RA6T1

Vector control for permanent magnetic synchronous motor with encoder - AB incremental type encoder

Abstract

This application note describes the sample program for a permanent magnetic synchronous motor drive with vector control using encoder based on Renesas RA6T1 microcontroller. This application note also describes how to use the motor control development support tool, 'Renesas Motor Workbench'.

The targeted software for this application is only to be used as reference purposes only and Renesas Electronics Corporation does not guarantee the operations. Please use this 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.

- RA6T1 (R7FA6T1AD3CFP)

Contents

1. Overview

This application note explains how to implement the vector control with encoder software that drives permanent magnetic synchronous motor (PMSM) using the RA6T1 microcontroller and how to use the motor control development support tool, 'Renesas Motor Workbench'.

Note that this software uses the algorithm described in the application note 'Vector control with encoder for permanent magnet synchronous motor (Algorithm)' (R01AN3789), so please refer to that for the details of the algorithm.

1.1 Development environment

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

Table 1-1 Hardware Development Environment

Microcontroller	Evaluation board (Note 1)	Motor (Note 2)	Sensor (Note 3)
RA6T1 (R7FA6T1AD3CFP)	48V inverter board RA6T1 CPU Card	BLY171D-24V-4000	AMT102-V

Table 1-2 Software Development Environment

e^{2}studio version	FSP version	Toolchain version
V2022-10	V4.1.0 and later	GCC ARM Embedded: V10.3.1.20210824

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

Notes:

1. 48V inverter board (RTK0EM0000B10020BJ) is included in the kit products RTK0EMA170S00020BJ and RTK0EMX270S00020BJ, and it is a product of Renesas Electronics Corporation.

   RA6T1 CPU Card (RTK0EMA170C00000BJ) is a product of Renesas Electronics Corporation.

2. BLY171D-24V-4000 is a product of Anaheim Automation.

   Anaheim Automation (https://www.anaheimautomation.com/)

3. AMT102-V is a product of CUI Devices.

   CUI Devices (https://www.cuidevices.com/)

2. System overview

Overview of this system is explained below.

2.1 Hardware configuration

The hardware configuration is shown below.

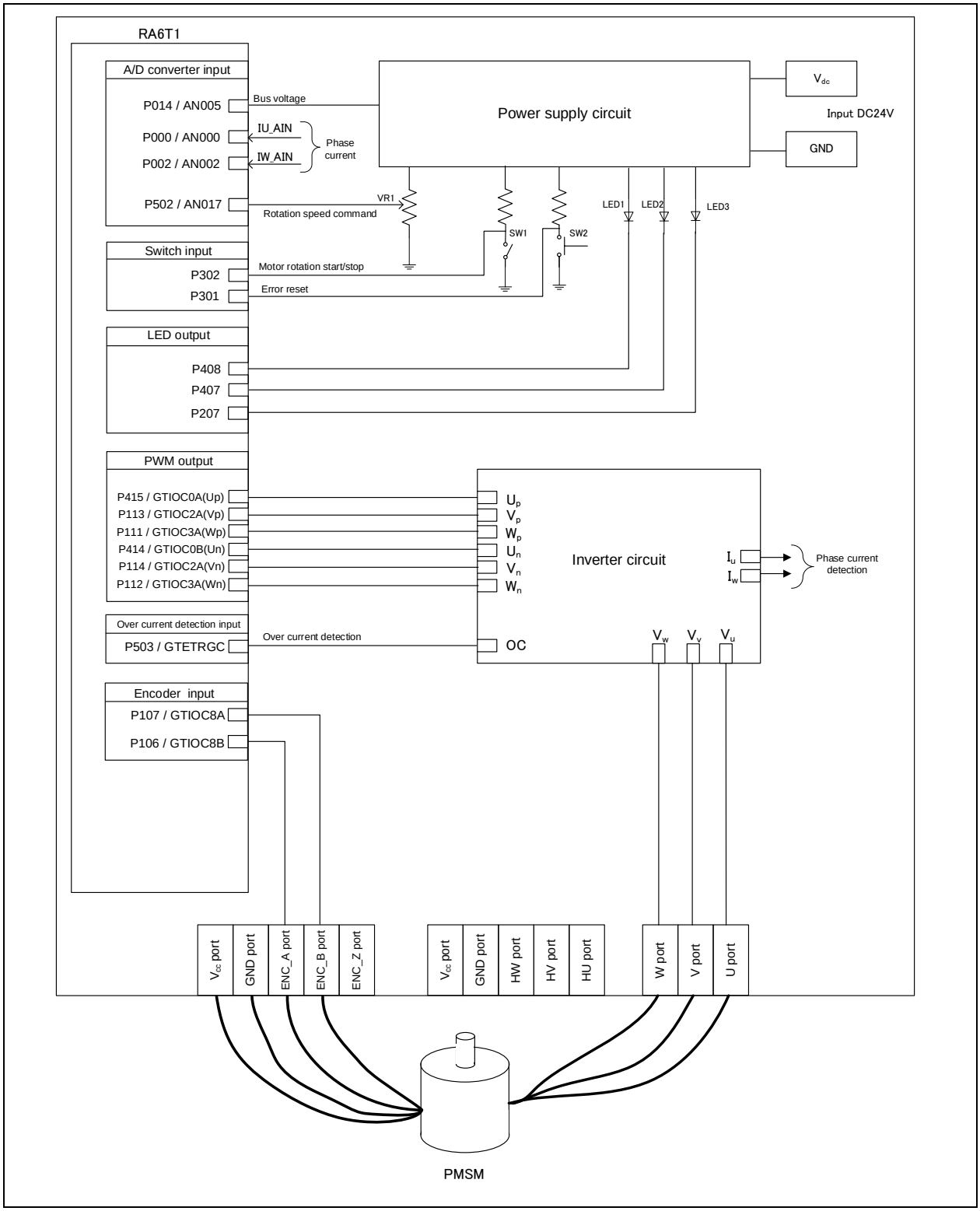


Figure 2-1 Hardware Configuration Diagram

2.2 Hardware specifications

2.2.1 User interface

Table 2-1 and Table 2-2 is lists of user interfaces of this system.

Table 2-1 User Interface

Item	Interface component	Function
Rotation speed command	Variable resistor (VR)	Reference value of rotation speed input (analog value)
START/STOP	Toggle switch (SW1)	Motor rotation start/stop command
ERROR RESET	Push switch (SW2)	Command of recovery from error status
LED1	Orange LED	- At the time of motor rotation: ON - At the time of stop: OFF
LED2	Orange LED	- At the time of error detection: ON - At the time of normal operation: OFF
LED2	Orange LED	- Positioning completed: ON - Positioning not completed: OFF

Table 2-2 CPU card user interface

Item	Interface component	Function
LED1	Green LED	- At the time of motor rotation: ON - At the time of stop: OFF
LED2	Green LED	- At the time of error detection: ON - At the time of normal operation: OFF
RESET	Push switch (RESET)	System reset

List of port interfaces of this system is given in Table 2-3.

Table 2-3 Port Interfaces

R7FA6T1AD3CFP port name	Function
P014 / AN005	Inverter bus voltage measurement
P502 / AN017	For inputting rotation speed/position command values (analog values)
P302	START/STOP toggle switch
P301	ERROR RESET push switch
P408	LED1 ON/OFF control
P407	LED2 ON/OFF control
P207	LED3 ON/OFF control
P000 / AN000	U phase current measurement
P001 / AN001	V phase current measurement
P002 / AN002	W phase current measurement
P415 / GTIOC0A	PWM output (Up) / Low Active
P113 / GTIOC2A	PWM output (Vp) / Low Active
P111 / GTIOC3A	PWM output (Wp) / Low Active
P414 / GTIOC0B	PWM output (Un) / High Active
P114 / GTIOC2B	PWM output (Vn) / High Active
P112 / GTIOC3B	PWM output (Wn) / High Active
P503 / GTETRGC	PWM emergency stop input at the time of overcurrent detection
P107 / GTIOC8A	Encoder A Phase input
P106 / GTIOC8B	Encoder B Phase input

2.2.2 Peripheral functions

List of the peripheral functions used in this system is given in Table 2-4.

Table 2-4 List of the Peripheral Functions

Peripheral	Resource	Purpose
12-bit A/D Converter	AN000, AN001, AN002, AN005, AN017	• Measure electric current of phase U,V and W • Measure inverter bus voltage • Rotational speed or position command input
AGT	AGT0	500 [µs] interval timer
GPT	CH0, CH1, CH2, CH3, CH8	• Complementary PWM outputs • Encoder phase counter • Encoder count capture
POEG	Group C	Sets ports executing PWM output to high impedance state when an overcurrent is detected by external circuit.

(1) 12-bit A/D Converter (12ADC)

U-phase current (Iu), V-phase current (Iv), W-phase current (Iw), inverter bus voltage1 (Vdc), and speed command (VR) are measured in "Single scan mode" (use a hardware trigger).

A/D conversion is implemented to be synchronized with carrier synchronized interrupt as GPT underflow (PWM valley).

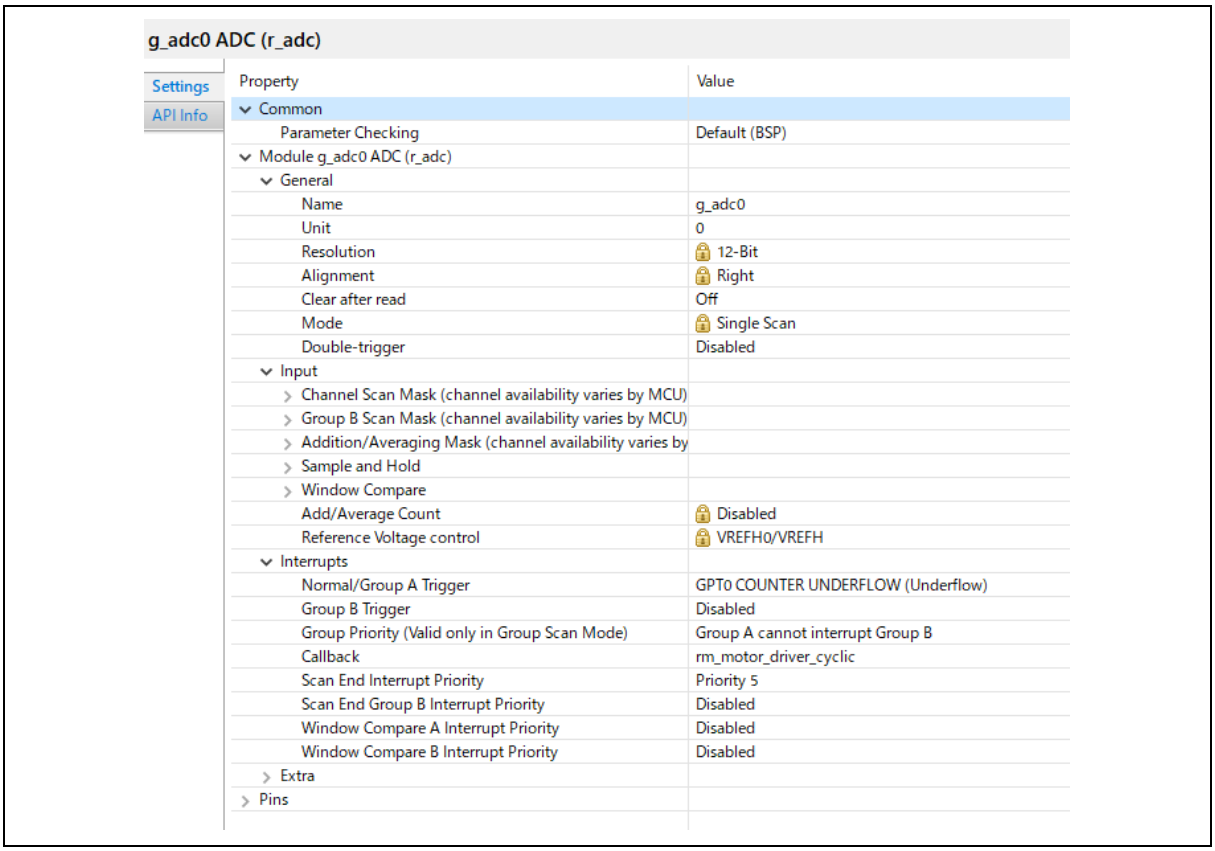


Figure 2-2 FSP Configuration of ADC Driver (FSP 3.5.0)

(2) Low Power Asynchronous General-Purpose Timer (AGT)

The AGT is used as 500 [µs] interval timer.

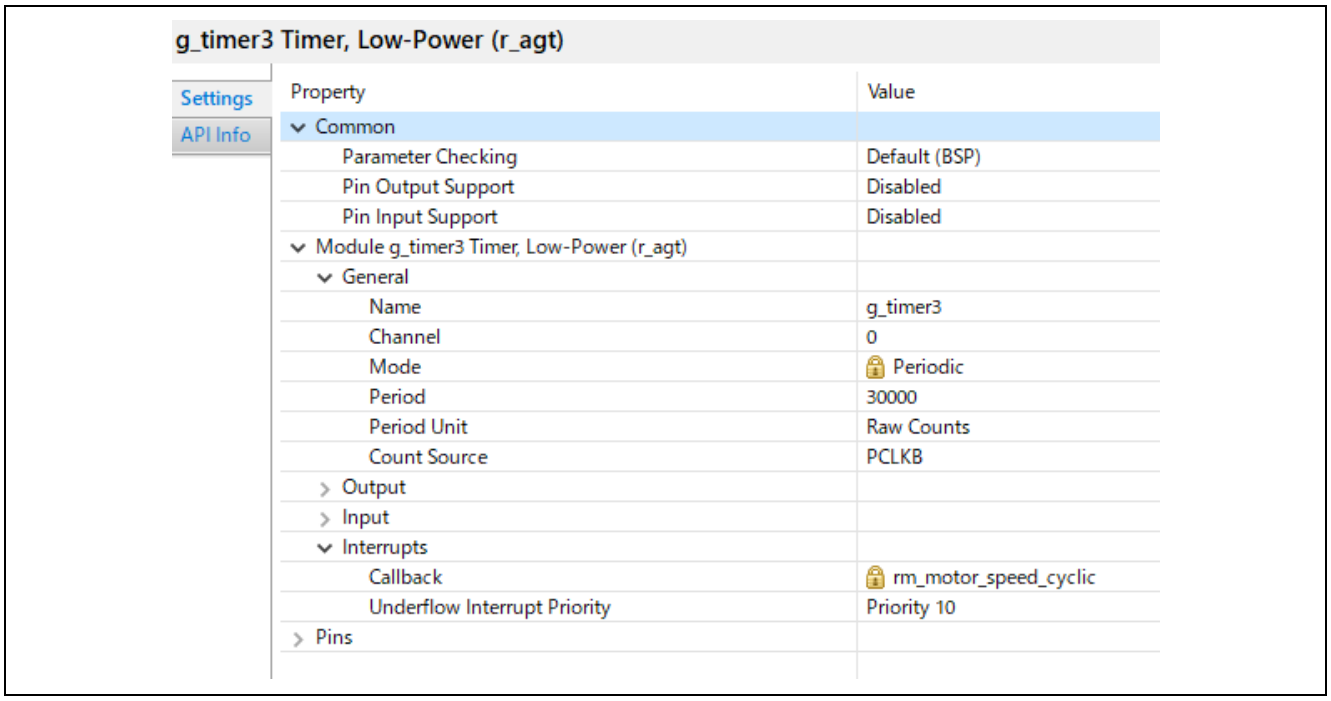


Figure 2-3 FSP Configuration of AGT Driver (FSP 3.5.0)

(3) General PWM Timer (GPT)

On the channel 0, 2, and 3, output with dead time is performed by using the complementary PWM Output Operating Mode.

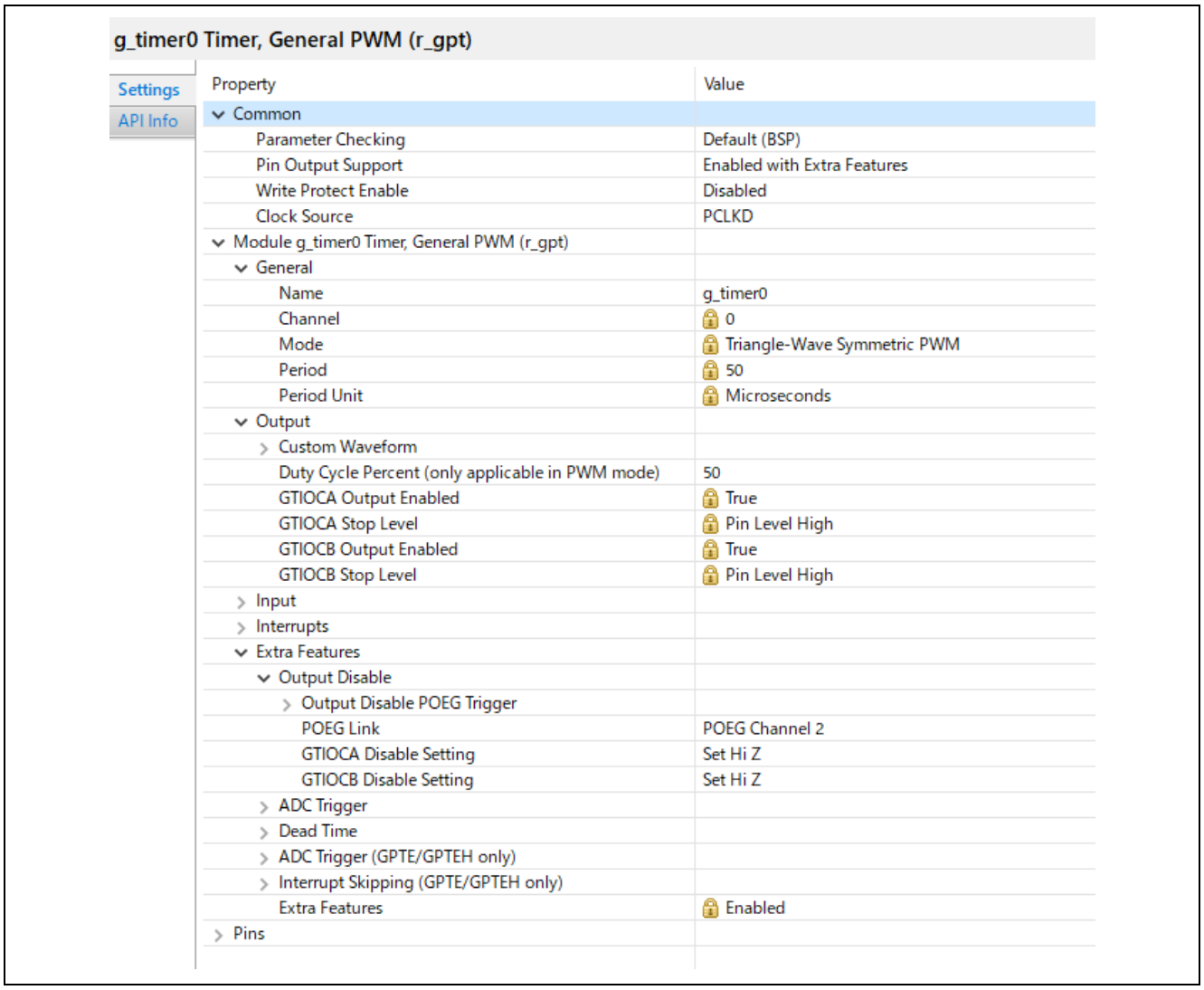


Figure 2-4 FSP Configuration of GPT Driver for PWM output (FSP 3.5.0)

(4) Phase Counting Timer (GPT)

Channel 8 is used as phase counting mode2 to measure the phase difference between encoder phase A and B signal and counts the encoder input.

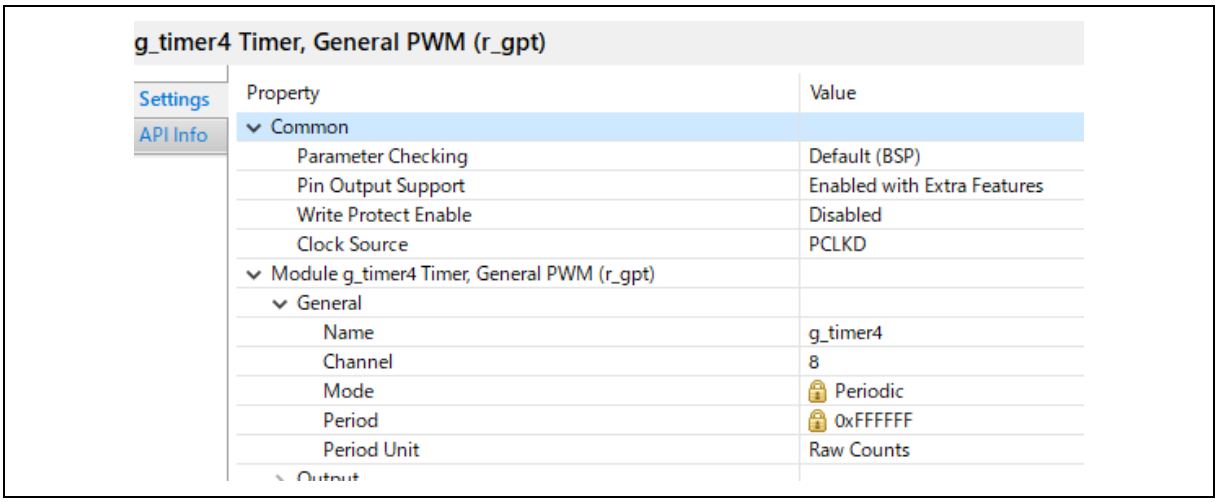


Figure 2-5 FSP Configuration of GPT Driver for phase counting timer (FSP 3.5.0) [1/5]

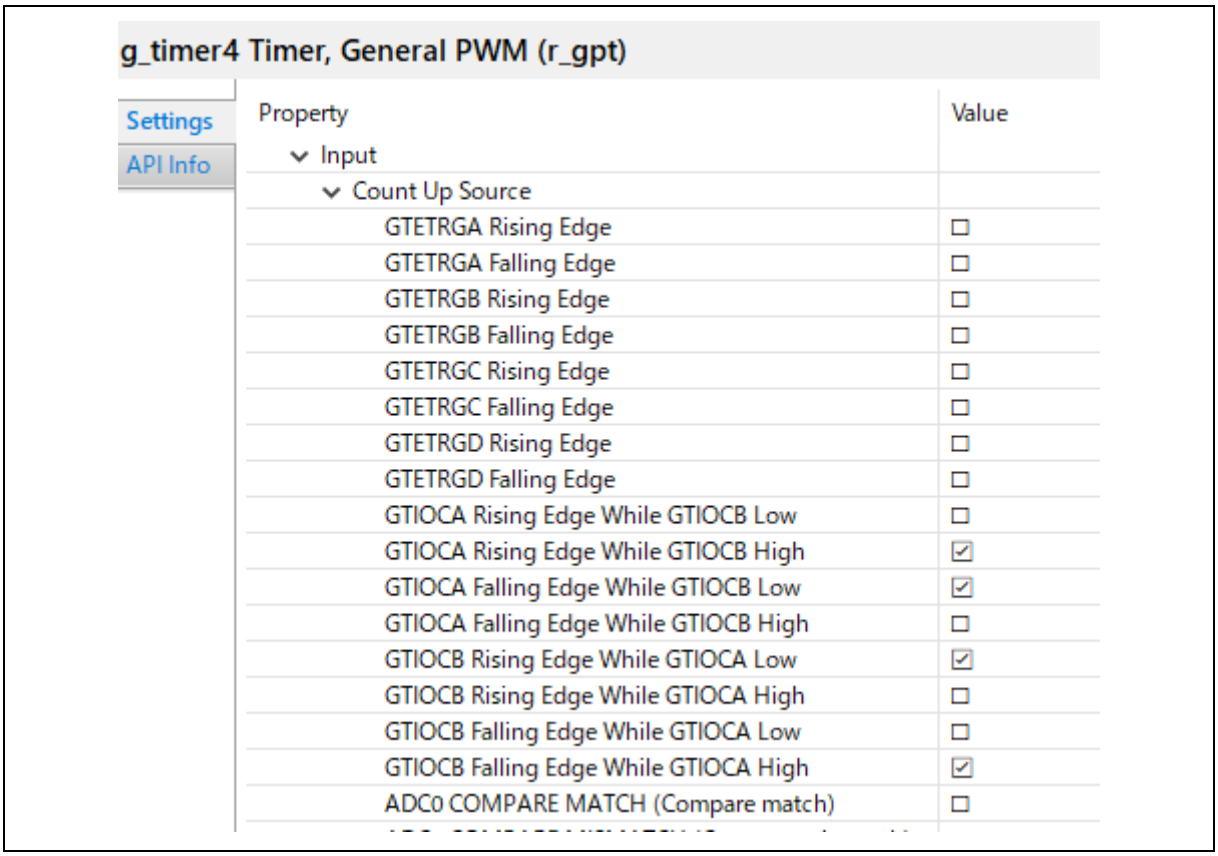


Figure 2-6 FSP Configuration of GPT Driver for phase counting timer (FSP 3.5.0) [2/5]

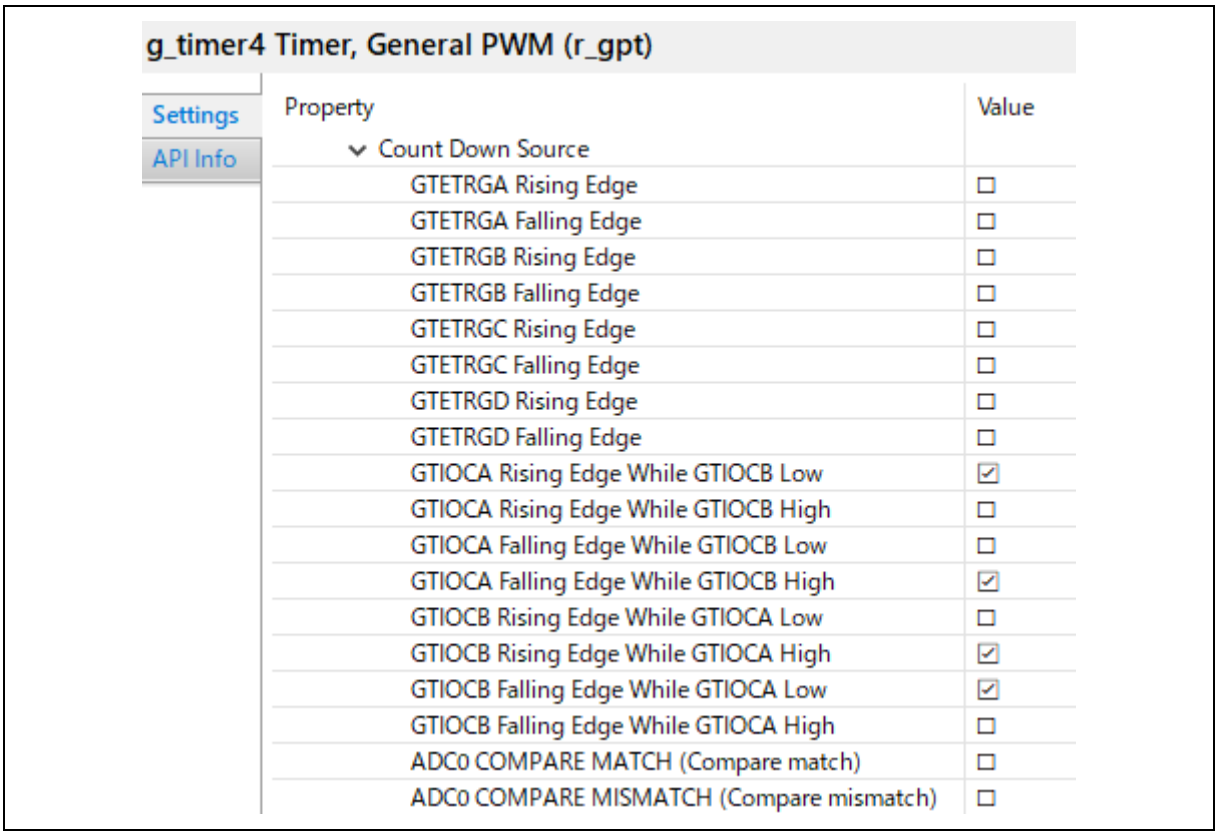


Figure 2-7 FSP Configuration of GPT Driver for phase counting timer (FSP 3.5.0) [3/5]

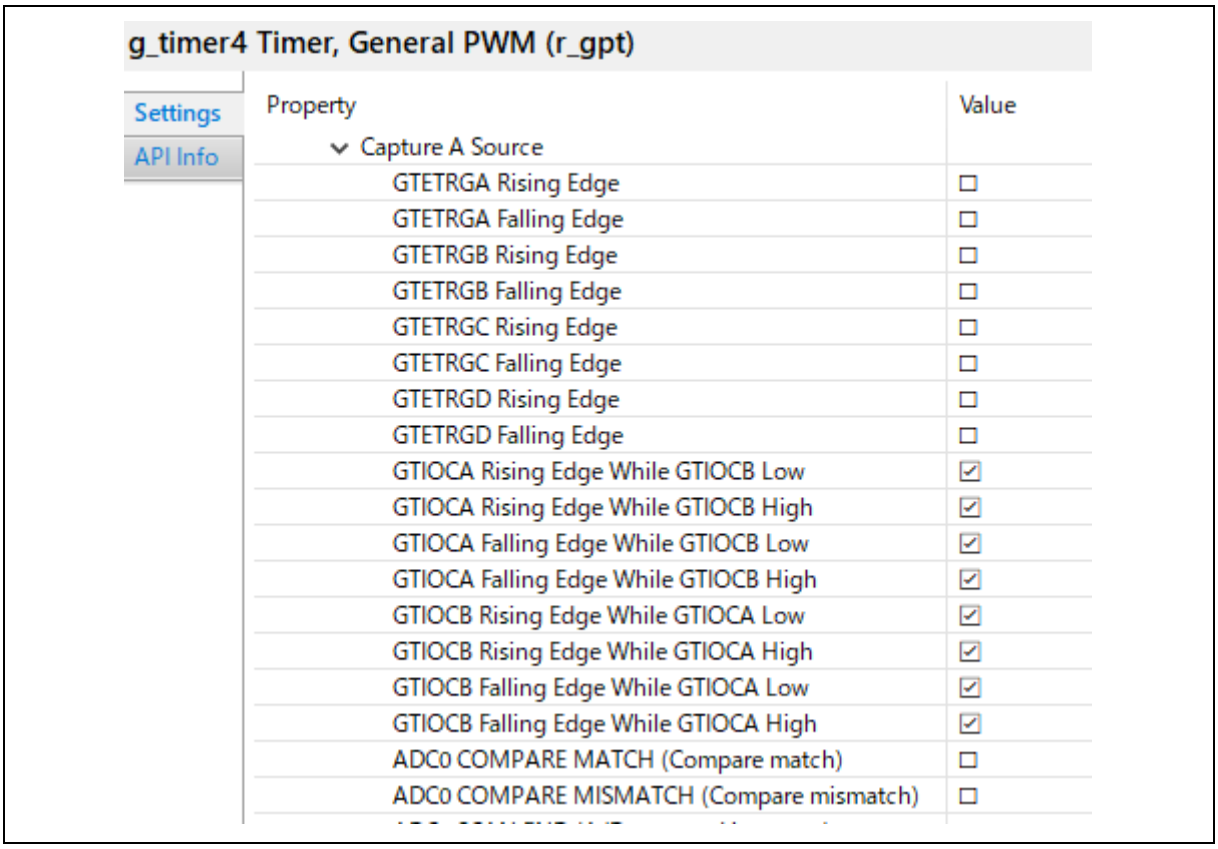


Figure 2-8 FSP Configuration of GPT Driver for phase counting timer (FSP 3.5.0) [4/5]

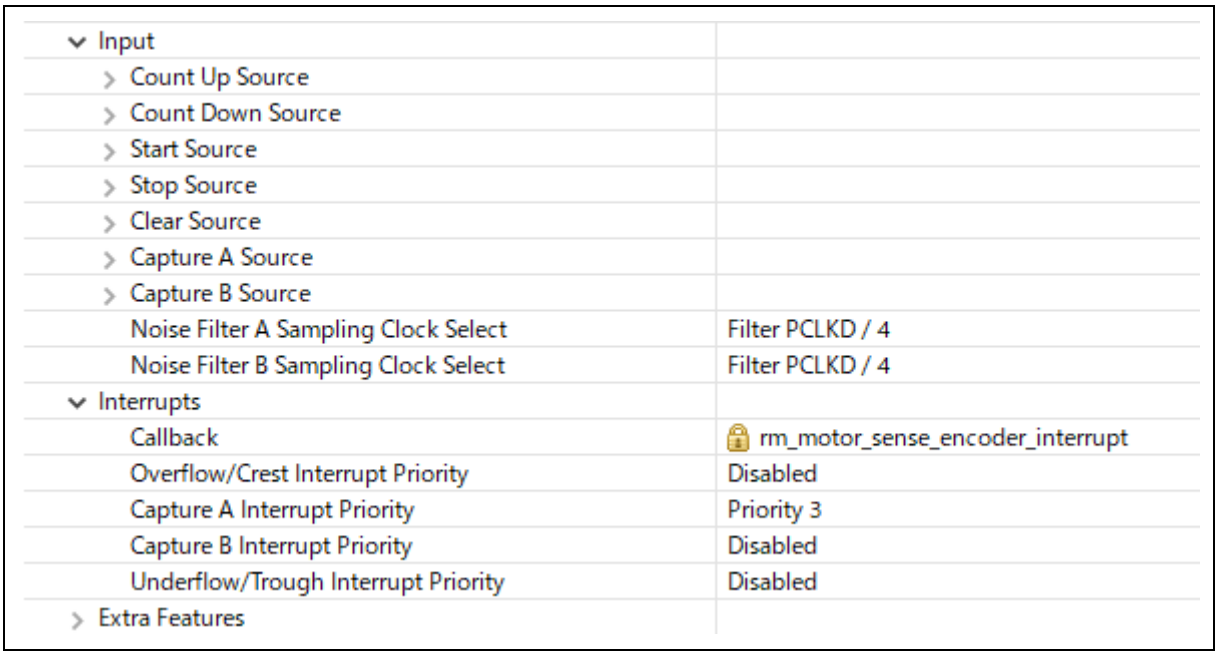


Figure 2-9 FSP Configuration of GPT Driver for phase counting timer (FSP 3.5.0) [5/5]

(5) Free run timer (GPT)

Channel 1 is used as a free run timer for speed measurement.

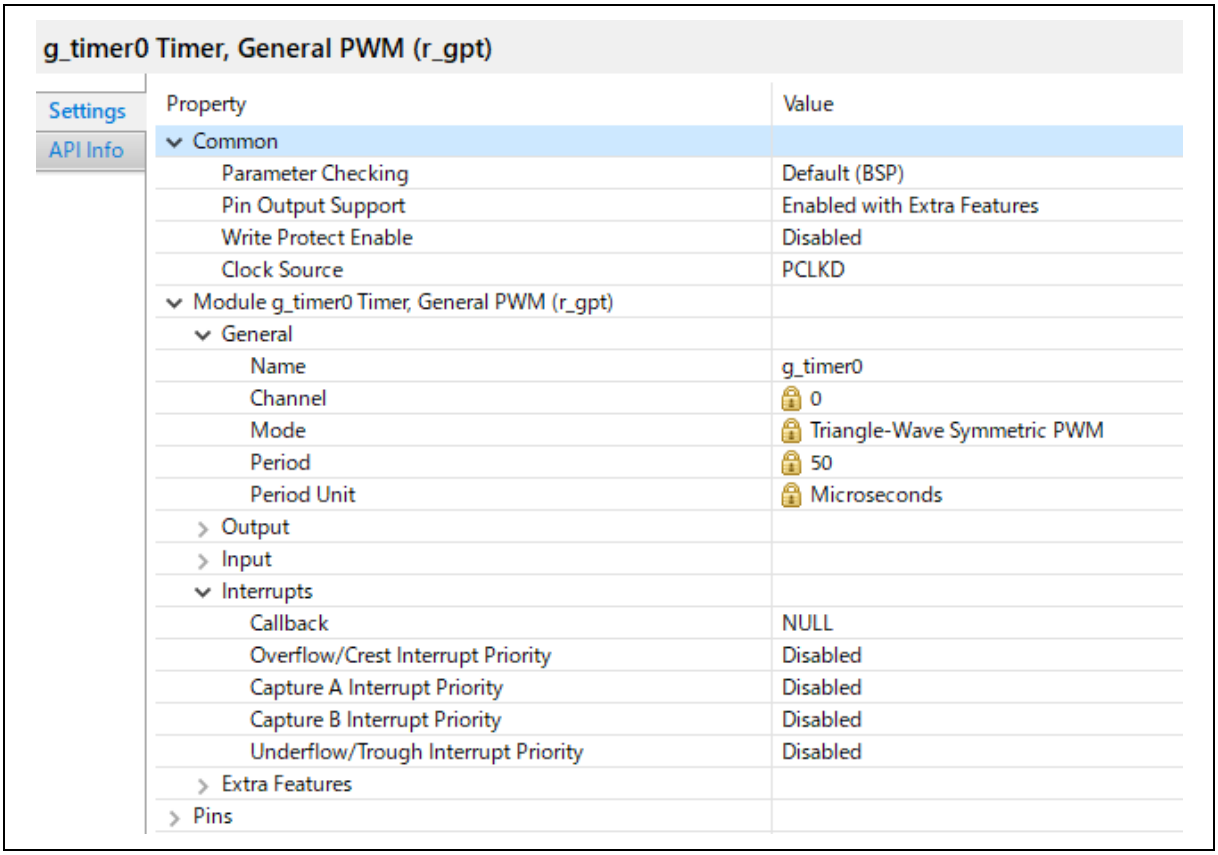


Figure 2-10 FSP Configuration of GPT Driver for free run timer (FSP 3.5.0)

(6) Port Output Enable for GPT (POEG)

The port executing PWM output are set to high impedance state when an overcurrent is detected (when a low level of the GTETRGC port is detected)

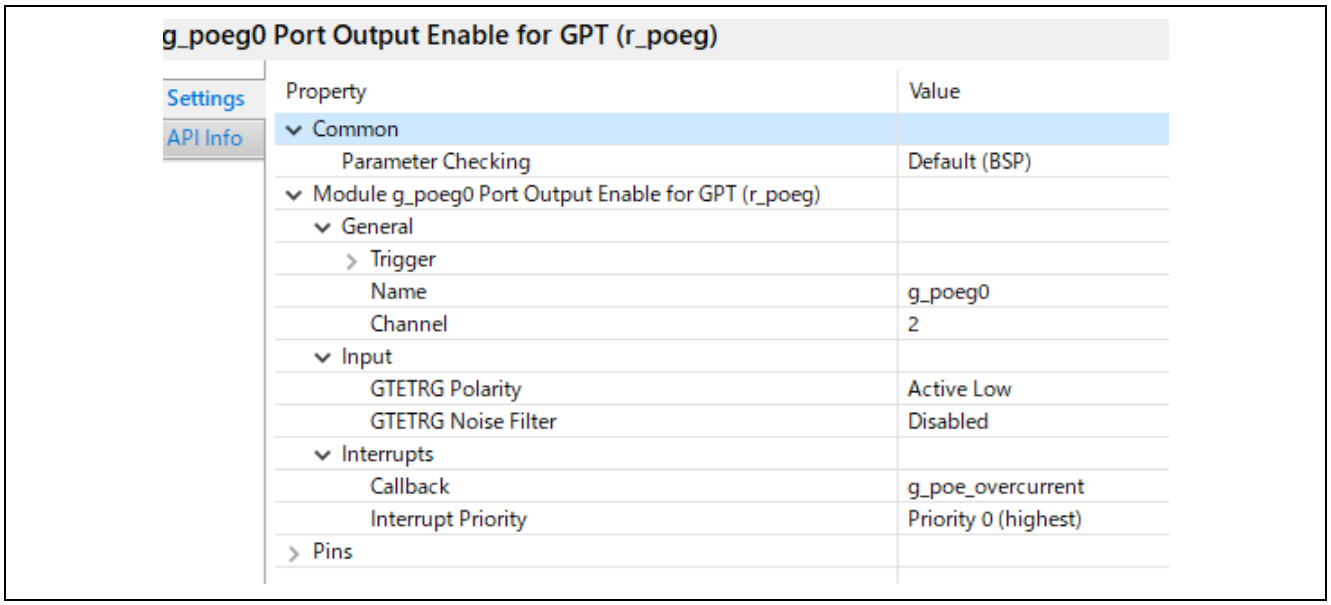
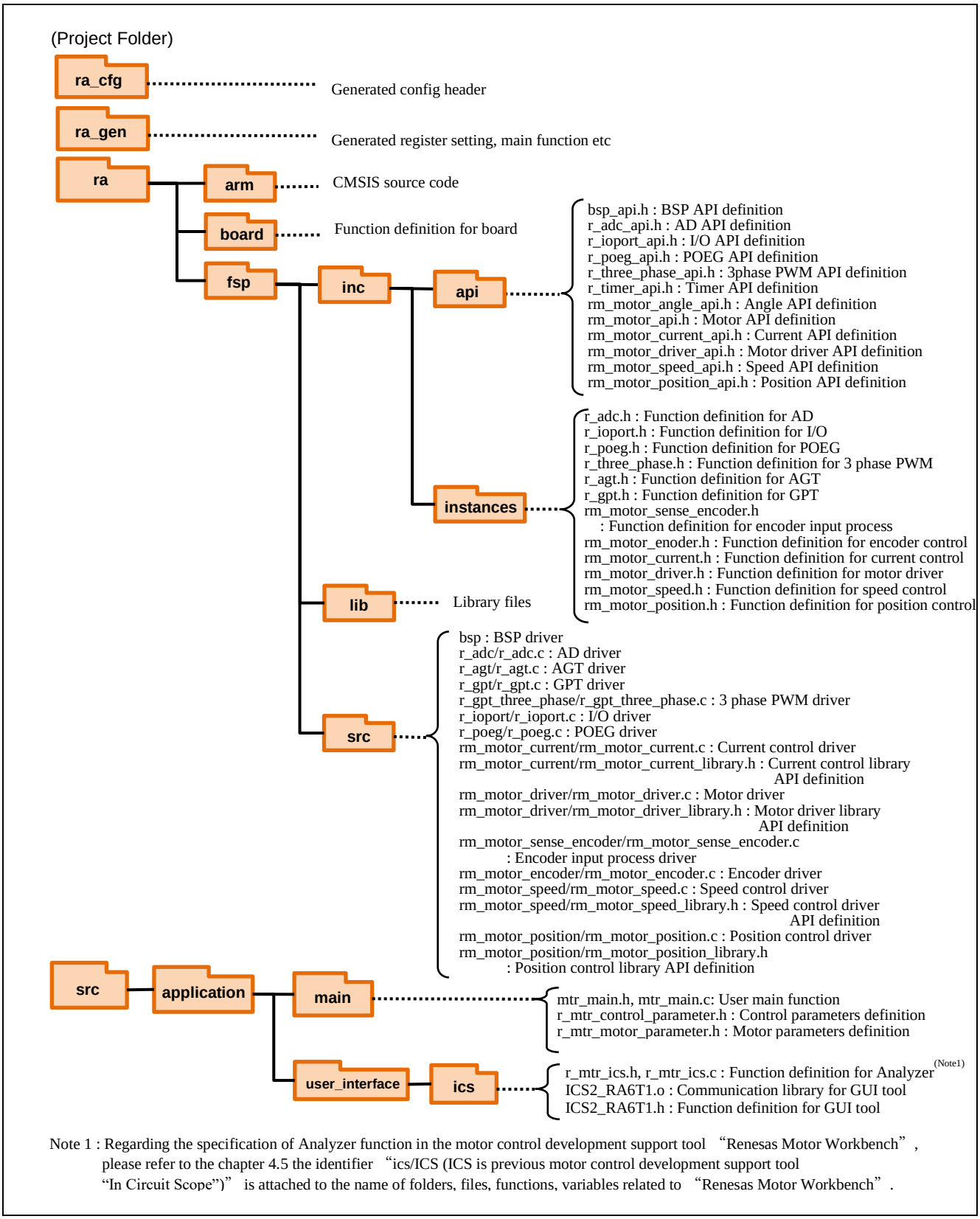


Figure 2-11 FSP Configuration of POEG Driver (FSP 3.5.0)

2.3 Software configuration

2.3.1 Software file configuration

Folder and file configuration of the software is given below.



Note 1 : Regarding the specification of Analyzer function in the motor control development support tool “Renesas Motor Workbench”, please refer to the chapter 4.5 the identifier “ics/ICS (ICS is previous motor control development support tool “In Circuit Scope”)” is attached to the name of folders, files, functions, variables related to “Renesas Motor Workbench”.

Figure 2-12 Folder and file configuration

2.3.2 Module configuration

Module configuration of the software is described below.

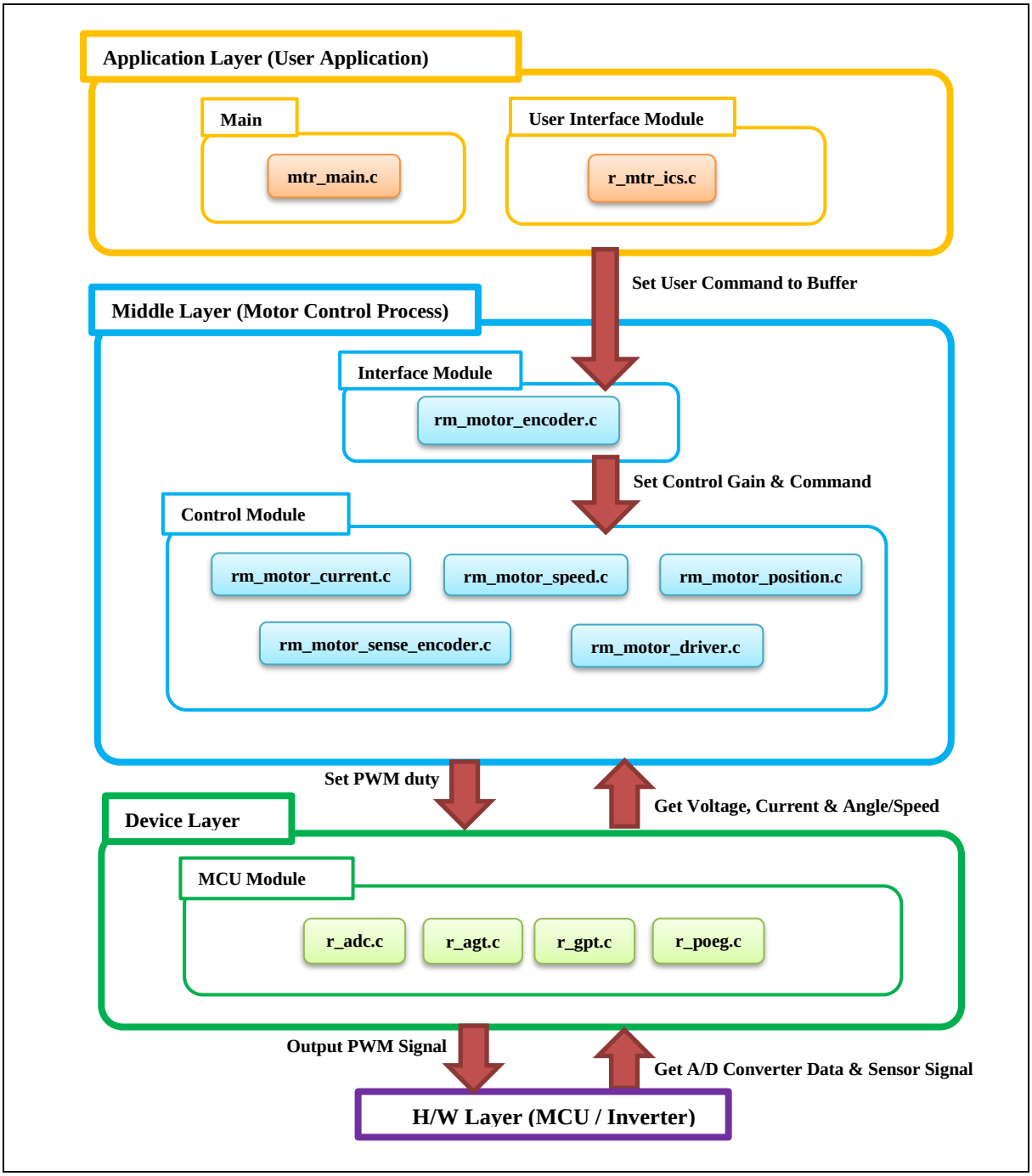


Figure 2-13 Module Configuration

2.4 Software specifications

Table 2-5 shows basic software specification of this system. For details of the vector control with encoder, refer to the application note 'Vector control with encoder for permanent magnet synchronous motor (Algorithm)' (R01AN3789).

Table 2-5 Basic Specifications of Vector Control with encoder Software

Item	Content	
Control method	Vector control	
Position detection method	Incremental encoder (A-B Phase)	
Motor rotation start/stop	SW1 input or input from 'Renesas Motor Workbench'	
Input voltage	DC 24 [V]	
Main clock frequency	120 [MHz]	
Carrier frequency (PWM)	20 [kHz] (Carrier period: 50 [µs])	
Dead time	2 [µs]	
Control period	Current control: 50 [µs] (the carrier period) Speed control: 500 [µs]	
Rotation speed control range	CW: 0 [rpm] to 4000 [rpm] CCW: 0 [rpm] to 4000 [rpm]	
Position control range	At board_ui	Position command generation: Voltage input divided by VR (input range) -180° to 180°
	At ics_ui	Position command generation: Position profile of trapezoidal curve for speed command value (input range) -32768° to 32767° (Max speed) CW / CCW: 4000[rpm]
Dead band of position*	Encoder count ±1 [cpr] (±0.09degree)	
Natural frequency of each control system	Current control system : 300 [Hz] Speed control system : 15 [Hz] Position estimation system : 5 [Hz]	
Optimization setting of compiler	Optimization level	Optimize more(-O2) (default setting)
Processing stop for protection	Disables the motor control signal output (six outputs), under any of the following conditions. 1. Instantaneous value of current of any phase exceeds 3.82(=1.8*sqrt (2)*1.5) [A] (monitored every 50 [µs]) 2. Inverter bus voltage exceeds 28 [V] (monitored every 50 [µs]) 3. Inverter bus voltage is less than 14 [V] (monitored every 50 [µs]) 4. Rotation speed exceeds 4500 [rpm] (monitored every 50 [µs]) When an external over current signal is detected (when a low level of the GTETRGC port is detected), the PWM output ports are set to high impedance state.	

* Dead band is provided to avoid hunting at the detection of target position.

2.5 Interrupt Priority

Table 2-6 shows the interrupt and priorities used in this system.

Table 2-6 Interrupt priority

Interrupt level	Priority	function
15	Min	
14		
13		
12		
11		
10		500 [µsec] Interrupt handling
9		
8		
7		
6		
5		A/D conversion complete interrupt
4		
3		GPT capture compare A interrupt (Input capture of encoder input)
2		
1		
0	Max	Over current error interrupt

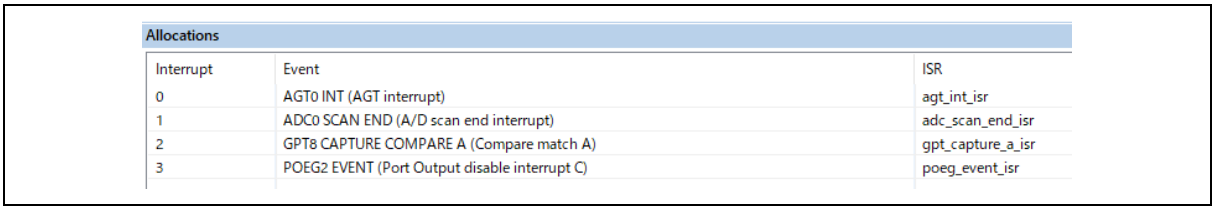


Figure 2-14 FSP Interrupts Configuration (FSP 3.5.0)

3. Descriptions of the control program

The target software of this application note is explained here.

3.1 Contents of control

3.1.1 Motor start/stop

Starting and stopping of the motor are controlled by input from ‘Renesas Motor Workbench’ or SW1.

A general-purpose port is assigned to SW1 and based upon its level the motor operation is controlled.

"Low" level → Motor Start

"High" level → Motor Stop

3.1.2 A/D Converter

(1) Motor rotation speed reference

The rotation speed and position command value of the motor is determined from the input from ‘Renesas Motor Workbench’ or the output value (analog value) of VR1. Rotation speed and position command value from VR1 is measured as shown in the table below.

Table 3-1 Conversion Ratio of the Rotation Speed Reference

Item	Conversion ratio (reference: A/D conversion value)		Channel
Rotation speed reference	CW	0 rpm to 4000 rpm: 0800H to 0FFFH	AN017
	CCW	0 rpm to 4000 rpm: 07FFH to 0000H	
Position reference	CW	0 rpm to 180 degrees: 0800H to 0FFFH	
	CCW	0 rpm to 180 degrees: 07FFH to 0000H	

(2) Inverter bus voltage

Inverter bus voltage is measured as shown in the table below. It is used for calculation of modulation rate and detection of overvoltage and undervoltage (PWM stops in case of abnormality).

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	AN005

(3) U, W phase current

The U, V and W phase currents are measured as shown in Table 3-3 and used for vector control. User can select only U and W phase currents to use as 2shunt resistances detection.

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

Item	Conversion ratio (U, V, W phase current: A/D conversion value)	Channel
U, W phase current	-12.5 [A] to 12.5 [A]: 0000H to 0E8BH (Note) Current = (5.0V-2.5V)/(0.01Ohm*20) = 12.5A In this system, the current detection circuit shifts the level from 5V to 3V, so 0E8BH is the upper limit of A / D conversion.	Iu: AN000 Iv: AN001 Iw: AN002

Note: For more details of A/D conversion characteristics, refer to “RA6T1 Group User’s Manual: Hardware”.

3.1.3 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.

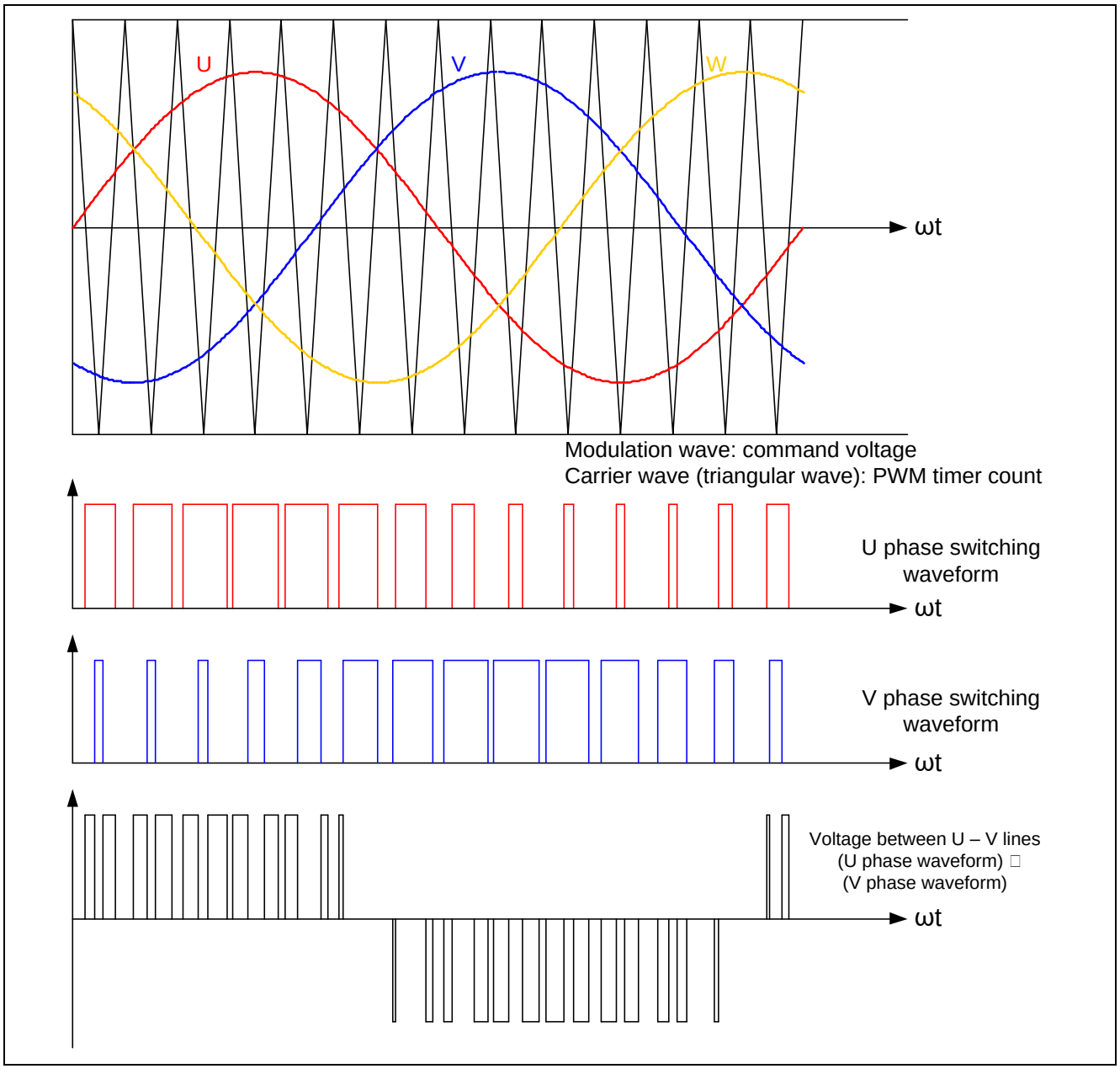


Figure 3-1 Conceptual Diagram of the Triangular Wave Comparison Method

As shown in the Figure 3-2, ratio of the output voltage pulse to the carrier wave is called duty.

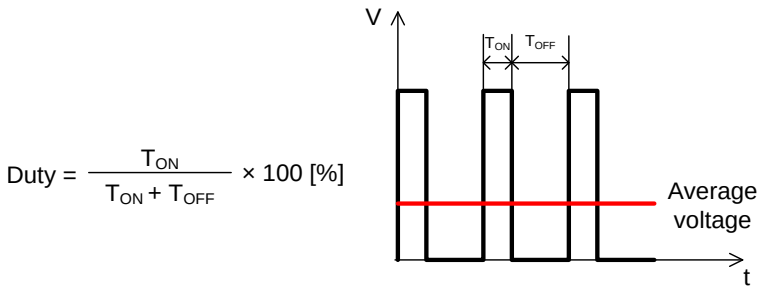


Figure 3-2 Definition of Duty

Modulation factor “m” is defined as follows.

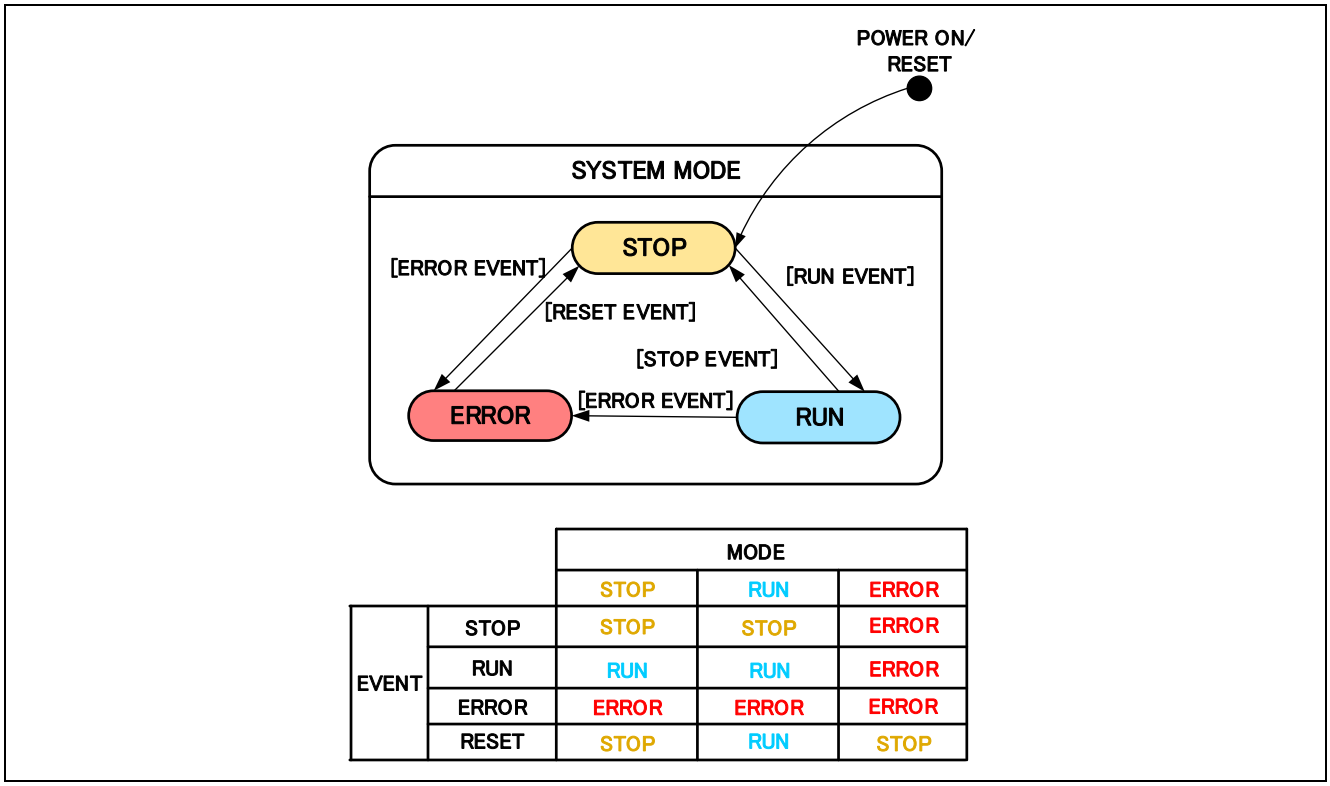
$$m = \frac{V}{E}$$

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

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

3.1.4 State transition

Figure 3-3 is a state transition diagram of the vector control with encoder software. In the target software of this application note, the software state is managed by “SYSTEM MODE”.



		MODE		
		STOP	RUN	ERROR
EVENT	STOP	STOP	STOP	ERROR
	RUN	RUN	RUN	ERROR
	ERROR	ERROR	ERROR	ERROR
	RESET	STOP	RUN	STOP

Figure 3-3 State Transition Diagram of Vector Control 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). EVENT

When “EVENT” occurs in each “SYSTEM MODE”, “SYSTEM MODE” changes as shown the table in Figure 3-3, according to that “EVENT”. The occurrence factors of each event are shown below.

Table 3-4 List of EVENT

EVENT name	occurrence factor
STOP	by user operation
RUN	by user operation
ERROR	when the system detects an error
RESET	by user operation

3.1.5 Start-up method

Figure 3-4 and Figure 3-5 show 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.

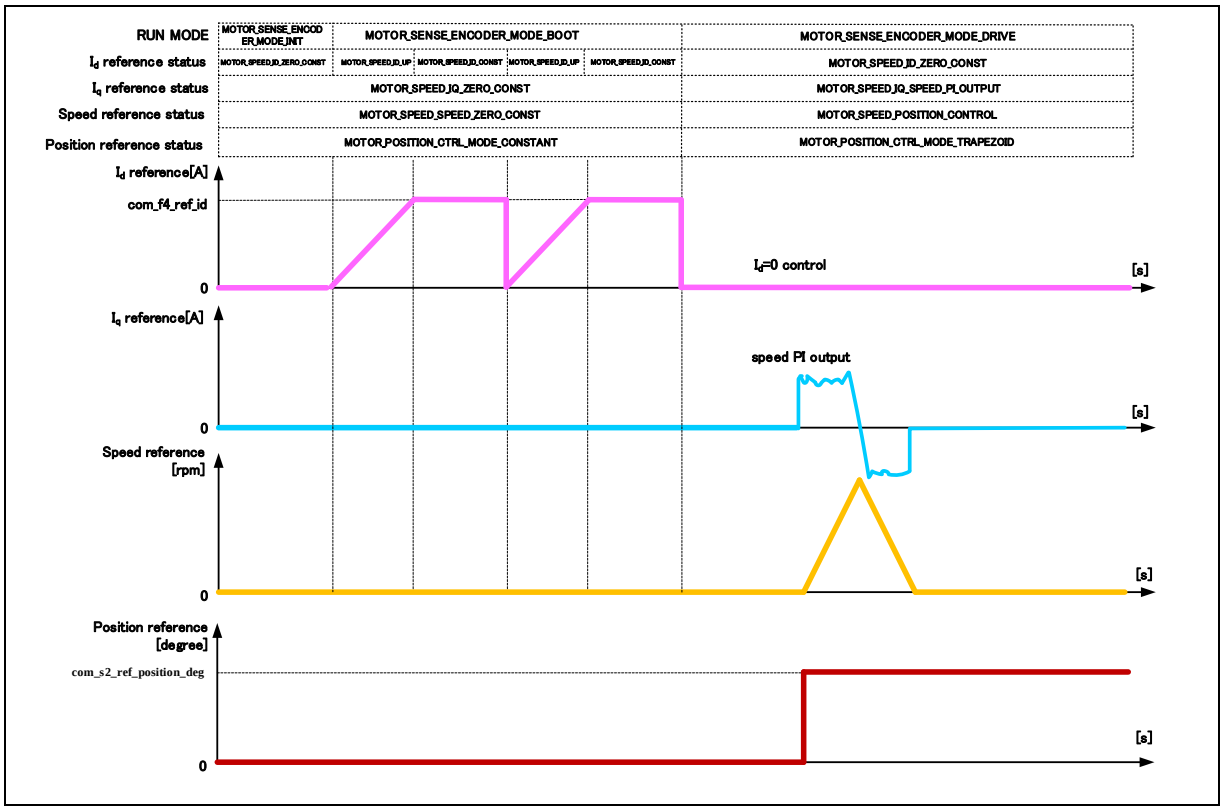


Figure 3-4 Startup Control of Vector Control with encoder Software (at position control)

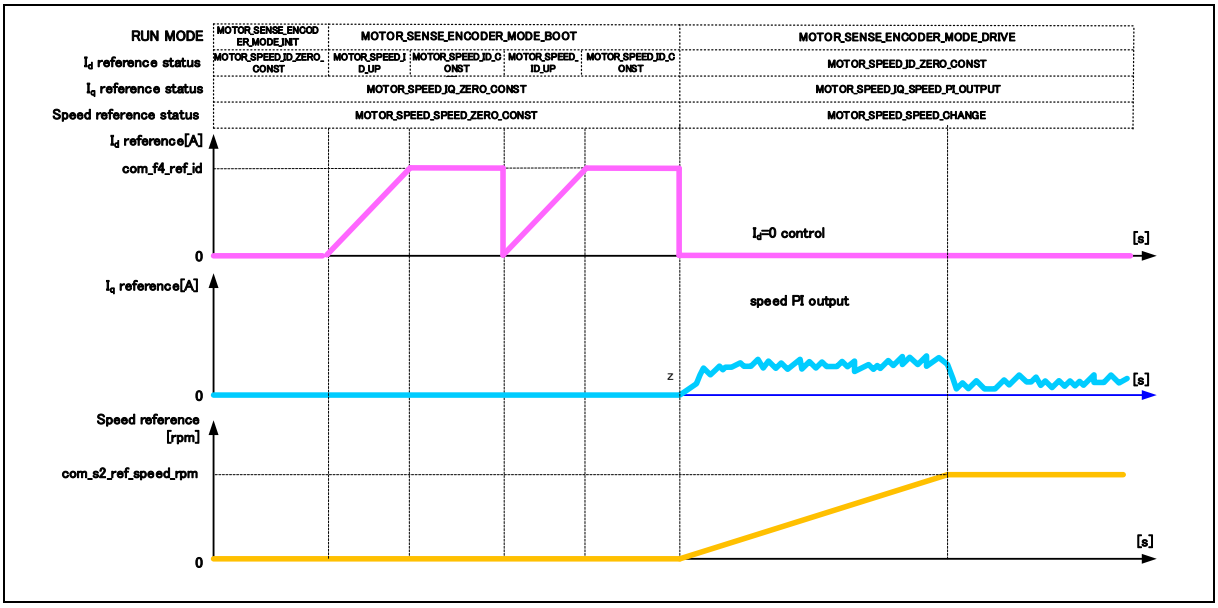


Figure 3-5 Startup Control of Vector Control with encoder Software (at speed control)

3.1.6 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.

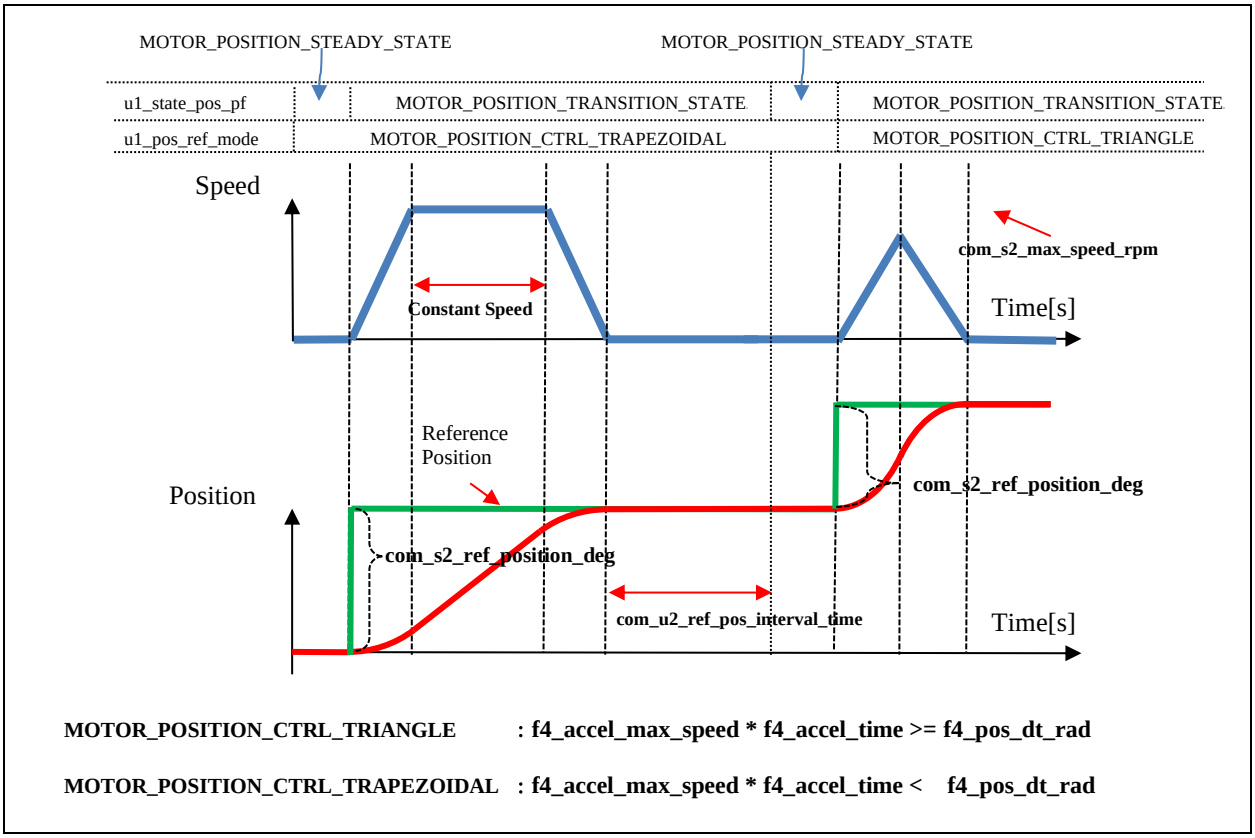


Figure 3-6 Position Profile Generation

following variables are required to create a command value

- Acceleration time (com_f4_accel_time)
- Maximum speed command value (com_f4_max_speed_rpm)
- Position stabilization wait time (com_u2_ref_pos_interval_time)

3.1.7 Speed Measurement

3.1.7.1 Speed calculation using encoder at low speed

The sample software calculates the speed from the encoder count (A, B phase signal edge) intervals for higher real-time and slow speed resolution. The speed is calculated from the count between the rising and falling edges of the A and B phases as shown in the figure.

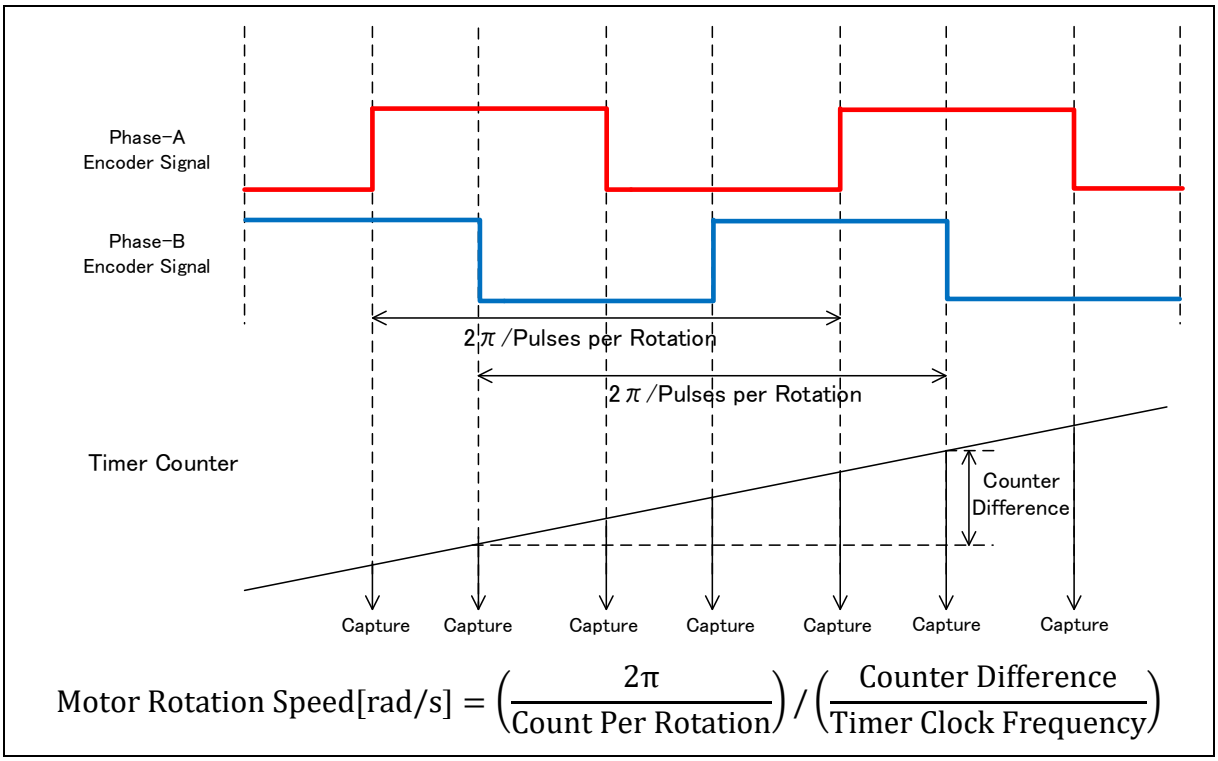
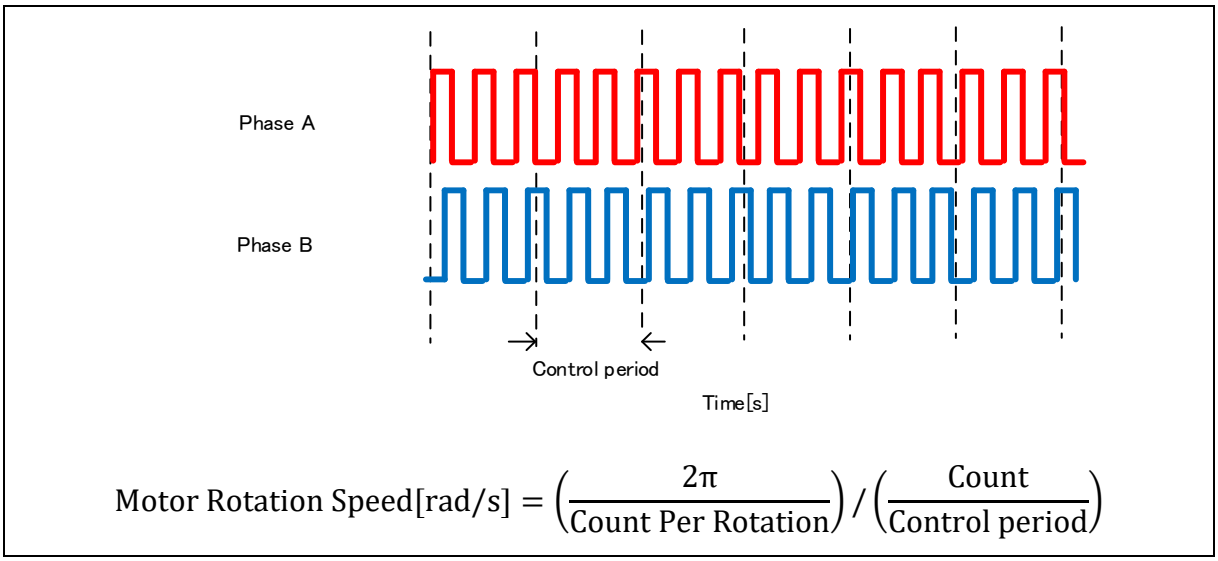


Figure 3-7 Speed calculation using encoder at low speed

3.1.7.2 Speed calculation using encoder at high speed

When using the encoder during high-speed rotation, the speed is calculated from the number of input pulses in the control cycle.



$$\text{Motor Rotation Speed}[\text{rad/s}] = \left(\frac{2\pi}{\text{Count Per Rotation}}\right) / \left(\frac{\text{Count}}{\text{Control period}}\right)$$

Figure 3-8 Speed calculation using encoder at high speed

3.1.8 System protection function

This control software has the following error status and executes emergency stop functions in case of occurrence of respective errors. Table 3-5 shows each software threshold for the system protection function.

- Over current error

There are two kind of overcurrent protection.

Hardware OCP: When emergency stop signal from the hardware is detected, PWM output ports are automatically set to high impedance output (without software intervention).

Software OCP: U, V, and W phase currents are monitored in over current monitoring cycle. When an over current is detected, the CPU executes emergency stop.

- Over voltage error

The inverter bus voltage is monitored in over voltage monitoring cycle. When an over voltage is detected (when the voltage exceeds the over voltage limit), the CPU performs emergency stop. Here, the over voltage limit is set in consideration of the error of resistance value of the detect circuit. When this error occurs, the CPU performs emergency stop in the side of the motor in which the error occurred.

- Low voltage error

The inverter bus voltage is monitored in low-voltage monitoring cycle. The CPU performs emergency stop when low voltage (when voltage falls below the limit) is detected. Here, the low voltage limit is set in consideration of the error of resistance value of the detect circuit. When this error occurs, the CPU performs emergency stop in the side of the motor in which the error occurred.

- 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. When this error occurs, the CPU performs emergency stop in the side of the motor in which the error occurred.

Table 3-5 Setting Values of the System Protection Function

Error name	Threshold	
Over current error	Over current limit [A]	3.82
	Monitoring cycle [µs]	50
Over voltage error	Over voltage limit [V]	28
	Monitoring cycle [µs]	50
Low voltage error	Low voltage limit [V]	14
	Monitoring cycle [µs]	50
Over speed error	Speed limit [rpm]	4500
	Monitoring cycle [µs]	50

3.1.9 AD triggers

Shows the timing of AD triggers and scan groups.

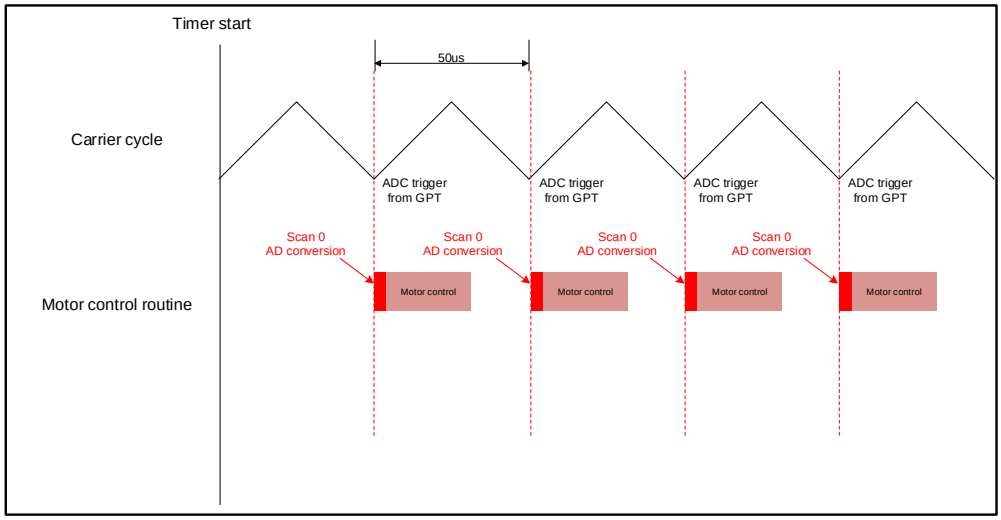


Figure 3-9 AD trigger timing

3.2 Function specifications of vector control with encoder software

The control process of the target software of this application note is mainly consisted of 50[μs] period interrupt (carrier interrupt) and 500[μs] period interrupt. As following Figure 3-10, the control process in the red broken line part is executed every 50[μs] cycle, and the control process in the blue broken line part is executed every 500[μs] cycle.

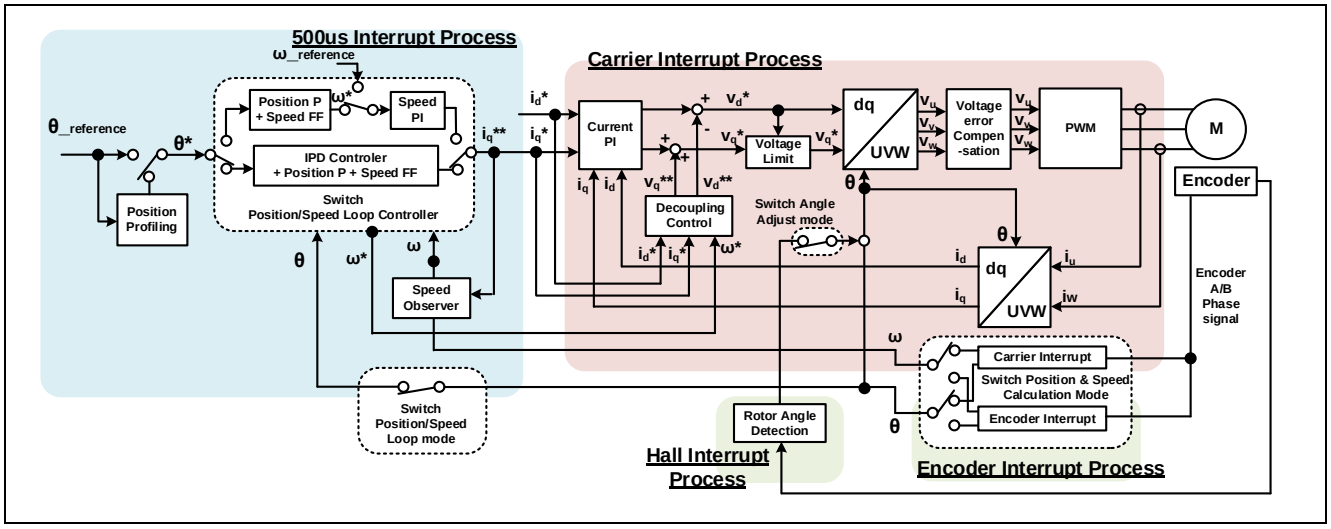


Figure 3-10 Block Diagram of Vector Control with encoder

This chapter shows the specification of 2 interrupt functions and functions executed in each interrupt cycle. In following tables, only primary functions of the vector control with encoder are listed. Regarding the specification of functions not listed in following tables, please refer to source codes.

Table 3-6 List of Functions Executed in 50[µs] Period Interrupt (1/4)

File name	Function name	Process overview
mtr_main.c	mtr_callback_event0 Input : (motor_encoder_callback_args_t *) p_args / Callback argument Output : None	Encoder control callback function
rm_motor_encoder.c	rm_motor_encoder_current_callback Input : (motor_current_callback_args_t *) p_args / Callback argument Output :None	Set the speed control output to the current control input
	RM_MOTOR_ENCODER_ErrorCheck Input : (motor_ctrl_t * const) p_ctrl / Pointer to control structure. (uint16_t * const) p_error / Pointer to get occured error Output : fsp_err_t / Execution result	Check the occurrence of Error.
	rm_motor_encoder_copy_speed_current Input : (motor_speed_output_t *) st_output / Pointer to the structure of Speed Control output (motor_current_input_t *) st_input / Pointer to the structure of Current Control input Output :None	Copy speed output data to current input data
rm_motor_driver.c	rm_motor_driver_cyclic Input : (adc_callback_args_t *) p_args / Callback argument Output :None	Motor driver callback function
	rm_motor_driver_current_get Input : (motor_driver_instance_ctrl_t *) p_ctrl / The pointer to the motor driver module instance Output :None	Get A/D converted data (Phase Current & Main Line Voltage)
	RM_MOTOR_DRIVER_FlagCurrentOffsetGet Input : (motor_driver_ctrl_t * const) p_ctrl / Pointer to control structure (uint8_t * const) p_flag_offset / Flag of finish current offset detection Output : fsp_err_t / Execution result	Measure current offset values
	RM_MOTOR_DRIVER_PhaseVoltageSet Input : (motor_driver_ctrl_t * const) p_ctrl / Pointer to control structure (float const) u_voltage / U phase voltage (float const) v_voltage / V phase voltage (float const) w_voltage / W phase voltage Output : fsp_err_t / Execution result	Set Phase Voltage Data to calculate PWM duty.
	rm_motor_driver_modulation Input : (motor_driver_instance_ctrl_t *) p_ctrl / The pointer to the motor driver module instance Output :None	Perform PWM modulation

Table 3-7 List of Functions Executed in 50[µs] Period Interrupt (2/4)

File name	Function name	Process overview
rm_motor_driver.c	rm_motor_driver_mod_run Input : (motor_driver_modulation_t *) p_mod / Pointer to the modulation data structure (const float *) p_f4_v_in / Pointer to the 3-phase input voltage (float *) p_f4_duty_out / Where to store the 3-phase output duty cycle Output :None	Calculates duty cycle from input 3-phase voltage (bipolar)
	rm_motor_driver_set_uvw_duty Input : (motor_driver_instance_ctrl_t *) p_ctrl / Pointer to Motor Driver instance (float) f_duty_u / The duty cycle of Phase-U (float) f_duty_v / The duty cycle of Phase-V (float) f_duty_w / The duty cycle of Phase-W Output : fsp_err_t / Execution result	PWM duty setting
	RM_MOTOR_DRIVER_CurrentGet Input : (motor_driver_ctrl_t * const) p_ctrl / Pointer to control structure (motor_driver_current_get_t * const) p_current_get / Pointer to get data structure Output : fsp_err_t / Execution result	Get calculated phase Current, Vdc & Va_max data
rm_motor_current.c	rm_motor_current_cyclic Input : (motor_driver_callback_args_t *) p_args / Callback argument Output :None	Current control cycle operation
	RM_MOTOR_CURRENT_ParameterSet Input : (motor_current_ctrl_t * const) p_ctrl / Pointer to control structure (motor_current_input_current_t const * const) p_st_input / Pointer to input data structure Output : fsp_err_t / Execution result	Set (Input) Parameter Data.
	RM_MOTOR_CURRENT_CurrentSet Input : (motor_current_ctrl_t * const) p_ctrl / Pointer to control structure (motor_current_input_current_t const * const) p_st_current / Pointer to input current structure (motor_current_input_voltage_t const * const) p_st_voltage / Pointer to input voltage structure Output : fsp_err_t / Execution result	Set d/q-axis Current & Voltage Data.
	RM_MOTOR_CURRENT_CurrentGet Input : (motor_current_ctrl_t * const) p_ctrl / Pointer to control structure (float * const) p_id / Pointer to get d-axis current (float * const) p_iq / Pointer to get q-axis current Output : fsp_err_t / Execution result	Get d/q-axis Current.
	motor_current_transform_uvw_dq_abs Input : (const float) f_angle / rotor angle (const float *) f_uvw / the pointer to the UVW-phase array in [U,V,W] format (float *) f_dq / where to store the [d,q] formated array on dq coordinates Output :None	Coordinate transform UVW to dq (absolute transform)

Table 3-8 List of Functions Executed in 50[µs] Period Interrupt (3/4)

File name	Function name	Process overview
rm_motor_current.c	motor_current_angle_cyclic Input : (motor_current_instance_t *) p_instance / The pointer to current control module control instance Output :None	Angle/Speed Process in Cyclic Process of Current Control
	RM_MOTOR_CURRENT_SpeedPhaseSet Input : (motor_current_ctrl_t * const) p_ctrl / Pointer to control structure (float const) speed_rad / Rotational speed (float const) phase_rad / Rotor phase Output : fsp_err_t / Execution result	Set Current Speed & rotor phase Data.
	RM_MOTOR_CURRENT_CurrentReferenceSet Input : (motor_current_ctrl_t * const) p_ctrl / Pointer to control structure (float const) speed_rad / D-axis current Reference (float const) phase_rad / Q-axis current Reference Output : fsp_err_t / Execution result	Set Current Reference Data
	RM_MOTOR_CURRENT_PhaseVoltageGet Input : (motor_current_ctrl_t * const) p_ctrl / Pointer to control structure (motor_current_get_voltage_t * const) p_voltage / Pointer to get Voltages Output : fsp_err_t / Execution result	Gets the set phase voltage.
	motor_current_pi_calculation Input : (motor_current_instance_ctrl_t *) p_instance / The pointer to the FOC current control structure Output :None	Calculates the output voltage vector from current vector command and actual current vector
	motor_current_pi_control Input : (motor_current_pi_params_t *) pi_ctrl / The pointer to the PI control structure Output : float / PI control output value	PI control
	motor_current_limit_abs Input : (float) f4_value / Target value (float) f4_limit_value / Limit Output : float / Limited value	Limit with absolute value
	motor_current_decoupling Input : (motor_current_instance_ctrl_t *) p_ctrl / The pointer to the FOC current control instance (float) f_speed_rad / The electrical speed (const motor_current_motor_parameter_t *) p_mtr / The pointer to the motor parameter data structure Output :None	Decoupling control
	motor_current_voltage_limit Input : (motor_current_instance_ctrl_t *) p_ctrl / The pointer to the FOC current control structure Output :None	Limit voltage vector

Table 3-9 List of Functions Executed in 50[µs] Period Interrupt (4/4)

File name	Function name	Process overview
rm_motor_current.c	motor_current_transform_dq_uvw_abs Input : (const float) f_angle / Rotor angle (const float *) f_dq / The pointer to the dq-axis value array in [D,Q] format (float *) f_uvw / Where to store the [U,V,W] formated 3-phase quantities array Output :None	Coordinate transform dq to UVW 3-phase (absolute transform)
librm_motor_current.a	rm_motor_voltage_error_compensation_main Input : (motor_currnt_voltage_compensation_t *) st_volt_comp / Voltage error compensation data (float *) p_f4_v_array / Reference voltage (float *) p_f4_i_array / Reference current (float) f4_vdc / Bus voltage Output :None	Voltage error compensation
rm_motor_sense_encoder.c	RM_MOTOR_SENSE_ENCODER_InternalCalculate Input : (motor_angle_ctrl_t * const) p_ctrl / Pointer to control structure Output : fsp_err_t / Execution result	Calculation of angle and speed according to measured encoder signal.
	RM_MOTOR_SENSE_ENCODER_AngleSpeedGet Input : (motor_angle_ctrl_t * const) p_ctrl / Pointer to control structure (float * const) p_angle / Pointer to get angle data (float * const) p_seed / Pointer to get speed data (float * const) p_phase_err / Pointer to get phase error data Output : fsp_err_t / Execution result	Get angle, speed and phase error data about the roter.
r_gpt_three_phase.c	R_GPT_THREE_PHASE_DutyCycleSet Input : (three_phase_ctrl_t * const) p_ctrl / Control block set in @ref three_phase_api_t::open call for this timer (three_phase_duty_cycle_t * const) p_duty_cycle / Duty cycle values for all three timer channels Output : fsp_err_t / Execution result	Sets duty cycle for all three timers.

Table 3-10 List of Functions Executed in 500[µs] Interrupt (1/2)

File name	Function name	Process overview
mtr_main.c	mtr_callback_encoder Input : (motor_encoder_callback_args_t *) p_args / Callback argument Output :None	Encoder control callback function
	get_vr1 Input :None Output : uint16_t / conversion value	Get VR1 A/D conversion value
rm_motor_current.c	RM_MOTOR_CURRENT_ParameterGet Input : (motor_current_ctrl_t * const) p_ctrl / Pointer to control structure (motor_current_input_current_t const * const) p_st_input / Pointer to output data structure Output : fsp_err_t / Execution result	Get speed control input data from current control
rm_motor_encoder.c	rm_motor_encoder_speed_callback Input : (motor_speed_callback_args_t *) p_args / Callback argument Output :None	Speed control callback function
	rm_motor_encoder_copy_current_speed Input : (motor_current_output_t *) st_output / Pointer to the structure of Current Control output (motor_speed_input_t *) st_input / Pointer to the structure of Speed Control input Output :None	Copy current output data to speed input data
rm_motor_speed.c	rm_motor_speed_cyclic Input : (timer_callback_args_t *) p_args/ Callback argument Output :None	Cyclic process of Speed Control (Call at timer interrupt)
	RM_MOTOR_SPEED_ParameterSet Input : (motor_speed_ctrl_t * const) p_ctrl / Pointer to control structure (motor_speed_input_t const * const) p_st_input / Pointer to structure to input parameters Output : fsp_err_t / Execution result	Set speed Input parameters
	RM_MOTOR_SPEED_SpeedControl Input : (motor_speed_ctrl_t * const) p_ctrl / Pointer to control structure Output : fsp_err_t / Execution result	Calculates the d/q-axis current reference.(Main process of Speed Control)
	rm_motor_speed_set_speed_ref Input : (motor_speed_instance_ctrl_t *) p_ctrl / The pointer to the FOC data instance Output : float / Speed reference	Updates the speed reference
	rm_motor_speed_set_iq_ref Input : (motor_speed_instance_ctrl_t *) p_ctrl / The pointer to the ctrl instance Output : float / Iq reference	Updates the q-axis current reference
	rm_motor_speed_set_id_ref Input : (motor_speed_instance_ctrl_t *) p_ctrl / The pointer to the ctrl instance Output : float / Id reference	Updates the d-axis current reference
	RM_MOTOR_SPEED_ParameterGet Input : (motor_speed_ctrl_t * const) p_ctrl / The pointer to the ctrl instance (motor_speed_output_t * const) p_st_output / Pointer to get speed control parameters Output : fsp_err_t / Execution result	Get speed control output parameters

Table 3-11 List of Functions Executed in 500[µs] Interrupt (2/2)

File name	Function name	Process overview
librm_motor_speed.a	rm_motor_speed_first_order_lpf Input : (motor_speed_lpf_t *) p_lpf / First order LPF structure (float) f4_omega / Natural frequency (float) f4_ctrl_period / Control period Output : None	First Order LPF
	rm_motor_speed_fluxwkn_set_vamax Input : (motor_speed_flux_weakening_t *) p_fluxwkn / The pointer to flux weakening structure (float) f4_va_max / maximum magnitude of voltage vector Output :None	Sets the maximum magnitude of voltage vector
	rm_motor_speed_fluxwkn_run Input : (motor_speed_flux_weakening_t *) p_fluxwkn / The pointer to flux weakening structure (float) f4_speed_rad / The electrical speed of motor (const float *) p_f4_idq / The pointer to the measured current vector in format d/q (float *) p_f4_idq_ref / The pointer to the reference current vector in format d/q Output :None	Executes the flux-weakening module

3.3 Contents of control

3.3.1 Configuration Options

The configuration options of the vector control with encoder module for motor can be configured using the RA Configurator. The changed options are automatically reflected to the hal_data.c when generating code. The option names and setting values are listed in the Table 3-12 shown as follows.

Table 3-12 Configuration Options for encoder module

Configuration Options (rm_motor_encoder.h)	
Options	**Description**
Limit of over current (A) Initial: 1.8F	When a phase current exceeds this value, PWM output ports are set to off.
Limit of over voltage (V) Initial: 28.0F	When an inverter voltage exceeds this value, PWM output ports are set to off.
Limit of over speed (rpm) Initial: 4500.0F	When a rotation speed exceeds this value, PWM output ports are set to off.
Limit of over speed (rpm) Initial: 14.0F	When an inverter voltage becomes below this value, PWM output ports are set to off.

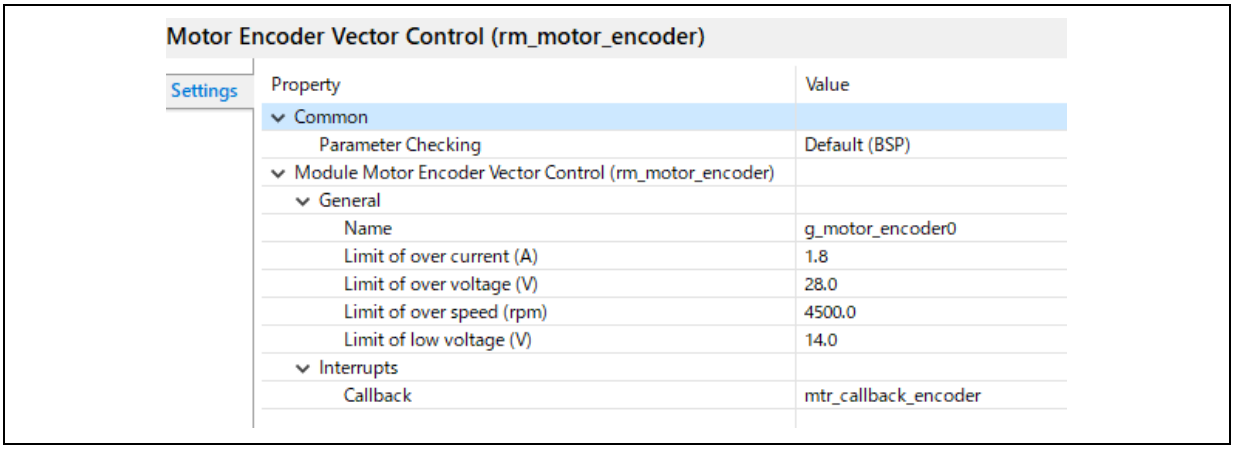


Figure 3-11 FSP Configuration of Motor vector control with encoder (FSP 3.5.0)

3.3.2 Configuration Options for included modules

The vector control with encoder module for motor includes below modules.

- Current Module
- Speed Module
- Position Module
- Angle Module
- Driver Module

And also these included modules have each configuration parameters as same as the vector control with encoder module. The option names and setting values are listed in the tables shown as follows.

Table 3-13 Configuration Options for Current Control

Configuration Options (rm_motor_current.h)	
Options	**Description**
Shunt type Initial: 2shunt	Selects how many shunt resistances to use current detection. Please set to “1shunt”.
Input voltage (V) Initial: 24.0F	Input voltage [V]
Voltage error compensation Initial: Enable	Selects whether to “enable” or “disable” voltage error compensation.
Design Parameter | Current PI loop omega Initial: 300.0F	Current PI control omega parameter [Hz].
Design Parameter | Current PI loop zeta Initial: 1.0F	Current PI control zeta parameter.
Motor Parameter | Pole pairs Initial: 4	Pole pairs of target motor.
Motor Parameter | Resistance (ohm) Initial: 0.84	Resistance of motor [ohm].
Motor Parameter | Inductance of d-axis (H) Initial: 0.0011	D-axis inductance [H].
Motor Parameter | Inductance of d-axis (H) Initial: 0.0011	Q-axis inductance [H].
Motor Parameter | Permanent magnetic flux (Wb) Initial: 0.00623	Magnetic flux [Wb].
Motor Parameter | Rotor inertia (kgm^2) Initial: 0.0000041	Rotor inertia [kgm^2].

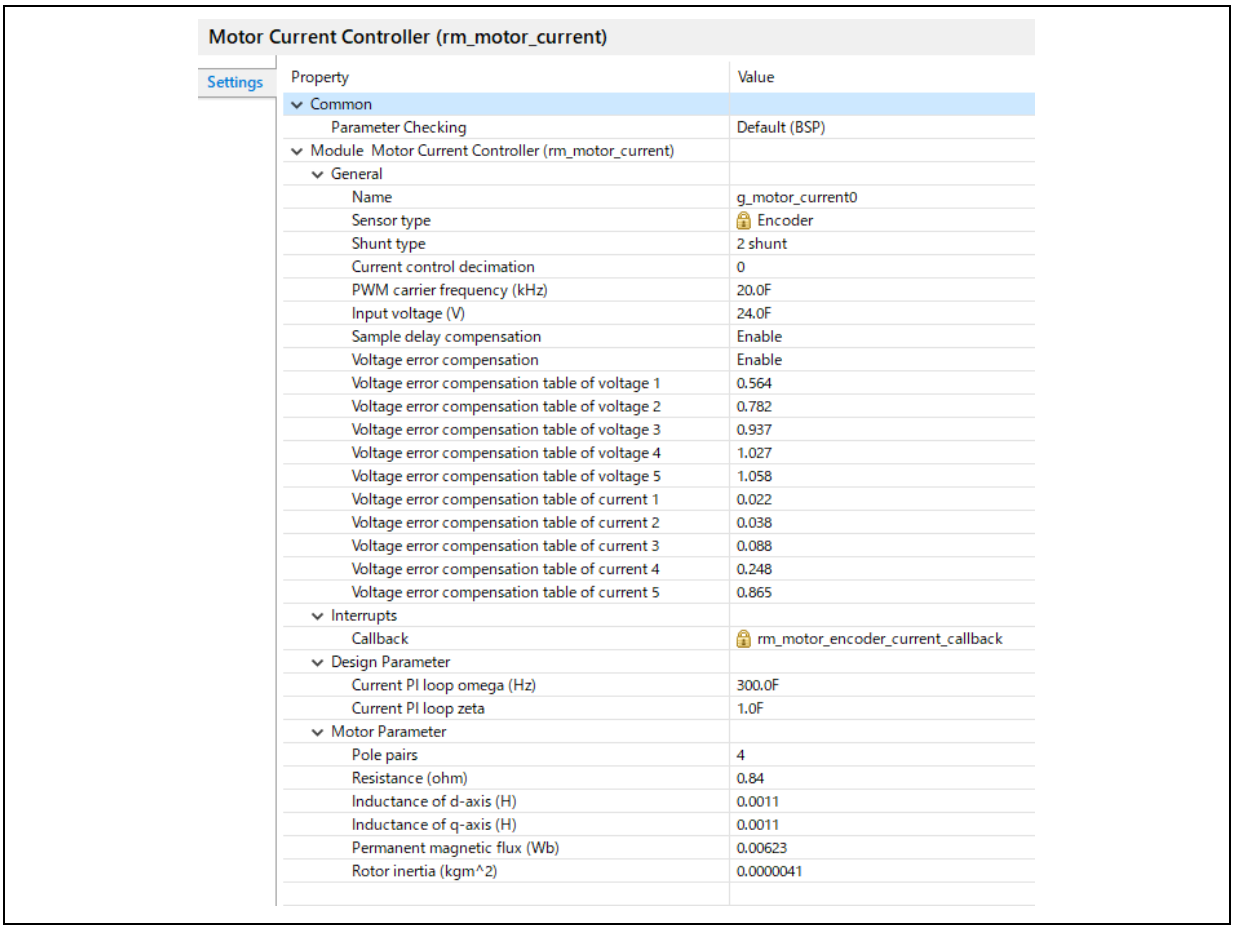


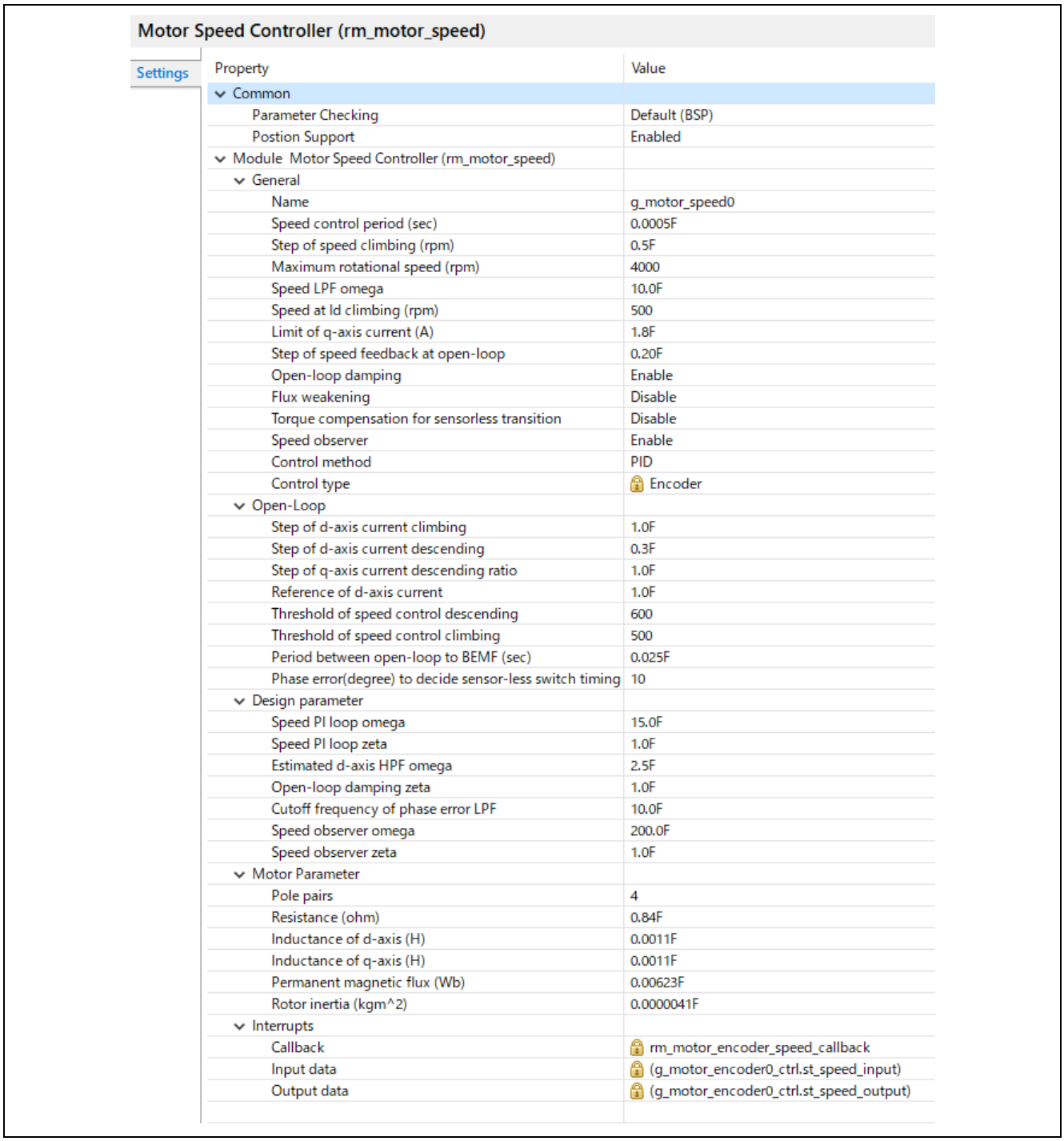
Figure 3-12 FSP Configuration of Motor Current Controller (FSP 3.5.0)

Table 3-14 Configuration Options for Speed Control [1/2]

Configuration Options (rm_motor_speed.h)	
Options	**Description**
Speed control period (sec) Initial: 0.0005F	The period of speed control process [sec].
Step of speed climbing (rpm) Initial: 0.5F	The step of speed fluctuation [rpm]. Program controls speed by this step at acceleration and deceleration.
Maximum rotational speed (rpm) Initial: 4000	Maximum rotational speed [rpm]
Speed LPF omega Initial: 10.0F	Speed LPF parameter omega [Hz].
Speed at Id climbing (rpm) Initial: 500	The threshold speed to control d-axis current increase [rad/s]. Program increases d-axis current at start up the motor rotation until the speed reaches this value.
Limit of q-axis current (A) Initial: 1.8F	Limit of q-axis current [A].
Flux weakening Initial: Disable	Select enable/disable of flux weakening control at high speed.

Table 3-15 Configuration Options for Speed Control [2/2]

Configuration Options (rm_motor_speed.h)	
Options	**Description**
Design parameter | Speed PI loop omega Initial: 15.0F	Speed PI Control parameter omega.
Design parameter | Speed PI loop zeta Initial: 1.0F	Speed PI Control parameter zeta.
Design parameter | Cutoff frequency of phase error LPF Initial: 10.0F	The cut-off frequency [Hz] of phase error LPF gain design.
Design parameter | Speed observer omega Initial: 200.0F	Speed observer omega.
Design parameter | Speed observer zeta Initial: 1.0F	Speed observer zeta.
Motor Parameter | Pole pairs Initial: 4	Pole pairs of target motor.
Motor Parameter | Resistance (ohm) Initial: 0.84F	Resistance of motor [ohm].
Motor Parameter | Inductance of d-axis (H) Initial: 0.0011F	D-axis inductance [H].
Motor Parameter | Inductance of q-axis (H) Initial: 0.0011F	Q-axis inductance [H].
Motor Parameter | Permanent magnetic flux (Wb) Initial: 0.00623F	Magnetic flux [Wb].
Motor Parameter | Rotor inertia (kgm^2) Initial: 0.0000041F	Rotor inertia [kgm^2].



Motor Speed Controller (rm_motor_speed)

Settings

Property	Value
Common	
Parameter Checking	Default (BSP)
Postion Support	Enabled
Module Motor Speed Controller (rm_motor_speed)	
General	
Name	g_motor_speed0
Speed control period (sec)	0.0005F
Step of speed climbing (rpm)	0.5F
Maximum rotational speed (rpm)	4000
Speed LPF omega	10.0F
Speed at Id climbing (rpm)	500
Limit of q-axis current (A)	1.8F
Step of speed feedback at open-loop	0.20F
Open-loop damping	Enable
Flux weakening	Disable
Torque compensation for sensorless transition	Disable
Speed observer	Enable
Control method	PID
Control type	Encoder
Open-Loop	
Step of d-axis current climbing	1.0F
Step of d-axis current descending	0.3F
Step of q-axis current descending ratio	1.0F
Reference of d-axis current	1.0F
Threshold of speed control descending	600
Threshold of speed control climbing	500
Period between open-loop to BEMF (sec)	0.025F
Phase error(degree) to decide sensor-less switch timing	10
Design parameter	
Speed PI loop omega	15.0F
Speed PI loop zeta	1.0F
Estimated d-axis HPF omega	2.5F
Open-loop damping zeta	1.0F
Cutoff frequency of phase error LPF	10.0F
Speed observer omega	200.0F
Speed observer zeta	1.0F
Motor Parameter	
Pole pairs	4
Resistance (ohm)	0.84F
Inductance of d-axis (H)	0.0011F
Inductance of q-axis (H)	0.0011F
Permanent magnetic flux (Wb)	0.00623F
Rotor inertia (kgm^2)	0.0000041F
Interrupts	
Callback	rm_motor_encoder_speed_callback
Input data	(g_motor_encoder0_ctrl.st_speed_input)
Output data	(g_motor_encoder0_ctrl.st_speed_output)

Figure 3-13 FSP Configuration of Motor Speed Controller (FSP 3.5.0)

Table 3-16 Configuration Options for Position control module [1/2]

Configuration Options (rm_motor_position.h)	
Options	**Description**
Position dead band Initial: 1U	Position dead band
Position band limit Initial: 3U	Zero position error range
Speed feedforward ratio Initial: 0.8F	Speed feedforward ratio
Encoder counts per one rotation Initial: 4000.0F	Encoder count per revolution
Position omega Initial: 10.0F	Position control omega parameter [Hz].
Period of speed control (sec) Initial: 0.0005F	Speed control execution cycle
Position Profiling | Interval time Initial: 400U	Position response steady-state waiting time
Position Profiling | Accel time Initial: 0.3F	Acceleration time
Position Profiling | Maximum accel time Initial: 11077.904F	Maximum acceleration time calculation parameter
Position Profiling | Accelaration maximu speed Initial: 4000.0F	Position profile maximum rotation speed
Position Profiling | Update step of timer Initial: 0.0005F	Position profile update cycle
Motor Parameter | Pole pairs Initial: 4	Pole pairs of target motor.
Motor Parameter | Resistance (ohm) Initial: 0.84F	Resistance of motor [ohm].
Motor Parameter | Inductance of d-axis (H) Initial: 0.0011F	D-axis inductance [H].
Motor Parameter | Inductance of q-axis (H) Initial: 0.0011F	Q-axis inductance [H].
Motor Parameter | Permanent magnetic flux (Wb) Initial: 0.00623F	Magnetic flux [Wb].
Motor Parameter | Rotor inertia (kgm^2) Initial: 0.0000041F	Rotor inertia [kgm^2].

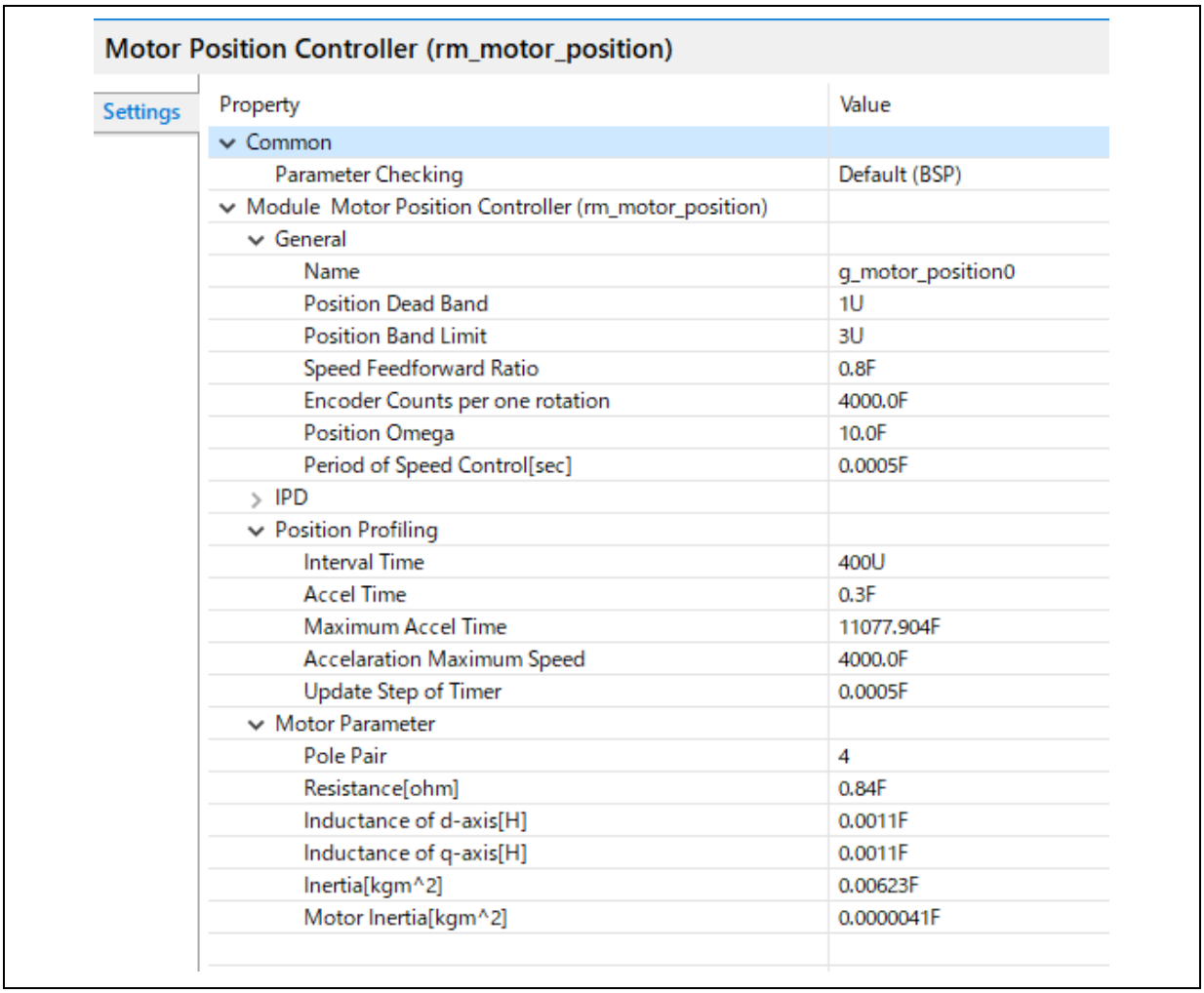


Figure 3-14 FSP Configuration of Position control module (FSP3.5.0)

Table 3-17 Configuration Options for Encoder sensing module

Configuration Options (rm_motor_sense_encoder.h)	
Options	**Description**
Motor Parameter | Pole pairs Initial: 4	Pole pairs of target motor.
Motor Parameter | Resistance (ohm) Initial: 0.84	Resistance of motor [ohm].
Motor Parameter | Inductance of d-axis (H) Initial: 0.0011	D-axis inductance [H].
Motor Parameter | Inductance of q-axis (H) Initial: 0.0011	Q-axis inductance [H].
Motor Parameter | Permanent magnetic flux (Wb) Initial: 0.00623	Magnetic flux [Wb].
Motor Parameter | Rotor inertia (kgm^2) Initial: 0.0000041	Rotor inertia [kgm^2].
Control type Initial: Speed	Selection of speed or position control
Period of current control (kHz) Initial: 20.0F	Period of current control [kHz]
Period of speed control (sec) Initial: 0.0005F	Period of speed control [sec]
PWM carrier frequency (kHz) Initial: 20.0F	PWM carrier frequency [kHz]
Decimation of interrupt Initial: 0U	Decimation counts of interrupt
Counts per rotation Initial: 4000	Encoder counts per one rotation
Counts for angle adjust Initial: 512U	Counts for adjustment of angle initialization
Zero speed counts Initial: 20000000U	Encoder counts to judge zero speed
Occupancy time Initial: 0.4F	Current control occupancy time
Carrier time Initial: 0.00002F	Current control time
Process time Initial: 0.000002F	Encoder count capture processing time
High speed change margin (rpm) Initial: 150U	Margin rotational speed to change high speed mode [rpm]
LPF parameter for high speed filter Initial: 0.1F	LPF parameter for high speed filter
Counts to change speed Initial: 8U	Speed calculation mode change count

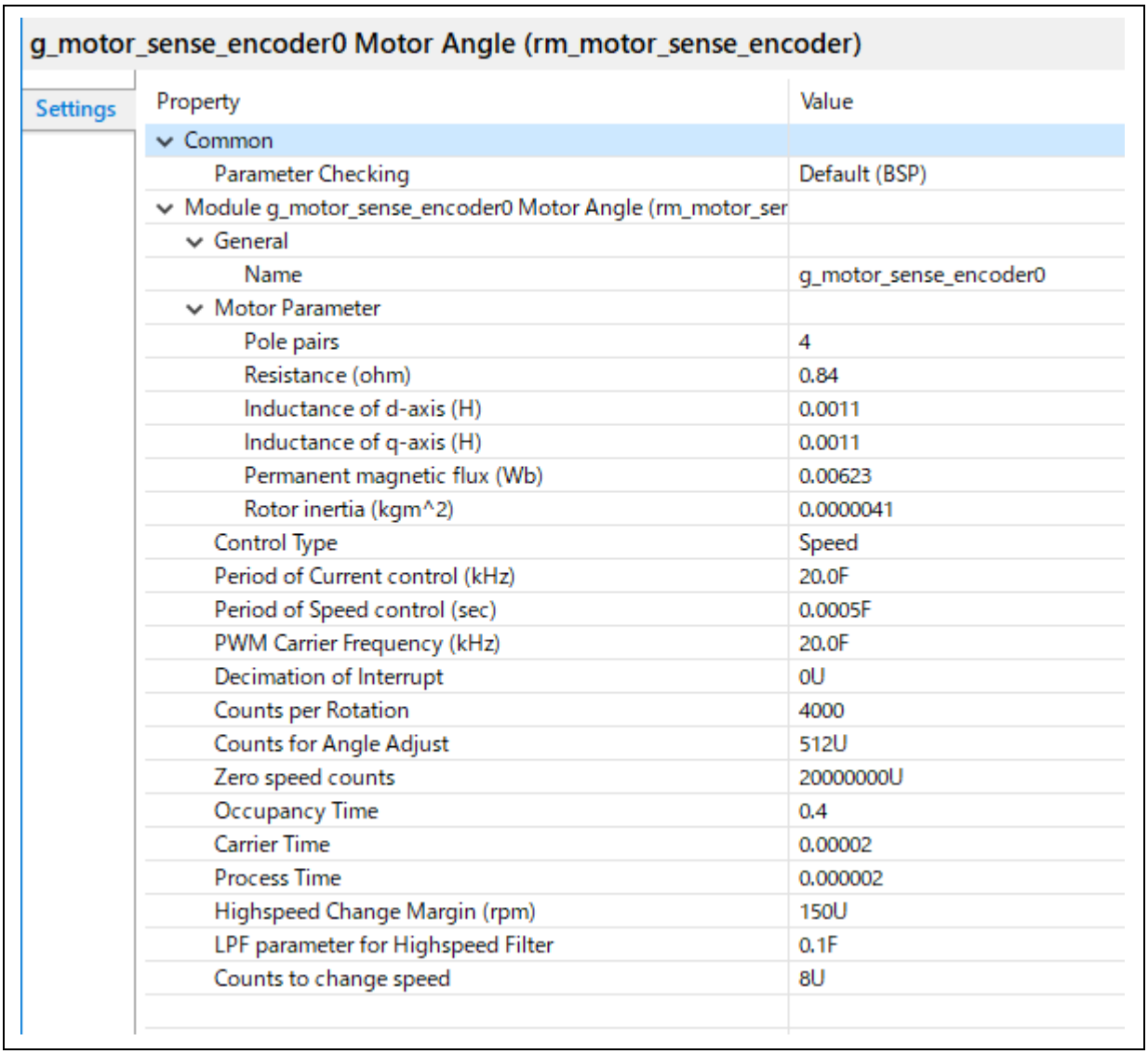


Figure 3-15 FSP Configuration of Encoder sensing module (FSP3.5.0)

Table 3-18 Configuration Options for Driver Access

Configuration Options (rm_motor_driver.h)	
Options	**Description**
Shunt type Initial: 2shunt	Selects how many shunt resistances to use current detection.
PWM timer frequency (MHz) Initial: 120	PWM Timer Clock Frequency [MHz]
PWM carrier period (micro seconds) Initial: 50	PWM Carrier Period [micro seconds]
Dead time (raw counts) Initial: 240	PWM Dead time [raw counts]
Current range (A) Initial: 27.5F	Measurement Range of Electric current [A]
Voltage range (V) Initial: 111.0F	Measurement Range of Inverter Voltage [V]
Counts for current offset measurement Initial: 500	Counts of measurement the offset of A/D Conversion at electric current input.
A/D conversion channel for U phase current Initial: 0	A/D channel for U-phase current
A/D conversion channel for W phase current Initial: 2	A/D channel for W-phase current
A/D conversion channel for main line voltage Initial: 5	A/D channel for main line voltage
A/D conversion channel for V phase current Initial: 1	A/D channel for V-phase current It is invalid at 2shunt detection.
Input voltage Initial: 24.0F	Range of input for main line voltage
Resolution of A/D conversion Initial: 0xFFF	Resolution of A/D conversion Please set same value with ADC module setting.
Offset of A/D conversion for current Initial: 0x745	Offset level of A/D conversion input for current Please set according to the circuit.
Conversion level of A/D conversion for voltage Initial: 0.66F	Conversion level of A/D conversion for voltage Please set when the CPU main voltage is different.
GTIOCA stop level Initial: Pin level High	Output level of upper arm at stop status
GTIOCB stop level Initial: Pin level High	Output level of lower arm at stop status
Maximum duty Initial: 0.9375F	Maximum duty of PWM Maximum duty except dead time.

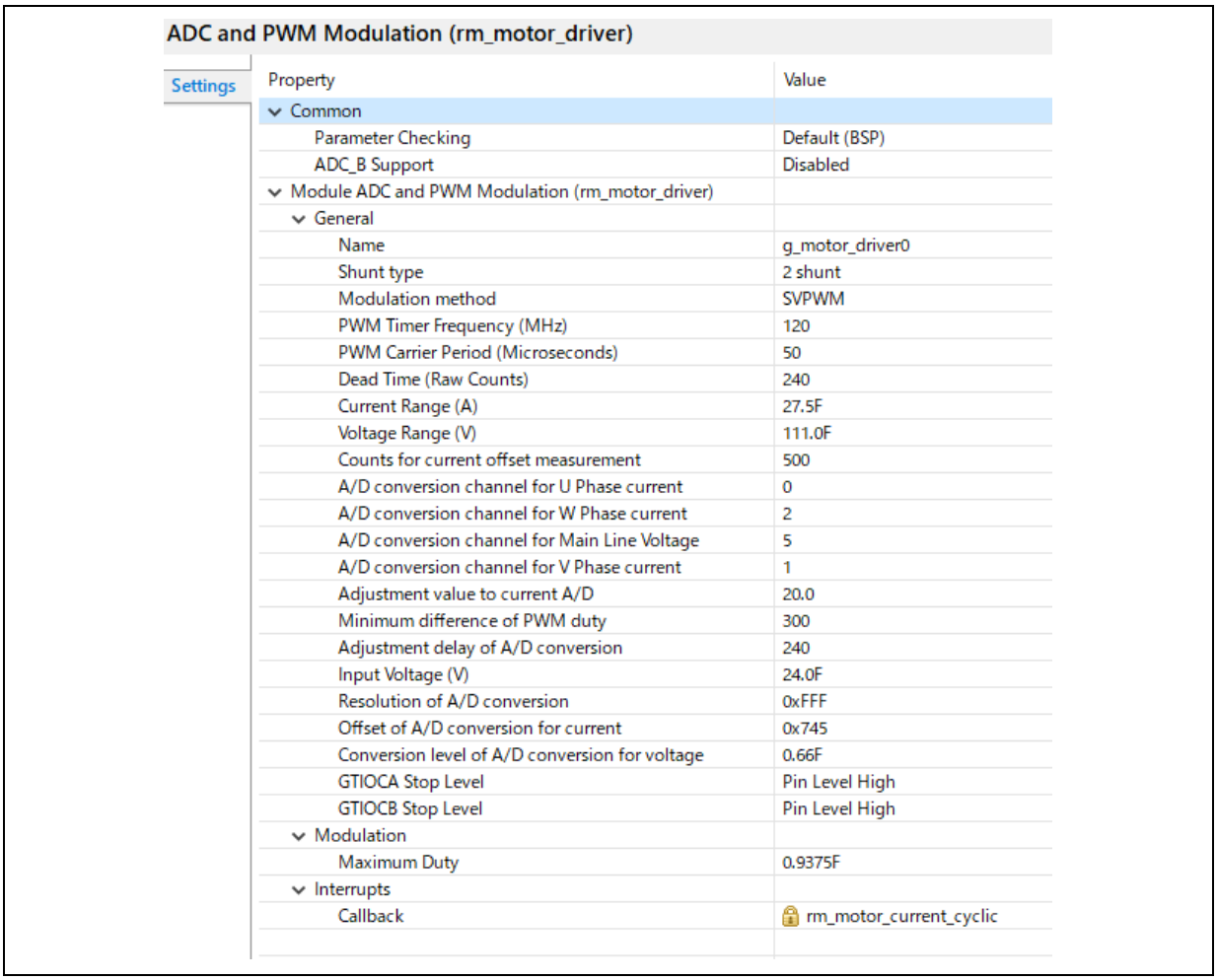


Figure 3-16 FSP Configuration of ADC and PWM Modulation Driver (FSP 3.5.0)

3.4 Control flowcharts

3.4.1 Main process

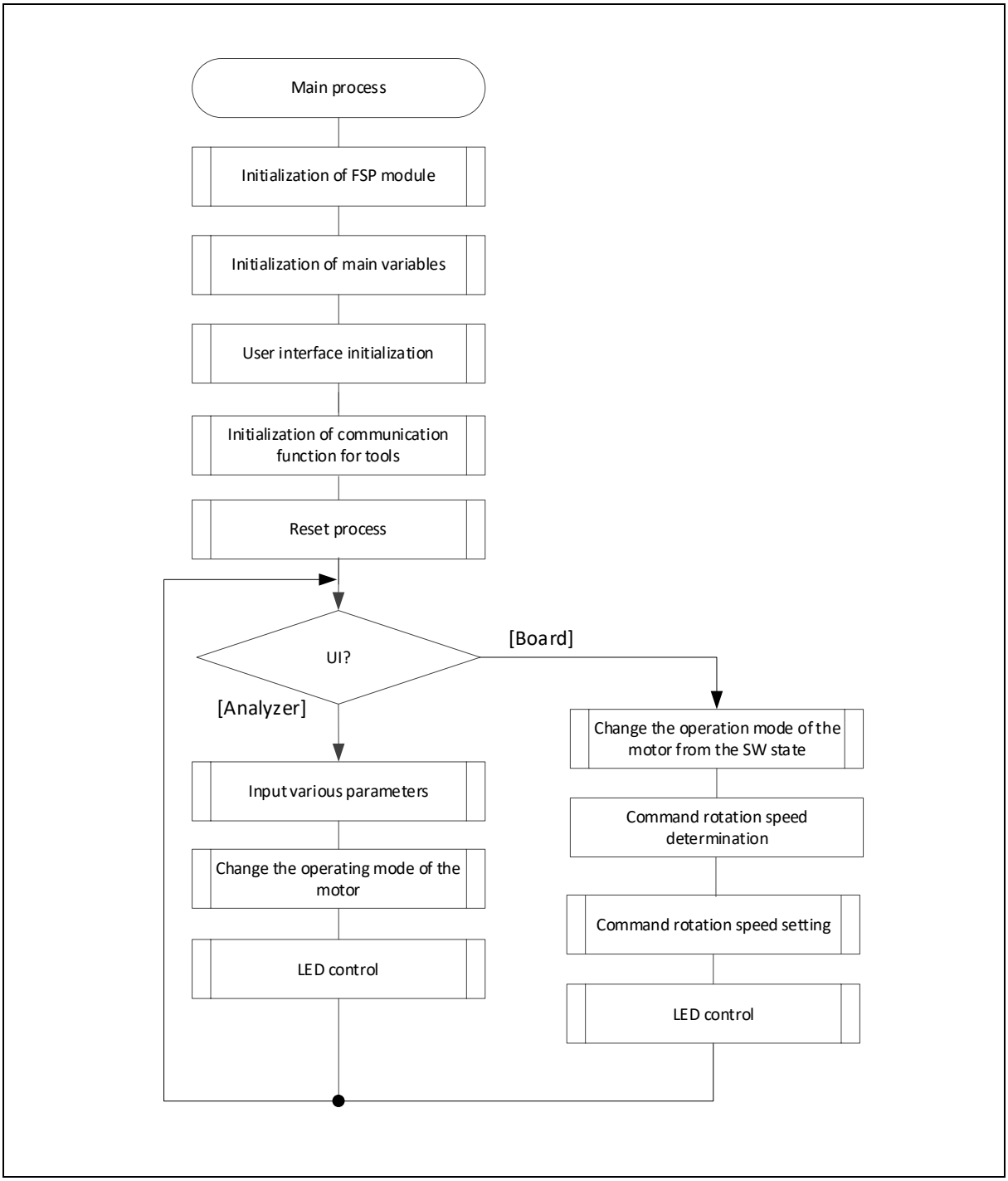


Figure 3-17 Main Process Flowchart

3.4.2 50[µs] Period Interrupt (Carrier synchronized Interrupt) Process

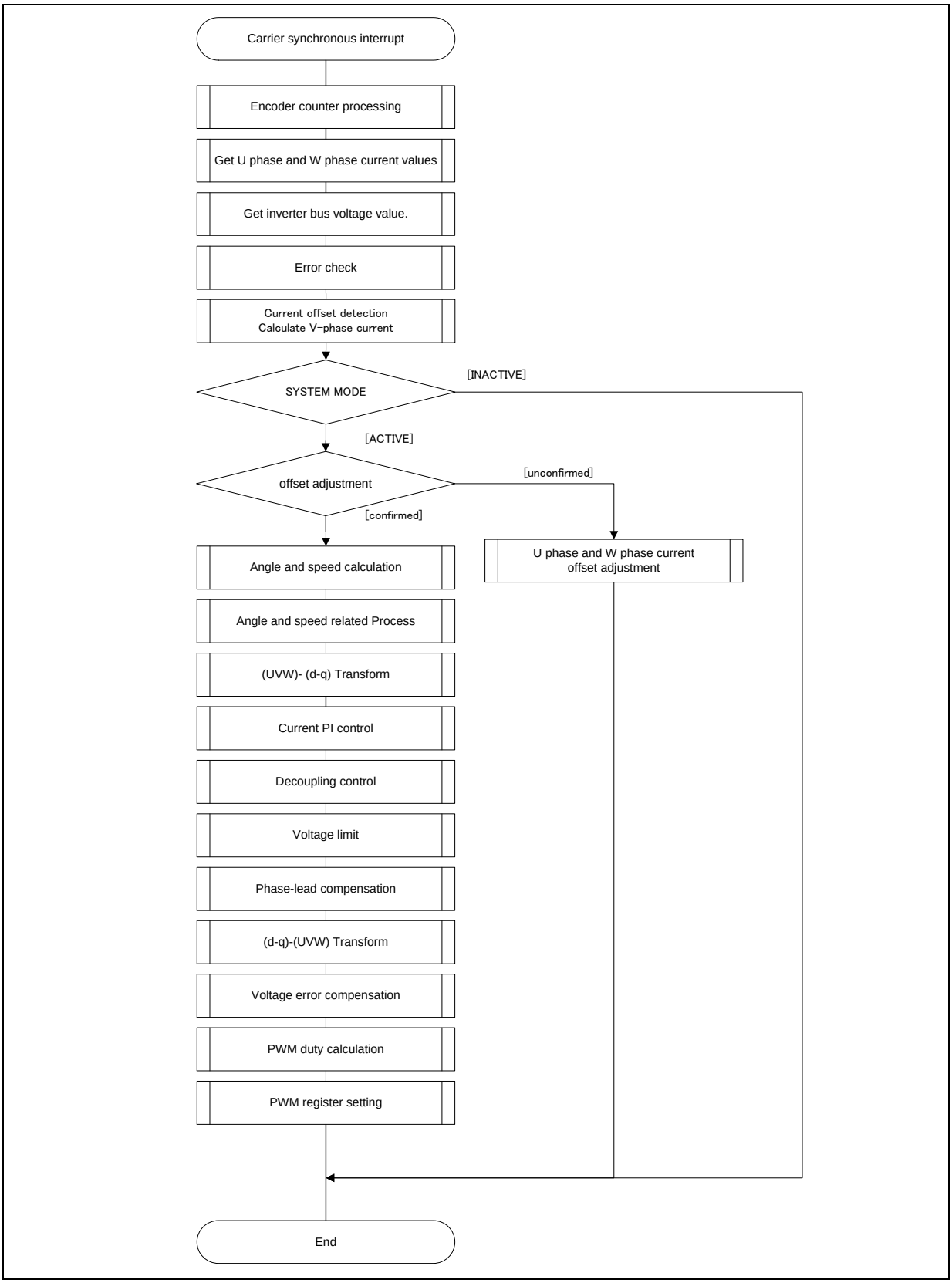


Figure 3-18 50[µs] Period Interrupt (Carrier Interrupt) Process Flowchart

3.4.3 500 [µs] Period Interrupt Process

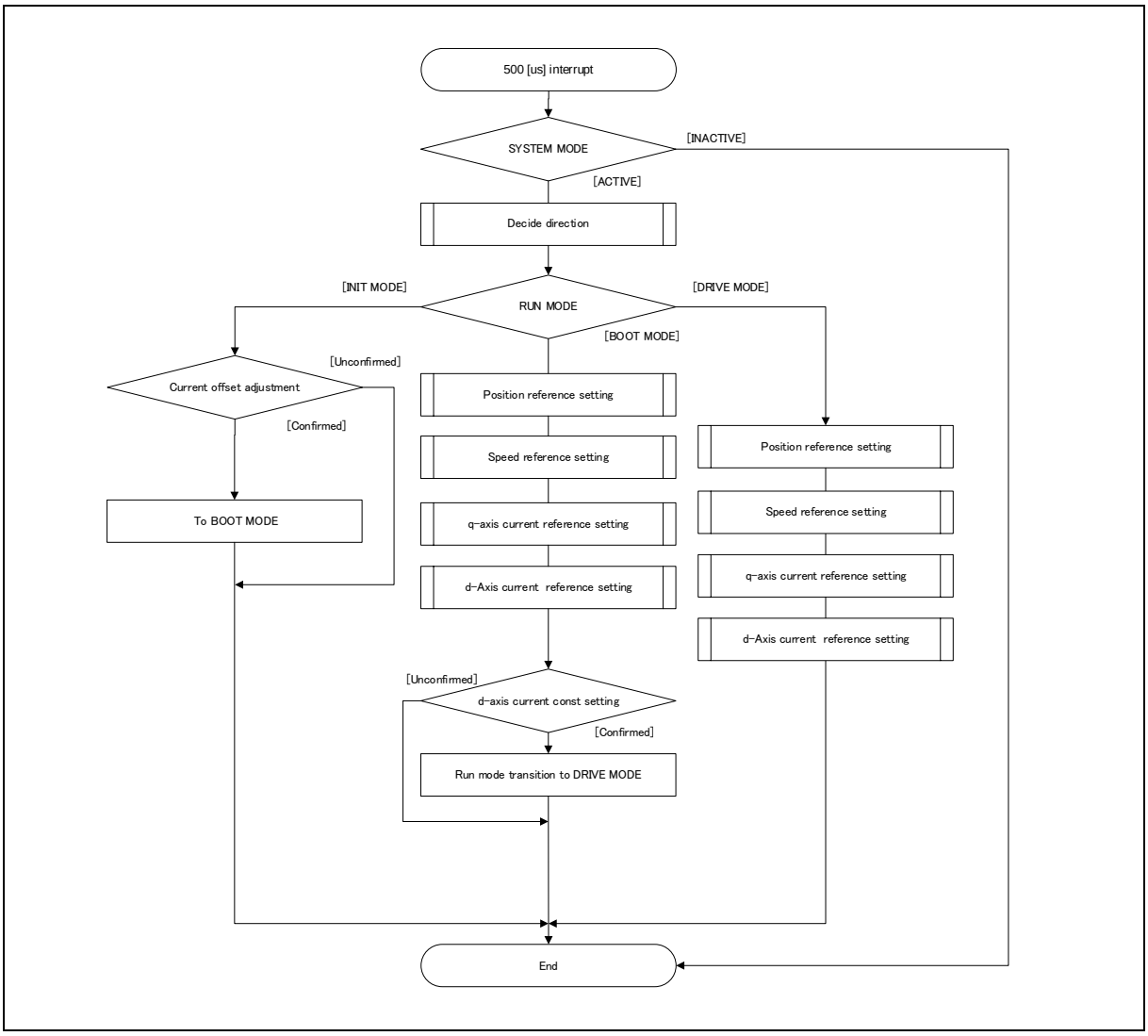


Figure 3-19 500[µs] Period Interrupt Process Flowchart

Flux weakening control can be activated by FSP setting.

3.4.4 Over Current Detection Interrupt Process

The overcurrent detection interrupt is an interrupt that occurs when either an external overcurrent detection signal is input at POEG pin, or an output short circuit is detected by the GTETRGC output level comparison operation. In both cases, the PWM output terminal are put in the high impedance state. Therefore, at the start of execution of this interrupt processing, the PWM output terminal is already in the high impedance state and the output to the motor had been stopped.

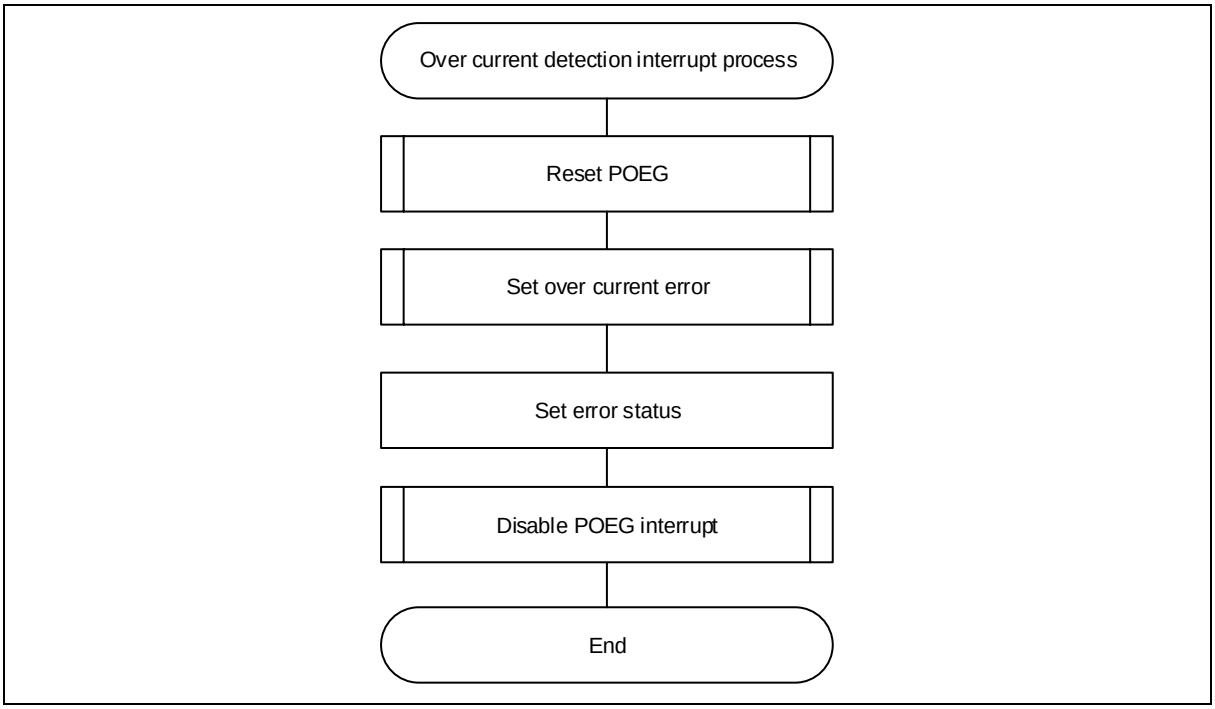


Figure 3-20 Over Current Detection Interrupt Process Flowchart

3.4.5 Encoder signal interrupt processing

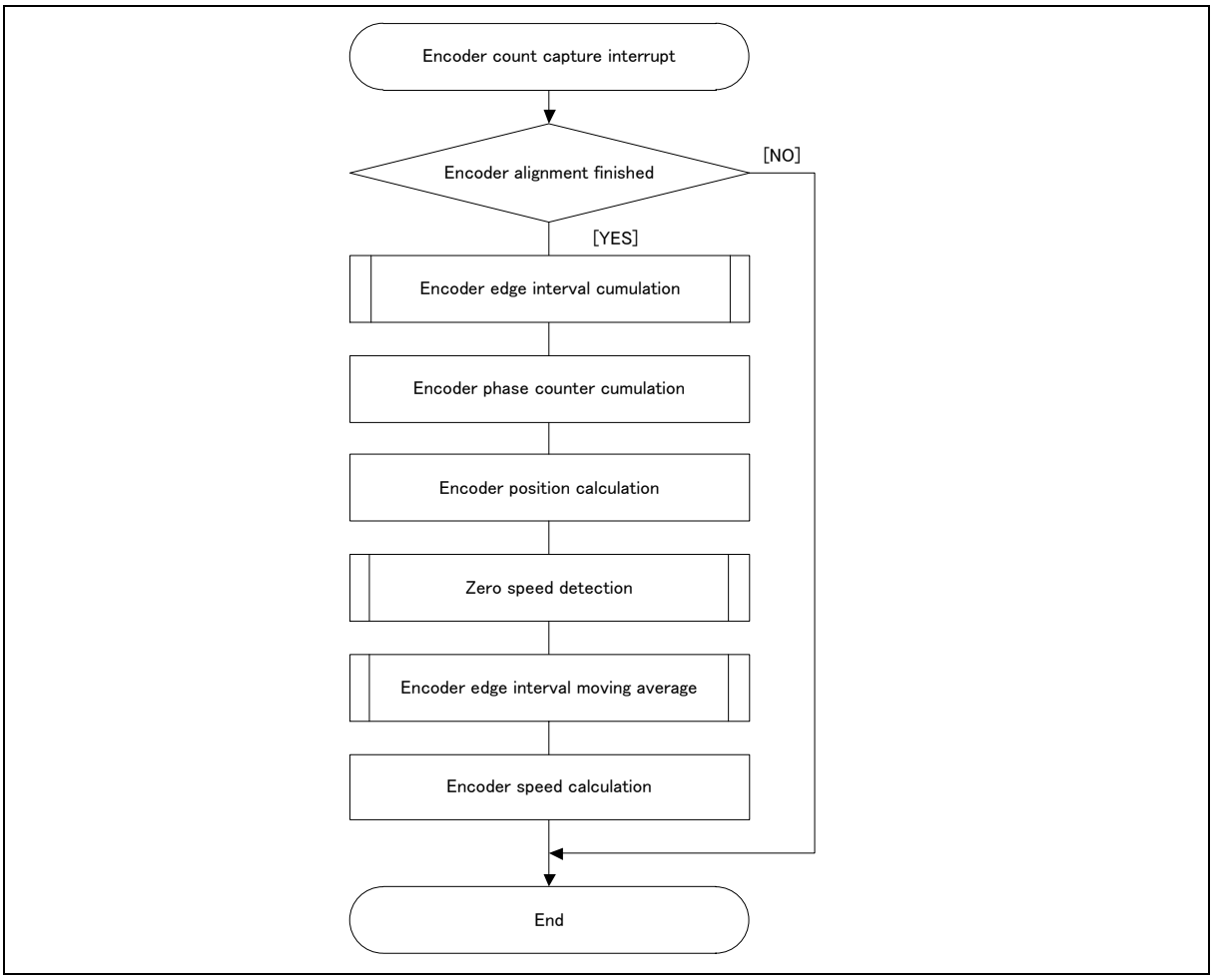


Figure 3-21 Encoder signal interrupt process Flowchart

4. Evaluation environment explanation

4.1 Operating Environment

Table 4-1 shows the hardware requirements for building and debugging Motor control software.

Table 4-1 Hardware Requirements

Hardware	Description
Inverter Board	RA6T1-RSSK [RTK0EM0000B10020BJ]
CPU Card	RA6T1 CPU Card [RTK0EMA170C00000BJ]
Motor	Brushless DC Motor (BLY171D-24V-4000)
On-chip debugging Emulator	The RA6T1 CPU Card has an on-board debugger (J-Link OB), so there is no need to prepare an emulator.

Table 4-2 shows the software requirements for build and debug Motor control software.

Table 4-2 Software Requirements

Software		Version	Description
GCC environment	e²studio	2021-10	Integrated development environment (IDE) for Renesas devices.
	GCC ARM Embedded	V10.3.1.20210824	C/C++ Compiler. (Download from e2studio installer)
	Renesas Flexible Software Package (FSP)	V3.5.0 (or later)	Software package for writing applications for the RA microcontroller series.

4.2 Importing the Demo Project

The sample application provided with this document may be imported into e²studio using the steps in this section.

1. Select File → Import.

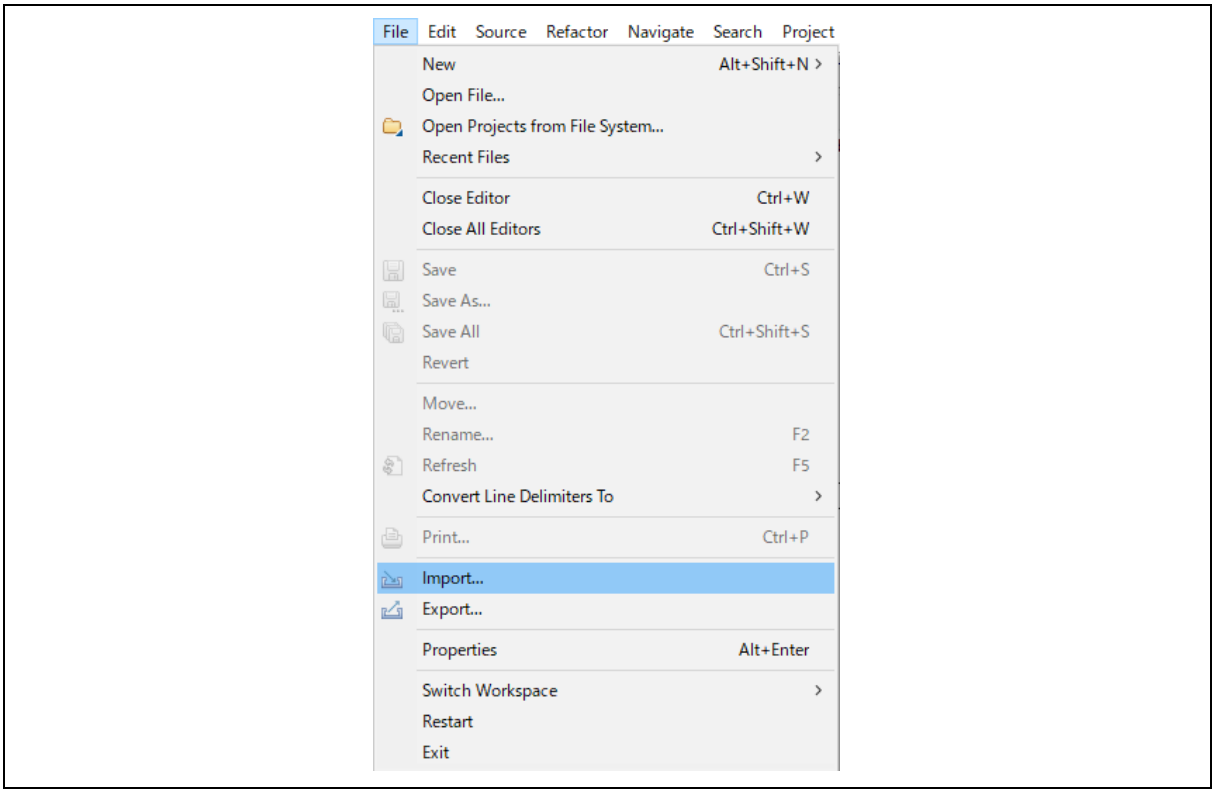


Figure 4-1 File Menu

2. Select “Existing Projects into Workspace”.

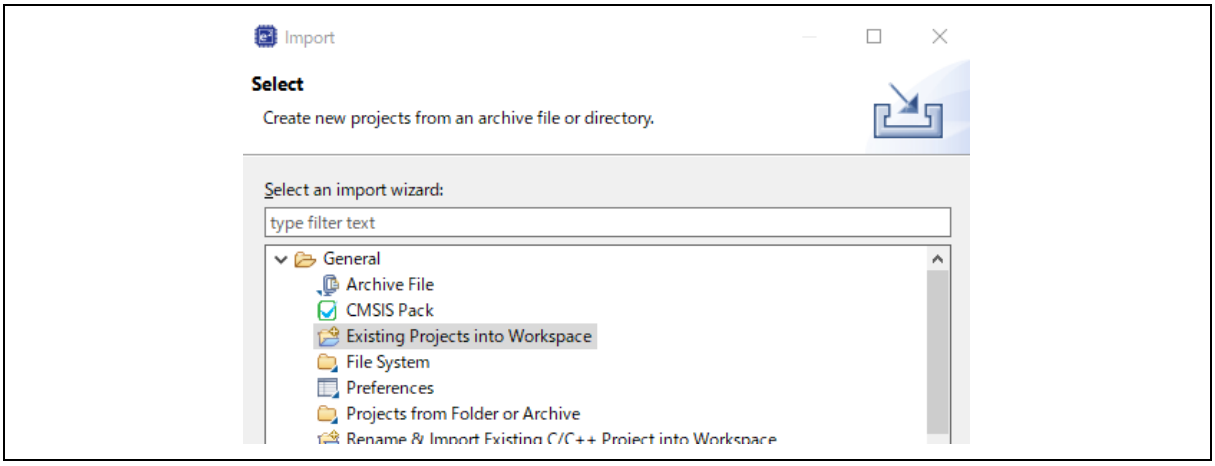


Figure 4-2 Import Wizard Selection

3. Click “Browse…” button and select the demo project. Click Finish button and the demo project is imported.

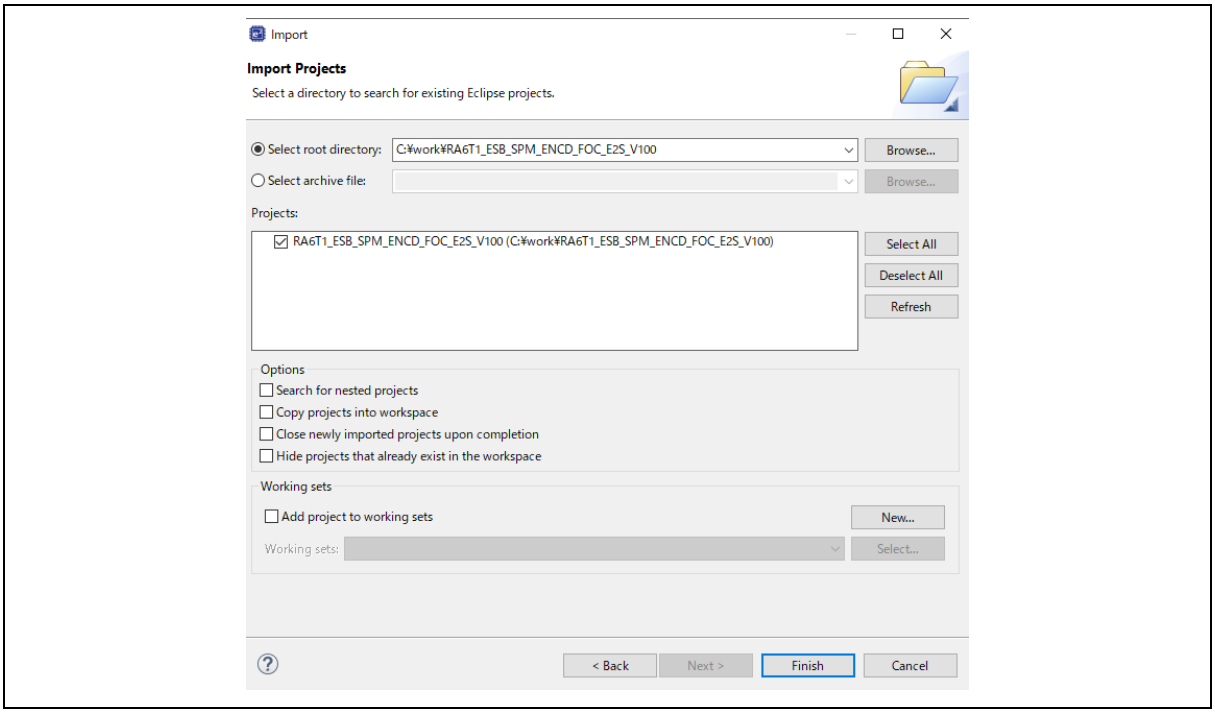


Figure 4-3 Import Projects

4.3 Building and Debugging

Refer to the "e²studio Getting Started Guide (R20UT4204)”.

4.4 Motor Demonstration Project Overview

4.4.1 Quick Start

When executing the sample code only in the evaluation environment without using Renesas motor workbench, the Quick Start Sample Project can be executed with the following procedure.

(1) After turning on stabilized power supply or executing reset, LED1, and LED2 on the inverter board are both off and the motor stops.

(2) IF the toggle switch (SW1) on the inverter board is turned on, the motor starts to rotate. Every time the toggle switch (SW1) is changed, motor rotation starts/stops alternately. If the motor rotates normally, LED1 is on. However, if LED2 on the inverter board is also on, error is occurring.

(3) In order to change the direction of the motor rotation, adjust it with the variable resistor (VR) on the inverter board.

- Turn the variable resistor (VR) right: Motor rotates clockwise
- Turn the variable resistor (VR) left: Motor rotates counterclockwise

(4) If error occurs, LED2 on the inverter board lighten, and the motor rotation stops. To restore, the toggle switch (SW1) on the inverter board needs to be turned off, then the switch (SW2) to be pushed and released.

(5) In order to stop the operation check, turn off the output of the stabilized power supply after making sure that the motor rotation has already stopped

4.5 Motor Control Development Support Tool ‘Renesas Motor Workbench’

4.5.1 Overview

In the target software of this application note, the motor control development support tool “Renesas Motor Workbench” is used as a user interface (rotating/stop motor, set rotation speed reference, etc). Please refer to ‘Renesas Motor Workbench User’s Manual’ for usage and more details.

You can find ’Renesas Motor Workbench’ on Renesas Electronics Corporation website.

Figure 4-4 Renesas Motor Workbench – Appearance

Set up for “Renesas Motor Workbench”

(1) Start ‘Renesas Motor Workbench’ by clicking this icon. 

(2) Drop down menu [File] → [Open RMT File(O)].

And select RMT file in ‘[Project Folder]/src/application/user_interface/ics/’.

(3) Use the ‘Connection’ [COM] select menu to choose the COM port.

(4) Click the Analyzer button of Select Tool to activate Analyzer function.

(5) Please refer to ‘4.5.2 Easy function operation example’ or ‘4.5.4 Operation Example for Analyzer’ for motor driving operation.

4.5.2 Easy function operation example

The following is an example of operating the motor using the Easy function.

4.5.2.1 Position control

- Set the control mode to Position
  (1) Select the Position tab.
  (2) Turn on “Position”.
  If it has already been turned on, turn it off and then turn it on again.

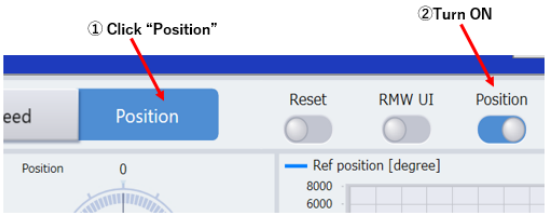


Figure 4-5 Procedure for setting the control mode to Position

- Change the user interface to use Renesas Motor Workbench
  (1) Turn on "RMW UI".
  If it has already been turned on, turn it off and then turn it on again.

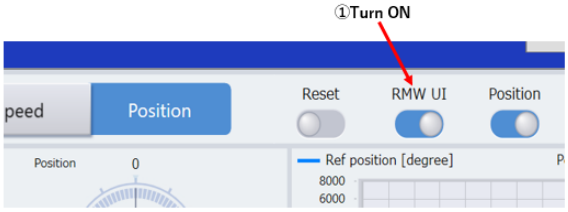


Figure 4-6 Procedure for changing to use Renesas Motor Workbench

- Run the motor
  (1) Press the “Run” button
  (2) Enter the command position with the "Ref position" slider.

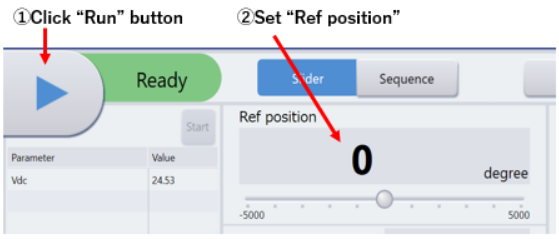


Figure 4-7 Motor rotation procedure

- Stop the motor

  (1) Press the “Stop” button

Figure 4-8 Motor stop procedure

- Processing when it stops (error)

  (1) Turn on "Reset" button.

  (2) Turn off “Reset” button

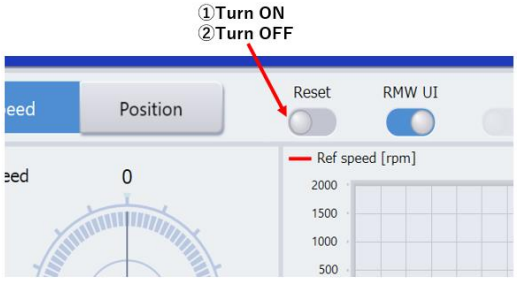


Figure 4-9 Error clearing procedure

4.5.2.2 Speed control

- Set the control mode to Speed
  (3) Select the Speed tab.
  (4) Turn on “Speed”.
  If it has already been turned on, turn it off and then turn it on again.

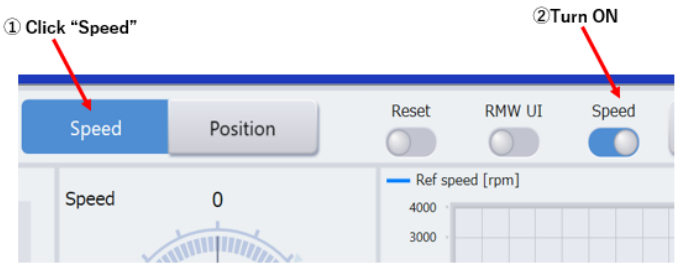


Figure 4-10 Procedure for setting the control mode to Speed

- Change the user interface to use Renesas Motor Workbench
  (2) Turn on "RMW UI".
  If it has already been turned on, turn it off and then turn it on again.

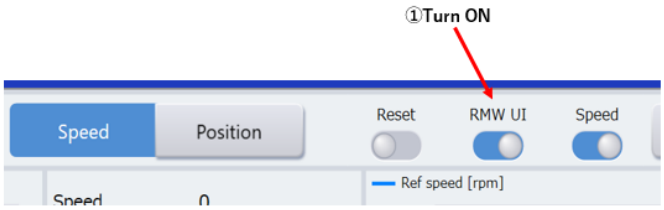


Figure 4-11 Procedure for changing to use Renesas Motor Workbench

- Run the motor
  (3) Press the “Run” button
  (4) Enter the command speed with the "Ref speed" slider.

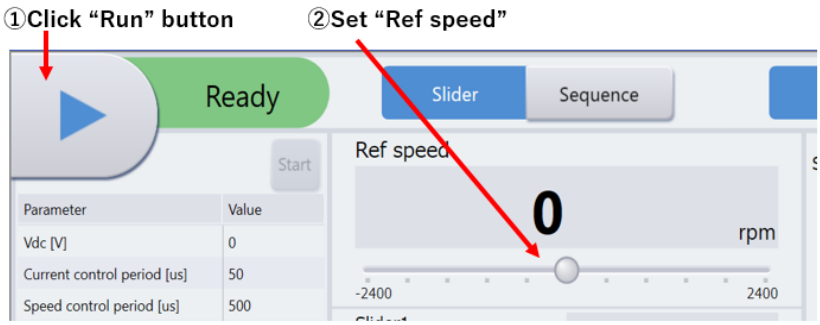


Figure 4-12 Motor rotation procedure

- Stop the motor

  (1) Press the “Stop” button

Figure 4-13 Motor stop procedure

- Processing when it stops (error)

  (1) Turn on "Reset" button.

  (2) Turn off “Reset” button

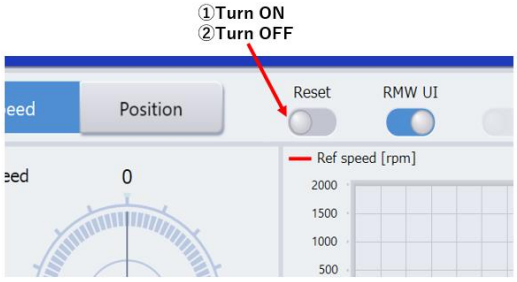


Figure 4-14 Error clearing procedure

4.5.3 List of variables for Analyzer function

Table 4-3 is a list of variables for Analyzer. These variables are reflected to the corresponding variables when the same values as 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 4-3 List of Variables for Analyzer

Variable name	Type	Content
com_u1_sw_userif (*)	uint8_t	User interface switch 0: GUI use 1: Board user interface use (default)
com_u1_mode_system (*)	uint8_t	State management 0: Stop mode 1: Run mode 3: Reset
com_u1_ctrl_loop_mode	uint8_t	Control loop mode switch 0: Speed control 1: Position control (default)
com_f4_ref_speed_rpm	float	Speed reference (Mechanical) [rpm]
com_s2_ref_position_deg	int16_t	Position command value [degree]
com_u2_mtr_pp	uint16_t	Number of pole pairs
com_f4_mtr_r	float	Resistance [Ω]
com_f4_mtr_ld	float	d-axis Inductance [H]
com_f4_mtr_lq	float	q-axis Inductance [H]
com_f4_mtr_m	float	Flux [Wb]
com_f4_mtr_j	float	Inertia [kgm^2]
com_f4_pos_omega	float	Natural frequency of current control system [Hz]
com_f4_sob_omega	float	Damping ratio of current control system
com_f4_sob_zeta	float	Natural frequency of speed control system [Hz]
com_f4_speed_omega	float	Damping ratio of speed control system
com_f4_speed_zeta	float	Natural frequency of the position loop[Hz]
com_f4_current_omega	float	Natural frequency of the speed observer [Hz]
com_f4_current_zeta	float	Damping ratio of the speed observer
com_f4_ol_ref_id	float	d-axis current reference in open loop mode [A]
com_f4_id_up_time	float	d-axis current command value addition time [ms]
com_f4_max_speed_rpm	float	Maximum speed
com_f4_speed_limit_rpm	float	Over speed limit
com_u2_pos_dead_band	uint16_t	Dead band of position
com_u2_pos_band_limit	uint16_t	Positioning complete range
com_u2_encd_cpr	uint16_t	Encoder pulse count (4 for multiplying)
com_u2_interval_time	uint16_t	Time interval of the position command changes
com_f4_accel_time	float	Acceleration time [s] (for position control)
com_f4_speed_rate_limit	float	Acceleration limit [s] (for speed control)
com_f4_overcurrent_limit	float	Over current limit
com_f4_iq_limit	float	q-axis current limit
com_u1_enable_write	uint8_t	Enable to rewriting variables

4.5.4 Operation Example for Analyzer

Following example shows motor driving operation using Analyzer. Operation is using “Control Window” as shown in Figure 4-4. Regarding specification of “Control Window”, refer to ‘Renesas Motor Workbench User’s Manual’.

- Change the user interface to Analyzer

(1) Confirm the check-boxes of column [W?] for ‘com_u1_sw_userif’ marks.
(2) Input ‘0’ in the [Write] box of ‘com_u1_sw_userif’.
(3) Click the ‘Write’ button.

- Driving the motor

(1) The [W?] check boxes contain checkmarks for “com_u1_mode_system1”,“com_s2_ref_position_deg”, “com_u1_enable_write”
(2) Type a reference position value in the [Write] box of “com_s2_ref_position_deg”.
(3) Click the “Write” button.
(4) Click the “Read” button. Confirm the [Read] box of “com_s2_ref_position_deg” , ”g_u1_enable_write”.
(5) Enter the same value of “g_u1_enable_write” in the [Write] box of “com_u1_enable_write”.
(6) Enter “1” in the [Write] box of “com_u1_mode_system”.
(7) Click the “Write” button.

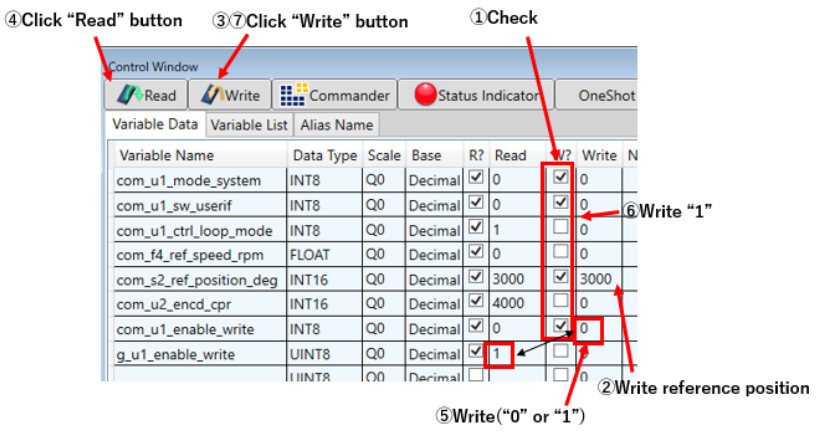


Figure 4-15 Procedure - Driving the motor

- Stop the motor

(1) Enter “0” in the [Write] box of “com_u1_mode_system”.
(2) Click the “Write” button.

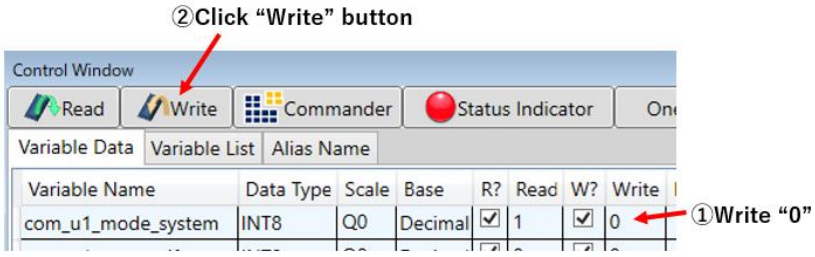


Figure 4-16 Procedure - Stop the motor

- Error cancel operation

(1) Enter “3” in the [Write] box of “com_u1_mode_system”.
(2) Click the “Write” button.

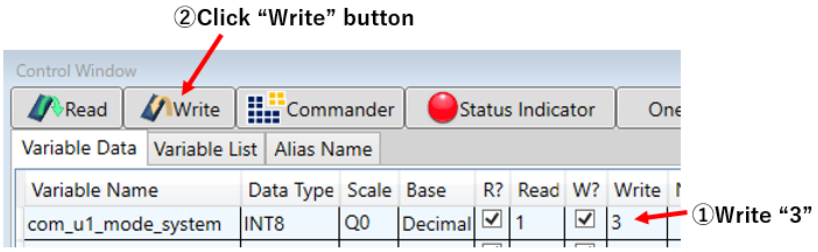


Figure 4-17 Procedure - Error cancel operation

4.5.5 Tuner function

To use the Tuner function, use the executable file provided by Renesas Motor Workbench.
For details on how to use the Tuner function, refer to the Tuner function manual (R20AN0528) included in the Renesas Motor Workbench download file.

5. Reference Documents

RA6T1 Group User's Manual: Hardware (R01UH0897)
RA Flexible Software Package Documentation
Application note: Vector control with encoder for permanent magnet synchronous motor (Algorithm)' (R01AN3789)
Renesas Motor Workbench User's Manual (R21UZ0004)
Evaluation System for BLDC Motor User's Manual (R12UZ0062)
Motor Control Evaluation System for RA Family (R12UZ0078)
RA6T1 CPU CARD User's Manual (R12UZ0077)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun 29, 2021	-	First release document
1.01	Sep 24, 2021	23	Addition of 3.5 chapters
1.10	Dec.17, 2021	-	For FSP V3.5.0
1.11	Oct.18, 2022	-	For FSP V4.1.0

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. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. 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 the 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.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.