RZ/T1 Group

Encoder-Based Vector Control (Position Control) of the Permanent Magnet Synchronous Motor for the RZ/T1 Motion Control Solution Kit

R01AN4052EJ0110
Rev.1.10
Mar. 16, 2018

Introduction

This application note is designed to describe the sample program which drives the permanent magnet synchronous motor under encoder-based vector control (position control) by using the functions of the RZ/T1 Group.

This sample program is provided for reference only. Renesas does not guarantee the operation of the sample program. Before using this sample program, fully evaluate it in an appropriate environment.

Target Device

The operation of the sample program has been verified for the following device:

- RZ/T1 Group (R7S910018CBG)

Table of Contents

1. Overview 3
   1.1 System Configuration 3
   1.2 Development Environment 3
2. System Overview 4
   2.1 Hardware Configuration 4
   2.2 Hardware Specifications 6
      2.2.1 User Interface 6
      2.2.2 Peripheral Functions 7
   2.3 Software Configuration 8
      2.3.1 Software File Configuration 8
      2.3.2 Module Configuration 9
   2.4 Software Specifications 10
3. Motor Control Method 11
   3.1 Voltage Equations of the Motor Control System 11
   3.2 Vector Control 13
   3.3 Starting the Motor 16
   3.4 Calculating the Speed 17
   3.5 Comparing the Triangular Wave 18
4. Control Program 20
   4.1 Control 20
      4.1.1 Starting and Stopping the Motor 20
      4.1.2 Motor Axis Position Target Value, Inverter Bus Voltage, and Motor 3-Phase Voltage 21
      4.1.3 Control Method 23
      4.1.4 System Protection Function 23
   4.2 Function Specifications 24
   4.3 Functions 29
   4.4 Macro Definitions 32
   4.5 Control Flow (Flowchart) 36
5. Reference Documents 40

1. Overview

This application note describes the sample program for encoder-based vector control (position control) of the permanent magnet synchronous motor (hereafter called SPMSM) by using the RZ/T1 Group.

1.1 System Configuration

For encoder-based vector control, this system uses an inverter biplane board and a surface-mounted permanent magnet synchronous motor (MB057GA140) included in the RZ/T1 motion control solution kit.

For the purchase or technical support of the RZ/T1 motion control solution kit, contact Renesas Electronics Corporation or an authorized Renesas Electronics Corporation product distributor.

1.2 Development Environment

(1) Software development environment

Integrated development environment	• Embedded Workbench for ARM (V7.80) from IAR Systems • Renesas e^{2}studio (V6.1.0)

(2) Hardware environment

On-chip debug emulator	• Embedded Workbench for ARM environment: I-jet (from IAR Systems) • e^{2}studio environment: J-Link (from SEGGER)
Microcomputer used	RZ/T1 Group (R7S910018CBG)
Inverter board	Biplane
Motor	MB057GA140 (SPMSM)

2. System Overview

The overview of this system is described below.

2.1 Hardware Configuration

The figure below shows the hardware configuration.

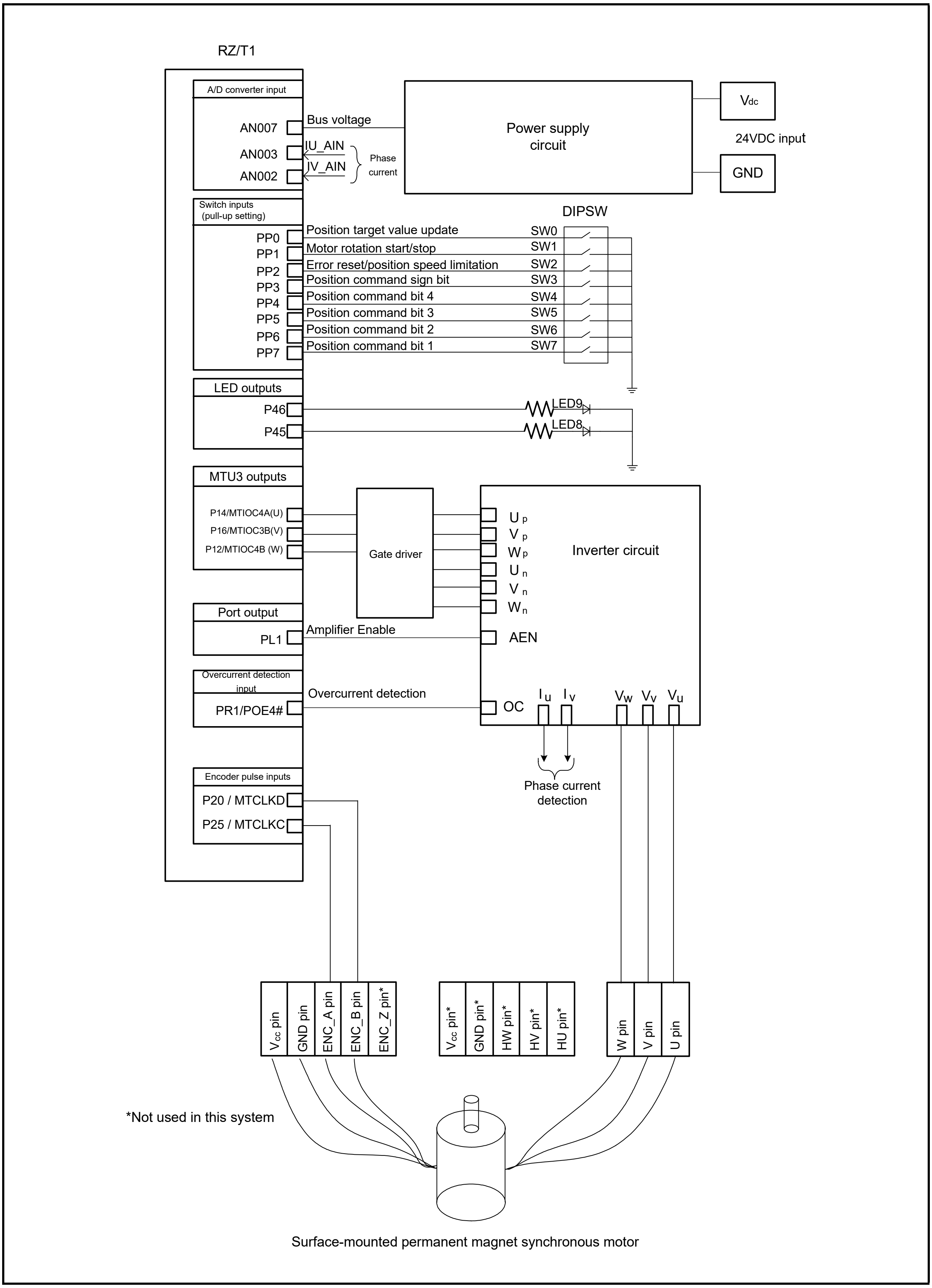


Figure 2.1 Hardware Configuration Diagram

2.2 Hardware Specifications

2.2.1 User Interface

Table 2.1 lists the user interface of this system.

This system operates only in Axis2. Connect the encoder connector to the P2 port and the motor connector to the J3 port.

Table 2.1 User Interface

Item	Interface Component	Function
Position target value update	DIP switch (SW0)	Updates the position target value (see Section 4.1.1)
Position target	DIP switch (SW3 to SW7)	Position target value input (see Section 4.1.2)
START/STOP	DIP switch (SW1)	Motor rotation start/stop command (see Section 4.1.1)
ERROR RESET	DIP switch (SW2)	Command to recover from the error state (Switched to ON and then to OFF if an error occurs)
Maximum speed control		Selects a maximum speed control value. (When the motor is rotating, ON setting: 25 rad/s, OFF setting: 100 rad/s)
LED9	Greenish yellow LED	• When the motor is rotating: Lit • When the motor is stopped: Extinguished
LED8	Greenish yellow LED	• When an error is detected: Lit • In normal operation: Extinguished
RESET	Push switch (RESET)	System reset

Table 2.2 lists the interface pins of this system (RZ/T1 Group).

Table 2.2 Interface Pins

Pin Name	Function
AN007	Inverter bus voltage measurement
PP0	DIP switch for updating the position target value
PP3	DIP switch for the position target sign bit
PP4	DIP switch for position target bit 4
PP5	DIP switch for position target bit 3
PP6	DIP switch for position target bit 2
PP7	DIP switch for position target bit 1
PP1	Start/stop DIP switch
PP2	Error reset DIP switch
P46	LED9 on/off control
P45	LED8 on/off control
AN003	U-phase current measurement
AN002	V-phase current measurement
P14 / MTIOC4A	PWM output (U)
P16 / MTIOC3B	PWM output (V)
P12 / MTIOC4B	PWM output (W)
PL1	AEN (Amplifier Enable)
PR1/POE4#	PWM emergency stop input when an overcurrent is detected
P25 / MTCLKC	Encoder A-phase input
P20 / MTCLKD	Encoder B-phase input

2.2.2 Peripheral Functions

Table 2.3 lists the peripheral functions used in this system.

Table 2.3 Peripheral Functions

Peripheral Function	Use
12-bit A/D converter (S12ADAa)	• Inverter bus voltage measurement • U-/V-phase current measurement
Compare match timer (CMT)	1-ms interval timer
Multi-function timer pulse unit 3 (MTU3a)	• PWM output (3 outputs) • Encoder input pulse count
Port output enable 3 (POE3)	When detecting an overcurrent, places the PWM output pin in a high-impedance state.

(1) 12-bit A/D converter

Measures the U-phase current (I_u), V-phase current (I_v) and inverter bus voltage (V_{dc}) by using the 12-bit A/D converter.

In unit 0, set the operating mode to 1-cycle scan mode that uses the sample & hold function.

(2) Compare match timer (CMT)

Uses channel 0 of the compare match timer as the 1-ms interval timer.

(3) Multi-function timer pulse unit 3 (MTU3a)

The operating mode differs between channels. Provides an output without a dead time (high active) by using complementary PWM mode through channels 3 and 4. Set so that only the normal phase is output because a counter phase is generated by the gate driver. Channel 2 counts the input pulses from the encoder by using phase count mode.

(4) Port output enable 3 (POE3)

Places the PWM output pin in the high-impedance state when detecting an overcurrent (falling edge of the POE4# pin) and a short circuit of the output.

2.3 Software Configuration

2.3.1 Software File Configuration

The table below lists the folders and file structure of the sample program.

The table includes only those necessary for the motor control sample program. Those files not directly related to motor control (such as board initialization) are not listed.

Table 2.4 Folders and File Structure of the Sample Program

RZT1_BIPLANE_SSNS_ENCD_FOC_CPOS	inc			app_common.h	Common application header
				iodefine.h	I/O register definition header in the RZ/T1 Group
				r_typedefs.h	Common TYPEDEF definition header
				r_common.h	Common driver header
				r_adc_if.h	S12ADC driver API header
				r_cmt_if.h	CMT driver API header
				r_io_port_if.h	I/O port driver API header
				r_mtu_if.h	MTU3a driver API header
				r_mtr_api.h	Drive library API header
				mtr_seq.h	Drive library internal header (sequence controller)
				mtr_ctrl.h	Drive library internal header (motor controller)
	src	drv	adc	r_adc_if.c	S12ADC driver function
			cmt	r_cmt_if.c	CMT driver function
			gpio	r_io_port_if.c	I/O port driver function
			mtr	r_mtr_api.c	Drive library API function
				mtr_seq.c	Drive library internal function (sequence controller)
				mtr_ctrl.c	Drive library internal function (motor controller)
			mtu	r_mtu_if.c	MTU3a driver function
		sample		main.c	Main function, user interface control
				app_interrupt.c	Application interrupt service routine
				app_common.c	Common application function

2.3.2 Module Configuration

Figure 2.2 shows the module configuration of the sample program.

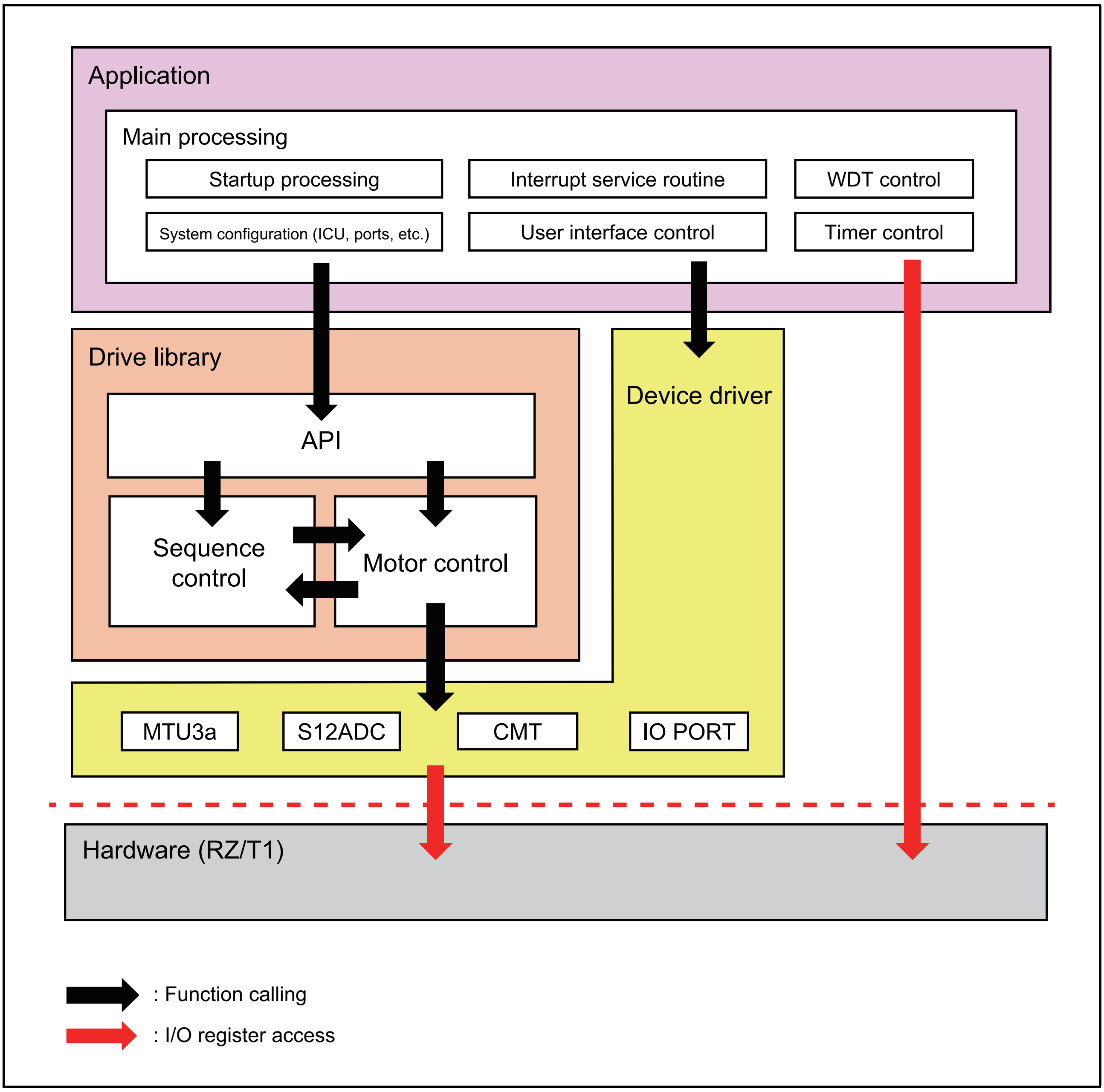


Figure 2.2 Module Configuration of the Sample Program

2.4 Software Specifications

Table 2.5 lists the basic specifications for the software in this system.

Table 2.5 Basic Software Specifications

Item	Description
Control method	Vector control (position control)
Motor rotation start/stop	Determined by the level of SW1 (PP1) (Low: rotation start, High: rotation stop)
Rotor magnetic pole position detection	Incremental encoder
Carrier frequency (PWM)	20 kHz
Control frequency	100 µs (twice the carrier frequency)
Setting range of the maximum speed control value	25 or 100 rad/s
Setting range of the torque limit	0.5 to 3.0 A
Position control range	–54000 to +54000 pulses
Protection stop	• Makes the three motor control signal outputs inactive if any of the following four conditions is met. 1. The current of each phase exceeds 4 A (monitored every 100 µs). 2. The inverter bus voltage exceeds 28 V (monitored every 100 µs). 3. The inverter bus voltage goes below 12 V (monitored every 100 µs). 4. The rotation speed exceeds 600 rad/s (electrical angle) (monitored every 100 µs). • Places the PWM output pin in a high-impedance state when detecting an overcurrent detection signal from outside (falling edge at the POE4# pin) and a short circuit of the output.

3. Motor Control Method

This section describes the vector control of the SPMSM used in the sample program.

3.1 Voltage Equations of the Motor Control System

The voltage equations of the permanent magnet synchronous motor (Figure 3.1) having a sine-wave magnetic flux distribution are expressed as follows.

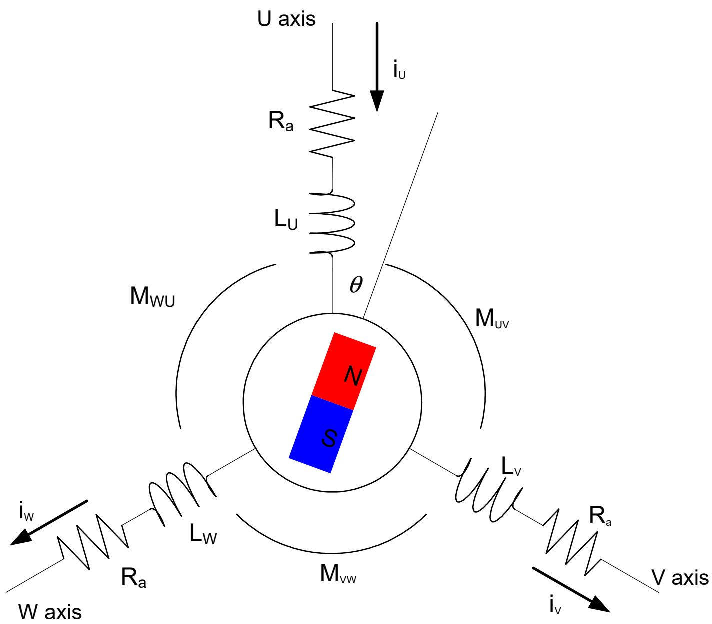


Figure 3.1 Conceptual Diagram of the 3-Phase Permanent Magnet Synchronous Motor

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

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

v_u, v_v, v_w : Armature voltage in each phase

i_u, i_v, i_w : Armature current in each phase

ϕ_u, ϕ_v, ϕ_w : Armature interlinkage magnetic flux in each phase

R_a : Armature current in each phase

p : Differential operator

L_u, L_v, L_w : Self inductance in each phase

M_{uv}, M_{vw}, M_{wu} : Mutual inductance in each phase

ψ : Maximum value of armature interlinkage magnetic flux by the permanent magnet

θ : Lead angle of the permanent magnet (rotor) from the U phase

Hence, the self inductance and the mutual inductance are represented as follows:

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

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

l_a: Leakage inductance per phase
L_a: Average of valid inductance per phase
L_{as}: Amplitude of valid inductance per phase

3.2 Vector Control

If the d axis is determined in the magnetic flux (N pole) direction of the permanent magnet of the rotor and the q axis in the direction 90 degrees ahead from the d axis, the following conversion matrix can be used to obtain the voltage equation of the permanent magnet synchronous motor viewed from the dq coordinate system.

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

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

After the coordinate transformation, the voltage equation of the permanent magnet synchronous motor in the dq coordinate system can be expressed as follows.

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

v_d , v_q: Armature voltage in each phase

i_d , i_q: Armature current in each phase

θ: Lead angle of the d axis (rotor) from the U phase

R_a: Armature resistance in each phase

L_d , L_q: Self inductance in each phase

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

ψ_a: Effective value of armature interlinkage magnetic flux by the permanent magnet

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

It can be considered here that the alternating current that was flowing through the quiescent 3-phase stator appears as the direct current in the 2-phase stator that is rotating in synchronization with the permanent magnet (rotor).

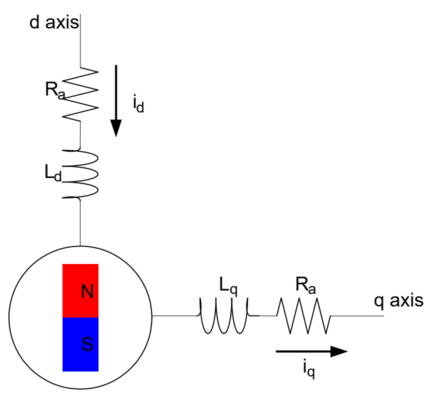


Figure 3.2 Conceptual Diagram of the 2-Phase DC Motor

The magnitude of the torque generated in the motor can be found as follows by the outer product of the current vector and the armature interlinkage magnetic flux. The first term on the right side of this equation is referred to as magnet torque, and the second term on the right side as 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

A motor which has no difference in inductance between the d axis and the q axis is referred to as a motor with no saliency. In this case, the torque increases in proportion to the q-axis current as the reluctance torque is 0. Therefore, the q-axis current is sometimes referred to as the torque current. On the other hand, the d-axis current, when its magnitude varies, seems as if the magnitude of the magnetic flux of the permanent magnet varies for the q-axis current. It is therefore referred to as an exciting current.

Generally, since the SPMSM does not have saliency, it controls the d-axis current (which is unnecessary for torque generation) to 0 in speed control. This is called id = 0 control. On the other hand, as the motion equation of the motor at this time is represented as follows, it can be seen that increasing q-axis current iq can boost the speed.

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

T_L: Load torque I: Inertia moment of the motor

The speed is controlled by PI control, not by solving this motion equation. The command value of the q-axis current can be obtained by speed PI control.

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

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

PI control is also performed for the current value to quickly stabilize it by the current command values for the d axis and the q axis. The command voltage value is obtained by the current PI control.

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

K_{Pi_d}: d-axis current PI proportional gain K_{Ii_d}: d-axis current PI integration gain

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

K_{Pi_q}: q-axis current PI proportional gain K_{Ii_q}: q-axis current PI integration gain

When the motor rotates, an inductive voltage is generated. The d-axis voltage is affected by the q-axis current and the q-axis voltage is affected by the permanent magnet magnetic flux more prominently as the speed increases. The interference between the d axis and the q axis may delay the stabilization of the current value. To avoid this, the voltage for each axis is calculated by feed forwarding the interference term in each axis so that it is canceled.

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

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

The method which eliminates the influence of the interference term like this is called non-interacting control. This enables independent control of the d axis and the q axis.

Vector control converts the three phases of the AC motor which could not be controlled independently into two phases which can be controlled independently to provide control while managing the torque, speed and position of the rotor.

The control flow of the vector control is shown below.

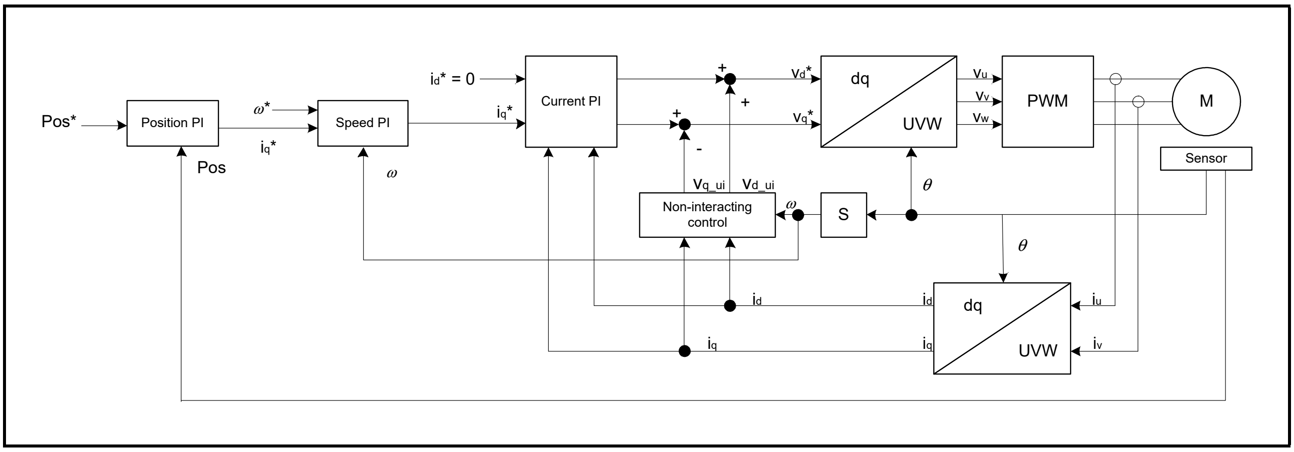


Figure 3.3 Control Flow of Vector Control (Position Control)

3.3 Starting the Motor

This system determines the magnetic pole position of the rotor by generating the current vector in the sequence shown in Figure 3.4 when starting the motor to match the direction of the d axis and the current vector. Figure 3.5 shows the motor start sequence.

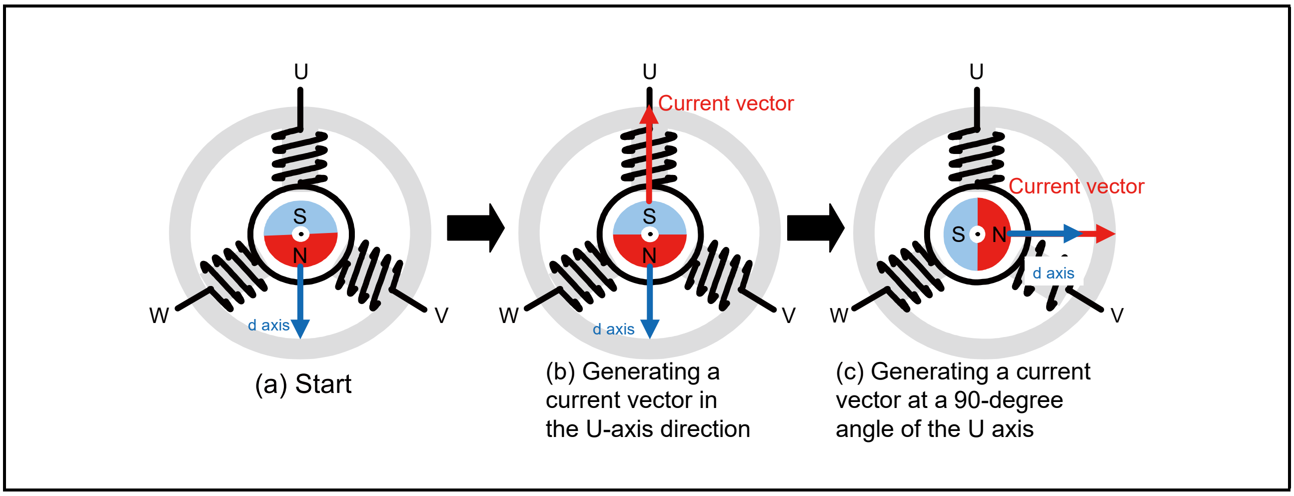


Figure 3.4 Determining the Position of the Permanent Magnet

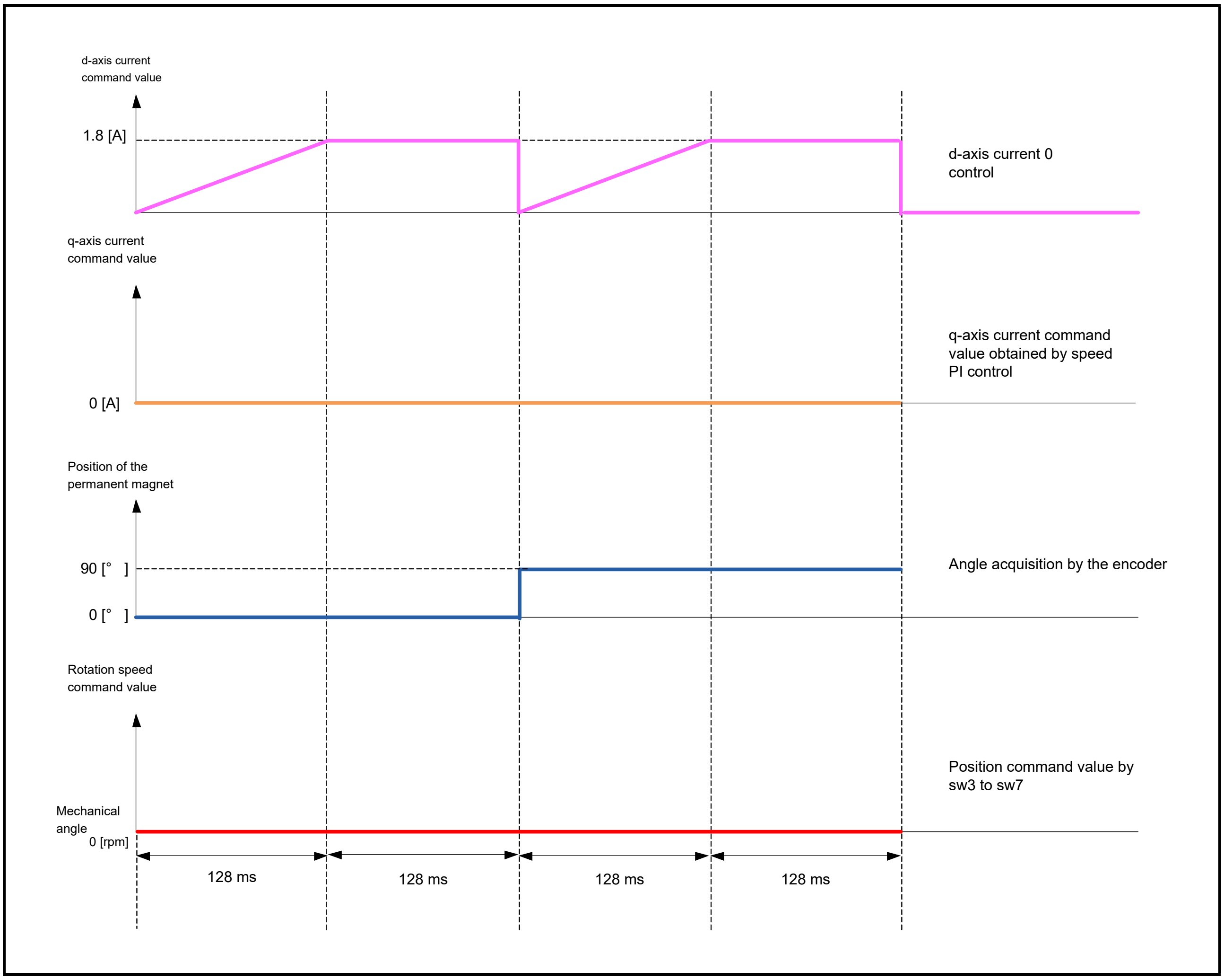


Figure 3.5 Motor Start Control

3.4 Calculating the Speed

In this system, the angular speed is calculated based on the encoder timer counter value as shown in Figure 3.6.

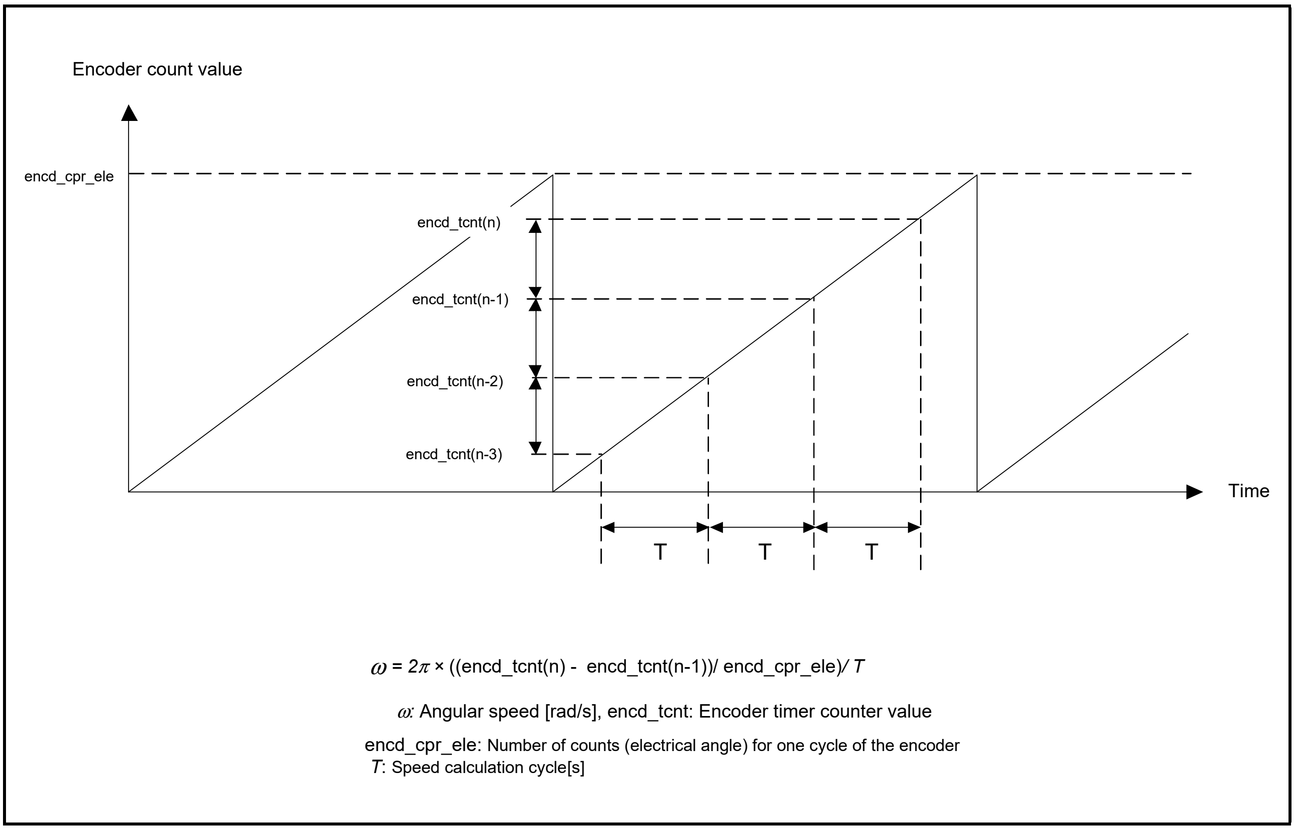


Figure 3.6 Calculating the Speed by Using the Encoder

3.5 Comparing the Triangular Wave

To output command value voltages, the triangular wave comparison method is used which determines the pulse width of the output voltage by comparing the carrier waveform (triangular wave) and the command value voltage waveform. This PWM method allows the sinusoidal command value voltage to be output artificially.

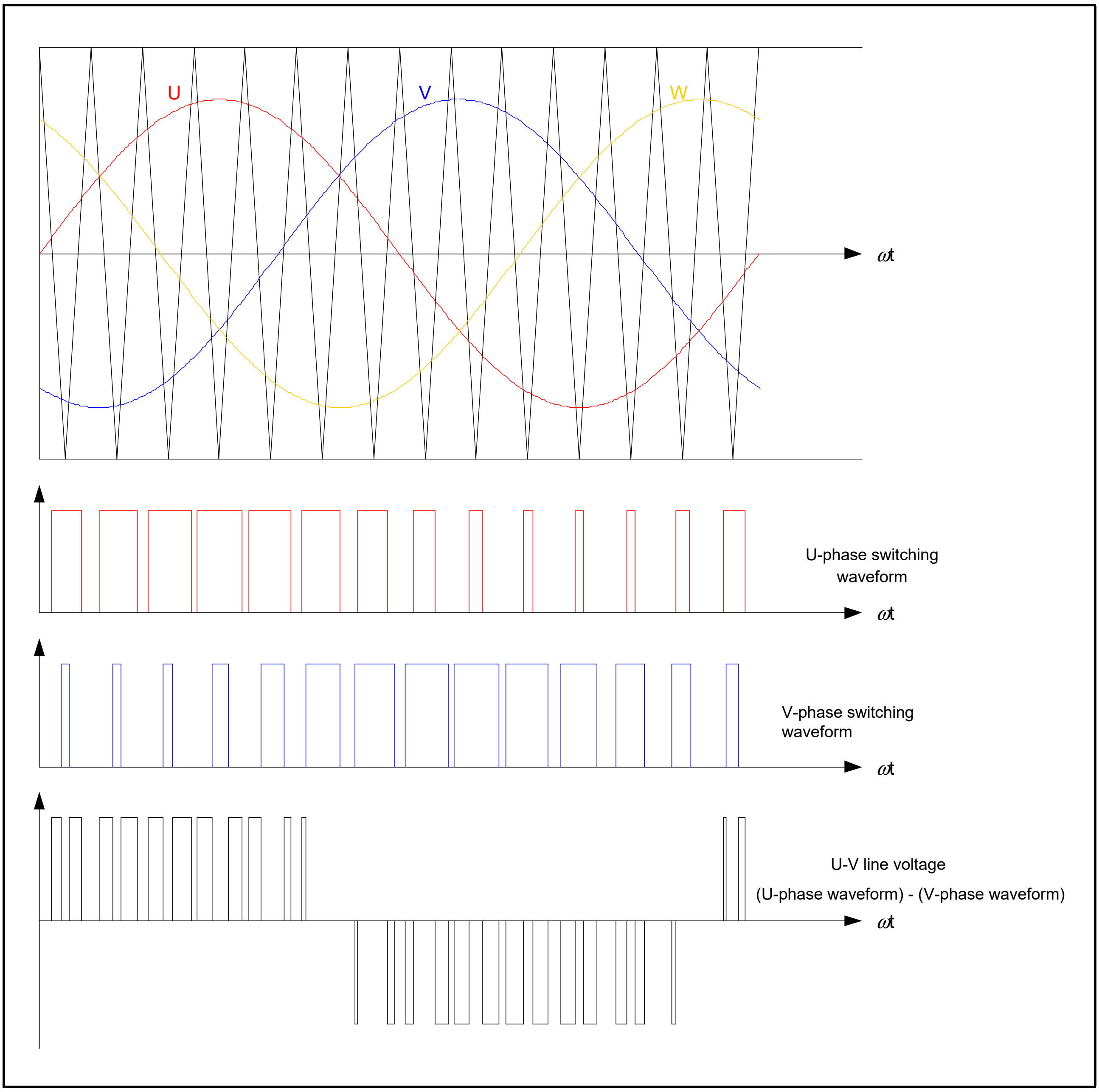


Figure 3.7 Conceptual Diagram of the Triangular Wave Comparison Method

Where, the ratio of the output voltage pulse to the carrier wave in Figure 3.8 is called "duty."

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

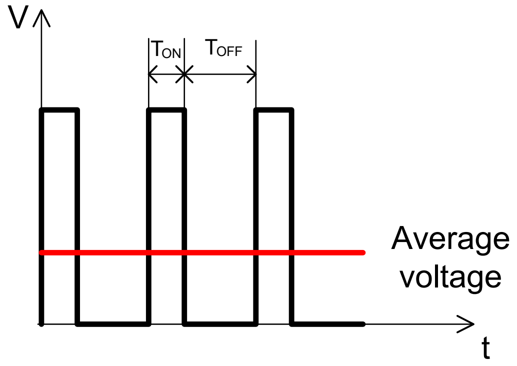


Figure 3.8 Definition of Duty

Modulation factor m is defined as follows:

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

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

Desired control is performed by reflecting this modulation factor in the register that determines the PWM duty.

4. Control Program

This section describes the control program.

4.1 Control

4.1.1 Starting and Stopping the Motor

The start and stop of the motor is controlled by the SW1 input.

A general port (PP1) is allocated to SW1. The program reads the PP1 pin in the main loop. If it is low (SW1 = SW ON), the motor is started. If it is high (SW1 = SW OFF), the motor is stopped. However, if the motor axis position command value is not updated once after the power is turned on, the motor does not start even when SW1 = SW ON is input. In this case, the level of SW0 must be first switched as described in Section 4.1.2 after setting SW1 to SW ON, and then the motor axis position command value must be updated.

4.1.2 Motor Axis Position Target Value, Inverter Bus Voltage, and Motor 3-Phase Voltage

(1) Motor axis position target value

Use the DIP switches to set the motor axis position target values as shown in Table 4.1. The motor axis position target value is updated by reading the general port (PP0) of SW0 and acquiring the SW3-SW7 settings if the level is different from the previous one. The calculation formula of the position target value is as follows:

Position target value = Maximum position of the axis rotation in CW direction (54000) × (n/15)

Table 4.1 Combination of Position Target Values

Item		DIP Switch Setting				
Position Target Value [Pulse]	n	Sign Bit (SW3)	Bit 4 (SW4)	Bit 3 (SW5)	Bit 2 (SW6)	Bit 1 (SW7)
54000	15	SW OFF	SW ON	SW ON	SW ON	SW ON
50400	14	SW OFF	SW ON	SW ON	SW ON	SW OFF
46800	13	SW OFF	SW ON	SW ON	SW OFF	SW ON
43200	12	SW OFF	SW ON	SW ON	SW OFF	SW OFF
39600	11	SW OFF	SW ON	SW OFF	SW ON	SW ON
36000	10	SW OFF	SW ON	SW OFF	SW ON	SW OFF
32400	9	SW OFF	SW ON	SW OFF	SW OFF	SW ON
28800	8	SW OFF	SW ON	SW OFF	SW OFF	SW OFF
25200	7	SW OFF	SW OFF	SW ON	SW ON	SW ON
21600	6	SW OFF	SW OFF	SW ON	SW ON	SW OFF
18000	5	SW OFF	SW OFF	SW ON	SW OFF	SW ON
14400	4	SW OFF	SW OFF	SW ON	SW OFF	SW OFF
10800	3	SW OFF	SW OFF	SW OFF	SW ON	SW ON
7200	2	SW OFF	SW OFF	SW OFF	SW ON	SW OFF
3600	1	SW OFF	SW OFF	SW OFF	SW OFF	SW ON
0	0	SW OFF or SW ON	SW OFF	SW OFF	SW OFF	SW OFF
-3600	-1	SW ON	SW OFF	SW OFF	SW OFF	SW ON
-7200	-2	SW ON	SW OFF	SW OFF	SW ON	SW OFF
-10800	-3	SW ON	SW OFF	SW OFF	SW ON	SW ON
-14400	-4	SW ON	SW OFF	SW ON	SW OFF	SW OFF
-18000	-5	SW ON	SW OFF	SW ON	SW OFF	SW ON
-21600	-6	SW ON	SW OFF	SW ON	SW ON	SW OFF
-25200	-7	SW ON	SW OFF	SW ON	SW ON	SW ON
-28800	-8	SW ON	SW ON	SW OFF	SW OFF	SW OFF
-32400	-9	SW ON	SW ON	SW OFF	SW OFF	SW ON
-36000	-10	SW ON	SW ON	SW OFF	SW ON	SW OFF
-39600	-11	SW ON	SW ON	SW OFF	SW ON	SW ON
-43200	-12	SW ON	SW ON	SW ON	SW OFF	SW OFF
-46800	-13	SW ON	SW ON	SW ON	SW OFF	SW ON
-50400	-14	SW ON	SW ON	SW ON	SW ON	SW OFF
-54000	-15	SW ON	SW ON	SW ON	SW ON	SW ON

(2) Inverter bus voltage

The inverter bus voltage is measured as shown in Table 4.2.

It is used for calculating the modulation factor and for detecting an overvoltage (PWM is stopped if an error was detected).

Table 4.2 Conversion Ratio of Inverter Bus Voltage

Item	Conversion Ratio (inverter bus voltage: A/D conversion value)	Channel
Inverter bus voltage	0 [V] to 280.0 [V]: 0000H to 0FFFH	AN007

(3) U- and V-phase currents

The U- and V-phase currents are measured for vector control as shown in Table 4.3.

Table 4.3 Conversion Ratio of U- and V-Phase Currents

Item	Conversion Ratio (U- and V-phase currents: A/D conversion value)	Channel
U- and V-phase currents	-37.5 [A] to 37.5 [A]: 0000H to 0FFFH	AN003, AN002

4.1.3 Control Method

When starting the motor, the program determines the magnetic pole position of the rotor and drives the motor based on the vector control by using an encoder after a predetermined period of time (see Section 3.3) (see the block diagram in Figure 3.3). PI control is used for speed and position control.

4.1.4 System Protection Function

This control program provides the following four error states and implements an emergency stop function in each error state.

- Overcurrent error

  Provides a high-impedance output to the PWM output pin in response to an emergency stop signal (overcurent detection) from the hardware (emergency stop with no CPU intervention).

  Also provides an emergency stop by the CPU when detecting an overcurrent (more than 4 A) by monitoring the U-, V-, and W-phase currents at 100-µs intervals.

- Overvoltage error

  Provides an emergency stop by the CPU when detecting an overvoltage (more than 28 V) by monitoring the inverter bus voltage at 100-µs intervals. This overvoltage limit value (28 V) is set considering the resistance error and the supply voltage error caused by the AC adapter.

- Low voltage error

  Provides an emergency stop by the CPU when detecting a lower voltage (less than 12 V) by monitoring the inverter bus voltage at 100-µs intervals.

- High-speed error

  Provides an emergency stop by the CPU when detecting a high speed exceeding 600 rad/s by monitoring the speed at 100-µs intervals.

4.2 Function Specifications

This control program uses multiple control functions. A list of control functions is shown below.

For detailed processing, see the flowchart or source file.

Table 4.4 Application Control Functions (1 / 2)

File Name	Description	Processing Overview
main.c	main Input: None Output: None	• Calls clock, port and interrupt setting functions. • Calls the hardware initialization function. • Calls the main processing variable initialization function. • Calls the state transition and event execution function. • Calls the watchdog timer and timer execution function. • Main processing ⇒ Calls the main processing execution function. ⇒ Calls the watchdog timer clearing function.
	ctrl_ui Input: None Output: None	• Changes the motor status. • Determines the position target value.
	software_init Input: None Output: None	Initializes the variables for use in main processing.
	sys_init Input: None Output: None	Sets the clock.
	set_ports Input: None Output: None	Sets the I/O ports.
	set_irqs Input: None Output: None	Sets interrupts.
	set_wdt Input: None Output: None	Sets the WDT.
	clear_wdt Input: None Output: None	Clears the WDT.
	set_timer Input: None Output: None	Sets the 1-ms timer.
	start_timer Input: None Output: None	Starts the 1-ms timer.
	stop_timer Input: None Output: None	Stops the 1-ms timer.
	get_sw_pos Input: None Output: Position setting	Obtains the position setting.
	get_sw1 Input: None Output: SW1 level	Obtains the SW1 state.
	get_sw2 Input: None Output: SW2 level	Obtains the SW2 state.
	led9_on Input: None Output: None	Lights LED9.

Table 4.4 Application Control Functions (2 / 2)

File Name	Description	Processing Overview
main.c	led8_on Input: None Output: None	Lights LED8.
	led9_off Input: None Output: None	Extinguishes LED9.
	led8_off Input: None Output: None	Extinguishes LED8.
app_common.c	APP_VIC_Init Input: None Output: None	Initializes the VIC.
	APP_CPU_EnableInterrupt Input: None Output: None	Enables acceptance of CPU interrupts.
	APP_CPU_DisableInterrupt Input: None Output: None	Disables acceptance of CPU interrupts.
	APP_VIC_EnableIRQ Input: Interrupt vector number Output: None	Enables VIC interrupts.
	APP_VIC_DisableIRQ Input: Interrupt vector number Output: None	Disables VIC interrupts.
	APP_VIC_ClearEdgeDetectIRQ Input: Interrupt vector number Output: None	Clears the VIC interrupt edge detection.
	APP_VIC_EndSeqIRQ Input: None Output: None	Ends the interrupt sequence.
app_interrupt.c	app_cmt_cmi0_isr Input: None Output: None	CMT: CMI0 interrupt service routine • Calls the R_MTR_Timer_Interrupt function.
	app_cmt_cmi1_isr Input: None Output: None	CMT: CMI1 interrupt service routine • Calls the R_CMT_CMI1_Interrupt function.
	app_mtu_tciv4_isr Input: None Output: None	MTU3: TCIV4 interrupt service routine • Calls the R_MTR_Pwm_Interrupt function.
	app_poe3_oei2_isr Input: None Output: None	POE3: OEI2 interrupt service routine • Calls the R_MTR_OverCurrent_Interrupt function.

Table 4.5 Drive Library Control Functions

File Name	Description	Processing Overview
r_mtr_api.c	R_MTR_InitHardware Input: None Output: None	Initializes the peripheral functions.
	R_MTR_InitSequence Input: None Output: None	Initializes the sequence processing.
	R_MTR_ExecEvent Input: Generated event Output: None	• Changes the status. • Calls a processing function appropriate for the generated event. - Starts/stops the motor. - Initializes the motor control processing.
	R_MTR_GetStatus Input: None Output: Status information	Obtains the status.
	R_MTR_GetErrorStatus Input: None Output: Error status information	Obtains the error status when an error occurred.
	R_MTR_SetMaxSpeed Input: Speed target value [rad/sec] Output: None	Sets the maximum speed control value. Input value: 25, 50, 100 (100 if any other value is set)
	R_MTR_GetSpeed Input: None Output: Speed value [rad/sec]	Obtains the calculated speed value (electrical angle).
	R_MTR_SetPosition Input: Position target value [pulse] Output: None	Sets the position control target. Input value: –54000 to +54000 (Minimum or maximum value if a value outside the range is specified)
	R_MTR_GetPosition Input: None Output: Current position value [pulse]	Obtains the current position value.
	R_MTR_SetMaxTorque Input: Torque control command value [A] Output: None	Sets the maximum torque. Input value: 0.5 to 3.0 (Minimum or maximum value if a value outside the range is specified)
	R_MTR_GetTorque Input: None Output: Current torque value [A]	Obtains the current torque value.
	R_MTR_GetCurrentAdjustStatus Input: None Output: Current offset detection status information	Obtains the current offset detection status information.
	R_MTR_ClearCurrentAdjustStatus Input: None Output: None	Clears the current offset detection status.
	R_MTR_OverCurrent_Interrupt Input: None Output: None	Detects an overcurrent status. (POE3: calls in the OEI2 interrupt service routine.) • Calls the event processing selection function. • Changes the motor status. • Calls the high-impedance status clearing function.
	R_MTR_Pwm_Interrupt Input: None Output: None	Calls at 100-μs intervals. (MTU3: Calls in the TCIV4 interrupt service routine.) • Vector control • Current PI control
	R_MTR_Timer_Interrupt Input: None Output: None	Calls at 1-ms intervals. (Calls in the timer interrupt service routine.) • Start control • Speed PI control

Table 4.6 Device Driver Control Functions (1 / 2)

File Name	Description	Processing Overview
r_adc_if.c	R_ADC_SetOpr Input: Unit, operation setting structure address, interrupt callback function address Output: Normal end, input parameter error	Sets the S12ADCa operation.
	R_ADC_GetDataAnalogInput Input: Unit, analog input channel Output: Input data value, input parameter error	Obtains S12ADCa analog input data.
	R_ADC_GetAdcStatus Input: Unit Output: None	Obtains the S12ADCa status.
	R_ADC_Start Input: Unit, trigger setting Output: Normal end, input parameter error	Starts S12ADCa conversion.
	R_ADC_Stop Input: Unit Output: Normal end, input parameter error	Stops S12ADCa conversion.
r_cmt_if.c	R_CMT_SetOpr Input: Channel, clock frequency division, interrupt enable/disable Output: Normal end, input parameter error	Sets the CMT operation.
	R_CMT_SetTimeCounter Input: Channel, counter setting value Output: Normal end, input parameter error	Sets the CMT time counter.
	R_CMT_SetCompareMatchCount Input: Channel, counter setting value Output: Normal end, input parameter error	Sets the CMT compare match count.
	R_CMT_Start Input: Channel Output: Normal end, input parameter error	Starts the CMT time counter.
	R_CMT_Stop Input: Channel Output: Normal end, input parameter error	Stops the CMT time counter.
r_io_port_if.c	R_IO_PORT_Set Input: Port, mode, input pull-up setting, function selection setting Output: Normal end, input parameter error	Sets the I/O ports.
	R_IO_PORT_GetInputData Input: Port Output: Input data value, input parameter error	Obtains the I/O port input data.
	R_IO_PORT_SetOutputData Input: Port, output setting value Output: Normal end, input parameter error	Sets the I/O port output data.
r_mtu_if.c	R_MTU_SetOpr Input: Channel, operation setting structure address, interrupt callback function address Output: Normal end, input parameter error	Sets the MTU3a operation.
	R_MTU_SetTimerCounter Input: Channel, counter setting value Output: Normal end, input parameter error	Sets the MTU3a timer counter.
	R_MTU_SetTimerGeneral Input: Channel, general counter, counter setting value Output: Normal end, input parameter error	Sets the MTU3a timer general counter.

Table 4.6 Device Driver Control Functions (2 / 2)

File Name	Description	Processing Overview
r_mtu_if.c	R_MTU_ClearTimerStatus Input: Channel, status value Output: Normal end, input parameter error	Clears the MTU3a status.
	R_MTU_SetTimerOutput Input: MTIO pin, timer output setting structure address Output: Normal end, input parameter error	Sets the MTU3a timer output.
	R_MTU_SetTimerGate Input: Timer gate enabled/disabled, timer gate setting structure address Output: Normal end, input parameter error	Sets the MTU3a timer gate.
	R_MTU_SetTimerCycle Input: Channel, data setting value Output: Normal end, input parameter error	Sets the MTU3a timer cycle.
	R_MTU_SetTimerDeadTime Input: Channel, dead time enabled/disabled counter setting value Output: Normal end, input parameter error	Sets the MTU3a dead time.
	R_MTU_SetInterruptSkip Input: Channel, timer interrupt skipping setting structure address Output: Normal end, input parameter error	Sets the MTU3a timer interrupt skipping.
	R_MTU_SetNoiseFilterMtclk Input: Noise filter setting structure address Output: Normal end, input parameter error	Sets the MTU3a noise filter (for MTCLK).
	R_MTU_StartTimerCounterA Input: Timer counter Output: Normal end	Starts the MTU3a timer counter.
	R_MTU_StopTimerCounterA Input: Timer counter Output: Normal end	Stops the MTU3a timer counter.
	R_MTU_StartTimerOutput Input: MTIO pin Output: Normal end, input parameter error	Starts the MTU3a timer output.
	R_MTU_StopTimerOutput Input: MTIO pin Output: Normal end, input parameter error	Stops the MTU3a timer output.

4.3 Functions

The table below lists the variables used by this program.

Table 4.7 Functions (1 / 3)

File	Variable Name	Type	Description	Remarks
main.c (Application)	g_u1_motor_status	uint8	User motor status management	0: Stopped 1: Rotating 2: Error
	g_u1_reset_req	uint8	Reset request flag	0: SW2 ON in an error 1: SW2 OFF in an error
	g_u1_sw1_cnt	uint8	SW1 judgment counter	Chattering removal
	g_u1_sw2_cnt	uint8	SW2 judgment counter	Chattering removal
	g_s4_setposition	uint32	Position target value	[Pulse]
	g_u1_sw0_port_old	uint8	Previous SW0 value	0: ON 1: OFF
	g_u1_pos_first_update	uint8	Position target initial update flag	1: Updated 0: Not updated
	g_u2_sw_pos_value	uint16	Position setting	0x0000 to 0x0FFF
r_mtr_api.c (drive library)	g_u1_def_state	uint8	Motor status definition	Array members • Stop mode • Run mode • Error mode
	gp_u1_def_action	uint8	Action definitions	Array members • Stop action • Run action • Error action • Reset action • No action
mtr_seq.c (drive library)	g_u1_seq_mode_system	uint8	State management	0: Stop mode 1: Run mode 2: Error mode
	g_u2_seq_run_mode	uint16	Operating mode management	2: Start mode 5: Normal operating mode
	g_u1_seq_error_status	uint8	Error status management	0: No error 1: Overcurrent error 2: Overvoltage error 3: High-speed error 7: Low voltage error 0xFF: Undefined error
mtr_ctrl.c (drive library)	g_f4_vdc_ad	float	Inverter bus voltage A/D value	[V]
	g_f4_vd_ref	float	d-axis voltage command value	Current PI control output value [V]
	g_f4_vq_ref	float	q-axis voltage command value	Current PI control output value [V]
	g_f4_iu_ad	float	U-phase current	[A]
	g_f4_pre_iu_ad	float	Previous U-phase current value	[A]
	g_f4_iv_ad	float	V-phase current	[A]
	g_f4_iw_ad	float	W-phase current	[A]
	g_f4_pre_iv_ad	float	Previous V-phase current value	[A]
	g_f4_offset_iu	float	U-phase current offset value	[A]
	g_f4_offset_iv	float	V-phase current offset value	[A]
	g_f4_id_lpf	float	d-axis current	[A]
	g_f4_iq_lpf	float	q-axis current	[A]

Table 4.7 Functions (2 / 3)

File	Variable Name	Type	Description	Remarks
mtr_ctrl.c (drive library)	g_f4_kp_id	float	d-axis current PI proportional term gain	
	g_f4_ki_id	float	d-axis current PI integral term gain	
	g_f4_kp_iq	float	q-axis current PI proportional term gain	
	g_f4_ki_iq	float	q-axis current PI integral term gain	
	g_f4_kp_speed	float	Speed PI control proportional term gain	
	g_f4_ki_speed	float	Speed PI control integral term gain	
	g_f4_lim_vd	float	d-axis current PI control output limit value	[V]
	g_f4_lim_vq	float	q-axis current PI control output limit value	[V]
	g_f4_ilim_vd	float	d-axis current PI control integral term limit value	[V]
	g_f4_ilim_vq	float	q-axis current PI control integral term limit value	[V]
	g_f4_lim_iq	float	Speed PI control output limit value	[A]
	g_f4_ilim_iq	float	Speed PI control integral term limit value	[A]
	g_f4_id_ref	float	d-axis current command value	[A]
	g_f4_iq_ref	float	q-axis current command value	[A]
	g_f4_speed_rad	float	Calculated speed value	Electrical angle [rad/s]
	g_f4_ref_speed_rad	float	Speed command value	Electrical angle [rad/s]
	g_f4_ref_speed_rad_ad	float	Speed adjustment value	Electrical angle [rad/s]
	g_f4_angle_rad	float	Rotor position	Electrical angle [rad]
	g_f4_max_speed_rad	float	Maximum speed value	Electrical angle [rad/s]
	g_f4_refu	float	U-phase voltage command value	[V]
	g_f4_refv	float	V-phase voltage command value	[V]
	g_f4_refw	float	W-phase voltage command value	[V]
	g_f4_inv_limit	float	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	
	speed	MTR_PI_CTRL	Speed PI control structure	
	position	MTR_PI_CTRL	Position PI control structure	
	g_u1_flag_id_open	uint8	Start mode judgment flag	
	g_u2_cnt_adjust	uint16	Counter for calculating the current offset	
	g_f4_id_open	float	d-axis current command value in start mode	[A]
	g_u2_cnt_adj_theta	uint16	Positioning time counter	
	g_f4_d_angle_rad	float	Rotor position difference	[rad]
	g_f4_encd_tcnt	float	Encoder timer counter value	
	g_f4_pre_encd_tcnt	float	Previous encoder timer counter value	
	g_u1_direction	uint8	Motor operating direction flag	
	g_s2_angle_count	int16	Speed measurement counter	

Table 4.7 Functions (3 / 3)

File	Variable Name	Type	Description	Remarks
mtr_ctrl.c (drive library)	g_u2_cnt_angle_pi	uint16	Counter for creating the position PI loop	
	g_s4_position_count	int32	Encoder position counter	
	g_s4_ref_position	int32	Position control target value	
	g_f4_lock_kp_position	float	Position PI control proportional term gain on stop	
	g_f4_lock_ki_position	float	Position PI control integral term gain on stop	
	g_f4_lock_limit_position	float	Position PI control output limit value on stop	
	g_f4_lock_ilimit_position	float	Position PI control integral term limit value on stop	
	g_f4_lock_kp_speed	float	Speed PI control proportional term gain on stop	
	g_f4_lock_ki_speed	float	Speed PI control integral term gain on stop	
	g_f4_nomal_kp_position	float	Position PI control proportional term gain during rotation	
	g_f4_nomal_ki_position	float	Position PI control integral term gain during rotation	
	g_f4_nomal_limit_position	float	Position PI control output limit value during rotation	
	g_f4_nomal_ilimit_position	float	Position PI control integral term limit value during rotation	
	g_f4_nomal_kp_speed	float	Speed PI control proportional term gain during rotation	
	g_f4_nomal_ki_speed	float	Speed PI control integral term gain during rotation	
	g_u1_flag_lock		Servo lock parameter switch flag	
	g_s2_lock_diff		Servo lock judgment position	
	g_u1_flag_pospi_speedpi		Position PI, speed PI control flag	

4.4 Macro Definitions

The table below lists the macros used by this program.

Table 4.8 Application Macro Definitions

File Name	Macro Name	Defined Value	Remarks
main.c	SW_ON	0	Low active
	SW_OFF	1	
	CHATTERING_CNT	200	Chattering removal
	PORT_SW1	IO_PORT_P(1)	SW1 input port (PP1)
	PORT_SW2	IO_PORT_P(2)	SW2 input port (PP2)
	PORT_LED9	IO_PORT_4(6)	LED9 output port (P46)
	PORT_LED8	IO_PORT_4(5)	LED8 output port (P46)
	LED_ON	1	High active
	LED_OFF	0	
	FLAG_ON	1	Flag ON
	FLAG_OFF	0	Flag OFF

Table 4.9 Drive Library Macro Definitions (1 / 4)

File Name	Macro Name	Defined Value	Remarks
r_mtr_api.h	MTR_DEADTIME	0	Dead time [μs]
	MTR_PWM_TIMER_FREQ	150.0f	PWM timer count frequency [MHz]
	MTR_CARRIER_FREQ	20.0f	Carrier frequency [kHz]
	MTR_DEADTIME_SET	MTR_DEADTIME * MTR_PWM_TIMER_FREQ	Dead time setting value
	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_LOCK_EN	1	Servo lock function provided/not provided
	MTR_INT_DECIMATION	1	Interrupt skipping 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_M	0.040107f	Magnetic flux [Wb]
	MTR_R	3.35f	Resistance [Ω]
	MTR_LD	0.00632f	d-axis inductance [H]
	MTR_LQ	0.00632f	q-axis inductance [H]
	MTR_POLE_PAIRS	2	Number of pole pairs
	MTR_ENCD_CPR_MECH	2000.0f	Number of counts for one cycle of the encoder (mechanical angle)
	MTR_ENCD_CPR_ELE	MTR_ENCD_CPR_MECH / MTR_POLE_PAIRS	Number of counts for one cycle of the encoder (electrical angle)
	MTR_SPEED_LIMIT	600	Speed limit value (electrical angle) [rad/s]
	MTR_OVERCURRENT_LIMIT	4	Current limit value [A]
	MTR_OVERVOLTAGE_LIMIT	28	High voltage limit value [V]

Table 4.9 Drive Library Macro Definitions (2 / 4)

File Name	Macro Name	Defined Value	Remarks
r_mtr_api.h	MTR_UNDERVOLTAGE_LIMIT	12	Low voltage limit value [V]
	MTR_TWOPI	2*3.14159265f	2π
	MTR_SQRT_2_3	0.81649658f	√(2/3)
	MTR_HALFPI	3.14159265f/2	(1/2)π
	MTR_HALF_VDC	12	Power-supply voltage / 2 [V]
	MTR_ADC_SCALING	0x7FF	ADC offset adjustment constant
	MTR_CURRENT_SCALING	75.0f /4095.0f	Current A/D conversion value resolution
	MTR_VDC_SCALING	280.0f /4095.0f	Inverter bus voltage A/D conversion value resolution
	MTR_ID_PI_KP	4	d-axis current PI control proportional term gain
	MTR_ID_PI_KI	0.21f	d-axis current PI control integral term gain
	MTR_IQ_PI_KP	4	q-axis current PI control proportional term gain
	MTR_IQ_PI_KI	0.21f	q-axis current PI control integral term gain
	MTR_SPEED_PI_KP	0.025f	Speed PI control proportional term gain
	MTR_SPEED_PI_KI	0.0001f	Speed PI control integral term gain
	MTR_SPEED_LPF_K	0.1f	Speed LPF gain
	MTR_CURRENT_LPF_K	0.1f	Current LPF gain
	MTR_LIMIT_VD	11	d-axis current PI control output limit value [V]
	MTR_LIMIT_VQ	11	q-axis current PI control output limit value [V]
	MTR_I_LIMIT_VD	11	d-axis current PI control integral term limit value [V]
	MTR_I_LIMIT_VQ	11	q-axis current PI control integral term limit value [V]
	MTR_LIMIT_IQ	3	Speed PI control output limit value [A]
	MTR_I_LIMIT_IQ	3	Speed PI control integral term limit value [A]
	MTR_MAX_SPEED_RAD	100.0f	Maximum speed (electrical angle) [rad/s]
	MTR_MIN_SPEED_RAD	0.0f	Minimum speed (electrical angle) [rad/s]
	MTR_START_REF_SPEED_UP_STEP	(MTR_MAX_SPEED_RAD – MTR_MIN_SPEED_RAD) / 2048.0f	Acceleration limiting constant
	MTR_START_REF_SPEED_DOWN_STEP	(MTR_MAX_SPEED_RAD – MTR_MIN_SPEED_RAD) / 2048.0f	Acceleration limiting constant
	MTR_START_OL_ID	1.8f	d-axis current in start mode [A]
	MTR_START_OL_ID_UP_TIME	128	d-axis current additional 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 additional value [A]
	MTR_ANGLE_ADJUST_TIME	128	Positioning time [ms]
	MTR_CHECK_ENCD_PERIOD	MTR_CTRL_PERIOD	Encoder timer counter sampling cycle [s]

Table 4.9 Drive Library Macro Definitions (3 / 4)

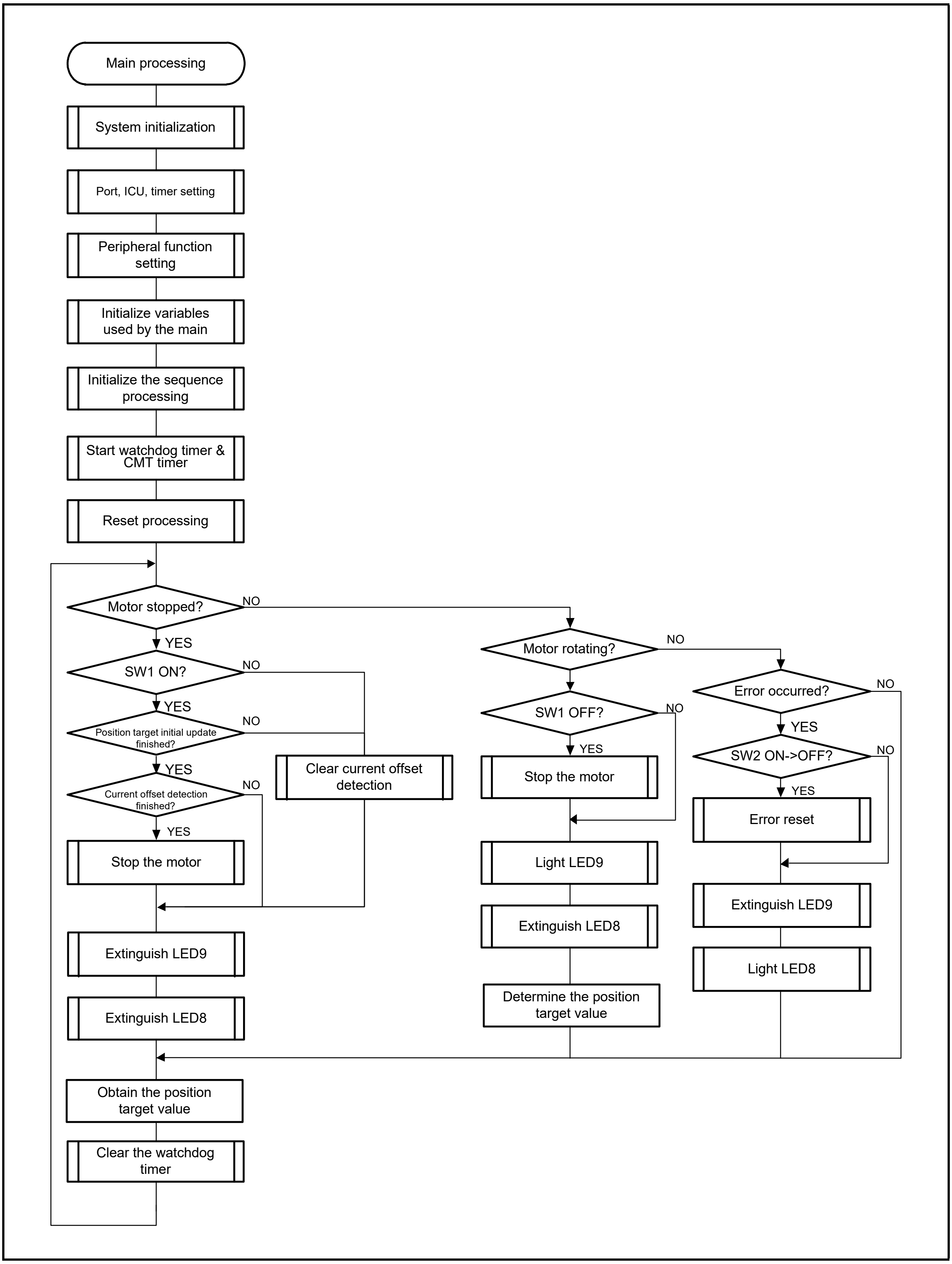
File Name	Macro Name	Defined Value	Remarks
r_mtr_api.h	MTR_POSITION_PI_KP_25	0.0195f	Position PI control proportional term gain during rotation (for 25 rad/sec)
	MTR_POSITION_PI_KP_50	0.0110f	Position PI control proportional term gain during rotation (for 50 rad/sec)
	MTR_POSITION_PI_KP_100	0.0058f	Position PI control proportional term gain during rotation (for 100 rad/sec)
	MTR_POSITION_PI_KI	0.000025f	Position PI control integral term gain during rotation
	MTR_POSITION_LIMIT_25	25.0f	Position PI control output limit value during rotation (for 25 rad/sec)
	MTR_POSITION_LIMIT_50	50.0f	Position PI control output limit value during rotation (for 50 rad/sec)
	MTR_POSITION_LIMIT_100	100.0f	Position PI control output limit value during rotation (for 100 rad/sec)
	MTR_POSITION_I_LIMIT	5.0f	Position PI control integral term limit value during rotation
	MTR_LOCK_DIFF	100	Encoder position difference definition
	MTR_LOCK_POSITION_PI_KP	0.10f	Position PI control proportional term gain on stop
	MTR_LOCK_POSITION_PI_KI	0.005f	Position PI control integral term gain on stop
	MTR_LOCK_POSITION_LIMIT	25.0f	Position PI control output limit value on stop
	MTR_LOCK_POSITION_I_LIMIT	2.0f	Position PI control integral term limit value on stop
	MTR_LOCK_SPEED_PI_KP	0.020f	Speed PI control proportional term gain on stop
	MTR_LOCK_SPEED_PI_KI	0.0025f	Speed PI control integral term gain on stop
	MTR_POS_PI_CTRL_CYC	5	Position PI control cycle [ms]
	MTR_CURRENT_ADJUST_TIME	100	Current stabilization wait time after reset [ms]
	MTR_ANGLE_MAX_COUNT	2000	Single-rotation pulse count [pulse]
	MTR_POSITION_CW_MAX	54000	Maximum position of the axis rotation in CW direction [pulse]
	MTR_POSITION_CCW_MAX	-54000	Maximum position of the axis rotation in CCW direction [pulse]
	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_MODE_STOP	0x00	Stopped state
	MTR_MODE_RUN	0x01	Rotating
	MTR_MODE_ERROR	0x02	Error state
	MTR_NONE_ERROR	0x00	No error state
	MTR_OVER_CURRENT_ERROR	0x01	Overcurrent error
	MTR_OVER_VOLTAGE_ERROR	0x02	Overvoltage error
	MTR_OVER_SPEED_ERROR	0x03	High-speed error
	MTR_UNDER_VOLTAGE_ERROR	0x07	Low voltage error
	MTR_UNKNOWN_ERROR	0xFF	Undefined error
	MTR_CURRENT_ADJUST_RUNNING	0x00	Detecting the current offset

Table 4.9 Drive Library Macro Definitions (4 / 4)

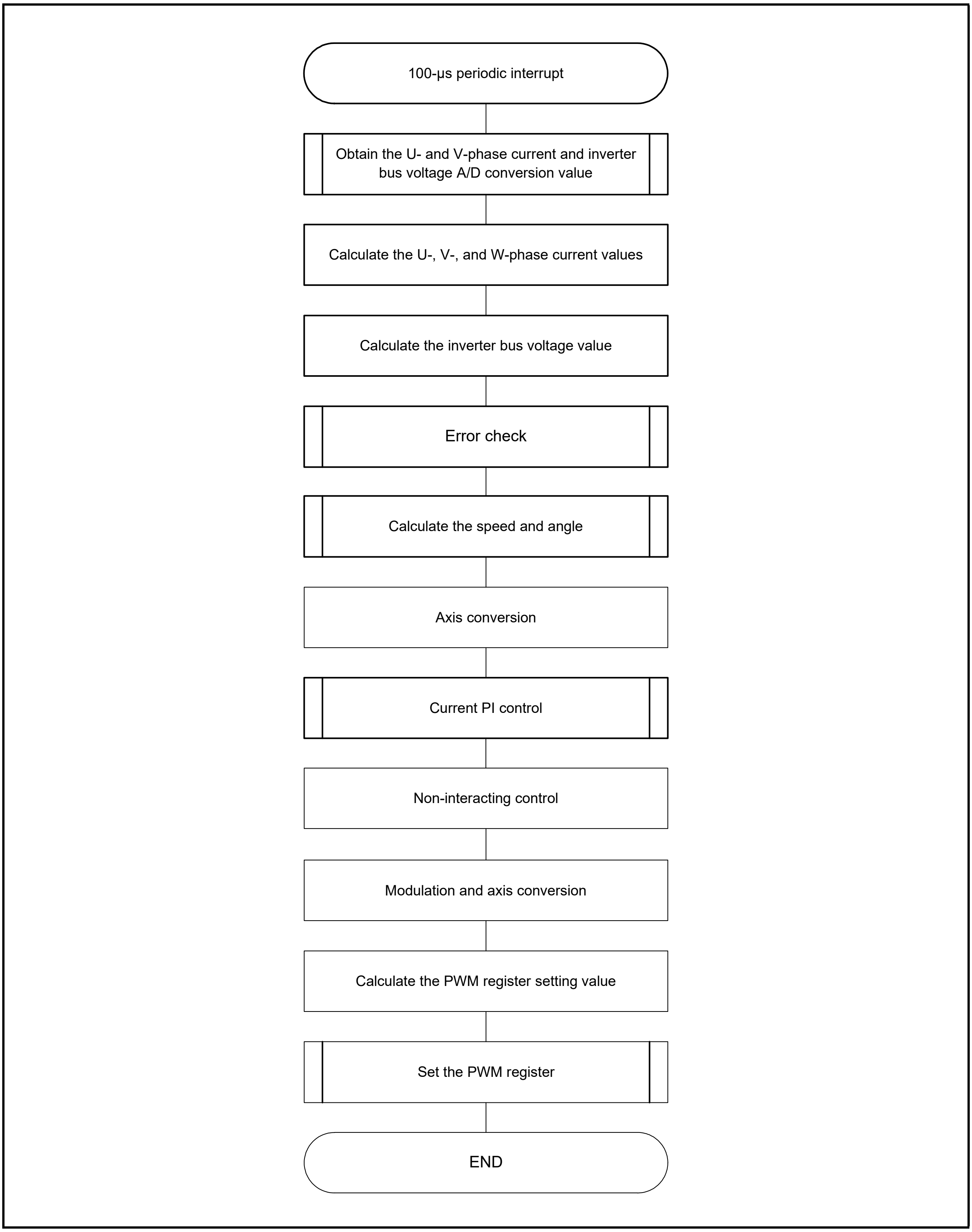
File Name	Macro Name	Defined Value	Remarks
r_mtr_api.h	MTR_CURRENT_ADJUST_COMPLETE	0x01	Current offset detection end

4.5 Control Flow (Flowchart)

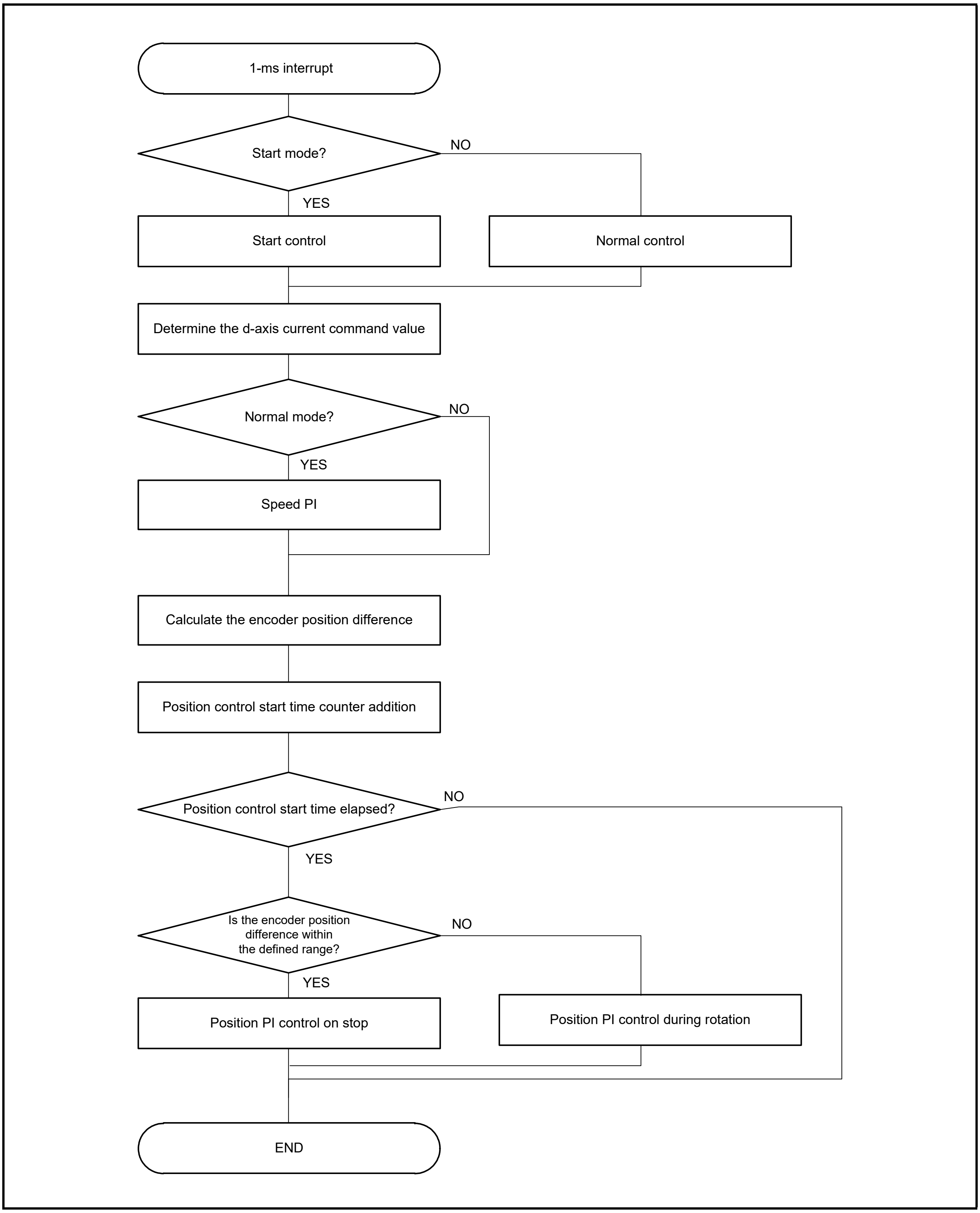
(1) Main processing



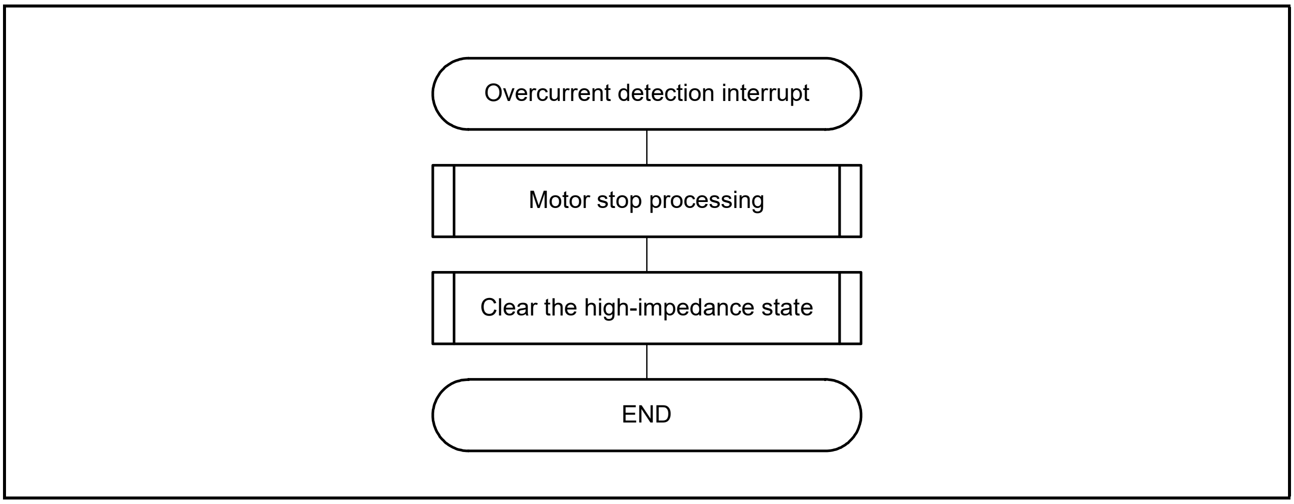
(2) 100-μs periodic interrupt processing



(3) 1-ms interrupt processing



(4) Overcurrent detection interrupt processing



5. Reference Documents

- RZ/T1 Group User's Manual: Hardware (R01UH0483EJ0130)
- RZ/T1 Group User's Manual: Solution Kit (R01UH0665EU0104)

Website and Support

Renesas Electronics website

http://www.renesas.com/

Inquiries

http://www.renesas.com/inquiry

Revision History	Application Note: Encoder-Based Vector Control (Position Control) of the Permanent Magnet Synchronous Motor for the RZ/T1 Motion Control Solution Kit

Rev.	Date	Description	
		Page	**Summary**
1.00	Jan. 12, 2018	—	First edition issued
1.10	Mar. 16, 2018	1. Overview	
		3	1.2 Development Environment: Information on GCC environment, added

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 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 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.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

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: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, 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 1207, 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 India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.0