RL78/G24

Sensorless vector control for permanent magnetic synchronous motor

Summary

This application note explains the sample programs for driving a permanent magnet synchronous motor in the sensorless vector 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 using the following device.

- RL78/G24(R7F101GLG2DFB)

Applicable sample programs

This application note regards the following sample programs.

<1 shunt current detection method>

- RL78G24_MCEK_1S_LESS_FOC_CSP_CC_V110 (IDE: CS+ for CC)
- RL78G24_MCEK_1S_LESS_FOC_E2S_CC_V110 (IDE: e[2]studio for CC)

<3 shunt current detection method>

- RL78G24_MCEK_3S_LESS_FOC_CSP_CC_V110 (IDE: CS+ for CC)
- RL78G24_MCEK_3S_LESS_FOC_E2S_CC_V110 (IDE: e[2]studio for CC)

References

- RL78/G24 Sensorless vector control for permanent magnetic synchronous motor FAA Library (R01AN7267EJ0110)
- 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 sensorless vector control sample programs of the permanent magnetic synchronous motor (PMSM) using the RL78/G24 microcontroller, and how to use the motor control development support tool Renesas Motor Workbench.

1.1 Development environment

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

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

Microcontroller	Evaluation board	Motor
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 (Software)

CS+ version	e2studio version	Build tool version	Smart Configurator	Debug tool
V8.11.00	2024-04	CC-RL V1.14.00	V1.10.0	-

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

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)

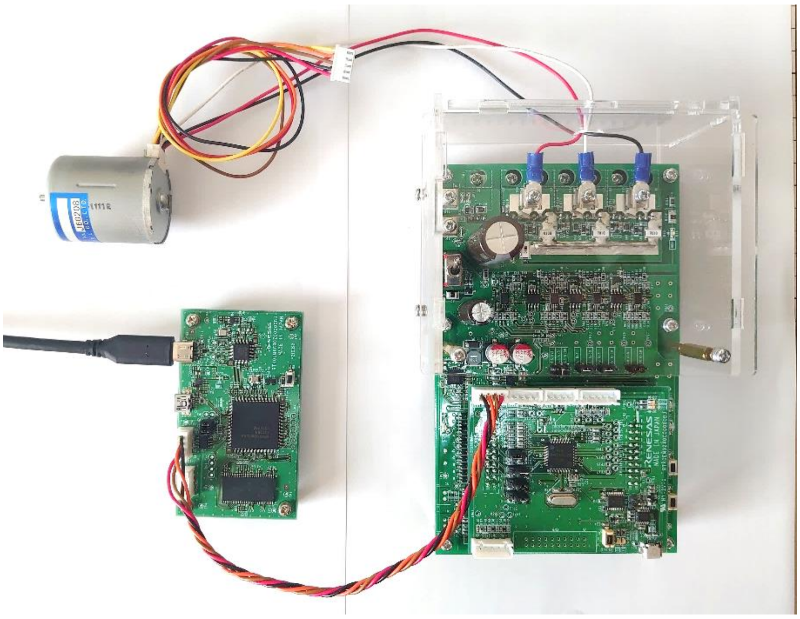
Item	Specification	
Product name	RL78/G24 Motor Control Evaluation Kit	
Kit model name	RTK0EMG24SS00000BJ	
Kit configuration	Low voltage Inverter Board LV400	RTK0EMGPLVB00000BJ
	RL78/G24 CPU Card	RTK0EMG240C00000BJ
	Communication Board MC-COM	RTK0EMXC90Z00000BJ
	PMSM	TG-55L-KA (TSUKASA Co., Ltd.)
Isolation	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)

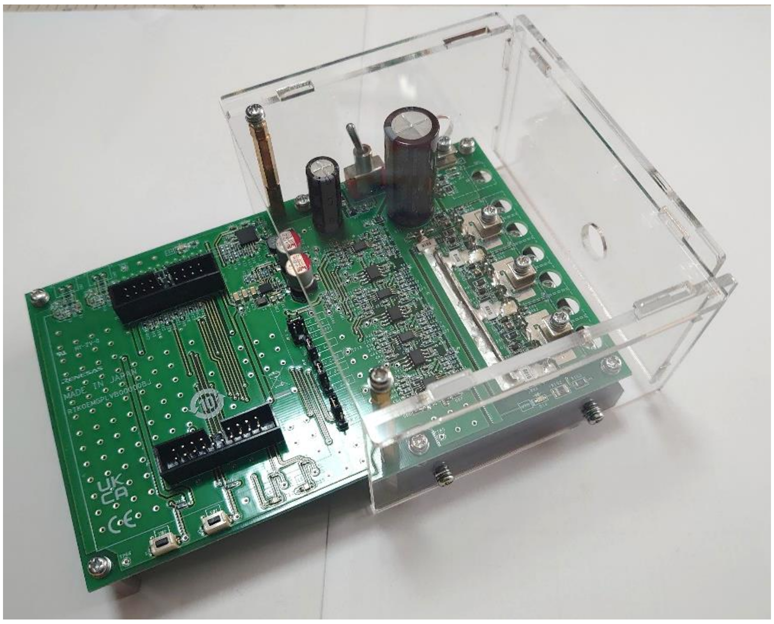
Item	Specification
Product name	Low voltage Inverter Board for Motor Control Evaluation Kit - LVI400
Board model name	RTK0EMGPLVB00000BJ
Operating input voltage	12V to 50V
Maximum output current	30A (peak current for each phase)
Driving motor	3-phase permanent magnet synchronous motor
Current detection 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

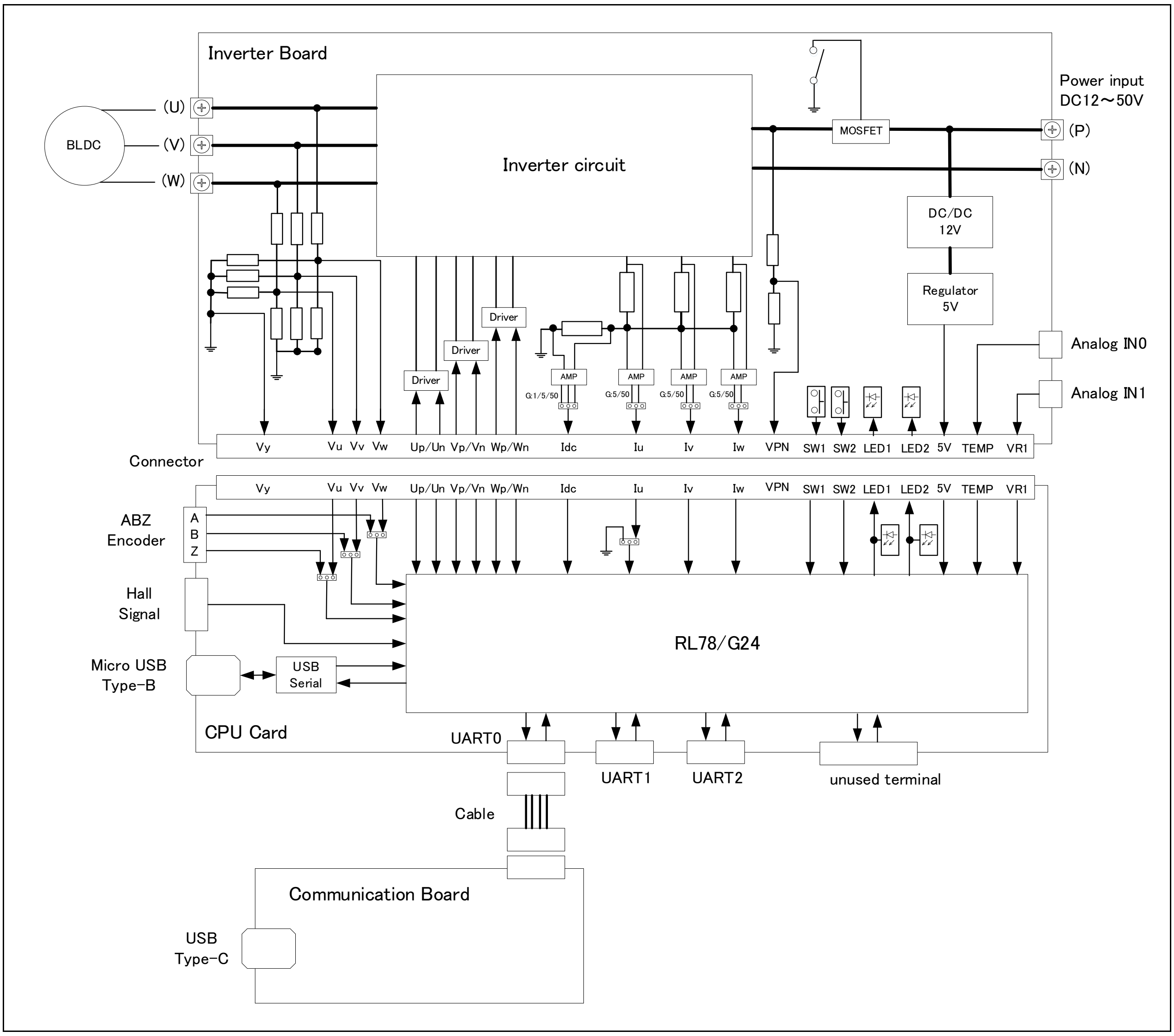


Figure 2-1 Hardware Configuration Diagram

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 system's connector interfaces 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 sensor signal input (not used in this system)
	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 system's Jumper setting are 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	W-Phase current [Used in 3-shunt mode]
Inverter board	JP7	5-6 Short	DC link current gain = 50x
	JP8	2-3 Short	U-phase current gain = 50x [Used in 3-shunt mode]
	JP9	2-3 Short	V-phase current gain = 50x [Used in 3-shunt mode]
	JP10	2-3 Short	W-phase current gain = 50x [Used in 3-shunt mode]

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

Table 2-8 – Port Interfaces

R7F101GLGFB port name	Function
P26 / ANI6	Inverter bus voltage detection
P147 / ANI18	DC link current detection
P20 / ANI0 / AVREFP	Analog reference voltage plus
P21 / ANI1 / AVREFM	Analog reference voltage minus
P22 / ANI2	U-phase current detection [Used in 3-shunt mode]
P24 / ANI4	V-phase current detection [Used in 3-shunt mode]
P23 / ANI3 / PGAGND	W-phase current detection [Used in 3-shunt mode]
P42	LED1 ON / OFF control
P43	LED2 ON / OFF control
P70 / TRDIOB0	PORT output / PWM output (U_p)
P72 / TRDIOA1	PORT output / PWM output (V_p)
P74 / TRDIOB1	PORT output / PWM output (W_p)
P71 / TRDIOD0	PORT output / PWM output (U_n)
P73 / TRDIOC1	PORT output / PWM output (V_n)
P75 / TRDIOD1	PORT output / PWM output (W_n)
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 sensor input(HU) (not used in this system)
P53 / INTP2	Hall sensor input(HV) (not used in this system)
P54 / INTP3	Hall sensor input(HW) (not used in this system)
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 (AD)	• Inverter bus voltage detection • 3-phase current detection [Used in 3-shunt mode] • DC link current detection [Used in 1-shunt mode]
Timer Array Unit (TAU)	• 1 [ms] interval timer • Periodic timer for serial communication during initial position detection
Timer RD2 (TRD2)	• PWM output using extended complementary PWM mode • 50[us] timer
PWM option unit A (PWMOPA)	• Forced shut-off of PWM output depending on CMP3 output
Timer RX(TRX)	• Current rising period measurement for initial position detection
Comparator(CMP3)	• Overcurrent detection • Current judgement for initial position detection
10bit D/A converter (DA)	
Watch dog timer (WDT)	• Program runaway detection

(1) 12-bit A/D converter (AD)
The U-phase current (Iu), W-phase current (Iw), and inverter bus voltage (Vdc) are measured using the 12-bit A/D converter. [Used in 3-shunt mode]
DC link current and inverter bus voltage (Vdc) are measured using the 12-bit A/D converter.
[Used in 1-shunt 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)
Uses 1 [ms] interval timer.
In addition, a periodic interrupt is generated for serial communication during initial position detection.

(3) Timer RD2 (TRD2)
AD conversion can be executed at any timing using two AD conversion trigger compare registers.
Asymmetric waveforms can be output using the extended compare register.

(4) 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.

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

(6) Comparator (CMP3)
Overcurrent detection or initial position detection by comparing to the internal reference value.

(7) 10bit D/A converter (DA)
Comparator (CMP3) is used for overcurrent or initial position detection. For the threshold value to be compared with the corresponding current value, input the reference value from the 10bit D/A converter.

(8) Watch Dog timer (WDT)
Watch Dog timer is used as program runaway detection.

2.4 Software structure

2.4.1 Software file structure

The folder and file configurations of the sample programs are given in Table 2-10 below.

Table 2-10 Folder and File Configurations of the Sample Programs

Folder		File	Content
config		r_mtr_config.h	Configuration definition
		r_mtr_motor_parameter.h	Motor parameter definition
		r_mtr_control_parameter.h	Control parameter definition
		r_mtr_inverter_parameter.h	Inverter parameter definition
		r_mtr_scaling_parameter.h	Scaling parameter definition
application	main	main.h main.c	Main function
	board	r_mtr_board.h r_mtr_board.c	Function definition for hardware UI Hardware UI
	ics	r_mtr_ics.h r_mtr_ics.c	Function definition for Analyzer(Note1) UI Analyzer(Note1) UI
		ICS_define.h	CPU definition for RMW
		RL78_vector.h RL78_vector.c	Interrupt vector function processing 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 Handling of unused pins
smc_gen		-	Smart Configurator generate file
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	Motor control parameter definition
		r_mtr_ctrl_gain .h r_mtr_ctrl_gain.obj	Gain design function definition Gain design
		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_foc_less_speed.h r_mtr_foc_less_speed.c	Sensorless vector control-related function definition Sensorless vector control-related
		r_mtr_interrupt.c	Interrupt handler function definition
		r_mtr_est_phase_err.h r_mtr_est_phase_err.obj	Phase error estimating function definition Phase error estimation
		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.obj	Open loop to Closed loop switch control function definition Open loop to Closed loop switch control
		r_mtr_disturb_suppress.h r_mtr_disturb_suppress.obj	Disturbance suppression function definition Disturbance suppression
		r_mtr_foc.h r_mtr_foc.c	Sensorless vector control function definition Sensorless vector control
		r_mtr_mtpa2fw_ctrl.h r_mtr_mtpa2fw_ctrl.obj	Field weakening control/maximum torque control function definition Field weakening control/maximum torque control
		r_mtr_offset.h r_mtr_offset.c	Current offset detection function definition Current offset detection
		r_mtr_prep.h r_mtr_prep.c	Checking of drive stop function definition Checking of drive stop

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 the module configuration of the sample programs.

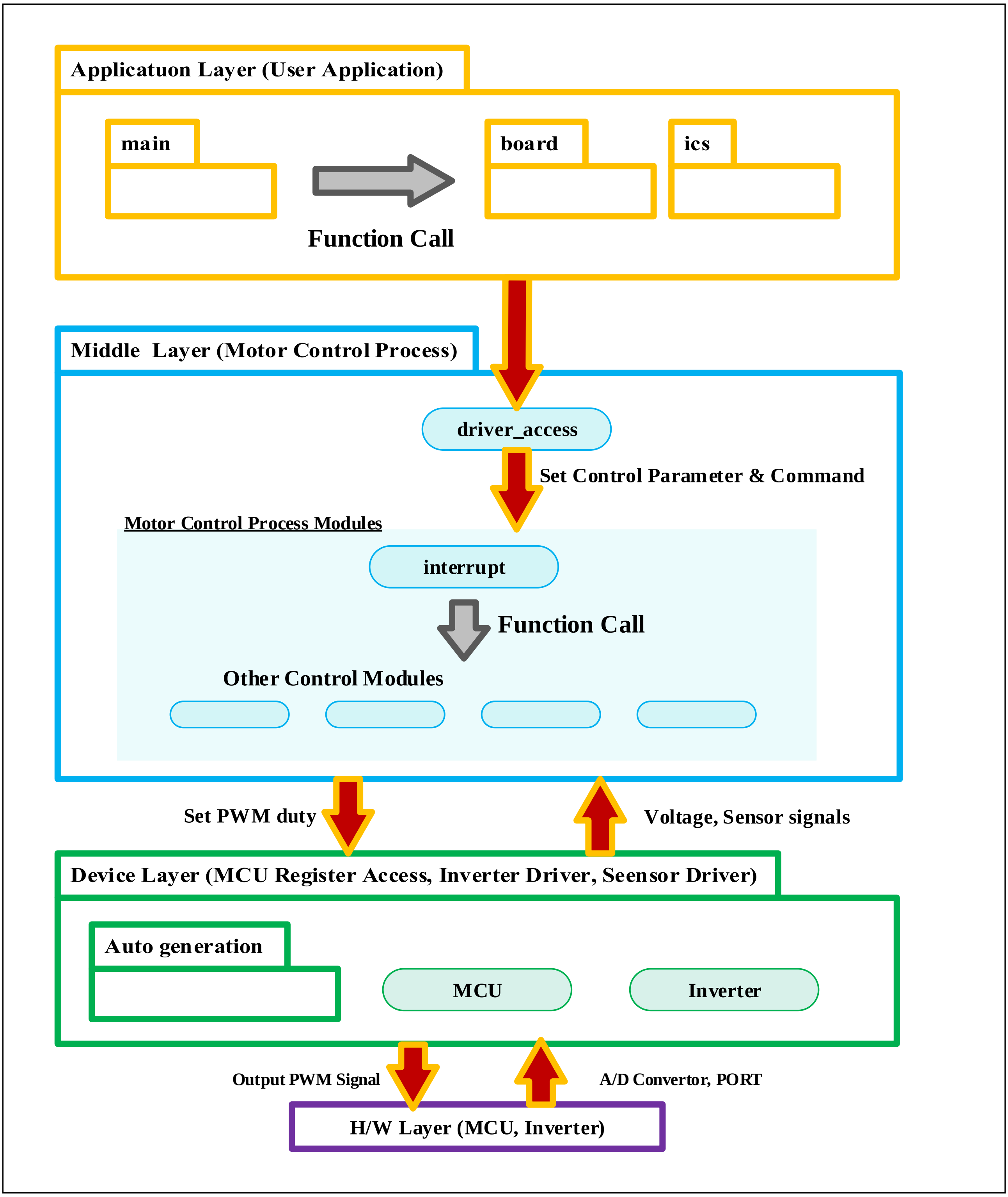


Figure 2-2 Module Configuration of the Sample Programs

2.5 Software specifications

The basic software specifications of the system are listed below.

Table 2-11 Software Specifications

<table>
<tr><th>Item</th><th colspan="2">Content</th></tr>
<tr><td>Control method</td><td colspan="2">Vector control</td></tr>
<tr><td>Current detection method</td><td colspan="2">Use the compile switch to switch between the method of detecting DC link current (1-shunt mode) and the method of detecting 3-phase current (3-shunt mode).
Switch 1-shunt mode or 3-shunt mode with the compile switch
(Default is 1-shunt mode)</td></tr>
<tr><td>Motor rotation start/stop</td><td colspan="2">Operation using the motor control development support tool[Note]</td></tr>
<tr><td>Position detection of rotor magnetic pole (Sensorless)</td><td colspan="2">Angle estimation using induced voltage
Initial position detection using motor polarity and magnetic saturation characteristics</td></tr>
<tr><td>Input voltage</td><td colspan="2">DC 24 [V]</td></tr>
<tr><td>Main clock frequency</td><td colspan="2">CPU clock: f_{CLK} 48 [MHz]
TRD clock: f_{HOCO} 96 [MHz]</td></tr>
<tr><td>Carrier frequency (PWM)</td><td colspan="2">20 [kHz]</td></tr>
<tr><td>Deadtime</td><td colspan="2">1 [µs]</td></tr>
<tr><td>Control cycle</td><td colspan="2">Current control location/speed estimate: 100 [µs] (twice the carrier cycle)
Speed control: 1 [ms]</td></tr>
<tr><td>Rotational speed range</td><td colspan="2">CW: 0 [rpm] - 3975 [rpm]
CCW: 0 [rpm] - 3975 [rpm]
However, driving is performed as an open loop at 795 [rpm] or less</td></tr>
<tr><td>Optimal setting</td><td colspan="2">-Olite</td></tr>
<tr><td rowspan="2">ROM/RAM Size</td><td>ROM</td><td>1-shunt mode : 31.647 KB 3-shunt mode : 29.517 KB</td></tr>
<tr><td>RAM</td><td>1-shunt mode : 1.830 KB 3-shunt mode : 1.726 KB</td></tr>
<tr><td>Processing stop for protection</td><td colspan="2">- Disables the motor control signal output (six outputs), under any of the following conditions.
1. Detects a value exceeding the inverter bus voltage of 28 [V]
2. Detects a value less than the inverter bus voltage of 12 [V]
3. Detects a value exceeding the rotation speed of 5590 [rpm] (1 shunt mode) / 4290 [rpm] (3 shunt mode)
4. Detects a value where the current of each phase exceeds 1.47 [A]
5. Detects a value where the current of each phase exceeds 4.5 [A](Forced shut-off for PWMOPA)
6. Detects TRX counter overflow during initial position detection processing
- When an overcurrent detection signal (CMP3) is detected, the PWM output port is made low-level (using PWMOPA).</td></tr>
</table>

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

2.6 User option bytes

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

Table 2-12 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. Explanation of Control Programs

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

3.1 Contents of control

3.1.1 Motor start/stop

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

3.1.2 A/D converter

(1) Inverter bus voltage

The inverter bus voltage is measured as given in Table 3-2. It is used for modulation factor calculation and over- and undervoltage detection. (When an abnormality is detected, PWM is stopped).

Table 3-1 Inverter Bus Voltage Conversion Ratio

Item	Conversion ratio (Inverter bus voltage: A/D conversion value)	Channel
Inverter bus voltage	0 [V] – 111 [V]: 0000H – 0FFFH	ANI6

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

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

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

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

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

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

Table 3-3 Conversion Ratios of 3-Phase Currents

Item	Conversion ratio (3-phase currents: A/D conversion value)	Channel
U-phase, V-phase and W-phase current	-5 [A] – 5 [A]: 0000H – 0FFFH	Iu: ANI2 Iv: ANI4 Iw: ANI3

[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 to detect overcurrent and judgment of the current threshold value for initial position detection.

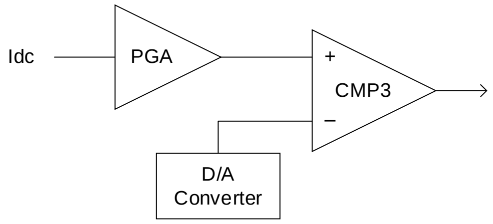


Figure 3-1 Overcurrent detection by CMP3

3.1.4 Voltage control by PWM

PWM control is used for controlling output voltage. PWM control is a control method that continuously adjusts the average voltage by varying the pulse duty, as shown in Figure 3-2.

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

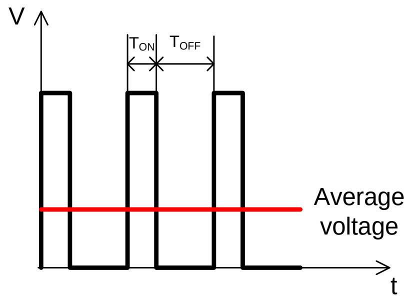


Figure 3-2 PWM Control

Here, the modulation factor *m* is defined as follows.

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

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

3.1.5 Modulation

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

(1) Triangle Wave Comparison Method

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

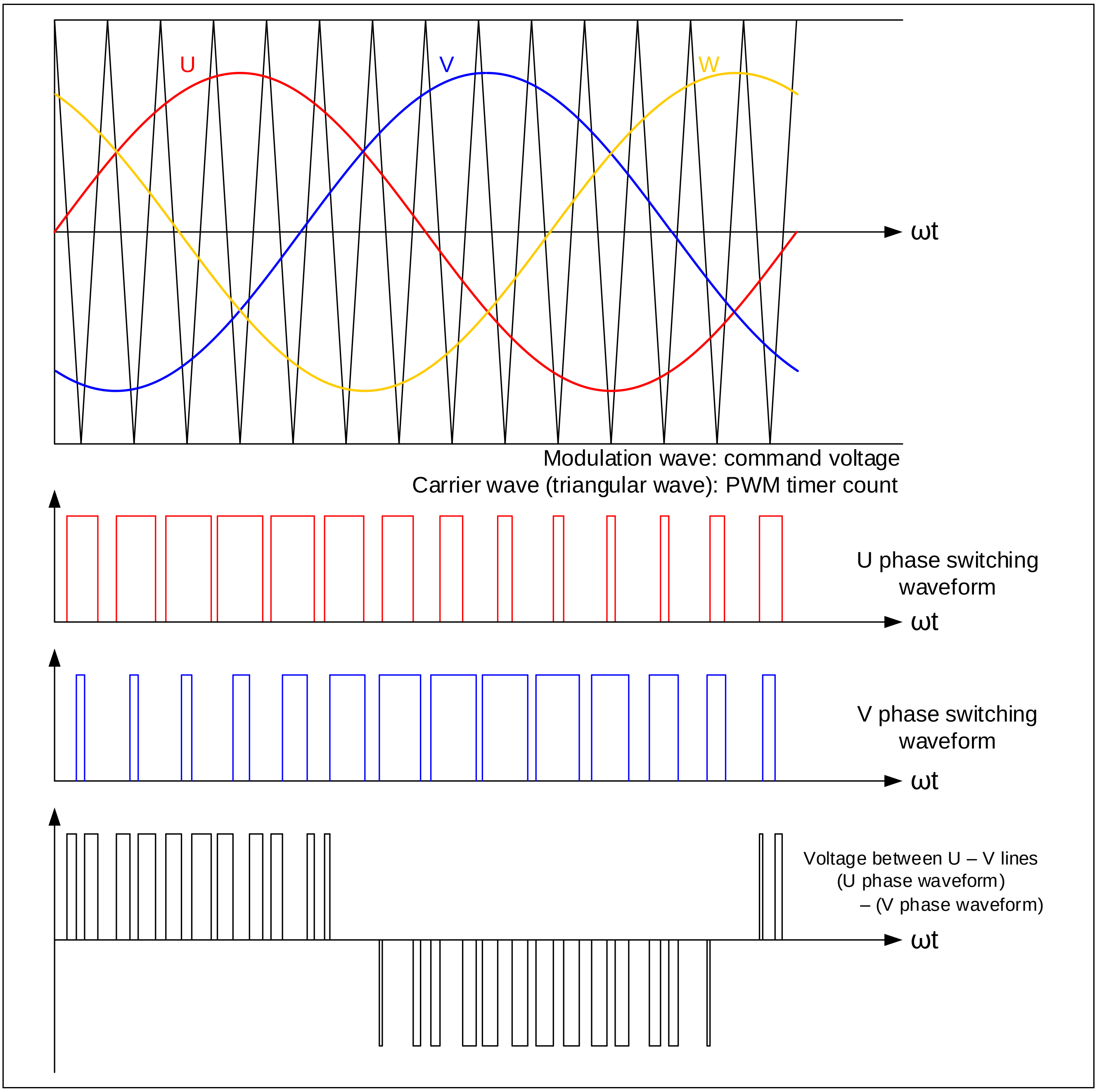


Figure 3-3 Conceptual Diagram of the Triangle Wave Comparison Method

(2) Third Harmonic Injection Method

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

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

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

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

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

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

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

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

(3) 2-phase Modulation Method

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

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

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

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

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

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

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

v_o : offset voltage

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

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

3.1.6 State transitions

The state transition diagram for this program is shown in Figure 3-4.

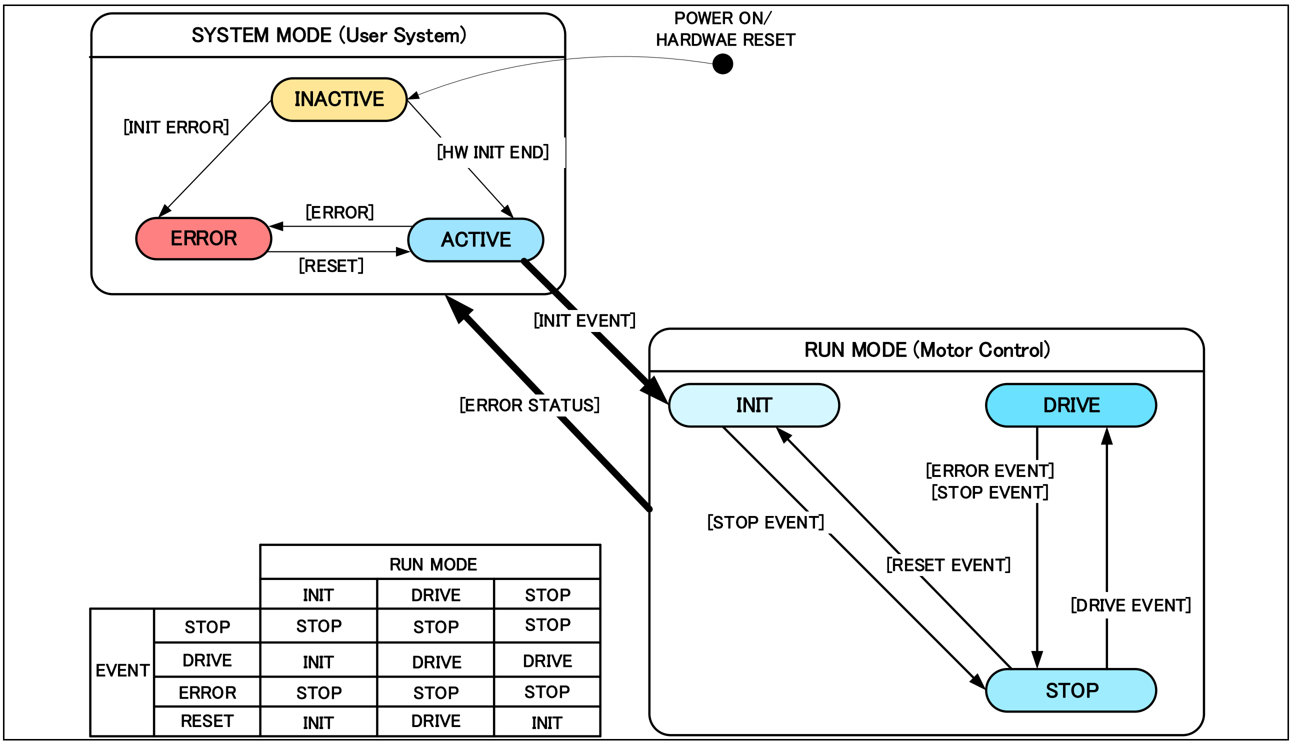


Figure 3-4 State Transition Diagram

(1) SYSTEM MODE

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

(2) RUN MODE

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

(3) EVENT

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

Table 3-4 EVENT List

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

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

Table 3-5 DRIVE status List

Status name	Content
MTR_OFFSET_CALC_EXE	Execution of current offset detection process
MTE_OFFSET_CALC_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_START	Motor starting operation (open loop drive)
MTR_DRIVE_ID_ZERO	Closed loop drive
MTR_DRIVE_BRAKE	Brake (Unimplemented)
MTR_DRIVE_END	Drive stop

3.1.7 Startup method

The description of startup control of the sensorless vector control software is shown in Figure 3-5. The mode is controlled by the states that control the reference values of the d-axis current, q-axis current, and speed.

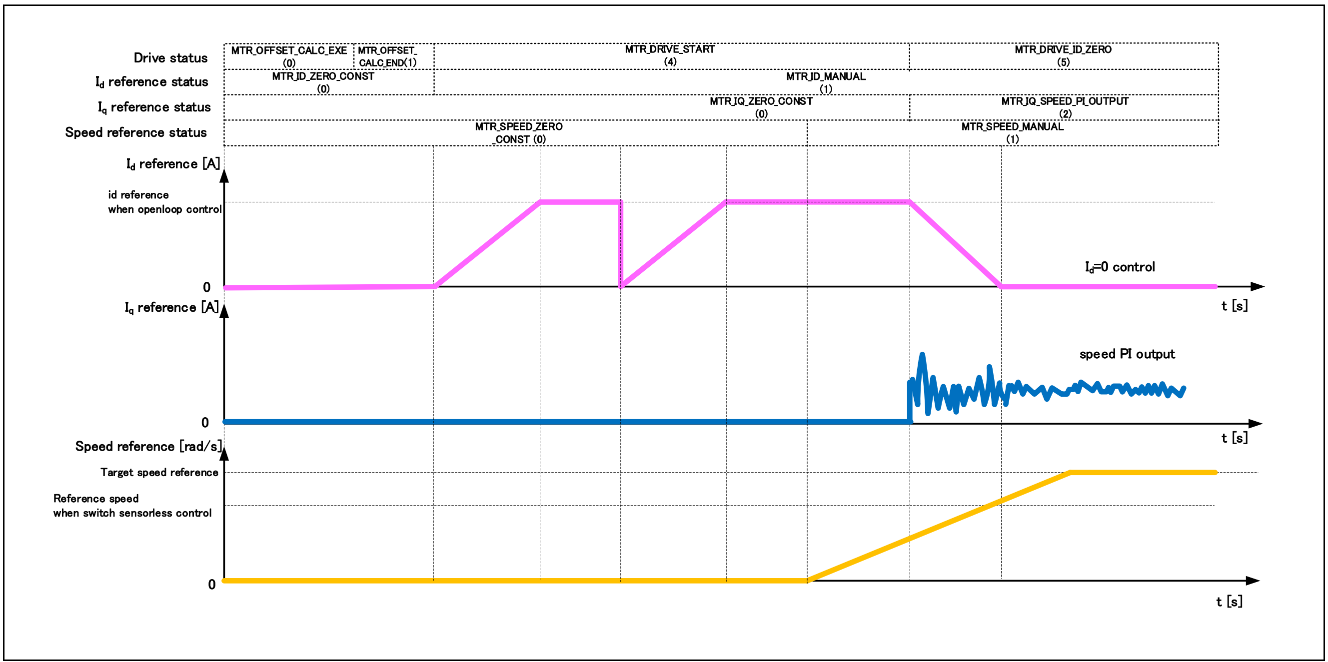


Figure 3-5 Description of Sensorless Speed Control Software Startup Control

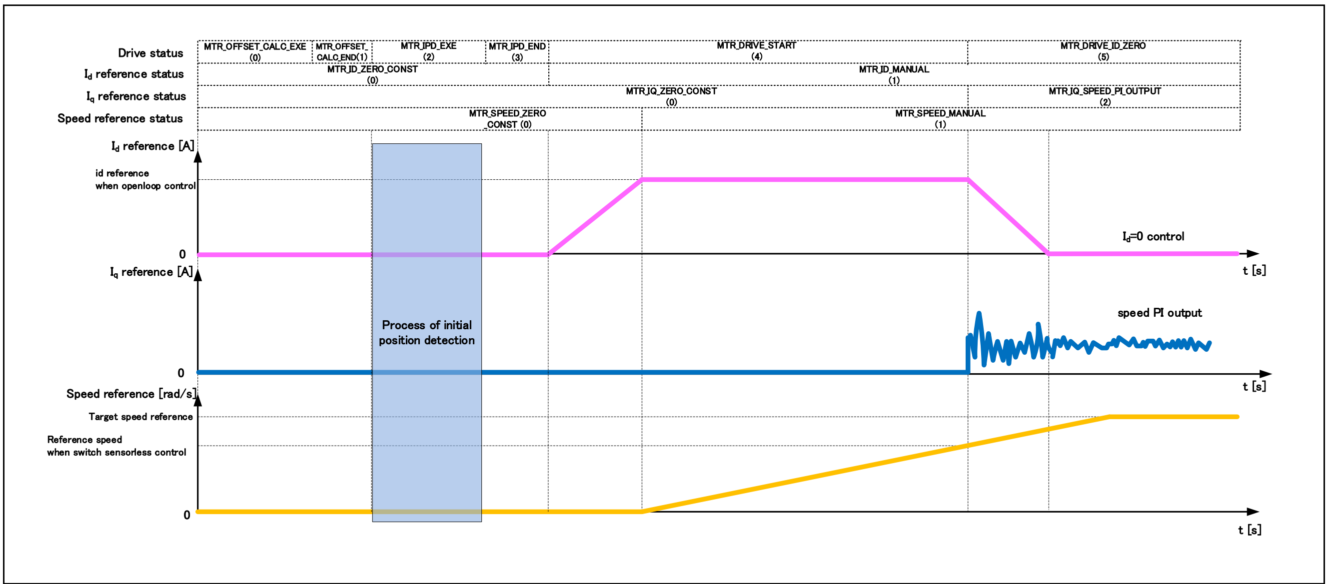


Figure 3-6 Description of Sensorless Speed Control Software Startup Control (IPD processing application)

3.1.8 Control method

The block diagram of the entire control system is shown in Figure 3-7. The control system is made up of a coordinate converter and decoupling controller, phase error estimator, PLL controller, Auto Speed Regulator (ASR), and Auto Current Regulator (ACR). Also, a primary LPF is set up for the estimated speed and δ-axis current in order to prevent phase error estimate pulsations, etc. due to factors such as disturbances.

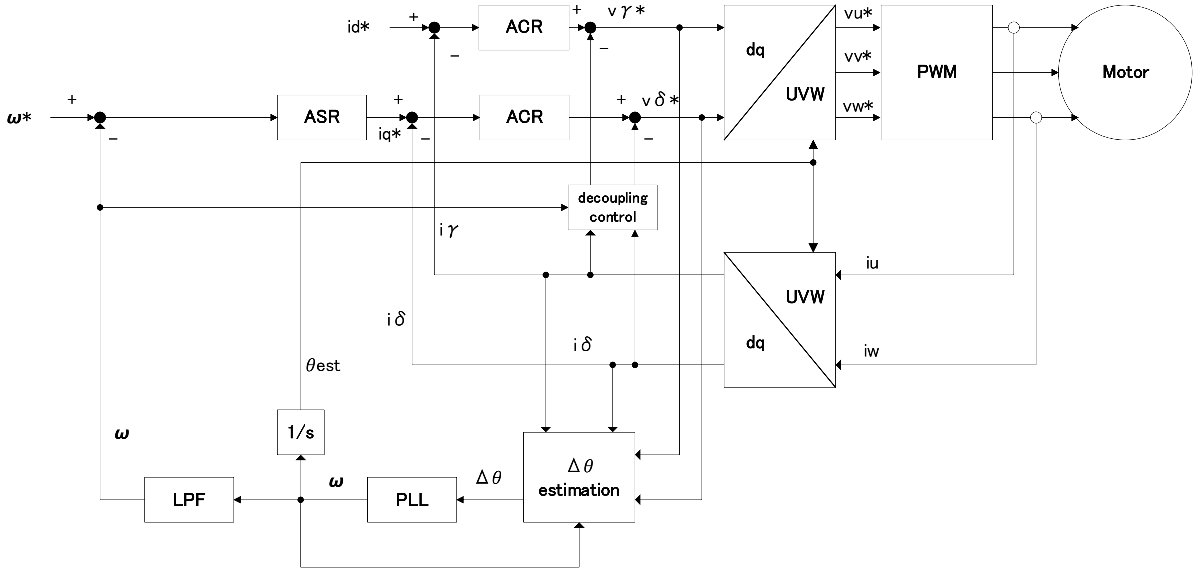


Figure 3-7 Control System Block Diagram

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

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

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

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

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

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

The BEMF constituent elements e_γ $e_{I3}$and e_δ of the γ-axis and δ-axis are calculated, and the phase error $\Delta\theta$ is obtained using the following equation. When calculating the BEMF, the γ-axis voltage and the δ-axis voltage are each used in approximation with reference voltage values v_γ^* and v_δ^*. The estimated speed ω is determined by constructing a feedback loop so that this phase error reaches 0 (PLL controller).

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

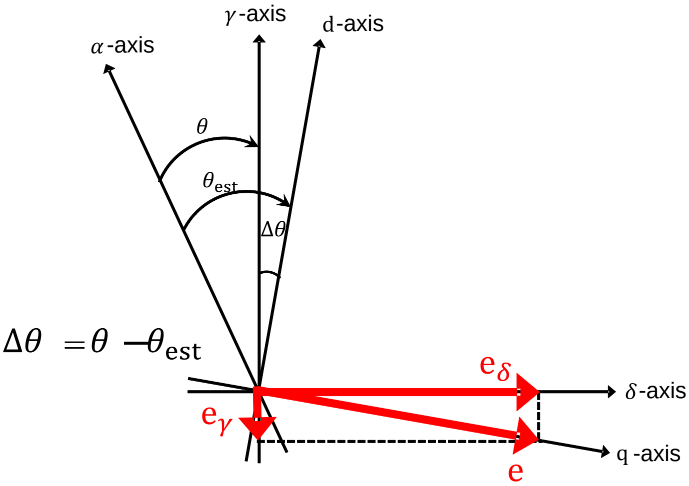


Figure 3-8 Phase Error and γ- and δ-axis Induced Voltage Constituent Elements

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

$$K_{p_{ACR}} = \omega_{CG} L, \qquad K_{i_{ACR}} = \omega_{CG} R$$

ω_{CG}: Current PI control natural frequency

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

R: Resistance

$$K_{p_{ASR}} = \frac{\omega_{SG} J}{P_n^2 \psi_a}, \qquad K_{i_{ASR}} = \frac{\omega_{SG} D_1}{a P_n^2 \psi_a}$$

ω_{SG}: Speed PI control natural frequency

J: Inertia

D_1: Kinetic friction coefficient

ψ_a: BEMF coefficient

P_n: Number of pole pairs

α: Adjustment parameters

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

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

3.1.9 System protection function

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

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

- Overcurrent error
3-phase current are monitored in the 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.

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

Table 3-6 System Protection Function Settings

Kinds of error	Threshold	
Overcurrent error for hardware	Overcurrent limit [A]	4.5
Overcurrent error	Overcurrent limit [A]	1.47
	Monitoring cycle [μs]	100
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 (1 shut mode) / 4290 (3 shunt mode)
	Monitoring cycle [μs]	1000

3.1.10 Per-unit method (PU)

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

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

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

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

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

Table 3-7 PU system base values

Category	Item	Definition	Unit
PU base physical quantity	Current	Rated current	[A]
	Voltage	Input voltage (inverter input)	[V]
	Angular frequency	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]
	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	[A/(rad/s)]
	Kidt	Current / angular frequency	[A/(rad/s)]
PLL control	Kp	Angular frequency / angle	[Hz]
	Kidt	Angular frequency / angle	[Hz]

3.1.11 Current measurement method with 1 shunt resistor

Current measurement is performed using 1 shunt resistor.

3.1.11.1 Timing to measure the 1-shunt resistor current

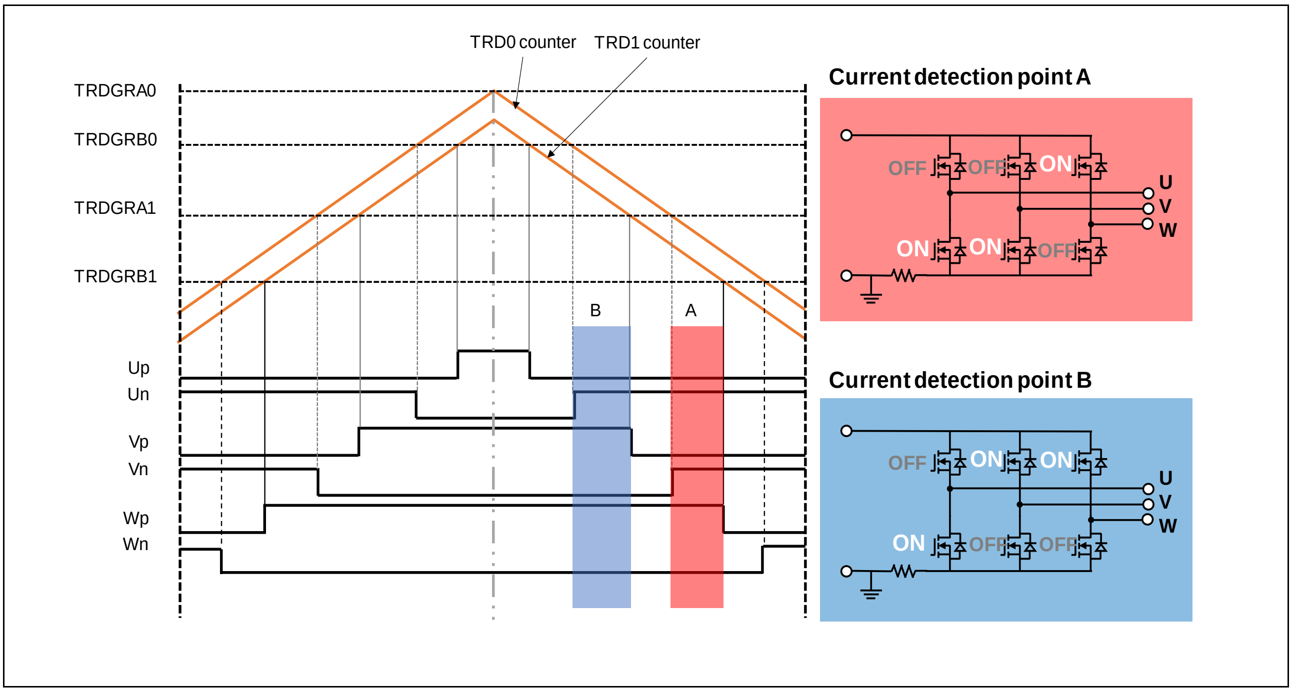


Figure 3-9 Complementary PWM waveform (For example, duty magnitude relation W > V > U).

In this program, the timer RD2 extended complementary PWM mode is used to control by 3-phase PWM output with dead time. Figure 3-9 shows complementary PWM waveforms (For example, duty magnitude relation W > V > U).

At point A in the figure, only the W-phase of the upper arm is ON as shown in the red frame on the upper right. In this case, the current flowing through the shunt resistor is the current flowing through the W-phase.

$$I_A = I_w$$

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

$$I_B = -I_u$$

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

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

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

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

3.1.11.2 How to measure 1-shunt resistor current using RL78/G24 functions

When current measurement by a 1-shunt resistor is performed as shown in 3.1.8.1, the conversion timing of the A/D converter must be adjusted according to the PWM duty setting. The program achieves this by using the following RL78/G24 functions:

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

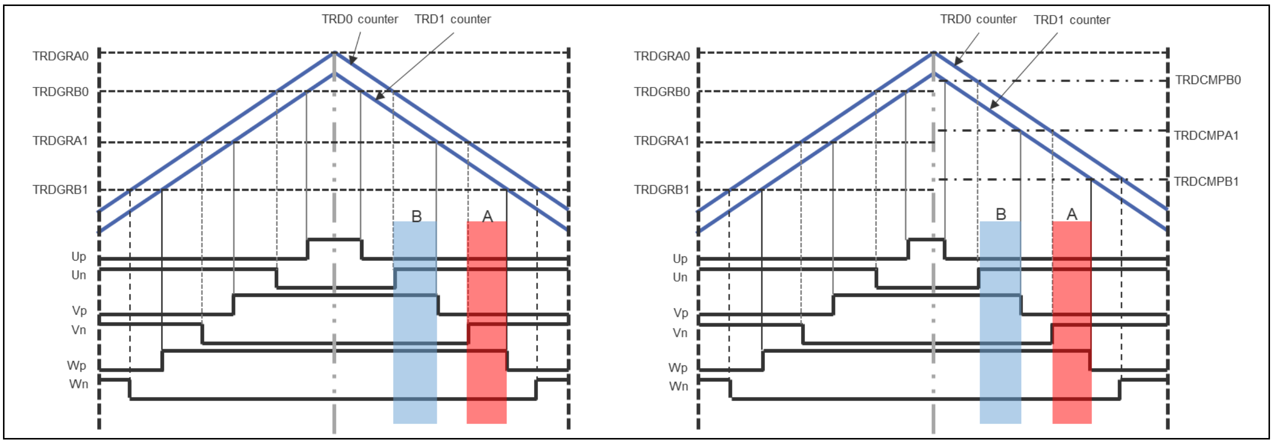


Figure 3-10 Timing of A/D Conversion

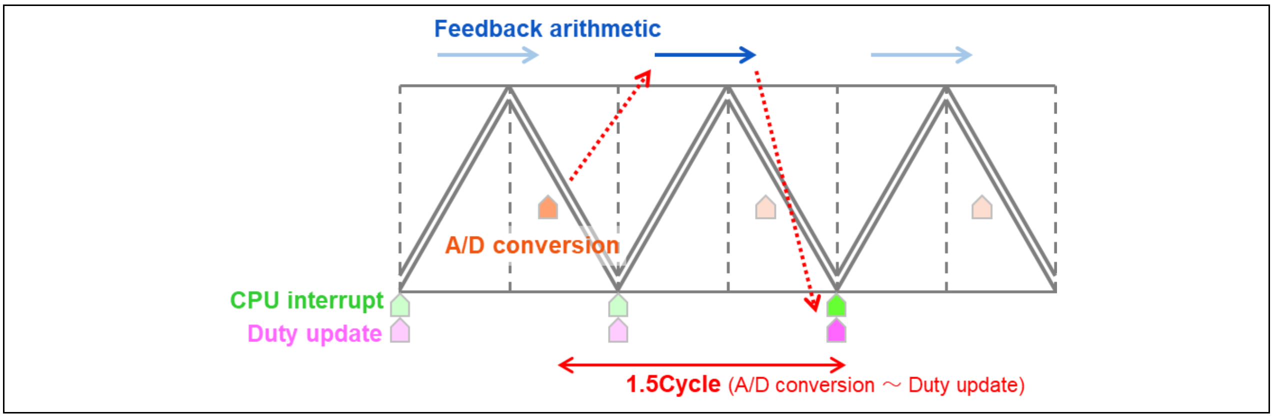


Figure 3-11 Timing of interrupt

The description in this section is based on Figure 3-10 and Figure 3-11 .
The Timer RD2 sets an interrupt to be generated at the valley of PWM carrier, where the duty, and the count value for the timings of A and B for the next cycle are calculated. For A/D conversion trigger timings, the left edge of the point A and B are set to the A/D conversion trigger buffer register 0/1 of Timer RD2 respectively.

The A/D converter performs A/D conversion using A/D conversion trigger 0/1 of Timer RD2 as a hardware trigger. The A/D conversion result is used for duty and the count value for the timings of A and B calculation at the next cycle, in which the calculated result is reflected at the valley of the next cycle (1.5 cycle later).

It’s possible for 3-phase current to be restored from A/D conversion value of two points which are set in A/D conversion resister (ADCRn). If the PWM waveform cannot be kept constant during the AD conversion time of two points, the AD conversion time is secured by outputting an asymmetric three-phase waveform in extended complementary PWM mode.

3.1.11.3 Duty adjustment

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

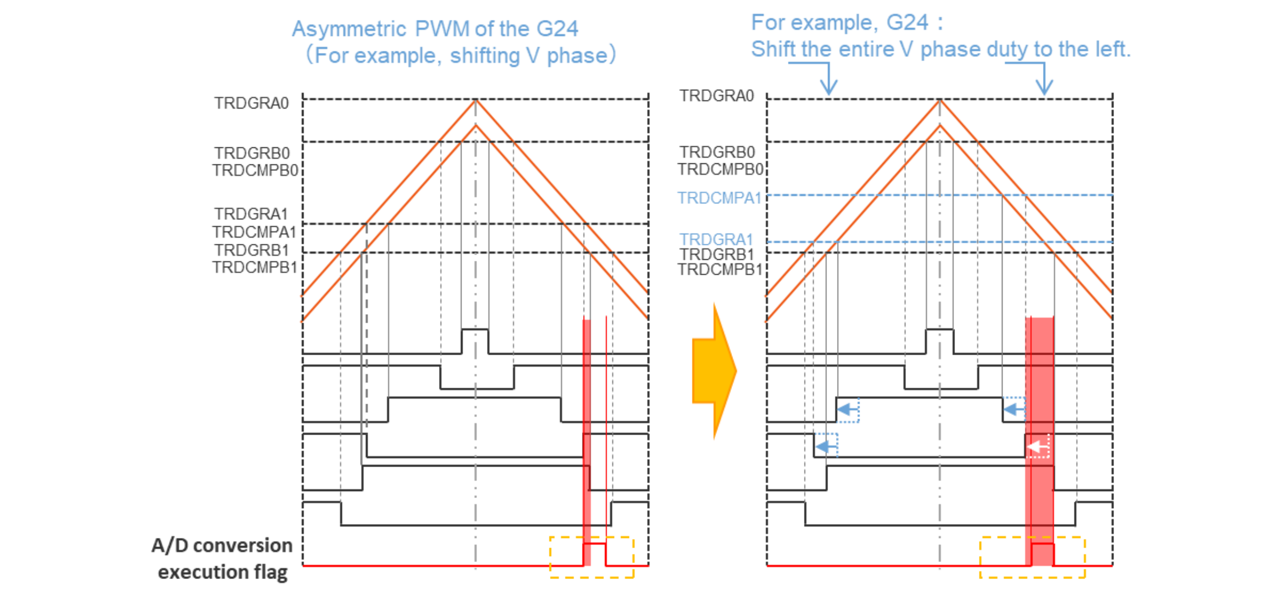


Figure 3-12 Duty adjustment (example)

3.1.12 Initial position detection by saliency

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 3-13, 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 3-14. It therefore takes longer when voltage is applied in the V→W direction than when voltage is applied in the W→U direction.

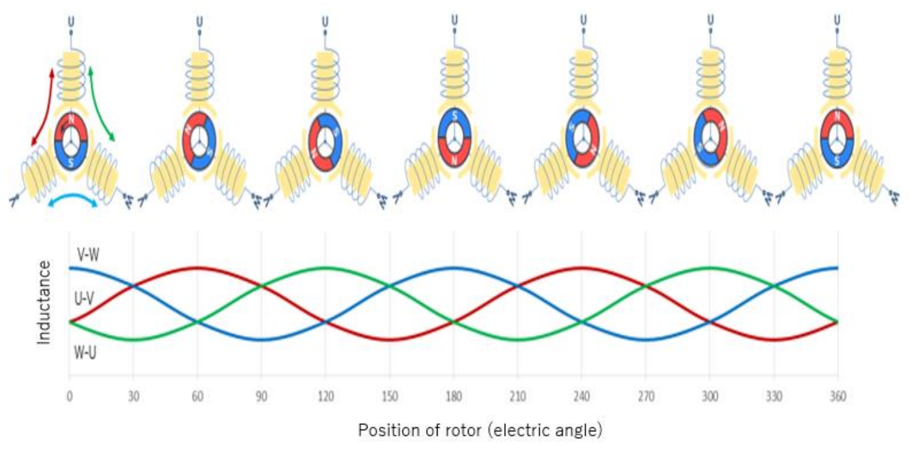


Figure 3-13 Changes in inductance according to the position of the rotor

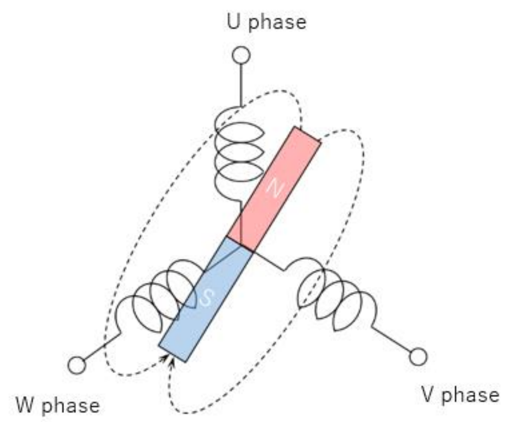


Figure 3-14 Relationship between rotor position and each phase

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 3-15. 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.

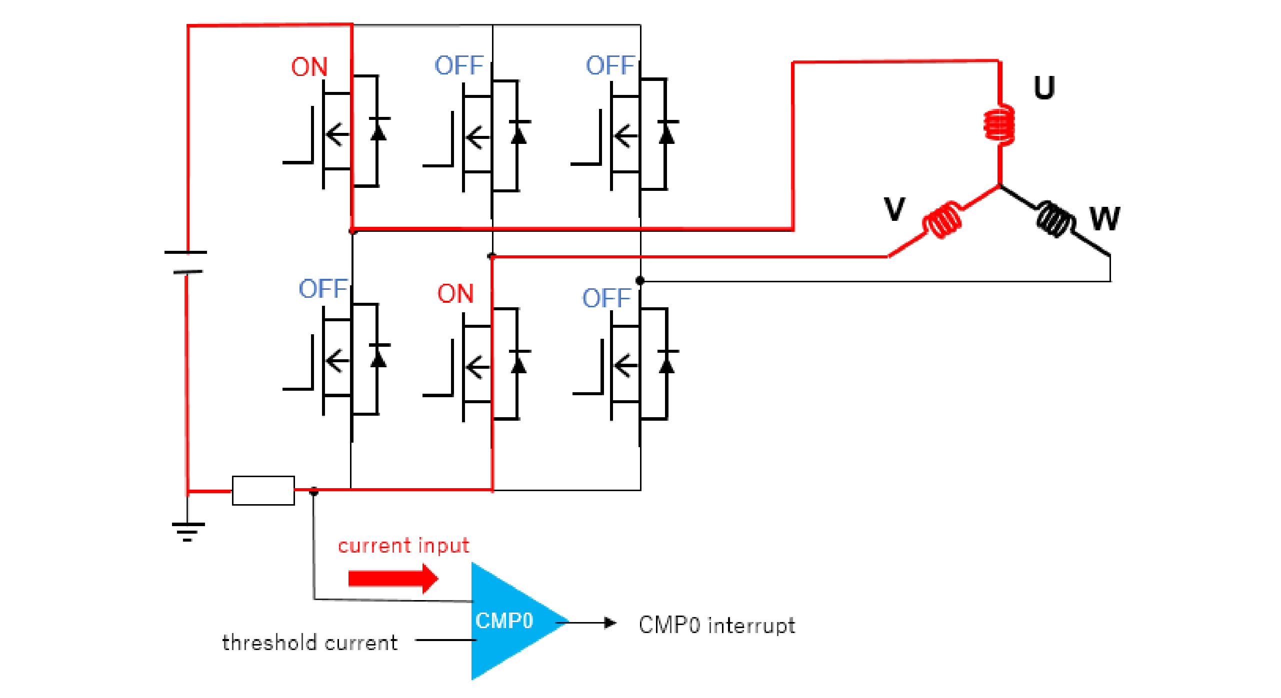


Figure 3-15 Angle detection diagram

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.

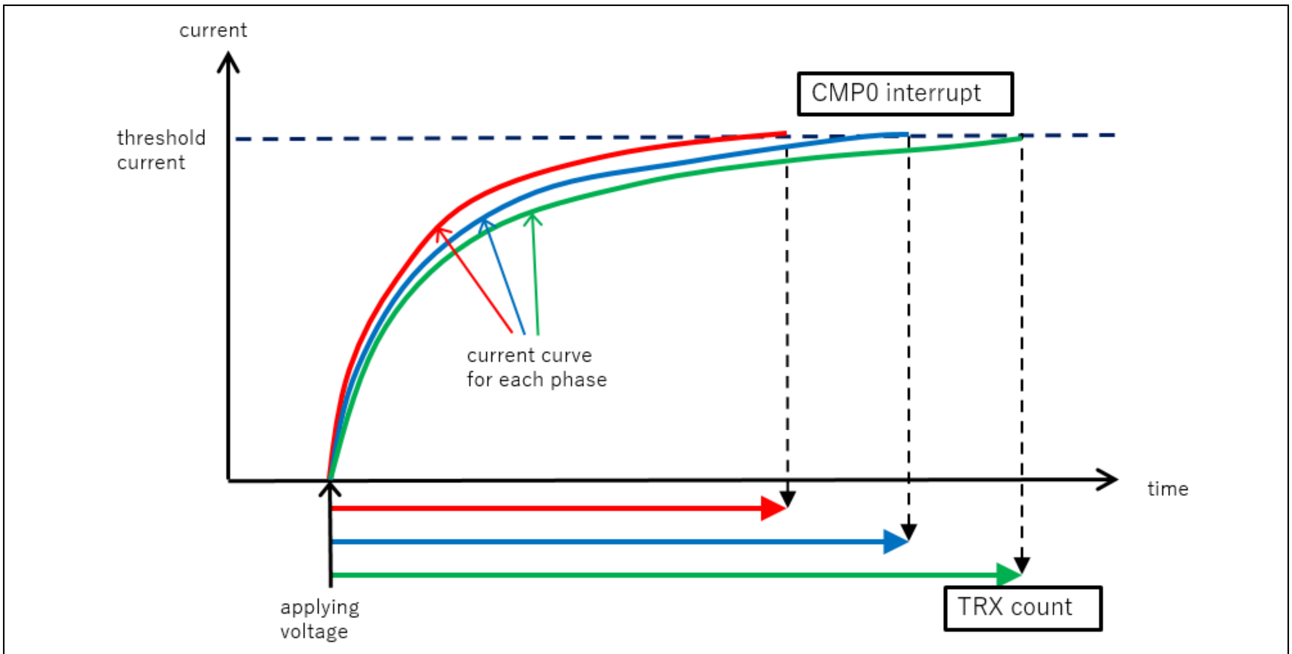


Figure 3-16 Current detection time differential among the 3-Phases

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 3-17, 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.

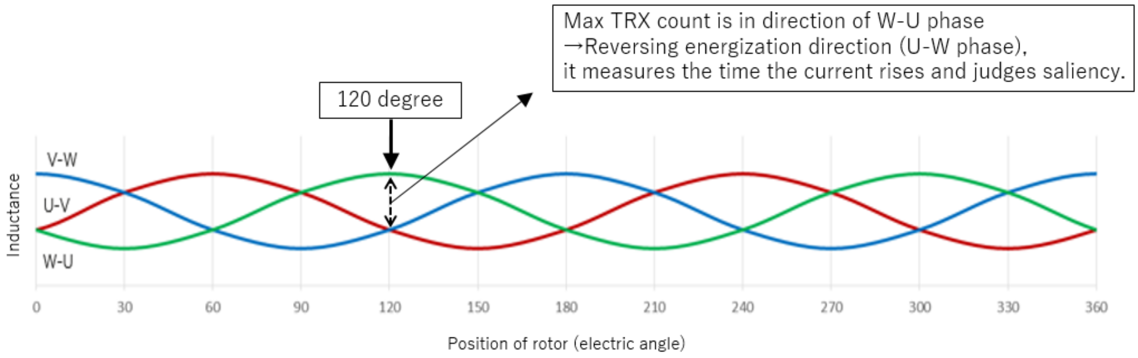


Figure 3-17 Saliency confirmation method

3.1.13 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.

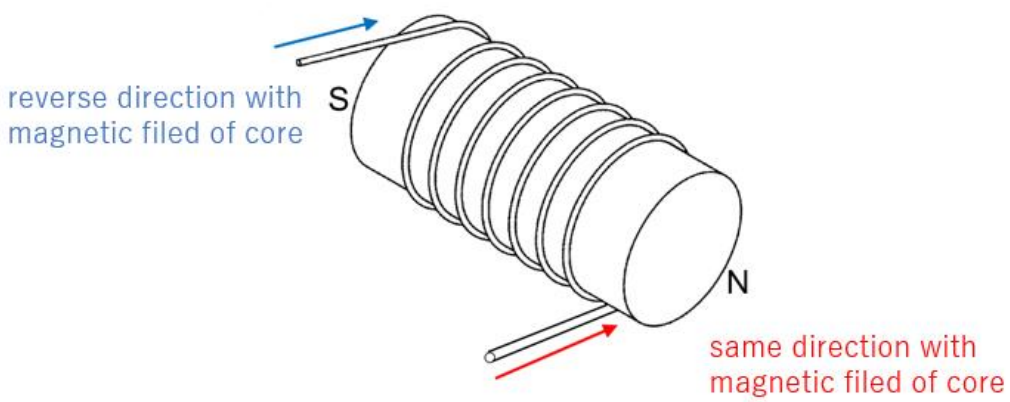


Figure 3-18 Example of magnetic pole wound with coil

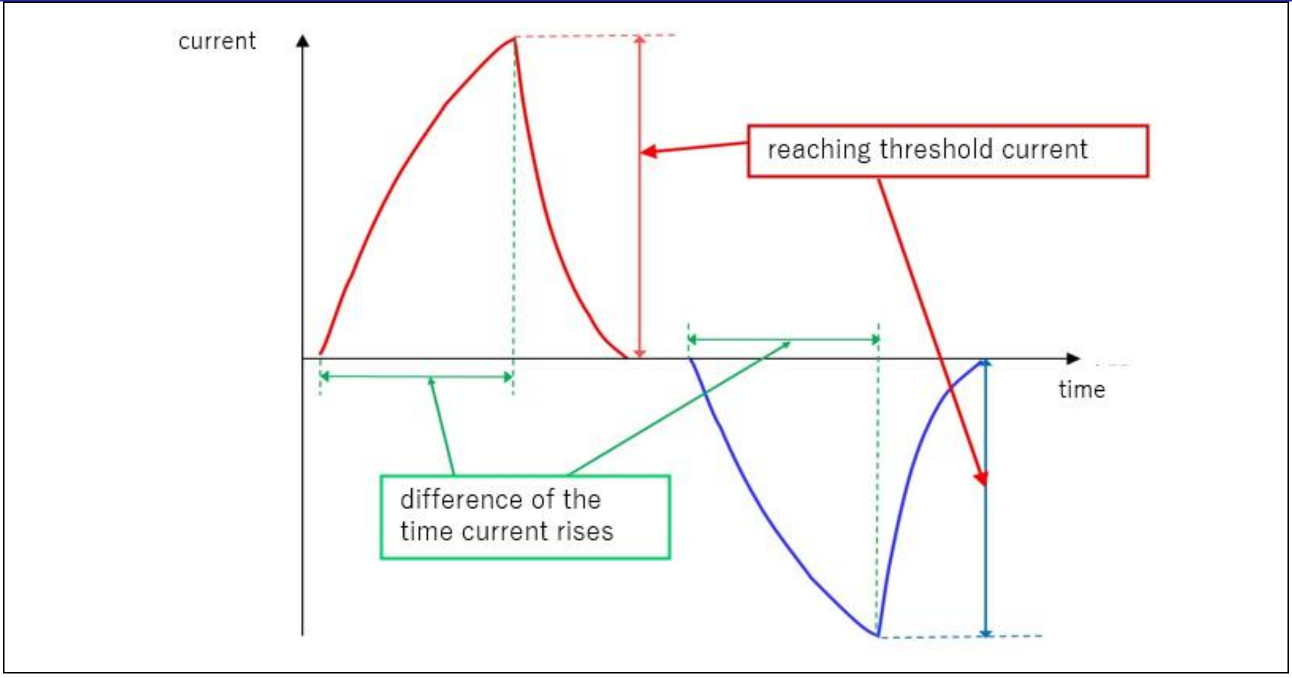


Figure 3-19 Current differential according to direction of applied current

Voltage is applied to the motor as shown in Figure 3-19, 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.

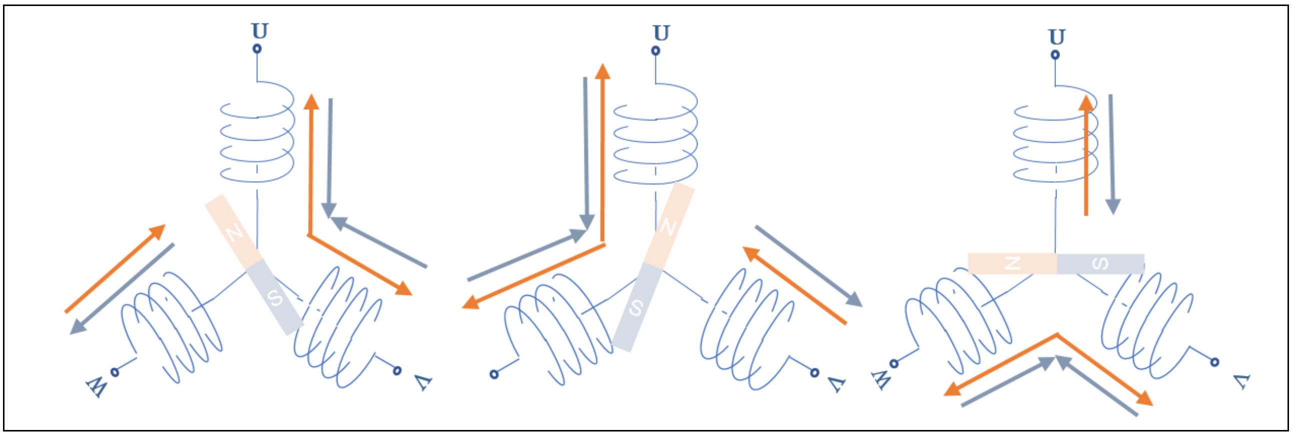


Figure 3-20 Example of applied voltage pattern due to initial position detection using magnetic saturation

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.

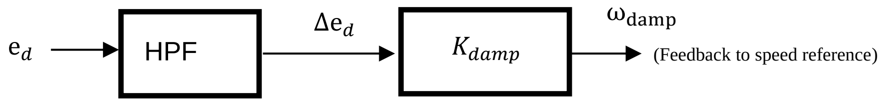
The initial position detection process to be performed changes depending on the value of the preparation method for start-up (PS_METHOD). Table 3-8 shows the processes executed in each mode.

Table 3-8 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	×	×	×	×

3.1.14 Open loop Control

Position estimation is not possible because the BEMF is small in the low-speed range. Therefore, a rotating magnetic field is generated using the d-axis current to force the motor to be driven synchronously to a speed at which position estimation is possible (open loop control). At that time, the speed of the motor oscillates at a natural frequency that depends on the current and motor parameters. Therefore, by performing damping control as shown in the block diagram in Figure 3-21, the vibration of the motor during open loop control in the low-speed range is reduced.



e_d: Estimated d-axis BEMF [V]
Δe_d: Vibration component of estimated d-axis BEMF [V]
K_{damp}:Damping control gain
ω_{damp}:Feedback speed for speed reference [rad/s (electrical angle)]

Figure 3-21 Block diagram of the open loop damping control

3.1.15 Open loop to closed loop switch control

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

3.1.16 Field weakening control/Maximum torque control

The BEMF increases as the rotation speed increases. For this reason, the reference current cannot be output adequately in the range where the BEMF is the same magnitude as the power supply voltage. There is a method, in which negative current is applied to the d-axis to simulatively reducing the BEMF, thereby expanding the drivable area (Field weakening control). Additionally, for motors with saliency, there is a method that uses reluctance torque to increase the output torque (Maximum torque control).

3.1.17 Disturbance suppression

Since the speed fluctuates due to disturbance, the influence can be reduced by estimating the disturbance and feeding forward the value. If the current control is functioning satisfactorily, the disturbance can be estimated from the inverse model of the motor, so the estimated value is multiplied by the LPF and reflected in the reference current to reduce the disturbance.

3.2 Sensorless Vector Control Software Function Specification

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

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

File	Function	Process overview
main.c	main Input: none Output: none	- Call hardware initialization function - Renesas Motor Workbench Communication Initialization - Call user interface initialization function - Call main processing use variable initialization function - Call state transition and event execution function - Call bus voltage stability waiting process - Main process ⇒Call user interface process ⇒Call watchdog timer clear function
	ics_ui Input: none Output: none	Uses Renesas Motor Workbench - Motor status change
	software_init Input: none Output: none	Initialization of variable used for main process

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

File	Function	Process overview
r_mtr_ics.c	R_MTR_SetIcs Input:none Output:none	Preparation for RMW communication
	R_MTR_SetCOMVariables Input: none Output: none	Preprocess to set control variables - Control variable rewrite Variable value (com variable) Input to control pass buffer variable (ICS variable) - input values of ICS variables to ICS buffer variables
	R_MTR_ICSVariablesInit Input: none Output: none	Initialization of com variables
	R_MTR_ICSIntLevel Input: uint8_t u1_level :: Priority of interrupt Output: none	Set ICS interrupt priority
	mtr_limit (inline function) Input: int16_t s2_value :: target value int16_t s2_max :: maximum value (int16_t) s2_min :: minimum limit Output: int16_t :: limited value	Limit between maximum and minimum values

Table 3-11 List of Functions in “ICS2_RL78G24.lib”

File name	Function	Processing overview
ICS2_RL78G24.lib	ics2_init argument: uint16_t addr :: DTC vector table start address uint16_t pin :: Pins used by SCI uint8_t level :: Interrupt level uint8_t num :: Top address of DTC structure uint8_t brr :: communication speed uint8_t mode :: Communication mode return: none	Communication initialization
	ics2_watchpoint argument: none return: none	Call transfer function Must be called at intervals of 300us or more.

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

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

Table 3-13 List of Functions in "r_mtr_rl78g24.c"

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

Table 3-14 List of Functions in "r_mtr_ctrl_gain.obj"

File	Function	Process overview
r_mtr_ctrl_gain.obj	R_MTR_CtrlGain Input: st_mtr_ctrl_gain_t *st_gain_buf :: Control Gain structure pointer const st_mtr_design_parameter_t *st_ctrl_param :: Design parameter structure pointer Output: none	Gain design process

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

File	Function	Process overview
r_mtr_driver_access.c	R_MTR_InitControl Input: none Output: none	Initialization of motor control system - initialization of motor status - initialization of control variables
	R_MTR_ExecEvent Input: uint8_t u1_event :: event Output: none	Change motor status and execute event process
	R_MTR_ChargeCapacitor Input: none Output: uint16_t :: timeout error	Waiting for stability of bus voltage
	R_MTR_SetSpeed Input: int16_t s2_ref_speed_rpm :: target rotational speed int16_t s2_ref_min_speed_rpm :: minimum value of target rotational speed Output: uint8_t :: motor stop flag	Set speed command value
	R_MTR_GetSpeed Input: none Output: int16_t :: rotational speed	Get speed
	R_MTR_SetDir Input: int8_t 1_dir :: direction of rotation Output: none	Set direction of rotation
	R_MTR_GetDir Input: none Output: int8_t :: direction of rotation	Get direction of rotation
	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_IcsInput Input: mtr_ctrl_input_t *st_ics_input ::ICS structure Output: none	Input values of ICS variables to ICS buffer variables
	R_MTR_SetVariables 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_UpdatePolling Input: none Output: none	Set control variables

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

File	Function	Process overview
r_mtr_statemachine.c	mtr_statemachine_init Input: (st_mtr_statemachine_t) *p_state_machine :: motor status structure Output: none	Initialization of motor status
	mtr_statemacine_reset Input: (st_mtr_statemachine_t) *p_state_machine :: motor status structure Output: none	Reset motor status
	mtr_state_machine_event Input: (st_mtr_statemachine_t) *p_state_machine :: motor status structure (void) *p_object ::structure for control variables (uint8_t) u1_event ::event Output: (uint8_t) p_state_machine->u2_error_status :: error status	Execute event
	mtr_statemachine_get_status Input: (st_mtr_statemachine_t) *p_state_machine :: motor status structure Output: (uint8_t) p_state_machine->u1_status ::motor status	Get motor status
	mtr_act_none Input: (st_mtr_statemachine_t) *st_stm :: motor status structure (void) *p_param ::structure for control variables Output: (uint8_t) action_ret :: execution result (always “0”)	No process is performed
	mtr_act_init Input: (st_mtr_statemachine_t) *st_stm :: motor status structure (void) *p_param ::structure for control variables Output: (uint8_t) action_ret :: execution result (always “0”)	Initialization of control variables
	mtr_act_error Input: (st_mtr_statemachine_t) *st_stm :: motor status structure (void) *p_param ::structure for control variables Output: (uint8_t) action_ret :: execution result (always “0”)	Post processing after error detection
	mtr_act_drive Input: (st_mtr_statemachine_t) *st_stm :: motor status structure (void) *p_param ::structure for control variables Output: (uint8_t) action_ret :: execution result (always “0”)	Start motor
	mtr_act_stop Input: (st_mtr_statemachine_t) *st_stm :: motor status structure (void) *p_param ::structure for control variables Output: (uint8_t) action_ret :: execution result (always “0”)	Stop motor

Table 3-17 List of Functions in “r_mtr_foc_less_speed.c”

File	Function	Process overview
r_mtr_foc_less_speed.c	R_MTR_FOCMotorDefaultInit Input: st_mtr_foc_t *st_foc :: FOC structure pointer st_mtr_cint_t *st_cint :: carrier interrupt structure pointer Output: none	Initialization of control variables
	R_MTR_FOCMotorReset Input: st_mtr_foc_t *st_foc :: FOC structure pointer st_mtr_cint_t *st_cint :: carrier interrupt structure pointer Output: none	Reset control variables

Table 3-18 List of Functions in "r_mtr_est_phase_err.obj"

File	Function	Process overview
r_mtr_est_phase_err.obj	R_MTR_EstPhaseError Input: st_mtr_est_phe_t *st_phe:: Phase error estimation structure pointer int16_t *p_s2_ref_vdq :: dq axis voltage command value variable pointer int16_t s2_speed_rad :: speed int16_t * p_s2_phe_ed :: d-axis Back-EMF voltage int16_t * p_s2_phe_phase_err_rad :: phase error Output: none	Phase error estimating process

Table 3-19 List of Functions in "R_DSP_RL78_CC_S.lib" (1/3)

File	Function	Process overview
R_DSP_RL78_CC_S.lib	R_motor_uw2ab_abs_pu_FIX13 Input: st_coordinate13 *p_coordinate13 :: Coordinate transformation structure pointer Input range: $-4\sqrt{2/3}+1/2048 \leqq u, w \leqq 4\sqrt{2/3}-1/2048$ (FIX13) Output: none Output range: -4 ≦a, b ≦4-1/8192 (FIX13)	Clark transformation (absolute transformation) Convert U-phase (FIX 13 u) and W-phase (FIX 13 w) values to a-axis and b-axis values and store them in (FIX 13 a) and (FIX 13 b), respectively.
	R_motor_uw2ab_abs_sat_pu_FIX13 Input: st_coordinate13 *p_coordinate13 :: Coordinate transformation structure pointer Input range: $-4\sqrt{2/3}+1/2048 \leqq u, w \leqq 4\sqrt{2/3}-1/2048$ (FIX13) Output: none Output range: -4 ≦a, b ≦4-1/8192 (FIX13)	Clark transformation (absolute transformation) with saturation processing Convert U-phase (FIX 13 u) and W-phase (FIX 13 w) values to a-axis and b-axis values and store them in (FIX 13 a) and (FIX 13 b), respectively.
	R_motor_ab2dq_pu_FIX13 input: st_coordinate13 *p_coordinate13 :: Coordinate transformation structure pointer Input range: -4 ≦a, b ≦4-1/8192 (FIX13) -1 ≦ angle.sin, angle.cos ≦ 1 (FIX14) Output: none Output range: -4 ≦d, q ≦4-1/8192 (FIX13)	Clark transformation Convert a-axis (FIX 13 a) and b-axis (FIX 13 b) values to d-axis and q-axis values and store them in (FIX 13 d) and (FIX 13 q), respectively.
	R_motor_ab2dq_sat_pu_FIX13 input: st_coordinate13 *p_coordinate13 :: Coordinate transformation structure pointer Input range: -4 ≦a, b ≦4-1/8192 (FIX13) -1 ≦ angle.sin, angle.cos ≦ 1 (FIX14) Output: none Output range: -4 ≦d, q ≦4-1/8192 (FIX13)	Clark transformation with saturation processing Convert a-axis (FIX 13 a) and b-axis (FIX 13 b) values to d-axis and q-axis values and store them in (FIX 13 d) and (FIX 13 q), respectively.
	R_motor_uw2dq_abs_pu_FIX13 Input: st_coordinate13 *p_coordinate13 :: Coordinate transformation structure pointer Input range: $-4\sqrt{2/3}+1/2048 \leqq u, w \leqq 4\sqrt{2/3}-1/2048$ (FIX13) -1 ≦ angle.sin, angle.cos ≦ 1 (FIX14) Output: none Output range: -4 ≦d, q ≦4-1/8192 (FIX13)	Composite transformation of Clark transformation and Park transformation (absolute transformation) Convert U-phase (FIX 13 u) and W-phase (FIX 13 w) values to d-axis and q-axis values and store them in (FIX 13 d) and (FIX 13 q), respectively.
	R_motor_uw2dq_abs_sat_pu_FIX13 input: st_coordinate13 *p_coordinate13 :: Coordinate transformation structure pointer Input range: $-4\sqrt{2/3} \leqq u, w \leqq 4\sqrt{2/3}$ (FIX13) -1 ≦ angle.sin, angle.cos ≦ 1 (FIX14) Output: none Output range: -4 ≦d, q ≦4-1/8192 (FIX13)	Combined conversion (absolute conversion) of Clark conversion and Park conversion, with saturation processing Convert U-phase (FIX 13 u) and W-phase (FIX 13 w) values to d-axis and q-axis values and store them in (FIX 13 d) and (FIX 13 q), respectively.

Table 3-20 List of Functions in “R_DSP_RL78_CC_S.lib” (2/3)

<table>
<tr><th>File</th><th>Function</th><th>Process overview</th></tr>
<tr><td rowspan="5">R_DSP_RL78_CC_S.lib</td><td>R_motor_dq2uvw_abs_pu_FIX13
input: st_coordinate13 *p_coordinate13 :: Coordinate transformation structure pointer
Input range: -4 ≦d, q ≦4-1/8192 (FIX13)
and $\sqrt{d^2+q^2}$ ≦4 (FIX13)
-1 ≦ angle.sin, angle.cos ≦ 1 (FIX14)
Output: none
Output range: -4$\sqrt{2/3}$ ≦u, v, w ≦4$\sqrt{2/3}$ (FIX13)</td><td>Composite transformation of inverse Clark transformation and inverse Park transformation (absolute transformation)
Converts d-axis (FIX 13 d) and q-axis (FIX 13 q) values to U-phase and W-phase values and stores them in (FIX 13 u) and (FIX 13 w), respectively.</td></tr>
<tr><td>R_motor_uv2dq_abs_sat_pu_FIX13
input: st_coordinate13 *p_coordinate13 :: Coordinate transformation structure pointer
Input range: -4$\sqrt{2/3}$+1/2048 ≦u, v ≦4$\sqrt{2/3}$-1/2048 (FIX13)
-1 ≦ angle.sin, angle.cos ≦ 1 (FIX14)
Output: none
Output range: -4 ≦d, q ≦4-1/8192 (FIX13)</td><td>Combined conversion (absolute conversion) of Clark conversion and Park conversion, with saturation processing
Convert U-phase (FIX 13 u) and V-phase (FIX 13 v) values to d-axis and q-axis values and store them in (FIX 13 d) and (FIX 13 q), respectively.</td></tr>
<tr><td>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)</td><td>Calculate sine and cosine values from the angle (FIX 12 theta) and store them in (FIX 14 sin) and (FIX 14 cos), respectively.</td></tr>
<tr><td>R_motor_atan2_pu_FIX12
input: int16_t x :: Input value x
int16_t y :: Input value y
Input range: -4 ≦x, y ≦4-1/8192 (FIX13)
Output: int16_t atan(y/x)
Output range: -π ≦atan ≦π(FIX12)</td><td>Calculate the principal value of arctangent to (y / x) from the input values x (FIX13 x) and y (FIX13 y) and output it as the (FIX12) type.
Output when the input contains 0<table><tr><th>x</th><th>y</th><th>atan</th></tr><tr><td>0</td><td>Positive</td><td>π/2</td></tr><tr><td>0</td><td>Negative</td><td>-π/2</td></tr><tr><td>0</td><td>0</td><td>π/4</td></tr><tr><td>Positive</td><td>0</td><td>0</td></tr><tr><td>Negative</td><td>0</td><td>-π</td></tr></table></td></tr>
<tr><td>R_motor_atan2_pu_FIX14
input: int16_t x :: Input value x
int16_t y :: Input value y
Input range: -4 ≦x, y ≦4-1/8192 (FIX13)
Output: int16_t atan(y/x)
Output range: -0.5 ≦atan ≦0.49993896484375 (FIX14)</td><td>Calculate the principal value of arctangent to (y / x) from the input values x (FIX13 x) and y (FIX13 y) and output it as the (FIX14) type.
Output when the input contains 0<table><tr><th>x</th><th>y</th><th>atan</th></tr><tr><td>0</td><td>Positive</td><td>π/2</td></tr><tr><td>0</td><td>Negative</td><td>-π/2</td></tr><tr><td>0</td><td>0</td><td>π/4</td></tr><tr><td>Positive</td><td>0</td><td>0</td></tr><tr><td>Negative</td><td>0</td><td>-π</td></tr></table></td></tr>
</table>

Table 3-21 List of Functions in “R_DSP_RL78_CC_S.lib” (3/3)

File	Function	Process overview
R_DSP_RL78_CC_S.lib	R_motor_sqrt_sum_pu_FIX13 input: int16_t x :: Input value x int16_t y :: Input value y Input range: -4 ≦x, y ≦4-1/8192 (FIX13) and $\sqrt{x^2+y^2}$ ≦4-1/8192 (FIX13) Output: int16_t $\sqrt{x^2+y^2}$ Output range: 0 ≦sqrt ≦ 4√2 (FIX13)	Calculate the value of $\sqrt{x^2+y^2}$ from the input values x and y and output as the (FIX13) type.
	R_motor_sqrt_dif_pu_FIX13 input: int16_t x :: Input value x int16_t y :: Input value y Input range: -4 ≦x, y ≦4-1/8192 (FIX13) Output: int16_t $\sqrt{x^2-y^2}$ Output range: 0 ≦ sqrt ≦ 4 (FIX13)	Calculate the value of $\sqrt{x^2-y^2}$ from the input values x and y and output as the (FIX13) type.

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

File	Function	Process overview
r_mtr_interrupt.c	mtr_carrier_interrupt Input: none Output: none	Cycle timer interrupt (Call using INTTRD1) Cycle: 100 µs (1time decimation of 50us) - Call vector control functions - Bus voltage detection - Call communication process - Call vector control preparation functions
	mtr_get_vdc Input: none Output: none	Bus voltage detection - Calculate bus voltage from AD converter - Calculate inverse voltage and voltage limit
	mtr_1ms_interrupt Input: none Output: none	Cycle timer interrupt (Call using INTTM00) Cycle: 1 ms - Startup control - Field weakening control/maximum torque control - Call command value setting process for d-axis and q-axis current and rotational speed - Call speed PI control process - Call error monitoring process
	mtr_lpf1_run (inline function) Input: st_mtr_lpf1_t *p_st_lpf :: LPF structure pointer int16_t s2_input :: LPF input const uint8_t u1_q :: Q value of LPF Output: LPF output	Primary LPF process
	mtr_set_speed_ref (inline function) Input: st_mtr_foc_t *p_st_foc :: FOC structure pointer Output: int16_t :: speed command value	Set command value for speed control
	mtr_pi_run_asr (inline function) Input: st_mtr_pi_t *p_st_pi :: PI control structure pointer int16_t s2_err :: deviation 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	Speed PI control process
	mtr_set_iq_ref (inline function) Input: st_mtr_foc_t *p_st_foc :: FOC structure pointer Output: int16_t :: q-axis current command value	Set q-axis current command value
	mtr_set_id_ref (inline function) Input: st_mtr_foc_t *p_st_foc :: FOC structure pointer Output: int16_t :: d-axis current command value	Set d-axis current command value
	mtr_error_check (inline function) Input: st_mtr_foc_t *p_st_foc :: FOC structure pointer Output: none	Error process - Overvoltage detection - Undervoltage detection - Excessive speed detection
	mtr_abs (inline function) Input: int16_t s2_value :: input value Output: int16_t :: output value	Output absolute value of input
	mtr_limit_abs (inline function) Input: int16_t s2_value :: input value int16_t s2_limit_value :: limit value Output: int16_t :: output value	Limit input by absolute value
	mtr_cmp_interrupt [Used in IPD mode] Input: none Output: none	Get TRX count during initial position detection

Table 3-23 List of Functions in "r_mtr_interrupt.c" 2/2)

File	Function	Process overview
r_mtr_interrupt.c	mtr_abs (inline function) Input: int16_t s2_value :: input value Output: int16_t :: output value	Output absolute value of input
	mtr_limit_abs (inline function) Input: int16_t s2_value :: input value int16_t s2_limit_value :: limit value Output: int16_t :: output value	Limit input by absolute value
	mtr_cmp_interrupt [Used in IPD mode] Input: none Output: none	Get TRX count during initial position detection

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

File	Function	Process overview
r_mtr_ipd.c	R_MTR_Ipd Input: st_mtr_ipd_t *p_st_ipd :: IPD structure uint16_t u2_current_offset :: current offset uint16_t *p_u2_error_status :: error status int8_t s1_ref_dir :: Saved value for direction of rotation command FIX12 *p_angle_theta :: ADC current coordinate system angle Output: uint8_t :: drive mode status	Initial position detection
	mtr_ipd_process Input: st_mtr_ipd_t *p_st_ipd :: IPD structure uint16_t u2_current_offset :: current offset uint16_t *p_u2_error_status :: error status Output : None	Initial position detection process
	mtr_reset_for_drive Input: st_mtr_ipd_t *p_st_ipd :: IPD structure uint16_t u2_current_offset :: current offset int8_t s1_dir :: direction of rotation FIX12 *p_theta :: angle of rotor Output: None	Reset peripheral functions for drive after initial position detection
	mtr_measure_inductance_effect Input: uint8_t u1_energized_phase :: number of energized phases uint8_t u1_v_pattern :: voltage pattern st_mtr_ipd_t *p_st_ipd :: IPD structure uint16_t *p_u2_error_status :: error status Output: None	Measures the time taken to reach threshold current during initial position detection
	mtr_salient_detect_angle Input: st_mtr_ipd_t *p_st_ipd :: IPD structure uint16_t u2_current_offset :: current offset uint16_t *p_u2_error_status :: error status Output: None	Angle detection process for initial position detection with a salient motor
	mtr_salient_detect_polarity Input: st_mtr_ipd_t *p_st_ipd :: IPD structure uint16_t u2_current_offset :: current offset uint16_t *p_u2_error_status :: error status Output: None	Polarity detection process for initial position detection with a salient motor
	mtr_non_salient_detect_angle Input: st_mtr_ipd_t *p_st_ipd :: IPD structure uint16_t u2_current_offset :: current offset uint16_t *p_u2_error_status :: error status Output: None	Angle detection process for initial position detection with a non-salient motor
	mtr_set_initial_position Input: uint8_t u1_position :: IPD result int8_t s1_dir :: direction of rotation FIX12 *p_theta :: angle of rotor Output: None	Set initial angle for open loop depending on the result of initial position detection
	mtr_lower_arm_on Input: uint16_t u2_low_on_period :: period for lower arms on Output: None	Set all lower arms on

Table 3-25 List of Functions in “r_mtr_ipd.c” (2/2)

File	Function	Process overview
r_mtr_ipd.c	mtr_prepare_energize_phase 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 Input: None Output: None	Stop output of voltage
	mtr_reset_timer Input: None Output: None	Reset TRD and TAU
	mtr_ebable_cmp_intr Input: None Output: None	Enable the interrupt of CMP
	mtr_disable_cmp_intr Input: None Output: None	Disable the interrupt of CMP

Table 3-26 List of Functions in “r_mtr_ipd.h”

File	Function	Process overview
r_mtr_ipd.h	R_MTR_SetCutoffSource Input: None Output: None	Set PWM forced cutoff source to CMP
	R_MTR_ClearCutoffSource Input: None Output: None	Release PWM forced cutoff source from CMP

Table 3-27 List of Functions in “r_mtr_ol2cl_ctrl.obj”

File	Function	Process overview
r_mtr_ol2cl_ctrl.obj	R_MTR_OL2CLTorqueCurrentCalc Input: st_mtr_ol2cl_t *st_ol2cl :: Open loop to Closed loop switch Control structure pointer int16_t s2_id_ref :: Id reference int8_t s1_dir :: direction of rotation Output: None	Torque current calculation for Open loop to Closed loop switch control
	R_MTR_OL2CLSwichCtrl Input: st_mtr_ol2cl_t *st_ol2cl :: Open loop to Closed loop switch Control structure pointer int16_t s2_id_ref :: Id reference Output: int16_t :: Iq reference	Iq reference calculation for Open loop to Closed loop switch control
	R_MTR_DampCtrl Input: st_mtr_damp_t *st_damp :: axis error estimate structure pointer int16_t s2_ed :: d-axis Back-EMF voltage int16_t s2_speed_ref :: speed reference Output: int16_t :: damping speed reference	Damping control process
	mtr_hpf1_run (inline function) Input: st_mtr_hpf1_t *st_hpf :: HPF structure pointer int16_t s2_input :: HPF input const uint8_t u1_q :: Q value of HPF Output: int16_t :: HPF output	Primary HPF process
	mtr_limit_abs (inline function) Input: int16_t s2_value :: input value int16_t s2_limit_value :: limit value Output : int16_t :: output value	Input limitation by absolute value
	mtr_abs (inline function) Input : int16_t s2_value :: input value Output : int16_t :: output value	Output to absolute value of input

Table 3-28 List of Functions in “r_mtr_disturb_suppress.obj”

File	Function	Process overview
r_mtr_disturb_suppress.obj	R_MTR_DisturbSuppress Input: st_mtr_do_t *st_do :: Disturbance suppression structure pointer int16_t s2_speed :: speed integral value int16_t s2_current :: Iq reference int16_t s2_d0_div_pm :: Static friction coefficient/(number of pole pairs × BEMF constant)int16_t s2_d1_div_p2m :: Kinetic friction coefficient/((number of pole pairs)^2 × BEMF constant) output: int16_t :: Id reference	Disturbance suppression process
	mtr_lpf1_run (inline function) Input: st_mtr_do_lpf1_t *p_st_lpf :: LPF structure pointer int16_t s2_input :: LPF input const uint8_t u1_q :: Q value of LPF Output: int16_t :: LPF output	Primary LPF process
	mtr_hpf1_run (inline function) Input: st_mtr_do_hpf1_t *st_hpf :: HPF structure pointer int16_t s2_input :: HPF input const uint8_t u1_q :: Q value of HPF Output: int16_t :: HPF output	Primary HPF process

Table 3-29 List of Functions in “r_mtr_foc.c” (1/4)

File	Function	Process overview
r_mtr_foc.c	MtrFocSequence Input: None Output: None	Field oriented control process
	mtr_get_current_ss (inline function) Input: st_mtr_sscs_t *p_st_sscs :: 1-shunt resistor current detection structure pointer int16_t *p_s2_i_uvw :: UVW-phase detection current pointer int16_t s2_limit_over_current :: over current limit value st_coordinate13* p_st_i_repro :: reproduction current coordinate system structure pointer Output: error status	current detection [Used in 1-shunt mode]
	mtr_get_current_ts (inline function) Input: st_mtr_tscs_t *p_st_tscs :: 3-shunt resistor current detection int16_t *p_s2_i_uvw :: UVW-phase detection current pointer int16_t s2_limit_over_current :: over current limit value Output: error status	current detection [Used in 3-shunt mode]
	mtr_uvw2dq_current (inline function) Input: st_coordinate13 *p_st_ad_i :: ADC current coordinate system structure pointer st_sincos12* p_ad_i_angle :: angle structure pointer Output: None	Conversion detection current (U,V,W-phase → d,q axis)
	mtr_est_phase_err (inline function) Input: st_coordinate13 *p_st_ad_i :: ADC current coordinate system structure pointer st_mtr_parameter_t *p_st_motor :: motor parameter structure pointer st_mtr_est_phe_t *p_st_phe :: phase error estimate structure pointer int16_t *p_s2_vdq :: dq axis reference voltage pointer st_mtr_common_cint_t *p_st_common :: 1ms /50us period common structure pointer Output: None	Axis error estimate process

Table 3-30 List of Functions in “r_mtr_foc.c” (2/4)

File	Function	Process overview
r_mtr_foc.c	mtr_ctrl_speed (inline function) Input: st_mtr_pll_t *p_st_pll :: PLL control structure pointer st_mtr_asr_cint_t *p_st_asr :: ASR structure pointer int16_t s2_phase_err_rad :: phase error int16_t s2_ctrl_period :: current control period int16_t *p_ad_i_theta :: ADC current coordinate system angle pointer int16_t *p_ref_v_theta :: reference voltage coordinate system angle pointer st_mtr_common_cint_t *p_st_common :: 1ms/50us period common structure pointer Output: None	Speed control
	mtr_ctrl_current (inline function) Input: st_mtr_est_phe_t *p_st_phe :: phase error estimate structure pointer int16_t *p_ad_i_theta :: ADC current coordinate system angle pointer int16_t *p_ref_v_theta :: reference voltage coordinate system angle pointer st_mtr_asr_cint_t *p_st_asr :: ASR structure pointer int16_t s2_ctrl_period :: current control period st_mtr_pll_t *p_st_pll :: PLL control structure pointer st_mtr_common_cint_t *p_st_common :: 1ms/50us period common structure pointer Output: None	Current control
	mtr_get_direction (inline function) Input: int8_t* p_s1_direction :: direction of rotate st_mtr_common_cint_t *p_st_common :: 1ms/50us period common structure pointer Output: None	Get a direction of the motor rotation
	mtr_ctrl_pi (inline function) Input: st_coordinate13 *p_st_ad_i :: ADC current coordinate system pointer st_coordinate13 *p_st_ref_i :: reference current coordinate system structure pointer st_coordinate13 *p_st_ref_v :: reference voltage coordinate system structure pointer st_mtr_acr_cint_t *p_st_acr :: ACR structure pointer Output: None	Current PI control
	mtr_ctrl_decoupling (inline function) Input: st_mtr_parameter_t *p_st_motor :: motor parameter structure pointer st_mtr_acr_cint_t *p_st_acr :: ACR structure pointer st_coordinate13 *p_st_ad_i :: ADC current coordinate system structure pointer st_coordinate13 *p_st_ref_v :: reference voltage coordinate system structure pointer st_mtr_common_cint_t *p_st_common :: 1ms/50us period common structure pointer Output: None	Decoupling controller
	mtr_dq2uvw_voltage (inline function) Input: st_coordinate13 *p_st_ref_v :: reference voltage coordinate system structure pointer Output: None	Conversion detection voltage (d,q axis → U,V,W - phase)
	mtr_set_pwm_ss (inline function) Input: st_coordinate13 *p_st_ref_v :: reference voltage coordinate system structure pointer st_mtr_mod_t *p_st_mod :: modulation structure pointer st_mtr_sscs_cint_t *p_st_sscs :: 1-shunt resistor current detection structure pointer Output: None	Setting of PWM output [Used in 1-shunt mode]

Table 3-31 List of Functions in “r_mtr_foc.c” (3/4)

File	Function	Process overview
r_mtr_foc.c	mtr_set_pwm_ts (inline function) Input: st_coordinate13 *p_st_ref_v :: reference voltage coordinate system structure pointer st_mtr_mod_t *p_st_mod :: modulation structure pointer st_mtr_tscs_cint_t *p_st_tscs :: 3-phase current detection structure pointer Output: None	Setting of PWM output [Used in 3-shunt mode]
	mtr_conv_ad_ss (inline function) Input: st_mtr_sscs_cint_t *p_st_sscs :: 1-shunt resistor current detection structure pointer Output: None	A/D conversion timing calculation [Used in 1-shunt mode]
	mtr_2phase_duty_cross (inline function) Input:st_coordinate13 *p_st_ad_i :: ADC current coordinate system structure pointer st_coordinate13 *p_st_i_repro :: reproduction current coordinate system structure pointer st_mtr_lpf1_t *p_st_id_repro :: Id for reproduction current LPF pointer st_mtr_lpf1_t *p_st_iq_repro :: Iq for reproduction current LPF pointer Output: None	2-phase cross current compensation
	mtr_deadtime_comp (inline function) Input: st_mtr_deadtime_comp_t* p_st_dtcomp :: deadtime compensation structure pointer st_coordinate13* p_st_i_uvw_dq,:: reference current coordinate system structure pointer st_coordinate13* p_st_v_uvw_dq :: reference voltage coordinate system structure pointer st_mtr_mod_t* p_st_mod :: modulation structure pointer Output: None	Deadtime compensation process
	mtr_mod_ss (inline function) Input: st_mtr_mod_t* p_st_mod :: modulation structure pointer st_coordinate13* p_st_ref_v :: reference voltage coordinate system structure pointer uint8_t* p_u1_drv_pat :: PWM magnitude relationship pattern st_mtr_sscs_cint_t* p_st_sscs :: 1-shunt resistor current detection structure pointer Output: None	Modulation process [Used in 1-shunt mode]
	mtr_mod_ts (inline function) Input: st_mtr_mod_t* p_st_mod :: modulation structure pointer st_coordinate13* p_st_ref_v :: reference voltage coordinate system structure pointer Output: None	Modulation process [Used in 3-shunt mode]
	mtr_pwm_duty_ss (inline function) Input: st_mtr_sscs_cint_t* p_st_sscs :: 1-shunt resistor current detection structure pointer Output: None	Duty calculation [Used in 1-shunt mode]
	mtr_pwm_duty_ts (inline function) Input: st_mtr_tscs_cint_t* p_st_tscs :: 3-phase current detection structure pointer int16_t s2_u :: U-phase modulation rate int16_t s2_v :: V-phase modulation rate int16_t s2_w :: W-phase modulation rate uint8_t u1_uvw_min_pattern :: Phase current minimum pattern Output: None	Duty calculation [Used in 3-shunt mode]
	mtr_set_duty_adj_ss (inline function) Input: st_mtr_sscs_cint_t* p_st_sscs :: 1-shunt resistor current detection structure pointer Output: None	Duty setting [Used in 1-shunt mode]

Table 3-32 List of Functions in "r_mtr_foc.c" (4/4)

File	Function	Process overview
r_mtr_foc.c	mtr_uvw_voltage_limit (inline function) Input: int16_t* p_s2_ref_v_uvw :: UVW-phase voltage pointer int16_t s2_voltage_limit :: voltage limit value Output: None	3-phase voltage limit process
	mtr_limit (inline function) Input: int16_t s2_value :: Target value int16_t s2_max :: maximum value int16_t s2_min :: minimum value Output: int16_t :: limit value	Maximum and minimum limitation process
	mtr_limit_or_zero (inline function) Input: int16_t s2_value :: Target value int16_t s2_max :: maximum value int16_t s2_min :: minimum value int16_t s2_zero_line :: 0 correction threshold Output: int16_t :: limit value	Maximum and minimum limitation process with 0 correction
	mtr_lpf1_run (inline function) Input: st_mtr_lpf1_t *p_st_lpf :: LPF structure pointer int16_t s2_input :: LPF input const uint8_t u1_q :: Q value of LPF Output: int16_t :: LPF output	Primary LPF process
	mtr_pi_run_acr【inline function】 Input: st_mtr_pi_t *p_st_pi :: PI control structure pointer int16_t s2_err :: deviation const uint8_t u1_kp_q :: Q value of proportional gain const uint8_t u1_kidt_q :: Q value of integral gain Output: int16_t :: PI output	Current PI control process
	mtr_abs (inline function) Input: int16_t s2_value :: input value Output: int16_t :: output value	Output to absolute value of input
	mtr_limit_abs (inline function) Input: int16_t s2_value :: input value int16_t s2_limit_value :: limit value Output: int16_t :: output value	limit input by absolute value

Table 3-33 List of Functions in “r_mtr_mtpa2fw_ctrl.obj”

File	Function	Process overview
r_mtr_mtpa2fw_ctrl.obj	R_MTR_SetVlimit Input: st_mtr_mtpa2fw_t *st_m2f :: Field-Weakening control/maximum torque control structure pointer int16_t s2_vdc :: power supply Output: None	Output voltage limit setting
	R_MTR_MTPA2FWCtrl Input: st_mtr_mtpa2fw_t *st_m2f :: Field-Weakening control/maximum torque control structure pointer int16_t *s2_ref_idq :: dq axis reference current pointer int16_t *s2_idq :: dq axis detection current pointer int16_t s2 speed_rad :: speed Output: None	Field-Weakening control/maximum torque control

Table 3-34 List of Functions in “r_mtr_offset.c”

File	Function	Process overview
r_mtr_offset.c	R_MTR_CalibCurrentOffsetSs Input: st_mtr_sscs_t *p_st_sscs :: 1-shunt resistor current detection structure pointer int16_t * p_s2_offset :: current offset value pointer Output: Drive mode status	Current offset detection [Used in 1-shunt mode]
	R_MTR_CalibCurrentOffsetTs Input: st_mtr_tscs_t *p_st_tscs :: 3-phase current detection structure pointer int16_t * p_s2_offset :: current offset value pointer Output: Drive mode status	Current offset detection [Used in 3-shunt mode]

Table 3-35 List of Functions in “r_mtr_prep.c”

File	Function	Process overview
r_mtr_prep.c	R_MTR_StopSequence Input: None Output: None	Vector control preparation

3.3 List of Sensorless Vector Control Software Function Variables

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

Table 3-36 List of Variables in "main.c"

Variable	Type	Qn	PU	Content	Remarks
g_u1_system_mode	static uint8_t	Q0	-	Mode system management	
g_u1_motor_status	static uint8_t	Q0	-	Motor status management	
g_u2_error_status	static uint16_t	Q0	-	Error status management	
g_u2_conf_hw	uint16_t	Q0	-	RMW configuration	
g_u2_conf_sw	uint16_t	Q0	-		
g_u2_conf_tool	uint16_t	Q0	-		
gui_u1_active_gui	uint8_t	Q0	-		
g_u2_conf_sw_ver	uint16_t	Q0	-		
com_u1_run_event	uint8_t	Q0	-	Change run mode	0: MTR_EVENT_STOP 1: MTR_EVENT_DRIVE 2: MTR_EVENT_ERROR 3: MTR_EVENT_RESET
g_u1_run_event	uint8_t	Q0	-		
g_u2_system_error	uint16_t	Q0	-	System error management	

Table 3-37 List of Variables in “r_mtr_ics.c”(1/2)

Variable	Type	Qn	PU	Content	Remarks
st_ics	st_mtr_ctrl_input_t	-	-	Structure for ICS variable transfer	Structure
com_u1_direction	uint8_t	Q0	-	Direction of rotation	0: CW 1: CCW
com_f4_mtr_r	float	-	-	Resistance [Ω]	
com_f4_mtr_ld	float	-	-	d-axis inductance [H/rad]	
com_f4_mtr_lq	float	-	-	q-axis inductance [H/rad]	
com_f4_mtr_m	float	-	-	BEMF constant [Vs/rad]	
com_f4_mtr_j	float	-	-	Rotor inertia [kgm^2/rad^2]	
com_f4_mtr_d0	float	-	-	Coefficient of friction at rest[kgm^2/(rad･s^2)]	
com_f4_mtr_d1	float	-	-	Coefficient of kinetic friction[kgm^2/(rad^2･s)]	
com_u2_mtr_pp	uint16_t	Q0	-	Number of pole pairs	
com_u2_offset_calc_cnt	uint16_t	Q0	-	Current offset detection time	
com_s2_ref_speed_rpm	int16_t	Q0	-	Reference rotational speed [rpm]	Mechanical angle
com_f4_ramp_limit_speed_rpm	float	-	-	Limit of acceleration [rpm/ms]	Mechanical angle
com_s2_max_speed_rpm	int16_t	Q0	-	Maximum speed [rpm]	Mechanical angle
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_lpf_cof_hz	float	-	-	Speed LPF cut-off frequency [Hz]	
com_f4_acr_lpf_cof_hz	float	-	-	Current LPF cut-off frequency [Hz]	
com_f4_pll_nf_hz	float	-	-	PLL natural frequency [Hz]	
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_pll_deadband_lsb	float	-	-	Deadband of phase locked loop calculation [LSB]	
com_f4_asr_ki_aug	float	-	-	Augmentation rate for integral part of ASR	
com_s2_cl2ol_speed_rpm	int16_t	Q0	-	Switching speed from sensorless to open loop [rpm]	Mechanical angle
com_s2_ol2cl_speed_rpm	int16_t	Q0	-	Switching speed from open loop to sensorless [rpm]	Mechanical angle
com_f4_ol_ref_id	float	-	-	Open loop d-axis command current [A]	
com_f4_draw_in_wait_time	float	-	-	Draw-in wait time count value [s]	
com_f4_init_asr_intg	float	-	-	ASR integral term initial value during sensorless transition	
com_f4_ramp_limit_current	float	-	-	Limit value for current rise [A/ms]	
com_s2_duty_diff_limit	int16_t	Q0	-	Limit of difference 1 between each phase duty	[Used in 1-shunt mode]
com_f4_i_repro_cof_hz	float	-	-	LPF cut-off frequency for current reproduction [Hz]	
com_s2_duty_diff_limit2	int16_t	Q0	-	Limit of difference 2 between each phase duty	
com_s2_ad_point_a_adj_cnt	int16_t	Q0	-	Adjustment for A/D delay counts for A point	
com_s2_ad_point_b_adj_cnt	int16_t	Q0	-	Adjustment for A/D delay counts for B point	

Table 3-38 List of Variables in “r_mtr_ics.c”(2/2)

Variable	Type	Qn	PU	Content	Remarks
com_s2_mod_3ph2ph_speed_rpm	int16_t	Q0	-	Switching speed from 3-phase modulation to 2-phase modulation [rpm]	[Used in 1-shunt mode and 2-phase modulation]
com_s2_mod_2ph3ph_speed_rpm	int16_t	Q0	-	Switching speed from 2-phase modulation to 3-phase modulation [rpm]	
com_s2_ad_point_a_adj_cnt_3ph	int16_t	Q0	-	Adjustment value of A/D conversion timing A (3-phase modulation)	
com_s2_ad_point_b_adj_cnt_3ph	int16_t	Q0	-	Adjustment value of A/D conversion timing B (3-phase modulation)	
com_s2_ad_point_a_adj_cnt_2ph	int16_t	Q0	-	Adjustment value of A/D conversion timing A (2-phase modulation)	
com_s2_ad_point_b_adj_cnt_2ph	int16_t	Q0	-	Adjustment value of A/D conversion timing B (2-phase modulation)	
com_s2_mod_2ph_bot_change_cnt	int16_t	Q0	-	Number of counts during 2-phase modulation transition	
com_f4_sal_angle_current	float	-	-	Threshold current for angle detection of saliency motor [A]	【IPD】
com_u4_sal_angle_th	uint32_t	Q0	-	TRX count value difference threshold for angle detection of saliency motor	
com_u2_sal_angle_discharge	uint16_t	Q0	-	Discharge time for angle detection of saliency motor	
com_f4_sal_polarity_current	float	-	-	Threshold current for polarity detection of salient motor [A]	
com_u4_sal_polarity_th	uint32_t	Q0	-	TRX count value differential of polarity detection for salient rotor	
com_u2_sal_polarity_discharge	uint16_t	Q0	-	Discharge period of polarity detection for salient motor	
com_f4_non_sal_current	float	-	-	Threshold current of angle detection for non-salient motor	
com_u4_non_sal_th	uint32_t	Q0	-	TRX count value differential of angle detection for non-salient motor	
com_u2_non_sal_discharge	uint16_t	Q0	-	Discharge period of angle detection for non-salient motor	
com_f4_fw_pi_nf_hz	float	-	-	PI controller natural frequency for Field-Weakening Control [Hz]	【Field-Weakening control/maximum torque control】
com_s2_fw_speed_err_th_rpm	int16_t	Q0	-	Speed error threshold for Field-Weakening Control [rpm]	
com_f4_fw_speed_err_lpf_cof_hz	float	-	-	Speed error LPF cutoff frequency [Hz]	
com_s2_fw_id_inc_const	int16_t	Q0	-	Field-Weakening Id control constant	
com_u1_fw_pi_intg_redct_cof	uint8_t	Q0	-	Field-Weakening Id pi integral term reduction coefficient	
com_f4_do_lpf_cof_hz	float	-	-	LPF cutoff frequency [Hz] for disturbance suppression	【Disturbance suppression】
com_f4_damp_hpf_cof_hz	float	-	-	HPF cutoff frequency for damping control [Hz]	【Open loop damping control 】
com_f4_damp_zeta	float	-	-	Damping coefficient of damping control	
com_f4_damp_speed_limit_rate	float	-	-	Damping control speed limit	
com_f4_pherr_lpf_cof_hz	float	-	-	Phase error LPF cutoff frequency for Open loop to Closed loop switch Control	【OL2CL】
com_f4_ol2cl_switch_time	float	-	-	Time[s] to switch open loop to sensor-less	
com_s2_enable_write	int16_t	Q0	-	Variable to allow variable rewriting	
g_s2_enable_write	int16_t	Q0	-	Variable to allow variable rewriting	
g_u1_cnt_ics	static uint8_t	Q0	-	Communication processing cycle delay variable	

Table 3-39 List of Variables in “r_mtr_driver_access.c”

Variable	Type	Qn	PU	Content	Remarks
st_ics_buff	mtr_ctrl_input_t	Q0	-	Buffer structure for ICS variable transfer	Structure
g_u1_trig_enable_write	uint8_t	Q0	-	Transfer completion flag	
g_u1_stop_req	uint8_t	Q0	-	Motor stop flag	
g_s2_cnt	int16_t	Q0	-	Counter	

Table 3-40 List of Variables in “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 3-41 List of Variables in “r_mtr_interrupt.c”

Variable	Type	Qn	PU	Content	Remarks
gst_foc	st_mtr_foc_t	-	-	Vector control structures	Structure
gst_cint	st_mtr_cint_t	-	-	Carrier interrupt counter	Structure

3.4 List of Sensorless Vector Control Software Structures

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

Table 3-42 List of Variables in “r_mtr_parameter.h” / Structure: “st_mtr_parameter_t”

Variable	Type	Qn	PU	Content	Remarks
u2_mtr_pp	uint16_t	Q0	-	Number of pole pairs	
s2_mtr_r	int16_t	Q17	Resistance (voltage/current)	Resistance [PU]	
s2_mtr_ld	int16_t	Q18 (1 shunt) / Q19 (3 shunt)	Inductance (resistance/angular frequency)	d-axis inductance [PU]	
s2_mtr_lq	int16_t	Q18 (1 shunt) / Q19 (3 shunt)	Inductance (resistance/angular frequency)	q-axis inductance [PU]	
s2_mtr_m	int16_t	Q15	BEMF constant (voltage/angular frequency)	BEMF constant [PU]	
s2_mtr_j	int16_t	Q9 (1 shunt) / Q10 (3 shunt)	Inertia (BEMF constant ×current× (number of pole pairs/angular frequency) ^2)	Inertia [PU]	

Table 3-43 List of Variables in “r_mtr_ctrl_gain.h” / Structure: “st_mtr_design_parameter_t”(1/2)

Variable	Type	Qn	PU	Content	Remarks
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_asr_lpf_cof_hz	float	Q0	-	Speed LPF cutoff frequency [Hz]	
f4_acr_lpf_cof_hz	float	Q0	-	Current LPF cut-off frequency [Hz]	
f4_pll_nf_hz	float	Q0	-	PLL natural frequency [Hz]	
f4_pll_deadband_lsb	float	Q0	-	Deadband of speed estimator PI calculation [LSB]	
f4_i_repro_cof_hz	float	Q0	-	LPF cut-off frequency for current reproduction [Hz]	
f4_asr_ki_aug	float	Q0	-	Augmentation rate for integral part of ASR	
f4_dt	float	Q0	-	Control period [sec]	
f4_dt_speed	float	Q0	-	Control period for speed loop [sec]	
f4_r	float	Q0	-	Resistance [Ω]	
f4_ld	float	Q0	-	d-axis inductance [H/rad]	
f4_lq	float	Q0	-	q-axis inductance [H/rad]	
f4_m	float	Q0	-	BEMF constant [V･s/rad]	
f4_j	float	Q0	-	Rotor inertia [kgm^2/rad^2]	
f4_rated_current	float	Q0	-	Rated current [Arms]	
f4_ol_ref_id	float	Q0	-	Open loop reference Id [A]	
f4_ol2cl_speed	float	Q0	-	Switching speed from open loop [rpm]	
f4_ramp_limit_speed	float	Q0	-	Limit of acceleration [rpm/ms] (mechanical)	
f4_do_lpf_cof_hz	float	Q0	-	Disturbance suppression LPF cutoff frequency [Hz]	
f4_do_hpf_cof_hz	float	Q0	-	Disturbance suppression HPF cutoff frequency [Hz]	
f4_d0_div_pm	float	Q0	-	Static friction coefficient/(number of pole pairs × BEMF constant) [kg･m^2/(rad･s^2)]	
f4_d1_div_p2m	float	Q0	-	Kinetic friction coefficient/((number of pole pairs)^2 × BEMF constant) [kg･m^2/(rad^2･s)]	
f4_damp_hpf_cof_hz	float	Q0	-	Damping control HPF cutoff frequency [Hz]	
f4_damp_zeta	float	Q0	-	Damping control damping coefficient	
f4_pu_sf_afreq	float	Q0	-	Frequency scale factor	
f4_pherr_lpf_cof_hz	float	Q0	-	Phase error lpf natural frequency [Hz]	
f4_ol2cl_switch_time	float	Q0	-	Time[s] to switch open loop to sensor-less	
f4_fw_pi_nf_hz	float	Q0	-	PI controller natural frequency for Field-Weakening Control [Hz]	
f4_fw_speed_err_lpf_cof_hz	float	Q0	-	Speed error LPF cutoff frequency [Hz]	
u2_mtr_pp	uint16_t	Q0	-	Number of pole pairs	

Table 3-44 List of Variables in “r_mtr_ctrl_gain.h” / Structure: “st_mtr_design_parameter_t”(2/2)

Variable	Type	Qn	PU	Content	Remarks
u1_q_current	uint8_t	Q0	-	Q-format of current	
u1_q_acr_kp	uint8_t	Q0	-	Q-format of Q-axis current PI proportional gain	
u1_q_acr_kidt	uint8_t	Q0	-	Q-format of Q-axis current PI ki× dt	
u1_q_acr_deadband	uint8_t	Q0	-	Q-format of current PI deadband	
u1_q_asr_kp	uint8_t	Q0	-	Q-format of Speed current PI proportional gain	
u1_q_asr_kidt	uint8_t	Q0	-	Q-format of Speed current PI ki × dt	
u1_q_asr_deadband	uint8_t	Q0	-	Q-format of speed PI deadband	
u1_q_pll_kp	uint8_t	Q0	-	Q-format of D-axis current PI proportional gain	
u1_q_pll_kidt	uint8_t	Q0	-	Q-format of D-axis current PI ki × dt	
u1_q_pll_deadband	uint8_t	Q0	-	Q-format of PLL deadband	
u1_q_acr_lpf_k	uint8_t	Q0	-	Q-format of Current LPF gain	
u1_q_asr_lpf_k	uint8_t	Q0	-	Q-format of Speed LPF gain	
u1_q_i_repro_lpf_k	uint8_t	Q0	-	Q-format of LPF numerator for current reproduction	
u1_q_do_lpf_k	uint8_t	Q0	-	Q-format of disturbance suppression LPF numerator	
u1_q_do_hpf_k	uint8_t	Q0	-	Q-format of disturbance suppression HPF numerator	
u1_q_j_div_tc_p2m	uint8_t	Q0	-	Q-format of inertia/time constant of LPF coefficient	
u1_q_damp_k	uint8_t	Q0	-	Q-format of Damping control gain	
u1_q_damp_hpf_k	uint8_t	Q0	-	Q-format of Damping control HPF	
u1_q_pherr_lpf_k	uint8_t	Q0	-	Q-format of Phase error LPF gain	
u1_q_ol2cl_current_k1	uint8_t	Q0	-	Q-format of Open loop to Closed loop switch Control gain	
u1_q_mtpa_cof_i	uint8_t	Q0	-	Q-format of (ld-lq) / ke calculation	
u1_q_fw_lpf_k	uint8_t	Q0	-	Q-format of LPF numerator for Field-Weakening Control	
u1_q_fw_kp	uint8_t	Q0	-	Q-format of D-axis FW PI proportional gain	
u1_q_fw_kidt	uint8_t	Q0	-	Q-format of D-axis FW PI ki * dt	

Table 3-45 List of Variables in “r_mtr_ctrl_gain.h” / Structure: “st_mtr_ctrl_gain_t”

Variable	Type	Qn	PU	Content	Remarks
s2_acr_id_kp	int16_t	Q17	Resistance	d-axis current control proportional gain	
s2_acr_id_kidt	int16_t	Q19	Resistance	d-axis current control integral gain×operation period	
s2_acr_iq_kp	int16_t	Q17	Resistance	q-axis current control proportional gain	
s2_acr_iq_kidt	int16_t	Q20	Resistance	q-axis current control integral gain×operation period	
s2_asr_kp	int16_t	Q12	Current/angular frequency	Speed control proportional gain	
s2_asr_kidt	int16_t	Q20	Current/angular frequency	Speed control integral gain×operation period	
s2_asr_deadband	int16_t	Q0	-	Speed integral part deadband	
s2_acr_deadband	int16_t	Q0	-	Current integral part deadband	
s2_asr_lpf_in_k	int16_t	Q15	-	Speed LPF input coefficient	
s2_acr_lpf_in_k	int16_t	Q15	-	Current LPF input coefficient	
s2_do_j_div_tc_p2m	int16_t	Q14	-	inertia/time constant of LPF coefficient	
s2_do_lpf_in_k	int16_t	Q15	-	Disturbance suppression LPF numerator	
s2_do_hpf_k	int16_t	Q15	-	Disturbance suppression HPF numerator	
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_i_repro_lpf_in_k	int16_t	Q15	-	LPF for current reproduction input coefficient	
s2_pll_kp	int16_t	Q15	Angular frequency/angle	PLL proportional gain	
s2_pll_kidt	int16_t	Q23	Angular frequency/angle	PLL integral gain×operation period	
s2_pll_deadband	int16_t	Q0	-	Speed estimation integral part deadband	
s2_damp_k	int16_t	Q13 (1 shunt) / Q12 (3 shunt)	Speed/Voltage	Damping control gain	
s2_damp_hpf_k	int16_t	Q15	-	Damping control HPF	
s2_ol2cl_theta2crnt_k0	int16_t	Q13	-	Zero order component of the angle-to-current conversion factor	
s2_ol2cl_theta2crnt_k1	int16_t	Q21	-	First-order component of the angle-to-current conversion factor	
s2_pherr_lpf_in_k	int16_t	Q15	-	Phase error LPF numerator	
s2_mtpa_cof_i	int16_t	Q13	-	MTPA current coefficient (ld - lq) / ke	
s2_fw_kp	int16_t	Q19	-	FW PI proportional gain	
s2_fw_kidt	int16_t	Q26	-	FW PI ki * dt	
s2_fw_lpf_in_k	int16_t	Q15	-	FW LPF numerator	

Table 3-46 List of Variables in “r_dsp_cc_s.h / Structure:” st_sincos12”

Variable	Type	Qn	PU	Content	Remarks
sin	int16_t	Q14	-	Sine (FIX14)	
cos	int16_t	Q14	-	Cosine (FIX14)	
theta	int16_t	Q12	1	Angle [rad] (FIX12)	

Table 3-47 List of Variables in “r_dsp_cc_s.h / Structure：” st_coordinate13”

Variable	Type	Qn	PU	Content	Remarks
u	int16_t	Q13	-	U-phase (FIX13)	
v	int16_t	Q13	-	V-phase (FIX13)	
w	int16_t	Q13	-	W-phase (FIX13)	
a	int16_t	Q13	-	α-axis (FIX13)	
b	int16_t	Q13	-	β-axis (FIX13)	
d	int16_t	Q13	-	d-axis (FIX13)	
q	int16_t	Q13	-	q-axis (FIX13)	
angle	st_sincos12	-	-	angle structure	

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

1 bit sign	15-n bit Integer part	n bit Decimal part

Table 3-48 List of Variables in “r_mtr_driver_access.h” / Structure: “st_mtr_ctrl_input_t” (1/2)

Variable	Type	Qn	PU	Content	Remarks
u1_direction	uint8_t	Q0	-	Direction of rotation	
u2_offset_calc_cnt	uint16_t	Q0	-	Offset detection time	
s2_ref_speed_rad	int16_t	Q14	Angular frequency	Reference rotational speed [PU]	Electric angle
s2_ramp_limit_speed_rad	int16_t	Q14	Angular frequency	Limit of acceleration [PU]	Electric angle
s2_max_speed_rad	int16_t	Q14	Angular frequency	Maximum speed [PU]	Electric angle
s2_cl2ol_speed_rad	int16_t	Q14	Angular frequency	Switching speed from sensorless to open loop [PU]	Electric angle
s2_ol2cl_speed_rad	int16_t	Q14	Angular frequency	Switching speed from open loop to sensorless [PU]	Electric angle
s2_ol_ref_id	int16_t	Q13	Current	Open loop d-axis reference current [PU]	
s2_draw_in_wait_cnt	int16_t	Q0	-	Draw-in wait time count value	
s2_init_intg	int16_t	Q13	Current	ASR integral term initial value during sensorless transition	
s2_ramp_limit_current	int16_t	Q13	Current	Limit value for current rise [PU/ms]	
s2_duty_diff_limit	int16_t	Q0	-	Minimum value of duty deviation between phases	[Used in 1-shunt mode]
s2_duty_diff_limit2	int16_t	Q0	-	Limit value for current rise [A/ms]	
s2_mod_3ph2ph_speed_rad	int16_t	Q0	-	Switching speed from 3-phase modulation to 2-phase modulation [rpm]	[Used in 1-shunt mode and 2-phase modulation]
s2_mod_2ph3ph_speed_rad	int16_t	Q0	-	Switching speed from 2-phase modulation to 3-phase modulation [rpm]	
s2_ad_point_a_adj_cnt_3ph	int16_t	Q0	-	Adjustment value of A/D conversion timing A (3-phase modulation)	
s2_ad_point_b_adj_cnt_3ph	int16_t	Q0	-	Adjustment value of A/D conversion timing B (3-phase modulation)	
s2_ad_point_a_adj_cnt_2ph	int16_t	Q0	-	Adjustment value of A/D conversion timing A (2-phase modulation)	
s2_ad_point_b_adj_cnt_2ph	int16_t	Q0	-	Adjustment value of A/D conversion timing B (2-phase modulation)	
s2_ad_point_a_adj_cnt	int16_t	Q0	-	Adjustment for A/D delay counts for A point	[Used in 1-shunt mode]
s2_ad_point_b_adj_cnt	int16_t	Q0	-	Adjustment for A/D delay counts for B point	
s2_mod_2ph_bot_change_cnt	int16_t	Q0	-	2-phase modulation transition counter	[Used in 2-phase modulation]

Table 3-49 List of Variables in "r_mtr_driver_access.h" / Structure: "st_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 rotor	【IPD】
u4_sal_angle_th	uint32_t	Q0	-	TRX count value differential of angle detection for salient rotor	
u2_sal_angle_discharge	uint16_t	Q0	-	Discharge period of angle detection for salient rotor	
u2_sal_polarity_current	uint16_t	Q0	-	Threshold current of polarity detection for salient rotor	
u4_sal_polarity_th	uint32_t	Q0	-	TRX count value differential of polarity detection for salient rotor	
u2_sal_polarity_ discharge	uint16_t	Q0	-	Discharge period of polarity detection for salient rotor	
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	-	Discharge period of angle detection for non-salient rotor	
s2_fw_speed_err_th_rad	int16_t	Q0	-	Speed error threshold for Field-Weakening Control [PU]	【Field-Weakening control/maximum torque control】
s2_fw_id_inc_const	int16_t	Q0	-	Field-Weakening Id control constant	
u1_fw_pi_intg_redct_cof	unt8_t	Q0	-	Field-Weakening Id pi integral term reduction coefficient	
s2_speed_limit_rate	int16_t	Q0	-	Damping control speed limit	【Open loop damping control 】
st_motor	st_mtr_parameter_t	-	-	Structure for motor parameter	Structure
st_ctrl_param	st_mtr_design_para meter_t	-	-	Structure for PI control	
st_gain_buf	st_mtr_ctrl_gain_t	-	-	Structure for Control Gain	

Table 3-50 List of Variables in "r_mtr_statemachine.h" / Structure: "st_mtr_statemachine_t"

Variable	Type	Qn	PU	Content	Remarks
u1_status	uint8_t	Q0	-	Motor status	
u1_status_next	uint8_t	Q0	-	Next motor status	
u2_error_status	uint16_t	Q0	-	Error status	
u1_current_event	uint8_t	Q0	-	Execution event	

表 3-51 List of Variables in "r_mtr_est_phase_err.h / Structure: "st_mtr_est_phe_t"

Variable	Type	Qn	PU	Content	Remarks
s2_eq	int16_t	Q13	Voltage	q-axis induced voltage	
s2_e	int16_t	Q13	Voltage	Induced voltage	
s2_r_id	int16_t	Q13	Voltage	R×id	
s2_r_iq	int16_t	Q13	Voltage	R×iq	
s2_speed_ld_id	int16_t	Q13	Voltage	Speed×Ld×id	
s2_speed_lq_iq	uint16_t	Q13	Voltage	Speed×Lq×iq	
s2_reci_m	int16_t	Q14	1/BEMF constant	Reciprocal of BEMF constant	

Table 3-52 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_lpf1_t”

Variable	Type	Qn	PU	Content	Remarks
s2_in_k	int16_t	Current: Q15 Speed: Q15 Phase: Q15	-	LPF input gain	
s2_out_k	int16_t	Current: Q15 Speed: Q15 Phase: Q15	-	LPF previous gain	
s2_pre_out	int16_t	Current: Q13 Speed: Q14 Phase: Q12	Current: current Speed: angular frequency Phase: angle	Previous output value	

Table 3-53 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_pi_t”

Variable	Type	Qn	PU	Content	Remarks
s2_kp	int16_t	Current: Q17 Speed: Q12 PLL: Q15	Current: Resistance Speed: Current/angular frequency PLL: Angular frequency/angle	Proportional gain	
s2_kidt	int16_t	Current: Q19 Speed: Q20 PLL: Q23	Current: Resistance Speed: Current/angular frequency PLL: Angular frequency/angle	Integral gain x control period	
s2_intg	int16_t	Current: Q13 Speed: Q14 PLL: Q12	Current: Voltage Speed: Current PLL: Angular frequency	Integral term	
s2_ilimit	int16_t	Current: Q13 Speed: Q14 PLL: Q12	Current: Voltage Speed: Current PLL: Angular frequency	Integral limit (up/down symmetry)	
s2_deadband	Int16_t	Current: Q13 Speed: Q14 PLL: Q12	Current: Current Speed: angular frequency PLL: angle	Deadband of integral part	
s2_decimal	Int16_t	Current: Q13 Speed: Q14 PLL: Q12	Current: Current Speed: angular frequency PLL: angle	Stored decimal part	

Table 3-54 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_acr_t”

Variable	Type	Qn	PU	Content	Remarks
s2_ref_id	int16_t	Q13	Current	d-axis reference current	
s2_ref_iq	int16_t	Q13	Current	q-axis reference current	
s2_limit_iq	int16_t	Q13	Current	q-axis current limit	
s2_ol_ref_id	int16_t	Q13	Current	Open loop d-axis current command value	
s2_ramp_limit_current	int16_t	Q13	Current	Limit value for current rise [PU/ms]	

Table 3-55 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_acr_cint_t”

Variable	Type	Qn	PU	Content	Remarks
u1_flag_err_zero	uint8_t	Q0	-	Current error zero flag	[Used in 1-shunt mode]
s2_ctrl_period	int16_t	Q18	Time	Current control cycle	
s2_pre_ref_vd	int16_t	Q13	Voltage	Previous d-axis reference voltage	
s2_pre_ref_vq	int16_t	Q13	Voltage	Previous q-axis reference voltage	
s2_iq_lpf	int16_t	Q13	Current	q-axis current LPF value	
s2_id_lpf	int16_t	Q13	Current	d-axis current LPF value	
st_iq_lpf	st_mtr_lpf1_t	-	-	q-axis current LPF structure	Structure
st_id_lpf	st_mtr_lpf1_t	-	-	d-axis current LPF structure	
st_pi_id	st_mtr_pi_t	-	-	d-axis current PI structure	
st_pi_iq	st_mtr_pi_t	-	-	q-axis current PI structure	

Table 3-56 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_pll_t”

Variable	Type	Qn	PU	Content	Remarks
s2_dt	int16_t	Q18	Time	Control cycle	
s2_speed_rad	int16_t	Q14	Frequencies	Speed	
st_pi	st_mtr_pi_t	-	-	PI structure	Structure

Table 3-57 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_deadtime_comp_t”

Variable	Type	Qn	PU	Content	Remarks
s2_deadtime_error_voltage	int16_t	Q13	Voltage	Voltage error	
s2_deadtime_limit_current	int16_t	Q13	Current	Current limit	
s2_delta_v_uvw[3]	int16_t	Q13	Voltage	3-phase voltage compensation value	

Table 3-58 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_asr_t”

Variable	Type	Qn	PU	Content	Remarks
s1_ref_dir	int8_t	-	-	Direction of rotation command	1: CW -1: CCW
s2_speed_ctrl_period	int16_t	Q15	Time	Speed control cycle	
s2_ref_speed_rad	int16_t	Q14	Angular frequency	Reference speed	
s2_ramp_limit_speed_rad	int16_t	Q14	Angular frequency	Limit of acceleration	
s2_ramp_deci_sample_cnt	int16_t	Q14	Angular frequency	Number of decimation of acceleration limit value	
s2_max_speed_rad	int16_t	Q14	Angular frequency	Maximum speed	
s2_limit_speed_rad	int16_t	Q14	Angular frequency	Limit of speed	
s2_init_intg	int16_t	Q13	Current	Integral term initial value during sensorless switching	
s2_cl2ol_speed_rad	int16_t	Q14	Angular frequency	Switching speed from closed loop to open loop	
s2_ol2cl_speed_rad	int16_t	Q14	Angular frequency	Switching speed from open loop to closed loop	
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)	
u2_cl2ol_judge_wait_cnt	uint16_t	Q0	-	Waiting counter for start to judge closed-loop to open-loop	
st_pi	st_mtr_pi_t	-	-	Speed PI structure	Structure

Table 3-59 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_asr_cint_t”

Variable	Type	Qn	PU	Content	Remarks
st_lpf	st_mtr_lpf1_t	-	-	Speed LPF structure	Structure

Table 3-60 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_mod_t”

Variable	Type	Qn	PU	Content	Remarks
u1_uvw_min_pattern	uint8_t	-	-	Phase current minimum pattern	
u1_pre_uvw_min_pattern	uint8_t	-	-	Previous value of minimum phase current pattern	
s2_2ph_bot_change_cnt	int16_t	-	-	2-phase modulation transition counter	[Used in 2-phase modulation]
s2_v_diff_sum	int16_t	-	-	Integration of the middle to minimum 2-phase voltage difference	
s2_com_v	int16_t	Q13	Voltage	Voltage offset	
s2_mod_u	int16_t	Q12	-	U-phase modulation factor	
s2_mod_v	int16_t	Q12	-	V-phase modulation factor	
s2_mod_w	int16_t	Q12	-	W-phase modulation factor	
s2_reci_vdc	int16_t	Q13	1/voltage	Inverse of voltage	
s2_limit_vout	int16_t	Q13	Voltage	Voltage limit	
s2_comp_v[3]	int16_t	Q13	Voltage	Deadtime compensation voltage	

Table 3-61 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_tscs_t [Used in 3-shunt mode]”

Variable	Type	Qn	PU	Content	Remarks
u2_offset_idc_adc	uint16_t	Q0	-	DC link current offset value	
u4_offset_iu_sum	uint32_t	Q0	-	U-phase current offset value integral value	
u4_offset_iv_sum	uint32_t	Q0	-	V-phase current offset value integral value	
u4_offset_iw_sum	uint32_t	Q0	-	W-phase current offset value integral value	
u4_offset_idc_ad_sum	uint32_t	Q0	-	DC link current offset value integral value	
u2_offset_calc_cnt	uint16_t	Q0	-	Offset current measurement count	
u2_offset_sample_cnt	uint16_t	Q0	-	Offset current measurement sample count	

Table 3-62 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_tscs_cint_t [Used in 3-shunt mode]”

Variable	Type	Qn	PU	Content	Remarks
s2_duty_u	int16_t	Q0	-	U-phase duty (PWM register setting)	
s2_duty_v	int16_t	Q0	-	V-phase duty (PWM register setting)	
s2_duty_w	int16_t	Q0	-	W-phase duty (PWM register setting)	
s2_offset_iu	int16_t	Q0	-	U-phase current offset value	
s2_offset_iv	int16_t	Q0	-	V-phase current offset value	
s2_offset_iw	int16_t	Q0	-	W-phase current offset value	
u2_crnt_ad[3]	uint16_t	Q13	-	UVW-phase current A/D conversion value	

Table 3-63 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_sscs_t [Used in 1-shunt mode]”

Variable	Type	Qn	PU	Content	Remarks
u2_offset_idc_adc	uint16_t	Q0	-	DC current offset	
u4_offset_ia_sum	uint32_t	Q0	-	Point A Current offset value Integral value	
u4_offset_ib_sum	uint32_t	Q0	-	Point B Current offset value Integral value	
u2_offset_calc_cnt	uint16_t	Q0	-	Number of offset current measurements	
u2_offset_sample_cnt	uint16_t	Q0	-	Number of offset current measurement samples	

Table 3-64 List of Variables in "r_mtr_foc_less_speed.h" / Structure: "st_mtr_sscs_cint_t [Used in 1-shunt mode]" (1/2)

Variable	Type	Qn	PU	Content	Remarks
u1_drv_pattern_get	uint8_t	Q0	-	PWM drive pattern to get	
u1_drv_pattern_set	uint8_t	Q0	-	PWM drive pattern to set	
u1_drv_pattern_put	uint8_t	Q0	-	PWM drive pattern to put	
u1_flag_ctrl_loop	uint8_t	Q0	-	Control loop flag	
u1_flag_mod_2ph	uint8_t	Q0	-	2-phase modulation flag	
s2_duty_max	int16_t	Q0	-	Maximum duty value	
s2_duty_mid	int16_t	Q0	-	Middle duty value	
s2_duty_min	int16_t	Q0	-	Minimum duty value	
s2_duty_u	int16_t	Q0	-	U-phase duty value	
s2_duty_v	int16_t	Q0	-	V-phase duty value	
s2_duty_w	int16_t	Q0	-	W-phase duty value	
s2_duty_max_adj	int16_t	Q0	-	Maximum duty adjustment value	
s2_duty_mid_adj	int16_t	Q0	-	Middle duty adjustment value	
s2_duty_min_adj	int16_t	Q0	-	Minimum duty adjustment value	
s2_duty_max_adj_comp	int16_t	Q0	-	Maximum duty adjustment compensation value	
s2_duty_mid_adj_comp	int16_t	Q0	-	Middle duty adjustment compensation value	
s2_duty_min_adj_comp	int16_t	Q0	-	Minimum duty adjustment compensation value	
s2_duty_u_adj	int16_t	Q0	-	U-phase duty adjustment value	
s2_duty_v_adj	int16_t	Q0	-	V-phase duty adjustment value	
s2_duty_w_adj	int16_t	Q0	-	W-phase duty adjustment value	
s2_duty_u_adjc	int16_t	Q0	-	U-phase duty adjustment compensation value	
s2_duty_v_adjc	int16_t	Q0	-	V-phase duty adjustment compensation value	
s2_duty_w_adjc	int16_t	Q0	-	W-phase duty adjustment compensation value	
s2_duty_diff_limit	int16_t	Q0	-	Minimum duty difference	
s2_duty_diff_limit_half	int16_t	Q0	-	Half of the minimum duty difference	
s2_duty_diff_limit2	int16_t	Q0	-	Minimum duty difference 2	
s2_ad_point_a_cnt	int16_t	Q0	-	A/D conversion point A timer count value	
s2_ad_point_b_cnt	int16_t	Q0	-	A/D conversion point B timer count value	
s2_mod_3ph2ph_speed_rad	int16_t	Q0	-	Switching speed from 3-phase modulation to 2-phase modulation	
s2_mod_2ph3ph_speed_rad	int16_t	Q0	-	Switching speed from 2-phase modulation to 3-phase modulation	
s2_ad_point_a_adj_cnt_2ph	int16_t	Q0	-	Adjustment value of A/D conversion point A count (2-phase modulation)	
s2_ad_point_b_adj_cnt_2ph	int16_t	Q0	-	Adjustment value of A/D conversion point B count (2-phase modulation)	
s2_ad_point_a_adj_cnt_3ph	int16_t	Q0	-	Adjustment value of A/D conversion point A count (3-phase modulation)	
s2_ad_point_b_adj_cnt_3ph	int16_t	Q0	-	Adjustment value of A/D conversion point B count (3-phase modulation)	
s2_ad_point_a_adj_cnt	int16_t	Q0	-	A/D conversion point A count adjustment value	
s2_ad_point_b_adj_cnt	int16_t	Q0	-	A/D conversion point B count adjustment value	

Table 3-65 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_sscs_cint_t [Used in 1-shunt mode]” (2/2)

Variable	Type	Qn	PU	Content	Remarks
s2_ad_ss_a	int16_t	Q0	-	Point A A/D conversion result	
s2_ad_ss_b	int16_t	Q0	-	Point A A/D conversion result	
s2_offset_ia	int16_t	Q0	-	Point A Current offset value	
s2_offset_ib	int16_t	Q0	-	Point B current offset value	
u2_state_duty_diff	uint16_t	Q0	-	2-phase match in 3-phase duty state	
u2_crnt_ad[2]	uint16_t	Q0	-	AB point current AD conversion value	

Table 3-66 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_common_foc_t”

Variable	Type	Qn	PU	Content	Remarks
u2_ctrl_conf	uint16_t	-	-	Control inputs	0x01: Current control 0x02: Speed control 0x04: Position control 0x08: Torque control 0x10: Voltage control
s1_direction	int8_t	-	-	Current direction of rotation	1: CW -1: CCW
s2_vdc_ad	int16_t	Q13	Voltage	Power source voltage	
s2_ref_v_d	int16_t	Q13	Voltage	d-axis reference Voltage	
s2_ref_v_q	int16_t	Q13	Voltage	q-axis reference Voltage	
s2_ref_i_d	int16_t	Q13	Current	d-axis reference Current	
s2_ref_i_q	int16_t	Q13	Current	q-axis reference Current	
s2_ad_i_d	int16_t	Q13	Current	d-axis ADC Current	
s2_ad_i_q	int16_t	Q13	Current	q-axis ADC Current	
s2_phe_ed	int16_t	Q13	Voltage	d-axis BEMF [PU(V)]	
s2_phe_phase_err_rad	int16_t	Q12	Angle	Phase error [PU(rad)]	
s2_asr_ref_speed_rad_ctrl	int16_t	Q14	Speed	Reference speed for speed PI control [PU(rad/s)]	
s2_asr_speed_rad	int16_t	Q14	Speed	Speed operation value [PU(rad/s)]	
s2_damp_ref_speed_rad_ctrl	int16_t	Q14	Speed	Reference speed for Open loop Damping Control	

Table 3-67 List of Variables in "r_mtr_foc_less_speed.h" / Structure: "st_mtr_common_cint_t"

Variable	Type	Qn	PU	Content	Remarks
u2_ctrl_conf	uint16_t	-	-	Control inputs	0x01: Current control 0x02: Speed control 0x04: Position control 0x08: Torque control 0x10: Voltage control
s1_direction	int8_t	-	-	Current direction of rotation	1: CW -1: CCW
s2_vdc_ad	int16_t	Q13	Voltage	Power source voltage	
s2_ref_v_d	int16_t	Q13	Voltage	d-axis reference Voltage	
s2_ref_v_q	int16_t	Q13	Voltage	q-axis reference Voltage	
s2_ref_i_d	int16_t	Q13	Current	d-axis reference Current	
s2_ref_i_q	int16_t	Q13	Current	q-axis reference Current	
s2_ad_i_d	int16_t	Q13	Current	d-axis ADC Current	
s2_ad_i_q	int16_t	Q13	Current	q-axis ADC Current	
s2_phe_ed	int16_t	Q13	Voltage	d-axis BEMF [PU(V)]	
s2_phe_phase_err_rad	int16_t	Q12	Angle	Phase error [PU(rad)]	
s2_asr_ref_speed_rad_ctrl	int16_t	Q14	Speed	Reference speed for speed PI control [PU(rad/s)]	
s2_asr_speed_rad	int16_t	Q14	Speed	Speed operation value [PU(rad/s)]	
s2_damp_ref_speed_rad_ctrl	int16_t	Q14	Speed	Reference speed for Open loop Damping Control	

Table 3-68 List of Variables in "r_mtr_foc_less_speed.h" / Structure: "st_mtr_foc_t" (1/2)

Variable	Type	Qn	PU	Content	Remarks
s2_damp_speed	int16_t	Q14	Speed	Damping speed output	
s2_damp_ref_speed_rad_ctrl	int16_t	Q14	Speed	Damping reference speed	
s2_limit_over_voltage	int16_t	Q13	Voltage	Overvoltage limit value	
s2_limit_under_voltage	int16_t	Q13	Voltage	Undervoltage limit value	
s2_phase_err_lpf_rad	int16_t	Q14	Speed	Phase error LPF	
u1_flag_charge_cap	uint8_t	-	-	Current offset value calculation flag	0: Execute offset calculation process 1: Offset calculation process completed
u1_flag_down_to_ol	uint8_t	-	-	Open loop transition flags	0: No transition 1: Execute transition
u1_flag_draw_in	uint8_t	-	-	Id draw-in flag	0: Unreached 1: Reach the reference
u1_state_drive	uint8_t	-	-	Drive mode status	0: Offset is being removed 1: Offset removal completed 2: IPD processing execution 3: IPD processing completed 4: Start driving 5: d-axis current 0 drive 6: Brake processing 7: Drive stop
u1_state_ref_id	uint8_t	-	-	d-axis current command value generation status	0: d-axis current 0 1: d-axis current manual control 2: Field weakening control/maximum torque control
u1_state_ref_iq	uint8_t	-	-	q-axis current command value generation status	0: q-axis current 0 1: q-axis current manual control 2: Open loop to Closed loop switch Control 3: Speed PI output
u1_state_ref_speed	uint8_t	-	-	Speed command value generation status	0: Speed 0 1: Speed change

Table 3-69 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_foc_t” (2/2)

Variable	Type	Qn	PU	Content	Remarks
u2_draw_in_wait_cnt	uint16_t	Q0	-	Wait times of Id draw-in	
u2_draw_in_time_calc_cnt	uint16_t	Q0	-	Wait times counter of Id draw-in	
u2_run_mode	uint16_t	-	-	Operating modes	0x00: Init mode 0x01: Boot mode 0x02: Drive mode 0x03: Analysis mode 0x04: Tune mode
u2_error_status	uint16_t	-	-	Error status	0x0000: No error 0x0001: Overcurrent error (hardware) 0x0002: Overvoltage error 0x0004: Rotational speed error 0x0008: Hall timeout error 0x0010: BEMF timeout error 0x0020: Hall pattern error 0x0040: BEMF pattern error 0x0080: Undervoltage error 0x0100001: Overcurrent error (software) 0x0200: TRX overflow error 0xFFFF: Undefined error
u1_flg_foc_run	uint8_t	-	-	Execution flag of 50us carrier control	0: Stop of carrier control 1: Run of carrier control
u1_flg_get_vdc	uint8_t	-	-	Get vdc flag	0: Get vdc 1: Don’t get vdc
u1_flag_rdy_trans_faa	uint8_t	-	-	-	
u1_flag_pack	uint8_t	-	-	-	
st_stm	st_mtr_statemachine_t	-	-	Structure for state machine	
st_tscs	st_mtr_tscs_t	-	-	Structure for 3-phase current detection	[Used in 3-shunt mode]
st_sscs	st_mtr_sscs_t	-	-	Structure for 1-shunt resistor current detection	[Used in 1-shunt mode]
st_acr	st_mtr_acr_t	-	-	ACR structure	Current PI control
st_asr	st_mtr_asr_t	-	-	ASR structure	Speed PI control
st_ipd	st_mtr_ipd_t	-	-	Structure for initial position detection	
st_m2f	st_mtr_mtpa2fw_t	-	-	Structure for maximum torque and field weakening control	
st_ol2cl	st_mtr_ol2cl_t			Structure for Open loop to Closed loop switch Control	
st_pe_lpf	st_mtr_lpf1_t			Structure for phase error lpf	
st_do	st_mtr_do_t	-	-	Disturbance suppression structure	
st_common	st_mtr_common_foc_t	-	-	1ms/50us frequency common structure	

Table 3-70 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_cint_t”

Variable	Type	Qn	PU	Content	Remarks
s2_limit_over_current	int16_t	Q13	Current	Over current limit value	
u1_flag_over_current	uint8_t	-	-	-	
st_ad_i	st_coordinate13	-	-	ADC current coordinate system	
st_ref_v	st_coordinate13	-	-	Reference voltage coordinate system	
st_ref_i	st_coordinate13	-	-	Reference current coordinate system	
st_i_repro	st_coordinate13	-	-	Reproduced current coordinate system	
st_iq_repro	st_mtr_lpf1_t	-	-	q-axis current LPF for current reproduction	
st_id_repro	st_mtr_lpf1_t	-	-	d-axis current LPF for current reproduction	
st_motor	st_mtr_parameter_t	-	-	Structure for motor parameter	
st_phe	st_mtr_est_phe_t	-	-	Structure for phase error estimate	
st_tscs	st_mtr_tscs_cint_t	-	-	Structure for 3-phase current detection	[Used in 3-shunt mode]
st_sscs	st_mtr_sscs_cint_t	-	-	Structure for 1-shunt resistor current detection	[Used in 1-shunt mode]
st_acr	st_mtr_acr_cint_t	-	-	ACR structure	Current PI control
st_asr	st_mtr_asr_cint_t	-	-	ASR structure	Speed PI control
st_mod	st_mtr_mod_t	-	-	Structure for modulation	
st_pll	st_mtr_pll_t	-	-	Structure for PLL control	
st_dt_comp	st_mtr_deadtime_comp_t	-	-	Structure for deadtime compensation	
st_common	st_mtr_common_cint_t	-	-	1ms/50us frequency common structure	

Table 3-71 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_data_t”

Variable	Type	Qn	PU	Content	Remarks
p_st_foc	st_mtr_foc_t	-	-	1ms periodic control structure pointer	
p_st_cint	st_mtr_cint_t	-	-	50us periodic control structure pointer	

Table 3-72 List of Variables in “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 judgement	
u1_flag_cmp0_intr	uint8_t	Q0	-	Flag for CMP0 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	uint16_t	Q0	-	Discharge period of angle detection for salient rotor	
u2_sal_polarity_discharge	uint16_t	Q0	-	Discharge period of polarity detection for salient rotor	
u2_non_sal _discharge	uint16_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 3-73 List of Variables in “r_mtr_mtpa2fw_ctrl.h” / Structure: “st_mtr_mtpa2fw_t” (1/2)

Variable	Type	Qn	PU	Content	Remarks
s2_ia_max	int16_t	Q13	Current	Maximum current [PU]	
s2_vlim	int16_t	Q13	Voltage	Limit voltage [PU]	
s2_id_lim	int16_t	Q13	Current	d-axis current limit [PU]	
s2_iq_lim	int16_t	Q13	Current	q-axis current limit [PU]	
s2_ref_id	int16_t	Q13	Current	Command d-axis current value [PU]	
s2_ref_iq	int16_t	Q13	Current	Command q-axis current value [PU]	
s2_mtr_r	int16_t	Q17	Resistance	Resistance [PU]	
s2_mtr_ld	int16_t	Q18 (1 shunt) / Q19 (3 shunt)	Inductance	d-axis inductance [PU]	
s2_mtr_lq	int16_t	Q18 (1 shunt) / Q19 (3 shunt)	Inductance	q-axis inductance [PU]	
s2_mtr_m	int16_t	Q15	Back-EMF constant	Back-EMF constant [PU]	
s2_mtpa_ref_id	int16_t	Q13	Current	d-axis current value for MTPA [PU]	
s2_va_max_gain_fp	int16_t	Q13	Voltage	Va_max gain for fixed point	
s2_mtpa_cof_i	int16_t	Q13	Current	MTPA current coeficiency (ld - lq) / ke	
s2_speed_err_th_rad	int16_t	Q14	Speed	Speed error threshold [PU]	
s2_speed_err_rad	int16_t	Q14	Speed	Speed error [PU]	
s2_speed_err_rad_lpf	int16_t	Q14	Speed	Speed error LPF [PU]	
s2_va	int16_t	Q13	Voltage	Voltage magnitude [PU]	
s2_vd	int16_t	Q13	Voltage	d-axis voltage [PU]	
s2_vq	int16_t	Q13	Voltage	q-axis voltage [PU]	
s2_fw_pi_ref_id	int16_t	Q13	Current	PI output d-axis current [PU]	
s2_fw_ref_id	int16_t	Q13	Current	Reference d-axis current [PU]	
s2_fw_ref_id_ctrl	int16_t	Q13	Current	Command reference d-axis current [PU]	
s2_limit_current	int16_t	Q13	Current	Limit of current value [PU]	
s2_ramp_limit_current	int16_t	Q13	Current	Limit of current value change [PU]	
s2_fw_id_inc_const	int16_t	Q13	Current	Field-Weakening Id control constant	
u1_fw_pi_intg_redct_cof	uint8_t	-	-	Field-Weakening Id pi integral term reduction coefficient	

Table 3-74 List of Variables in “r_mtr_mtpa2fw_ctrl.h” / Structure: “st_mtr_mtpa2fw_t” (2/2)

Variable	Type	Qn	PU	Content	Remarks
u1_q_iq_mtpa_cofi_calc	uint8_t	-	-	Q format of iq * (ld - lq) / ke	
u1_q_voltage	uint8_t	-	-	Q-format of voltage	
u1_q_current	uint8_t	-	-	Q-format of current	
u1_q_id_mtpa_calc	uint8_t	-	-	Q-format of id reference for MTPA	
u1_q_r_mul_i	uint8_t	-	-	Q format of resistance multiplied by current	
u1_q_speed_mul_l	uint8_t	-	-	Q format of speed multiplied by inductance	
u1_q_speed_mul_m	uint8_t	-	-	Q-format of speed multiplied by Back-EMF constant	
u1_q_wl_mu_i	uint8_t	-	-	Q-format of wl multiplied by current	
u1_q_fw_kp	uint8_t	-	-	Q-format of FW PI proportional gain	
u1_q_fw_kidt	uint8_t	-	-	Q-format of FW PI ki * dt	
u1_q_fw_lpf_co	uint8_t	-	-	Q-format of LPF numerator for Field-Weakening Control	
u1_pi_on_flag	uint8_t	-	-	PI ON flag	
u1_hys_flag	uint8_t	-	-	hysteresis flag	
st_pi	st_mtr_fw_pi_t	-	-	PI controller parameter structure	
st_lpf	st_mtr_fw_lpf1_t	-	-	1st order LPF structure	

Table 3-75 List of Variables in “r_mtr_mtpa2fw_ctrl.h” / Structure: “st_mtr_fw_pi_t”

Variable	Type	Qn	PU	Content	Remarks
s2_kp	int16_t	Q12	Voltage/Current	Proportional gain	
s2_kidt	int16_t	Q19	Voltage/Current	Integral gain x control period	
s2_intg	int16_t	Q26	Voltage	Integral term	
s2_ilimit	int16_t	Q26	Voltage	Integral limit (up/down symmetry)	
s2_deadband	int16_t	Q26	Current	Deadband of integral part	

Table 3-76 List of Variables in “r_mtr_mtpa2fw_ctrl.h” / Structure: “st_mtr_fw_lpf1_t”

Variable	Type	Qn	PU	Content	Remarks
s2_in_k	int16_t	Q15	-	LPF input gain	
s2_out_k	int16_t	Q15	-	LPF previous output gain	
s2_pre_out	int16_t	Q13	Current	Previous output value	

Table 3-77 List of Variables in “r_mtr_ol2cl.h” / Structure: “st_mtr_hpf1_t”

Variable	Type	Qn	PU	Content	Remarks
s2_k	int16_t	Q15	-	HPF gain	
s2_pre_input	int16_t	Q13	Voltage	Previous value of input	
s2_pre_output	int16_t	Q13	Voltage	Previous value of output	
u1_q_hpf_co	int16_t	Q0	-	HPF gain Q value	

Table 3-78 List of Variables in “r_mtr_ol2cl.h” / Structure: “st_mtr_damp_t”

Variable	Type	Qn	PU	Content	Remarks
s2_k	int16_t	Q13 (1 shunt) / Q12 (3 shunt)	Speed/Voltage	Damping control gain	
s2_speed_limit_rate	int16_t	Q14	Speed	Speed limit	
u1_q_damp_speed_calc	int8_t	Q0	-	Damping control speed Q value	
u1_q_damp_speed_limit_calc	Int8_t	Q0	-	Damping control speed limit Q value	
st_hpf	st_mtr_hpf1_t	-	-	HPF structure	

Table 3-79 List of Variables in “r_mtr_ol2cl.h” / Structure: “st_mtr_ol2cl_t”

Variable	Type	Qn	PU	Content	Remarks
s2_ramp_limit_current	int16_t	Q13	Current	Current rise limit value during switching process [A/ms]	
s2_temp_ramp_limit_current	int16_t	Q13	Current	Temporarily saved data of current rise limit value [A/ms]	
s2_q_axis_current	int16_t	Q13	Current	q-axis current during switching process	
s2_theta2crnt_k0	int16_t	Q13	-	Reference current = k0 + k1 × (phase error)	
s2_theta2crnt_k1	int16_t	Q21	-	Reference current = k0 + k1 × (phase error)	
s2_cl_swich_phase_err_rad	int16_t	Q12	Angle	Phase error during sensorless switching processing	
u2_switch_cnt	uint16_t	Q0	-	Switching time [cnt]	
u1_q_theta2crnt_k1	uint8_t	Q0	-	Q value for angular error of reference current	
u1_q_idelta_calc	uint8_t	Q0	-	Q value for calculating the q-axis component of current	
st_ph_err	st_sincos12	-	-	Phase error angle structure	

Table 3-80 List of Variables in “r_mtr_disturb_suppress.h” / Structure: “st_mtr_do_lpf1_t”

Variable	Type	Qn	PU	Content	Remarks
s2_in_k	int16_t	Q15	-	LPF input gain	
s2_out_k	int16_t	Q15	-	LPF previous output gain	
s2_pre_out	int16_t	Q13	Current	Previous output value	

Table 3-81 List of Variables in “r_mtr_disturb_suppress.h” / Structure: “st_mtr_do_hpf1_t”

Variable	Type	Qn	PU	Content	Remarks
s2_k	int16_t	Q15	-	HPF gain	
s2_pre_input	int16_t	Q13	Current	Previous value of input	
s2_pre_output	int16_t	Q13	Current	Previous value of output	
u1_q_hpf_co	uint8_t	Q0	-	HPF gain Q value	

Table 3-82 List of Variables in “r_mtr_disturb_suppress.h” / Structure: “st_mtr_do_t”

Variable	Type	Qn	PU	Content	Remarks
u1_q_calc_hpf	uint8_t	Q0	-	Q value for HPF coefficient calculation	
u1_q_calc_lpf	uint8_t	Q0	-	Q value for LPF coefficient calculation	
u1_q_lpf_lpf_co	uint8_t	Q0	-	Q value of LPF	
s2_ref_current_buff	int16_t	Q13	Current	Buffer value of reference current	
s2_j_div_tc_p2m	int16_t	Q26	-	(Inertia) / (Time constant of LPF × pole pair^2 × induced voltage constant)	
s2_disturbance	int16_t	Q13	Current	Disturbance	
st_lpf	st_mtr_do_lpf1_t	-	-	Primary LPF structure	
st_hpf	st_mtr_do_hpf1_t	-	-	Primary HPF structure	

3.5 List of Sensorless Vector Control Software Macro Definitions

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

Table 3-83 List of Macro Definitions in “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	
SINGLE_SHUNT	0	1-shunt mode	
THREE_SHUNT	1	3-shunt mode	
CURRENT_SENS_METHOD	SINGLE_SHUNT THREE_SHUNT	Current detection method(Note)	
USE_DEADTIME_COMP	0 : 1	Select deadtime compensation process (1:Enable 0:Disable)	Default setting 1
USE_SPEED_LPF	0 : 1	Select speed LPF (1:Enable 0:Disable)	Default setting 1
USE_CURRENT_LPF_IQ	0 : 1	Select q-axis current LPF (1:Enable 0:Disable)	Default setting 0
USE_CURRENT_LPF_ID	0 : 1	Select d-axis current LPF (1:Enable 0:Disable)	Default setting 0
USE_MTPA_AND_FW	0 : 1	Select Field-Weakening control/maximum torque control (1:Enable 0:Disable)	Default setting 1
USE_OPENLOOP_DAMPING	0 : 1	Select Openloop damping Control (1:Enable 0:Disable)	Default setting 1
USE_OL2CL_CTRL	0 : 1	Select Open loop to Closed loop switch Control(1:Enable 0:Disable)	Default setting 1
USE_DUTY_2PH_CROS_COMP	0 : 1	Current compensation for 2-phase Duty cross	Default setting 0
USE_DISTURBANCE_SUPPRESS	0 : 1	Select disturbance suppression function	Default setting 0
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
MOD_3PH_SPWM	0	Sine wave modulation	
MOD_3PH_TOW	1	Third harmonic calculation	
MOD_2PH_BOT	2	2-phase modulation	
MOD_METHOD	MOD_3PH_SPWM MOD_3PH_TOW MOD_2PH_BOT	Modulation method	Default setting MOD_3PH_TOW

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

Table 3-84 List of Macro Definitions in “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/m]	
MP_ROTOR_INERTIA	0.00000205f	Inertia [kgm^2/rad^2]	
MP_FRICTION_0TH_ORDER	0.002748f	Static friction coefficient [kgm^2/(rad･s^2)]	
MP_FRICTION_1ST_ORDER	0.000001873f	Kinetic friction coefficient [kgm^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 3-85 List of Macro Definitions in “r_mtr_control_parameter.h” [1/3]

Macro	Definition value	Description	Remarks
CP_PWM_TIMER_FREQ	96.0f	PWM timer frequency [MHz]	
CP_INTVAL_TIMER_FREQ	48.0f	Interval timer frequency [MHz]	
CP_CARRIER_FREQ	20.0f	PWM carrier frequency	
CP_TRX_TIMER_FREQ	96.0f	TRX timer frequency [MHz]	
CP_INT_DECIMATION	1	Interrupt thinning count	
CP_V_PHASE_LEAD_COEF	1	Voltage leading phase coefficient	
CP_SPEED_CTRL_PERIOD	0.001f	Speed control period [seconds]	
CP_AD_CONVERSION_TIME	1.14583f	A/D conversion time	
CP_AD_RINGING_WAIT_CNT	264	Current ringing wait time at AD conversion	
CP_AD_RINGING_WAIT_2PH_CNT	264	Current ringing wait time at AD conversion for 2-phase modulation	
CP_ACR_NF_HZ	500	Current PI control natural frequency [Hz]	
CP_ASR_NF_HZ	11.19f	Speed PI control natural frequency [Hz]	
CP_PLL_NF_HZ	55.95f	PLL control natural frequency [Hz]	
CP_ACR_DEADBAND_LSB	0.05f	Current PI deadband [LSB]	
CP_ASR_DEADBAND_LSB	0.05f	Speed PI deadband [LSB]	
CP_PLL_DEADBAND_LSB	0.05f	PLL control deadband [LSB]	
CP_ASR_LPF_COF_HZ	139.88f	Speed LPF cutoff frequency [Hz]	
CP_ACR_LPF_COF_HZ	2000	Current LPF cutoff frequency [Hz]	
CP_I_REPRO_COF_HZ	2000	Cutoff frequency of LPF for current reproduction [Hz]	
CP_ASR_KI_AUG	8	Augmentation rate for integral part of ASR	
CP_MAX_SPEED_RPM	4300 (1 shunt) / 3900 (3shunt)	Maximum speed (mechanical angle) [rpm]	
CP_SPEED_LIMIT_RPM	5590 (1 shunt) / 4290 (3 shunt)	Speed limit (mechanical angle) [rpm]	
CP_OC_LIMIT	1.47f	Overcurrent limit value [A]	
CP_OL_REF_ID	0.42f	d-axis current command value [A]	

Table 3-86 List of Macro Definitions in “r_mtr_control_parameter.h” [2/3]

Macro	Definition value	Description	Remarks
CP_DRAW_IN_WAIT_TIME	0.2f	Draw-in wait time count value	
CP_INIT_ASR_INTEG	0. 181244f	q-axis current PI integral term PI initial value [A]	
CP_LAMP_LIMIT_CURRENT	0.00084f	Limit value for current rise [PU/ms]	
CP_MOD_3PH2PH_SPEED_RPM	1060	Switching speed from 3-phase to 2-phase modulation [rpm]	[Used in 1-shunt mode and 2-phase modulation]
CP_MOD_2PH3PH_SPEED_RPM	927	Switching speed from 2-phase to 3-phase modulation [rpm]	
CP_OL2CL_SPEED_RPM	795	Speed to start decreasing id [rpm]	
CP_CL2OL_SPEED_RPM	530	Speed to start increasing id [rpm]	
CP_CL2OL_JUDGE_WAIT_TIME	28.302f	Waiting time to judge down to open loop [ms]	
CP_LAMP_LIMIT_SPEED_RPM	1.677845f	Limit of acceleration [rpm/ms]	
CP_RAMP_SPEED_CNT_DECIMATION	0	Number of decimations out of acceleration limit	
CP_OFFSET_CALC_TIME	128	Current offset value calculation time [ms]	
CP_OFFSET_CALC_ST_WAIT_CNT	100	Offset calculation stable waiting time	
CP_AD_POINT_A_ADJ_CNT	264	Adjustment for A/D delay counts for A point	[Used in 1-shunt mode]
CP_AD_POINT_B_ADJ_CNT	264	Adjustment for A/D delay counts for B point	
CP_AD_POINT_A_ADJ_2PH_CNT	264	Adjustment for A/D delay counts for A point	[Used in 1-shunt mode and 2-phase modulation]
CP_AD_POINT_B_ADJ_2PH_CNT	100	Adjustment for A/D delay counts for B point	
CP_MOD_2PH_BOT_CHANGE_CNT	200	2-phase modulation phase switching judgment counter	
CP_SAL_ANGLE_CURRENT	0.25f	Threshold current of angle detection for salient rotor	【IPD】
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_DISCHARGE	689	Discharge period of angle detection for non-salient rotor	
CP_IPD_NOISE_AVOID_CNT	97	Noise avoidance count at IPD	
CP_FW_CURRNT_LIMIT_RATIO	0.5f	Current limit ratio for Field-weakening contro	【MTPA/FW】
CP_FW_PI_NF_HZ	0	PI natural frequency for Field-Weakening Control	
CP_FW_SPEED_ERR_THRESHOLD_RPM	100	Speed error threshold for Field-Weakening Control	
CP_FW_SPEED_ERR_LPF_COF_HZ	10	Speed error LPF for Field-Weakening Control	
CP_FW_ID_INC_CONST	1	Field-Weakening Id control constant	
CP_FW_PI_INTG_REDCT_COEF	2	Field-Weakening Id pi integral term reduction coefficient	

Table 3-87 List of Macro Definitions in “r_mtr_control_parameter.h” [3/3]

Macro	Definition value	Description	Remarks
CP_DAMP_HPF_COF_HZ	5	HPF cutoff frequency for damping control[Hz]	【Open loop damping】
CP_DAMP_ZETA	1	Damping control damping coefficient	
CP_DAMP_SPEED_LIMIT_RATE	0.2f	Damping control speed limit	
CP_PHASE_ERR_LPF_COF_HZ	10	Phase error LPF cutoff frequency for Open loop to Closed loop switch control [Hz]	【OL2CL】
CP_OL2CL_SWITCH_TIME	0.1095f	Time[s] to switch open loop to sensor-less	
CP_OL2CL_SWITCH_ANGLE_MIN	5	Minimum angle for switching to open loop [deg]	
CP_DO_LPF_COF_HZ	0.001f	Filter coefficients for disturbance suppression [Hz]	【Disturbance observer】

Table 3-88 List of Macro Definitions in “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 3-89 List of Macro Definitions “r_mtr_scaling_parameter.h” [1/2]

Macro	Definition value	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	Current standard value [A]	
PU_BASE_VOLTAGE_V	IP_INPUT_V	Voltage standard value [A]	
PU_BASE_FREQ_Hz	MTR_TWOPI*(CP_MAX_SPEED_RPM+1)*MP_POLE_PAIRS/60	Frequency standard value [Hz]	
PU_BASE_ANGLE_Rad	1.0f	Angle standard value [rad]	
PU_SF_CURRENT	1.0f / PU_BASE_CURRENT_A	Current scale [1/A]	
PU_SF_VOLTAGE	1.0f / PU_BASE_VOLTAGE_V	Voltage scale [1/V]	
PU_SF_AFREQ	1.0f / PU_BASE_FREQ_Hz	Angular frequency scale [s/rad]	
PU_SF_ANGLE	1.0f / PU_BASE_ANGLE_Rad	Angle scale [1/rad]	
PU_SF_TIME	PU_SF_ANGLE / PU_SF_AFREQ	Time scale [1/s]	
PU_SF_RES	PU_SF_VOLTAGE / PU_SF_CURRENT	Resistance scale [1/Ω]	
PU_SF_IND	PU_SF_RES / PU_SF_AFREQ	Inductance scale [rad/H]	
PU_SF_BEMF_CONST	PU_SF_VOLTAGE / PU_SF_AFREQ	BEMF constant scale [rad/(V･s)]	
PU_SF_INERTIA	PU_SF_BEMF_CONST * PU_SF_CURRENT / (MP_POLE_PAIRS * MP_POLE_PAIRS * PU_SF_AFREQ * PU_SF_AFREQ)	Inertia scale [rad^2/kg･m^2]	
PU_SF_D1_DIV_P2M	(PU_SF_CURRENT / PU_SF_AFREQ)	Scale pf kinetic friction coefficient/((number of pole pairs)^2 × BEMF constant) [A･s/rad]	
PU_SF_RPM_RAD	1.0f / (CP_MAX_SPEED_RPM+1)	Scale of conversion from [rpm] to [rad/s]	
PU_SF_RAD_RPM	CP_MAX_SPEED_RPM+1	Scale of conversion from [rad/s] to [rpm]	
PU_SF_ACR_KP	PU_SF_RES	Current PI proportional gain scale	
PU_SF_ACR_KIDT	PU_SF_RES	Current PI integral gain scale	
PU_SF_ASR_KP	PU_SF_CURRENT / PU_SF_AFREQ	Speed PI proportional gain scale	
PU_SF_ASR_KIDT	PU_SF_CURRENT / PU_SF_AFREQ	Speed PI integral gain scale	
PU_SF_PLL_KP	PU_SF_AFREQ / PU_SF_ANGLE	PLL proportional gain scale	
PU_SF_PLL_KIDT	PU_SF_AFREQ / PU_SF_ANGLE	PLL integral gain scale	
MTR_Q_ANGLE	12	Q-format of angle	
MTR_Q_CURRENT	13	Q-format of current	
MTR_Q_VOLTAGE	13	Q-format of voltage	
MTR_Q_VMOD	12	Q-format of PWM modulation factor	
MTR_Q_AFREQ	14	Q-format of angular frequency	
MTR_Q_CTRL_TIME	18	Q-format of FOC control cycle	
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 BEMF constant	
MTR_Q_INERTIA	9 (1 shunt) / 10 (3 shunt)	Q-format of inertia	

Table 3-90 List of Macro Definitions “r_mtr_scaling_parameter.h” [2/2]

Macro	Definition value	Description	Remarks
MTR_Q_D1_DIV_P2M	18	Q-format of kinetic friction coefficient/ ((number of pole pairs)^2 × BEMF constant)	
MTR_Q_RECIV	13	Q-format of inverse voltage	
MTR_Q_RECIM	14	Q-format of reciprocal of BEMF constant	
MTR_Q_ACR_KP	17	Q-format of speed PI proportional gain	
MTR_Q_ACR_KIDT	19	Q-format of speed PI integral gain×control period	
MTR_Q_ASR_KP	12	Q-format of current PI proportional gain	
MTR_Q_ASR_KIDT	20	Q-format of current PI integral gain×control period	
MTR_Q_PLL_KP	15	Q-format of PLL proportional gain	
MTR_Q_PLL_KIDT	23	Q-format of PLL integral gain×control period	
MTR_Q_SPEED_LPF_CO	15	Q-format of speed LPF gain	
MTR_Q_CURRENT_LPF_CO	15	Q-format of current LPF gain	
MTR_Q_CURRENT_REPO_LPF_CO	15	Q-format of current reproduction LPF gain	
MTR_Q_DAMP_K	13 (1 shunt) / 12 (3 shunt)	Q-format of damping control gain	
MTR_Q_DAMP_HPF_CO	15	Q-format of damping control HPF gain	
MTR_Q_DAMP_SL_RATE	17	Q-format of damping control speed limit	
MTR_Q_PHERR_LPF_CO	15	Q-format of Phase error LPF gain	
MTR_Q_OL2CL_K1	21	Q-format of Open loop to Closed loop switch Control gain	
MTR_Q_MTPA_COEF_I	21	Q-format of ((q-axis inductance- d-axis inductance))/BEMF-const	
MTR_Q_I_MUL_COEF_I	19	Q-format of iq * iq* ((q-axis inductance- d-axis inductance))/BEMF-const calculation	
MTR_Q_FW_KP	19	Q-format of PI proportional gain for Field-Weakening Control	
MTR_Q_FW_KIDT	26	Q-format of PI (integral gain) * (sampling period) for Field-Weakening Control	
MTR_Q_FW_LPF_CO	15	Q-format of speed error LPF coefficient for Field-Weakening Control	
MTR_Q_DO_LPF_CO	15	Q-format of filter coefficients for disturbance suppression	
MTR_Q_J_DIV_TC_P2M	26	Q-format of kinetic friction coefficient/ ((number of pole pairs)^2 × BEMF constant)	
MTR_Q_V_PHASE_LEAD_COEF	14	Q-format of voltage leading phase coefficient	
MTR_Q_DIV_DSP	16	Q-format of DSP function division	
MTR_Q_SIN_COS_DSP	14	Q-format of trigonometric function of DSP function	

Table 3-91 List of Macro Definitions in “main.h”

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

Table 3-92 List of Macro Definitions in “ICS_define.h”

Macro	Definition value	Description	Remarks
RL78	-	CPU definition	

Table 3-93 List of Macro Definitions in “r_mtr_ics.h”

Macro	Definition value	Description	Remarks
TS_300US	-	ICS communication speed	
ICS_BRR	23	ICS bit rate register selection	
MTR_ICS_DECIMATION	2	Number of pixels skipped in ICS processing	
ICS_ADDR	0xFE00	Address of ICS	
ICS_INT_LEVEL	3	ICS interrupt level setting	
ICS_NUM	0x40	Data size of ICS communication	
ICS_MODE	0	ICS interrupt mode setting	

Table 3-94 List of Macro Definitions in “r_mtr_rl78g24.h” [1/2]

Macro	Definition value	Description	Remarks
USE_PWMOPA	MTR_SET	Select of overcurrent use by PWMOPA	
MTR_MAIN_CLOCK_FREQ	48.0f	CPU main clock [MHz]	
MTR_INT_DECIMATION	CP_INT_DECIMATION	Interrupt processing carrier pixel skipping	
TRD_SKIPPING_COUNT	CP_INT_DECIMATION	TRD interrupt skipping count	
MTR_PWM_TIMER_FREQ	CP_PWM_TIMER_FREQ	PWM timer frequency [kHz]	
MTR_INTVAL_TIMER_FREQ	CP_INTVAL_TIMER_FREQ	Interval timer frequency [kHz]	
MTR_CARRIER_FREQ	CP_CARRIER_FREQ	Carrier interrupt frequency [kHz]	
MTR_INVTVAL_PERIOD	(MTR_INT_DECIMATION + 1) * 1000.0f / (MTR_CARRIER_FREQ)	Interval timer cycle [μs]	
MTR_DEADTIME	IP_DEADTIME	Deadtime [μs]	
MTR_DEADTIME_CNT	(int16)(MTR_DEADTIME * MTR_PWM_TIMER_FREQ)	Deadtime settings	
MTR_CARRIER_CNT	(uint16_t)(MTR_PWM_TIMER_FREQ * 1000 / MTR_CARRIER_FREQ * 0.5f)	Carrier settings	
MTR_HALF_CARRIER_CNT	(uint16)(MTR_CARRIER_SET * 0.5f)	Carrier settings (intermediate value)	
MTR_CARRIER_DOWN_CNT	(uint16_t)(MTR_PWM_TIMER_FREQ * 1000 / MTR_CARRIER_FREQ * 0.5f)	Max setting value of carrier period	
MTR_HALF_CARRIER_DOWN_CNT	(uint16_t)(MTR_CARRIER_DOWN_CNT * 0.5f)	Half of duty value	
MTR_HALF_DEADTIME_CNT	(uint16_t)(MTR_DEADTIME_CNT * 0.5f)	Half of the dead time setting value	
MTR_DUTY_RANGE_CNT	(uint16_t)((MTR_CARRIER_CNT-MTR_DEADTIME_CNT)*0.5f)	Count value of modulation rate range	
MTR_CURRENT_ADCONV_TIME	CP_AD_CONVERSION_TIME * 2.0f	Time [μs] taken for 2-phase current A/D conversion	[Used in 3-shunt mode]
MTR_VOLTAGE_LIMIT_OFFSET	(int16_t)(((MTR_CURRENT_ADCONV_TIME + MTR_DEADTIME * 2) / (1000/MTR_CARRIER_FREQ)) * 0.5f * (1 << MTR_Q_VOLTAGE))	Voltage offset limit [PU (V)]	
MTR_CENTER_AMPLITUDE_CNT	(uint16_t)((MTR_CARRIER_CNT+CP_AD_RINGING_WAIT_CNT+MTR_DEADTIME_CNT) * 0.5f)	PWM timer center amplitude	
MTR_CURRENT_ADCONV_TIME	CP_AD_CONVERSION_TIME * 2.0f	Time [μs] taken for 2-phase current A/D conversion	[Used in 3-shunt mode and 2-phase modulation]
MTR_VOLTAGE_LIMIT_OFFSET	0	Voltage offset limit [PU (V)]	
MTR_CENTER_AMPLITUDE_CNT	(int16_t)(MTR_HALF_CARRIER_CNT + MTR_DEADTIME_CNT)	PWM timer center amplitude	
MTR_AD_MINIMUM_TIME	(int16_t)(CP_AD_CONVERSION_TIME*MTR_PWM_TIMER_FREQ+CP_AD_RINGING_WAIT_CNT+MTR_DEADTIME_CNT)	A/D conversion time count value	[Used in 1-shunt mode]
MTR_AD_MINIMUM_TIME2	50	Minimum A/D conversion time	
MTR_VOLTAGE_LIMIT_OFFSET	(int16_t)(((MTR_DEADTIME*2)/(1000/MTR_CARRIER_FREQ))*0.5f*(1<<MTR_Q_VOLTAGE))	Voltage offset limit [PU (V)]	
MTR_AD_MINIMUM_TIME	(int16_t)(CP_AD_CONVERSION_TIME*MTR_PWM_TIMER_FREQ+CP_AD_RINGING_WAIT_CNT+MTR_DEADTIME_CNT)	A/D conversion time count value	[Used in 1-shunt mode and 2-phase modulation]
MTR_AD_MINIMUM_TIME2	50	Minimum A/D conversion time	
MTR_CURRENT_ADCONV_TIME	CP_AD_CONVERSION_TIME * 2.0f	Time [μs] taken for 2-phase current A/D conversion	
MTR_VOLTAGE_LIMIT_OFFSET	0	Voltage offset limit [PU (V)]	
MTR_CENTER_AMPLITUDE_CNT	(int16_t)(MTR_HALF_CARRIER_CNT + MTR_DEADTIME_CNT)	PWM timer center amplitude	

Table 3-95 List of Macro Definitions in “r_mtr_rl78g24.h” [2/2]

Macro	Definition value	Description	Remarks
MTR_OFFSET_CALC_ST_WAIT_CNT	CP_OFFSET_CALC_ST_WAIT_CNT	Stable waiting time before acquiring current offset	
MTR_DEADTIME_RATIO	(MTR_DEADTIME * MTR_CARRIER_FREQ * (1 << MTR_Q_VOLTAGE)) /1000	Deadtime compensation coefficient	
MTR_DEADTIME_CURRENT_LIMIT	MP_RATED_CURRENT * 0.1f	Current limit value	
MTR_CTRL_PERIOD	(MTR_INT_DECIMATION + 1) / (MTR_CARRIER_FREQ * 1000)	Current control cycle	
MTR_SPEED_CTRL_PERIOD	CP_SPEED_CTRL_PERIOD	Speed control cycle	
MTR_ADC_DATA_SHIFT	0	A/D conversion value shift amount	
MTR_ADC_OFFSET	0x7FF	A/D conversion value offset	
ERROR_NONE	0x00	No error	
ERROR_CHANGE_CLK_TIMEOUT	0x01	Timeout error for clock settings	
ERROR_CHARGE_CAP_TIMEOUT	0x02	Capacitor charging timeout error	
MTR_CHARGE_CAP_WAIT_CNT	IP_CHARGE_CAP_WAIT_CNT	Charge time of DC capacitor	
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	
MTR_V_PHASE_LEAD_COEF	FIX_fromfloat(CP_V_PHASE_LEAD_COEF, TR_Q_V_PHASE_LEAD_COEF)	Voltage leading phase coefficient	
MTR_LOWEST_DUTY_CNT	MTR_CARRIER_CNT + MTR_DEADTIME_CNT	Timer count: lowest	
MTR_MAX_DUTY_CNT	MTR_CARRIER_CNT - 3	Timer count: max	
MTR_MIN_DUTY_CNT	MTR_DEADTIME_CNT + 1	Timer count: min	
CARRIER_UL_CNT	MTR_CARRIER_CNT - 2	Timer count	

Table 3-96 List of Macro Definitions in “Config_ADC.h” [Used in 1-shunt mode]

Macro	Definition value	Description	Remarks
R_MTR_ADCR_IDC_A	ADCR2	Select ADCR register of point A	
R_MTR_ADCR_IDC_B	ADCR0	Select ADCR register of point B	
R_MTR_ADCR_VDC	ADCR3	Select ADCR register of inverter bus voltage	
R_MTR_WaitIdc	macro function	Waiting for AD convert	

Table 3-97 List of Macro Definitions in “Config_ADC.h” [Used in 3-shunt mode]

Macro	Definition value	Description	Remarks
R_MTR_ADCR_IU	ADCR1	Select ADCR register of Iu	
R_MTR_ADCR_IV	ADCR0	Select ADCR register of Iv	
R_MTR_ADCR_IW	ADCR2	Select ADCR register of Iw	
R_MTR_ADCR_VDC	ADCR3	Select ADCR register of inverter bus voltage	
R_MTR_ADCR_IDC	ADCR3	Select ADCR register of DC link current	
R_MTR_ADS_CHANNEL_VDC	_06_AD_ADV_INPUT_ANI6	Select AD channel of inverter bus voltage	
R_MTR_ADS_CHANNEL_IDC	_12_AD_ADV_INPUT_ANI18	Select AD channel of DC link current	
R_MTR_ChangeChanelDefault	macro function	Changing AD channel of inverter bus voltage	
R_MTR_ChangeChanelIdc	macro function	Changing AD channel of DC link current	
R_MTR_WaitIdc	macro function	Waiting for AD convert	

Table 3-98 List of Macro Definitions in “Config_Port.h”

Macro	Definition value	Description	Remarks
R_MTR_PORT_LED1	P4_bit.no2	Output port of LED1	
R_MTR_PORT_LED2	P4_bit.no3	Output port of LED2	

Table 3-99 List of Macro Definitions in “Config_TRD0_TRD1.h”

Macro	Definition value	Description	Remarks
R_MTR_PORT_UP	P7_bit.no0	U-phase (positive phase) voltage output port	
R_MTR_PORT_UN	P7_bit.no1	U-phase (negative-phase) voltage output port	
R_MTR_PORT_VP	P7_bit.no2	V-phase (positive-phase) voltage output port	
R_MTR_PORT_VN	P7_bit.no3	V-phase (negative-phase) voltage output port	
R_MTR_PORT_WP	P7_bit.no4	W-phase (positive-phase) voltage output port	
R_MTR_PORT_WN	P7_bit.no5	W-phase (negative-phase) voltage output port	
R_MTR_StartADCTrigger	macro function	Setting AD start trigger	
R_MTR_SetPwmDuty	macro function	Setting PWM duty in 1 shunt mode	
R_MTR_SetPWMDutySym	macro function	Setting PWM duty in 3 shunt mode	

Table 3-100 List of Macro Definitions in “r_mtr_common.h”

Macro	Definition value	Description	Remarks
MTR_TWOPI	2*3.14159265359f	2π	
MTR_SQRT_2	1.4142136f	√2	
MTR_SQRT_3	1.7320508f	√3	
MTR_SQRT_3_half	1.7320508f / 2	√3 ÷ 2	
MTR_CW	1	CW	
MTR_CCW	-1	CCW	
MTR_ON	0	ON	
MTR_OFF	1	OFF	
MTR_CLR	0	Flag clear	
MTR_SET	1	Flag set	
MTR_OPL	0	Open loop	
MTR_CLL	1	Closed loop	
MTR_UNREACHED	0	Unreached	
MTR_REACHED	1	Reached	
MTR_WAIT_CNT_5US	30	Wait for 5us	
MTR_WAIT_CNT_20US	120	Wait for 20us	
MTR_WAIT_CNT_300US	1500	Wait for 300us	
MTR_ADC_12BIT	12	12bit ADC	
MTR_ADC_10BIT	10	10bit ADC	

Table 3-101 List of Macro Definitions in “r_mtr_parameter.h”

Macro	Definition value	Description	Remarks
MTR_PWM_DUTY_RANGE	4095	Duty range	
MTR_INPUT_V	IP_INPUT_V	Input voltage	
MTR_HALF_VDC	MTR_INPUT_V * 0.5f	50% of voltage	
MTR_MCU_ON_V	MTR_INPUT_V * 0.8f	80% of voltage	
MTR_OVERVOLTAGE_LIMIT	IP_OVERVOLTAGE_LIMIT	Overvoltage limit value	
MTR_UNDERVOLTAGE_LIMIT	IP_UNDERVOLTAGE_LIMIT	Undervoltage limit value	
MTR_ANGLE_RANGE	(int16_t)(MTR_TWOPI* PU_SF_ANGLE * (1<<MTR_Q_ANGLE))	Angle range 2π	
MTR_ANGLE_HALF_RANGE	(int16_t)(MTR_ANGLE_RANGE/2))	Angle range π	
MTR_ANGLE_QUAT_RANGE	(int16_t)(MTR_ANGLE_RANGE/4))	Angle range π/2	
MTR_OVERCURRENT_LIMIT	CP_OC_LIMIT	Current limit value	
MTR_I_LIMIT_VD	IP_INPUT_V * 0.5f	Vd current PI limit	
MTR_I_LIMIT_VQ	IP_INPUT_V * 0.5f	Vq current PI limit	
MTR_RPM_RAD	(MP_POLE_PAIRS * MTR_TWOPI) / 60.0f	Conversion from [rpm] to [rad/s]	
MTR_SPEED_LIMIT_RAD	CP_SPEED_LIMIT_RPM * MTR_RPM_RAD	Speed limit value [rad/s]	
MTR_MAX_SPEED_RAD	CP_MAX_SPEED_RPM * MTR_RPM_RAD	Maximum speed [rad/s]	
MTR_LIMIT_IQ	MP_ RATED_CURRENT * MTR_SQRT_3	Speed PI output limit value	
MTR_I_LIMIT_IQ	MP_ RATED_CURRENT * MTR_SQRT_3	Limit value for speed PI integral term output	
MTR_CL2OL_SPEED_RAD	CP_LESS2OL_SPEED_RPM * MTR_RPM_RAD	Switching speed from sensorless to open loop [rad/s]	
MTR_OL2LESS_SPEED_RAD	CP_OL2LESS_SPEED_RPM * MTR_RPM_RAD	Switching speed from open loop to sensorless [rad/s]	
MTR_RECIM	(1.0f/(MP_BEMF_CONSTANT * PU_SF_BEMF_CONST))	1/ BEMF constant [PU]	
MTR_DRAW_IN_WAIT_CNT	CP_DRAW_IN_WAIT_TIME/MTR_SPEED_CTRL_PERIOD	Draw-in wait time count value	
MTR_SWITCH_COUNT	(uint16_t)(CP_OL2CL_SWITCH_TIME/MTR_SPEED_CTRL_PERIOD)	Time[cnt] to switch open loop to closed loop	
MTR_CL2OL_JUDGE_WAIT_CNT	(uint16_t)(CP_CL2OL_JUDGE_WAIT_TIME / (CP_SPEED_CTRL_PERIOD * 1000))	Wait counts for open loop judgement	

Table 3-102 List of Macro Definitions in “r_mtr_statemachine.h”

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

Table 3-103 List of Macro Definitions in “r_mtr_foc_less_speed.h” [1/2]

Macro	Definition value	Description	Remarks
MTR_CONTROL_CURRENT	0x01	Current control	
MTR_CONTROL_SPEED	0x02	Speed control	
MTR_CONTROL_POSITION	0x04	Position control	
MTR_CONTROL_TORQUE	0x08	Torque control	
MTR_CONTROL_VOLTAGE	0x10	Voltage control	
MTR_ERROR_NONE	0x0000	No error	
MTR_ERROR_OVER_CURRENT	0x0001	Overcurrent error	
MTR_ERROR_OVER_VOLTAGE	0x0002	Overvoltage error	
MTR_ERROR_OVER_SPEED	0x0004	Excessive speed error	
MTR_ERROR_HALL_TIMEOUT	0x0008	Timeout error of Hall sensor detection Mismatch error of Hall sensor signal	
MTR_ERROR_BEMF_TIMEOUT	0x0010	BEMF timeout error	
MTR_ERROR_HALL_PATTERN	0x0020	Hall pattern error	
MTR_ERROR_BEMF_PATTERN	0x0040	BEMF pattern error	
MTR_ERROR_UNDER_VOLTAGE	0x0080	Undervoltage error	
MTR_ERROR_OVERCURRENT_SW	0x0100	Overcurrent error for SW	
MTR_ERROR_IPD_TRX_OVERFLOW	0x0200	TRX overflow error during initial position detection	
MTR_ERROR_UNKNOWN	0xffff	Undefined error	
MTR_ID_ZERO_CONST	0	d-axis current 0 control	
MTR_ID_MANUAL	1	d-axis current manual control	
MTR_ID_MTPA2FW	2	d-axis current MTPA and Field weakening control	
MTR_IQ_ZERO_CONST	0	q-axis current 0 control	
MTR_IQ_MANUAL	1	q-axis current manual control	
MTR_IQ_OL2CL	2	q-axis current for Open loop to Closed loop switch Control	
MTR_IQ_SPEED_PI_OUTPUT	3	Speed PI control output	
MTR_SPEED_ZERO_CONST	0	Speed 0 control	
MTR_SPEED_MANUAL	1	Speed manual control	
MTR_OFFSET_CALC_EXE	0	Offset is being removed	
MTE_OFFSET_CALC_END	1	Offset removal completed	
MTR_IPD_EXE	2	IPD processing execution	
MTR_IPD_END	3	IPD processing completed	
MTR_DRIVE_START	4	Start driving	
MTR_DRIVE_ID_ZERO	5	d-axis current 0 drive	
MTR_DRIVE_BRAKE	6	Brake processing	
MTR_DRIVE_END	7	Drive stop	
MTR_PHASE_U	0	U-phase	
MTR_PHASE_V	1	V-phase	
MTR_PHASE_W	2	W-phase	

Table 3-104 List of Macro Definitions in “r_mtr_foc_less_speed.h” [2/2]

Macro	Definition value	Description	Remarks
MTR_DRV_UVW	1	Duty size relationship U>V>W	[Used in 1-shunt mode]
MTR_DRV_UWV	2	Duty size relationship U>W>V	
MTR_DRV_VUW	3	Duty size relationship V>U>W	
MTR_DRV_VWU	4	Duty size relationship V>W>U	
MTR_DRV_WUV	5	Duty size relationship W>U>V	
MTR_DRV_WVU	6	Duty size relationship W>V>U	
MTR_DRV_MASK	0x0F	Mask of modulation drive pattern	
MTR_DRV_USE_REPRO	0x10	Flag of reproduction current	
MTR_CROSS_MID_MIN	1	Duty cross for middle and minimum	
MTR_CROSS_MAX_MID	2	Duty cross for maximum and middle	

Table 3-105 List of Macro Definitions in “r_mtr_ipd.h”

Macro	Definition value	Description	Remarks
MTR_REF_CURRENT_BASE	(float)IP_DC_AMPLIFICATION_GAIN * IP_DC_SHUNT_RESISTANCE * 1024 / 5)	Scaling factor for setting threshold current of CMP0	【IPD】
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	30	Percentage of TRX differential threshold of angle detection for non-salient rotor	
MTR_PERCENTAGE	100	Calculation for percentage	
MTR_ENERGIZE_2_PHASES	0	2-phases energized	
MTR_ENERGIZE_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	
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_TRX_CNT	TRX	TRX count register	

3.6 Interrupt Processing Specifications

This section describes interrupt processing for the sample code.
Interrupt processing in both 1-shunt mode and 3-shunt mode are composed of two cycle interrupts: a carrier cycle interrupt (100-us : 1-time decimations of 50-us) and a 1-ms cycle interrupt.

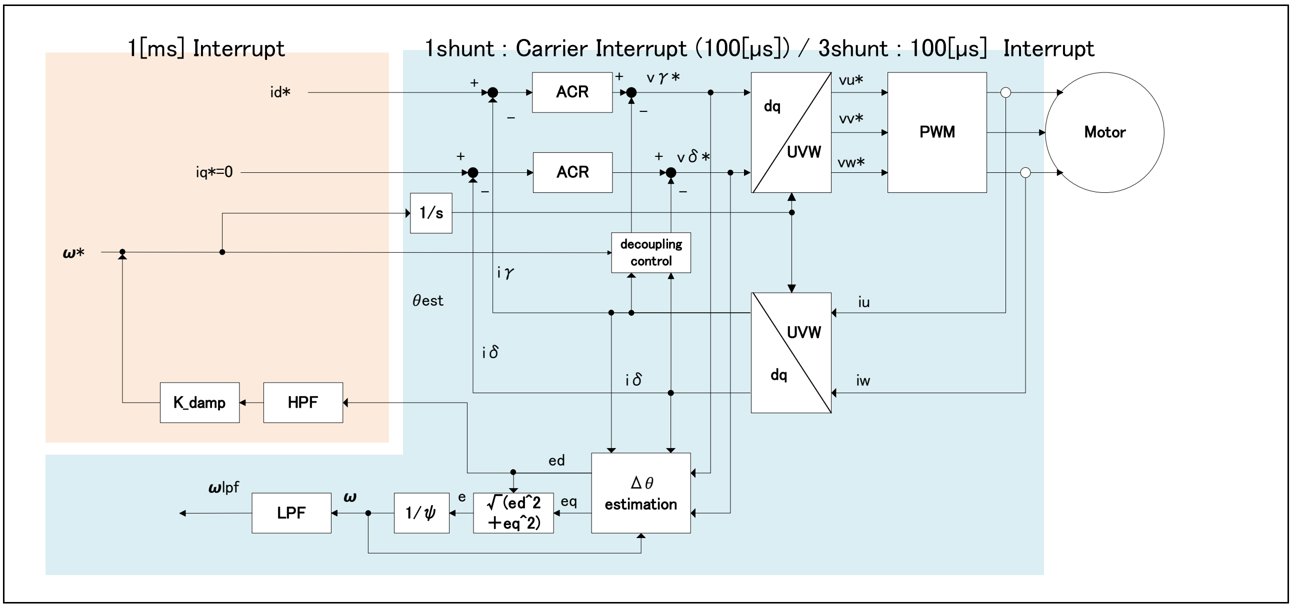


Figure 3-22 Interrupt processing inside control block (during open loop control)

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

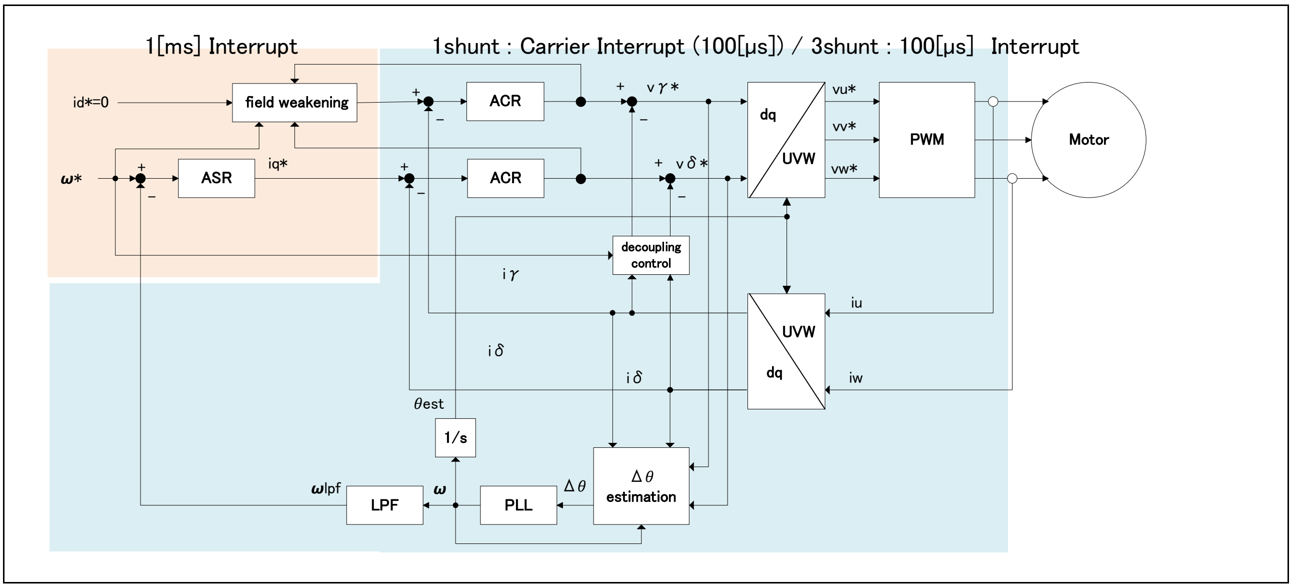


Figure 3-23 Interrupt processing inside control block (during closed loop control)

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

To keep the interrupt occupancy rate about 60%, the carrier cycle interrupt sets the 1-time decimations. The control cycle of the current control system inside the carrier interrupt has one 50-μs skip and thus is 100 μs.

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

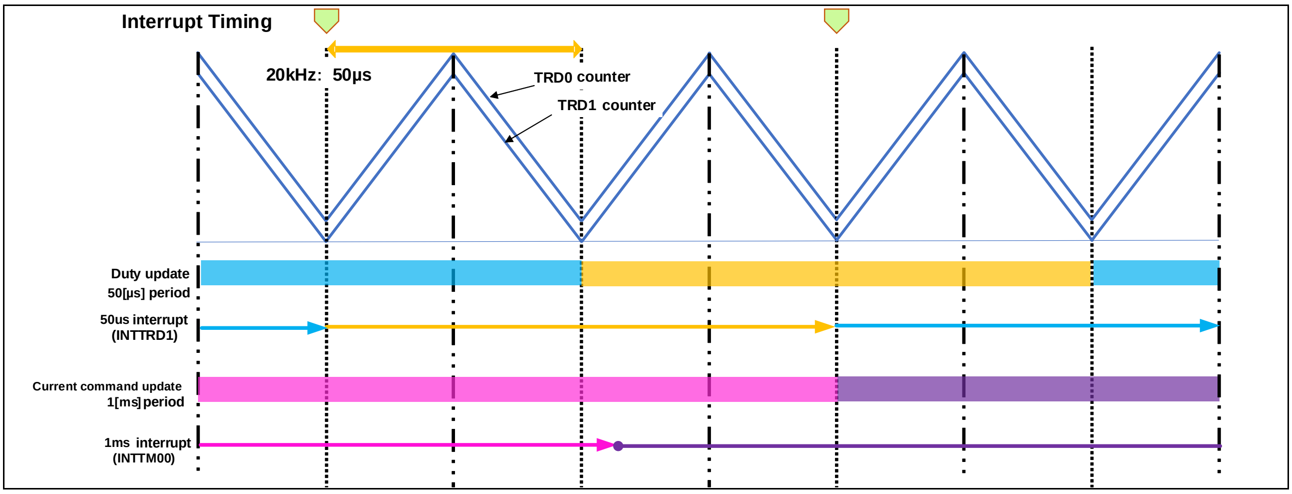


Figure 3-24 Image of interrupt generation and update of output value

3.7 Control flows (flowcharts)

3.7.1 Main process

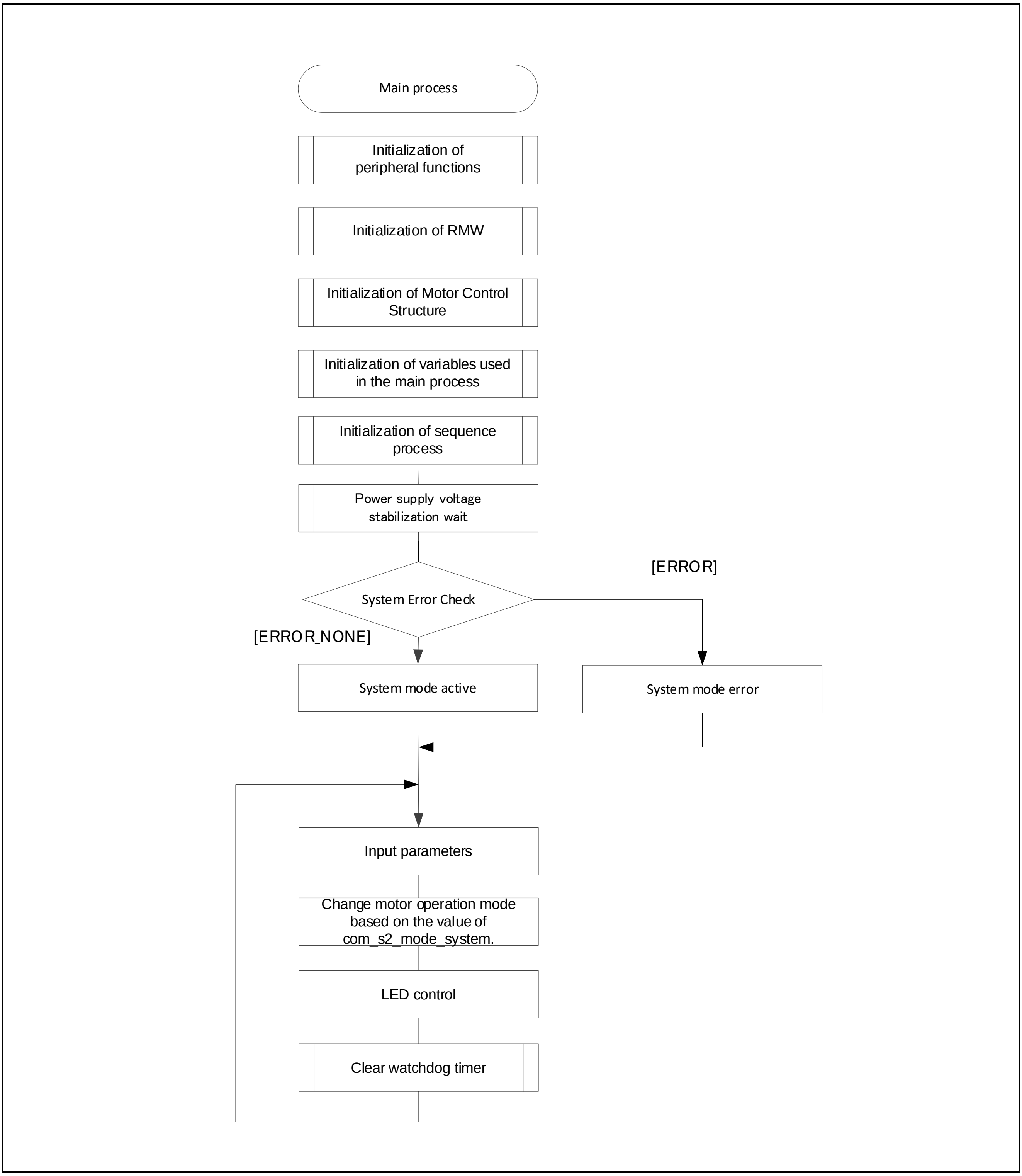


Figure 3-25 Main Process Flowchart

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

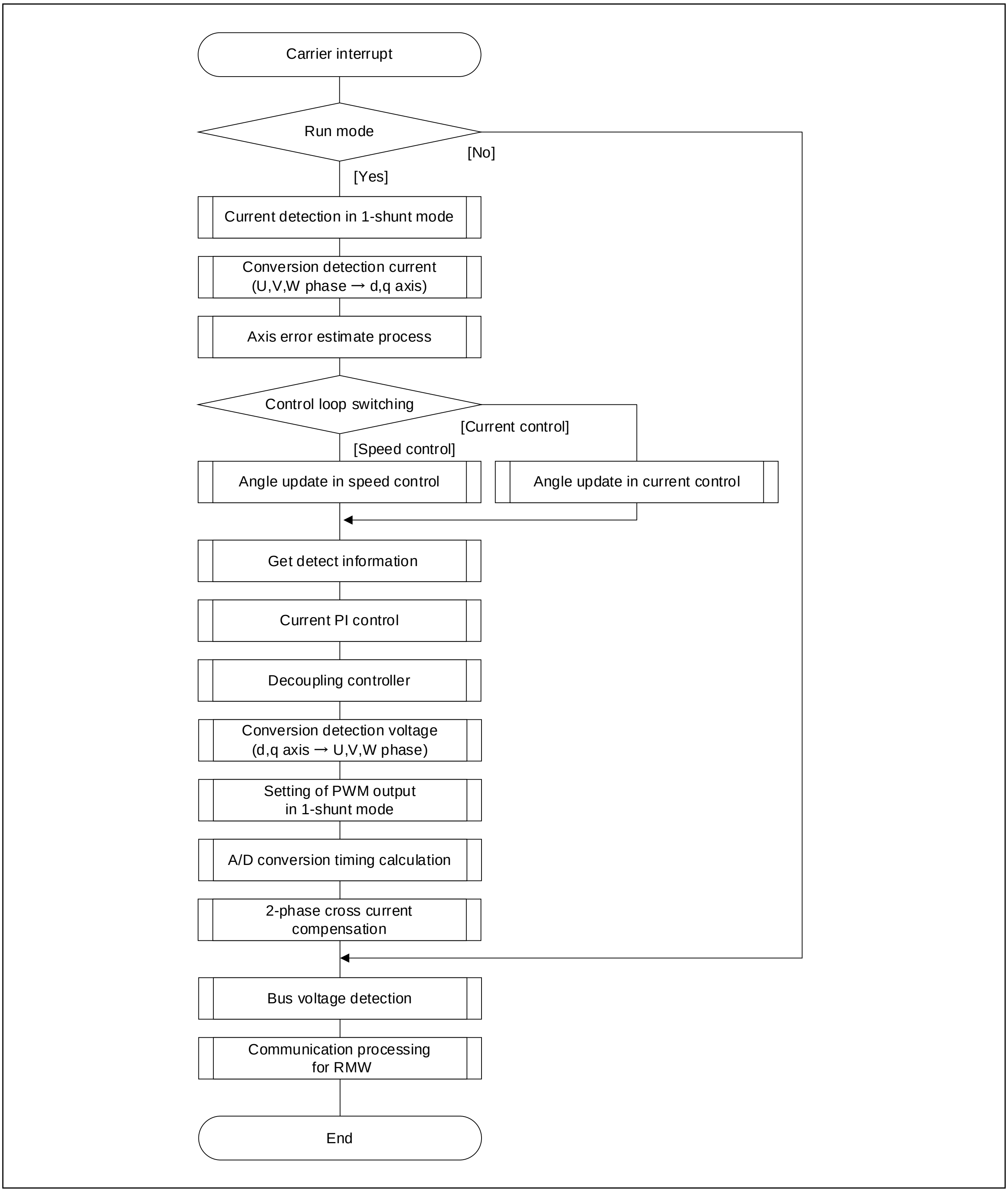


Figure 3-26 Carrier Cycle Interrupt Handling Flowchart

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

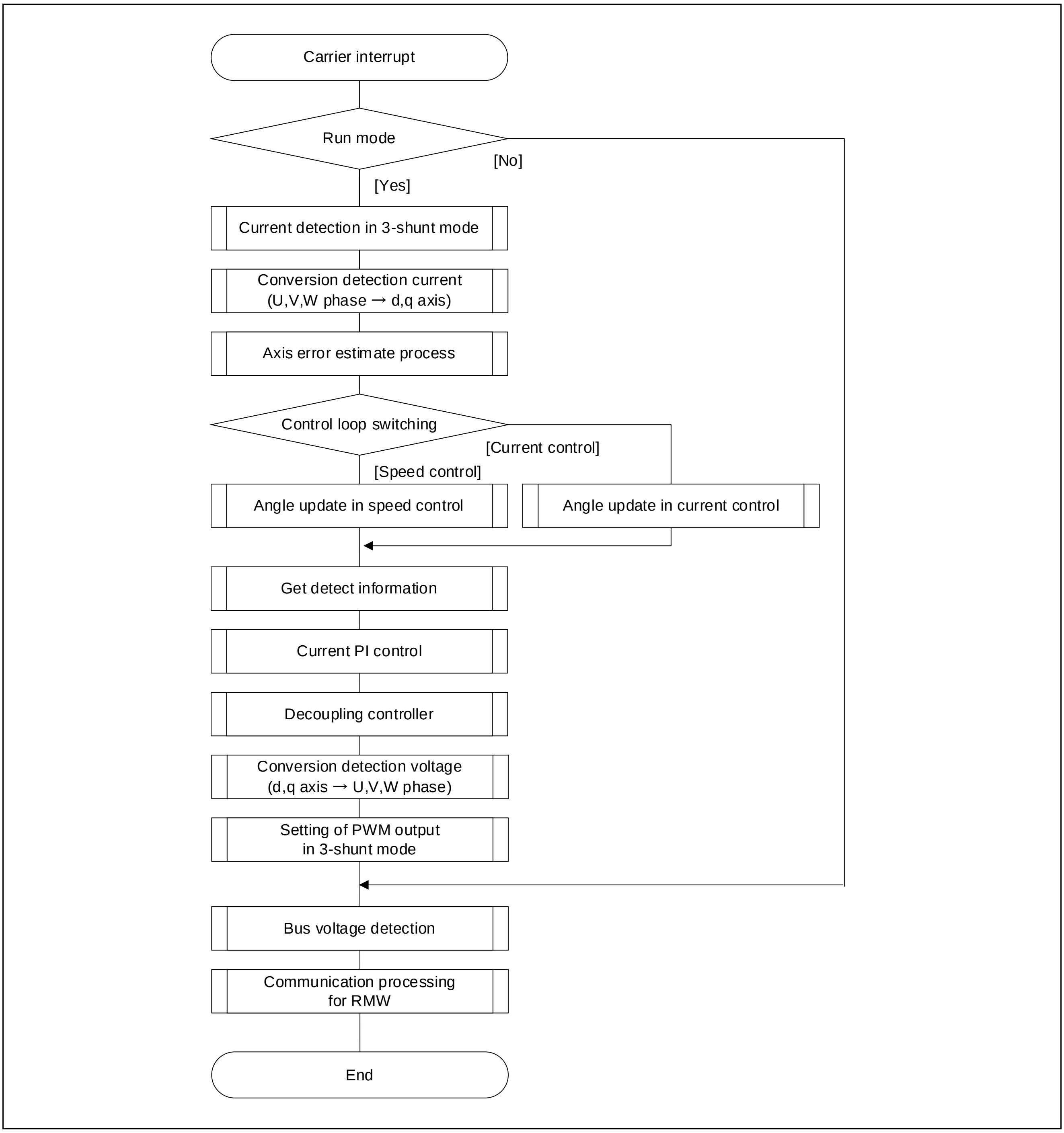


Figure 3-27 Carrier Cycle Interrupt Handling Flowchart

3.7.4 1-ms interrupt handling

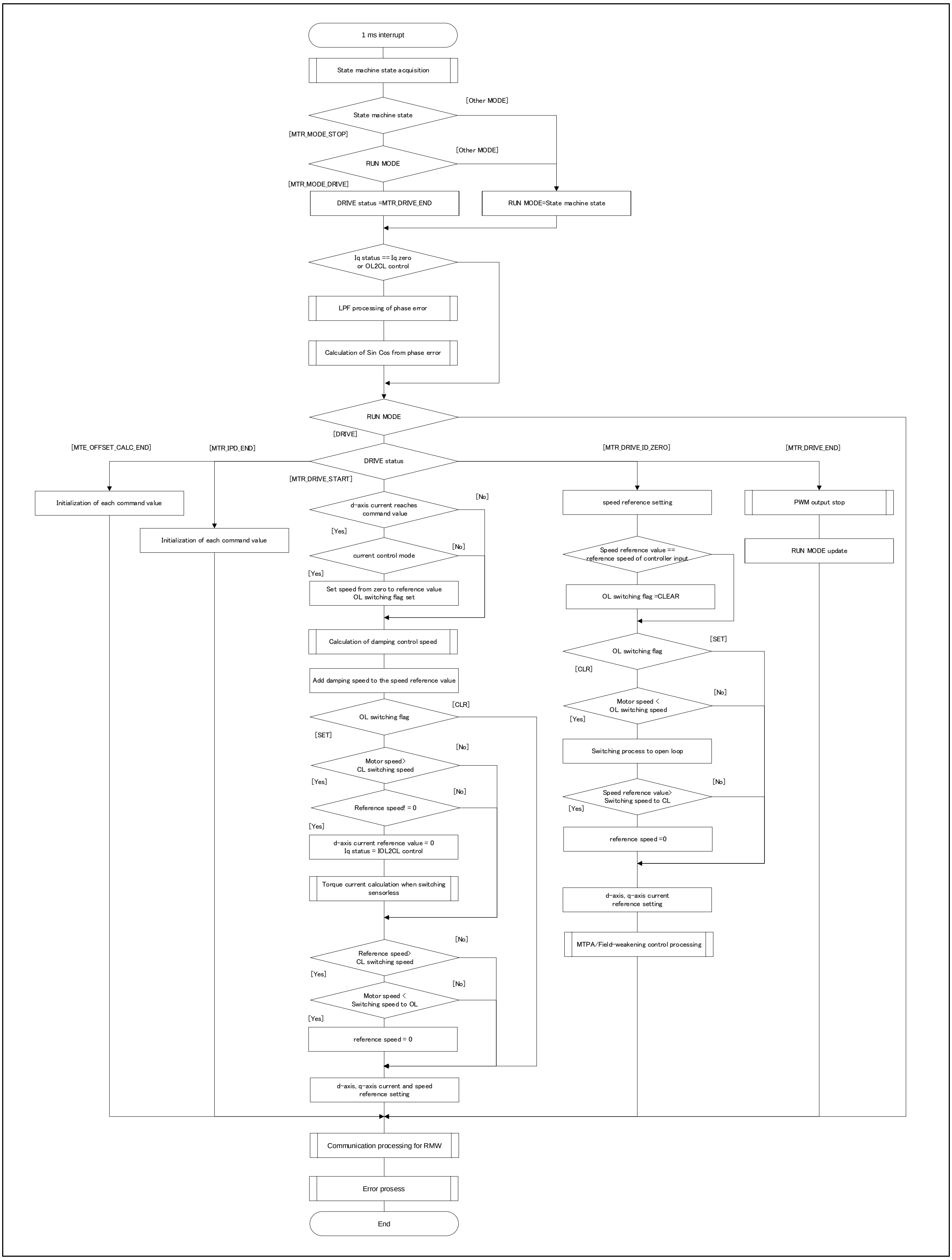


Figure 3-28 1-ms Interrupt Handling Flowchart

3.7.5 Comparator 3 interrupt handling

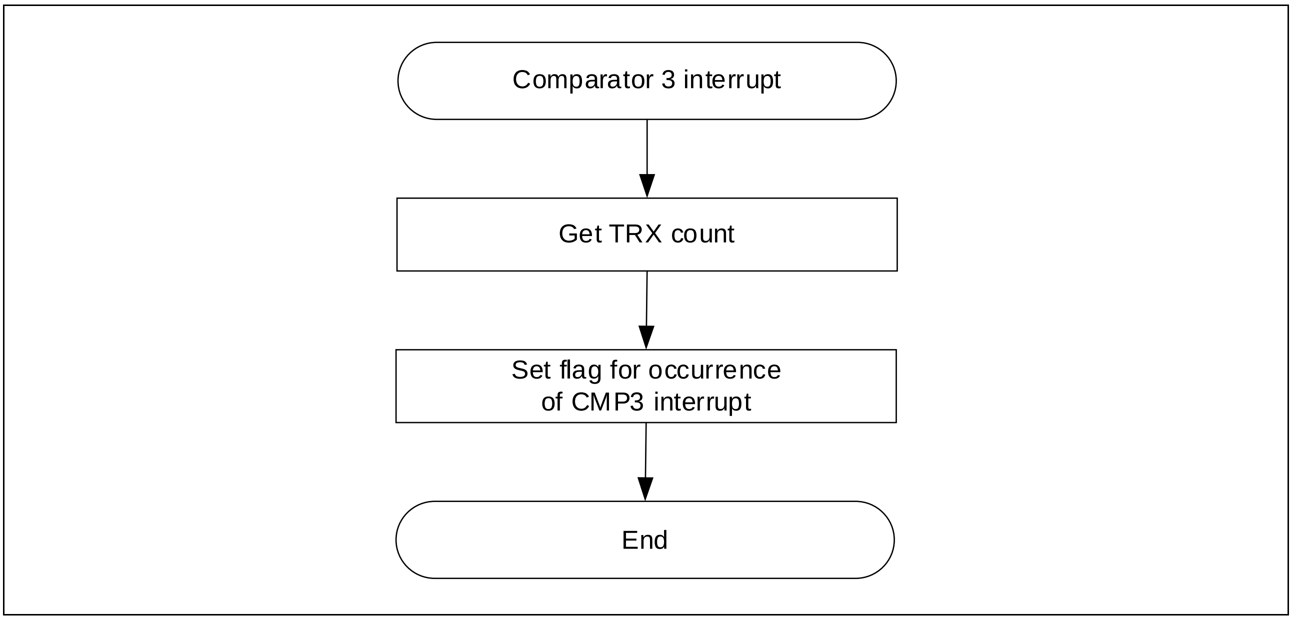


Figure 3-29 Comparator 3 interrupt handling

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.

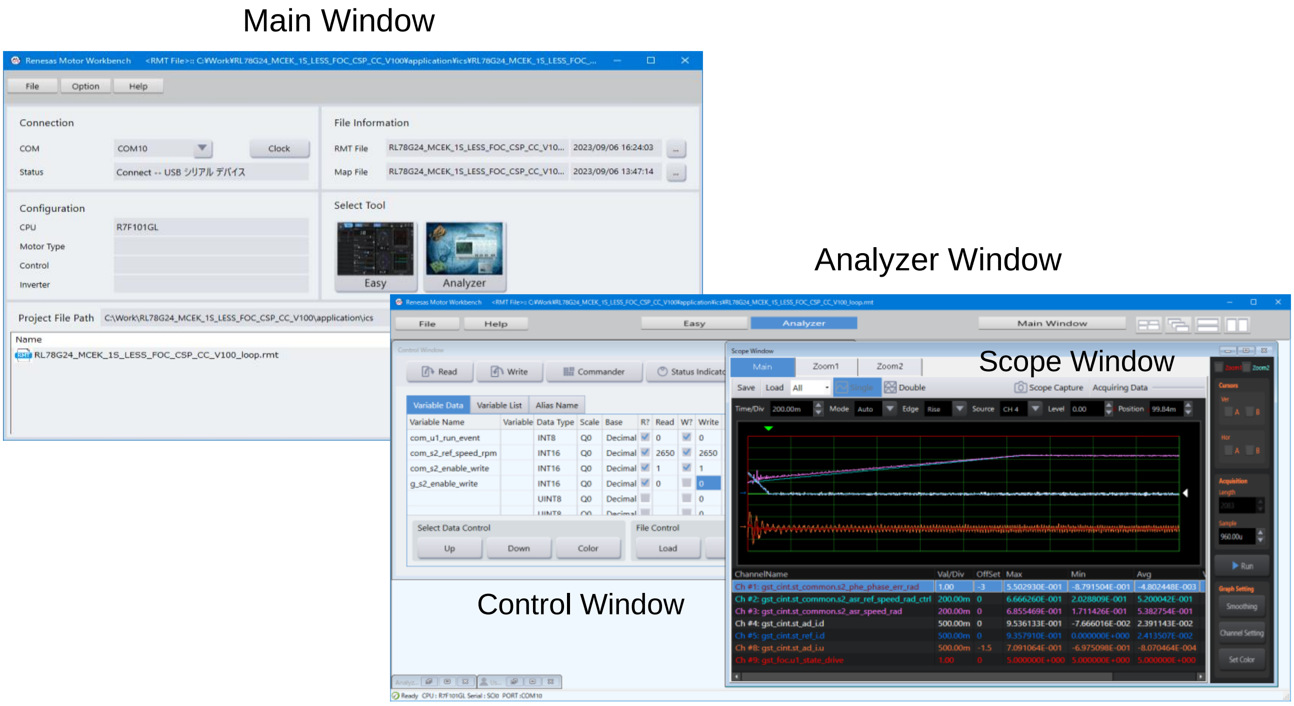


Figure 4-1 Screenshots of Renesas Motor Workbench

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

(1) Start Renesas Motor Workbench by clicking this icon 

.

(2) From the menu bar in the main window, select [File] -> [Open RMT File(O)].
Select RMT file in '[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 is a list of variables for the Analyzer. These variable values are reflected to the protect variables when the same values as g_s2_enable_write are written to com_s2_enable_write. However, note that variables with (*) do not depend on com_s2_enable_write.

Table 4-1 List of Input Variables for Analyzer (1/3)

Variable	Type	Content	Remarks ([ ]: reflection variable name)
com_u1_run_event (*)	uint8_t	Change run mode 0: Stop event 1: Drive event 2: Error event 3: Reset event	[g_u1_run_event]
com_u1_direction	uint8_t	Direction of rotation 0: CW 1: CCW	[gst_foc.st_asr.s1_ref_dir]
com_f4_mtr_r	float	Resistance [Ω]	[gst_cint.st_motor.s2_mtr_r]
com_f4_mtr_ld	float	d-axis inductance [H]	[gst_cint.st_motor.s2_mtr_ld]
com_f4_mtr_lq	float	q-axis inductance [H]	[gst_cint.st_motor.s2_mtr_lq]
com_f4_mtr_m	float	BEMF constant [Vs/rad]	[gst_cint.st_motor.s2_mtr_m]
com_f4_mtr_j	float	Inertia [kgm^2/(rad･s^2)]	[gst_cint.st_motor.s2_mtr_j]
com_f4_mtr_d0	float	Static friction coefficient[kgm^2/(rads^2)]	[gst_foc.st_asr.s2_d0_div_pm]
com_f4_mtr_d1	float	Kinetic friction coefficient[kgm^2/(rads^2)]	[gst_foc.st_asr.s2_d1_div_p2m]
com_u2_mtr_pp	uint16_t	Number of pole pairs	[gst_cint.st_motor.u2_mtr_pp]
com_u2_offset_calc_cnt	uint16_t	Current offset detection time	[gst_foc.st_sscs.u2_offset_calc_cnt]
com_s2_ref_speed_rpm	int16_t	Command rotational speed [rpm]	[gst_foc.st_asr.s2_ref_speed_rad]
com_f4_ramp_limit_speed_rpm	float	Limit of acceleration [rpm/ms]	[gst_foc.st_asr.s2_ramp_limit_speed_rad]
com_s2_max_speed_rpm	int16_t	Maximum speed [rpm]	[gst_foc.st_asr.s2_max_speed_rad]
com_f4_acr_nf_hz	float	Current PI control natural frequency [Hz]	[gst_cint.st_acr.st_pi_id.s2_kp] [gst_cint.st_acr.st_pi_id.s2_kidt] [gst_cint.st_acr.st_pi_iq.s2_kp] [gst_cint.st_acr.st_pi_iq.s2_kidt]
com_f4_asr_nf_hz	float	Speed PI control natural frequency [Hz]	[gst_cint.st_asr.st_pi.s2_kp] [gst_cint.st_asr.st_pi.s2_kidt]
com_f4_asr_lpf_cof_hz	float	ASR LPF natural frequency [Hz]	[gst_cint.st_asr.st_lpf.s2_in_k] [gst_cint.st_asr.st_lpf.s2_out_k]
com_f4_acr_lpf_cof_hz	float	ACR LPF natural frequency [Hz]	[gst_cint.st_acr.st_iq_lpf.s2_in_k] [gst_cint.st_acr.st_iq_lpf.s2_out_k] [gst_cint.st_acr.st_id_lpf.s2_in_k] [gst_cint.st_acr.st_id_lpf.s2_out_k]
com_f4_pll_nf_hz	float	PLL natural frequency [Hz]	[gst_cint.st_pll.st_pi.s2_kp] [gst_cint.st_pll.st_pi.s2_kidt]
com_f4_acr_deadband_lsb	float	Deadband of current PI calculation [LSB]	[gst_cint.st_acr.st_pi_id.s2_deadband] [gst_cint.st_acr.st_pi_iq.s2_deadband]
com_f4_asr_deadband_lsb	float	Deadband of speed PI calculation [LSB]	[gst_foc.st_asr.st_pi.s2_deadband]
com_f4_pll_deadband_lsb	float	Deadband of phase locked loop calculation [LSB]	[gst_foc.st_asr.st_pi.s2_deadband]
com_f4_asr_ki_aug	float	Augmentation rate for integral part of ASR	[gst_foc.st_asr.st_pi.s2_kidt]
com_s2_cl2ol_speed_rpm	int16_t	Switching speed from sensorless to open loop [rpm]	[gst_foc.st_asr.s2_cl2ol_speed_rad]
com_s2_ol2cl_speed_rpm	int16_t	Switching speed from open loop to sensorless [rpm]	[gst_foc.st_asr.s2_ol2cl_speed_rad]
com_f4_ol_ref_id	float	Open loop d-axis command current [A]	[gst_foc.st_acr.s2_ol_ref_id]
com_f4_draw_in_wait_time	float	Draw-in wait time count value [s]	[gst_foc.u2_draw_in_wait_cnt]
com_f4_init_asr_intg	float	ASR integral term initial value during sensorless transition	[gst_foc.st_asr.s2_init_intg]
com_f4_ramp_limit_current	float	Limit value for current rise [A/ms]	[gst_foc.st_acr.s2_ramp_limit_current]
com_s2_duty_diff_limit	int16_t	Minimum value of duty deviation between phases	[gst_cint.st_sscs.s2_duty_diff_limit]

Table 4-2 List of Input Variables for Analyzer (2/3)

variable	type	content	remarks ([ ]: reflection variable name)
com_f4_i_repro_cof_hz	float	LPF cut-off frequency for current reproduction [Hz]	[gst_cint.st_id_repro.s2_in_k] [gst_cint.st_id_repro.s2_out_k] [gst_cint.st_iq_repro.s2_in_k] [gst_cint.st_id_repro.s2_out_k]
com_s2_duty_diff_limit2	int16_t	Limit of difference 2 between each phase duty	[gst_cint.st_sscs.s2_duty_diff_limit2]
com_s2_mod_3ph2ph_speed_rpm	int16_t	Switching speed from 3-phase modulation to 2-phase modulation [rpm]	[gst_cint.st_sscs.s2_mod_3ph2ph_speed_rad]
com_s2_mod_2ph3ph_speed_rpm	int16_t	Switching speed from 2-phase modulation to 3-phase modulation [rpm]	[gst_cint.st_sscs.s2_mod_2ph3ph_speed_rad]
com_s2_ad_point_a_adj_cnt_3ph	int16_t	Adjustment value of A/D conversion timing A (3-phase modulation)	[gst_cint.st_sscs.s2_ad_point_a_adj_cnt_3ph]
com_s2_ad_point_b_adj_cnt_3ph	int16_t	Adjustment value of A/D conversion timing B (3-phase modulation)	[gst_cint.st_sscs.s2_ad_point_b_adj_cnt_3ph]
com_s2_ad_point_a_adj_cnt_2ph	int16_t	Adjustment value of A/D conversion timing A (2-phase modulation)	[gst_cint.st_sscs.s2_ad_point_a_adj_cnt_2ph]
com_s2_ad_point_b_adj_cnt_2ph	int16_t	Adjustment value of A/D conversion timing B (2-phase modulation)	[gst_cint.st_sscs.s2_ad_point_b_adj_cnt_2ph]
com_s2_ad_point_a_adj_cnt	int16_t	Adjustment for A/D delay counts for A point	[gst_cint.st_sscs.s2_ad_point_a_adj_cnt]
com_s2_ad_point_b_adj_cnt	int16_t	Adjustment for A/D delay counts for B point	[gst_cint.st_sscs.s2_ad_point_b_adj_cnt]
com_s2_mod_2ph_bot_change_cnt	int16_t	Number of counts during 2-phase modulation transition	[gst_cint.st_mod.s2_2ph_bot_change_cnt]
com_f4_sal_angle_current	float	TRX count value differential of angle detection for salient rotor	[gst_foc.st_ipd.u2_sal_angle_current]
com_u4_sal_angle_th	uint32_t	Maximum number of times salient rotor angle detection is measured	[gst_foc.st_ipd. u4_sal_angle_th]
com_u2_sal_angle_discharge	uint16_t	Discharge period of angle detection for salient rotor	[gst_foc.st_ipd. u2_sal_angle_discharge]
com_f4_sal_polarity_current	float	TRX count value differential of polarity detection for salient rotor	[gst_foc.st_ipd.u2_sal_polarity_current]
com_u4_sal_polarity_th	uint32_t	Maximum number of times salient rotor polarity detection is measured	[gst_foc.st_ipd. u4_sal_polarity_th]
com_u2_sal_polarity_discharge	uint16_t	Discharge period of polarity detection for salient rotor	[gst_foc.st_ipd. u2_sal_polarity_discharge]
com_f4_non_sal_current	float	TRX count value differential of angle detection for non-salient rotor	[gst_foc.st_ipd. U2_non_sal_current]
com_u4_non_sal_th	uint32_t	Maximum number of times non-salient rotor angle detection is measured	[gst_foc.st_ipd. u4_dnon_sal_th]
com_u2_non_sal_discharge	uint16_t	Discharge period of angle detection for non-salient rotor	[gst_foc.st_ipd. u2_non_sal_discharge]
com_f4_fw_pi_nf_hz	float	PI controller natural frequency for Field-Weakening Control [Hz]	[gst_foc.st_m2f.st_pi.s2_kp] [gst_foc.st_m2f.st_pi.s2_kidt]
com_s2_fw_speed_err_th_rpm	int16_t	Speed error threshold for Field-Weakening Control [rpm]	[gst_foc.st_m2f.s2_speed_err_th_rad]
com_f4_fw_speed_err_lpf_cof_hz	float	Speed error LPF cutoff frequency [Hz]	[gst_foc.st_m2f.s2_speed_err_th_rad] [gst_foc.st_m2f.st_lpf.s2_in_k] [gst_foc.st_m2f.st_lpf.s2_out_k]
com_s2_fw_id_inc_const	int16_t	Field-Weakening Id control constant	[gst_foc.st_m2f. s2_fw_id_inc_const]
com_u1_fw_pi_intg_redct_cof	uint8_t	Field-Weakening Id pi integral term reduction coefficient	[gst_foc.st_m2f.u1_fw_pi_intg_redct_cof]
com_f4_do_lpf_cof_hz	float	lpf cutoff frequency [Hz] for disturbance suppression	[gst_foc.st_do.st_lpf.s2_in_k] [gst_foc.st_do.st_lpf.s2_out_k]

Table 4-3 List of Input Variables for Analyzer (3/3)

variable	type	content	remarks ([ ]: reflection variable name)
com_f4_damp_hpf_cof_hz	float	HPF cutoff frequency for damping control [Hz]	[gst_foc.st_damp.st_hpf.s2_k]
com_f4_damp_zeta	float	Damping coefficient of damping control	[gst_foc.st_damp.s2_k]
com_f4_damp_speed_limit_rate	float	Damping control speed limit	[gst_foc.st_damp.s2_speed_limit_rate]
com_f4_pherr_lpf_cof_hz	float	Phase error LPF cutoff frequency for Open loop to Closed loop switch Control	[gst_foc.st_pe_lpf.s2_in_k] [gst_foc.st_pe_lpf.s2_out_k]
com_f4_ol2cl_switch_time	float	Time[s] to switch open loop to sensor-less	[gst_foc.st_ol2cl.u2_switch_cnt] [gst_foc.st_ol2cl.s2_theta2crnt_k1]
com_s2_enable_write	int16_t	Variable to allow to variable writing	[g_s2_enable_write]

4.3 Operation Example for Analyzer

An example of a motor driving operation using Analyzer is shown below. For the operation, the "Control Window" shown in Figure 4-1 is used. Refer to the ‘Renesas Motor Workbench V 3.1 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.

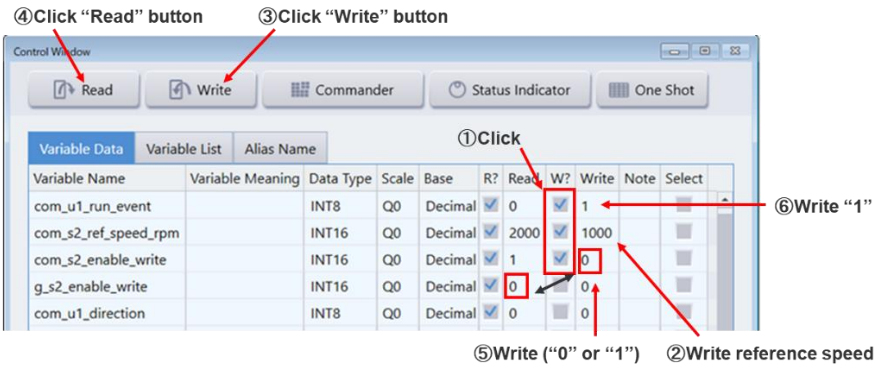


Figure 4-2 Procedure - Driving the motor

- Stop the motor

① Input a value of “0” in the [Write] box of “com_u1_run_event.”

② Click the “Write” button.

Figure 4-3 Procedure - Stop the motor

- Error cancel operation

① Input a value of “3” in the [Write] box of “com_u1_run_event.”

② Click the “Write” button.

Figure 4-4 Procedure - Error cancel operation

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Nov.27.2023	-	New create
1.10	Jul.16.2024	-	Update of revision information from the FAA version of the software.

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/.