

RX24T/RX24U

Vector Control for Permanent Magnet Synchronous Motor with Encoder (Implementation)

Abstract

This application note aims to explain the sample programs for a permanent magnet synchronous motor with encoder, by using functions of RX24T/RX24U. The explanation includes, how to use the library of 'Renesas Motor Workbench' tool, that is support tool for motor control development.

The target software of this application note is only to be used as reference and Renesas Electronics Corporation does not guarantee the operations. Please use them after carrying out a thorough evaluation in a suitable environment.

Operation Checking Device

Operations of the target software of this application note are checked by using the following device.

- RX24T (R5F524TAADFP)
- RX24U (R5F524UEADFB)

Target Software

The target programs of this application note are as follows.

- RX24T_MRSSK_SPM_ENCD_FOC_CSP_RV101 (IDE: CS+)
- RX24T_MRSSK_SPM_ENCD_FOC_E2S_RV101 (IDE: e²studio)
- RX24U_MRSSK_SPM_ENCD_FOC_CSP_RV101 (IDE: CS+)
- RX24U_MRSSK_SPM_ENCD_FOC_CSP_RV101 (IDE: e²studio)

RX24T/RX24U vector control with encoder software for '24V Motor Control Evaluation System for RX23T and RX24T/RX24U CPU CARD'

Reference

- RX24T Group User's Manual: Hardware (R01UH0576EJ0200)
- RX24U Group User's Manual: Hardware (R01UH0658EJ0100)
- Application note: 'Vector control for permanent magnet synchronous motor with encoder (Algorithm)' (R01AN3789EJ0100)
- Renesas Motor Workbench V.1.00 User's Manual (R21UZ0004EJ0100)
- Renesas Solution Starter Kit 24V Motor Control Evaluation System for RX23T User's Manual (R20UT3697EJ0100)
- RX24T CPU CARD User's Manual (R20UT3696EJ0110)
- RX24U CPU CARD User's Manual (R12TU0018EJ0100)

Contents

1. Overview 3

2. System Overview..... 4

3. Descriptions of the Control Program 10

4. Motor Control Development Support Tool ‘Renesas Motor Workbench’ 29

1. Overview

This application note aims to explain the sample programs for a permanent magnet synchronous motor with encoder, by using functions of RX24T/RX24U. The explanation includes, how to use the library of 'Renesas Motor Workbench' tool, that is support tool for motor control development.

Note that these sample programs use the algorithm described in the application note 'Vector control for permanent magnet synchronous motor with encoder (Algorithm)'.

1.1 Development Environment

Table 1-1 and Table 1-2 show development environment of the sample programs explained in this application note.

Table 1-1 Hardware Development Environment

Microcontroller	Evaluation board	Motor
RX24T(R5F524TAADFP) RX24U(R5F524UEADFB)	24V inverter circuit & RX24T/RX24U CPU Card (Note 1)	FH6S20E-X81 (Note 2)

Table 1-2 Software Development Environment

CS+ version	e2studio version	Toolchain version
V5.00.00	V5.3.0.023	CC-RX V2.06.00

For purchase and technical support, contact sales representatives and dealers of Renesas Electronics Corporation.

Notes:1. 24V inverter board & RX24T CPU Card (RTK0EM0009C03402BJ) / RX24U CPU Card (RTK0EMX590C02000BJ) are products of Renesas Electronics Corporation.

2. FH6S20E-X81 is a product of NIDEC SERVO CORPORATION.
NIDEC SERVO (<http://www.nidec-servo.com/>)

2. System Overview

Overview of this system is explained below.

2.1 Hardware Configuration

The hardware configuration is shown below.

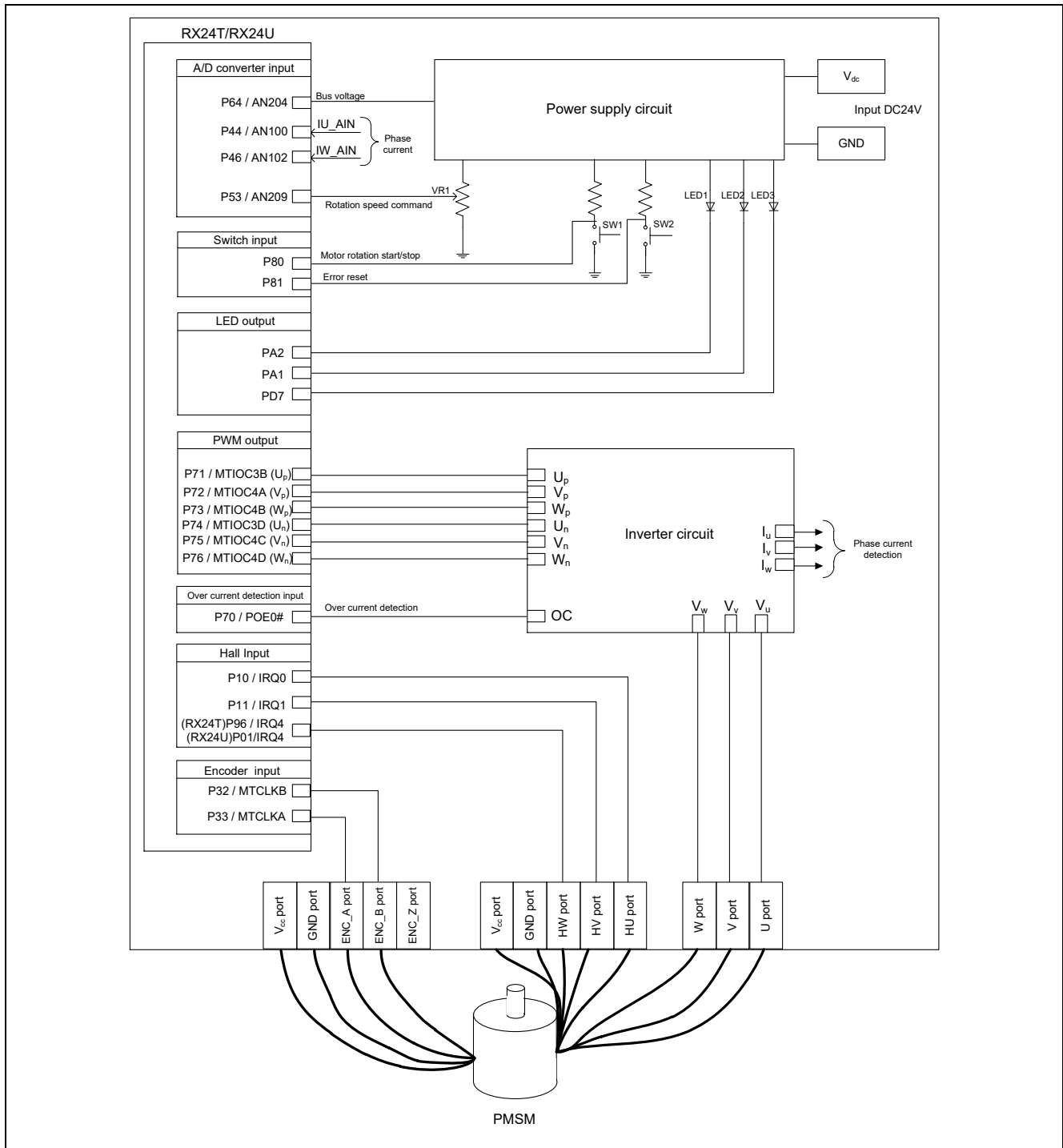


Figure 2-1 Hardware Configuration Diagram

2.2 Hardware Specifications

2.2.1 User Interfaces

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

Table 2-1 User Interfaces

Item	Interface component	Function
Rotation position / speed	Variable resistor (VR1)	Reference value of rotation position / speed input (analog value)
START/STOP	Toggle switch (SW1)	Motor rotation start/stop command
ERROR RESET	Toggle switch (SW2)	Command of recovery from error status
LED1	Yellow green LED	- At the time of motor rotation: ON - At the time of stop: OFF
LED2	Yellow green LED	- At the time of error detection: ON - At the time of normal operation: OFF
LED3	Yellow green LED	- Complete of positioning: ON - Uncomplete of positioning: OFF
RESET	Push switch (RESET1)	System reset

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

Table 2-2 Port Interfaces

R5F524TAADFP Port name	R5F524UEADFB Port name	Function
P64 / AN204		Inverter bus voltage measurement
P53 / AN209		For position / speed command value input (analog value)
P80		START/STOP toggle switch
P81		ERROR RESET toggle switch
PA2		LED1 ON/OFF control
PA1		LED2 ON/OFF control
PD7		LED3 ON/OFF control
P44 / AN100		U phase current measurement
P46 / AN102		W phase current measurement
P71 / MTIOC3B		PWM output (U_p)
P72 / MTIOC4A		PWM output (V_p)
P73 / MTIOC4B		PWM output (W_p)
P74 / MTIOC3D		PWM output (U_n)
P75 / MTIOC4C		PWM output (V_n)
P76 / MTIOC4D		PWM output (W_n)
P10 / IRQ0		Hall Phase-U signal input
P11 / IRQ1		Hall Phase-V signal input
P96 / IRQ4	P01 / IRQ4	Hall Phase-W signal input
P32 / MTCLKB		Encoder Phase-B signal input
P33 / MTCLKA		Encoder Phase-A signal input
P70 / POE0#		PWM emergency stop input at the time of over-current detection

2.2.2 Peripheral Functions

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

Table 2-3 List of the Peripheral Functions

12-bit A/D	CMT	MTU3	POE3
<ul style="list-style-type: none"> - Rotation speed command value input - Current of each phase U and W measurement - Inverter bus voltage measurement 	1 [ms] interval timer	<ul style="list-style-type: none"> - Complementary PWM output - Encoder phase counter - Encoder count capture 	Set PWM output ports to high impedance state to stop the PWM output.

(1) 12-bit A/D converter (S12ADE)

U phase current (I_u), W phase current (I_w), inverter bus voltage (V_{dc}) and rotation speed reference are measured by using the single scan mode (use hardware trigger). The sample-and-hold function is used for U phase current (I_u) and W phase current (I_w) measurement.

(2) Compare match timer (CMT)

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

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

The operation mode varies depending on channels. On the channels 3 and 4, output (active level: high) with dead time is performed by using the complementary PWM mode.

The channel 1 of MTU3 operate in phase counting mode, the counter is incremented or decremented according to the phase difference between Phase-A and Phase-B signals from the encoder.

The channel 0 of MTU3 is used as free-run timer for speed measurement.

(4) Port output enable 3 (POE3)

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

2.3 Software Configuration

2.3.1 Software File Configuration

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

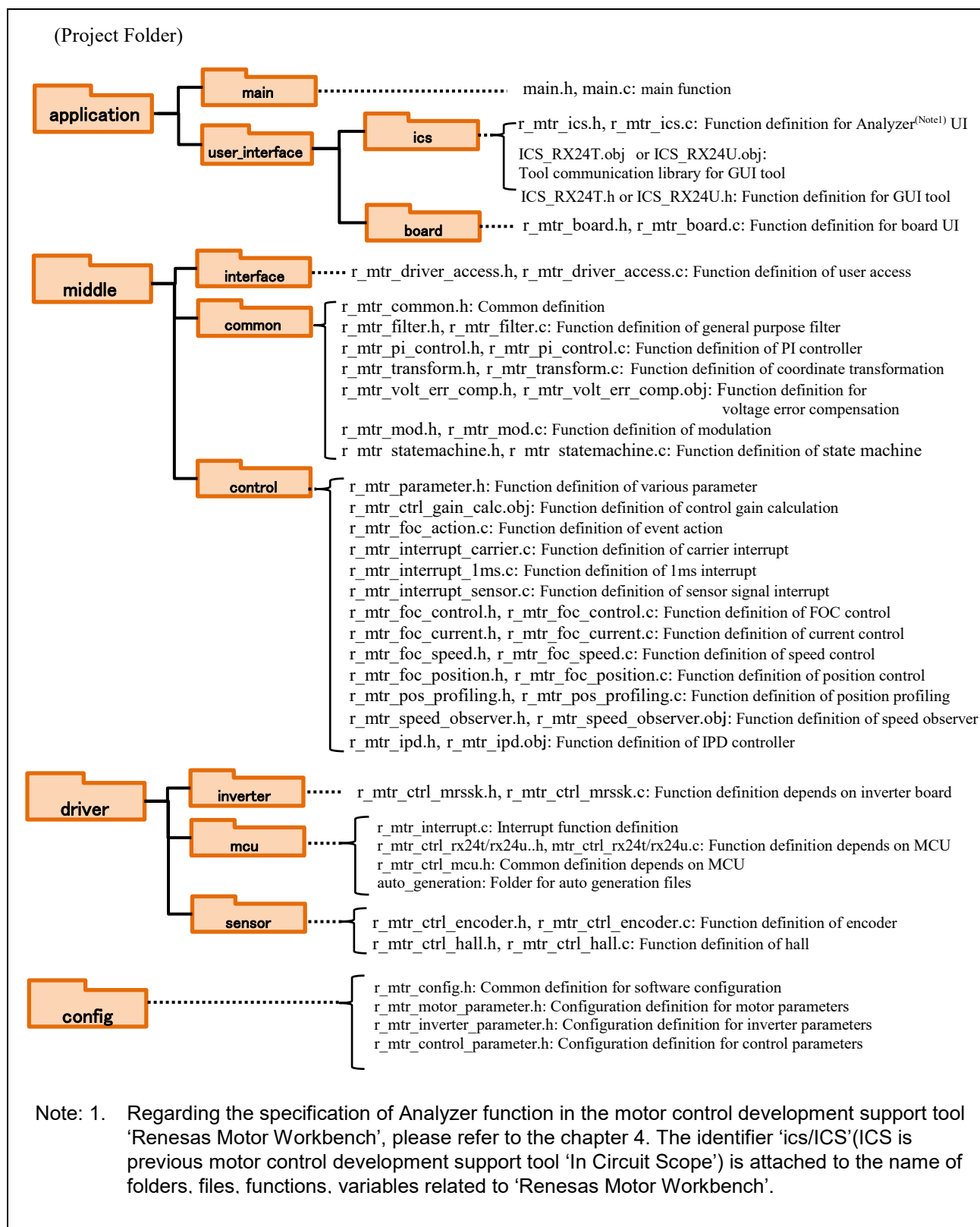


Figure 2-2 Folder and File Configuration

2.3.2 Module Configuration

Module configuration of the sample programs is described below.

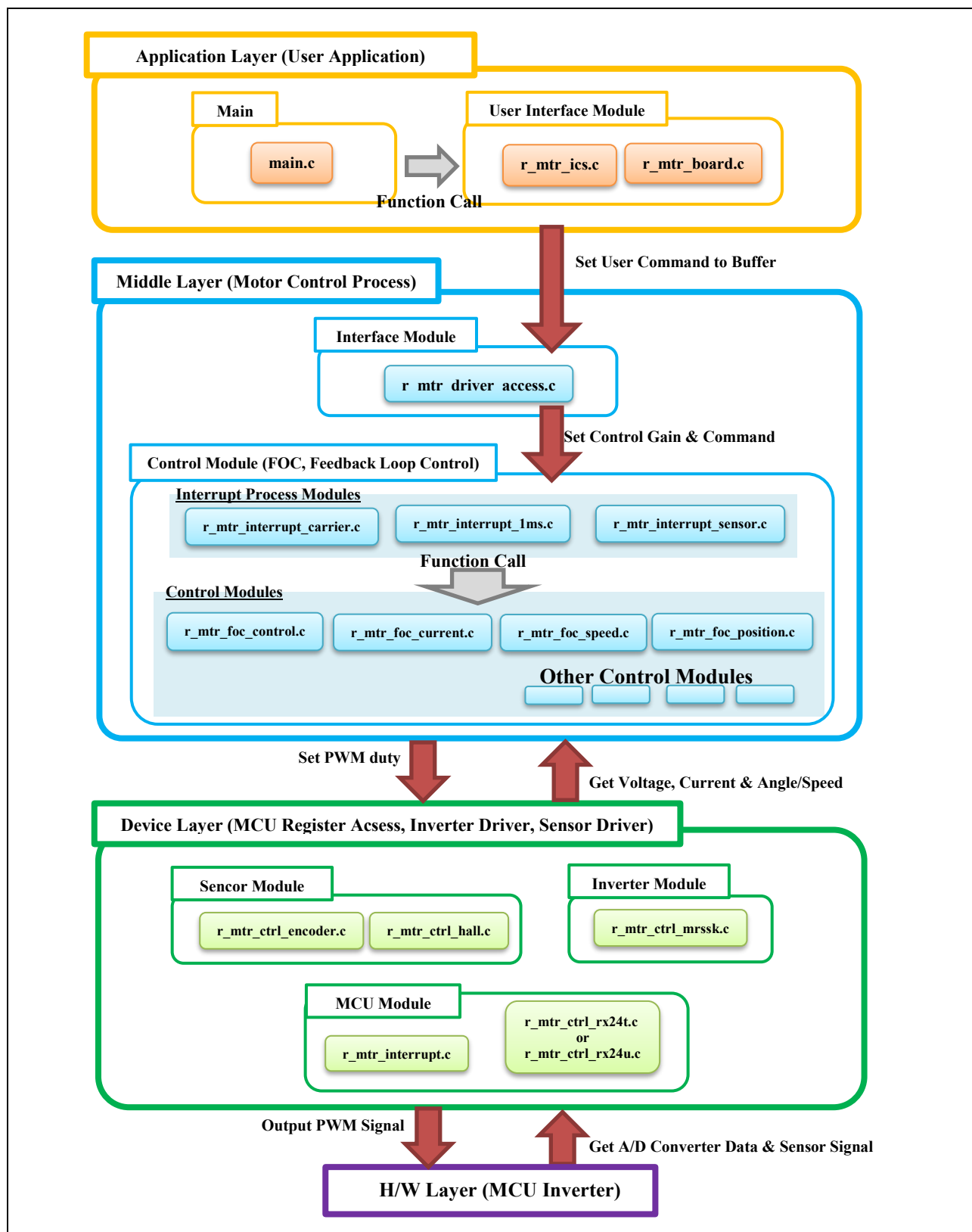


Figure 2-3 Module Configuration

2.4 Software Specifications

Table 2-4 shows basic software specification of this system. For details of the vector control, refer to the application note 'Vector control of permanent magnet synchronous motor with encoder: algorithm'.

Table 2-4 Basic Specifications of Vector Control PMSM with Encoder Software

Item	Content	
Control method	Vector control	
Motor position control start/stop	Determined depending on the level of SW1 ('Low': control start 'High': stop) or input from Analyzer	
Position detection of rotor magnetic pole	Incremental encoder (A-B Phase), Hall sensor (UVW Phase)	
Input voltage	DC 24 [V]	
Main clock frequency	80 [MHz]	
Carrier frequency (PWM)	20 [kHz] (carrier cycle: 50[us])	
Dead time	2 [μs]	
Control cycle (Current loop)	100 [μs] (twice the carrier cycle)	
Control cycle (Speed and Position loop)	1 [ms]	
Management of position command value	Board UI	Position command generation: Direct input of VR1 (input range) -180° ~ 180°
	ICS UI	Position command generation: Position profile of trapezoidal curve for speed command value (input range) - 32768° ~ 32767° (Max speed) CW / CCW: 2000[rpm]
Management of speed command value	CW: 0 [rpm] to 2000rpm CCW: 0 [rpm] to 2000rpm	
Accuracy of position	0.3° (Encoder pulse: 300[ppr] 4 for multiplying 1200[cpr])	
Dead band of position (Note)	Encoder count ±1 [cpr] (±0.3°)	
Natural frequency of each control system	Current control system: 300Hz Speed control system: 30Hz Position control system: 10Hz	
Optimization setting for compiler	Optimization level	2 (-optimize=2) (default)
	Optimization method	Size priority (default)
ROM/RAM size	ROM: 17.9KB RAM: 4.6KB	
Processing stop for protection	<p>Motor control signal outputs (six outputs) will be disabled, under any of the following conditions.</p> <ol style="list-style-type: none"> 1. Current of each phase exceeds 3.28 [A] (monitored every 100 [μs]) 2. Inverter bus voltage exceeds 28 [V] (monitored every 100 [μs]) 3. Inverter bus voltage is less than 14 [V] (monitored every 100 [μs]) 4. Rotation speed exceeds 3000 [rpm] (monitored every 100 [μs]) <p>When an external over-current signal is detected (when a falling edge of the POE0# port is detected) and when the output short circuit is detected, the PWM output ports are set to high impedance state.</p>	

Note: Dead zone is provided to prevent hunting in positioning.

3. Descriptions of the Control Program

The target sample programs of this application note are explained here.

3.1 Contents of Control

3.1.1 Motor Start/Stop

The start and stop of the motor are controlled by input from Analyzer function of 'Renesas Motor Workbench' or SW1 switch of RSSK board.

A general-purpose port is assigned to SW1. The port is read within the main loop. When the port is at a 'Low' level, the software determines that the motor should be started. Conversely, when the level is switched to 'High', the program determines that the motor should be stopped.

3.1.2 A/D Converter

(1) Motor Rotation Position and Speed Command Value

The motor rotation position and speed command value can be set by Analyzer input or A/D conversion of the VR1 output value (analog value). The A/D converted VR1 value is used as rotation speed command value, as shown below.

Table 3-1 Conversion Ratio of the Rotation Speed Command Value

Item	Conversion ratio (Command value: A/D conversion value)		Channel
Rotation position command value	CW	0° ~ 180°: 0800H ~ 0FFFH	AN209
	CCW	0° ~ -180°: 07FFH ~ 0000H	
Rotation speed command value	CW	0 [rpm] ~ 2000[rpm]: 0800H ~ 0FFFH	
	CCW	0 [rpm] ~ 2000[rpm]: 07FFH ~ 0000H	

(2) Inverter Bus Voltage

Inverter bus voltage is measured as given in Table 3-2.

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

Table 3-2 Inverter Bus Voltage Conversion Ratio

Item	Conversion ratio (Inverter bus voltage: A/D conversion value)	Channel
Inverter bus voltage	0 [V] to 111 [V]: 0000H to 0FFFH	AN204

(3) U, W Phase Current

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

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

Item	Conversion ratio (U, W phase current: A/D conversion value)	Channel
U, W phase current	-10 [A] to 10 [A]: 0000H to 0FFFH (Note 1)	Iu : AN100 Iw : AN102

Notes:1 For more details of A/D conversion characteristics, refer to 'RX24T Group User's Manual: Hardware', 'RX24U Group User's Manual: Hardware'

3.1.3 Position Profile Generation

(Position Profile of Trapezoidal Curve for Speed Command Value)

In vector control software for PMSM with encoder, the position profile generation is used to create command value (input position value). The implementation of command value is each control cycle is used as method of managing acceleration and the maximum speed value with respect to target position value.

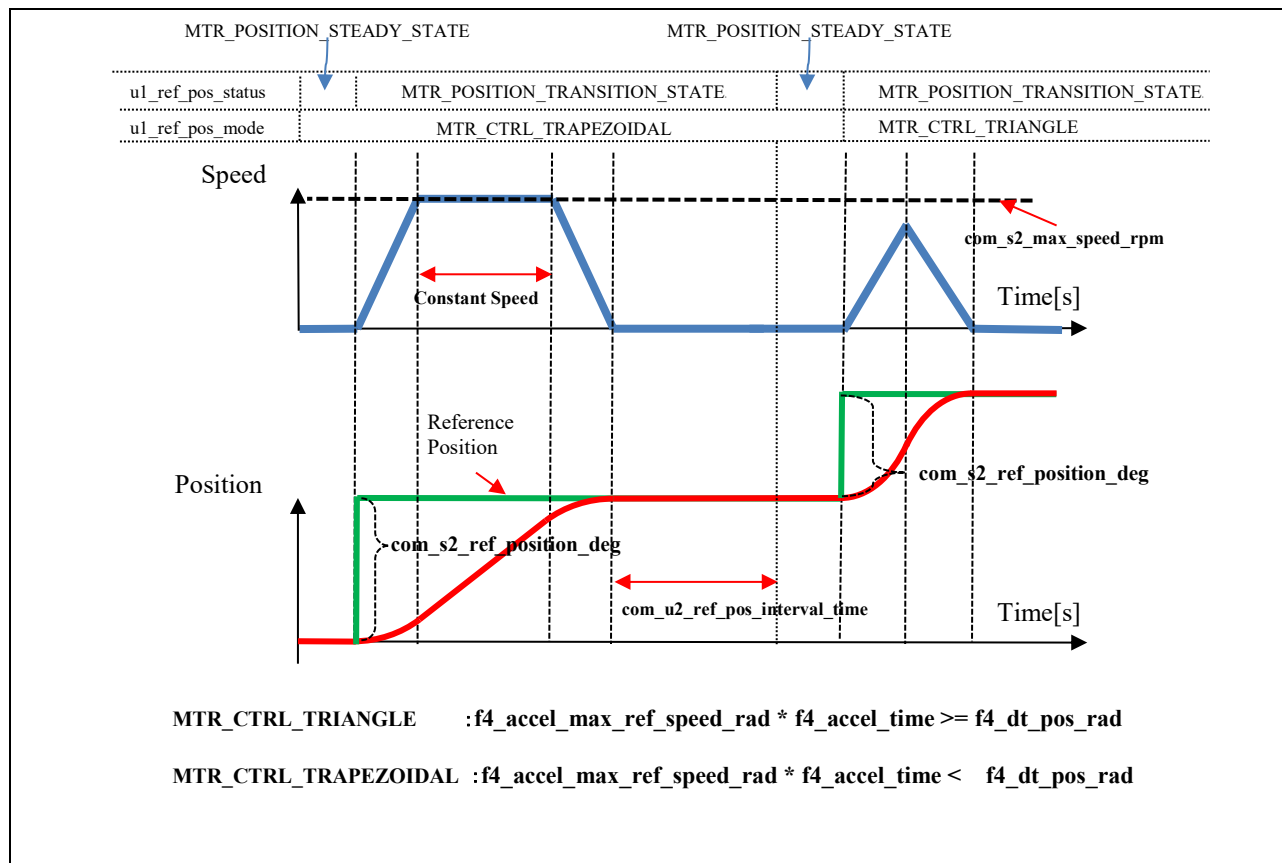


Figure 3-1 Position Profile Generation

Enter the following variables from the Analyzer to create a command value.

- Position reference [degree] (`com_s2_ref_position_deg`)
- Acceleration time (`com_f4_accel_time`)
- Maximum speed command value (`com_f4_accel_max_ref_speed_rad`)
- Position stabilization wait time (`com_u2_ref_pos_interval_time`)

3.1.4 Speed Measurement

In order to obtain better real-time performance and higher speed resolution at low speed, this system use encoder signal edge interval to calculate speed, the speed extrapolation is used in PI control calculation. In addition, taking the difference between rise time and fall time and the accuracy of quadrature of encoder signal into consideration, the speed is calculated with time elapsed and angle changed in one period of encoder Phase-A or Phase-B signals.

(1) Speed Calculation

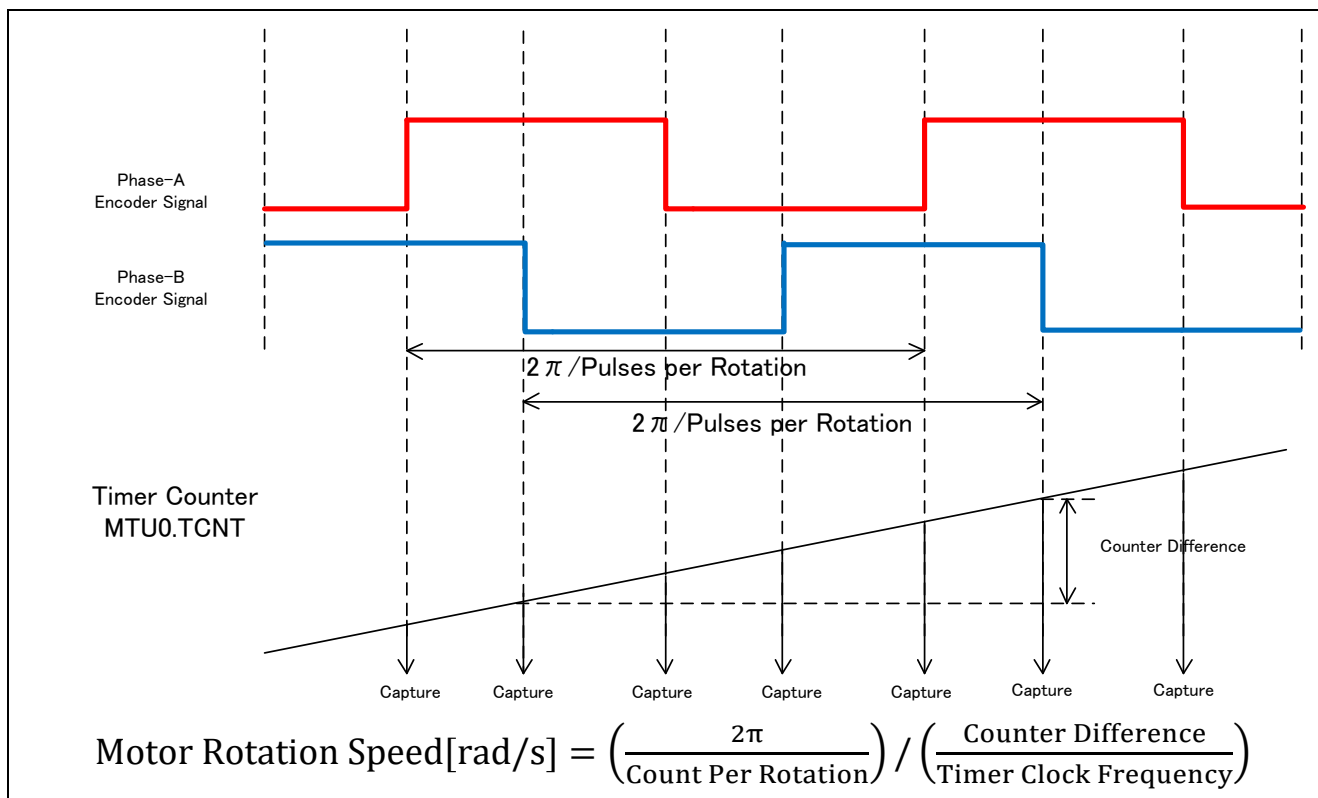


Figure 3-2 Speed Calculation using Encoder

3.1.5 Modulation

The target software of this application note uses pulse width modulation (hereinafter called PWM) to generate the input voltage to the motor. And the PWM waveform is generated by the triangular wave comparison method.

(1) Triangular Wave Comparison Method

The triangular wave comparison method is used to output the voltage command value. By this method, the pulse width of the output voltage can be determined by comparing the carrier waveform (triangular wave) and voltage command value waveform. The voltage command value of the pseudo sinusoidal wave can be output by turning the switch on or off when the voltage command value is larger or smaller than the carrier wave respectively.

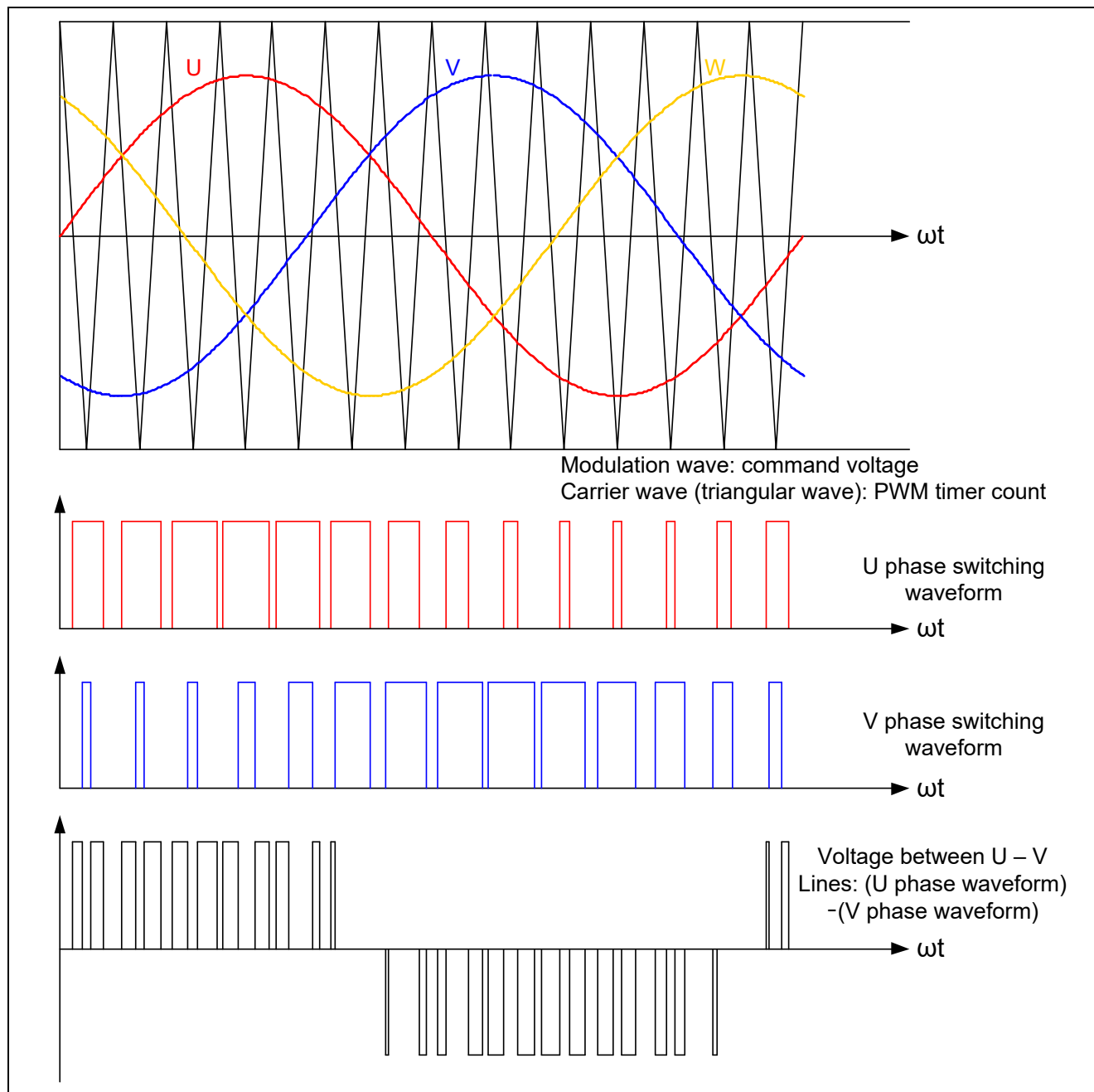


Figure 3-3 Conceptual Diagram of the Triangular Wave Comparison Method

Here, as shown in Figure 3-4, ratio of the output voltage pulse to the carrier wave is called duty.

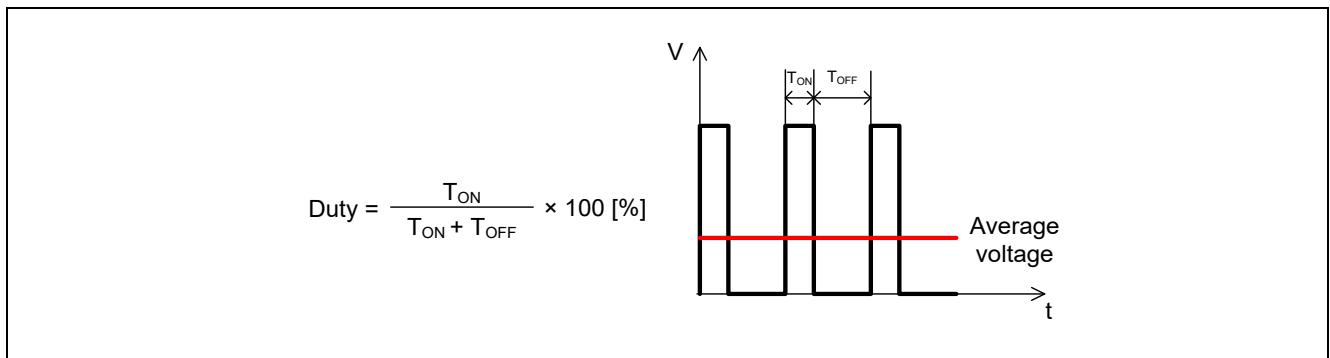


Figure 3-4 Definition of Duty

Modulation factor m is defined as follows.

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

m : Modulation factor V : Voltage command value E : Inverter bus voltage

The voltage command can be generated by setting PWM compare register properly to obtain the desired duty.

3.1.6 State Transition

Figure 3-5 is a state transition diagram of the vector control software. In the target software of this application note, the software state is managed by 'SYSTEM MODE' and 'RUN MODE'. And 'Control Config' shows the active control system in the software.

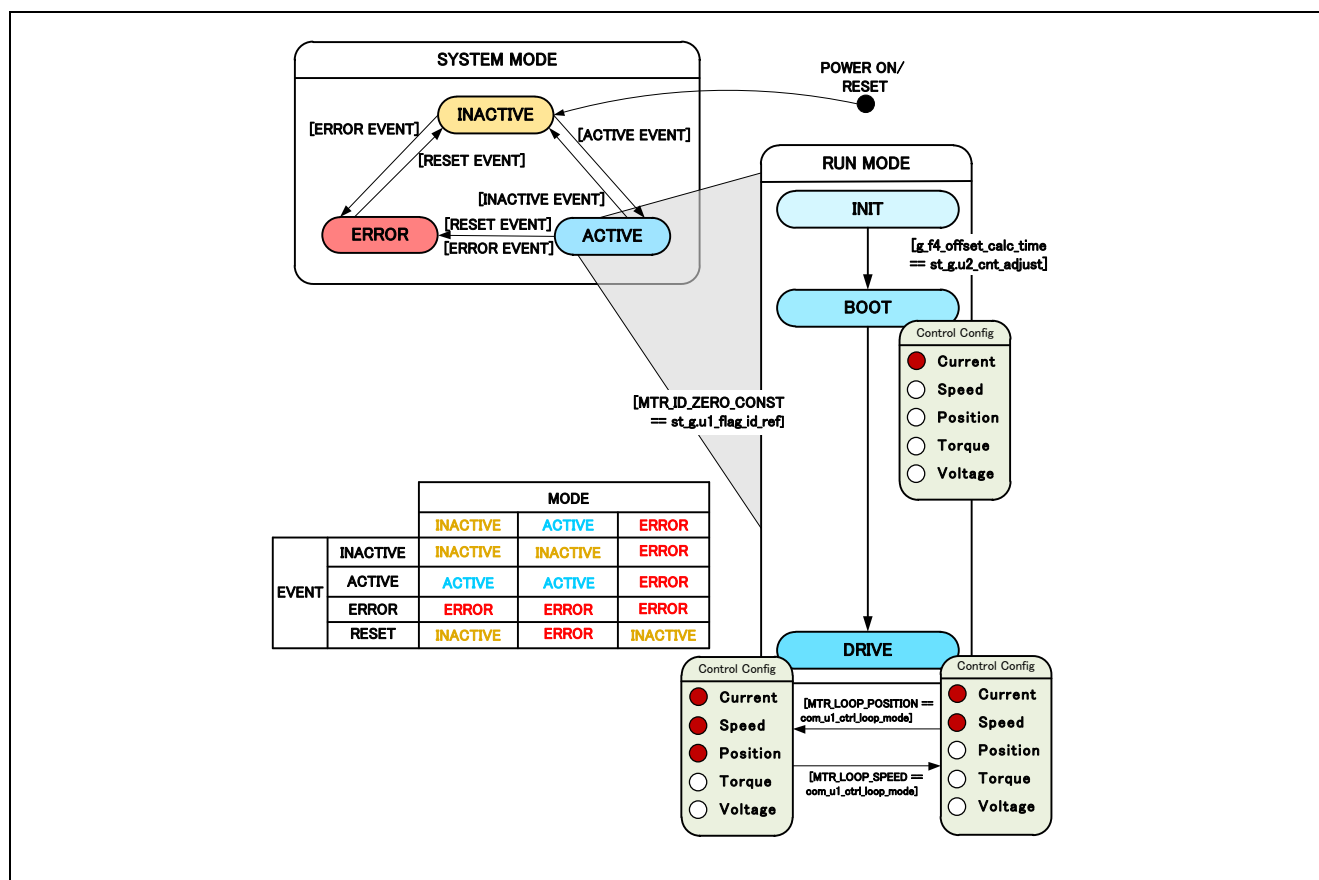


Figure 3-5 State Transition Diagram of Vector Control PMSM with Encoder Software

(1). SYSTEM MODE

'SYSTEM MODE' indicates the operating states of the system. The state transits on occurrence of each event (EVENT). 'SYSTEM MODE' has 3 states that are motor drive stop (INACTIVE), motor drive (ACTIVE), and abnormal condition (ERROR).

(2). RUN MODE

'RUN MODE' indicates the condition of the motor control. 'RUN MODE' transits sequentially as shown in Figure 3-5 when 'SYSTEM MODE' is 'ACTIVE'.

(3). EVENT

When 'EVENT' occurs in each 'SYSTEM MODE', 'SYSTEM MODE' changes as shown the table of Figure 3-5, according to that 'EVENT'.

Table 3-1 List of EVENT

EVENT name	Occurrence factor
INACTIVE	by user operation
ACTIVE	by user operation
ERROR	when the system detects an error
RESET	by user operation

3.1.7 Startup Method

Figure 3-6 shows the software implementation of d-axis and encoder alignment method. The d-axis alignment method used as startup control of position control method, in initialization mode (MTR_MODE_INIT) and Boot mode (MTR_MODE_BOOT). In drive mode (MTR_MODE_DRIVE) vector control is implemented for PMSM with Encoder. Each reference value setting of d-axis current, q-axis current and speed is managed by respective status.

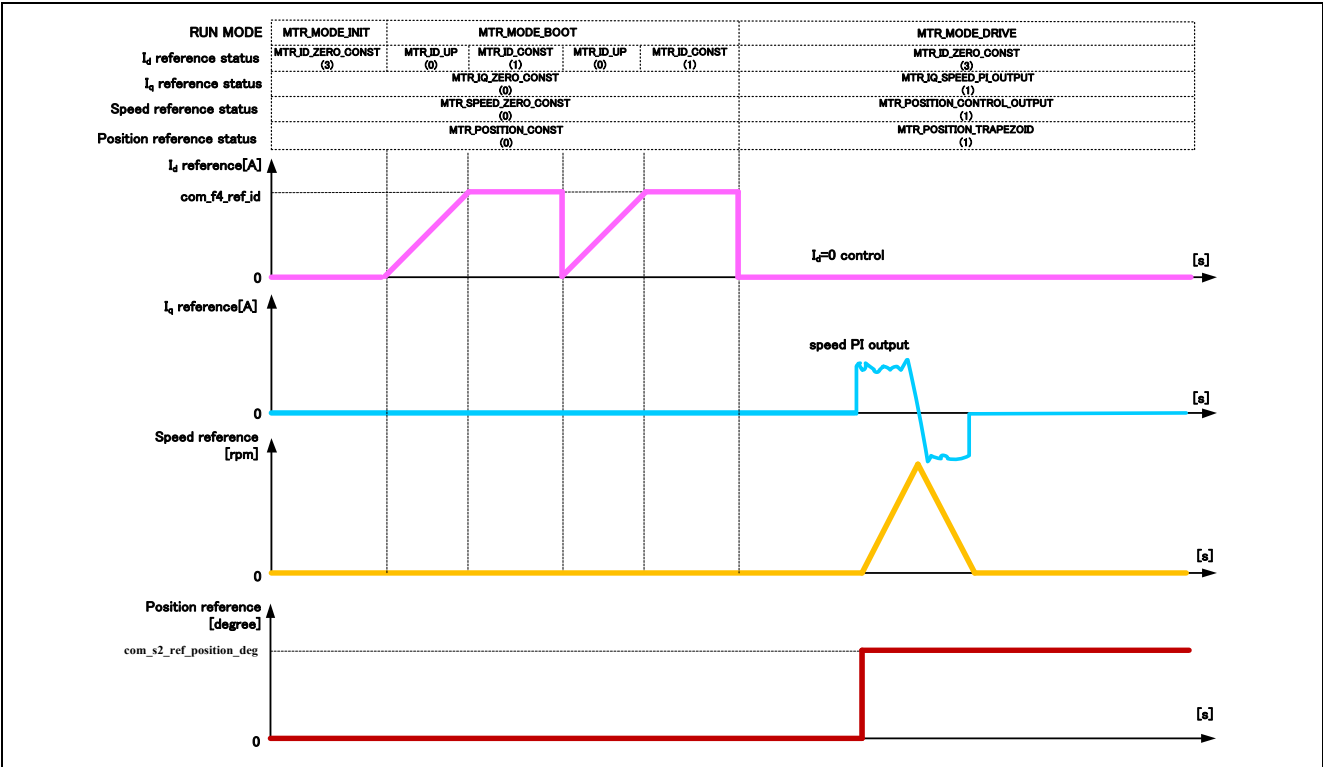


Figure 3-6 Startup Position Control of Vector Control PMSM with encoder Software

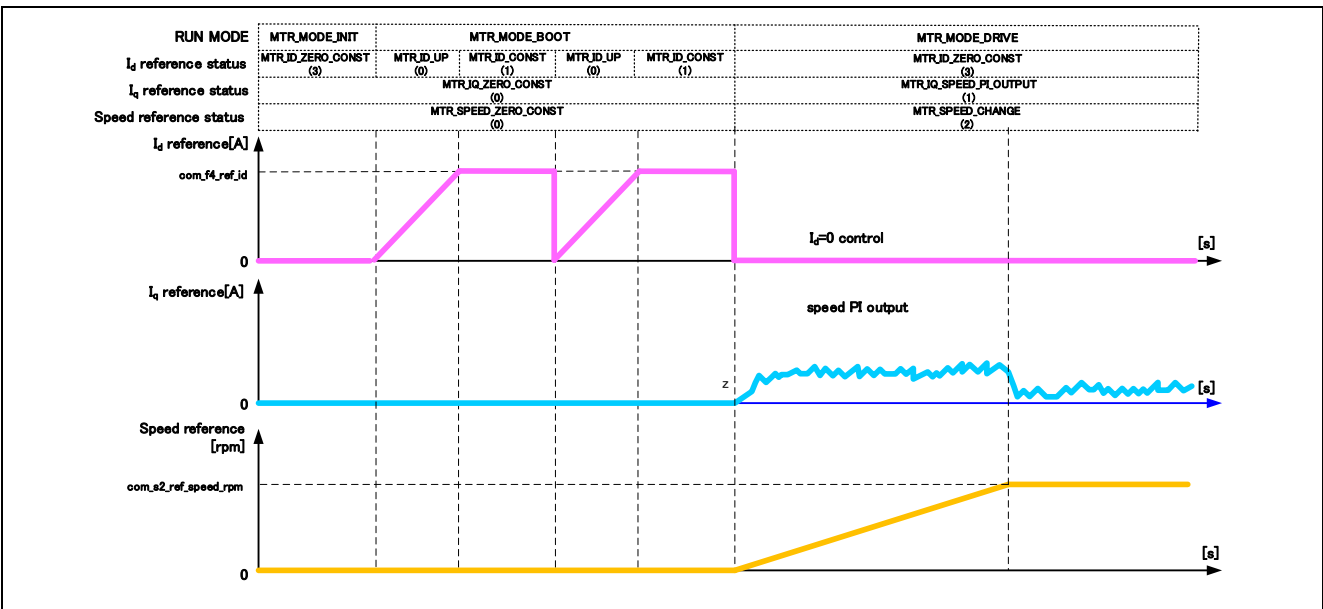


Figure 3-7 Startup Speed Control of Vector Control PMSM with Encoder Software

For details of the position control of a vector controlled PMSM using encoder, refer to the application note 'Vector control of permanent magnet synchronous motor with encoder: algorithm'.

3.1.8 System Protection Function

This control program has the following error status and executes emergency stop functions in case of occurrence of respective errors. Table 3-4 shows each setting value for the system protection function.

- Over-current error

The over current detection is performed by both hardware detection method as well as software detection method. In response to over current detection an emergency stop signal is generated from the hardware (hardware detection). When the emergency stop signal is generated, the PWM output ports are set to high impedance state.

In addition, U, V, and W phase currents are monitored in over current monitoring cycle. When an over current is detected, the CPU executes emergency stop (software detection). The over current limit value is calculated from the nominal current of the motor [MP_NOMINAL_CURRENT_RMS].

- Over voltage error

The inverter bus voltage is monitored in over-voltage monitoring cycle. When an over-voltage is detected, the CPU performs emergency stop. Here, the over-voltage limit value is set in consideration of the error of resistance value of the detect circuit.

- Under-voltage error

The inverter bus voltage is monitored in under-voltage monitoring cycle. The CPU performs emergency stop when under-voltage is detected. Here, the low voltage limit value is set in consideration of the error of resistance value of the detect circuit.

- Over-speed error

The rotation speed is monitored in rotation speed monitoring cycle. The CPU performs emergency stop when the speed is over the limit value.

Table 3-4 Setting Values of the System Protection Function

Over-current error	Over-current limit value [A]	3.82
	Monitoring cycle [μ s]	100
Over-voltage error	Over-voltage limit value [V]	28
	Monitoring cycle [μ s]	100
Under-voltage error	Under-voltage limit value [V]	14
	Monitoring cycle [μ s]	100
Over-speed error	Speed limit value [rpm]	3000
	Monitoring cycle [μ s]	100

3.2 Function Specifications of Vector Control using Encoder Software

The control process of the target software of this application note is mainly consisted of 100[us] period interrupt (carrier interrupt) and 1[ms] period interrupt. In following Figure 3-8, the control process in the red broken line part is executed every 100[us] period, and the control process in the blue broken line part is executed every 1[ms] period.

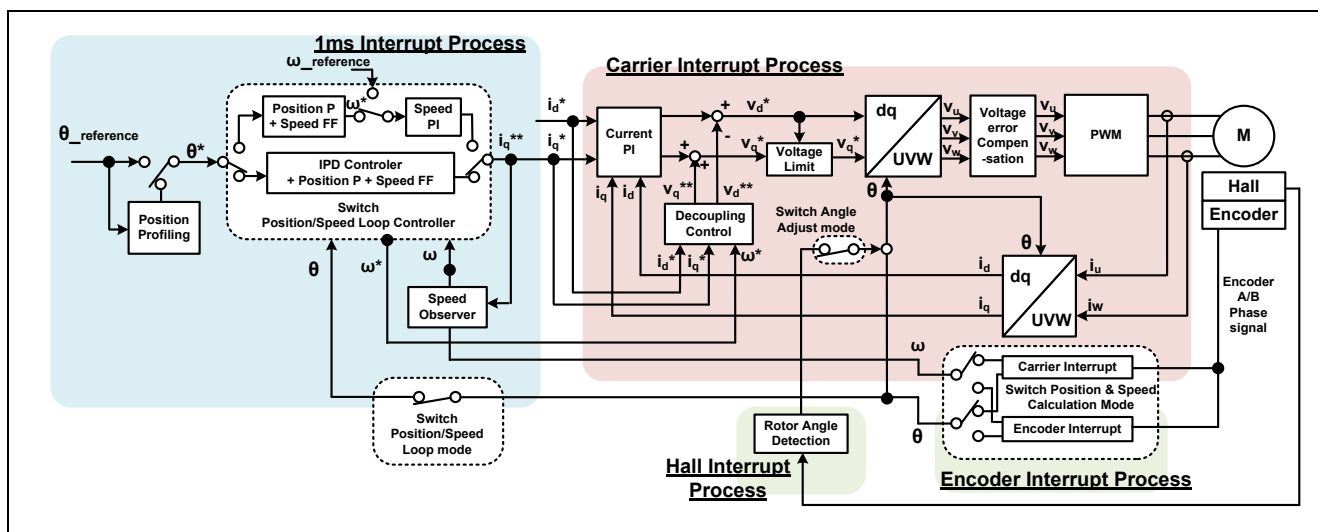


Figure 3-8 System Block of Vector Control with Encoder

This chapter shows the specification of 4 interrupt functions and functions executed in each interrupt cycle. In the following tables, only essential functions of the vector control are listed. Regarding the specification of functions not listed in following tables, refer to source codes.

Table 3-5 List of Control Functions 'mtr_interrupt.c'

File name	Function name	Process overview
r_mtr_interrupt_carrier.c	mtr_foc_carrier_interrupt Input: (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Calling every 100 [us] - Current and voltage monitoring - Error detection - Current offset detection - Vector calculation - Current PI control
r_mtr_interrupt_1ms.c	mtr_foc_1ms_interrupt Input: (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Calling every 1 [ms] - Startup control - d-axis/q-axis current and speed reference set - Speed PI control
r_mtr_interrupt_sensor.c	mtr_angle_adj_hall_interrupt Input: (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Called when the Hall phase signals (Phase-U/V/W) - Get Hall signal - Rotor phase calculation - Hall error process - Disable Hall interrupt
	mtr_encd_pos_speed_calc_interrupt Input (mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Called when the encoder phase counts (Phase-A and B) - Rotor phase calculation - Speed calculation

Table 3-6 List of Functions for 100us interrupt [1/2]

File name	Function name	Process overview
r_mtr_ctrl_mrssh.c	mtr_get_current_iuiw Input:(float*) f4_iu_ad / U phase current A/D conversion value (float*) f4_iw_ad / W phase current A/D conversion value (uint8_t) u1_id / Motor ID Output: None	Obtaining the UVW phase current
	mtr_get_vdc Input:(uint8_t) u1_id / Motor ID Output:(float) f4_temp_vdc / Vdc value	Obtaining the Vdc
r_mtr_foc_control.c	mtr_error_check Input:(mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Error monitoring
	mtr_current_offset_adjustment Input:(mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	UVW phase current offset adjustment
	mtr_calib_current_offset Input:(mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	UVW phase current offset calculation
	mtr_encd_pos_speed_calc Input:(mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Position and speed calculation for encoder pulse
	mtr_angle_speed Input:(mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Rotor phase and speed related process (Switching calculation method)
	mtr_foc_voltage_limit Input:(mtr_foc_control_t *) st_foc / Structure pointer for vector control Output:None	Voltage command value limit
r_mtr_foc_current.c	mtr_current_pi_control Input:(mtr_foc_control_t *) st_foc / Structure pointer for vector control Output: None	Current PI
	mtr_decoupling_control Input:(mtr_current_control_t *) st_cc / Current control structure (float)f4_speed_rad / speed (mtr_parameter_t *) mtr_para / motor parameter structure Output: None	Decupling control
r_mtr_transform.c	mtr_transform_uvw_dq_abs Input: (const mtr_rotor_angle_t *) p_angle / Structure pointer for phase management (const float*) f4_uvw / UVV phase pointer (float*) f4_dq / dq-axis pointer Output: None	Coordinate transform UVW to dq
	mtr_transform_dq_uvw_abs Input: (const mtr_rotor_angle_t *) p_angle / Structure pointer for phase management (const float*) f4_dq / dq-axis pointer (float*) f4_uvw / UVW phase pointer Output: None	Coordinate transform dq to UVW

Table 3-7 List of Functions for 100us Interrupt [2/2]

File name	Function name	Process overview
r_mtr_volt_err_comp.c	mtr_volt_err_comp_main Input:(mtr_volt_comp_t *) st_volt_comp / Voltage error compensation structure (float*) p_f4_v_array / Three phase voltage compensation value array pointer (float*) p_f4_i_array / Three phase current compensation value array pointer (float)f4_vdc / Vdc value Output: None	Voltage error compensation
r_mtr_ctrl_rx24t.c / r_mtr_ctrl_rx24u.c	mtr_inv_set_uvw Input:(float) f4_modu / U phase modulation factor (float) f4_modv / V phase modulation factor (float) f4_modw / W phase modulation factor (uint8_t) u1_id / Motor ID Output: None	PWM output setting

Table 3-8 List of Functions for 1ms Interrupt

File name	Function name	Process overview
r_mtr_ctrl_hall.c	mtr_angle_adj_hall_init Input:(mtr_hall_t *) st_hc / Hall sensor structure Output:(float) f4_hall_angle_rad / angle of signal detection for Hall sensor	Initialize rotor angle detection for Hall sensor
r_mtr_foc_control.c	mtr_hall_error Input:(mtr_foc_control_t *) st_foc / FOC motor structure (float) f4_hall_angle_rad / angle of Hall Output: None	Hall sensor error process
r_mtr_ctrl_encoder.c	mtr_set_encd_tcmt Input:(uint8_t) u1_id / Motor ID (uint16_t) u2_cnt_value / counter value Output: None	Set for encoder count resister
	mtr_encd_cnt_reset Input:(uint8_t) u1_id / Motor ID (uint16_t) u2_cnt_value / counter value Output: None	Initialize encoder timer counter value
r_mtr_ctrl_rx24t.c / r_mtr_ctrl_rx24u.c	mtr_speed_calc_timer_start Input:(uint8_t) u1_id / Motor ID Output: None	Start for encoder timer
	mtr_irq_interrupt_enable Input:(uint8_t) u1_id / Motor ID Output: None	Enable Hall interrupt
r_mtr_foc_control.c	mtr_set_pos_ref Input:(mtr_foc_control_t *) st_foc / FOC motor structure Output:(float32) f4_ref_pos_rad_calc / position command value	Setting the command value for position control
	mtr_set_speed_ref Input:(mtr_foc_control_t *) st_foc / FOC motor structure Output:(float32) f4_speed_ref_rad_calc / speed command value	Setting the command value for speed control
	mtr_set_iq_ref Input:(mtr_foc_control_t *) st_foc / FOC motor structure Output:(float32) f4_iq_ref_calc / q-axis current command value	Setting the q axis current command value
	mtr_set_id_ref Input:(mtr_foc_control_t *) st_foc / FOC motor structure Output:(float32) f4_id_ref_calc / d-axis current command value	Setting the d axis current command value

3.3 Macro Definitions of Vector Control Software Using Encoder

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

Table 3-9 List of Macro Definitions 'r_mtr_motor_parameter.h'

File name	Macro name	Definition value	Remarks
r_mtr_motor_parameter.h	MP_POLE_PAIRS	7	Number of pole pairs
	MP_MAGNETIC_FLUX	0.006198f	Flux [Wb]
	MP_RESISTANCE	0.453f	Resistance [Ω]
	MP_D_INDUCTANCE	0.0009447f	d-axis Inductance [H]
	MP_Q_INDUCTANCE	0.0009447f	q-axis Inductance [H]
	MP_ROTOR_INERTIA	0.00000962f	Rotor inertia [kgm ²]
	MP_NOMINAL_CURRENT_RMS	1.8f	Nominal torque [Arms]

Table 3-10 List of Macro Definitions 'r_mtr_control_parameter.h'

File name	Macro name	Definition value	Remarks
r_mtr_control_parameter.h	CP_POS_OMEGA	10f	Natural frequency of the position loop[Hz]
	CP_SPEED_OMEGA	30f	Natural frequency of the speed loop[Hz]
	CP_SPEED_ZETA	1.0f	Damping ratio of the speed loop
	CP_CURRENT_OMEGA	300f	Natural frequency of the current loop[Hz]
	CP_CURRENT_ZETA	1.0f	Damping ratio of the current loop
	CP_SOB_OMEGA	200f	Natural frequency of the speed observer[Hz]
	CP_SOB_ZETA	1.0f	Damping ratio of the speed observer
	CP_MIN_SPEED_RPM	0	Minimum speed (mechanical) [rpm]
	CP_MAX_SPEED_RPM	2000	Maximum speed (mechanical) [rpm]
	CP_SPEED_LIMIT_RPM	3000	Limit speed (mechanical) [rpm]
	CP_REF_ID	1.5f	d-axis current command value [A]

Table 3-11 List of Macro Definitions 'r_mtr_inverter_parameter.h'

File name	Macro name	Definition value	Remarks
r_mtr_inverter_parameter.h	IP_DEADTIME	2.0f	Deadtime [us]
	IP_CURRENT_RANGE	20.0f	current sensing range (-10[A] ~ 10[A])
	IP_VDC_RANGE	111.0f	voltage sensing range (0[V] ~ 111[V])
	IP_INPUT_V	24.0f	input DC voltage [V]
	IP_CURRENT_LIMIT	5.0f	Current limit[A] ^(Note)
	IP_OVERVOLTAGE_LIMIT	28.0f	Over voltage limit [V]
	IP_UNDERVOLTAGE_LIMIT	14.0f	Under voltage limit [V]

Note: This value is calculated from the rated power of the shunt resistance

Table 3-12 List of Macro Definitions 'r_mtr_config.h'

File name	Macro name	Definition value	Remarks
r_mtr_config.h	IP_MRSSK	-	Inverter select macro
	RX24T_MRSSK / RX24U_MRSSK	-	MCU select macro
	MP_FH6S20EX81	-	Motor select macro
	CP_FH6S20EX81	-	
	CONFIG_DEFAULT_UI	ICS_UI	Select default UI ICS_UI: Use the Analyzer for RMW BOARD_UI: Use board interface
	USE_VOLT_ERR_COMP	1	Voltage error compensation 0: Disable 1: Enable
	ANGLE_ADJUST_MODE	MTR_ANGLE_ADJ_EXCIT	Select angle adjust mode MTR_ANGLE_ADJ_EXCIT: Forced excitation mode MTR_ANGLE_ADJ_HALL: Hall mode
	POS_CTRL_MODE	MTR_CTRL_IPD	Select position control mode MTR_CTRL_PID: PID controller MTR_CTRL_IPD: IPD controller
	LOOP_MODE	MTR_LOOP_POSITION	Select control loop mode MTR_LOOP_SPEED: speed loop MTR_LOOP_POSITION: position loop
	GAIN_MODE	MTR_GAIN_DESIGN_MODE	Gain mode MTR_GAIN_DESIGN_MODE: PI gain design mode MTR_GAIN_DIRECT_MODE: PI gain direct input mode
	MOD_METHOD	MOD_METHOD_SVPWM	modulation method MOD_METHOD_SPWM: Sinusoidal PWM MOD_METHOD_SVPWM: Space Vector PWM

3.4 Control Flowcharts

3.4.1 Main Process

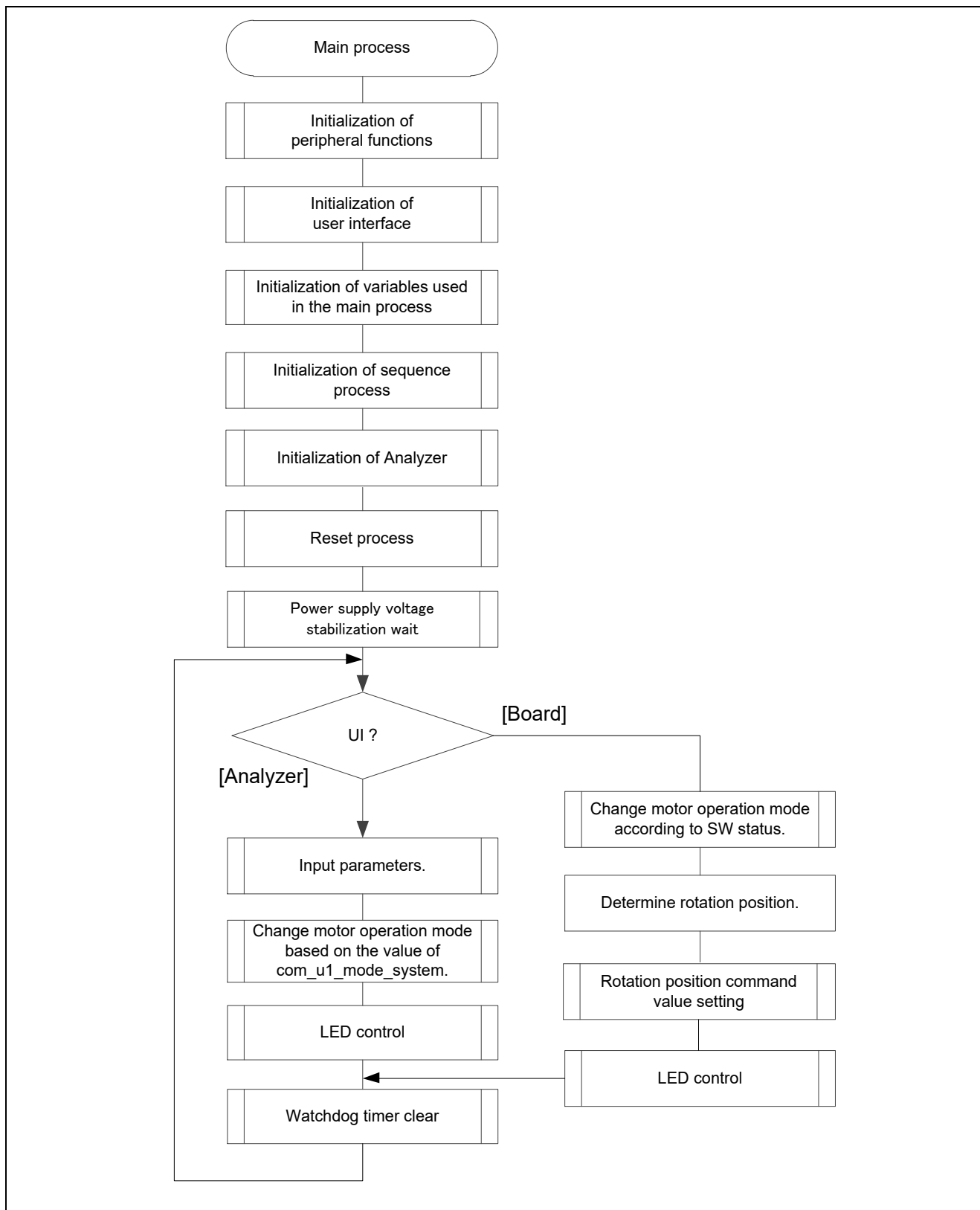


Figure 3-9 Main Process Flowchart

3.4.2 Carrier Synchronous Interrupt Handling (Vector Control using Encoder)

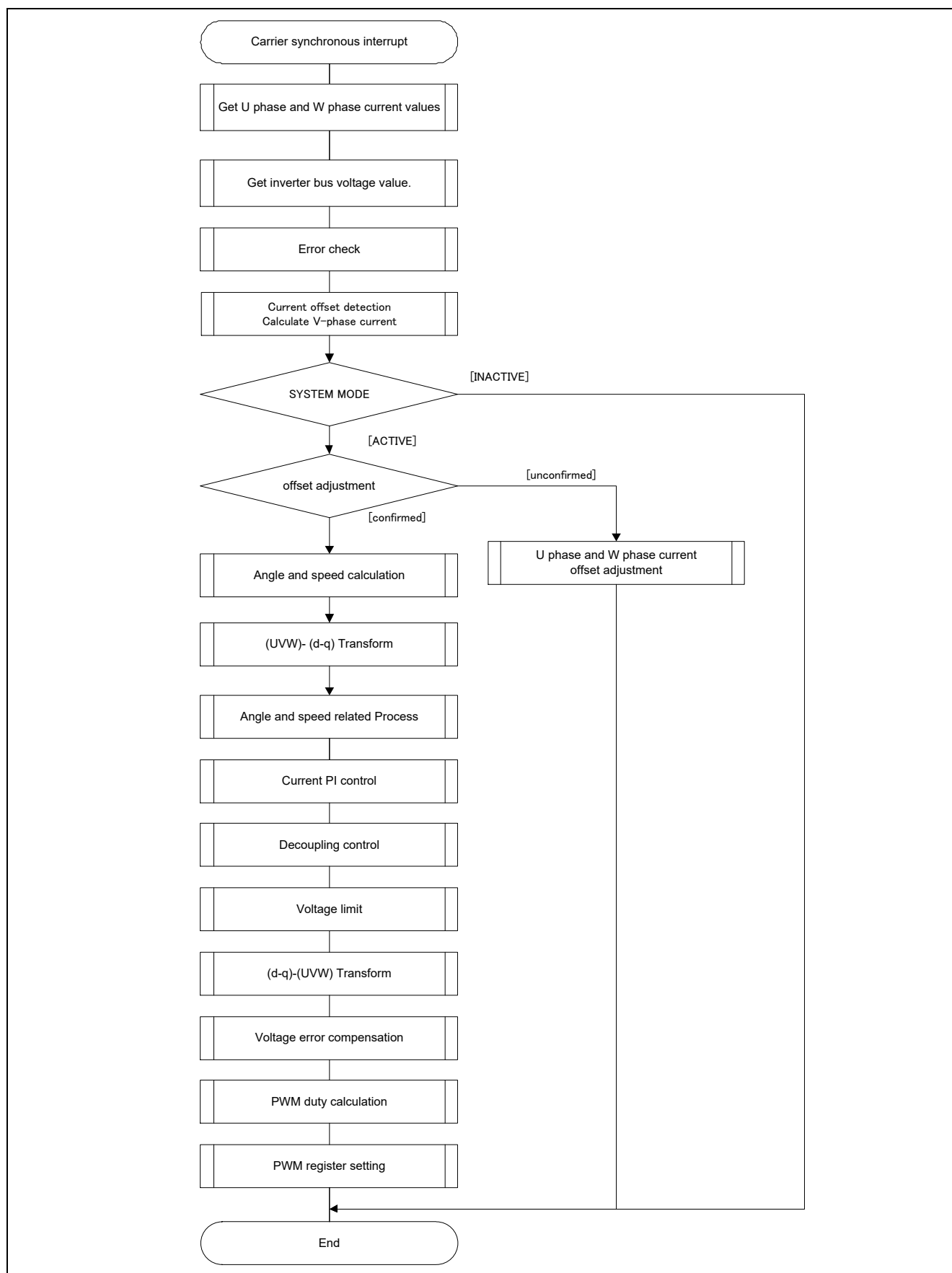


Figure 3-10 100 [μs] Cycle Interrupt Handling

3.4.3 1 [ms] Interrupt Handling

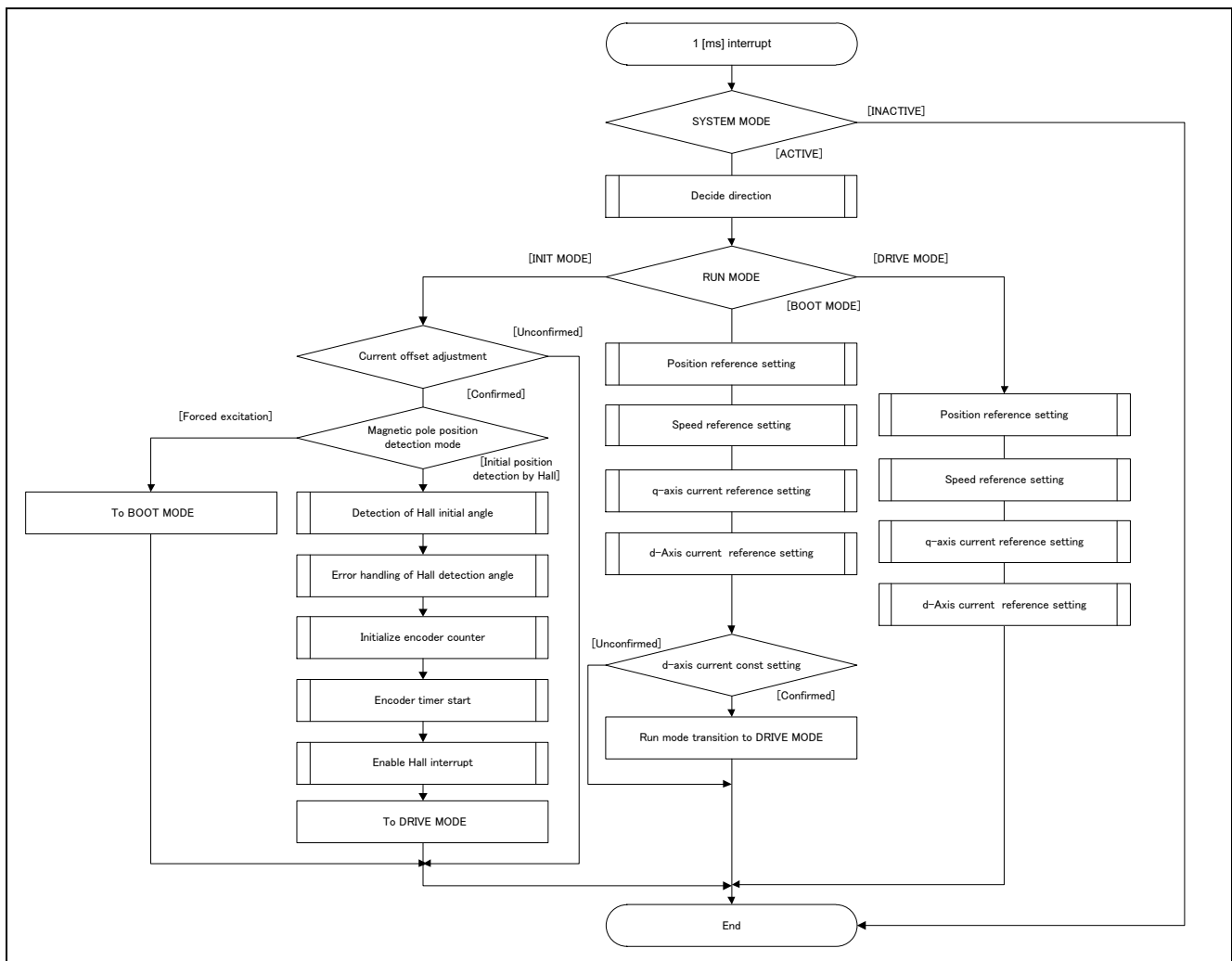


Figure 3-11 1 [ms] Interrupt Handling

3.4.4 Over Current Detection Interrupt Handling

The over current detection interrupt occurs when POE0# pin detects falling-edge or when output levels of the MTU complementary PWM output pins are compared and simultaneous active-level output continues for one cycle or more. Therefore when this interrupt process is executed, PWM output pins are already in high-impedance state and the output to the motor is stopped.

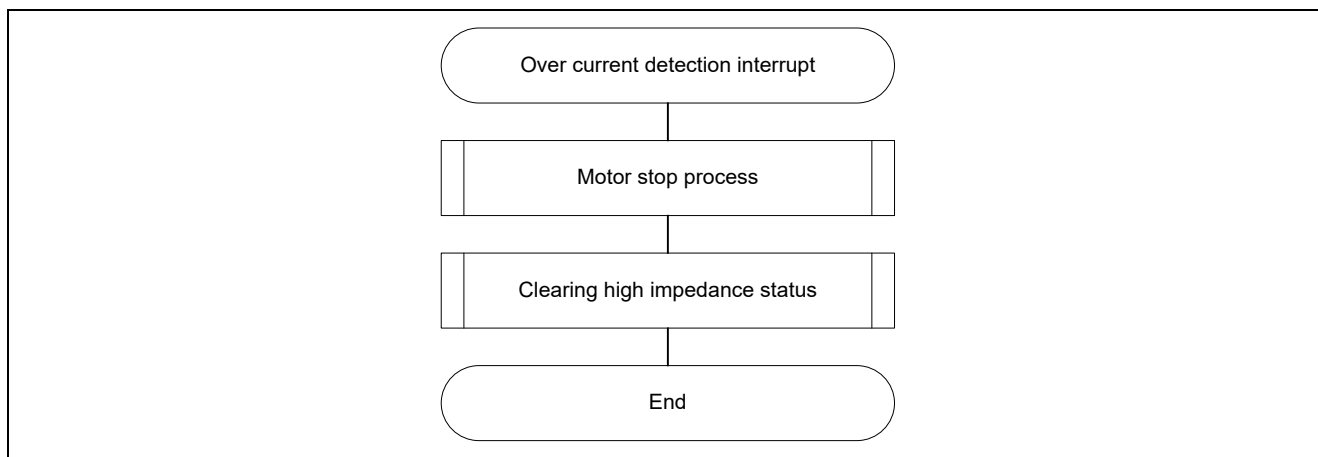


Figure 3-12 Over Current Detection Interrupt Handling

3.4.5 Encoder Count Capture Interrupt Handling

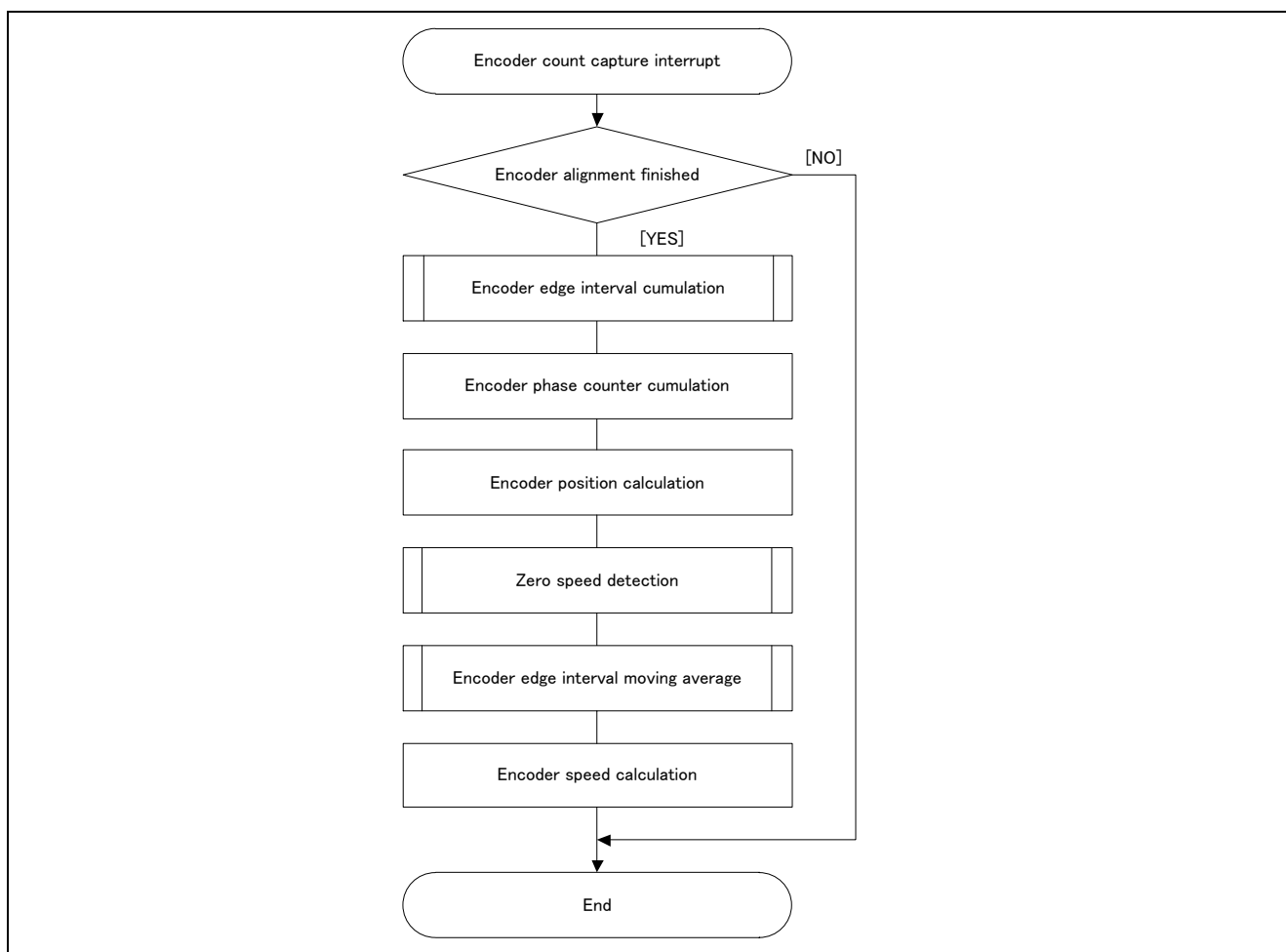


Figure 3-13 Encoder Count Capture Interrupt Handling

3.4.1 Hall Signal Interrupt Handling

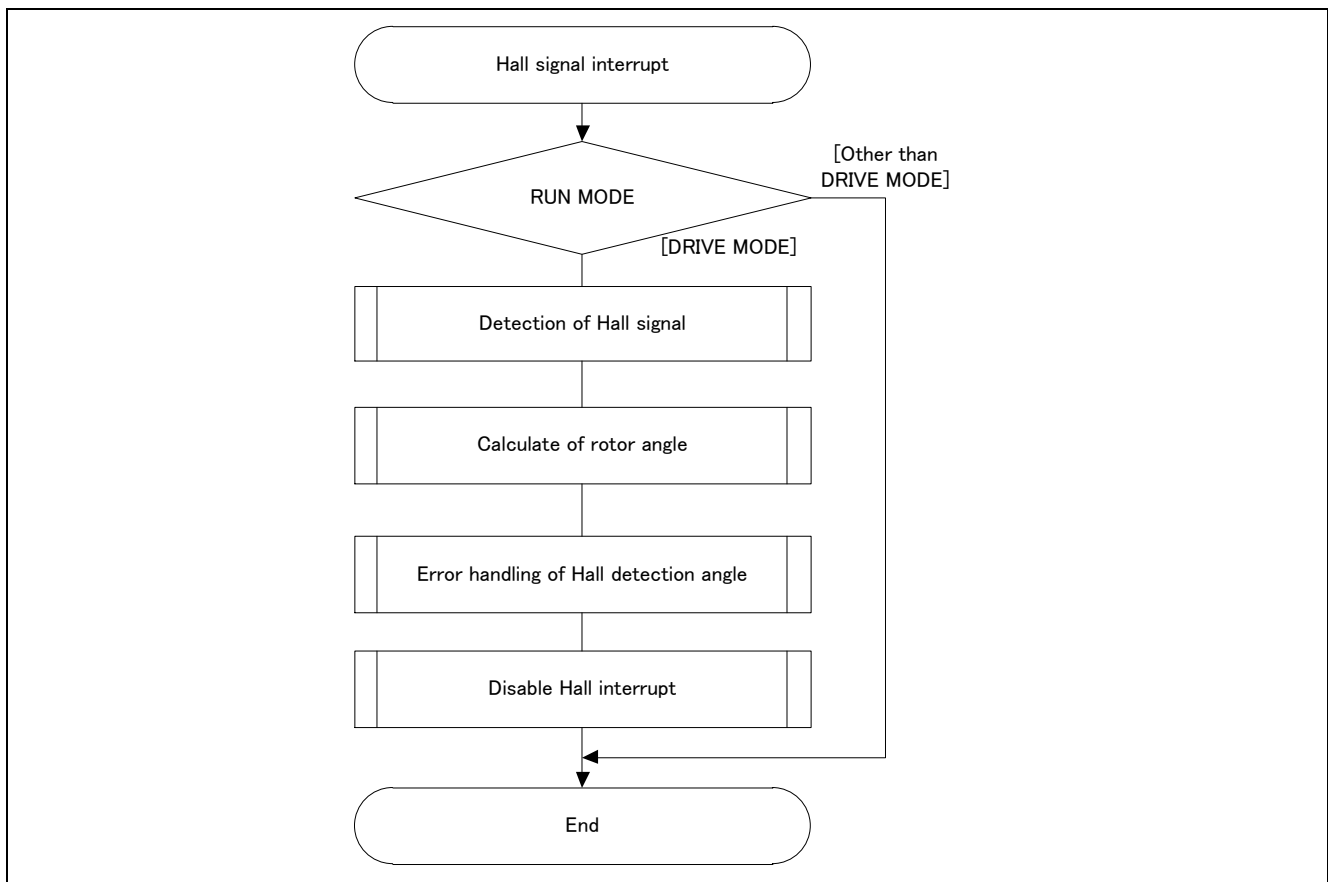


Figure 3-14 Hall Signal Interrupt Handling

4. Motor Control Development Support Tool ‘Renesas Motor Workbench’

4.1 Overview

‘Renesas Motor Workbench’ is support tool for development of motor control system. ‘Renesas Motor Workbench’ can be used with target software of this application note to analyze the control performance. The user interfaces of ‘Renesas Motor Workbench’ provide functions like rotating/stop command, setting rotation speed reference, etc. Please refer to ‘Renesas Motor Workbench V.1.00 User’s Manual’ for usage and more details. ‘Renesas Motor Workbench’ can be downloaded from Renesas Electronics Corporation website.

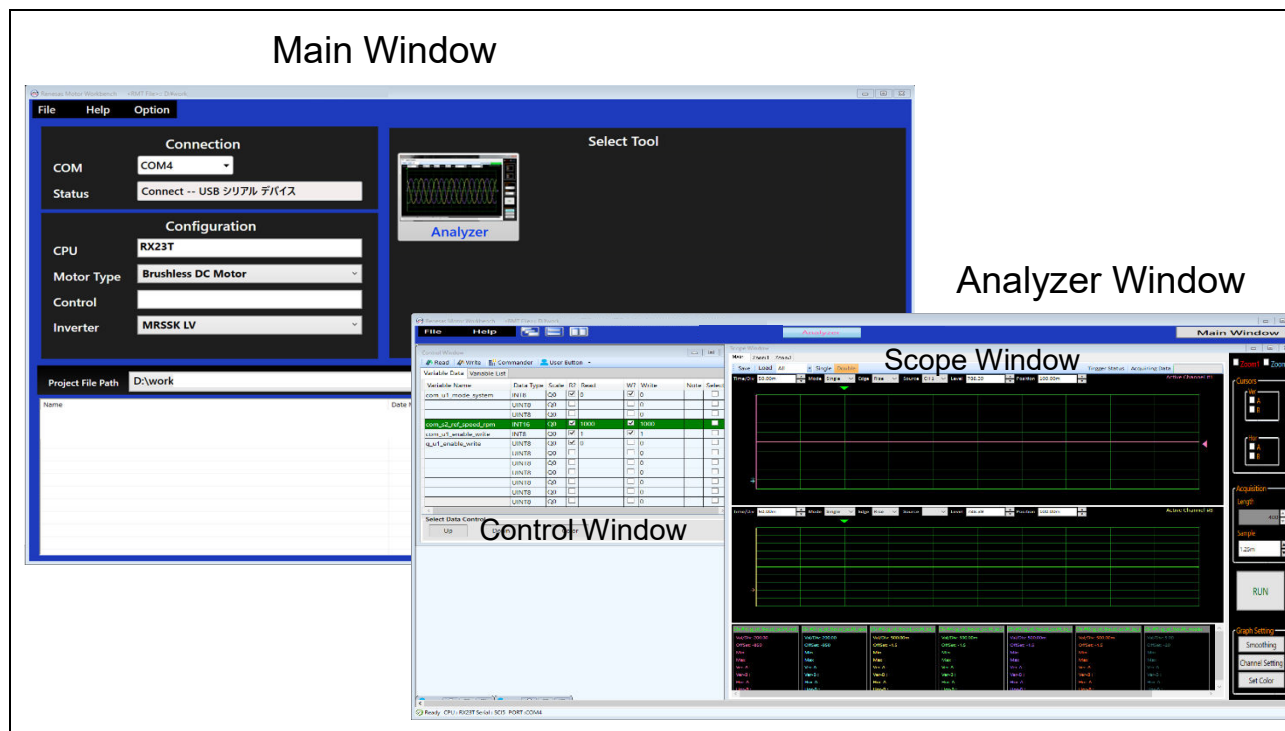
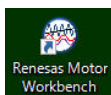


Figure 4-1 Renesas Motor Workbench – Appearance

Set up for ‘Renesas Motor Workbench’



- (1) Start ‘Renesas Motor Workbench’ by clicking this icon.
- (2) Click on [File] and select [Open RMT File(O)] from drop down Menu.
Select the RMT file from following location of e2studio/CS+ project folder.
‘[Project Folder]/ application/user_interface/ics/’
- (3) Use the ‘Connection’ [COM] select menu to choose the COM port.
- (4) Click on the ‘Analyzer’ icon of Select Tool panel to open Analyzer function window.
- (5) Please refer to ‘4.3Operation Example for Analyzer’ for motor driving operation.

4.2 List of Variables for Scope Function 'Analyzer'

Table 4-1 is a list of variables for Analyzer. These variable values are reflected to the protect variables when the same values as of `g_u1_enable_write` are written to `com_u1_enable_write`. However, note that variables with (*) do not depend on `com_u1_enable_write`.

Table 4-1 List of Variables for Analyzer

Variable name	Type	Content
<code>com_u1_sw_userif (*)</code>	<code>uint8_t</code>	User interface switch 0: ICS user interface use (default) 1: Board user interface use
<code>com_u1_mode_system (*)</code>	<code>uint8_t</code>	State management 0: Stop mode 1: Run mode 3: Reset
<code>com_u1_direction</code>	<code>uint8_t</code>	Rotation direction 0: CW 1: CCW
<code>com_u1_ctrl_loop_mode</code>	<code>uint8_t</code>	Control loop mode switch 0: Speed control 1: Position control (default)
<code>com_u1_ctrl_method_mode</code>	<code>uint8_t</code>	Control method switch 0: PID control (Position P/Speed PI/Current PI) 1: IPD control(position + Speed IPD +Position FF+ Speed FF+Position P/ Current PI) (default) FF:Feed-forward control
<code>com_u1_position_input_mode</code>	<code>uint8_t</code>	Position reference input mode switch 0:0 output 1:direct input 2:Position profiling (default)
<code>com_u1_encd_angle_adj_mode</code>	<code>uint8_t</code>	Angle detection mode switch 0: Forced excitation(default) 1: Position detection using Hall signal
<code>com_s2_ref_position_deg</code>	<code>int16_t</code>	Position command value [degree]
<code>com_s2_ref_speed_rpm</code>	<code>int16_t</code>	Speed command value [rpm]
<code>com_u2_min_speed_rpm</code>	<code>uint16_t</code>	Minimum speed [[rpm]
<code>com_u2_max_speed_rpm</code>	<code>uint16_t</code>	Maximum speed [rpm]
<code>com_u2_overspeed_limit_rpm</code>	<code>uint16_t</code>	Overspeed Limit [rpm]
<code>com_u2_hs_change_speed_rpm</code>	<code>uint16_t</code>	Speed calculation mode switch speed [rpm]
<code>com_u2_hs_change_margin_rpm</code>	<code>uint16_t</code>	Speed calculation mode switch margin speed [rpm]
<code>com_u2_pos_interval_time</code>	<code>uint16_t</code>	Time interval of the position command changes [s]
<code>com_u2_pos_dead_band</code>	<code>uint16_t</code>	Dead band of position
<code>com_u2_pos_band_limit</code>	<code>uint16_t</code>	Positioning complete range
<code>com_u2_encd_cpr_mech</code>	<code>uint16_t</code>	Encoder pulse count (4 for multiplying)
<code>com_u2_offset_calc_time</code>	<code>uint16_t</code>	Current offset value calculation time [ms]
<code>com_u2_mtr_pp</code>	<code>uint16_t</code>	Number of pole pairs
<code>com_f4_mtr_r</code>	<code>float</code>	Resistance [Ω]
<code>com_f4_mtr_ld</code>	<code>float</code>	d-axis Inductance [H]
<code>com_f4_mtr_lq</code>	<code>float</code>	q-axis Inductance [H]
<code>com_f4_mtr_m</code>	<code>float</code>	Flux [Wb]
<code>com_f4_mtr_j</code>	<code>float</code>	Inertia [kgm^2]
<code>com_f4_nominal_current_rms</code>	<code>float</code>	Nominal current [Arms]

Table 4-2 List of Variables for Analyzer

Variable name	Type	Content
com_f4_current_omega	float	Natural frequency of the current loop[Hz]
com_f4_current_zeta	float	Damping ratio of the current loop
com_f4_speed_omega	float	Natural frequency of the speed loop[Hz]
com_f4_speed_zeta	float	Damping ratio of the speed loop
com_f4_pos_omega	float	Natural frequency of the position loop[Hz]
com_f4_sob_omega	float	Natural frequency of the speed observer [Hz]
com_f4_sob_zeta	float	Damping ratio of the speed observer
com_f4_id_kp	float	d axis current PI control proportional term gain
com_f4_id_ki	float	d axis current PI control integral term gain
com_f4_iq_kp	float	q axis current PI control proportional term gain
com_f4_iq_ki	float	q axis current PI control integral term gain
com_f4_speed_kp	float	Speed PI control proportional term gain
com_f4_speed_ki	float	Speed PI control integral term gain
com_f4_pos_kp	float	Position control proportional term gain
com_f4_ipd_speed_k_ratio	float	Speed control gain ratio for IPD
com_f4_ipd_pos_kp_ratio	float	Position control proportional term gain ratio for IPD
com_f4_ipd_err_limit_1	float	Position error limit for IPD
com_f4_ipd_err_limit_2	float	Position error limit for IPD
com_f4_accel_time	float	Acceleration time [s] (for position control)
com_f4_id_ref_open	float	d-axis current command value [A]
com_f4_id_up_time	float	d-axis current command value addition time [ms]
com_f4_limit_speed_change	float	Acceleration limit [s] (for speed control)
com_u1_enable_write	uint8_t	Enabled to rewriting variables

The primary variables that are frequently observed during the motor driving evaluation are listed in Table 4-3 Please refer when using Analyzer function. Regarding variables not listed in Table 4-3, refer to source codes.

Table 4-3 List of Primary Variables for Encoder Vector Control

Name of primary variable for Encoder Vector Control	Type	Content
st_foc.u2_error_status	uint16_t	error status
st_foc.st_cc.f4_id_ref	float	d-axis current command value [A]
st_foc.st_cc.f4_id_ad	float	d-axis current [A]
st_foc.st_cc.f4_iq_ref	float	q-axis current command value [A]
st_foc.st_cc.f4_iq_ad	float	q-axis current [A]
st_foc.f4_iu_ad	float	W phase current A/D conversion value [A]
st_foc.f4_iv_ad	float	V phase current A/D conversion value [A]
st_foc.f4_iw_ad	float	W phase current A/D conversion value [A]
st_foc.st_cc.f4_vd_ref	float	d-axis output voltage command value [V]
st_foc.st_cc.f4_vq_ref	float	q-axis output voltage command value [V]
st_foc.f4_refu	float	U phase voltage command value [V]
st_foc.f4_refv	float	V phase voltage command value [V]
st_foc.f4_refw	float	W phase voltage command value [V]
st_foc.st_sc.f4_ref_speed_rad_ctrl	float	Command value for speed PI control (Electrica) [rad/s]
st_foc.st_sc.f4_speed_rad	float	Speed (Electrical) [rad/s]
st_foc.st_pc.f4_ref_pos_rad_ctrl	float	Command value for Position control (Electrical) [rad]
st_foc.st_pc.f4_pos_rad	float	Position (Electrical) [rad]

4.3 Operation Example for Analyzer

This section shows an example below for motor driving operation using Analyzer. Operation is using 'Control Window' of analyzer. Regarding specification of 'Control Window', refer to 'Renesas Motor Workbench V.1.00 User's Manual'.

- Driving the motor

- ① Confirm the check-boxes of column [W?] for 'com_u1_mode_system', 'com_s2_ref_speed_rpm', 'com_u1_enable_write'
- ② Input a reference speed value in the [Write] box of 'com_s2_ref_speed_rpm'.
- ③ Click the 'Write' button.
- ④ Click the 'Read' button. Confirm the [Read] box of 'com_s2_ref_speed_rpm', 'g_u1_enable_write'.
- ⑤ Set a same value of 'g_u1_enable_write' in the [Write] box of 'com_u1_enable_write'.
- ⑥ Write '1' in the [Write] box of 'com_u1_mode_system'.
- ⑦ Click the 'Write' button.

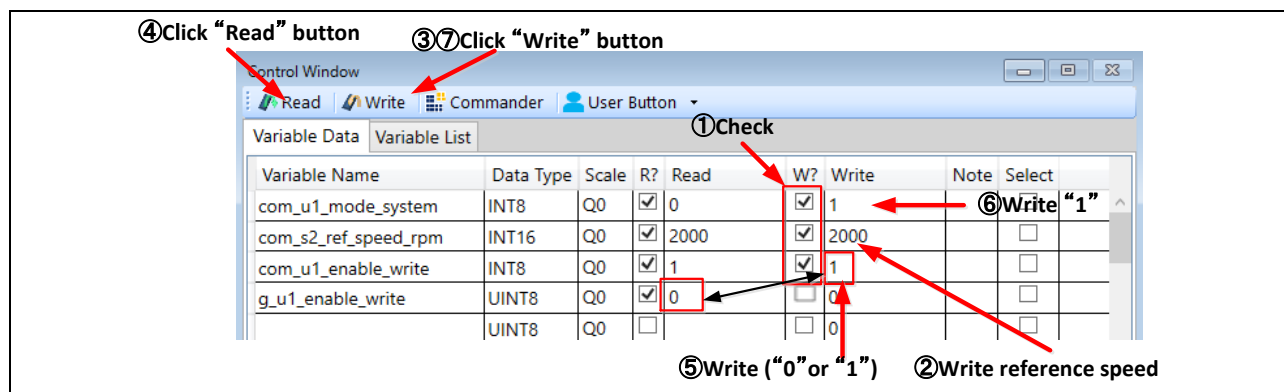


Figure 4-2 Procedure - Driving the Motor

- Stop the motor

- ① Write '0' in the [Write] box of 'com_u1_mode_system'
- ② Click the 'Write' button.

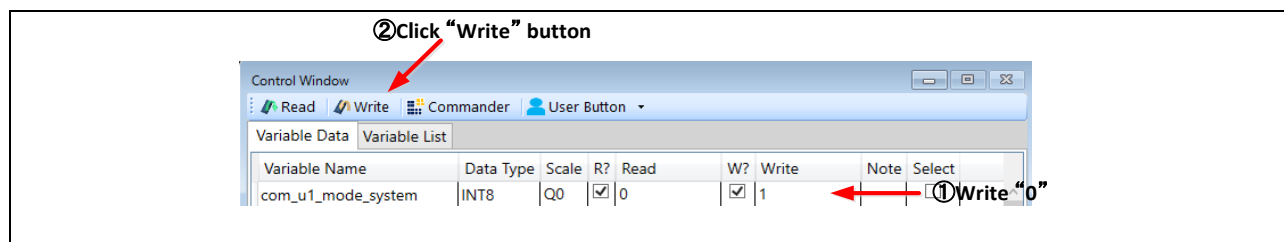


Figure 4-3 Procedure - Stop the Motor

- Error cancel operation

- ① Write '3' in the [Write] box of 'com_u1_mode_system'
- ② Click the 'Write' button.

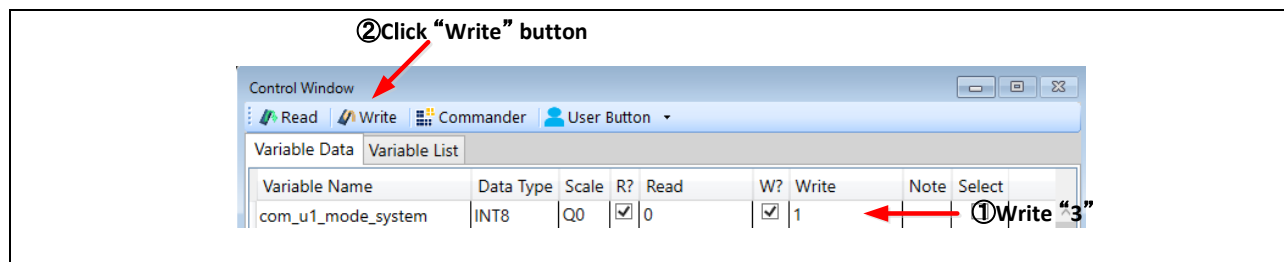


Figure 4-4 Procedure - Error Cancel Operation

Website and Support

Renesas Electronics Website

<http://www.renesas.com/>

Inquiries

<http://www.renesas.com/contact/>

All trademarks and registered trademarks are the property of their respective owners.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Apr.05. 2017	-	First edition issued
1.01	July 07. 2017	-	Update for software version 1.01 Fixed typo error in document

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

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

1. 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 Microprocessing unit or Microcontroller unit products 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.