RA6T1

120-degree conducting control of permanent magnetic synchronous motor using hall sensors

Summary

This application note describes the sample program for a permanent magnetic synchronous motor drive with 120- degree conducting method using hall sensors 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 sample programs have been checked by using the following device.

- RA6T1 (R7FA6T1AD3CFP)

Contents

1. Overview 3
1.1 Development environment 3

2. System overview 4
2.1 Hardware configuration 4
2.2 Hardware specifications 5
2.2.1 User interface 5
2.2.2 Peripheral functions 7
2.3 Software structure 13
2.3.1 Software file structure 13
2.3.2 Module configuration 14
2.4 Software specifications 15
2.5 Interrupt priority 16

3. Descriptions of the control software 17
3.1 Contents of control 17
3.1.1 Motor start / stop 17
3.1.2 A/D converter 17
3.1.3 Speed control 18
3.1.4 Voltage control by PWM 19
3.1.5 State transition 21
3.1.6 Start-up method 22
3.1.7 Sytem protection function 23
3.1.8 AD triggers 24
3.2 Function specifications of 120-degree conducting control software using hall sensors software 25
3.3 Contents of control 36

3.3.1 Configuration Options ... 36
3.3.2 Configuration Options for included modules ... 36
3.4 Control flows (flow charts) ... 41
3.4.1 Main process ... 41
3.4.2 50 [µs] Periodic interrupt (carrier periodic interrupt) handling ... 42
3.4.3 1 [ms] periodic interrupt handling ... 43
3.4.4 Overcurrent interrupt handling ... 44

4. Evaluation environment explanation ... 45
4.1 Operating environment ... 45
4.2 Project import ... 46
4.3 Build and debug ... 47
4.4 Operation outline of sample software ... 48
4.4.1 Quick start ... 48
4.5 Motor Control Development Support Tool, 'Renesas Motor Workbench' ... 49
4.5.1 Overview ... 49
4.5.2 Easy function operation example ... 50
4.5.3 List of variables for Analyzer function ... 52
4.5.4 Operation Example for Analyzer ... 53

5. Reference document ... 55

1. Overview

This application note explains how to implement the 120-degree conducting control sample program of permanent magnetic synchronous motor (PMSM) using hall sensors based on the RA6T1 microcontroller and how to use the motor control development support tool, 'Renesas Motor Workbench'. Note that these sample programs use the algorithm described in the application note '120-degree conducting control of permanent magnetic synchronous motor: algorithm'.

1.1 Development environment

Table 1-1 and Table 1-2 show the development environment of software targeted by this application note.

Table 1-1 Hardware development environment

Microcontroller	Evaluation board (Note 1)	Motor (Note 2)
RA6T1 (R7FA6T1AD3CFP)	48V inverter board (2 unit) RA6T1 CPU Card	TG-55L-KA 24V

Table 1-2 Software development environment

e^{2}studio version	FSP version	Toolchain version
V2021-10	Since V3.5.0	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. TG-55L KA,24V is a product of TSUKASA ELECTRIC.

TSUKASA ELECTRIC (https://www.tsukasa-d.co.jp/en/)

2. System overview

Overview of this system is explained below.

2.1 Hardware configuration

The hardware configuration is shown below.

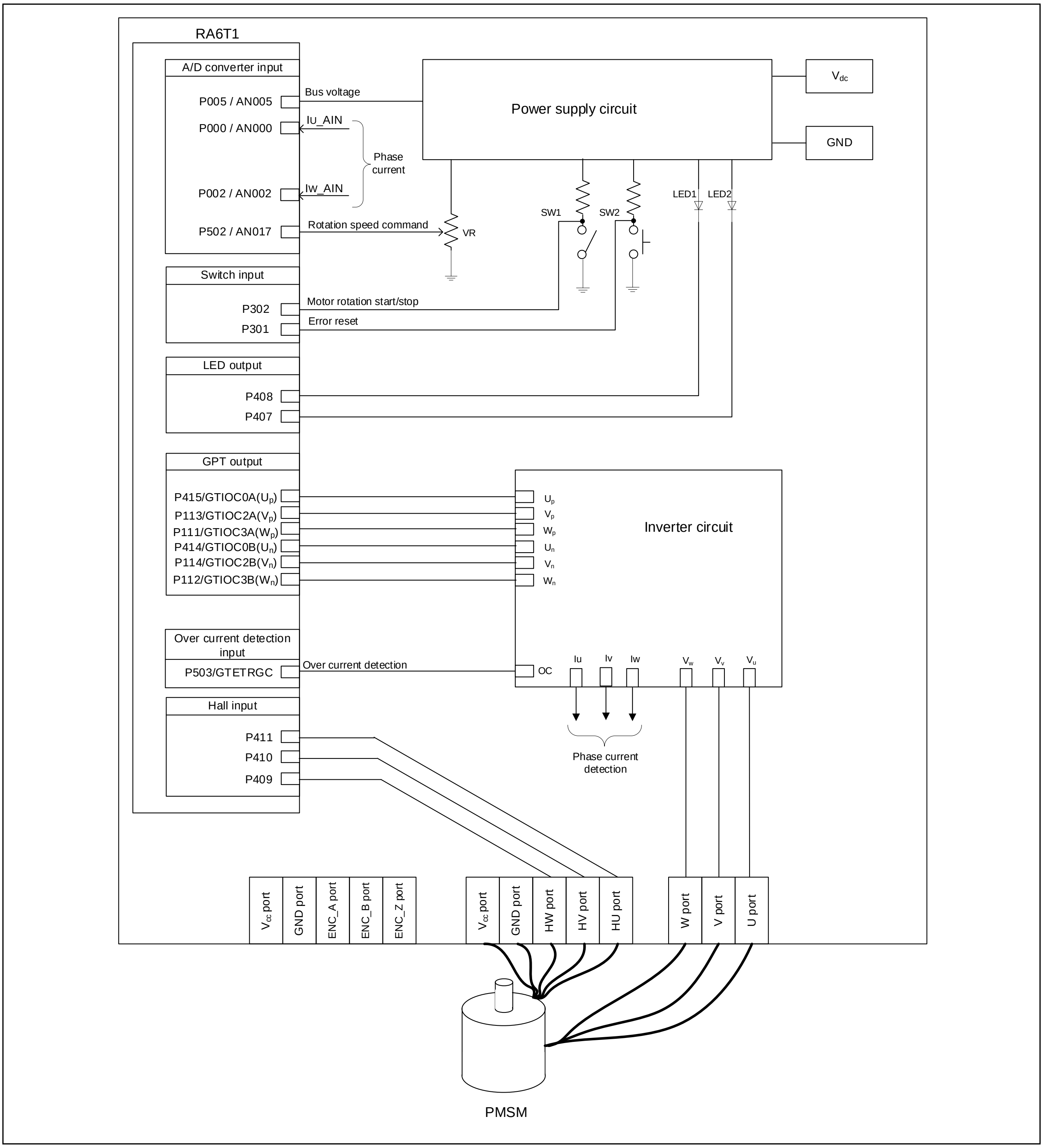


Figure 2-1 Hardware Configuration Diagram

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 Inverter board user interface

Item	Interface component	Function
Rotation speed	Variable resistance(VR1)	Rotation speed command value input(analog values)
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
LED3	Orange LED	Not used in this system

Talbe 2-2 CPU card user interface

Item	Interface component	Function
LED1	Yellow green LED	- At the time of Motor rotation : ON - At the time of stop : OFF
LED2	Yellow green LED	- At the time of error detection : ON - At the time of normal operation : OFF
RESET	Push switch(SW1)	System reset

Table 2-3 is a list of port interfaces of RA6T1 microcontroller of this system.

Table 2-3 Port Interface

R7FA6T1AD3CFP Port name	Function
P014 / AN005	Inverter bus voltage measurement
P502 / AN017	For inputting rotation speed command values (analog values)
P302	START/STOP toggle switch
P301	ERROR RESET push switch
P408	LED1 ON/OFF control
P407	LED2 ON/OFF control
P000 / AN000	U 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
P411 / IRQ4	Hall sensor input (HU)
P410 / IRQ5	Hall sensor input (HV)
P409 / IRQ6	Hall sensor input (HW)

2.2.2 Peripheral functions

Table 2-4 is a list of peripheral functions used in this system.

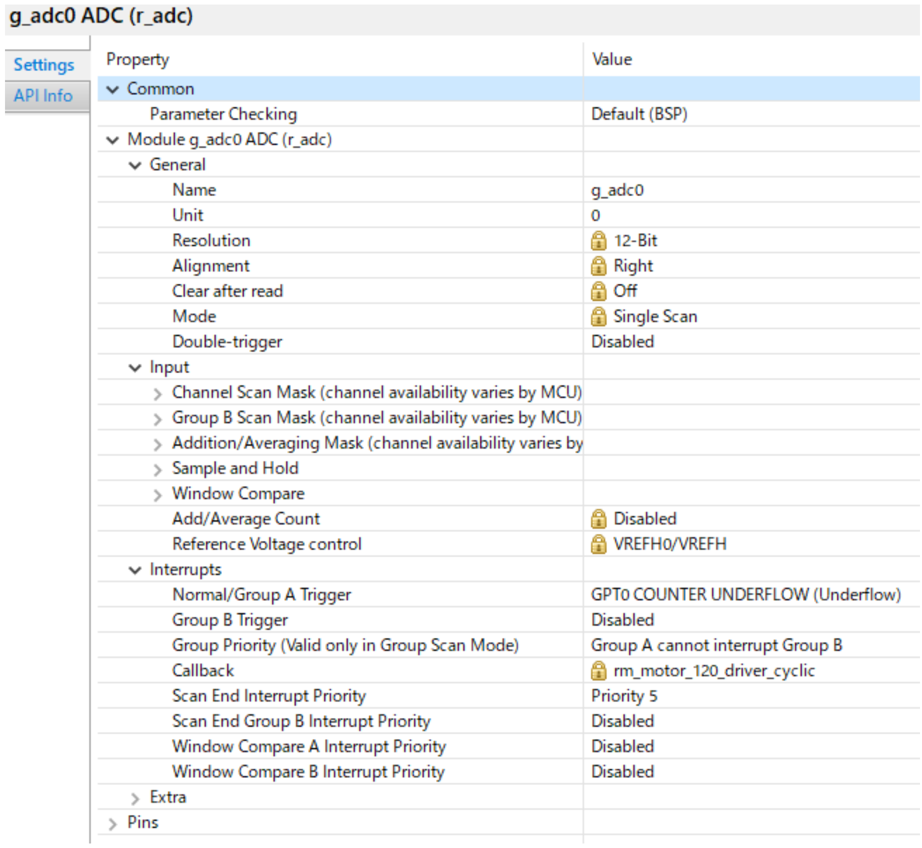
Table 2-4 Peripheral Functions List

Peripheral Function	Resource	Use
12-bit A/D converter	AN000, AN002, AN005, AN017	- Rotation speed command value input - Current of each phase U, V, and W - Inverter bus voltage measurement
AGT	AGT0	1ms interval timer
GPT	CH0, CH1, CH2, CH3	- Complementary PWM outputs - Free run timer for rotation speed measurement
POEG	Group C	Sets ports executing PWM output to high impedance state when an overcurrent is detected.
External interrupt (IRQ)	CH4, CH5, CH6	External interrupt by hall sensors’ signals (both edge)

(1). 12-bit A/D converter (ADC12)

U-phase current (Iu), W-phase current (Iw), the rotation speed command value input, inverter bus voltage(Vdc) are measured by using the '12 bit A/D converter'.

The operation modes must be set to the 'Signle scan mode'(use a hardware trigger).



g_adc0 ADC (r_adc)

Settings
API Info

Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_adc0 ADC (r_adc)	
General	
Name	g_adc0
Unit	0
Resolution	12-Bit
Alignment	Right
Clear after read	Off
Mode	Single Scan
Double-trigger	Disabled
Input	
Channel Scan Mask (channel availability varies by MCU)	
Group B Scan Mask (channel availability varies by MCU)	
Addition/Averaging Mask (channel availability varies by	
Sample and Hold	
Window Compare	
Add/Average Count	Disabled
Reference Voltage control	VREFH0/VREFH
Interrupts	
Normal/Group A Trigger	GPT0 COUNTER UNDERFLOW (Underflow)
Group B Trigger	Disabled
Group Priority (Valid only in Group Scan Mode)	Group A cannot interrupt Group B
Callback	rm_motor_120_driver_cyclic
Scan End Interrupt Priority	Priority 5
Scan End Group B Interrupt Priority	Disabled
Window Compare A Interrupt Priority	Disabled
Window Compare B Interrupt Priority	Disabled
Extra	
Pins	

Figure 2-2 FSP configuration of ADC driver (FSP3.5.0)

(2). Low power consumption asynchronous general-purpose timer (AGT)

Used as a 1ms interval taimer.

g_timer3 Timer, Low-Power (r_agt)

Settings | API Info

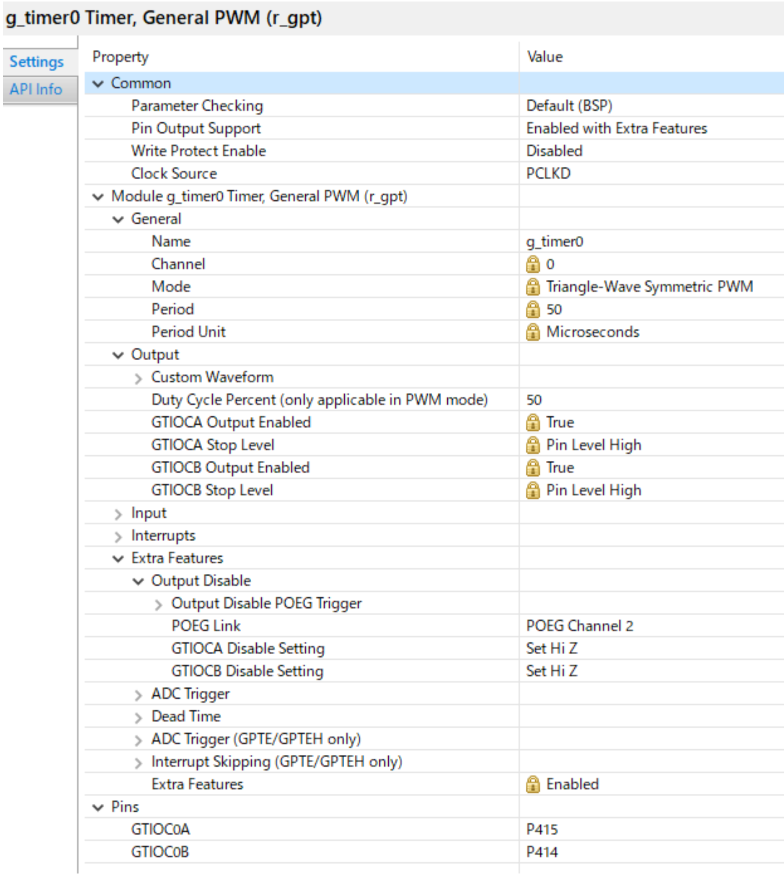
Property	Value
∨ Common	
Parameter Checking	Default (BSP)
Pin Output Support	Disabled
Pin Input Support	Disabled
∨ Module g_timer3 Timer, Low-Power (r_agt)	
∨ General	
Name	g_timer3
Channel	0
Mode	Periodic
Period	1
Period Unit	Milliseconds
Count Source	PCLKB
› Output	
› Input	
› Interrupts	
› Pins	

Figure 2-3 FSP configuration of AGT driver (FSP3.5.0)

(3). General-purpose PWM timer (GPT)

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

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



g_timer0 Timer, General PWM (r_gpt)

Settings | API Info

Property	Value
Common	
Parameter Checking	Default (BSP)
Pin Output Support	Enabled with Extra Features
Write Protect Enable	Disabled
Clock Source	PCLKD
Module g_timer0 Timer, General PWM (r_gpt)	
General	
Name	g_timer0
Channel	0
Mode	Triangle-Wave Symmetric PWM
Period	50
Period Unit	Microseconds
Output	
Custom Waveform	
Duty Cycle Percent (only applicable in PWM mode)	50
GTIOCA Output Enabled	True
GTIOCA Stop Level	Pin Level High
GTIOCB Output Enabled	True
GTIOCB Stop Level	Pin Level High
Input	
Interrupts	
Extra Features	
Output Disable	
Output Disable POEG Trigger	
POEG Link	POEG Channel 2
GTIOCA Disable Setting	Set Hi Z
GTIOCB Disable Setting	Set Hi Z
ADC Trigger	
Dead Time	
ADC Trigger (GPTE/GPTEH only)	
Interrupt Skipping (GPTE/GPTEH only)	
Extra Features	Enabled
Pins	
GTIOC0A	P415
GTIOC0B	P414

Figure 2-4 FSP configuration of GPT driver(complementary PWM output) (FSP3.5.0)

g_timer4 Timer, General PWM (r_gpt)

Settings | API Info

Property	Value
Common	
Parameter Checking	Default (BSP)
Pin Output Support	Enabled with Extra Features
Write Protect Enable	Disabled
Clock Source	PCLKD
Module g_timer4 Timer, General PWM (r_gpt)	
General	
Name	g_timer4
Channel	1
Mode	Periodic
Period	0x100000000
Period Unit	Raw Counts
Output	
Input	
Interrupts	
Extra Features	
Pins	
GTIOC1A	<unavailable>
GTIOC1B	<unavailable>

Figure 2-5 FSP configuration of GPT driver (free run timer for speed measurement) (FSP3.5.0)

(4). Port output enable for GPT(POEG)

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

g_poeg0 Port Output Enable for GPT (r_poeg)

Settings | API Info

Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_poeg0 Port Output Enable for GPT (r_poeg)	
General	
Trigger	
Name	g_poeg0
Channel	2
Input	
Interrupts	
Callback	g_poe_overcurrent
Interrupt Priority	Priority 0 (highest)
Pins	
GTETRGC	P503

Figure 2-6 FSP configuration of POEG driver (FSP3.5.0)

(5). External interrup (IRQ)

The hall sensors' signals are inputted for detection of rotor position.

Both edge mode is used.

When the interrupt occurs, following operations are performed

- detection of rotor position
- rotation speed measurement
- conduction pattern chabge

g_external_irq0 External IRQ (r_icu)

Settings | API Info

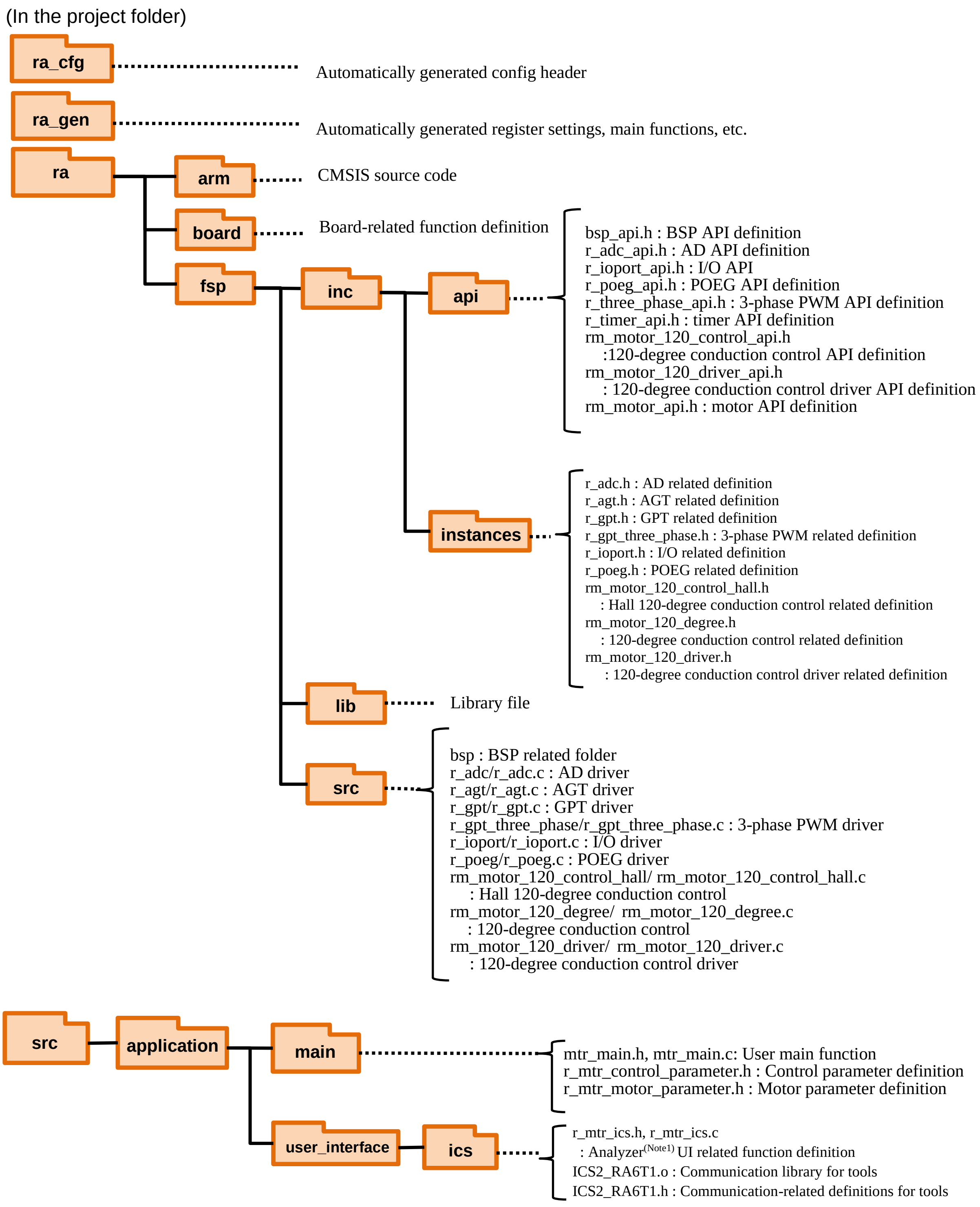
Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_external_irq0 External IRQ (r_icu)	
Name	g_external_irq0
Channel	4
Trigger	Both Edges
Digital Filtering	Disabled
Digital Filtering Sample Clock (Only valid when Digital Filte	PCLK / 64
Callback	rm_motor_120_control_hall_interrupt
Pin Interrupt Priority	Priority 3
Pins	
IRQ04	P411

Figure 2-7 FSP configuration of IRQ driver (FSP3.5.0)

2.3 Software structure

2.3.1 Software file structure

The folder and file configurations of the sample programs are given below.



Note1 : For details on the Analyzer function of the motor control development support tool "Renesas Motor Workbench", refer to Chapter 4.5.
Also, for the names of folders, files, functions, and variables related to the motor control development support tool "Renesas Motor Workbench", The identifier "ics / ICS (formerly Renesas Motor Control Development Support Tool" In Circuit Scope ")" may be added.

Figure 2-8 Folder and file structure

2.3.2 Module configuration

Figure 2-9 show module configuration of software.

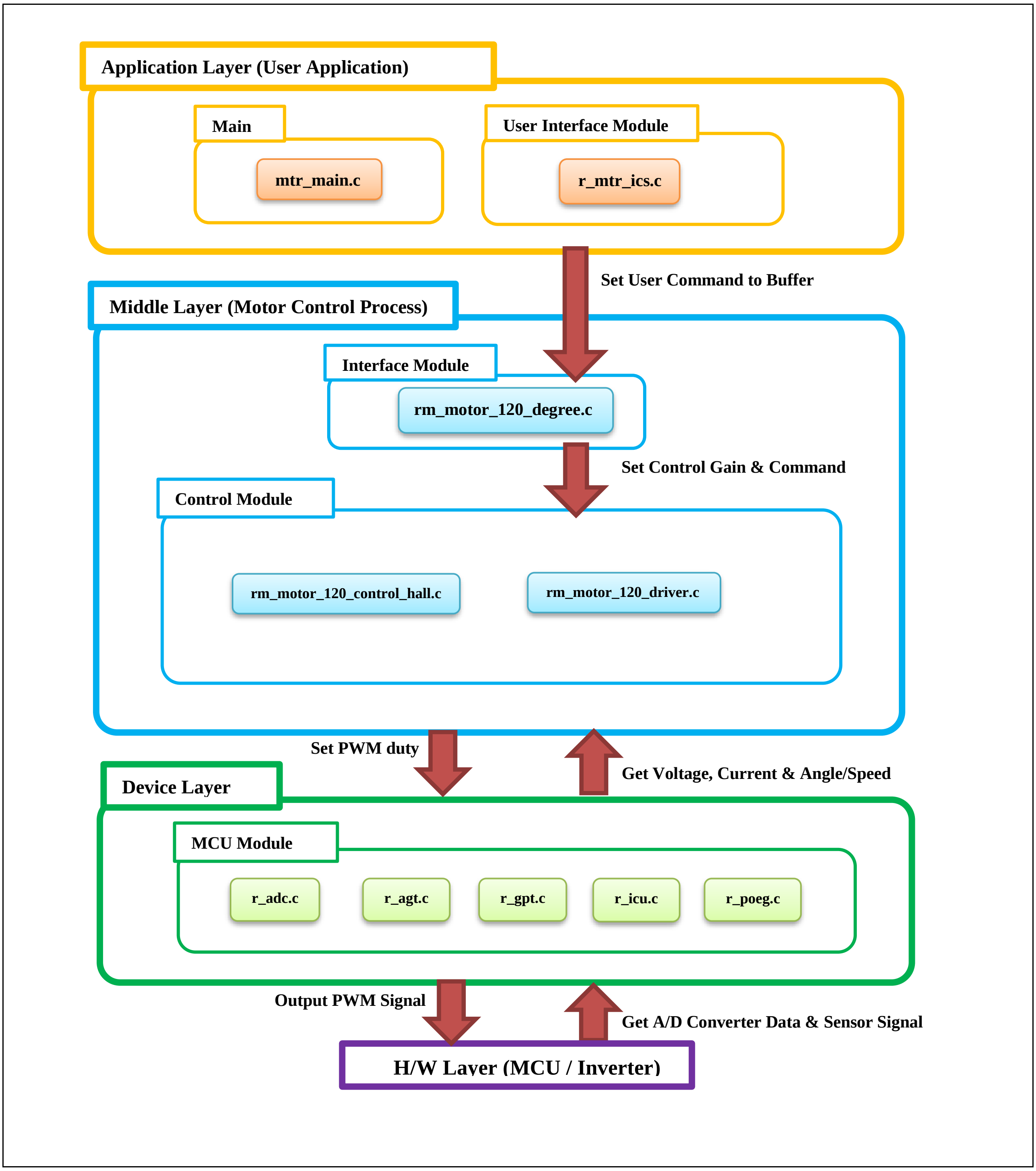


Figure 2-9 Module Configuration

2.4 Software specifications

Table 2-5 shows the basic specifications of target software of this application note. For details of 120-degree conducting control, refer to the application note '120-degree conducting control of permanent magnetic synchronous motor: algorithm' (R01AN2657)

Table 2-5 Basic Specifications of Software

Item	Content	
Control method	120-degree conducting control	
Rotor magnetic pole position detection	Position detection by signals of hall sensors (by each 60 degrees)	
Motor rotation start/stop	SW1 input or input from 'Renesas Motor Workbench'	
Input voltage	DC 24V	
Main clock frequency	120 [MHz]	
Carrier frequency (PWM)	20 [kHz] (Carrier cycle：50 [μs])	
dead time	2 [μs]	
Control frequency	Speed control：1 [ms]	
Rotation speed control range	Both CW and CCW: 550 [rpm] to 2650 [rpm]	
Compiler optimization settings	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.Current of any phase exceeds 0.89(=0.42*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 3000 [rpm] (monitored every 50 [μs]) 5.During motor rotation, the interrupt of hall sensors' signals is not detected for 200 [ms]. 6.Fault detection of hall sensor pattern (position information). 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.	

2.5 Interrupt priority

The interrupts and priorities used in this system are shown below.

Table 2-6 Interrupt priority

Interrupt level	Priority	Process
15	Min	
14		
13		
12		
11		
10		1[ms] interrupt processing
9		
8		
7		
6		
5		A/D conversion completion interrupt
4		
3		Hall sensor interrupt
2		
1		
0	Max	Overcurrent error interrupt

Allocations

Interrupt	Event	ISR
0	AGT0 INT (AGT interrupt)	agt_int_isr
1	ICU IRQ4 (External pin interrupt 4)	r_icu_isr
2	ICU IRQ5 (External pin interrupt 5)	r_icu_isr
3	ICU IRQ6 (External pin interrupt 6)	r_icu_isr
4	ADC0 SCAN END (A/D scan end interrupt)	adc_scan_end_isr
5	POEG2 EVENT (Port Output disable interrupt C)	poeg_event_isr

Figure 2-10 FSP configuration of interrupt (FSP3.5.0)

3. Descriptions of the control software

The target sample 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 command value

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

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

Item	Conversion ratio (Command value: A/D conversion value)		Channel
Rotation speed command value	CW	0 [rpm]～2650[rpm]：0800H～0FFFH	AN017
	CCW	0 [rpm]～2650[rpm]：07FFH～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]～111 [V]：0000H～0FFFH	AN005

(3). U and W phase current

The U and W phase currents are measured as shown in Table 3-3.

Table 3-3 Conversion Ratio U and W Phase Current

Item	Conversion ratio (U and W phase current：A/D conversion value)	Channel
U and 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 Iw: AN002

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

3.1.3 Speed control

In this system, the motor rotation speed is calculated from a difference of the current timer value and the timer value 2π [rad] before. The timer values are obtained from compare match value of GPT0 (free running timer) at every external interrupt due to hall sensor signals.

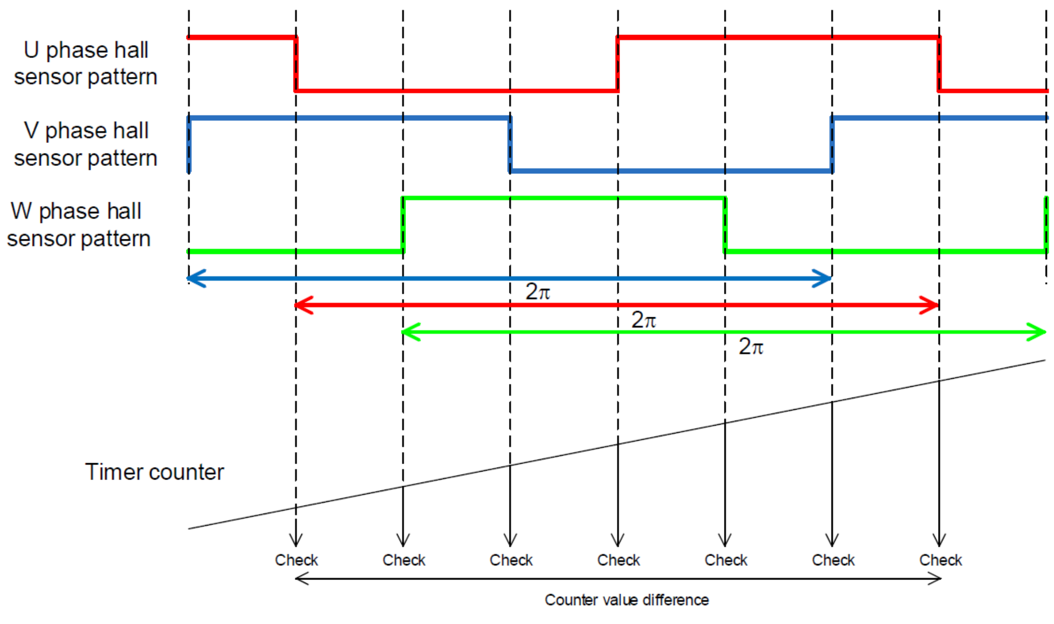


Figure 3-1 Motor Rotation Speed Calculation Method

Speed control in the software targeted by this application note is performed by PI control. Obtain the voltage command value by the following speed control PI control.

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

v^*: Voltage command value, ω^*: Speed command value, ω: Rotation speed

$Kp\omega$: Speed PI proportional gain, $KI\omega$: Speed PI integral gain, s: Laplace operator

3.1.4 Voltage control by PWM

PWM control is used for controlling output voltage. The PWM control is a control method that continuously adjusts the average voltage by varying the duty of pulse, as shown in Figure 3-2.

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

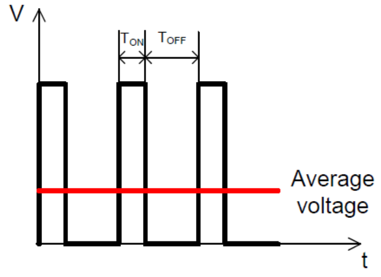


Figure 3-2 PWM Control

Here, modulation factor m is defined as follows.

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

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

This modulation factor is reflected in the setting value of the register that determines the PWM duty.

In the target software of this application note, first-60-degree chopping is used to control the output voltage and speed. Figure 3-3 shows an example of motor control signal output waveforms at non-complimentary first-60-degree Chopping. Figure 3-4 shows an example of motor control signal output waveforms at Complimentary first 60-degree Chopping.

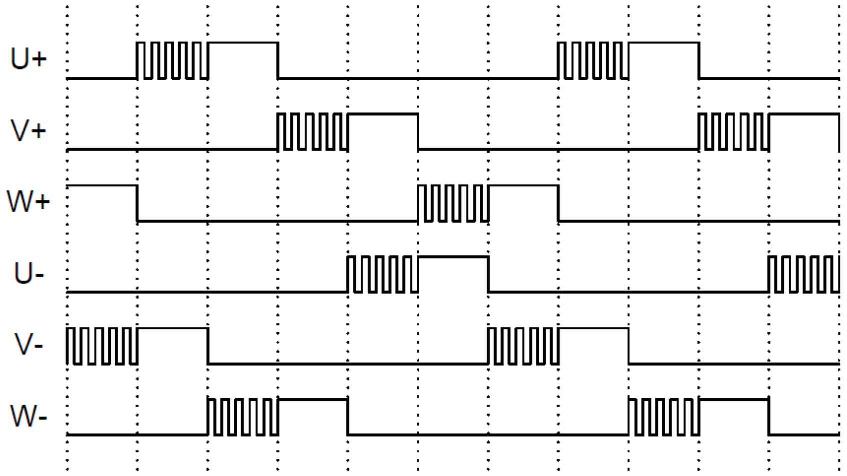


Figure 3-3 non-complimentary first 60-degree Chopping

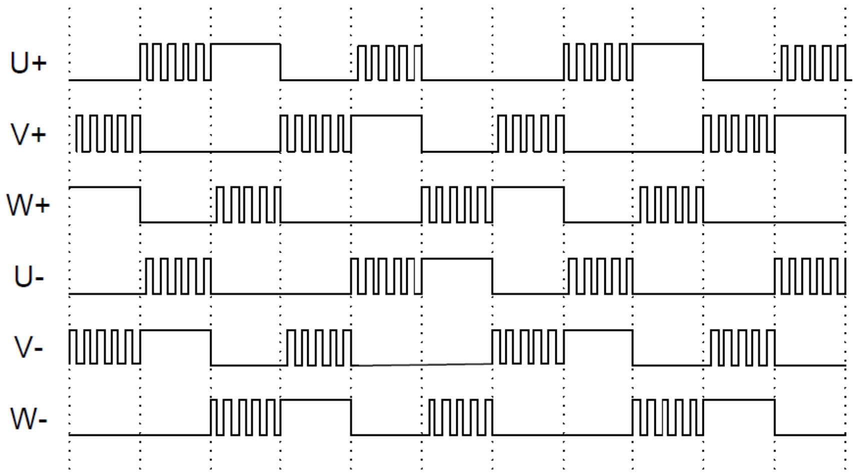
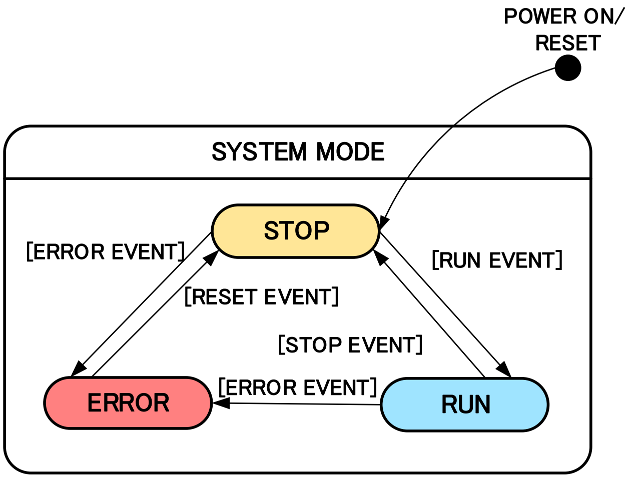


Figure 3-4 Complimentary first 60-degree Chopping

3.1.5 State transition

Figure 3-5 show state transition diagrams of 120-degree conducting control software. In this application note target software, the status 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 3-5 State Transition Diagram of 120-degree Conducting Control Using Hall Sensors Software

(1) SYSTEM MODE

Represents the operating status of the system. The state changes when each event (EVENT) occurs. The operating status of the system includes motor drive stop (INACTIVE), motor drive (ACTIVE), and abnormal status (ERROR).

(2) EVENT

When an EVENT occurs during each SYSTEM MODE, the system operating status changes as shown in the table in Fig. 35 according to the EVENT. The causes of each EVENT are as follows.

Table 3-4 EVENT List

Event	Occurrence factor
STOP	Caused by user operation.
RUN	Caused by user operation.
ERROR	Occurs when the system detects an anomaly.
RESET	Caused by user operation.

3.1.6 Start-up method

In the case of 120-degree conducting control using hall sensors, the rotor position can be determined by hall sensors' signals. Therefore, the conduction pattern at start-up is also determined.

When the control is changed to PI control, at least the motor needs to rotate one time (refer to 3.1.3). At start-up the motor is controlled in open loop with a constant voltage until the motor rotate one time.

Figure 3-6 shows the start-up method in this sample software. In "MTR_MODE_BOOT", open loop with a constant voltage.

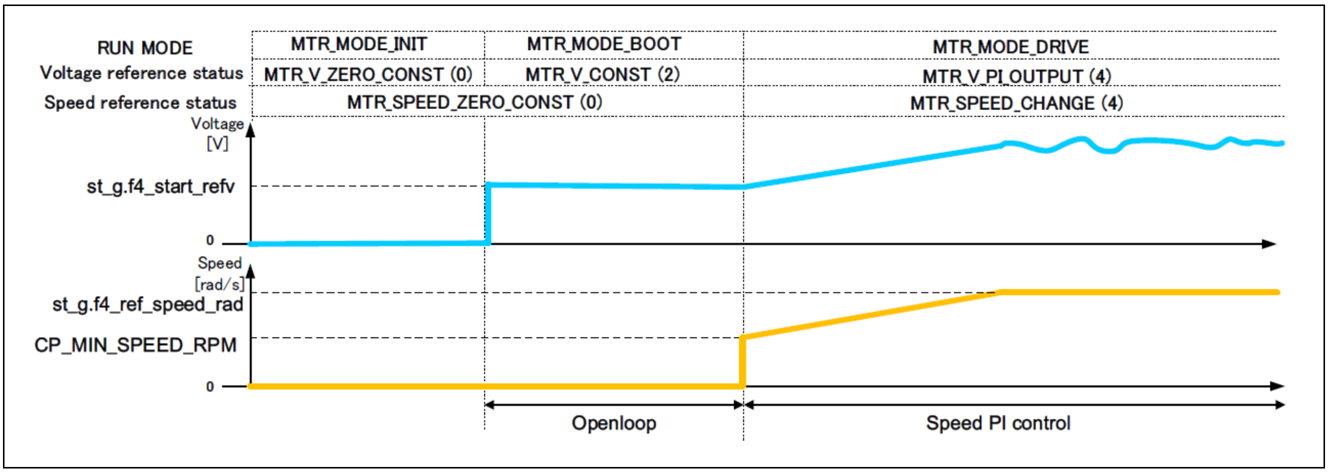


Figure 3-6 Start-up Method (Example)

3.1.7 Sytem protection function

This system has the following six types of error status and enables emergency stop functions in case of occurrence of respective error. Refer to Table 3-5 for settings.

・Overcurrent 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.

・Overvoltage error

The inverter bus voltage is monitored at the overvoltage monitoring cycle. When an over voltage is detected (when the voltage exceeds the limit value), CPU performs an emergency stop. The threshold value of the overvoltage is set in consideration of the error of resistance value of the detection circuit.

・Undervoltage error

The inverter bus voltage is monitored at the undervoltage monitoring cycle. When an under voltage is detected (when the voltage lowers the limit value), CPU performs an emergency stop. The threshold value of the overvoltage is set in consideration of the error of resistance value of the detection circuit.

・Rotation speed abnormality error

The rotation speed is monitored at the rotation speed monitoring cycle. When the speed exceeds the limit value, CPU performs an emergency stop.

・Timeout error of hall interrupt detection

When the interrupt by hall sensors' signal doesn't occur during defined period, CPU performs an emergency stop.

・Hall sensor pattern (position information) error

When an error is detected in hall sensor patterns (position information) generated at hall interrupts, CPU performs an emergency stop.

Table 3-5 Setting Value of Each System Protection Function

Error	Threshold	
Overcurrent error	Overcurrent limit value [A]	0.89
	Monitoring cycle [µs]	50
Overvoltage error	Overvoltage limit value [V]	28
	Monitoring cycle [µs]	50
Undervoltage error	Undervoltage limit value [V]	14
	Monitoring cycle [µs]	50
Rotation speed abnormality error	Speed limit value [rpm]	3000
	Monitoring cycle [µs]	50
Timeout error of hall interrupt detection	Timeout value [ms]	200

3.1.8 AD triggers

Shows the timing of AD triggers and scan groups.

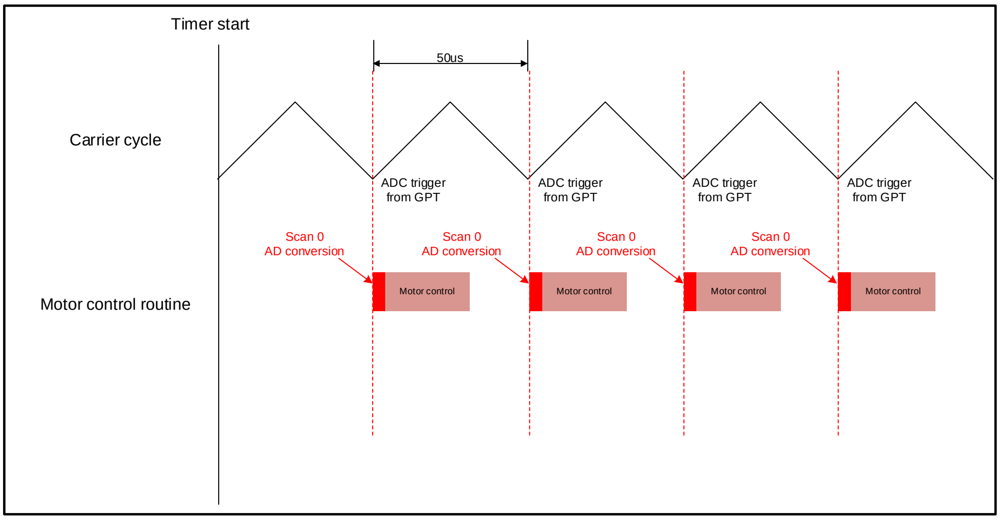


Figure 3-7 AD trigger timing

3.2 Function specifications of 120-degree conducting control software using hall sensors software

Multiple control functions are used in this control program.

Table 3-6 List of Functions “mtr_main.c” [1/2]

File name	Function name	Process overview
mtr_main.c	mtr_init Input : None Output : None	Initialization process
	mtr_main Input : None Output : None	Main processing
	board_ui Input : None Output : None	Board user interface use
	ics_ui Input : None Output : None	GUI tool user interface use
	software_init Input : None Output : None	Initialization of variables used in the main process
	g_poe_overcurrent Input : (poeg_callback_args_t *) p_args / callback function parameter Output : None	POEG interrupt processing
	motor_fsp_init Input : None Output : None	FSP module initialization process
	mtr_callback_120_degree Input : (motor_callback_args_t *) p_args / callback function parameter Output : None	120 degree control callback function
	mtr_board_led_control Input : (uint8_t) u1_motor_status / motor status Output : None	LED pattern setting process
	mtr_remove_sw_chattering Input : (uint8_t) u1_sw / switch type (uint8_t) u1_on_off / ON/OFF state Output : None	Swith chattering removal process
	get_vr1 Input : None Output : None	Get the state of VR1

Table 3-7 List of Functions “mtr_main.c” [2/2]

ファイル名	関数名	処理概要
mtr_main.c	get_sw1 Input : None Output : None	Get the status of SW1
	get_sw2 Input : None Output : None	Get the status of SW2
	led1_on Input : None Output : None	LED1 on
	led2_on Input : None Output : None	LED2 on
	led3_on Input : None Output : None	LED3 on
	led1_off Input : None Output : None	LED1 off
	led2_off Input : None Output : None	LED2 off
	led3_off Input : None Output : None	LED3 off

Table 3-8 List of Functions “r_mtr_ics.c”

File name	Function name	Process overview
r_mtr_ics.c	mtr_set_com_variables Input : None Output : None	Set value from GUI tool
	mtr_ics_variables_init Input : None Output : None	Initialization of variables used by GUI tool
	mtr_ics_interrupt_process Input : None Output : None	Reflect the set value in motor control

Table 3-9 List of Functions “rm_motor_120_degree.c” [1/3]

File name	Function name	Process overview
rm_motor_120_degree.c	RM_MOTOR_120_DEGREE_Open Input : (motor_ctrl_t * const) p_ctrl / Instance parameter (motor_cfg_t const * const) p_cfg / Configuration parameter Output : (fsp_err_t) err / Result	120 degree control start process
	RM_MOTOR_120_DEGREE_Close Input : (motor_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	120 degree control end process
	RM_MOTOR_120_DEGREE_Reset Input : (motor_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	Error state reset process
	RM_MOTOR_120_DEGREE_Run Input : (motor_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	Motor rotation start process
	RM_MOTOR_120_DEGREE_Stop Input : (motor_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	Motor rotation stop process
	RM_MOTOR_120_DEGREE_ErrorSet Input : (motor_ctrl_t * const) p_ctrl / Instance parameter (motor_cfg_t const * const) p_cfg / Configuration parameter Output : (fsp_err_t) err / Result	Error status setting process
	RM_MOTOR_120_DEGREE_SpeedSet Input : (motor_ctrl_t * const) p_ctrl / Instance parameter (float const) speed_rpm / Rotation speed command value [RPM] Output : (fsp_err_t) err / Result	Motor rotation command value setting process
	RM_MOTOR_120_DEGREE_StatusGet Input : (motor_ctrl_t * const) p_ctrl / Instance parameter (uint8_t * const) p_status / Motor control status Output : (fsp_err_t) err / Result	Acquisition of motor control status
	RM_MOTOR_120_DEGREE_SpeedGet Input : (motor_ctrl_t * const) p_ctrl / Instance parameter (float * const) p_speed_rpm / Rotation speed command value [RPM] Output : (fsp_err_t) err / Result	Motor rotation speed acquisition process
	RM_MOTOR_120_DEGREE_WaitStopFlagGet Input : (motor_ctrl_t * const) p_ctrl / Instance parameter (uint8_t * const) p_flg_wait_stop / Motor stopped state Output : (fsp_err_t) err / Result	Acquisition of motor stop state
	RM_MOTOR_120_DEGREE_ErrorCheck Input : (motor_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result (uint16_t * const) p_error / Error status	Error check process

Table 3-10 List of Functions “rm_motor_120_degree.c” [2/3]

File name	Function name	Process overview
rm_motor_120_degree.c	rm_motor_120_degree_active Input : (motor_120_degree_instance_ctrl_t *) p_ctrl / Instance parameter Output : (uint8_t) err / Result	Motor rotation start process
	rm_motor_120_degree_inactive Input : (motor_120_degree_instance_ctrl_t *) p_ctrl / Instance parameter Output : (uint8_t) err / Result	Motor rotation stop process
	rm_motor_120_degree_nowork Input : (motor_120_degree_instance_ctrl_t *) p_ctrl / Instance parameter Output : (uint8_t) err / Result	Blank process
	rm_motor_120_degree_reset Input : (motor_120_degree_instance_ctrl_t *) p_ctrl / Instance parameter Output : (uint8_t) err / Result	Error state reset process
	rm_motor_120_degree_error Input : (motor_120_degree_instance_ctrl_t *) p_ctrl / Instance parameter Output : (uint8_t) err / Result	Post-processing in case of error
	rm_motor_120_degree_statemachine_init Input : (motor_120_degree_statemachine_t *) p_state_machine / State machine Output : None	State machine initialization process
	rm_motor_120_degree_statemachine_reset Input : (motor_120_degree_statemachine_t *) p_state_machine / State machine Output : None	State machine reset process
	rm_motor_120_degree_statemachine_event Input : (motor_120_degree_instance_ctrl_t *) p_ctrl / Instance parameter (motor_120_degree_ctrl_event_t) u1_event / event Output : None	State transition process
	rm_motor_check_over_speed_error Input : (float) f4_speed_rad / Rotation speed [RPM] (float) f4_speed_limit_rad / Rotation speed upper limit [RPM] Output : (uint16_t) u2_temp0 / Error flag	Overspeed error detection processing
	rm_motor_check_over_voltage_error Input : (float) f4_vdc / Inverter bus voltage [V] (float) f4_overvoltage_limit / voltage upper limit [V] Output : (uint16_t) u2_temp0 / Error flag	Overvoltage error detection process
	rm_motor_check_low_voltage_error Input : (float) f4_vdc / Inverter bus voltage [V] (float) f4_lowvoltage_limit / voltage lower limit [V] Output : (uint16_t) u2_temp0 / Error flag	Low voltage error detection process

Table 3-11 List of Functions “rm_motor_120_degree.c” [3/3]

File name	Function name	Process overview
rm_motor_120_degree.c	rm_motor_check_over_current_error Input : (float) f4_iu / U phase current [A] (float) f4_iv / V phase current [A] (float) f4_iw / W phase current [A] (float) f4_oc_limit / Current upper limit [A] Output : (uint16_t) u2_temp0 / Error flag	Overcurrent error detection process
	rm_motor_120_degree_error_check Input : (motor_120_degree_instance_ctrl_t *) p_ctrl / Instance parameter (float) f_iu / U phase current [A] (float) f_iv / V phase current [A] (float) f_iw / W phase current [A] (float) f_vdc / Inverter bus voltage[V] (float) f_speed / Rotation speed [RPM] Output : (uint16_t) u2_error_flags / Error flag	Error detection process
	rm_motor_120_degree_120_control_callback Input : (motor_120_control_callback_args_t *) p_args / Callback function parameter Output : None	120contorol module callback process

Table 3-12 List of Functions “rm_motor_120_control_hall.c” [1/3]

File name	Function name	Process overview
rm_motor_120_control_hall.c	RM_MOTOR_120_CONTROL_HALL_Open Input : (motor_120_control_ctrl_t * const) p_ctrl / Instance parameter (motor_120_control_cfg_t const * const) p_cfg / Configuration parameter Output : (fsp_err_t) err / Result	Hall sensor control start process
	RM_MOTOR_120_CONTROL_HALL_Close Input : (motor_120_control_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	Hall sensor control end process
	RM_MOTOR_120_CONTROL_HALL_Run Input : (motor_120_control_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	Motor rotation start process (Hall sensor control)
	RM_MOTOR_120_CONTROL_HALL_Stop Input : (motor_120_control_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	Motor rotation stop process (Hall sensor control)
	RM_MOTOR_120_CONTROL_HALL_Reset Input : (motor_120_control_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	Error state reset process
	RM_MOTOR_120_CONTROL_HALL_SpeedSet Input : (motor_120_control_ctrl_t* const) p_ctrl / Instance parameter (float * const) p_speed_rpm / Rotation command value [RPM] Output : (fsp_err_t) err / Result	Motor rotation command value setting process
	RM_MOTOR_120_CONTROL_HALL_SpeedGet Input : (motor_120_control_ctrl_t * const) p_ctrl / Instance parameter (float * const) p_speed_rpm / Rotation speed value [RPM] Output : (fsp_err_t) err / Result	Motor rotation speed acquisition process
	RM_MOTOR_120_CONTROL_HALL_CurrentGet Input : (motor_120_control_ctrl_t * const) p_ctrl / Instance parameter (motor_120_driver_current_status_t * const) p_current_status / Current status Output : (fsp_err_t) err / Result	Current and voltage data acquisition process
	RM_MOTOR_120_CONTROL_HALL_WaitStopFlagGet Input : (motor_120_control_ctrl_t * const) p_ctrl / Instance parameter (motor_120_control_wait_stop_flag_t * const) p_flg_wait_stop / Motor stopped state Output : (fsp_err_t) err / Result	Motor stop state acquisition process
	RM_MOTOR_120_CONTROL_HALL_TimeoutErrorFlagGet Input : (motor_120_control_ctrl_t * const) p_ctrl / Instance parameter (motor_120_control_timeout_error_flag_t * const) p_timeout_error_flag / Hall sensor detection timeout error status Output : (fsp_err_t) err / Result	Hall sensor detection timeout error status acquisition process

Table 3-13 List of Functions “rm_motor_120_control_hall.c” [2/3]

File name	Function name	Process overview
rm_motor_120_control_hall.c	RM_MOTOR_120_CONTROL_HALL_PatternErrorFlagGet Input : (motor_120_control_ctrl_t * const) p_ctrl / Instance parameter (motor_120_control_pattern_error_flag_t * const) p_pattern_error_flag / Hall pattern error status Output : (fsp_err_t) err / Result	Hall sensor pattern error status acquisition process
	RM_MOTOR_120_CONTROL_HALL_VoltageRefGet Input : (motor_120_control_ctrl_t * const) p_ctrl / Instance parameter (motor_120_control_voltage_ref_t * const) p_voltage_ref / Voltage setting status Output : (fsp_err_t) err / Result	Obtain voltage setting status
	RM_MOTOR_120_CONTROL_HALL_ParameterUpdate Input : (motor_120_control_ctrl_t * const) p_ctrl / Instance parameter (motor_120_control_cfg_t const * const) p_cfg / Configuration parameter Output : (fsp_err_t) err / Result	Parameter update process
	rm_motor_120_control_hall_interrupt Input : (external_irq_callback_args_t *) p_args / Callback function parameter Output : None	Hall sensor interrupt process
	rm_motor_120_control_hall_speed_cyclic Input : (timer_callback_args_t *) p_args / Callback function parameter Output : None	Callback function for speed control
	rm_motor_120_control_hall_driver_callback Input : (motor_120_driver_callback_args_t *) p_args / Callback function parameter Output : None	A/D conversion complete callback function
	rm_motor_120_control_hall_reset Input : (motor_120_control_hall_instance_ctrl_t *) p_ctrl / Instance parameter Output : None	Instance parameter reset process
	rm_motor_120_control_hall_speed_calc Input : (motor_120_control_hall_instance_ctrl_t *) p_ctrl / Instance parameter Output : None	Rotation speed calculation process
	rm_motor_120_control_hall_wait_motorstop Input : (motor_120_control_hall_instance_ctrl_t *) p_ctrl / Instance parameter Output : None	Rotation stop check process
	rm_motor_120_control_hall_pattern_set Input : (motor_120_control_hall_instance_ctrl_t *) p_ctrl / Instance parameter Output : None	Voltage pattern setting process

Table 3-14 List of Functions “rm_motor_120_control_hall.c” [3/3]

File name	Function name	Process overview
rm_motor_120_control_hall.c	rm_motor_120_control_hall_pattern_first60 Input : (motor_120_control_hall_instance_ctrl_t *) p_ctrl / Instance parameter (uint8_t) u1_signal / Voltage pattern Output : None	Non-complementary first 60 degree chopping process
	rm_motor_120_control_hall_pattern_first60_comp Input : (motor_120_control_hall_instance_ctrl_t *) p_ctrl / Instance parameter (uint8_t) u1_signal / Voltage pattern Output : None	Complementary first 60 degree chopping process
	rm_motor_120_control_hall_speed_ref_set Input : (motor_120_control_hall_instance_ctrl_t *) p_ctrl / Instance parameter Output : None	Speed control command value setting process
	rm_motor_120_control_hall_voltage_ref_set Input : (motor_120_control_hall_instance_ctrl_t *) p_ctrl / Instance parameter Output : None	Voltage command value setting process
	rm_motor_120_control_hall_pi_ctrl Input : (motor_120_control_hall_instance_ctrl_t *) p_ctrl / Instance parameter Output : None	PI control processing
	rm_motor_120_control_hall_check_timeout_error Input : (motor_120_control_hall_instance_ctrl_t *) p_ctrl / Instance parameter Output : None	Hall sensor pattern error judgment processing
	rm_motor_120_control_hall_lpff Input : (float) f4_lpf_input / LPF input value (float) f4_pre_lpf_output / Previous LPF output value (float) f4_lpf_k / LPF gain Output : (float) f4_temp / LPF output value	LPF process
	rm_motor_120_control_hall_limitf Input : (float) f4_value / Input value (float) f4_max / Upper limit (float) f4_min / Lower limit Output : (float) f4_temp / Output value	Upper and lower limit process
	rm_motor_120_control_hall_limitf_h Input : (float) f4_value / Input value (float) f4_max / Upper limit Output : (float) f4_temp / Output value	Upper limit process
	rm_motor_120_control_hall_limitf_l Input : (float) f4_value / Input value (float) f4_min / Lower limit Output : (float) f4_temp / Output value	Lower limit process
	rm_motor_120_control_hall_limitf_abs Input : (float) f4_value / Input value (float) f4_limit_value / Limit value Output : (float) f4_temp / Output value	Absolute value limit process

Table 3-15 List of Functions “rm_motor_120_driver.c” [1/2]

File name	Function name	Process overview
rm_motor_120_driver.c	RM_MOTOR_120_DRIVER_Open Input : (motor_120_driver_ctrl_t * const) p_ctrl / Instance parameter (motor_120_driver_cfg_t const * const) p_cfg / Configuration parameter Output : (fsp_err_t) err / Result	Driver start processing for 120 degree control
	RM_MOTOR_120_DRIVER_Close Input : (motor_120_driver_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	120 degree control driver termination process
	RM_MOTOR_120_DRIVER_Run Input : (motor_120_driver_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	Motor rotation start process
	RM_MOTOR_120_DRIVER_Stop Input : (motor_120_driver_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	Motor rotation stop process
	RM_MOTOR_120_DRIVER_Reset Input : (motor_120_driver_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	Error state reset process
	RM_MOTOR_120_DRIVER_PhaseVoltageSet Input : (motor_120_driver_ctrl_t * const) p_ctrl / Instance parameter (float const) u_voltage / U phase PWM duty (float const) v_voltage / V phase PWM duty (float const) w_voltage / W phase PWM duty Output : (fsp_err_t) err / Result	PWM duty setting process
	RM_MOTOR_120_DRIVER_PhasePatternSet Input : (motor_120_driver_ctrl_t * const) p_ctrl / Instance parameter (motor_120_driver_phase_pattern_t const) pattern / Voltage pattern Output : (fsp_err_t) err / Result	PWM output state switching process
	RM_MOTOR_120_DRIVER_CurrentGet Input : (motor_120_driver_ctrl_t * const) p_ctrl / Instance parameter (motor_120_driver_current_status_t * const) p_current_status / Current status Output : (fsp_err_t) err / Result	Current and voltage data acquisition process
	RM_MOTOR_120_DRIVER_CurrentOffsetCalc Input : (motor_120_driver_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	A/D value offset calculation process
	RM_MOTOR_120_DRIVER_FlagCurrentOffsetGet Input : (motor_120_driver_ctrl_t * const) p_ctrl / Instance parameter Output : (fsp_err_t) err / Result	A/D value offset calculation status acquisition process

Table 3-16 List of Functions “rm_motor_120_driver.c” [2/2]

File name	Function name	Process overview
rm_motor_120_driver.c	RM_MOTOR_120_DRIVER_ParameterUpdate Input : (motor_120_driver_ctrl_t * const) p_ctrl / Instance parameter (motor_120_driver_cfg_t const * const) p_cfg / Configuration parameter Output : (fsp_err_t) err / Result	Parameter update process
	rm_motor_120_driver_reset Input : (motor_120_driver_instance_ctrl_t *) p_ctrl / Instance parameter Output : None	Instance parameter reset process
	rm_motor_120_driver_output_pwm Input : (motor_120_driver_instance_ctrl_t *) p_ctrl / Instance parameter Output : None	PWM output start process
	rm_motor_120_driver_ctrl_start Input : (motor_120_driver_instance_ctrl_t *) p_ctrl / Instance parameter Output : None	PWM output start processing for motor control
	rm_motor_120_driver_ctrl_stop Input : (motor_120_driver_instance_ctrl_t *) p_ctrl / Instance parameter Output : None	PWM output stop process
	rm_motor_120_driver_set_uvw_duty Input : (motor_120_driver_instance_ctrl_t *) p_ctrl / Instance parameter (float) f_duty_u / U phase PWM duty (float) f_duty_v / V phase PWM duty (float) f_duty_w / W phase PWM duty Output : None	PWM duty setting process
	rm_motor_120_driver_current_get Input : (motor_120_driver_instance_ctrl_t *) p_ctrl / Instance parameter Output : None	A/D value acquisition process
	rm_motor_120_driver_mod_set_max_duty Input : (motor_120_driver_modulation_t *) p_mod/ PWM setting (float) f4_max_duty / Maximum duty Output : None	Maximum duty setting process
	rm_motor_120_driver_mod_set_min_duty Input : (motor_120_driver_modulation_t *) p_mod/ PWM setting (float) f4_max_duty / Minimum duty Output : None	Minimum duty setting process
	rm_motor_120_driver_pin_cfg Input : (bsp_io_port_pin_t) pin / Pin number (uint32_t) cfg / Set value Output : None	Pin configuration setting process
	rm_motor_120_driver_cyclic Input : (adc_callback_args_t *) p_args / Callback function parameter Output : None	A/D conversion complete callback function

3.3 Contents of control

3.3.1 Configuration Options

The configuration options for the 120-degree control using hall sensors module for the motor can be configured using the RA Configurator. The changed options are automatically reflected in hal_data.c when the code is generated. Option names and settings are listed in Table 3-17 Configuration Options below.

Table 3-17 Configuration Options

Configuration Options (rm_motor_120_degree.h)	
Option name	Contents
Limit of over current (A) Initial: 0.42F	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: 3000.0F	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.

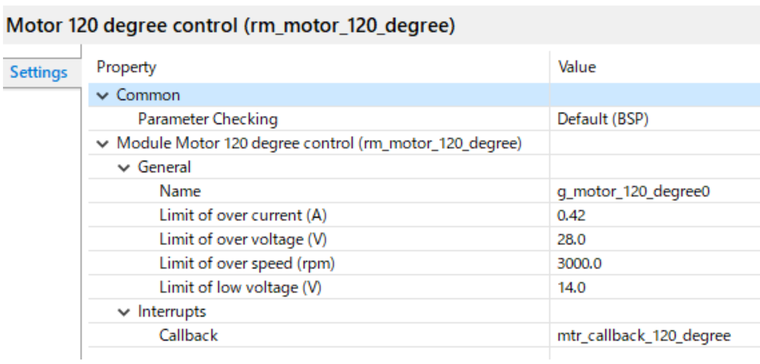


Figure 3-8 FSP configuration of 120-degree conducting control (FSP3.5.0)

3.3.2 Configuration Options for included modules

The 120-degree conducting control using hall sensors module for motors includes the following modules.

・120 control hall module

・120 driver module

In addition, these modules have the same configuration parameters as the 120-degree conducting control using hall sensors module. The option names and setting values are shown in the table below.

Table 3-18 Configuration Options

Configuration Options (rm_motor_120_control_api.h)	
Option name	Contents
Conduction type Initial: First 60-degree PWM	Switching of first 60 degree chopping control
Timeout counts (msec) Initial: 200	Stop judgement counter [ms]
Maximum voltage (V) Initial: 20.0	Maximum command voltage [V]
Minimum voltage (V) Initial: 3.0	Minimum command voltage [V]
Speed PI decimation Initial: 0	Interrupt thinning number for speed PI control
Free run timer frequency (MHz) Initial: 120	Free run timer frequency [MHz]
Speed LPF K Initial: 1.0	Speed LPF parameter
Step of speed change Initial: 0.2	Speed command maximum increase limit
PI control KP Initial: 0.02	Speed PI proportional gain
PI control KI Initial: 0.0005F	Speed PI Integral gain
PI control limit Initial: 24.0F	Voltage PI control output limit value [V]
Motor Parameter | Pole pairs Initial: 2	Number of pole pairs

Table 3-19 Configuration Options

Configuration Options (rm_motor_120_control_hall.h)	
Option name	Contents
Start reference voltage (V) Initial: 5.8	Reference voltage for start-up [V]
Hall wait counts Initial: 12	Wait counts of hall interrupts to start speed calculation
Stop judge time Initial: 1000	Stop judge count
Minimum limit speed (rpm) Initial: 550	Rotation speed command minimum value (mechanical angle) [rpm]
Hall interrupt mask value Initial: 15	Hall interrupt decimation number

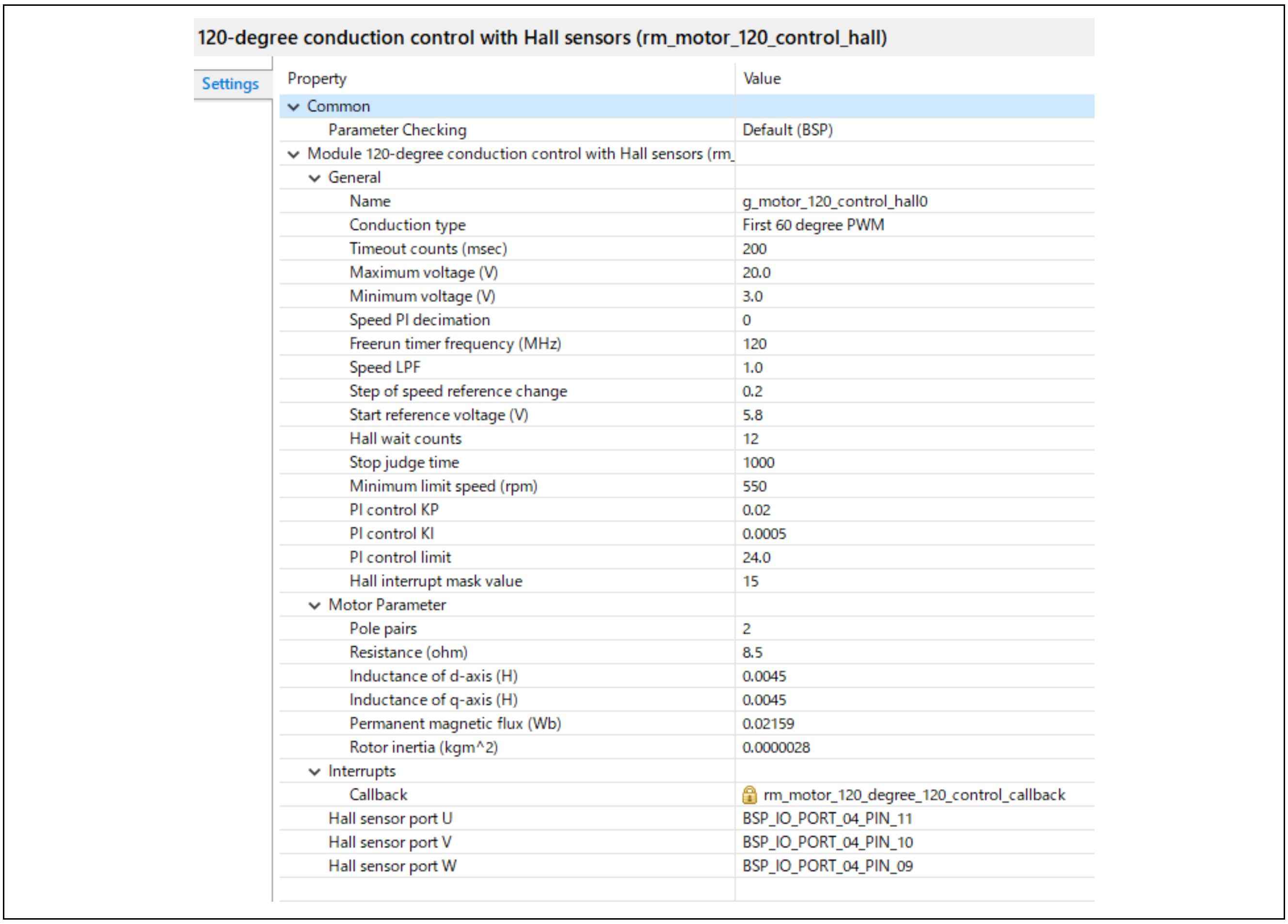
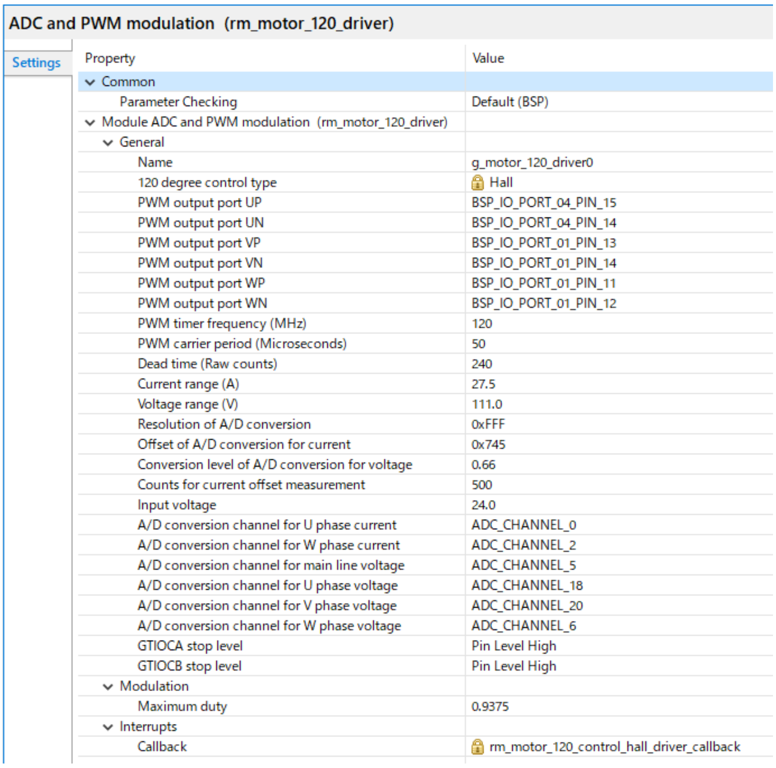


Figure 3-9 FSP configuration of 120-degree conducting control using hall sensors (FSP3.5.0)

Table 3-20 Configuration Options [1/2]

Configuration Options (rm_motor_120_driver.h)	
Option name	Contents
120 degree control type Initial: Hall	120 degree energization control sensor type (Please select HALL)
PWM output port UP Initial: BSP_IO_PORT_04_PIN_15	PWM output (Up) port
PWM output port UN Initial: BSP_IO_PORT_04_PIN_14	PWM output (Un) port
PWM output port VP Initial: BSP_IO_PORT_01_PIN_13	PWM output (Vp) port
PWM output port VN Initial: BSP_IO_PORT_01_PIN_14	PWM output (Vn) port
PWM output port WP Initial: BSP_IO_PORT_01_PIN_11	PWM output (Wp) port
PWM output port WN Initial: BSP_IO_PORT_01_PIN_12	PWM output (Wn) port
PWM timer frequency (MHz) Initial: 120	PWM timer frequency [MHz]
PWM carrier period (Microseconds) Initial: 50	PWM carrier frequency [Micro seconds]
Dead time (Raw counts) Initial: 240	Dead time count [Raw counts]
Current range (A) Initial: 27.5	Current detection range [A]
Voltage range (V) Initial: 111.0	Voltage detection range [V]
Resolution of A/D conversion Initial: 0xFFF	A/D conversion value
Offset of A/D conversion for current Initial: 0x745	A/D conversion offset
Conversion level of A/D conversion for voltage Initial: 0.66	Voltage A/D conversion rate
Counts for current offset measurement Initial: 500	Offset value calculation count
Input voltage Initial: 24.0	Bus voltage
A/D conversion channel for U phase current Initial: ADC_CHANNEL_0	U phase current detection channel
A/D conversion channel for W phase current Initial: ADC_CHANNEL_2	W phase current detection channel
A/D conversion channel for main line voltage Initial: ADC_CHANNEL_5	Inverter bus voltage detection channel
GTIOCA stop level Initial: Pin Level High	Level when the upper arm is stopped
GTIOCB stop level Initial: Pin Level High	Level when lower arm is stopped
Modulation | Maximum duty Initial: 0.9375	PWM maximum duty



ADC and PWM modulation (rm_motor_120_driver)

Settings

Property	Value
∨ Common	
Parameter Checking	Default (BSP)
∨ Module ADC and PWM modulation (rm_motor_120_driver)	
∨ General	
Name	g_motor_120_driver0
120 degree control type	Hall
PWM output port UP	BSP_IO_PORT_04_PIN_15
PWM output port UN	BSP_IO_PORT_04_PIN_14
PWM output port VP	BSP_IO_PORT_01_PIN_13
PWM output port VN	BSP_IO_PORT_01_PIN_14
PWM output port WP	BSP_IO_PORT_01_PIN_11
PWM output port WN	BSP_IO_PORT_01_PIN_12
PWM timer frequency (MHz)	120
PWM carrier period (Microseconds)	50
Dead time (Raw counts)	240
Current range (A)	27.5
Voltage range (V)	111.0
Resolution of A/D conversion	0xFFF
Offset of A/D conversion for current	0x745
Conversion level of A/D conversion for voltage	0.66
Counts for current offset measurement	500
Input voltage	24.0
A/D conversion channel for U phase current	ADC_CHANNEL_0
A/D conversion channel for W phase current	ADC_CHANNEL_2
A/D conversion channel for main line voltage	ADC_CHANNEL_5
A/D conversion channel for U phase voltage	ADC_CHANNEL_18
A/D conversion channel for V phase voltage	ADC_CHANNEL_20
A/D conversion channel for W phase voltage	ADC_CHANNEL_6
GTIOCA stop level	Pin Level High
GTIOCB stop level	Pin Level High
∨ Modulation	
Maximum duty	0.9375
∨ Interrupts	
Callback	rm_motor_120_control_hall_driver_callback

Figure 3-10 FSP configuration of ADC and PWM modulation driver for 120-degree conducting control (FSP3.5.0)

3.4 Control flows (flow charts)

3.4.1 Main process

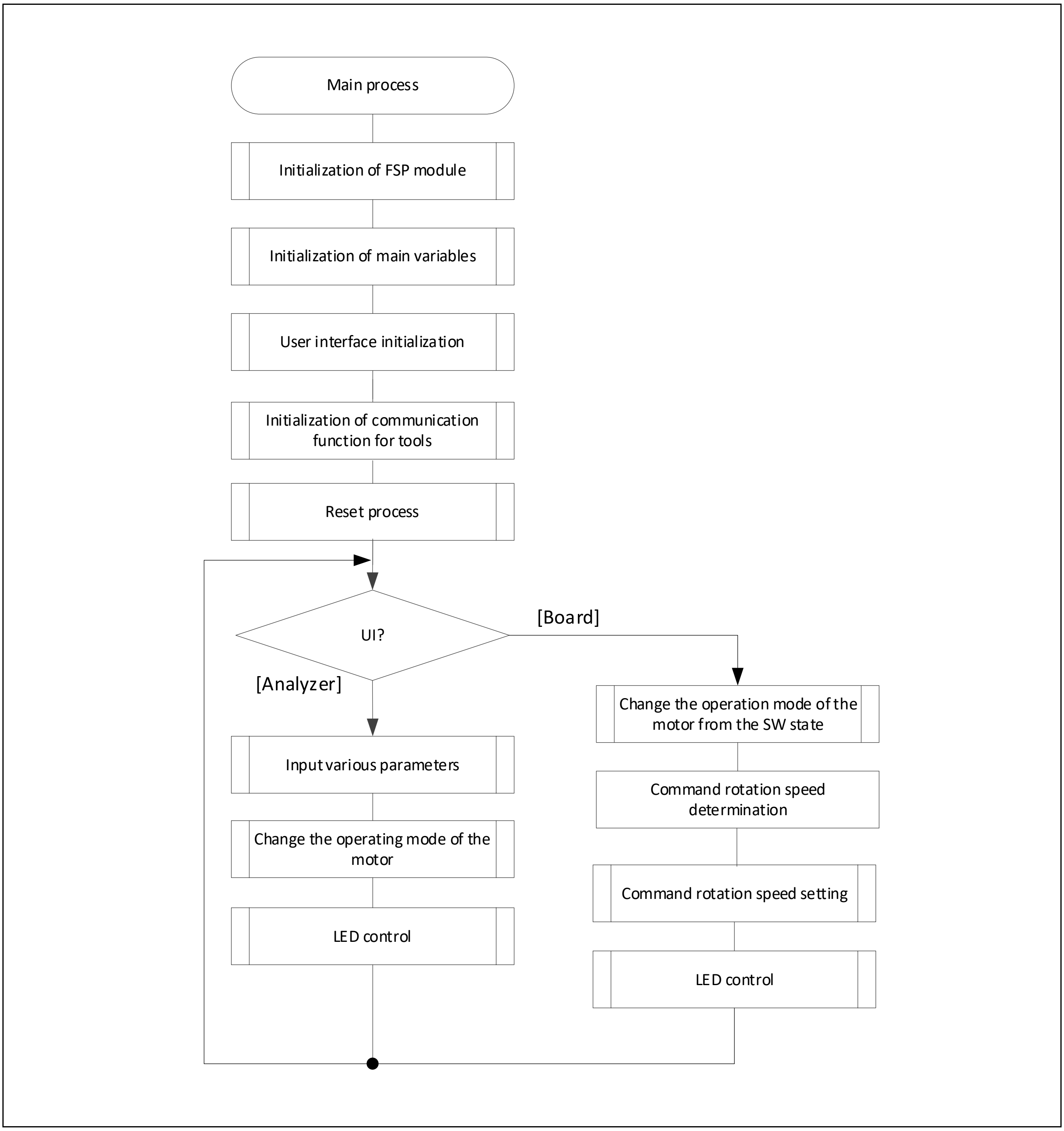


Figure 3-11 Main process Flowchart

3.4.2 50 [µs] Periodic interrupt (carrier periodic interrupt) handling

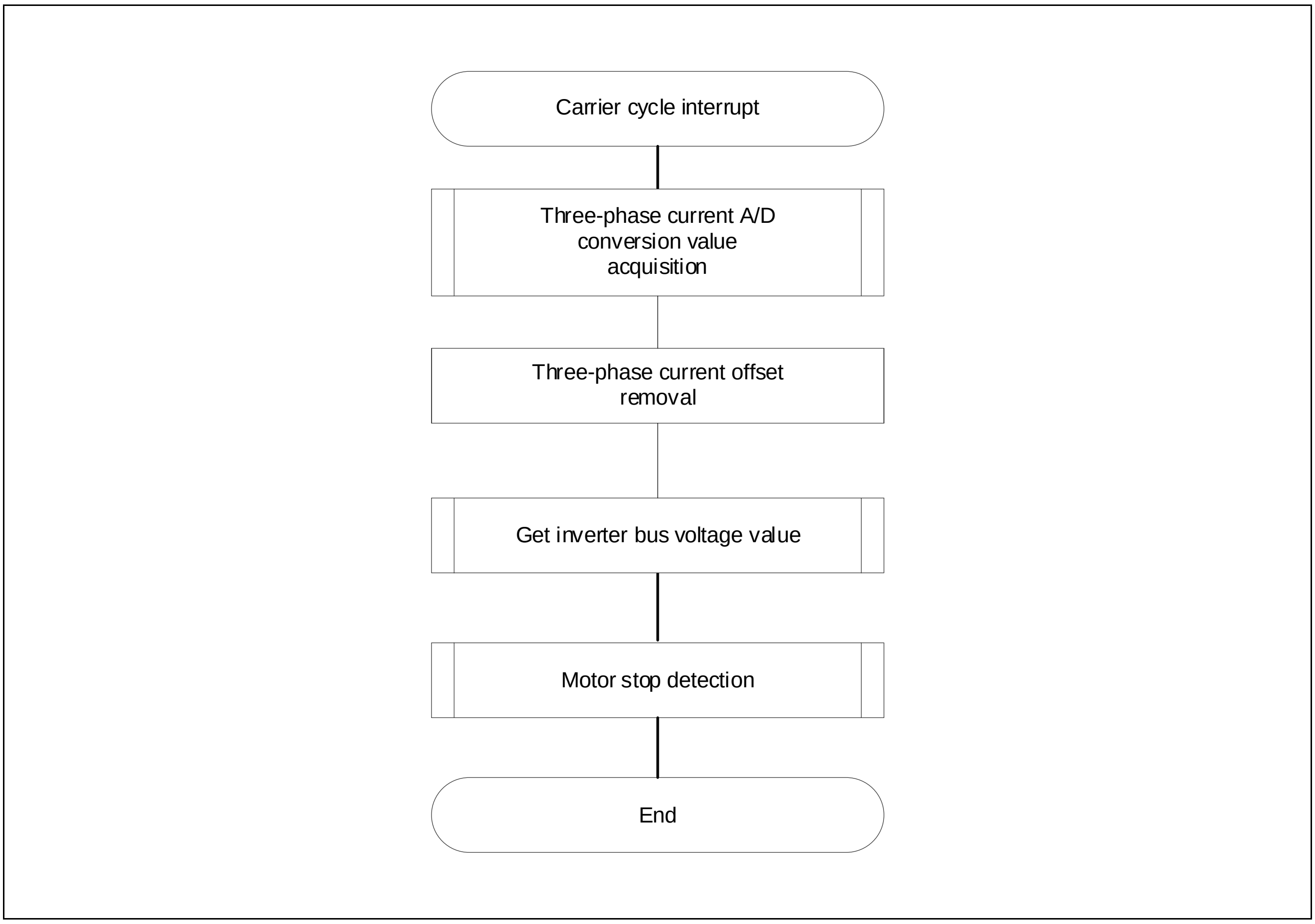


Figure 3-12 50 [µs] Periodic Interrupt Handling

3.4.3 1 [ms] periodic interrupt handling

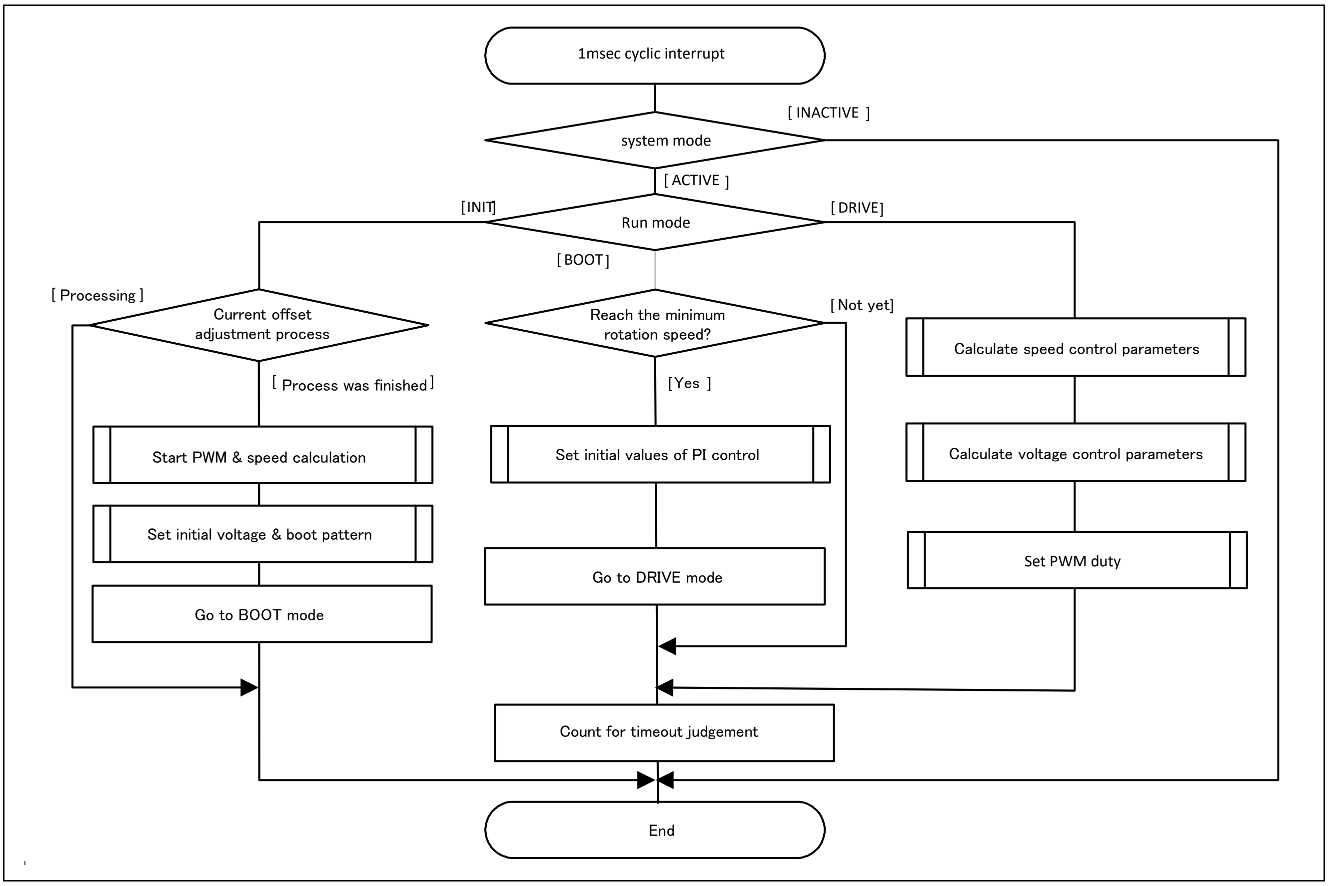


Figure 3-13 1 [ms] Periodic Interrupt Handling

3.4.4 Overcurrent interrupt handling

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.

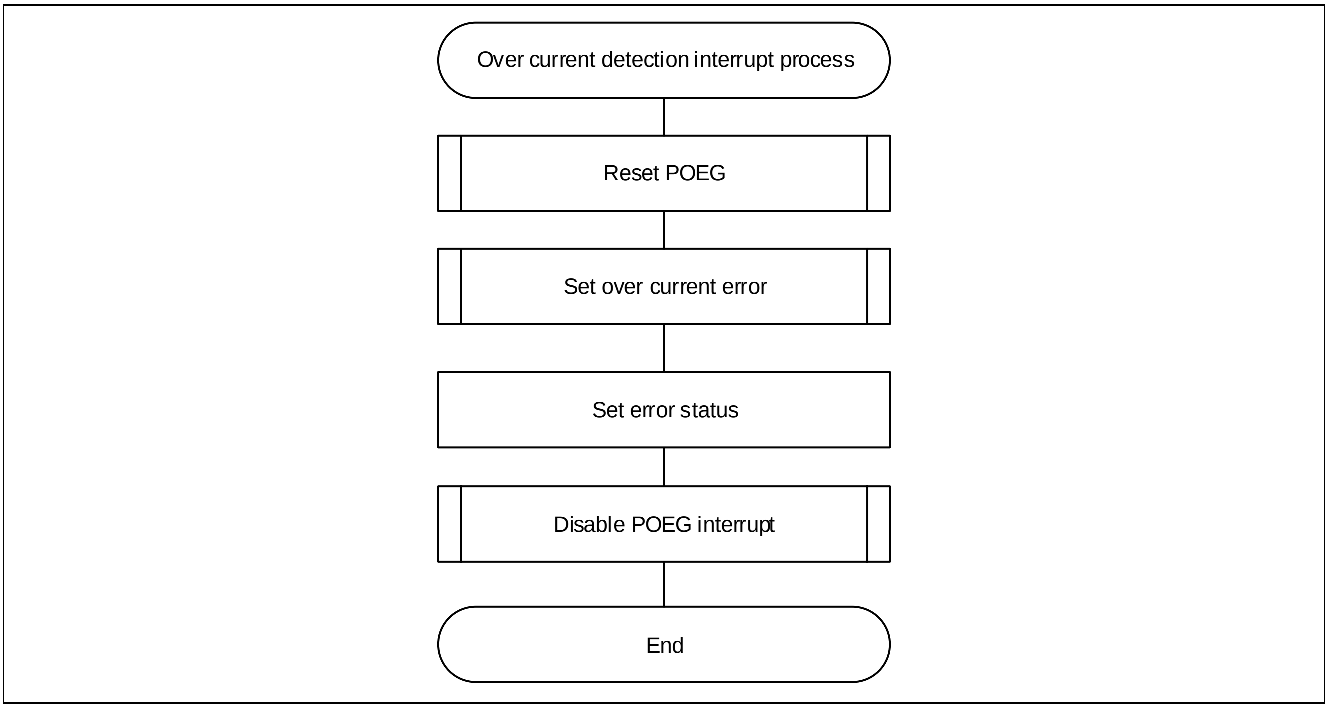


Figure 3-14 Overcurrent Detection Interrupt Handling

4. Evaluation environment explanation

This application note describes the target software.

4.1 Operating environment

Table 4-1 lists the hardware requirements for building and debugging motor control software.

Table 4-1 120-degree conducting control using hall sensors hardware basic specifications

Item	Contents
Inverter Board	RA6T1-RSSK [RTK0EM0000B10020BJ]
CPU Card	RA6T1 CPU Card [RTK0EMA170C00000BJ]
Motor	Brushless DC Motor (TG-55L-KA 24V)
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 lists the software requirements for building and debugging motor control software.

Table 4-2 120-degree conducting control using hall sensors software basic specifications

Item		Version	Contents
GCC	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 Project import

The sample software can be imported into e2 studio by following the steps below.

1. File → Import

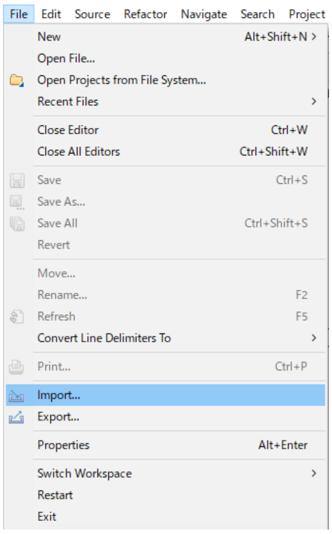


Figure 4-1 File Menu

2. Select "Existing Projects into Workspace" and click the [Next] button.

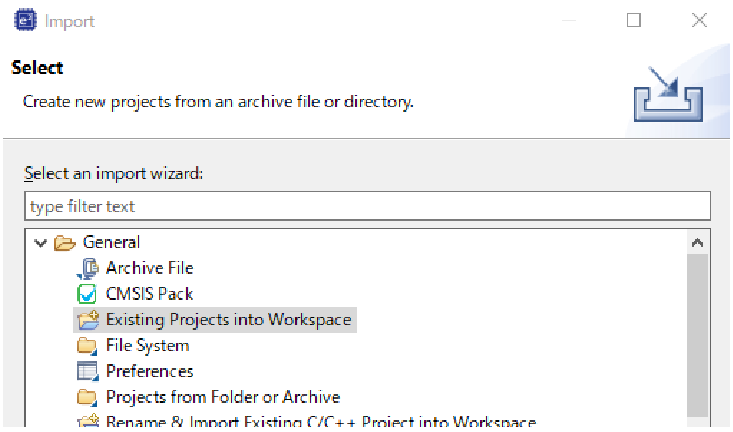


Figure 4-2 Import Menu

3. Select a project file. Click the Finish button to import the project.

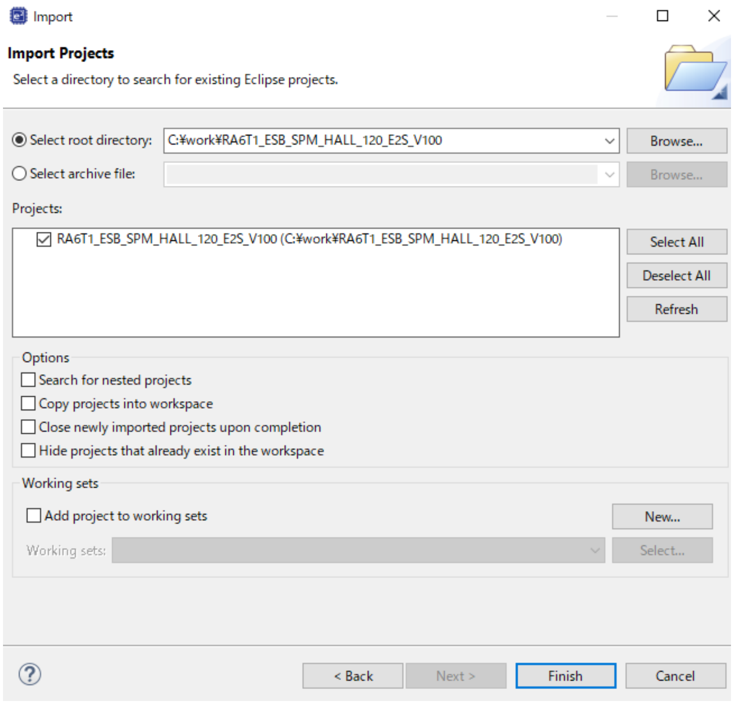


Figure 4-3 Project Import

4.3 Build and debug

Please refer to “e2 studio User’s Manual Starting Guide (R20UT4204)”.

4.4 Operation outline of sample software

4.4.1 Quick start

To operate the sample code without using the motor control development support tool "Renesas Motor Workbench", execute the quick start sample project according to the following procedure.

(1) After the regulated power is turned on or reset, all LEDs 1 and 2 on the inverter board are off and the motor is stopped.

(2) Turn on the toggle switch (SW1) on the inverter board to rotate the motor. Every time the toggle switch (SW1) is switched, the motor starts / stops rotating repeatedly. When the motor is rotating normally, LED1 on the inverter board lights up. At this time, if LED2 on the inverter board is lit, an error has occurred.

(3) When changing the rotation direction of the motor, adjust it with the volume resistor (VR1) on the inverter board.

- Turn volume resistor (VR1) clockwise : Motor rotates clockwise
- Turn the volume resistor (VR1) counterclockwise : The motor rotates counterclockwise

(4) If an error occurs, LED2 on the inverter board lights up and rotation stops. To recover, turn off the toggle switch (SW1) on the inverter board and then press the push switch (SW2).

(5) When you finish the operation check, check that the rotation of the motor is stopped, and turn off the output of the regulated power supply.

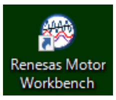
4.5 Motor Control Development Support Tool, ‘Renesas Motor Workbench’

4.5.1 Overview

In the target sample programs described in this application note, user interfaces (rotating/stop command, rotation speed command, etc.) based on the motor control development support tool, ’Renesas Motor Workbench’ can be used. 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 ‘Motor RSSK Support Tool’ by clicking this icon. 

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

And select RMT file in “src/application/user_interface/ics”.

③ Use the ‘Connection’ COM select menu to choose the COM port for Motor RSSK.

④ Click on the ‘Analyzer’ icon of Select Tool panel to open Analyzer function window.

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.

- Change the user interface to use Renesas Motor Workbench

(1) Turn on "RMW UI".

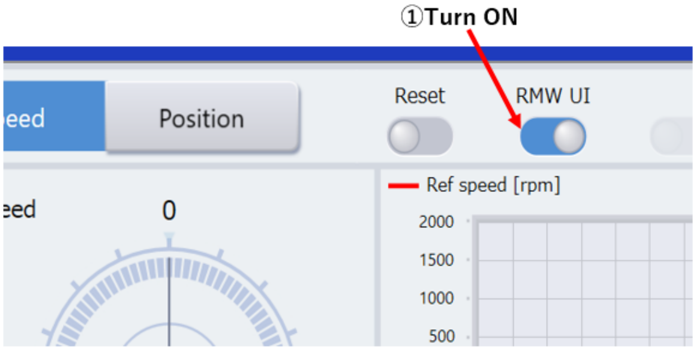


Figure 4-1 Procedure for changing to use Renesas Motor Workbench

- Change the user interface to use Renesas Motor Workbench

(1) Press the “Run” button

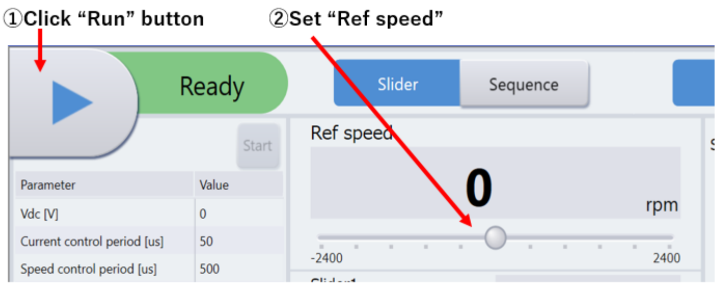


Figure 4-2 Motor rotation procedure

- Stop the motor

(1) Press the “Stop” button

Figure 4-3 Motor rotation procedure

- Processing when it stops (error)
  (1) Turn on "Reset" button.
  (2) Turn off “Reset” button

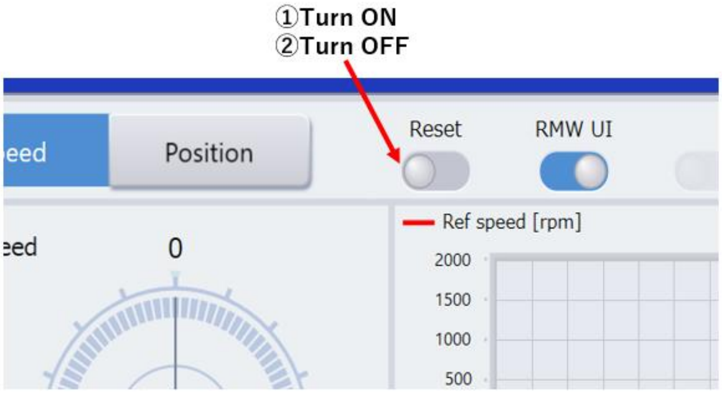


Figure 4-4 Error clearing procedure

4.5.3 List of variables for Analyzer function

Table 4-3 shows a list of input variables when using the Analyzer user interface. The input values to these variables will be reflected in the corresponding variables when the same values as g_u1_enable_write are written to com_u1_enable_write. However, variables marked with (*) do not depend on com_u1_enable_write.

Table 4-3 List of variables for analyzer function input

Variable name	Type	Contents
com_u1_mode_system1(*)	uint8_t	State management 0 : Stop mode, 1 : Run mode, 3 : Reset
com_f4_ref_speed_rpm	float	Speed command value (mechanical angle) ［rpm］
com_f4_overcurrent_limit	float	High current limit value [A]
com_f4_overvoltage_limit	float	High voltage limit value [V]
com_f4_overspeed_limit_rpm	float	Speed limit value (mechanical angle) [rpm]
com_f4_lowvoltage_limit	float	Low voltage limit value [V]
com_u4_timeout_cnt	uint32_t	Timeout count limit
com_f4_max_drive_v	float	Maximum command voltage [V]
com_f4_min_drive_v	float	Minimum command voltage [V]
com_f4_speed_lpf_k	float	Speed LPF parameter
com_f4_limit_speed_change	float	Command speed changing limit
com_f4_start_ref_v	float	Reference voltage for start-up [V]
com_f4_pi_ctrl_kp	float	Speed PI proportional gain
com_f4_pi_ctrl_ki	float	Speed PI Integral gain
com_f4_pi_ctrl_ilimit	float	Voltage PI control output limit value [V]
com_u4_hall_interrupt_mask_value	uint32_t	Hall interrupt decimation number
com_u4_mtr_pp	uint32_t	Motor Pole pairs
com_u1_enable_write	uint8_t	Enable to rewriting variables (Write permission when the same value as g_u1_enable_write is written)

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 use Analyzer
  ① Make sure that "check" is entered in the [W?] box of "com_u1_sw_userif".
  ② Enter 0 in the [Write] box.
  ③ Click the “Write” button.

- Driving the motor
  ① The [W?] check boxes contain checkmarks for “com_u1_mode_system”,“com_f4_ref_speed_rpm”, “com_u1_enable_write”.
  ② Type a reference speed value in the [Write] box of “com_f4_ref_speed_rpm”.
  ③ Click the “Write” button.
  ④ Click the “Read” button. Confirm the [Read] box of “com_f4_ref_speed_rpm”,” g_u1_enable_write”.
  ⑤ Enter the 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.

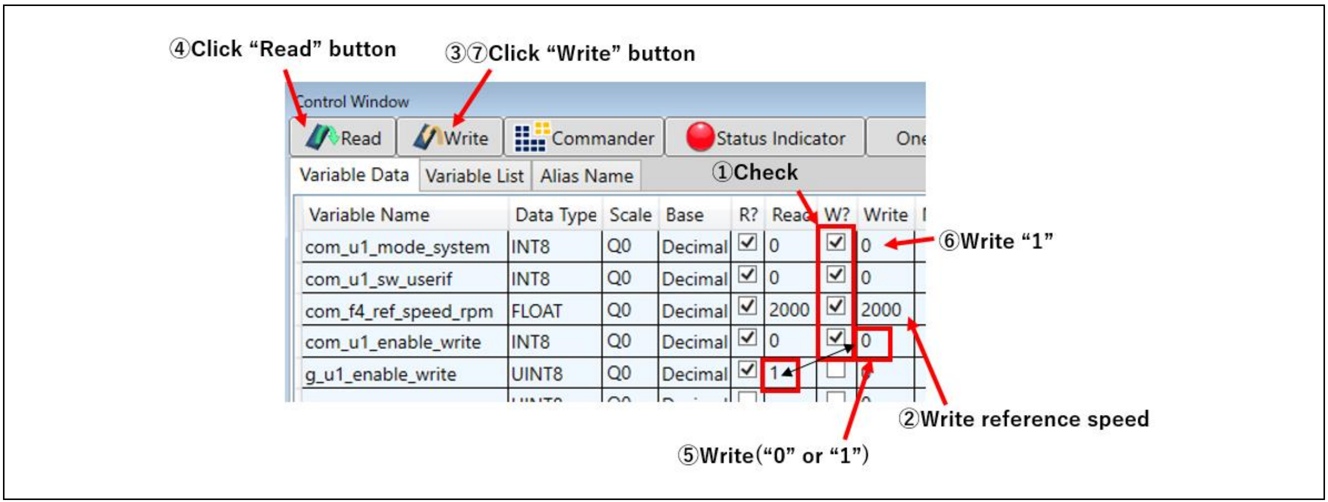


Figure 4-5 Procedure – Driving the motor

- Stop the motor
  ① Enter “0” in the [Write] box of “com_u1_mode_system”.
  ② Click the “Write” button.

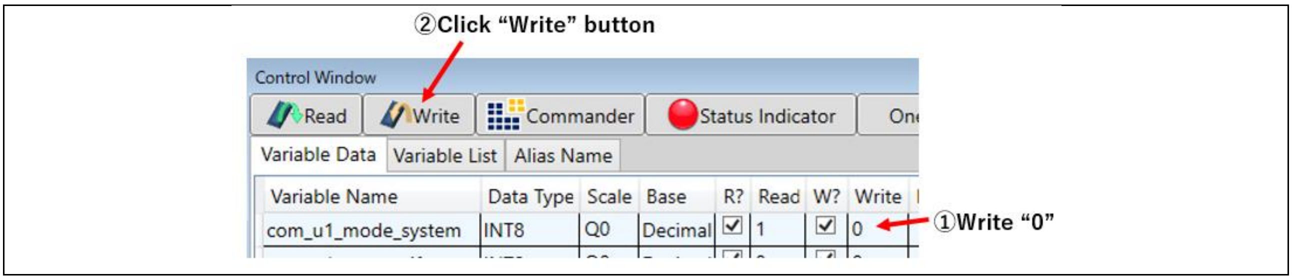


Figure 4-6 Procedure – Stop the motor

- Error cancel operation
  ① Enter “3” in the [Write] box of “com_u1_mode_system”.
  ② Click the “Write” botton.

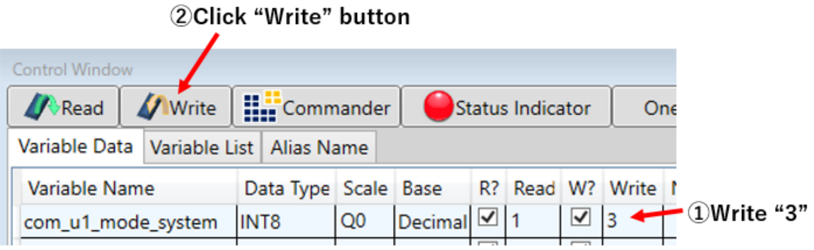


Figure 4-7 Procedure – Error cancel operation

5. Reference document

RA6T1 Group User's Manual: Hardware (R01UH0897)
RA Flexible Software Package Documentation
120-degree conducting control of permanent magnetic synchronous motor (Algorithm) (R01AN2657)
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 Record

Rev.	Data of issue	Descriptions	
		Page	Summary
1.00	2021.12.17	—	First edition issued.

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.

Corporate Headquarters

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

Trademarks

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/.