Servo control sample program

RA Family

Abstract

This application note describes the sample program about inertia estimation and return origin function which are added to vector control using encoder or induction sensor based on Renesas microcontrollers. About the vector control for permanent magnet synchronous motor with encoder or induction sensor, please refer to each application note “RA01AN5923” ,“RA01AN6208” and “R01AN6467”.

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

Operation checking device

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

- RA6T1 (R7FA6T1AD3CFP)

- RA6T2 (R7FA6T2BD3CFP)

Target software

The following shows the target software for this application note:

- RA6T1_ESB_SPM_ENCD_FOC_SERVO_E2S_V102
- RA6T2_MCILV1_SPM_ENCD_FOC_SERVO_E2S_V111
- RA6T2_MCILV1_SPM_IS_FOC_SERVO_E2S_V111

Contents

1. Overview

This application note explains about inertia estimate and return origin function which are added to the vector control with encoder or induction sensor software that drives permanent magnetic synchronous motor (PMSM) using the RA6T1/RA6T2 microcontroller.

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

1.1 Development environment

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

Table 1-1 Hardware Development Environment

Classification	Product used
Microcomputer / CPU card P/N	RA6T1 (R7FA6T1AD3CFP) / RTK0EMA170C00000BJ RA6T2 (R7FA6T2BD3CFP) / RTK0EMA270C00000BJ
Inverter board	RA6T1 : RTK0EM0000B10020BJ RA6T2 : MCI-LV-1 / RTK0EM0000S04020BJ
Motor	BLY171D-24V-4000
Sensor	Encoder : AMT102-V (CUI Devices)

Table 1-2 Software Development Environment

e^{2}studio version	FSP version	Toolchain version
e^{2}studio：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. System overview

Overview of this system is explained below.

2.1 Hardware configuration

The hardware configuration is shown below.

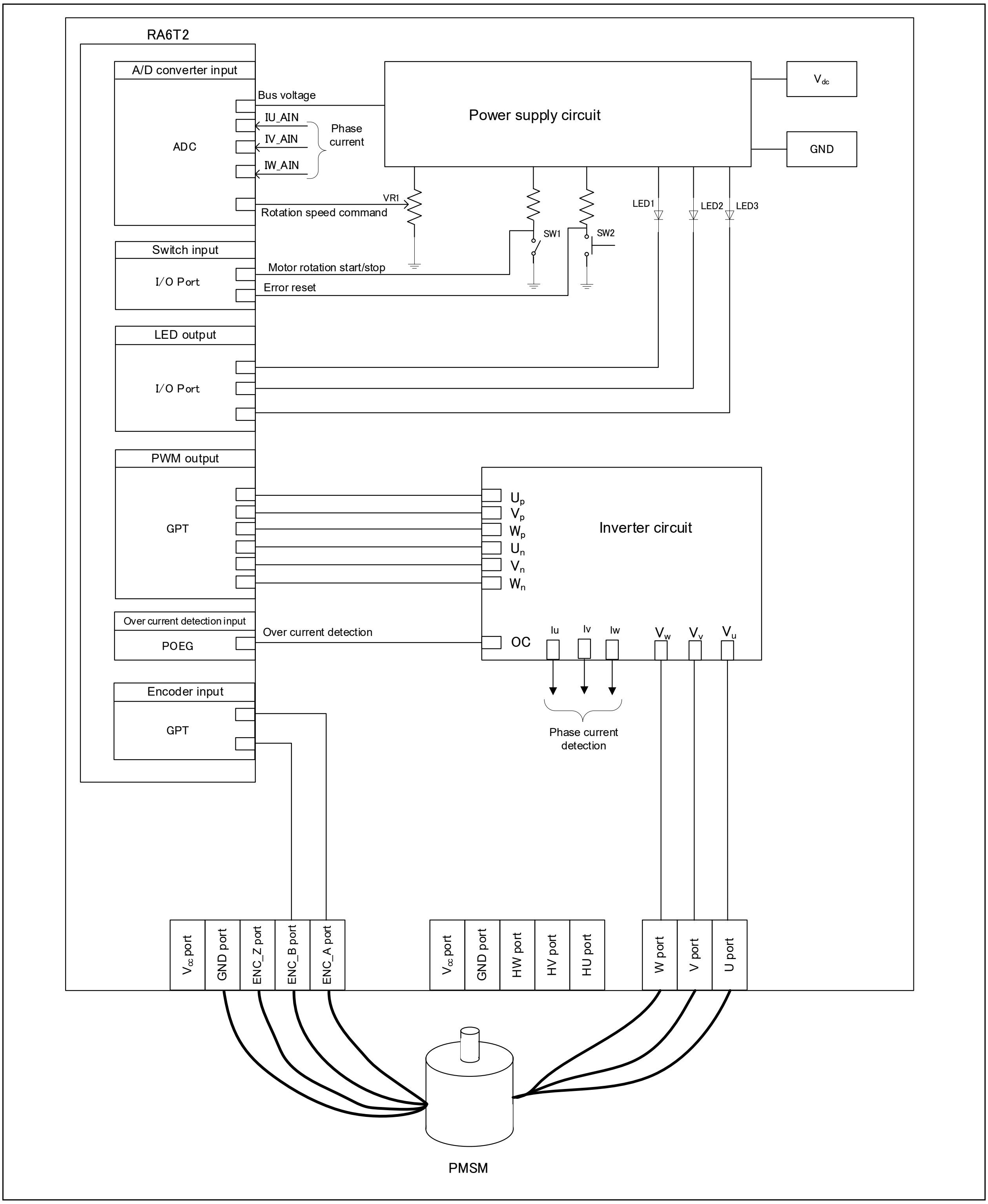


Figure 2-1 Hardware Configuration Diagram : Encoder

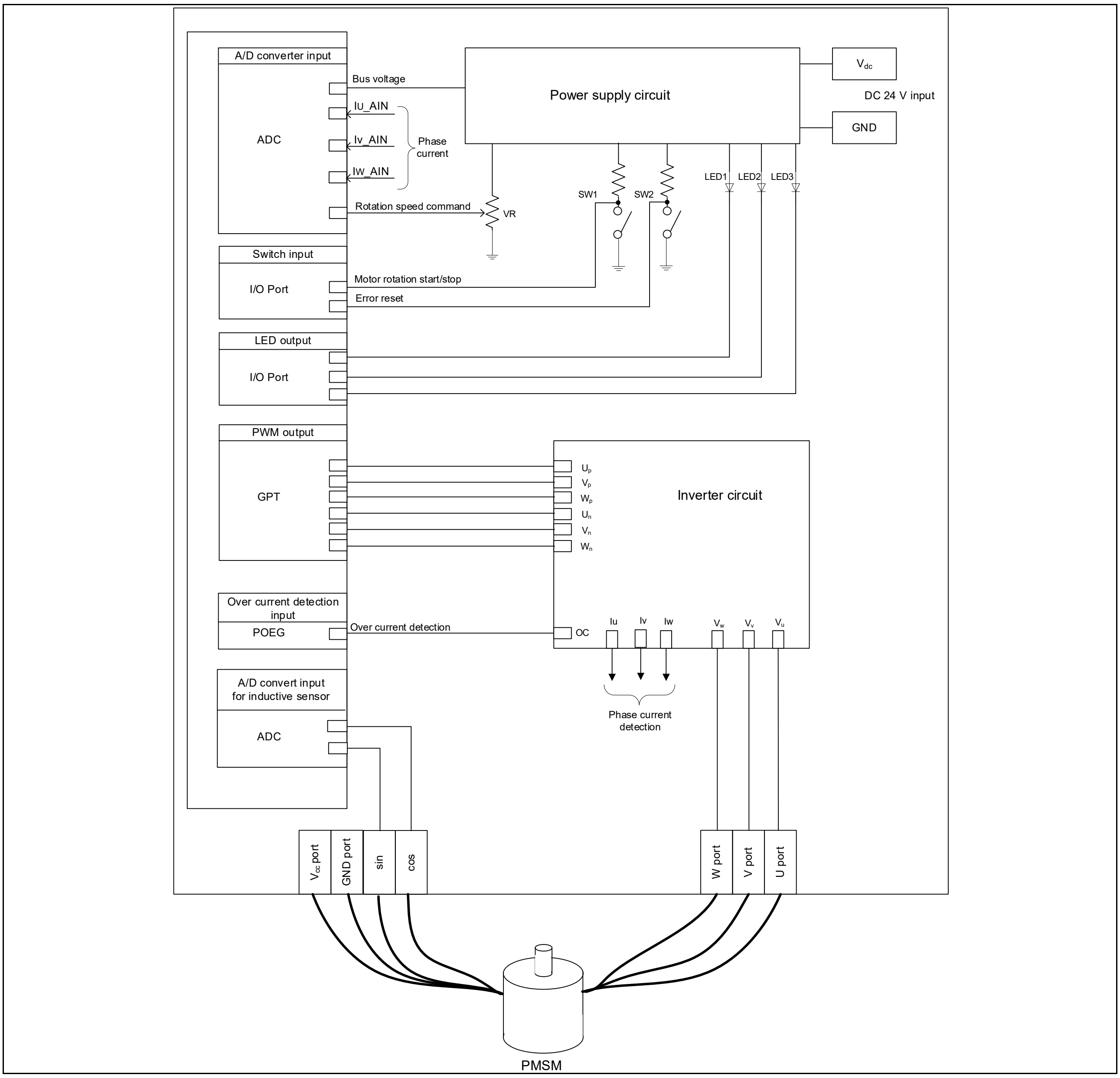


Figure 2-2 Hardware Configuration Diagram : Induction sensor

2.2 Hardware specifications

2.2.1 User interface

The lists of user interfaces of this system are shown below.

Table 2-1 User Interface

Item	Interface component	Function
Rotation speed command	Variable resistor (VR)	Reference value of rotation speed input (analog value)
START/STOP	Toggle switch (SW1)	Motor rotation start/stop command
ERROR RESET	Push switch (SW2)	Command of recovery from error status
LED1	Orange LED	- At the time of motor rotation: ON - At the time of stop: OFF
LED2	Orange LED	- At the time of error detection: ON - At the time of normal operation: OFF
LED3	Orange LED	Not use in this system.

Table 2-2 CPU card user interface

Item	Interface component	Function
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 use in this system.
RESET	Push switch (RESET)	System reset

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

Table 2-3 Port Interfaces

Function	RA6T2	RA6T1
Inverter bus voltage measurement	PA06 / AN006	P014 / AN005
For rotation speed command value input (analog value)	PB00 / AN008	P502 / AN017
START/STOP toggle switch (SW1)	PD04	P302
ERROR RESET push switch (SW2)	PD07	P301
LED1 ON/OFF control	PD01	P408
LED2 ON/OFF control	PD02	P407
U phase current measurement	PA04 / AN004	P000 / AN000
V phase current measurement	PA02 / AN002	P001 / AN001
W phase current measurement	PA00 / AN000	P002 / AN002
PWM output (U_p)	PB04 / GTIOC4A	P415 / GTIOC0A
PWM output (V_p)	PB06 / GTIOC5A	P113 / GTIOC2A
PWM output (W_p)	PB08 / GTIOC6A	P111 / GTIOC3A
PWM output (U_n)	PB05 / GTIOC4B	P414 / GTIOC0B
PWM output (V_n)	PB07 / GTIOC5B	P114 / GTIOC2B
PWM output (W_n)	PB09 / GTIOC6B	P112 / GTIOC3B
Encoder A phase input	PC14 / GTIOC3A	P107 / GTIOC8A
Encoder B phase input	PC15 / GTIOC3B	P106 / GTIOC8B
Cos input for induction sensor	PE11 / AN023	Not supported by RA6T1
Sin input for induction sensor	PE12 / AN024	Not supported by RA6T1
PWM emergency stop input at the time of overcurrent detection	PC13 / GTETRGD	P503 / GTETRGC

2.3 Stack structure of FSP

2.3.1 Stack structure of FSP

Stack structure of FSP for each project are shown below.

The module “Inertia estimate” and “Return origin” which are red squared are new added modules.

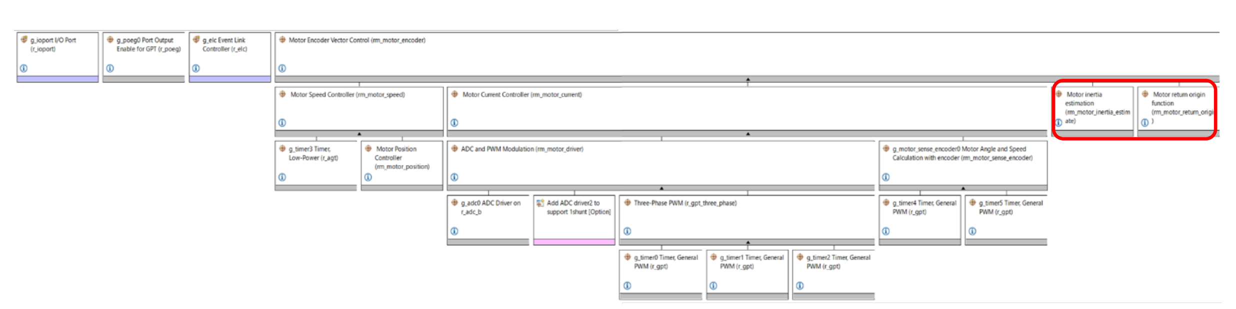

g_ioport I/O Port (r_ioport)
g_poeg0 Port Output Enable for GPT (r_poeg)
g_elc Event Link Controller (r_elc)
Motor Encoder Vector Control (rm_motor_encoder)
Motor Speed Controller (rm_motor_speed)
Motor Current Controller (rm_motor_current)
Motor inertia estimation (rm_motor_inertia_estimate)
Motor return origin function (rm_motor_return_origi)
g_timer3 Timer, Low-Power (r_agt)
Motor Position Controller (rm_motor_position)
ADC and PWM Modulation (rm_motor_driver)
g_motor_sense_encoder0 Motor Angle and Speed Calculation with encoder (rm_motor_sense_encoder)
g_adc0 ADC Driver on r_adc_b
Add ADC driver2 to support 1shunt [Option]
Three-Phase PWM (r_gpt_three_phase)
g_timer4 Timer, General PWM (r_gpt)
g_timer5 Timer, General PWM (r_gpt)
g_timer0 Timer, General PWM (r_gpt)
g_timer1 Timer, General PWM (r_gpt)
g_timer2 Timer, General PWM (r_gpt)

Figure 2-3 Stack structure of FSP for RA6T2 using encoder

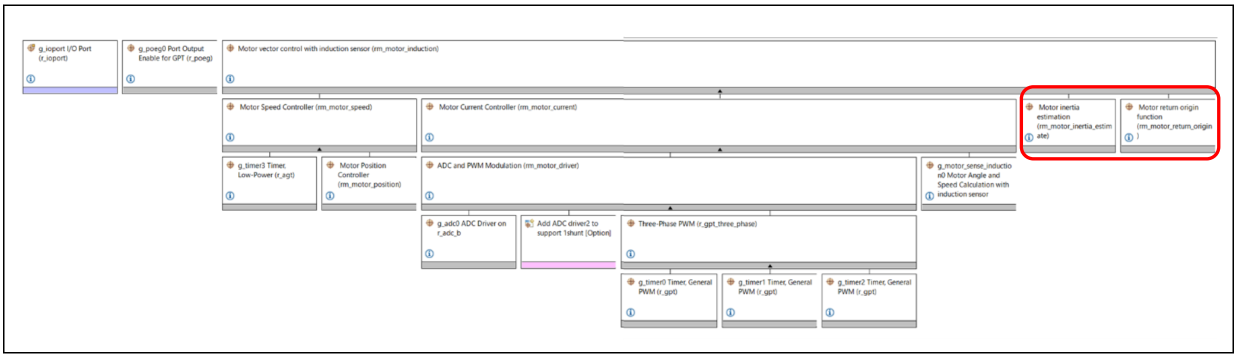

g_ioport I/O Port (r_ioport)
g_poeg0 Port Output Enable for GPT (r_poeg)
Motor vector control with induction sensor (rm_motor_induction)
Motor Speed Controller (rm_motor_speed)
Motor Current Controller (rm_motor_current)
Motor inertia estimation (rm_motor_inertia_estimate)
Motor return origin function (rm_motor_return_origin)
g_timer3 Timer, Low-Power (r_agt)
Motor Position Controller (rm_motor_position)
ADC and PWM Modulation (rm_motor_driver)
g_motor_sense_induction0 Motor Angle and Speed Calculation with induction sensor
g_adc0 ADC Driver on r_adc_b
Add ADC driver2 to support 1shunt [Option]
Three-Phase PWM (r_gpt_three_phase)
g_timer0 Timer, General PWM (r_gpt)
g_timer1 Timer, General PWM (r_gpt)
g_timer2 Timer, General PWM (r_gpt)

Figure 2-4 Stack structure of FSP for RA6T2 using induction sensor

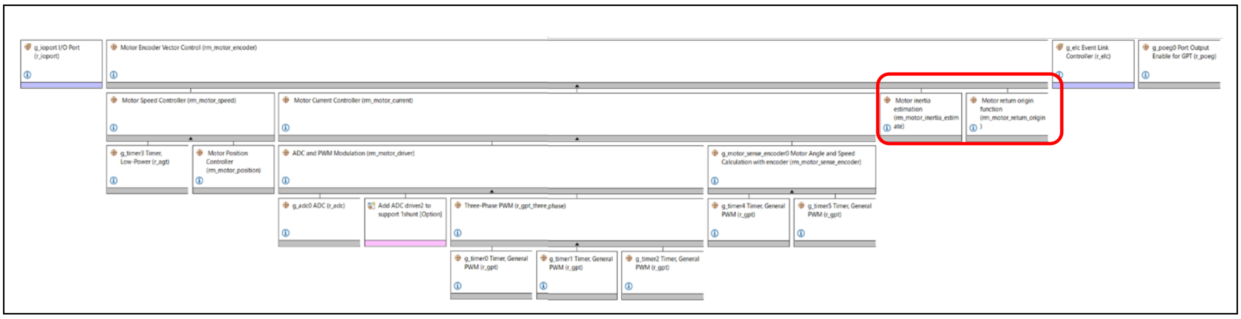
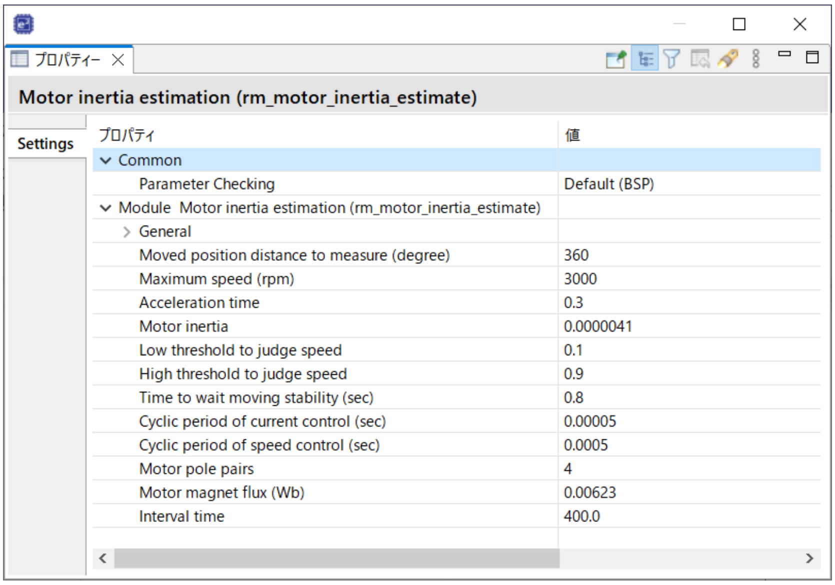

g_ioport I/O Port (r_ioport)
Motor Encoder Vector Control (rm_motor_encoder)
g_elc Event Link Controller (r_elc)
g_poeg0 Port Output Enable for GPT (r_poeg)
Motor Speed Controller (rm_motor_speed)
Motor Current Controller (rm_motor_current)
Motor inertia estimation (rm_motor_inertia_estimate)
Motor return origin function (rm_motor_return_origin)
g_timer3 Timer, Low-Power (r_agt)
Motor Position Controller (rm_motor_position)
ADC and PWM Modulation (rm_motor_driver)
g_motor_sense_encoder0 Motor Angle and Speed Calculation with encoder (rm_motor_sense_encoder)
g_adc0 ADC (r_adc)
Add ADC driver2 to support 1shunt [Option]
Three-Phase PWM (r_gpt_three_phase)
g_timer4 Timer, General PWM (r_gpt)
g_timer5 Timer, General PWM (r_gpt)
g_timer0 Timer, General PWM (r_gpt)
g_timer1 Timer, General PWM (r_gpt)
g_timer2 Timer, General PWM (r_gpt)

Figure 2-5 Stack structure of FSP for RA6T1 using encoder

2.3.2 FSP configuration

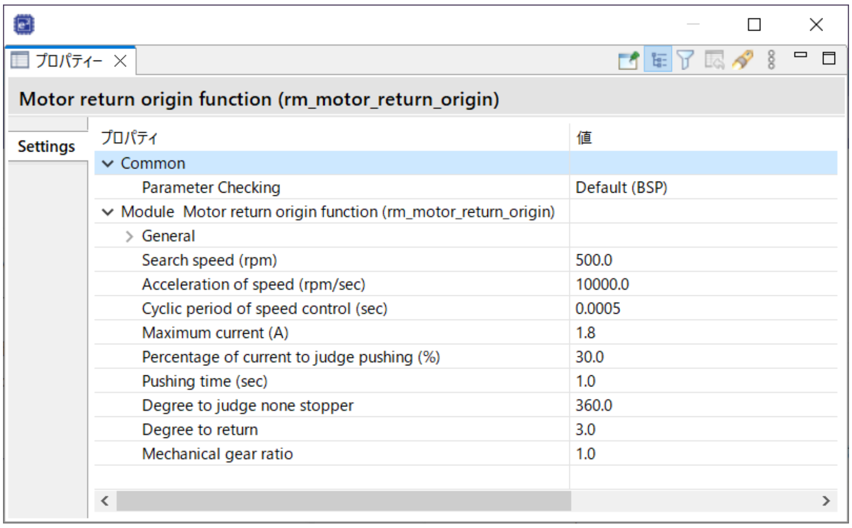
FSP configuration for inertia estimate and return origin module are shown below.



Motor inertia estimation (rm_motor_inertia_estimate)

プロパティ	値
Common	
Parameter Checking	Default (BSP)
Module Motor inertia estimation (rm_motor_inertia_estimate)	
General	
Moved position distance to measure (degree)	360
Maximum speed (rpm)	3000
Acceleration time	0.3
Motor inertia	0.0000041
Low threshold to judge speed	0.1
High threshold to judge speed	0.9
Time to wait moving stability (sec)	0.8
Cyclic period of current control (sec)	0.00005
Cyclic period of speed control (sec)	0.0005
Motor pole pairs	4
Motor magnet flux (Wb)	0.00623
Interval time	400.0

Figure 2-6 FSP Configuration of inertia estimation (FSP 4.3.0)



Motor return origin function (rm_motor_return_origin)

プロパティ	値
Common	
Parameter Checking	Default (BSP)
Module Motor return origin function (rm_motor_return_origin)	
General	
Search speed (rpm)	500.0
Acceleration of speed (rpm/sec)	10000.0
Cyclic period of speed control (sec)	0.0005
Maximum current (A)	1.8
Percentage of current to judge pushing (%)	30.0
Pushing time (sec)	1.0
Degree to judge none stopper	360.0
Degree to return	3.0
Mechanical gear ratio	1.0

Figure 2-7 FSP Configuration of return origin (FSP 4.3.0)

2.4 Folder and file structure

2.4.1 Folder and file structure

Folder and file structure of the software is shown below.

Table 2-4 Folder and file structure [1/2]

Folder	Subfolder	File	Contents
ra_cfg			Configuration header by auto setting
ra_gen			Register setting auto generated
ra	arm		CMSIS source code
	board		Definitions about circuit board
	fsp/inc/api	bsp_api.h	Definitions of BSP API
		r_adc_api.h	Definitions of AD API
		r_elc_api.h	Definitions of ELC API
		r_ioport_api.h	Definitions of I/O ports API
		r_poeg_api.h	Definitions of POEG API
		r_three_phase_api.h	Definitions of three phase PWM API
		r_timer_api.h	Definitions of timer API
		r_transfer_api.h	Definitions of transfer API
		rm_motor_angle_api.h	Definitions of angle detection API
		rm_motor_api.h	Definitions of motor application API
		rm_motor_current_api.h	Definitions of current control API
		rm_motor_driver_api.h	Definitions of motor driver API
		rm_motor_inertia_estimate_api.h	Definitions of inertia estimate API
		rm_motor_position_api.h	Definitions of position control API
		rm_motor_return_origin_api.h	Definitions of return origin API
		rm_motor_speed_api.h	Definitions of speed control AP

	fsp/inc/instances	r_adc_b.h(RA6T2)
		r_adc.h(RA6T1)
			Definitions about A/D conversion

		r_agt.h	Definitions about AGT
		r_elc.h	Definitions about ELC
		r_gpt_three_phase.h	Definitions about three phase PWM
		r_gpt.h	Definitions about GPT
		r_ioport.h	Definitions about I/O ports
		r_poeg.h	Definitions about POEG
		rm_motor_current.h	Definitions about current control
		rm_motor_driver.h	Definitions about motor driver
		rm_motor_encoder.h	Definitions about motor application with using encoder
		rm_motor_induction.h	Definitions about motor application with using induction sensor
		rm_motor_position_api.h	Definitions about position control
		rm_motor_sense_encoder.h	Definitions about process for encoder input
		rm_motor_sense_induction.h	Definitions about process for induction sensor input
		rm_motor_speed.h	Definitions about speed control
	fsp/lib		Libraries

Table 2-5 Folder and file structure [2/2]

Folder	Subfolder	File	Contents
ra	fsp/src	bsp	Files about BSP
		r_adc_b/r_adc_b.c (RA6T2) r_adc/r_adc.c (RA6T1)	Driver of AD converter
		r_agt/r_agt.c	Driver of AGT
		r_elc/r_elc.c	Driver of ELC
		r_gpt/r_gpt.c	Driver of GPT
		r_gpt_three_phase/ r_gpt_three_phase.c	Driver of three phase PWM
		r_ioport/r_ioport.c	Driver of IO ports
		r_poeg/r_poeg.c	Driver of POEG
		rm_motor_current/rm_motor_current.c	Driver of current control
		rm_motor_current/rm_motor_current_library.h	Definitions of current control library functions
		rm_motor_driver/rm_motor_driver.c	Driver of motor hardware
		rm_motor_encoder/rm_motor_encoder.c	Application code of motor control with encoder
		rm_motor_induction/rm_motor_induction.c	Application code of motor control with induction sensor
		rm_motor_position/rm_motor_position.c	Driver of position control
		rm_motor_position/rm_motor_position_library.h	Definitions of position control library functions
		Rm_motor_sense_encoder /rm_motor_sense_encoder.c	Handler of encoder input (only Encoder use)
		Rm_motor_sense_induction /rm_motor_sense_induction.c	Handler of induction sensor input (only Induction sensor use)
		rm_motor_speed/rm_motor_speed.c	Driver of speed control
		rm_motor_speed/rm_motor_speed_library.h	Definitions of speed control library functions
		rm_motor_inertia_estimate /rm_motor_inertia_estimate.c	Module of inertia estimation
		rm_motor_return_origin/rm_motor_return_origin.c	Module of return origin
src	application/main	mtr_main.h , mtr_main.c	Functions of user interface
		r_mtr_control_parameter.h	Definitions of control parameters
		r_mtr_motor_parameter.h	Definitions of motor parameters
		mtr_adjust_function.h , mtr_adjust_function.c	Process to support servo GUI
	application /user_interface/ics	r_mtr_ics.h , r_mtr_ics.c	Process to support analyzer GUI
		ICS2_RA6T2.h , ICS2_RA6T1.h	Definitions of interface with GUI tool
		ICS2_RA6T2.o , ICS2_RA6T1.o	Library of interface with GUI tool

2.4.2 Module configuration

Module configuration of the software is described below.

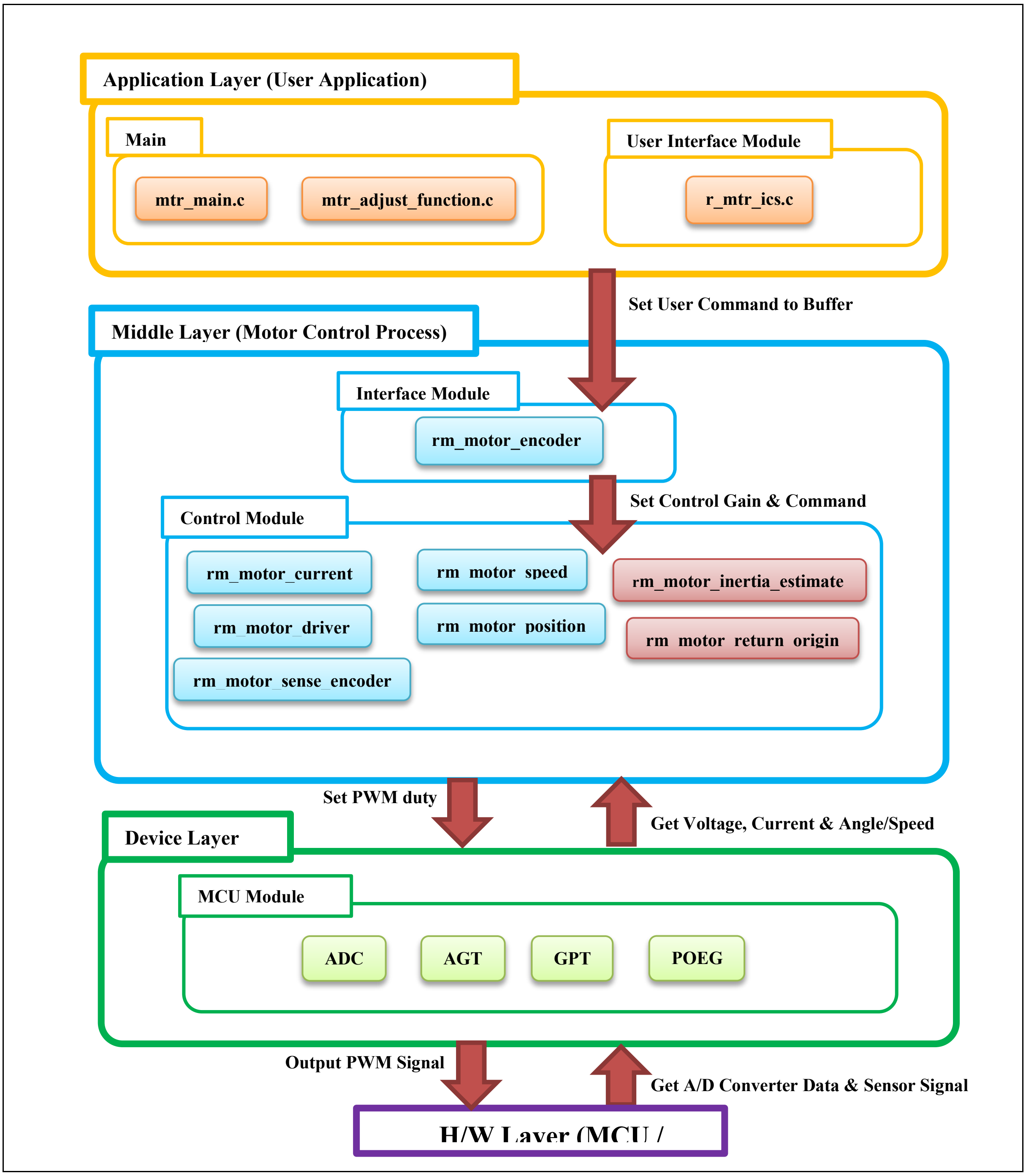


Figure 2-8 Module Configuration

3. Description of functions

Basic functions are not changed from original software “Vector control with encoder” or “Vector control with induction sensor”. Therefore, in this chapter, only “Inertia estimation” and “Return origin” functions are described.

3.1 Contents of control

3.1.1 Method to estimate inertia

The inertia estimation function estimates the rotor and load inertia connected to the motor shaft. Since inertia is an important parameter to drive the motor properly, this function is used to estimate it.

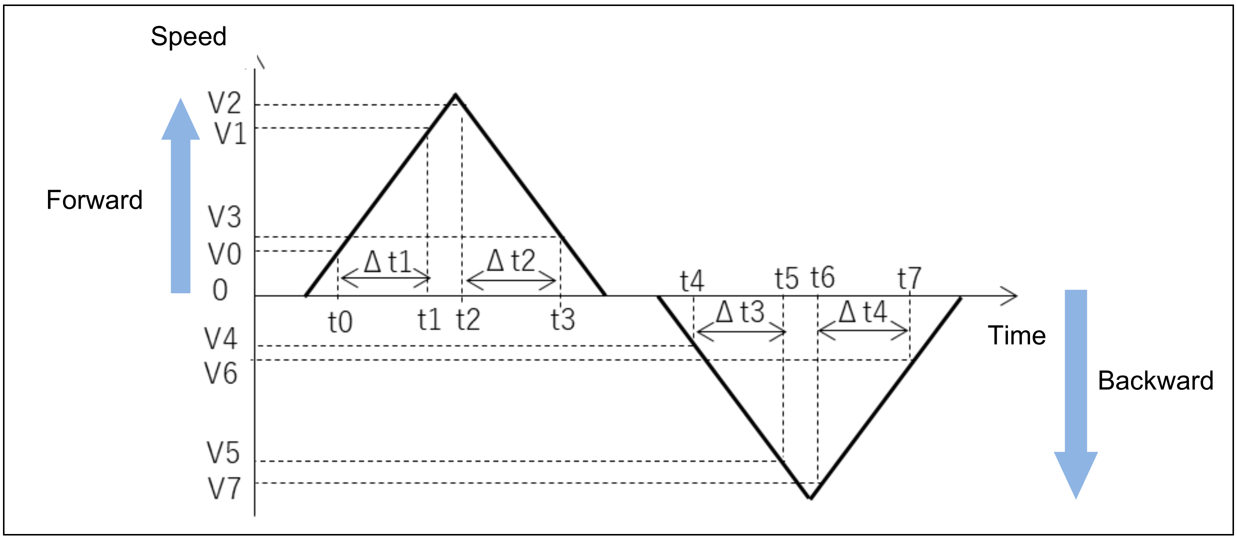


Figure 3-1 Action at inertia estimation

Inertia estimation is performed automatically after the function is started. The motor is rotated forward and backward in same angle with position control like as above figure. Inertia is estimated with the acceleration period at forward(Δt1), deceleration period at forward(Δt2), acceleration period at backward(Δt3), and deceleration at backward(Δt4), the average of q-axis current and acceleration in each period.

3.1.2 Return origin

Return to origin is an action that positions the motor to the mechanical reference position. It is used to return to the reference position when the reference position is not known at startup.

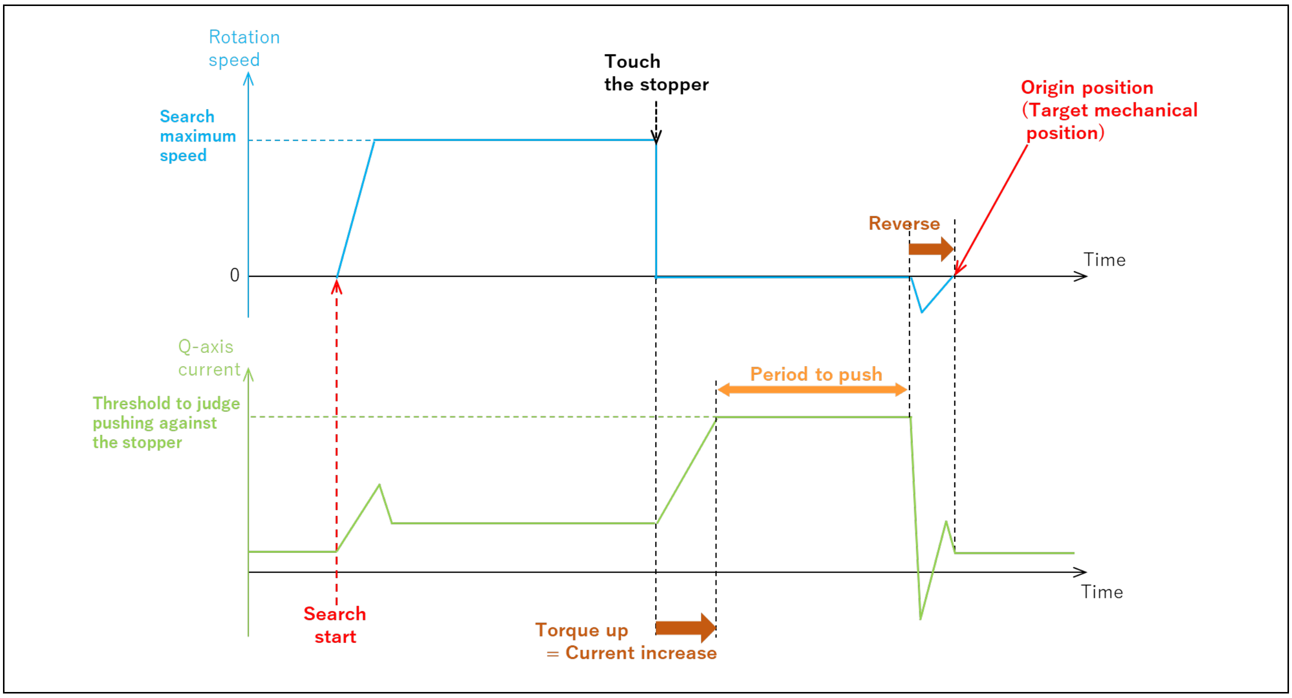


Figure 3-2 Example of return origin movement with pushing method

After the function is started, the motor starts rotation according to set search speed and direction. When the motor hits the stopper, torque is increased. At that time, q-axis current is also increased according to torque. When the q-axis current reaches set threshold value, the motor is judged to hit the stopper. After waiting for set period for pushing, the motor rotate reverse according to set reverse angle, then the function finish.

3.2 Function specifications

3.2.1 API functions for inertia estimation

The list of API functions for inertia estimation is shown below.

Table 3-1 API functions for inertia estimation

API function	Arguments	Contents
RM_MOTOR_INERTIA_ESTIMATE_Open	motor_inertia_estimate_ctrl_t * const p_ctrl / Pointer of structure for inertia estimate module variables motor_inertia_estimate_cfg_t const * const p_cfg / Pointer of structure for configuration of inertia estimate	Open inertia estimate module
RM_MOTOR_INERTIA_ESTIMATE_Close	motor_inertia_estimate_ctrl_t * const p_ctrl / Pointer of structure for inertia estimate module variables	Close inertia estimate module
RM_MOTOR_INERTIA_ESTIMATE_Start	motor_inertia_estimate_ctrl_t * const p_ctrl / Pointer of structure for inertia estimate module variables	Start inertia estimation
RM_MOTOR_INERTIA_ESTIMATE_Stop	motor_inertia_estimate_ctrl_t * const p_ctrl / Pointer of structure for inertia estimate module variables	Stop (cancel) inertia estimation
RM_MOTOR_INERTIA_ESTIMATE_Reset	motor_inertia_estimate_ctrl_t * const p_ctrl / Pointer of structure for inertia estimate module variables	Reset inertia estimate module
RM_MOTOR_INERTIA_ESTIMATE_InfoGet	motor_inertia_estimate_ctrl_t * const p_ctrl / Pointer of structure for inertia estimate module variables motor_inertia_estimate_info_t * const p_info / Pointer of structure to get information from inertia estimate module	Get the information from inertia estimate process
RM_MOTOR_INERTIA_ESTIMATE_DataSet	motor_inertia_estimate_ctrl_t * const p_ctrl / Pointer of structure for inertia estimate module variables motor_inertia_estimate_set_data_t * const p_set_data / Pointer of structure to set data to inertia estimate module	Set necessary data to inertia estimate process
RM_MOTOR_INERTIA_ESTIMATE_SpeedCyclic	motor_inertia_estimate_ctrl_t * const p_ctrl / Pointer of structure for inertia estimate module variables	Inertia estimate process which should be performed in speed control cyclic
RM_MOTOR_INERTIA_ESTIMATE_CurrentCyclic	motor_inertia_estimate_ctrl_t * const p_ctrl / Pointer of structure for inertia estimate module variables	Inertia estimate process which should be performed in current control cyclic
RM_MOTOR_INERTIA_ESTIMATE_ParameterUpdate	motor_inertia_estimate_ctrl_t * const p_ctrl / Pointer of structure for inertia estimate module variables motor_inertia_estimate_cfg_t const * const p_cfg / Pointer of structure for configuration of inertia estimate	Update configuration data about inertia estimation

3.2.2 API functions for return origin

The list of API functions for return origin is shown below.

Table 3-2 API functions for return origin

API function	Arguments	Contents
RM_MOTOR_RETURN_ORIGIN_Open	motor_return_origin_ctrl_t * const p_ctrl / Pointer of structure for return origin module variables motor_return_origin_cfg_t const * const p_cfg / Pointer of structure for configuration of return origin	Open return origin module
RM_MOTOR_RETURN_ORIGIN_Close	motor_return_origin_ctrl_t * const p_ctrl / Pointer of structure for return origin module variables	Close return origin module
RM_MOTOR_RETURN_ORIGIN_Start	motor_return_origin_ctrl_t * const p_ctrl / Pointer of structure for return origin module variables	Start return origin function
RM_MOTOR_RETURN_ORIGIN_Stop	motor_return_origin_ctrl_t * const p_ctrl / Pointer of structure for return origin module variables	Stop (cancel) return origin function
RM_MOTOR_RETURN_ORIGIN_Reset	motor_return_origin_ctrl_t * const p_ctrl / Pointer of structure for return origin module variables	Reset return origin module
RM_MOTOR_RETURN_ORIGIN_InfoGet	motor_return_origin_ctrl_t * const p_ctrl / Pointer of structure for return origin module variables motor_return_origin_info_t * const p_info / Pointer of structure to get information from return origin module	Get information from return origin module
RM_MOTOR_RETURN_ORIGIN_DataSet	motor_return_origin_ctrl_t * const p_ctrl / Pointer of structure for return origin module variables motor_return_origin_set_data_t * const p_set_data / Pointer of structure to set data to return origin module	Set necessary data to return origin module
RM_MOTOR_RETURN_ORIGIN_SpeedCyclic	motor_return_origin_ctrl_t * const p_ctrl / Pointer of structure for return origin module variables	Return origin process which should be performed in speed control cyclic
RM_MOTOR_RETURN_ORIGIN_ParameterUpdate	motor_return_origin_ctrl_t * const p_ctrl / Pointer of structure for return origin module variables motor_return_origin_cfg_t const * const p_cfg / Pointer of structure for configuration of return origin	Update configuration data about return origin

3.3 Contents of control

3.3.1 Configuration Options

The configuration options of the vector control with encoder module for motor can be configured using the RA Configurator. The changed options are automatically reflected to the hal_data.c when generating code.

Table 3-3 Configuration Options for Current Control for inertia estimation

Configuration Options (rm_motor_inertia_estimate.h)	
Options	**Description**
Moved position distance to measure (degree)	Moving angle at inertia estimation [degree]
Maximum speed (rpm)	Maximum rotation speed at inertia estimation [rpm]
Acceleration time	Acceleration of speed [rpm/sec]
Motor inertia	Rotor inertia at initialization (Please set motor specified inertia value.)
Low threshold to judge speed	Ratio to judge the beginning of measurement for speed [%] (Please set ratio against maximum speed.)
High threshold to judge speed	Ratio to judge the end of measurement for speed [%] (Please set ratio against maximum speed.)
Time to wait moving stability (sec)	Time to wait the stability of position control [sec]
Cyclic period of current control (sec)	Cyclic period of current control [sec]
Cyclic period of speed control (sec)	Cyclic period of speed control [sec]
Motor pole pairs	Pole pairs of target motor
Motor magnet flux (Wb)	Magnetic flux [Wb]
Interval time	Interval time to change forward -> backward control.

Table 3-4 Configuration Options for Current Control for return origin

Configuration Options (rm_motor_return_origin.h)	
Options	**Description**
Search speed (rpm)	Maximum speed at search origin [rpm]
Acceleration of speed (rpm/sec)	Accel speed at reverse movement from stopper [rpm/sec]
Cyclic period of speed control (sec)	Cyclic period of speed control [sec]
Maximum current (A)	Maximum of current [A]
Percentage of current to judge pushing (%)	Ratio to judge pushing at the stopper (Please set the ratio to maximum current) [%]
Pushing time (sec)	Period to pushing against the stopper [sec]
Degree to judge none stopper	If the motor rotates over this value, the stopper is judged not to be detected.[degree]
Degree to return	Reverse angle from the stopper [degree]
Mechanical gear ratio	If an external gear is set, please set the ratio.

4. Operation with Renesas Motor Workbench (RMW)

Servo tuning functions can be performed easily with the tool “Renesas Motor Workbench” (from here “RMW”). In this chapter, the operation how to use is described.

4.1 Import the sample project

Sample software can be imported to e2studio by below operation.

1. File -> Import

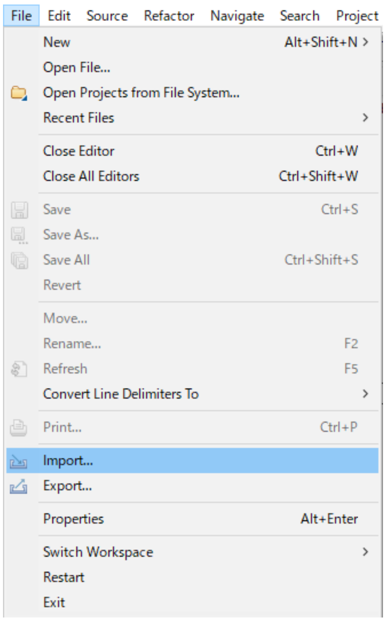


Figure 4-1 File menu

2. Select “Existing Projects into Workspace”, then click “Next” button.

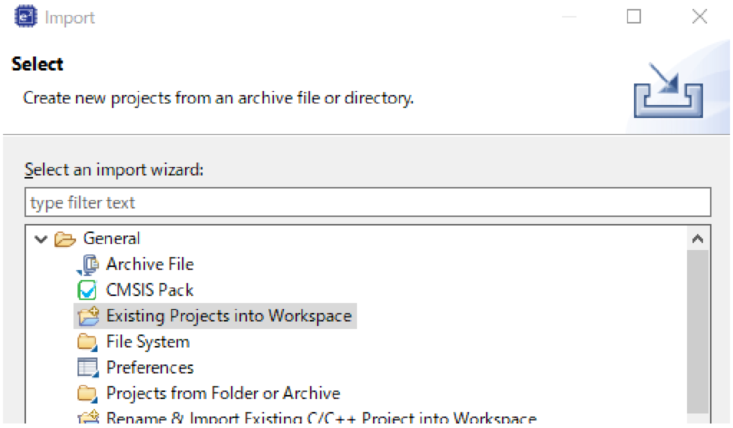
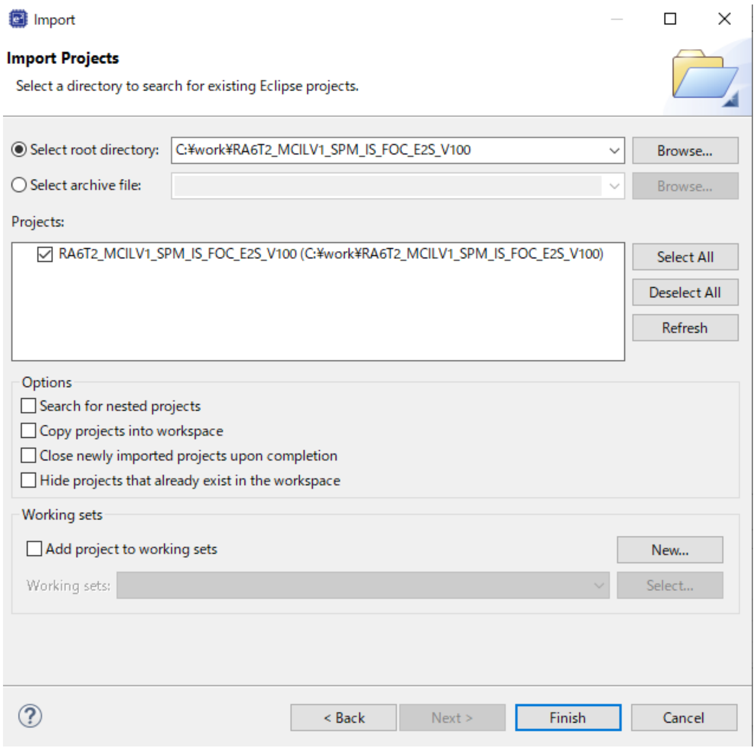

Import
Select
Create new projects from an archive file or directory.
Select an import wizard:
type filter text
General
Archive File
CMSIS Pack
Existing Projects into Workspace
File System
Preferences
Projects from Folder or Archive
Rename & Import Existing C/C++ Project into Workspace

Figure 4-2 Import menu

3. Select the project file. After click “Finish” button, selected project is imported.



Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:¥work¥RA6T2_MCILV1_SPM_IS_FOC_E2S_V100

○ Select archive file:

Projects:

☑ RA6T2_MCILV1_SPM_IS_FOC_E2S_V100 (C:¥work¥RA6T2_MCILV1_SPM_IS_FOC_E2S_V100)

Options

☐ Search for nested projects

☐ Copy projects into workspace

☐ Close newly imported projects upon completion

☐ Hide projects that already exist in the workspace

Working sets

☐ Add project to working sets

Working sets:

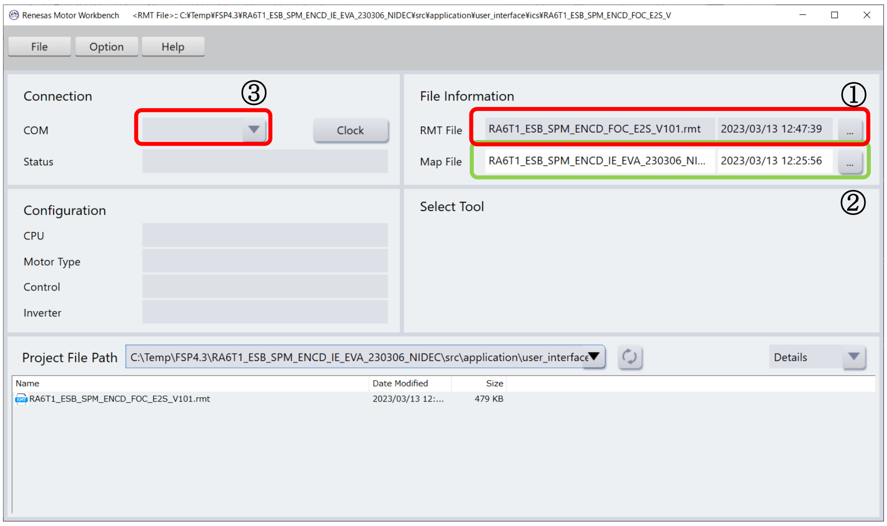
Figure 4-3 Import “Project”

4.2 Build and Debug

Please refer the manual “Basic guide e2studio users manual (R20UT4204)”.

4.3 Operation at the beginning

When RMW application is started, the below window is displayed.



Renesas Motor Workbench <RMT File>:: C:¥Temp¥FSP4.3¥RA6T1_ESB_SPM_ENCD_IE_EVA_230306_NIDEC¥src¥application¥user_interface¥ics¥RA6T1_ESB_SPM_ENCD_FOC_E2S_V

RMT File: RA6T1_ESB_SPM_ENCD_FOC_E2S_V101.rmt 2023/03/13 12:47:39

Map File: RA6T1_ESB_SPM_ENCD_IE_EVA_230306_NI... 2023/03/13 12:25:56

Project File Path: C:\Temp\FSP4.3\RA6T1_ESB_SPM_ENCD_IE_EVA_230306_NIDEC\src\application\user_interface

Name	Date Modified	Size
RA6T1_ESB_SPM_ENCD_FOC_E2S_V101.rmt	2023/03/13 12:...	479 KB

Figure 4-4 First window of RMW

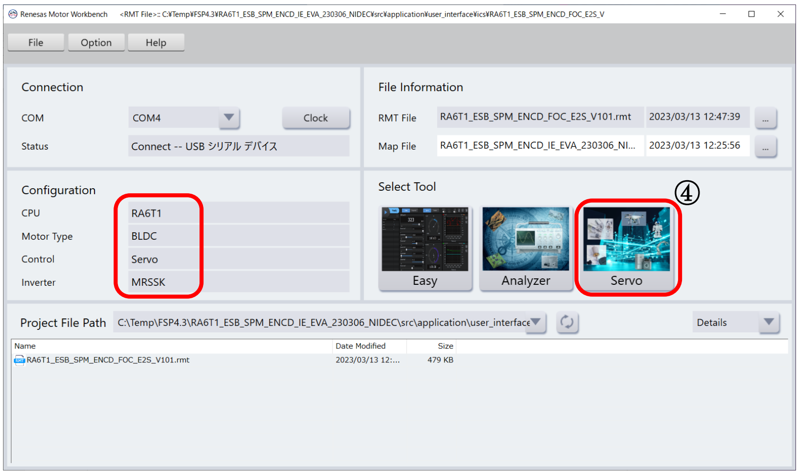
Please perform below procedures.

i. Click the icon "…" of right side of "File information" / "RMT File". And then select RMT file (***.rmt) which is included below the folder "src/application/user_interface/ics".

ii. Click the icon "…" of right side of "Map File". And then select a map file (***.map) which is included below same folder.

After that, a window to reflect variable map is popped up, please click "Set".

iii. After above procedure finished, connect the application to the target. Click the icon "▼" of right side of "Connection"/"COM". And then select correct COM port from the pull down menu.



Renesas Motor Workbench <RMT File>:: C:¥Temp¥FSP4.3¥RA6T1_ESB_SPM_ENCD_IE_EVA_230306_NIDEC¥src¥application¥user_interface¥ics¥RA6T1_ESB_SPM_ENCD_FOC_E2S_V

Connection

COM: COM4

Status: Connect -- USB シリアル デバイス

File Information

RMT File: RA6T1_ESB_SPM_ENCD_FOC_E2S_V101.rmt 2023/03/13 12:47:39

Map File: RA6T1_ESB_SPM_ENCD_IE_EVA_230306_NI... 2023/03/13 12:25:56

Configuration

CPU: RA6T1

Motor Type: BLDC

Control: Servo

Inverter: MRSSK

Project File Path: C:\Temp\FSP4.3\RA6T1_ESB_SPM_ENCD_IE_EVA_230306_NIDEC\src\application\user_interface

Name	Date Modified	Size
RA6T1_ESB_SPM_ENCD_FOC_E2S_V101.rmt	2023/03/13 12:...	479 KB

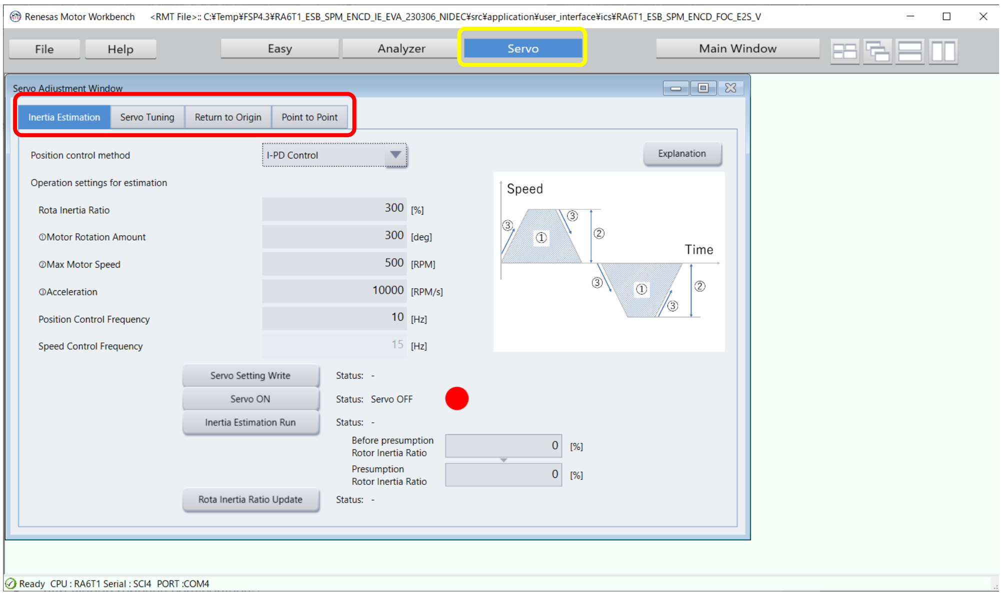
Figure 4-5 Main window of RMW

If the connection is valid, then the window changes like above.

Please confirm the information in "Configuration" like above. (Figure 4-5 is an example with RA6T1.)

If the information is displayed correctly,

iv. Click "Servo" icon in "Select Tool".



Renesas Motor Workbench <RMT File>:: C:¥Temp¥FSP4.3¥RA6T1_ESB_SPM_ENCD_IE_EVA_230306_NIDEC¥src¥application¥user_interface¥ics¥RA6T1_ESB_SPM_ENCD_FOC_E2S_V

Servo Adjustment Window

Position control method: I-PD Control

Operation settings for estimation

Rota Inertia Ratio	300	[%]
①Motor Rotation Amount	300	[deg]
②Max Motor Speed	500	[RPM]
③Acceleration	10000	[RPM/s]
Position Control Frequency	10	[Hz]
Speed Control Frequency	15	[Hz]

Servo Setting Write — Status: -

Servo ON — Status: Servo OFF ●

Inertia Estimation Run — Status: -

Before presumption Rotor Inertia Ratio: 0 [%]

Presumption Rotor Inertia Ratio: 0 [%]

Rota Inertia Ratio Update — Status: -

Ready CPU : RA6T1 Serial : SCI4 PORT :COM4

Figure 4-6 Servo function window (selected "Inertia Estimation")

In the case that the application and CPU work correctly, above window is displayed.

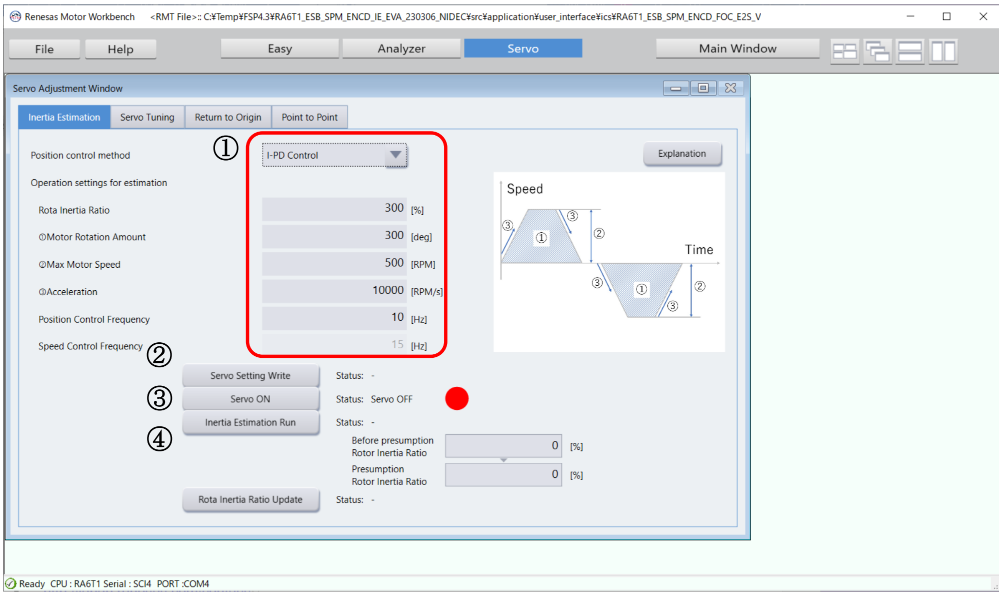
This window is "Servo function" window that is displayed upper side (yellow flamed).

In this window, user can use several functions with selection of function TAB (red flamed).

- Inertia Estimation
- Return to Origin
- Servo Tuning
- Point to Point

How to use these functions are explained below.

4.4 Operation of “Inertia Estimation”



Renesas Motor Workbench <RMT File>:: C:¥Temp¥FSP4.3¥RA6T1_ESB_SPM_ENCD_IE_EVA_230306_NIDEC¥src¥application¥user_interface¥ics¥RA6T1_ESB_SPM_ENCD_FOC_E2S_V

Servo Adjustment Window

Position control method: I-PD Control

Operation settings for estimation

Rota Inertia Ratio: 300 [%]

①Motor Rotation Amount: 300 [deg]

②Max Motor Speed: 500 [RPM]

③Acceleration: 10000 [RPM/s]

Position Control Frequency: 10 [Hz]

Speed Control Frequency: 15 [Hz]

Status: -

Status: Servo OFF

Status: -

Before presumption Rotor Inertia Ratio: 0 [%]

Presumption Rotor Inertia Ratio: 0 [%]

Status: -

Ready CPU : RA6T1 Serial : SCI4 PORT :COM4

Figure 4-7 Operation window of “Inertia Estimation”

User can perform inertia estimation easily with below procedure in this window.

i. Set parameters for inertia estimation.

Table 4-1 Parameters for inertia estimation

Options	Description
Position control method	Specify the method for position control.
Rota Inertia Ratio	Set the rotor inertia ratio.
Motor Rotation Amount	Set the motor rotation amount.
Max Motor Speed	Set the maximum motor speed.
Acceleration	Set the acceleration.
Position Control Frequency	Set the position control natural frequency.
Speed Control Frequency	Set the speed control natural frequency. You can enter it when “Position control method” is “PID Control”.

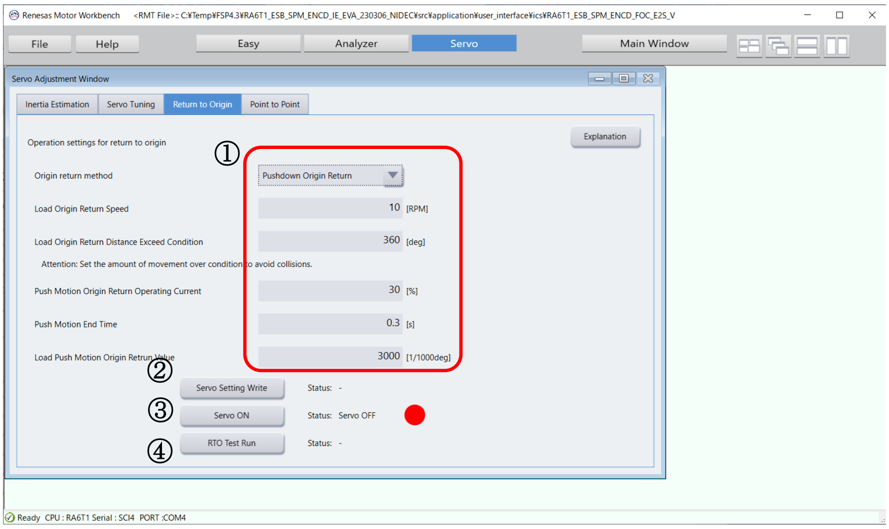
ii. Click “Servo Setting Write” button to reflect the parameters. Even if you don’t change parameters, please click the button to reflect initial values.

iii. Click “Servo ON” button. Then the status changes to “Servo ON” and red indicator changes to green. The motor is turned on “position control”. And clicked button name also changes to “Servo OFF”.

iv. Click “Inertia Estimation Run” button. And then, inertia estimation starts. Clicked button name changes to “Inertia Estimation Stop”. After the process finished correctly, the button returns to “Inertia Estimation Run” automatically. When an error happened, error window is popped up.

When you change to other window, click “Servo OFF” button to cancel “position control”.

4.5 Operation of return origin



Renesas Motor Workbench <RMT File>:: C:¥Temp¥FSP4.3¥RA6T1_ESB_SPM_ENCD_IE_EVA_230306_NIDEC¥src¥application¥user_interface¥ics¥RA6T1_ESB_SPM_ENCD_FOC_E2S_V

Operation settings for return to origin

Origin return method: Pushdown Origin Return

Load Origin Return Speed: 10 [RPM]

Load Origin Return Distance Exceed Condition: 360 [deg]

Attention: Set the amount of movement over condition to avoid collisions.

Push Motion Origin Return Operating Current: 30 [%]

Push Motion End Time: 0.3 [s]

Load Push Motion Origin Retrun Value: 3000 [1/1000deg]

Servo Setting Write — Status: -

Servo ON — Status: Servo OFF ●

RTO Test Run — Status: -

Ready CPU : RA6T1 Serial : SCI4 PORT :COM4

Figure 4-8 Operation window of "Return Origin"

User can perform return origin easily with below procedure in this window.

i. Set parameters for return origin.

Table 4-2 Parameters for return origin

Options	Description
Load Origin Return Speed	Set the load speed for return to origin.
Load Origin Return Distance Exceed Condition	Set the angle to determine impossibility of search during origin search.
Push Motion Origin Return Operating Current	Set the push motion origin return operating current.
Push Motion End Time	Set the push motion ending time.
Load Push Motion Origin Return Value	Set the return value of the load push motion origin return.

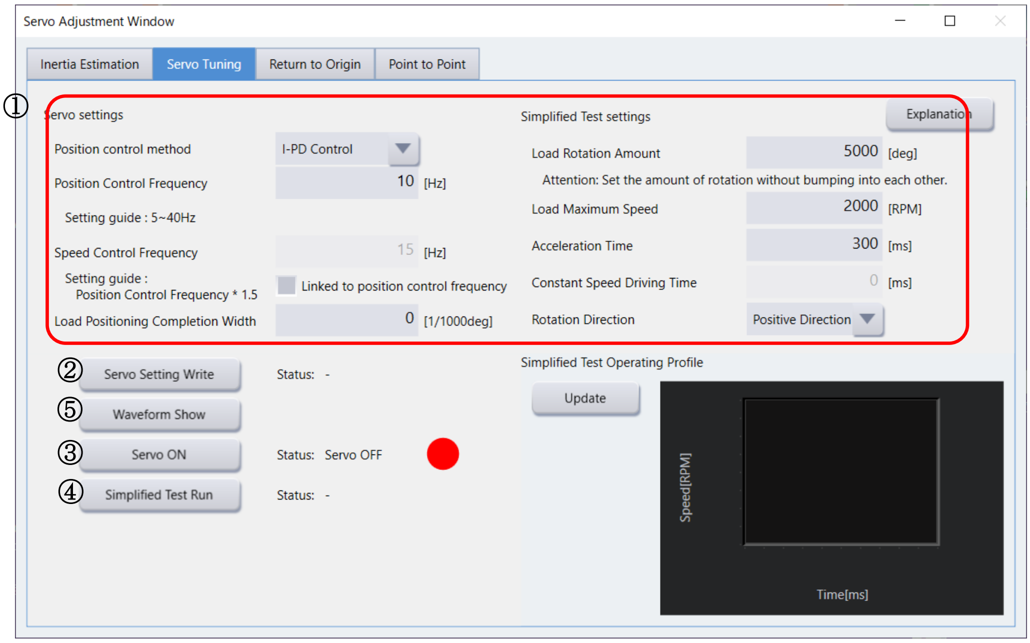
ii. Click "Servo Setting Write" button to reflect the parameters. Even if you don't change parameters, please click the button to reflect initial values.

iii. Click "Servo ON" button. Then the status changes to "Servo ON" and red indicator changes to green. The motor is turned on "position control". And clicked button name also changes to "Servo OFF".

iv. Click "RTO Test Run" button. And then, return origin starts. Clicked button name changes to "RTO Test Stop". After the process finished correctly, the button returns to "RTO Test Run" automatically. When an error happened, error window is popped up.

When you change to other window, click "Servo OFF" button to cancel "position control".

4.6 Operation of servo tuning



Servo settings

Position control method: I-PD Control

Position Control Frequency: 10 [Hz]

Setting guide : 5~40Hz

Speed Control Frequency: 15 [Hz]

Setting guide : Position Control Frequency * 1.5

□ Linked to position control frequency

Load Positioning Completion Width: 0 [1/1000deg]

Simplified Test settings

Load Rotation Amount: 5000 [deg]

Attention: Set the amount of rotation without bumping into each other.

Load Maximum Speed: 2000 [RPM]

Acceleration Time: 300 [ms]

Constant Speed Driving Time: 0 [ms]

Rotation Direction: Positive Direction

Status: -

Status: Servo OFF ●

Status: -

Figure 4-9 Operation window of “Servo Tuning”

User can perform servo tuning easily with below procedure in this window.

i. Set parameters for servo tuning.

Table 4-3 Parameters for servo tuning

Options	Description
Position control method	Specify the method for position control
Position Control Frequency	Set the position control natural frequency.
Speed Control Frequency	Set the speed control natural frequency. You can enter it when “Position control method” is “PID Control”.
Linked to position control frequency	When checked, the position control natural frequency is multiplied by 1.5, and the value is set to the speed control natural frequency automatically. The value set in No.3 is overwritten.
Load Positioning Completion	Set the range of the load positioning completion.
Load Rotation Amount	Set the load rotation amount.
Load Maximum Speed	Set the load maximum speed.
Acceleration Time	Set the acceleration time.
Rotation Direction	Set the rotation direction.

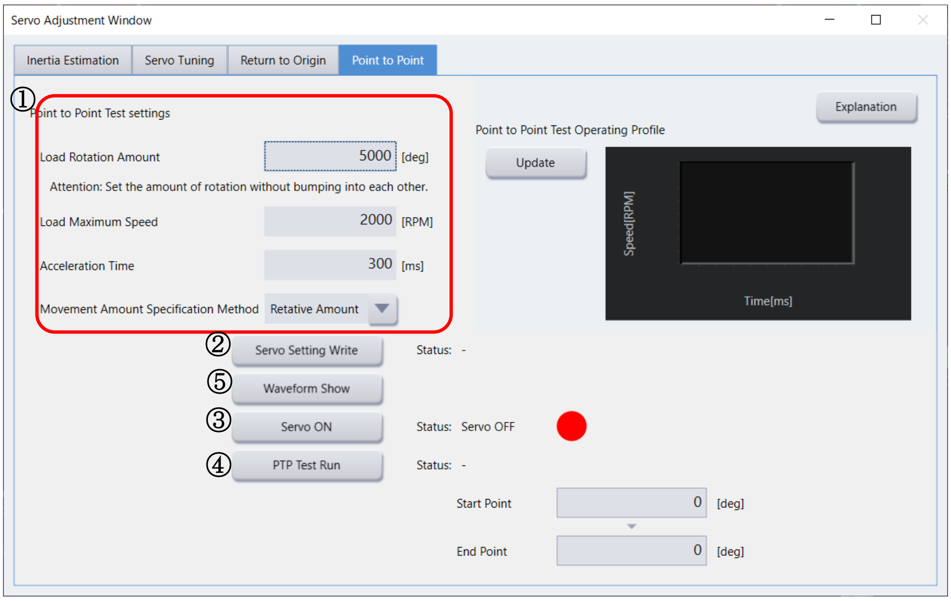
ii. Click “Servo Setting Write” button to reflect the parameters. Even if you don’t change parameters, please click the button to reflect initial values.

iii. Click “Servo ON” button. Then the status changes to “Servo ON” and red indicator changes to green. The motor is turned on “position control”. And clicked button name also changes to “Servo OFF”.

iv. Click “Simplified Test Run” button. And then, servo tuning starts. Clicked button name changes to “Simplified Test Stop”. After the process finished correctly, the button returns to “Simplified Test Run” automatically. When an error happened, error window is popped up.

When you change to other window, click “Servo OFF” button to cancel “position control”.

4.7 Operation of “Point to Point”



Point to Point Test settings

Load Rotation Amount 5000 [deg]

Attention: Set the amount of rotation without bumping into each other.

Load Maximum Speed 2000 [RPM]

Acceleration Time 300 [ms]

Movement Amount Specification Method Retative Amount

Point to Point Test Operating Profile

Status: -

Status: Servo OFF

Status: -

Start Point 0 [deg]

End Point 0 [deg]

Figure 4-10 Operation window or “Point to point”

User can perform “Point to point” easily with below procedure in this window.

i. Set parameters for “Point to point”.

Table 4-4 Parameters for “Point to point”

Options	Description
Load Rotation Amount	Set the load rotation amount.
Load Maximum Speed	Set the load maximum speed.
Acceleration Time	Set the acceleration time.
Movement Amount Specification Method	Set the method for specifying movement amount.

ii. Click “Servo Setting Write” button to reflect the parameters. Even if you don’t change parameters, please click the button to reflect initial values.

iii. Click “Servo ON” button. Then the status changes to “Servo ON” and red indicator changes to green. The motor is turned on “position control”. And clicked button name also changes to “Servo OFF”.

iv. Click “PTP Test Run” button. And then, servo tuning starts. Clicked button name changes to “PTP Test Stop”. After the process finished correctly, the button returns to “PTP Test Run” automatically. When an error happened, error window is popped up.

When you change to other window, click “Servo OFF” button to cancel “position control”.

5. Reference Documents

Renesas Motor Workbench User’s Manual (R21UZ0004)
Renesas Motor Workbench Quick start guide (R21QS0011)
RA6T1 Group User’s Manual: Hardware (R01UH0897)
RA6T2 Group User’s Manual: Hardware (R01UH0951)
Application note: ‘Encoder vector control for permanent magnet synchronous motor (Algorithm)’ (R01AN3789)
Motor Control Evaluation System for RA Family (R12UZ0078)
MCK-RA6T2 User's Manual (R12UZ0091)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	May 30, 2023	-	First edition issued
1.01	Aug 30, 2024	1	Updated Target software

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

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

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

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

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

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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