RZ/T1 Group

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

R01AN4148EJ0100
Rev.1.00
Mar. 05, 2018

Introduction

This application note is designed to describe the sample program that drives a permanent magnet synchronous motor under encoder-based vector control (speed 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

This application note describes the sample program for encoder-based vector control (speed 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 (sample program) 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.

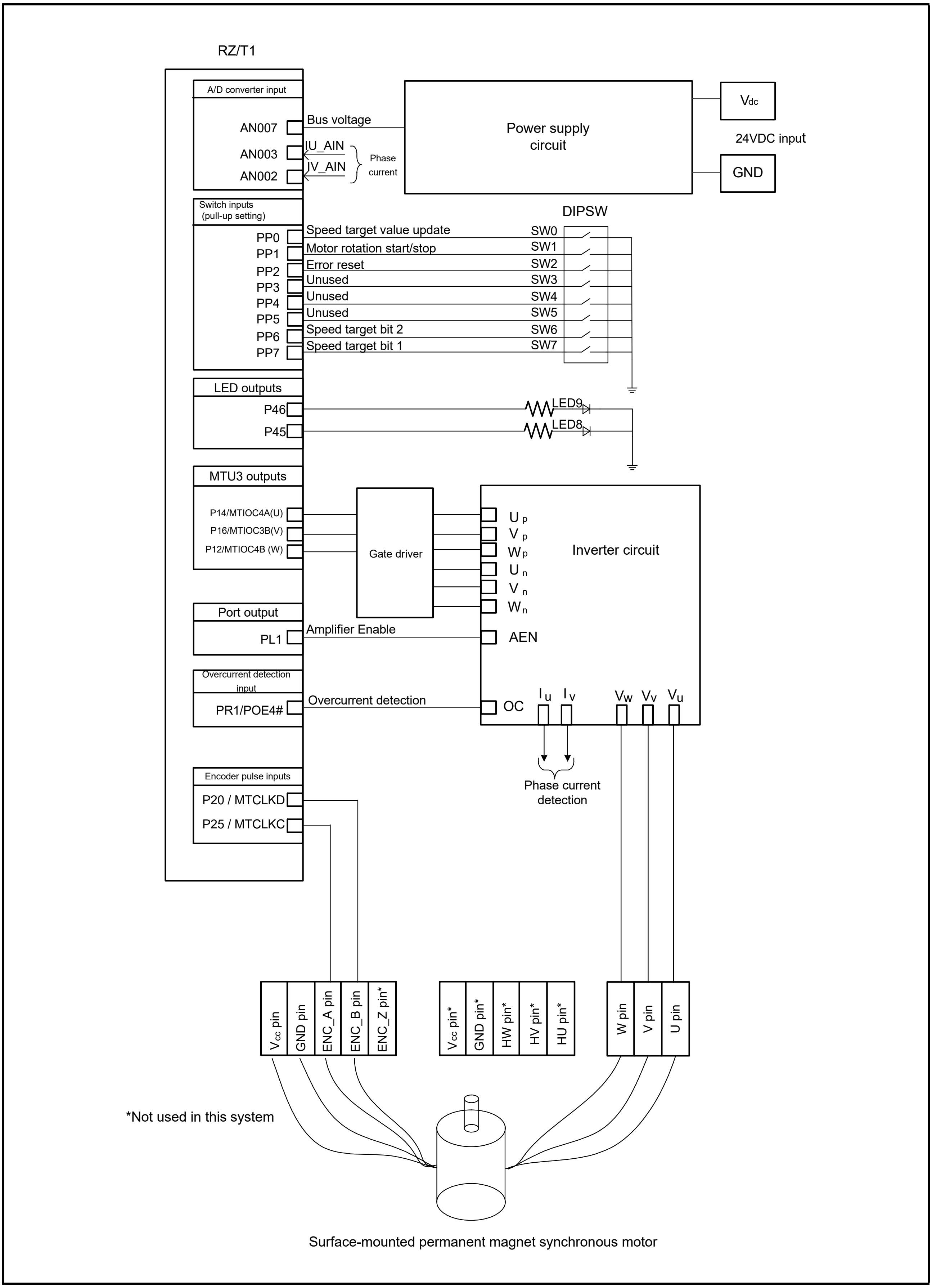


Figure 2.1 Hardware Configuration Diagram

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
Speed target value update	DIP switch (SW0)	Updates the rotation speed target value (see Section 4.1.1)
Rotation speed target	DIP switch (SW6, SW7)	Rotation speed 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
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

R7S910018CBG Pin Name	Function
AN007	Inverter bus voltage measurement
PP0	DIP switch for updating the speed target value
PP6	DIP switch for speed target bit 2
PP7	DIP switch for speed 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
RESET#	Reset

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_CSP	inc			iodefine.h	I/O register definition header in the RZ/T1 Group
				main.h	Main function, user interface control header
				mtr_common.h	Header for common definitions
				mtr_ctrl_biplane.h	Header of the processing section that depends on the board
				mtr_ctrl_rzt1.h	Header of the processing section that depends on the RZ/T1 Group
				mtr_ssns_encd_foc.h	Header of the section that depends on encoder-based vector control
	src	drv	mtr	mtr_ctrl_biplane.c	Processing section that depends on the board
				mtr_ctrl_rzt1.c	Processing section that depends on the RZ/T1 Group
				mtr_interrupt.c	Interrupt handler
				mtr_ssns_encd_foc.c	Section that depends on encoder-based vector control
	sample			main.c	Main function, user interface control

2.3.2 Module Configuration

Figure 2.2 shows the module configuration of the sample program.

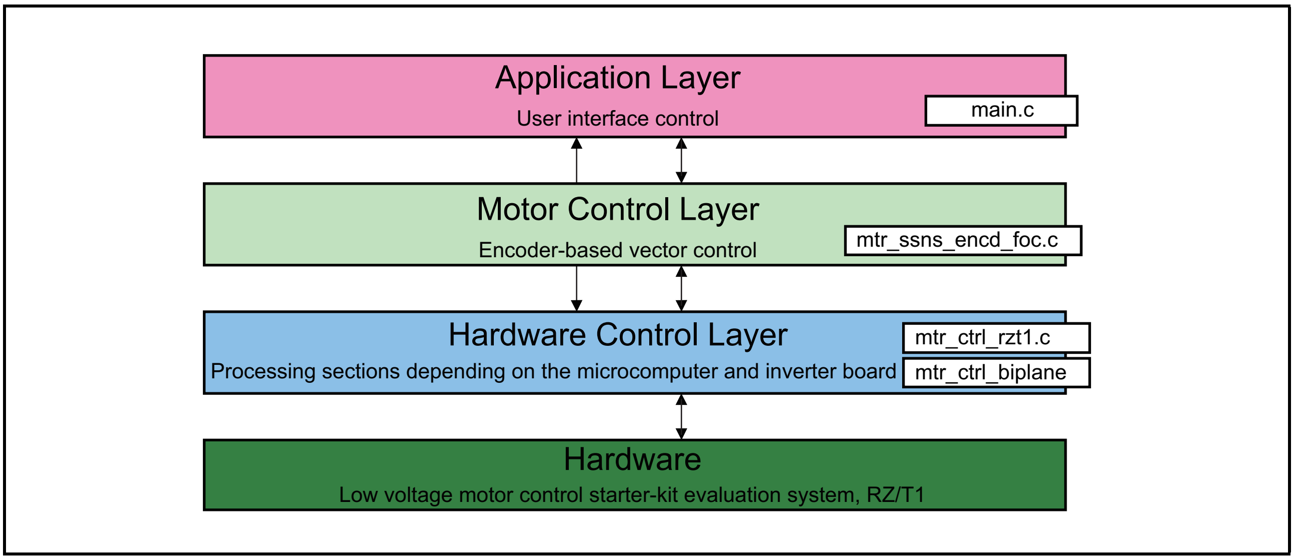


Figure 2.2 Module Configuration of the Sample Program

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 (speed control)
Motor rotation start/stop	Determined by the level of SW1 (PP1) (Low: rotation start, High: rotation stop)
Rotor magnetic pole position detection	Encoder
Carrier frequency (PWM)	20 kHz
Control period	100 µs (twice the carrier period)
Rotation speed controllable range	CW: 600 to 1,500 rpm
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.

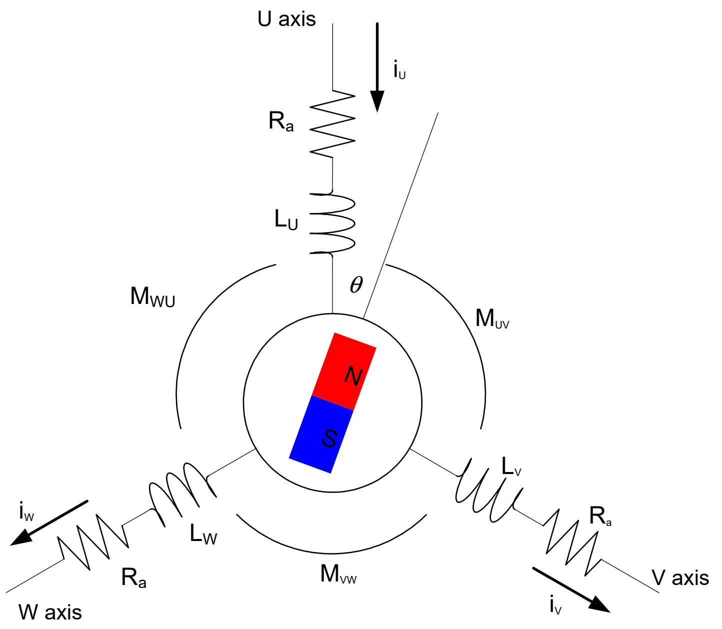


Figure 3.1 Conceptual Diagram of the 3-Phase Permanent Magnet Synchronous Motor

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

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

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).

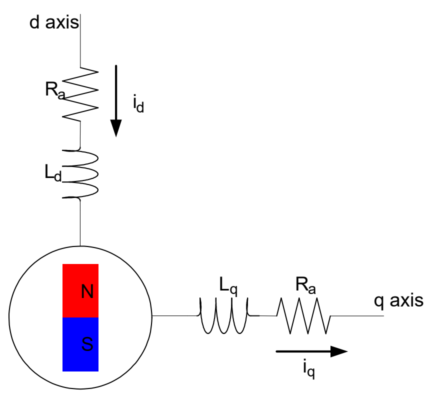


Figure 3.2 Conceptual Diagram of the 2-Phase DC Motor

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.

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

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 target value of the q-axis current can be obtained by speed PI control.

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

$K_{P\omega}$: 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 target 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.

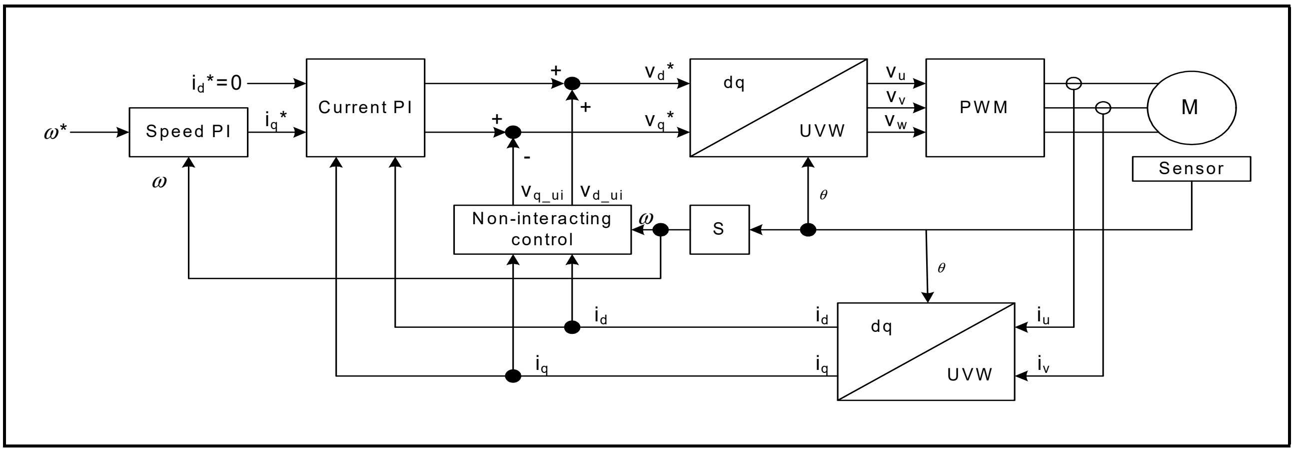


Figure 3.3 Control Flow of Vector Control

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.

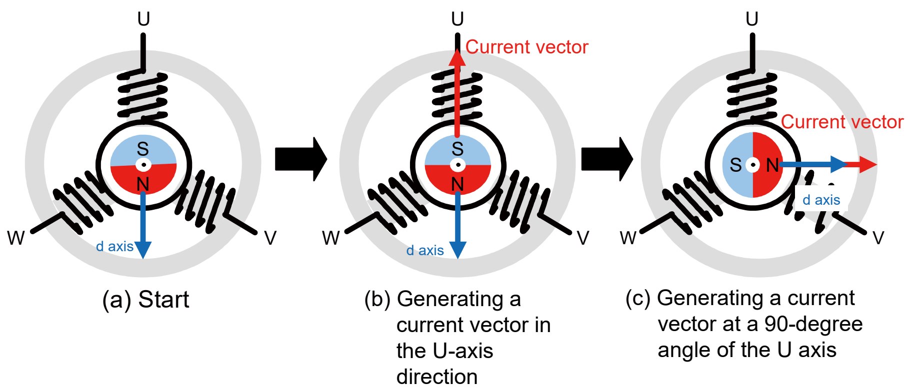


Figure 3.4 Determining the Position of the Permanent Magnet

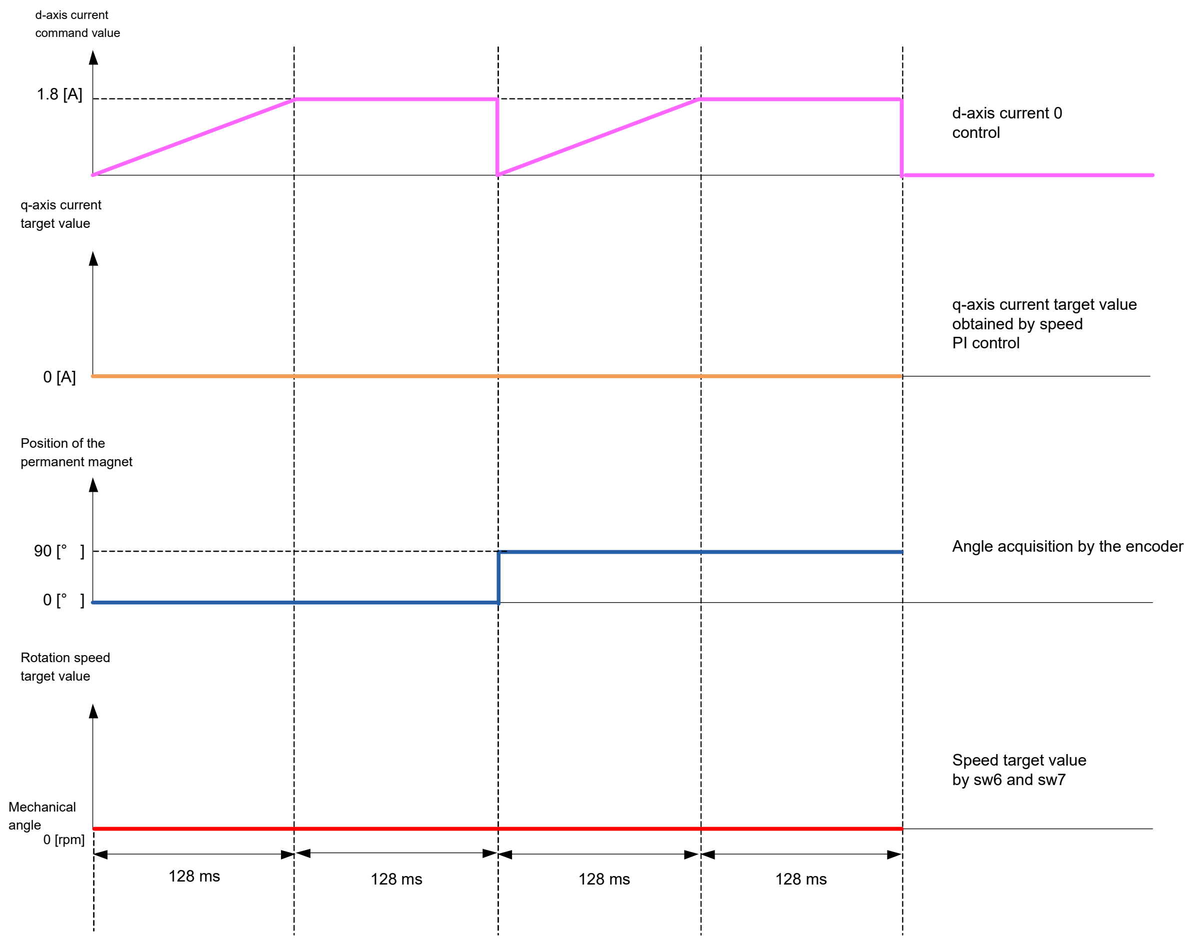
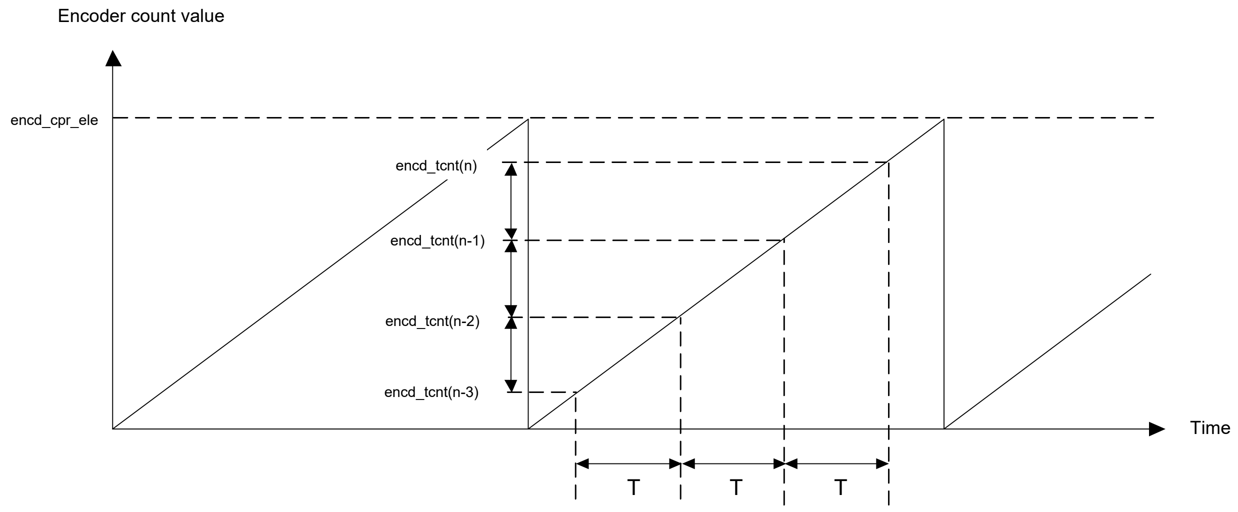


Figure 3.5 Motor Start Control

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.



$\omega = 2\pi \times ((encd\_tcnt(n) - encd\_tcnt(n-1)) / encd\_cpr\_ele) / T$

ω: Angular speed [rad/s]
encd_tcnt: Encoder timer counter value
encd_cpr_ele: Number of counts (electrical angle) for one cycle of the encoder
T: Speed calculation cycle[s]

Figure 3.6 Calculating the Speed by Using the Encoder

3.5 Comparing the Triangular Wave

To output target 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 target value voltage waveform. This PWM method allows the sinusoidal target value voltage to be output artificially.

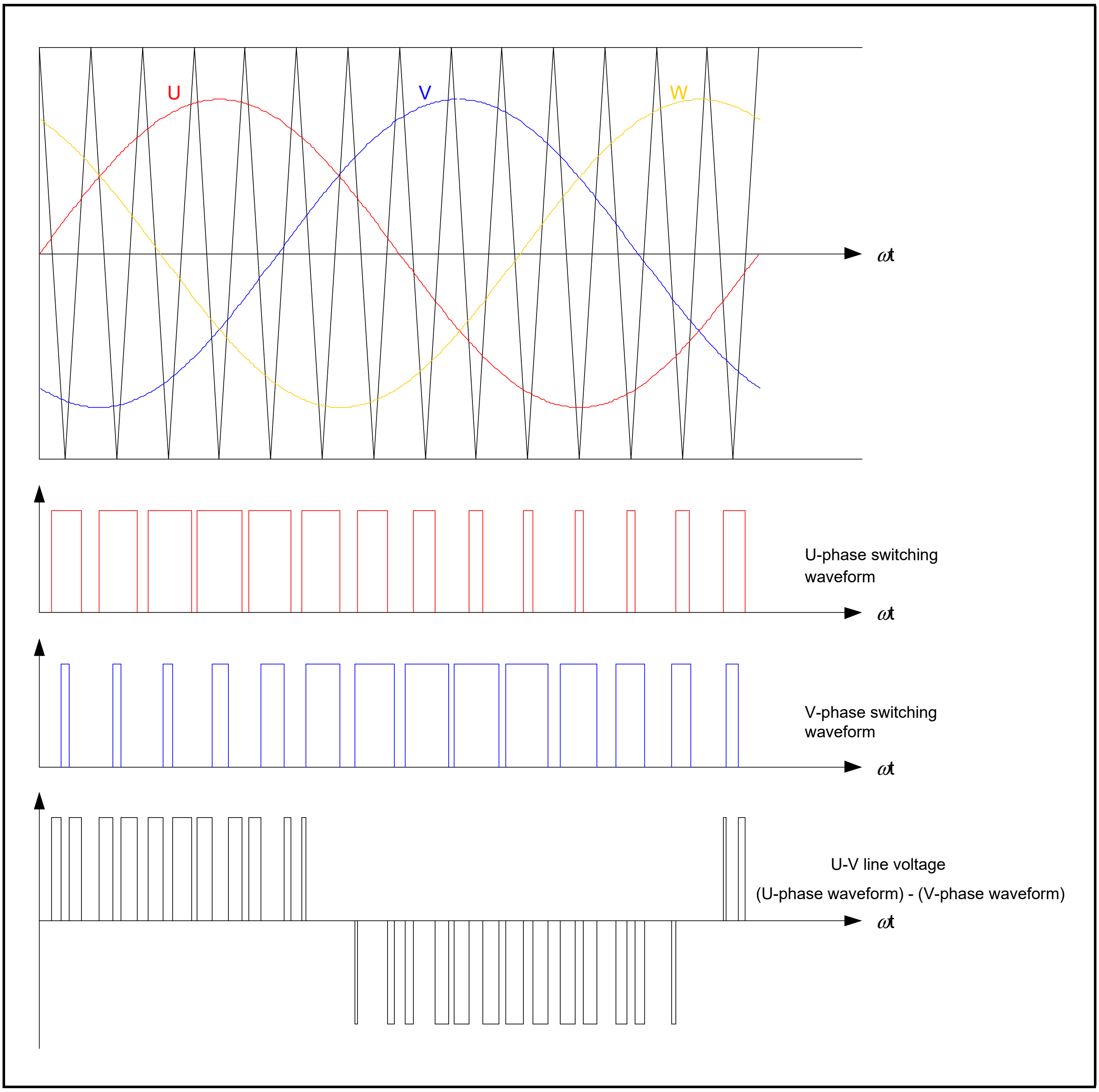


Figure 3.7 Conceptual Diagram of the Triangular Wave Comparison Method

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

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

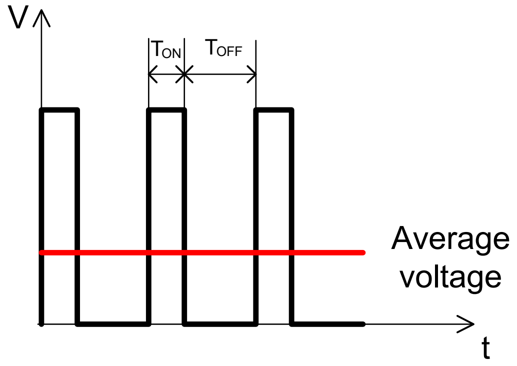


Figure 3.8 Definition of Duty

Modulation factor m is defined as follows:

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

m: Modulation factor V: Target 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 rotation speed target value is not updated at least once after the power is turned on, the motor does not start even when SW1 = SW ON is input. In this case, SW0 must be first switched as described in Section 4.1.2 after setting SW1 to SW ON, and then the motor axis rotation speed target value must be updated.

4.1.2 Motor Rotation Speed Target Value, Inverter Bus Voltage, and Motor 3-Phase Voltage

(1) Motor rotation speed target value

The motor rotation speed target value is set by using DIP switches as shown in Table 4.1. If the general port pin (PP0) of SW0 has been read in the main loop and if SW0 is switched, the level that was read in the previous loop might have been changed. Therefore, in such a case, the speed target value is updated.

Table 4.1 Combination of Speed Target Values

Item	DIP Switch Setting	
Position Target Value [Pulse]	**SW6**	**SW7**
1500	SW ON	SW ON
1200	SW ON	SW OFF
900	SW OFF	SW ON
600	SW OFF	SW OFF

(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 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 (overcurrent 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 List of Control Functions (1 / 4)

File Name	Description	Processing Overview
main.c	main Input: None Output: None	• Calls the hardware initialization function. • Calls the user interface initialization function. • Calls the main processing variable initialization function. • Calls the state transition and event 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 rotation speed target value.
	software_init Input: None Output: None	Initializes the variables for use in main processing.
mtr_ctrl_biplane.c	get_sw_speed Input: None Output: (uint16) speed_value / Speed setting	Obtains the speed setting.
	get_sw1 Input: None Output: (uint8) tmp_port / SW1 level	Obtains the SW1 state.
	get_sw2 Input: None Output: (uint8) tmp_port / SW2 level	Obtains the SW2 state.
	led9_on Input: None Output: None	Lights LED9.
	led8_on Input: None Output: None	Lights LED8.
	led9_off Input: None Output: None	Extinguishes LED9.
	led8_off Input: None Output: None	Extinguishes LED8.
mtr_ctrl_rzt1.c	R_MTR_InitHardware Input: None Output: None	Initializes the clock and peripheral functions.
	Init_ui Input: None Output: None	Initializes the UI.
	mtr_ctrl_start Input: None Output: None	Starts the motor.
	mtr_ctrl_stop Input: None Output: None	Stops the motor.
	mtr_get_sw_speed Input: None Output: (uint16) Speed setting	Obtains the speed setting.

Table 4.4 List of Control Functions (2 / 4)

File Name	Description	Processing Overview
mtr_ctrl_rzt1.c	mtr_get_speed_first_update Input: None Output: (uint8) g_u1_speed_first_update / Speed target initial update flag	Obtains the speed target initial update flag.
	mtr_get_iuivvdc Input: (float32) *f4_iu_ad / U-phase current A/D conversion value (float32) *f4_iv_ad / V-phase current A/D conversion value (float32) *f4_vdc_ad / Vdc A/D conversion value Output: None	Inverter bus voltage A/D conversion for U- and V-phase currents
	clear_wdt Input: None Output: None	Clears the WDT.
	mtr_clear_poe3_oei1_flag Input: None Output: None	Clears a high-impedance state.
	mtr_clear_poe3_oei2_flag Input: None Output: None	Clears a high-impedance state.
	mtr_clear_mtu4_flag Input: None Output: None	Clears an interrupt flag.
	mtr_clear_cmt0_flag Input: None Output: None	Clears an 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
	mtr_get_encd_tcnt Input: None Output: (float32) f4_temp / Encoder timer counter value	Obtains the encoder timer counter value.
	mtr_clear_encd_tcnt Input: None Output: None	Clears the encoder timer counter value.
	mtr_power_on Input: (uint32) axis / Channel Output: None	Turns on the amplifier.
	mtr_amp_enable Input: (uint32) axis / Channel Output: None	Enables the amplifier.
	mtr_amp_disable Input: (uint32) axis / Channel Output: None	Disables the amplifier.
	mtr_amp_fault Input: (uint32) axis / Channel Output: (uint32) result / Check result	Checks the amplifier.

Table 4.4 List of Control Functions (3 / 4)

File Name	Description	Processing Overview
mtr_interrupt.c	mtr_poe3_oei1_interrupt Input: None Output: None	Detects an overcurrent status. • Calls the event processing selection function. • Changes the motor status. • Calls the high-impedance status clearing function.
	mtr_poe3_oei2_interrupt Input: None Output: None	Detects an overcurrent status. • Calls the event processing selection function. • Changes the motor status. • Calls the high-impedance status clearing function.
	mtr_mtu4_interrupt Input: None Output: None	Calls at 100-μs intervals. • Vector control • Current PI control
	mtr_cmt0_interrupt Input: None Output: None	Calls at 1-ms intervals. • Start control • Speed PI control
mtr_ssns_encd_foc.c	R_MTR_InitSequence Input: None Output: None	Initializes the sequence processing.
	R_MTR_ExecEvent Input: (uint8)u1_event / Generated event Output: None	• Changes the status. • Calls a processing function appropriate for the generated event.
	mtr_act_run Input: (uint8)u1_state / Motor status Output: (uint8)u1_state / Motor status	• Calls the variable initialization function when starting the motor. • Calls the motor control start function.
	mtr_act_stop Input: (uint8)u1_state / Motor status Output: (uint8)u1_state / Motor status	Calls the motor control termination function.
	mtr_act_none Input: (uint8)u1_state / Motor status Output: (uint8)u1_state / Motor status	No processing
	mtr_act_reset Input: (uint8)u1_state / Motor status Output: (uint8)u1_state / Motor status	Initializes the global variables.
	mtr_act_error Input: (uint8)u1_state / Motor status Output: (uint8)u1_state / Motor status	Calls the motor control termination function.
	mtr_angle_speed Input: None Output: None	Calculates the position and speed.
	mtr_start_init Input: None Output: None	Initializes only the necessary variables when starting the motor.
	mtr_pi_ctrl Input: MTR_PI_CTRL *vdq/ PI control structure Output: (float32)f4_ref / PI control output value	PI control
	R_MTR_SetSpeed Input: (float32)ref_speed / Speed target value Output: None	Sets the speed target value.
	R_MTR_GetSpeed Input: None Output: (float32) g_f4_speed_rad / Speed	Obtains the calculated speed value (electrical angle).
	R_MTR_GetStatus Input: None Output: (uint8)g_u1_mode_system / Motor status	Obtains the motor status.
	mtr_error_check Input: None Output: None	Monitors and detects errors.

Table 4.4 List of Control Functions (4 / 4)

File Name	Description	Processing Overview
mtr_ssns_encd_foc.c	R_MTR_GetCurrentAdjustStatus Input: None Output: (uint8)u1_temp / 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.

4.3 List of Variables

The table below lists the variables used by this control program. Note that this list does not include local variables.

Table 4.5 List of Variables (1 / 3)

File	Variable Name	Type	Description	Remarks
main.c (Application)	g_f4_max_mecha_speed_rad	float32	Maximum speed target value	Mechanical angle [rad/s]
	g_f4_min_mecha_speed_rad	float32	Minimum speed target value	Mechanical angle [rad/s]
	g_f4_set_speed	float32	User speed target value	Electrical angle [rad/s]
	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 state 1: SW2 OFF in an error state
	g_u1_sw1_cnt	uint8	SW1 judgment counter	Chattering removal
	g_u1_sw2_cnt	uint8	SW2 judgment counter	Chattering removal
mtr_ssns_encd_foc.c (Motor control)	g_u1_mode_system	uint8	State management	0: Stop mode 1: Run mode 2: Error mode
	g_u2_run_mode	uint16	Operating mode management	2: Start mode 5: Normal operating mode
	g_u1_error_status	uint8	Error status management	1: Overcurrent error 2: Overvoltage error 3: High-speed error 7: Low voltage error FFh: Undefined error
	g_f4_vdc_ad	float32	Inverter bus voltage A/D value	[V]
	g_f4_vd_ref	float32	d-axis voltage target value	Current PI control output value [V]
	g_f4_vq_ref	float32	q-axis voltage target value	Current PI control output value [V]
	g_f4_iu_ad	float32	U-phase current	[A]
	g_f4_pre_iu_ad	float32	Previous U-phase current value	[A]
	g_f4_iv_ad	float32	V-phase current	[A]
	g_f4_pre_iv_ad	float32	Previous V-phase current value	[A]
	g_f4_iw_ad	float32	W-phase current	[A]
	g_f4_offset_iu	float32	U-phase current offset value	[A]
	g_f4_offset_iv	float32	V-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_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	
mtr_ssns_encd_foc.c (H/W Control)	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_vd	float32	d-axis current PI control output limit value	[V]

Table 4.5 List of Variables (2 / 3)

File	Variable Name	Type	Description	Remarks
mtr_ssns_encd_foc.c (H/W Control)	g_f4_lim_vq	float32	q-axis current PI control output limit value	[V]
	g_f4_ilim_vd	float32	d-axis current PI control integral term limit value	[V]
	g_f4_ilim_vq	float32	q-axis current PI control integral term limit value	[V]
	g_f4_lim_iq	float32	Speed PI control output limit value	[A]
	g_f4_ilim_iq	float32	Speed PI control integral term limit value	[A]
	g_f4_id_ref	float32	d-axis current target value	[A]
	g_f4_iq_ref	float32	q-axis current target value	[A]
	g_f4_speed_rad	float32	Calculated speed value	Electrical angle [rad/s]
	g_f4_ref_speed_rad	float32	Speed target 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_refu	float32	U-phase voltage target value	[V]
	g_f4_refv	float32	V-phase voltage target value	[V]
	g_f4_refw	float32	W-phase voltage target 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	
	speed	MTR_PI_CTRL	Speed 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	float32	d-axis current target value in start mode	[A]
	g_u2_cnt_adj_theta	uint16	Positioning time counter	
	g_f4_d_angle_rad	float32	Rotor position difference	[rad]
	g_f4_encd_tcnt	float32	Encoder timer counter value	
	g_f4_pre_encd_tcnt	float32	Previous encoder timer counter value	
	g_s2_angle_count	int16	Speed measurement counter	
	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

Table 4.5 List of Variables (3 / 3)

File	Variable Name	Type	Description	Remarks
mtr_ctrl_rzt1.c	g_u1_sw0_port_old	uint8	Previous SW0 value	0: ON 1: OFF
	g_u1_speed_first_update	uint8	Speed target initial update flag	1: ON 0: OFF
	g_u2_sw_speed_value	uint16	Speed setting	
	g_u2_def_speed_ref	uint16	Speed setting definition	

4.4 Macro Definitions

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

Table 4.6 List of Macro Definitions (1 / 4)

File Name	Macro Name	Defined Value	Remarks
main.h	MAX_SPEED	1,500	Maximum rotation speed target value (mechanical angle) [rpm]
	MIN_SPEED	600	Minimum rotation speed target value (mechanical angle) [rpm]
	PI	3.14159265f	Circle ratio
	RPM_RAD	(2*PI)/60	[rpm] to [rad/s] Unit conversion constant
	SW_ON	0	Low active
	SW_OFF	1	
	CHATTERING_CNT	200	Chattering removal
	FLAG_ON	1	Flag ON
	FLAG_OFF	0	Flag OFF
mtr_ctrl_rzt1.h	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_PORT_UP	PORT1.PODR.BIT.B4	U-phase (normal phase) output port
	MTR_PORT_VP	PORT1.PODR.BIT.B6	V-phase (normal phase) output port
	MTR_PORT_WP	PORT1.PODR.BIT.B2	W-phase (normal phase) output port
	MTR_PORT_SW1	PORTP.PIDR.BIT.B1	SW1 input port
	MTR_PORT_SW2	PORTP.PIDR.BIT.B2	SW2 input port
	MTR_PORT_LED9	PORT4.PODR.BIT.B6	LED9 output port
	MTR_PORT_LED8	PORT4.PODR.BIT.B5	LED8 output port
	MTR_LED_ON	1	High active
	MTR_LED_OFF	0	
	MTR_ENCD_TCNT	MTU2.TCNT	Encoder timer counter
	MTR_NUM_REF	4	Number of defined speed settings
	INIT_AXIS_0	0	Channel 0
	INIT_AXIS_1	1	Channel 1

Table 4.6 List of Macro Definitions (2 / 4)

File Name	Macro Name	Defined Value	Remarks
mtr_ssns_encd_foc.h	MTR_DEADTIME	0	Dead time [μs]
	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	electrical angle [A]
	MTR_OVERVOLTAGE_LIMIT	28	High voltage limit value [V]
	MTR_UNDERVOLTAGE_LIMIT	12	Low voltage limit value [V]
	MTR_TWOPI	2*3.14159265f	2π
	MTR_SQRT_2_3	0.81649658f	$\sqrt{(2/3)}$
	MTR_HALFPI	3.14159265f / 2	$(1/2)\pi$
	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	0.21	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	0.21	q-axis current PI control integral term gain
	MTR_SPEED_PI_KP	0.025	Speed PI control proportional term gain
	MTR_SPEED_PI_KI	0.00010	Speed PI control integral term gain
	MTR_SPEED_LPF_K	0.1	Speed LPF gain
	MTR_CURRENT_LPF_K	0.1	Current LPF gain

Table 4.6 List of Macro Definitions (3 / 4)

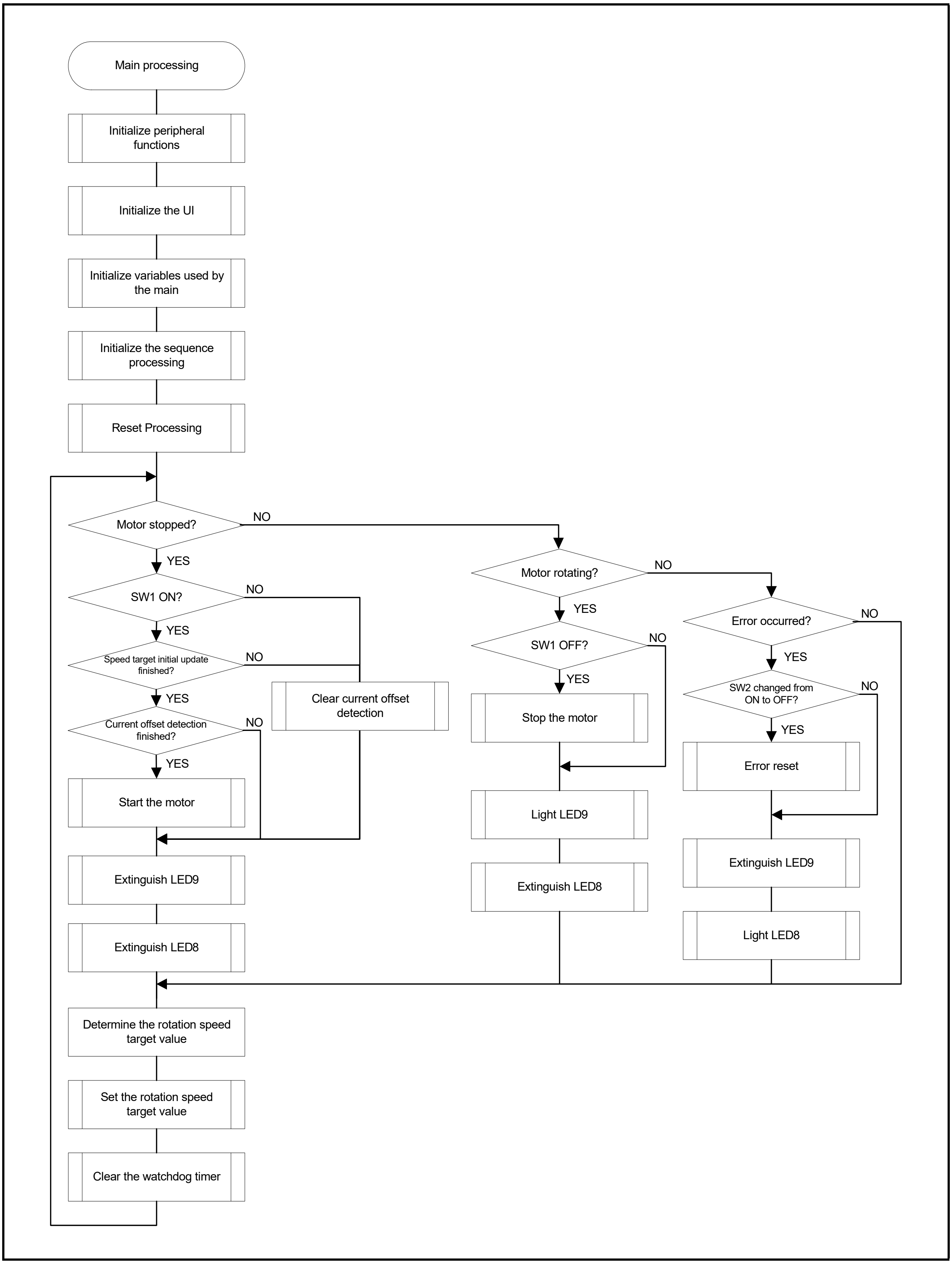
File Name	Macro Name	Defined Value	Remarks
mtr_ssns_encd_foc.h	MTR_LIMIT_VD	11	d-axis current PI control output limit value [V]
	MTR_LIMIT_VQ	11	q-axis current PI control output limit value [V]
	MTR_I_LIMIT_VD	11	d-axis current PI control integral term limit value [V]
	MTR_I_LIMIT_VQ	11	q-axis current 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	314.1593f	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 target 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 4.6 List of Macro Definitions (4 / 4)

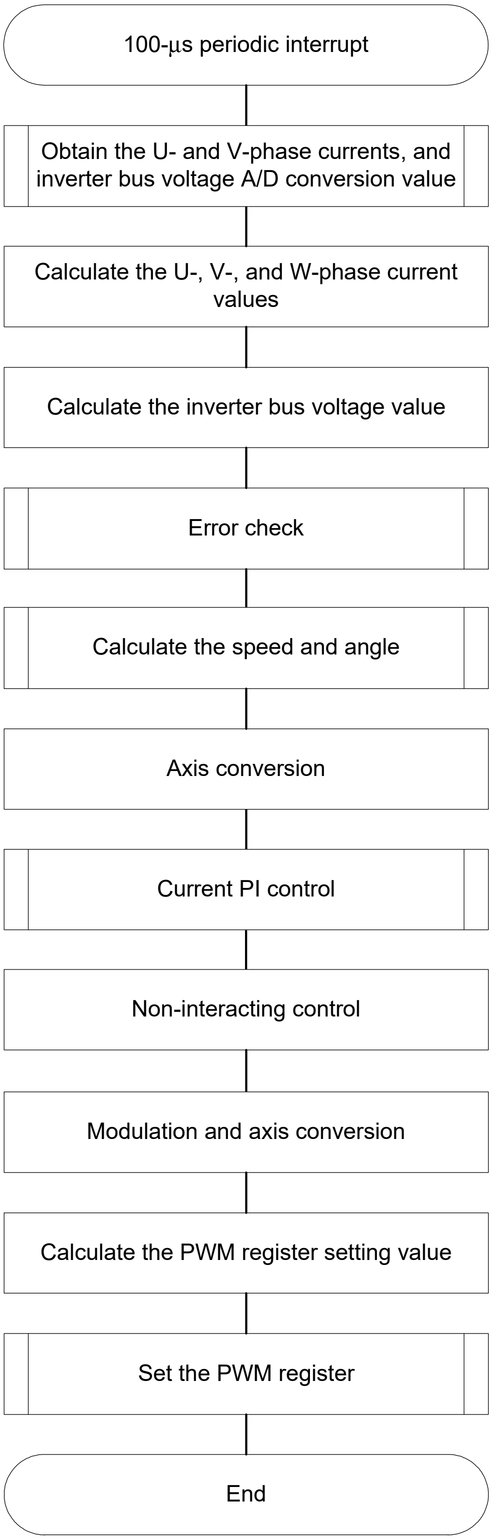
File Name	Macro Name	Defined Value	Remarks
mtr_ssns_encd_foc.h	MTR_BOOT_MODE	0x00	Boot mode
	MTR_OPENLOOP_MODE	0x01	Open loop mode
	MTR_START_MODE	0x02	Start mode
	MTR_HALL_120_MODE	0x03	Hall sensor 120-degree operation mode
	MTR_BEMF_120_MODE	0x04	BEMF sensorless 120-degree operation mode
	MTR_ENCD_FOC_MODE	0x05	Encoder vector operation mode
	MTR_LESS_FOC_MODE	0x06	Sensorless vector operation mode
	MTR_OVER_CURRENT_ERROR	0x01	Overcurrent error
	MTR_OVER_VOLTAGE_ERROR	0x02	Overvoltage error
	MTR_OVER_SPEED_ERROR	0x03	Overspeed error
	MTR_TIMEOUT_ERROR	0x04	Timeout error
	MTR_UNDER_VOLTAGE_ERROR	0x07	Undervoltage error
	MTR_UNKNOWN_ERROR	0xff	Undefined error
	MTR_MODE_STOP	0x00	Stopped state
	MTR_MODE_RUN	0x01	Rotating
	MTR_MODE_ERROR	0x02	Error state
	MTR_SIZE_STATE	3	Number of status
	MTR_EVENT_STOP	0x00	Motor stop event
	MTR_EVENT_RUN	0x01	Motor start event
	MTR_EVENT_ERROR	0x02	Motor error event
	MTR_EVENT_RESET	0x03	Motor reset event
	MTR_SIZE_EVENT	4	Number of events
	MTR_CURRENT_ADJUST_RUNNING	0x00	Detecting the current offset
	MTR_CURRENT_ADJUST_COMPLETE	0x01	Current offset detection end

4.5 Control Flow (Flowchart)

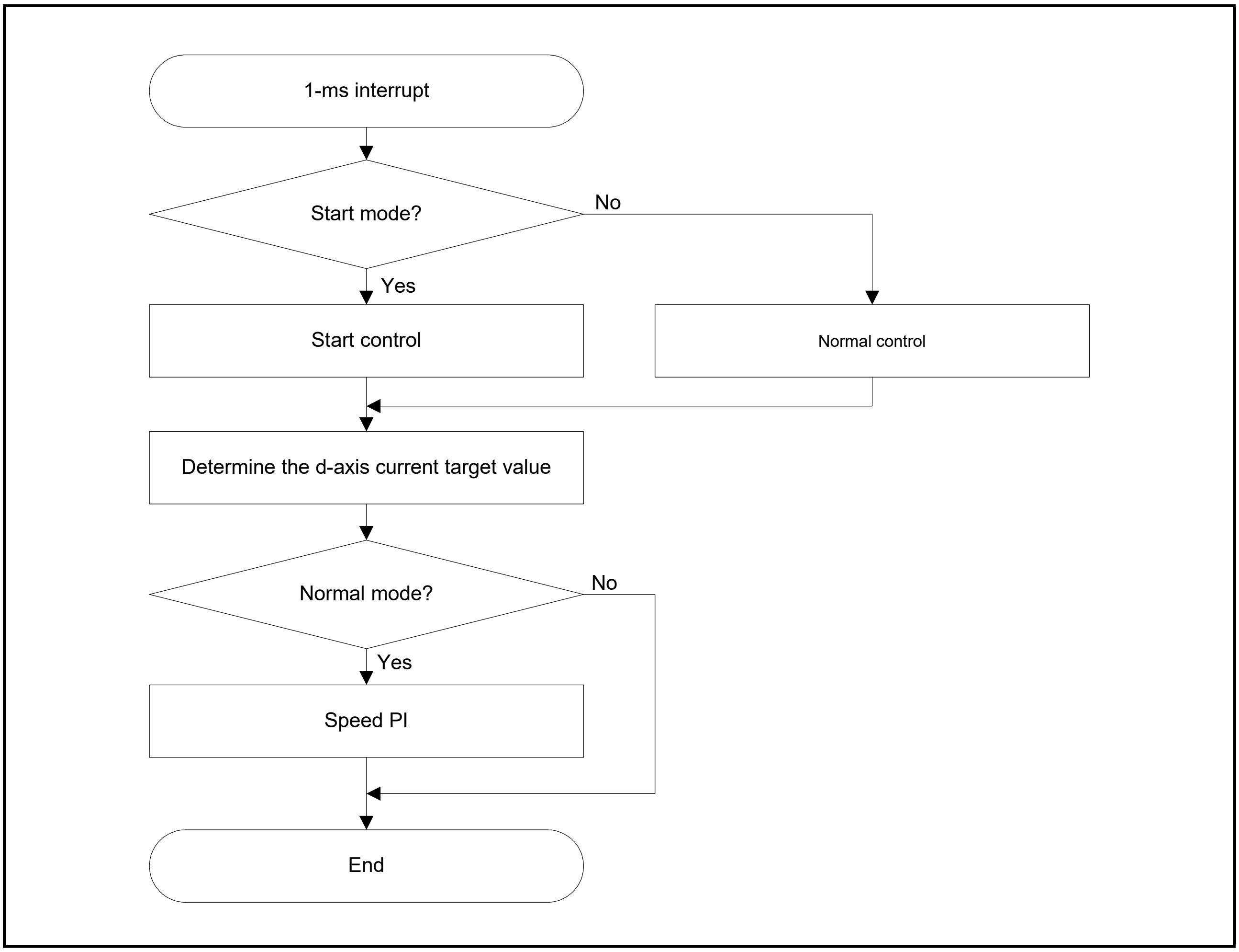
(1) Main processing



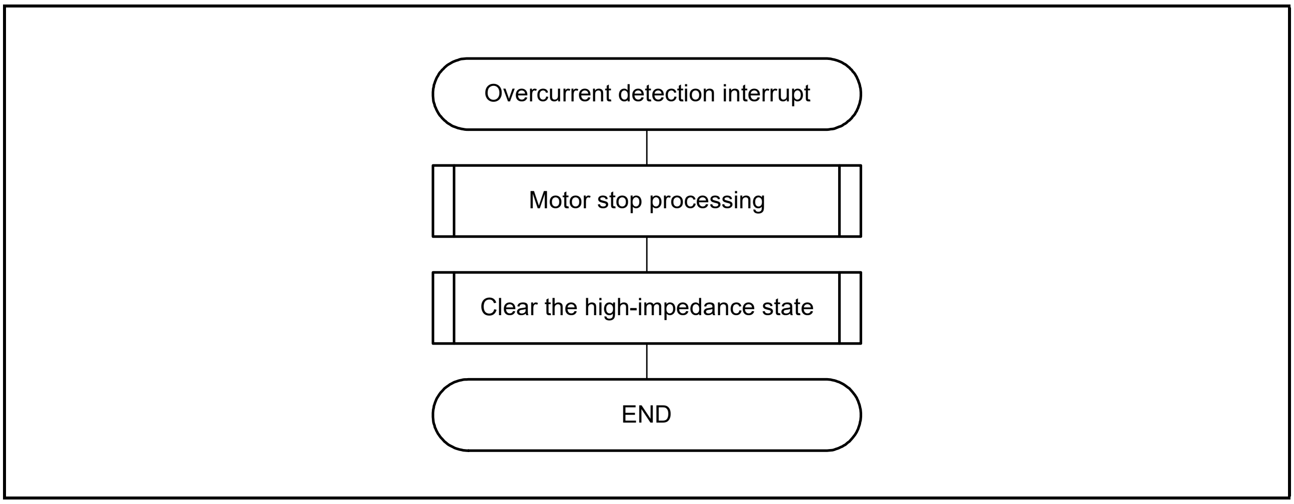
(2) 100-μs periodic interrupt processing



(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 (Speed Control) of the Permanent Magnet Synchronous Motor for the RZ/T1 Motion Control Solution Kit

Rev.	Date	Description	
		Page	Summary
1.00	Mar. 05, 2018	—	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 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