RX62T

R01AN1662EJ0100
Rev.1.00
Apr 9, 2013

Motor control by RX62T micro controller
Sensorless vector control of permanent magnetic synchronous motor

Summary

This application note aims at explaining the sample program for operating the sensorless vector control of permanent magnetic synchronous motor, by using functions of RX62T.

The sample program is only to be used as reference and Renesas Electronics Corporation does not guarantee the operations. Please use this sample program after carrying 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 explains the sample program of the sensorless vector control of permanent magnetic synchronous motor (henceforth referred to as PMSM) by using the RX62T micro controller.

1.1 Usage of the system

This system (sample program) enables sensorless vector control by using RSSK (Note 1) for motor control (Low Voltage Motor Control Starter-Kit Evaluation System and surface 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+ (V1.03.00)

(2) Hardware environment

On-chip debug emulator	E1
Microcomputer used	RX62T (F562TAADFM)
Inverter board for motor control	Low Voltage Motor Control Starter-Kit Evaluation System (P03401-D1-001)
Motor	FH6S20E-X81 (SPMSM)

2. System overview

Overview of this system is explained below.

2.1 Hardware configuration

The hardware configuration is shown below.

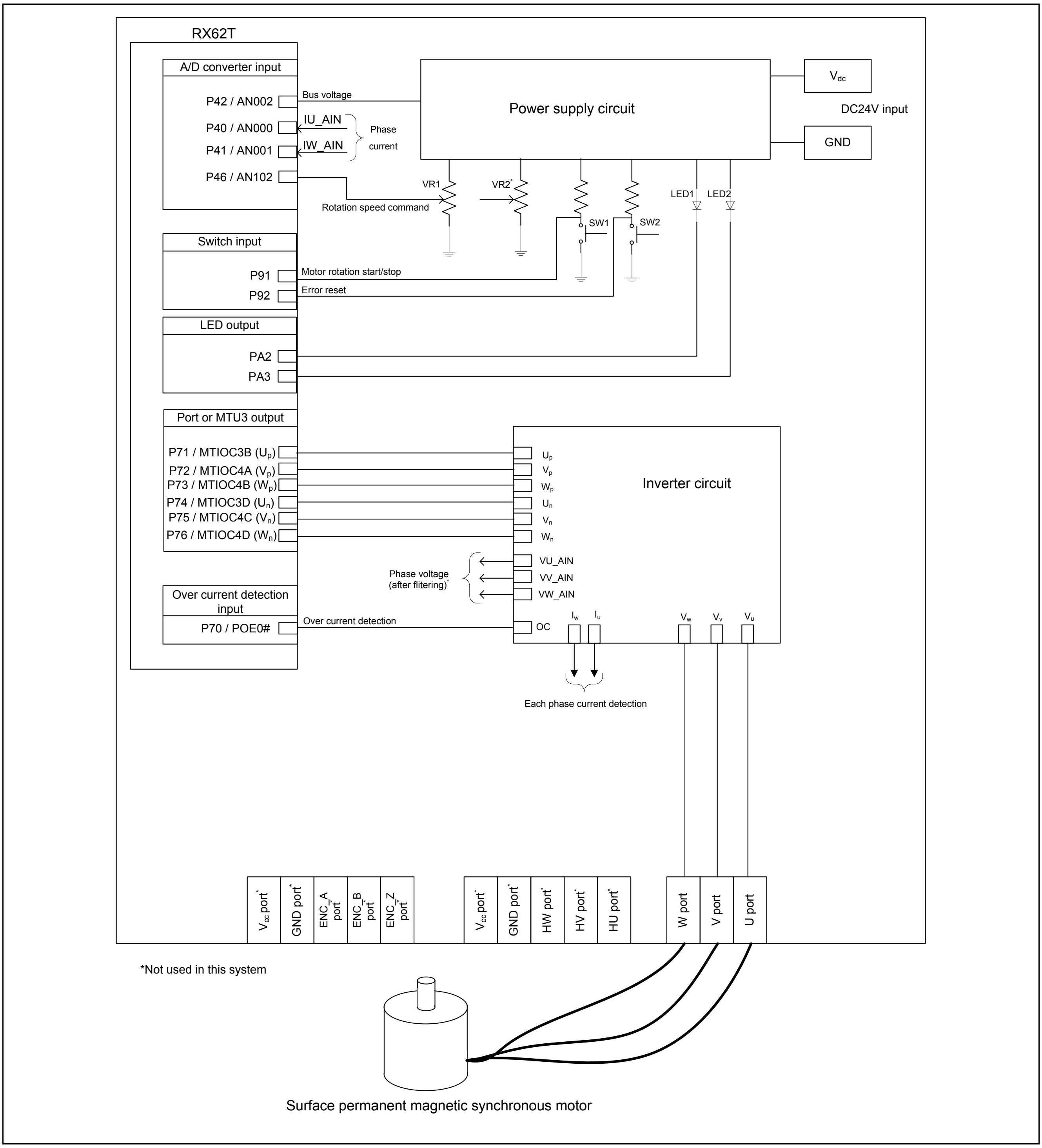


Figure 2-1 Hardware Configuration Diagram

2.2 Hardware specifications

2.2.1 User interface

List of user interfaces of this system is given in Table 2-1.

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

List of port interfaces of RX62T micro controller of this system is given in Table 2-2.

Table 2-2 Port Interfaces

Port name	Function
P42 / AN002	Inverter bus voltage measurement
P46 / AN102	For rotation speed command value input (analog value)
P91	START/STOP push switch
P92	ERROR RESET push switch
PA2	LED1 ON/OFF control
PA3	LED2 ON/OFF control
P40 / AN000	U phase current measurement
P41 / AN001	W phase current measurement
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)
P70 / POE0#	PWM emergency stop input at the time of over current detection
RESET#	RESET

2.2.2 Peripheral functions

List of the peripheral functions used in this system is given in Table 2-3.

Table 2-3 List of the Peripheral Functions

Peripheral function	Usage
12-bit A/D converter (S12ADA)	• Rotation speed command value input • Inverter bus voltage measurement • U, W phase current measurement
Compare match timer (CMT)	1 [ms] interval timer
Multi-function timer pulse unit 3 (MTU3)	Complementary PWM output (six outputs)
Port output enable 3 (POE3)	In the case of over current detection, set PWM output to high impedance

(1) 12-bit A/D converter

The rotation speed command value input, U phase current (I_u), W phase current (I_w), and inverter bus voltage (V_{dc}) are measured by using '12-bit A/D converter'.

The operation mode varies depending on units. For the Unit 0, set the 'Single-cycle Scan mode' with sample-and-hold function, and for the Unit 1, set the 'Single mode' (use software trigger).

(2) Compare match timer (CMT)

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

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

The 6-phase PWM output with dead time (high active) is performed by using the complementary PWM mode.

(4) Port output enable 3 (POE3)

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

2.3 Software configuration

2.3.1 Software file configuration

Folder and file configuration of the sample program is given below.

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

RX62T_RSSK_SSNS_LESS_FOC_ICS_CSP_V100	inc	ics_rx62t_uart0.h	ICS header
		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_less_foc.h	Sensorless vector control dependent part header
	lib	ics.lib	ICS library
		angle_speed.lib	Estimating position and speed 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 _less_foc.c	Sensorless vector control dependent part

2.3.2 Module configuration

Module configuration of the sample program is described below.

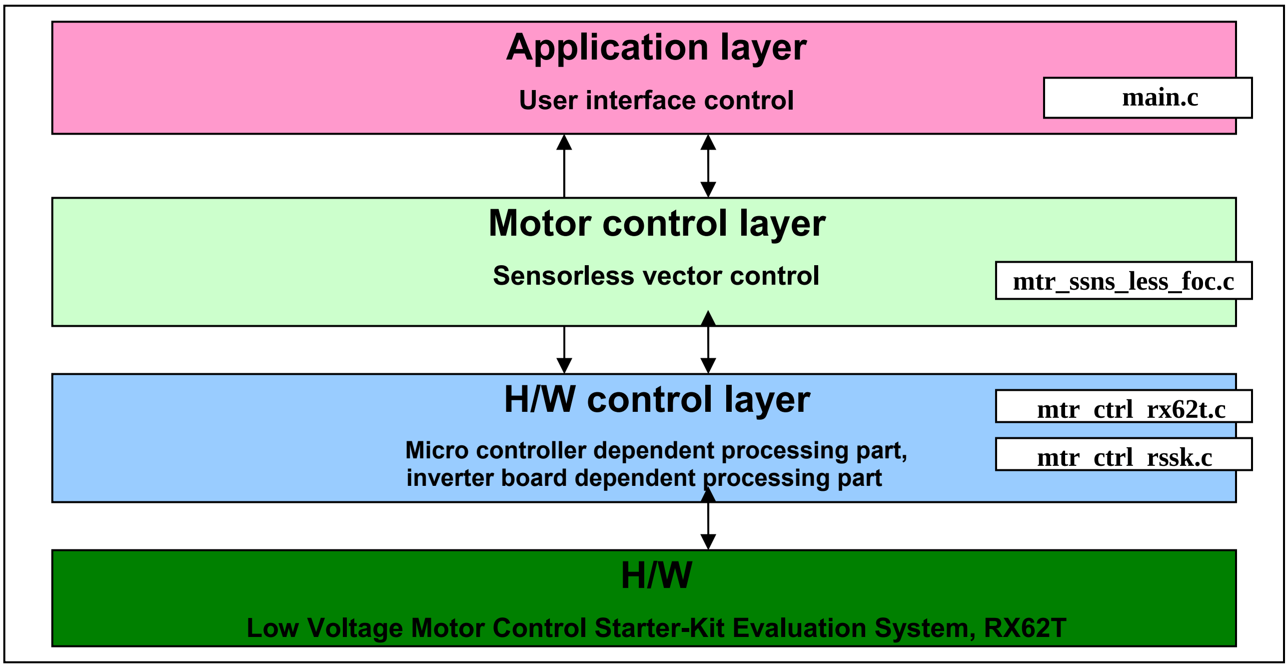


Figure 2-2 Module Configuration of the Sample Program

2.4 Software specifications

Basic specifications of software of this system are given in Table 2-5.

Table 2-5 Basic Specifications of the Software

Item	Content
Control method	Vector control
Motor rotation start/stop	Determined depending on the level of SW1 (P91) ("Low": rotation start "High": stop)
Position detection of rotor magnetic pole	Sensorless
Carrier frequency (PWM)	20 [kHz]
Control cycle	100 [μs] (carrier cycle*2)
Rotation speed control range	CW: 600 [rpm] to 2000 [rpm]
Processing stop for protection	･Disables the motor control signal output (six outputs), under any of the following four conditions. 1. Current of each phase exceeds 10 [A] (monitored per 100 [μs]) 2. Inverter bus voltage exceeds 28 [V] (monitored per 100 [μs]) 3. Inverter bus voltage is less than 0 [V] (monitored per 100 [μs]) 4. Rotation speed exceeds 1600 [rad/s] (electrical angle) (monitored per 100 [μs]) ･In the case of over current detection, set the PWM output to high impedance ("Low" is input to the POE0# port)

3. Motor control method

The SPMSM vector control used in the sample program is explained here.

3.1 Voltage equation of the motor control system

Voltage equation of the permanent magnetic synchronous motor (Figure 3-1) having the magnetic flux distribution of sine-wave shape can be expressed as follows.

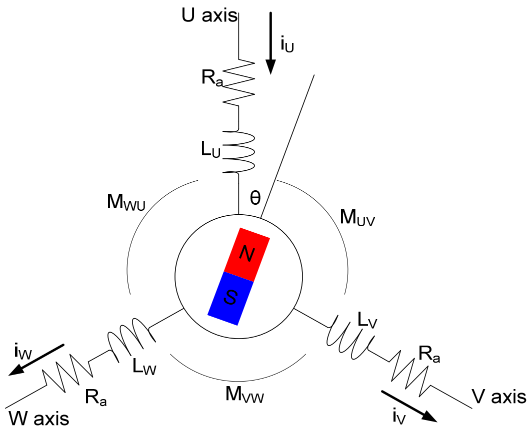


Figure 3-1 Conceptual diagram of the three phase permanent magnetic synchronous motor

$$\begin{pmatrix} v_u \\ v_v \\ v_w \end{pmatrix} = R_a \begin{pmatrix} i_u \\ i_v \\ i_w \end{pmatrix} + p \begin{pmatrix} \phi_u \\ \phi_v \\ \phi_w \end{pmatrix}$$

$$\begin{pmatrix} \phi_u \\ \phi_v \\ \phi_w \end{pmatrix} = \begin{pmatrix} L_u & M_{uv} & M_{wu} \\ M_{uv} & L_v & M_{vw} \\ M_{wu} & M_{vw} & L_w \end{pmatrix} \begin{pmatrix} i_u \\ i_v \\ i_w \end{pmatrix} + \psi \begin{pmatrix} \cos\theta \\ \cos(\theta - 2\pi/3) \\ \cos(\theta + 2\pi/3) \end{pmatrix}$$

v_u, v_v, v_w : Each phase armature voltage L_u, L_v, L_w : Each phase self inductance

i_u, i_v, i_w : Each phase armature current M_{uv}, M_{vw}, M_{wu} : Each phase mutual inductance

ϕ_u, ϕ_v, ϕ_w : Each phase armature interlinkage flux

ψ :Maximum value of armature interlinkage flux depending on permanent magnet

R_a : Each phase armature resistance

θ : Lead angle of permanent magnet (rotor) from U phase p : Differential operator

Here, self-inductance and mutual inductance are expressed as shown in the following formula.

$$\begin{cases} L_u = l_a + L_a - L_{as}\cos(2\theta) \\ L_u = l_a + L_a - L_{as}\cos(2\theta + 2\pi/3) \\ L_u = l_a + L_a - L_{as}\cos(2\theta - 2\pi/3) \end{cases}$$

$$\begin{cases} M_{uv} = -L_a/2 - L_{as}\cos(2\theta - 2\pi/3) \\ M_{vw} = -L_a/2 - L_{as}\cos 2\theta \\ M_{wu} = -L_a/2 - L_{as}\cos(\theta + 2\pi/3) \end{cases}$$

l_a : Leakage inductance for one phase

L_a : Average value of effective inductance for one phase

L_{as} : Amplitude of effective inductance for one phase

3.2 Vector control

The d axis is set in the direction of the magnetic flux (N pole) of the permanent magnet and the q axis is set in the direction which progresses by 90 degrees from the d axis. Then by using the following conversion matrix, coordinate conversion is performed.

$$C = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos\theta & \cos(\theta - 2\pi/3) & \cos(\theta + 2\pi/3) \\ -\sin\theta & -\sin(\theta - 2\pi/3) & -\sin(\theta + 2\pi/3) \end{pmatrix}$$

$$\begin{pmatrix} v_d \\ v_q \end{pmatrix} = C \begin{pmatrix} v_u \\ v_v \\ v_w \end{pmatrix}$$

The voltage equation in the dq coordinate system is obtained as follow.

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

v_d, v_q : Each phase armature voltage

L_d, L_q : Each phase self inductance

i_d, i_q : Each phase armature current

$L_d = l_a + 3/2(L_a - L_{as})$, $L_q = l_a + 3/2(L_a + L_{as})$

ψ_a : Value of armature interlinkage flux depending on permanent magnet

$\psi_a = \sqrt{3/2}\psi$

R_a : Each phase armature resistance

Based on this, it can be assumed that 3 phase alternating current motor system is 2 phase direct current motor system.

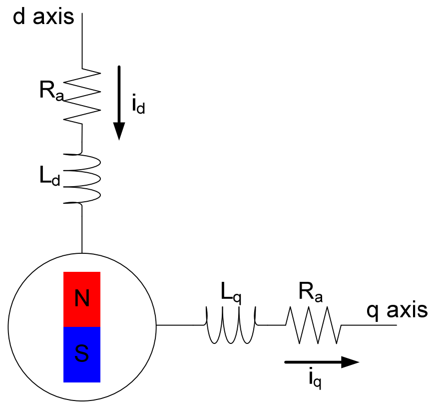


Figure 3-2 Conceptual diagram of the two phase direct current motor

Size of the torque generated in the motor can be obtained as follows from the exterior product of the electric current vector and armature inter-linkage magnetic flux. The first term on the right side of this formula is called magnetic torque and the second term on the right side of this formula is the reluctance torque.

$$T = P_n \left\{ \psi_a i_q + (L_d - L_q) i_d i_q \right\}$$

T : Motor torque P_n : Number of pole pairs

The motor which has no difference between the d axis and q axis inductance is defined as a motor which does not have saliency. In this case, as the reluctance torque is 0,the torque increases proportionally to the q axis current. Due to this, the q axis current is called torque current. On the other hand, d axis current is sometimes called excitation current, because the d axis current's operation to change its size can be assumed that the size of magnetic flux of permanent magnet is changing for q axis voltage.

As SPMSM generally does not have saliency, the d axis current unnecessary for generating torque is controlled to 0 while controlling the speed. This is known as id = 0 control. On one hand, the motion equation of the motor in this case is expressed as follows. This equation shows that motor speed is increased by increasing the q axis current.

$$I \frac{d\omega}{dt} = P_n \psi_a i_q - T_L$$

T_L : Load torque I : Motor intertia momentum

This system uses not motion equation but PI control for speed control. The q axis current command value is calculated by the following formula.

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

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

To achieve early stabilization, the PI control is also used for the d axis and q axis current values. A command voltage value is acquired by current PI control.

$$v_d^* = (K_{Pi_d} + \frac{K_{Ii_d}}{s})(i_d^* - i_d)$$

K_{Pi_d} : d axis current PI propotional gain K_{Ii_d} : d axis current PI integral gain

$$v_q^* = (K_{Pi_q} + \frac{K_{Ii_q}}{s})(i_q^* - i_q)$$

K_{Pi_q} : q axis current PI propotional gain K_{Ii_q} : q axis current PI integral gain

Inductive voltage is generated when the motor is rotated. The effect on d axis voltage due to q axis current and on q axis voltage due to d axis current and magnetic flux of permanent magnet becomes significant along with the increase in speed. This d axis and q axis interference may delay the stability of a current value. In order to avoid this, the voltage of each axis is calculated by performing feedforward so that the interference term of each axis can be canceled beforehand.

$$v_d^* = (K_{Pi_d} + \frac{K_{Ii_d}}{s})(i_d^* - i_d) - \omega L_q i_q$$

$$v_q^* = (K_{Pi_q} + \frac{K_{Ii_q}}{s})(i_q^* - i_q) + \omega (L_d i_d + \psi_a)$$

This method to eliminate the effect of the interference term is known as decoupling control. This enables to control the d axis and q axis independently.

Vector control is a method by which the 3 phase alternating current motor is converted to the 2 phase direct current motor that can be controlled each phase (d,q) independently while managing the position, speed of the torque and rotor. Control flow of the vector control is shown below.

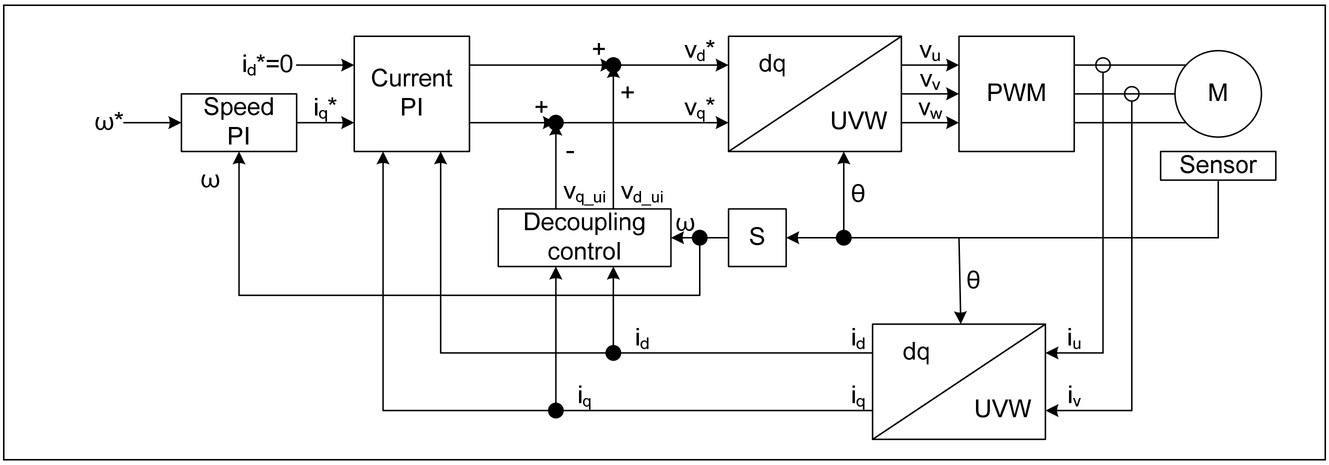


Figure 3-3 Control Flow of the Vector Control

3.3 Sensorless vector control based on the current estimation error

For the vector control, position sensors of the encoder and resolver etc are required as voltage is set according to the rotor position. When the position sensors are not used, in other words, in the case of the sensorless vector control, it is necessary to estimate the position by some methods. These days, the demand for motor control by sensorless has increased and several methods are provided for estimating the position. This part introduces the sensorless vector control used in this system, which is using current estimation error.

Position of the d axis is not clear as the position information of the actual motor is not available. As shown in the below figure, when γ axis is set in the location which lags behind by Δθ from the d axis and δ axis is set in the location 90 degrees ahead of the γ axis, the conversion formula from d q axis to the γ δ axis can be indicated as follows.

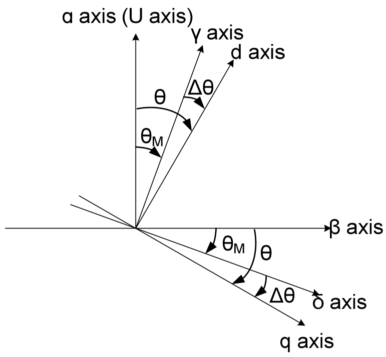


Figure 3-4 Relation between d q axis and γ δ axis

$$\begin{pmatrix}\gamma\\ \delta\end{pmatrix}=\begin{pmatrix}\cos\Delta\theta & \sin\Delta\theta\\ -\sin\Delta\theta & \cos\Delta\theta\end{pmatrix}\begin{pmatrix}d\\ q\end{pmatrix}$$

The equation in which above is applied to the SPMSM voltage equation and written in the electric current state equation format is as follows.

$$p\begin{pmatrix}i_\gamma\\ i_\delta\end{pmatrix}=-\begin{pmatrix}\frac{R}{L} & -\dot{\theta}_M\\ \dot{\theta}_M & \frac{R}{L}\end{pmatrix}\begin{pmatrix}i_\gamma\\ i_\delta\end{pmatrix}+\frac{1}{L}\begin{pmatrix}v_\gamma\\ v_\delta\end{pmatrix}-\frac{K_E\dot{\theta}}{L}\begin{pmatrix}-\sin\Delta\theta\\ \cos\Delta\theta\end{pmatrix}$$

Discretization is performed by using backward differential approximation (Euler’s approximation) to this state equation.

$$\begin{pmatrix}i_\gamma(n)\\ i_\delta(n)\end{pmatrix}=\begin{pmatrix}i_\gamma(n-1)\\ i_\delta(n-1)\end{pmatrix}+\frac{T}{L}\left\{\begin{pmatrix}v_\gamma(n-1)\\ v_\delta(n-1)\end{pmatrix}-R\begin{pmatrix}i_\gamma(n-1)\\ i_\delta(n-1)\end{pmatrix}-\dot{\theta}_M(n-1)L\begin{pmatrix}-i_\delta(n-1)\\ i_\gamma(n-1)\end{pmatrix}-e(n-1)\begin{pmatrix}-\sin\Delta\theta(n-1)\\ \cos\Delta\theta(n-1)\end{pmatrix}\right\}$$

$$\because e(n-1)=K_E\dot{\theta}(n-1)$$

As a motor model here, given that the motor parameters are written as R_M, L_M and e_M which are sufficiently equal to motor parameters of an actual motor and Δθ is set to 0, the current value at a sample point n can be represented as follows.

$$\begin{pmatrix} i_{\gamma M}(n) \\ i_{\delta M}(n) \end{pmatrix} = \begin{pmatrix} i_\gamma(n-1) \\ i_\delta(n-1) \end{pmatrix} + \frac{T}{L_M}\left\{ \begin{pmatrix} v_\gamma(n-1) \\ v_\delta(n-1) \end{pmatrix} - R_M \begin{pmatrix} i_\gamma(n-1) \\ i_\delta(n-1) \end{pmatrix} - \dot{\theta}_M(n-1) L_M \begin{pmatrix} -i_\delta(n-1) \\ i_\gamma(n-1) \end{pmatrix} - e_M(n-1)\begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

Depending on the difference between actual motor current and motor model current, the current estimation error can be indicated as follows.

$$\begin{pmatrix} \Delta i_\gamma(n) \\ \Delta i_\delta(n) \end{pmatrix} = \frac{T}{L}\begin{pmatrix} e(n-1)\sin\Delta\theta(n-1) \\ e_M(n-1) - e(n-1)\cos\Delta\theta(n-1) \end{pmatrix}$$

When Δθ is sufficiently small, the current estimation error can be approximated as follows.

$$\begin{pmatrix} \Delta i_\gamma(n) \\ \Delta i_\delta(n) \end{pmatrix} \approx \frac{T}{L}\begin{pmatrix} e(n-1)\Delta\theta(n-1) \\ -\Delta e(n-1) \end{pmatrix}$$
$$\Delta e(n-1) = e(n-1) - e_M(n-1)$$

If both Δe and Δθ are 0, it can be considered that the actual model is synchronized with the motor model. e_M is estimated by feeding back Δi_δ such that Δe becomes 0. Similarly, the θ_M value is estimated by feeding back Δi_γ such that Δθ becomes 0. The motor model is thus matched with the actual model. The e_M estimation equation can be expressed as follows.

$$e_M(n) = e_M(n-1) - K_e \Delta i_\delta(n)$$

Here, K_e is the speed electromotive force gain. Similarly, the θ_M estimation equation can be written as follows.

$$\theta_M(n) = \theta_M(n-1) + \frac{T}{K_{EM}} e_M(n) + K_\theta \operatorname{sgn}\{\dot{\theta}_M(n-1)\}\Delta i_\gamma(n)$$
$$\operatorname{sgn}\{\dot{\theta}_M(n-1)\} = \begin{cases} 1 & ;\ \dot{\theta}_M(n-1) \geq 0 \\ -1 & ;\ \dot{\theta}_M(n-1) < 0 \end{cases}$$

Here, K_{EM} is the electromotive force coefficient of the motor model and K_θ is the position estimation gain. Also, $p\theta_M$ sign is used instead of the pθ sign. The speed can be written as follows based on the formula shown above.

$$\dot{\theta}_M = \frac{1}{T}\{\theta_M(n) - \theta_M(n-1)\} = \frac{e_M}{K_{EM}} + \Delta\dot{\theta}_M(n)$$

$$\Delta\dot{\theta}_M(n) = \frac{K_\theta}{T}\operatorname{sgn}\{\dot{\theta}_M(n-1)\}\Delta i_\gamma(n)$$

In the control, LPF for the speed correction term is used as follows. Here, 0 < K < 1.

$$\dot{\theta}_{Mo}(n) = \frac{e_M(n)}{K_{EM}} + \Delta\dot{\theta}_{Mo}(n)$$

$$\Delta\dot{\theta}_{Mo}(n) = \Delta\dot{\theta}_{Mo}(n-1) + K\{\Delta\dot{\theta}_M(n) - \Delta\dot{\theta}_{Mo}(n-1)\}$$

Control flow of this control method is shown below.

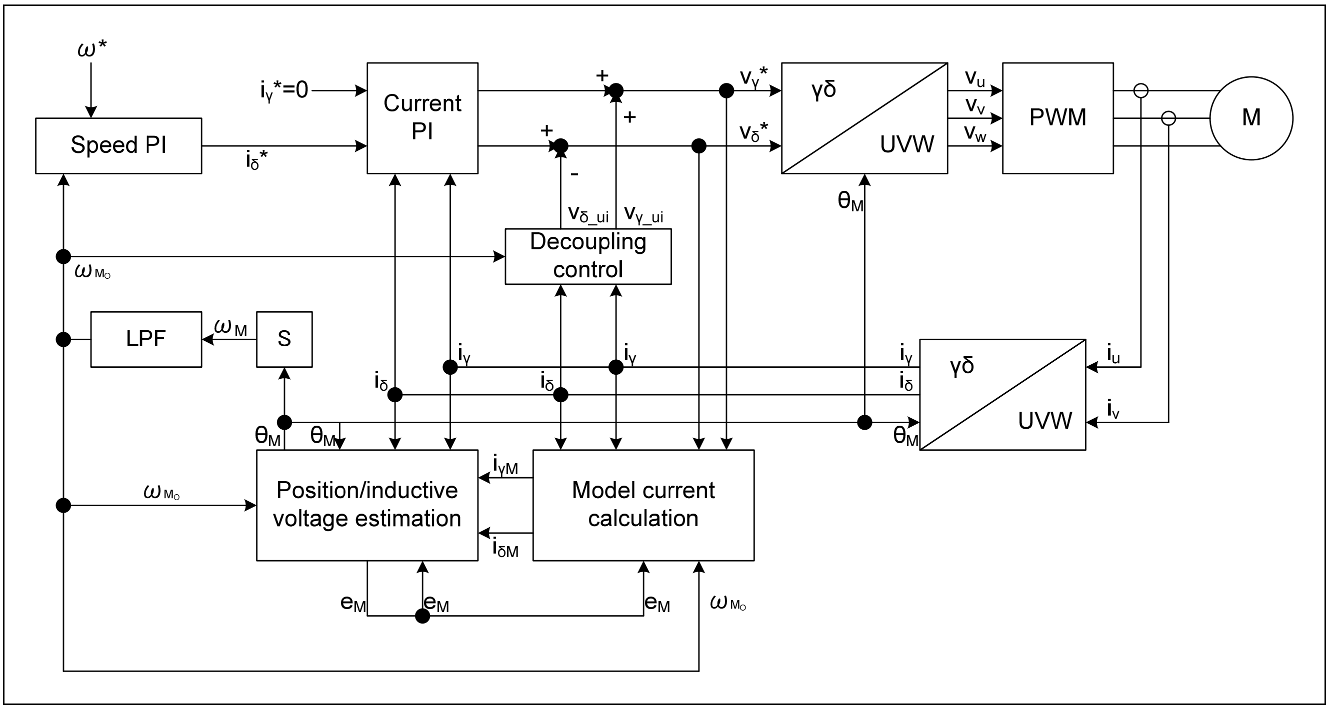


Figure 3-5 Control flow of the sensorless vector control based on the current estimation error method

3.4 Triangular wave comparison method

In order to actually output the command value voltage, the triangular wave comparison method which determines the pulse width of the output voltage by comparing the carrier waveform (triangular wave) and voltage command value waveform is used. By using this PWM formula, output of the voltage command value of the pseudo sinusoidal wave can be performed.

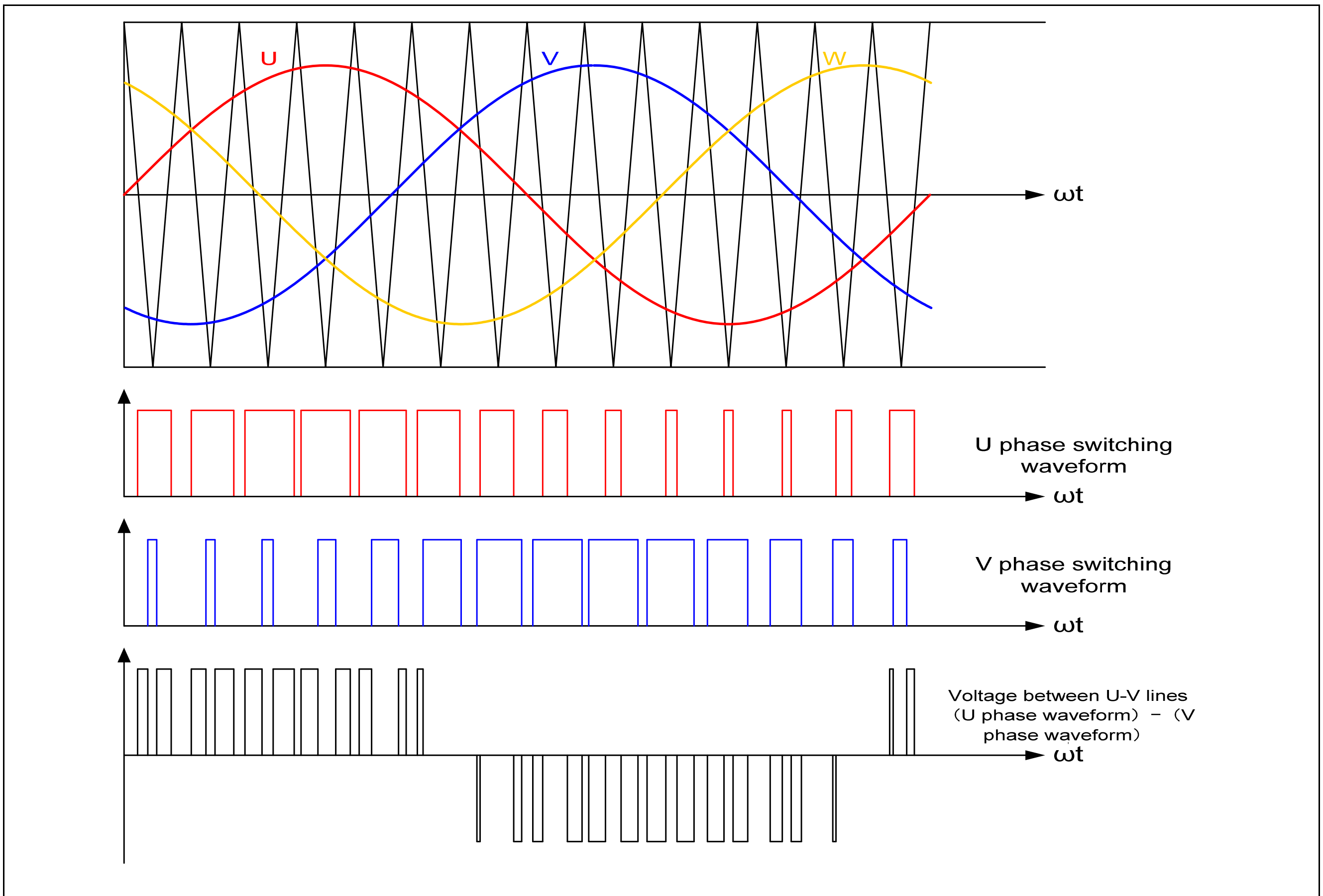


Figure 3-6 Conceptual diagram of the triangular wave comparison method

Here, as shown in the Figure 3-7, ratio of the output voltage pulse to the carrier wave is called as duty.

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

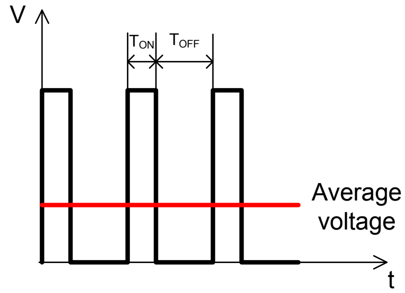


Figure 3-7 Definition of duty

Modulation factor m is defined as follows.

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

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

A requested control can be performed by setting this modulation factor to the register which determines PWM duty.

4. Description of the control program

Control program of this system is explained here.

4.1 Contents of Control

4.1.1 Motor start/stop

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

A general-purpose port (P91) is assigned to SW1. The P91 port is read within the main loop. When P91 is at a “Low” level, it is determined that the start switch is being pressed. Conversely, when the level is switched to “High”, the program determines that the motor should be stopped.

4.1.2 Motor rotation speed command value, inverter bus voltage, motor 3 phase voltage

(1) Motor rotation speed command value

The motor rotation speed command value can be set by A/D conversion of the VR1 value (analog value). 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 the Speed Command Value

Item		Conversion ratio (Command value: A/D conversion value)	Channel
Rotation speed command value	CW	600 [rpm] to 2000 [rpm]: 0000H to 0FFFH	AN102

(2) Inverter bus voltage

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

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

Table 4-2 Inverter Bus Voltage Conversion Ratio

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

(3) U, W phase current

The U, W phase currents are measured as shown in Table 4-3 and used in vector control.

Table 4-3 Conversion Ratio of U and W Phase Current

Item	Conversion ratio (U, W phase current: A/D conversion value)	Channel
U, W phase current	-10 [A] to 10 [A]: 0000H to 0FFFH	AN000, AN001

4.1.3 Control method

The motor is driven in an open loop at the time of startup. After a fixed time has passed, the motor is driven by the sensorless vector control based on the current estimation error explained in chapter 3 (please refer to the block diagram in Figure 3-5). PI control is used to control the speed.

4.1.4 System protection function

This control program has the following four types of error status and executes emergency stop functions in case of occurrence of respective errors.

・Over current error

High impedance output is made to the PWM output port in response to an emergency stop signal (over current detection) from the hardware (emergency stop without involving CPU). In addition, U, V, and W phase currents are monitored by 100 [μs] intervals. When an over current (when the current exceeds 10 [A]) is detected, the CPU executes emergency stop.

・Over voltage error

The inverter bus voltage is monitored by 100 [μs] intervals. When an over voltage is detected (when the voltage exceeds 28 [V]), the CPU performs emergency stop. Here, the over voltage limit value 28 [V] is set by considering the error of resistance value and error of supply voltage by AC adapter etc.

・Low voltage error

The inverter bus voltage is monitored by 100 [μs] intervals. The CPU performs emergency stop when low voltage (when voltage falls below 0 [V]) is detected.

・Over speed error

The rotation speed is monitored by 100 [μs] intervals. The CPU performs emergency stop when the speed is over 1600 [rad/s] (electrical angle)

4.2 Function Specifications

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

For detailed processing, please refer to flowcharts or source files.

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

File name	Function name	Process overview
main.c	main Input: None Output: None	・Hardware initialization function call ・User interface initialization function call ・Initialization function call of the variable used in the main process ・Status transition and event execution function call ・Main process ⇒Main process execution function call ⇒Watchdog timer clear function call
	ctrl_ui Input: None Output: None	・Motor status change ・Determination of rotation speed command value
	software_init Input: None Output: None	Initialization of variables used in the main process
mtr_ctrl_rssk.c	get_vr1 Input: None Output: (int16) ad_data / A/D conversion result	VR1 status acquisition
	get_sw1 Input: None Output: (uint8) tmp_port / SW1 level	SW1 status acquisition
	get_sw2 Input: None Output: (uint8) tmp_port / SW2 level	SW2 status acquisition
	led1_on Input: None Output: None	Making LED1 ON
	led2_on Input: None Output: None	Making LED2 ON
	led1_off Input: None Output: None	Making LED1 OFF
	led2_off Input: None Output: None	Making LED2 OFF

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

File name	Function name	Process overview
mtr_ctrl_rx62t.c	R_MTR_InitHardware Input: None Output: None	Initialization of the clock and peripheral functions
	init_ui Input: None Output: None	Initialization of the peripheral functions used by the user
	mtr_ctrl_start Input: None Output: None	Motor startup process
	mtr_ctrl_stop Input: None Output: None	Motor stop process
	mtr_get_vr1 Input: None Output: (uint16)u2_temp /VR1 AD conversion value	VR1 AD conversion
	mtr_get_iuiwvdc Input: (float32) *f4_iu_ad / U phase current AD conversion value : (float32) *f4_iw_ad / W phase current AD conversion value : (float32) *f4_vdc_ad / Vdc AD conversion value Output: None	AD conversion of U phase current, W phase current, and inverter bus voltage
	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_mtu4_flag Input: None Output: None	Clearing the interrupt flag
	mtr_clear_cmt0_flag Input: None Output: None	Clearing the interrupt flag
	mtr_inv_set_uvw Input: (float32) f4_u / U phase voltage : (float32) f4_v / V phase voltage : (float32) f4_w / W phase voltage : (float32) f4_vdc / Vdc Output: None	PWM output setting

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

File name	Function name	Process overview
mtr_interrupt.c	mtr_over_current_interrupt Input: None Output: None	Overcurrent detection process ・Event processing selection function call ・Changing the motor status ・High impedance state clearing function call
	mtr_mtu4_interrupt Input: None Output: None	Calling per 100 [μs] ・Vector control ・Current PI control
	mtr_cmt0_interrupt Input: None Output: None	Calling per 1 [ms] ・Start control ・Speed PI control

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

File name	Function name	Process overview
mtr_ssns_less_foc.c	R_MTR_InitSequence Input: None Output: None	Initialization of the sequence process
	R_MTR_ExecEvent Input: (uint8)u1_event/ occurred event Output: None	・Changing the status ・Calling an appropriate process 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
	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_angle_speed Input: None Output: None	Position and speed calculation process
	mtr_start_init Input: None Output: None	Initializing only the variables required for motor startup
	mtr_pi_ctrl Input: MTR_PI_CTRL *vdq/ PI control structure Output: (float32)f4_ref / PI control output value	Current PI control
	R_MTR_SetSpeed Input: (float32)ref_speed / Speed command value Output: None	Speed command value setting
	R_MTR_GetSpeed Input: None Output: (float32) g_f4_speed_rad / speed	Obtaining the speed calculation value
	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

4.3 List of variables

Lists of variables used in this control program are given below. However, the local variables are not mentioned.

Table 4-5 List of Variables (1/2)

Variable name	Type	Content	Remarks
g_f4_max_mecha_speed_rad	float32	Speed command maximum value	Mechanical angle [rad/s]
g_f4_min_mecha_speed_rad	float32	Speed command minimum value	Mechanical angle [rad/s]
g_f4_set_speed	float32	User rotation speed command value	Electrical angle [rad/s]
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 at the time of error status 1: Turning SW2 OFF at the time of error status
g_u1_sw1_cnt	uint8	SW1 determining counter	Chattering removal
g_u1_sw2_cnt	uint8	SW2 determining counter	Chattering removal
g_u1_mode_system	uint8	State management	0: Stop mode 1: Run mode 2: Error mode
g_u2_run_mode	uint16	Operation mode management	2: Start mode 6: Normal operation mode
g_u1_error_status	uint8	Error status management	1: Over current error 2: Over voltage error 3: Over speed error 7: Low voltage error 0xFF: Undefined error
g_u1_cnt_ics	uint8	Counter for ICS call	
g_f4_vdc_ad	float32	Inverter bus voltage A/D value	[V]
g_f4_vd_ref	float32	d axis voltage command value	Current PI control output value [V]
g_f4_vq_ref	float32	q axis voltage command value	Current PI control output value [V]
g_f4_iu_ad	float32	U phase current	[A]
g_f4_iv_ad	float32	V phase current	[A]
g_f4_iw_ad	float32	W phase current	[A]
g_f4_offset_iu	float32	U phase current offset value	[A]
g_f4_offset_iw	float32	W phase current offset value	[A]
g_f4_id_lpf	float32	d axis current	[A]
g_f4_iq_lpf	float32	q axis current	[A]
g_f4_ex_id_lpf	float32	Previous value of d axis current	[A]
g_f4_ex_iq_lpf	float32	Previous value of q axis current	[A]
g_f4_kp_id	float32	d axis current PI proportional term gain	
g_f4_ki_id	float32	d axis current PI integral term gain	
g_f4_kp_iq	float32	q axis current PI proportional term gain	
g_f4_ki_iq	float32	q axis current PI integral term gain	

Table 4-5 List of Variables (2/2)

Variable name	Type	Content	Remarks
g_f4_kp_speed	float32	Speed PI control proportional term gain	
g_f4_ki_speed	float32	Speed PI control integral term gain	
g_f4_lim_id	float32	d axis current PI control output limit value	[V]
g_f4_lim_iq	float32	q axis current PI control output limit value	[V]
g_f4_ilim_id	float32	d axis current PI control integral term limit value	[V]
g_f4_ilim_iq	float32	q axis current PI control integral term limit value	[V]
g_f4_id_ref	float32	d axis current command value	[A]
g_f4_iq_ref	float32	q axis current command value	[A]
g_f4_speed_rad	float32	Speed operation value	Electrical angle [rad/s]
g_f4_ref_speed_rad	float32	Speed command value	Electrical angle [rad/s]
g_f4_ref_speed_rad_ad	float32	Speed adjustment value	Electrical angle [rad/s]
g_f4_angle_rad	float32	Rotor position	Electrical angle [rad]
g_f4_max_speed_rad	float32	Maximum speed value	Electrical angle [rad/s]
g_f4_min_speed_rad	float32	Minimum speed value	Electrical angle [rad/s]
g_f4_iq_pip	float32	Speed PI control proportional term	[A]
g_f4_iq_pii	float32	Speed PI control integral term	[A]
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]
vd	MTR_PI_CTRL	d axis current PI control structure	
vq	MTR_PI_CTRL	q axis current PI control structure	
g_u1_flag_id_open	uint8	Start mode determining flag 1	
g_u1_flag_wr_open	uint8	Start mode determining flag 2	
g_u2_cnt_adjust	uint16	Counter for current offset calculation	
g_f4_id_open	float32	d axis current command value in start mode	[A]
g_f4_ol_speed_rad	float32	Speed in start mode	[rad/s]
g_u2_cnt_wr_open	uint16	Counter in start mode	
g_f4_i_gamma	float32	γ axis current	[A]
g_f4_i_delta	float32	δ axis current	[A]
g_f4_di_gamma	float32	γ axis current error	[A]
g_f4_di_delta	float32	δ axis current error	[A]
g_f4_emf_est	float32	Estimation value of inductive voltage	[V]
g_f4_k_emf	float32	Speed electromotive force estimation gain	
g_f4_k_theta	float32	Position estimation gain	
f4_tdspeed_lpf	float32	Control cycle × difference in speed	
g_u1_def_state	uint8	Motor status definition	Array members ・Stop mode ・Run mode ・Error mode
gp_u1_def_action	uint8	Action definition	Array members ・Stop action ・Run action ・Error action ・Reset action ・No action

4.4 Macro definitions

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

Table 4-6 List of Macro Definitions (1/5)

File name	Macro name	Definition value	Remarks
main.h	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	Constant for creating rotation speed command (mechanical angle) [rpm]
	MARGIN_MAX_SPEED	MAX_SPEED + MARGIN_SPEED	Constant for creating rotation speed command maximum value (mechanical angle) [rpm]
	MARGIN_MIN_SPEED	MIN_SPEED - MARGIN_SPEED	Constant for creating rotation speed command minimum value (mechanical angle) [rpm]
	PI	3.14159265f	Circular constant (pi)
	RPM_RAD	(2*PI)/60	Constant for converting unit: [rpm] to [rad/s]
	SW_ON	0	Active in case of “Low”
	SW_OFF	1	
	CHATTERING_CNT	10	Chattering removal
	VR1_SCALING	(MARGIN_MAX_SPEED - MARGIN_MIN_SPEED) / 4095.0f	Rotation speed command value creation constant
	POLE_PAIR	7	Number of pole pairs

Table 4-6 List of Macro Definitions (2/5)

File name	Macro name	Definition value	Remarks
mtr_ctrl_rx62t.h	MTR_PWM_TIMER_FREQ	96.0f	PWM timer count frequency [MHz]
	MTR_CARRIER_FREQ	20.0f	Carrier frequency [kHz]
	MTR_DEADTIME_SET	MTR_DEADTIME * MTR_PWM_TIMER_FREQ	Dead time
	MTR_CARRIER_SET	((MTR_PWM_TIMER_FREQ * 1000 / MTR_CARRIER_FREQ / 2)+ MTR_DEADTIME_SET)	Carrier setting value
	MTR_HALF_CARRIER_SET	MTR_CARRIER_SET / 2	Carrier setting value /2
	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	

Table 4-6 List of Macro Definitions (3/5)

File name	Macro name	Definition value	Remarks
mtr_ssns_less_foc.h	MTR_DEADTIME	2	Dead time [μs]
	MTR_INT_DECIMATION	1	Number of interrupt decimation times
	MTR_CTRL_PERIOD	(MTR_INT_DECIMATION + 1)/(MTR_CARRIER_FREQ*1000)	Control cycle [s]
	MTR_CONTROL_FREQ	(MTR_CARRIER_FREQ*1000)/(MTR_INT_DECIMATION + 1)	Control frequency [Hz]
	MTR_M	0.006198f	Magnetic Flux [Wb]
	MTR_R	0.453f	Resistance [Ω]
	MTR_L	0.0009447f	Inductance [H]
	MTR_T_L	MTR_CTRL_PERIOD/MTR_L	T/L [s/H]
	MTR_T_M	MTR_CTRL_PERIOD/MTR_M	T/M [s/Wb]
	MTR_SPEED_LIMIT	1600	Speed limit value (electrical angle) [rad/s]
	MTR_OVERCURRENT_LIMIT	10	Current limit value [A]
	MTR_OVERVOLTAGE_LIMIT	28	Upper limit of voltage value [V]
	MTR_UNDERVOLTAGE_LIMIT	0	Lower limit of voltage value [V]
	MTR_TWOPI	2*3.14159265	2π
	MTR_SQRT_2_3	0.81649658	$\sqrt{(2/3)}$
	MTR_HALF_VDC	12	Power supply voltage/2 [V]
	MTR_ADC_SCALING	0x7FF	Constant for ADC offset adjustment
	MTR_CURRENT_SCALING	20.0f/4095.0f	Current A/D conversion value resolution
	MTR_VDC_SCALING	30.0f/4095.0f	Inverter bus voltage A/D conversion value resolution
	MTR_ID_PI_KP	3	d axis current PI control proportional term gain
	MTR_ID_PI_KI	0.0001	d axis current PI control integral term gain
	MTR_IQ_PI_KP	3	q axis current PI control proportional term gain
	MTR_IQ_PI_KI	0.0005	q axis current PI control integral term gain
	MTR_SPEED_PI_KP	0.001	Speed PI control proportional term gain
	MTR_SPEED_PI_KI	0.0005	Speed PI control integral term gain
	MTR_EMF_EST_K	0.1	Speed electro-motive force estimation gain
	MTR_THETA_EST_K	0.1	Position estimation gain
	MTR_LPF_K	0.04	LPF coefficient
	MTR_IQ_LIMIT	3	q axis current command limit value [A]

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

File name	Macro name	Definition value	Remarks
mtr_ssns_less_foc.h	MTR_LIMIT_VD	11	d axis current PI control output limit value [A]
	MTR_LIMIT_VQ	11	q axis current PI control output limit value [A]
	MTR_I_LIMIT_VD	11	d axis current PI control integral term limit value [A]
	MTR_I_LIMIT_VQ	11	q axis current PI control integral term limit value [A]
	MTR_MAX_SPEED_RAD	1470	Maximum speed (electrical angle) [rad/s]
	MTR_MIN_SPEED_RAD	440	Minimum speed (electrical angle) [rad/s]
	MTR_START_OL_ID	1.0f	d axis current in start mode [A]
	MTR_START_OL_ID_UP_TIME	256.0f	d axis current adding time[ms]
	MTR_START_OL_ID_DOWN_TIME	256.0f	d axis current subtracting time [ms]
	MTR_START_OL_REF_ID	MTR_START_OL_ID	d axis current command value in start mode [A]
	MTR_START_OL_ID_UP_STEP	MTR_START_OL_ID/ MTR_START_OL_ID_UP_TIME	Command d axis current adding value [A]
	MTR_START_OL_ID_DOWN_STEP	MTR_START_OL_ID/ MTR_START_OL_ID_DOWN_TIME	Command d axis current subtracting value [A]
	MTR_START_OL_IQ	0.4f	q axis current value at the speed PI control start [A]
	MTR_START_OL_IQ_UP_STEP	MTR_START_OL_IQ/ MTR_START_OL_SPEED_CONST_TIME	Command q axis current adding value [A]
	MTR_START_OL_SPEED	70*MTR_TWOPI	Maximum speed in start mode [rad/s]
	MTR_START_OL_SPEED_UP_TIME	1024	Speed adding time in open loop mode [ms]
	MTR_START_OL_SPEED_CONST_TIME	128	Constant speed time in open loop mode [ms]
	MTR_START_REF_SPEED_CONST_TIME	512	Time during which speed command value is constant after speed PI control start [ms]
	MTR_START_OL_REF_SPEED	MTR_START_OL_SPEED	Speed command value in start mode (electrical angle) [rad/s]
	MTR_START_OL_SPEED_UP_STEP	MTR_START_OL_REF_SPEED /MTR_START_OL_SPEED_UP_TIME	Command speed adding time in open loop mode (electrical angle) [rad/s]
	MTR_START_REF_SPEED_UP_STEP	(MTR_MAX_SPEED_RAD - MTR_MIN_SPEED_RAD)/ MTR_START_OL_ID_DOWN_TIME	Adding value for reflecting command speed by VR1 in start mode (electrical angle) [rad/s]
	MTR_START_REF_SPEED_DOWN_STEP	(MTR_MAX_SPEED_RAD - MTR_MIN_SPEED_RAD)/ MTR_START_OL_ID_DOWN_TIME	Subtracting value for reflecting command speed by VR1 in start mode (electrical angle) [rad/s]

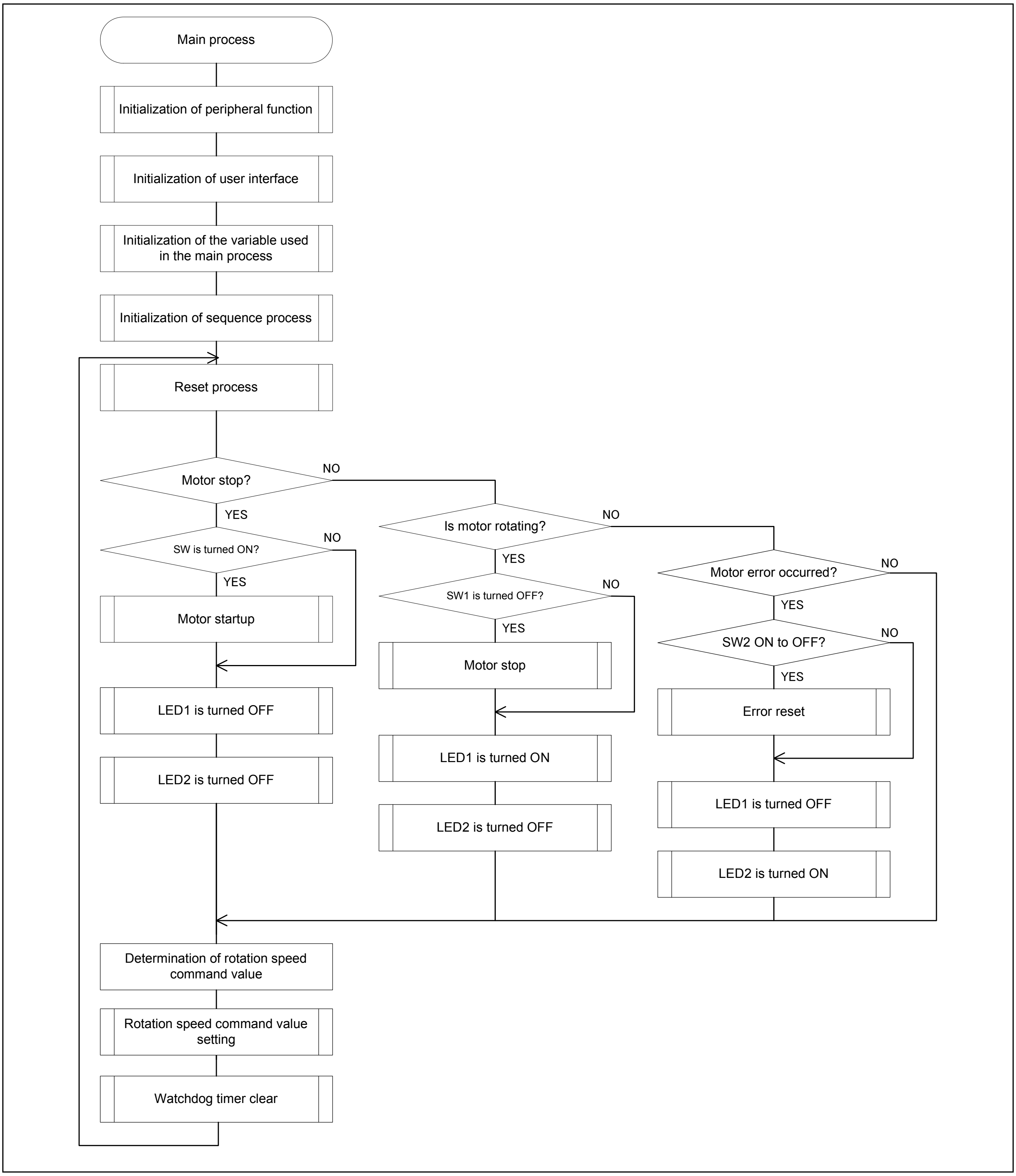
RENESAS

Table 5-10 List of Macro Definitions (5/5)

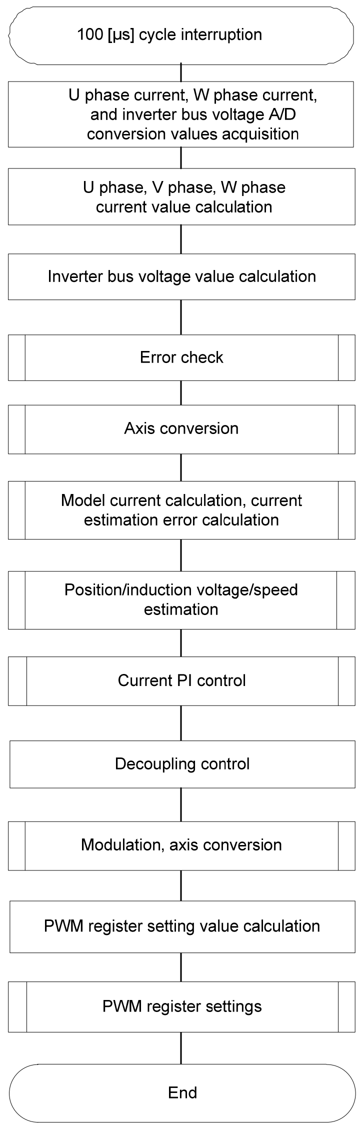
File name	Macro name	Definition value	Remark
mtr_ssns_less_foc.h	MTR_BOOT_MODE	0x00	Boot mode
	MTR_OPENLOOP_MODE	0x01	Open loop mode
	MTR_START_MODE	0x02	Start mode
	MTR_HALL_120_MODE	0x03	Hall sensor 120-degree operation mode
	MTR_BEMF_120_MODE	0x04	BEMF sensorless 120-degree operation mode
	MTR_ENCD_FOC_MODE	0x05	Encoder vector operation mode
	MTR_LESS_FOC_MODE	0x06	Sensorless vector operation mode
	MTR_OVER_CURRENT_ERROR	0x01	Over current error
	MTR_OVER_VOLTAGE_ERROR	0x02	Over voltage error
	MTR_OVER_SPEED_ERROR	0x03	Over speed error
	MTR_TIMEOUT_ERROR	0x04	Timeout 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	0x03	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

4.5 Control flow (flow chart)

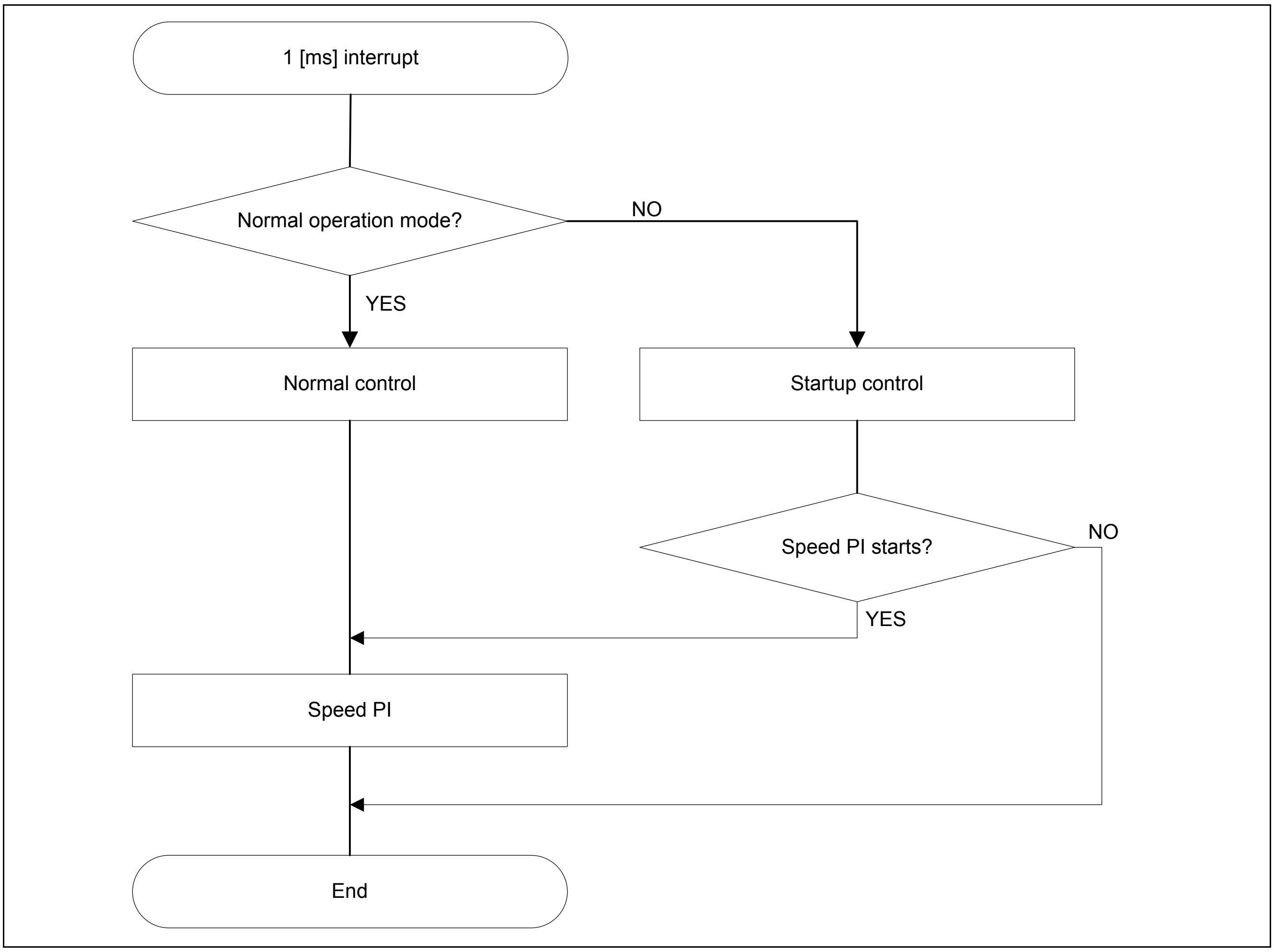
(1) Main process



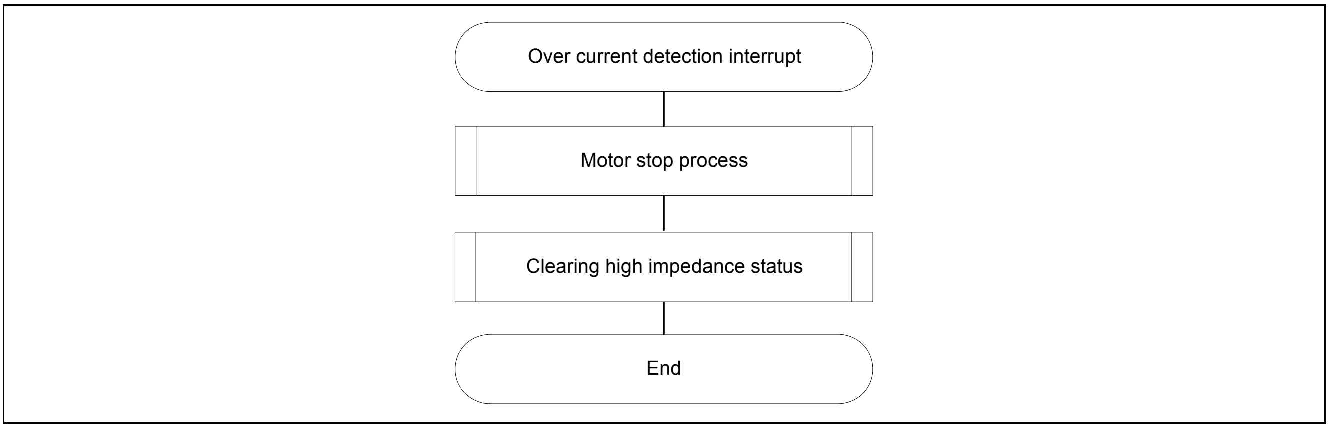
(2) 100 [μs] cycle interrupt process



(3) 1 [ms] interrupt process



(4) Over current interrupt process



Website and Support

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

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

Rev.	Issued on	Revision Details	
		Page	Summary
1.00	Apr 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 part 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