Vector Control of Three-Phase Induction Motor Used in Driving a Pump

RX13T Implementation

Introduction

This application note describes how to use the sample program to drive a three phase induction motor (pump motor) with vector control using the RX13T microcontroller and the motor control development support tool 'Renesas Motor Workbench 2.0'.

The sample program is only provided for reference purposes and Renesas does not guarantee its operation. This sample program should only be used after thorough evaluation in an appropriate operating environment.

In particular, high-voltage environments are extremely dangerous. The information provided here should only be used after reading all the user's manuals for the development environment and observing all safety precautions. Renesas Electronics assumes no responsibility for an accident or loss occurring from the use of the development environments described in this document.

Target Device

Operation of the sample program provided with this application note has been verified for the following device.

- RX13T (R5F513T5ADFL)

Target Sample Program

The sample program discussed in this application note is the following.

[1] RX13T48_T1102_3IM_LESS_FOC_CSP_PUMP_V110
RX13T48 (R5F513T5ADFL) T1102 sample program: Vector Control of Three-Phase Induction Motor Used in Driving a Pump

Reference Documents

- RX13T Group User's Manual; Hardware (R01UH0822EJ0100)
- Motor Control Application: Vector Control of Three-Phase Induction Motor (Algorithms) (R01AN2193EJ0100)
- Motor Control Development Support Tool 'Renesas Motor Workbench 2.0'
  Download from https://www.renesas.com/us/en/software/D3017970.html
- Trial series "T1102" 3kW 4kVA Inverter Unit User's Manual
- RX13T CPU Card (RTK0EMXA10C00000BJ) Schematic (R12TU0062EJ0100)

Contents

1. Overview

This application note describes how to implement a sample program for driving three-phase induction motor by vector control from the RX13T microcontroller, and how to use the library of 'Renesas Motor Workbench 2.0' (RMW)*1, that is support tool for motor control development. Note that the sample program described here uses the algorithm described in the Motor Control Application: Vector Control of Three-Phase Induction Motor (Algorithms).

1.1 Development Environment

Table 1.1 lists the elements of the development environment for the sample program covered in this application note.

Table 1.1 Sample Program Development Environment

Sample Program	Microcontroller	Inverter Board	Motor	Version of CS+
[1]	R5F513T5ADFL	T1102 *2	PE2-256-0.4T *3	V8.05.00

Contact your sales representative or authorized Renesas Electronics distributors for details on purchasing the T1102 inverter board and technical support.

Note 1. Motor Control Development Support Tool 'Renesas Motor Workbench 2.0' is products of Renesas Electronics Corporation.

Note 2. The T1102 inverter board and the In Circuit Scope development support tool are products of Desk Top Laboratories Inc.
Website: http://desktoplab.co.jp/ (provided in Japanese only)

Note 3. PE2-256-0.4T is a product of Kawamoto Pump Mfg. Co., Ltd.
Website: https://www.kawamoto-global.com/

2. System Overview

This section gives an overview of the system described in this application note.

2.1 Hardware Configuration

The hardware configuration is shown below.

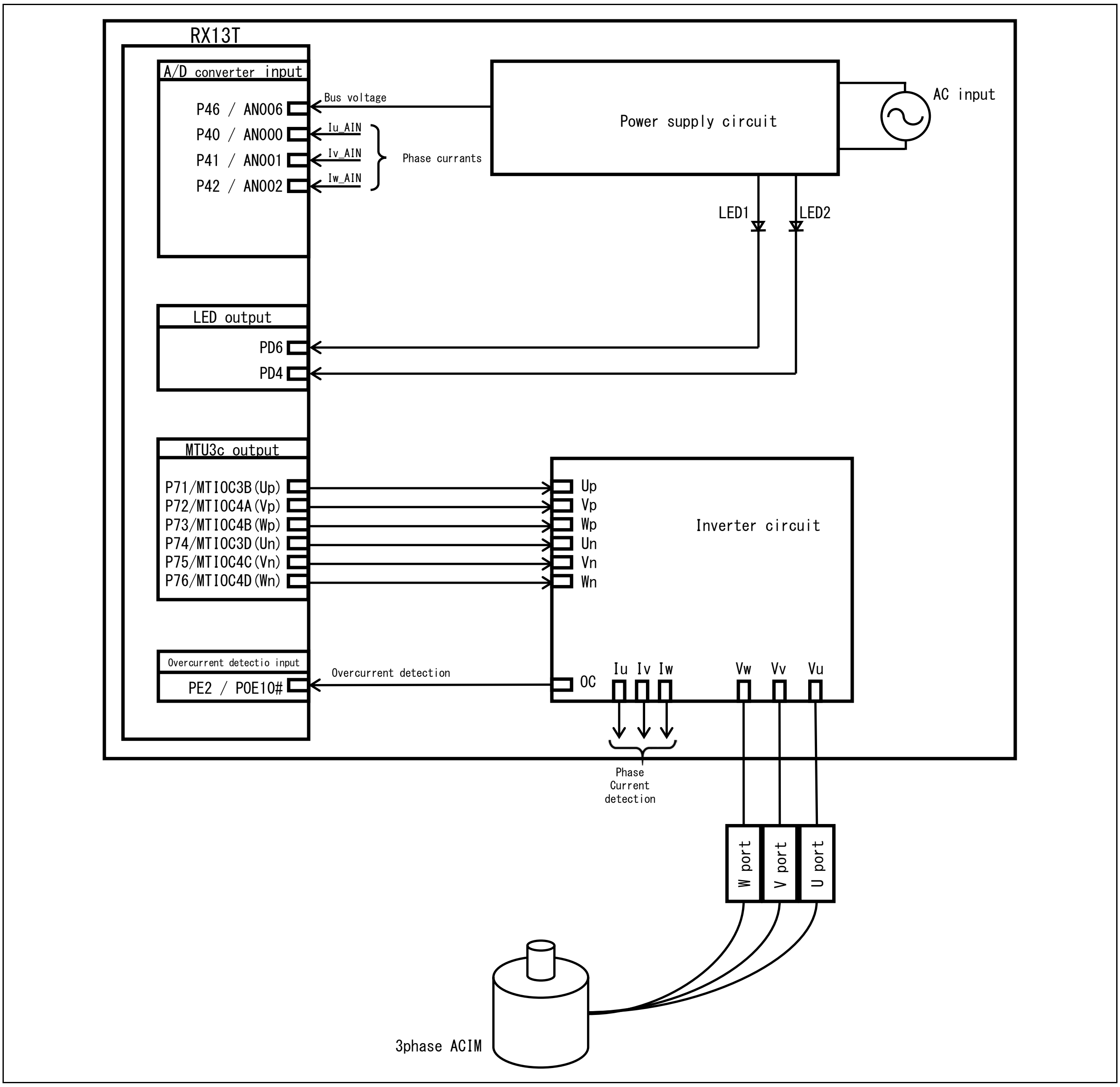

Figure 2.1 Hardware Configuration

2.2 Hardware Specifications

2.2.1 User Interfaces

Table 2.1 lists the user interfaces for use in this system.

Table 2.1 User Interfaces

Item	Interface Component	Function
LED1	Yellow-green LED	• Motor is running: On • Motor is stopped: Off
LED2	Yellow-green LED	• An error is detected: On • Normal operation: Off
RESET	Pushbutton switch RESET1	System reset

Table 2.2 lists the pin interfaces for use in this system.

Table 2.2 Pin Interfaces

R5F513T5ADFL Pin Name	Function
P46/AN006	Inverter bus voltage measurement
PD6	LED1 on/off control
PD4	LED2 on/off control
P40/AN000	Measurement of the U-phase current
P41/AN001	Measurement of the V-phase current
P42/AN002	Measurement of the W-phase current
P45/AN005*1	Measurement of the intelligent power module (IPM) temperature
P71/MTIOC3B	Complementary PWM output (U_p)
P72/MTIOC4A	Complementary PWM output (V_p)
P73/MTIOC4B	Complementary PWM output (W_p)
P74/MTIOC3D	Complementary PWM output (U_n)
P75/MTIOC4C	Complementary PWM output (V_n)
P76/MTIOC4D	Complementary PWM output (W_n)
PE2/POE10#	Input for the emergency signal for stopping the PWM output on detection of an overcurrent

Note 1. Not connected on the CPU board (function is disabled)

2.2.2 Peripheral Modules

The peripheral modules for use with this system are listed below.

Table 2.3 Peripheral Modules for Use with the Sample Program

MCU	12-bit ADC	CMT	MTU3c	POE3C
RX13T	• Individual currents of U/V/W phases • Inverter bus voltage	1-ms and 100-ms interval timers	Complementary PWM output	Initialization of the complementary PWM output port (The pins being used for PWM output are placed in the high-impedance state and PWM output is stopped)

(1) 12-bit A/D converter
Using 12-bit A/D converters to measure the U-, V-, and W-phase currents (I_u, I_v, and I_w), inverter bus voltage (V_{dc}).
The operating mode is set to group scan mode, with the use of the sample-and-hold function, and the synchronous trigger to start a conversion.

(2) Compare match timer (CMT)
Channel 0 of the compare match timer is used as a 1-ms interval timer.
Channel 1 of the compare match timer is used as a 100-ms interval timer.

(3) Multi-function timer pulse unit 3 (MTU3c)
The operating mode varies with channels, with channels 3 and 4 being used in complementary PWM mode to output an active-high signal that includes dead time.

(4) Port output enable 3 (POE3C)
When an overcurrent is detected (indicated by a falling edge on the POE10# pin) or when an output short-circuit is detected, the pins being used for PWM output are placed in the high-impedance state, PWM output is stopped, and the complementary PWM output port pins are initialized.

2.3 Software Configuration

2.3.1 File Configuration

Table 2.4 lists the folders and files for this sample program.

Table 2.4 Folders and Files for the Sample Program [1]

Sample Program	Folder Name	File Name	Description
RX13T48_T1102_3IM_LESS_FOC_CSP_PUMP_V100	inc	main.h	Main function and user interface control header file
		mtr_common.h	Common definitions header file
		mtr_api.h	Application interface processing header file
		mtr_ctrl_t1102.h	Board-dependent processing header file
		mtr_ctrl_rx13t48.h	RX13T-dependent processing header file
		mtr_3im_less_foc.h	Sensorless vector control header file
		control_parameter.h	Control parameter header file
		motor_parameter.h	Motor parameter header file
		mtr_ctrl_rx13t48_t1102.h	Board- and RX13T-dependent processing header file
		r_init_clock.h	Hader file for initial setting of the clock signals for the RX13T
	ics	ICS2_RX13T.lib	ICS library
		ICS2_RX13T.h	ICS library header file
	src	main.c	Main function and user interface control
		mtr_api.c	Application interface processing
		mtr_ctrl_t1102.c	Board-dependent processing
		mtr_ctrl_rx13t48.c	RX13T-dependent processing
		mtr_interrupt.c	Interrupt handlers
		mtr_3im_less_foc.c	Sensorless vector control
		mtr_ctrl_rx13t48_t1102.c	Board- and RX13T-dependent processing
		r_init_clock.c	Initial setting of the clock signals for the RX13T

2.3.2 Configuration of the Sample Program

The software modules used in this sample program are shown in Figure 2.2 and Table 2.5.

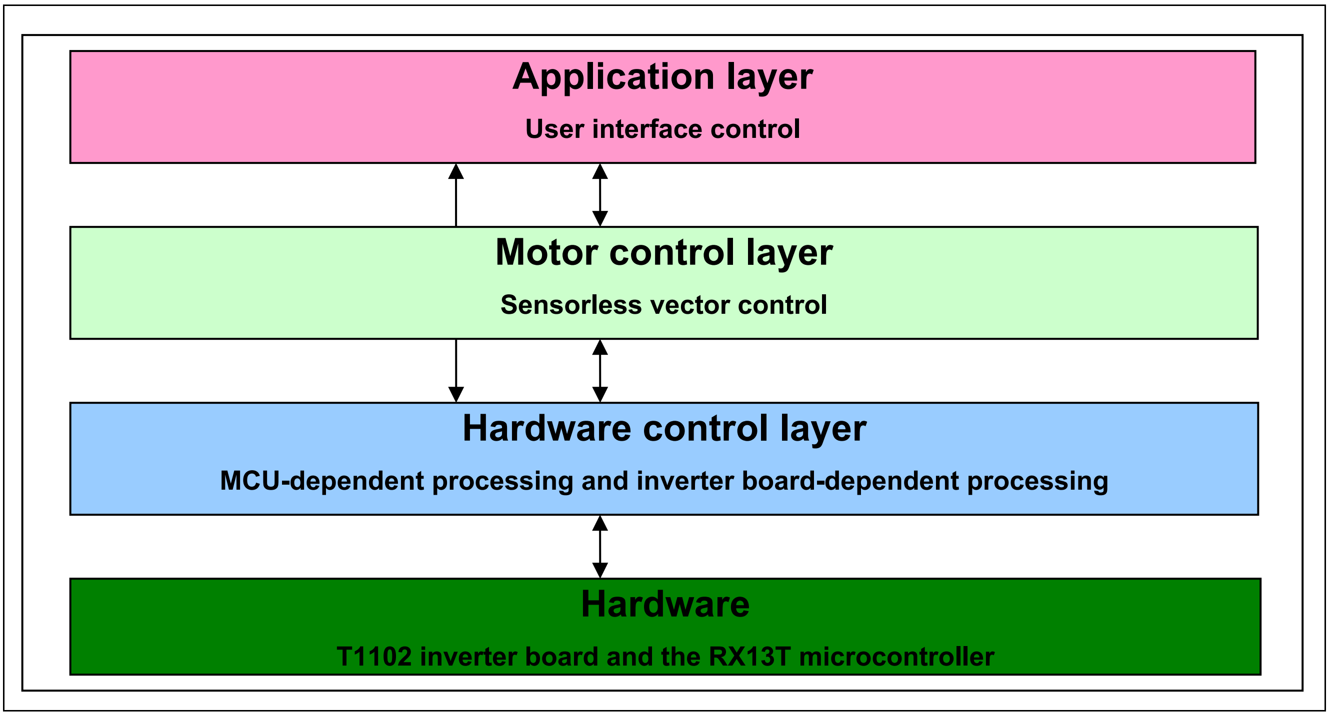

Figure 2.2 Configuration of the Software Modules Used in the Sample Program

Table 2.5 Configuration of the Software Modules Used in the sample Program [1]

Layer	File
Application layer	main.c mtr_api.c
Motor control layer	mtr_3im_less_foc.c
Hardware control layer	mtr_ctrl_rx13t48_t1102.c mtr_ctrl_rx13t48.c mtr_ctrl_t1102.c r_init_clock.c r_init_port_initialize.c r_init_rom_cache.c r_init_stop_module.c

2.4 Software Specifications

Table 2.6 lists the basic specifications of this system software. See the Motor Control Application: Vector Control of Three-Phase Induction Motor (Algorithms) for details on the vector control.

Table 2.6 Basic Specifications of the Vector Control Program (for Sample Program [1])

Item	Description
Control method	Vector control (Sensorless control mode and open loop mode are switched according to the rotational speed command value.)
Starting and stopping of motor rotation	Handled by RMW (See the ‘com_u1_operation variable’ in Table 4.1.)
Detection of rotor's magnetic pole position	Sensorless
Input voltage	220 VAC
Carrier frequency (PWM)	16 kHz
Control period	125 μs (twice the carrier frequency)
Rotational speed range	50 rpm to 3600 rpm *1
System protection	• For system protection, the motor control signal outputs (6 lines) are set to the inactive level in response to any of the following four conditions. 1. The current in any phase exceeds 7 A (monitored once every 125 μs). 2. The inverter bus voltage exceeds 420 V (monitored once every 125 μs). 3. The inverter bus voltage falls below 0 V (monitored once every 125 μs). 4. The rotational speed exceeds 4680 rpm (monitored once every 125 μs). • The pins being used for PWM output are placed in the high-impedance state in response to external input of an overcurrent detection signal (detection of a falling edge on the POE10# pin). • The motor control signal outputs (6 lines) are set to the inactive level in response to any of the following two conditions of application errors. 1. Overload error: The resultant current of the motor is 6 A or more (monitored within the main loop). 2. Dry running error: When the rotational speed is 2000 rpm or more, the resultant current of the motor is 1 A or less (monitored within the main loop).

Note 1. There may be a difference between the actual speed and the reference speed depending on the working environment.

3. Control Program

This section describes the sample program covered in this application note.

3.1 Control

3.1.1 Starting and Stopping the Motor

Starting and stopping of the motor are controlled by using RMW to set a value to the motor operation variable 'com_u1_operation'. 'com_u1_operation' reads in the main loop and performs an operation for each bit of the variable.

The following table shows the behavior of each bit when a value is set.

bit0: write '0' to stop the motor, write '1' to start motor.

bit1: write '0' to disable timer control, write '1' to enable timer control

when the timer control is enabled, the motor starts and stops at a fixed cycle.

bit2: error reset when the value changes from '0' to '1'. In error mode, changing the value from '0' to '1' resets the error.

3.1.2 Values as Motor Rotational Speed Commands

Using RMW to set rotation speed command value in 'com_s2_ref_speed_rpm'. The unit of the speed command value is rpm.

3.1.3 Inverter Bus Voltage

As shown in the table below, the measured values of the inverter bus voltage are used in producing the modulation factor and for overvoltage detection.

Detection of abnormal voltages leads to stopping of the PWM output.

Table 3.1 Conversion Ratio for Inverter Bus Voltage

Item	Sample Program	Conversion Ratio (Inverter bus voltage: A/D converted value)	Channel
Inverter bus voltage	[1]	0 V to 686.5 V: 0000H to 0FFFH	AN006

3.1.4 Phase Current

As shown in the table below, the measured values of U-, V-, and W-phase currents are used for vector control and overcurrent detection.

Table 3.2 Conversion Ratio for U-, V-, W-Phase Currents

Item	Sample Program	Conversion Ratio (U-, V-, W-phase currents: A/D converted value)	Channel
U-, V-, W-phase currents	[1]	-50 A to 50 A: 0000H to 0FFFH	Iu: AN000 Iv: AN001 Iw:AN002

3.1.5 Mode Switching between Sensorless Control and Open Loop

Open loop mode is used at low speeds and sensorless control mode is used at medium and high speeds.

Sensorless control mode and open loop mode are switched according to the motor rotational speed command value.

3.1.6 Modulation

In this sample program, the voltage for input to the motor (the voltage output from the inverter) is generated by pulse width modulation (PWM). Comparison of the command voltage waveform with a triangular waveform determines the PWM pulse width.

(1) Triangle Wave Comparison Method

This is the method for the physical output of the desired voltage. The pulse width for the voltage to be output is determined on the basis of the results of comparing the command voltage waveform with the carrier waveform (triangle wave). The desired voltage is output as a pseudo-sinusoidal waveform by switching the output on when the voltage is greater than that produced by the carrier wave and off when the voltage is lower than that produced by the carrier wave.

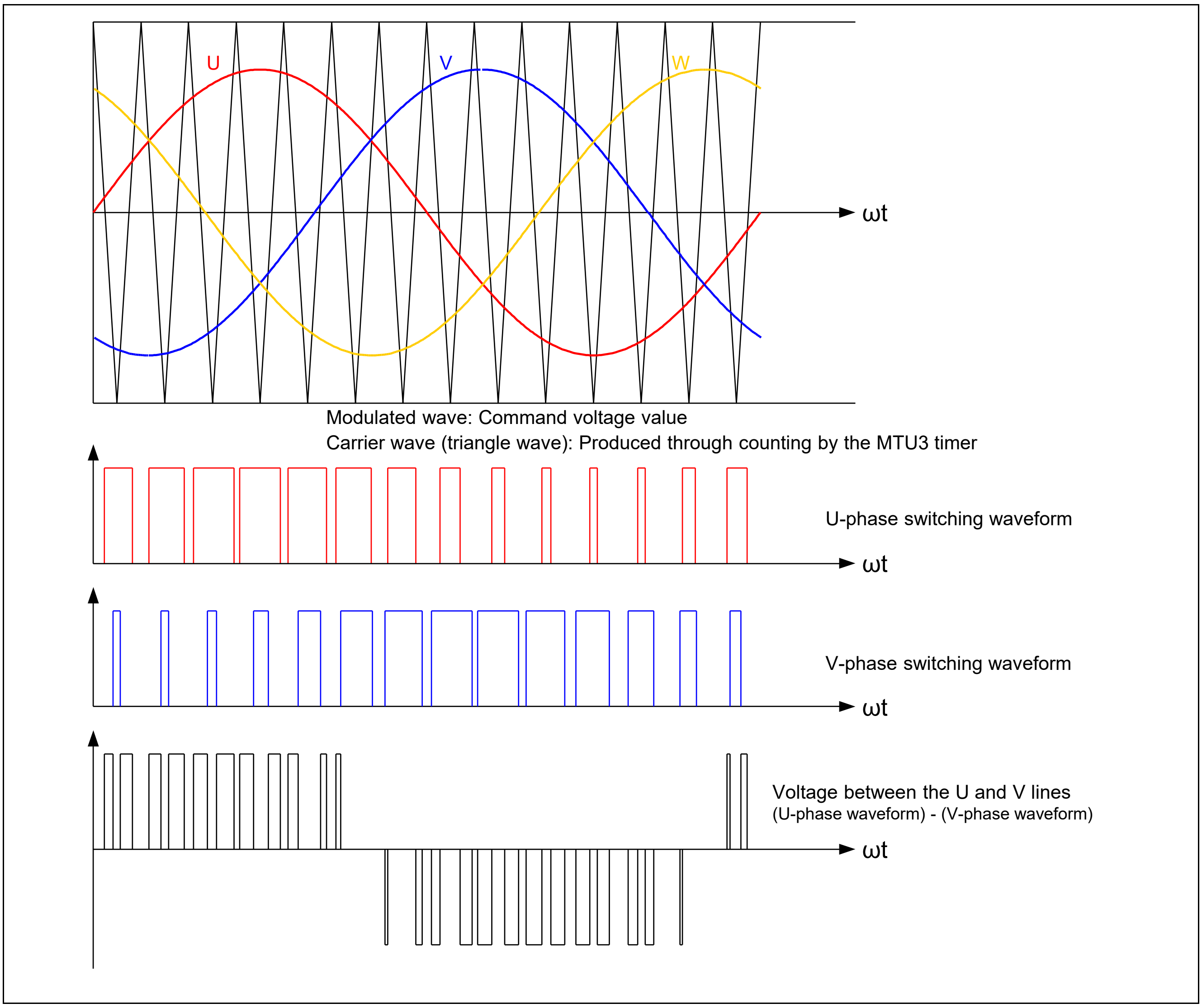

Figure 3.1 Concepts of Triangle Wave Comparison Method

Here, as shown in Figure 3.2, the ratio of the output voltage pulse to the carrier wave is called duty.

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

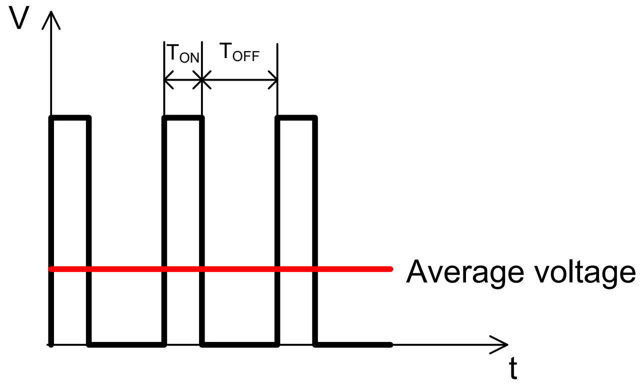

Figure 3.2 Definition of Duty

The modulation factor *m* is defined as follows.

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

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

A desired control is accomplished by setting this modulation factor in the register for use in determining the PWM duty.

3.1.7 State Transitions of the Application

Figure 3.3 shows the state transitions of the application within the sensorless vector control program.

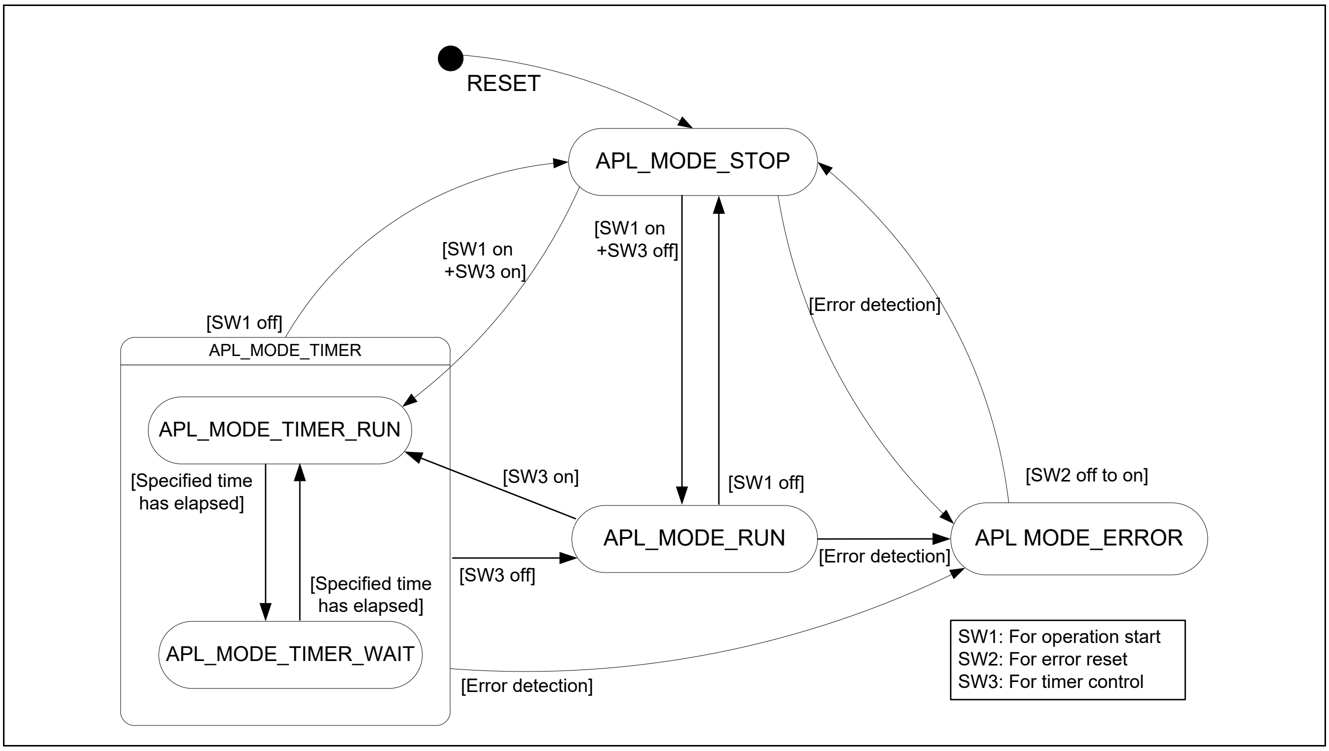

Figure 3.3 State Transitions of the Application within the Sensorless Vector Control Program

3.1.8 State Transition of Motor Control

Figure 3.4 shows the state transitions of motor control within the sensorless vector control program.

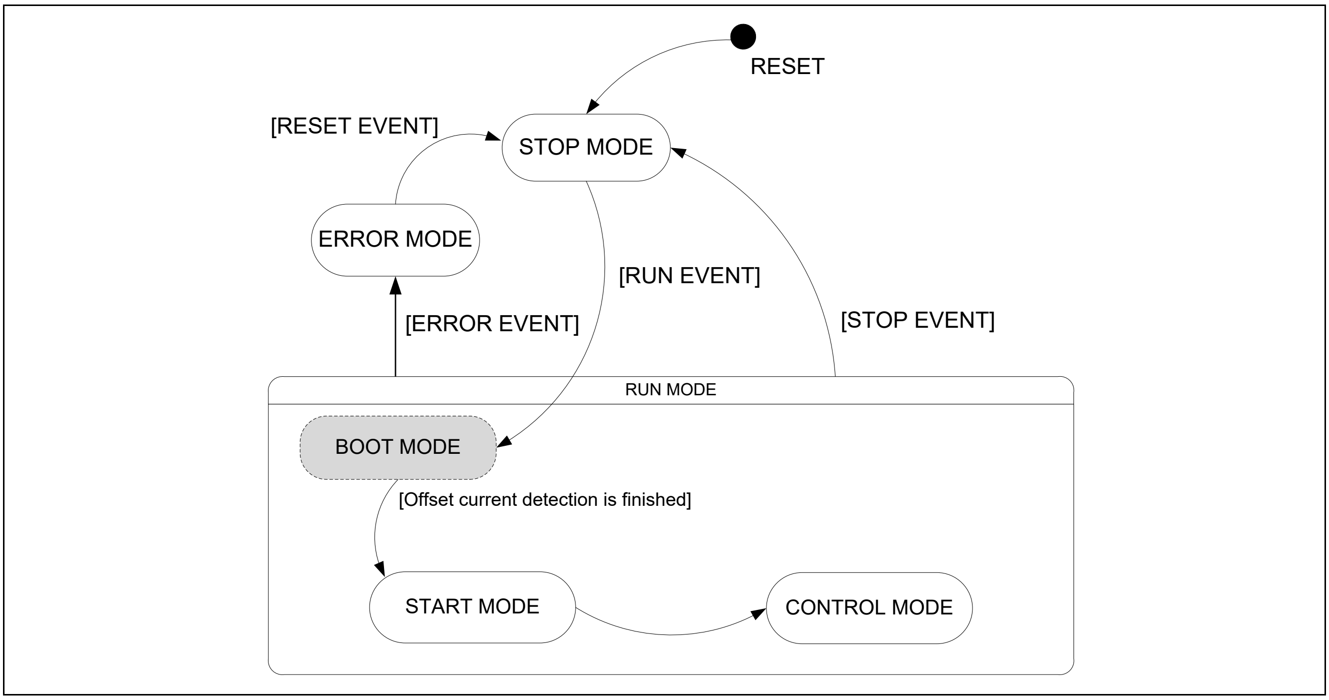

Figure 3.4 State Transitions of Motor Control within the Sensorless Vector Control Program

3.1.9 System Protection Functions

This control program detects the following four errors and initiates an emergency stop in response to each of them. See Table 3.3 for the values used for the system protection functions.

- Overcurrent error
  The PWM output pins are placed in the high-impedance state in response to an emergency stop signal (overcurrent detection) from the hardware.
  In addition, U-, V-, and W-phase currents are monitored in overcurrent monitoring cycles. When an overcurrent (the current exceeding the overcurrent limit value) is detected, the CPU initiates an emergency stop of the PWM output (in response to detection by the software).

- Overvoltage error
  The inverter bus voltage is monitored in overvoltage monitoring cycles. When an overvoltage (the voltage exceeding the overvoltage limit value) is detected, the CPU initiates an emergency stop of the PWM output.

- Undervoltage error
  The inverter bus voltage is monitored in low-voltage monitoring cycles. When low voltage (the voltage falls below the limit value) is detected, the CPU initiates an emergency stop of the PWM output.

- Rotational speed error
  Rotational speed is monitored in speed monitoring cycles and if the speed limit is exceeded, the CPU initiates an emergency stop of the PWM output.

Table 3.3 Values for the System Protection Functions in Sample Program [1]

Error	Item	Value
Overcurrent error	Overcurrent limit value [A]	7
	Monitoring cycle [μs]	125
Overvoltage error	Overvoltage limit value [V]	420
	Monitoring cycle [μs]	125
Undervoltage error	Undervoltage limit value [V]	0
	Monitoring cycle [μs]	125
Rotational speed error	Speed limit value [rad/s] (electrical angle)	490
	Monitoring cycle [μs]	125

3.1.10 Detection of Application Errors

This control program detects the following two application errors and initiates an emergency stop in response to each of them. For the values that are taken to represent the error conditions, see Table 3.4.

- Overload error
  The control program monitors the resultant current of the motor within the main loop and performs an emergency stop upon detecting an overload i.e. when the resultant current equal to or greater than the threshold for overload detection.
- Dry running error
  The control program monitors the resultant current of the motor within the main loop and performs an emergency stop upon detecting a dry running state(when the rotational velocity equal to or greater than the threshold for the dry running rotational velocity detection and resultant current equal to or less than the threshold for the dry running current detection).

Table 3.4 Values for the Detection of Application Errors

Error	Item	Value
Overload error	Threshold value for the resultant current for overload determination [A]	6
Dry running error	Threshold value for the resultant current for dry running determination [A]	1
	Threshold value for the rotational speed at which dry running determination starts [rpm]	2000

3.2 Functions for Use in Vector Control Software Program

The control program uses multiple control functions as listed in the table below.

See the flowcharts and the source code for more detailed information on the processing performed by these functions.

Table 3.5 Control Functions (1/9)

File Name	Function Name	Processing Overview
main.c	main Input: None Output: None	• Calls the hardware initialization function • Calls the user interface initialization function • Calls the main processing variables initialization function • Calls the function to execute state transitions and events • Main processing ⇒ Calls the function that performs the main processing ⇒ Calls the watchdog timer clearing function
	ics_ui Input: None Output: None	For use by the ICS user interface
	software_init Input: None Output: None	Initializes variables used in the main processing
	APL_mode_stop Input: None Output: None	Processing performed when the application is in stopped mode
	APL_mode_run Input: None Output: None	Processing performed when the application is in running mode
	APL_mode_timer_run Input: None Output: None	Processing performed when the application is running in timer-controlled operation mode
	APL_mode_timer_wait Input: None Output: None	Processing performed when the application is waiting for timer-controlled operation mode
	APL_mode_error Input: None Output: None	Processing performed when the application is in error mode

Table 3.5 Control Functions (2/9)

File Name	Function Name	Processing Overview
mtr_api.c	API_set_start Input: None Output: (uint16) u2_result/ execution result	Starts motor control
	API_set_stop Input: None Output: (uint16) u2_result/ execution result	Stops motor control
	API_set_reset Input: None Output: None	Resets motor control
	API_get_status Input: None Output: (uint8) u1_result/ motor control state	Acquires the motor control state
	API_get_error Input: None Output: (uint8) u1_error_status/ error state	Acquires information about motor control errors
	API_set_speed Input: (uint16) u2_temp/ rotational speed command Output: (uint16) u2_result/ execution result	Sets the rotational speed command for motor control
	API_get_motor_current Input: None Output: (float32) f4_i1_lpf/ resultant current of the motor	Acquires the resultant current of the motor

Table 3.5 Control Functions (3/9)

File Name	Function Name	Processing Overview
mtr_ctrl_t1102.c	R_MTR_ChargeCapacitor Input: None Output: None	Waits for the charging time for the smoothing capacitor
	ic_gate_on Input: None Output: None	Sets the gate signal used for inrush prevention to the ON state
	led1_on Input: None Output: None	Turns on LED1
	led2_on Input: None Output: None	Turns on LED2
	led1_off Input: None Output: None	Turns off LED1
	led2_off Input: None Output: None	Turns off LED2
	get_sw1 Input: None Output: (uint8) u1_temp/ the state of SW1	Gets the state of SW1
	get_sw2 Input: None Output: (uint8) u1_temp/ the state of SW2	Gets the state of SW2
	get_sw3 Input: None Output: (uint8) u1_temp/ the state of SW3	Gets the state of SW3
	get_vr1 Input: None Output: (uint16) u2_temp/ rotational speed command	Gets the A/D converted value of the rotational speed command

Table 3.5 Control Functions (4/9)

File Name	Function Name	Processing Overview
mtr_ctrl_rx13t48.c	R_MTR_InitHardware Input: None Output: None	Initializes clock signals and peripheral modules
	mtr_init_module Input: None Output: None	Releases the module stop state
	mtr_init_cmt Input: None Output: None	Initializes the CMT
	mtr_init_poe3 Input: None Output: None	Initializes the POE3B
	init_wdt Input: None Output: None	Initializes the WDT
	clear_wdt Input: None Output: None	Clears the WDT
	mtr_clear_oc_flag Input: None Output: None	Releases the high-impedance state
	mtr_clear_cmt0_flag Input: None Output: None	Clears the CMT0 interrupt flag
	mtr_clear_cmt1_flag Input: None Output: None	Clears the CMT1 interrupt flag

Table 3.5 Control Functions (5/9)

File Name	Function Name	Processing Overview
mtr_interrupt.c	mtr_over_current_interrupt Input: None Output: None	Overcurrent detection processing ● Calls the event processing selection function ● Changes the motor status ● Calls the function that clears the high-impedance state
	mtr_mtu4_interrupt Input: None Output: None	Called once every 125 μs ● Vector control ● Calls the sensorless control processing ● Calls the open loop control processing
	mtr_less_foc_interrupt Input: None Output: None	Sensorless control processing ● Current PI control ● Voltage value calculation
	mtr_open_loop_interrupt Input: None Output: None	Open loop control processing ● Current PI control ● Voltage value calculation
	mtr_cmt0_interrupt Input: None Output: None	Called once every 1 ms ● Start control ● Speed PI control
	mtr_cmt1_interrupt Input: None Output: None	Called once every 100 ms ● Timer control count processing ● Count processing for starting error determination

Table 3.5 Control Functions (6/9)

File Name	Function Name	Processing Overview
mtr_3im_less_foc.c	R_MTR_InitSequence Input: None Output: None	Sequence processing initialization
	R_MTR_ExecEvent Input: (uint8)u1_event/ event that occur Output: None	• Updates the motor state • Calls the function that handles the appropriate processing in response to events that occur
	mtr_act_run Input: (uint8)u1_state/ motor status Output: (uint8)u1_state/ motor status	• Calls the function that initializes the variables used at motor startup • Calls the function that starts motor control
	mtr_act_stop Input: (uint8)u1_state/ motor status Output: (uint8)u1_state/ motor status	Calls the function that terminates motor control
	mtr_act_none Input: (uint8)u1_state/ motor status Output: (uint8)u1_state/ motor status	No processing
	mtr_act_reset Input: (uint8)u1_state/ motor status Output: (uint8)u1_state/ motor status	Initializes global variables
	mtr_act_error Input: (uint8)u1_state/ motor status Output: (uint8)u1_state/ motor status	Calls the function that terminates motor control
	mtr_start_init Input: None Output: None	Initializes only those variables needed at motor startup
	mtr_stop_init Input: None Output: None	Initializes variables needed at motor stop
	mtr_pi_ctrl Input: MTR_PI_CTRL *pi_ctrl/ structure for PI control Output: (float32)f4_ref/ PI control output value	Used in PI control
	mtr_set_variables Input: None Output: None	Sets the motor variables
	R_MTR_IcsInput Input: MTR_ICS_INPUT *ics_input/ structure for ICS Output: None	Sets up buffers
	R_MTR_SetSpeed Input: (float32)ref_speed/ rotational speed command value Output: None	Sets the rotational speed command value
	R_MTR_SetDir Input: (uint8)dir/ rotational direction Output: None	Sets the direction of rotation
	R_MTR_GetSpeed Input: None Output: (float32)f4_speed_rpm/ speed	Acquires the speed calculation value
	R_MTR_GetDir Input: None Output: (uint8) g_u1_direction	Acquires the value of the rotational direction
	R_MTR_GetStatus Input: None Output: (uint8)g_u1_mode_system/ motor status	Acquires the motor status

Table 3.5 Control Functions (7/9)

File Name	Function Name	Processing Overview
mtr_3im_less_foc.c	mtr_error_check Input: None Output: None	Error monitoring and detection
	mtr_set_speed_ref Input: None Output: None	Sets the command used for speed control
	mtr_set_iq_ref Input: None Output: None	Sets the δ axis current command
	mtr_set_id_ref Input: None Output: None	Sets the γ axis current command
	mtr_calc_mod Input: (float32) f4_vu/ U-phase voltage (float32) f4_vv/ V-phase voltage (float32) f4_vw/ W-phase voltage (float32) f4_vdc/ bus voltage Output: None	Modulation factor calculation

Table 3.5 Control Functions (8/9)

File Name	Function Name	Processing Overview
mtr_ctrl_rx13t48_t1102.c	mtr_init_mtu Input: None Output: None	Initialization of the MTU3d
	mtr_init_io_port Input: None Output: None	Initialization of the I/O ports
	mtr_init_ad_converter Input: None Output: None	Initialization of the A/D converters
	init_ui Input: None Output: None	Initialization of the user interfaces
	mtr_ctrl_start Input: None Output: None	Motor startup processing
	mtr_ctrl_stop Input: None Output: None	Motor stop processing
	mtr_get_iuiviwvdc Input: (float32)*f4_iu_ad/ A/D converted value of U-phase current (float32)*f4_iv_ad/ A/D converted value of V-phase current (float32)*f4_iw_ad/ A/D converted value of W-phase current (float32)*f4_vdc_ad/ A/D converted value of Vdc Output: None	A/D conversion of U-, V-, and W-phase currents and inverter bus voltage
	mtr_get_ipm_temperature Input: None Output: (int16)s2_temp/ A/D converted value of the IPM temperature voltage	A/D conversion of the IPM temperature
	mtr_clear_mtu4_flag Input: None Output: None	Clears the interrupt flag
	mtr_inv_set_uvw Input: (float32)f4_modu/ U phase modulation factor (float32)f4_modv/ V phase modulation factor (float32)f4_modw/ W phase modulation factor Output: None	PWM output setting
	mtr_init_register Input: None Output: None	Initialization for the PWM value comparison

Table 3.5 Control Functions (9/9)

File Name	Function Name	Processing Overview
r_init_clock.c	R_INIT_Clock Input: None Output: None	Initialization of the clock signals
	CGC_oscillation_main Input: None Output: None	Main clock oscillation setting
	CGC_oscillation_PLL Input: None Output: None	PLL clock oscillation setting
	CGC_oscillation_HOCO Input: None Output: None	HOCO clock oscillation setting

3.3 Software Variables Used in the Sensorless Vector Control Program

The table below lists the variables used in the control program.

Table 3.6 List of Variables (1/3)

Variable Name	Type	Description	Remarks
g_u1_motor_status	uint8	Motor control state	0: Stop mode 1: Run mode 2: Error mode
g_u1_drive_sw_state	uint8	State of the operation start switch	
g_u1_err_reset_sw_state	uint8	State of the error reset switch	
g_u1_timer_mode_sw_state	uint8	State of the timer control switch	
g_f4_ref_speed_ad	float32	A/D converted value of the rotational speed command	
g_s2_ref_speed_rpm	int16	Rotational speed command value	[rpm]
apl_u1_mode_status	uint8	Application state	0: APL_MODE_STOP 1: APL_MODE_RUN 2: APL_MODE_TIMER_RUN 3: APL_MODE_TIMER_WAIT 4: APL_MODE_ERROR
apl_u4_cnt_timer	uint32	Timer control mode counter	
apl_u1_error	uint8	Application error state management	0: No error 1: Overload error 2: Dry running error 4: Motor control error
apl_u4_err_chk_start_time	uint32	Counter for starting application error detection	
g_u1_mode_system	uint8	State management	0: Stop mode 1: Run mode 2: Error mode
g_u2_run_mode	uint16	Operating mode management	0: Boot mode 2: Control mode
g_u2_ctrl_mode	uint16	Vector control mode	1: Open loop mode 2: Sensorless vector control mode
g_u1_error_status	uint8	Motor control error state management	1: Overcurrent error 2: Overvoltage error 3: Rotational speed error 7: Low voltage error 8: IPM temperature error 0xFF: Undefined error
g_f4_vdc_ad	float32	Inverter bus voltage	[V]
g_f4_vd_ref	float32	γ axis output voltage command	[V]
g_f4_vq_ref	float32	δ axis output voltage command	[V]
g_f4_iu_ad	float32	U-phase current	[A]
g_f4_pre_iu_ad	float32	Previous U-phase current value	[A]
g_f4_iv_ad	float32	V-phase current	[A]
g_f4_pre_iv_ad	float32	Previous V-phase current value	[A]
g_f4_iw_ad	float32	W-phase current	[A]
g_f4_pre_iw_ad	float32	Previous W-phase current value	[A]

Table 3.6 List of Variables (2/3)

Variable Name	Type	Description	Remarks
g_f4_offset_iu	float32	U-phase current offset	[A]
g_f4_offset_iv	float32	V-phase current offset	[A]
g_f4_offset_iw	float32	W-phase current offset	[A]
g_f4_id_lpf	float32	γ axis current	[A]
g_f4_iq_lpf	float32	δ axis current	[A]
g_f4_i1_lpf	float32	Resultant current of the motor	[A]
g_f4_pre_id_lpf	float32	Previous γ axis current value	[A]
g_f4_pre_iq_lpf	float32	Previous δ axis current value	[A]
g_f4_kp_id	float32	γ axis current PI control proportional gain	
g_f4_ki_id	float32	γ axis current PI control integral gain	
g_f4_lim_id	float32	γ axis current PI control limit value	[A]
g_f4_ilim_id	float32	γ axis current PI control integral term limit value	[A]
g_f4_kp_iq	float32	δ axis current PI control proportional gain	
g_f4_ki_iq	float32	δ axis current PI control integral gain	
g_f4_lim_rotor_speed_rad	float32	Rotor speed PI control limit value	Electrical angle [rad/s]
g_f4_ilim_rotor_speed_rad	float32	Rotor speed PI control integral term limit value	Electrical angle [rad/s]
g_f4_id_ref	float32	γ axis current command	[A]
g_f4_id_ref2	float32	γ axis current command 2	[A]
g_f4_iq_ref	float32	δ axis current command	[A]
g_f4_ref_stator_speed_rad	float32	Stator rotational speed command value	Electrical angle [rad/s]
g_f4_slip_speed_rad	float32	Slip speed	Electrical angle [rad/s]
g_f4_slip_k	float32	Slip speed gain	
g_f4_speed_rad	float32	Calculated speed value	Electrical angle [rad/s]
g_f4_ref_speed_rad_pi	float32	Command value for speed PI control	Electrical angle [rad/s]
g_f4_ref_speed_rad	float32	Rotational speed command value	Electrical angle [rad/s]
g_f4_angle_rad	float32	Rotor interlinkage flux phase	[rad]
g_f4_refu	float32	U-phase voltage command value	[V]
g_f4_refv	float32	V-phase voltage command value	[V]
g_f4_refw	float32	W-phase voltage command value	[V]
g_f4_inv_limit	float32	Phase voltage limit value	[V]
g_f4_speed_lpf_k	float32	Speed LPF gain	
g_f4_current_lpf_k	float32	Current LPF gain	
g_f4_offset_lpf_k	float32	Current offset value LPF gain	
g_u1_direction	uint8	Rotational direction command	0: CW 1: CCW
g_u1_enable_write	uint8	Variable for use by the user interface of the ICS	
g_u2_cnt_adjust	uint16	Counter for use in current offset calculation	
g_u1_flag_id_ref	uint8	γ axis current command management flag	0: γ axis current increases 1: γ axis current is constant 2: γ axis current is constant (with speed control)
g_f4_temp_speed_rad	float32	Variable for storing the speed value	Electrical angle [rad/s]
g_f4_temp_ref_speed_rad	float32	Variable for storing rotational speed command value	Electrical angle [rad/s]
g_f4_angle_compensation	float32	Phase compensation constant	
g_f4_offset_calc_time	float32	Current offset calculation time	[ms]
g_f4_voltage_drop	float32	Voltage drop correction threshold	[V]
g_f4_voltage_drop_k	float32	Voltage drop correction gain	

Table 3.6 List of Variables (3/3)

Variable Name	Type	Description	Remarks
g_f4_modu	float32	U phase modulation factor	
g_f4_modv	float32	V phase modulation factor	
g_f4_modw	float32	W phase modulation factor	
rotor_speed	MTR_PI_CTRL	Rotor speed PI control structure	
id_ACR	MTR_PI_CTRL	γ axis current PI control structure	
g_f4_kp_speed	float32	Speed PI control proportional gain	
g_f4_ki_speed	float32	Speed PI control integral gain	
g_f4_lim_iq	float32	δ axis current PI control limit value	[A]
g_f4_ilim_iq	float32	δ axis current PI control integral term limit value	[A]
g_f4_max_speed_rad	float32	Maximum rotational speed command value	[rad/s]
g_f4_min_speed_rad	float32	Minimum rotational speed command value	[rad/s]
g_u1_dir_buff	uint8	Variable for storing the commanded rotational direction	0: CW 1: CCW
g_f4_id_ref_buff	float32	Variable for storing γ axis current command	[A]
g_f4_iq_ref_buff	float32	Variable for storing δ axis current command	[A]
g_u1_flag_iq_ref	uint8	δ axis current command management flag	0: δ axis current = 0 1: Speed PI output
g_u1_flag_speed_ref	uint8	Rotational speed command management flag	0: Speed = 0 1: Speed changes
mtr_p	MTR_PARAMETER	Motor parameters and control parameters	
g_u1_flag_offset_calc	uint8	Current offset calculation flag	0: Calculated at transition to boot mode 1: Calculated at transition to boot mode (first time only)
g_f4_boot_id_up_step	float32	γ axis current additional value at startup	[A]
g_f4_fluctuation_limit	float32	Speed fluctuation limit	[rad/s]
g_f4_ctrl_ref_id	float32	γ axis current command	[A]
g_u2_cnt_id_const	uint16	γ axis current flux stabilization wait time counter	
g_f4_id_const_time	float32	γ axis current flux stabilization wait time	[ms]
g_f4_accel	float32	Acceleration	[rad/s^2]
g_f4_ipm_temperature_ad	float32	IPM temperature that is converted to voltage	[V]
speed	MTR_PI_CTRL	Structure for use in speed PI control	
ics_input_buff	MTR_ICS_INPUT	Structure for the ICS user interface	
g_s2_enable_write	int16	ICS write enable flag	
ics_input	MTR_ICS_INPUT	Structure for the ICS user interface	

3.4 Structures Used in the Sensorless Vector Control Software

The table below lists the structures used in the control program.

Table 3.7 List of Structures

	Member Name	Type	Description	Remarks
MTR_PI_CTRL	f4_err	float32	Differential	
	f4_kp	float32	PI control proportional gain	
	f4_ki	float32	PI control integral gain	
	f4_limit	float32	PI control output limit value	
	f4_refi	float32	PI control integral term output value	
	f4_ilimit	float32	PI control integral term limit value	
MTR_PARAMETER	f4_mtr_rs	float32	Stator resistance	[Ω]
	f4_mtr_rr	float32	Rotor resistance	[Ω]
	f4_mtr_m	float32	Magnetizing inductance	[H]
	f4_mtr_ls	float32	Stator leakage inductance	[H]
	f4_mtr_lr	float32	Rotor leakage inductance	[H]
	f4_mtr_m_lr	float32	f4_mtr_m/f4_mtr_lr	
	f4_mtr_rr_lr	float32	f4_mtr_rr/f4_mtr_lr	
	f4_mtr_sigma	float32	1.0 - f4_mtr_m / f4_mtr_ls * f4_mtr_m_lr	
	f4_mtr_ls_sigma	float32	f4_mtr_ls * f4_mtr_sigma	
MTR_ICS_INPUT	s2_ref_speed	int16	Rotational speed command	Mechanical angle [rpm]
	s2_direction	int16	Rotational direction	0: CW 1: CCW
	f4_kp_speed	float32	Speed PI control proportional gain	
	f4_ki_speed	float32	Speed PI control integral gain	
	f4_kp_iq	float32	δ axis current PI control proportional gain	
	f4_ki_iq	float32	δ axis current PI control integral gain	
	f4_speed_lpf_k	float32	Speed LPF gain	
	f4_current_lpf_k	float32	Current LPF gain	
	f4_mtr_rs	float32	Stator resistance	[Ω]
	f4_mtr_rr	float32	Rotor resistance	[Ω]
	f4_mtr_m	float32	Magnetizing inductance	[H]
	f4_mtr_lls	float32	Stator leakage inductance	[H]
	f4_mtr_llr	float32	Rotor leakage inductance	[H]
	f4_offset_lpf_k	float32	Current offset value LPF gain	
	s2_max_speed	int16	Maximum speed	Mechanical angle [rpm]
	s2_min_speed	int16	Minimum speed	Mechanical angle [rpm]
	f4_ctrl_ref_id	float32	γ axis current command	[A]
	f4_boot_id_up_time	float32	Rise time at γ axis current startup	[ms]
	f4_id_const_time	float32	γ axis current flux stabilization wait time	[ms]
	f4_accel	float32	Rotational speed command acceleration/deceleration step size	
	f4_fluctuation_limit	float32	Speed fluctuation limit	[rad/s]
	f4_delay	float32	Voltage output delay compensation coefficient	
	f4_offset_calc_time	float32	Current offset adjustment time	[ms]
	f4_voltage_drop	float32	Voltage drop correction threshold	[V]
	f4_voltage_drop_k	float32	Voltage drop correction gain	

3.5 Sensorless Vector Control Software Macros

The table below lists the macro definitions used in this control program.

Table 3.8 List of Macro Definitions (1/11)

File Name	Macro Name	Definition Value	Remarks
main.h	MAX_SPEED	CP_MAX_SPEED_RPM	Maximum value of the speed command (mechanical angle) [rpm]
	MIN_SPEED	CP_MIN_SPEED_RPM	Minimum value of the speed command (mechanical angle) [rpm]
	IQ_PI_KP	CP_IQ_PI_KP	δ axis current PI control proportional gain
	IQ_PI_KI	CP_IQ_PI_KI	δ axis current PI control integral gain
	SPEED_PI_KP	CP_SPEED_PI_KP	Speed PI control proportional gain
	SPEED_PI_KI	CP_SPEED_PI_KI	Speed PI control integral gain
	SPEED_LPF_K	CP_SPEED_LPF_K	Speed LPF gain
	CURRENT_LPF_K	CP_CURRENT_LPF_K	Current LPF gain
	STATOR_RESISTANCE	MP_STATOR_RESISTANCE	Stator resistance [Ω]
	ROTOR_RESISTANCE	MP_ROTOR_RESISTANCE	Rotor resistance [Ω]
	MUTUAL_INDUCTANCE	MP_MUTUAL_INDUCTANCE	Magnetizing inductance [H]
	STATOR_LEAKAGE_INDUCTANCE	MP_STATOR_LEAKAGE_INDUCTANCE	Stator leakage inductance [H]
	ROTOR_LEAKAGE_INDUCTANCE	MP_ROTOR_LEAKAGE_INDUCTANCE	Rotor leakage inductance [H]
	OFFSET_LPF_K	CP_OFFSET_LPF_K	Current offset value LPF gain
	CTRL_REF_ID	CP_CTRL_REF_ID	γ axis current command [A]
	BOOT_ID_UP_TIME	CP_BOOT_ID_UP_TIME	Rise time at γ axis current startup [ms]
	ID_CONST_TIME	CP_ID_CONST_TIME	γ axis current flux stabilization wait time [ms]
	ACCEL_MODE0	CP_ACCEL_MODE0	Acceleration
	FLUCTUATION_LIMIT	CP_FLUCTUATION_LIMIT	Speed fluctuation limit
	DELAY	CP_DELAY	Voltage output delay compensation coefficient
	OFFSET_CALC_TIME	CP_OFFSET_CALC_TIME	Current offset calculation time [ms]
	VOLTAGE_DROP	CP_VOLTAGE_DROP	Voltage drop correction threshold [V]
	VOLTAGE_DROP_K	CP_VOLTAGE_DROP_K	Voltage drop correction gain
	POLE_PAIRS	MP_POLE_PAIRS	Constant used for pole pairs count correction
	M_CW	0	Rotational direction
	M_CCW	1	
	ICS_INT_LEVEL	6	ICS interrupt priority level
	SW_ON	1	Active-high
	SW_OFF	0	

Table 3.9 List of Macro Definitions (2/11)

File Name	Macro Name	Definition Value	Remarks
main.h	VR1_SCALING	4000.0/4095.0	Rotational speed command scale
	APL_MODE_STOP	0	Stopped state
	APL_MODE_RUN	1	Running state
	APL_MODE_TIMER_RUN	2	Running in timer-controlled operation mode
	APL_MODE_TIMER_WAIT	3	Waiting for timer-controlled operation mode
	APL_MODE_ERROR	4	Error state
	APL_NO_ERROR	0x00	No error occurred
	APL_OVERCURRENT_ERROR	0x01	Overload error (overcurrent)
	APL_UNDERCURRENT_ERROR	0x02	Dry running error (undercurrent)
	APL_CONTROLER_ERROR	0x04	Motor control error
	APL_OVERCURRENT_LIMIT	6.0	Threshold value for the resultant current for overload determination [A]
	APL_UNDERCURRENT_LIMIT	1.0	Threshold value for the resultant current for dry running determination [A]
	APL_CHK_START_UC_RPM	2000	Threshold value for the rotational speed for dry running determination (mechanical angle) [rpm]
	APL_ERR_CHK_START_TIME	1.0	Wait time before starting application error determination [s]
	APL_ERR_CHK_START_COUNT	APL_ERR_CHK_START_TIME * 10	Counting for performing error determination (multiplication value for the count interval of 100 ms)
	APL_TIMER_RUN_TIME	10	Run time in timer-controlled operation mode [s]
	APL_TIMER_WAIT_TIME	10	Wait time before entering timer-controlled operation mode [s]
	APL_TIMER_RUN_COUNT	APL_TIMER_RUN_TIME * 10	Counting for run time in timer-controlled operation mode
	APL_TIMER_WAIT_COUNT	APL_TIMER_WAIT_TIME * 10	Counting for wait time before entering timer-controlled operation mode

Table 3.10 List of Macro Definitions (3/11)

File Name	Macro Name	Definition Value	Remarks
mtr_api.h	API_MAX_SPEED	CP_MAX_SPEED_RPM	Maximum value of the speed command (mechanical angle) [rpm]
	API_MIN_SPEED	CP_MIN_SPEED_RPM	Minimum value of the speed command (mechanical angle) [rpm]
	API_SUCCESS	0	Normal termination
	API_ERR_FAILED	-1	Abnormal termination
	API_CONVERT_LIMIT	1	Terminated because the rotational speed command value reached a preliminary specified limitation.

Table 3.11 List of Macro Definitions (4/11)

File Name	Macro Name	Definition Value	Remarks
motor_parameter.h	MP_POLE_PAIRS	1	Pole pairs count
	MP_STATOR_RESISTANCE	5.15	Stator resistance [Ω]
	MP_ROTOR_RESISTANCE	3.14	Rotor resistance [Ω]
	MP_MUTUAL_INDUCTANCE	0.10	Magnetizing inductance [H]
	MP_STATOR_LEAKAGE_INDUCTANCE	0.0088	Stator leakage inductance [H]
	MP_INDUCTANCE	0.0108	Rotor leakage inductance [H]

Table 3.12 List of Macro Definitions (5/11)

File Name	Macro Name	Definition Value	Remarks
mtr_ctrl_rx13t48_t1102.h	MTR_PWM_TIMER_FREQ	160.0	PWM timer count frequency [MHz]
	MTR_CARRIER_FREQ	16.0	Carrier frequency [kHz]
	MTR_DEADTIME	2.5	Dead time [μs]
	MTR_DEADTIME_SET	MTR_DEADTIME * MTR_PWM_TIMER_FREQ	Dead time setting
	MTR_AD_FREQ	40.0	A/D converter operating frequency [MHz]
	MTR_AD_SAMPLING_CYCLE	45.0	A/D conversion cycle count
	MTR_AD_SAMPLING_TIME	MTR_AD_SAMPLING_CYCLE / MTR_AD_FREQ	A/D conversion time [μs]
	MTR_AD_TIME_SET	MTR_PWM_TIMER_FREQ * MTR_AD_SAMPLING_TIME	Setting used to assure the A/D conversion time
	MTR_CARRIER_SET	(MTR_PWM_TIMER_FREQ * 1000 / MTR_CARRIER_FREQ / 2) + MTR_DEADTIME_SET	Carrier setting
	MTR_HALF_CARRIER_SET	MTR_CARRIER_SET / 2	Carrier setting (intermediate value)
	MTR_PORT_UP	PORT7.PODR.BIT.B1	U phase (positive phase) output port
	MTR_PORT_UN	PORT7.PODR.BIT.B4	U phase (negative phase) output port
	MTR_PORT_VP	PORT7.PODR.BIT.B2	V phase (positive phase) output port
	MTR_PORT_VN	PORT7.PODR.BIT.B5	V phase (negative phase) output port
	MTR_PORT_WP	PORT7.PODR.BIT.B3	W phase (positive phase) output port
	MTR_PORT_WN	PORT7.PODR.BIT.B6	W phase (negative phase) output port
	MTR_PORT_LED1	PORTD.PODR.BIT.B6	LED1 output port
	MTR_PORT_LED2	PORTD.PODR.BIT.B4	LED2 output port
	MTR_LED_ON	0	Low active
	MTR_LED_OFF	1	
	MTR_INPUT_V	220 * 1.41421356	Power supply voltage [V]
	MTR_IC_GATE_ON_V	MTR_INPUT_V * 0.8	Power supply voltage × 80% [V]
	MTR_HALF_VDC	MTR_INPUT_V/2.0	Power supply voltage / 2 [V]
	MTR_ADC_SCALING	0x7FF	ADC offset adjustment constant
	MTR_CURRENT_SCALING	100.0f/4095.0	Resolution for use in A/D conversion of the measured current
	MTR_VDC_SCALING	686.0f/4095.0	Resolution for use in A/D conversion of the measured inverter bus voltage
	MTR_IPMTEMPERATURE_SCALING	5.0f/4095.0	Resolution for use in A/D conversion of the measured IPM temperature
	MTR_OVERCURRENT_LIMIT	7.0	Upper limit of the current value [A]
	MTR_OVERVOLTAGE_LIMIT	420.0	Upper limit of the voltage value [V]
	MTR_UNDERVOLTAGE_LIMIT	0.0	Lower limit of the voltage value [V]
	MTR_OVERIPMTEMPERATURE_LIMIT	3	Upper limit of the IPM temperature [V]
	MTR_PORT_IC_GATE	PORTB.PODR.BIT.B2	Inrush current prevention circuit ports
	MTR_IC_GATE_ON	1	
	MTR_PORT_SW1	PORTB.PIDR.BIT.B5	SW1 input port
	MTR_PORT_SW2	PORTB.PIDR.BIT.B4	SW2 input port
	MTR_PORT_SW3	PORT7.PIDR.BIT.B0	SW3 input port

Table 3.13 List of Macro Definitions (6/11)

File Name	Macro Name	Definition Value	Remarks
mtr_3im_less_foc.h	MTR_INT_DECIMATION	1	Interrupt decimation count
	MTR_CTRL_PERIOD	(MTR_INT_DECIMATION + 1)/ (MTR_CARRIER_FREQ * 1000)	Control period [s]
	MTR_CONTROL_FREQ	(MTR_CARRIER_FREQ * 1000)/ (MTR_INT_DECIMATION + 1)	Control frequency [Hz]
	MTR_POLE_PAIRS	MP_POLE_PAIRS	Pole pairs count
	MTR_RS	MP_STATOR_RESISTANCE	Stator resistance [Ω]
	MTR_RR	MP_ROTOR_RESISTANCE	Rotor resistance [Ω]
	MTR_M	MP_MUTUAL_INDUCTANCE	Magnetizing inductance [H]
	MTR_LLS	MP_STATOR_LEAKAGE_INDUCTANCE	Stator leakage inductance [H]
	MTR_LLR	MP_ROTOR_LEAKAGE_INDUCTANCE	Rotor leakage inductance [H]
	MTR_LS	MTR_M + MTR_LLS	
	MTR_LR	MTR_M + MTR_LLR	
	MTR_M_LR	MTR_M / MTR_LR	
	MTR_RR_LR	MTR_RR / MTR_LR	
	MTR_SIGMA	1.0f - MTR_M / MTR_LS * MTR_M_LR	
	MTR_LS_SIGMA	MTR_LS * MTR_SIGMA	
	MTR_TWOPI	2 * 3.14159265	2π
	MTR_TWOPI_3	MTR_TWOPI/3	$2\pi / 3$
	MTR_SQRT_2	1.41421356f	$\sqrt{2}$
	MTR_SQRT_3	1.7320508f	$\sqrt{3}$
	MTR_SQRT_2_3	0.81649658f	$\sqrt{(2/3)}$
	MTR_RPM_RAD	MTR_TWOPI/60	$2\pi / 60$
	MTR_ID_PI_KP	CP_ID_PI_KP	γ axis current PI control proportional gain
	MTR_ID_PI_KI	CP_ID_PI_KI	γ axis current PI control integral gain
	MTR_IQ_PI_KP	CP_IQ_PI_KP	δ axis current PI control proportional gain
	MTR_IQ_PI_KI	CP_IQ_PI_KI	δ axis current PI control integral gain
	MTR_SPEED_PI_KP	CP_SPEED_PI_KP	Speed PI control proportional gain
	MTR_SPEED_PI_KI	CP_SPEED_PI_KI	Speed PI control integral gain
	MTR_SPEED_LPF_K	CP_SPEED_LPF_K	Speed LPF gain
	MTR_CURRENT_LPF_K	CP_CURRENT_LPF_K	Current LPF gain
	MTR_OFFSET_LPF_K	CP_OFFSET_LPF_K	Current offset value LPF gain
	MTR_LIMIT_ID	3.5	γ axis current PI control output limit value [A]
	MTR_I_LIMIT_ID	3.5	γ axis current PI control integral term limit value [A]
	MTR_LIMIT_IQ	3.5	Speed PI control output limit value [A]
	MTR_I_LIMIT_IQ	3.5	Speed PI control integral term limit value [A]

Table 3.14 List of Macro Definitions (7/11)

File Name	Macro Name	Definition Value	Remarks
mtr_3im_less_foc.h	MTR_MAX_SPEED_RPM	CP_MAX_SPEED_RPM	Maximum speed (mechanical angle) [rpm]
	MTR_MAX_SPEED_RAD	MTR_MAX_SPEED_RPM * MTR_POLE_PAIRS * MTR_TWOPI/60	Maximum speed (electrical angle) [rad/s]
	MTR_MIN_SPEED_RPM	CP_MIN_SPEED_RPM	Minimum speed (mechanical angle) [rpm]
	MTR_MIN_SPEED_RAD	MTR_MIN_SPEED_RPM * MTR_POLE_PAIRS * MTR_TWOPI/60	Minimum speed (electrical angle) [rad/s]
	MTR_CHG_OPEN_RPM	CP_CHG_OPEN_RPM	The value at which the vector control is switched to open loop mode (mechanical angle) [rpm]
	MTR_CHG_OPEN_RAD	MTR_CHG_OPEN_RPM * MTR_POLE_PAIRS * MTR_TWOPI/60	The value at which the vector control is switched to open loop mode (electrical angle) [rad/s]
	MTR_CHG_FOC_RPM	CP_CHG_FOC_RPM	The value at which the vector control is switched to sensorless mode (mechanical angle) [rpm]
	MTR_CHG_FOC_RAD	MTR_CHG_FOC_RPM * MTR_POLE_PAIRS * MTR_TWOPI/60	The value at which the vector control is switched to sensorless mode (electrical angle) [rad/s]
	MTR_SPEED_LIMIT	MTR_MAX_SPEED_RAD * 1.3	Speed limit value [rad/s]
	MTR_LIMIT_ROTOR_SPEED_RAD	MTR_MAX_SPEED_RAD * 1.2	δ axis current PI control output limit value [rad/s]
	MTR_I_LIMIT_ROTOR_SPEED_RAD	MTR_MAX_SPEED_RAD * 1.2	δ axis current PI control integral term limit value [rad/s]
	MTR_CTRL_REF_ID	CP_CTRL_REF_ID	γ axis current command
	MTR_BOOT_ID_UP_TIME	CP_BOOT_ID_UP_TIME	Rise time at γ axis current startup [ms]
	MTR_BOOT_ID_UP_STEP	CP_CTRL_REF_ID/MTR_BOOT_ID_UP_TIME	Rise step size at γ axis current startup
	MTR_ID_CONST_TIME	CP_ID_CONST_TIME	γ axis current flux stabilization wait time [ms]
	MTR_ACCEL_MODE0	CP_ACCEL_MODE0	Acceleration
	MTR_FLUCTUATION_LIMIT	CP_FLUCTUATION_LIMIT	Speed fluctuation limit [rad/s]
	MTR_DELAY	CP_DELAY	Phase compensation constant
	MTR_ANGLE_COMPENSATION	MTR_DELAY * MTR_CTRL_PERIOD	
	MTR_OFFSET_CALC_TIME	CP_OFFSET_CALC_TIME	Current offset calculation time [ms]
	MTR_VOLTAGE_DROP	CP_VOLTAGE_DROP	Voltage drop correction threshold [V]
	MTR_VOLTAGE_DROP_K	CP_VOLTAGE_DROP_K	Voltage drop correction gain

Table 3.15 List of Macro Definitions (8/11)

File Name	Macro Name	Definition Value	Remarks
mtr_3im_less_foc.h	MTR_EVERY_TIME	0	Current value calculation
	MTR_ONE_TIME	1	Current offset value calculation (first time only)
	MTR_CW	0	Rotational direction
	MTR_CCW	1	
	MTR_FLG_CLR	0	Flag management
	MTR_FLG_SET	1	
	MTR_ID_UP	0	γ axis current increases
	MTR_ID_CONST	1	γ axis current is constant
	MTR_ID_CONST_CTRL	2	Normal operation
	MTR_IQ_ZERO	0	δ axis current is 0
	MTR_IQ_SPEED_PI_OUTPUT	1	Normal operation
	MTR_SPEED_ZERO	0	Speed command is 0
	MTR_SPEED_CHANGE	1	Speed command change
	MTR_BOOT_MODE	0x00	Boot mode
	MTR_START_MODE	0x01	Start mode
	MTR_CTRL_MODE	0x02	Control mode
	MTR_OPENLOOP_MODE	0x01	Open loop mode
	MTR_LESS_FOC_MODE	0x02	Sensorless vector control mode
	MTR_OVER_CURRENT_ERROR	0x01	Overcurrent error
	MTR_OVER_VOLTAGE_ERROR	0x02	Overvoltage error
	MTR_OVER_SPEED_ERROR	0x03	Excessive speed error
	MTR_TIMEOUT_ERROR	0x04	Timeout error
	MTR_UNDER_VOLTAGE_ERROR	0x07	Low voltage error
	MTR_OVER_IPMTEMPERATURE_ERROR	0x08	IPM temperature abnormality error
	MTR_UNKNOWN_ERROR	0xff	Undefined error
	MTR_MODE_STOP	0x00	Stopped state
	MTR_MODE_RUN	0x01	Motor running state
	MTR_MODE_ERROR	0x02	Error state
	MTR_SIZE_STATE	3	Number of states
	MTR_EVENT_STOP	0x00	Motor stop event
	MTR_EVENT_RUN	0x01	Motor start event
	MTR_EVENT_ERROR	0x02	Motor error event
	MTR_EVENT_RESET	0x03	Motor reset event
	MTR_SIZE_EVENT	4	Number of events

Table 3.16 List of Macro Definitions (9/11)

File Name	Macro Name	Definition Value	Remarks
control_parameter.h	CP_ID_PI_KP	1.9	γ axis current PI control proportional gain
	CP_ID_PI_KI	0.08	γ axis current PI control integral gain
	CP_IQ_PI_KP	1.9	δ axis current PI control proportional gain
	CP_IQ_PI_KI	0.08	δ axis current PI control integral gain
	CP_SPEED_PI_KP	0.01	Speed PI control proportional gain
	CP_SPEED_PI_KI	0.00013	Speed PI control integral gain
	CP_SPEED_LPF_K	0.3	Speed LPF gain
	CP_CURRENT_LPF_K	1.0	Current LPF gain
	CP_OFFSET_LPF_K	0.1	Current offset value LPF gain
	CP_MAX_SPEED_RPM	3600	Maximum speed (mechanical angle) [rpm]
	CP_MIN_SPEED_RPM	50	Minimum speed (mechanical angle) [rpm]
	CP_CTRL_REF_ID	2.0	γ axis current command
	CP_BOOT_ID_UP_TIME	1000.0	Rise time at γ axis current startup [ms]
	CP_ID_CONST_TIME	500.0	γ axis current flux stabilization wait time [ms]
	CP_ACCEL_MODE0	0.1	Acceleration during start mode [rad/s^2]
	CP_FLUCTUATION_LIMIT	200.0	Speed fluctuation limit [rad/s]
	CP_DELAY	1.0	Phase compensation constant
	CP_OFFSET_CALC_TIME	256	Current offset calculation time [ms]
	CP_VOLTAGE_DROP	8.0	Voltage drop correction threshold [V]
	CP_VOLTAGE_DROP_K	100.0	Voltage drop correction gain
	CP_CHG_OPEN_RPM	200	The value at which the vector control is switched to open loop mode (mechanical angle) [rpm]
	CP_CHG_FOC_RPM	300	The value at which the vector control is switched to sensorless mode (mechanical angle) [rpm]

Table 3.17 List of Macro Definitions (10/11)

File Name	Macro Name	Definition Value	Remarks
r_init_clock.h	B_NOT_USE	0	Not in use
	B_USE	1	In use
	B_USE_PLL_MAIN	2	Use the PLL clock. (Clock source: the main clock)
	B_USE_PLL_HOCO	3	Use the PLL clock. (Clock source: HOCO)
	SEL_MAIN	B_USE	Used in selecting oscillation or stopping of the main clock. B_USE: In use (the main clock oscillates) B_NOT_USE: Not in use (the main clock is stopped)
	MAIN_CLOCK_Hz	8000000L	Number of frequencies of the main clock oscillator (Hz)
	REG_MOFCR	30h	For setting the driving ability of the main clock oscillator (the value to be set in the MOFCR register)
	REG_MOSCWTCR	53h	The value set in the wait control register for the main clock
	REG_PLLCR *1,*2	0700h	Setting of the division ratio and multiplication factor for the PLL (the value to be set in the PLLCR register).
	SEL_PLL	B_USE_PLL_HOCO	Used in selecting oscillation or stopping of the PLL clock. B_USE_PLL_MAIN: In use (with the main clock as the source for the PLL) B_USE_PLL_HOCO: In use (with the HOCO clock as the source for the PLL) B_NOT_USE: Not in use (the PLL clock is stopped)
	SEL_HOCO	B_USE	Used in selecting oscillation or stopping of the HOCO clock. B_USE: In use (the HOCO clock oscillates) B_NOT_USE: Not in use (the HOCO clock is stopped)

Note 1. The meanings of the values depend on the clock source selected for the system clock.
Note 2. Change the setting value of the PLLCR register as required in accord with the description of the RX13T Group User's Manual: Hardware.

Table 3.18 List of Macro Definitions (11/11)

File Name	Macro Name	Definition Value	Remarks
r_init_clock.h	CLK_HOCO	0100h	Clock source: PLL
	CLK_MAIN	0200h	Clock source: HOCO
	CLK_PLL	0400h	Clock source: The main clock
	SEL_SYSCLK	CLK_HOCO	The clock source for the system clock. CLK_PLL: PLL CLK_ HOCO: HOCO CLK_ MAIN: The main clock
	REG_SCKCR *1, *2	0000 0000h	Used in setting the division ratio for the internal clock signals (value to be set in the SCKCR register)

Note 1. The meanings of the values depend on the clock source selected for the system clock.
Note 2. Change the setting value of the SCKCR register as required in accord with the description of the RX13T Group User's Manual: Hardware.

3.6 Control Flow (Flowcharts)

3.6.1 Main Processing

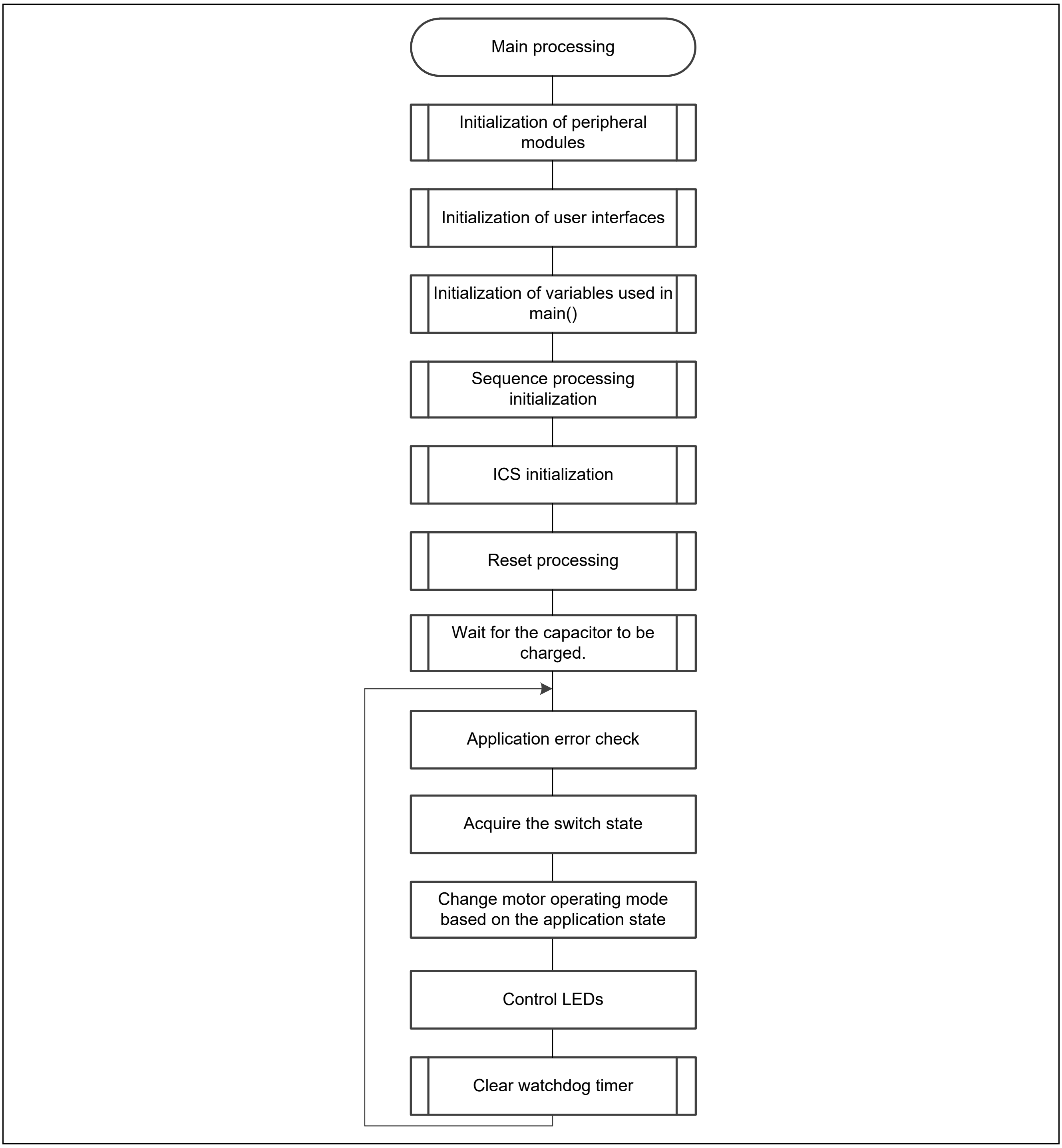

Figure 3.5 Main Processing

3.6.2 125-μs Period Interrupt Handling

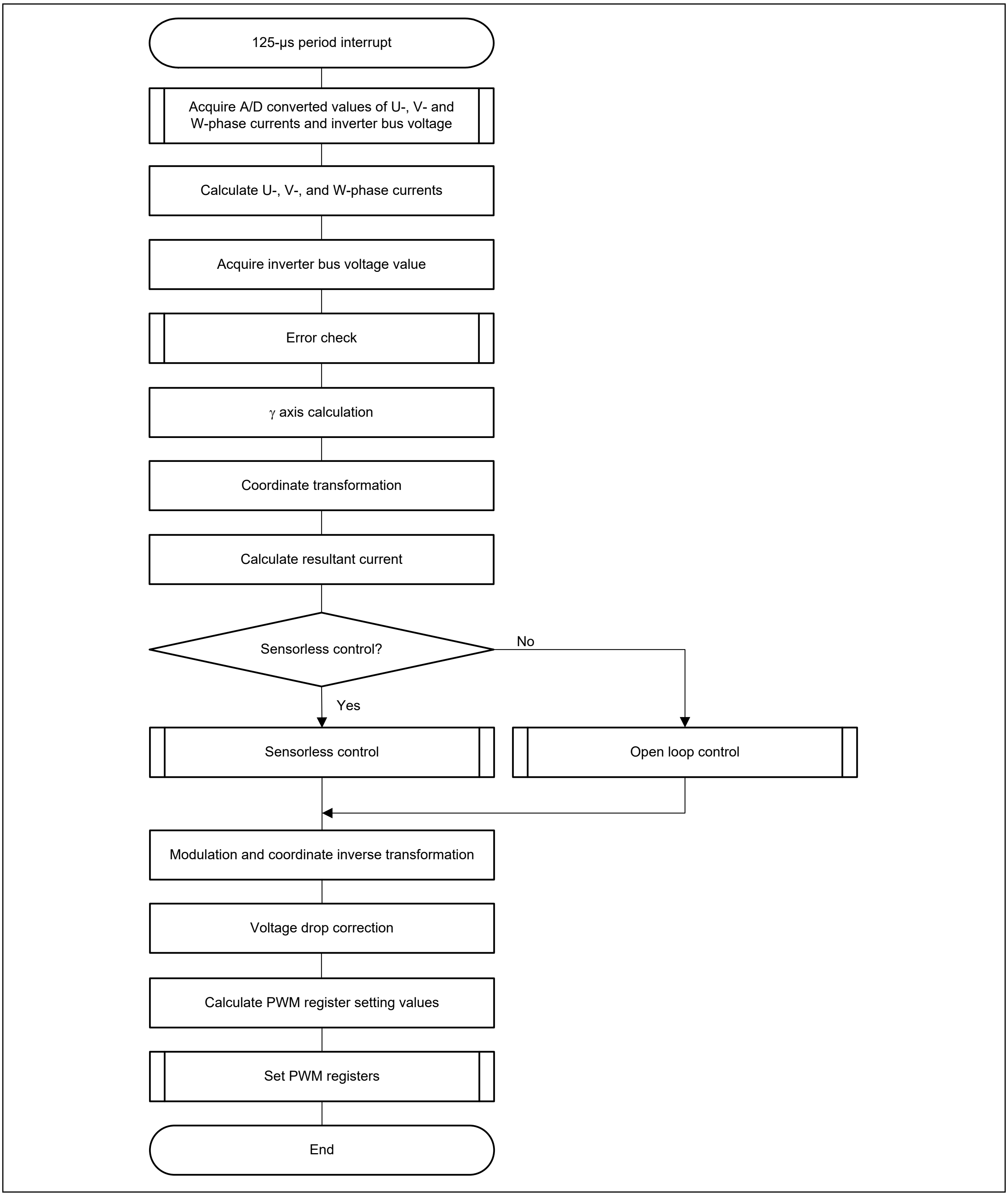

Figure 3.6 125-μs Period Interrupt Handling

3.6.3 1-ms Interrupt Handling

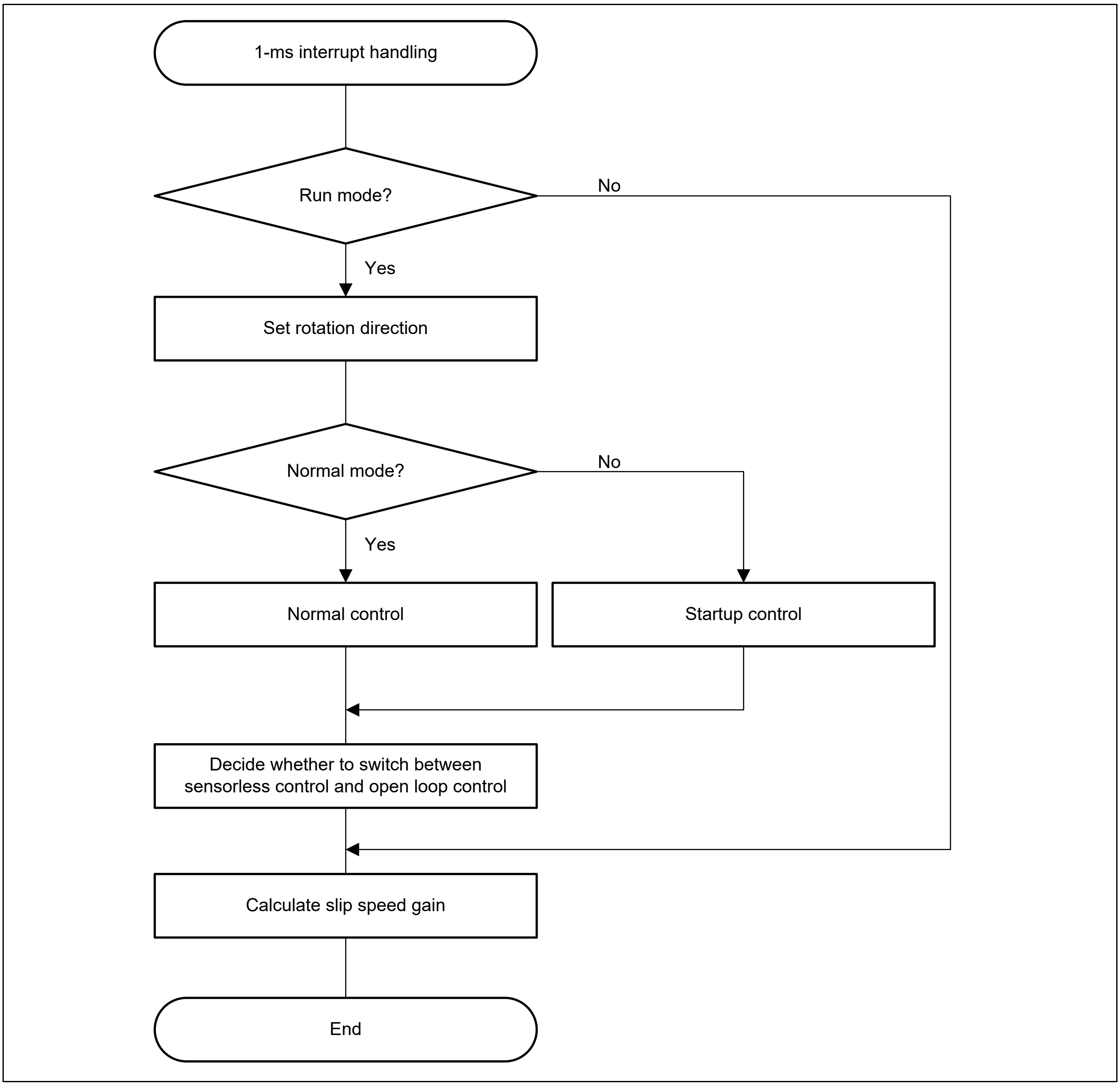

Figure 3.7 1-ms Interrupt Handling

3.6.4 100-ms Interrupt Handling

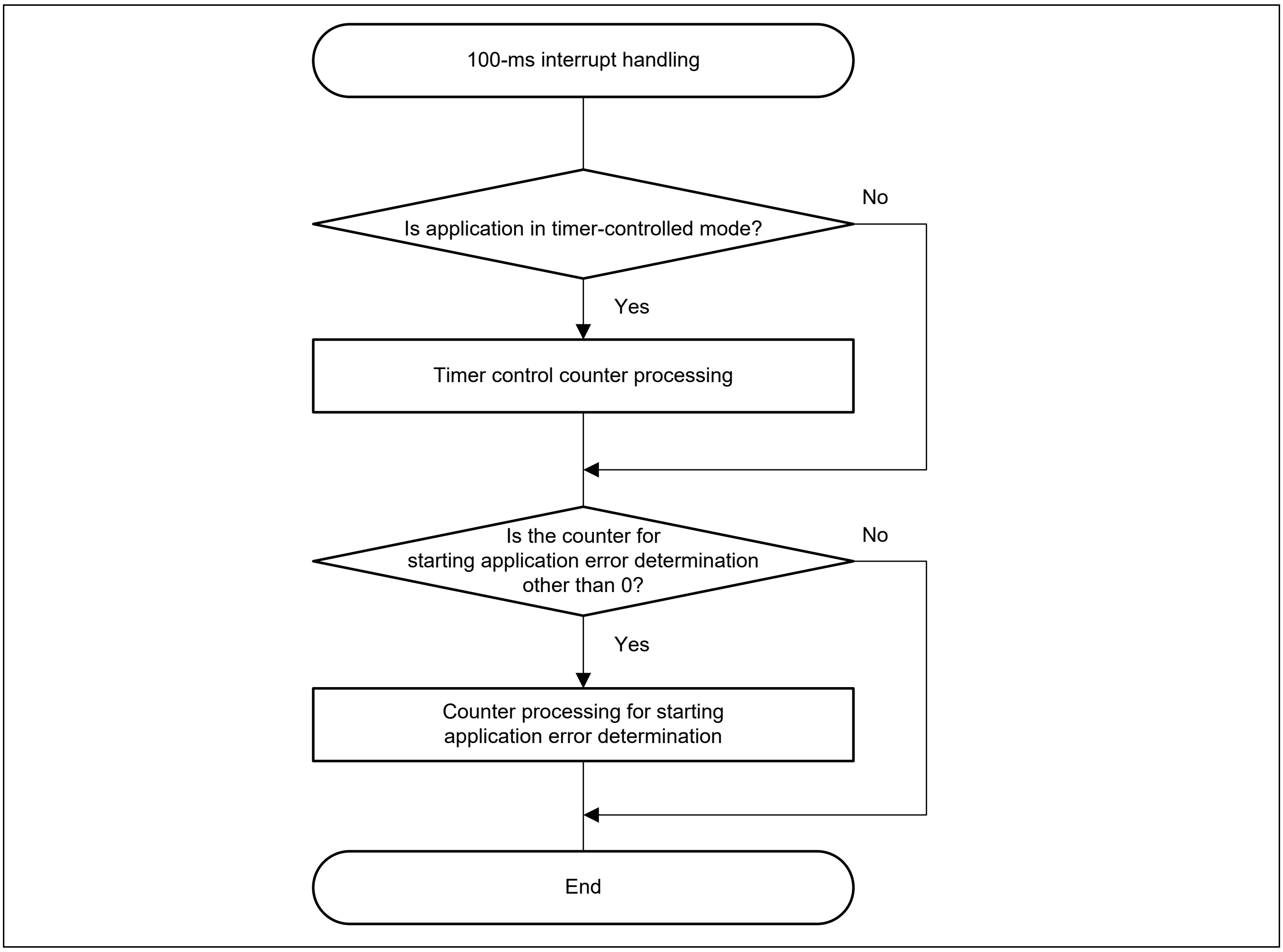

Figure 3.8 100-ms Interrupt Handling

3.6.5 Overcurrent Detection Interrupt Handling

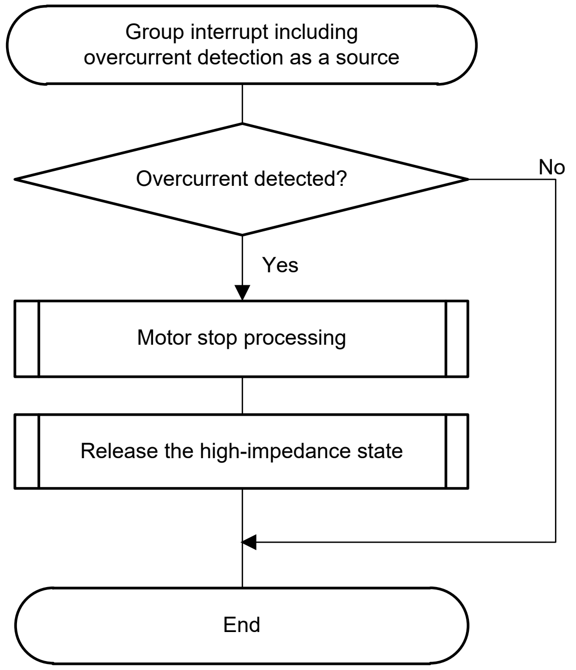

Figure 3.9 Overcurrent Detection Interrupt Handling

4. Motor Control Development Support Tool 'Renesas Motor Workbench 2.0'

4.1 Overview

'Renesas Motor Workbench 2.0' is support tool for development of motor control system. 'Renesas Motor Workbench 2.0' can be used with target software of this application note to analyze the control performance. The user interfaces of 'Renesas Motor Workbench 2.0' provide functions like rotating start/stop command, setting rotation speed command, etc…Please refer to 'Renesas Motor Workbench 2.0 User's Manual' for usage and more details.

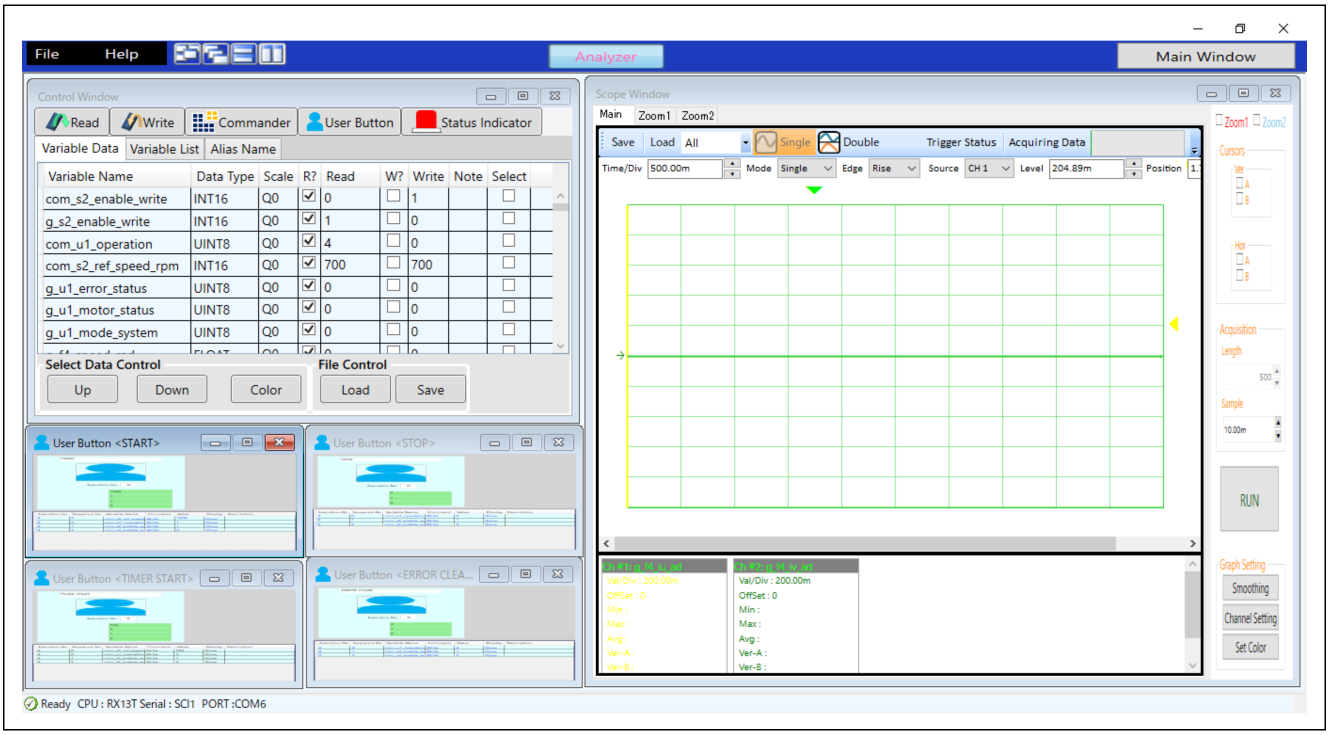

Figure 4.1 Renesas Motor Workbench-Appearance

4.2 The Usage of RMW

Following shows how to use RMW to operate a motor. The RMW screen is shown in Figure 4.2. The screen consists of three parts of the window. the control window is located in the upper left corner, the user button is located int the lower left corner, and the scope window is located in the right side. In RMW, the user button is used for basic operations, such as starting/stopping the motor. The function of each button is shown below.

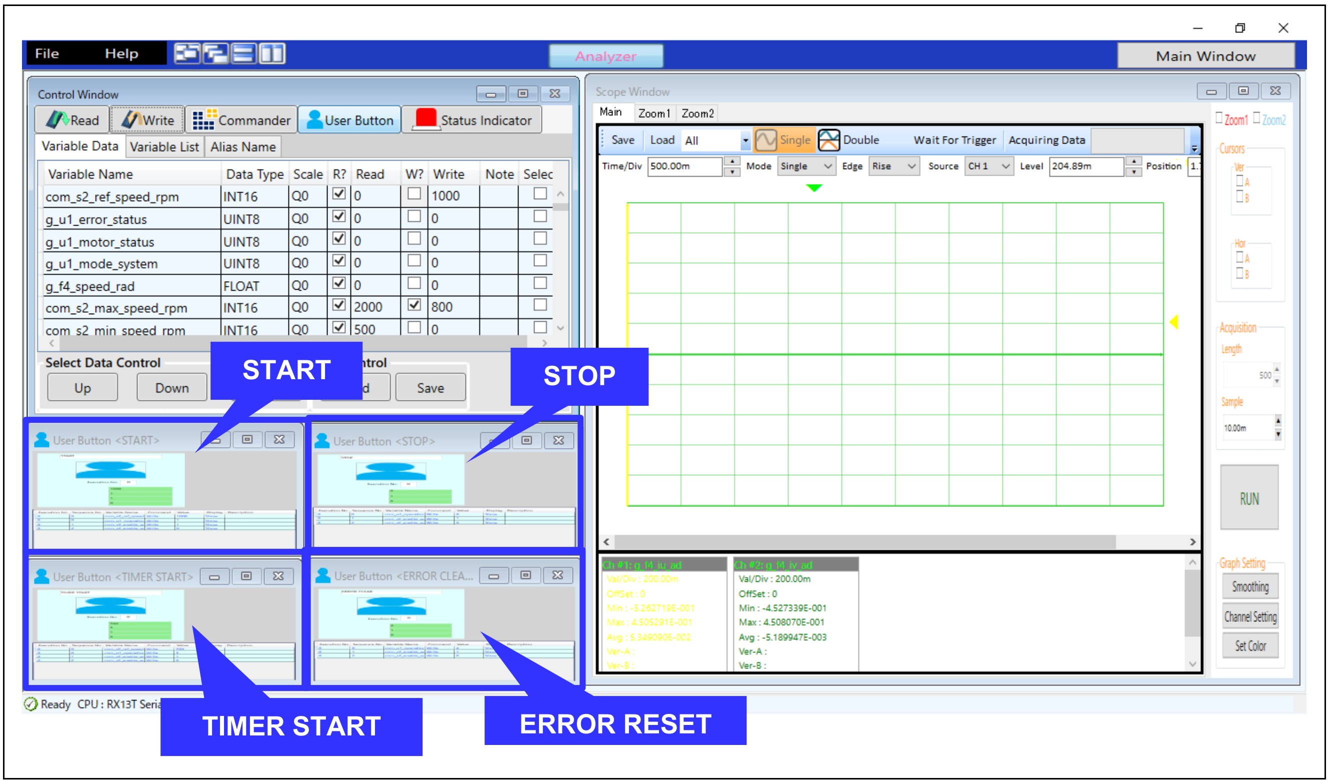

Figure 4.2 The Usage of RMW

4.2.1 START Button

The user button located in the upper left corner is the START button. use this button to set speed command value and operate the motor. Enter the speed command value in rpm unit in the value column of 'com_s2_ref_speed_rpm'. Then click the image part in the button to transition to the motor operating state and accelerate to the set speed command value. However if errors occurred, the motor will not operating.

To change the motor rotation speed in the operating state, change the speed command value and click the image part of the button to accelerate or decelerate the motor.

4.2.2 STOP Button

The user button located in the upper right corner is the STOP button. Click the image part in the button to switch to the motor stop state.

4.2.3 ERROR RESET Button

The user button located in the lower right corner is the ERROR RESET button. Click the image part in the button to resets the error status.

4.2.4 TIMER START Button

The user button located in the lower left corner is the TIMER START button. This button is used to set the speed command value and to operate the motor timer. Enter the speed command value in rpm unit in the value column of ‘com_s2_ref_speed_rpm’. Then click the image part in the button to transition to the motor operating state and repeats starting and stopping of the motor for the set period of time. However if errors occurred, the motor will not operating.

4.3 RMW Variables

Table 4.1 lists the variables for use with the RMW. Note that modifications to these variables will not be reflected in the motor control layer variables at the point these RMW variables are modified. The motor control layer variables are written and modified at the point the value of g_s2_enable_write is written to com_s2_enable_write.

Table 4.1 RMW Variables

RMW Variable	Type	Usage	Target Variable (Motor Control Layer Variable)
com_u1_operation	Uint8	Motor operation bit0: motor start/stop bit1: timer control enable/disable bit2: error reset	g_u1_drive_sw_state g_u1_timer_mode_sw_state g_u1_err_reset_sw_state
com_s2_direction	int16	Rotational direction	g_u1_dir_buff
com_s2_ref_speed_rpm	int16	Speed command	g_f4_ref_speed_rad
com_f4_kp_speed	float32	Speed PI control proportional gain	g_f4_kp_speed
com_f4_ki_speed	float32	Speed PI control integral gain	g_f4_ki_speed
com_f4_kp_iq	float32	δ axis current PI control proportional gain	g_f4_kp_iq
com_f4_ki_iq	float32	δ axis current PI control integral gain	g_f4_ki_iq
com_f4_speed_lpf_k	float32	Speed LPF gain	g_f4_speed_lpf_k
com_f4_current_lpf_k	float32	Current LPF gain	g_f4_current_lpf_k
com_f4_mtr_rs	float32	Stator resistance	mtr_p.f4_mtr_rs
com_f4_mtr_rr	float32	Rotor resistance	mtr_p.f4_mtr_rr
com_f4_mtr_m	float32	Magnetizing inductance	mtr_p.f4_mtr_m
com_f4_mtr_lls	float32	Stator leakage inductance	mtr_p.f4_mtr_ls
com_f4_mtr_llr	float32	Rotor leakage inductance	mtr_p.f4_mtr_lr
com_f4_offset_lpf_k	float32	Current offset value LPF gain	g_f4_offset_lpf_k
com_s2_max_speed_rpm	int16	Maximum speed	g_f4_max_speed_rad
com_s2_min_speed_rpm	int16	Minimum speed	g_f4_min_speed_rad
com_f4_ctrl_ref_id	float32	γ axis current command	g_f4_ctrl_ref_id
com_f4_boot_id_up_time	float32	Rise time at γ axis current startup	g_f4_boot_id_up_step
com_f4_id_const_time	float32	γ axis current flux stabilization wait time	g_f4_id_const_time
com_f4_accel	float32	Rotational speed command acceleration/deceleration step size	g_f4_accel
com_f4_fluctuation_limit	float32	Speed fluctuation limit	g_f4_fluctuation_limit
com_f4_offset_calc_time	float32	Current offset adjustment time	g_f4_offset_calc_time
com_f4_delay	float32	Voltage output delay compensation coefficient	g_f4_angle_compensation
com_f4_voltage_drop	float32	Voltage drop correction threshold	g_f4_voltage_drop
com_f4_voltage_drop_k	float32	Voltage drop correction gain	g_f4_voltage_drop_k
com_s2_enable_write	int16	Variable rewrite enable	—

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Aug 31, 20	—	First edition issued
1.10	Apr 21, 21	4	Updated version of CS+ to 8.05.00
		34	Updated Table 3.8 List of Macro Definitions(5/11)
		38	Updated Table 3.8 List of Macro Definitions(9/11)

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

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

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

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

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

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

Notice

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

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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