R01AN1848EJ0100
Rev.1.00
2013.10.09

RX62T

Motor control by RX62T microcontroller
120-degree conducting control of permanent magnetic synchronous motor with hall sensors

Summary

This application note aims at explaining a sample program for operating a three-phase permanent magnetic synchronous motor with hall sensors by 120-degree conducting method, by using functions of RX62T.

The sample program is only to be used as reference and Renesas Electronics Corporation does not guarantee the operations. Before using this sample program, carry out a thorough evaluation in a suitable environment.

Operation checking device

Operations of the sample program are checked by using the following device.

- RX62T (F562TAADFM)

Contents

1. Overview

This application note describes an example of speed control by permanent magnetic synchronous motor with hall sensors by using the microcontroller RX62T. The speed control is performed by the 120-degree conducting method.

1.1 Usage of the system

This system (sample program) enables 120-degree conducting control by using RSSK (Note 1) for motor control (Low Voltage Motor Control Starter-Kit Evaluation System and permanent magnetic synchronous motor (FH6S20E-X81 (Note 2)).

For installation and technical support of 'RSSK for motor control', contact Sales representatives and dealers of Renesas Electronics Corporation.

Notes:

1. RSSK (Renesas Solution Starter Kit) is the product of Renesas Electronics Corporation.
2. FH6S20E-X81 is the product of NIDEC SERVO CORPORATION.
   NIDEC SERVO CORPORATION. (http://www.nidec-servo.com/en/index.html)

1.2 Development environment

(1) Software development environment

Integrated development environment	CubeSuite+ (V2.00.00)

(2) Hardware environment

On-chip debug emulator	E1
Microcontroller used	RX62T (F562TAADFM)
Inverter board for motor control	Low Voltage Motor Control Starter-Kit Evaluation System (P03401-D1-002)
Permanent magnetic synchronous motor	FH6S20E-X81

2. System overview

Overview of this system is explained below.

2.1 Hardware configuration

Hardware configuration is shown below.

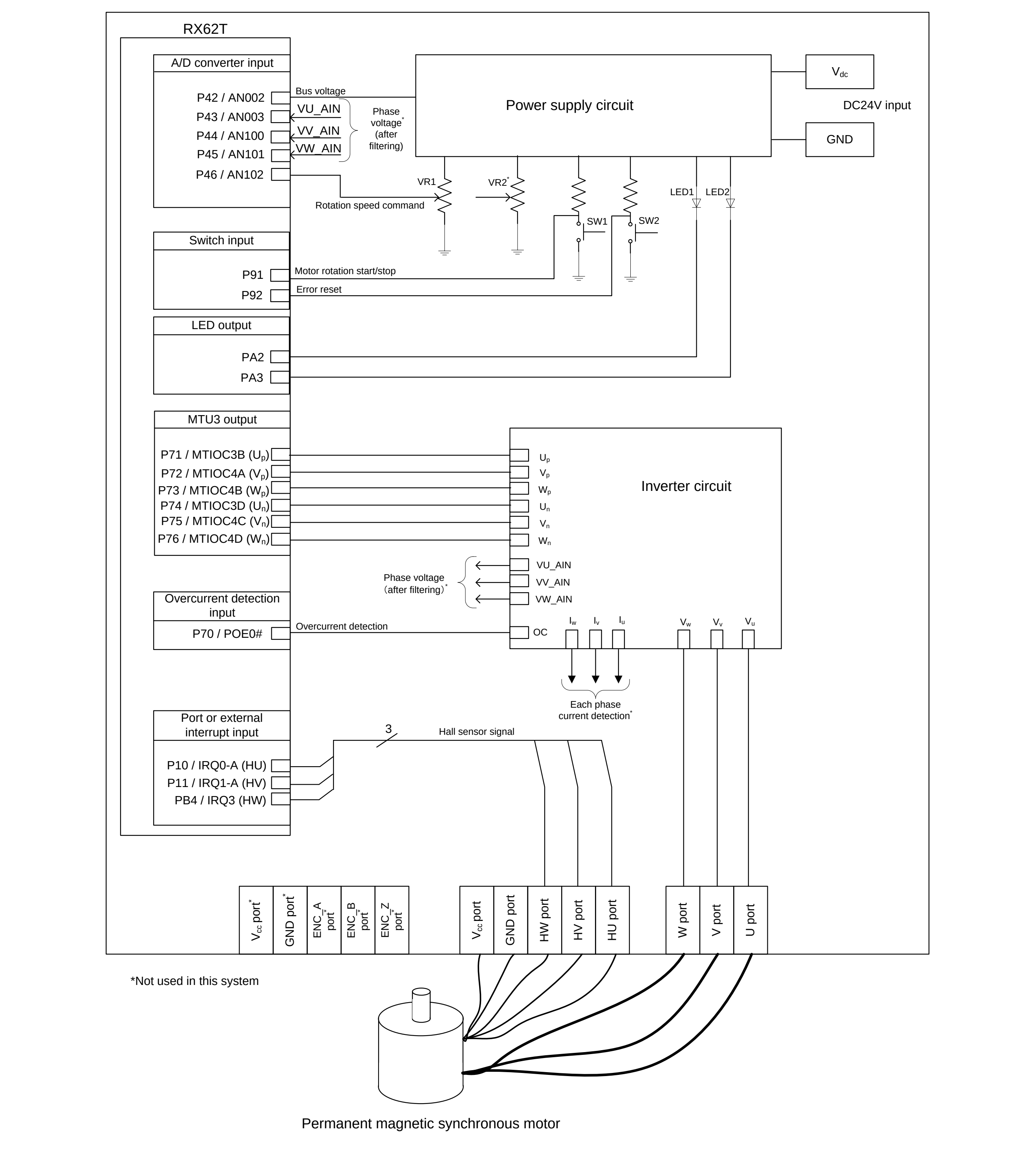


Figure 2-1 Hardware Configuration Diagram

2.2 Hardware specifications

2.2.1 User interface

Table 2-1 is a list of user interfaces of this system.

Table 2-1 User Interface

Item	Interface component	Function
Rotation speed	Variable resistance (VR1)	Rotation speed command value input (analog value)
START/STOP	Push switch (SW1)	Motor rotation start/stop command
ERROR RESET	Push switch (SW2)	Command of recovery from error status
LED1	Yellow green LED	- At the time of motor rotation: ON - At the time of stop: OFF
LED2	Yellow green LED	- At the time of error detection: ON - At the time of normal operation: OFF
RESET	Push switch (RESET)	System reset

Table 2-2 is a list of port interfaces of RX62T microcontroller of this system.

Table 2-2 Port Interface

Terminal name	Function
P42 / AN002	Inverter bus voltage measurement
P46 / AN102	For inputting rotation speed command values (analog values)
P91	START/STOP push switch
P92	ERROR RESET push switch
PA2	LED1 ON/OFF control
PA3	LED2 ON/OFF control
P10 / IRQ0-A	Hall sensor input (HU)
P11 / IRQ1-A	Hall sensor input (HV)
PB4 / IRQ3	Hall sensor input (HW)
P71 / MTIOC3B	Non-complimentary PWM output (U_p)
P72 / MTIOC4A	Non-complimentary PWM output (V_p)
P73 / MTIOC4B	Non-complimentary PWM output (W_p)
P74 / MTIOC3D	Non-complimentary PWM output (U_n)
P75 / MTIOC4C	Non-complimentary PWM output (V_n)
P76 / MTIOC4D	Non-complimentary PWM output (W_n)
P70 / POE0#	PWM emergency stop input at the time of overcurrent detection
RESET#	RESET

2.2.2 Peripheral functions

Table 2-3 is a list of peripheral functions used in this system.

Table 2-3 Peripheral Functions List

Peripheral function	Usage
12-bit A/D converter (S12ADA)	- Rotation speed command value input - Inverter bus voltage measurement
Port/external interrupt (P10/IRQ0-A, P11/IRQ1-A, PB4/IRQ3)	- Hall sensor signal input (position detection) - Hall sensor read out and external interrupt (both edges)
Compare match timer (CMT)	- 1 [ms] interval timer - Free-running timer for rotation speed measurement
Multi-function timer pulse unit 3 (MTU3)	Non-complimentary PWM output (six outputs) using the reset synchronous PWM mode
Port output enable 3 (POE3)	Sets PWM output to high impedance when an overcurrent is detected.

(1) 12-bit A/D converter

The rotation speed command value input and inverter bus voltage (V_{dc}) are measured by using the ‘12-bit A/D converter’.

The operation modes both for the Unit 0 and Unit 1 are set to the ‘Single mode’ (use software trigger).

(2) Compare match timer (CMT)

a. 1 [ms] interval timer

The channel 0 of the compare match timer (CMT) is used as 1 [ms] interval timer.

b. Free-running timer for measuring speed

The channel 1 of the compare match timer is used as free-running timer for speed measurement. Note that interrupts are not used.

(3) Multi-function timer pulse unit 3 (MTU3)

Six-phase PWM output (saw-tooth wave modulation, no dead time) is performed using the reset synchronous PWM mode.

(4) General-purpose port

The motor magnetic pole position detection signals (hall sensor signals) are input to the general-purpose port. As edges of the position signals are needed to be detected, a port which is also used for external interrupt must be selected.

(5) Port output enable 3 (POE3)

The ports executing PWM output are set to high impedance state when an overcurrent is detected (when a falling edge of the POE0# port is detected) and when an output short circuit is detected.

2.3 Software structure

2.3.1 Software file structure

The folder and file configuration of the sample program is shown below.

Table 2-4 Folder and File Configuration of the Sample Program

RX62T_RSSK_SSNS_HALL_120_ICS_CSP_V100	inc	main.h	Main function, user interface control header
		mtr_common.h	Common definition header
		mtr_ctrl_rssk.h	Board dependent processing part header
		mtr_ctrl_rx62t.h	RX62T dependent processing part header
		mtr_ssns_hall_120.h	Hall sensors use 120-degree conducting control dependent part header
	ics	ics_rx62t_uart0.h	Header for ICS
		ics.lib	ICS library
	src	main.c	Main function, user interface control
		mtr_ctrl_rssk.c	Board dependent processing part
		mtr_ctrl_rx62t.c	RX62T dependent processing part
		mtr_interrupt.c	Interrupt handler
		mtr_ssns_hall_120.c	Hall sensors use 120-degree conducting control depending part

2.3.2 Module structure

Module structure of the sample program is described below.

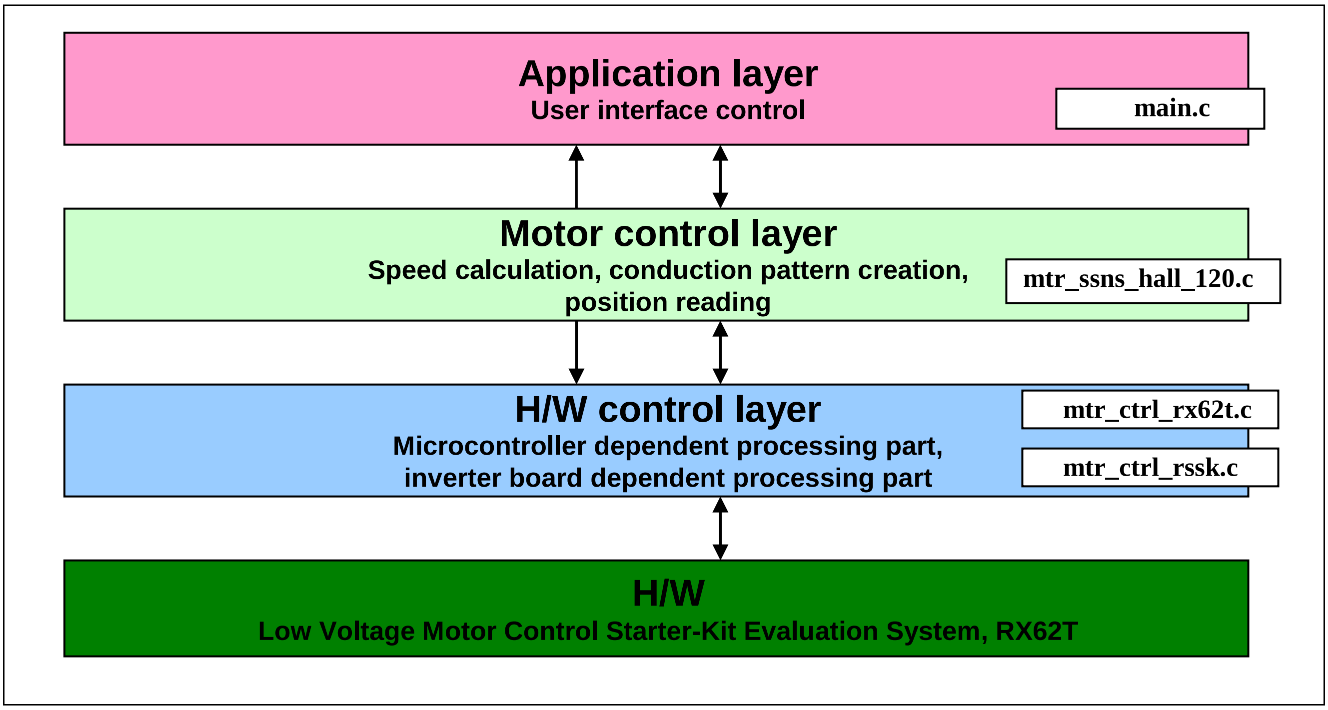


Figure 2-2 Module Structure of Sample Program

2.4 Software specifications

Table 2-5 shows basic software specifications of this system.

Table 2-5 Basic Software Specifications

Item	Content
Control method	120-degree conducting method (chopping at the first 60 degrees)
Motor rotation start/stop	Determined depending on the level of SW1 (P91) (Low: rotation start, High: stop)
Position detection of rotor magnetic pole	Position detection by hall sensors (by 60 degrees)
Carrier frequency (PWM)	20 [kHz]
Control cycle	Every time a hall sensor interrupt occurs. (Speed PI control by 3 [ms])
Rotation speed control range	600 [rpm] to 2000 [rpm] for both CW/CCW
Processing stop for protection	- Disables the motor control signal output (six outputs) under any of the following four conditions: 1. Inverter bus voltage exceeds 28 [V] (monitored per 1 [ms]). 2. Rotation speed exceeds 16000 [rpm] (electrical angle) (monitored per 1 [ms]). 3. No hall sensor interrupt is generated for 20 [ms] while the motor is driving. 4. A hall sensor error pattern is detected. - Changes the port executing the PWM output to high impedance when an overcurrent detection signal (a falling edge of the POE0# port) and output short-circuit are detected from external.

3. Motor control method

This part explains 120-degree conducting control of the permanent magnetic synchronous motor with hall sensors used in the sample program.

3.1 120-degree conducting control of the permanent magnetic synchronous motor with hall sensors

In this system, hall sensors are used to detect the position of the permanent magnet, and signals from the hall IC (hall sensor signals) are input to the microcontroller as position information.

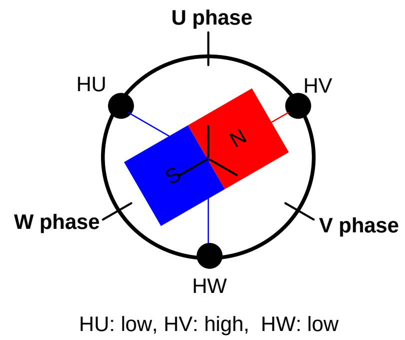


Figure 3-1 Example of Position of Hall Sensors (HU, HV, HW) and Position Signals

As shown in Figure 3-1, hall sensors are allocated every 120 degrees and the respective hall sensor signals are switched depending on direction of rotating magnetic poles. Combining these three hall sensor signals enables to obtain position information every 60 degrees (six patterns for one cycle).

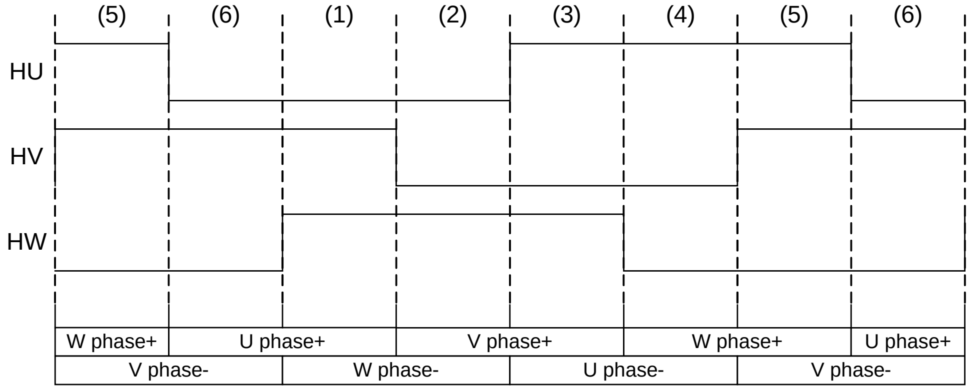


Figure 3-2 Relation between Hall Sensor Signals and Conduction Patterns (rotation direction: CW)

If the conduction patterns of each phase are changed at the switching timing of these hall sensor signals as shown in Figure 3-2, rotating coil flux generates a torque with the permanent magnet and the rotor rotates as shown in Figure 3-3.

As a conduction session of each switching element is 120 degrees, this control method is referred to as 120-degree conducting control.

The relation between above-mentioned six conduction patterns and rotor position ranges is shown in Figure 3-3.

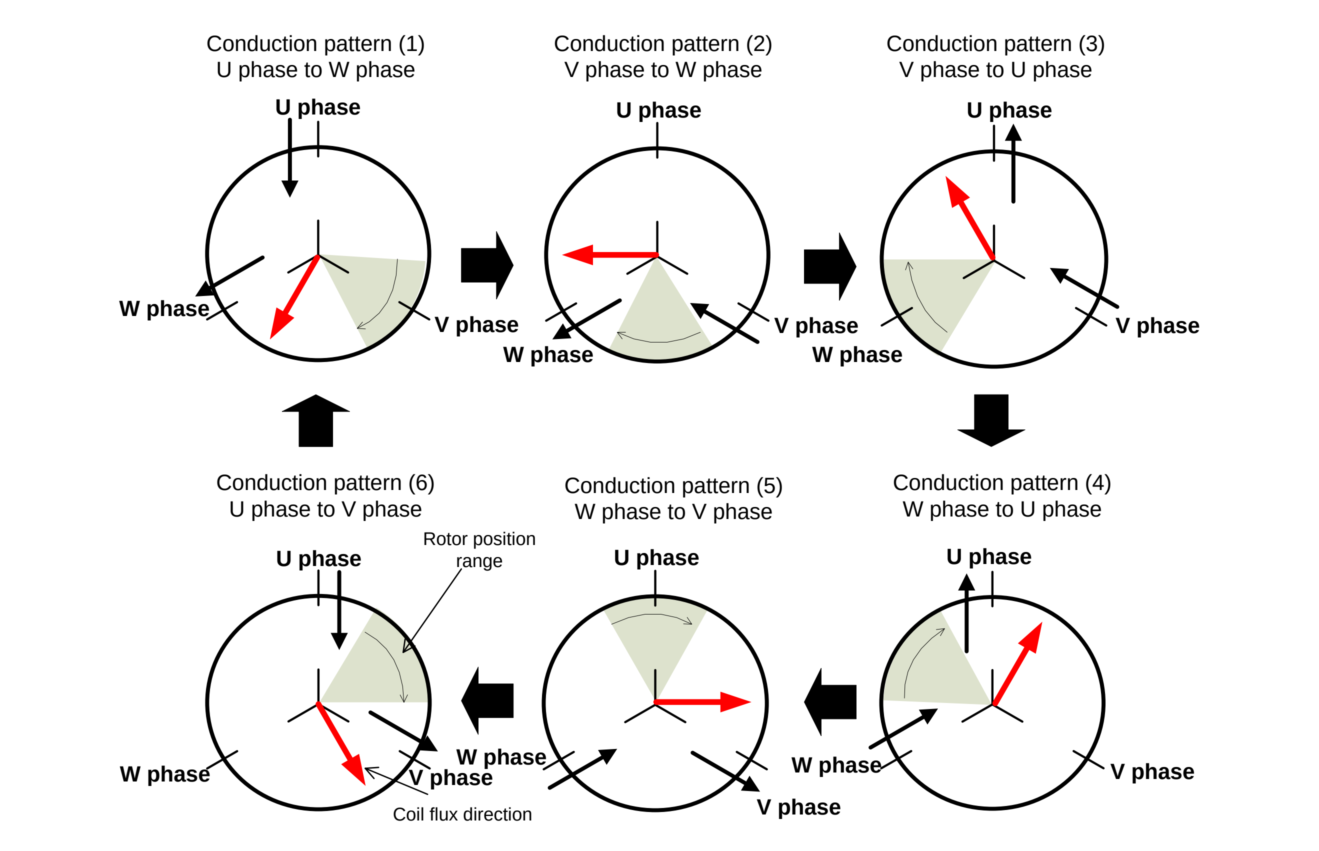


Figure 3-3 Six Conduction Patterns and Rotor Position Ranges

Notes:

1. The relation between hall sensor signals and conduction patterns shown in Figure 3-3 is set to be suitable for this system. A different motor specification requires setting different conduction patterns appropriate to the system.
2. In the 120-degree conducting control, only six types of conduction patterns are generated for one cycle and hence in principle, a torque ripple occurs without fail.

3.2 Voltage control by PWM

PWM control is used for controlling the output voltage. The PWM control is a control method that continuously adjusts the average voltage by varying the duty of pulse, as shown in Figure 3-4.

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

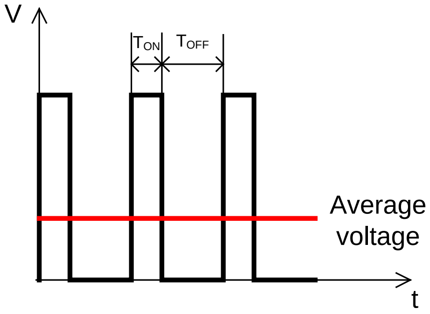


Figure 3-4 PWM Control

Here, modulation factor m is defined as follows.

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

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

This modulation factor is reflected in the setting value of the register that determines the PWM duty.

In this system, first-60-degree chopping is used to control the output voltage and speed. Figure 3-5 shows an example of motor control signal output waveforms at the time of first-60-degree chopping.

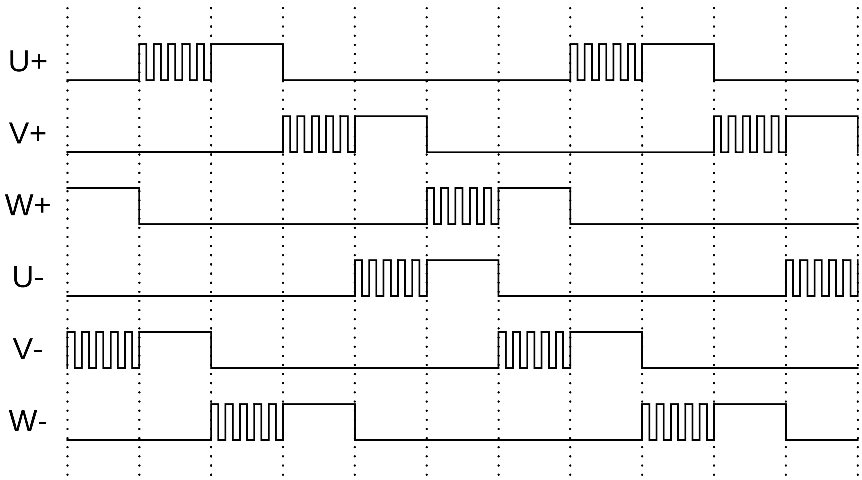


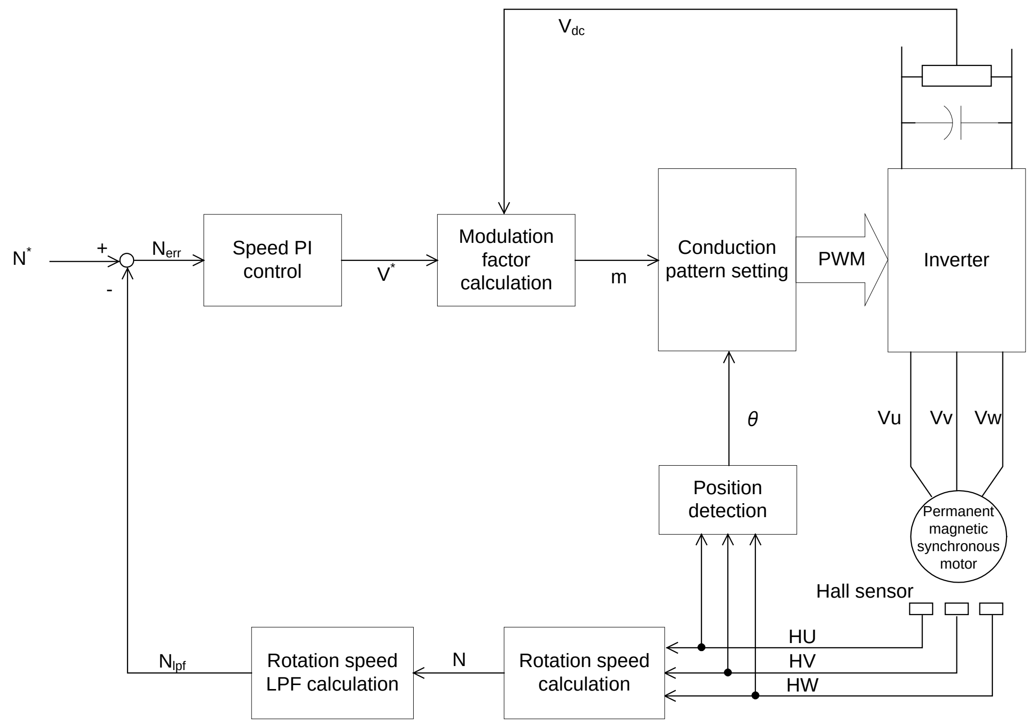
Figure 3-5 First-60-degree Chopping

4. Description of control program

This part explains the control program of this system.

4.1 Control block diagram

In this system, the motor is started to operate by open loop control. After that, control is performed according to the below block diagram.



Name	Meaning	Remarks
N	Rotation speed	
N_{lpf}	Rotation speed after LPF	
N^*	Rotation speed command value	Set in VR1
N_{err}	Rotation speed deviation	
V^*	Voltage command value	Output from speed PI control
V_{dc}	Inverter bus voltage	
m	Modulation factor	Common to each phase
PWM	PWM output signal	
V_u, V_v, V_w	Phase voltage	
HU, HV, HW	Hall sensor signals	
θ	Rotor position	Detected every 60 degrees

Figure 4-1 Control Block Diagram

Functions are as follows:

(1) Position detection of the permanent magnet

Position of the permanent magnet is detected by making interrupts occur at the both edges of hall sensor signals (HU, HV, and HW) and reading the port values within the hall sensor interrupt function. The hall sensor signals input to the microcontroller are assumed to be digital signals.

(2) Rotation speed calculation

Rotation speed is calculated by obtaining the timer counter value (CMT1.CMCNT) within the hall sensor signals (HU, HV, and HW) interrupt function. The rotation speed calculation value is used in calculating speed control.

(3) Speed control

Speed PI control is performed by using the speed command value and rotation speed calculation value. The output value of speed PI control is set as a voltage command value.

(4) Processing stop for protection

Processing stop for protection prevents the motor or inverter from breakage due to overcurrent or overvoltage.

4.2 Contents of control

4.2.1 Motor start/stop

Starting and stopping the motor are controlled by input from SW1 and VR1.

A general-purpose port (P91) is assigned to SW1. The sample program reads the P91 port within the main loop. When P91 is at low level, it is judged that the start switch is being pressed. On the other hand, when the level is switched to high, the program determines to stop the motor.

Also, an analog input port (AN102) is assigned to VR1. Input to AN102 is A/D converted within the main loop, and a rotation speed command value is generated. (As for creation of the rotation speed command value, refer to 4.2.2.) When the rotation speed command value is less than 550 [rpm], the program determines to stop the motor.

4.2.2 Motor rotation speed command value and inverter bus voltage

(1) Motor rotation speed command value

The motor rotation speed command value N* can be set by A/D converting the output values (analog values) of VR1. The A/D converted VR1 values are used as rotation speed command values, as shown in Table 4-1.

Table 4-1 Conversion Ratio of Speed Command Value

Item		Conversion ratio (Command value N*: A/D conversion value)	Channel
Rotation speed command value	CW	0 [rpm] to 2048 [rpm]: 07FFH to 0FFFH	AN102
	CCW	0 [rpm] to 2048 [rpm]: 0000H to 07FFH	

(2) Inverter bus voltage

Inverter bus voltage is measured as given in Table 4-2.

It is used for modulation factor calculation and overvoltage detection. (When an overvoltage is detected, PWM is stopped.)

Table 4-2 Inverter Voltage Conversion Ratio

Item	Conversion ratio (Inverter voltage Vdc: A/D conversion value)	Channel
Inverter bus voltage	0 [V] to 30 [V]: 0000H to 0FFFH	AN002

4.2.3 Speed control

In this system, the motor rotation speed is calculated from a difference of the current timer value and the timer value 2π [rad] before. The timer values are obtained through the external interrupt routine by hall sensor signals while having the timer of channel 1 of the compare match timer performed free running. This method is applicable even if the hall sensors are not placed at equal spaces. The LPF (low-pass filter) processing is conducted to the calculated result.

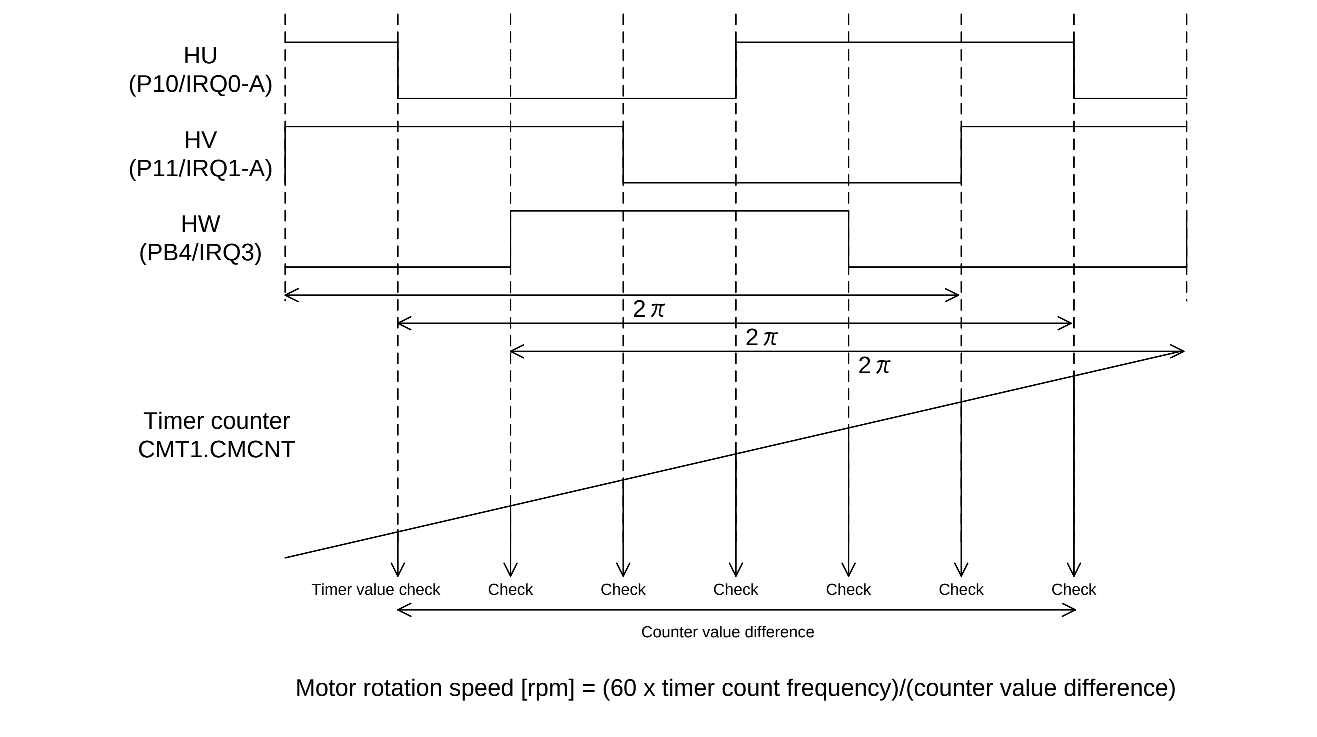


Figure 4-2 Method to Calculate Motor Rotation Speed

This system uses PI control for speed control. A voltage command value is calculated by the following formula of speed PI control (in 3 [ms] cycles).

$$v^* = (K_{P\omega} + \frac{K_{I\omega}}{s})(\omega^* - \omega)$$

v^*: Voltage command value ω^*: Speed command value ω: Rotation speed

$K_{P\omega}$: Speed PI proportional gain $K_{I\omega}$: Speed PI integral gain s: Laplace operator

For more details of PI control, please refer to specialized books.

4.2.4 System protection function

This control program has the following five types of error status and enables emergency stop functions in case of occurrence of respective error.

- Overcurrent error

  An emergency stop signal (over current detection) from hardware forces the program to execute high impedance output to the PWM output port.

- Overvoltage error

  The inverter bus voltage is monitored with 1 [ms] interval. When an over voltage is detected (when the voltage exceeds 28 [V]), CPU performs an emergency stop. The threshold of the overvoltage error 28 [V] is set in consideration of the error of resistance and supply voltage from AC adapter, etc.

- Rotation speed abnormality error

  The rotation speed calculation value is monitored with 1 [ms] interval. When an error value is detected in rotation speed values (in case of value over 16000 [rpm] (electrical angle), CPU performs an emergency stop.

- Timeout error

  When no pattern switching by hall sensor detection occurs for a certain period (20 [ms]), CPU performs an emergency stop.

- Hall sensor signal pattern error

  Hall sensor signal patterns are monitored upon each hall sensor interrupt handling. When an error pattern is detected, CPU performs emergency stop.

4.3 Function specifications

Multiple control functions are used in this control program. Lists of control functions are given below.

For detailed processing, refer to flowcharts or source files.

Table 4-3 Control Functions List (1/4)

File name	Function overview	Processing overview
main.c	main() Input: None Output : None	- Hardware initialization function call - User interface initialization function call - Main processing use variable initialization function call - Status transition and event execution function call - Main processing → Main processing execution function call → Watchdog timer clear function call
	ctrl_ui() Input: None Output : None	- Motor status change - Determination of rotation speed command value and rotation direction
	software_init() Input: None Output : None	Initialization of variables used in the main processing
mtr_ctrl_rssk.c	get_vr1() Input: None Output: (uint16) ad_data / A/D conversion result	A/D conversion execution function call
	get_sw1() Input: None Output: (uint8) tmp_port / level of SW1	Obtaining the status of SW1
	get_sw2() Input: None Output: (uint8) tmp_port / level of SW2	Obtaining the status of SW2
	led1_on() Input: None Output : None	Turning LED1 ON
	led2_on() Input: None Output : None	Turning LED2 ON
	led1_off() Input: None Output : None	Turning LED1 OFF
	led2_off() Input: None Output : None	Turning LED2 OFF

Table 4-4 Control Functions List (2/4)

File name	Function overview	Processing overview
mtr_ssns_hall_120.c	R_MTR_InitSequence() Input: None Output : None	Initialization of sequence processing
	R_MTR_ExecEvent() Input: (uint8)u1_event /occurred event Output : None	- Changing the status - Calling an appropriate processing execution function for the occurred event
	mtr_act_run () Input: (uint8)u1_state /motor status Output: (uint8)u1_state /motor status	- Variable initialization function call upon motor startup - Motor control startup function call - Conduction pattern determination function call
	mtr_act_stop () Input: (uint8)u1_state /motor status Output: (uint8)u1_state /motor status	Motor control stop function call
	mtr_act_none () Input: (uint8)u1_state /motor status Output: (uint8)u1_state /motor status	No processing is performed.
	mtr_act_reset () Input: (uint8)u1_state /motor status Output: (uint8)u1_state /motor status	Global variable initialization
	mtr_act_error() Input: (uint8)u1_state /motor status Output: (uint8)u1_state /motor status	Motor control stop function call
	mtr_pattern_set() Input: None Output : None	- Speed measurement function call - Obtaining hall sensor patterns - Conduction pattern determination - Motor control signal creation function call
	mtr_speed_calc () Input: None Output : None	Speed measurement calculation processing
	mtr_start_init () Input: None Output : None	Initializing only the variables required for motor startup
	mtr_pi_ctrl () Input: MTR_PI_CTRL *pi_ctrl/structure for PI control Output: (float32)f4_ref / PI control output value	Speed PI control
	R_MTR_SetSpeed () Input: (int16)ref_speed/rotation speed command value Output : None	Rotation speed command value setting
	R_MTR_SetDir() Input: (uint8)dir /rotation direction command value Output : None	Rotation direction setting
	R_MTR_GetSpeed () Input: None Output: (int16)s2_speed_rpm/rotation speed calculation value	Obtaining the rotation speed calculation value (electrical angle)
	R_MTR_GetStatus () Input: None Output: (uint8)g_u1_mode_system/motor status	Obtaining the motor status
	mtr_error_check() Input: None Output : None	Error monitoring and detection

Table 4-5 Control Functions List (3/4)

File name	Function overview	Processing overview
mtr_interrupt.c	mtr_hall_u_interrupt () Input: None Output : None	Hall interrupt function call
	mtr_hall_v_interrupt () Input: None Output : None	Hall interrupt function call
	mtr_hall_w_interrupt () Input: None Output : None	Hall interrupt function call
	mtr_hall_interrupt () Input: None Output : None	Conduction pattern determination function call
	mtr_over_current_interrupt () Input: None Output : None	- Changing the motor status - Event processing select function call - Pulse output forced shutdown flag clearing function call
	mtr_cmt0_interrupt () Input: None Output : None	- Error check function call - Boot mode time measurement - Calling speed PI control function every 3 [ms]
	mtr_mtu3_interrupt () Input: None Output : None	Waiting for motor rotation stop

Table 4-6 Control Functions List (4/4)

File name	Function overview	Processing overview
mtr_ctrl_rx62t.c	R_MTR_InitHardware () Input: None Output : None	Initializing the clock and peripheral functions
	init_ui() Input: None Output : None	Initializing user usage peripheral functions
	mtr_ctrl_start () Input: None Output : None	Motor startup processing
	mtr_ctrl_stop() Input: None Output : None	Motor stop processing
	mtr_get_vdc() Input: None Output: (float32)f4_temp	Inverter bus voltage A/D conversion
	mtr_get_vr1() Input: None Output: (uint16)u2_temp	VR1 A/D conversion
	clear_wdt() Input: None Output : None	Clearing the watchdog timer
	mtr_clear_oc_flag() Input: None Output : None	Clearing the high impedance state
	mtr_clear_mtu3_flag() Input: None Output : None	Clearing the interrupt flag
	mtr_clear_cmt0_flg() Input: None Output : None	Clearing the interrupt flag
	mtr_change_pattern() Input: (uint8)pattern / conduction pattern Output : None	- Setting a conduction pattern - Changing the motor status when an output pattern error occurs - Event processing selection function call

4.4 Variables list

Lists of variables used in this control program are given below. Note that local variables are not described.

Table 4-7 Variables List (1/2)

Variable name	Type	Content	Remark
g_s2_max_speed	int16	Rotation speed command maximum value	Mechanical angle [rpm]
g_s2_min_speed	int16	Rotation speed command minimum value	Mechanical angle [rpm]
g_s2_margin_min_speed	int16	Rotation speed command minimum value for motor stop	Mechanical angle [rpm]
g_s2_ref_speed	int16	User setting rotation speed	Electrical angle [rpm]
g_u1_rot_dir	uint8	User setting rotation direction	0: CW 1: CCW
g_u1_motor_status	uint8	User motor status management	0: Stop 1: Rotating 2: Error
g_u1_reset_req	uint8	Reset request flag	0: Turning SW2 ON in error status 1: Turning SW2 OFF in error status
g_u1_sw1_cnt	uint8	SW1 determination counter	Chattering removal
g_u1_sw2_cnt	uint8	SW2 determination counter	Chattering removal
g_u1_stop_req	uint8	VR1 stop command flag	Stop is determined when the rotation speed command value is less than 550 [rpm].
g_u1_cnt_speed_pi	uint8	Interrupt decimation counter for speed PI control	Speed PI control cycle 3 [ms] is counted.
g_u2_cnt_boot_mode	uint16	BOOT mode time measurement counter	20 [ms] after motor startup is counted.
g_u2_cnt_wait_stop	uint16	Motor rotation stop waiting counter	10 [ms] after motor stop processing is counted. (Note that the count is reset when a hall sensor interrupt is detected.)
g_u1_flg_wait_stop	uint8	Motor rotation stop waiting flag	The flag set upon motor stop command. When no hall sensor interrupt is detected for 10 [ms] after motor stop processing, the flag is cleared.
g_f4_v_ref	float32	Voltage command value	Speed PI control output value [V]
g_f4_vdc_ad	float32	Inverter bus voltage A/D value	[V]
g_s2_pwm_duty	int16	Timer RD compare register setting value	-
g_f4_ref_speed_rad	float32	Speed command value	Electrical angle [rad/s]
g_f4_ref_speed_rad_pi	float32	Speed command value	Electrical angle [rad/s]
g_f4_speed_rad	float32	Speed calculation value	Electrical angle [rad/s]
g_f4_kp_speed	float32	Speed PI control proportional gain	-
g_f4_ki_speed	float32	Speed PI control integral gain	-
g_f4_lim_v	float32	Voltage PI control output limit value	[V]
g_f4_ilim_v	float32	Voltage PI control integral term limit value	[V]
speed	MTR_PI_CTRL	Structure for speed PI control	-
g_u1_cnt_ics	uint8	ICS function call interval counter	-

Table 4-8 Variables List (2/2)

Variable name	Type	Content	Remark
g_u2_run_mode	uint16	Operation mode management	0: Boot mode 3: Normal operation mode
g_u1_error_status	uint8	Error status management	1: Over current error 2: Over voltage error 3: Rotation speed abnormality error 4: Timeout error 6: Hall sensor pattern error (0xff: Undefined error)
g_u1_mode_system	uint8	Status management	0: Stop mode 1: Run mode 2: Error mode
g_u1_hall_signal	uint8	Three-phase hall sensor input value	-
g_u1_v_pattern	uint8	Conduction pattern	-
g_u2_cnt_timeout	uint16	Stop determination time measurement counter	Cleared when the conducting pattern is switched.
g_u1_direction	uint8	Rotation direction management	0: CW 1: CCW
g_u2_hall_timer_cnt	uint16	Free run timer count value	CMT1.CMCNT
g_u2_pre_hall_timer_cnt	uint16	Last free run timer count value	-
g_s4_timer_cnt_ave	float32	Speed measurement timer count difference for 2π	-
g_u2_timer_cnt_buf	uint16	Speed measurement timer count buffer	-
g_u2_timer_cnt_num	uint16	Speed measurement timer count buffer number	-

4.5 Macro definitions

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

Table 4-9 Macro Definitions List (1/5)

File name	Macro name	Definition value	Remark
main.h	M_CW	0	User setting rotation direction : CW
	M_CCW	1	User setting rotation direction : CCW
	MAX_SPEED	2000	Rotation speed command maximum value (mechanical angle) [rpm]
	MIN_SPEED	600	Rotation speed command minimum value (mechanical angle) [rpm]
	MARGIN_SPEED	50	Rotation speed command minimum value creation constants for stop (mechanical angle) [rpm]
	MARGIN_MIN_SPEED	MIN_SPEED - MARGIN_SPEED	Rotation speed command minimum value for motor stop (mechanical angle) [rpm]
	SW_ON	0	Active in case of “Low”
	SW_OFF	1	Active in case of “Low”
	CHATTERING_CNT	10	Chattering removal
	VR1_SCALING	1	Speed command value creation constant
	ADJUST_OFFSET	0x7FF	Speed command value offset adjustment constant
	POLE_PAIR	7	Constant for correcting number of pole pairs (seven pairs)
	REQ_CLR	0	VR1 stop command flag clearing
	REQ_SET	1	VR1 stop command flag setting

Table 4-10 Macro Definitions List (2/5)

File name	Macro name	Definition value	Remark
mtr_ctrl_rx62t.h	MTR_PWM_TIMER_FREQ	96.0f	PWM timer count frequency [MHz]
	MTR_CARRIER_FREQ	20.0f	Carrier frequency [kHz]
	MTR_CARRIER_SET	(MTR_PWM_TIMER_FREQ * 1000 / MTR_CARRIER_FREQ) - 1	Carrier setting value
	MTR_PORT_HALL_U	PORT1.PORT.BIT.B0	U phase hall sensor input port
	MTR_PORT_HALL_V	PORT1.PORT.BIT.B1	V phase hall sensor input port
	MTR_PORT_HALL_W	PORTB.PORT.BIT.B4	W phase hall sensor input port
	MTR_PORT_UP	PORT7.DR.BIT.B1	U phase (positive phase) output port
	MTR_PORT_UN	PORT7.DR.BIT.B4	U phase (negative phase) output port
	MTR_PORT_VP	PORT7.DR.BIT.B2	V phase (positive phase) output port
	MTR_PORT_VN	PORT7.DR.BIT.B5	V phase (negative phase) output port
	MTR_PORT_WP	PORT7.DR.BIT.B3	W phase (positive phase) output port
	MTR_PORT_WN	PORT7.DR.BIT.B6	W phase (negative phase) output port
	MTR_PORT_SW1	PORT9.PORT.BIT.B1	SW1 input port
	MTR_PORT_SW2	PORT9.PORT.BIT.B2	SW2 input port
	MTR_PORT_LED1	PORTA.DR.BIT.B2	LED1 output port
	MTR_PORT_LED2	PORTA.DR.BIT.B3	LED2 output port
	MTR_LED_ON	0	Active in case of “Low”
	MTR_LED_OFF	1	
	MTR_SPEED_TCNT	CMT1.CMCNT	Timer count register for speed measurement

Table 4-11 Macro Definitions List (3/5)

File name	Macro name	Definition value	Remark
mtr_ssns_ hall_120.h	MTR_TWOPI	2 * 3.14159265f	2 π
	MTR_RPM_RAD	MTR_TWOPI / 60	Constant to change units from [rpm] to [rad/s]
	MTR_POLE_PAIRS	7	Constant for correcting number of pole pairs (seven pairs)
	MTR_CW	0	Rotation direction setting value: CW
	MTR_CCW	1	Rotation direction setting value: CCW
	MTR_START_REFV	3.0f	Command voltage at startup [V]
	MTR_SPEED_LIMIT_RPM	2500	Speed limit value (mechanical angle) [rpm]
	MTR_SPEED_LIMIT	MTR_SPEED_LIMIT_RPM * MTR_POLE_PAIRS * MTR_TWOPI / 60	Speed limit value (electrical angle) [rad/s]
	MTR_OVERVOLTAGE_LIMIT	28	Over voltage error determination value [V]
	MTR_VDC_SCALING	30.0f / 4095.0f	Inverter bus voltage A/D conversion value resolution
	MTR_PATTERN_CW_V_U	5	CW hall sensor values
	MTR_PATTERN_CW_W_U	4	
	MTR_PATTERN_CW_W_V	6	
	MTR_PATTERN_CW_U_V	2	
	MTR_PATTERN_CW_U_W	3	
	MTR_PATTERN_CW_V_W	1	
	MTR_PATTERN_CCW_V_U	2	CCW hall sensor values
	MTR_PATTERN_CCW_V_W	6	
	MTR_PATTERN_CCW_U_W	4	
	MTR_PATTERN_CCW_U_V	5	
	MTR_PATTERN_CCW_W_V	1	
	MTR_PATTERN_CCW_W_U	3	
	MTR_REF_SPEED_UP_STEP	1.0f	Command speed adding value [rad/s]
	MTR_REF_SPEED_DOWN_STEP	1.0f	Command speed subtracting value [rad/s]
	MTR_SPEED_PI_DECIMATION	2	Number of interrupt decimation times for speed PI control
	MTR_SPEED_PI_KP	0.001f	Proportional gain
	MTR_SPEED_PI_KI	0.0001f	Integral gain
	MTR_SPEED_PI_LIMIT_V	24	Voltage PI control output limit value [V]
	MTR_SPEED_PI_I_LIMIT_V	24	Voltage PI control integral term limit value [V]
	MTR_MAX_DRIVE_V	15.0f	Maximum command voltage [V]
	MTR_MIN_DRIVE_V	3.8f	Minimum command voltage [V]
	MTR_TIMEOUT_CNT	20	Stop determination time [ms]
	MTR_START_CNT	100	Boot mode period [ms]
	MTR_SPEED_CALC_BASE	MTR_TWOPI * 6000000	Constant for speed measurement

Table 4-12 Macro Definitions List (4/5)

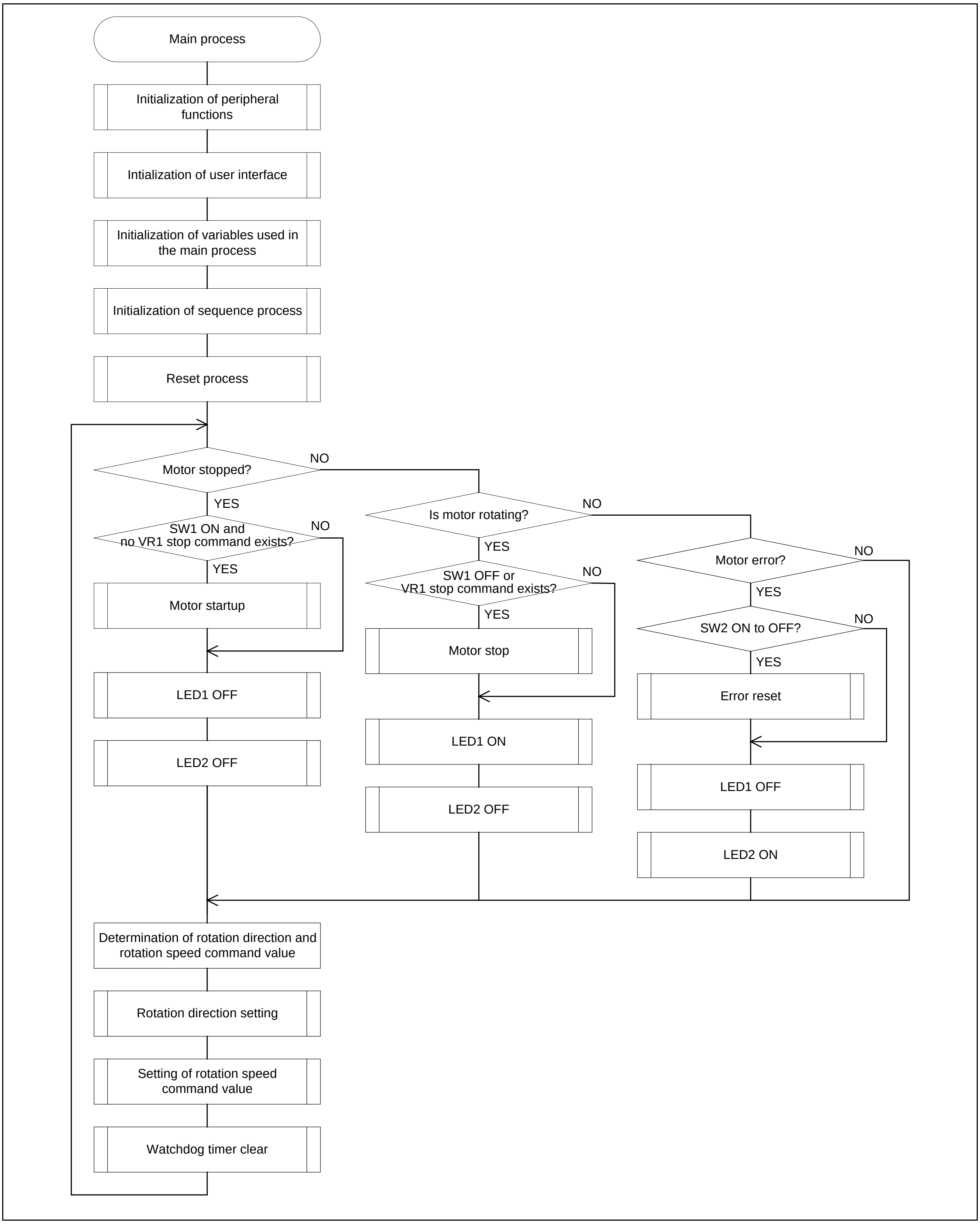
File name	Macro name	Definition value	Remark
mtr_ssns_hall_120.h	MTR_PATTERN_ERROR	0	Conduction pattern
	MTR_UP_PWM_VN_ON	1	
	MTR_UP_PWM_WN_ON	2	
	MTR_VP_PWM_UN_ON	3	
	MTR_VP_PWM_WN_ON	4	
	MTR_WP_PWM_UN_ON	5	
	MTR_WP_PWM_VN_ON	6	
	MTR_UP_ON_VN_PWM	7	
	MTR_UP_ON_WN_PWM	8	
	MTR_VP_ON_UN_PWM	9	
	MTR_VP_ON_WN_PWM	10	
	MTR_WP_ON_UN_PWM	11	
	MTR_WP_ON_VN_PWM	12	
	MTR_SPEED_LPF_K	0.3	LPF coefficient for speed
	MTR_FLG_CLR	0	Constant for flag clear
	MTR_FLG_SET	1	Constant for flag setting
	MTR_STOP_WAIT_CNT	200	Period to wait for stop [ms]
	MTR_SPEED_PI_CHANGE_RPM	4200	Handover speed when changing modes (electrical angle) [rpm]
	MTR_ICS_DECIMATION	4	Number of function call decimation times for ICS

Table 4-13 Macro Definitions List (5/5)

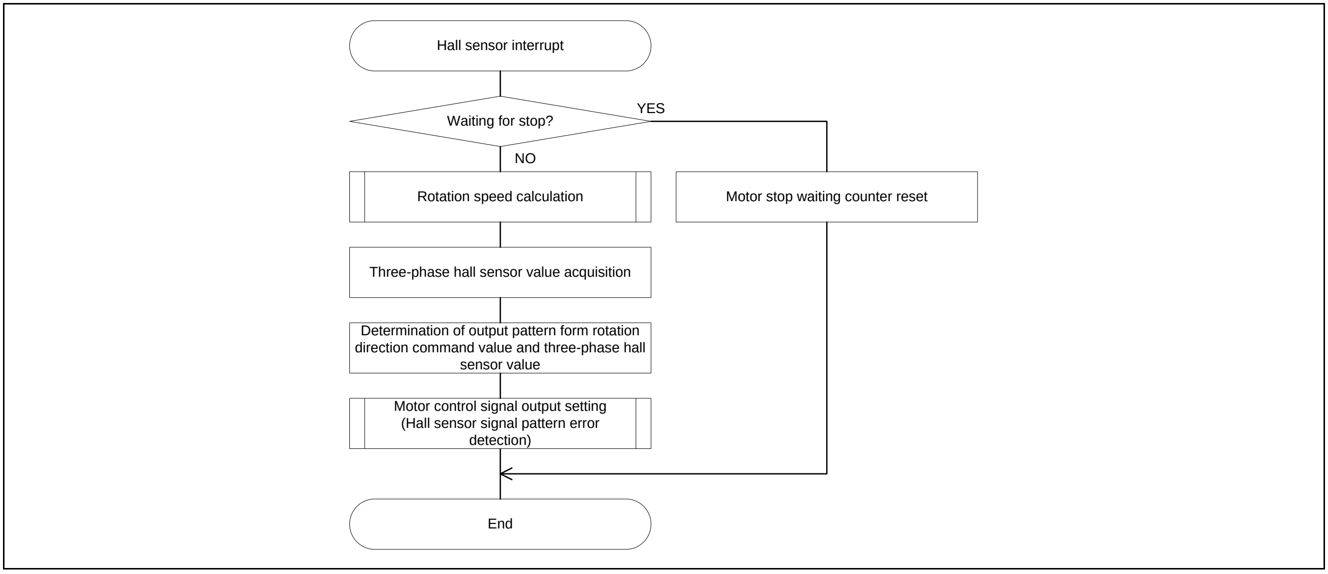
File name	Macro name	Definition value	Remark
mtr_ssns_hall_120.h	MTR_BOOT_MODE	0x00	Boot mode
	MTR_OPENLOOP_MODE	0x01	Open loop mode
	MTR_START_MODE	0x02	Start mode
	MTR_HALL_120_MODE	0x03	Hall sensor 120-degree operation mode
	MTR_BEMF_120_MODE	0x04	BEMF sensorless 120-degeree operation mode
	MTR_ENCD_FOC_MODE	0x05	Encoder vector operation mode
	MTR_LESS_FOC_MODE	0x06	Sensorless vector operation mode
	MTR_OVER_CURRENT_ERROR	0x01	Overcurrent error
	MTR_OVER_VOLTAGE_ERROR	0x02	Overvoltage error
	MTR_OVER_SPEED_ERROR	0x03	Rotation speed abnormality error
	MTR_TIMEOUT_ERROR	0x04	Timeout error
	MTR_HALL_ERROR	0x05	Hall sensor pattern error
	MTR_BEMF_ERROR	0x06	BEMF pattern error
	MTR_UNDER_VOLTAGE_ERROR	0x07	Low voltage error
	MTR_UNKNOWN_ERROR	0xff	Undefined error
	MTR_MODE_STOP	0x00	Stop status
	MTR_MODE_RUN	0x01	Rotating status
	MTR_MODE_ERROR	0x02	Error status
	MTR_SIZE_STATE	3	Status count
	MTR_EVENT_STOP	0x00	Motor stop event
	MTR_EVENT_RUN	0x01	Motor startup event
	MTR_EVENT_ERROR	0x02	Motor error event
	MTR_EVENT_RESET	0x03	Motor reset event
	MTR_SIZE_EVENT	4	Events count

4.6 Control flow (flow chart)

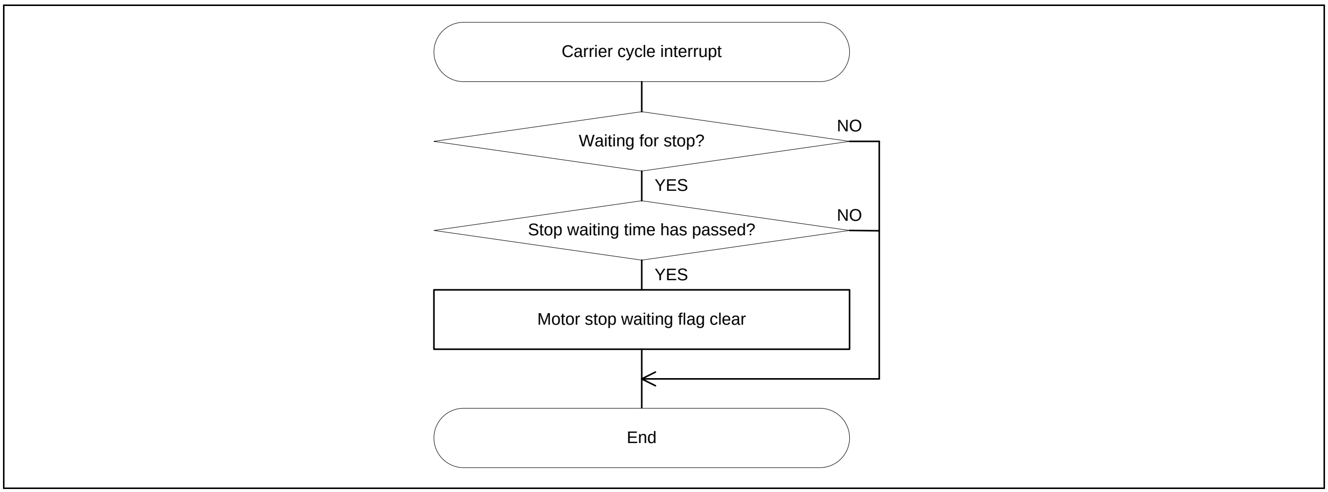
(1) Main processing



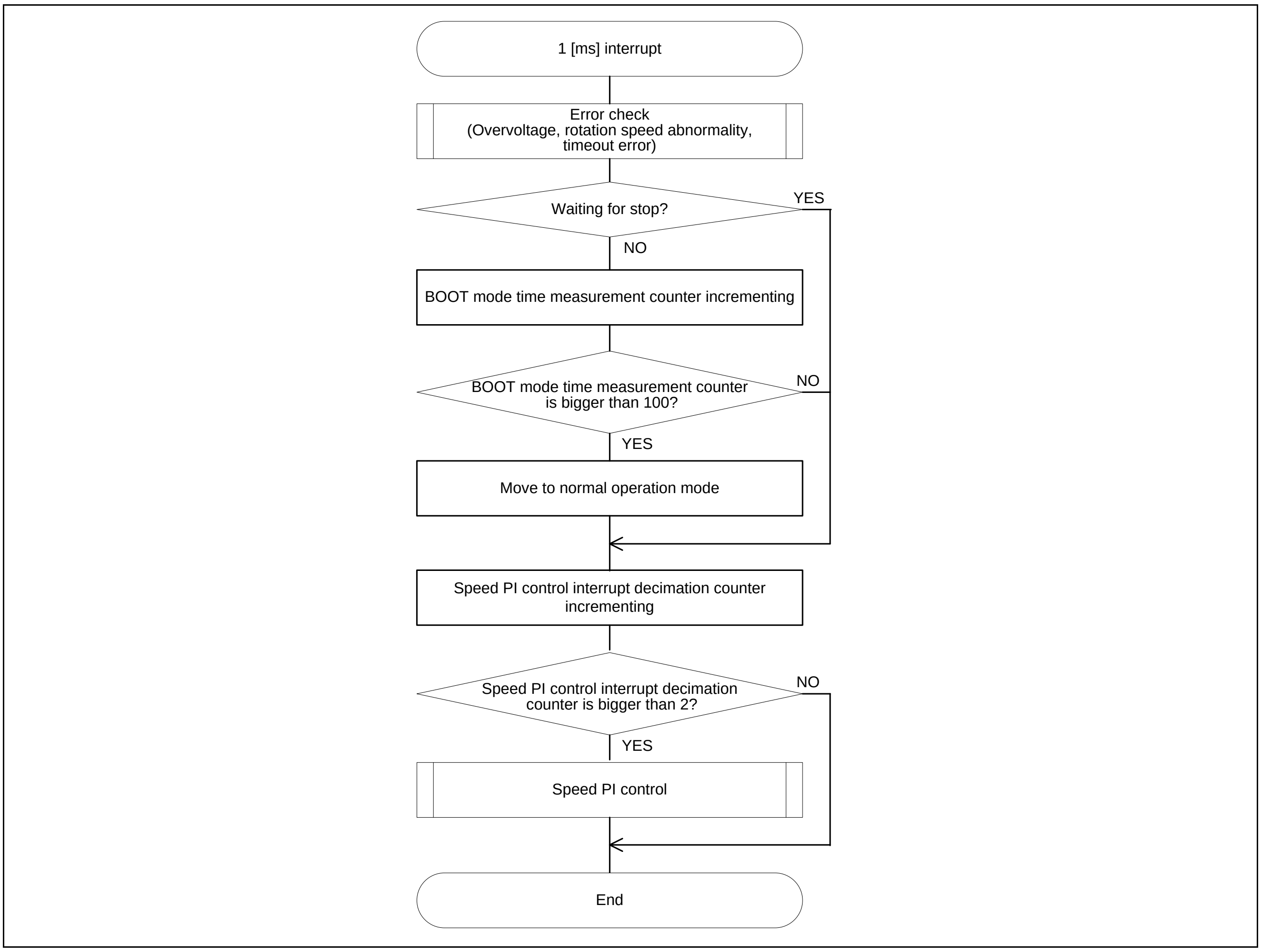
(2) Hall sensor interrupt handling



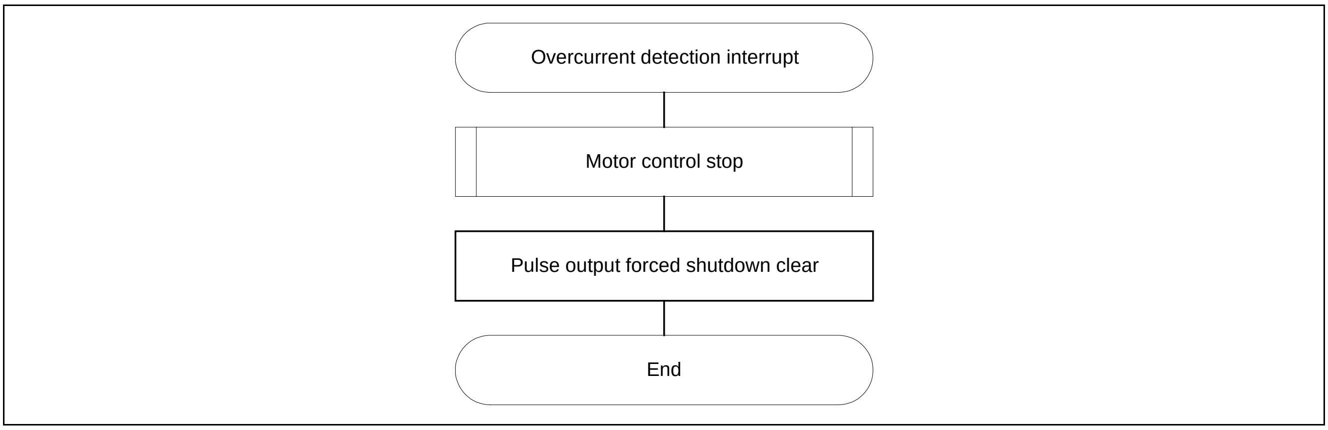
(3) Carrier cycle interrupt handling



(4) 1 [ms] interrupt handling



(5) Overcurrent interrupt handling



Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Oct. 9, 2013	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU 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. Handling of Unused Pins
   Handle unused pins in accord 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.
5. Differences between Products
   Before changing from one product to another, i.e. to a product with a different type number, confirm that the change will not lead to problems.
   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the 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.

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 2.2