RL78/G1F

Sensorless vector control for permanent magnetic synchronous motor with 1-shunt resistor current detection

Summary

This application note explains the sample programs for driving a permanent magnet synchronous motor in the sensorless vector method using the RL78/G1F 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.

Operation checking device

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

- RL78/G1F(R5F11BGEAFB)

Applicable sample programs

This application note regards the following sample programs.

- RL78G1F_GB01_1S_LESS_FOC_CSP_CC_V220 (IDE: CS+ for CC)
- RL78G1F_GB01_1S_LESS_FOC_E2S_CC_V220 (IDE: e^{2}studio)

Sample program for RL78/G1F sensorless vector control with 1-shunt resistor current detection
on inverter for evaluation GB01

References

- RL78/G1F Group User's Manual: Hardware (R01UH0516EJ0112)
- Renesas Motor Workbench 2.0 User's Manual (R21UZ0004JJ0202: Renesas-Motor-Workbench-V2_0f)
- RL78/G1F Motor Driver Board GB01 User's Manual (R12UT0012EJ0100)

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/G1F 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/G1F (R5F11BGEAFB)	RL78/G1F Motor Driver Board GB01 (P13130-D1-003)[1] Communication board (RTK0EMX6B0Z00000BJ) [1,2]	TSUKASA(Note 3) TG-55L

Table 1.2 Development Environment of the Sample Programs (Software)

CS+ version	Build tool version	Debug tool
V8.06.00	CC-RL V1.10.00	E2 Lite

e^{2}studio version	Build tool version	Debug tool
2021-10	CC-RL V1.10.00	E2 Lite

For purchasing information and technical support, please contact Renesas Electronics Corporation sales representatives and dealers.

Notes:

1. The RL78 / G1F demo board (P13130-D1-003) and communication board (RTK0EMX6B0Z00000BJ) are for testing purposes and are not for sale. For technical support, rental, etc., please contact our sales office or distributor.
2. As for the communication board used for debugging, ICS ++ (In Circuit Scope plus) manufactured by Desk Top Laboratories Inc. can also be used.

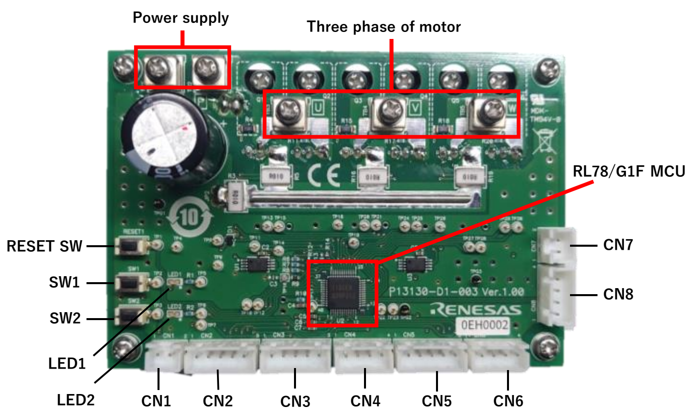
   Desk Top Laboratories Inc. (http://desktoplab.co.jp/)
3. TG-55L is a product of TSUKASA ELECTRIC.
   TSUKASA ELECTRIC. (https://www.tsukasa-d.co.jp/en/)

2. System overview

An overview of this system is provided below.

2.1 Hardware Specifications

Table 2.1 Specifications of 78/G1F Motor Driver Board GB01 (P13130-D1-003)

Item	Specification
Operating input voltage	12V - 50V
Maximum output current	30 A (peak current for each phase)
Motor to be driven	3-phase permanent magnet synchronous motor
Current detection method	3-phase current detection or current detection using DC link shunt resistor
DC bus voltage detection	Detection by voltage divider
Three-phase output voltage detection	Detection by voltage divider
PWM logic	Positive logic in both upper and lower arms
Overcurrent detection	Implemented by RL78/G1F MCU setting (PWMOPA); detection circuit not yet implemented
Dead time	1 µs or more
Switch	3 tact switches (one of which is a CPU reset)
LED	2
connectors	• Connector for emulator connection (using conversion cable): CN3 • I2C communication connector: CN6 • 2 serial communication connectors: CN4, CN8 • ABZ encoder signal input connector: CN2 • Hall sensor signal input connector: CN5 • 2 analog signal inputs: CN1, CN7
Exterior view	
Heat dissipation	Natural air cooling by heat sink
Names of parts	

2.2 Hardware configuration

The hardware configuration is shown below.

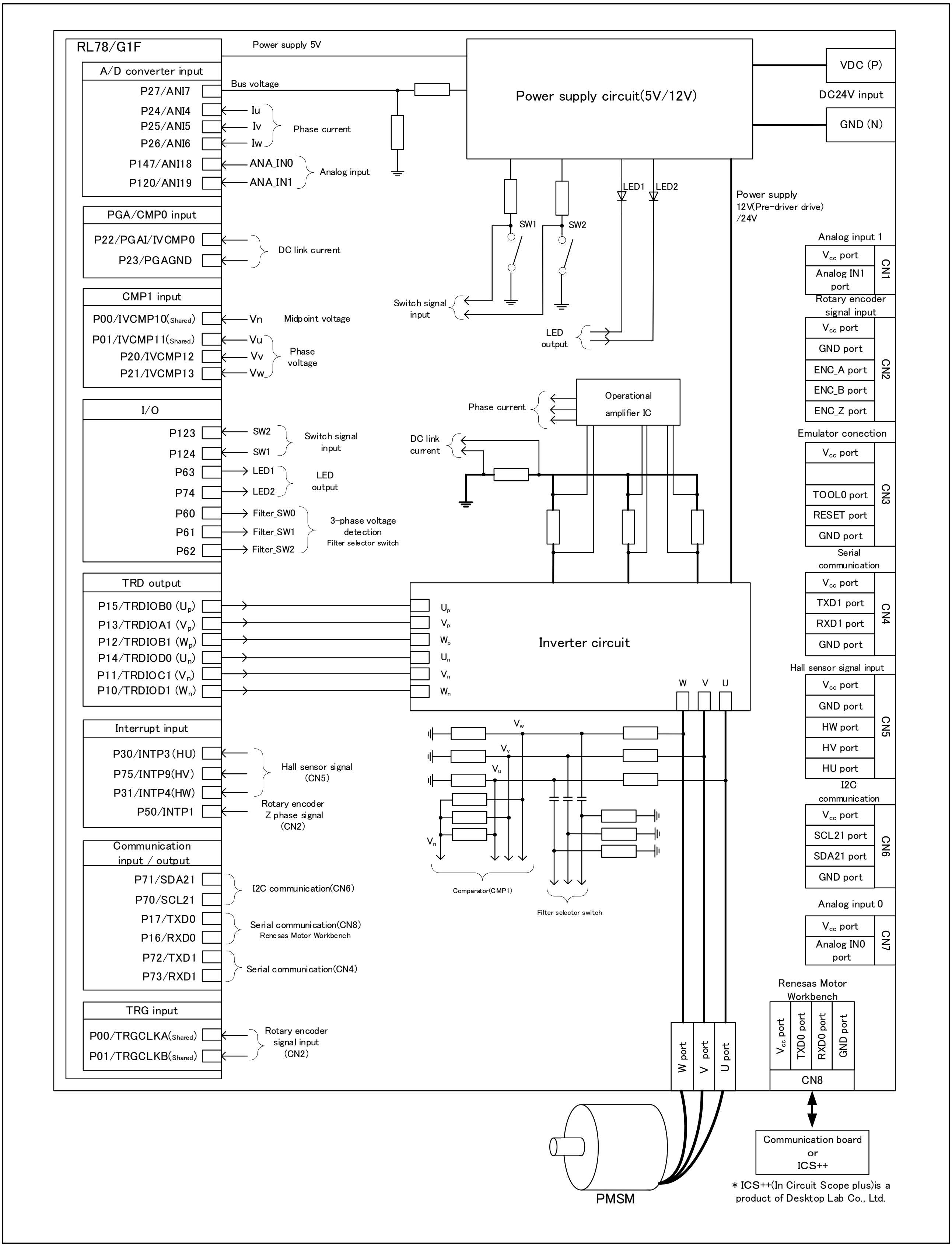


Figure 2-1 Hardware Configuration Diagram

2.3 Hardware specifications

2.3.1 User interface

Table 2.2 is a list of user interfaces of this system.

Table 2.2 – User Interfaces

<table>
<tr><th>Item</th><th>Interface component</th><th>Function</th></tr>
<tr><td>SW1</td><td>Tact switch</td><td>(not used in this system)</td></tr>
<tr><td>SW2</td><td>Tact switch</td><td>(not used in this system)</td></tr>
<tr><td>RESET</td><td>Tact switch (RESET1)</td><td>System reset</td></tr>
<tr><td>LED1</td><td>Yellow 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 off</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>Yellow green LED</td></tr>
</table>

The system's connector interfaces are listed in Table 2.3.

Table 2.3 – Connector Interfaces

Item	Number of ports	Function
CN1	2	Analog input (not used in this system)
CN2	5	ABZ encoder signal input (not used in this system)
CN3	5	Emulator connection (using conversion board)
CN4	4	Serial communication (SCI1) (not used in this system)
CN5	5	Hall sensor signal input (not used in this system)
CN6	4	I2C communication (not used in this system)
CN7	2	Analog input (not used in this system)
CN8	4	Serial communication (SCI0) communication with Renesas Motor Workbench

Table 2.4 is a list of port interfaces of the RL78/G1F microcontroller of this system.

Table 2.4 – Port Interfaces

R5F11BGEAFB port name	Function
P27 / ANI7	Inverter bus voltage detection
P24 / ANI4	U-phase current detection [Used in 3-shunt mode]
P25 / ANI5	V-phase current detection [Used in 3-shunt mode]
P26 / ANI6	W-phase current detection [Used in 3-shunt mode]
P147 / ANI18	Analog input (not used in this system)
P120 / ANI19	Analog input (not used in this system)
P22 / PGAI / IVCMP0	DC link current detection (A/D, PGA, CMP)
P23 / PGAGND	DC link current detection GND
P00 / IVCMP10	3-phase midpoint voltage detection (CMP1) (shared port)
P01 / IVCMP11	Uphase voltage detection (CMP1) (shared port) (not used in this system)
P20 / IVCMP12	Vphase voltage detection (CMP1) (not used in this system)
P21 / IVCMP13	Wphase voltage detection (CMP1) (not used in this system)
P123	Tact switch (SW2) (not used in this system)
P124	Tact switch (SW1) (not used in this system)
P63	LED1 ON / OFF control
P74	LED2 ON / OFF control
P60	U-phase voltage detection filter change switch (not used in this system)
P61	V-phase voltage detection filter change switch (not used in this system)
P62	W-phase voltage detection filter change switch (not used in this system)
P15 / TRDIOB0	PORT output / PWM output (U_p)
P13 / TRDIOA1	PORT output / PWM output (V_p)
P12 / TRDIOB1	PORT output / PWM output (W_p)
P14 / TRDIOD0	PORT output / PWM output (U_n)
P11 / TRDIOC1	PORT output / PWM output (V_n)
P10 / TRDIOD1	PORT output / PWM output (W_n)
P30 / INTP3	Hall sensor input (HU) (not used in this system)
P75 / INTP9	Hall sensor input (HV) (not used in this system)
P31 / INTP4	Hall sensor input (HW) (not used in this system)
P50 / INTP1	Encoder Z-phase input (not used in this system)
P71 / SDA21	I2C communication (not used in this system)
P70 / SCL21	I2C communication (not used in this system)
P17 / TXD0	Serial communication (SCI0)
P16 / RXD0	Serial communication (SCI0)
P72 / TXD1	Serial communication (SCI1) (not used in this system)
P73 / RXD1	Serial communication (SCI1) (not used in this system)
P00 / TRDCLKA	Encoder A-phase input (shared port) (not used in this system)
P01 / TRDCLKB	Encoder B-phase input (shared port) (not used in this system)
P40 / TOOL0	Data I/O for debugger
P125 / 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

2.3.2 Peripheral functions

Table 2.5 is a list of peripheral functions used in this system.

Table 2.5 List of Peripheral Functions

Peripheral Function	Usage
10-bit A/D converter (AD)	• Inverter bus voltage detection • U and W phase current detection (V phase current is calculated from U and W phase currents) [Used in 3-shunt mode] • DC link current detection [Used in 1-shunt mode]
Timer Array Unit (TAU)	• 1-ms interval timer • 100-us Control cycle timer [Used in 3-shunt mode] • Current detection AD trigger timer [Used in 1-shunt mode]
Timer RD (TRD)	PWM output for complementary PWM mode use
Timer RX(TRX)	Current rising period measurement
PWM option unit A (PWMOPA)	Forced cut-off of PWM output depending on CMP0 output
Programmable gain amplifier (PGA)	DC link current detection amplification
Comparator (CMP0)	• Overcurrent detection • Judgement for reaching threshold current at initial position detection
Data transfer controller (DTC)	Transfer of conversion result to memory by A/D conversion end interrupt

(1) 10-bit A/D converter (AD)

The U phase current (Iu), W phase current (Iw), and inverter bus voltage (Vdc) are measured using the 10-bit A/D converter. [Used in 3-shunt mode]

DC link current and inverter bus voltage (Vdc) are measured using the 10-bit A/D converter. [Used in 1-shunt mode]

A/D conversion sets the channel selection mode to Select mode and the conversion operation mode to One-shot Conversion mode. (Uses a software trigger).

(2) Timer Array Unit (TAU)

a. 1-ms interval timer

Uses channel 0 of the Timer Array Unit as the 1-ms interval timer.

b. Control cycle timer

Uses channel 1 of the Timer Array Unit.
It is used as a 100-μs interval timer.

(3) Timer RD (TRD)

Using complementary PWM mode, outputs six-phase PWM with deadtime.

Also, using a pulse output forced blocking function, the PWM output port outputs high impedance when overcurrent is detected.

(4)Timer RX (TRX)

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

(5) PWM option unit A (PWMOPA)

Force the PWM output to be cut off from the overcurrent signal detected in CMP0.

After detecting the cause of the cut-off release (CMP0 falling edge), the forced cut-off of the output is released from the software.

(6) Programmable gain amplifier (PGA)
Amplifies the voltage across the shunt resistor for 1-shunt resistor current detection.

(7) Comparator (CMP0)
Overcurrent detection or initial position detection by comparing to the internal reference value.

(8) Data transfer controller (DTC)
The A/D conversion result by 1-shunt resistor current detection is saved in the memory by the conversion end interrupt.

2.4 Software structure

2.4.1 Software file structure

The folder and file configurations of the sample programs are given in Table 2.6 below.

Table 2.6 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 hardwareUI Hardware UI processing
	ics	r_mtr_ics.h r_mtr_ics.c	Function definition for Analyzer(Note1) UI Analyzer(Note1) UI processing
		ICS_define.h	CPU definition for RMW
		RL78G1F_vector.c	Interrupt vector function processing for RMW
		Ics2_RL78G1F.h	Function declaration for RMW communication
		ics_RL78G1F_Lx.h	MCU serial communication definition for RMW communication
		ICS2_RL78G1F.obj	Library for RMW communication
driver	auto_generation	cstart.asm hdwinit.asm stkinit.asm iodefine.h	Auto generation files
		r_mtr_rl78g1f.h r_mtr_rl78g1f.c	Function definition for MCU control MCU settings
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 processing
		r_mtr_statemachine.h r_mtr_statemachine.c	Function definition for state machine State machine processing
		r_mtr_foc_less_speed.h r_mtr_foc_less_speed.c	Sensorless vector control-related function definition Sensorless vector control-related processing
		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 processing
		r_mtr_ipd.h r_mtr_ipd.c	Initial position detection function definition Initial position detection processing
		r_mtr_fw_ctrl.h r_mtr_fw_ctrl.obj	Field-Weakening Control function definition Field-Weakening Control processing
		r_mtr_damp_ctrl.h r_mtr_damp_ctrl.obj	Damping control function definition Damping control processing

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.

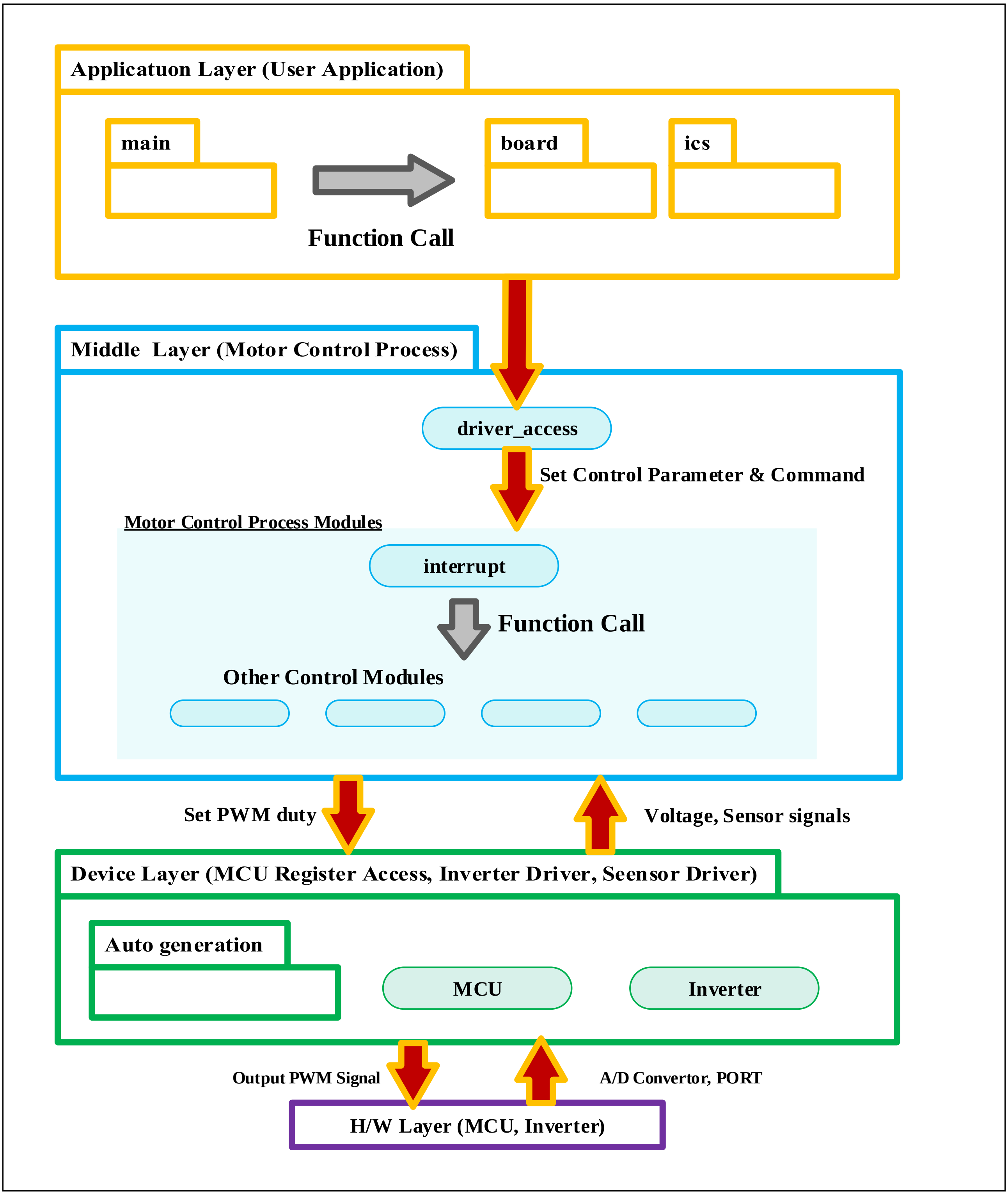


Figure 2-2 Module Configuration of the Sample Programs

2.5 Software specifications

The basic software specifications of the system are listed below.

Table 2.7 Software Specifications

Item	Content	
Control method	Vector control	
Current detection method	Use the compile switch to switch between the method of detecting DC link current (1-shunt mode) and the method of detecting 3-phase current (3-shunt mode). Switch 1-shunt mode or 3-shunt mode with the compile switch (Default is 1-shunt mode)	
Motor rotation start/stop	Operation using the motor control development support tool[Note]	
Position detection of rotor magnetic pole (Sensorless)	Angle estimation using induced voltage Initial position detection using motor polarity and magnetic saturation characteristics	
Input voltage	DC 24 V	
Main clock frequency	CPU clock: f_{CLK} 32 MHz TRD clock: f_{HOCO} 64 MHz	
Carrier frequency (PWM)	20 kHz	
Deadtime	1 μs	
Control cycle	Current control location/speed estimate: 100 μs (twice the carrier cycle) Speed control: 1 ms	
Rotational speed range	CW: 0 rpm - 3975 rpm CCW: 0 rpm - 3975 rpm However, driving is performed as an open loop at 1060 rpm or less	
Optimal setting	Default setting	
ROM/RAM Size	ROM	1-shunt mode : 20.545KB 3-shunt mode : 19.329KB
	RAM	1-shunt mode : 1.178KB 3-shunt mode : 1.110KB
Processing stop for protection	- Disables the motor control signal output (six outputs), under any of the following conditions. 1. Inverter bus voltage exceeds 28 V 2. Inverter bus voltage falls below 12 V 3. Rotational speed exceeds 5300 rpm 4. Each current phase exceeds 1.47 A - When an overcurrent detection signal (CMP0) is detected, the PWM output port is made high impedance (using PWMOPA).	

[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/G1F flash memory are shown below.

Table 2.8 User option byte settings

Setting	Address	Value	Description
787BF8	000C0H /010C0H	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: 136 ms - Watchdog timer counter operation control: In HALT/STOP mode, counter operation stops
	000C1H /010C1H	01110011B	- LVD settings (reset mode) Rising edge: 2.92 V Falling edge: 2.86 V
	000C2H /010C2H	11111000B	- Flash operation mode setting: HS (high-speed main) mode - High-speed on-chip oscillator/block frequency fHOCO: 64 MHz fIH: 32 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 – 03FFH	ANI7

(2) DC link shunt resistor current [Used in 1-shunt mode]
As shown in the table below, DC link shunt resistor current are 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	-6.25 A – 6.25 A: 0000H – 03FFH	PGAI：ANI25

(3) U phase and W phase current [Used in 3-shunt mode]
As shown in the table below, U phase and W phase current are measured and used for vector control.

Table 3.3 Conversion Ratios of U and W Phase Currents

Item	Conversion ratio (U phase and W phase currents: A/D conversion value)	Channel
U phase and W phase current	-5 A – 5 A: 0000H – 03FFH	Iu: ANI2 Iw: ANI3

[Note] For more information about A/D conversion characteristics, see “RL78/G1F User’s Manual - Hardware.”

3.1.3 PGA (Programmable gain amplifier)

(1) DC link shunt resistor current detection
The PGA is used to detect the DC link shunt resistor current in the inverter circuit.

Table 3.4 – DC link current detection

Item	Description	Remarks
PGA gain selection	4x	Choose from 4/8/16/32x
GND selection	PGAGND	Vss or PGAGND

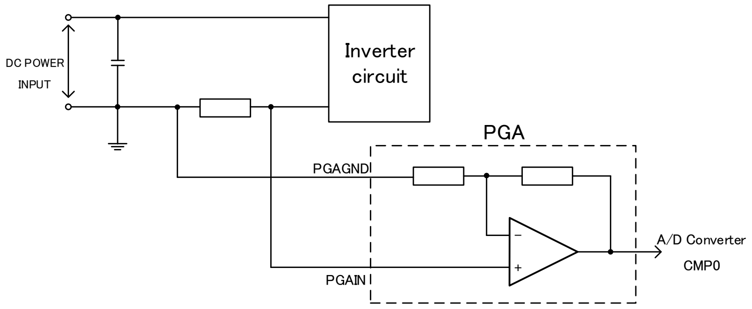


Figure 3-1 Connecting the PGA to the inverter circuit

3.1.4 Comparator

(1) Overcurrent detection (CMP0)

The output of the PGA 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.

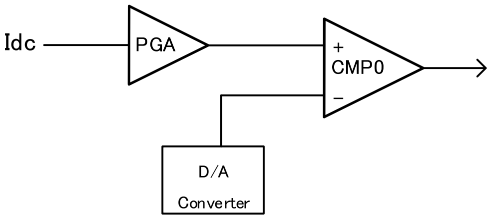


Figure 3-2 Overcurrent detection by CMP0

3.1.5 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 3-3.

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

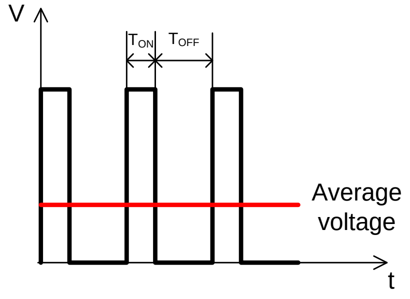


Figure 3-3 PWM Control

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

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

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

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

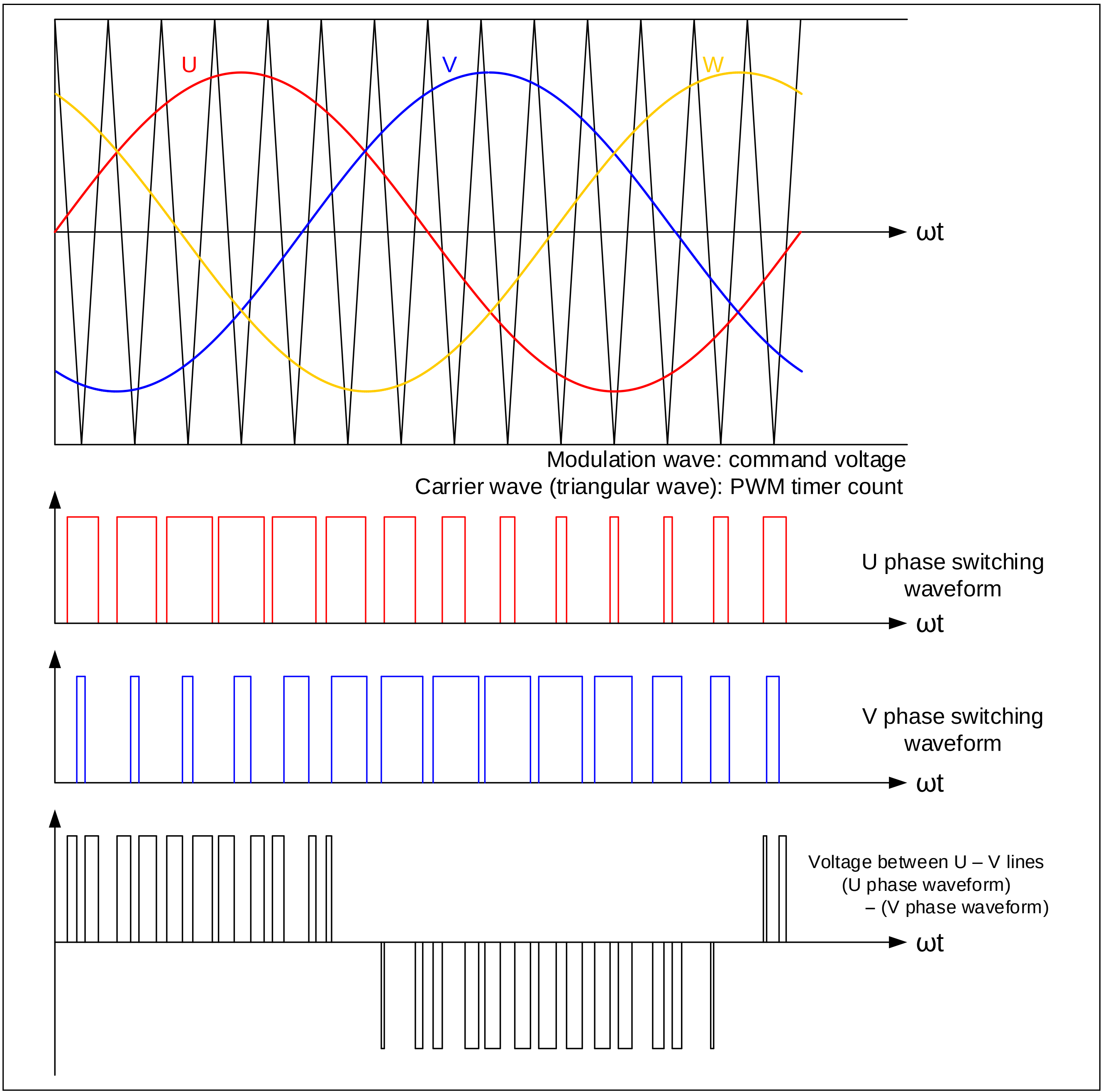


Figure 3-4 Conceptual Diagram of the Triangle Wave Comparison Method

(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 command voltage as shown below, the command 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 UVW phase command voltage

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

v_o : Resistance

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

3.1.7 State transitions

The state transition diagram for this program is shown in Figure 3-5.

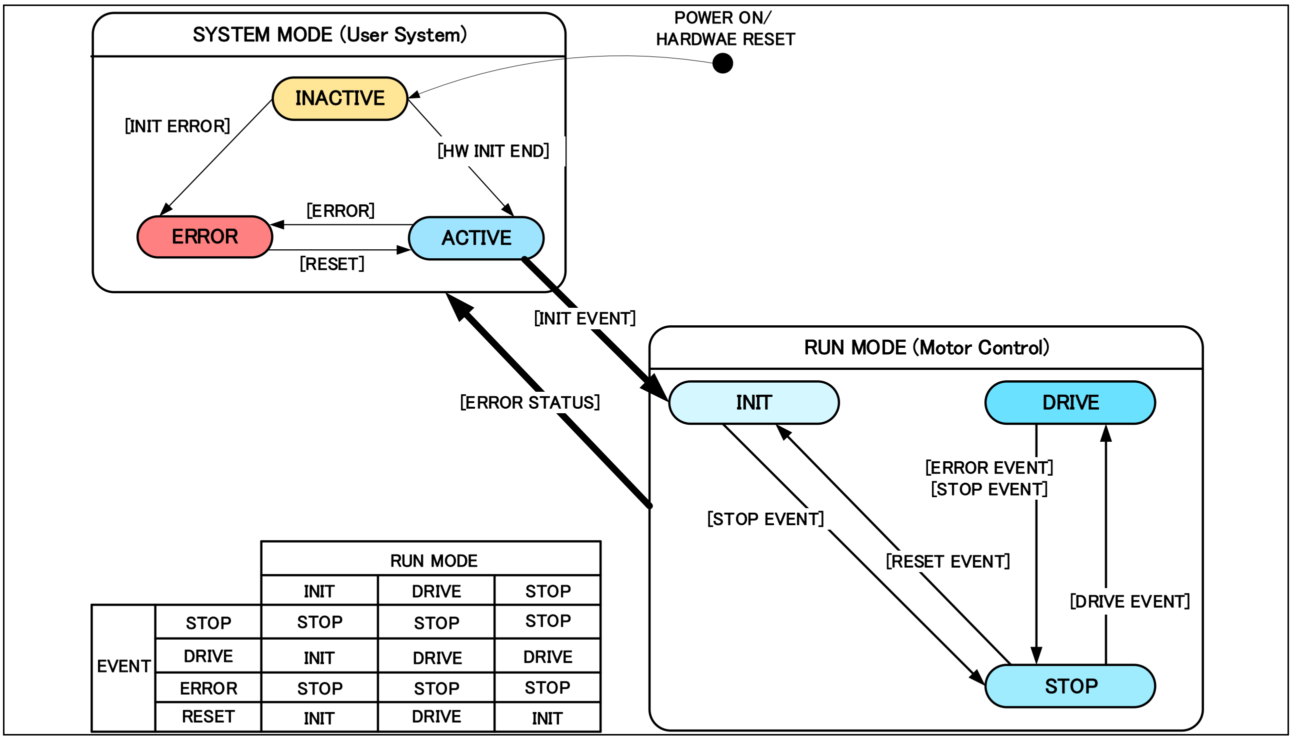


Figure 3-5 State Transition Diagram

(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 3-5. Each EVENT is caused by an occurrence as shown in Table 3.5.

Table 3.5 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 3.6 according to the drive status of the motor.

Table 3.6 DRIVE status List

status name	content
MTR_OFFSET_CALC_EXE	Execution of current offset detection processing
MTE_OFFSET_CALC_END	Completion of current offset detection processing
MTR_IPD_EXE	Execution of initial position detection processing
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

Startup method

The description of startup control of the sensorless vector control software is shown in Figure 3-6. The mode is controlled by the states that control the command values of the d-axis current, q-axis current, and speed.

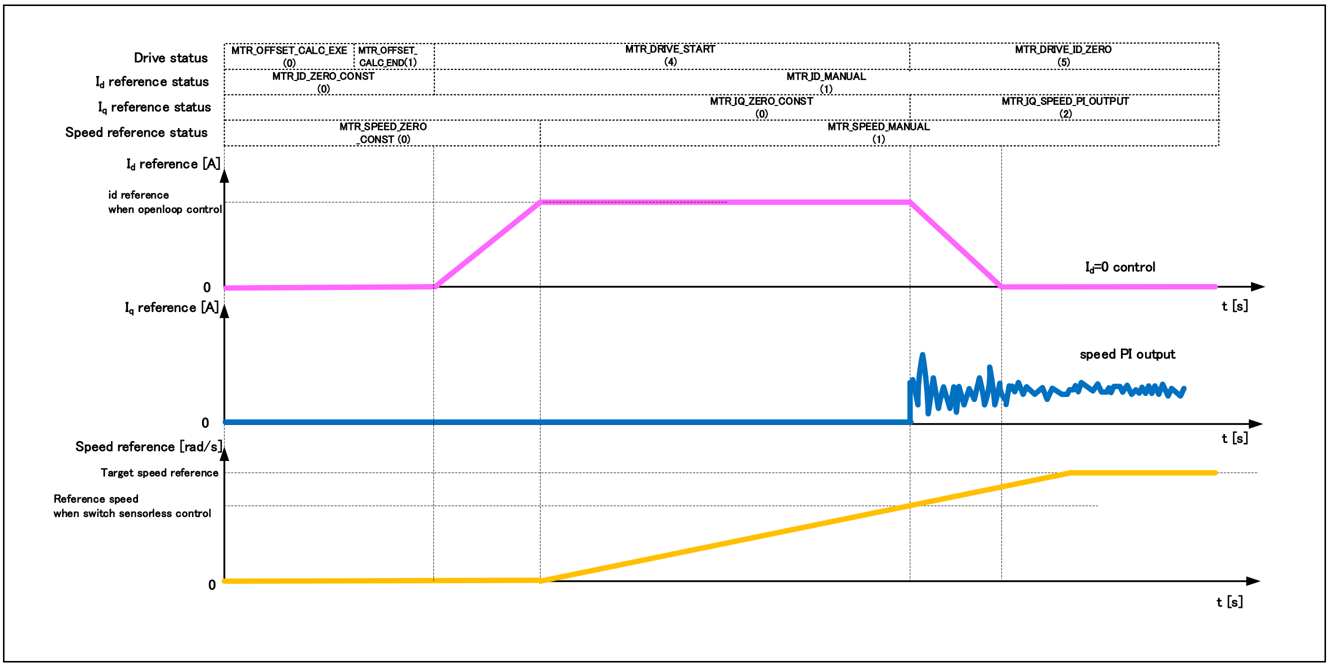


Figure 3-6 Description of Sensorless Speed Control Software Startup Control

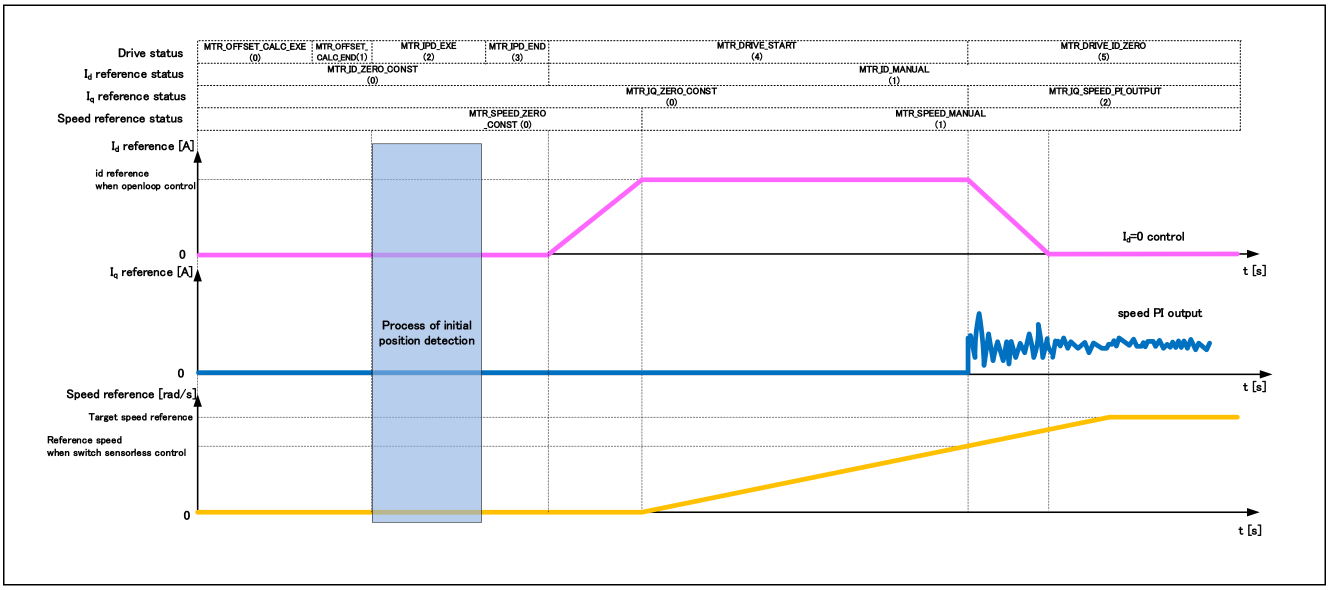


Figure 3-7 Description of Sensorless Speed Control Software Startup Control (IPD processing application)

3.1.8 Control method

The block diagram of the entire control system is shown in Figure 3-8. 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.

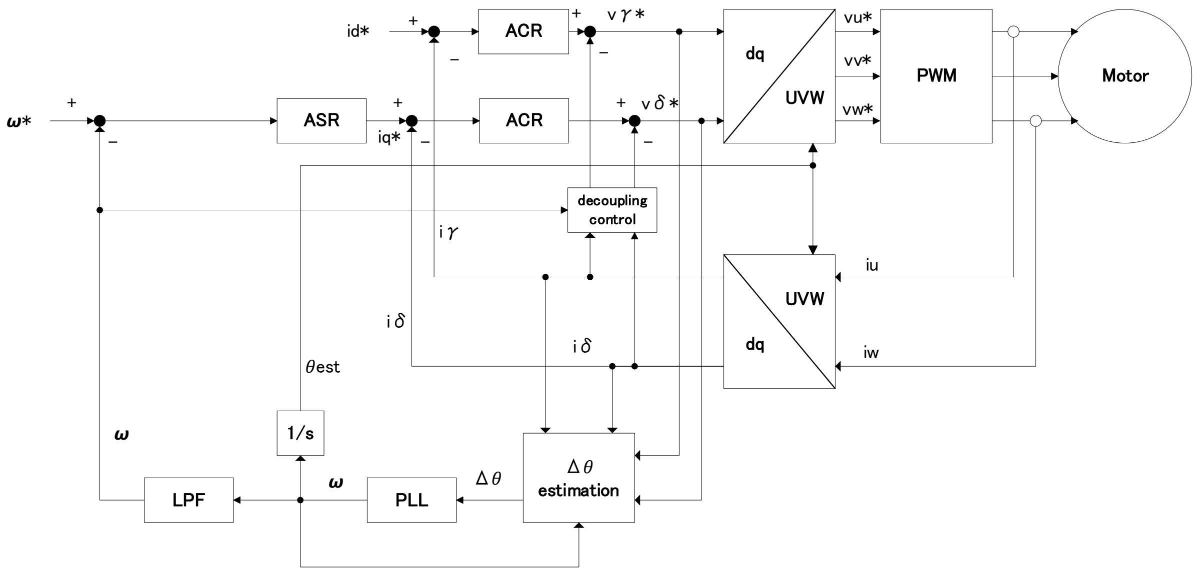


Figure 3-8 Control System Block Diagram

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 with which the above equation is multiplied.

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

The induced voltage 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 induced voltage, the γ-axis voltage and the δ-axis voltage are each used in approximation with voltage command values v_γ^* and v_δ^*. The estimated speed ω is found by constructing a feedback loop so that this phase error reaches 0 (PLL controller).

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

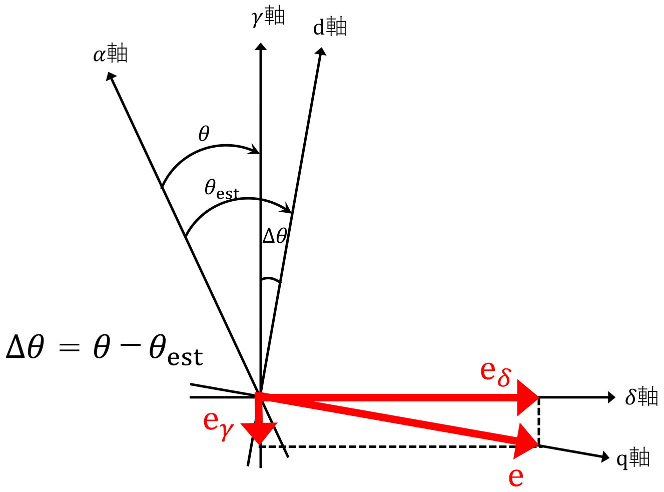


Figure 3-9 Phase Error and γ- and δ-axis Induced Voltage Constituent Elements

The ACR, ASR, and PLL controller are achieved by using the PI controller. Their gain requires suitable adjustment combined with the desired controls. The current PI control gain $K_{p_{ACR}}$ and $K_{i_{ACR}}$, the speed PI control gain $K_{p_{ACR}}$ and $K_{i_{ACR}}$, and the PLL control gain $K_{p_{PLL}}$ 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

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

ω_{SG}: Speed PI control natural frequency

J: Inertia

ψ_a: Induced voltage coefficient

P_n: Number of pole pairs

a: 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 3.7 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
U phase, V phase, and W 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 3.7 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]	100
Undervoltage error	Undervoltage limit [V]	12
	Monitoring cycle [μs]	100
Rotational speed error	Speed limit [rpm]	5300
	Monitoring cycle [μs]	100

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 physical value serving as a standard, 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 3.8. These values are defined in r_mtr_scaling_parameter.h.

Table 3.8 PU system base values

Category	Item	Definition	Unit
PU base physical quantity	Current	Rated current	[A]
	Voltage	Input voltage (inverter input)	[V]
	Angular frequency	2 π × 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]
	Back-EMF constant	Voltage / Angular frequency	[Wb]
	Inertia	Back-EMF constant × current × (number of pole pairs / angular frequency)^2	[kgm^2/rad]
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

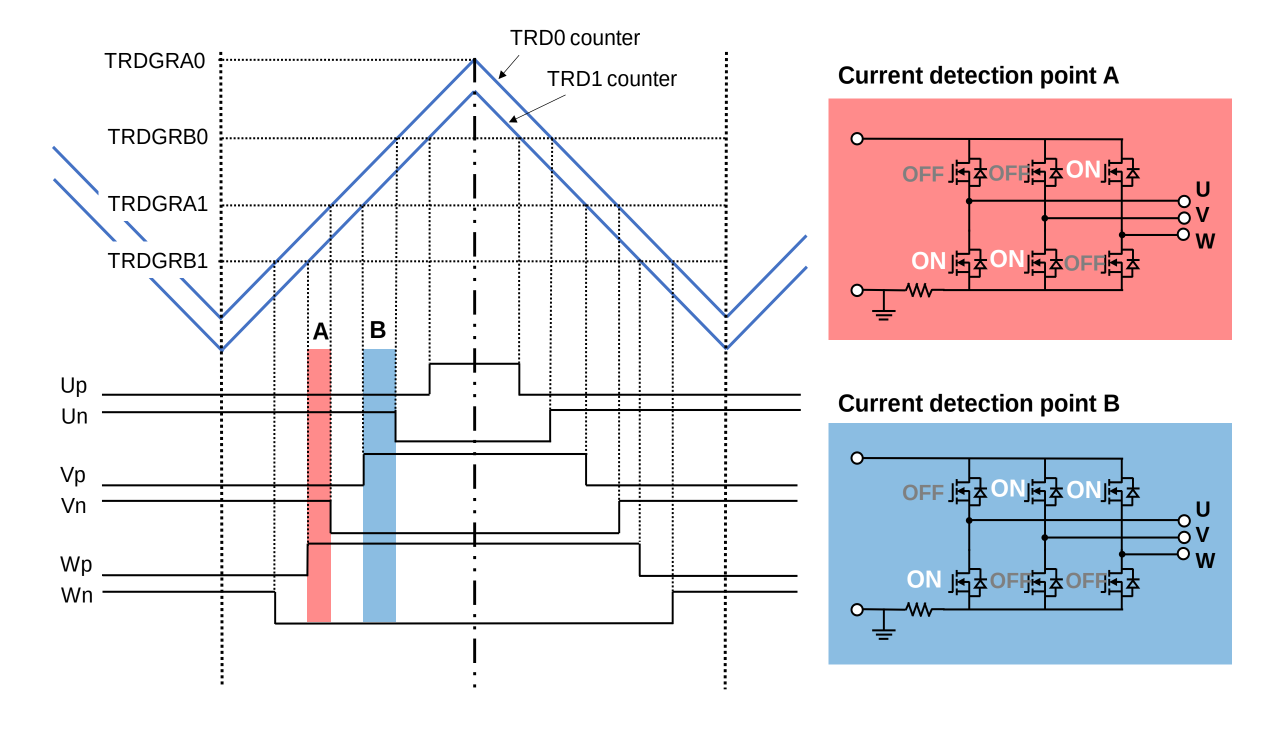


Figure 3-10 Complementally PWM waveform (For example, duty magnitude relation W > V > U).

In this program, the timer RD is used to control by three-phase PWM output with dead time in complementary PWM mode. Figure 3-10 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 invert the positive and negative.

$$I_B = -I_u$$

Since point C, which is the remaining one phase, is the current flowing in the V phase, the combined current between the U and W phases 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 three-phase current can be restored.

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/G1F 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 controlled according to the PWM duty setting. The program achieves this by using the following RL78/G1F functions:

- A/D converter hardware trigger no-wait mode
Perform A/D conversion triggered by the completion of the TAU0 channel 1 count.
In this case, select mode is used as the A/D conversion channel selection mode.
- Data transfer controller (DTC)
Allocate two A/D conversion values in RAM using the end of A/D conversion as the transfer trigger.
- Timer array unit (TAU)
Create timings for two A/D conversions by having channel 1 operate as an interval timer.

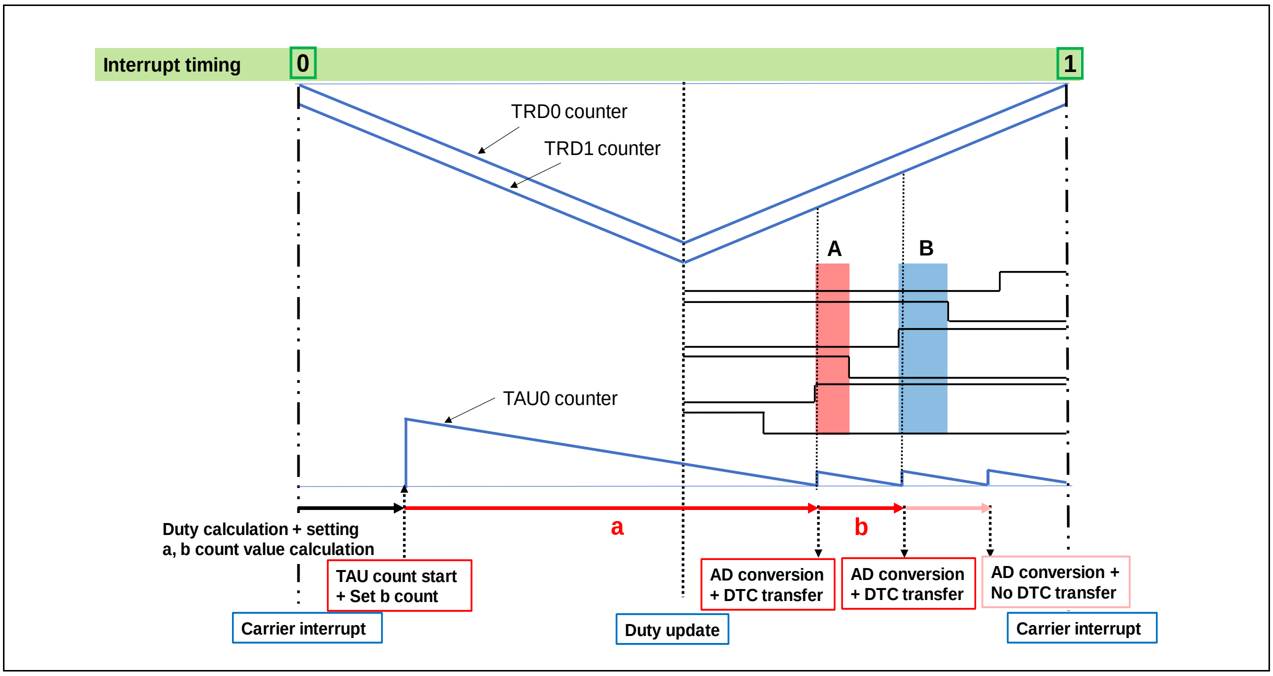


Figure 3-11 Timing of A/D Conversion

The description in this section is based on Figure 3-11.
The timer RD setting causes an interrupt at the carrier peak, the next duty is calculated, and then the count value for the timings of *a* and *b* are calculated. The duty that is set here is updated by reflecting the valley.
PWM is calculated at interrupt timing 0, and then the time data for *a* and *b* is calculated from that value. First, the value of *a* is set as the interval timer value for TAU0 channel 1. Next, TAU0 channel 1 is started. Once it is started, the interval timer value of TAU0 channel 1 is updated to *b*.
TAU0 channel 1 performs a countdown operation with a value of *a* and generates a count completion interrupt when the counter reaches 0. The A/D converter performs A/D conversion using this interrupt as a hardware trigger. When the A/D conversion is finished, the DTC transfers the A/D value at this point to the RAM for the first time.
TAU0 channel 1 then reloads the value of *b* and starts the countdown. At the end of the countdown with this value, another count completion interrupt is generated, A/D conversion is performed, and the DTC transfers the A/D value for the second time.
TAU0 channel 1 again reloads the value of *b* and starts the countdown. When this countdown ends, the count completion interrupt is generated again. However, since the number of DTC transfers is set to two, the data transfer at this interrupt is ignored and does not occur. (Internally, an interrupt is generated and A/D conversion is performed, but is not retained as a value.) TAU0 channel 1 then repeats the interval timer operation with a value of *b*, but no DTC transfer occurs.
The next control process can be executed at interrupt timing 1. Therefore, the 1-shunt resistor current's A/D conversion trigger is stopped by stopping the TAU0 channel 1 counter. The three-phase current can then be restored from the two A/D data points transferred by the DTC.

The values of *a* and *b* are calculated as follows.

[Count value of *a*]
Timer RD count value + minimum duty value
Since the counter clock of timer RD is twice that of TAU, the *a* value is shifted by 1 bit after these operations and set as the counter value of TAU. In addition, an adjustment value is used to account for physical effects (such as time difference due to external LPF).
The result is

a = (timer RD count value + minimum duty value + adjustment value) >> 1

[Count value of *b*]
This is the difference between the minimum duty value and the intermediate duty value. In addition, as above, an adjustment value is used to account for the effects of external conditions.

b = (intermediate duty value - minimum duty value + adjustment value) >> 1

3.1.11.1 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.

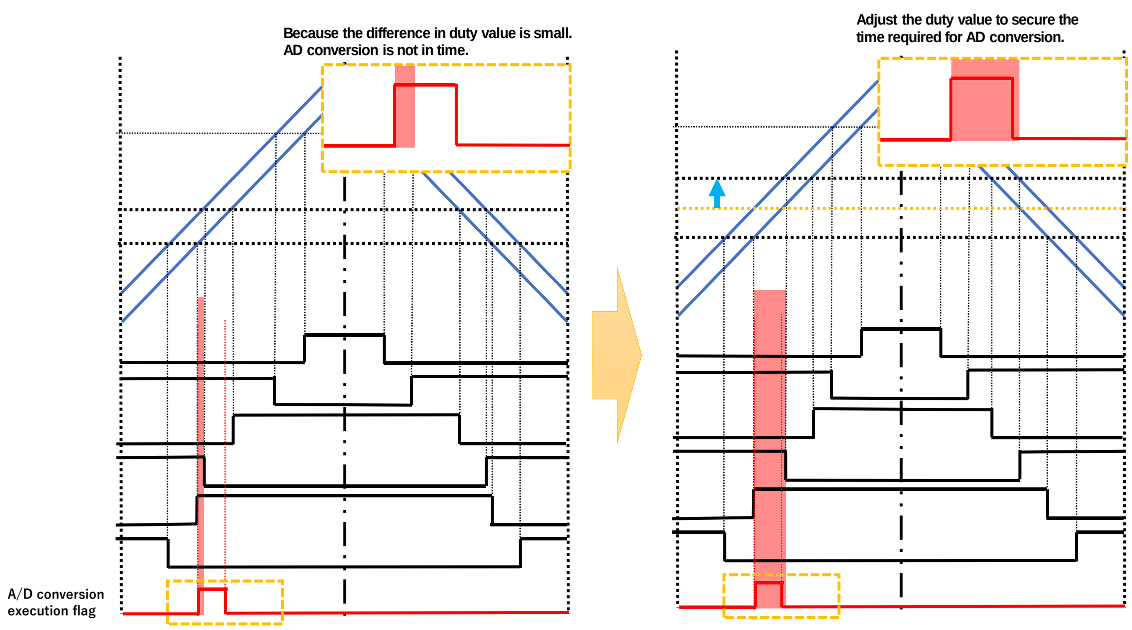


Figure 3-12 Duty adjustment (example)

The adjusted duty is compensated for in the next cycle. In other words, the duty is controlled such that the average thereof is equal to the command duty over two cycles. (See Figure 3-13)

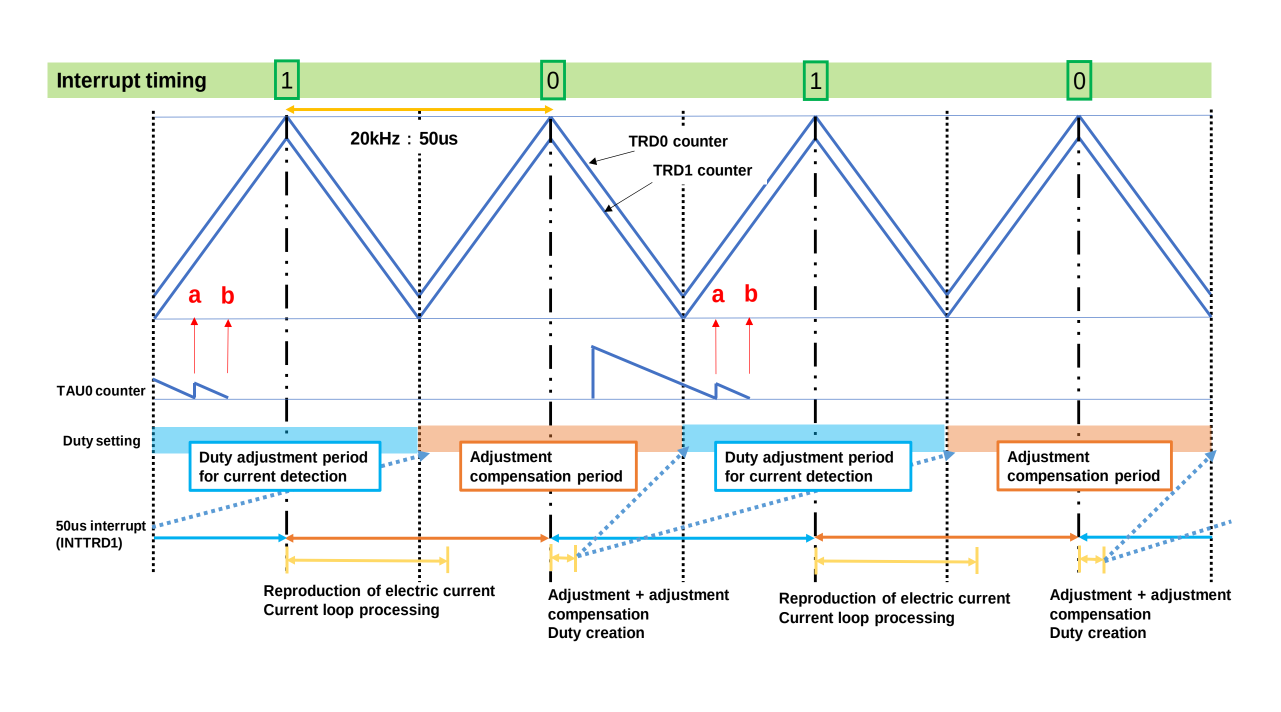


Figure 3-13 Periodic duty adjustment and compensation

3.1.12 Using saliency to detect initial position during stop

A motor is said to have saliency if the rotor position causes a different amount of rotor magnet magnetic flux through each phase coil. In other words, the magnetic resistance changes according to the position of the rotor. If the magnetic resistance changes into a sine-wave shape, the inductance will also change into a sine-wave shape. As shown in Figure 3-14, inductance changes to twice the number of cycles for each lap 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 3-15. It therefore takes longer when voltage is applied in the V→W direction than when voltage is applied in the W→U direction.

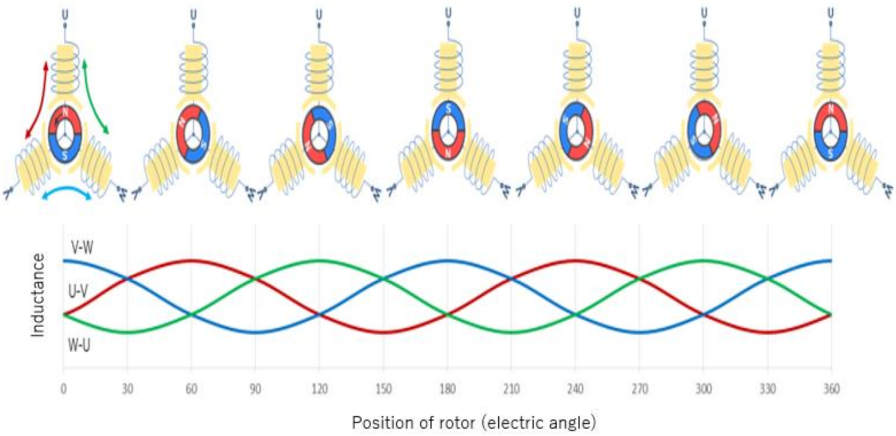


Figure 3-14 Changes in inductance according to rotor position

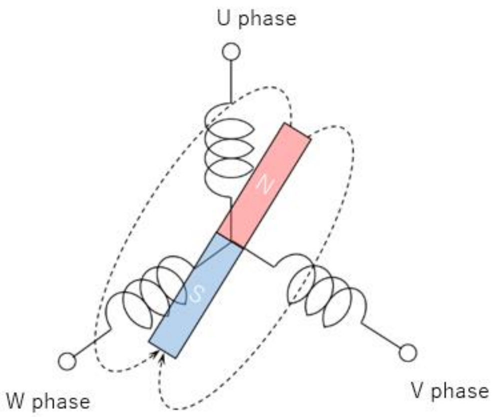


Figure 3-15 Relationship between rotor position and each phase

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 3-16. It is distinguished by applying 3 patterns of voltage, measuring the time taken until the current that flows in the shunt resistor reaches the threshold current, and comparing these to detect which direction the rotor is facing each 60 degrees within the 180 degrees of electrical angle.

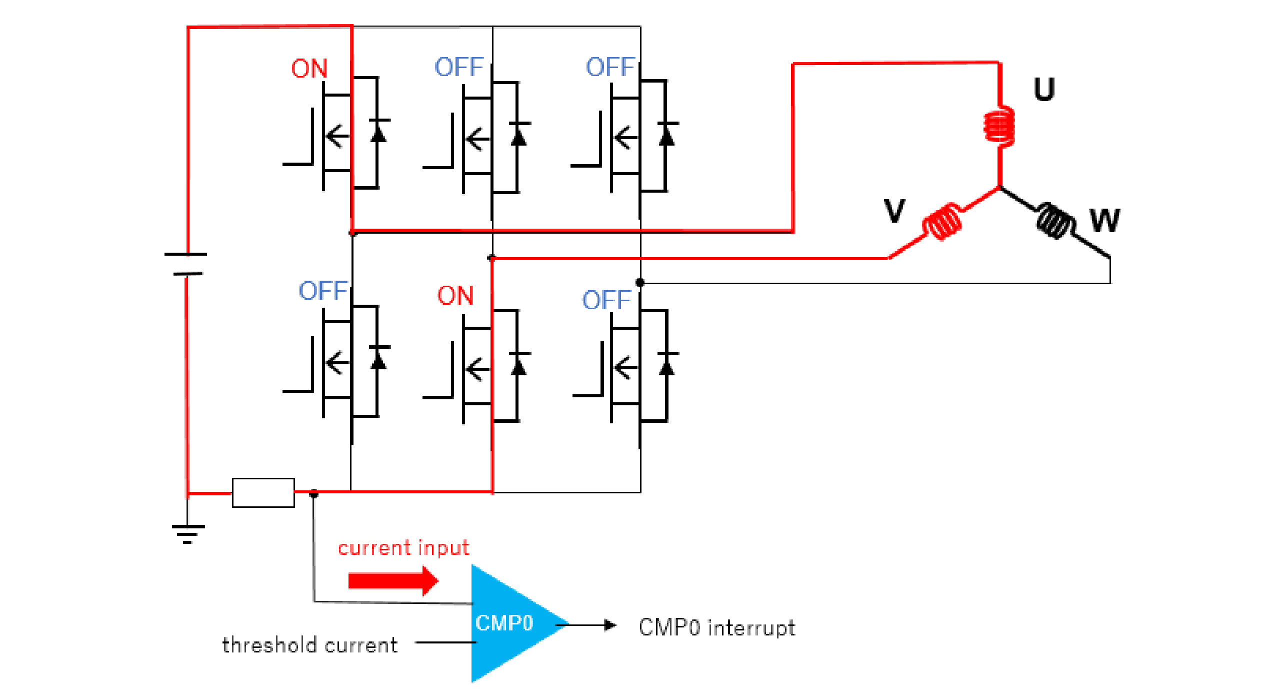


Figure 3-16 Angle detection diagram

The algorithm used in this system detects the time taken to reach the internal base current value using an RL78/G1F timer RX (TRX) and a comparator 0 (CMP0). It uses the TRD complementary PWM mode to apply pulse-shaped voltage to each phase. At this time, the count of the TRX starts at the rising edge synchronization of the TRD. The CMP0 generates an interrupt when it detects that the current flowing through the shunt resistor has reached the threshold current, and it measures the time required to increase the current.

Angle detection is performed every 60 degrees within the 180 degrees of electrical angle from the comparison with the cumulative time measured at each phase. Measurement stops when the cumulative measured time becomes greater than or equal to the threshold in the differential between the highest phase and lowest phase. However, if the differential does not reach the threshold after the maximum number of measurements, it is assumed that saliency is insufficient, and a transition is made to angle detection using saturation characteristics.

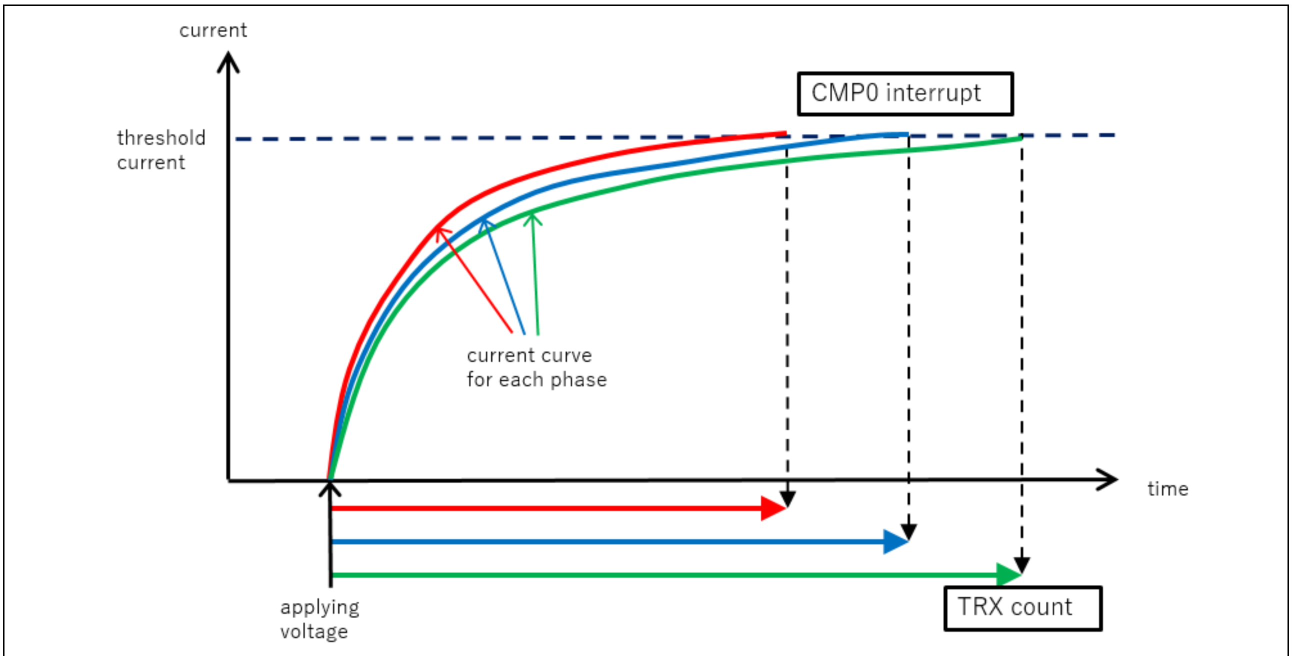


Figure 3-17 Current detection time differential among the 3 Phases

This rotor position detection method also takes measurements to confirm that the motor rotor has sufficient saliency to estimate its initial position. It compares the difference between the maximum value and median value with the difference between the median value and the minimum value of the TRX count, which increments with the current rise of the measured 3 phases, and identifies the maximum value or minimum value phases with the largest difference. 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 voltage orientation reversed. If the TRX count of the reversed phase has the same magnitude correlation, saliency is judged to be sufficient, and if it does not, saliency is judged to be too low.

For example, as shown in Figure 3-18, 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. It reverses the energization direction from the W-U phase, applies voltage to the U-W phase, and measures the time the current rises. It compares the median value of the U-V, V-W, and W-U phase TRX counts to the U-W phase TRX counts. 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.

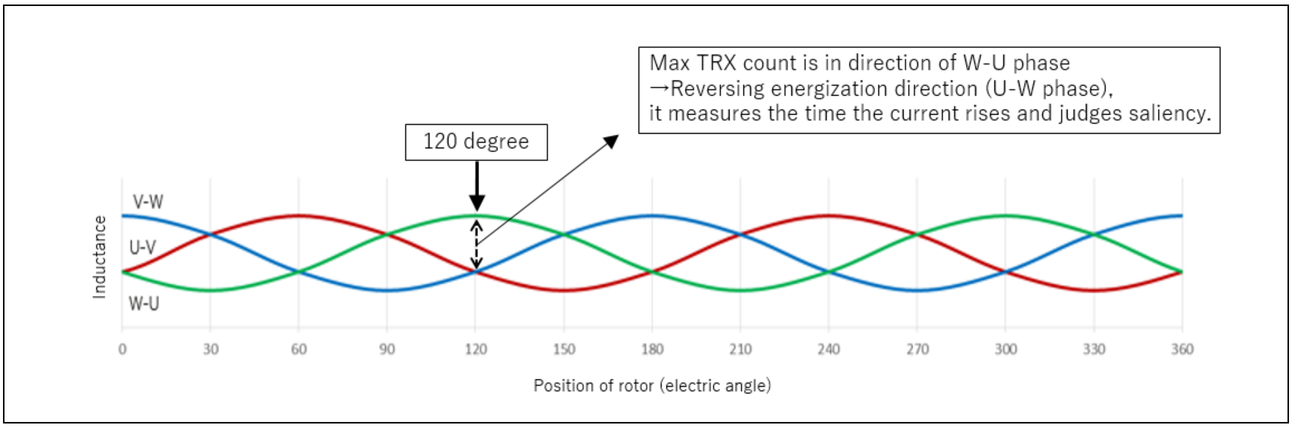


Figure 3-18 Saliency confirmation method

3.1.13 Initial position detection during stop using saturation characteristics

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.

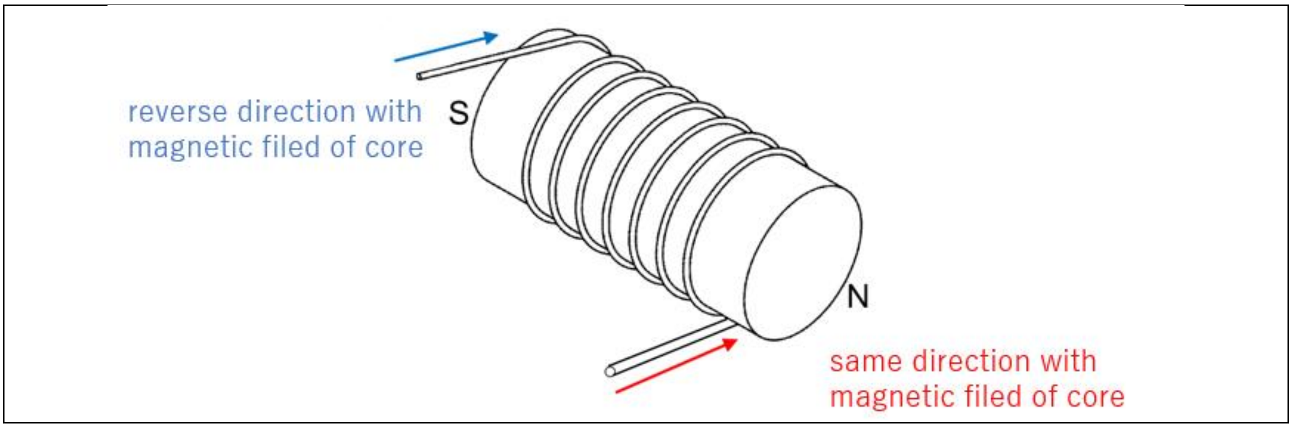


Figure 3-19 Example of magnetic pole wound with coil

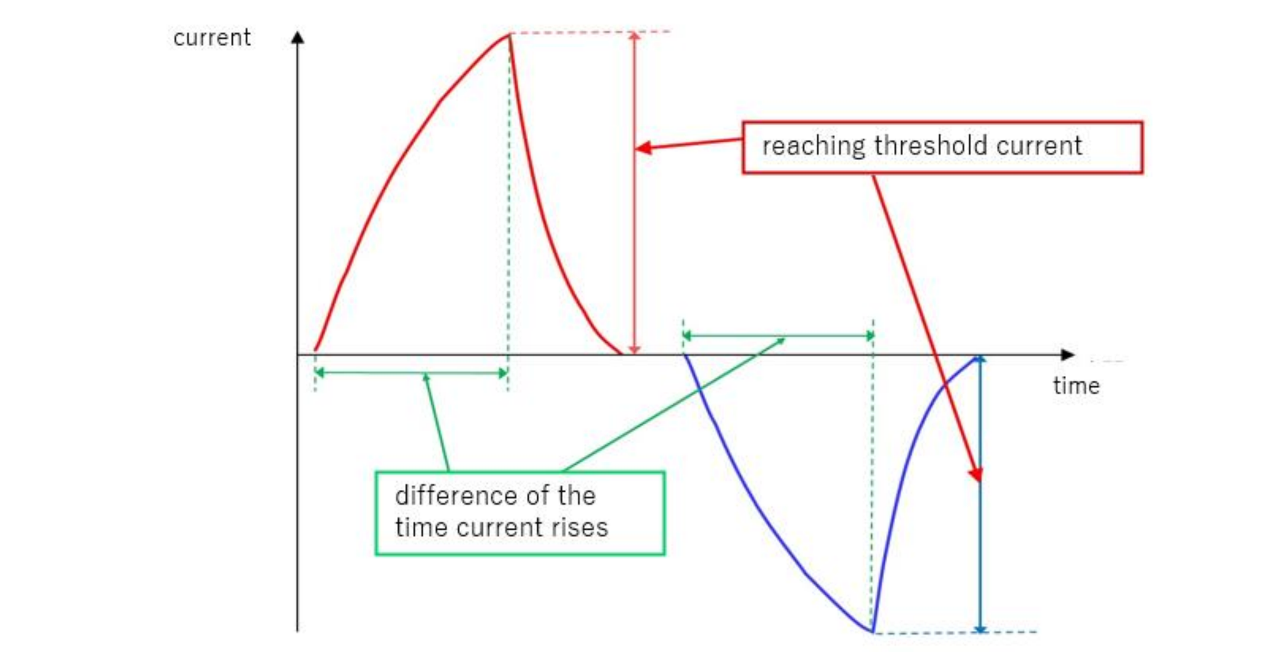


Figure 3-20 Current differential according to direction of applied current

Voltage is applied to the motor as shown in Figure 3-20, 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 application of the voltage matches the direction of rotation, so this tendency is used to estimate the orientation of the rotor.

If the motor is judged to be salient, in order to detect polarity of rotor, voltage is applied forward and backward based on angle information obtained using saliency. The current rise time is measured, and the polarity of the rotor is determined by comparing the magnitude correlation. If the motor is judged to be non-salient, the current rise time is measured by applying voltage in 6 directions, and the rotor position is estimated as if the rotor were oriented toward the phase with the minimum TRX count value.

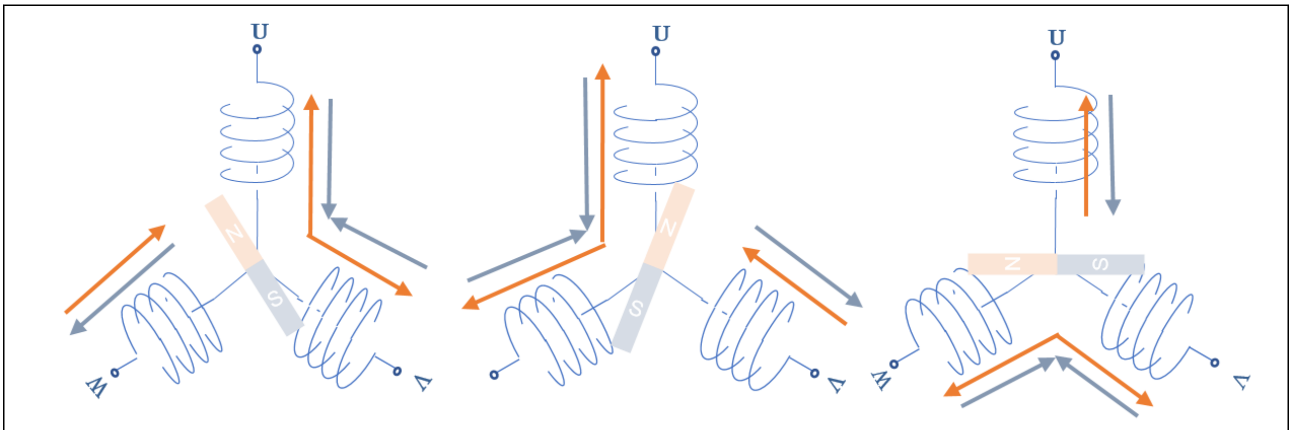
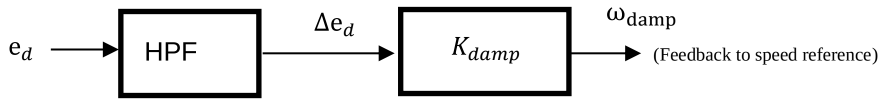


Figure 3-21 Example of applied voltage pattern due to initial position detection using magnetic saturation

Measurement stops when the differential among the directions exceeds the discrimination threshold. However, when the differential integral value does not reach the discrimination threshold even when measurement is carried out at the maximum measurement frequency, if the differential integral value is greater than the discrimination threshold, it is judged as an initial position detection succeeded. if the differential integral value is less than the discrimination threshold, judged as an initial position detection failure.

3.1.14 Open-loop Control

Position estimation is not possible because the induced voltage 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 3-22, 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 3-22 Block diagram of the open loop damping control

3.1.15 Field weakening control

The BEMF increases as the rotation speed increases. Therefore, in the high-speed range where the BEMF is about the same as the applied voltage, the torque current cannot be increased any more, and it may not be possible to increase the speed anymore.
At that time, there is a method called weak magnetic flux control in which a negative current is passed through the d-axis to reduce the BEMF in a pseudo manner to increase the torque current and increase the speed.
In this program, the drive able speed range is expanded by appropriately passing a d-axis current when a certain speed and command voltage are exceeded.

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_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
	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 s2_temp :: limited value	Limit between maximum and minimum values

Table 3.11 – List of Functions in “ics_RL78G1F.obj”

File name	Function name	Processing overview
ics_RL78G1F.obj	ics2_init argument: unsigned int addr :: DTC vector table start address char pin :: Pins used by SCI char level :: Interrupt level char num :: Top address of DTC structure char brr :: communication speed char mode :: Communication mode return: none	Communication initialization
	ics2_watchpoint argument: none return: none	Call transfer function Must be called at intervals of 250us 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_ctrl_rl78g1f.c”(1/2)

File	Function	Process overview
mtr_ctrl_rl78g1f.c	R_MTR_InitHardware Input: none Output: none	Initialization of clock and peripheral functions
	R_MTR_CtrlStart Input: none Output: none	Timer RD (PWM) output authorization
	R_MTR_CtrlStop Input: none Output: none	Timer RD (PWM) output stopped Initialization of register
	R_MTR_CtrlBrake Input: none Output: none	Timer RD (PWM) output stopped Initialization of register Low arm ON
	R_MTR_GetAdc Input: uint8_t u1_ad_ch :: A/D channel Output: uint16_t :: A/D conversion result	A/D conversion
	R_MTR_GetIuwAdc Input: uint16_t *u2_ad_iuwvdc :: UW phase pointer Output: none	UW phase current detection A/D conversion
	R_MTR_GetVdcAdc Input: none Output: none	Voltage detection A/D conversion
	R_MTR_InitAdcTrigger Input: none Output: none	Initialization of A/D conversion trigger settings ・ A/D mode setting ・ Reset DTC
	R_MTR_ResetAdcTrigger Input: none Output: none	Reset A/D conversion trigger ・ Reset DTC
	R_MTR_StartAdcTrigger Input: uint16_t u2_ad_1st :: Trigger timing count value A uint16_t u2_ad_2nd :: Trigger timing count value B Output: none	Start of A/D conversion trigger ・ A/D conversion settings ・ A/D conversion trigger timer setting and timer start
	R_MTR_RecoverForcedShutdown Input: none Output: none	Forced cutoff flag monitoring and cutoff release processing
	mtr_init_unused_pins (inline function) Input: none Output: none	Initialization of unused terminals
	mtr_init_clock (inline function) Input: none Output: uint16_t / clock setting error	Initialization of clock
	mtr_init_ui (inline function) Input: none Output: none	Initialization of user interface
	mtr_init_tau (inline function) Input: none Output: none	Initialization of timer array unit (TAU)
	mtr_init_inttm00_interrupt (inline function) Input: none Output: none	TAU00 interrupt setting initialization

Table 3.14 List of Functions in “r_mtr_ctrl_rl78g1f.c”(2/2)

File	Function	Process overview
mtr_ctrl_rl78g1f.c	mtr_init_inttm01_interrupt (inline function) Input: none Output: none	TAU01 interrupt setting initialization
	mtr_init_trd (inline function) Input: none Output: none	Initialization of TRD
	mtr_init_trx (inline function) Input: none Output: none	Initialization of TRX
	mtr_init_ad_converter (inline function) Input: none Output: none	Initialization of A/D converter
	mtr_init_pwm_register (inline function) Input: none Output: none	Initialization of PWM output register
	mtr_init_dtc (inline function) Input: none Output: none	Initialization of Data transfer controller (DTC)
	mtr_init_pga (inline function) Input: none Output: none	Initialization of Programmable gain amplifier (PGA)
	mtr_init_cmp0 (inline function) Input: uint8_t u1_level :: Overcurrent level Output: none	Initialization of Comparator (CMP0)
	mtr_init_pwmopa (inline function) Input: none Output: none	Initialization of PWM option unit A (PWMOPA)

Table 3.15 List of Functions in “r_mtr_ctrl_rl78g1f.h”

File	Function	Process overview
mtr_ctrl_rl78g1f.h	R_MTR_ClearPWMInterruptFlag Input: none Output: none	Clear interrupt flag
	R_MTR_EnableDTC Input: none Output: none	DTC enable
	R_MTR_DisableDTC Input: none Output: none	DTC disable
	R_MTR_StartTrd() argument: none return: none	Start TRD
	R_MTR_SetC0RVM Input: (uint8_t) i :: Internal reference voltage 0 ≦ i ≦ 265 Output: none	Comparator internal reference voltage setting
	R_MTR_ClearWDT Input: none Output: none	Clear watchdog timer (WDT)

Table 3.16 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, const :: Control Gain structure pointer st_mtr_design_parameter_t *st_ctrl_param :: Design parameter structure pointer Output: none	Gain design process

Table 3.17 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) u2_charge_cap_error :: timeout error	Waiting for stability of bus voltage
	R_MTR_SetSpeed Input: (int16_t) s2_ref_speed_rpm / target rotational speed Output: none	Set speed command value
	R_MTR_GetSpeed Input: none Output: int16_ts2_speed_rpm :: rotational speed	Get speed
	R_MTR_SetDir Input: (int8_t) gst_foc.s1_dir :: direction of rotation Output: none	Set direction of rotation
	R_MTR_Get_Dir Input: none Output: (int8_t) gst_foc.s1_dir :: direction of rotation	Get direction of rotation
	R_MTR_GetStatus Input: none Output: (uint8_t) mtr_statemachine_get_status(g_st_120.st_stm) :: motor status	Get motor status
	R_MTR_GetErrorStatus Input: none Output: (uint16_t) g_st_120.u2_error_status :: 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 (inline function) Input: none Output: none	Input values of ICS buffer variables to control variables
	R_MTR_InputBuffParamReset Input: none Output: none	Reset ICS buffer variables
	R_MTR_UpdatePolling Input: none Output: none	Set control variables

Table 3.18 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: none	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: none	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: none	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: none	Stop motor
	mtr_act_drive Input: (st_mtr_statemachine_t) *st_stm :: motor status structure (void) *p_param ::structure for control variables Output: none	Reset control variables
	mtr_act_stop Input: (st_mtr_statemachine_t) *st_stm :: motor status structure (void) *p_param ::structure for control variables Output: none	Stop motor

Table 3.19 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 Output: none	Initialization of control variables
	R_MTR_FOCMotorReset Input: st_mtr_foc_t *st_foc :: FOC structure pointer Output: none	Reset control variables

Table 3.20 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 *s2_ref_vdq :: dq axis voltage command value variable pointer int16_t s2_speed_rad :: speed Output: none	Phase error estimating process

Table 3.21 List of Functions in “R_DSP_RL78_CC_S.lib”(1/2)

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 ≦u, w ≦$4\sqrt{2/3}$-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 ≦u, w ≦$4\sqrt{2/3}$-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) 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) 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 ≦u, w ≦$4\sqrt{2/3}$-1/2048 (FIX13) 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: $-4\sqrt{2/3}$ ≦u, w ≦$4\sqrt{2/3}$ (FIX13) 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.
	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) Output: none Output range: $-4\sqrt{2/3}$ ≦u, v, w ≦$4\sqrt{2/3}$ (FIX13)	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.

Table 3.22 List of Functions in “R_DSP_RL78_CC_S.lib”(2/2)

<table>
<tr><th>File</th><th>Function</th><th>Process overview</th></tr>
<tr><td></td><td>R_motor_uv2dq_abs_sat_pu_FIX13
input: st_coordinate13 *p_coordinate13 :: Coordinate transformation structure pointer
Input range: $-4\sqrt{2/3} \leqq u, v \leqq 4\sqrt{2/3}$ (FIX13)
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></td><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></td><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></td><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>
<tr><td></td><td>R_motor_sqrt2_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 $\sqrt{x^2+y^2} \leqq$ 4-1/8192 (FIX13)
Output: int16_t $\sqrt{x^2+y^2}$
Output range: 0 ≦sqrt ≦4-1/8192 (FIX13)</td><td>Calculate the value of $\sqrt{x^2+y^2}$ from the input values x and y and output as the (FIX12) type.</td></tr>
</table>

Table 3.23 List of Functions in "r_mtr_interrupt.c" (1/4)

File	Function	Process overview
r_mtr_interrupt.c	mtr_carrier_interrupt [Used in 1-shunt mode] Input: none Output: none	Cycle timer interrupt (Call using INTTRD0) Cycle: 50 µs - current detection - Current detection offset correction process - Initial position detection process - Call overcurrent error monitoring process - Vector calculation - Call decoupling control process - Position/speed estimation calculation - Call current PI control process - Call deadtime compensation process - Call modulation process - Call PWM duty setting process - Call communication process
	mtr_100usec_interrupt [Used in 3-shunt mode] Input: none Output: none	Cycle timer interrupt (Call using INTTM01) Cycle: 100 µs - U, W phase current detection - Current detection offset correction process - Initial position detection process - Call overcurrent error monitoring process - Vector calculation - Call decoupling control process - Position/speed estimation calculation - Call current PI control process - Call deadtime compensation process - Call modulation process - Call PWM duty setting process - Call communication process
	mtr_1ms_interrupt Input: none Output: none	Cycle timer interrupt (Call using INTTM00) Cycle: 1 ms - Bus voltage detection - Calculation of inverse voltage - Startup control - Field-weakening 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_set_duty_adj (inline function) Input : st_mtr_sscs_t * st_sscs :: 1-shunt resistor current detection structure pointer Output : None	PWM adjustment / adjustment compensation duty setting [Used in 1-shunt mode]
	mtr_calib_current_offset_dc (inline function) Input : st_mtr_sscs_t * st_sscs :: three-phase current detection structure pointer Output : None	PWM adjustment compensation duty setting [Used in 3-shunt mode]
	mtr_calib_current_offset_uw (inline function) Input: st_mtr_tscs_t *st_tscs :: three-phase current detection structure pointer uint16_t *s2_iuw_ad :: UW current pointer Output: uint8_t :: current offset detection process completion flag	Current offset detection process [Used in 3-shunt mode]

Table 3.24 List of Functions in “r_mtr_interrupt.c” (2/4)

File	Function	Process overview
r_mtr_interrupt.c	mtr_current_offset_adjustment_uvw (inline function) Input: st_mtr_tscs_t *st_tscs :: three-phase current detection structure pointer int16_t *s2_iuw_ad :: UVW current pointer int16_t s2_limit_over_current :: overcurrent limit value Output: uint16_t :: error status	Offset elimination process and overcurrent error detection [Used in 3-shunt mode]
	mtr_lpf1_run (inline function) Input: st_mtr_lpf1_t *st_lpf :: LPF structure pointer int16_t s2_input :: LPF input const uint8_t u1_q :: Q value of LPF Output: none	Primary LPF process
	mtr_current_pi_ctrl (inline function) Input: st_mtr_acr_t *st_acr :: ACR structure pointer int16_t *s2_idq :: dq axis current variable pointer int16_t *s2_ref_idq :: dq axis current command value pointer int16_t *s2_ref_vdq :: dq axis voltage command value pointer Output: none	Current PI process
	mtr_decoupling (inline function) Input: st_mtr_foc_t *st_foc :: FOC structure pointer int16_t s2_speed_rad :: Speed const st_mtr_parameter_t *p_mtr :: Motor parameter structure pointer Output: none	Decoupling control process
	mtr_deadtime_comp (inline function) Input: st_mtr_deadtime_comp_t *st_dtcomp :: deadtime compensation structure pointer st_coordinate13 * st_i_uvw_dq :: Current coordinate system structure pointer st_coordinate13 * st_v_uvw_dq :: Voltage coordinate system structure pointer Output: none	Deadtime compensation process
	mtr_uvw_voltage_limit (inline function) Input: int16_t *s2_ref_v_uvw :: UVW phase voltage pointer int16_t s2_voltage_limit :: voltage limit value Output: none	Three-phase voltage limit processing
	mtr_mod_ts (inline function) Input: st_mtr_mod_t *st_mod :: modulation structure pointer st_coordinate13 * st_ref_v :: Voltage command value Coordinate system structure pointer Output: none	Modulation process [Used in 3-shunt mode]
	mtr_pwm_duty_ts (inline function) Input: st_mtr_tscs_t *st_tscs :: three-phase current detection structure pointer int16 s2_u :: u phase modulation factor int16 s2_v :: v phase modulation factor int16 s2_w :: w phase modulation factor Output: none	Duty calculation

Table 3.25 List of Functions in "r_mtr_interrupt.c" (3/4)

File	Function	Process overview
r_mtr_interrupt.c	mtr_mod_ss (inline function) Input: st_mtr_mod_t *st_mod :: modulation structure pointer st_coordinate13 * st_ref_v :: Voltage command value Coordinate system structure pointer uint8_t *u1_drv_pat :: PWM magnitude relationship pattern Output: none	Modulation process [Used in 1-shunt mode]
	mtr_calib_current_offset_ss (inline function) Input: st_mtr_sscs_t * st_sscs :: 1-shunt resistor current detection structure pointer Output: None	Offset removal processing when 1-shunt resistor current detected [Used in 1-shunt mode]
	mtr_pwm_duty_ss (inline function) Input: st_mtr_sscs_t * st_sscs :: 1-shunt resistor current detection structure pointer int16_t s2_u :: U phase duty int16_t s2_v :: V phase duty int16_t s2_w :: W phase duty Output: None	Calculation of adjustment duty when 1-shunt resistor current detected [Used in 1-shunt mode]
	mtr_repro_current (inline function) Input: st_mtr_sscs_t * st_sscs :: 1-shunt resistor current detection structure pointer int16_t * s2_i_uvw :: 3-phase current pointer int16_t s2_limit_over_current :: Overcurrent limit value Output: None	Current reproduction processing [Used in 1-shunt mode]
	mtr_limit (inline function) Input: int16_t s2_value :: Input value int16_t s2_max :: upper limit value int16_t s2_min :: Lower limit value output:: None	Limit processing [Used in 1-shunt mode]
	mtr_set_speed_ref (inline function) Input: st_mtr_foc_t *st_foc :: FOC structure pointer Output: int16 s2_speed_rad_ref_buff :: speed command value	Set command value for speed control
	mtr_pi_run (inline function) Input: st_mtr_pi_t *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 s2_pi_out :: PI output	PI control process
	mtr_set_iq_ref (inline function) Input: st_mtr_foc_t *st_foc :: FOC structure pointer Output: int16 s2_iq_ref_buff :: q-axis current command value	Set q-axis current command value
	mtr_set_id_ref (inline function) Input: st_mtr_foc_t *st_foc :: FOC structure pointer Output: int16 s2_id_ref_buff :: d-axis current command value	Set d-axis current command value

Table 3.26 List of Functions in “r_mtr_interrupt.c” (4/4)

File	Function	Process overview
r_mtr_interrupt.c	mtr_error_check (inline function) Input: st_mtr_foc_t *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_cmp0_interrupt [Used in IPD mode] Input: none Output: none	Get TRX count during initial position detection

Table 3.27 List of Functions in “r_mtr_ipd.c” (1/2)

File	Function	Process overview
r_mtr_ipd.c	R_MTR_IpdProcess Input: st_mtr_ipd_t *st_ipd :: IPD structure uint8_t u1_current_offset :: current offset uint16_t u2_error_status :: error status Output: None	Initial position detection process
	R_MTR_ResetForDrive Input: st_mtr_ipd_t *st_ipd :: IPD structure uint8_t u1_current_offset :: current offset int8_t s1_dir :: direction of rotation FIX12 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 *st_ipd :: IPD structure uint16_t 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 *st_ipd :: IPD structure int8_t u1_current_offset :: current offset uint16_t 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 *st_ipd :: IPD structure uint8_t u1_current_offset :: current offset uint16_t 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 *st_ipd :: IPD structure uint8_t u1_current_offset :: current offset uint16_t u2_error_status :: error status Output: None	Angle detection process for initial position detection with a non-salient motor
	mtr_set_initial_position Input: None 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

表 3.28 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_cmp0_intr Input: None Output: None	Enable the interrupt of CMP0
	mtr_disable_cmp0_intr Input: None Output: None	Disable the interrupt of CMP0

Table 3.29 List of Functions in “r_mtr_ipd.h”

File	Function	Process overview
r_mtr_ipd.h	R_MTR_SetCutoffSource() Input: None Output: None	Set cutoff source of PWMOPA for CMP0
	R_MTR_ClearCutoffSource() Input: None Output: None	Clear cutoff source of PWMOPA
	R_MTR_StartTrx() Input: None Output: None	Start TRX
	R_MTR_StopTrx() Input: None Output: None	Start TRX
	R_MTR_StopTrd() Input: None Output: None	Stop TRD
	R_MTR_StartTau() Input:None Output: None	Start TAU
	R_MTR_StopTau() Input: None Output: None	Stop TAU
	R_MTR_EnableTauIntr Input: None Output: None	Enable TAU interrupt
	R_MTR_DisableTauIntr Input: None Output: None	Disable TAU interrupt

Table 3.30 List of Functions in “r_mtr_fw_ctrl.obj”

File	Function	Process overview
r_mtr_fw_ctrl.obj	R_MTR_WeakenControl Input: st_mtr_fw_t *st_fw :: Field-Weakening Control structure pointer int16_t *s2_ref_vdq :: dq axis voltage command value variable pointer int16_t s2_iq :: q-axis current int16_t s2_speed_rad :: speed Output: d-axis current	Field-Weakening control process

Table 3.31 List of Functions in ““r_mtr_damp_ctrl.obj”

File	Function	Process overview
r_mtr_damp_ctrl.obj	R_MTR_DampCtrl Input: st_mtr_damp_t *st_damp :: Phase error estimation structure pointer int16_t *s2_ref_vdq :: d-axis induced voltage command value variable pointer int16_t s2_speed_ref :: reference speed Output: reference speed	Damping control process

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 mentioned. Also, 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 scale field expresses that the factional part is n bits. 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 3.32 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 3.33 List of Variables in "r_mtr_ics.c"(1/2)

Variable	Type	Qn	PU	Content	Remarks
com_u1_direction	uint8_t	Q0	-	Direction of rotation	0: CW 1: CCW
com_f4_mtr_r	float	-	-	Resistance [Ω]	
com_f4_mtr_ld	float	-	-	d-axis inductance [H]	
com_f4_mtr_lq	float	-	-	q-axis inductance [H]	
com_f4_mtr_m	float	-	-	Induced voltage constant [Vs/rad]	
com_f4_mtr_j	float	-	-	Rotor inertia [kgm^2]	
com_u2_mtr_pp	uint16_t	Q0	-	Number of pole pairs	
com_u2_offset_calc_time	uint16_t	Q0	-	Current offset detection time	
com_s2_ref_speed_rpm	int16_t	Q0	-	Command 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_s2_less2ol_speed_rpm	int16_t	Q0	-	Switching speed from sensorless to open loop [rpm]	Mechanical angle
com_s2_ol2less_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_init_asr_intg	float	-	-	ASR integral term initial value during sensorless transition	
com_f4_asr_ki_adj	float	-	-	Speed PI control integral term adjustment parameter	
com_f4_ramp_limit_current	float	-	-	Limit value for current rise [A/ms]	
com_s2_duty_diff_limit	int16_t	Q0	-	Minimum value of duty deviation between phases	
com_s2_ad_point_a_adj_cnt	int16_t	Q0	-	Adjustment for A/D delay counts for A point	
com_s2_ad_point_b_adj_cnt	int16_t	Q0	-	Adjustment for A/D delay counts for B point	
com_f4_sal_angle_current	float	-	-	Threshold current of angle detection for salient rotor	【IPD】
com_u4_sal_angle_th	uint32_t	Q0	-	TRX count value differential of angle detection for salient rotor	
com_u2_sal_angle_discharge	uint16_t	Q0	-	Discharge period of angle detection for salient rotor	
com_f4_sal_polarity_current	float	-	-	Threshold current of polarity detection for salient rotor	
com_u4_sal_polarity_th	uint16_t	Q0	-	TRX count value differential of polarity detection for salient rotor	
com_u2_sal_polarity_discharge	uint8_t	Q0	-	Discharge period of polarity detection for salient rotor	
com_f4_non_sal_current	float	-	-	Threshold current of angle detection for non-salient rotor	
com_u4_non_sal_th	uint16_t	Q0	-	TRX count value differential of angle detection for non-salient rotor	

Table 3.34 List of Variables in “r_mtr_ics.c”(2/2)

Variable	Type	Qn	PU	Content	Remarks
com_u2_non_sal_discharge	uint8_t	Q0	-	Discharge period of angle detection for non-salient rotor	
com_s2_speed_th_rpm	int16_t	Q0	-	Field-Weakening Control speed threshold [rpm]	
com_f4_v_mag_th	float	-	-	Maximum output voltage [V]	
com_f4_delta_id	float	-	-	Id change amount of field-weakening Control [A]	
com_f4_damp_hpf_cof_hz	float	-	-	HPF cutoff frequency for damping control [Hz]	
com_f4_damp_zeta	float	-	-	Damping coefficient of damping control	
com_f4_damp_speed_limit_rate	float	-	-	Damping control speed limit	
com_s2_enable_write	int16_t	Q0	-	Variable to allow variable rewriting	
g_s2_enable_write	int16_t	Q0	-	Variable to allow variable rewriting	
st_ics_input	mtr_ctrl_input_t		-	Structure for ICS variable transfer	Structure

Table 3.35 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	-	countor	

Table 3.36 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 3.37 List of Variables in “r_mtr_interrupt.c”

Variable	Type	Qn	PU	Content	Remarks
gst_foc	st_mtr_foc_t	Q0	-	Vector control structures	Structure
g_u1_cnt_ics	static uint8_t	Q0	-	Communication process cycle pixel skipping variable	
g_u1_carrier_intr_counter	static uint8_t	Q0	-	Carrier interrupt counter	

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 3.38 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	Q19	Inductance (resistance/angular frequency)	d-axis inductance [PU]	
s2_mtr_lq	int16_t	Q19	Inductance (resistance/angular frequency)	q-axis inductance [PU]	
s2_mtr_m	int16_t	Q15	Induced voltage constant (voltage/angular frequency)	Induced voltage constant [PU]	
s2_mtr_j	int16_t	Q11	Inertia (Induced voltage constant × current × (number of pole pairs/angular frequency)^2)	Inertia [PU]	

Table 3.39 List of Variables in “r_mtr_ctrl_gain.h” / Structure: “st_mtr_design_parameter_t”

Variable	Type	Qn	PU	Content	Remarks
s2_acr_nf_hz	float	Q0	-	Current PI control natural frequency [Hz]	
s2_asr_nf_fz	float	Q0	-	Speed PI control natural frequency [Hz]	
s2_asr_lpf_nf_hz	float	Q0	-	Speed LPF cut-off frequency [Hz]	
f4_acr_lpf_cof_hz	float	Q0	-	Current LPF cut-off frequency [Hz]	
s2_pll_nf_hz	float	Q0	-	PLL natural frequency [Hz]	
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]	
f4_lq	float	Q0	-	q-axis inductance [H]	
f4_m	float	Q0	-	Back-EMF constant [V･s/m]	
f4_j	float	Q0	-	Rotor inertia [[kgm^2]	
f4_ol_ref_id	float	Q0	-	Open Loop reference Id [A]	
f4_ol2cl_speed	float	Q0	-	Switching speed from open loop [rpm]	
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_pu_sf_afreq	float	Q0	-	Integral term adjustment parameters for speed PI control	
f4_asr_ki_adj	float	Q0	-	Number of pole pairs	
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_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_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_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_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	

Table 3.40 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	Q17	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	Q18	Resistance	q-axis current control integral gain*operation period	
s2_asr_pi_kp	int16_t	Q12	Current/angular frequency	Speed control proportional gain	
s2_asr_pi_kidt	int16_t	Q16	Current/angular frequency	Speed control integral gain*operation period	
s2_asr_lpf_in_k	int16_t	Q14	-	Speed LPF input coefficient	
s2_acr_lpf_in_k	int16_t	Q14	-	Current LPF input coefficient	
s2_pll_kp	int16_t	Q14	Angular frequency/angle	PLL proportional gain	
s2_pll_kidt	int16_t	Q18	1 angular frequency/angle	PLL integral gain*operation period	
s2_damp_k	int16_t	Q12	Speed/Voltage	Damping control gain	
s2_damp_hpf_k	int16_t	Q15		Damping control HPF	

Table 3.41 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 3.42 List of Variables in “r_dsp_cc_s.h / 構造体：” st_coordinate12”

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 phase (FIX13)	
b	int16_t	Q13	-	β phase (FIX13)	
d	int16_t	Q13	-	d phase (FIX13)	
q	int16_t	Q13	-	q phase (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 3.43 List of Variables in “r_mtr_driver_access.h” / Structure: “st_mtr_ctrl_input_t”

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_less2ol_speed_rad	int16_t	Q14	Angular frequency	Switching speed from sensorless to open loop [PU]	Electric angle
s2_ol2less_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 command current [PU]	
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	
s2_ad_point_a_adj_cnt	int16_t	Q0	-	Adjustment for A/D delay counts for A point	
s2_ad_point_b_adj_cnt	int16_t	Q0	-	Adjustment for A/D delay counts for B point	
u1_sal_angle_current	uint8_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	
u1_sal_polarity_current	uint8_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	
u1_non_sal_ref_i	uint8_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_speed_th_rad	int16_t	Q0	-	Field-Weakening Control speed threshold [rpm]	
s2_v_mag_th	int16_t	Q0	-	Maximum output voltage [V]	
s2_delta_id	int16_t	Q0	-	Id change amount of field-weakening Control [A]	
s2_speed_limit_rate	int16_t	Q0	-	Damping control speed limit	
st_motor	st_mtr_parameter_t	-	-	Structure for motor parameter	Structure
st_ctrl_params	st_mtr_design_parameter_t	-	-	Structure for PI control	

Table 3.44 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	
u1_current_event	uint8_t	Q0	-	Execution event	

表 3.45 “r_mtr_est_phase_err.h / 構造体：”st_mtr_est_phe_t” 変数一覧

Variable	Type	Qn	PU	Content	Remarks
s2_ed	int16_t	Q13	Voltage	d-axis induced voltage	
s2_eq	int16_t	Q13	Voltage	q-axis induced voltage	
s2_e	int16_t	Q13	Voltage	Induced voltage	
s2_phase_err_rad	int16_t	Q12	Angle	Phase error	
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	Q13	1/Back-EMF constant	Reciprocal of Back-EMF constant	

Table 3.46 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	Current: Q13 Speed: Q14	-	LPF input gain	
s2_out_k	int16	Current: Q13 Speed: Q14	-	LPF previous gain	
s2_pre_out	int16	Current: Q13 Speed: Q14	Current: current Speed: angular frequency	Previous output value	

Table 3.47 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: Q14 PLL: Q17	-	Proportional gain	
s2_kidt	int16_t	Current: Q20 Speed: Q21 PLL: Q22	-	Integral gain x control period	
s2_intg	int16_t	Current: Q17 Speed: Q13 PLL: Q14	Current: Resistance Speed: Current/angular frequency PLL: Angular frequency/angle	Integral term	
s2_ilimit	int16_t	Current: Q17 Speed: Q13 PLL: Q14	Current: Resistance Speed: Current/angular frequency PLL: Angular frequency/angle	Integral limit (up/down symmetry)	

Table 3.48 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_acr_t”

Variable	Type	Qn	PU	Content	Remarks
s2_ctrl_period	int16_t	Q18	Time	Current control cycle	
s2_pre_ref_vd	int16_t	Q13	Voltage	Previous d-axis output voltage command value	
s2_pre_ref_vq	int16_t	Q13	Voltage	Previous q-axis output voltage command value	
s2_ref_id	int16_t	Q13	Current	d-axis current command	
s2_ref_iq	int16_t	Q13	Current	q-axis current command	
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]	
s2_iq_lpf	int16_t	Q14	Current	q-axis current LPF value	
s2_id_lpf	int16_t	Q14	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 3.49 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 3.50 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	Q12	Voltage	Voltage error	
s2_deadtime_limit_current	int16_t	Q12	Current	Current limit	
s2_delta_v_uvw[3]	int16_t	Q12	Voltage	Three-phase voltage compensation value	

Table 3.51 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	Command rotational speed	
s2_ref_speed_rad_ctrl;	int16_t	Q14	Angular frequency	Command speed control value	
s2_speed_rad_origin;	int16_t	Q14	Angular frequency	Speed (no filter)	
s2_speed_rad;	int16_t	Q14	Angular frequency	Speed (with filter)	
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_less2ol_speed_rad;	int16_t	Q14	Angular frequency	Switching speed from sensorless to open loop	
s2_ol2less_speed_rad;	int16_t	Q14	Angular frequency	Switching speed from open loop to sensorless	
st_pi;	st_mtr_pi_t	-	-	Speed PI structure	Structure
st_lpf;	st_mtr_lpf1_t	-	-	Speed LPF structure	

Table 3.52 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_mod_t”

Variable	Type	Qn	PU	Content	Remarks
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	

Table 3.53 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_tscs_t”

Variable	Type	Qn	PU	Content	Remarks
u1_offset_idc_adc	uint8_t	Q0	-	DC link current offset value	
s2_duty_u	int16_t	Q0	-	U phase duty (PWM register setting)	
s2_duty_v	int16_t	Q0	-	V phase duty (PWM register setting)	
s2_duty_w	int16_t	Q0	-	W phase duty (PWM register setting)	
s2_offset_iu	int16_t	Q13	Current	U phase current offset value	
s2_offset_iw	int16_t	Q13	Current	W phase current offset value	
u4_offset_iu_sum	uint32_t	Q13	Current	U phase current offset value integral value	
u4_offset_iw_sum	uint32_t	Q13	Current	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	
u2_crnt_ad[2]	uint16_t	Q13	-	UW phase current A/D conversion value	

Table 3.54 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_sscs_t”

Variable	Type	Qn	PU	Content	Remarks
u1_drv_pattern	uint8_t	Q0	-	Modulation rate magnitude relation pattern	
u1_offset_idc_adc	uint8_t	Q0	-	dc current offset	
u1_flag_overl_duty	uint8_t	Q0	-	Duty limit over flag	
s2_duty_max	uint16_t	Q0	-	Maximum duty value	
s2_duty_mid	uint16_t	Q0	-	Middle duty value	
s2_duty_min	uint16_t	Q0	-	Minimum duty value	
s2_duty_u	uint16_t	Q0	-	U-phase duty value	
s2_duty_v	uint16_t	Q0	-	V-phase duty value	
s2_duty_w	uint16_t	Q0	-	W-phase duty value	
s2_duty_max_adj	uint16_t	Q0	-	Maximum duty adjustment value	
s2_duty_mid_adj	uint16_t	Q0	-	Middle duty adjustment value	
s2_duty_min_adj	uint16_t	Q0	-	Minimum duty adjustment value	
s2_duty_max_adj_comp	uint16_t	Q0	-	Maximum duty adjustment compensation value	
s2_duty_mid_adj_comp	uint16_t	Q0	-	Middle duty adjustment compensation value	
s2_duty_min_adj_comp	uint16_t	Q0	-	Minimum duty adjustment compensation value	
s2_duty_u_adj	uint16_t	Q0	-	U-phase duty adjustment value	
s2_duty_v_adj	uint16_t	Q0	-	V-phase duty adjustment value	
s2_duty_w_adj	uint16_t	Q0	-	W-phase duty adjustment value	
s2_duty_u_adjc	uint16_t	Q0	-	U-phase duty adjustment compensation value	
s2_duty_v_adjc	uint16_t	Q0	-	V-phase duty adjustment compensation value	
s2_duty_w_adjc	uint16_t	Q0	-	W-phase duty adjustment compensation value	
s2_duty_diff_limit	uint16_t	Q0	-	Minimum duty difference	
s2_ad_point_a_cnt	uint16_t	Q0	-	A/D conversion point A timer count value	
s2_ad_point_b_cnt	uint16_t	Q0	-	A/D conversion point B timer count value	
s2_ad_point_a_adj_cnt	uint16_t	Q0	-	A/D conversion point A count adjustment value	
s2_ad_point_b_adj_cnt	uint16_t	Q0	-	A/D conversion point B count adjustment value	
s2_ad_ss_a	uint16_t	Q0	-	Point A A/D conversion result	
s2_ad_ss_b	uint16_t	Q0	-	Point A A/D conversion result	
s2_offset_ia	uint16_t	Q13	Current	Point A Current offset value	
s2_offset_ib	uint16_t	Q13	Current	Point B current offset value	
s4_offset_ia_sum	int32_t	Q13	Current	Point A Current offset value Integral value	
s4_offset_ia_sum	int32_t	Q13	Current	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	
u2_crnt_ad[2]	uint16_t	Q13	-	AB point current AD conversion value	

Table 3.55 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_foc_t” (1/2)

Variable	Type	Qn	PU	Content	Remarks
u2_run_mode	uint16_t	-	-	Operating modes	0x00: Init mode 0x01: Boot mode 0x02: Drive mode 0x03: Analysis mode 0x04: Tune mode
u2_ctrl_conf	uint16_t	-	-	Control inputs	0x01: Current control 0x02: Speed control 0x04: Position control 0x08: Torque control 0x10: Voltage control
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: Induced voltage timeout error 0x0020: Hall pattern error 0x0040: Induced voltage pattern error 0x0080: Undervoltage error 0x0100001: Overcurrent error (software) 0xFFFF: Undefined error
s1_direction	int8_t	-	-	Current direction of rotation	1: CW -1: CCW
u1_flag_charge_cap	uint8_t	-	-	Current offset value calculation flag	0: Execute offset calculation process 1: Offset calculation process completed
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 increase 1: d-axis current constant 2: d-axis current decrease 3: d-axis current 0
u1_state_ref_iq	uint8_t	-	-	q-axis current command value generation status	0: q-axis current 0 1: Speed PI output 2: q-axis current decrease
u1_state_ref_speed	uint8_t	-	-	Speed command value generation status	0: Speed 0 1: Speed change
u1_flag_down_to_ol	uint8_t	-	-	Open loop transition flags	0: No transition 1: Execute transition
s2_vdc_ad	int16_t	Q13	Voltage	Power source voltage	
s2_limit_over_current	int16_t	Q13	Current	Overcurrent limit value	
s2_limit_over_voltage	int16_t	Q13	Voltage	Overvoltage limit value	
s2_limit_under_voltage	int16_t	Q13	Voltage	Undervoltage limit value	

Table 3.56 List of Variables in “r_mtr_foc_less_speed.h” / Structure: “st_mtr_foc_t” (2/2)

Variable	Type	Qn	PU	Content	Remarks
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_stm	st_mtr_statemachine_t	-	-	Structure for state machine	
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_t	-	-	Structure for three-phase current detection	
st_sscs	st_mtr_sscs_t	-	-	Structure for 1-shunt resistor current detection	
st_acr	st_mtr_acr_t	-	-	ACR structure	Current PI control
st_asr	st_mtr_asr_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_ipd	st_mtr_ipd_t	-	-	Structure for initial position detection	
st_fw	st_mtr_fw_t	-	-	Field-Weakening Control structure	
st_damp	st_mtr_damp_t	-	-	Damping control structure	
s2_damp_speed	int16_t	Q14	speed	Damping speed output	
s2_damp_ref_speed_rad_ctrl	int16_t	Q14	speed	Damping reference speed	

Table 3.57 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	
u1_sal_angle_current	uint8_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	
u1_sal_polarity_current	uint8_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	
u1_non_sal_current	uint8_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 3.58 List of Variables in “r_mtr_fw_ctrl.h” / Structure: “st_mtr_fw_t”

Variable	Type	Qn	PU	Content	Remarks
s2_speed_th_rad	int16_t	Q14	speed	Field-Weakening Control speed threshold [rpm]	
s2_v_mag	int16_t	Q13	voltage	Maximum output voltage [V]	
s2_v_mag_th	int16_t	Q13	voltage	Output voltage threshold [V]	
s2_delta_id	int16_t	Q13	current	Id change amount of field-weakening Control [A]	
s2_limit_id	int16_t	Q13	current	d-axis current limit	
s2_output_id	int16_t	Q13	current	d-axis current output	
s2_i_max	int16_t	Q13	current	Maximum current	

Table 3.59 List of Variables in “r_mtr_damp_ctrl.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	Q15	-	HPF gain Q value	

Table 3.60 List of Variables in “r_mtr_damp_ctrl.h” / Structure: “st_mtr_damp_t”

Variable	Type	Qn	PU	Content	Remarks
s2_k	int16_t	Q12	Speed/Voltage	Damping control gain	
s2_speed_limit_rate	int16_t	Q14	Speed	Speed limit	
u1_q_damp_speed_calc	int16_t	Q14	Speed	Damping control speed Q value	
u1_q_damp_speed_limit_calc	int16_t	Q14	Speed	Damping control speed limit Q value	
st_hpf	st_mtr_hpf1_t	-	-	LPF 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 3.61 List of Macro Definitions in “r_mtr_config.h”

Macro	Definition value	Description	Remarks
IP_GB01_x4	-	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	0 : 1	Current detection method(Note)	Default setting 0
USE_DEADTIME_COMP	0:1	Select deadtime compensation process	Default setting 1
USE_SPEED_LPF	0:1	Select speed LPF	Default setting 1
USE_CURRENT_LPF_IQ	0:1	Select q-axis current PPF	Default setting 0
USE_CURRENT_LPF_ID	0:1	Select d-axis current PPF	Default setting 0
USE_IPD	0:1	Select IPD	Default setting 0
USE_FIELD_WEAKENING	0 : 1	Select Field-Weakening Control	Default setting 1
USE_OPENLOOP_DAMPING	0 : 1	Select Openloop damping Control	Default setting 1
MOD_3PH_SPWM	0	Sine wave modulation	
MOD_3PH_TOW	1	Third harmonic calculation	
MOD_METHOD	0:1	Modulation method	Default setting 1

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

Table 3.62 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]	
MP_Q_INDUCTANCE	0.004315f	q-axis inductance [H]	
MP_BEMF_CONSTANT	0.02144f	Back-EMF constant [V･s/m]	
MP_ROTOR_INERTIA	0.000002050f	Inertia [kgm^2]	
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	0:1	Selection of motor wiring connection	Star / Delta

Table 3.63 List of Macro Definitions in "control_parameter.h"

Macro	Definition value	Description	Remarks
CP_ACR_NF_HZ	300.0f	Current PI control natural frequency [Hz]	
CP_ASR_NF_HZ	10.0f	Speed PI control natural frequency [Hz]	
CP_PLL_NF_HZ	40.0f	PLL control natural frequency [Hz]	
CP_ASR_LPF_COF_HZ	40.0f	cutoff frequency [Hz] of speed LPF	
CP_ACR_LPF_COF_HZ	1200.0f	cutoff frequency [Hz] of current LPF	
CP_MAX_SPEED_RPM	3975	Maximum speed (mechanical angle) [rpm]	
CP_SPEED_LIMIT_RPM	5300	Limit of speed (mechanical angle) [rpm]	
CP_OC_LIMIT	1.47		
CP_OL_REF_ID	0.42	d-axis current command value [A]	
CP_INIT_ASR_INTEG	0.05f	q-axis current PI integral term PI initial value [A]	
CP_ASR_KI_ADJ	0.2f	Adjustment parameters for the velocity PI integral term	
CP_LAMP_LIMIT_CURRENT	0.01f	Limit value for current rise [PU/ms]	
CP_OL2LESS_SPEED_RPM	795	Switching speed from sensorless to open loop (mechanical angle) [rpm]	
CP_LESS2OL_SPEED_RPM	530	Switching speed from open loop to sensorless (mechanical angle) [rpm]	
CP_LAMP_LIMIT_SPEED_RPM	1.677845f	Limit of acceleration [rpm/ms]	
CP_RAMP_SPEED_CNT_DECIMATION	0	Number of decimation out of acceleration limit	
CP_OFFSET_CALC_TIME	128	Current offset value calculation time [ms]	
CP_AD_POINT_A_ADJ_CNT	0	Adjustment for A/D delay counts for A point	
CP_AD_POINT_B_ADJ_CNT	0	Adjustment for A/D delay counts for B point	
CP_SAL_ANGLE_CURRENT	0.3f	Threshold current of angle detection for salient rotor	【IPD】
CP_ SAL _ANGLE_TRX_THRESHOLD	1728	TRX count value differential of angle detection for salient rotor	
CP_ SAL _ANGLE_DISCHARGE	613	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	884	TRX count value differential of polarity detection for salient rotor	
CP_ SAL _POLARITY_DISCHARGE	696	Discharge period of polarity detection for salient rotor	
CP_NON_SAL_CURRENT	0.42f	Threshold current of angle detection for non-salient rotor	
CP_NON_SAL_TRX_ THRESHOLD	1767	TRX count value differential of angle detection for non-salient rotor	
CP_NON_SAL_DISCHARGE	696	Discharge period of angle detection for non-salient rotor	
CP_FW_SPEED_THRESHOLD	2650	Field-weaking control speed threshold [rpm]	
CP_FW_V_MAG_THRESHOLD	19.2f	Output voltage threshold [V]	
CP_FW_DELTA_ID	0.000104f	Id change amount of field-weakening Control [A]	
CP_DAMP_HPF_COF_HZ	5	HPF cutoff frequency for damping control	
CP_DAMP_ZETA	1	Damping control damping coefficient	
CP_DAMP_SPEED_LIMIT_RATE	0.2f	Damping control speed limit	

Table 3.64 List of Macro Definitions in “r_mtr_inverter_parameter.h”

Macro	Definition value	Description	Remarks
IP_DEADTIME	1.0f	Deadtime	
IP_CURRENT_RANGE	12.5f	Current scaling range [A]	
IP_VDC_RANGE	111.0f	Voltage scaling range [V]	
IP_INPUT_V	24.0f	Input voltage [V]	
IP_CURRENT_LIMIT	4.5f	Current limit value [A]	
IP_OVERVOLTAGE_LIMIT	28.0f	Overvoltage limit [V]	
IP_UNDERVOLTAGE_LIMIT	12.0f	Undervoltage limit [V]	
IP_DC_SHUNT_RESISTANCE	0.1f	DC Link Shunt Resistance [ohm]	
IP_DC_AMPLIFICATION_GAIN	4	DC Link Current Amplification Gain	
IP_BSC_CHARGE_TIME	100	Period of charging bootstrap capacitor	

Table 3.65 List of Macro Definitions “r_mtr_scaling_parameter.h”

Macro	Definition value	Description	Remarks
FP_SF_VOLTAGE	37	Voltage PU conversion value (((IP_VDC_RANGE/1023)*PU_SF_VOLTAGE) * (1<<MTR_Q_VOLTAGE))	
FP_SF_CURRENT	238	Current PU conversion value (((IP_CURRENT_RANGE/1023)*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 *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 [PU/A]	
PU_SF_VOLTAGE	1.0f / PU_BASE_VOLTAGE_V	Voltage scale [PU/V]	
PU_SF_AFREQ	1.0f / PU_BASE_FREQ_Hz	Angular frequency scale [PU/(rad/s)]	
PU_SF_ANGLE	1.0f / PU_BASE_ANGLE_Rad	Angle scale [PU/rad]	
PU_SF_TIME	PU_SF_ANGLE / PU_SF_AFREQ	Time scale [PU/s]	
PU_SF_RES	PU_SF_VOLTAGE / PU_SF_CURRENT	Resistance scale [PU/ohm]	
PU_SF_IND	PU_SF_RES / PU_SF_AFREQ	Inductance scale [PU/H]	
PU_SF_BEMF_CONST	PU_SF_VOLTAGE / PU_SF_AFREQ	Back-EMF constant scale [PU/Wb]	
PU_SF_INERTIA	PU_SF_BEMF_CONST * PU_SF_CURRENT / (MP_POLE_PAIRS * MP_POLE_PAIRS * PU_SF_AFREQ * PU_SF_AFREQ)	Inertia scale [PU/(rad/kgm^2)]	
PU_SF_RPM_RAD	1.0f / CP_MAX_SPEED_RPM	Scale of conversion from [rpm] to [rad/s]	
PU_SF_RAD_RPM	CP_MAX_SPEED_RPM	Scale of conversion from [rad/s] to [rpm]	
PU_SF_ACR_KP	PU_SF_RES	Current PI proportional gain scale	
PU_SF_ACR_KIDT	PU_SF_RES	Current PI integral gain scale	
PU_SF_ASR_KP	PU_SF_CURRENT / PU_SF_AFREQ	Speed PI proportional gain scale	
PU_SF_ASR_KIDT	PU_SF_CURRENT / PU_SF_AFREQ	Speed PI integral gain scale	
PU_SF_PLL_KP	PU_SF_AFREQ / PU_SF_ANGLE	PLL proportional gain scale	
PU_SF_PLL_KIDT	PU_SF_AFREQ / PU_SF_ANGLE	PLL integral gain scale	
MTR_Q_ANGLE	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 Back-EMF constant	
MTR_Q_INERTIA	10	Q-format of inertia	
MTR_Q_RECIV	13	Q-format of inverse voltage	
MTR_Q_RECIM	14	Q-format of reciprocal of Back-EMF constant	

Table 3.66 List of Macro Definitions "r_mtr_scaling_parameter.h"

Macro	Definition value	Description	Remarks
MTR_Q_ACR_KP	17	Q-format of speed PI proportional gain	
MTR_Q_ACR_KIDT	20	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	19	Q-format of current PI integral gain * control period	
MTR_Q_PLL_KP	15	Q-format of PLL proportional gain	
MTR_Q_PLL_KIDT	22	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_DAMP_K	12	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_DIV_DSP	16	Q-format of DSP function division	
MTR_Q_SIN_COS_DSP	14	Q-format of trinodal function of DSP function	

Table 3.67 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 3.68 List of Macro Definitions in "ICS_define.h"

Macro	Definition value	Description	Remarks
RL78	-	CPU definition	

Table 3.69 List of Macro Definitions in "r_mtr_ics.h"

Macro	Definition value	Description	Remarks
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_BRR	15	ICS bit rate register selection	
ICS_MODE	0	ICS interrupt mode setting	

Table 3.70 List of Macro Definitions in “r_mtr_ctrl_rl78g1f.h” [1/2]

Macro	Definition value	Description	Remarks
USE_PWMOPA	MTR_SET	Select of overcurrent use by PWMOPA	
MTR_INT_DECIMATION	1	Interrupt processing carrier pixel skipping	
MTR_PWM_TIMER_FREQ	64.0f	PWM timer frequency [kHz]	
MTR_INTVAL_TIMER_FREQ	32.0f	Interval timer frequency [kHz]	
MTR_CARRIER_FREQ	20.0f	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_CURRENT_ADCONV_TIME	6.5f	Time [µs] taken for two-phase current A/D conversion	[Used in 3-shunt mode]
MTR_AD_TIME_ADJUST	0.2f	AD timing adjustment	
MTR_AD_TIME_CNT	(uint16)(MTR_PWM_TIMER_FREQ * MTR_PWM_LA_MIN_ONTIME)	A/D conversion time counter value g	
MTR_AD_START_SHIFT_CNT	MTR_AD_TIME_CNT - (uint16_t)(MTR_PWM_TIMER_FREQ * MTR_AD_TIME_ADJUST)	Shift count value for A/D conversion	
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-(MTR_AD_TIME_CNT+MTR_DEADTIME_CNT)) * 0.5f + MTR_AD_TIME_CNT+MTR_DEADTIME_CNT)	PWM timer center amplitude	
MTR_AD_MINIMUM_TIME	220	A/D conversion time count value	[Used in 1-shunt mode]
MTR_HALF_DEADTIME_CNT	(uint16_t)(MTR_DEADTIME_CNT*0.5f)	half of the dead time setting value	
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_OFFSET_CALC_ST_WAIT_CNT	500	Stable waiting time before acquiring current offset	
MTR_DEADTIME_RATIO	(MTR_DEADTIME/1000000) * (MTR_CARRIER_FREQ * 1000)/2	Deadtime compensation coefficient	
MTR_DEADTIME_CURRENT_LIMIT	MP_RATED_CRRENT * 0.1f	Current limit value	
MTR_CTRL_PERIOD	(MTR_INT_DECIMATION + 1) / (MTR_CARRIER_FREQ * 1000)	Current control cycle	
MTR_SPEED_CTRL_PERIOD	0.001f	Speed control cycle	
MTR_PORT_UP	P1_bit.no5	U phase (positive phase) voltage output port	
MTR_PORT_UN	P1_bit.no4	U phase (negative phase) voltage output port	
MTR_PORT_VP	P1_bit.no3	V phase (positive phase) voltage output port	
MTR_PORT_VN	P1_bit.no1	V phase (negative phase) voltage output port	
MTR_PORT_WP	P1_bit.no2	W phase (positive phase) voltage output port	
MTR_PORT_WN	P1_bit.no0	W phase (negative phase) voltage output port	
MTR_PORT_ENC_A	P0_bit.no0	Encoder A phase input port	
MTR_PORT_ENC_B	P0_bit.no1	Encoder B phase input port	
MTR_PORT_ENC_Z	P5_bit.no0	Encoder Z phase input port	

Table 3.71 List of Macro Definitions in “r_mtr_ctrl_rl78g1f.h” [2/2]

Macro	Definition value	Description	Remarks
MTR_PORT_HALL_U	P3_bit.no0	U phase Hall effect sensor input port	
MTR_PORT_HALL_V	P7_bit.no5	V phase Hall effect sensor input port	
MTR_PORT_HALL_W	P3_bit.no1	W phase Hall effect sensor input port	
MTR_PORT_ANA0	P14_bit.no7	Analog Input 0	
MTR_PORT_ANA1	P12_bit.no0	Analog Input 1	
MTR_PORT_LED1	P6_bit.no3	LED1 output port	
MTR_PORT_LED2	P7_bit.no4	LED2 output port	
MTR_PORT_SW1	P12_bit.no4	SW1 input port	
MTR_PORT_SW2	P12_bit.no3	SW2 input port	
MTR_DUTY_U	TRDGRD0	Timer RD general register	
MTR_DUTY_V	TRDGRC1	Timer RD general register	
MTR_DUTY_W	TRDGRD1	Timer RD general register	
MTR_ADCCH_IU	4	A/D converter channel of U phase current	
MTR_ADCCH_IV	5	A/D converter channel of V phase current	
MTR_ADCCH_IW	6	A/D converter channel of W phase current	
MTR_ADCCH_VDC	7	A/D converter channel of bus voltage	
MTR_ADCCH_VU	16	A/D converter channel of U phase voltage	
MTR_ADCCH_VV	0	A/D converter channel of V phase voltage	
MTR_ADCCH_VW	1	A/D converter channel of W phase voltage	
MTR_ADCCH_PGA	25	A/D converter channel of PGA	
MTR_ADC_DATA_SHIFT	6	A/D conversion value shift amount	
MTR_ADC_OFFSET	0x1FF	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_GET_PWM_COUNT	TRD1	PWM count value	
MTR_OC_DETECT_REF	(uint8_t)(2.0f+(4 * 0.1f*IP_CURRENT_LIMIT))*(255/5) [4 times] (uint8_t)((8 *IP_DC_SHUNT_RESISTANCE *IP_CURRENT_LIMIT)*(255/5)) [8 times]	Overcurrent limit reference value	
MTR_OC_DETECT_OFSET	(uint8_t)(2.02f*(255/5)) [4 times] (uint8_t)(3.98f*(255/5)) [8 times]	Overcurrent offset initial value	
MTR_ADC_MODE_CURRENT	0xA0	Current detection mode	[Used in 1-shunt mode]
MTR_ADC_MODE_VDC	0x21	Voltage detection mode	
MTR_DTC_BASE_ADDRESS	0xFE	DTC base address	
MTR_DTC_ADC_ADDRESS	0xFE0A	DTC activation factor Address of A/D conversion end	
MTR_DTC_ADC_CTRL_ADRS	0x48	Control data area address	
MTR_DTC_ADC_CTRL_BASE	0xFE00+MTR_DTC_ADC_CTRL_ADRS	Control database address	
MTR_DTC_ADC_MODE	0x48	DTC control register setting value	
MTR_DTC_ADC_TRSF_SIZE	1	DTC block size	
MTR_DTC_ADC_TRSF_COUNTS	2	DTC transfer count	
MTR_DTC_ADC_RELOAD_COUNTS	2	DTC transfer count reload	

Table 3.72 List of Macro Definitions in “r_mtr_common.h”

Macro	Definition value	Description	Remarks
MTR_TWOPI	2*3.14159265359f	2π	
MTR_SQRT_3	1.7320508f	$\sqrt{3}$	
MTR_CW	1	CW	
MTR_CCW	-1	CCW	
MTR_ON	0	ON	
MTR_OFF	1	OFF	
MTR_CLR	0	Flag clear	
MTR_SET	1	Flag set	

Table 3.73 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_VDC_SCALING	IP_VDC_RANGE / 1023.0f	Voltage scaling value	
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 * 4096)	Angle range 2 π	
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_CURRENT_SCALING	IP_CURRENT_RANGE / 1023.0f	Current scaling value	
MTR_OVERCURRENT_LIMIT	IP_CURRENT_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_LESS2OL_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/ Back-EMF constant [PU]	

Table 3.74 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 3.75 List of Macro Definitions in “r_mtr_foc_less_speed.h”

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	Hall timeout error	
MTR_ERROR_BEMF_TIMEOUT	0x0010	Induced voltage timeout error	
MTR_ERROR_HALL_PATTERN	0x0020	Hall pattern error	
MTR_ERROR_BEMF_PATTERN	0x0040	Induced voltage pattern error	
MTR_ERROR_UNDER_VOLTAGE	0x0080	Undervoltage error	
	0x0100		
MTR_ERROR_IPD_TRX_OVERFLOW	0x0200	TRX overflow error during initial position detection	
MTR_ERROR_UNKNOWN	0xff	Undefined error	
MTR_ID_ZERO_CONST	0	d-axis current 0 control	
MTR_ID_MANUAL	1	d-axis current manual control	
MTR_IQ_ZERO_CONST	0	q-axis current 0 control	
MTR_IQ_MANUAL	1	q-axis current manual control	
MTR_IQ_SPEED_PI_OUTPUT	2	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_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	

Table 3.76 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*256/5	Scaling factor for setting threshold current of CMP0	【IPD】
MTR_DETECT_ANGLE_MAX_CNT	20	Maximum number of times salient rotor angle detection is measured	
MTR_DETECT_ANGLE_PERCENTAGE	30	Percentage of TRX differential threshold of angle detection for salient rotor	
MTR_DETECT_POLARITY_MAX_CNT	20	Maximum number of times salient rotor polarity detection is measured	
MTR_DETECT_POLARITY_PERCENTAGE	30	Percentage of TRX differential threshold of polarity detection for salient rotor	
MTR_NON_SALIENT_MAX_CNT	20	Maximum number of times non-salient rotor angle detection is measured	
MTR_NON_SALIENT_PERCENTAGE	30	Percentage of TRX differential threshold of angle detection for non-salient rotor	
MTR_WAIT_FOR_CMP0_ENABLE	50	Waiting for CMP0 enabled	
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_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 1-shunt mode is composed of two cycle interrupts: a carrier cycle (50 µs) cycle interrupt and a 1-ms cycle interrupt. In 3-shunt mode, interrupt processing is composed of two cycle interrupts: a 100-µs cycle interrupt and a 1-ms cycle interrupt.

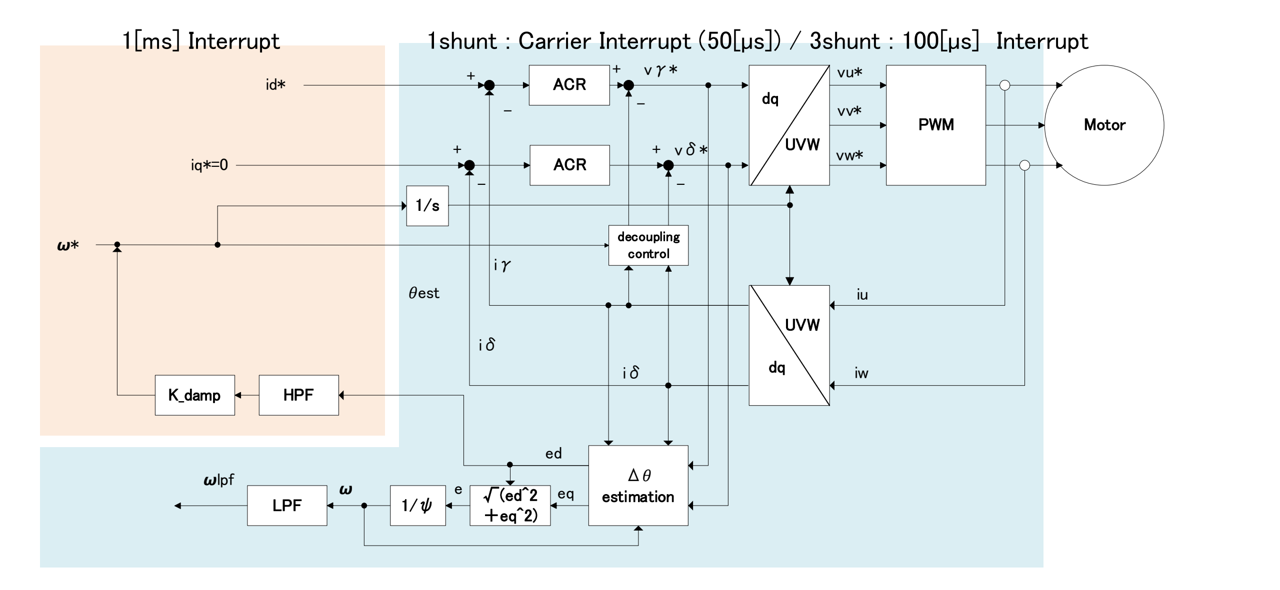


Figure 3-23 Interrupt processing inside control block (during open loop control)

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

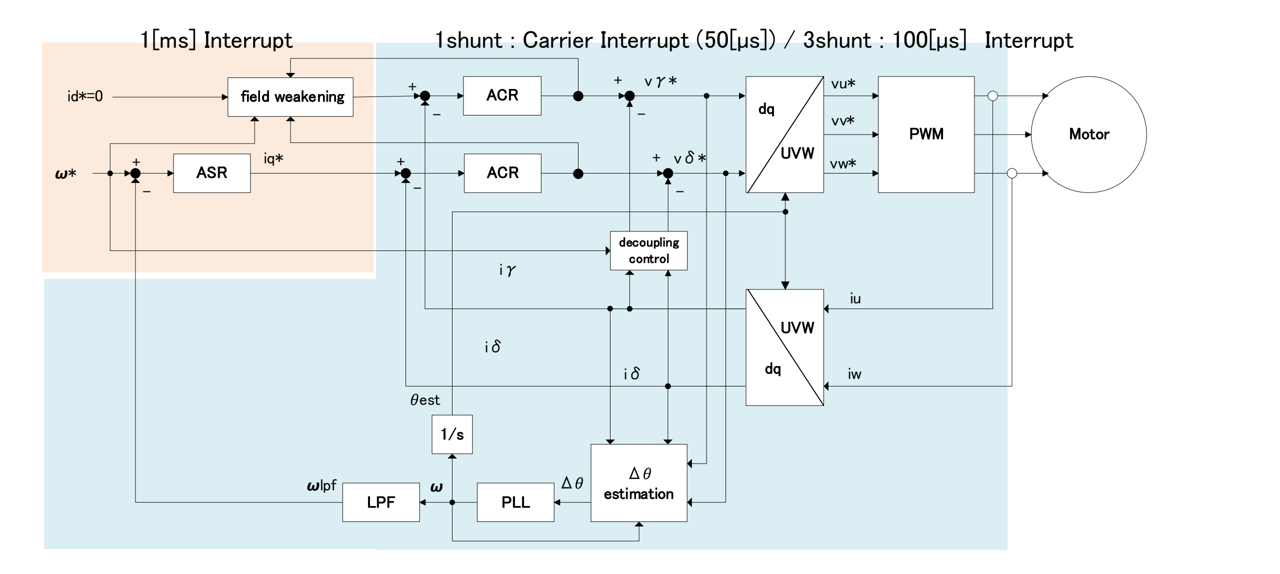


Figure 3-24 Interrupt processing inside control block (during closed loop control)

Here is an outline of driving force in a closed loop. The q-axis current command value is obtained as the output when the d-axis current is 0 and the estimated speed deviation, obtained from the speed command value and the phase error output from the phase error estimator, is input into a speed controller. The angle is updated by using angle information in which the estimated speed is integrated.

Processing inside the carrier cycle interrupt in 1-shunt mode is divided into two cycles because the processing share of the adjustment processing and interrupt processing is about 60% for the PWM duty for 1-shunt resistor current detection. The control cycle of the current control system inside the carrier interrupt has one 50-µs skip and thus is 100 µs. The 3-shunt mode interrupt cycle has the same 100-µs current control system control cycle.
Speed control system processing and dq-axis current command values are output during the 1-ms cycle interrupt. The control cycles for the speed control system are 1 ms.
Carrier cycle interrupt processing in 1-shunt mode is divided into two cycles, Interrupt timing 1 and 0, as shown in Fig. Figure 3-25. The PWM duty is updated every carrier cycle (50 µs).

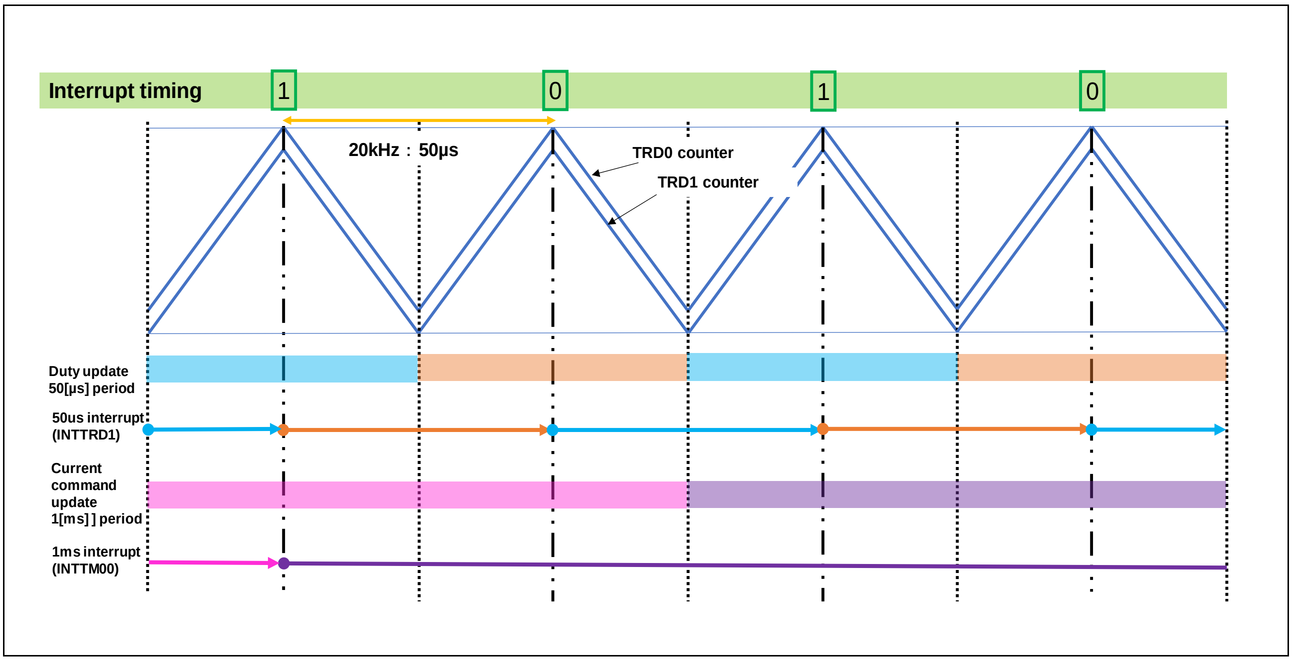


Figure 3-25 Image of interrupt generation and update of output value in 1-shunt mode

In 3-shunt mode, the PWM duty is updated every 100 µs by means of 100-µs interrupts.

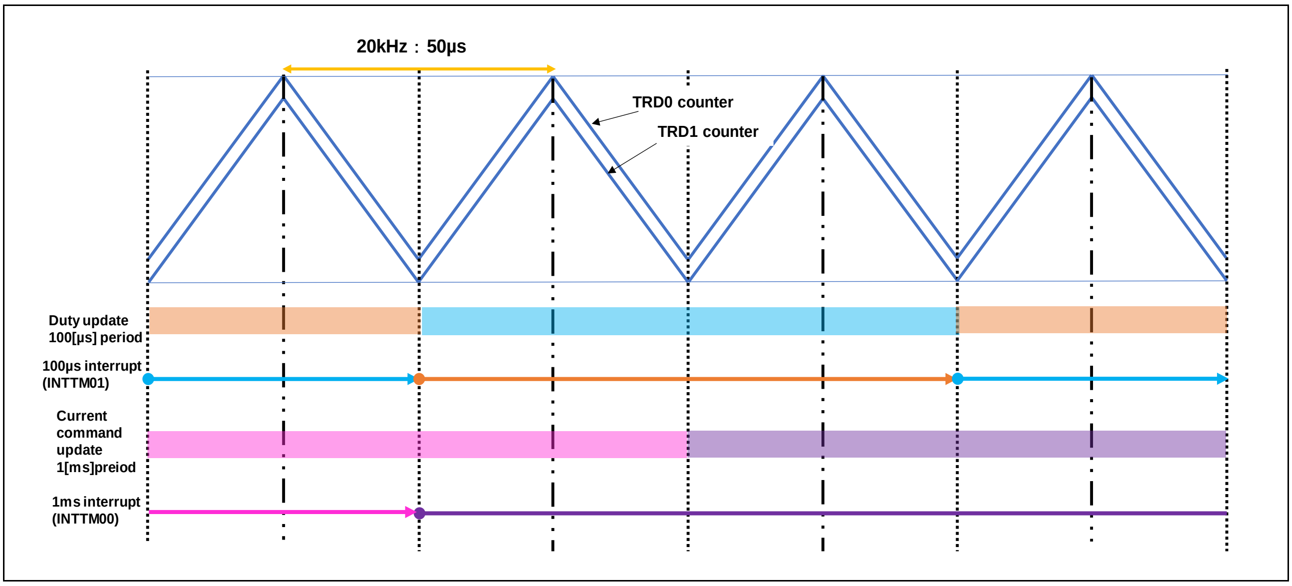


Figure 3-26 Image of interrupt generation and update of output value in 3-shunt mode

Also, in 1-shunt mode, interrupts are generated at the carrier peaks, and the setting values are reflected in the PWM at the troughs. In 3-shunt mode, interrupts are generated at the troughs and the setting values are reflected in the PWM at the troughs.

3.7 Control flows (flowcharts)

3.7.1 Main process

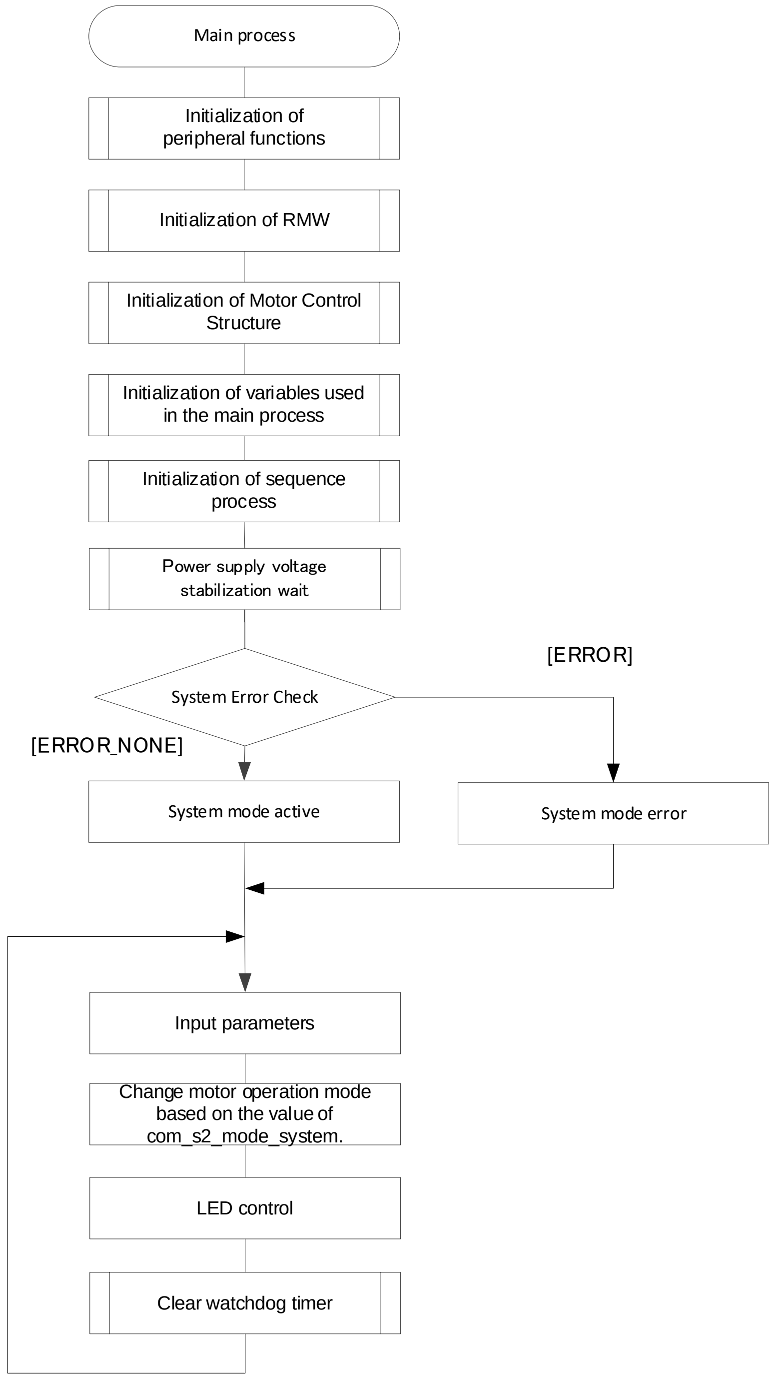
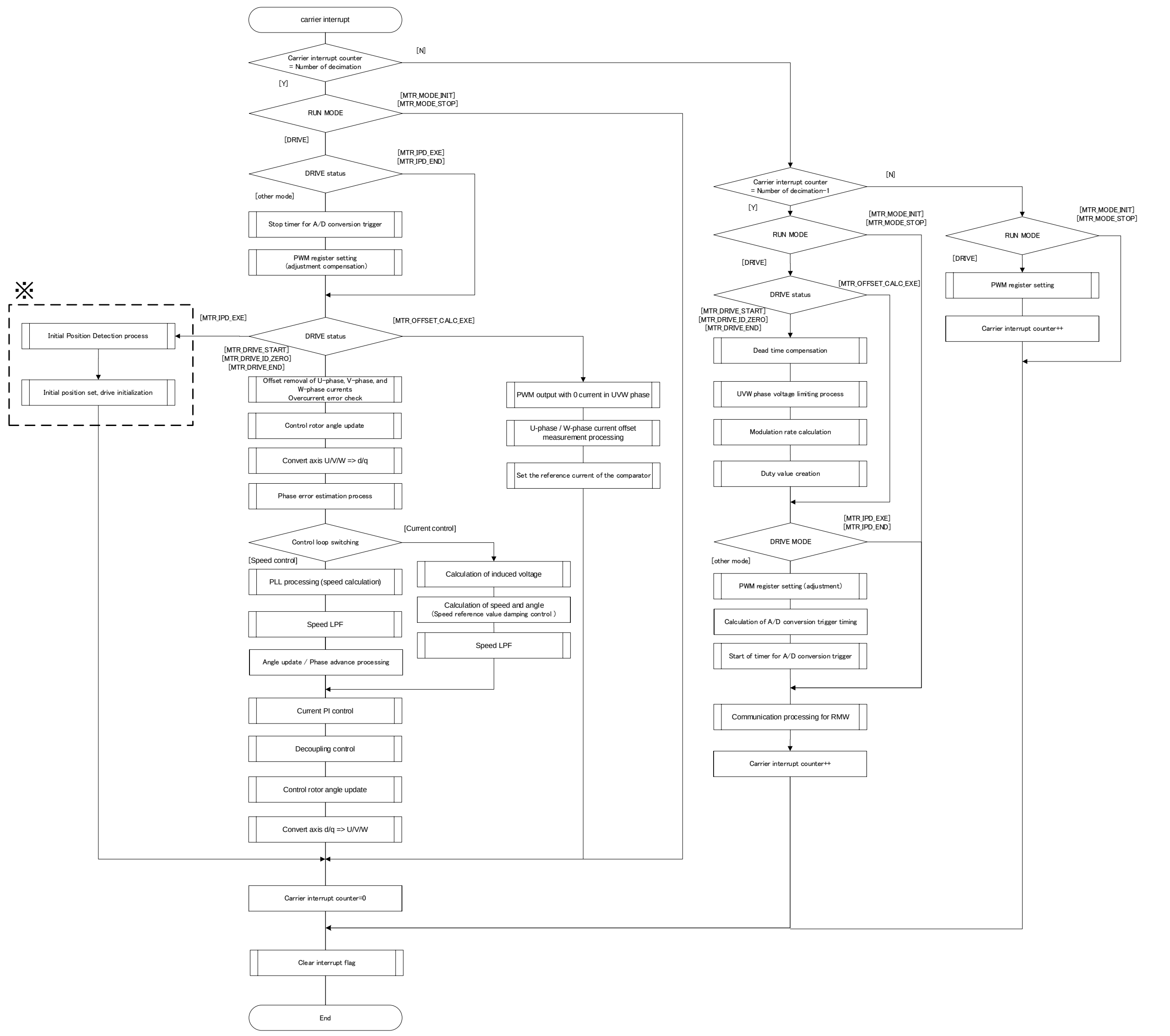


Figure 3-27 Main Process Flowchart

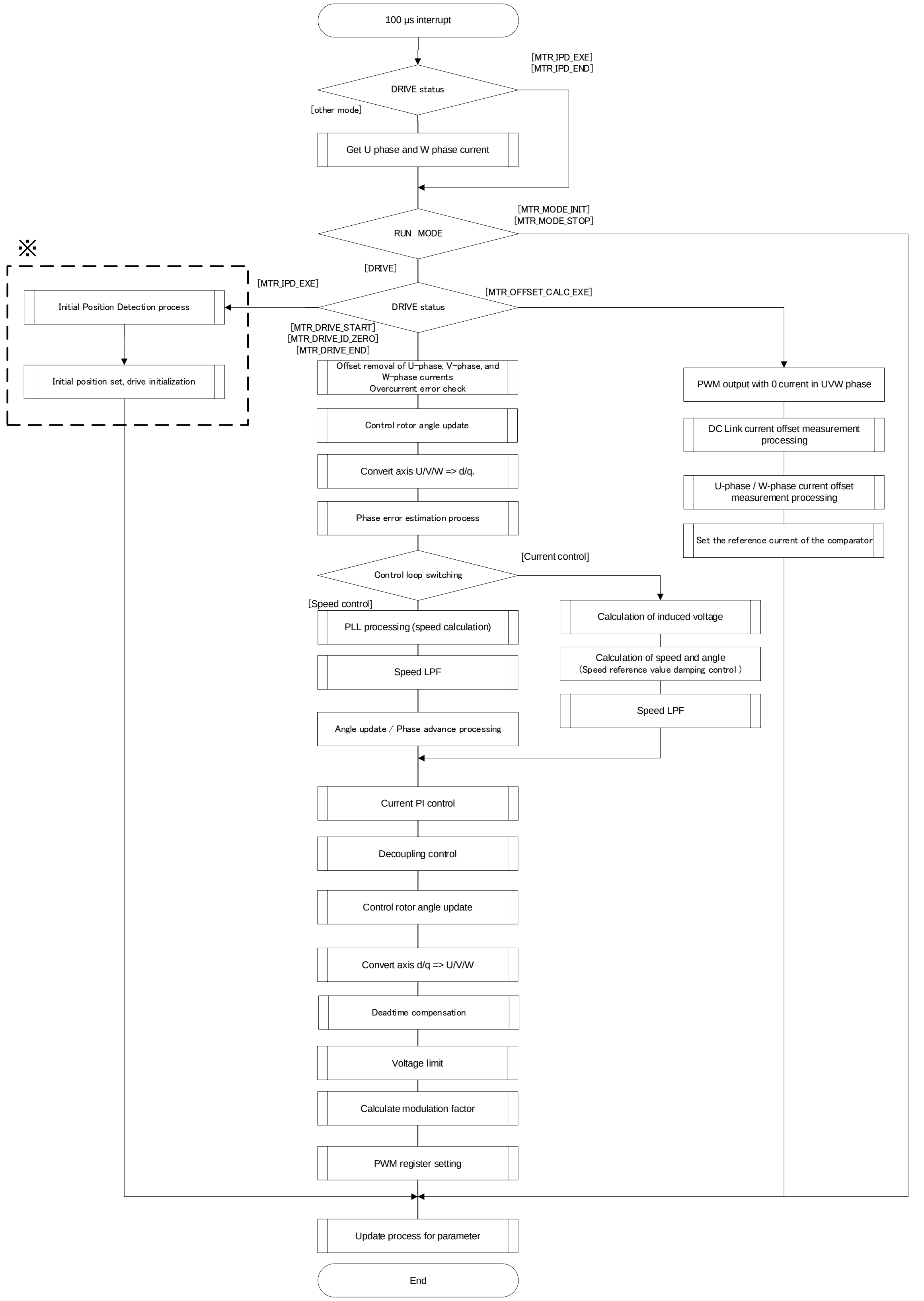
3.7.2 carrier interrupt handling (Used in 1-shunt mode)



※ If you do not use the initial position detection function, this process will not be performed.

Figure 3-28 Carrier Cycle Interrupt Handling Flowchart

3.7.3 100 us interrupt handling (Used in 3-shunt mode)



※ If you do not use the initial position detection function, this process will not be performed.

Figure 3-29 Carrier Cycle Interrupt Handling Flowchart

3.7.4 1-ms interrupt handling (Used in 1-shunt mode)

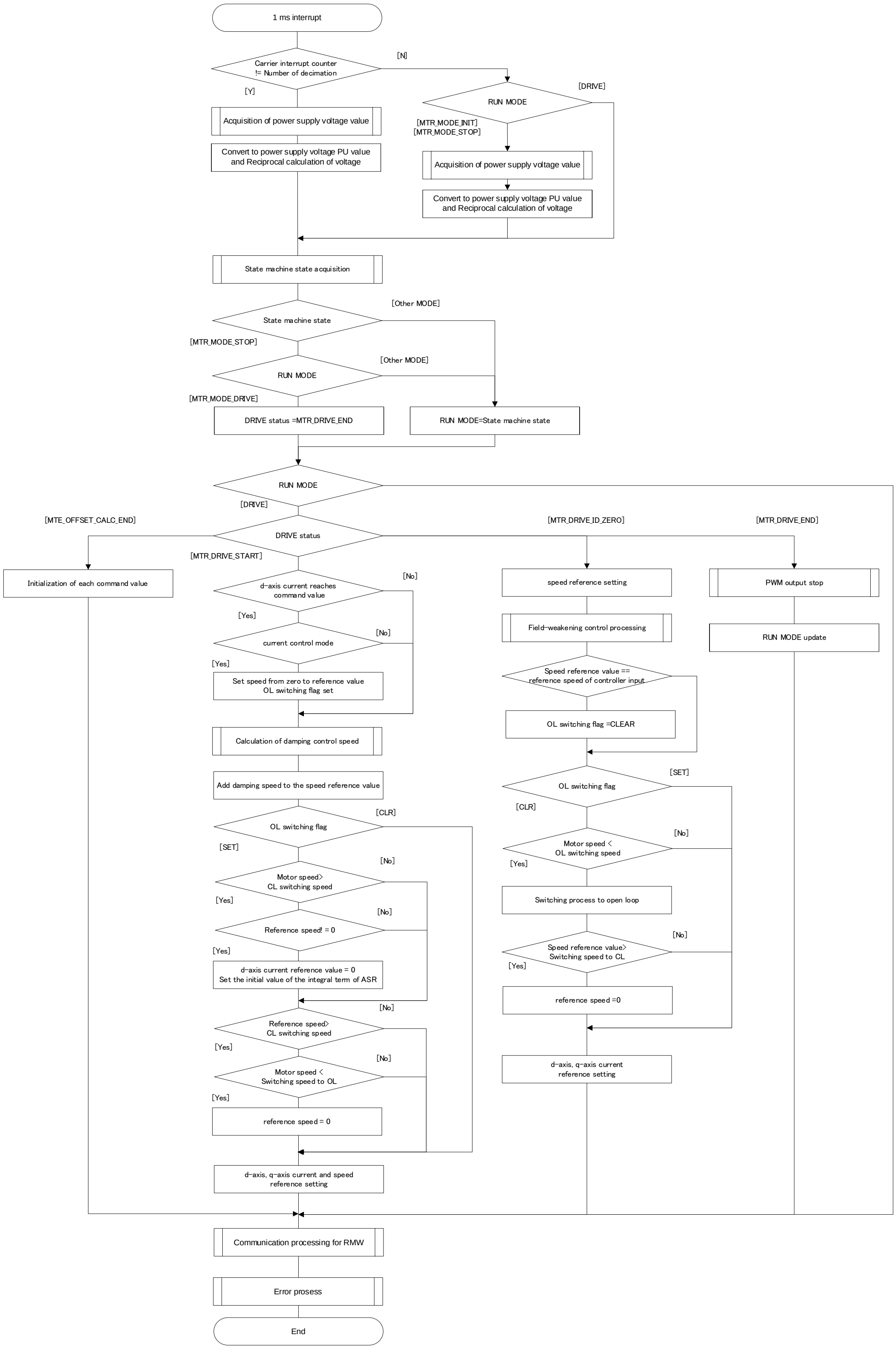


Figure 3-30 1-ms Interrupt Handling Flowchart

3.7.5 1-ms interrupt handling(Used in 3-shunt mode)

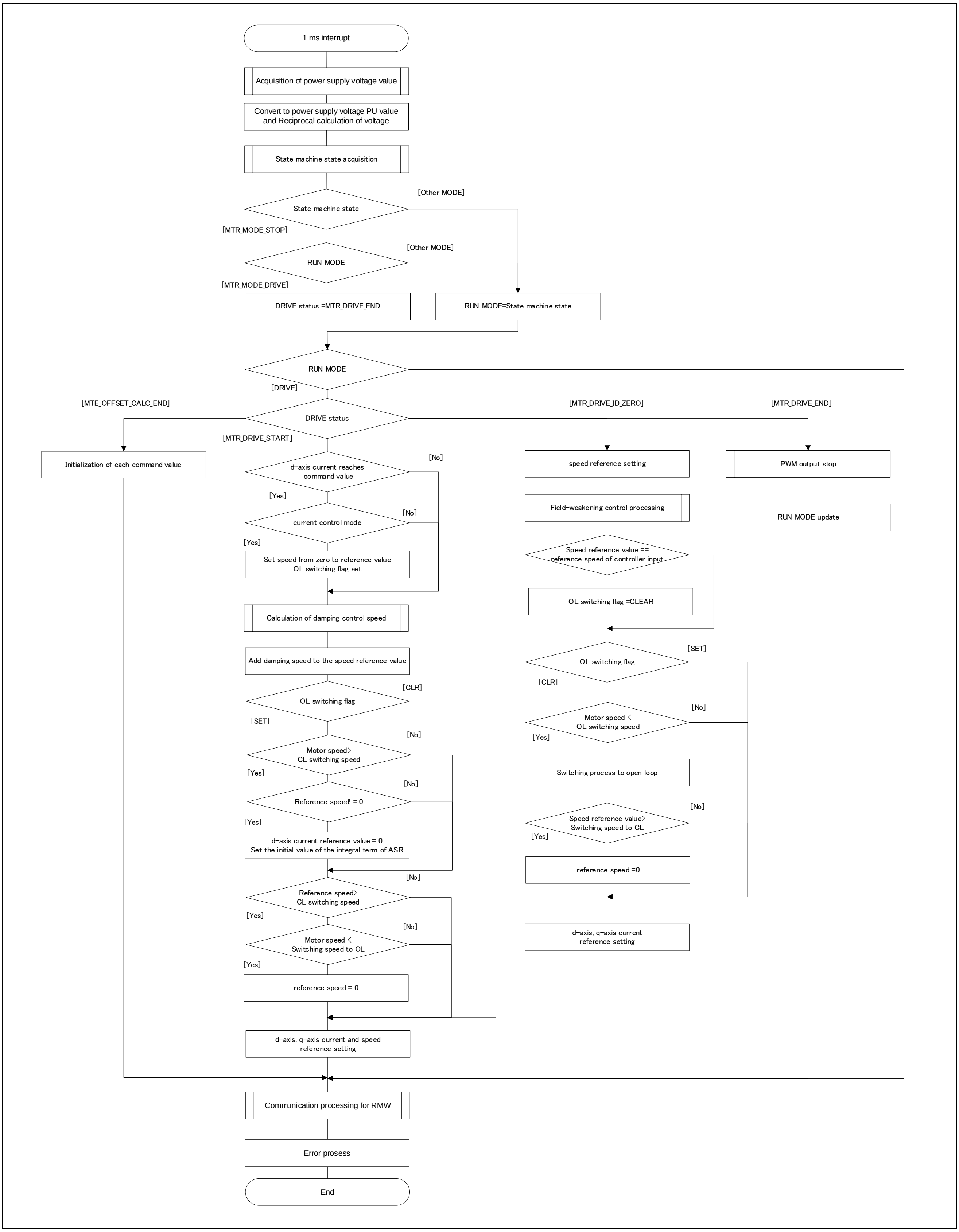


Figure 3-31 1-ms Interrupt Handling Flowchart

3.7.6 Comparator 0 interrupt handling

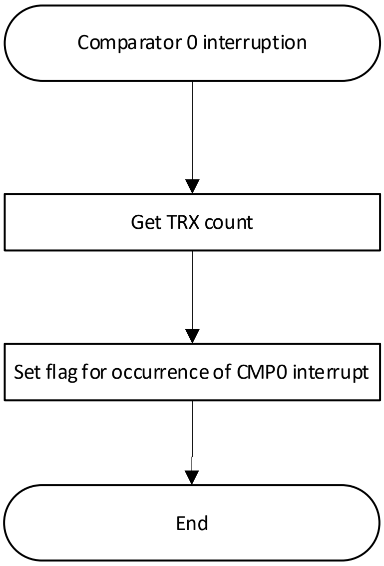


Figure 3-32 Comparator 0 interrupt handling

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 V 2.0 User's Manual' for usage and more details. You can find the 'Renesas Motor Workbench' on Renesas Electronics Corporation's website.

Main Window

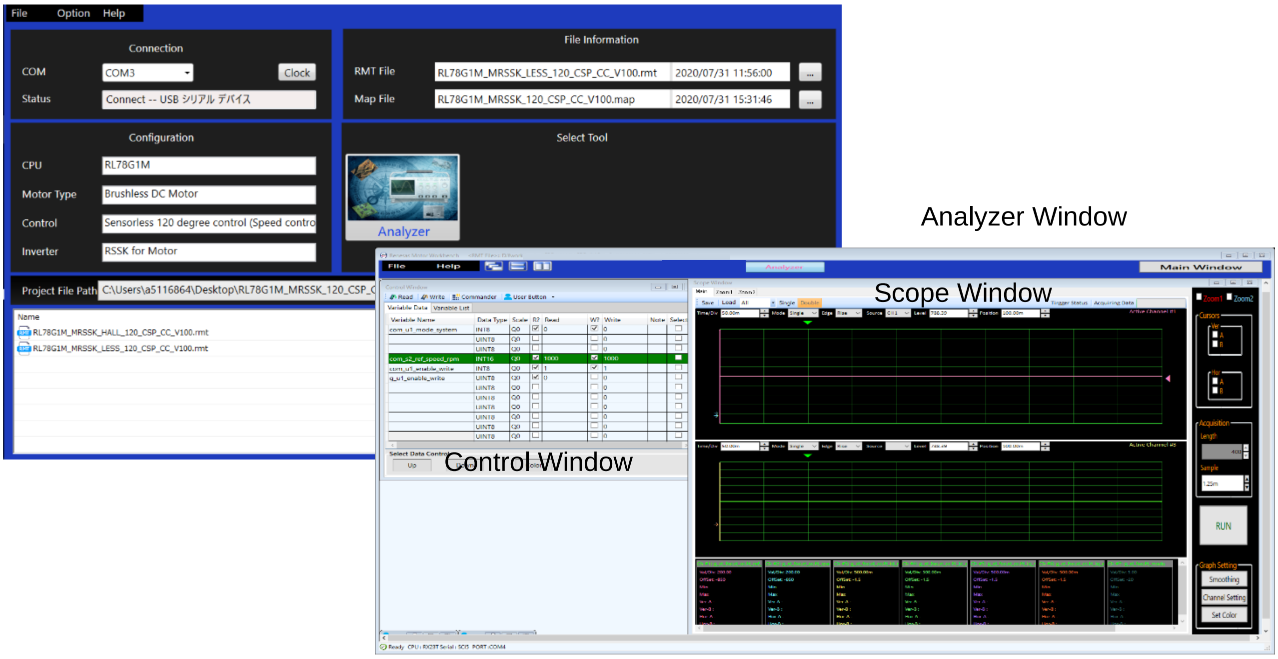
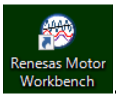


Figure 4-1 Screenshots of Renesas Motor Workbench

<u>How to use the motor control development support tool, Renesas Motor Workbench</u>

(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 for Motor RSSK.
(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(1/2)

Variable	Type	Content	Remarks ([ ]: reflection variable name)
com_u1_run_event (*)	uint8_t	Change run mode 0: Stop event 1: Drive event 2: Error event 3: Reset event	[g_u1_run_event]
com_s2_sw_userif (*)	int16_t	Management variable for UI 0: ICS UI (default) 1: Board UI	[g_s2_sw_userif]
com_u1_direction	uint8_t	Direction of rotation 0: CW 1: CCW	[gst_foc.st_asr.u1_ref_dir]
com_f4_mtr_r	float	Resistance [Ω]	[gst_foc.st_motor.s2_mtr_r]
com_f4_mtr_ld	float	d-axis inductance [H]	[gst_foc.st_motor.s2_mtr_ld]
com_f4_mtr_lq	float	q-axis inductance [H]	[gst_foc.st_motor.s2_mtr_lq]
com_f4_mtr_m	float	Induced voltage constant [Vs/rad]	[gst_foc.st_motor.s2_mtr_m]
com_f4_mtr_j	float	Inertia [kgm^2]	[gst_foc.st_motor.s2_mtr_j]
com_u2_mtr_pp	uint16_t	Number of pole pairs	[gst_foc.st_motor.u2_mtr_pp]
com_u2_offset_calc_cnt	uint16_t	Current offset detection time	[gst_foc.st_sscs.u2_offset_calc_time]
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_foc.st_acr.st_pi_id.s2_kp] [gst_foc.st_acr.st_pi_id.s2_kidt] [gst_foc.st_acr.st_pi_iq.s2_kp] [gst_foc.st_acr.st_pi_iq.s2_kidt]
com_f4_asr_nf_hz	float	Speed PI control natural frequency [Hz]	[gst_foc.st_asr.st_pi.s2_kp] [gst_foc.st_asr.st_pi.s2_kidt]
com_f4_asr_lpf_cof_hz	float	ASR LPF natural frequency [Hz]	[gst_foc.st_asr.st_lpf.s2_in_k] [gst_foc.st_asr.st_lpf.s2_out_k]
com_f4_acr_lpf_cof_hz	float	ACR LPF natural frequency [Hz]	[gst_foc.st_acr.st_iq_lpf.s2_in_k] [gst_foc.st_acr.st_iq_lpf.s2_out_k] [gst_foc.st_acr.st_id_lpf.s2_in_k] [gst_foc.st_acr.st_id_lpf.s2_out_k]
com_f4_pll_nf_hz	float	PLL natural frequency [Hz]	[gst_foc.st_pll.st_pi.s2_kp] [gst_foc.st_pll.st_pi.s2_kidt]
com_s2_less2ol_speed_rpm	int16_t	Switching speed from sensorless to open loop [rpm]	[gst_foc.st_asr.s2_cl2ol_speed_rad]
com_s2_ol2less_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_init_asr_intg	float	ASR integral term initial value during sensorless transition	[gst_foc.st_asr.s2_init_intg]
com_f4_asr_ki_adj	float	Speed PI control integral term adjustment parameter	[gst_foc.st_pll.st_pi.s2_kidt]
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_foc.st_sscs.s2_duty_diff_limit]

Table 4.2 List of Input Variables for Analyzer(2/2)

variable	type	content	remarks ([ ]: reflection variable name)
com_s2_ad_point_a_adj_cnt	int16_t	Adjustment for A/D delay counts for A point	[gst_foc.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_foc.st_sscs.s2_ad_point_b_adj_cnt]
com_f4_sal_angle_current	float	TRX count value differential of angle detection for salient rotor	【gst_foc.st_ipd. u1_sal_angle_current】
com_u4_sal_angle_threshold	uint32_t	Maximum number of times salient rotor angle detection is measured	【gst_foc.st_ipd. u4_sal_angle_threshold】
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. u1_sal_polarity_current】
com_u4_sal_polarity_threshold	uint32_t	Maximum number of times salient rotor polarity detection is measured	【gst_foc.st_ipd. u4_sal_polarity_threshold】
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. u1_non_sal_current】
com_u4_non_sal_threshold	uint32_t	Maximum number of times non-salient rotor angle detection is measured	【gst_foc.st_ipd. u4_dnon_sal_threshold】
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_s2_speed_th_rpm	int16_t	Field-Weakening Control speed threshold [rpm]	【gst_foc.st_fw.s2_speed_th_rad】
com_f4_v_mag_th	float	Maximum output voltage [V]	【gst_foc.st_fw.s2_v_mag_th】
com_f4_delta_id	float	Id change amount of field-weakening Control [A]	【gst_foc.st_fw.s2_delta_id】
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_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 2.0 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.

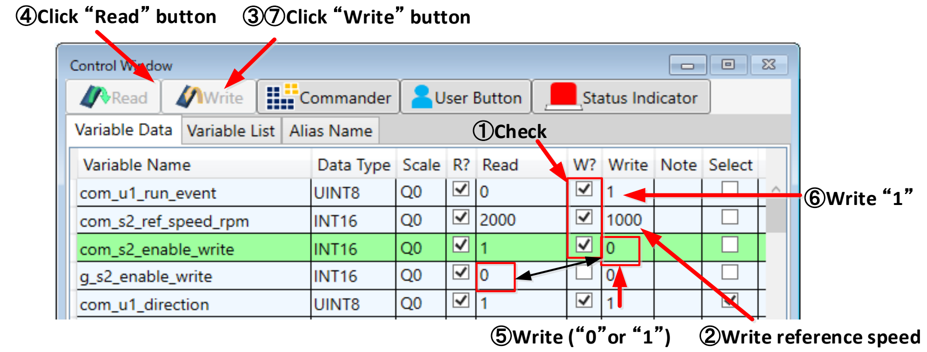


Figure 4-2 Procedure - Driving the motor

- Stop the motor

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

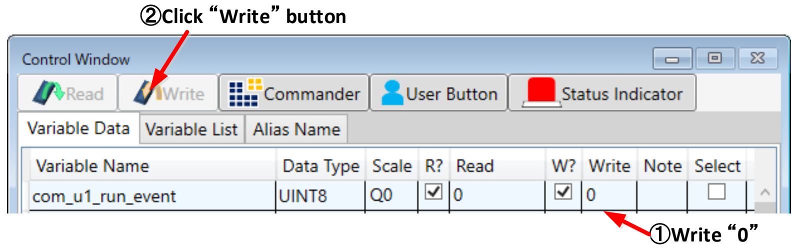


Figure 4-3 Procedure - Stop the motor

- Error cancel operation

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

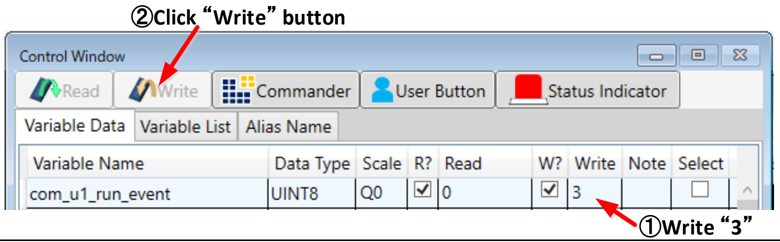


Figure 4-4 Procedure - Error cancel operation

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul, 31, 2017	-	First edition issued
2.10	Jun, 1, 2021	-	Changed motor control algorithm and scaling structure of the software Updated DSP library
2.20	Nov, 29, 2021	-	Addition of initial position detection, damping control, and weak magnetic flux control function Update DSP library

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.

Corporate Headquarters

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

Trademarks

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/.