER_FREQ	96.0f	TRX timer frequency [MHz]	
CP_TAU1_TIMER_FREQ	187.5f	TAU1 timer frequency [kHz]	
CP_SPEED_CTRL_PERIOD	0.001f	Speed control cycle [sec]	
CP_TRJ_TIMER_FREQ	48.0f	TRJ timer frequency [MHz]	[IPD]
CP_AD_TRG_DEFAULT_CNT	200.0f	A/D trigger count default value	
CP_OFFSET_CALC_CNT	64	Number of measurements of DC link current offset	
CP_OFFSET_CALC_ST_WAIT_CNT	100	Offset calculation stable waiting time	
CP_AD_TRG_LIMIT_CNT	100.0f	A/D trigger count lower limit value	
CP_AD_TRG_DUTY_THRESHOLD_CNT	1800.0f	Duty threshold value for A/D trigger count setting	
CP_AD_TRG_ADJUST_UP_CNT	300.0f	A/D trigger adjustment value for ascending BEMF	
CP_AD_TRG_ADJUST_DOWN_CNT	300.0f	A/D trigger adjustment value for descending BEMF	
CP_AD_TRG_DIFF_CNT	150.0f	Difference in A/D trigger count between previous and current carrier	
CP_MAX_SPEED_RPM	4300	Maximum limit of command rotational speed [rpm]	Mechanical angle
CP _SPEED_LIMIT_RPM	5590	Maximum limit of estimated rotational speed [rpm]	Mechanical angle
CP_MIN_SPEED_RPM	265 [Hall effect sensor mode] 530 [Sensorless mode]	Minimum limit of command rotational speed [rpm]	Mechanical angle
CP_OC_LIMIT	1.47f	Overcurrent limit [A]	
CP_RAMP_LIMIT_SPEED_RPM_LESS	1.677845f	Limit of acceleration [rpm/ms]	Mechanical angle
CP_OL2CL_SPEED_RPM	530	Speed allowed to transition to PI control [rpm]	Mechanical angle
CP_OL2CL_SPEED_MARGIN	53	Rotational speed margin for transition of control mode [rpm]	
CP_SKIP_SPIKE_NOSE	8	Carrier counts for avoiding spike noise	
CP_LIMIT_V	4 [Sensorless current control mode] 4.5f [Sensorless voltage control mode]	Lower limit of voltage [V]	
CP_OL2CL_SPEED_RAMP_RPM	2	Acceleration at transition to PI control [rpm/ms]	Mechanical angle
CP_OL2CL_ZC_CNT	3	Number of zero-crossings for transition to PI control	
CP_ACR_NF_HZ	300	Current PI control natural frequency [Hz]	
CP_ASR_NF_HZ	5.04f	Speed PI control natural frequency [Hz]	
CP_ACR_DEADBAND_LSB	0.5f	Deadband for current PI control [LSB]	
CP_ASR_DEADBAND_LSB	0.5f	Deadband for speed PI control [LSB]	
CP_ASR_KI_AUG	8	Augmentation rate for integral part of ASR	

Table 3-67 List of Macro definitions “r_mtr_control_parameter.h”[2/2]

Macro	Definition value	Description	Remarks
CP_RAMP_LIMIT_CURRENT	0.008311f	Limit for variation of current [A/ms]	
CP_DRAW_IN_REF_IDC	0.42f	Reference current at draw-in[V]	
CP_OL_REF_IDC	0.1f	Reference current at open-loop [V]	
CP_ASR_OMETA_HZ	12	Frequency for speed PI control [Hz]	
CP_ASR_ZETA	1	Damping factor for speed PI control	
CP_RAMP_LIMIT_V_LESS	0.13f	Limit for variation of voltage [V]	
CP_DRAW_IN_REF_V	7.67f	Reference voltage at draw-in[V]	
CP_OL_REF_V	3.5f	Reference voltage at open-loop [V]	
CP_RAMP_LIMIT_SPEED_RPM_HALL	10.067065 f	Limit of acceleration [rpm/ms]	
CP_RAMP_LIMIT_V_HALL	0.29f	Limit for variation of voltage	
CP_START_REF_V	2.6f	Reference voltage for start-up [V]	
CP_START_REF_IDC	0.042f	Reference current for start-up[A]	
CP_HALL2OLREV_SPEED_RPM	530	Transition speed to PI control when direction of rotation is reversed [rpm]	
CP_SAL_ANGLE_CURRENT	0.25f	Threshold current of angle detection for salient rotor	
CP_SAL_ANGLE_TRX_THRESHOLD	1012	TRX count value differential of angle detection for salient rotor	
CP_SAL_ANGLE_DISCHARGE	601	Discharge period of angle detection for salient rotor	
CP_SAL_POLARITY_CURRENT	0.42f	Threshold current of polarity detection for salient rotor	
CP_SAL_POLARITY_TRX_THRESHOLD	1019	TRX count value differential of polarity detection for salient rotor	
CP_SAL_POLARITY_DISCHARGE	745	Discharge period of polarity detection for salient rotor	
CP_NON_SAL_CURRENT	0.4f	Threshold current of angle detection for non-salient rotor	
CP_NON_SAL_TRX_THRESHOLD	852	TRX count value differential of angle detection for non-salient rotor	
CP_NON_SAL_TRX_DISCHARGE	689	Discharge period of angle detection for non-salient rotor	
CP_IPD_NOISE_AVOID_CNT	97	Noise avoidance count at IPD	
CP_AA_PI_NF_HZ	10	PI natural frequency for Advanced Angle Control	
CP_AA_SPEED_ERR_THRESHOLD_RPM	50	Speed error threshold for Advanced Angle Control	
CP_AA_SPEED_ERR_LPF_COF_HZ	50	Speed error LPF for Advanced Angle Control	
CP_AA_RAMP_LIMIT_V	0.3f	Limit of rate of neutral voltage change for Advanced Angle Control [V/ms]	
CP_AA_RAMP_LIMIT_ANGLE	0.001f	Limit of rate of angle offset change for Advanced Angle Control [rad/ms]	
CP_AA_VLIM	17.52f	Voltage limit for Advanced Angle Control[V]	
CP_AA_CURRENT_LIMIT_RATIO	0.2	Current limit ration for Advanced Angle Control	

Table 3-68 List of Macro definitions “r_mtr_scaling_parameter.h”[1/2]

Macro	Definition value	Qn	PU	Description	Remarks
FP_SF_VOLTAGE	37888	-	-	Voltage PU conversion value ((IP_VDC_RANGE*PU_SF_VOLTAGE) × (1<<MTR_Q_VOLTAGE))	
FP_SF_CURRENT	195047			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_FREQ	1.0f / PU_BASE_FREQ_Hz	-	-	Frequency scale factor [PU/Hz]	
PU_SF_AFREQ	1.0f / PU_BASE_ANGLE_Rad	-	-	Angular frequency scale factor [PU/(rad/s)]	
PU_SF_TIME	PU_SF_ANGLE / PU_SF_AFREQ	-	-	Time scale factor [PU/s]	
PU_SF_RES	PU_SF_VOLTAGE / PU_SF_CURRENT	-	-	Impedeance (resistance) scale factor [PU/Ω]	
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	Current mode： PU_SF_CURRENT / PU_SF_AFREQ Voltage mode： PU_SF_VOLTAGE / PU_SF_AFREQ	-	-	Speed PI proportional gain scale factor	
PU_SF_ASR_KIDT	Current mode： PU_SF_CURRENT / PU_SF_AFREQ Voltage mode： PU_SF_VOLTAGE / PU_SF_AFREQ	-	-	Speed PI integral gain scale factor	

Table 3-69 List of Macro definitions “r_mtr_scaling_parameter.h”[2/2]

Macro	Definition value	Description	Remarks
MTR_Q_ANGLE	12	Q-formtat of angle	
MTR_Q_CURRENT	13	Q-format of current	
MTR_Q_VOLTAGE	13	Q-format of voltage	
MTR_Q_AFREQ	14	Q-format of angular frequency	
MTR_Q_CTRL_TIME_SPEED	15	Q-format of speed control cycle	
MTR_Q_RESISTANCE	17	Q-format of resistance	
MTR_Q_INDUCTANCE	18	Q-format of inductance	
MTR_Q_BEMF_CONST	15	Q-format of induced voltage constant	
MTR_Q_INERTIA	9	Q-format of inertia	
MTR_A_D1_DIV_P2M	19	Q-format of kinetic friction coefficient /((number of pole pairs)^2 x BEMF constant)	
MTR_Q_ACR_KP	17	Q-format of current PI proportional gain	
MTR_Q_ACR_KIDT	21	A-format of current PI integral gain x control period	
MTR_Q_ASR_KP	Current control mode：14 Voltage control mode：13	Q-format of speed PI proportional gain	
MTR_Q_ASR_KIDT	Current control mode：21 Voltage control mode：15	Q-format of speed PI integral gain x control period	
MTR_Q_CURRENT_LPF_CO	15	Q-format of current LPF gain	
RSFT_AFREQ_BEMF_CONST_2VOLTAGE	MTR_Q_BEMF_CONST + MTR_Q_AFREQ – MTR_Q_VOLTAGE	Right shift, (speed * induced voltage) to voltage	
MTR_Q_AA_KP	21	Q-format of proportional gain for Advanced Angle Control	
MTR_Q_AA_KIDT	21	Q-format of integral gain x control period for Advanced Angle Control	
MTR_Q_AA_LPF_CO	15	Q-format of speed error LPF coefficient for Advanced Angle Control	
MTR_Q_SIN_COS_DSP	14	Q-format of trigonometric function of DSP function	

Table 3-70 List of Macro definitions “main.h”

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

Table 3-71 List of Macro definitions “ICS_define.h”

Macro	Definition value	Qn	PU	Description	Remarks
RL78	-	-	-	CPU definition	

Table 3-72 List of Macro definitions “r_mtr_ics.h”

Macro	Definition value	Qn	PU	Description	Remarks
TS_100US	-	-	-	ICS communication speed	
MTR_ICS_DECIMATION	5	-	-	Decimation for ICS communication	
ICS_BRR	23	-	-	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 3-73 List of Macro definitions “r_mtr_ctrl_rl78g24.h”[1/2]

Macro	Definition value	Description	Remarks
USE_PWMOPA	MTR_SET	Selection of overcurrent usage by PWMOPA	
MTR_CARRIER_FREQ	CP_CARRIER_FREQ	Frequency of carrier [kHz]	
MTR_TAU1_TIMER_FREQ	CP_TAU1_TIMER_FREQ	Frequency of TAU1 [kHz]	
MTR_PWM_TIMER_FREQ	CP_PWM_TIMER_FREQ	Frequency of PWM timer [MHz]	
MTR_TRG_TIMER_FREQ	CP_TRG_TIMER_FREQ	Frequency of TRG [MHz]	
MTR_TRJ_TIMER_FREQ	CP_TRJ_TIMER_FREQ	Frequency of TRJ [MHz]	[Initial Position Detection]
MTR_SPEED_CTRL_PERIOD	CP_SPEED_CTRL_PERIOD	Control period for speed loop [sec]	
MTR_CTRL_PERIOD	(1.0f / (MTR_CARRIER_FREQ * 1000))	Control period fo current loop [sec]	
MTR_WINDOW_DELAY_OFFSET	3	Delay count for timer window	
MTR_WINDOW_ADJUST	0	Adjust count for timer window	
MTR_OL2CL_BEMF_THRESH	3	Threshold for measuring induced voltage	
MTR_DEADTIME	IP_DEADTIME	Dead-time [μs]	
MTR_NOT_CARRIER_SET	(MTR_PWM_TIMER_FREQ*1000/MTR_CARRIER_FREQ/2) -2	Resister counts of carrier without deadtime	
MTR_CARRIER_SET	(MTR_PWM_TIMER_FREQ*1000/MTR_CARRIER_FREQ/2) -2 [Non-complementary PWM] / MTR_NOT_CARRIER_SET+ MTR_DEADTIME_SET [Complementary PWM]	Carrier count of PWM	
MTR_MAX_WINDOW_WIDTH	(MTR_NOT_CARRIER_SET * 0.9)	Maximum count for timer window	
MTR_MAX_DUYT_SET	MTR_NOT_CARRIER_SET – 2	Maximum count for duty setting	
MTR_HALF_CARRIER_SET	MTR_CARRIER_SET/2	Resister counts of half carrier	
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_AD_TRG_DEFAULT_CNT	CP_AD_TRG_DEFAULT_CNT	A/D trigger default count	
MTR_AD_TRG_LIMIT_CNT	CP_AD_TRG_LIMIT_CNT	A/D trigger minimum count	
MTR_AD_TRG_DUTY_THRESHOLD_CNT	CP_AD_TRG_DUTY_THRESHOLD_CNT	PWM duty threshold	
MTR_AD_TRG_ADJUST_UP_CNT	CP_AD_TRG_ADJUST_UP_CNT	Adjustment count of A/D trigger for ascending non-energized phase voltage	
MTR_AD_TRG_ADJUST_DOWN_CNT	CP_AD_TRG_ADJUST_DOWN_CNT	A/D trigger adjustment count ofr descending BEMF	
MTR_AD_TRG_DIFF_CNT	CP_AD_TRG_DIFF_CNT	Count difference of A/D between previous carrier and current carrie	
MTR_PATTERN_ERROR	0	Voltage pattern	
MTR_UP_PWM_VN_ON	1		
MTR_UP_PWM_WN_ON	2		
MTR_VP_PWM_UN_ON	3		
MTR_VP_PWM_WN_ON	4		
MTR_WP_PWM_UN_ON	5		
MTR_WP_PWM_VN_ON	6		
MTR_UP_ON_VN_PWM	7		
MTR_UP_ON_WN_PWM	8		
MTR_VP_ON_UN_PWM	9		

Table 3-74 List of Macro definitions "r_mtr_ctrl_rl78g24.h"[2/2]

Macro	Definition value	Description	Remarks
MTR_VP_ON_WN_PWM	10	Voltage pattern	
MTR_WP_ON_UN_PWM	11		
MTR_WP_ON_VN_PWM	12		
MTR_U_PWM_VN_ON	13		
MTR_U_PWM_WN_ON	14		[Complementary PWM]
MTR_V_PWM_UN_ON	15		
MTR_V_PWM_WN_ON	16		
MTR_W_PWM_UN_ON	17		
MTR_W_PWM_VN_ON	18		
MTR_UP_ON_V_PWM	19		
MTR_UP_ON_W_PWM	20		
MTR_VP_ON_U_PWM	21		
MTR_VP_ON_W_PWM	22		
MTR_WP_ON_U_PWM	23		
MTR_WP_ON_V_PWM	24		
ERROR_NONE	0x00	Default	
ERROR_CHANGE_CLK_TIMEOUT	0x01	Timeout error of change of resister for clock	
ERROR_CHARGE_CAP_TIMEOUT	0x02	Timeout error of capacitor charge	
ERROR_DRIVING	0x03	ERROR at driving mode	
MTR_OC_DETECT_REF	(uint16_t)((IP_DC_AMPLIFICATION_GAIN * IP_DC_SHUNT_RESISTANCE * IP_CURRENT_LIMIT) * (1024 / 5))	Overcurrent detection reference value	
MTR_OC_DETECT_OFSET	(uint16_t)(2.5f*(1024/5))	Overcurrent detection offset value	

Table 3-75 List of Macro definitions "r_mtr_common.h"

Macro	Definition value	Description	Remarks
MTR_TWOPI	2*3.14159265359f	2π	
MTR_TWOPI_60	MTR_TWOPI/60	2π/60	
MTR_SQRT_3	1.7320508f	√3	
MTR_SQRT_2_3	0.8164966f	√(2/3)	
MTR_CW	1	CW	
MTR_CCW	-1	CCW	
MTR_CLR	0	Clear flag	
MTR_SET	1	Set flag	
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_10BIT	10	10bit ADC	
MTR_ADC_12BIT	12	12bit ADC	

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

Macro	Definition value	Qn	PU	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 * 0.0f	-	-	Minimum output voltage	
MTR_MAX_DRIVE_V	FIX_fromfloat(MTR_US_MAX_DRIVE_V * PU_SF_VOLTAGE, MTR_Q_VOLTAGE)	Q13	Voltage	Maximum output voltage	
MTR_MIN_DRIVE_V	FIX_fromfloat(MTR_US_MIN_DRIVE_V * PU_SF_VOLTAGE, MTR_Q_VOLTAGE)	Q13	Voltage	Minimum output voltage	
MTR_US_SPEED_CALC_BASE	MTR_TAU1_TIMER_FREQ* 1000 *MTR_TWOPI	-	-	Calculation parameter to convert the timer counter to rotational speed	
MTR_SPEED_CALC_BASE	FIX32_fromfloat(MTR_US_SPEED_CALC_BASE * PU_SF_AFREQ, MTR_Q_AFREQ)	Q14	Angular frequency	Calculation parameter to convert the timer counter to rotational speed	
MTR_OL_SPEED_CALC_BASE	MTR_CARRIER_FREQ * 1000 * MTR_TWOPI / MTR_PATTERN_NUM	-	-	Calculation parameter to convert rotational speed to timer counter at open-loop drive	
MTR_PU_Q_OL_SPEED_CALC_BASE	FIX32_fromfloat(MTR_US_OL_SPEED_CALC_BASE * PU_SF_AFREQ,MTR_Q_AFREQ)	Q14	Angular frequency	Calculation parameter to convert rotational speed to timer counter at open-loop drive	
MTR_ASR_DEADBAND	0	-	-	Minimum value of deadband for integral term	
MTR_SPEED_CALC_BASE_1ST	MTR_SPEED_CALC_BASE/6	Q14	Angular frequency	Calculation parameter to convert the timer counter to rotational speed at first speed calculation	
MTR_SPEED_CALC_BASE_2ND	MTR_SPEED_CALC_BASE/3	Q14	Angular frequency	Calculation parameter to convert the timer counter to rotational speed at second speed calculation	
MTR_SPEED_CALC_BASE_3RD	MTR_SPEED_CALC_BASE/2	Q14	Angular frequency	Calculation parameter to convert the timer counter to rotational speed at third speed calculation	
MTR_SPEED_CALC_BASE_4TH	MTR_SPEED_CALC_BASE*2/3	Q14	Angular frequency	Calculation parameter to convert the timer counter to rotational speed at fourth speed calculation	
MTR_SPEED_CALC_BASE_5TH	MTR_SPEED_CALC_BASE*5/6	Q14	Angular frequency	Calculation parameter to convert the timer counter to rotational speed at fifth speed calculation	
MTR_BIT_SFT_NUM_FOR_DUTY_CALC	15	-	-	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	Q13	Current	Speed PI output limit value	[Current control mode]
MTR_I_LIMIT_CURRENT	MP_RATED_CURRENT * MTR_SQRT_3	Q13	Current	Limit value for speed PI integral term output	
MTR_I_LIMIT_VQ	IP_INPUT_V * 0.9f	Q13	Current	Current PI Integral term limit	
MTR_MIN_IDC_REF	0	Q13	Current	Minimum value for reference current	

Table 3-77 List of Macro definitions “r_mtr_statemachine.h”

Macro	Definition value	Qn	PU	Description	Remarks
MTR_MODE_INIT	0x00	Q0	-	Initialization mode	
MTR_MODE_DRIVE	0x01	Q0	-	Drive mode	
MTR_MODE_STOP	0x02	Q0	-	Stop mode	
MTR_SIZE_STATE	3	Q0	-	Number of states	
MTR_EVENT_STOP	0x00	Q0	-	Stop event	
MTR_EVENT_DRIVE	0x01	Q0	-	Run event	
MTR_EVENT_ERROR	0x02	Q0	-	Error event	
MTR_EVENT_RESET	0x03	Q0	-	Reset event	
MTR_SIZE_EVENT	4	Q0	-	Number of events	

Table 3-78 List of Macro definitions “r_ol2cl_ctrl.h”

Macro	Definition value	Qn	Pu	Description	Remarks
MTR_PHASE_LEAD	0	-	-	Leading phase of rotor	[Sensorless mode]
MTR_PHASE_LAG	1	-	-	Lagging phase of rotor	
MTR_1ST_ZC_CNT	1	-	-	First zerocross detected during transition	

Table 3-79 List of Macro definitions “r_mtr_aa_ctrl.h”

Macro	Definition value	Qn	PU	Description	Remarks
MTR_60_DEGREE	FIX_fromfloat((1.0/3.0) * PU_SF_ANGLE, MTR_Q_ANGLE)	Q12	Angle	π/3	[Advanced angle control]
MTR_AA_M	FIX_fromfloat((MP_BEMF_CONSTANT * MTR_SQRT_2_3 * PU_SF_BEMF_CONST), MTR_Q_BEMF_CONST)	Q15	BEMF constant	BEMF const for calculation	
MTR_VOLTAGE_MODE	0	-	-	Voltage control mode	
MTR_CURRENT_MODE	1	-	-	Current control mode	

Table 3-80 List of Macro definitions “r_mtr_120.h”[1/2]

Macro	Definition value	Qn	Pu	Description	Remarks
MTR_TIMEOUT_CNT	(uint16_t)(1.0f/((float)MP_POLE_PAIRS*CP_MIN_SPEED_RPM/60)*1000*2)	-	-	Counts for timeout	
MTR_HALL2OL_REV_SPEED_RAD	FIX_fromfloat(CP_HALL2OL_REV_SPEED_RPM * PU_SF_RPM_RAD, MTR_Q_AFREQ)	Q14	Angular frequency	Transition speed when rotor reverses	[Hall effect sensor mode]
MTR_DRAW_IN_1ST_PATTERN	1	-	-	Voltage pattern at first draw-in	
MTR_DRAW_IN_2ND_PATTERN	2	-	-	Voltage pattern at second draw-in	
MTR_PATTERN_CW_V_U	2	-	-	Voltage patten at CW rotation	
MTR_PATTERN_CW_W_U	3	-	-		
MTR_PATTERN_CW_W_V	1	-	-		
MTR_PATTERN_CW_U_V	5	-	-		
MTR_PATTERN_CW_U_W	4	-	-		
MTR_PATTERN_CW_V_W	6	-	-		
MTR_PATTERN_CCW_V_U	3	-	-	Voltage pattern at CCW rotation	
MTR_PATTERN_CCW_V_W	2	-	-		
MTR_PATTERN_CCW_U_W	6	-	-		
MTR_PATTERN_CCW_U_V	4	-	-		
MTR_PATTERN_CCW_W_V	5	-	-		
MTR_PATTERN_CCW_W_U	1	-	-		
MTR_PATTERN_NUM	6	-	-	Number of voltage patterns	
MTR_CMP3_IPD	0	-	-	Initial position detection	
MTR_CMP3_OC	1	-	-	Overcurrent	
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 control mode	
MTR_ERROR_BEMF_TIMEOUT	0x0010	-	-	Timeout error for sensorless control mode	
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_OVER_CURRENT_SW	0x0100	-	-	Software over current error	
MTR_ERROR_IPD_TRX_OVERFLOW	0x0200	-	-	TRX overflow error during initial position detection	
MTR_ERROR_UNKNOWN	0xffff	-	-	Undefined error	
MTR_DRAW_IN_NONE	0x0001	-	-	No operation	
MTR_DRAW_IN_1ST	1	-	-	First draw-in	
MTR_DRAW_IN_2ND	2	-	-	Second draw-in	
MTR_DRAW_IN_FINISH	3	-	-	Draw-in finished	
MTR_SPEED_ZERO_CONST	0	-	-	Reference speed 0 mode	
MTR_SPEED_MANUAL	1	-	-	Reference speed manual input mode	
MTR_V_ZERO_CONST	0	-	-	Reference voltage 0 mode	[Voltage control mode]
MTR_V_MANUAL	1	-	-	Reference voltage manual input mode	
MTR_V_PI_OUTPUT	2	-	-	Reference voltage PI output mode	
MTR_IDC_ZERO_CONST	0	-	-	Reference current 0 mode	[Current control mode]
MTR_IDC_MANUAL	1	-	-	Reference current manual input mode	
MTR_IDC_SPEED_PI_OUTPUT	2	-	-	Reference current PI output mode	
MTR_CURRENT_OFFSET_NONE	0	-	-	Default state for measurement DC link current offset	
MTR_CURRENT_OFFSET_MEASURE	1	-	-	Measuring DC link current offset	

Table 3-81 List of Macro definitions “r_mtr_120.h”[2/2]

Macro	Definition value	Qn	Pu	Description	Remarks
MTR_CURRENT_OFFSET_MEASURE	1	-	-	Measuring DC link current offset	
MTR_CURRENT_OFFSET_FINISH	2	-	-	Finish measurement of DC link current offset	
MTR_IPD	0	-	-	Start-up	
MTR_OPENLOOP	1	-	-	Open-loop control	
MTR_SPEED_CONTROL	2	-	-	Speed PI control	
MTR_STOP	0	-	-	Motor stop	
MTR_LOW_SPEED	1	-	-	Low-speed driving	
MTR_MIDDLE_SPEED	2	-	-	Middle-speed driving	
MTR_HIGH_SPEED	3	-	-	High-speed driving	
MTR_OFFSET_CALC_EXE	0	-	-	Current offset measurement for calibration executing	
MTE_OFFSET_CALC_END	1	-	-	Current offset measurement for calibration finished	
MTR_IPD_EXE	2	-	-	Initial position detection executing	
MTR_IPD_END	3	-	-	Initial position detection finished	
MTR_DRIVE_OL	4	-	-	Open-loop drive	
MTR_DRIVE_ASR_CL	5	-	-	Speed PI drive	
MTR_DRIVE_BRAKE	6	-	-	Brake mode	
MTR_DRIVE_END	7	-	-	Driving finished	

Table 3-82 List of Macro definitions “r_mtr_ipd.h”[1/2]

Macro	Definition value	Qn	Pu	Description	Remarks
MTR_REF_CURRENT_BASE	(float)IP_DC_AMPLIFICATION_GAIN * IP_DC_SHUNT_RESISTANCE * 1024 / 5	-	-	Scaling factor for setting threshold current of CMP3	[Initial position detection]
MTR_IPD_NOISE_AVOID_CNT	CP_IPD_NOISE_AVOID_CNT	-	-	Noise avoidance count at IPD	
MTR_SAL_ANGLE_MAX_CNT	20	-	-	Maximum number of times salient rotor angle detection is measured	
MTR_SAL_ANGLE_PERCENTAGE	30	-	-	Percentage of TRX differential threshold of angle detection for salient rotor	
MTR_SAL_POLARITY_MAX_CNT	20	-	-	Maximum number of times salient rotor polarity detection is measured	
MTR_SAL_POLARITY_PERCENTAGE	30	-	-	Percentage of TRX differential threshold of polarity detection for salient rotor	
MTR_NON_SAL_MAX_CNT	20	-	-	Maximum number of times non-salient rotor angle detection is measured	
MTR_NON_SAL_PERCENTAGE	10	-	-	Percentage of TRX differential threshold of angle detection for non-salient rotor	
MTR_PERCENTAGE	100	-	-	Percentage calculation	
MTR_ENERGIZED_2_PHASES	0	-	-	2 phases energized	
MTR_ENERGIZED_3_PHASES	1	-	-	3 phases energized	
MTR_REVERSE_DIRECTION	3	-	-	Polarity inversion	
MTR_PRE_JUDGE	0	-	-	Pre-judgement of salient	
MTR_SALIENT	1	-	-	Salient	
MTR_NON_SALIENT	2	-	-	Non-salient	

Table 3-83 List of Macro definitions “r_mtr_ipd.h”[2/2]

Macro	Definition value	Qn	Pu	Description	Remarks
MTR_ENERGIZE_U2V	0	-	-	Voltage pattern during 2-phase energizing	
MTR_ENERGIZE_V2W	1	-	-		
MTR_ENERGIZE_W2U	2	-	-		
MTR_ENERGIZE_V2U	3	-	-		
MTR_ENERGIZE_W2V	4	-	-		
MTR_ENERGIZE_U2W	5	-	-		
MTR_ENERGIZE_U2VW	0	-	-	Voltage pattern during 3-phase energizing	
MTR_ENERGIZE_V2WU	1	-	-		
MTR_ENERGIZE_W2UV	2	-	-		
MTR_ENERGIZE_VW2U	3	-	-		
MTR_ENERGIZE_WU2V	4	-	-		
MTR_ENERGIZE_UV2W	5	-	-		
MTR_MAX_PHASE	0	-	-	Max count phase	
MTR_MN_PHASE	1	-	-	Minimum count phase	
MTR_POLARITY_NONE	0	-	-	Polarity default value	
MTR_POLARITY_POSITIVE	1	-	-	Polarity positive direction	
MTR_POLARITY_NEGATIVE	2	-	-	Polarity negative direction	
MTR_IPD_NONE	0	-	-	Default state	
MTR_IPD_SAL_ANGLE	1	-	-	Angle detection for salient rotor	
MTR_IPD_SAL_POLARITY	2	-	-	Polarity detection for salient rotor	
MTR_IPD_NON_SAL	3	-	-	Angle detection for non-salient rotor	
MTR_IPD_FINISH	4	-	-	Finish initial position detection	
MTR_IPD_UNDETECTED	5	-	-	Initial position detection failure	
MTR_IPD_ERROR	6	-	-	Initial position detection error	
MTR_OL_V_PATTERN_1	1	-	-	Voltage pattern number for open-loop drive	
MTR_OL_V_PATTERN_2	2	-	-		
MTR_OL_V_PATTERN_3	3	-	-		
MTR_OL_V_PATTERN_4	4	-	-		
MTR_OL_V_PATTERN_5	5	-	-		
MTR_OL_V_PATTERN_6	6	-	-		
MTR_TRX_CNT	TRX	-	-	TRX count register	

Table 3-84 List of Macro definitions "Config_TAU0_1.h"

Macro	Definition value	Qn	Pu	Description	Remarks
MTR_TAU1_CNT	TCR01	-	-	TAU1 count register	

Table 3-85 List of Macro definitions "Config_INTC.h"

Macro	Definition value	Qn	Pu	Description	Remarks
MTR_PORT_HALL_U	P5_bit.no2	-	-	U phase Hall effect sensor input port	[Hall effect sensor mode]
MTR_PORT_HALL_V	P5_bit.no3	-	-	V phase Hall effect sensor input port	
MTR_PORT_HALL_W	P5_bit.no4	-	-	W phase Hall effect sensor input port	

Table 3-86 List of Macro definitions "Config_INTC.h"

Macro	Definition value	Qn	Pu	Description	Remarks
R_MTR_PORT_LED1	P4_bit.no2	-	-	LED1 output port	
R_MTR_PORT_LED2	P4_bit.no3	-	-	LED2 output port	

Table 3-87 List of Macro definitions "Config_TRD0_TRD1.h"

Macro	Definition value	Qn	Pu	Description	Remarks
R_MTR_PORT_UP	P7_bit.no0	-	-	U phase (positive phase) output port	-
R_MTR_PORT_UN	P7_bit.no1	-	-	U phase (negative phase) output port	-
R_MTR_PORT_VP	P7_bit.no2	-	-	V phase (positive phase) output port	-
R_MTR_PORT_VN	P7_bit.no3	-	-	V phase (negative phase) output port	-
R_MTR_PORT_WP	P7_bit.no4	-	-	W phase (positive phase) output port	-
R_MTR_PORT_WN	P7_bit.no5	-	-	W phase (negative phase) output port	-
MTR_TRD_OUTPUT_ENABLE	TRDOER1	-	-	TRD output enable register for PWM	-

Table 3-88 List of Macro definitions "Config_ADC.h"

Macro	Definition value	Qn	Pu	Description	Remarks
MTR_ADCCH_VDC	6	-	-	A/D converter channel of bus voltage	-
MTR_ADCCH_VU	19	-	-	A/D converter channel of U phase voltage	-
MTR_ADCCH_VV	29	-	-	A/D converter channel of V phase voltage	-
MTR_ADCCH_VW	30	-	-	A/D converter channel of W phase voltage	-
MTR_ADCCH_IDC	18	-	-	A/D converter channel of DC link current	-
MTR_ADCCH_IU	2	-	-	A/D converter channel of U phase current	-
MTR_ADCCH_IV	3	-	-	A/D converter channel of V phase current	-
MTR_ADCCH_IW	4	-	-	A/D converter channel of W phase current	
MTR_AD_TRG_IDC	0	-	-	DC link current	
MTR_AD_TRG_VDC	1	-	-	Bus voltage	

3.5 Control flows (flow charts)

3.5.1 Main process

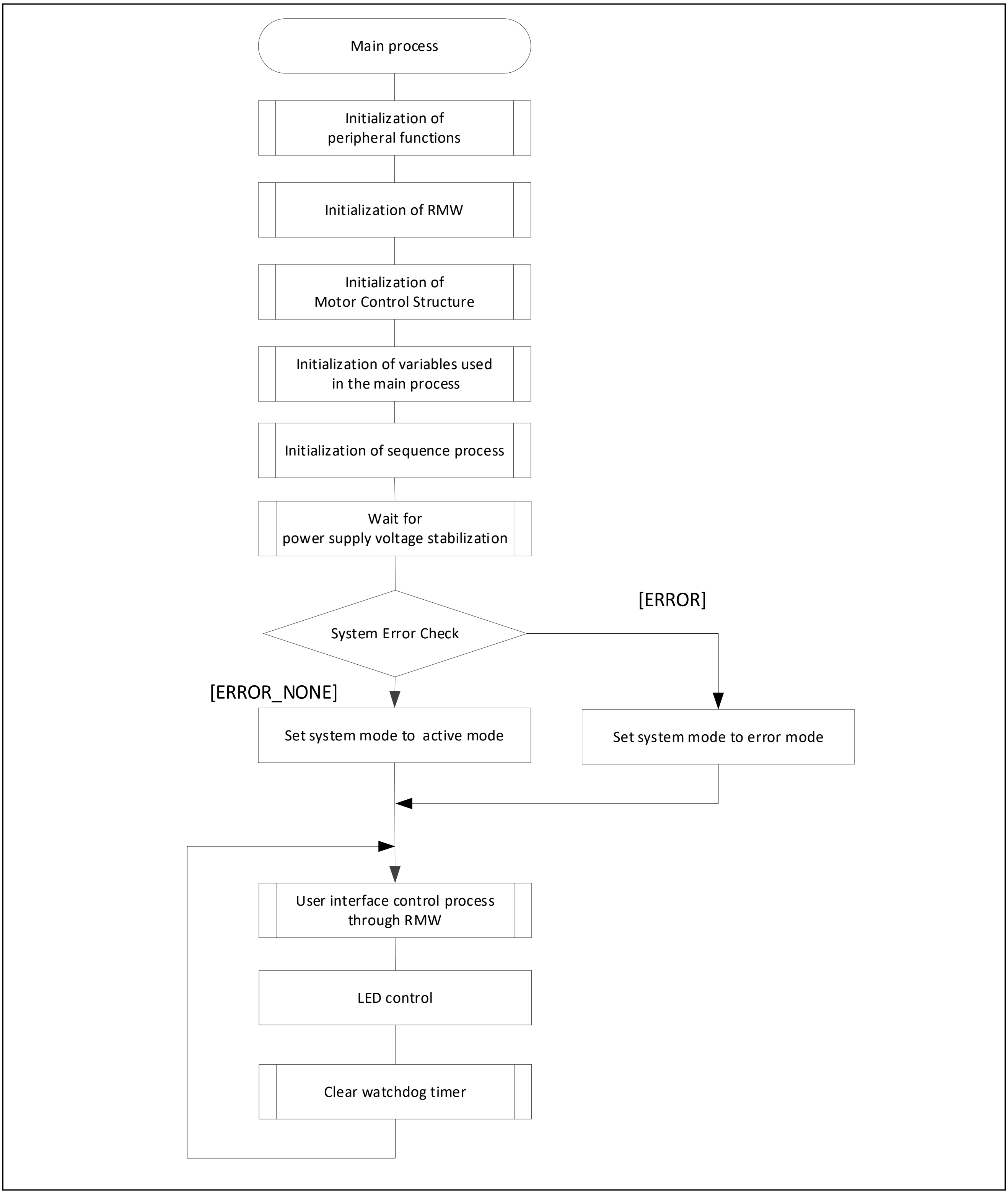
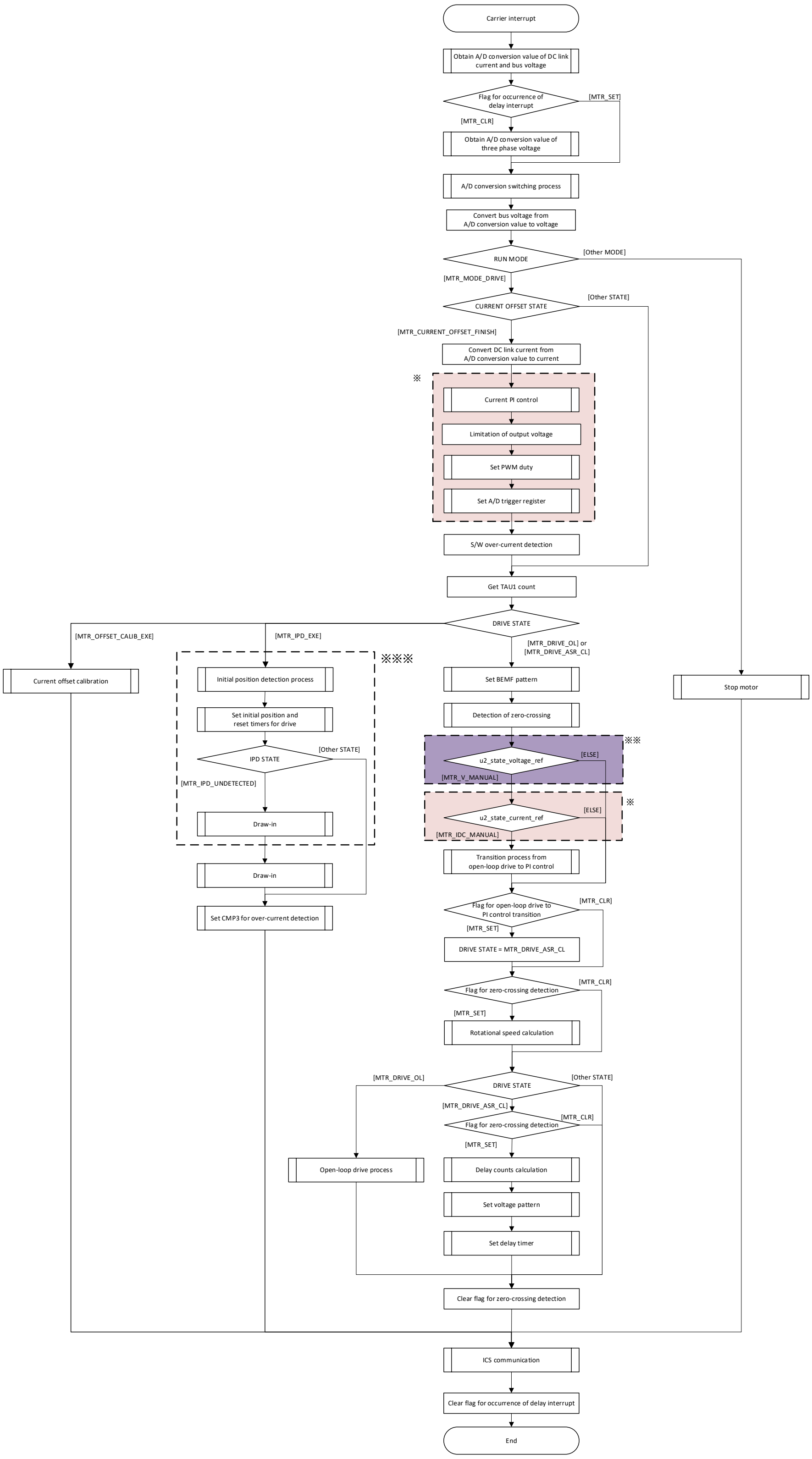

Main process
Initialization of peripheral functions
Initialization of RMW
Initialization of Motor Control Structure
Initialization of variables used in the main process
Initialization of sequence process
Wait for power supply voltage stabilization
System Error Check
[ERROR]
[ERROR_NONE]
Set system mode to active mode
Set system mode to error mode
User interface control process through RMW
LED control
Clear watchdog timer

Figure 3-32 Main Process Flowchart

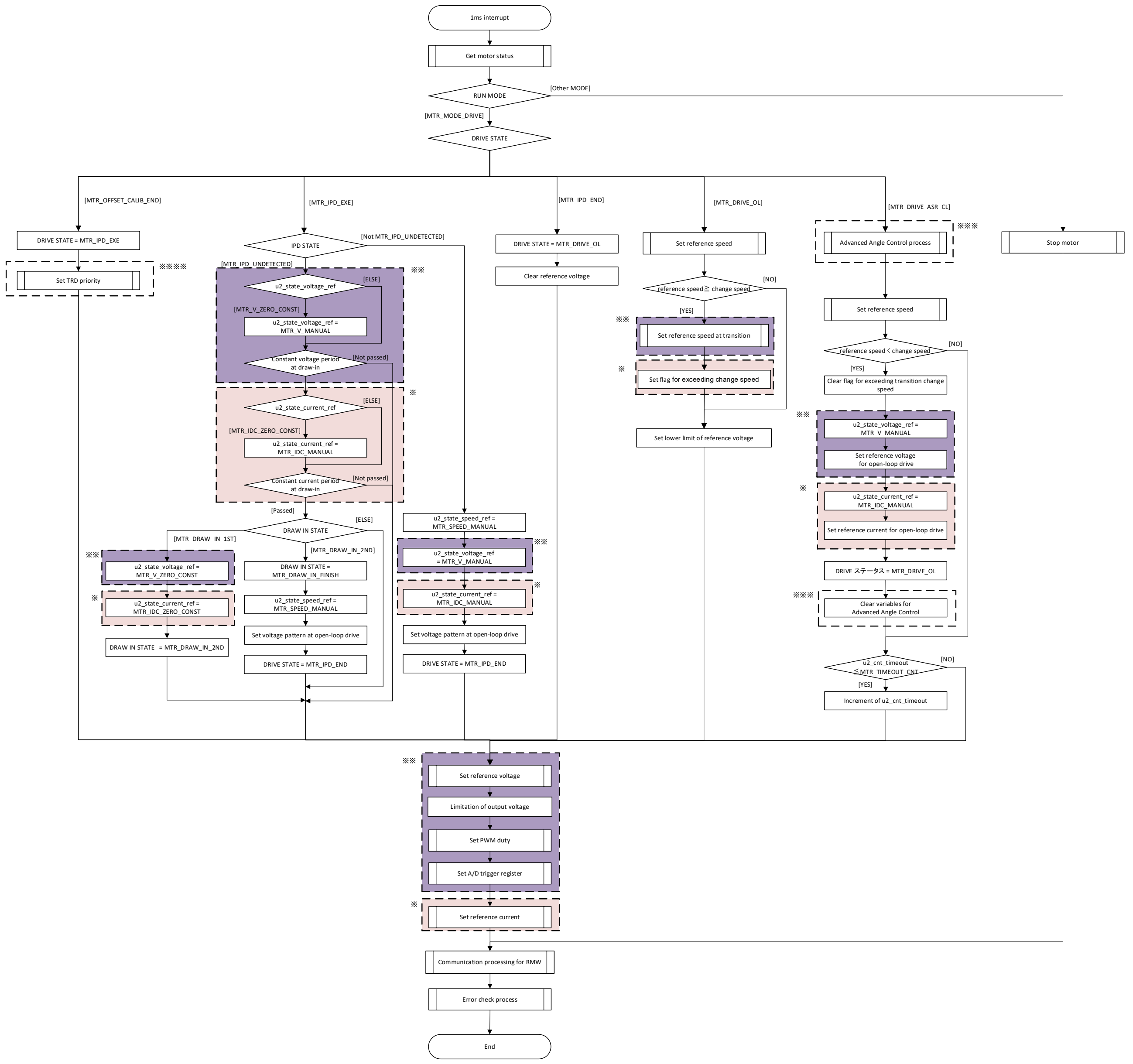
3.5.3 Carrier cycle interrupt handling (Sensorless mode)


Carrier interrupt
Obtain A/D conversion value of DC link current and bus voltage
Flag for occurrence of delay interrupt
[MTR_SET]
[MTR_CLR]
Obtain A/D conversion value of three phase voltage
A/D conversion switching process
Convert bus voltage from A/D conversion value to voltage
RUN MODE
[Other MODE]
[MTR_MODE_DRIVE]
CURRENT OFFSET STATE
[Other STATE]
[MTR_CURRENT_OFFSET_FINISH]
Convert DC link current from A/D conversion value to current
※
Current PI control
Limitation of output voltage
Set PWM duty
Set A/D trigger register
S/W over-current detection
Get TAU1 count
DRIVE STATE
[MTR_OFFSET_CALIB_EXE]
[MTR_IPD_EXE]
[MTR_DRIVE_OL] or [MTR_DRIVE_ASR_CL]
※※※
Current offset calibration
Initial position detection process
Set initial position and reset timers for drive
IPD STATE
[Other STATE]
[MTR_IPD_UNDETECTED]
Draw-in
Draw-in
Set CMP3 for over-current detection
Set BEMF pattern
Detection of zero-crossing
※※
u2_state_voltage_ref
[ELSE]
[MTR_V_MANUAL]
※
u2_state_current_ref
[ELSE]
[MTR_IDC_MANUAL]
Transition process from open-loop drive to PI control
Flag for open-loop drive to PI control transition
[MTR_CLR]
[MTR_SET]
DRIVE STATE = MTR_DRIVE_ASR_CL
Flag for zero-crossing detection
[MTR_CLR]
[MTR_SET]
Rotational speed calculation
DRIVE STATE
[MTR_DRIVE_OL]
[Other STATE]
[MTR_DRIVE_ASR_CL]
Flag for zero-crossing detection
[MTR_CLR]
[MTR_SET]
Open-loop drive process
Delay counts calculation
Set voltage pattern
Set delay timer
Clear flag for zero-crossing detection
Stop motor
ICS communication
Clear flag for occurrence of delay interrupt
End

※ When Control Mode is Voltage PI Control Mode, this process is not executed.
※※ When Control Mode is Current PI Control Mode, this process is not executed.
※※※When Initial Position Detection function is not used, this process is not executed.

Figure 3-33 Carrier Cycle Interrupt Handling (Sensorless mode)

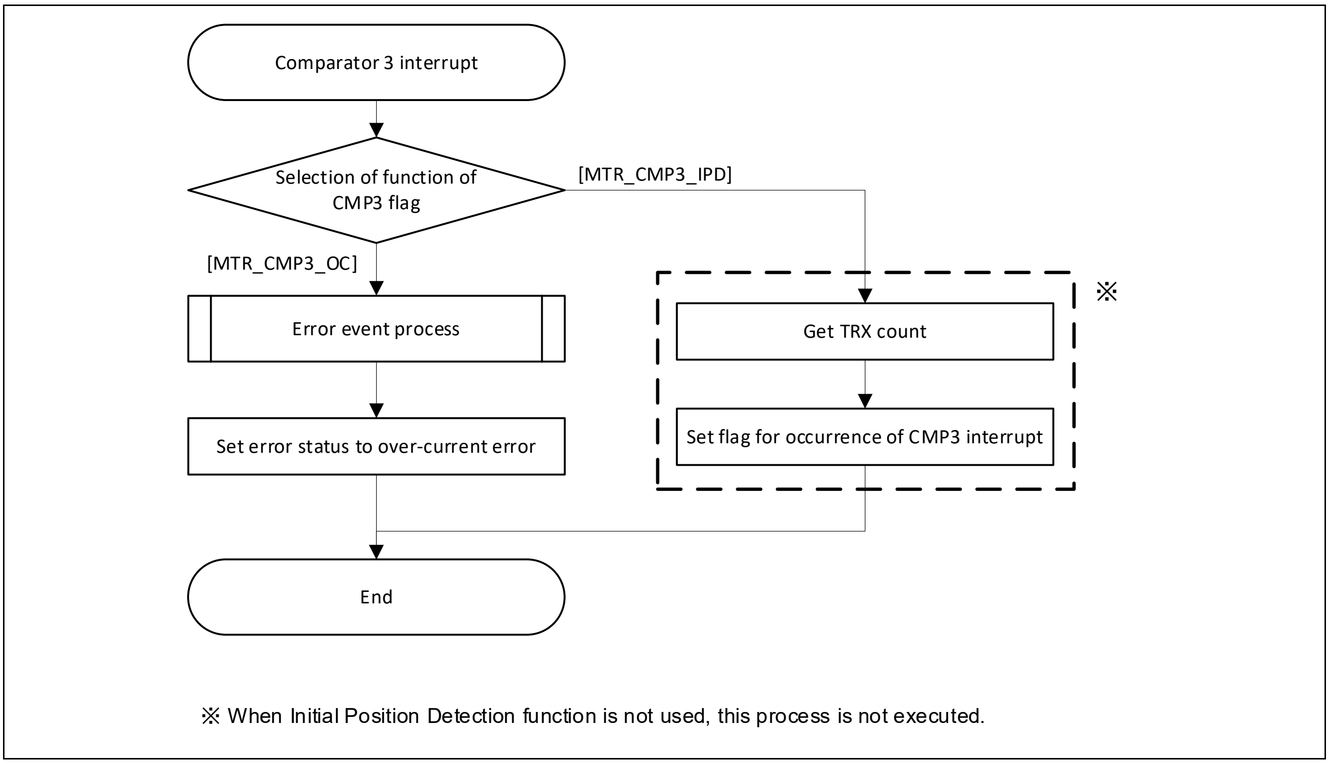
3.5.4 1 [ms] interrupt handling (Sensorless mode)


1ms interrupt
Get motor status
RUN MODE
[Other MODE]
[MTR_MODE_DRIVE]
DRIVE STATE
[MTR_OFFSET_CALIB_END]
[MTR_IPD_EXE]
[MTR_IPD_END]
[MTR_DRIVE_OL]
[MTR_DRIVE_ASR_CL]
DRIVE STATE = MTR_IPD_EXE
※※※※
Set TRD priority
IPD STATE
[Not MTR_IPD_UNDETECTED]
[MTR_IPD_UNDETECTED]
※※
u2_state_voltage_ref
[ELSE]
[MTR_V_ZERO_CONST]
u2_state_voltage_ref = MTR_V_MANUAL
Constant voltage period at draw-in
[Not passed]
※
u2_state_current_ref
[ELSE]
[MTR_IDC_ZERO_CONST]
u2_state_current_ref = MTR_IDC_MANUAL
Constant current period at draw-in
[Not passed]
[Passed]
DRAW IN STATE
[ELSE]
[MTR_DRAW_IN_1ST]
[MTR_DRAW_IN_2ND]
※※
u2_state_voltage_ref = MTR_V_ZERO_CONST
※
u2_state_current_ref = MTR_IDC_ZERO_CONST
DRAW IN STATE = MTR_DRAW_IN_2ND
DRAW IN STATE = MTR_DRAW_IN_FINISH
u2_state_speed_ref = MTR_SPEED_MANUAL
Set voltage pattern at open-loop drive
DRIVE STATE = MTR_IPD_END
u2_state_speed_ref = MTR_SPEED_MANUAL
※※
u2_state_voltage_ref = MTR_V_MANUAL
※
u2_state_current_ref = MTR_IDC_MANUAL
Set voltage pattern at open-loop drive
DRIVE STATE = MTR_IPD_END
DRIVE STATE = MTR_DRIVE_OL
Clear reference voltage
Set reference speed
reference speed≧ change speed
[NO]
[YES]
※※
Set reference speed at transition
※
Set flag for exceeding change speed
Set lower limit of reference voltage
※※※
Advanced Angle Control process
Set reference speed
reference speed < change speed
[NO]
[YES]
Clear flag for exceeding transition change speed
※※
u2_state_voltage_ref = MTR_V_MANUAL
Set reference voltage for open-loop drive
※
u2_state_current_ref = MTR_IDC_MANUAL
Set reference current for open-loop drive
DRIVE ステータス = MTR_DRIVE_OL
※※※
Clear variables for Advanced Angle Control
u2_cnt_timeout ≦MTR_TIMEOUT_CNT
[NO]
[YES]
Increment of u2_cnt_timeout
Stop motor
※※
Set reference voltage
Limitation of output voltage
Set PWM duty
Set A/D trigger register
※
Set reference current
Communication processing for RMW
Error check process
End

※ When Control Mode is Voltage PI Control Mode, this process is not executed.
※※ When Control Mode is Current PI Control Mode, this process is not executed.
※※※ When Advanced Angle Control function is not used, this process is not executed.
※※※※When Initial Position Detection function is not used, this process is not executed.

Figure 3-34 1 [ms] Interrupt Handling (Sensorless mode)

3.5.5 Comparator 3 interrupt handling


Comparator 3 interrupt
Selection of function of CMP3 flag
[MTR_CMP3_IPD]
[MTR_CMP3_OC]
Error event process
Set error status to over-current error
Get TRX count
Set flag for occurrence of CMP3 interrupt
※
End

※ When Initial Position Detection function is not used, this process is not executed.

Figure 3-35 Over Current Detection Interrupt Handling

3.5.6 Delay timer interrupt handling

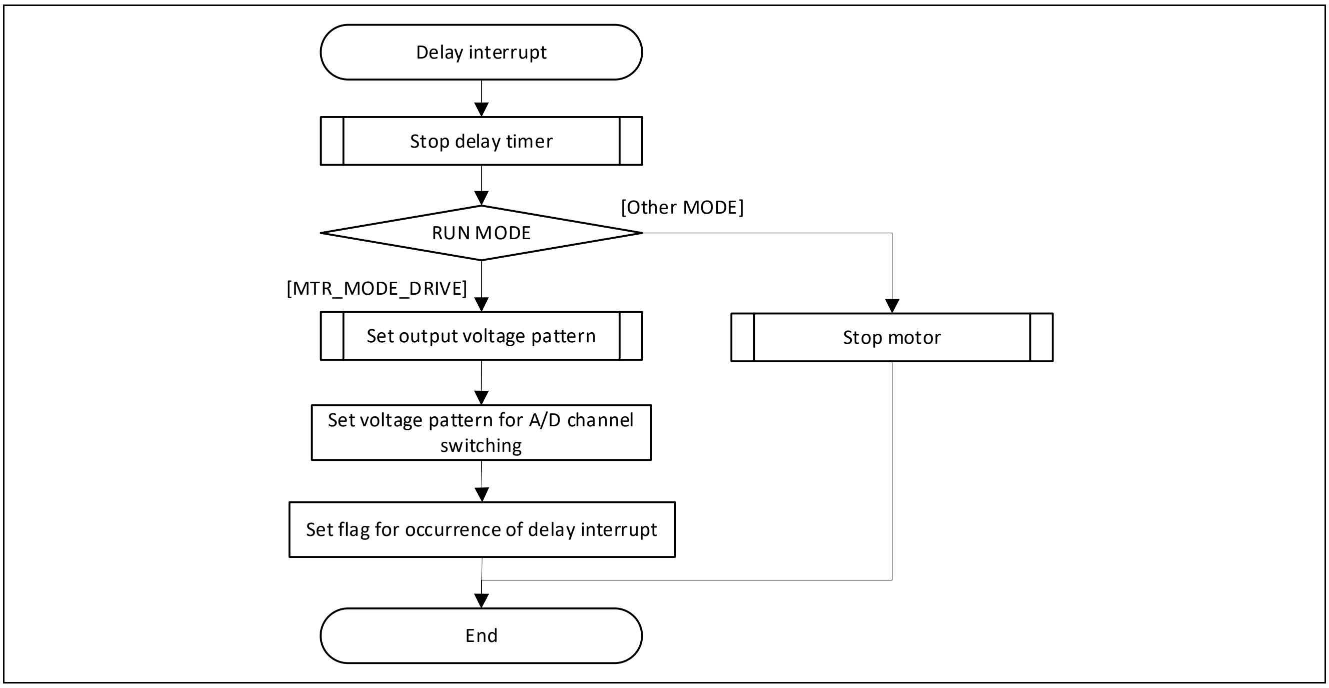
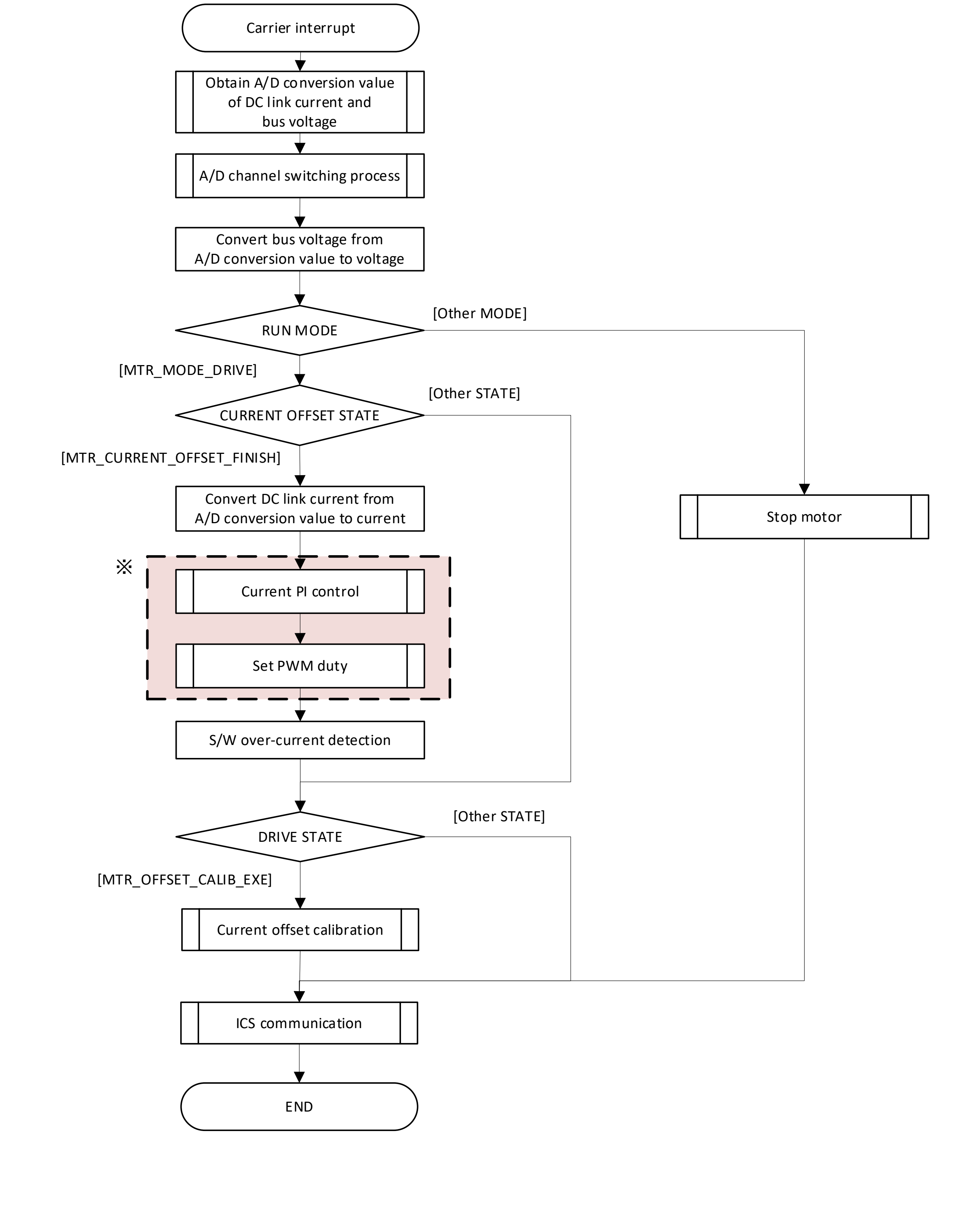

Delay interrupt
Stop delay timer
RUN MODE
[Other MODE]
[MTR_MODE_DRIVE]
Set output voltage pattern
Stop motor
Set voltage pattern for A/D channel switching
Set flag for occurrence of delay interrupt
End

Figure 3-36 Delay Timer Interrupt Handling

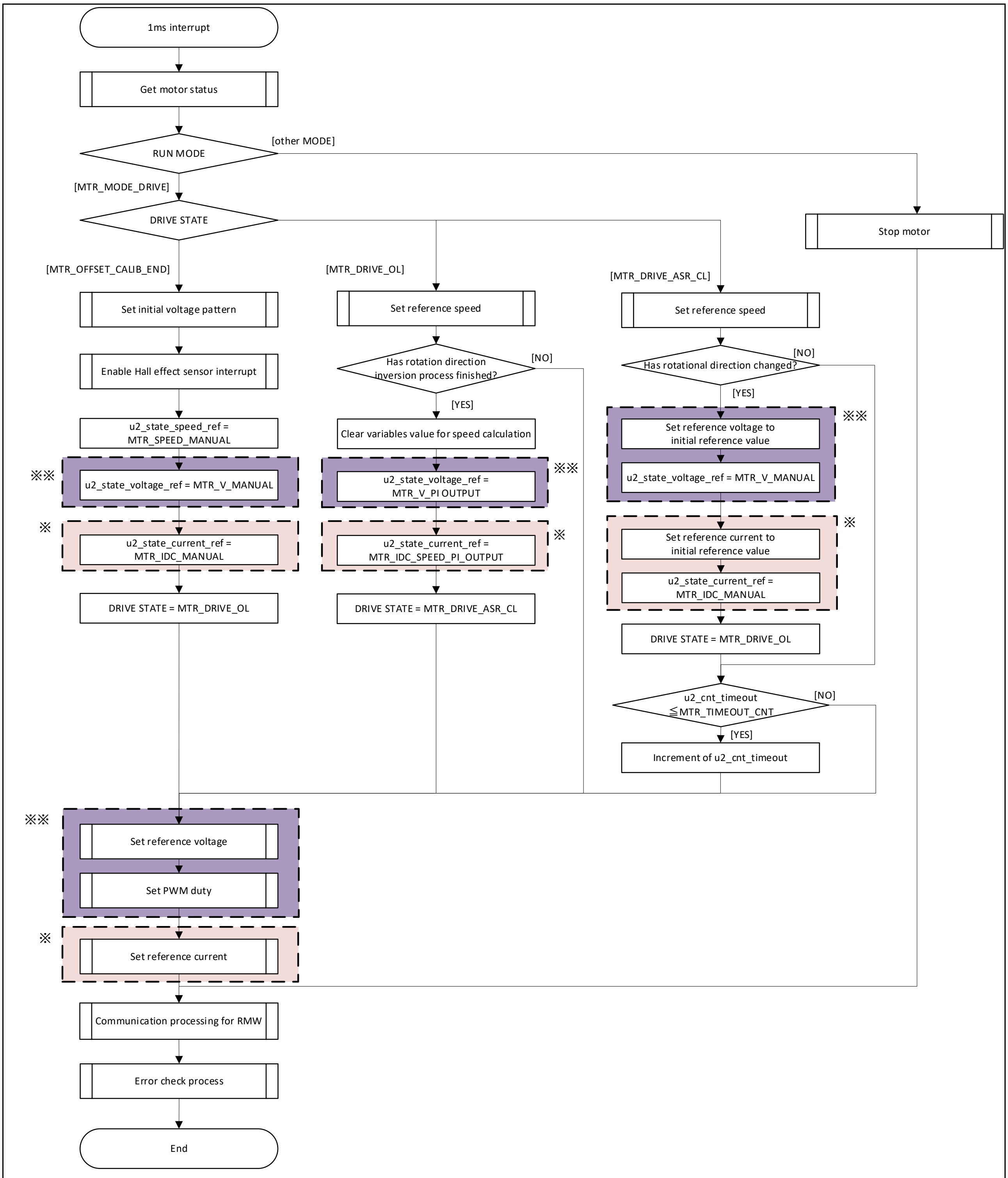
3.5.7 Carrier cycle interrupt handling (Hall effect sensor mode)


Carrier interrupt
Obtain A/D conversion value of DC link current and bus voltage
A/D channel switching process
Convert bus voltage from A/D conversion value to voltage
RUN MODE
[Other MODE]
[MTR_MODE_DRIVE]
CURRENT OFFSET STATE
[Other STATE]
[MTR_CURRENT_OFFSET_FINISH]
Convert DC link current from A/D conversion value to current
※
Current PI control
Set PWM duty
S/W over-current detection
Stop motor
DRIVE STATE
[Other STATE]
[MTR_OFFSET_CALIB_EXE]
Current offset calibration
ICS communication
END

※ When Control Mode is Voltage PI Control Mode, this process is not executed.

Figure 3-37 Carrier Cycle Interrupt Handling (Hall effect sensor mode)

3.5.8 1 [ms] interrupt handling (Hall effect sensor mode)


1ms interrupt
Get motor status
RUN MODE
[other MODE]
[MTR_MODE_DRIVE]
DRIVE STATE
Stop motor
[MTR_OFFSET_CALIB_END]
[MTR_DRIVE_OL]
[MTR_DRIVE_ASR_CL]
Set initial voltage pattern
Enable Hall effect sensor interrupt
u2_state_speed_ref = MTR_SPEED_MANUAL
※※
u2_state_voltage_ref = MTR_V_MANUAL
※
u2_state_current_ref = MTR_IDC_MANUAL
DRIVE STATE = MTR_DRIVE_OL
Set reference speed
Has rotation direction inversion process finished?
[NO]
[YES]
Clear variables value for speed calculation
u2_state_voltage_ref = MTR_V_PI OUTPUT
※※
u2_state_current_ref = MTR_IDC_SPEED_PI_OUTPUT
※
DRIVE STATE = MTR_DRIVE_ASR_CL
Set reference speed
Has rotational direction changed?
[NO]
[YES]
Set reference voltage to initial reference value
u2_state_voltage_ref = MTR_V_MANUAL
※※
Set reference current to initial reference value
u2_state_current_ref = MTR_IDC_MANUAL
※
DRIVE STATE = MTR_DRIVE_OL
u2_cnt_timeout ≦MTR_TIMEOUT_CNT
[NO]
[YES]
Increment of u2_cnt_timeout
※※
Set reference voltage
Set PWM duty
※
Set reference current
Communication processing for RMW
Error check process
End

※ When Control Mode is Voltage PI Control Mode, this process is not executed.

※※When Control Mode is Current PI Control Mode, this process is not executed.

Figure 3-38 1[ms] Cycle Interrupt Handling (Hall effect sensor mode)

3.5.9 Hall effect sensor interrupt handling

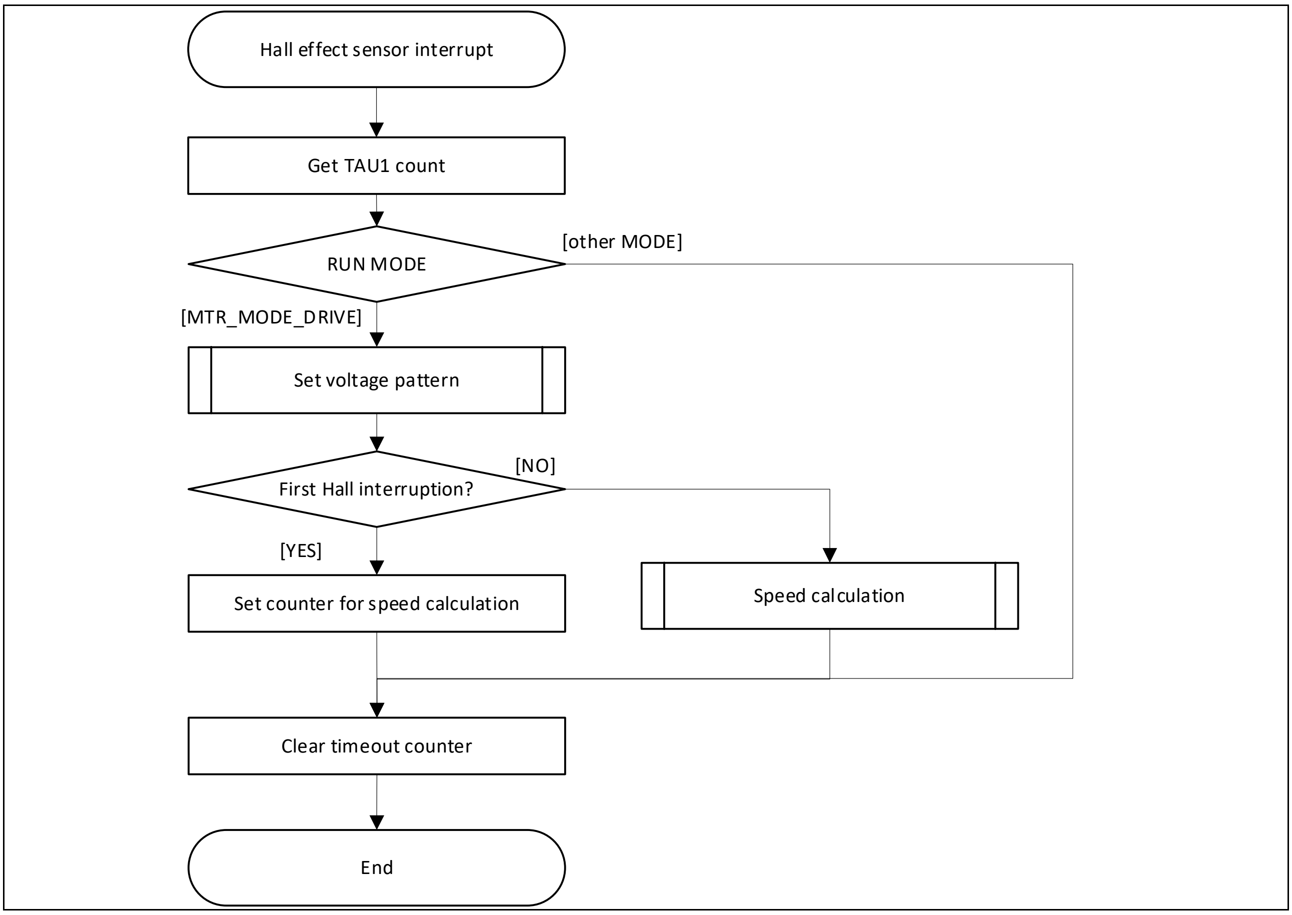
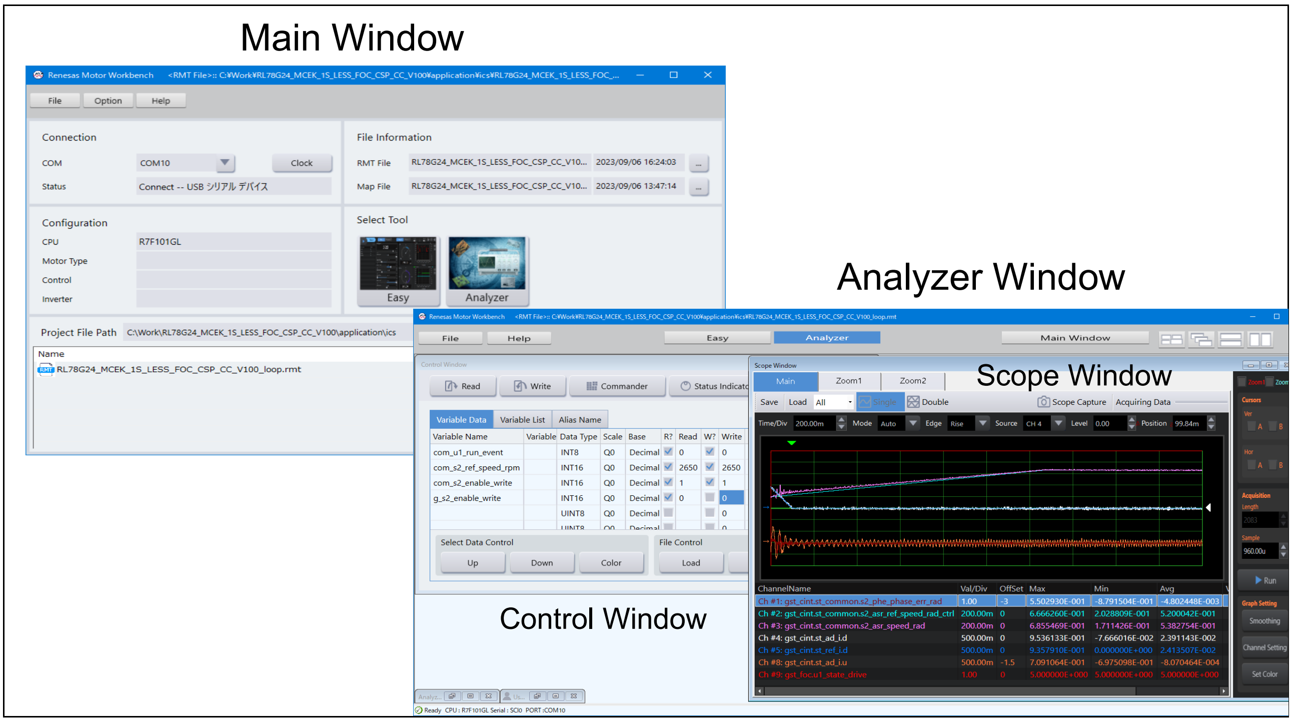


Figure 3-39 Hall effect sensor Interrupt Handling

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.



<RMT File>:: C:¥Work¥RL78G24_MCEK_1S_LESS_FOC_CSP_CC_V100¥application¥ics¥RL78G24_MCEK_1S_LESS_FOC_...

Connection
COM: COM10
Status: Connect -- USB シリアル デバイス

File Information
RMT File: RL78G24_MCEK_1S_LESS_FOC_CSP_CC_V10... 2023/09/06 16:24:03
Map File: RL78G24_MCEK_1S_LESS_FOC_CSP_CC_V10... 2023/09/06 13:47:14

Configuration
CPU: R7F101GL
Motor Type:
Control:
Inverter:

Select Tool

Project File Path: C:\Work\RL78G24_MCEK_1S_LESS_FOC_CSP_CC_V100\application\ics

Name
RL78G24_MCEK_1S_LESS_FOC_CSP_CC_V100_loop.rmt

<RMT File>:: C:¥Work¥RL78G24_MCEK_1S_LESS_FOC_CSP_CC_V100¥application¥ics¥RL78G24_MCEK_1S_LESS_FOC_CSP_CC_V100_loop.rmt

Variable Name	Variable	Data Type	Scale	Base	R?	Read	W?	Write
com_u1_run_event		INT8	Q0	Decimal	☑	0	☑	0
com_s2_ref_speed_rpm		INT16	Q0	Decimal	☑	2650	☑	2650
com_s2_enable_write		INT16	Q0	Decimal	☑	1	☑	1
g_s2_enable_write		INT16	Q0	Decimal	☑	0	□	0
		UINT8	Q0	Decimal	□		□	0
		UINT8	Q0	Decimal	□		□	0

Time/Div 200.00m Mode Auto Edge Rise Source CH 4 Level 0.00 Position 99.84m

Length 2083
Sample 960.00u

ChannelName	Val/Div	OffSet	Max	Min	Avg
Ch #1: gst_cint.st_common.s2_phe_phase_err_rad	1.00	-3	5.502930E-001	-8.791504E-001	-4.802448E-003
Ch #2: gst_cint.st_common.s2_asr_ref_speed_rad_ctrl	200.00m	0	6.666260E-001	2.028809E-001	5.200042E-001
Ch #3: gst_cint.st_common.s2_asr_speed_rad	200.00m	0	6.855469E-001	1.711426E-001	5.382754E-001
Ch #4: gst_cint.st_ad_i.d	500.00m	0	9.536133E-001	-7.666016E-002	2.391143E-002
Ch #5: gst_cint.st_ref_i.d	500.00m	0	9.357910E-001	0.000000E+000	2.413507E-002
Ch #8: gst_cint.st_ad_i.u	500.00m	-1.5	7.091064E-001	-6.975098E-001	-8.070464E-004
Ch #9: gst_foc.u1_state_drive	1.00	0	5.000000E+000	5.000000E+000	5.000000E+000

Ready CPU : R7F101GL Serial : SCI0 PORT :COM10

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 Renesas Motor Workbench.
(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 com_s2_enable_write. However, note that variables with (*) do not depend on 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]
com_u1_direction	uint8_t	Direction of rotation 0：CW 1：CCW	[g_st_120.s1_ref_dir]
com_u2_mtr_pp	uint16_t	Number of pole pairs	[g_st_120.st_mtr.u2_mtr_pp]
com_f4_mtr_r	float	Resistance [Ω]	[g_st_120.st_mtr.s2_mtr_r]
com_f4_mtr_ld	float	D-axis inductance[H/rad]	[g_st_120.st_mtr.s2_mtr_ld]
com_f4_mtr_lq	float	Q-axis inductance[H/rad]	[g_st_120.st_mtr.s2_mtr_lq]
com_f4_mtr_m	float	BEMF voltage constant [V･s/rad]	[g_st_120.st_mtr.s2_mtr_m]
com_f4_mtr_j	float	Rotor inertia [kgm^2/(rad s^2]	[g_st_120.st_mtr.s2_mtr_j]
com_f4_limit_v_ref	float	Lower limit of voltage reference [V]	[g_st_120.s2_limit_v_ref]
com_s2_ramp_limit_speed_rpm	int16_t	Limit of acceleration [rpm/ms]	[g_st_120.s2_ramp_limit_speed_rad]
com_s2_ref_speed_rpm	int16_t	Command rotational speed ［rpm］	[g_st_120.s2_ref_speed_rad]
com_u2_offset_calc_cnt	uint16_t	Current offset detection time	[g_st_120.u2_offset_calc_cnt]
com_f4_ramp_limit_v	float	Limit of variation of voltage [V/ms]	[g_st_120.s2_ramp_limit_v]
com_f4_asr_omega_hz	float	Frequency for speed PI control [Hz]	[g_st_120.st_asr.s2_kp]
com_f4_asr_zeta	float	Damping factor for speed PI control	[g_st_120.st_asr.s2_kidt]
com_f4_ramp_limit_current	float	Limit of variation of current [A/ms]	[g_st_120.st_acr.s2_ramp_limit_current]
com_f4_mtr_d0	float	Static friction coefficient [kgm^2/(rads^2)]	[g_st_120.st_asr.s2_d0_div_pm]
com_f4_mtr_d1	float	Kinetic friction coefficient [kg･m^2/(rad^2･s)]	[g_st_120.st_asr.s2_d1_div_p2m]
com_f4_acr_nf_hz	float	Current PI control natural frequency [Hz]	[g_st_120.st_acr.st_pi.s2_kp] [g_st_120.st_acr.st_pi.s2_kidt]
com_f4_asr_nf_hz	float	Speed PI control natural frequency [Hz]	[g_st_120.st_asr.st_pi.s2_kp] [g_st_120.st_asr.st_pi.s2_kidt]
com_f4_asr_ki_aug	float	Augmentation rate for integral part of ASR	[g_st_120.st_asr.st_pi.s2_kp] [g_st_120.st_asr.st_pi.s2_kidt]
com_f4_acr_deadband_lsb	float	Current PI control deadband [LSB]	[g_st_120.st_acr.st_pi.s2_deadband]
com_f4_asr_deadband_lsb	float	Speed PI control deadband [LSB]	[g_st_120.st_asr.s2_deadband][Voltage control mode] [g_st_120.st_asr.st_pi.s2_deadband][Current control mode]
com_f4_start_ref_v	float	Reference voltage at start-up [V]	[g_st_120.st_hall.s2_start_ref_v]
com_f4_start_ref_idc	float	Reference current at start-up [A]	[g_st_120.st_hall.s2_start_ref_idc]
com_f4_draw_in_ref_v	float	Reference voltage at draw-in [V]	[g_st_120.st_less.s2_draw_in_ref_v]

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

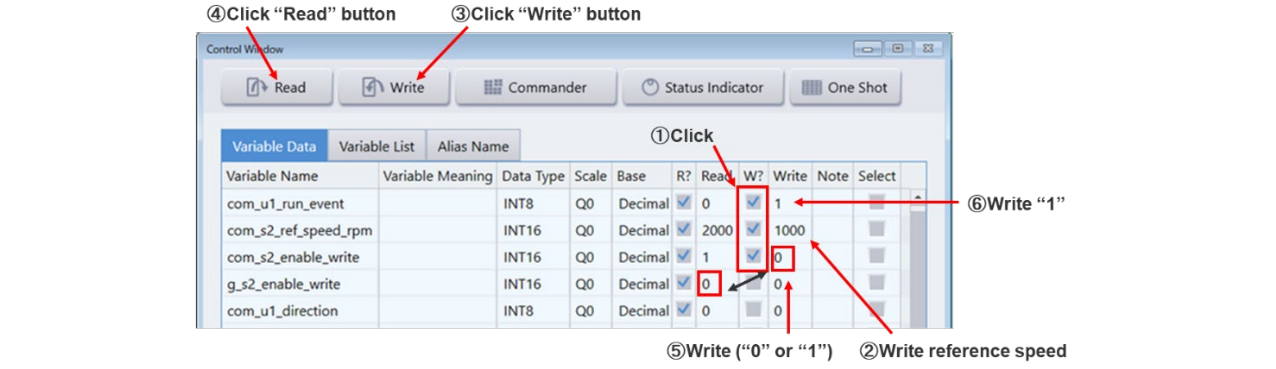
Variable	Type	Content	Remarks ([ ]: Corresponding variable name)
com_f4_ol_ref_v	float	Reference voltage at open-loop drive [V]	[g_st_120.st_less.s2_ol_ref_v]
com_f4_draw_in_ref_idc	float	Reference current at draw-in [A]	[g_st_120.st_less.s2_draw_in_ref_idc]
com_f4_ol_ref_idc	float	Reference current at open-loop drive [A]	[g_st_120.st_less.s2_ol_ref_idc]
com_s2_ol2cl_speed_rpm	int16_t	Speed allowed to transition to PI control[rpm]	[g_st_120.st_less.st_ol2cl. s2_change_speed_rad]
com_s2_ol2cl_ramp_speed_rpm	int16_t	Acceleration at transition to PI control [rpm/ms]	[g_st_120.st_less.st_ol2cl. s2_ramp_speed_rad]
com_s2_angle_shift_adjust	int16_t	Adjust delay counts	[g_st_120.st_less.s2_angle_shift_adjust]
com_s2_ad_trg_adjust_up_cnt	int16_t	Adjustment count of A/D trigger for ascending non-energized phase voltage	[g_st_120.st_less.s2_ad_trg_adjust_up_cnt]
com_s2_ad_trg_adjust_down_cnt	int16_t	Adjustment count of A/D trigger for descending non-energized phase voltage	[g_st_120.st_less.s2_ad_trg_adjust_down_cnt]
com_s2_ad_trg_diff_cnt	int16_t	Count difference of A/D between previous carrier and current carrier	[g_st_120.st_less.s2_ad_trg_diff_cnt]
com_f4_sal_angle_current	float	TRX count value differential of angle detection for salient rotor	[g_st_120.st_ipd.u2_sal_angle_current]
com_u4_sal_angle_th	uint32_t	Maximum number of times salient rotor angle detection is measured	[g_st_120.st_ipd.u4_sal_angle_th]
com_u2_sal_angle_discharge	uint8_t	Discharge period of angle detection for salient rotor	[g_st_120.st_ipd.u2_sal_angle_discharge]
com_f4_sal_polarity_current	float	TRX count value differential of polarity detection for salient rotor	[g_st_120.st_ipd.u2_sal_polarity_current]
com_u4_sal_polarity_th	uint32_t	Maximum number of times salient rotor polarity detection is measured	[g_st_120.st_ipd.u4_sal_polarity_th]
com_u2_sal_polarity_discharge	uint8_t	Discharge period of polarity detection for salient rotor	[g_st_120.st_ipd.u2_sal_polarity_discharge]
com_f4_non_sal_current	float	TRX count value differential of angle detection for non-salient rotor	[g_st_120.st_ipd.u2_non_sal_current]
com_u4_non_sal_th	uint32_t	Maximum number of times non-salient rotor angle detection is measured	[g_st_120.st_ipd. u4_non_sal_th]
com_u2_non_sal_discharge	uint8_t	Discharge period of angle detection for non-salient rotor	[g_st_120.st_ipd.u2_non_sal_discharge]
com_s2_aa_speed_err_th_rpm	uint16_t	Speed error threshold for Advanced angle control [rpm]	[g_st_120.st_aa. s2_speed_err_th_rad]
com_f4_aa_pi_nf_hz	float	PI controller natural frequency for Advanced angle control [Hz]	[g_st_120.st_aa.st_pi.s2_kp] [g_st_120.st_aa.st_pi.s2_kidt]
com_f4_aa_speed_err_lpf_cof_hz	float	Speed error LPF cutoff frequency [Hz]	[g_st_120.st_aa. s2_speed_err_rad_lpf]
com_f4_aa_ramp_limit_v	float	Limit of rate of neutral voltage change for Advanced angle control [V/ms]	[g_st_120.st_aa.s2_ramp_limit_voltage]
com_f4_aa_ramp_limit_angle	float	Limit of rate of angle offset change for Advanced angle control [rad/ms]	[g_st_120.st_aa.s2_ramp_limit_angle]
com_f4_aa_vlim	float	Voltage limit [V]	[g_st_120.st_aa.s2_vlim]
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, the "Control Window" shown in “Renesas Motor Workbench” is used. Refer to the ‘Renesas Motor Workbench V 3.1.2 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”, “com_s2_ref_speed_rpm”, and “com_s2_enable_write.”
② Input a reference rotational speed value in the [Write] box of “com_s2_ref_speed_rpm.”
③ Click the “Write” button.
④ Click the “Read” button. Confirm the [Read] box of “com_s2_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 “com_s2_enable_write.”
⑥ Input a value of “1” in the [Write] box of “com_u1_run_event.”
⑦ Click the “Write” button.



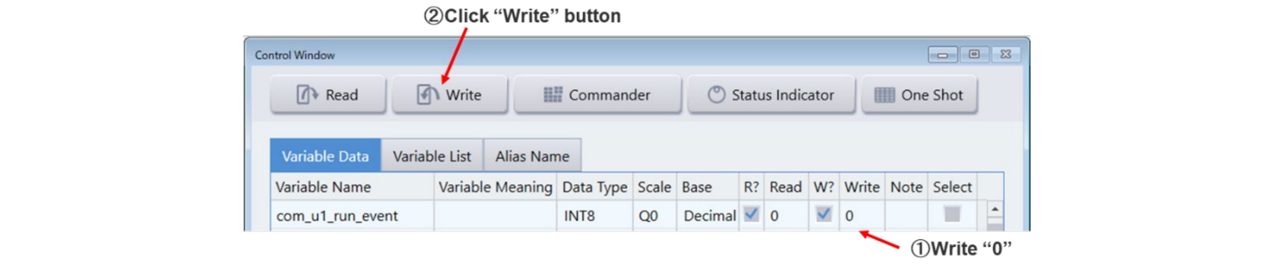
Variable Name	Variable Meaning	Data Type	Scale	Base	R?	Read	W?	Write	Note	Select
com_u1_run_event		INT8	Q0	Decimal	☑	0	☑	1		□
com_s2_ref_speed_rpm		INT16	Q0	Decimal	☑	2000	☑	1000		□
com_s2_enable_write		INT16	Q0	Decimal	☑	1	☑	0		□
g_s2_enable_write		INT16	Q0	Decimal	☑	0	□	0		□
com_u1_direction		INT8	Q0	Decimal	☑	0	□	0		□

④Click “Read” button
③Click “Write” button
①Click
⑥Write “1”
⑤Write (“0” or “1”)
②Write reference speed

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.



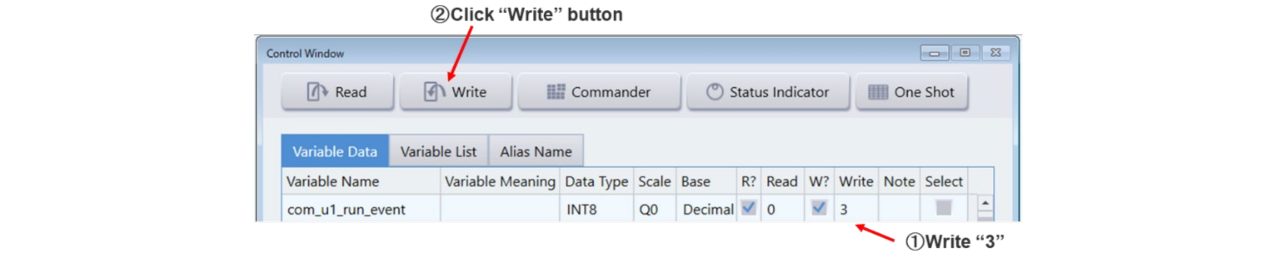
Variable Name	Variable Meaning	Data Type	Scale	Base	R?	Read	W?	Write	Note	Select
com_u1_run_event		INT8	Q0	Decimal	☑	0	☑	0		□

②Click “Write” button
①Write “0”

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.



Variable Name	Variable Meaning	Data Type	Scale	Base	R?	Read	W?	Write	Note	Select
com_u1_run_event		INT8	Q0	Decimal	☑	0	☑	3		□

②Click “Write” button
①Write “3”

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	09.09.2024	—	First edition issued

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

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

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

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

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

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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