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

For Renesas Flexible Motor Control Series

Abstract

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 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.

- RA6T2 (R7FA6T2BD3CFP)
- RA6T3 (R7FA6T3BB3CFM)
- RA4T1 (R7FA4T1BB3CFM)

Target software

The following shows the target software for this application:

- RA6T2_MCILV1_SPM_HALL_120_E2S_V110
- RA6T3_MCILV1_SPM_HALL_120_E2S_V100
- RA4T1_MCILV1_SPM_HALL_120_E2S_V100

Contents

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 RA series 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'.

2. Development environment

2.1 Operation check environment

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

Table 2-1 Hardware development environment

Classification	Product used
Microcomputer / CPU card P/N	RA6T2 (R7FA6T2BD3CFP) / RTK0EMA270C00000BJ RA4T1 (R7FA4T1BB3CFM) / RTK0EMA430C00000BJ RA6T3 (R7FA6T3BB3CFM) / RTK0EMA330C00000BJ
Inverter board	MCI-LV-1 / RTK0EM0000S04020BJ
motor	R42BLD30L3 (manufactured by MOONS')

Table 2-2 Software development environment

e^2studio version	FSP version	Toolchain version
e^2studio：2023-04	V4.4.0	GCC ARM Embedded : V10.3.1.20210824

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

2.2 Hardware specifications

2.2.1 Hardware configuration diagram

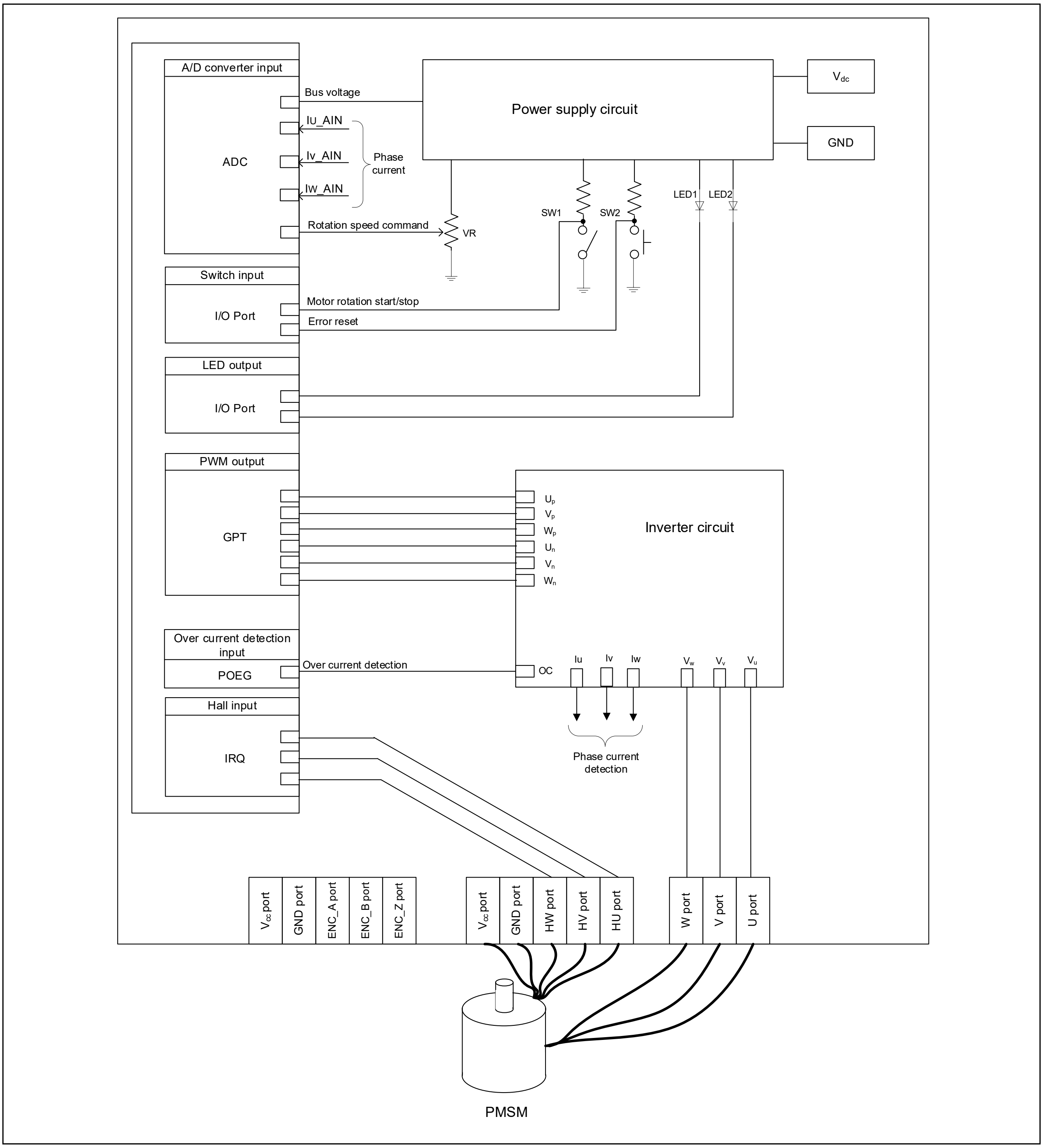


Figure 2-1 Hardware Configuration Diagram

2.2.2 User interface

Table 2-3 is lists of user interfaces of this system.

Table 2-3 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	Yellow green LED	- At the time of Motor rotation : ON - At the time of stop : OFF
LED2	Yellow green LED	- At the time of error detection : ON - At the time of normal operation : OFF
LED3	Yellow green LED	Not used in this system
RESET	Push switch(SW1)	System reset

List of port interfaces of this system.

Table 2-4 Port Interface

Function	RA6T2	RA4T1	RA6T3
Inverter bus voltage measurement	PA06 / AN006	P004 / AN004	P004 / AN004
For rotation speed command value input (analog value)	PB00 / AN008	P005 / AN005	P005 / AN005
START/STOP toggle switch (SW1)	PD04	P304	P304
ERROR RESET push switch (SW2)	PD07	P200	P200
LED1 ON/OFF control	PD01	P113	P113
LED2 ON/OFF control	PD02	P106	P106
U phase current measurement	PA04 / AN004	P000 / AN000	P000 / AN000
V phase current measurement	PA02 / AN002	P001 / AN001	P001 / AN001
W phase current measurement	PA00 / AN000	P002 / AN002	P002 / AN002
PWM output (U_p)	PB04 / GTIOC4A	P409 / GTIOC1A	P409 / GTIOC1A
PWM output (V_p)	PB06 / GTIOC5A	P103 / GTIOC2A	P103 / GTIOC2A
PWM output (W_p)	PB08 / GTIOC6A	P111 / GTIOC3A	P111 / GTIOC3A
PWM output (U_n)	PB05 / GTIOC4B	P408 / GTIOC1B	P408 / GTIOC1B
PWM output (V_n)	PB07 / GTIOC5B	P102 / GTIOC2B	P102 / GTIOC2B
PWM output (W_n)	PB09 / GTIOC6B	P112 / GTIOC3B	P112 / GTIOC3B
PWM emergency stop input at the time of overcurrent detection	PC13 / GTETRGD	P104 / GTETRGB	P104 / GTETRGB
Hall sensor input (HU)	PC04 / IRQ10	P008 / IRQ12	P008 / IRQ12
Hall sensor input (HV)	PC05 / IRQ11	P006 / IRQ11	P006 / IRQ11
Hall sensor input (HW)	PB01 / IRQ1	P015 / IRQ13	P015 / IRQ13

List of port interfaces of the sensor.

Table 2-1 Port Interfaces

Function	MCI-LV-1
GND	CN6 1pin
+5V	CN6 2pin
Hall sensor input (HU)	CN6 3pin
Hall sensor input (HV)	CN6 4pin
Hall sensor input (HW)	CN6 5pin

2.2.3 Peripheral functions

Table 2-5 is a list of peripheral functions used in this system.

Table 2-5 Peripheral Functions List

Peripheral	Purpose	RA6T2	RA4T1	RA6T3
12-bit A/D Converter	U phase current measurement	AN004	AN000	AN000
	V phase current measurement	AN002	AN001	AN001
	W phase current measurement	AN000	AN002	AN002
	Inverter bus voltage measurement	AN006	AN004	AN004
	For rotation speed command value input (analog value)	AN008	AN005	AN005
AGT	Speed control interval timer	AGT0	AGT0	AGT0
	Free run timer for rotation speed measurement	-	AGT1	AGT1
GPT	U phase PWM output	CH4	CN1	CN1
	V phase PWM output	CH5	CN2	CN2
	W phase PWM output	CH6	CN3	CN3
	Free run timer for rotation speed measurement	CH0	-	-
POEG	PWM emergency stop input at the time of overcurrent detection	Group D	Group B	Group B

2.2.3.1 RA6T2

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

U-phase current (Iu), V-phase current (Iv), 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).

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

Used as a 1ms interval timer.

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

On the channel 4, 5, and 6, output with dead time is performed by using the complementary PWM Output Operating Mode.

Channel 0 is used as a free run timer for speed measurement.

(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 GTETRGD port is detected).

(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

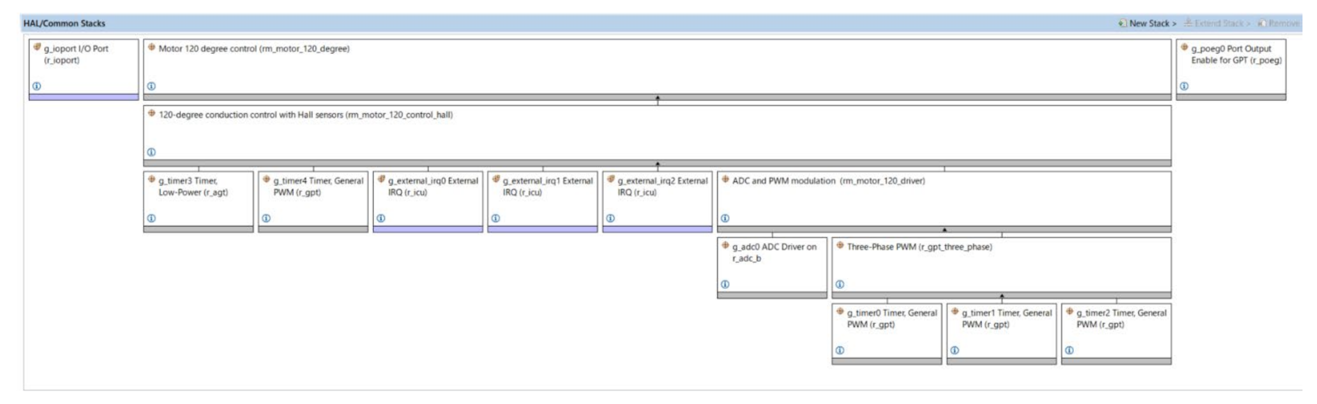
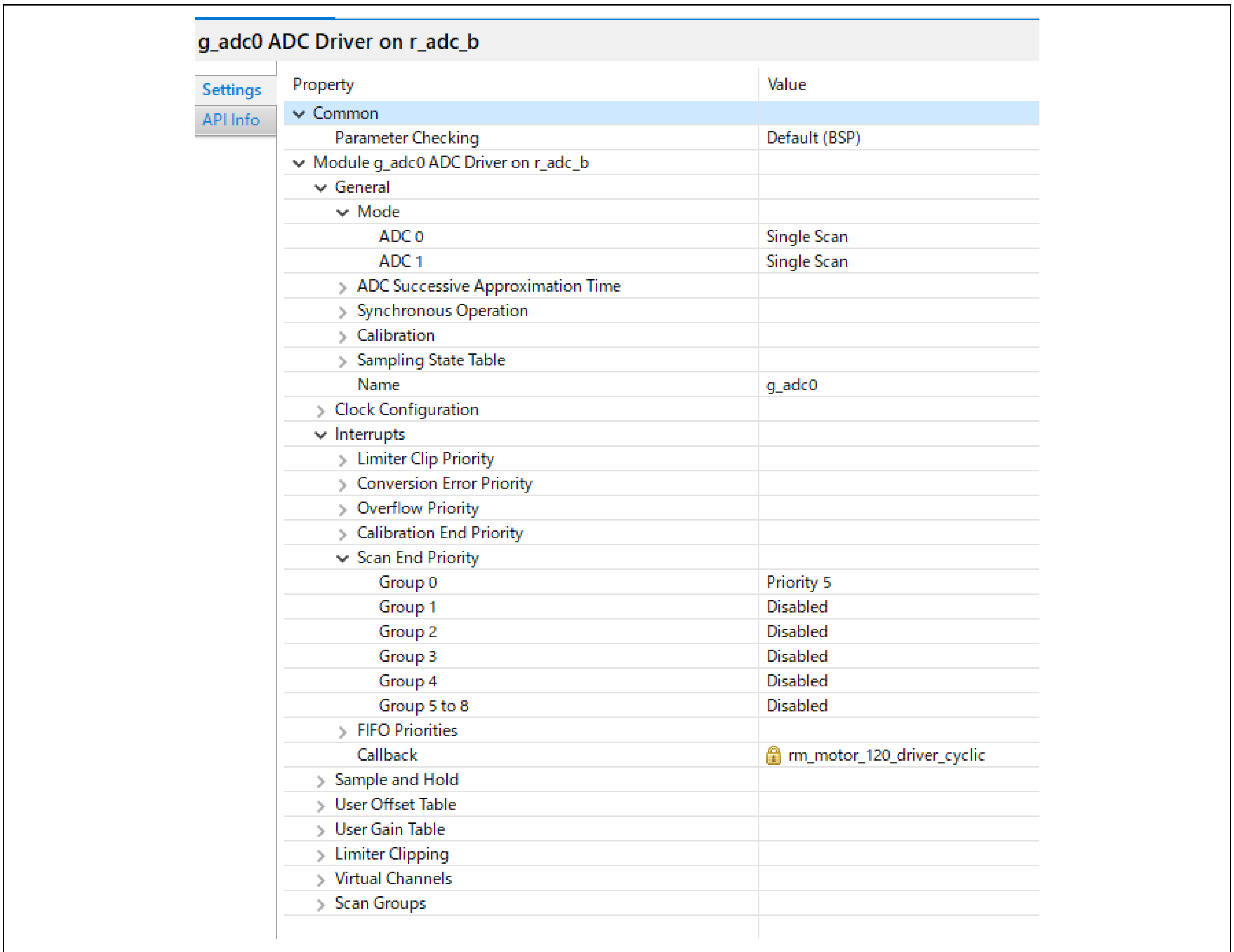


Figure 2-2 Overall FSP Stacks diagram



g_adc0 ADC Driver on r_adc_b

Settings
API Info

Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_adc0 ADC Driver on r_adc_b	
General	
Mode	
ADC 0	Single Scan
ADC 1	Single Scan
ADC Successive Approximation Time	
Synchronous Operation	
Calibration	
Sampling State Table	
Name	g_adc0
Clock Configuration	
Interrupts	
Limiter Clip Priority	
Conversion Error Priority	
Overflow Priority	
Calibration End Priority	
Scan End Priority	
Group 0	Priority 5
Group 1	Disabled
Group 2	Disabled
Group 3	Disabled
Group 4	Disabled
Group 5 to 8	Disabled
FIFO Priorities	
Callback	rm_motor_120_driver_cyclic
Sample and Hold	
User Offset Table	
User Gain Table	
Limiter Clipping	
Virtual Channels	
Scan Groups	

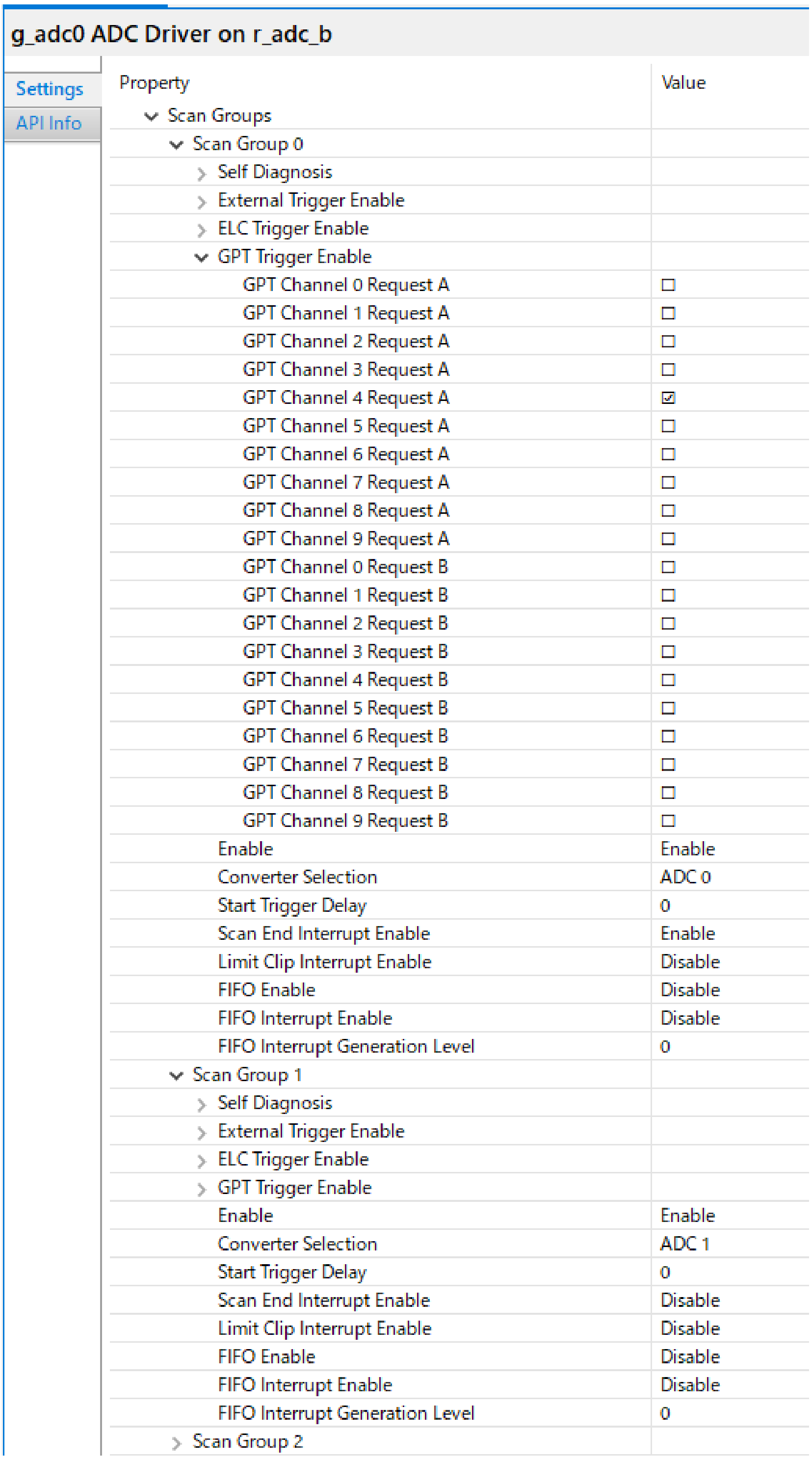
Figure 2-3 FSP configuration of ADC driver [1/3]

g_adc0 ADC Driver on r_adc_b

Settings
API Info

Property	Value
Virtual Channel 0	
Scan Group	Scan Group 0
Channel Select	AN000
Sampling State Table ID	Sampling State Entry 0
Channel Gain Table	Disabled
Channel Offset Table	Disabled
Add/Average Mode	Disabled
Add/Average Count	1-time conversion (Normal Conversion)
Limit Clip Table Id	Disabled
Conversion Resolution Format Select	12-bit Data Format
Virtual Channel 1	
Scan Group	Scan Group 0
Channel Select	AN002
Sampling State Table ID	Sampling State Entry 0
Channel Gain Table	Disabled
Channel Offset Table	Disabled
Add/Average Mode	Disabled
Add/Average Count	1-time conversion (Normal Conversion)
Limit Clip Table Id	Disabled
Conversion Resolution Format Select	12-bit Data Format
Virtual Channel 2	
Scan Group	Scan Group 0
Channel Select	AN004
Sampling State Table ID	Sampling State Entry 0
Channel Gain Table	Disabled
Channel Offset Table	Disabled
Add/Average Mode	Disabled
Add/Average Count	1-time conversion (Normal Conversion)
Limit Clip Table Id	Disabled
Conversion Resolution Format Select	12-bit Data Format
Virtual Channel 3	
Scan Group	Scan Group 1
Channel Select	AN006
Sampling State Table ID	Sampling State Entry 0
Channel Gain Table	Disabled
Channel Offset Table	Disabled
Add/Average Mode	Disabled
Add/Average Count	1-time conversion (Normal Conversion)
Limit Clip Table Id	Disabled
Conversion Resolution Format Select	12-bit Data Format
Virtual Channel 4	
Scan Group	Scan Group 1
Channel Select	AN008
Sampling State Table ID	Sampling State Entry 0
Channel Gain Table	Disabled
Channel Offset Table	Disabled
Add/Average Mode	Disabled
Add/Average Count	1-time conversion (Normal Conversion)
Limit Clip Table Id	Disabled

Figure 2-4 FSP configuration of ADC driver [2/3]



g_adc0 ADC Driver on r_adc_b

Settings
API Info

Property	Value
⌄ Scan Groups	
⌄ Scan Group 0	
› Self Diagnosis	
› External Trigger Enable	
› ELC Trigger Enable	
⌄ GPT Trigger Enable	
GPT Channel 0 Request A	☐
GPT Channel 1 Request A	☐
GPT Channel 2 Request A	☐
GPT Channel 3 Request A	☐
GPT Channel 4 Request A	☑
GPT Channel 5 Request A	☐
GPT Channel 6 Request A	☐
GPT Channel 7 Request A	☐
GPT Channel 8 Request A	☐
GPT Channel 9 Request A	☐
GPT Channel 0 Request B	☐
GPT Channel 1 Request B	☐
GPT Channel 2 Request B	☐
GPT Channel 3 Request B	☐
GPT Channel 4 Request B	☐
GPT Channel 5 Request B	☐
GPT Channel 6 Request B	☐
GPT Channel 7 Request B	☐
GPT Channel 8 Request B	☐
GPT Channel 9 Request B	☐
Enable	Enable
Converter Selection	ADC 0
Start Trigger Delay	0
Scan End Interrupt Enable	Enable
Limit Clip Interrupt Enable	Disable
FIFO Enable	Disable
FIFO Interrupt Enable	Disable
FIFO Interrupt Generation Level	0
⌄ Scan Group 1	
› Self Diagnosis	
› External Trigger Enable	
› ELC Trigger Enable	
› GPT Trigger Enable	
Enable	Enable
Converter Selection	ADC 1
Start Trigger Delay	0
Scan End Interrupt Enable	Disable
Limit Clip Interrupt Enable	Disable
FIFO Enable	Disable
FIFO Interrupt Enable	Disable
FIFO Interrupt Generation Level	0
› Scan Group 2	

Figure 2-5 FSP configuration of ADC driver [3/3]

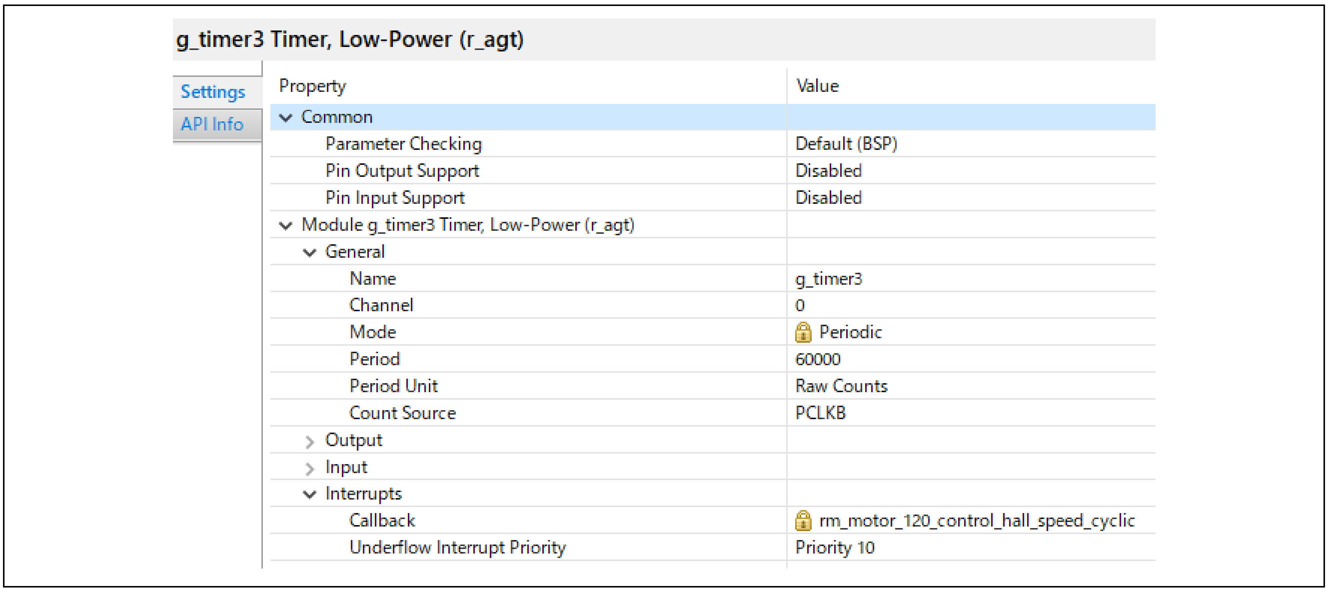


Figure 2-6 FSP configuration of AGT driver

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	4
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 Low
GTIOCB Output Enabled	True
GTIOCB Stop Level	Pin Level High
Input	
Interrupts	
Extra Features	
Output Disable	
ADC Trigger	
Start Event Trigger (GPTE/GPTEH only)	
Trigger Event A/D Converter Start Request A Durin	☐
Trigger Event A/D Converter Start Request A Durin	☑
Trigger Event A/D Converter Start Request B Durin	☐
Trigger Event A/D Converter Start Request B Durin	☐
Dead Time	
ADC Trigger (GPTE/GPTEH only)	
Interrupt Skipping (GPTE/GPTEH only)	
Extra Features	Enabled
Pins	
GTIOC4A	PB04
GTIOC4B	PB05

Figure 2-7 FSP configuration of GPT driver(complementary PWM output)

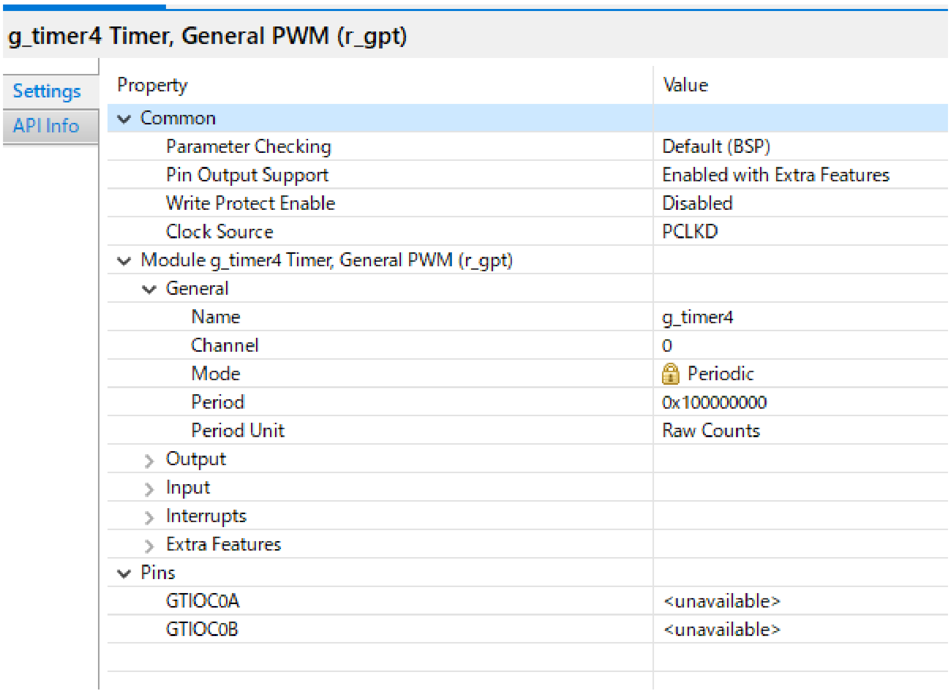
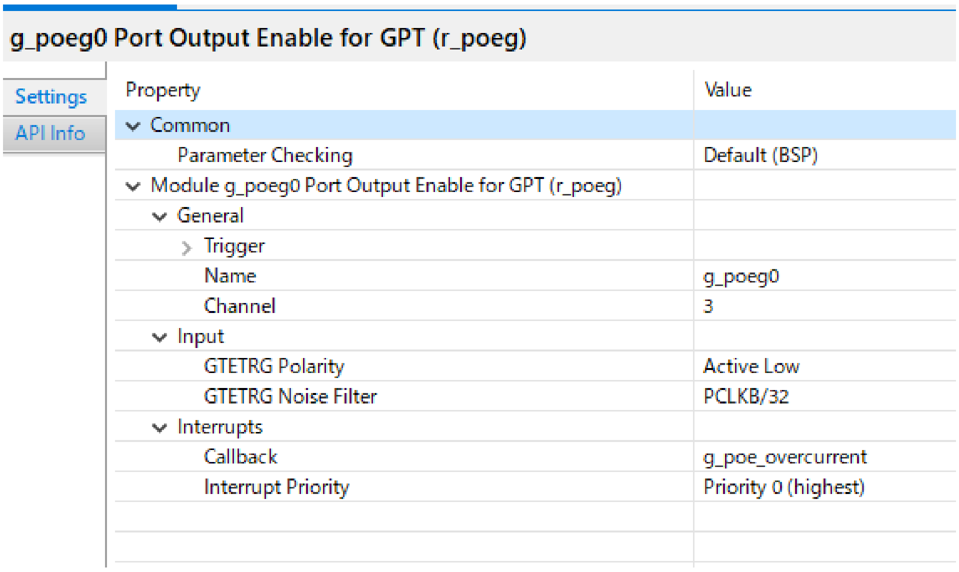


Figure 2-8 FSP configuration of GPT driver (free run timer for speed measurement)

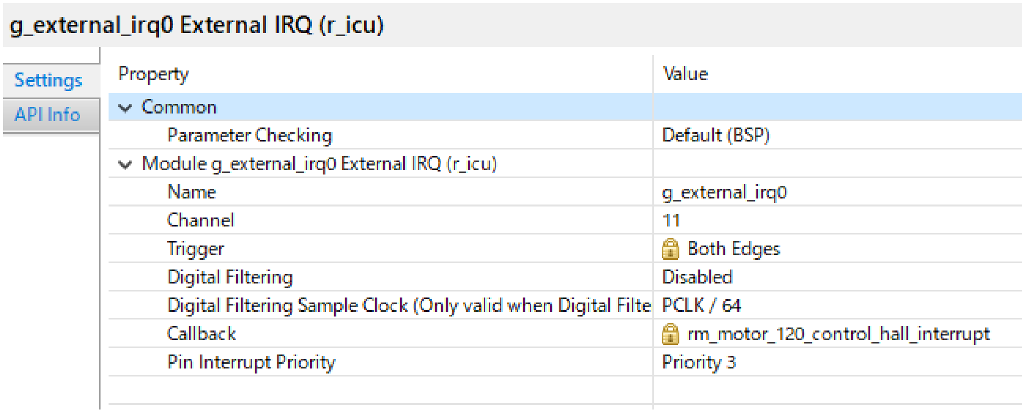


g_poeg0 Port Output Enable for GPT (r_poeg)

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	3
Input	
GTETRG Polarity	Active Low
GTETRG Noise Filter	PCLKB/32
Interrupts	
Callback	g_poe_overcurrent
Interrupt Priority	Priority 0 (highest)

Figure 2-9 FSP configuration of POEG driver



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	11
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

Figure 2-10 FSP configuration of IRQ driver

2.2.3.2 RA4T1

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

U-phase current (Iu), V-phase current (Iv), 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).

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

Used as a 1ms interval timer.

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

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

On the channel 1, 2, and 3, output with dead time is performed by using the complementary PWM Output Operating Mode.

(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 GTETRGB port is detected).

(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

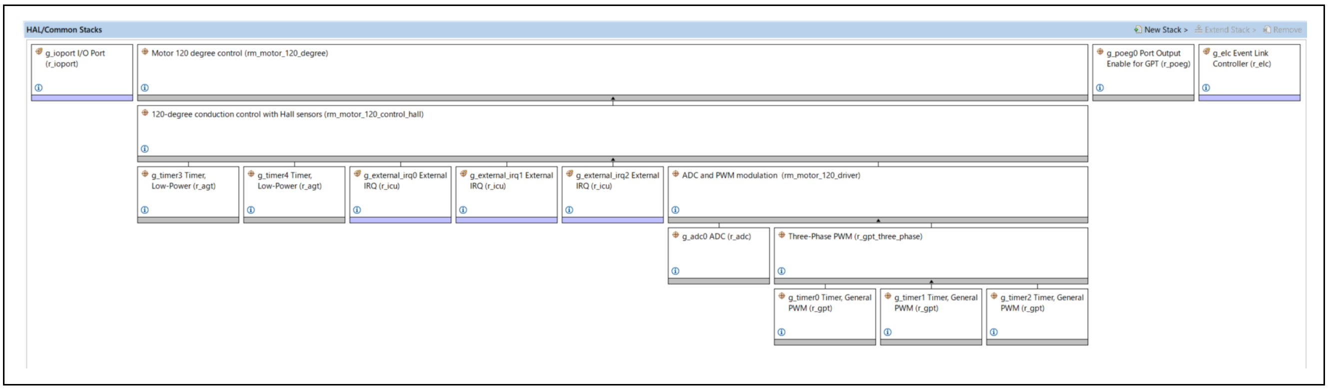


Figure 2-11 Overall FSP Stacks diagram

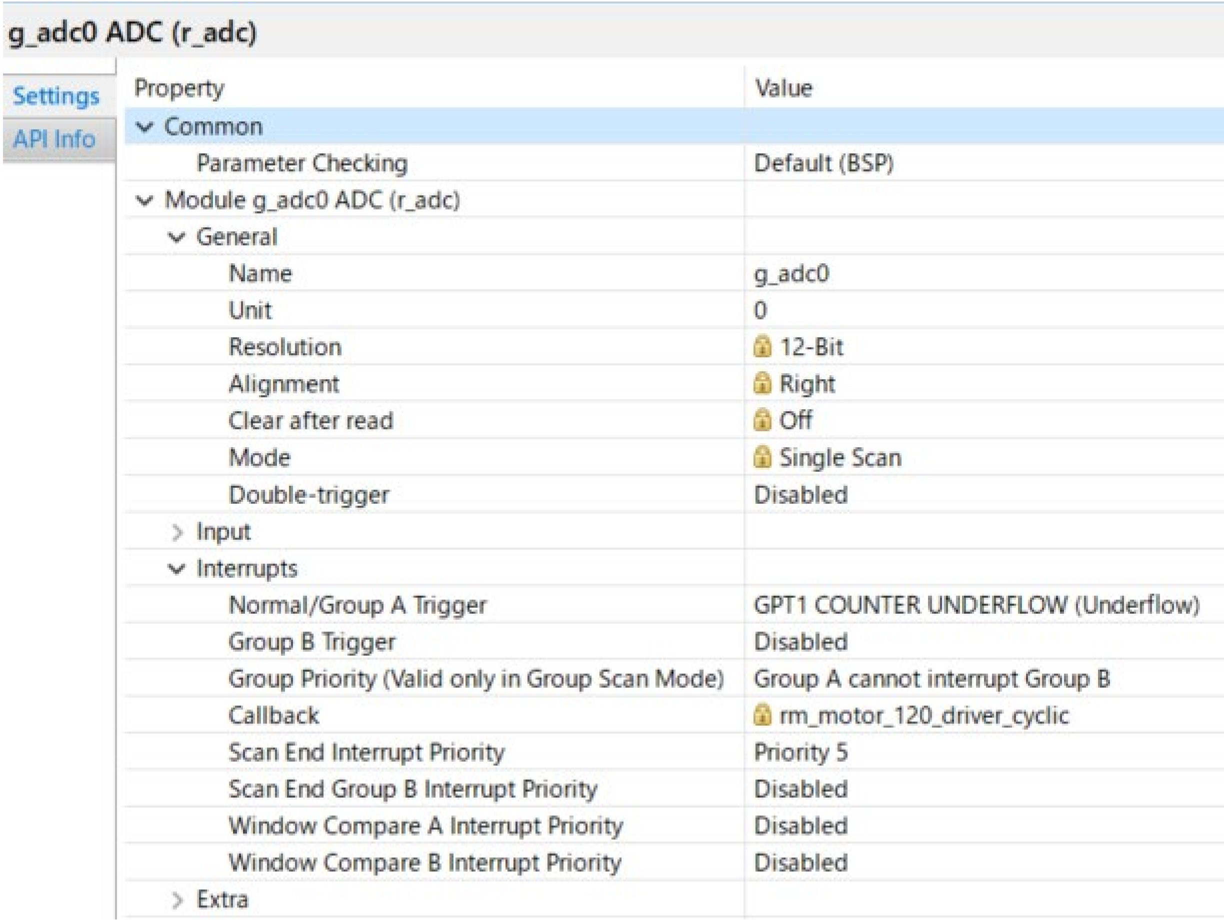


Figure 2-12 FSP Configuration of ADC Driver [1/2]

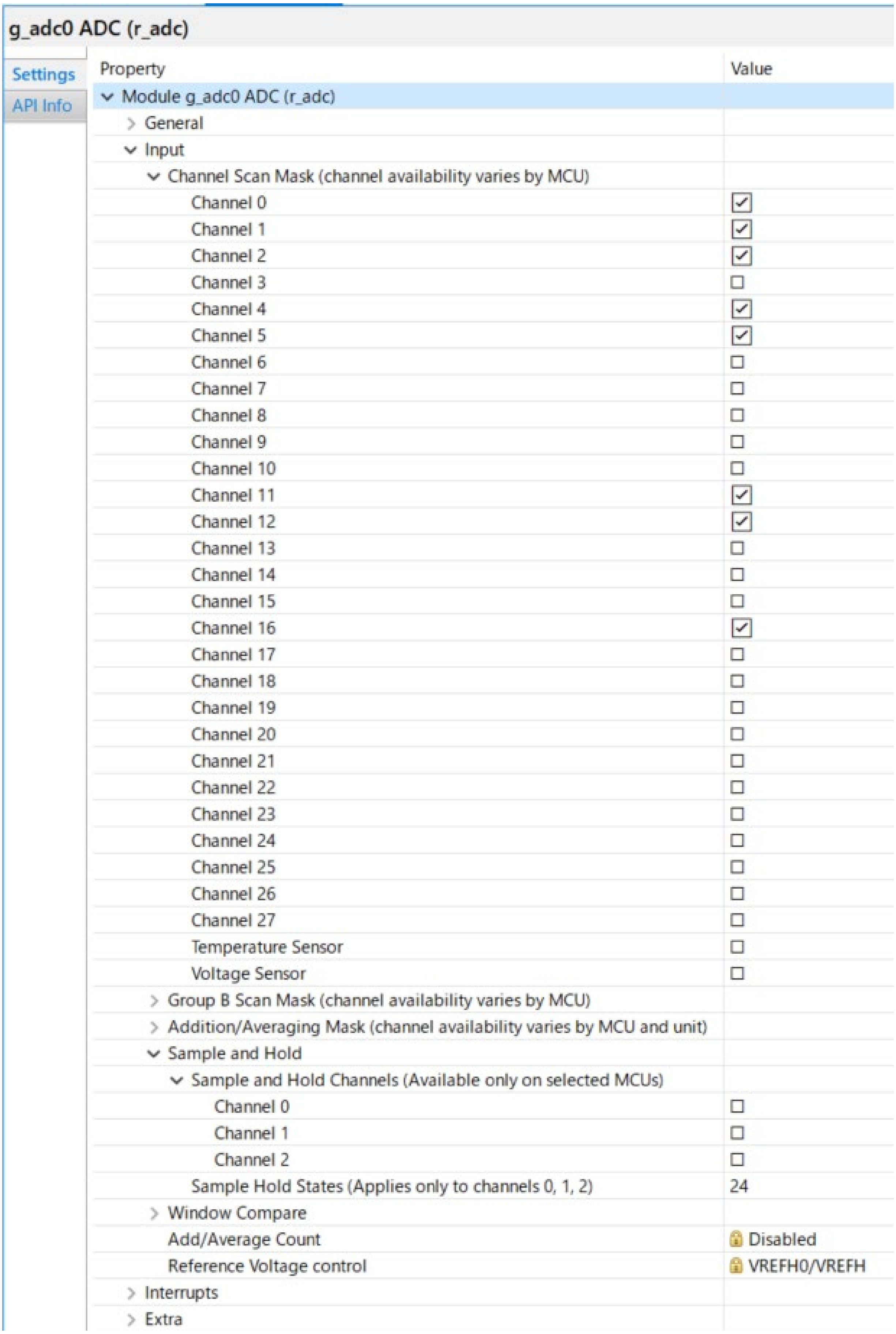
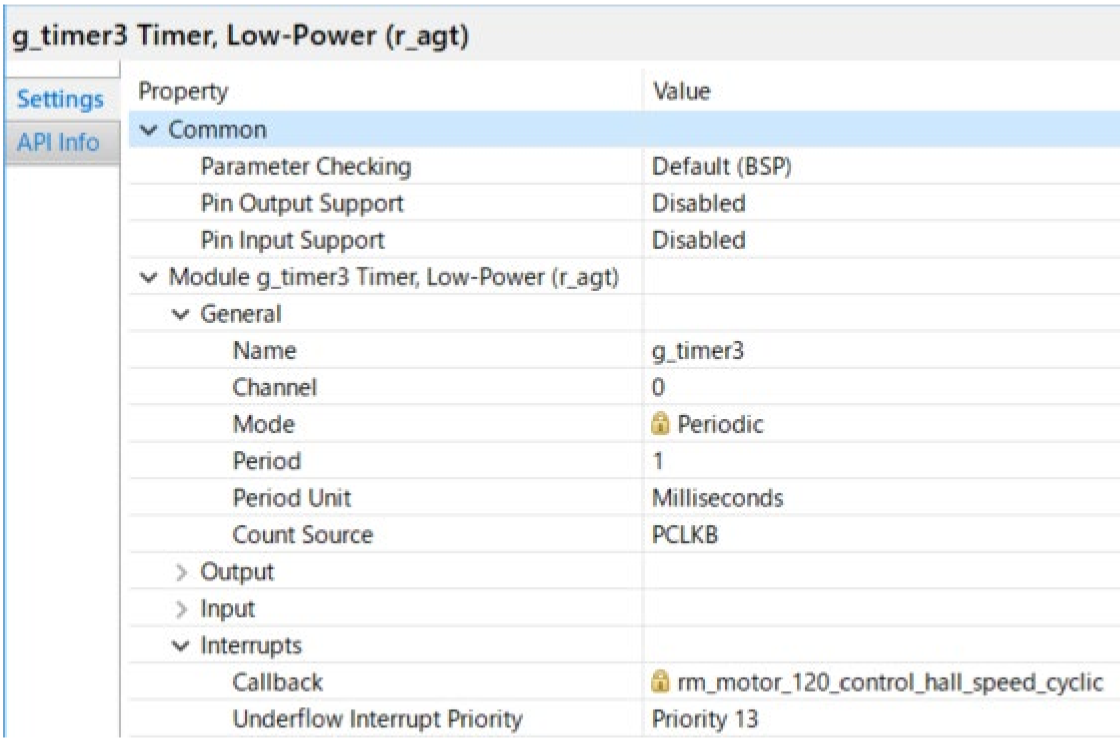


Figure 2-13 FSP Configuration of ADC Driver [2/2]

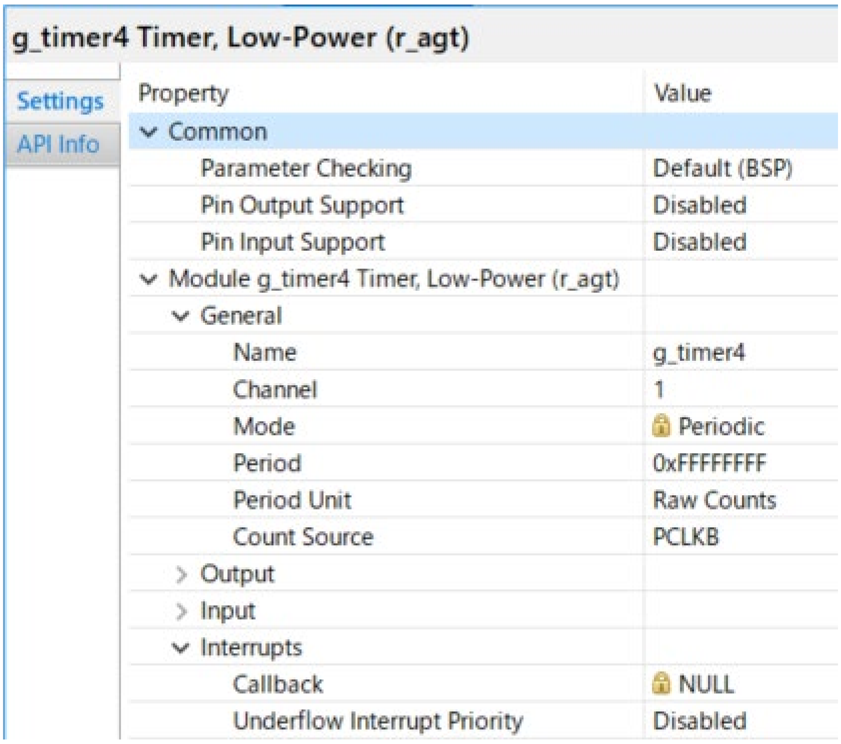


g_timer3 Timer, Low-Power (r_agt)

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	
Callback	rm_motor_120_control_hall_speed_cyclic
Underflow Interrupt Priority	Priority 13

Figure 2-14 FSP Configuration of AGT Driver

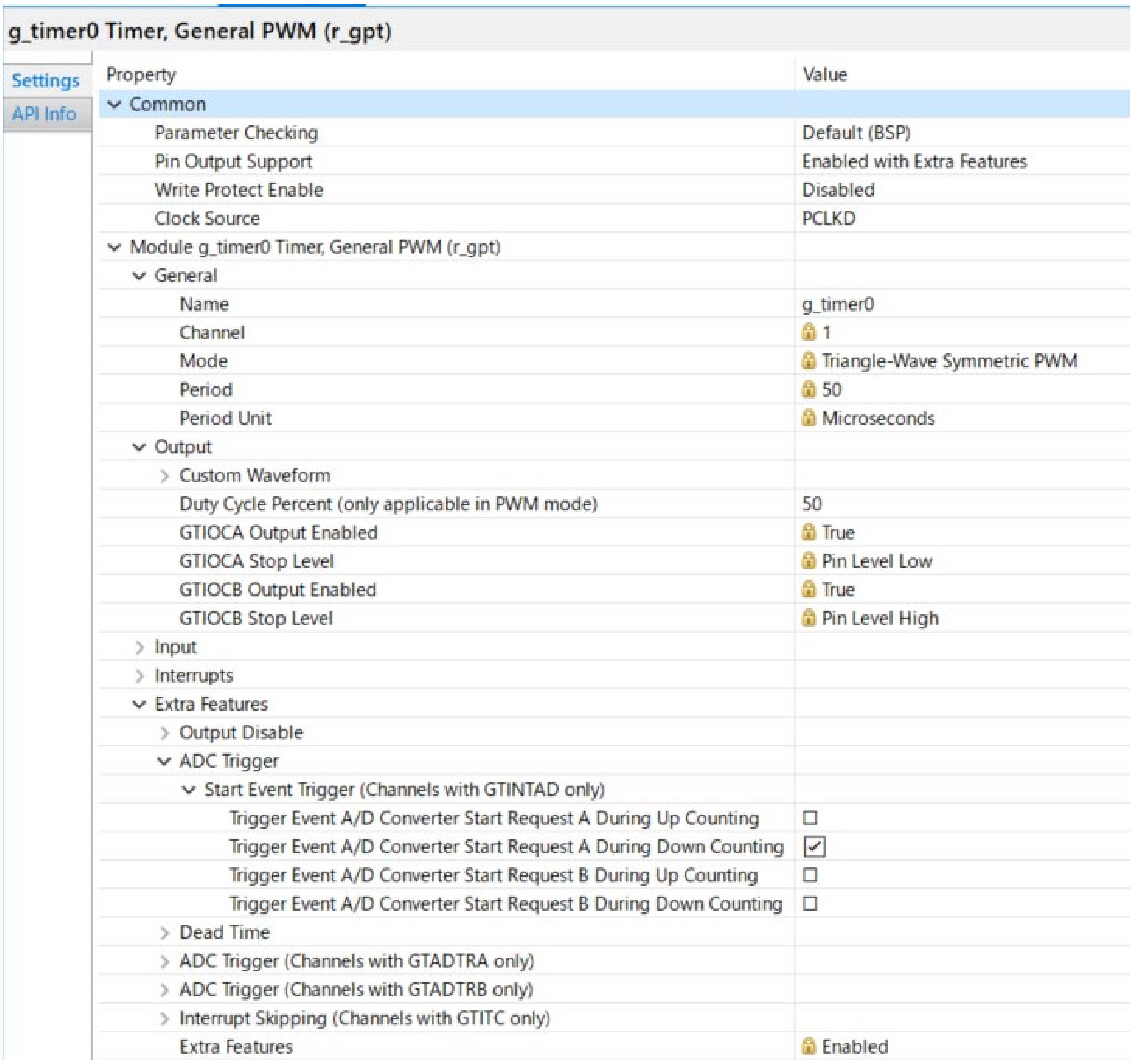


g_timer4 Timer, Low-Power (r_agt)

Settings
API Info

Property	Value
Common	
Parameter Checking	Default (BSP)
Pin Output Support	Disabled
Pin Input Support	Disabled
Module g_timer4 Timer, Low-Power (r_agt)	
General	
Name	g_timer4
Channel	1
Mode	Periodic
Period	0xFFFFFFFF
Period Unit	Raw Counts
Count Source	PCLKB
Output	
Input	
Interrupts	
Callback	NULL
Underflow Interrupt Priority	Disabled

Figure 2-15 FSP Configuration of AGT Driver (free run timer for speed measurement)

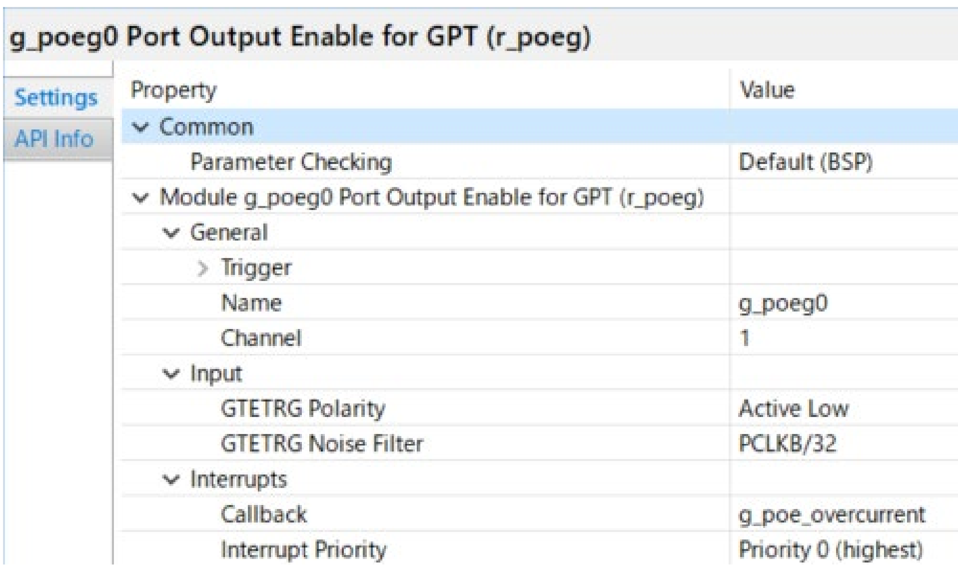


g_timer0 Timer, General PWM (r_gpt)

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	1
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 Low
GTIOCB Output Enabled	True
GTIOCB Stop Level	Pin Level High
Input	
Interrupts	
Extra Features	
Output Disable	
ADC Trigger	
Start Event Trigger (Channels with GTINTAD only)	
Trigger Event A/D Converter Start Request A During Up Counting	☐
Trigger Event A/D Converter Start Request A During Down Counting	☑
Trigger Event A/D Converter Start Request B During Up Counting	☐
Trigger Event A/D Converter Start Request B During Down Counting	☐
Dead Time	
ADC Trigger (Channels with GTADTRA only)	
ADC Trigger (Channels with GTADTRB only)	
Interrupt Skipping (Channels with GTITC only)	
Extra Features	Enabled

Figure 2-16 FSP Configuration of GPT Driver



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	1
Input	
GTETRG Polarity	Active Low
GTETRG Noise Filter	PCLKB/32
Interrupts	
Callback	g_poe_overcurrent
Interrupt Priority	Priority 0 (highest)

Figure 2-17 FSP Configuration of POEG Driver

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	12
Trigger	Both Edges
Digital Filtering	Disabled
Digital Filtering Sample Clock (Only valid when Digital Filtering is Enabled)	PCLK / 64
Callback	rm_motor_120_control_hall_interrupt
Pin Interrupt Priority	Priority 8

Figure 2-18 FSP configuration of IRQ driver

2.2.3.3 RA6T3

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

U-phase current (Iu), V-phase current (Iv), 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).

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

Used as a 1ms interval timer.

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

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

On the channel 1, 2, and 3, output with dead time is performed by using the complementary PWM Output Operating Mode.

(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 GTETRGB port is detected).

(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

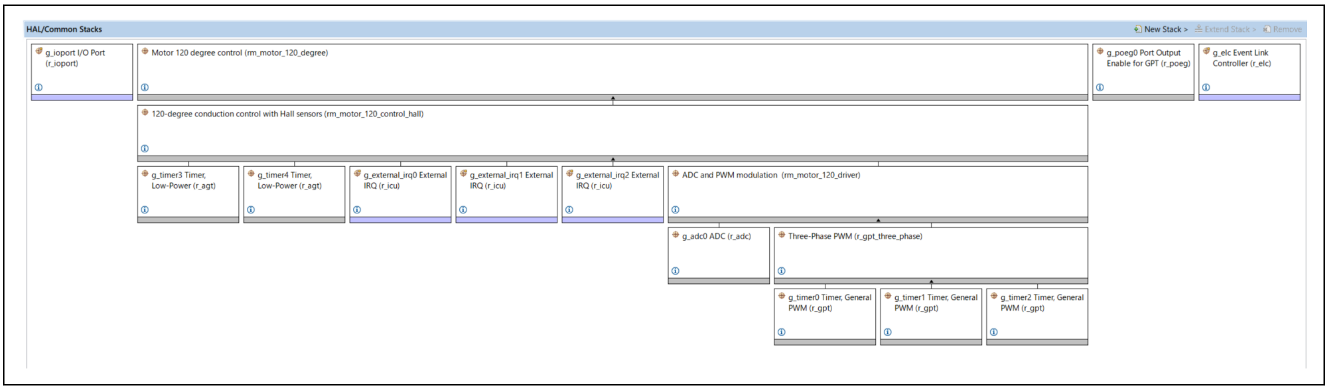
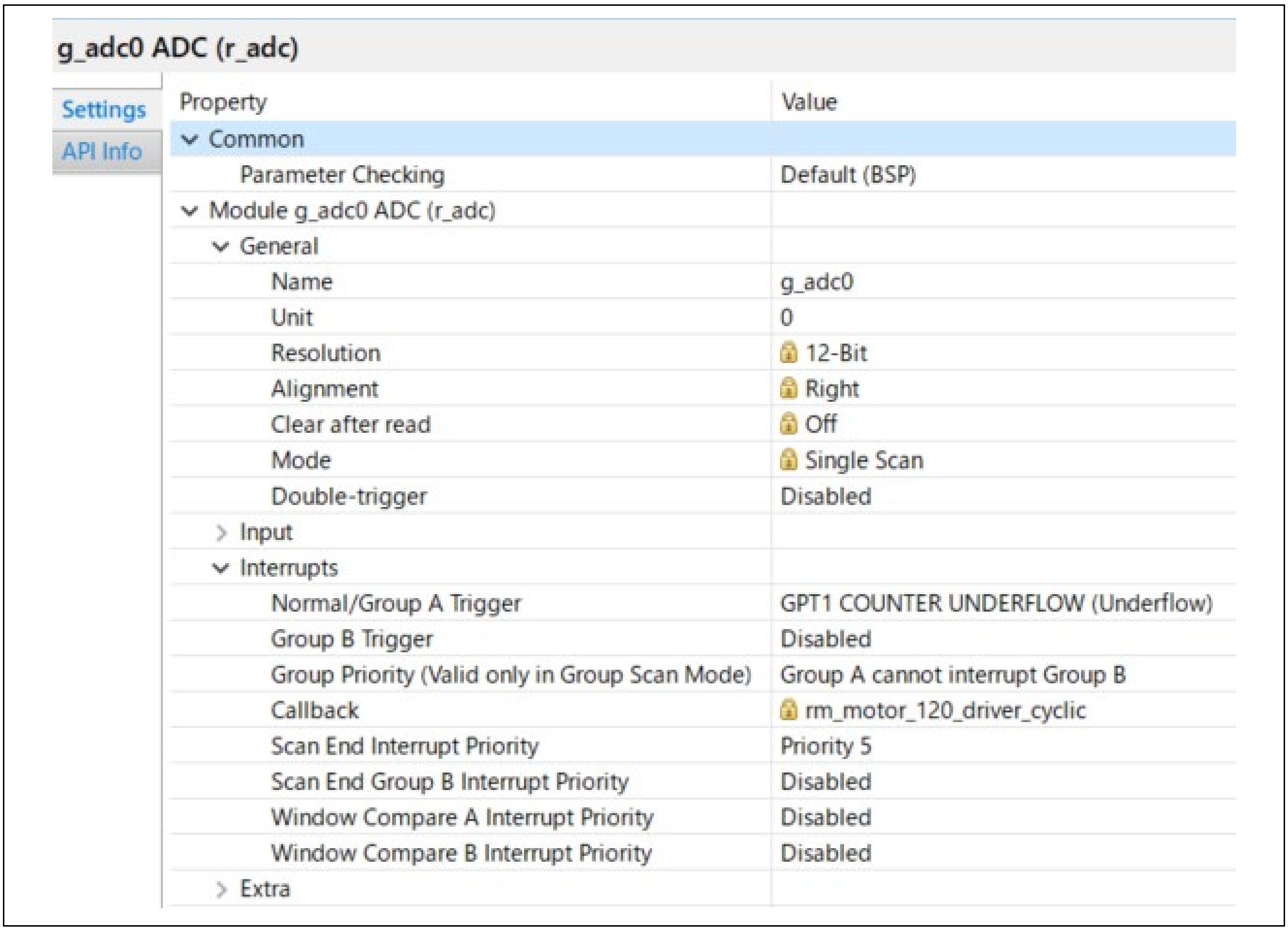


Figure 2-19 Overall FSP Stacks diagram



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	
Interrupts	
Normal/Group A Trigger	GPT1 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	

Figure 2-20 FSP Configuration of ADC Driver [1/2]

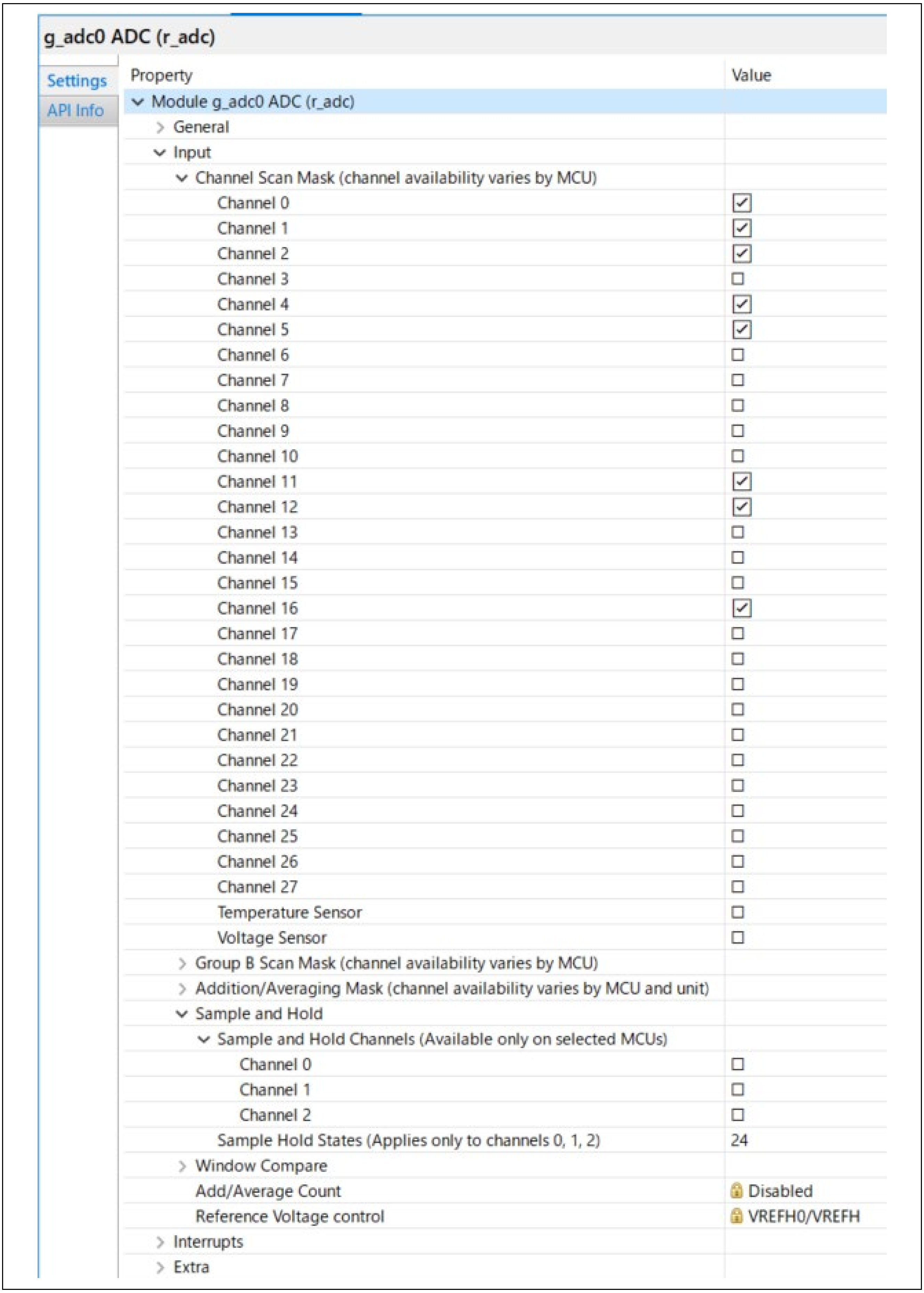
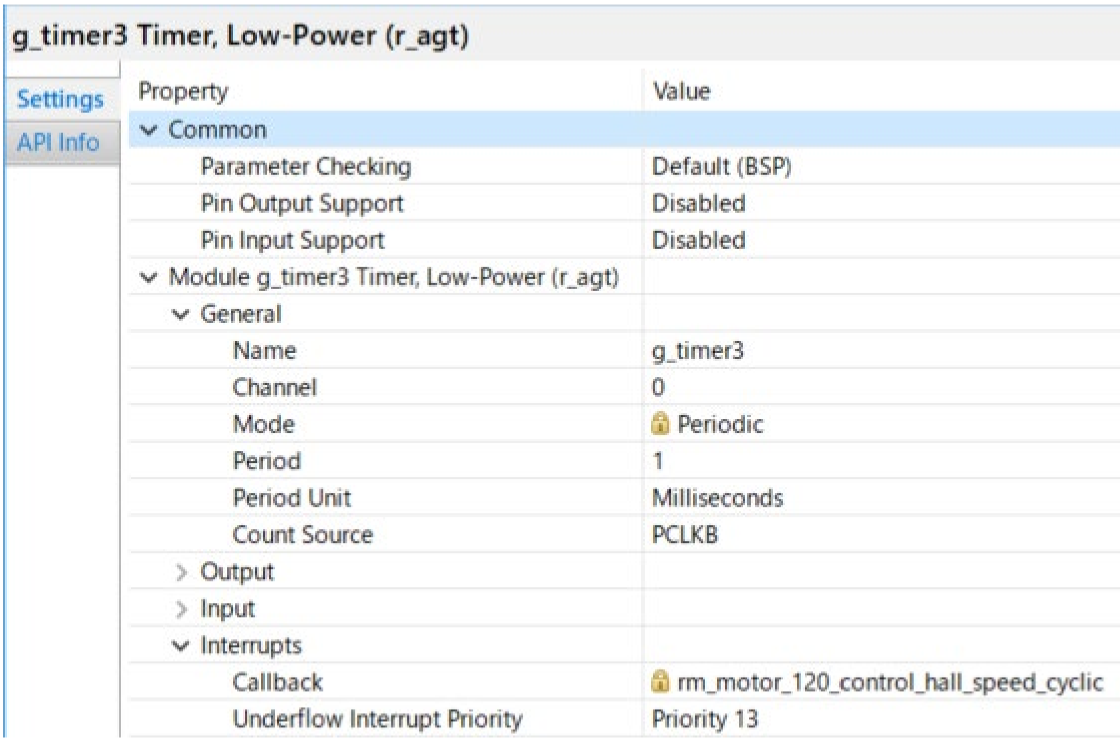


Figure 2-21 FSP Configuration of ADC Driver [2/2]

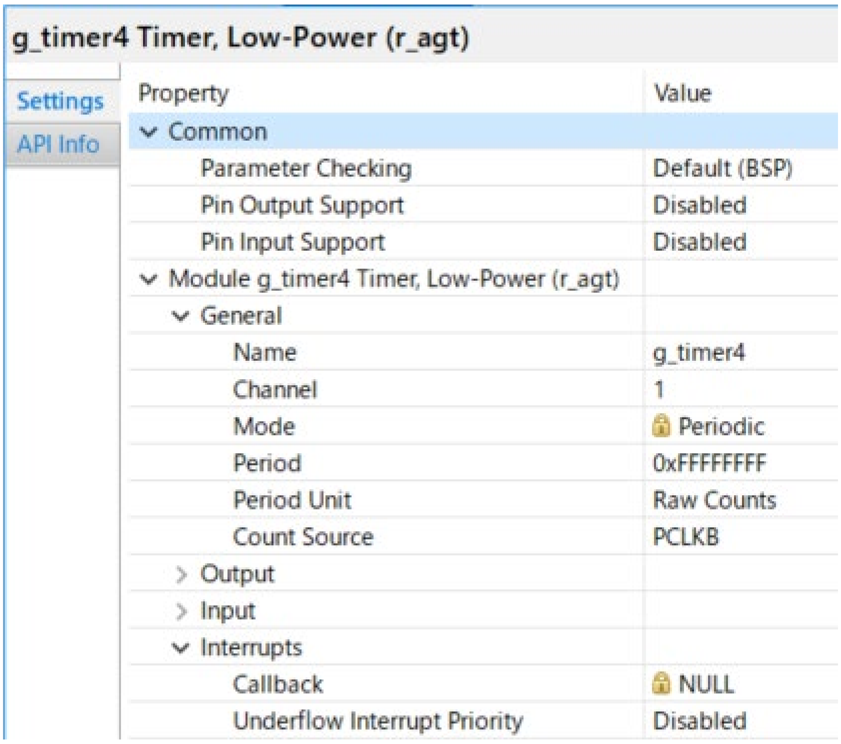


g_timer3 Timer, Low-Power (r_agt)

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	
Callback	rm_motor_120_control_hall_speed_cyclic
Underflow Interrupt Priority	Priority 13

Figure 2-22 FSP Configuration of AGT Driver



g_timer4 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_timer4 Timer, Low-Power (r_agt)	
General	
Name	g_timer4
Channel	1
Mode	Periodic
Period	0xFFFFFFFF
Period Unit	Raw Counts
Count Source	PCLKB
Output	
Input	
Interrupts	
Callback	NULL
Underflow Interrupt Priority	Disabled

Figure 2-23 FSP Configuration of AGT Driver (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	1
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 Low
GTIOCB Output Enabled	True
GTIOCB Stop Level	Pin Level High
Input	
Interrupts	
Extra Features	
Output Disable	
ADC Trigger	
Start Event Trigger (Channels with GTINTAD only)	
Trigger Event A/D Converter Start Request A During Up Counting	☐
Trigger Event A/D Converter Start Request A During Down Counting	☑
Trigger Event A/D Converter Start Request B During Up Counting	☐
Trigger Event A/D Converter Start Request B During Down Counting	☐
Dead Time	
ADC Trigger (Channels with GTADTRA only)	
ADC Trigger (Channels with GTADTRB only)	
Interrupt Skipping (Channels with GTITC only)	
Extra Features	Enabled

Figure 2-24 FSP Configuration of GPT Driver

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	1
Input	
GTETRG Polarity	Active Low
GTETRG Noise Filter	PCLKB/32
Interrupts	
Callback	g_poe_overcurrent
Interrupt Priority	Priority 0 (highest)

Figure 2-25 FSP Configuration of POEG Driver

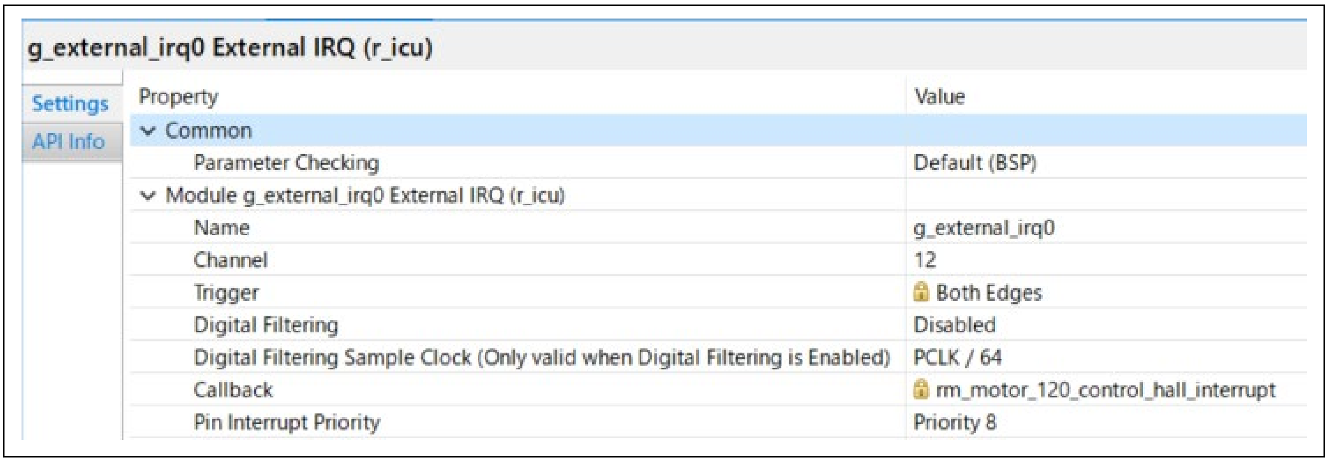


Figure 2-26 FSP configuration of IRQ driver

2.3 Software structure

2.3.1 Software file structure

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

Table 2-6 File and folder configuration[1/2]

Folder	Subfolder	File	Remarks
ra_cfg			Generated config header
ra_gen			Generated register setting, main function etc
ra	arm		CMSIS source code
	board		Function definition for board
	fsp/inc/api	bsp_api.h	BSP API definition
		r_adc_api.h	AD API definition
		r_elc_api.h	ELC API definition
		r_external_irq_api.h	IRQ API definition
		r_ioport_api.h	I/O API definition
		r_poeg_api.h	POEG API definition
		r_three_phase_api.h	3phase PWM API definition
		r_timer_api.h	Timer API definition
		r_transfer_api.h	Transfer API definition
		rm_motor_120_control_api.h	120-degree conduction control API definition
		rm_motor_120_driver_api.h	120-degree conduction control driver API definition
		rm_motor_angle_api.h	Angle API definition
		rm_motor_api.h	Motor API definition
		rm_motor_current_api.h	Current API definition
		rm_motor_driver_api.h	Motor driver API definition
		rm_motor_position_api.h	Position API definition
		rm_motor_speed_api.h	Speed API definition
	fsp/inc/instances	r_adc_b.h(RA6T2) r_adc.h(RA4T1,RA6T3)	Function definition for AD
		r_agt.h	Function definition for AGT
		r_elc.h(Only RA4T1 and RA6T3)	Function definition for ELC
		r_gpt_three_phase.h	Function definition for 3 Phase PWM
		r_icu.h	Function definition for IRQ
		r_gpt.h	Function definition for GPT

Table 2-7 File and folder configuration[2/2]

Folder	Subfolder	File	Remarks
ra	fsp/inc/instances	r_ioport.h	Function definition for I/O
		r_poeg.h	Function definition for POEG
		rm_motor_120_control_hall.h	Hall 120-degree conduction control related definition
		rm_motor_120_degree.h	120-degree conduction control related definition
		rm_motor_120_driver.h	120-degree conduction control driver related definition
	fsp/src	bsp	BSP driver
		r_adc_b/r_adc_b.c(RA6T2) r_adc/r_adc.c(RA4T1,RA6T3)	AD driver
		r_agt/r_agt.c	AGT driver
		r_elc/r_elc.c(Only RA4T1 and RA6T3)	ELC driver
		r_gpt/r_gpt.c	GPT driver
		r_gpt_three_phase/ r_gpt_three_phase.c	3 phase PWM driver
		r_icu/r_icu.c	IRQ driver
		r_ioport/r_ioport.c	I/O driver
		r_poeg/r_poeg.c	POEG driver
		rm_motor_120_control_hall/ rm_motor_120_control_hall.c	Hall 120-degree conduction control
		rm_motor_120_degree/ rm_motor_120_degree.c	120-degree conduction control
		rm_motor_120_driver/ rm_motor_120_driver.c	120-degree conduction control driver
src	application/main	mtr_main.h , mtr_main.c	User main function
		r_mtr_control_parameter.h (Only RA4T1 and RA6T3)	Control parameters definition
		r_mtr_motor_parameter.h (Only RA4T1 and RA6T3)	Motor parameters definition
	application/user_interface/ics	r_mtr_ics.h , r_mtr_ics.c	Function definition for Analyzer
		ICS2_RA6T2.h , ICS2_RA4T1.h , ICS2_RA6T3.h	Function definition for GUI tool
		ICS2_RA6T2.o , ICS2_RA4T1.o , ICS2_RA6T3.o	Communication library for GUI tool

2.3.2 Module configuration

Figure 2-27 Module Configuration show module configuration of software.

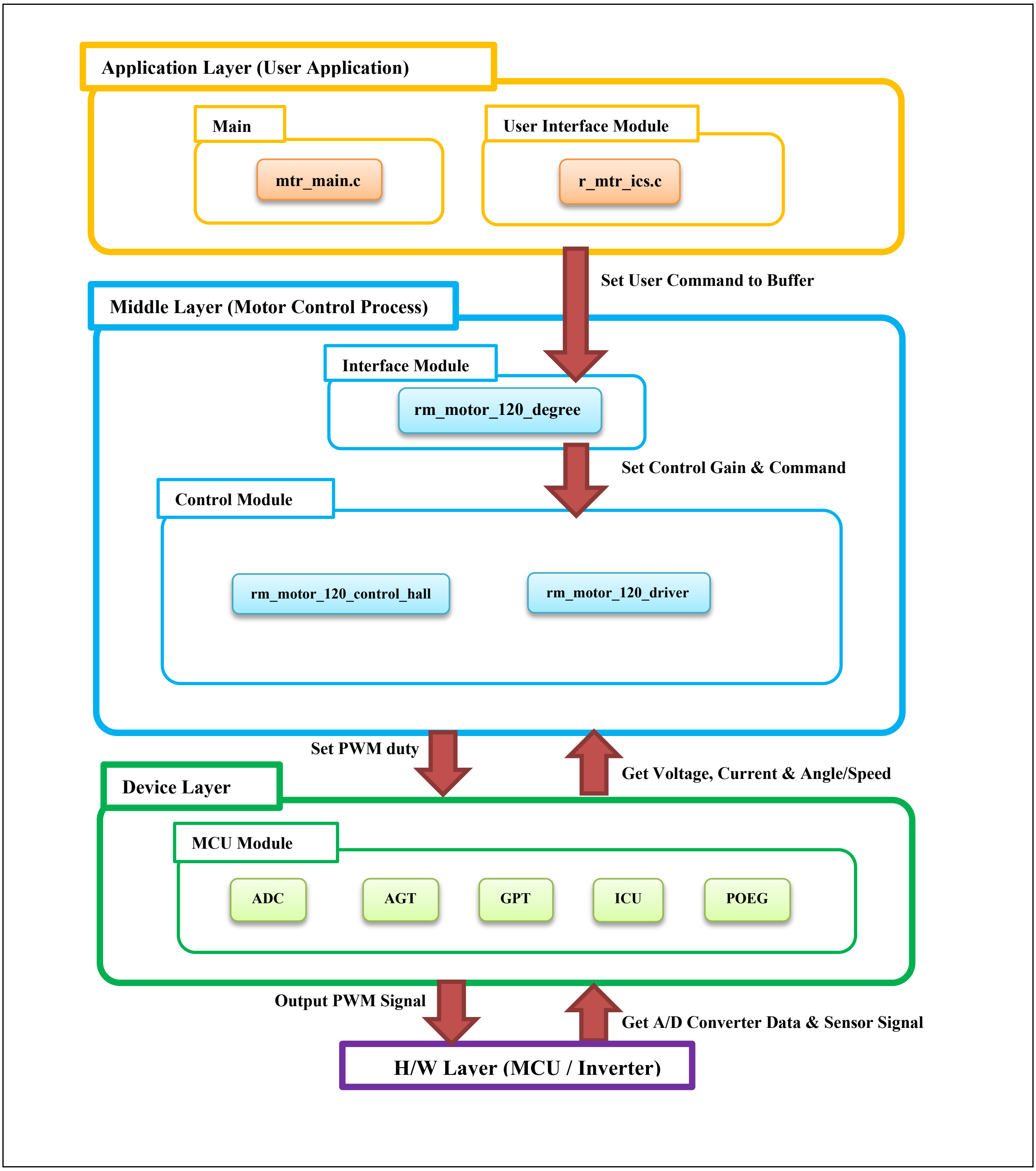


Figure 2-27 Module Configuration

2.4 Software specifications

Table 2-8 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 2-8 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	RA6T2: 240MHz] RA6T3: 200 [MHz] RA4T1: 100 [MHz]	
Carrier frequency (PWM)	20 [kHz] (Carrier cycle：50 [μs])	
dead time	2 [μs]	
Control frequency(Speed)	RA6T2: 1 [ms] RA6T3: 1 [ms] RA4T1: 1 [ms]	
Rotation speed control range	Both CW and CCW: 550 [rpm] to 2400 [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 3.54(=1.67*sqrt (2)*1.5) [A](monitored in current control period) 2. Inverter bus voltage exceeds 60 [V] (monitored in current control period) 3. Inverter bus voltage is less than 8 [V] (monitored in current control period) 4. Rotation speed exceeds 4500 [rpm] (monitored in current control period) 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 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 2-9 Interrupt priority

Interrupt level	Priority	function
15	Min	
14		
13		
12		
11		
10		AGT0 INT Speed control Interrupt
9		
8		
7		
6		
5		ADC0 ADI0(RA6T2) ADC0 SCAN END(RA4T1,RA6T3) A/D conversion complete interrupt
4		
3		ICU IRQ11 , ICU IRQ10 , ICU IRQ1 (RA6T2) ICU IRQ12 , ICU IRQ11 , ICU IRQ13 (RA4T1, RA6T3) Hall sensor interrupt
2		
1		
0	Max	POEG3 EVENT(RA6T2) POEG1 EVENT(RA4T1,RA6T3) Over current error interrupt

Allocations

Interrupt	Event	ISR
0	AGT0 INT (AGT interrupt)	agt_int_isr
1	ICU IRQ10 (External pin interrupt 10)	r_icu_isr
2	ICU IRQ11 (External pin interrupt 11)	r_icu_isr
3	ICU IRQ1 (External pin interrupt 1)	r_icu_isr
4	ADC0 ADI0 (End of A/D scanning operation(Gr.0))	adc_b_adi0_isr
5	POEG3 EVENT (Port Output disable interrupt D)	poeg_event_isr

Figure 2-28 RA6T2 FSP configuration of interrupt

Allocations

Interrupt	Event	ISR
0	AGT0 INT (AGT interrupt)	agt_int_isr
1	ICU IRQ12 (External pin interrupt 12)	r_icu_isr
2	ICU IRQ11 (External pin interrupt 11)	r_icu_isr
3	ICU IRQ13 (External pin interrupt 13)	r_icu_isr
4	ADC0 SCAN END (A/D scan end interrupt)	adc_scan_end_isr
5	POEG1 EVENT (Port Output disable interrupt B)	poeg_event_isr

Figure 2-29 RA4T1/RA6T3 FSP Interrupts Configuration

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.

"High" level → Motor Start

"Low" 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)	
Rotation speed command value	CW	0 [rpm]～2400[rpm]：0800H～0FFFH
	CCW	0 [rpm]～2400[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 low voltage (PWM stops in case of abnormality).

Table 3-2 Inverter Bus Voltage Conversion Ratio

Item	Conversion ratio (Inverter bus voltage：A/D conversion value)
Inverter bus voltage	0 [V]～73.26 [V]：0000H～0FFFH

(3). U, V, and W phase current

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

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

Item	Conversion ratio (U, V, and W phase current：A/D conversion value)
U, V, and W phase current	-8.25 [A] to 8.25 [A]: 0000H to 0FFFH (Note) Current = (3.3V-1.65V)/(0.01Ohm＊20) =8.25A

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.

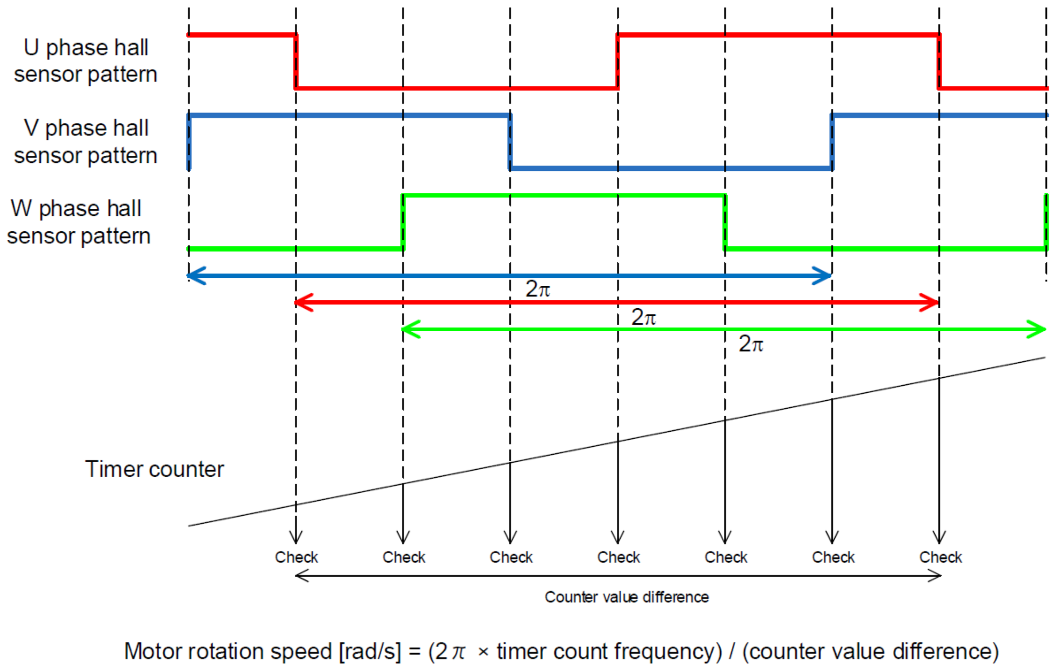


Figure 3-1 Motor Rotation Speed Calculation Method

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.

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

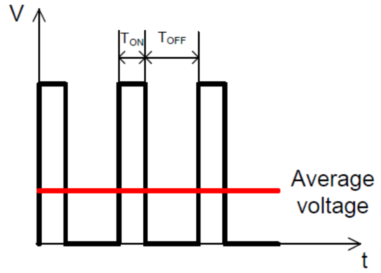


Figure 3-2 PWM Control

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.

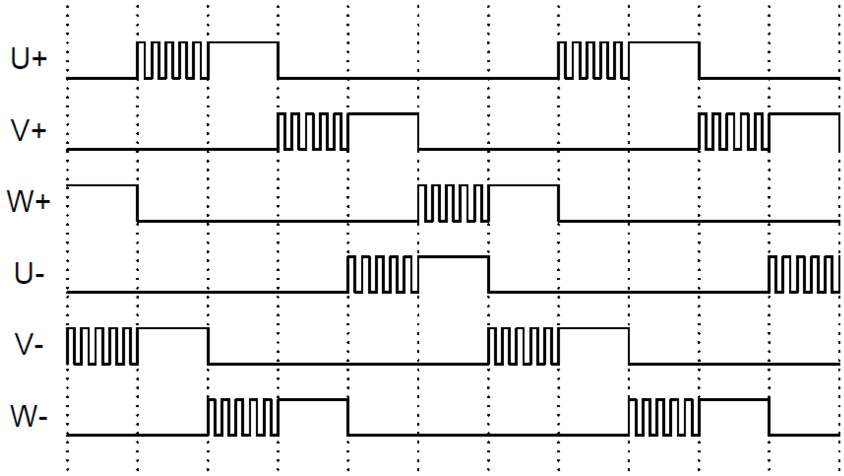


Figure 3-3 non-complimentary first 60-degree Chopping

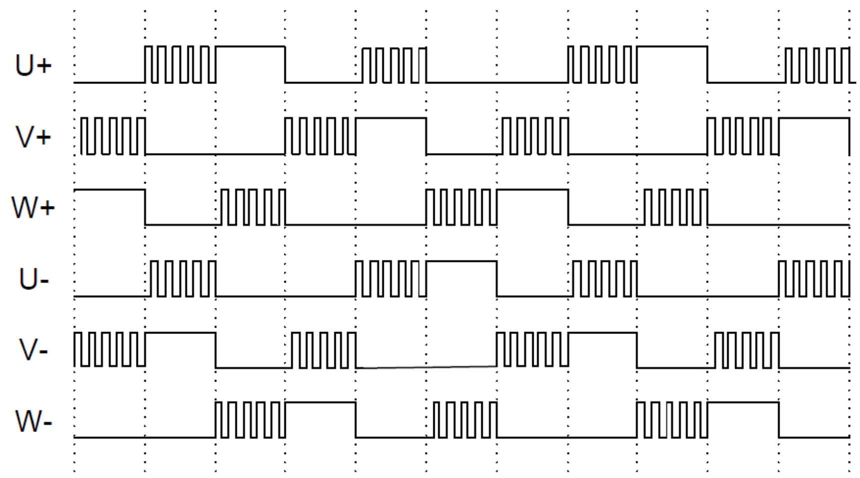
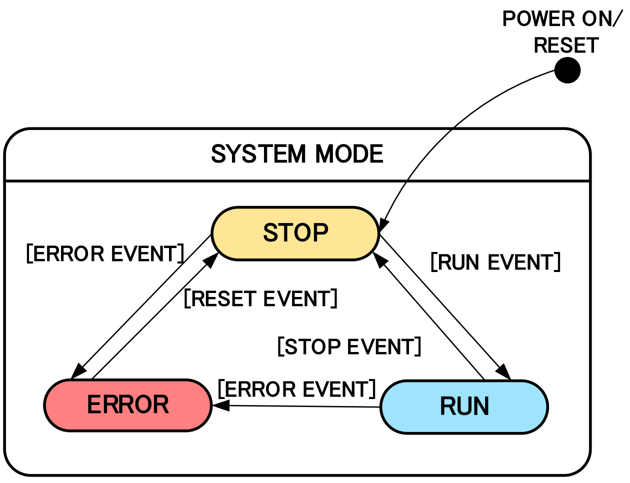


Figure 3-4 Complimentary first 60-degree Chopping

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.

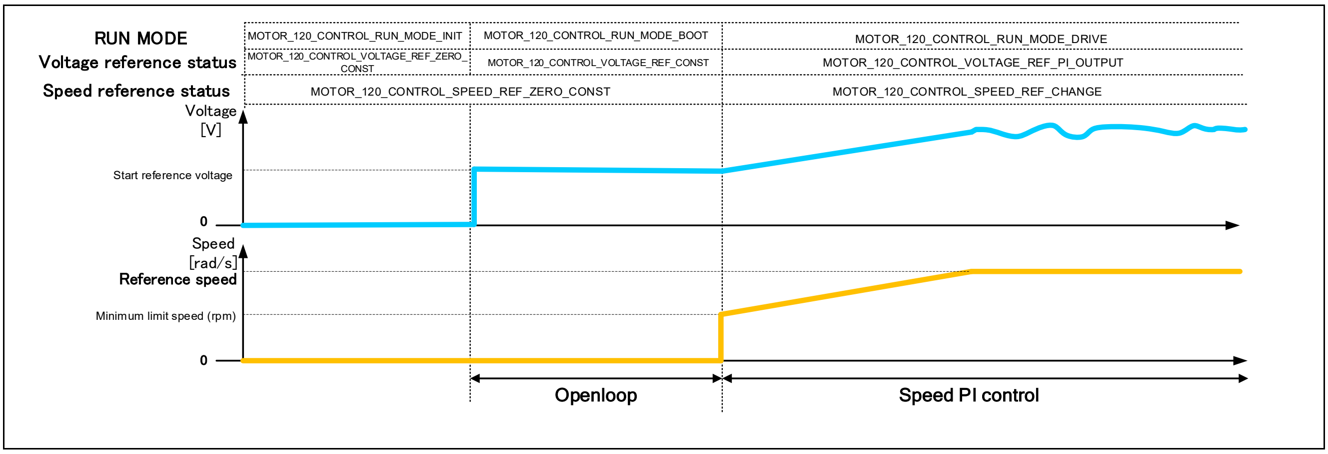


Figure 3-6 Start-up Method (Example)

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 name	Threshold		Monitoring cycle
Over current error	Over current limit [A]	3.54	Current control period
Over voltage error	Over voltage limit [V]	60	Current control period
Low voltage error	Low voltage limit [V]	8	Current control period
Over speed error	Speed limit [rpm]	4500	Current control period
Timeout error of hall interrupt detection	Timeout value [ms]	200	

3.1.8 AD triggers

Shows the timing of AD triggers and scan groups.

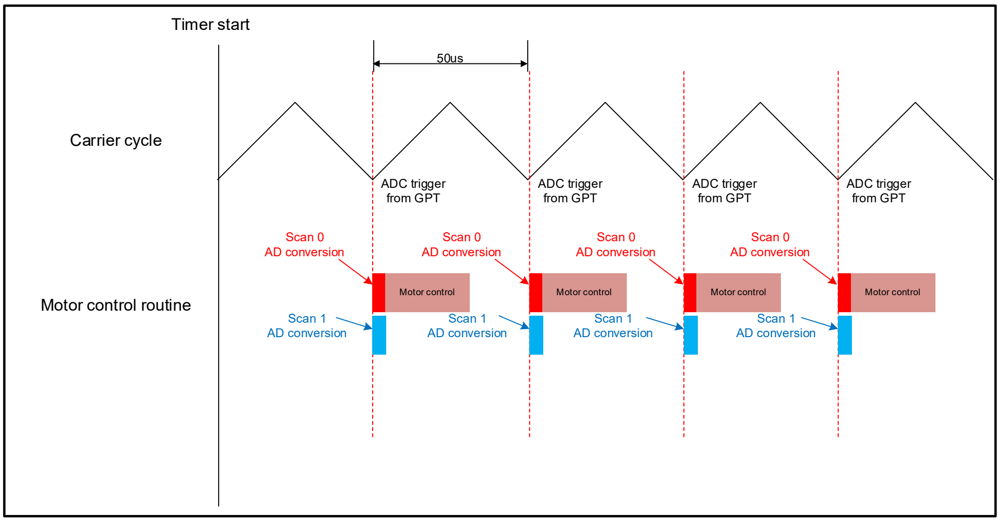


Figure 3-7 AD trigger timing

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]

File name	Function name	Process overview
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)	When a phase current exceeds this value, PWM output ports are set to off.
Limit of over voltage (V)	When an inverter voltage exceeds this value, PWM output ports are set to off.
Limit of over speed (rpm)	When a rotation speed exceeds this value, PWM output ports are set to off.
Limit of low voltage (V)	When an inverter voltage becomes below this value, PWM output ports are set to off.

Table 3-18 Configuration Options Initial Value(rm_motor_120_degree.h)

Options	RA6T2	RA4T1	RA6T3
Limit of over current (A)	1.67	1.67	1.67
Limit of over voltage (V)	60.0	60.0	60.0
Limit of over speed (rpm)	4500.0	4500.0	4500.0
Limit of over speed (rpm)	8.0	8.0	8.0

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 3-19 Configuration Options

Configuration Options (rm_motor_120_control_api.h)	
Option name	**Contents**
Conduction type	Switching of first 60 degree chopping control
Timeout counts (msec)	Stop judgement counter [ms]
Maximum voltage (V)	Maximum command voltage [V]
Minimum voltage (V)	Minimum command voltage [V]
Speed PI decimation	Interrupt thinning number for speed PI control
Free run timer frequency (MHz)	Free run timer frequency [MHz]
Speed LPF K	Speed LPF parameter
Step of speed change	Speed command maximum increase limit
PI control KP	Speed PI proportional gain
PI control KI	Speed PI Integral gain
PI control limit	Voltage PI control output limit value [V]
Motor Parameter | Pole pairs	Number of pole pairs

Table 3-20 Configuration Options Initial Value(rm_motor_120_control_api.h)

Options	RA6T2	RA4T1	RA6T3
Conduction type	First 60 degree PWM	First 60 degree PWM	First 60 degree PWM
Timeout counts (msec)	200	200	200
Maximum voltage (V)	20.0	22.0	22.0
Minimum voltage (V)	3.0	3.0	3.0
Speed PI decimation	0	0	0
Free run timer frequency (MHz)	120	50	50
Speed LPF K	1.0	1.0	1.0
Step of speed reference change	0.2	0.2	0.2
PI control KP	0.02	0.02	0.02
PI control KI	0.0005	0.0005	0.0005
PI control limit	24.0	24.0	24.0
Motor Parameter | Pole pairs	4	4	4

Table 3-21 Configuration Options

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

Table 3-22 Configuration Options Initial Value(rm_motor_120_control_hall.h)

Options	RA6T2	RA4T1	RA6T3
Start reference voltage (V)	5.8	5.8	5.8
Hall wait counts	12	12	12
Stop judge time	1000	1000	1000
Minimum limit speed (rpm)	550	550	550
Hall interrupt mask value	15	15	15

Table 3-23 Configuration Options

Configuration Options (rm_motor_120_driver.h)	
Option name	**Contents**
120 degree control type	120 degree energization control sensor type (Please select HALL)
PWM output port UP	PWM output (Up) port
PWM output port UN	PWM output (Un) port
PWM output port VP	PWM output (Vp) port
PWM output port VN	PWM output (Vn) port
PWM output port WP	PWM output (Wp) port
PWM output port WN	PWM output (Wn) port
PWM timer frequency (MHz)	PWM timer frequency [MHz]
PWM carrier period (Microseconds)	PWM carrier frequency [Micro seconds]
Dead time (Raw counts)	Dead time count [Raw counts]
Current range (A)	Current detection range [A]
Voltage range (V)	Voltage detection range [V]
Resolution of A/D conversion	A/D conversion value
Offset of A/D conversion for current	A/D conversion offset
Conversion level of A/D conversion for voltage	Voltage A/D conversion rate
Counts for current offset measurement	Offset value calculation count
Input voltage	Bus voltage
A/D conversion channel for U phase current	U phase current detection channel
A/D conversion channel for W phase current	W phase current detection channel
A/D conversion channel for main line voltage	Inverter bus voltage detection channel
GTIOCA stop level	Level when the upper arm is stopped
GTIOCB stop level	Level when lower arm is stopped
Modulation | Maximum duty	PWM maximum duty

Table 3-24 Configuration Options Initial Value(rm_motor_120_driver.h)

Options	RA6T2	RA4T1	RA6T3
120 degree control type	Hall	Hall	Hall
PWM output port UP	BSP_IO_PORT_11_PIN_04	BSP_IO_PORT_04_PIN_09	BSP_IO_PORT_04_PIN_09
PWM output port UN	BSP_IO_PORT_11_PIN_05	BSP_IO_PORT_04_PIN_08	BSP_IO_PORT_04_PIN_08
PWM output port VP	BSP_IO_PORT_11_PIN_06	BSP_IO_PORT_01_PIN_03	BSP_IO_PORT_01_PIN_03
PWM output port VN	BSP_IO_PORT_11_PIN_07	BSP_IO_PORT_01_PIN_02	BSP_IO_PORT_01_PIN_02
PWM output port WP	BSP_IO_PORT_11_PIN_08	BSP_IO_PORT_01_PIN_11	BSP_IO_PORT_01_PIN_11
PWM output port WN	BSP_IO_PORT_11_PIN_09	BSP_IO_PORT_01_PIN_12	BSP_IO_PORT_01_PIN_12
PWM timer frequency (MHz)	120	100	100
PWM carrier period (Microseconds)	50	50	50
Dead time (Raw counts)	240	200	200
Current range (A)	16.5	16.5	16.5
Voltage range (V)	73.26	73.26	73.26
Resolution of A/D conversion	0xFFF	0xFFF	0xFFF
Offset of A/D conversion for current	0x7FF	0x7FF	0x7FF
Conversion level of A/D conversion for voltage	1.0	1.0	1.0
Counts for current offset measurement	500	500	500
Input voltage	24.0	24.0	24.0
A/D conversion channel for U phase current	ADC_CHANNEL_4	ADC_CHANNEL_0	ADC_CHANNEL_0
A/D conversion channel for W phase current	ADC_CHANNEL_0	ADC_CHANNEL_2	ADC_CHANNEL_2
A/D conversion channel for main line voltage	ADC_CHANNEL_6	ADC_CHANNEL_4	ADC_CHANNEL_4
GTIOCA stop level	Pin Level Low	Pin Level Low	Pin Level Low
GTIOCB stop level	Pin Level High	Pin Level High	Pin Level High
Modulation | Maximum duty	0.9375	0.9375	0.9375

3.4 Control flows (flow charts)

3.4.1 Main process

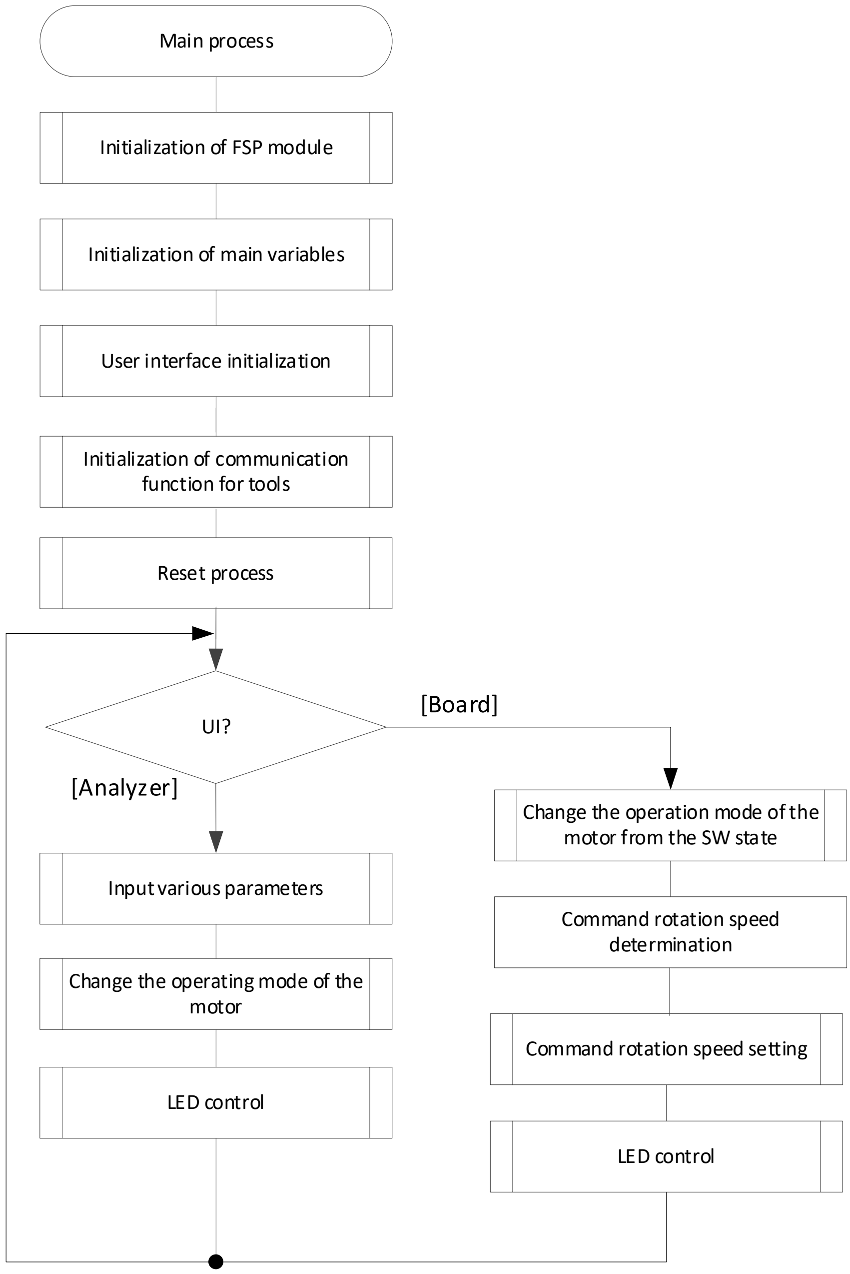


Figure 3-8 Main process Flowchart

3.4.2 Current Control Periodic interrupt (carrier periodic interrupt) handling

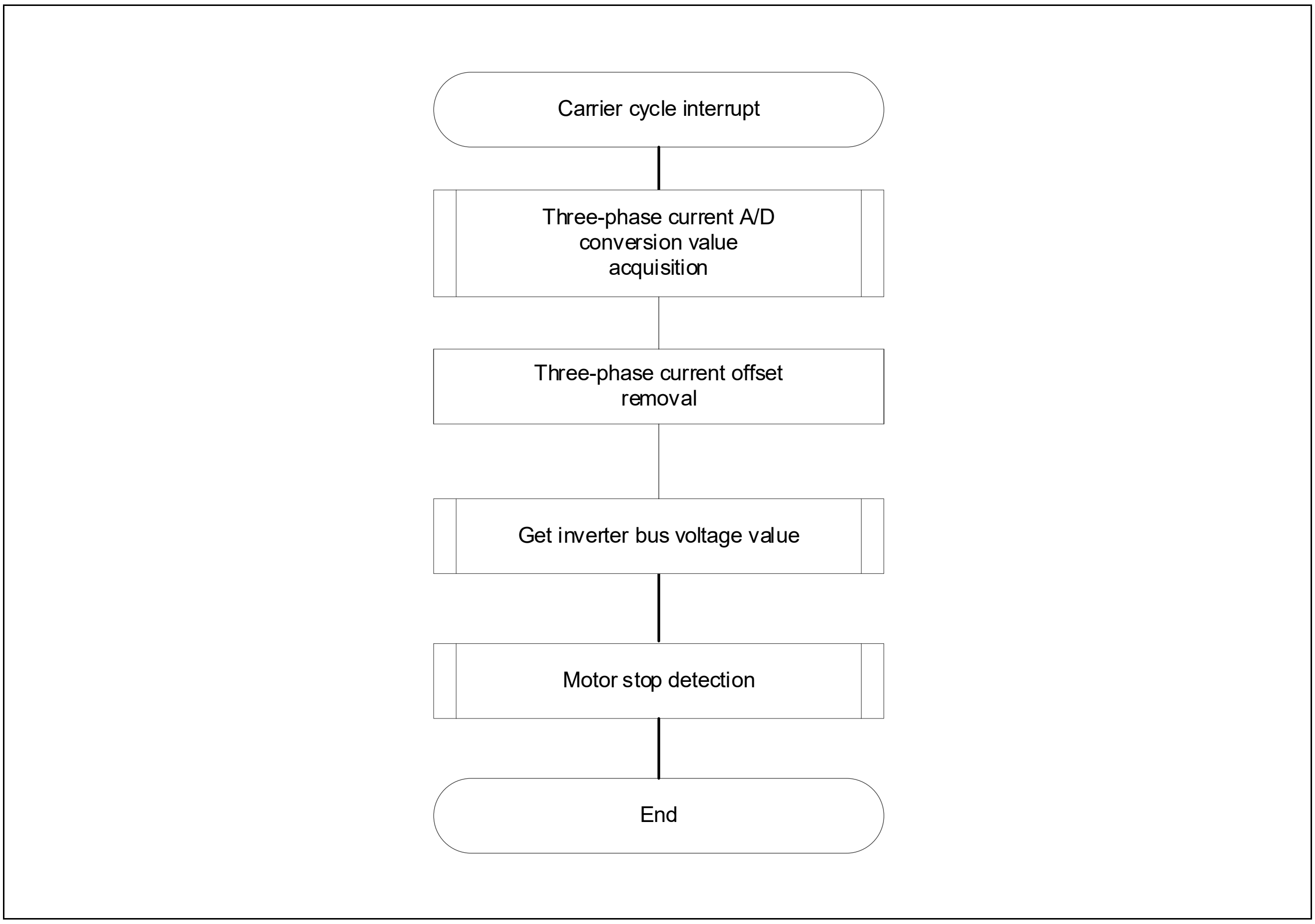


Figure 3-9 Current Control Periodic Interrupt Handling

3.4.3 Speed Control periodic interrupt handling

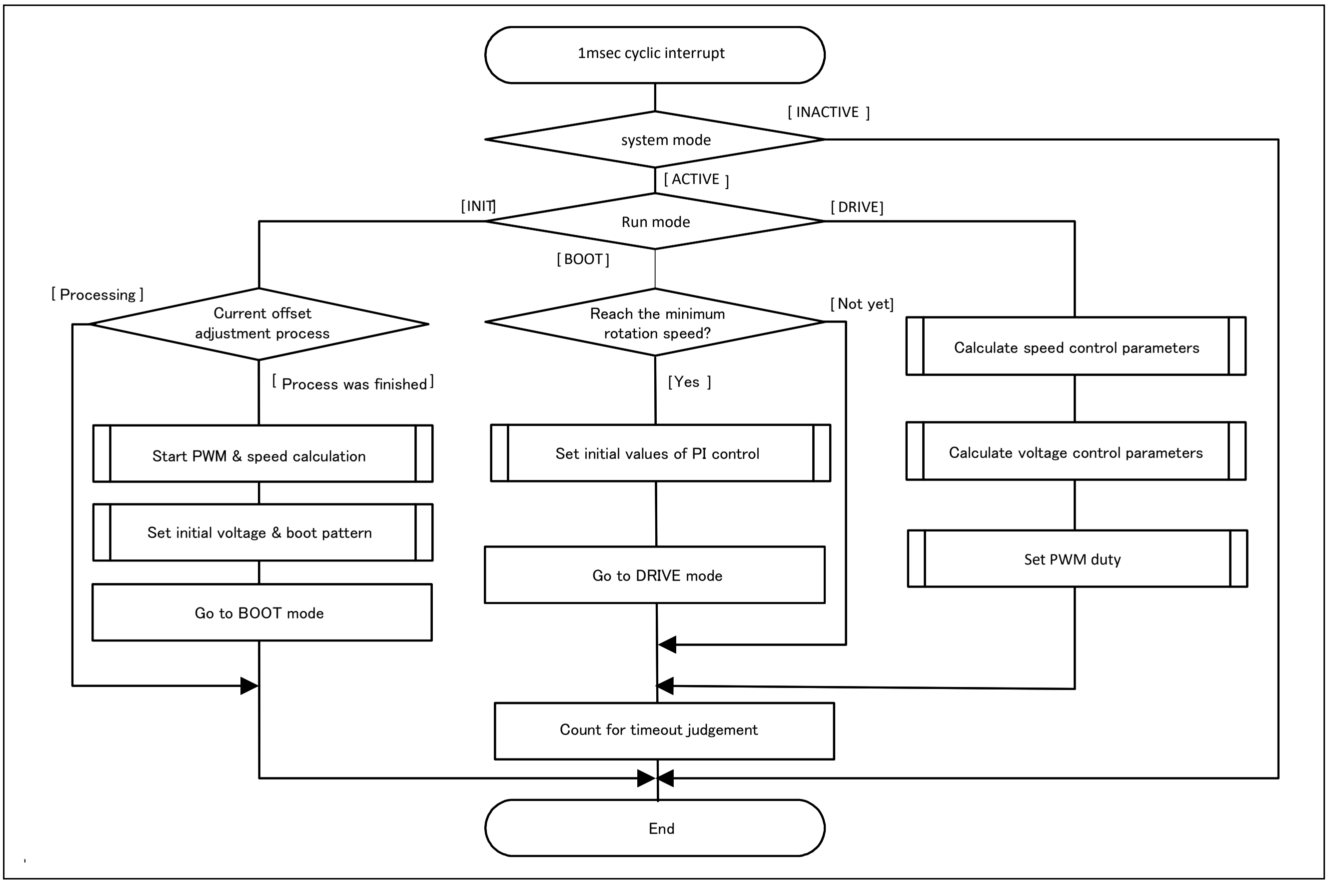


Figure 3-10 Speed Control Periodic Interrupt Handling

3.4.4 Overcurrent interrupt handling

The overcurrent detection interrupt is an interrupt that occurs when an external overcurrent detection signal is input. 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.

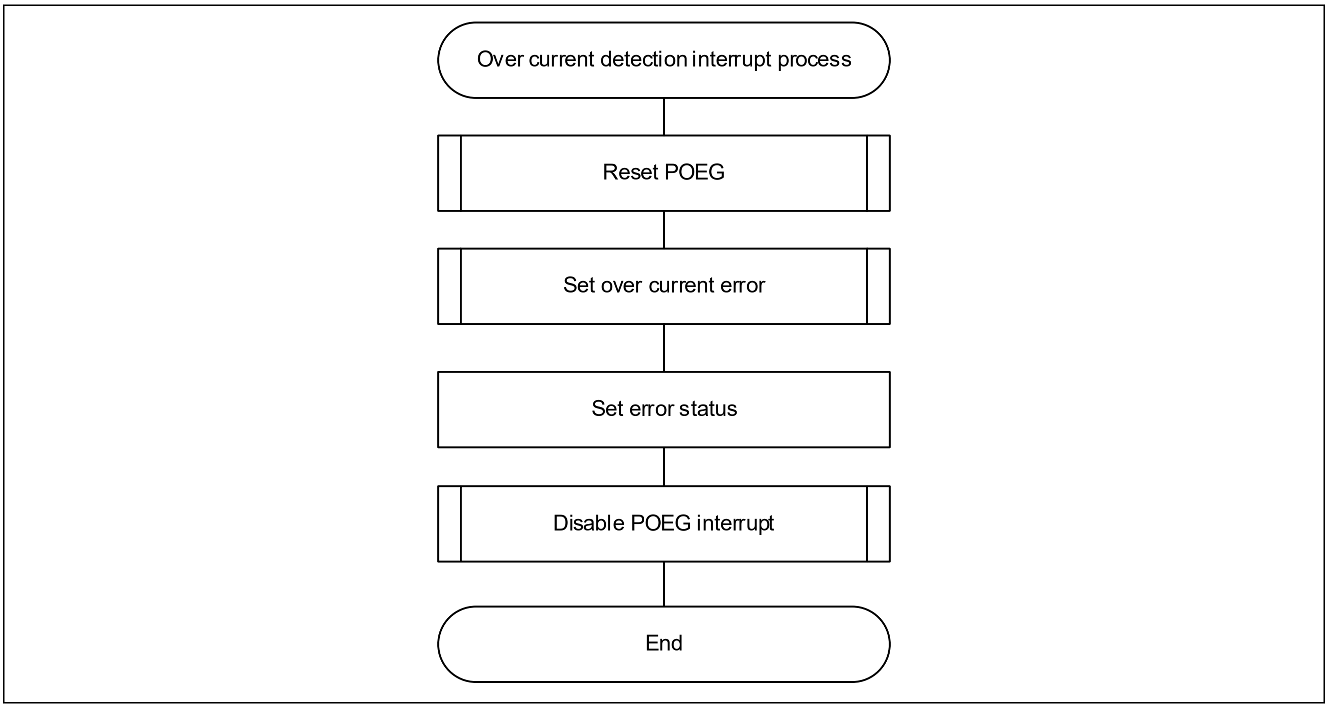


Figure 3-11 Overcurrent Detection Interrupt Handling

4. Evaluation environment explanation

This application note describes the target software.

4.1 Project import

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

1. File → Import

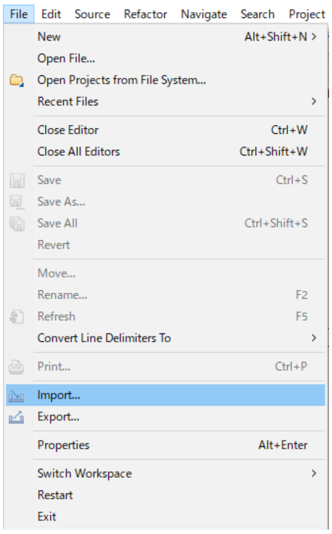


Figure 4-1 File Menu

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

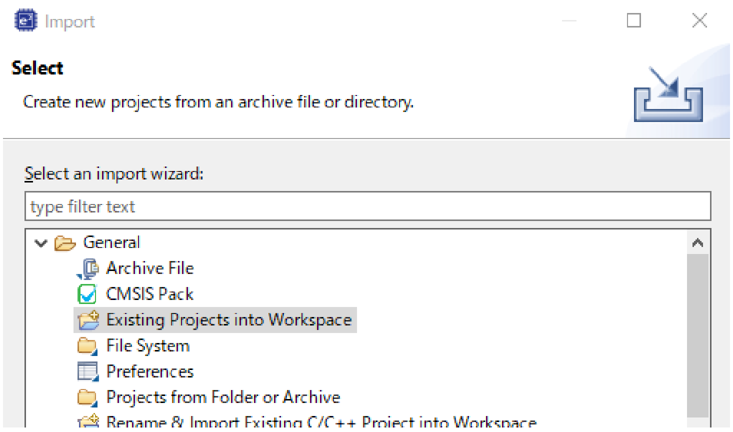


Figure 4-2 Import Menu

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

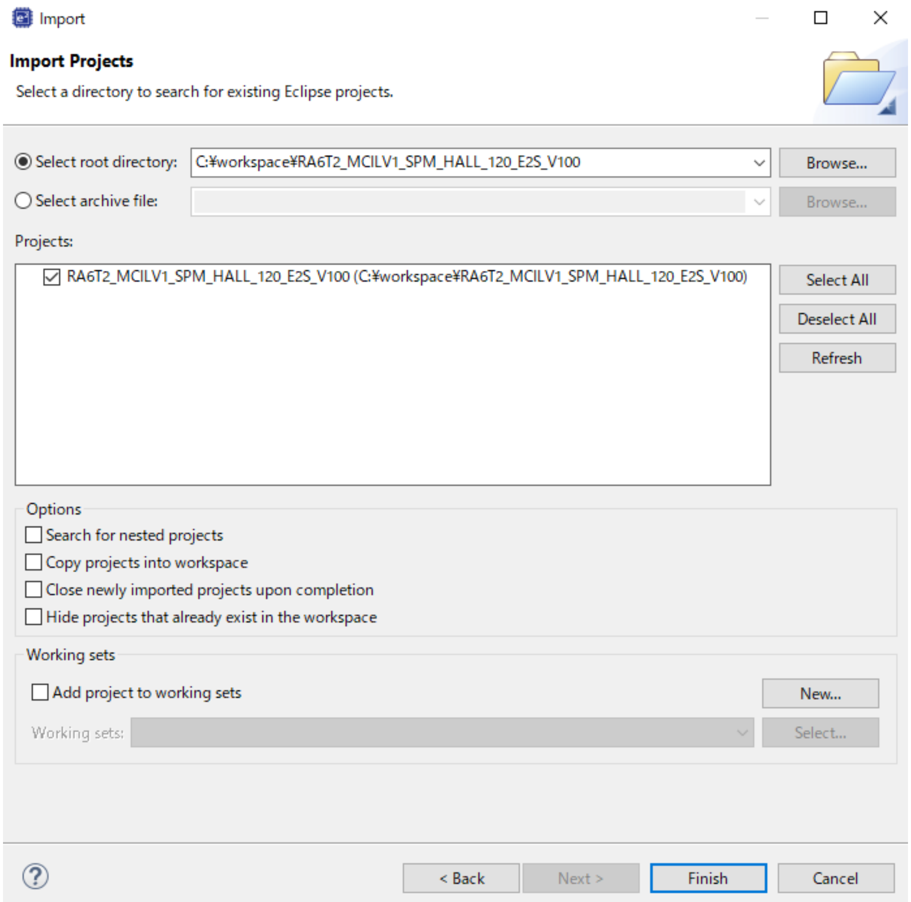


Figure 4-3 Project Import

4.2 Build and debug

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

4.3 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.

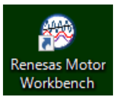
4.4 Motor Control Development Support Tool, ‘Renesas Motor Workbench’

4.4.1 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

<u>Set up for Renesas Motor Workbench</u>

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.

4.4.2 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".

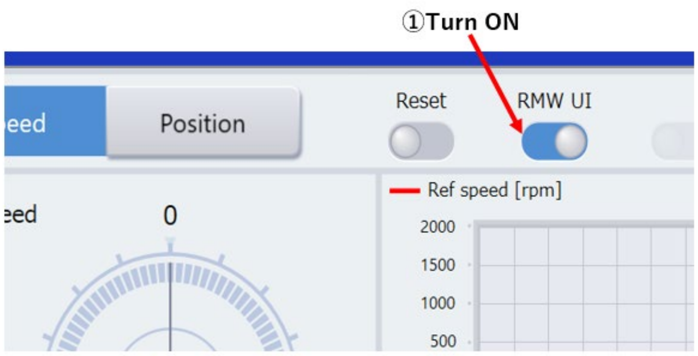


Figure 4-1 Procedure for changing to use Renesas Motor Workbench

- Change the user interface to use Renesas Motor Workbench
  (1) Press the “Run” button

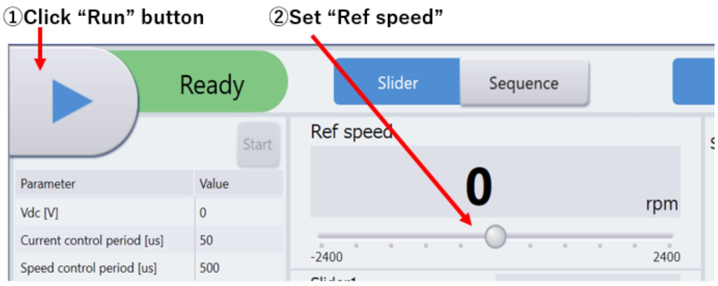


Figure 4-2 Motor rotation procedure

- Stop the motor
  (1) Press the “Stop” button

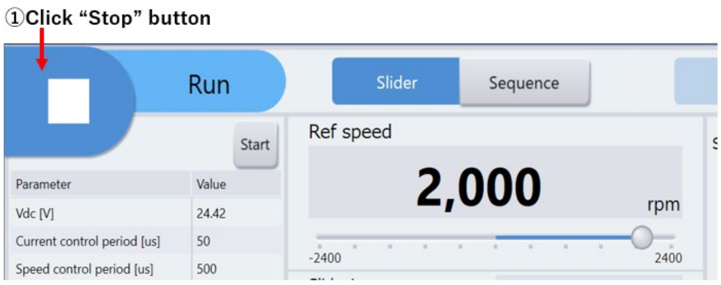


Figure 4-3 Motor rotation procedure

- Processing when it stops (error)

  (1) Turn on "Reset" button.

  (2) Turn off “Reset” button

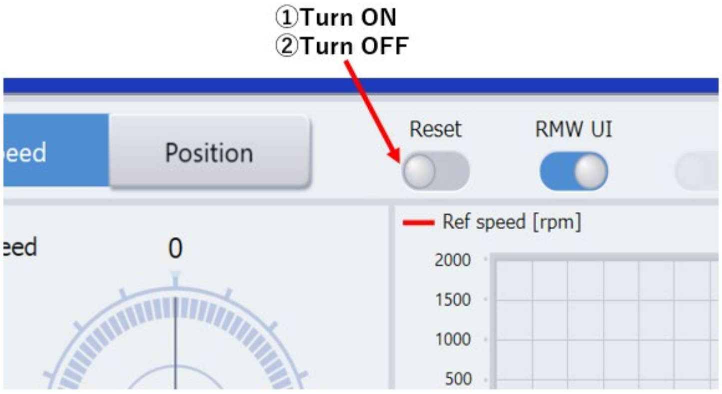


Figure 4-4 Error clearing procedure

4.4.3 List of variables for Analyzer function

Table 4-1 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 4-1 List of variables for analyzer function input

Variable name	Type	Contents
com_u1_mode_system(*)	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)

4.4.4 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.

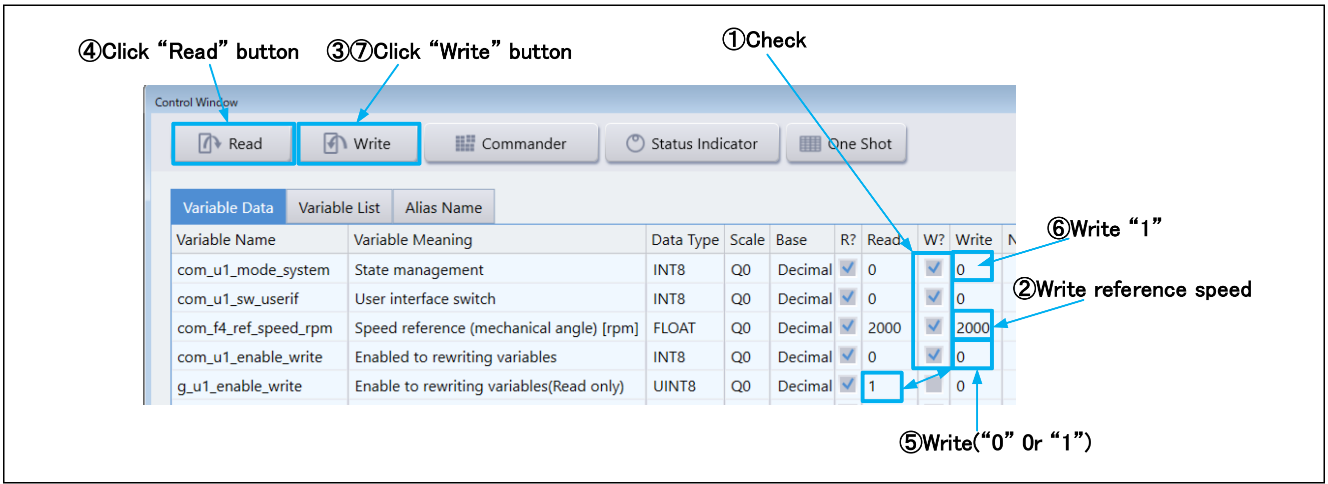


Figure 4-5 Procedure – Driving the motor

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

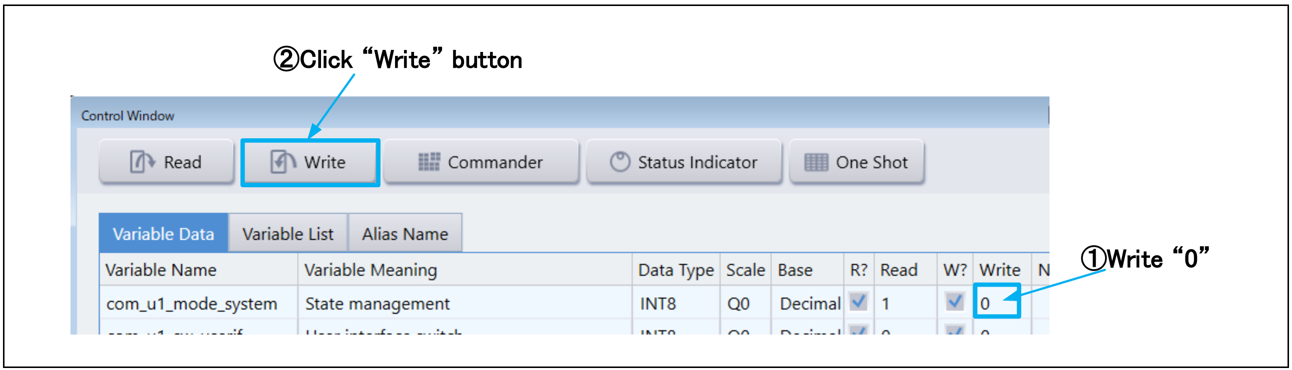


Figure 4-6 Procedure – Stop the motor

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

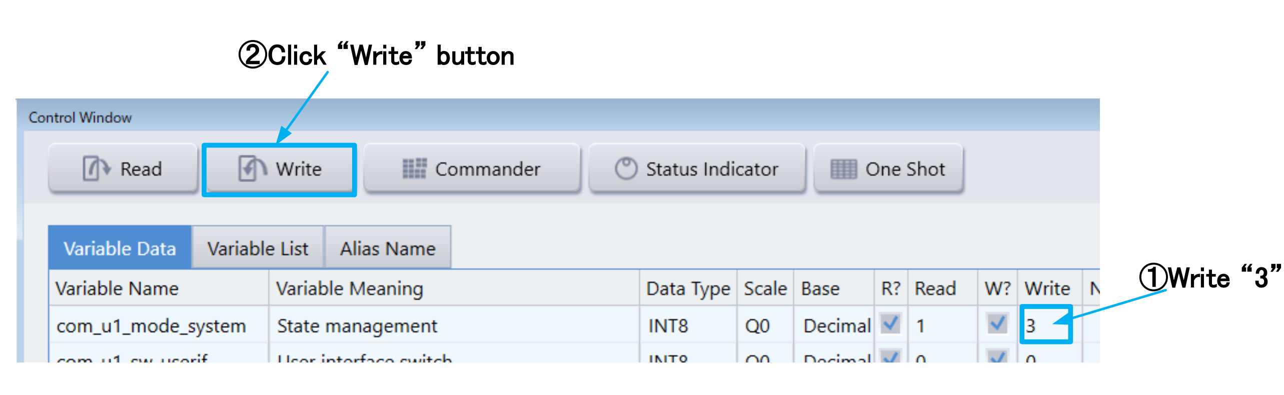


Figure 4-7 Procedure – Error cancel operation

4.4.5 Example of changing communication speed

The procedure for changing the communication speed of Renesas Motor Workbench with the sample software is shown below. See the Renesas Motor Workbench User's Manual for the values to change.

- Change the communication speed setting of the sample software (when the required communication rate is 10 Mbps)
  (1) Change the value of ICS_BRR in r_mtr_ics.h to 1.
  (2) Change the value of MTR_ICS_DECIMATION in r_mtr_ics.h to 1.

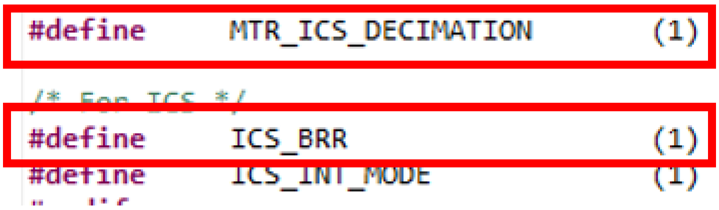


Figure 4-5 Modification of r_mtr_ics.h

- Change the communication speed setting of Renesas Motor Workbench to connect
  (1) Press the Clock button on the Main Window to change the value to 80,000,000
  This value was calculated by multiplying the default 8,000,000 by 10
  because the UART communication baud rate was changed from 1Mbps to 10Mbps.
  (2) Select the COM of the connected kit in the COM of Connection

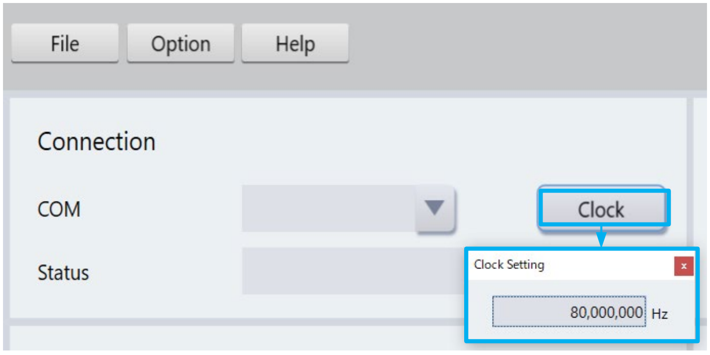


Figure 4-6 Clock frequency setting

If the connection fails, repeat the procedure for reconnecting after resetting the communication board.

4.4.6 How to use the built-in communication library

The procedure for connecting to Renesas Motor Workbench using the built-in communication library without using the communication board with the sample software is shown below.

- Connection between PC and CPU board

(1) Connect the CPU board and PC via a USB / serial conversion board, etc.

- Preparing a project for built-in communication (example of RA6T2 921600bps)

(1) Cancel the registration of ICS2_RA6T2.o

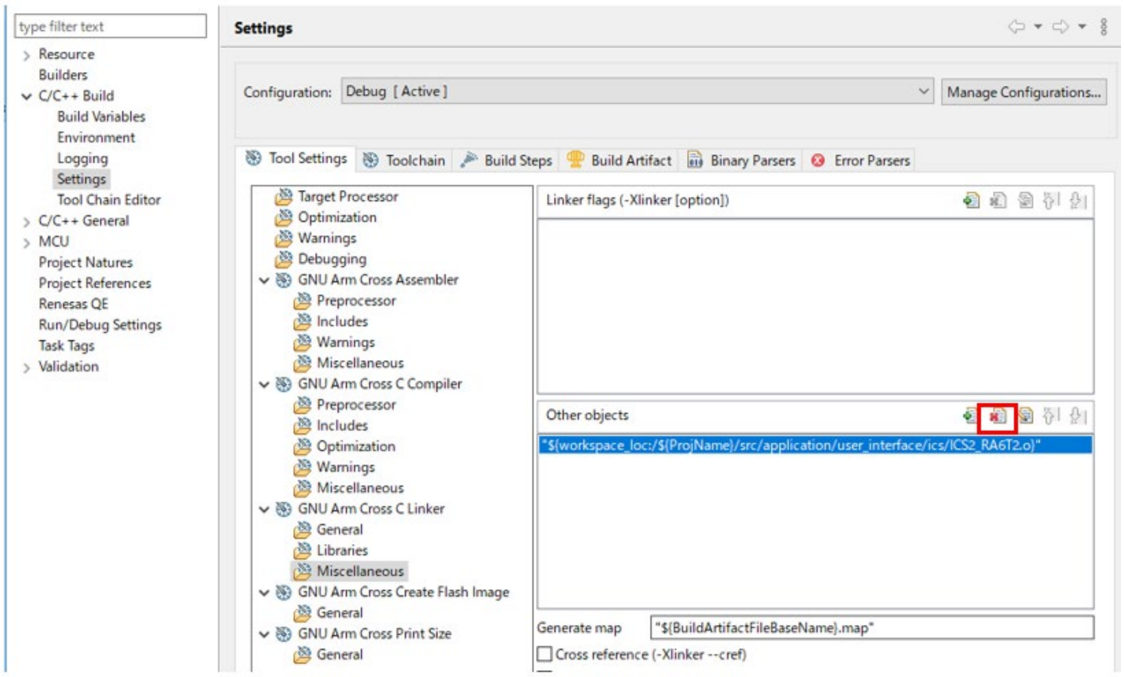


Figure 4-7 Unregister ICS2_RA6T2.o

(2) Register ICS2_RA6T2_Built_in.o

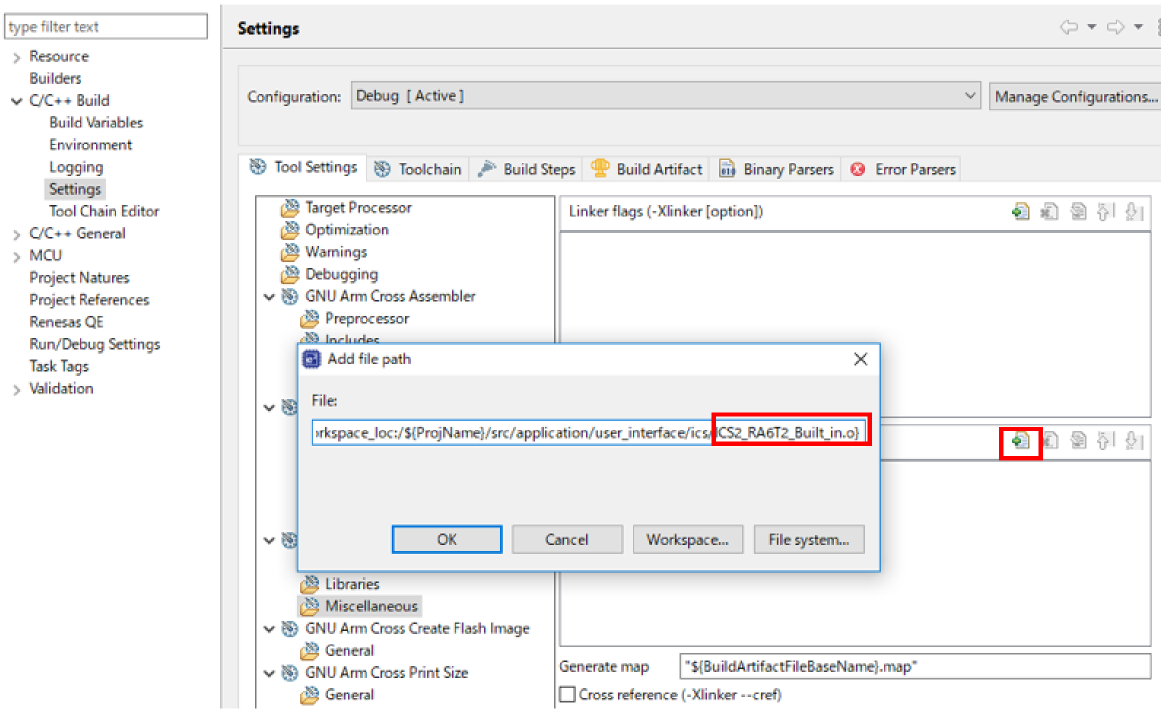


Figure 4-8 Register ICS2_RA6T2.o

(3) Change the value of USE_BUILT_IN in r_mtr_ics.h to 1.

```
#define     USE_BUILT_IN                (1)

#if USE_BUILT_IN
#define     MTR_ICS_DECIMATION          (1)

/* For ICS */
#define     ICS_BRR                     (21)
#define     ICS_INT_MODE                (1)
```

Figure 4-9 Modification of r_mtr_ics.h

- Change the communication baud rate setting of Renesas Motor Workbench to connect
  (1) Change the value to 921,600 with Baud rate Dialog from the Option menu of the Main Window.
  (2) Select the COM port of the connected kit in the COM of Connection.

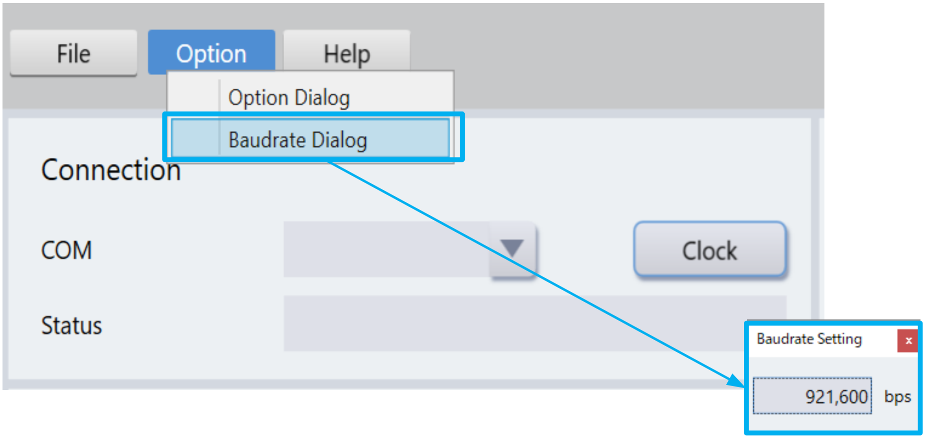


Figure 4-10 Baud rate setting

5. Reference document

RA6T2 Group User’s Manual: Hardware (R01UH0951)
RA4T1 Group User’s Manual: Hardware (R01UH0998)
RA6T3 Group User’s Manual: Hardware (R01UH0999)
RA Flexible Software Package Documentation
120-degree conducting control of permanent magnetic synchronous motor (Algorithm) (R01AN2657)
Renesas Motor Workbench User’s Manual (R21UZ0004)
Renesas Motor Workbench Quick start guide (R21QS0011)
MCK-RA6T2 User's Manual (R12UZ0091)
MCK-RA4T1 User's Manual (R12UZ0114)
MCK-RA6T3 User's Manual (R12UZ0115)

Revision History

Rev.	Data of issue	Descriptions	
		Page	Summary
1.00	May 23, 2023	—	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

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/.