User's Manual

Revised Edition

Motor Control Development Support Tool Renesas Motor Workbench 3.3.1

User's Manual

RA/RX/RL78 Family

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Technology Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev 4.21 Mar 2026

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/.

Table of Contents

1.Renesas Motor Workbench Overview

This document describes the various functions and operating procedures of the Motor Control Development Support Tool Renesas Motor Workbench (hereafter, RMW). Read this manual carefully to ensure safe use.

The description on functions and operations in this document assumes that the target is correctly connected. Some features may not be available or may be restricted if it is in offline mode or there is a problem with connection.

1.1 RMW Functions

RMW is a development support tool for debugging, analyzing, and tuning motor control programs. RMW provides the following features:

- Analyzer tool
  - Dynamically reads and writes variables in a MCU.
  - Displays changes in variables as waveforms in real-time.
  - Sets trigger and performs zoom analysis.
- Easy tool
  - Intuitive operations enable you to control speed and position of the motor easily.
  - You can check the drive status at a glance with meters and graphs.
- Tuner tool
  - Automatically obtains and tunes the parameters necessary for vector control.
  - Enables fine tuning by means of a manual tuning function.
  - Outputs the tuning results (header file, PDF).
- Servo tool
  - Inertia Estimation
    Estimates the load inertia and the inertia of the rotor and the shaft connected with the motor-axis by driving the motor actually.
  - Servo Tuning
    Configures the settings for servo operation such as position control method and control parameters.
  - Return to Origin
    Sets the method for return to origin and the return speed, etc.
  - Point to Point
    Performs PTP (Point to Point) operation for one axis.

1.2 Displaying and Entering Decimal Points

When displaying numerical values or entering numerical values in RMW, a period is used for the decimal point when dealing with decimals.
Note that because this is a fixed setting that does not depend on the language setting of the OS, a period is used for displaying and entering numerical values in RMW even if the language uses a comma for the decimal point.

1.3 Related Document

For installation, environment settings, and preparations of RMW, refer to the following Quick Start Guide.

- Renesas Motor Workbench Quick Start Guide (R21QS0011)

We release some functions of RMW (communication function to the target MCU, etc) in the form of DLL. When developing an original application on a PC, please refer to the following documentation.

- Renesas Motor Workbench DLL for communication Function Manual (R20AN0683)

2.Main Window

2.1 Overview

Main Window is a window that opens when RMW is started. This window is for basic operations of RMW, such as connecting to a communication board, loading project information, and launching tools.

2.2 Window Structure

The structure of Main Window is shown below.

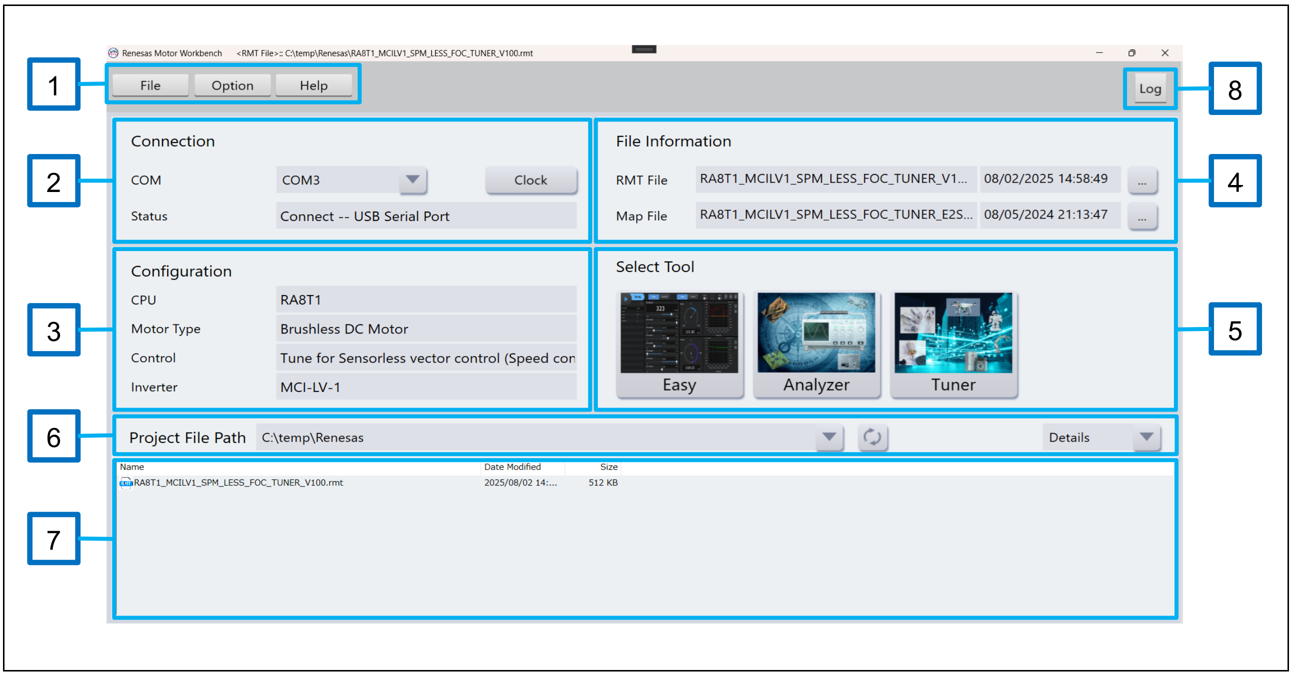


Figure 2-1 Main Window Structure

Table 2-1 Names of Each Part in Main Window (No. Corresponds to the Above Figure.)

No.	Name	Explanation	Reference Section
1	Menu bar	Menu bar (File, Option, Help)	2.3.1
2	Connection	Set communication with the target system.	2.3.2
3	Configuration	Displays information on the target system **(only for the execution file provided by Renesas).**	2.3.3
4	File Information	Displays the RMT file and Map file that are currently valid.	2.3.4
5	Select Tool	Displays icons for available tools.	2.3.5
6	Project File Path	• Displays the paths to the RMT files that were loaded before in a pull-down list. (max. 10) • Lists the RMT files included in the chosen folder into File List below. (The RMT file is not loaded at this point.)	2.3.6
7	File List	• Displays a list of the RMT files in the folder specified in Project File Path. • Double-click to load the RMT file. • Users can directly add an RMT file here by drag & drop as well.	
8	Display log	• Display operation logs.	2.3.7

2.3 Explanation of Items on Main Window

2.3.1 Menu Bar

2.3.1.1 File Menu

The functions of File menu are shown below.

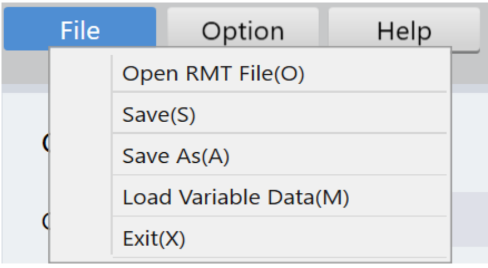


Figure 2-2 File Menu

Table 2-2 Functions of File Menu

Name	Explanation
Open RMT File	Loads an RMT file [Note]
Save	Overwrites and saves the loaded information about variables and RMW settings into the RMT file.
Save As	Saves the loaded information about variables and RMW settings into a new RMT file with a new name.
Load Variable Data	Loads a list of the global variables (“variable information”) held in the user program from a Map file.
Exit	Terminates RMW (displays the Exit screen.)

Note: RMT project file (including both variable information and information on settings configured by RMT tools)

2.3.1.2 Option Menu

The functions of Option menu are shown below.

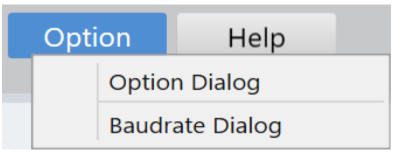


Figure 2-3 Option Menu

Table 2-3 Functions of Option Menu

Name	Explanation
Option Dialog	Configures various RMW settings. Displays the Option Dialog screen.
Baudrate Dialog	Sets the baud rate when the built-in type communication library is used.

2.3.1.3 Help Menu

The functions of Help menu are shown below.

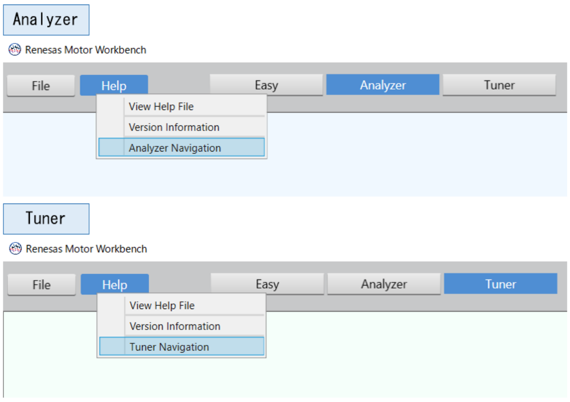


Figure 2-4 Help Menu

Table 2-4 Functions of Help Menu

Name	Explanation
View Help File	Displays RMW help information.
Version Information	Displays the RMW version information.
Select Navigation	Selectable from the menu while Main Window is displayed. Displays the dialog to load an RMT or Map file. Select Navigation is displayed by default on RMW launching. (It is possible to set not to display.)
Analyzer Navigation	Selectable from the menu while Analyzer tool is displayed. Displays the dialog to check the steps for using Analyzer functions. Analyzer Navigation is displayed by default when Analyzer starts. (It is possible not to display.)
Tuner Navigation	Selectable from the menu while Tuner tool is displayed. Displays the dialog to check the steps for using Tuner functions. Tuner Navigation is displayed by default when Tuner starts. (It is possible not to display.)

2.3.2 Connection Area

The functions of the Connection area are shown below.

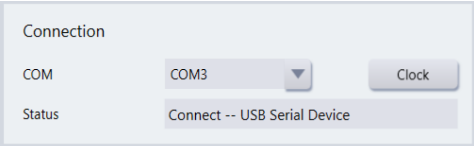


Figure 2-5 Connection Area

Table 2-5 Functions of Connection

Name	Explanation
COM	• Connectable COM numbers or "Offline Mode" are displayed as a pull-down list. • When a COM number is selected, connection to that COM is attempted.
Status	• Displays the connection status when a COM number is selected and connection to that COM is being attempted. • Displays "Connect - USB Serial Port" when the connection is successful.
Clock button	• Click this button to set a communication frequency. • For Motor RSSK, the value is fixed to 8 MHz.

2.3.3 Configuration

The functions of Configuration [Note] are shown below. Configuration area displays the contents defined in the control program.

Note: Displayed only when downloading the sample code provided by Renesas into an MCU.

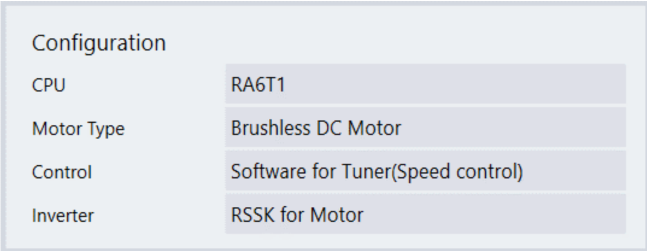


Figure 2-6 Configuration Area

Table 2-6 Functions of Configuration

Name	Explanation
CPU	Displays the name of the target CPU that is connected.
Motor Type	Displays the type of motor that the control program targets.
Control	Displays the control method of the control program.
Inverter	Displays information about the inverter board that the control program targets.

2.3.4 File Information

The functions of File Information are shown below. Hovering a mouse cursor on a file name displays its information, and clicking the right button loads the file.

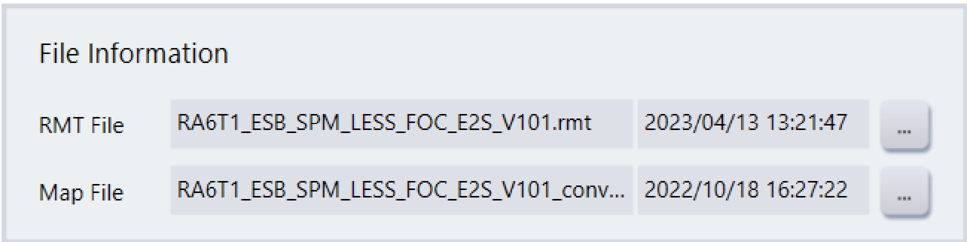


Figure 2-7 File Information Area

Table 2-7 Functions of File Information

Name	Explanation
RMT File	Displays the name of the RMT file being loaded and its most recent modification date.
Map File	Displays the name of the Map file being loaded and its most recent modification date.

2.3.5 Select Tool

Select Tool displays icons for available tools. Clicking a tool icon starts that tool.

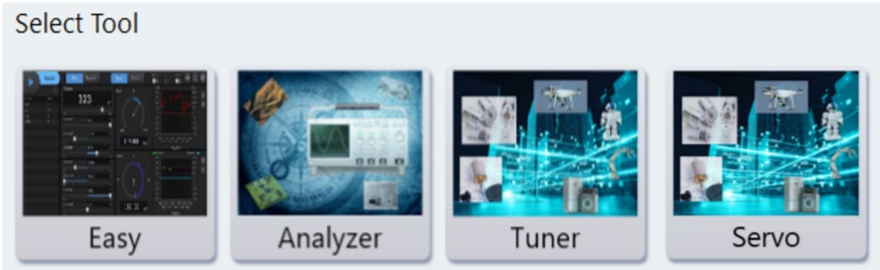


Figure 2-8 Select Tool Area

Table 2-8 Functions of Select Tool

Name	Explanation
Select Tool (Icon display)	• Displays icons for available tools (Easy, Analyzer, Tuner, Servo) according to the program. • Clicking an icon starts that tool and switches the window.

The available tools are determined according to the value of the variable “g_u2_conf_tool” that is defined in the motor control program.

The variable value is loaded on COM connections, and available tools are displayed according to that value.

The conditions of availability of the tools are shown below.

Table 2-9 Availability Conditions of Tools

Name	Availability condition
Easy	Always available
Analyzer	Always available
Tuner	Available when the value of the variable “g_u2_conf_tool” is 0x600.
Servo	Available when the value of the variable “g_u2_conf_tool” is 0xA00.

2.3.6 Project File Path and File List

Project File Path and File List are for managing project folders of RMW.

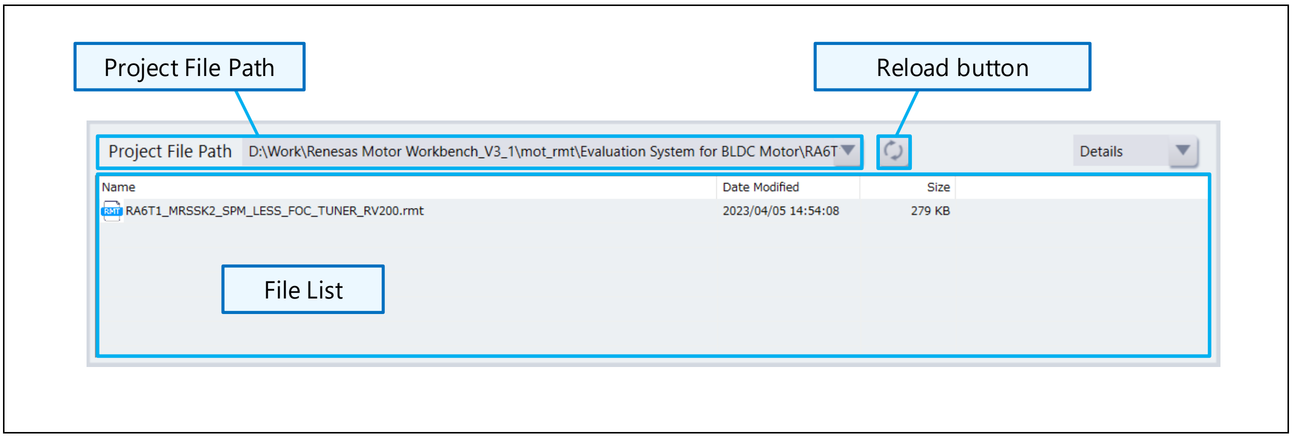


Figure 2-9 Project File Path and File List and Relation to Project Folder

Table 2-10 Functions of Project File Path and File List

Name	Explanation
Project File Path	• Displays paths to the project folders (folders in which the loaded RMT file exists) that have been loaded before in a pull-down menu. (max.10) • You cannot enter a path here directly. • Select a path, and the RMT files in the selected project folder will be displayed in File List below. (The RMT file has not been loaded yet at the time it is displayed in the File List.) • When you click the reload button on the right of Project File Path, the display of File List (list of the RMT files in the project folder) is refreshed.
File List	• Displays a list of the RMT files in the project folder specified by Project File Path. • When you double-click the displayed RMT file, it is loaded into RMW. • If you added/deleted an RMT file in the project folder on a PC after specifying a path in Project File Path, press the reload button next to the Project File Path to refresh the display. (It is not reflected automatically.) • **Notice for file operation:** - If you add an RMT file on a PC directly to File List by drag & drop, that file is displayed in File List and copied to the project folder on the PC. - If you delete an RMT file by the right-click menu on the File List, the RMT file on the PC is deleted as well.

2.3.7 Display Log

Pressing the Log button on the menu bar displays the operation log on the right side of the window. Pressing the Log button again closes the log display.

Logs can be displayed not only in the Main Window but also in the tool window.

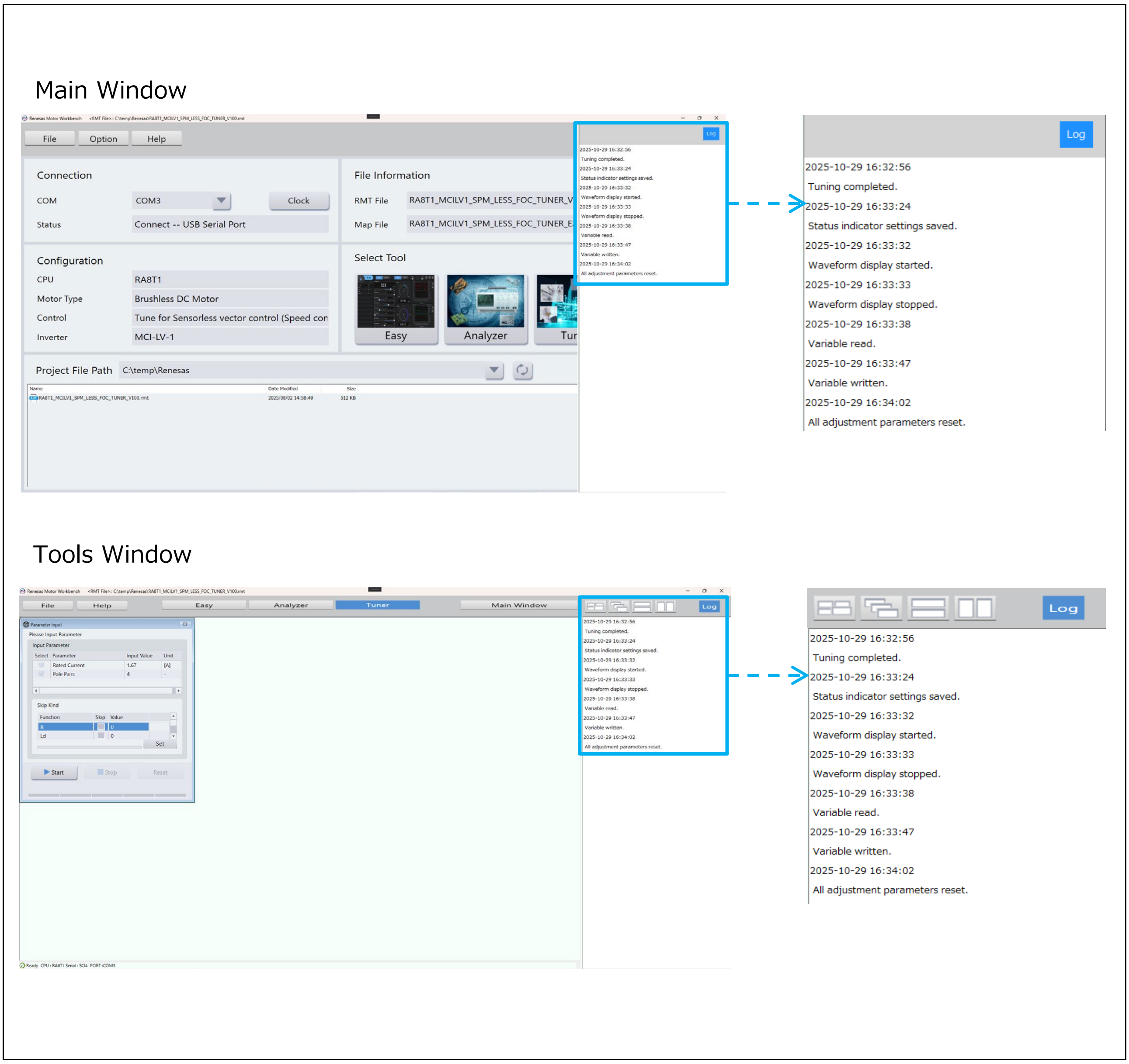


Figure 2-10 Display Logs

2.4 Main Window Operation

2.4.1 Loading RMT File

An RMW project file (RMT file) stores both the variable information loaded from a Map file and the RMW tool setting information.

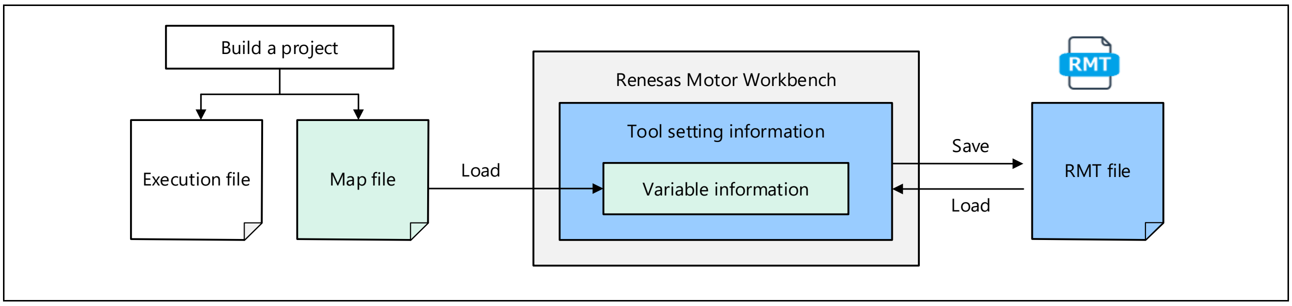


Figure 2-11 RMT File

Loading a project file means loading an RMT file into RMW. When an RMT file is loaded, information loaded previously is cleared. The methods for loading an RMT file are described below.

2.4.1.1 Loading from “File” Menu or “Open RMT File” Button

Select “Open RMT File” from “File” menu or click the “Open RMT File” button next to the RMT File field in File Information, and a dialog for opening an RMT file will be displayed. Specify a file and click the “Open” button, and the RMT file will be loaded.

When the RMT file is loaded, its path and name are displayed in the title bar of Main Window, and the contents of File Information, Project File Path, and File List are updated.

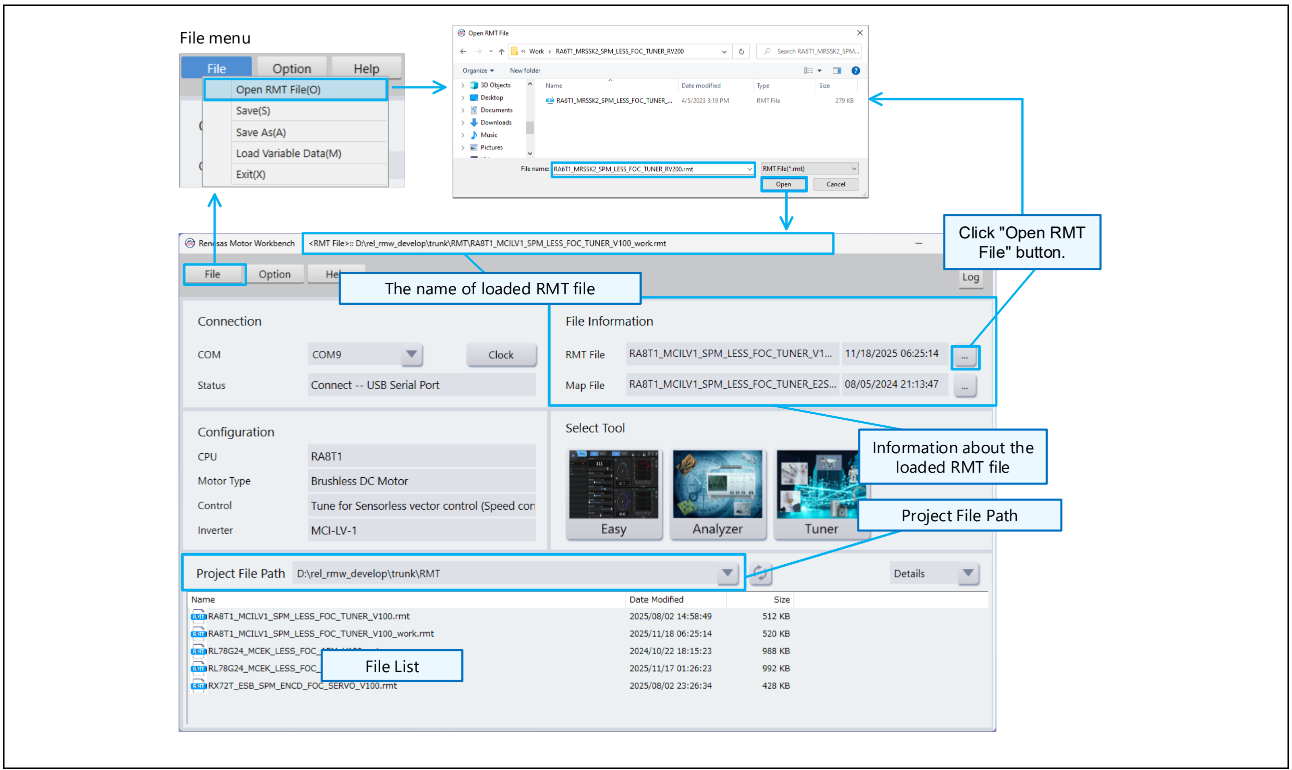


Figure 2-12 Loading RMT File From File Menu

2.4.1.2 Loading with Project File Path and File List

In the pull-down list of Project File Path, the paths to the RMT files loaded most recently are displayed up to 10 paths. When you select a path, RMT files in the project folder you select are shown in File List. You cannot directly enter a path name.

Double-click the RMT file in File List to load it. When the RMT file is loaded, the title bar of Main Window and the contents in File information are updated.

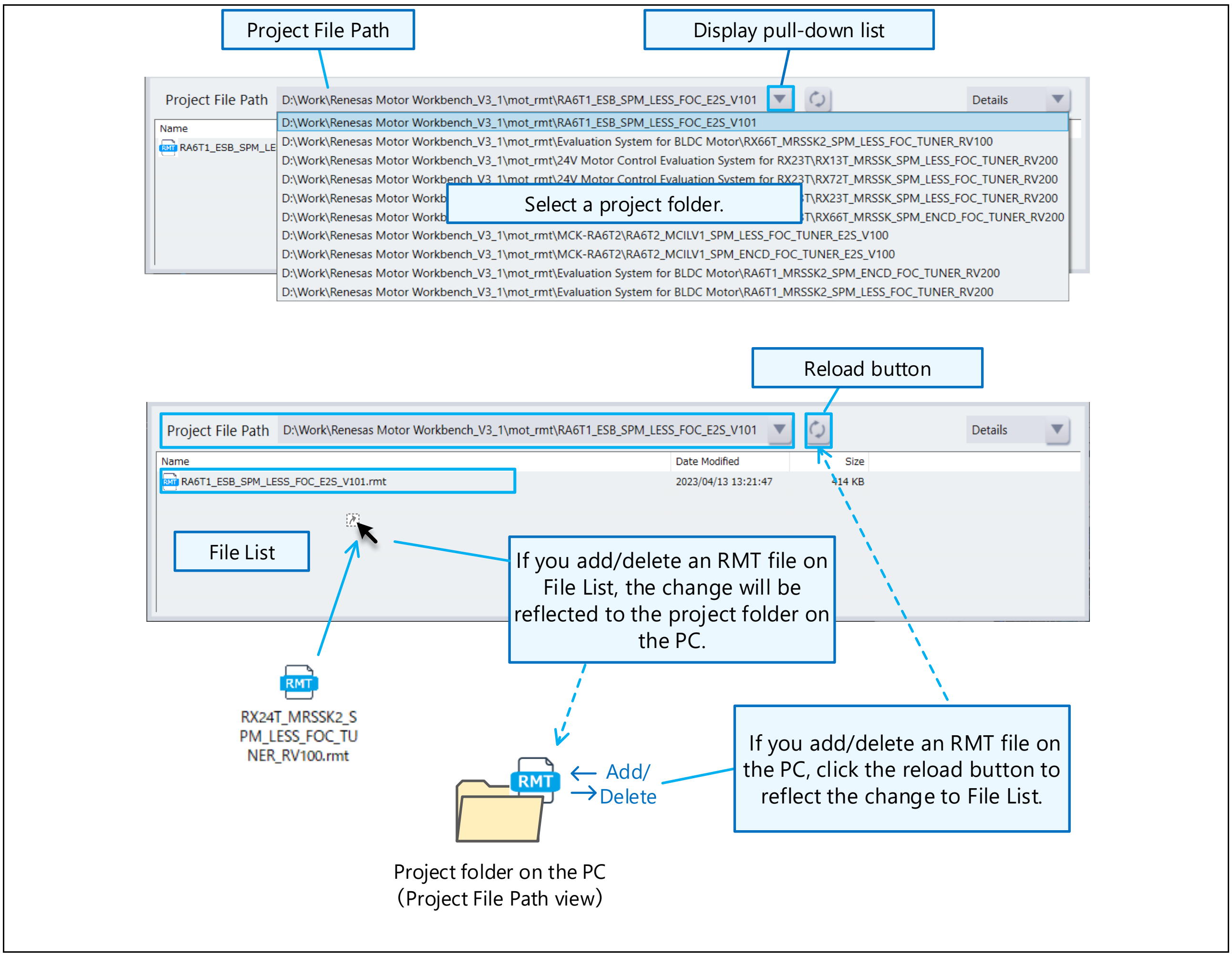


Figure 2-13 Loading RMT File from File List

You can also add an RMT file from a PC to File List by drag & drop operation. The file is copied to the project folder on the PC (specified by Project File Path) by this operation.

In addition, you can delete an RMT file by the right-click menu on File List, and the RMT file in the project folder on the PC is deleted as well.

If you added/deleted an RMT file in the project folder on a PC after specifying a path in Project File Path, press the reload button next to the Project File Path to update the File List. (It is not reflected automatically.)

2.4.1.3 Loading with Select Navigation

You can also use the “Select Navigation” function to load an RMT file. For Select Navigation, see Section 2.5.

2.4.2 Saving RMT File

How to save an RMT file is described below.

2.4.2.1 Saving with “Save” or “Save As” from File Menu

When you select “Save” from Main Window’s File menu, the information is saved into the loaded RMT file by overwriting it. When you select “Save As”, the Save As screen is displayed. Specify an RMT file name and save it.

2.4.2.2 Saving RMT File when Terminating RMW

When terminating RMW, you can select how to save the RMT file on the exit screen. The procedures are described below.

1. Select “Exit” from “File” menu or press the close button on the upper right corner of Main Window, and the exit screen will be displayed.
2. If you want to save the RMT file, select “Save” or “Save As” on the exit screen.

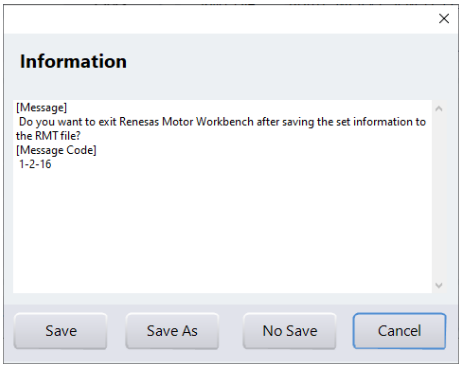


Figure 2-14 Renesas Motor Workbench Exit Screen

Table 2-11 Functions of Buttons on Exit Screen

Button	Function
Save	Saves the RMW project file in the loaded RMT file by overwriting it and exits.
Save As	Saves the RMW project file into a new RMT file with a new name specified and exits.
No Save	Exits RMW without saving the RMW project file.
Cancel	Cancel the termination process.

2.4.3 Loading Map File (Variable List)

Load a Map file that was generated in building a user program into RMW. By this operation, the variable list of global variables (variable information) is updated. If the user program has been changed and rebuilt, it is necessary to load the Map file again.

The procedure is described below.

2.4.3.1 Loading from File Menu's "Load Variable Data"

1. Select "Load Variable Data" from Main Window's File menu, and the screen to select a Map file will be displayed.
2. Select a Map file, and the "User Setting Form" screen will be displayed. (If "Data Type" is set in "Load Variable Data" from Option menu->Option Dialog.)
3. "User Setting Form" displays the variable list that reflects the settings.
4. After confirming the information in User Setting Form, click the "Set" button to load the variable information.

You cannot change "Data Type" of the variable list by using other RMW functions. To make a change, configure the variable information by using the Load Variable Data function again.

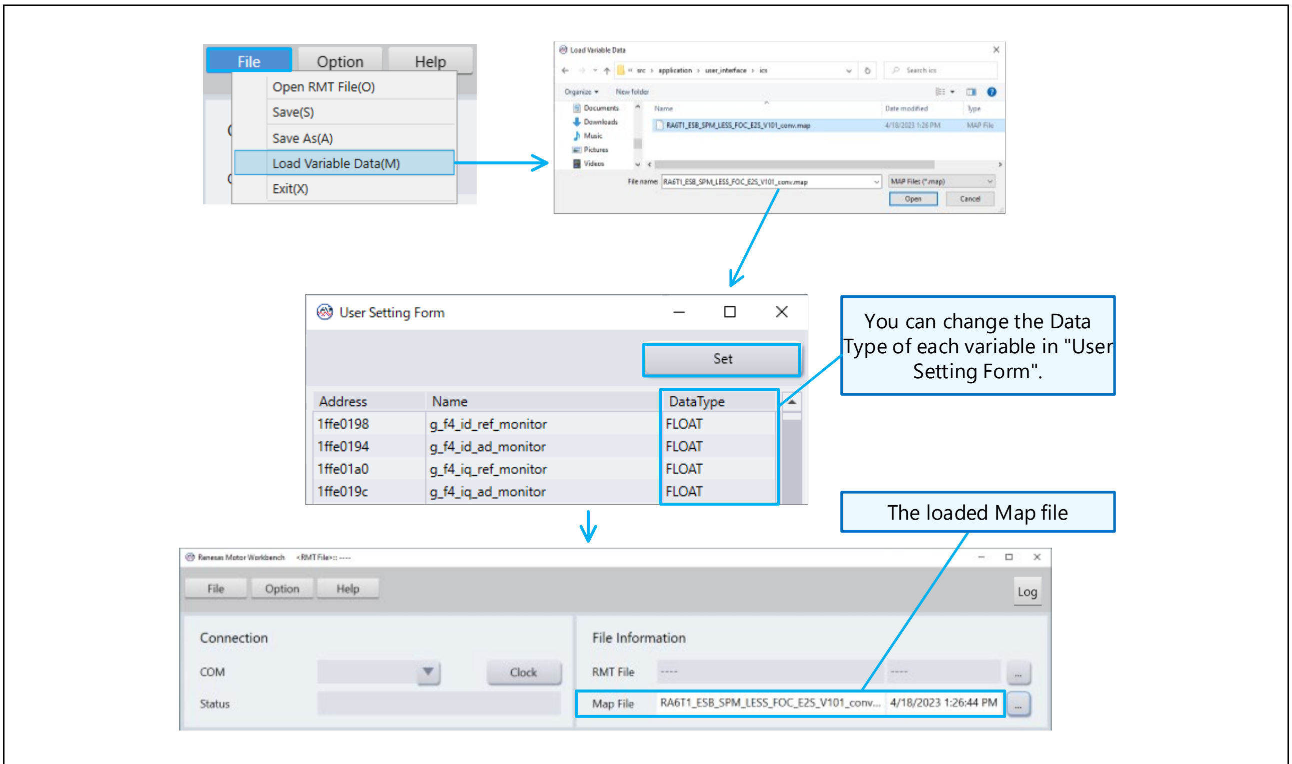


Figure 2-15 Loading Variable List (Map File)

2.4.3.2 Loading with Select Navigation

A Map file can also be loaded by using the Select Navigation function. For Select Navigation, see Section2.5.

2.4.4 Option Dialog from Option Menu

By selecting “Option Dialog” from Option menu, you can configure the settings for loading Map files and for preventing switch of the Analyzer function.

To use RMW with the same settings next time, you must save the RMT file.

Table 2-12 List of Option Dialog Menu

Tab	Function	Explanation
Main Window	Load Variable Data	• Specify a Data Type to be used when user program’s global variables are loaded from a Map file (when using the Load Variable Data function.) • Prefix Setting - Specify prefixes for variable names when you want to change a Data Type of variables. - You can specify up to four prefixes, separated by commas.
	Load Variable Meaning Data	• Specify the path of the CSV file that contains information about the variable meanings.
Analyzer	Control Window	• To ensure safety, you can prevent switching to another tool while a motor is being operated. - Variable Name: Specify the name of the variable to be referenced for confirming switching prevention. - Value: Specify the variable value to be referenced for confirming switching prevention (the specified value is the value used for prevention).

2.4.4.1 Load Variable Data Tab in Main Window Tab

When you use prefixes to identify variables with the same data type (UINT8, INT8, etc.), the data types specified here are set for all variables with the matching prefixes during loading of the variable list (Map file) (by executing the Load Variable function.)

For variables to be treated as an array, specify the data type in “Array of <Data Type>.” For non-array variables, specify the data type in “<Data Type>.” Up to four comma-separated values can be specified to each Data Type.

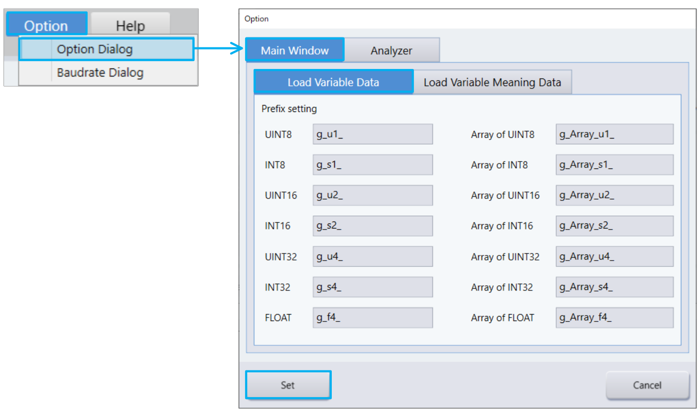


Figure 2-16 Load Variable Data Settings in Main Window Tab

2.4.4.2 Load Variable Meaning Data (Setting File for Variable Meaning Information)

You can load the information about the meanings (purposes) of variables in a CSV file format. The purpose and unit for each variable is displayed.

Create a CSV file as the following format:

<Variable name>,<Meaning>,<Unit>

The loaded information of the CSV file can be saved in a RMT file by the procedure described in Section 2.4.2 Saving RMT File. When a RMT file storing CSV file information is loaded, the saved variable meanings and units will be displayed.

Variable Name / Meaning / Unit

	A	B	C
1	com_u1_sw_userif	Switching of the user interface	
2	com_u1_system_mode	Managing the state	
3	com_f4_ref_speed_rpm	Speed command value (mechanical angle)	[rpm]
4	g_st_sensorless_vector.u2_error_status	Error status	
5	g_st_cc.f4_id_ref	d-axis current command value	[A]
6	g_st_cc.f4_id_ad	d-axis current detection value	[A]
7	g_st_cc.f4_iq_ref	q-axis current command value	[A]
8	g_st_cc.f4_iq_ad	q-axis current detection value	[A]
9	g_st_cc.f4_iu_ad	U-phase current detection value	[A]
10	g_st_cc.f4_iv_ad	V-phase current detection value	[A]
11	g_st_cc.f4_iw_ad	W-phase current detection value	[A]
12	g_st_cc.f4_vd_ref	d-axis voltage command value	[V]
13	g_st_cc.f4_vq_ref	q-axis voltage command value	[V]
14	g_st_cc.f4_refu	U-phase voltage command value	[V]
15	g_st_cc.f4_refv	V-phase voltage command value	[V]
16	g_st_cc.f4_refw	W-phase voltage command value	[V]
17	g_st_sc.f4_ref_speed_rad_ctrl	Speed command value (mechanical angle)	[rad/s]
18	g_st_sc.f4_speed_rad	Speed detection value	[rad/s]

Figure 2-17 Example of Creating Variable Meaning Information File (csv Format)

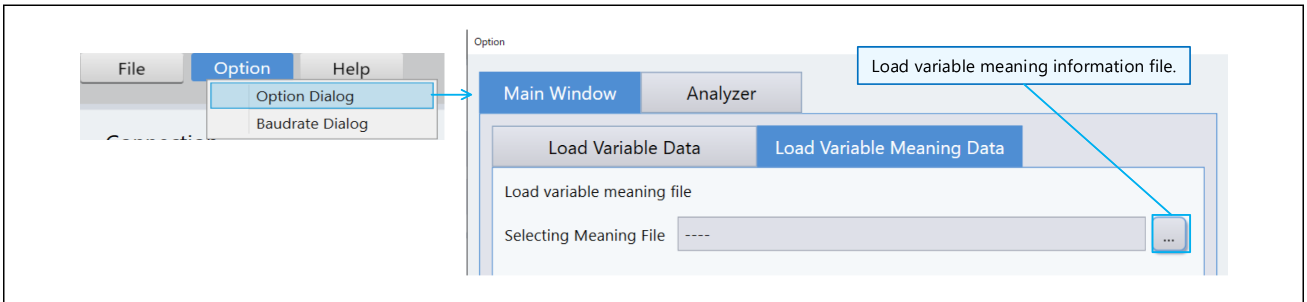


Figure 2-18 Example of Setting Load Variable Meaning Data

2.4.4.3 Control Window in Analyzer Tab

You can prevent the screen from switching to another tool while driving a motor by a user program.

Specify a variable name in the "Variable Name" field and a value that indicates "being operated (being processed)" in the "Value" field. Tool switching will be prevented when the value of the specified variable matches the value specified in the "Value" field. The message "Cannot changeover TAB of tools while Motor is in." is displayed while tool switching prevention is in effect.

The default value of "Variable Name" filed is blank. If this field is left blank, switching prevention will not be enabled. (RMW can be operated in this state.)

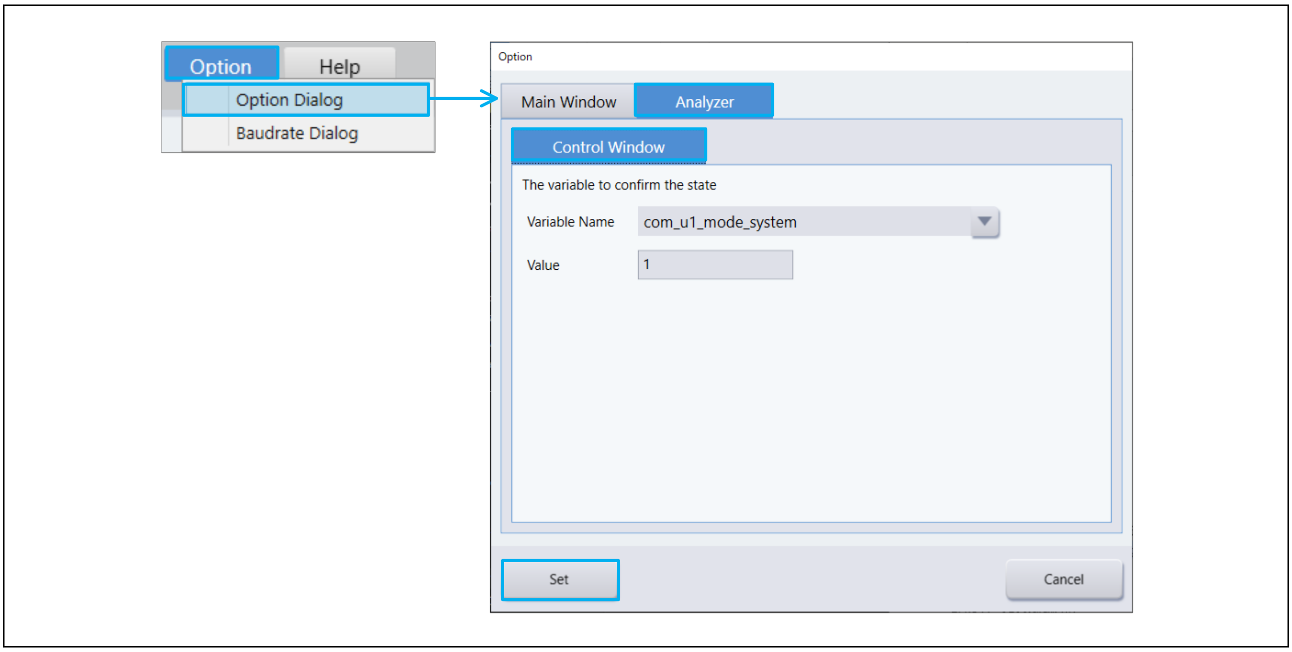


Figure 2-19 Control Window Settings In Analyzer Tab (Example of Setting "motor is being operated" As Judgement Condition)

2.4.5 USB Connection

2.4.5.1 Starting USB Communication

When expanding the COM information list in Main Window's "Connection", you can see the COM number or "Offline Mode" in a pull-down menu.

If you select "COM**" from the COM selection list, The COM connection process is started. The process status is displayed in "Status". When COM connection process is completed successfully, "Connect – USB Serial Port" is displayed.

In this state, tools are enabled.

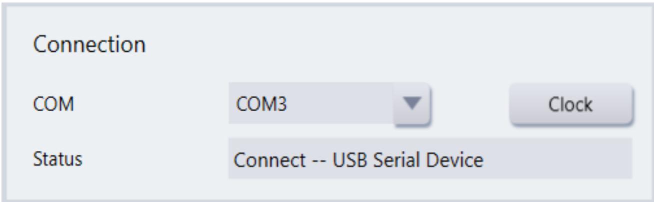


Figure 2-20 Status Displayed in Connection Screen When Connection is Established

Table 2-13 Main Window Display When Connection is Established

Name	Display
Connection Status	Displays "Connect."
Configuration	Displays the connection environment information. (This function is only for the execution file provided by Renesas.)
Select Tool	• Displays the icons for the available tools. • When "Offline Mode" is selected in the COM list, the available tools are limited.

If an error message is displayed when you select "COM**" from the COM selection list, the causes could be as follows; the USB connection may have a problem, the target board may not be turned on, or the communication clock setting may be wrong.

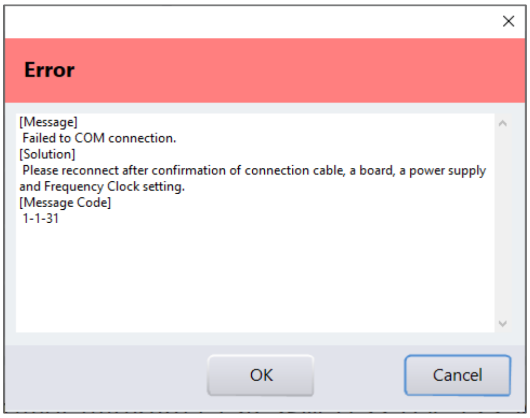


Figure 2-21 Error message under COM connection

2.4.5.2 In Case of USB Connection Trouble

If any problem occurs on USB connection, solve it by the following operations.

- Check the connection to the evaluation board and the power supply to the board.
- Confirm that the execution program of the evaluation board and the variable list loaded into Renesas Motor Workbench are both generated in the same build.
- Reset the evaluation board, reconnect the USB cable, or change the USB port to another one, etc.

2.4.5.3 Setting Clock Frequency (when using communication board for tools)

When using a communication board for tools (MC-COM by Renesas or W2002 by Desk Top Laboratories), specify the communication speed by the following operation.

Click "Clock" button in Main Window's "Connection" to display the Clock Setting dialog. The Clock Setting dialog displays the clock frequency that is currently set, so you can change this value. The changed clock frequency will take effect at the time of COM connection.

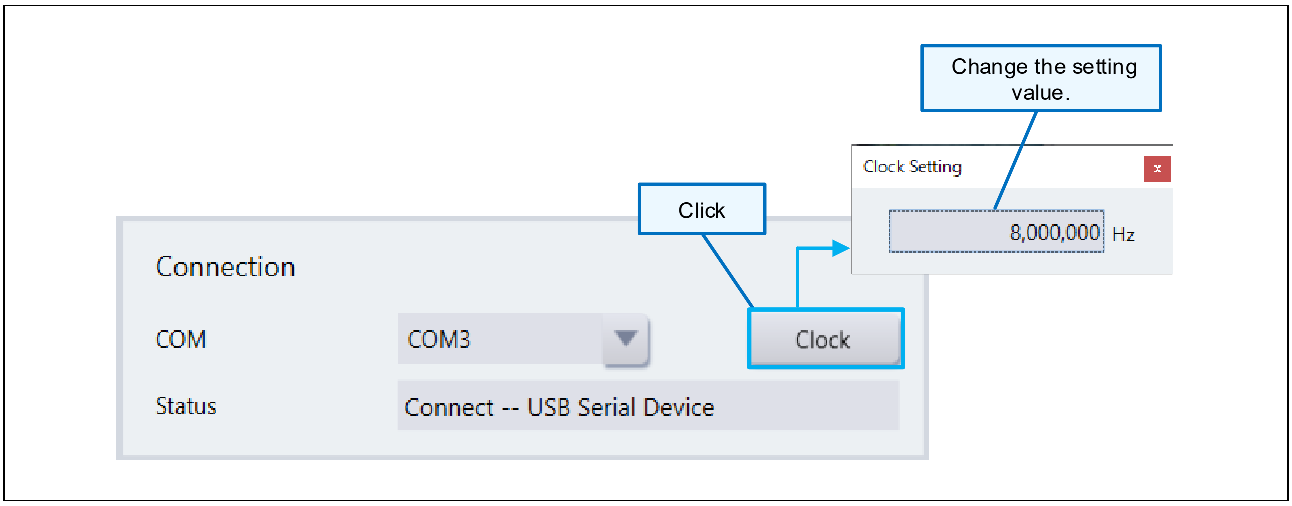


Figure 2-22 Clock Setting Dialog

To set the clock frequency, obtain the value by multiplying the communication rate by 8.

Example: When the communication rate is 1 Mbps, set the clock frequency to 8 MHz (8,000,000 Hz).
When the communication rate is 5 Mbps, set the clock frequency to 40 MHz (40,000,000 Hz).

When using a MC-COM (Renesas communication board for tools), you can select the following communication rates (clock frequencies). Set it based on the jumper (JP2) of the MC-COM.

Table 2-14 Settings of MC-COM JP2 and Selectable Clock Frequencies

JP2	Selectable clock frequency
Short	1 Mbps (8 MHz), 5 Mbps (40 MHz), 7.5 Mbps (60 MHz), 10 Mbps (80 MHz), 15 Mbps (120 MHz)
Open	6.25 Mbps (50 MHz), 8.33 Mbps (66666666 Hz), 12.5 Mbps (10 MHz), 16.66 Mbps (133333333 Hz)

2.4.5.4 Setting Baud Rate (when using built-in type communication library)

When using the built-in type communication library (using a commercially available USB serial conversion module), set the baud rate by the following operation.

Note: For the built-in type communication library, refer to 17 “Built-in Communication Library”.

Select "Baud rate Dialog" from the Main Window’s "Option" menu to display the Baud rate Setting dialog. The dialog displays the baud rate that is currently set, so you can change this value. The changed baud rate will take effect at the time of COM connection.

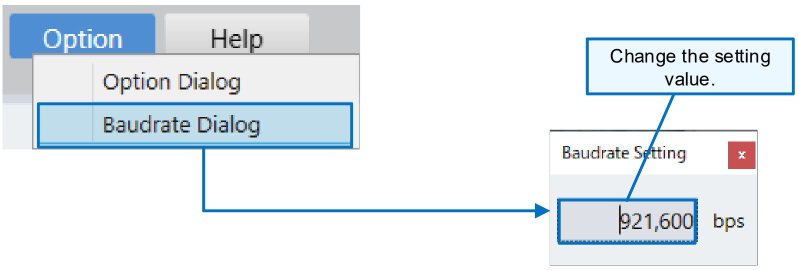


Figure 2-23 Baud Rate Setting

Set the baud rate to the value that is set by the program including the built-in type communication library. In addition, the set value must be a value that can be set from a PC to the connected USB serial conversion board. You can check the configurable values in the properties of the COM port (port setting) from Device Manager of Windows control panel.

2.4.6 Basic Window Operation

Main Window's "Select Tool" displays icons for the available tools. Click the icon to start the tool. The available tools vary depending on the user program and the status of COM connection.

The screen of each tool (Analyzer, Tuner, Easy) has a button to switch tools and a button to go back to Main Window on the top of the screen. However, if the condition matches the setting configured by Menu bar -> Option -> Option Dialog -> Analyzer tab -> "Control Window", tool switching is prevented while motor is being operated. (For details, see Section 2.4.4.)

2.4.6.1 Tool Switch Button

When a tool is started from Main Window, "Tool Switch button" is displayed on the top of the tool's screen. Click this button to switch to another tool.

Note that if you have set prevention of switching described in Section 2.4.4.2, you cannot switch tools while the state matches the preset condition.

2.4.6.2 Main Window Switch Button

When a tool is started from Main Window, the "Main Window" button is displayed on the upper right of the tool's screen. Click this button to go back to Main Window.

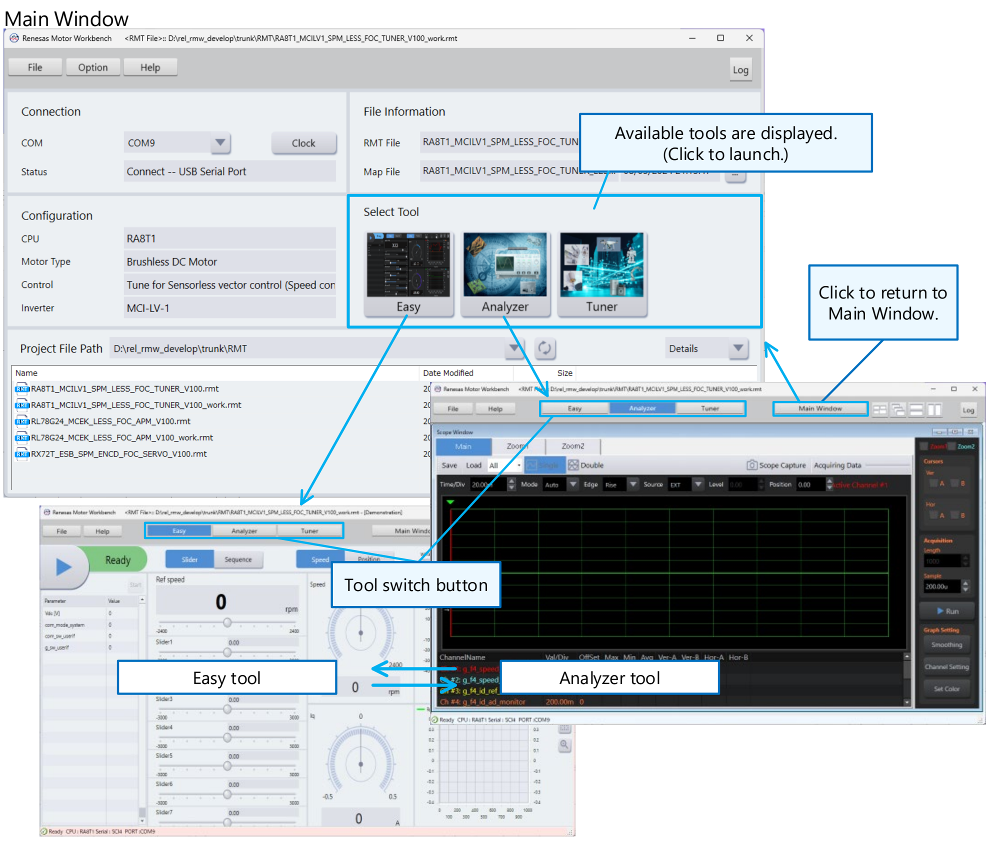


Figure 2-24 Tool Switch Button and Main Window Button

Note that you cannot switch to Main Window in the state like driving motor.

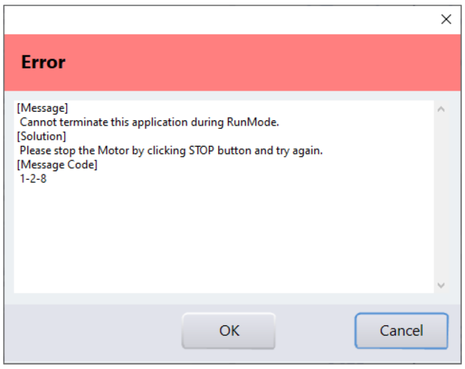


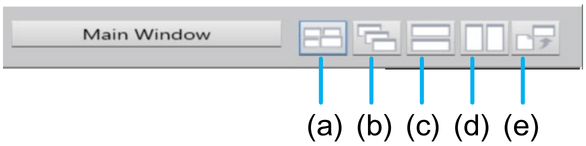
Figure 2-25 Error Message for Switching Tools During Processing

2.4.6.3 Window View Switch Button

In the screen of each tool (Analyzer, Servo, Tuner), you can change the window view by clicking the following buttons. You can also release the maximized window.

You can select the frontmost window with the window list button (button (a) in the figure below). However, when a window is framed out of the tool, the window is not displayed in the list of active windows. (In this case, select the window from Windows Task Bar.)

＜Window View Switch Button＞



(a) Display windows list
(b) Arrange windows in cascade
(c) Arrange windows up and down
(d) Arrange windows side by side
(e) Back to original size (displayed only for the maximized window)

＜Switching views by window list button (a)＞

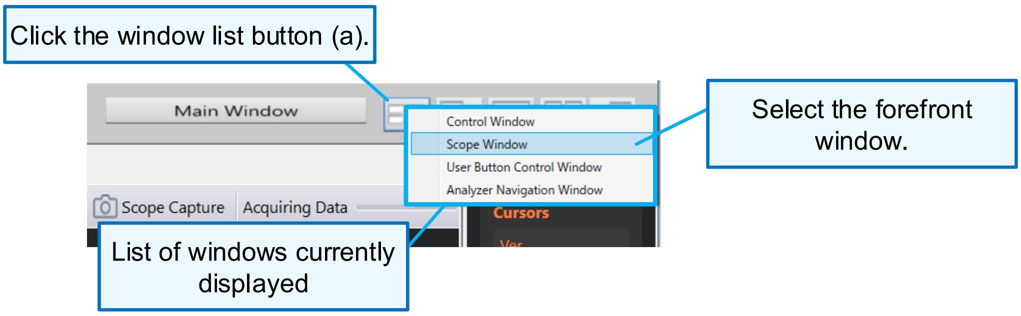


Figure 2-26 Window View Switch Button

2.4.6.4 Framing Out/In Window

You can move (frame out) tool windows (except for some tools) outside of the tool frame by dragging the title. In reverse, you can move (frame in) the window into the original tool frame by dragging it.

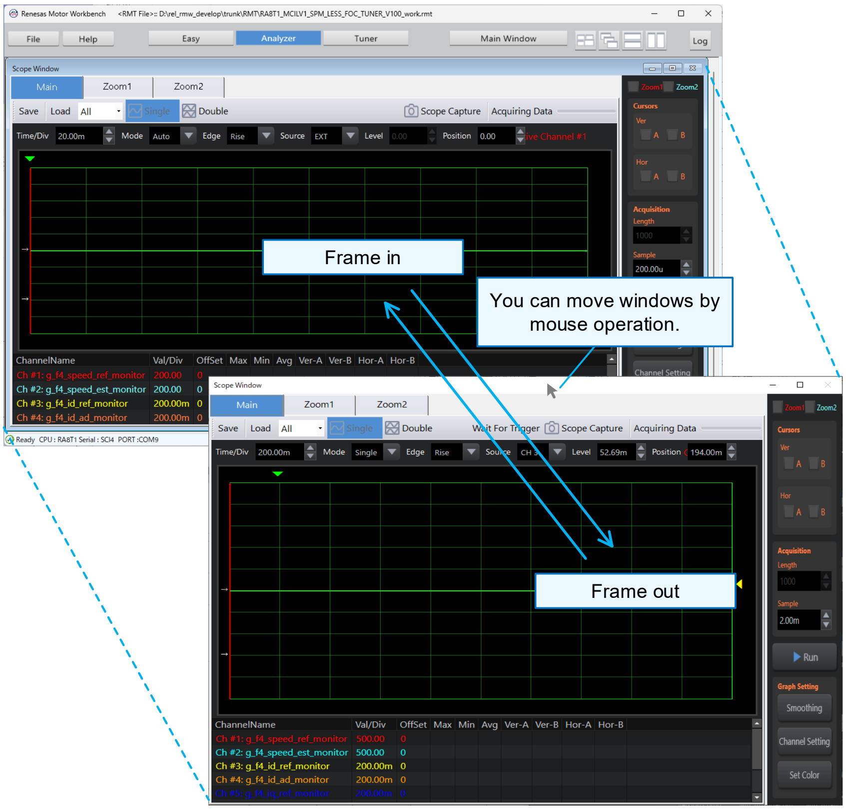


Figure 2-27 Window Frame Out/Frame In

2.4.7 Confirming Version

You can check the version of RMW from “Version Information” under the Help menu.

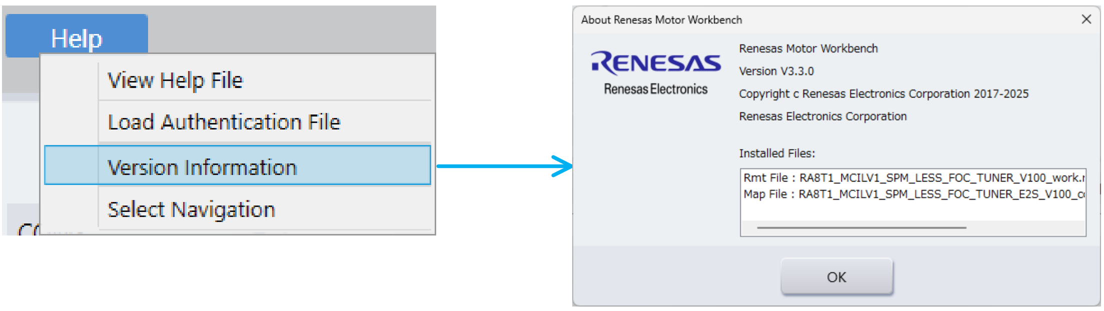


Figure 2-28 Displaying Version Information

2.5 Select Navigation Function

Select Navigation is a function that assists the basic settings for RMW, supporting such as loading RMT/Map files and setting a COM port in the dialog.

Select Navigation is displayed when RMW is started. (If you check “Do not show this features in the future”, it will not be displayed from the next time.) In addition, you can display it also by selecting “Select Navigation” from the Main Window’s Help menu.

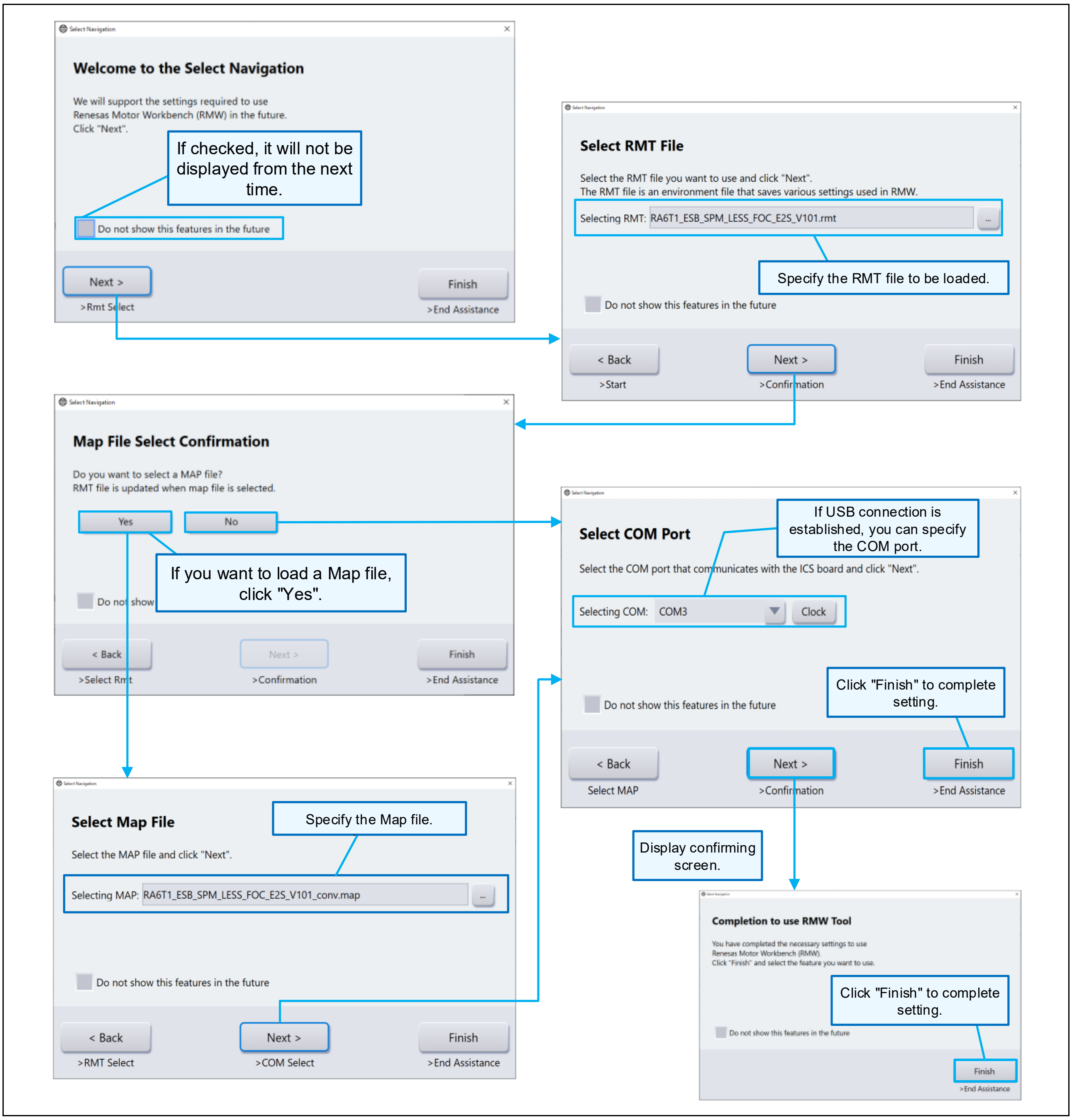


Figure 2-29 Select Navigation Function

3.Analyzer Tool

3.1 Overview

Analyzer is a tool that can read/write variables and display waveforms in real-time without stopping the program while a motor is being operated.

In motor control, if program execution is stopped suddenly, a large current may flow, and the inverter board may be damaged depending on the state of PWM output. Therefore, the variables in the MCU cannot be checked by setting brake points like general applications.

Additionally, in motor control, particularly in vector control, the current values calculated by the program, such as current values of “d-axis” and “q-axis”, are used for control, so those values cannot be checked with an oscilloscope, etc. This tool enables users to check the current values directly, which is one of key features.

Furthermore, Analyzer tool has various functions such as triggering according to the sequence, zooming the waveforms, etc.

By utilizing these functions of Analyzer tool, you can proceed your development far more efficiently than by outputting data with DA converters or external paths, or by saving them in a memory to analyze later.

3.2 Analyzer Tool Structure

Analyzer tool has three basic functions (windows) such as:

- Control Window
- Scope Window
- User Button Window
- Analyzer Navigation Window

and three sub tools (windows) that can be started from the above Control Window.

- Commander
- Status Indicator
- One Shot
- Parameter Output

The basic functions of Analyzer tool are displayed when you start Analyzer tool. While the Analyzer tool is active, you cannot close the window for the basic functions.

Every window in Analyzer tool can be framed out of the Analyzer tool window. (For detail, refer to Section 2.4.6 Basic Window Operation.)

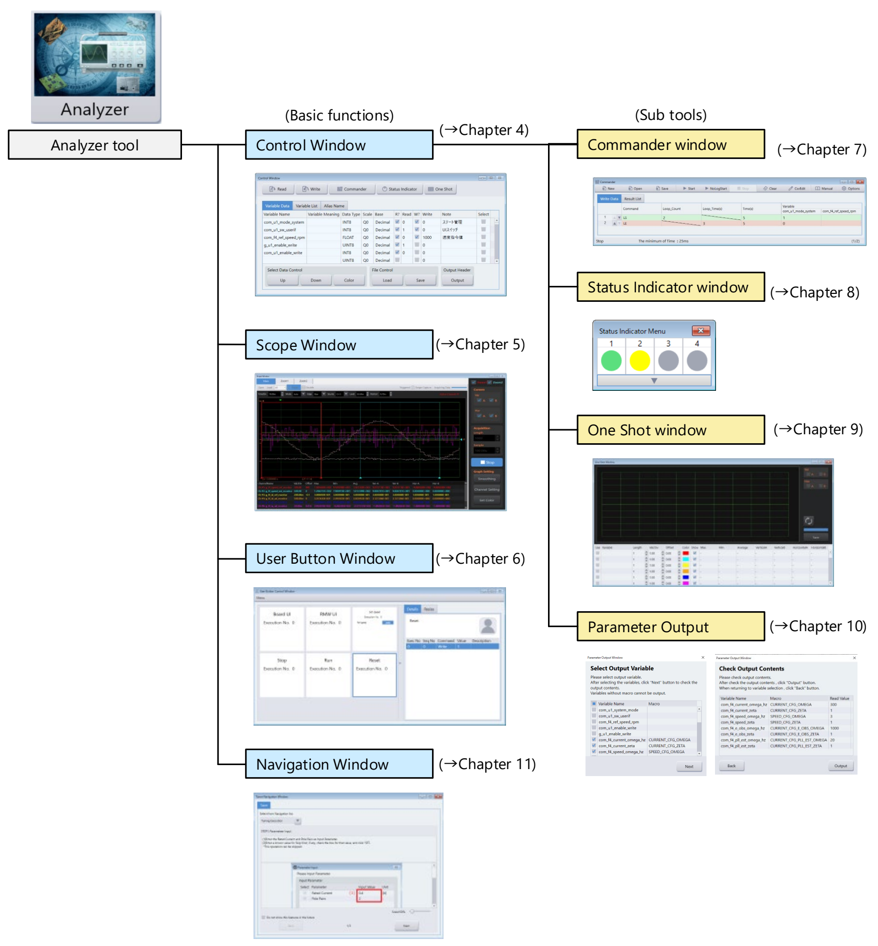


Figure 3-1 Analyzer Tool Structure

3.3 Analyzer tool View

The view of Analyzer tool is as below (the windows are arranged vertically).

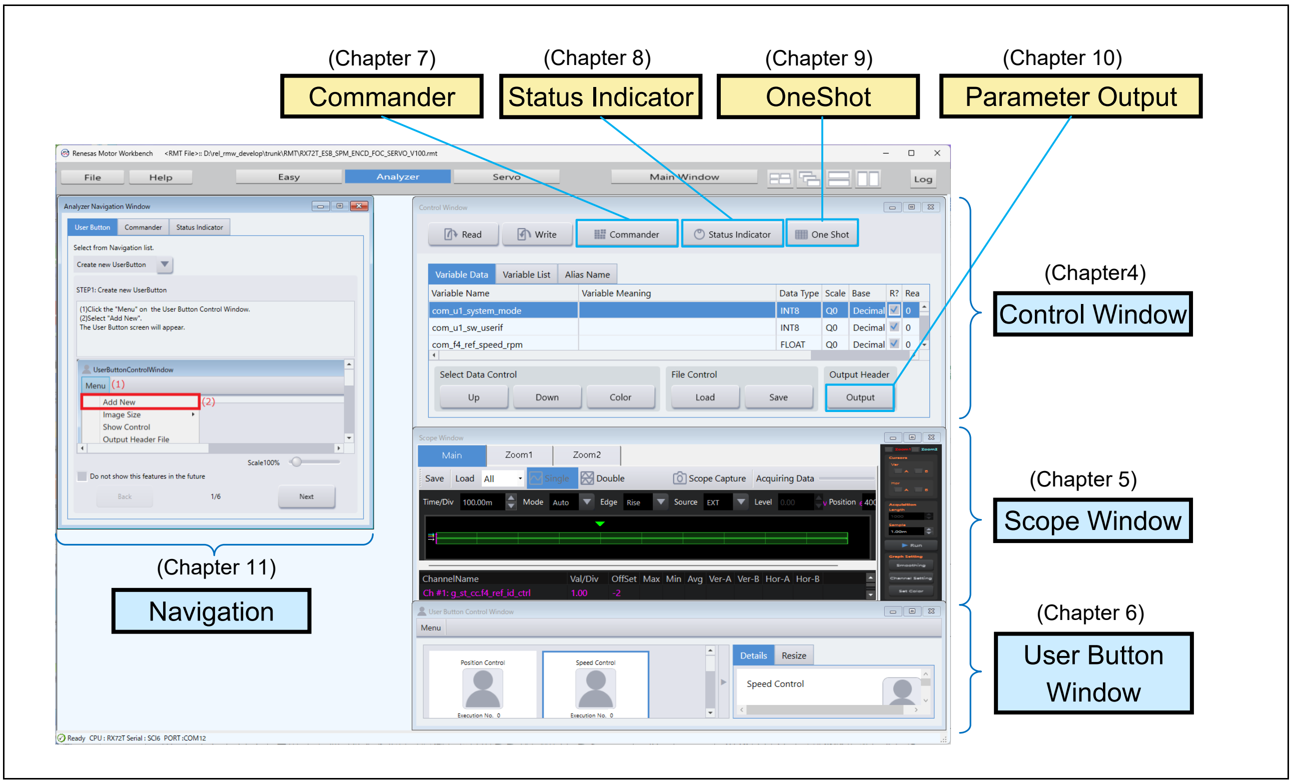


Figure 3-2 Analyzer Tool View

Table 3-1 Windows of Analyzer

Window		Explanation	Reference
Basic function	Control Window	Reads and writes variables in real-time. You can specify the target variable to operate from variables loaded from a Map file.	Chapter 4
	Scope Window	Provides waveform display of selected variables like an oscilloscope. It is possible to zoom and capture the display.	Chapter 5
	User Button Window	Executes the preregistered sequence sequentially by user instructions (clicking button).	Chapter 6
	Navigation Window	Explains how to operate each tool of Renesas Motor Workbench.	Chapter 11
Sub tool	Commander window	Preregisters a sequence of instructions for writing data into variables and executes it. It is possible to specify the intervals and perform loop processing.	Chapter 7
	Status Indicator window	Monitors variable values and turns on the indication light when a monitored result matches a preset condition (a threshold is exceeded).	Chapter 8
	One Shot window	Displays the buffered data (data of consecutive addresses from the specified variable) as a waveform.	Chapter 9
	Parameter Output	Outputs the parameters adjusted by Analyzer as a header file to be included in the motor control program.	Chapter 10

4. ［Analyzer］ Control Window

4.1 Overview

Control Window can read and write values of variables loaded from a Map file in real-time. Control Window is displayed when you launch Analyzer.

Control Window has three tabs such as Variable Data tab, Variable List tab, and Alias Name tab. You can also activate the sub windows (Commander window, Status Indicator window, and One Shot window) from the buttons on Control Window.

4.2 Features

- Reads and writes variables in real-time. Ideal for acquiring values instantaneously.
- Automatically displays a list of potential candidates when you enter characters to select the target variables.

4.3 Window Structure

The structure and functions of Control Window are shown below.

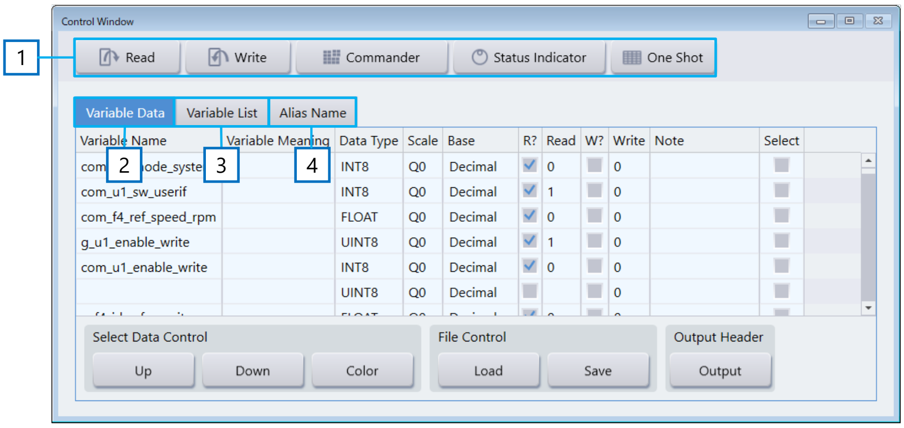


Figure 4-1 Control Window Structure

Table 4-1 Explanation of Each Part of Control Window

No.	Name	Explanation
1	Operation buttons	• You can load/write values of variables by clicking the Read/Write buttons. (You cannot when the target is not connected.) • Click Commander/Status Indicator/One Shot buttons to start each function.
2	Variable Data tab	You can specify the variable names from the “variable information” loaded from a Map file to load/write the values.
3	Variable List tab	Displays the list of “variable information” loaded from a Map file.
4	Alias Name tab	You can specify aliases for variable names in the “variable information” loaded from a Map file

4.3.1 Operation Buttons

The names and functions of operation buttons are explained below.

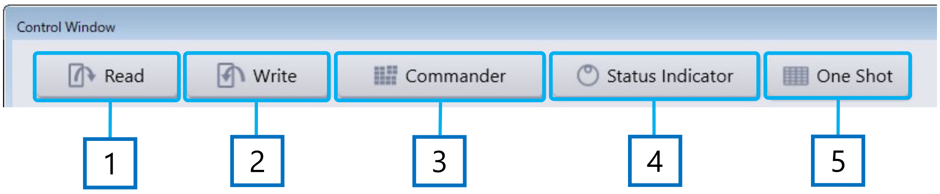


Figure 4-2 Operation Buttons

Table 4-2 Operation Button Function

No.	Name	Function
1	Read	Loads variable values.
2	Write	Writes values into variables.
3	Commander	Starts Commander window. (For details, see Section 7. ［Analyzer］ Commander Window.)
4	Status Indicator	Starts Status Indicator Menu window. (For detail, see Section 8. ［Analyzer］ Status Indicator Window.)
5	One Shot	Starts One Shot window. (For details, see Section 9. ［Analyzer］ One Shot Window.)

4.3.2 Variable Data Tab

You use Variable Data tab to load variable values and write values into variables by specifying the target variables.

The names and functions of items in Variable Data tab are shown below.

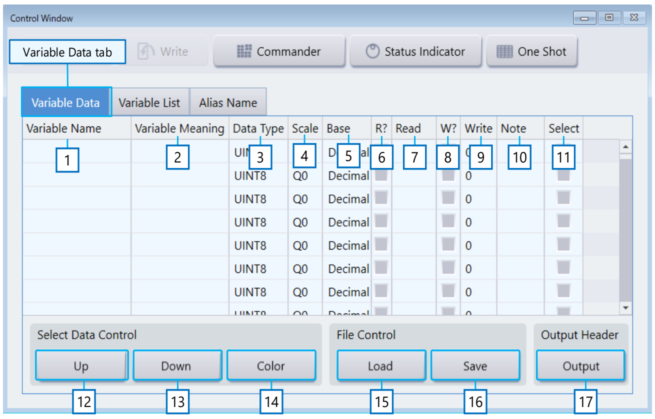


Figure 4-3 Variable Data Tab View

Table 4-3 Variable Data Tab Functions

No.	Name	Explanation
1	Variable Name	• Specify the name of a variable to be loaded or written. • You can specify a name by directly entering it, selecting the name from a list, or using the Variable Find function. (For details about the specification methods, see Section 4.4.1 Specifying Variable Name.)
2	Variable Meaning	• Displays the meaning and unit of the Variable Name
3	Data Type	• Displays the data type of the variable loaded from a Map file. • You can change the data type, but the change is effective only on this screen. (It is not reflected to the Variable List.)
4	Scale	Specify a scale value for the variable.
5	Base	Set the base number to be displayed.
6	R?	Specify whether the value is/isn't loaded when the Read button is clicked.
7	Read	Displays the loaded value.
8	W?	Specify whether the value is/isn't written when the Write button is clicked.
9	Write	Directly enter the value to be written.
10	Note	Any comments can be entered. (The entered information is saved in an RMT file.)
11	Select	By selecting boxes in this column, you can move rows or change background color in a batch operation. (You can select multiple items.)
12	Up button	Moves up the row(s) selected in Select by one row.
13	Down button	Moves down the row(s) selected in Select by one row.
14	Color button	Changes the background color of the row(s) selected in Select.
15	Load button	Loads setting information on variables to be read/written from a CSV file.
16	Save button	Saves information that has been set in this tab into a CSV file.
17	Output button	Activates Parameter Output window.

4.3.3 Variable List Tab

Variable List tab displays all variable information loaded from a Map file. In this tab, you cannot edit the information other than the "Description" field.

The names and functions of items in Variable List tab are shown below.

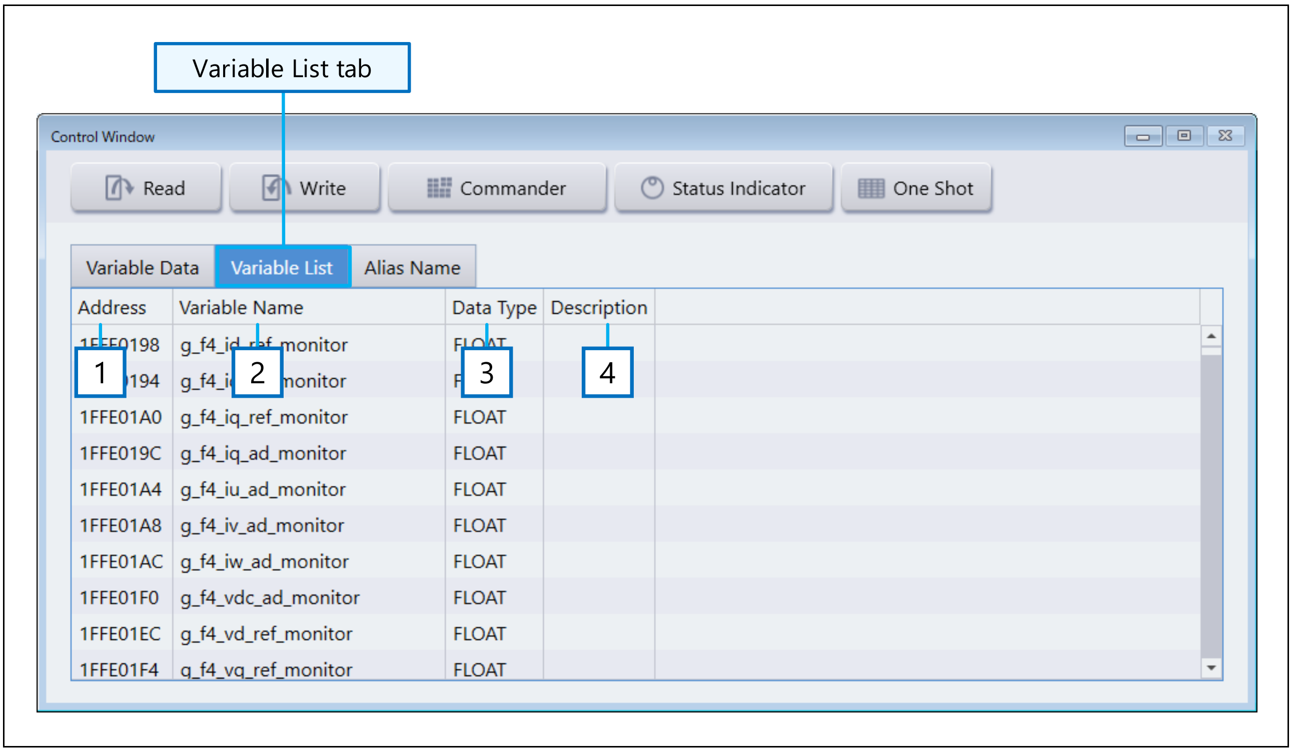


Figure 4-4 Variable List Tab View

Table 4-4 Variable List Tab Function

No.	Name	Explanation
1	Address	Displays the address of a variable (cannot be edited).
2	Variable Name	Displays a variable's name (cannot be edited).
3	Data Type	Displays a variable's data type (cannot be edited).
4	Description	Any comments can be entered. (The entered information is saved in an RMT file.)

4.3.4 Alias Name Tab

In Alias Name tab, you can specify an alias name for the variable loaded from a Map file. The specified name is used as Alias Name in the following windows.

- Channel Setting window in Scope Window
- User Button Window
- Commander window
- Variable Find window (searching variables)

The names and functions of items in Alias Name tab are shown below.

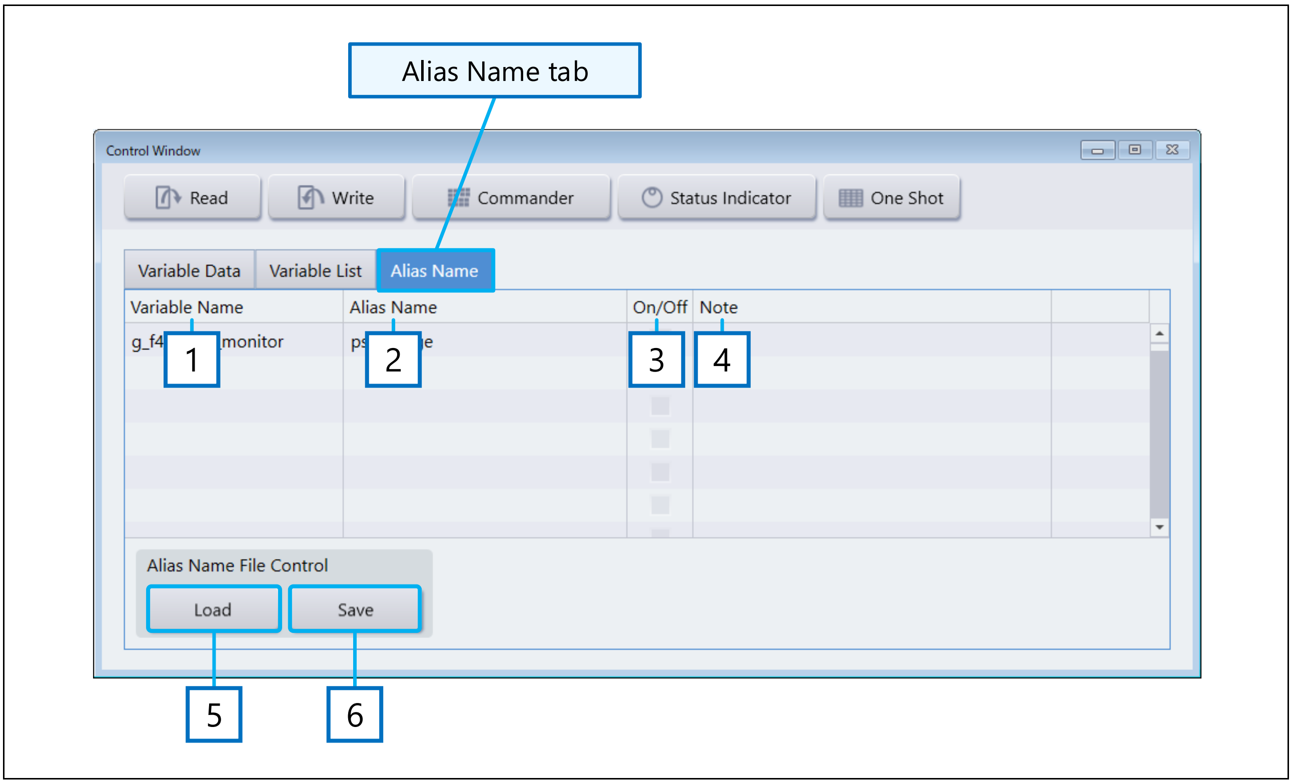


Figure 4-5 Alias Name Tab View

Table 4-5 Alias Name Tab Functions

No.	Name	Explanation
1	Variable Name	• Specify the variable name for which an alias is to be specified. • You can specify a variable name by directly entering it, selecting from a list, or using Variable Find. (For details, see Section 4.4.1 Specifying Variable Name.)
2	Alias Name	Specify the name to be used as the alias name.
3	On/Off	Enables/disables the alias name. (On: Setting enabled; Off: Setting disabled).
4	Note	Any comments can be entered. (The entered information is saved in an RMT file.)
5	Load	Loads alias name setting information from a CSV file.
6	Save	Saves the specified alias name information in a CSV file.

4.4 Specifying Variable Name (Variable Data Tab)

4.4.1 Specifying Variable Name

To load variable values or write values into variables, specify the target variable name in the Control Window's Variable Data tab. Specify a variable name by following one of the steps Section 4.4.1.1 to 4.4.1.3 below.

4.4.1.1 Enter Directly into the Variable Name Cell

Click a Variable Name cell in Variable Data tab to select it (the cell color is changed), then click it again to make it editable. You can enter a variable name directly in this state.

When you start entering characters, RMW automatically displays a list of potential matching candidates from the loaded variable information. You can select one of these variables.

Not that if you release the selected cell in which characters are entered halfway without specifying a variable, the entered information will be cleared.

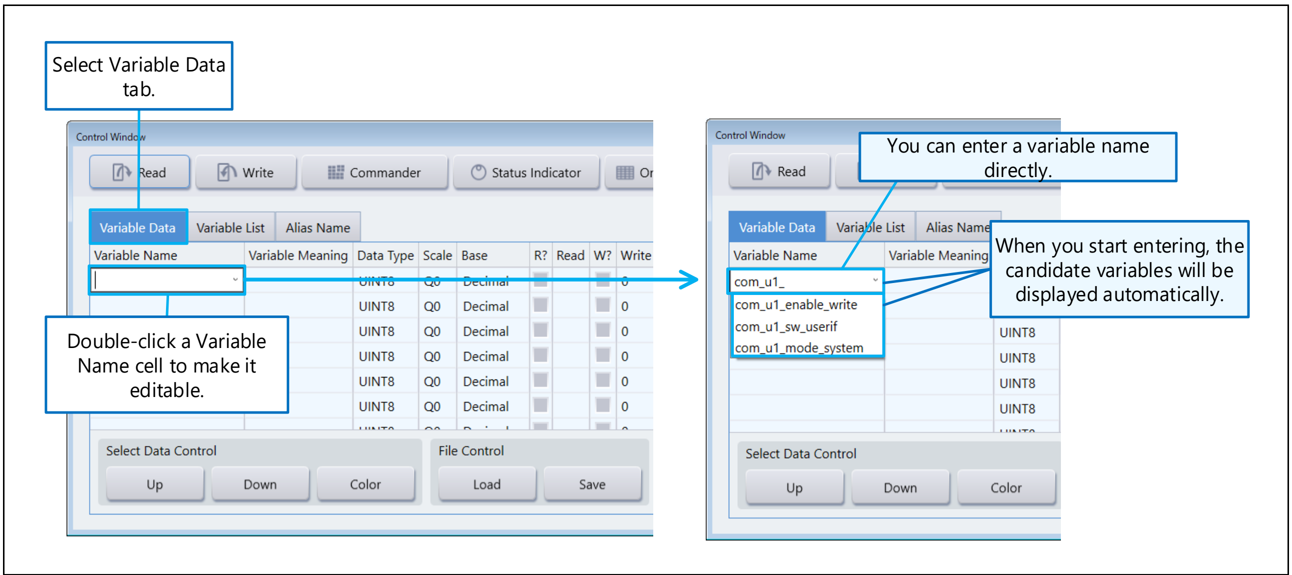


Figure 4-6 Direct Input to Cell

4.4.1.2 Select Variable From the List

Click a Variable Name cell in Variable Data tab to select it (the cell color is changed), then click it again to make it editable. In this state, "v" is displayed to the right of the cell. Click this "v" to display a list of the loaded variable information. Select a variable from this list.

If you enter a variable name partway and click the "v", a list of potential matching candidates is automatically displayed. You can select one of these variables. (the same operation as in Section 4.4.1.1)

Not that if you release the selected cell in which characters are entered halfway without specifying a variable, the entered information will be cleared.

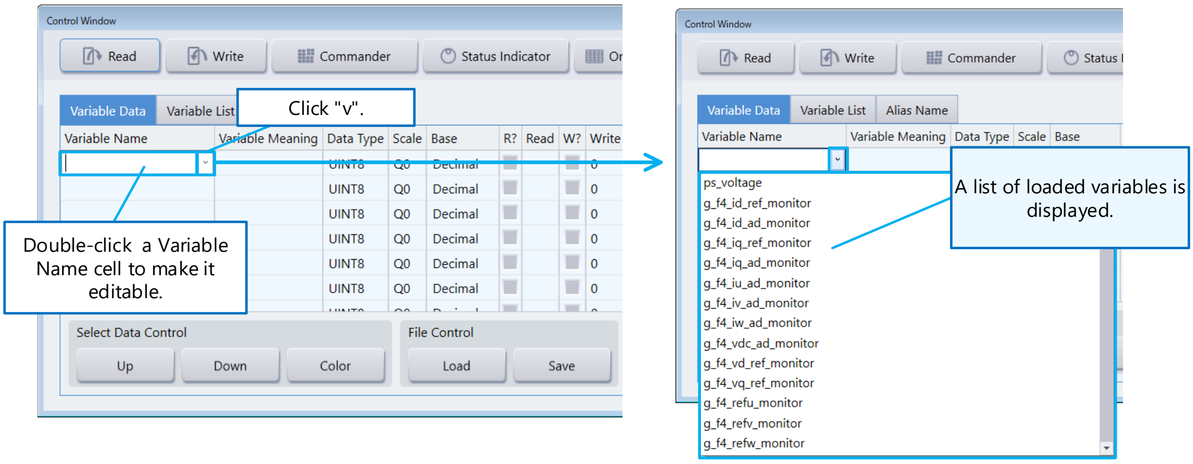


Figure 4-7 Selecting Variable from List

4.4.1.3 Select Variable Using Variable Find Function

Right-click on a Variable Name cell to open a menu. Select “Variable Find” from this menu, and the Variable Find screen will be displayed. On this screen, you can select a variable by narrowing down with keywords or meaning information.

Enter a keyword into the text input field on the Variable Find screen, and then click the “Find” button. You can enter multiple keywords separated with spaces for AND search. When checking the box at the bottom of Variable Find screen, you can also narrow down your search to only the variables whose meaning information has been loaded.

The following shows how to operate the Variable Find function.

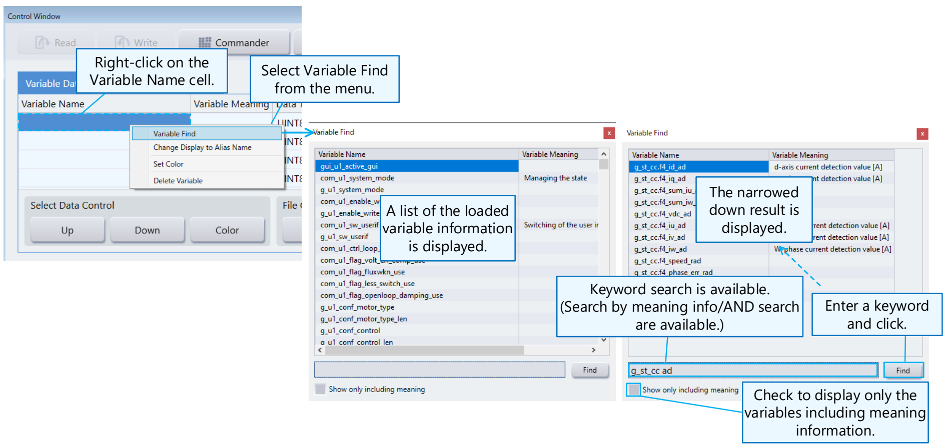


Figure 4-8 Variable Selection Using Variable Find Function

4.4.2 Setting Data Type

Control Window's Variable Data tab has a Data Type field, which automatically displays the data type of the variable information loaded from a Map file when the variable name is specified in Variable Name.

In a Data Type field, you can change the data type of the variable to use in Read/Write operations on Control Window. When you click the "v" to the right of a Data Type field, a selection list is displayed. You can change the data type by selecting one from the list.

Table 4-6 Data Type List

Data Type	Explanation
UINT8	8-bit, unsigned integer
INT8	8-bit, signed integer
UINT16	16-bit, unsigned integer
INT16	16-bit, signed integer
UINT32	32-bit, unsigned integer
INT32	32-bit, signed integer
FLOAT	32-bit, floating point
BOOL	Data type that takes either of two values (True or False)
LOGIC	Bit Field is displayed in bits

4.4.3 Setting Scale

Control Window's Variable Data tab has a Scale field, which automatically displays "Q0" as the default value when a variable name is specified in Variable Name.

In a Scale field, you can change the scale for the data to be used in Read/Write operations on Control Window. You can enter a value directly to this field. The valid values are Qn (n=0-31), positive integers, and positive decimals.

Table 4-7 Scale Input Values

Input value	Formula for calculating data
Qn (n=0-31)	Data × (1 ÷ (n-th power of 2))
Positive integer	Data × (1 ÷ integer)
Positive decimal	Data × (1 ÷ decimal)

If there is a mistake in the direct entry of a Scale field, an error notification is displayed.

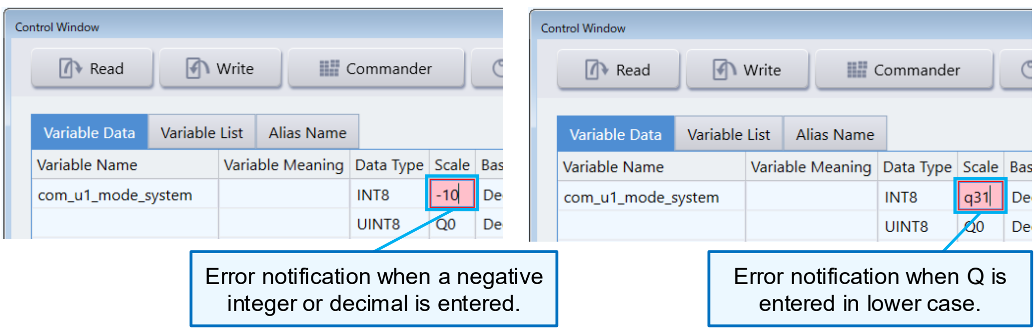


Figure 4-9 Error Notification in Scale Entry

4.4.4 Setting Base

Control Window's Variable Data tab has a Base field, which displays "Decimal" as the default value.

In a Base field, you can change the base for the data to be used in Read/Write operations on Control Window. Choose from Decimal, Binary, Octal, and Hex.

4.4.5 Text Notes for Variable Information

Control Window's Variable Data tab has a Note field, in which the user can enter any comments, such as a note on the variable information.

The information entered in the Note field is saved in an RMT file. (If you save RMT file and exit, the information will be displayed at the next startup.)

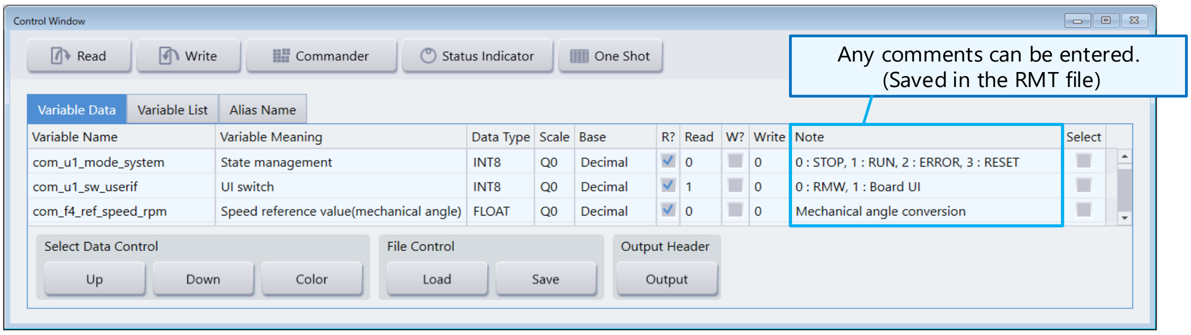


Figure 4-10 Note Field in Variable Data Tab

4.4.6 Loading Variable Values

Control Window's Variable Data tab is used to load variable values.

Check the checkbox in the “R?” field, then click the “Read” button to display the loaded value in the “Read” field. (If no target is connected, the “Read” button is disabled.)

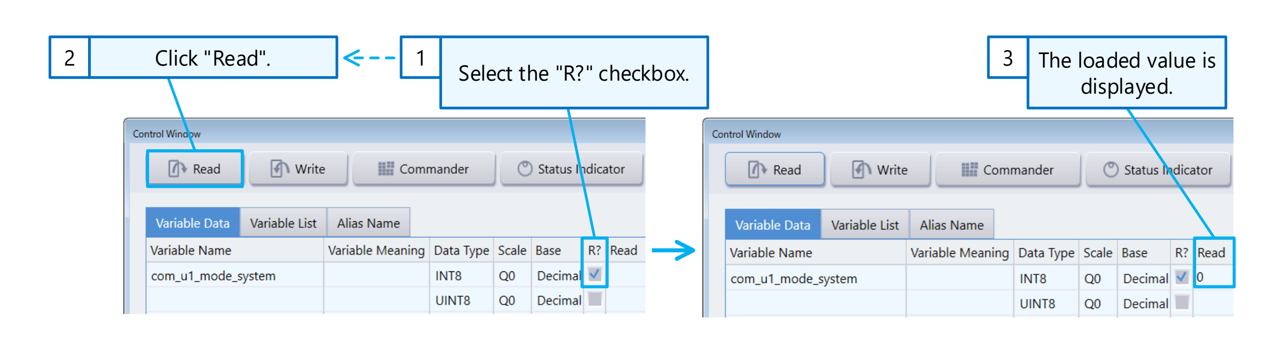


Figure 4-11 Loading Variable Values

You can switch ON/OFF of the checkboxes in the “R?” column all at once.

4.4.6.1 Select Multiple Cells to Switch

When multiple cells are selected (the cells turn blue), right-click and select “Select Set” or “Select Clear” from the menu, and you can switch the checkboxes of the selected cells all at once.

4.4.6.2 Switch All Cells

Select “All Set” or “All Clear” from the right-click menu of a “R?” cell, and you can switch the checkboxes of all the cells in the “R?” column at once.

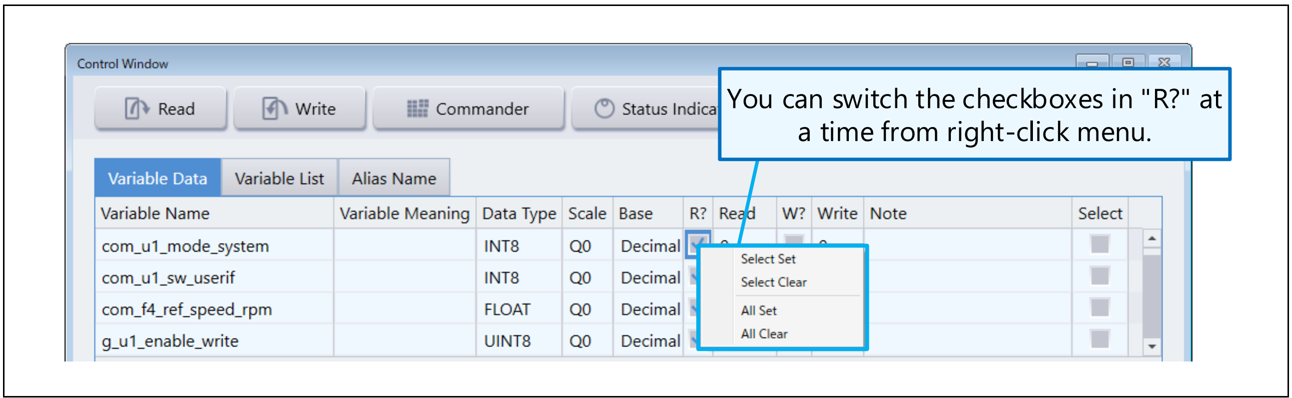


Figure 4-12 Switching ”R?” Checkboxes at Once

4.4.7 Writing Values into Variables

You can use Control Window's Variable Data tab to write values to variables.

Turn on the checkbox in the "W?" column, then enter a value to be written into the variable in the "Write" field. When you click the "Write" button, writing operation will be performed. (If no target is connected, Write button is disabled.)

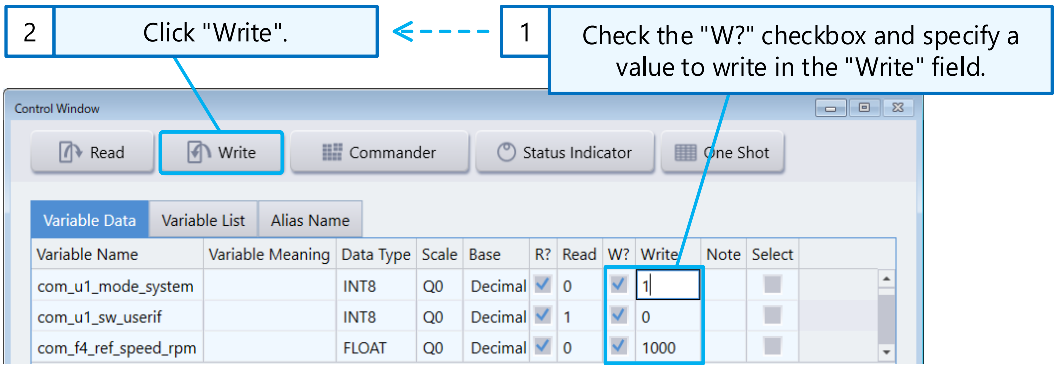


Figure 4-13 Writing Variable Values

You can check the written result using the operations described in Section 4.4.6 Loading Variable Values.

You can switch ON/OFF of the checkboxes in the "W?" column all at once.

4.4.7.1 Select Multiple Cells to Switch

When multiple cells are selected (the cells turn blue), right-click and select "Select Set" or "Select Clear" from the menu, and you can switch the checkboxes of the selected cells all at once.

4.4.7.2 Switch All Cells

Select "All Set" or "All Clear" from the right-click menu of the "W?" cell, and you can switch the checkboxes of all the cells in the "W?" column at once.

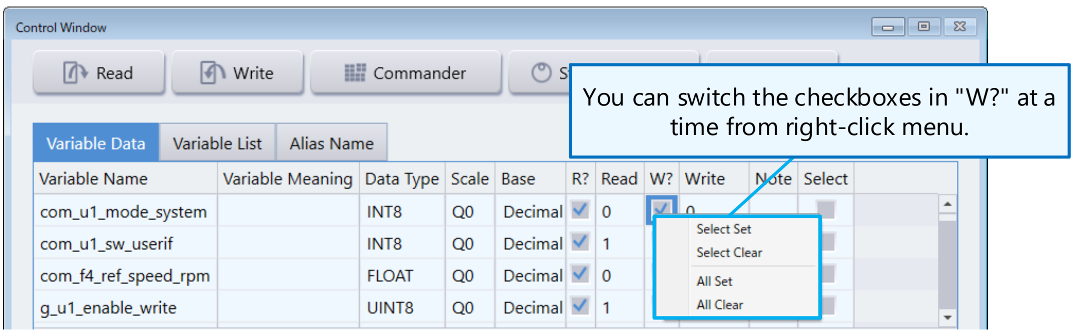


Figure 4-14 Switching "W?" Checkboxes at Once

4.4.8 Rearranging Variable Display

You can rearrange the order of the variable names set in the Control Window's Variable Data tab. Select a checkbox in the Select field for a row that you want to move, then click the “Up” or “Down” button at the bottom of the screen to move the row. You can also select multiple rows and move them at once.

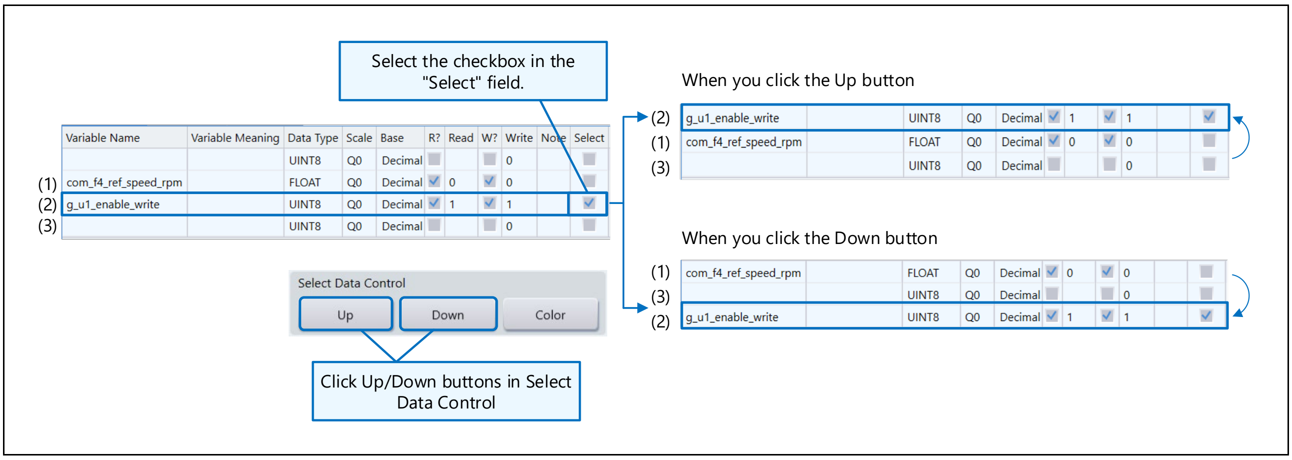


Figure 4-15 Rearranging Variable Display

4.4.9 Changing Background Color of Variable Display

You can change the background color of rows in Control Window’s Variable Data tab to make them stand out. There are two ways to change.

4.4.9.1 Change from the Right-Click Menu

Right-click on the row whose color you want to change, and select “Set Color” from the menu, and the color setting screen will be displayed. Select a color to change the background color.

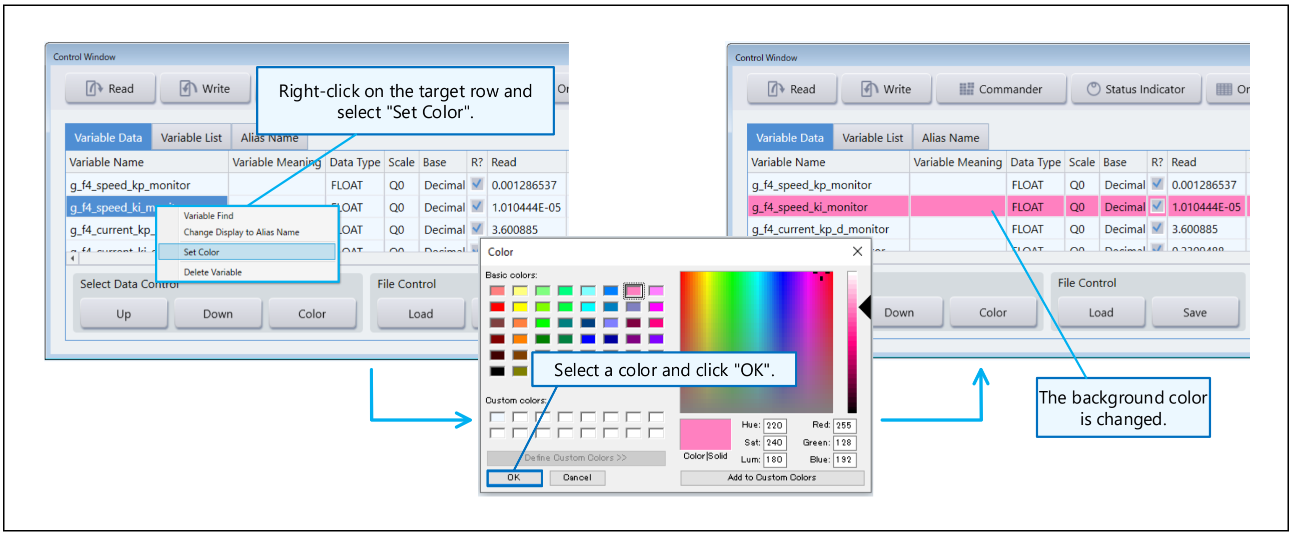


Figure 4-16 Changing Background Display Color of Variables (Right-Click Menu)

4.4.9.2 Select Rows by Using Select Field

Turn on the checkbox in the “Select” field of the row whose background color you want to change. Then click the “Color” button at the bottom of the screen, and the color setting screen will be displayed. Select a color to change the background color. You can also select multiple rows and change the color at once.

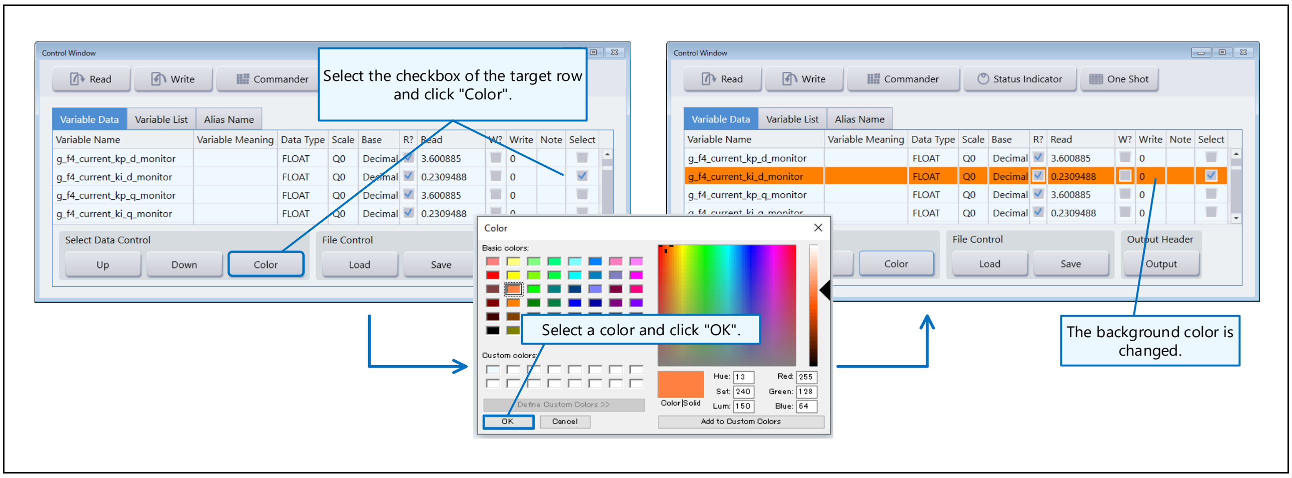


Figure 4-17 Changing Background Display Color of Variables (Selecting Rows by Using Select Field)

4.4.10 Deleting Variables

There are two ways to delete variable names set in the Control Window’s Variable Data tab. When a variable name is deleted, all the information on its row will be reset.

4.4.10.1 Delete by Delate Key

To delete a variable, select the variable name and press the Delete key.

4.4.10.2 Delete from Right-Click Menu

Right-click on the row of the variable name to be deleted (except on the R? /W?/Select fields), and you can delete it by selecting "Delete Variable" from the right-click menu.

If multiple variable names are selected, you can delete the multiple rows at once by the above operations Section 4.4.10.1 or 4.4.10.2. When a variable name is deleted, all the information set on its row will be reset to the initial state.

Note, however, that you cannot delete variables by selecting their checkboxes in the Select field.

4.4.11 Saving/Loading Variable Information

You can save and load the variable information set in Variable Data tab in CSV file format.

4.4.11.1 Saving Operation

Click the “Save” button on Variable Data tab to save the setting information into a CSV file.

4.4.11.2 Loading Operation

Click the “Load” button on Variable Data tab to load the setting information from the CSV file and update (overwrite) the list in Variable Data tab.

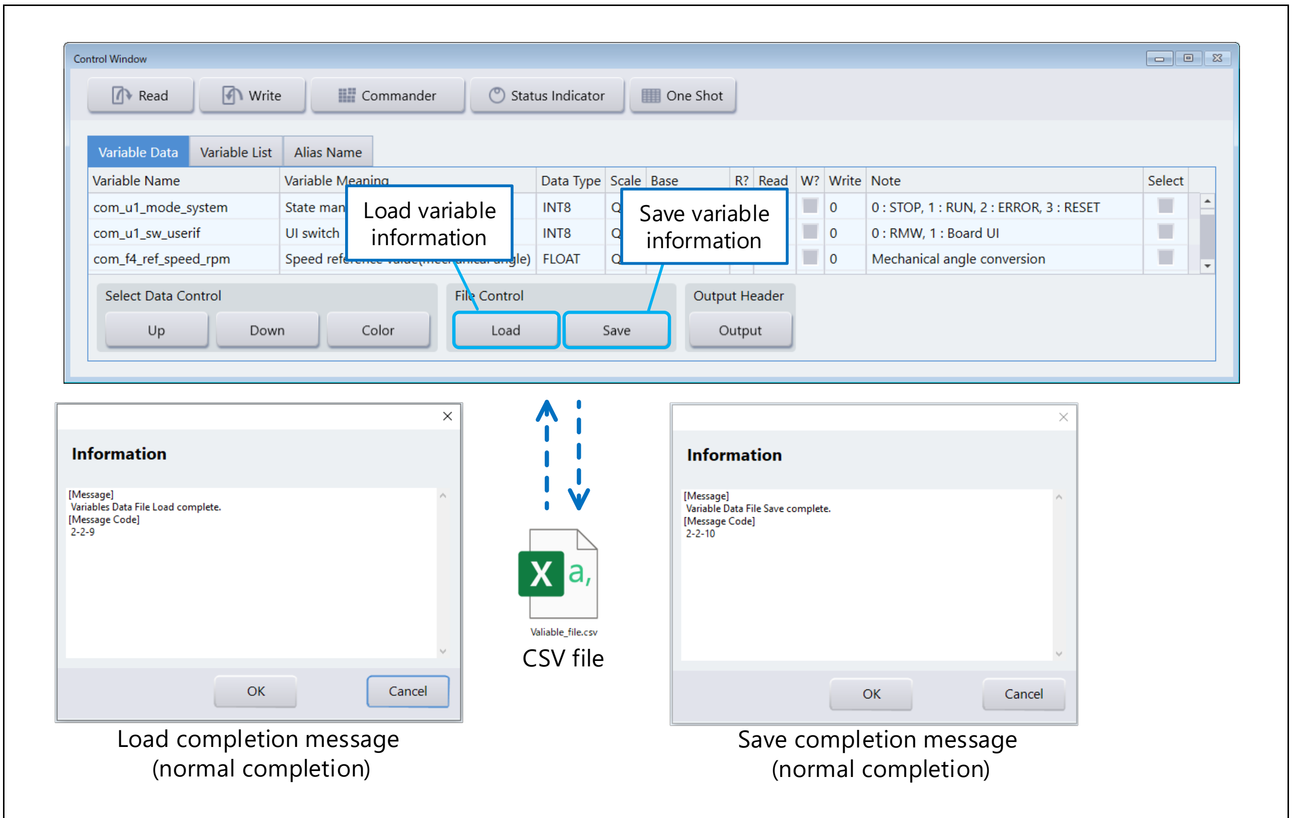


Figure 4-18 Saving/Loading Variable Information (CSV File)

Note that when you save an RMT file, all information set in each Control Window’s tab (Variable Data tab, Variable List tab, and Alias Name tab) is also saved. This information will be restored when the saved RMT file is loaded.

4.5 Listing Variables (Variable List Tab)

Variable List tab displays all the "variable information" loaded from a Map file. In this tab, you cannot edit the information other than the “Description” field.

For the screen structure, see Section 4.3.3.

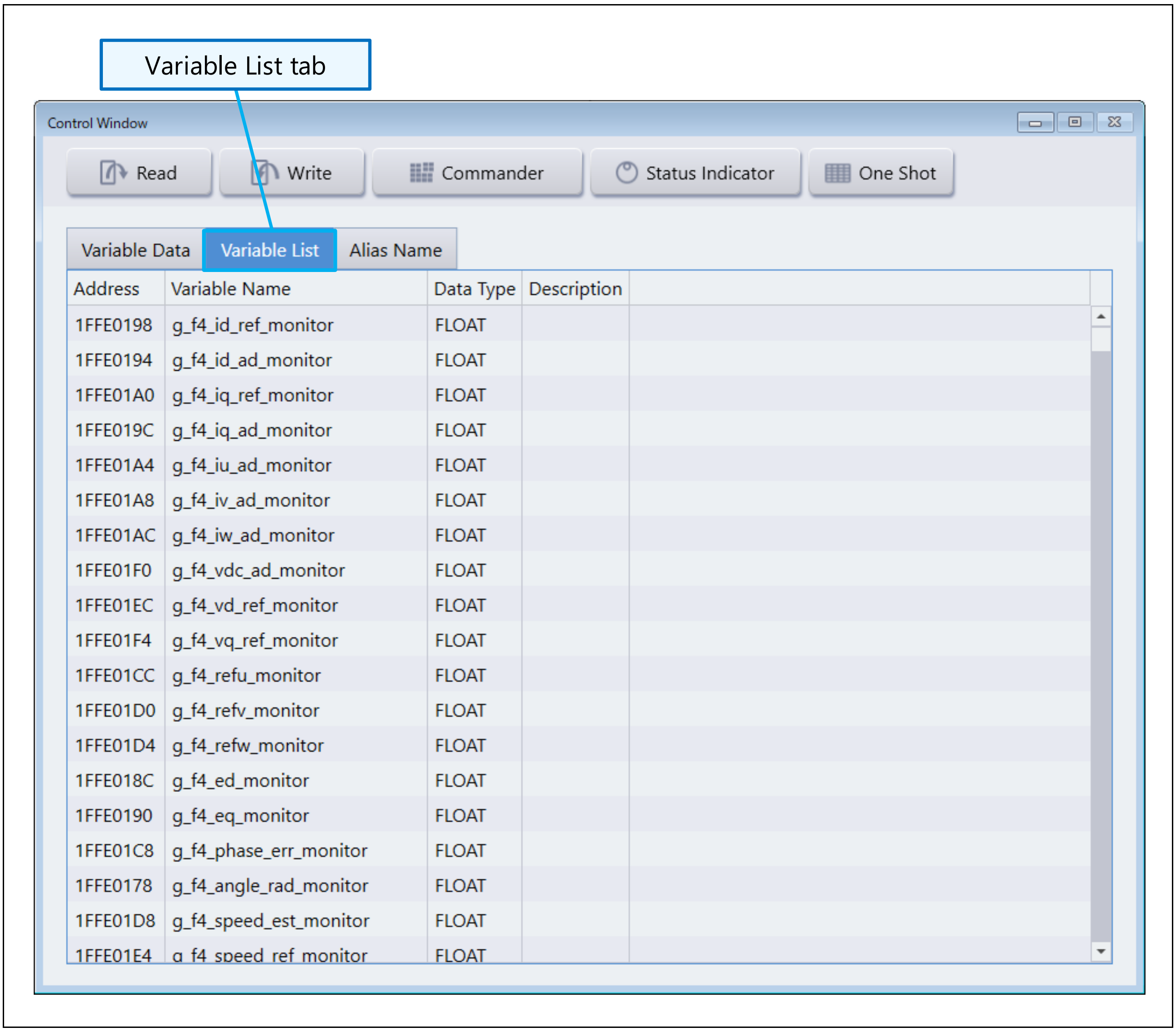


Figure 4-19 Variable List Tab View

4.6 Specifying Alias Name (Alias Name Tab)

In Control Window's Alias Name tab, you can specify an alias for the variable name. Aliases are displayed in the variable list in each function as well as regular variable names.

Alias Name can be displayed in Control Window's Variable Data tab, Variable Find screen for variable-search, Commander window, User Button window, and Channel Setting window in Scope Window.

4.6.1 Specify Target Variable

To define Alias Name, first specify the target variable in the Variable Name field of Alias Name tab.

You can enter a variable name in the same way as in Variable Name fields of Variable Data tab. (For details, see Section 4.4.1 Specifying Variable Name.)

4.6.2 Set Alias Name

Set an alias for the variable in the Alias Name field of Alias Name tab.

4.6.3 Enable or Disable Alias Name

You can enable or disable the alias name setting in the On/Off checkbox in Alias Name tab.

Check the checkbox to enable the alias, and uncheck to disable the alias. Though you can have duplicates variable names and alias names in the list, the only one of alias Name can be On.

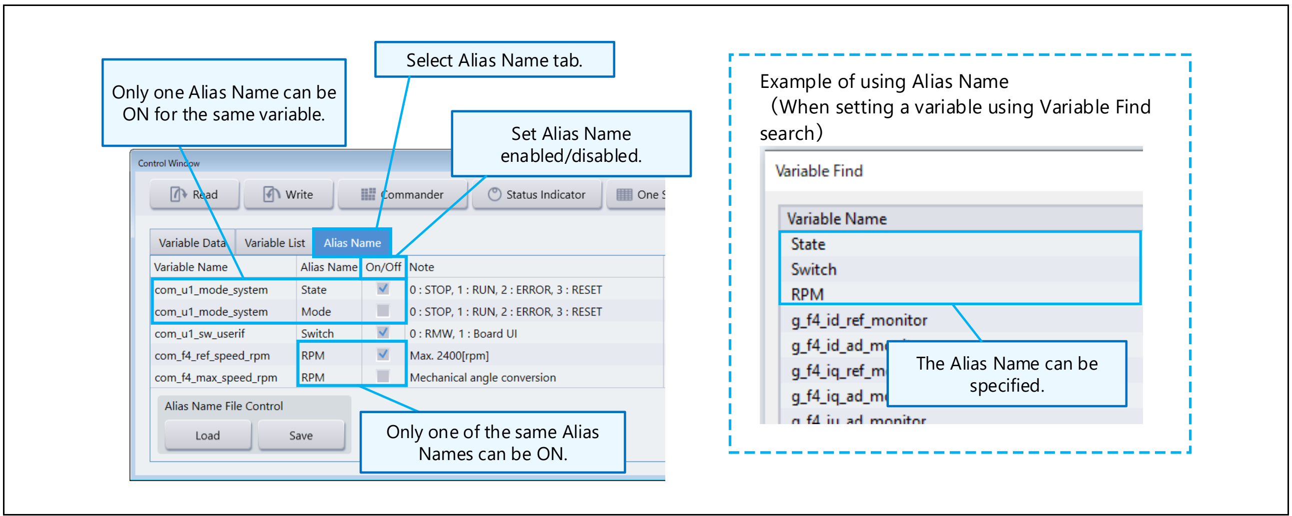


Figure 4-20 Enabling/Disabling Alias Name

Note: If you switch the Alias Name setting from "On" to "Off" while the other function has specified the alias name, the alias name will be switched automatically to the variable name. However, if you switch it from "Off" to "On", the variable name will not be switched automatically to the alias name. Therefore, you need to specify the alias name again from the function you use.

4.6.3.1 Set Note

The Note field on Alias Name tab can be used for entering any comments as a note.

4.6.3.2 Save/Load Alias Name List

You can save and load the setting information on Alias Name tab in a comma-separated CSV file format. When a CSV file is loaded, the information on Alias Name tab is overwritten.

When you save an RMT file, the information on Alias Name tab is also save.

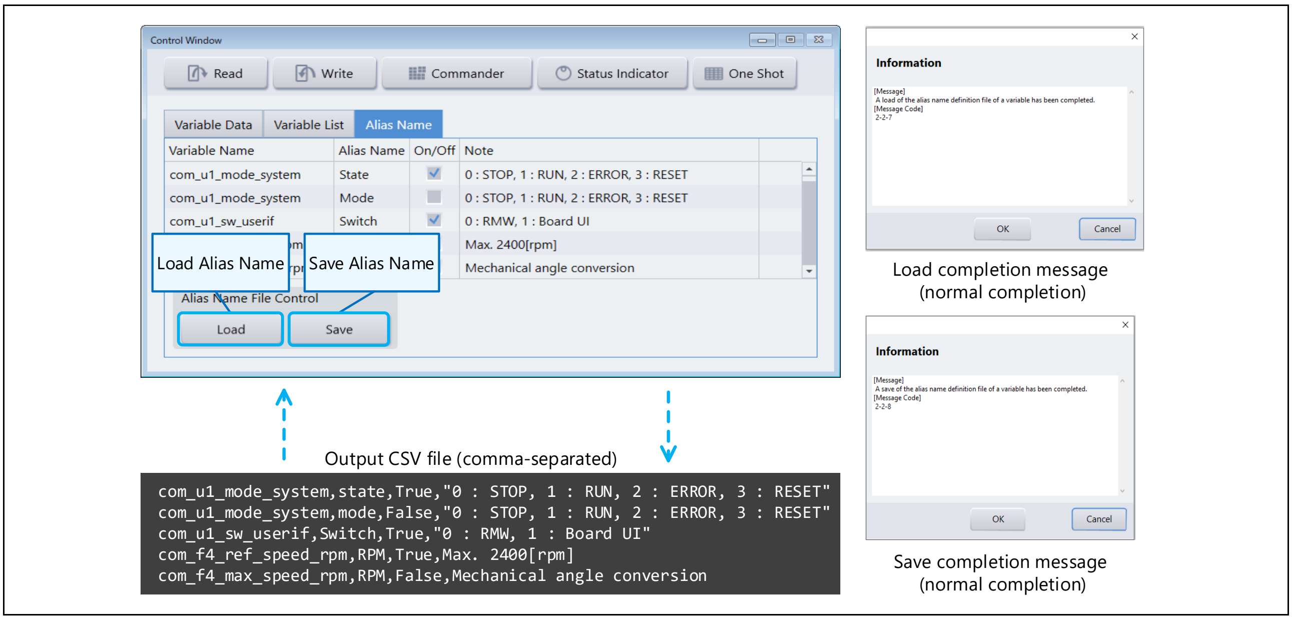


Figure 4-21 Saving/Loading Alias Name List (CSV File)

4.6.3.3 Switch Display Between Alias Name and Variable Name

Once you have set an alias name, you can use the alias name when specifying the variable name for other functions.

When you right-click a variable name on windows that support Alias Name display (except for Variable Find screen for variable-search), a menu is displayed such as “Change Display to Variable Name” and “Change Display to Alias Name”. You can switch displays by selecting them.

- Change Display to Variable Name :Switch display from Alias Name to Variable Name
- Change Display to Alias Name :Switch display from Variable Name to Alias Name

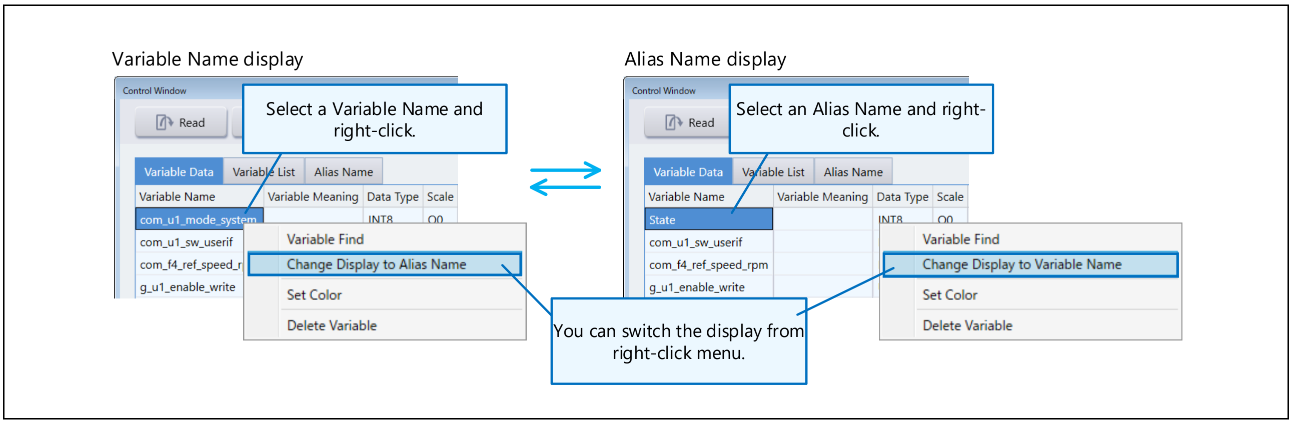


Figure 4-22 Switching Displays Between Alias Name and Variable Name

5. ［Analyzer］ Scope Window

5.1 Overview

Scope Window of Analyzer shows changes in the values of the selected variables as waveforms like an oscilloscope. Scope Window is displayed when you launch Analyzer.

5.2 Features

- Oscilloscope-like easy operations (trigger, offset, zoom, cursor, etc.)
- The sampling timing of each channel can be synchronized.
- The measured waveform information can be saved, loaded, and output in a report.

5.3 Window Structure

The window structure and functions of Scope Window are shown below.

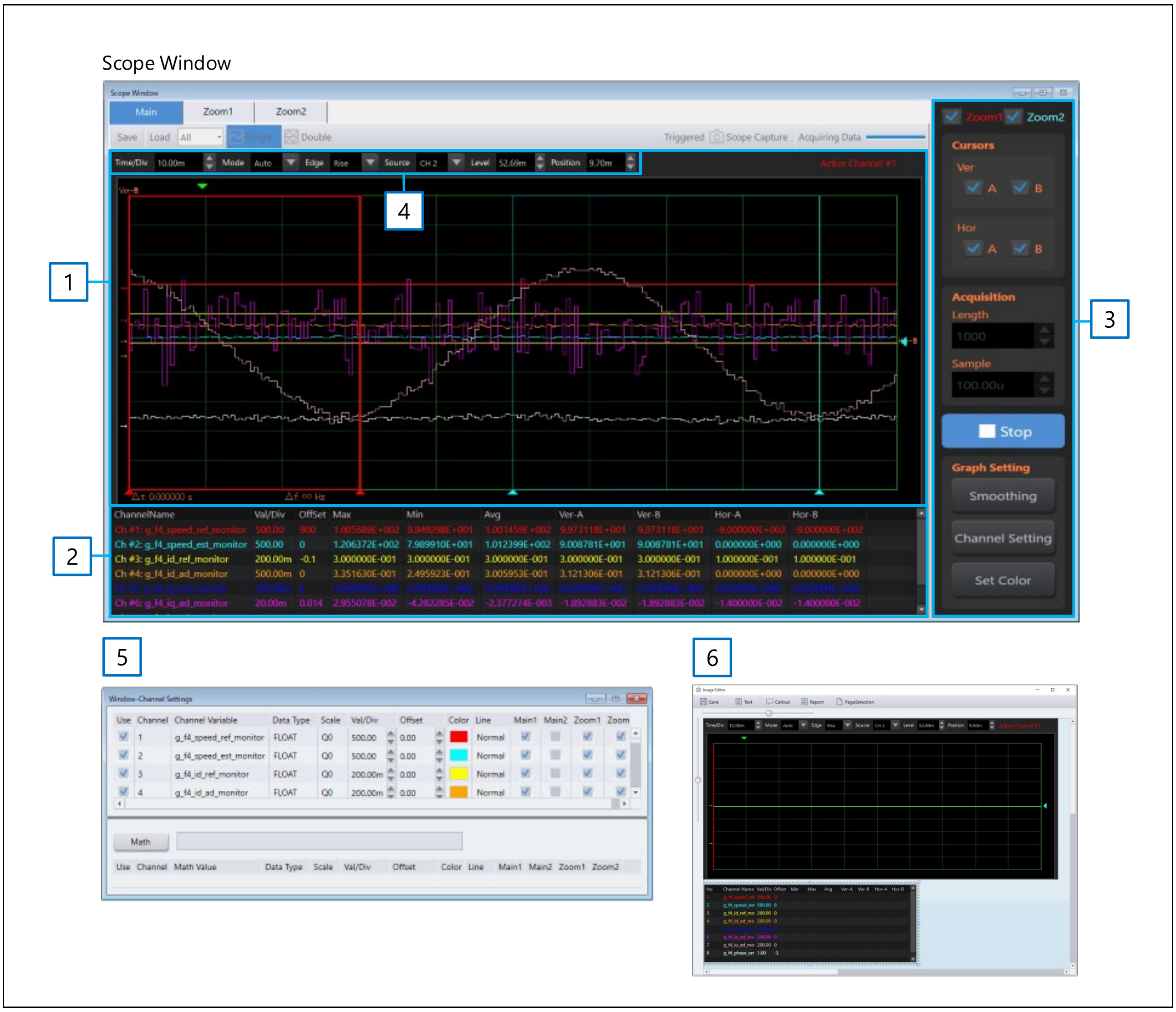


Figure 5-1 Scope Window Structure

Table 5-1 Explanation of Each Area of Scope Window

No.	Name	Explanation
1	Waveform display area	Displays variable vales as waveforms.
2	Channel information display area	Displays channel information.
3	Measurement settings area	Starts/stops waveform measurement and specifies various setting.
4	Trigger settings area	Specify the horizontal axis of waveform display area and settings related to trigger events.
5	Window-Channel Setting	Clicking “Channel Setting” button displays this screen. Register channels and configure detailed settings.
6	Image Editor screen	Clicking “Scope Capture” button displays this screen. Adjusts the waveform display image and outputs it in PDF or image format.

5.3.1 Waveform Display Area

The structure of the waveform display area is shown below.

5.3.1.1 Display Switch Tab

Figure 5-2 Display Switch Tab

Table 5-2 Explanation of Display Switch Tab

No.	Name	Explanation
1	Main tab	Displays the Main waveform.
2	Zoom1 tab / Zoom2 tab	Displays the Zoom waveform.

5.3.1.2 Tool Bar

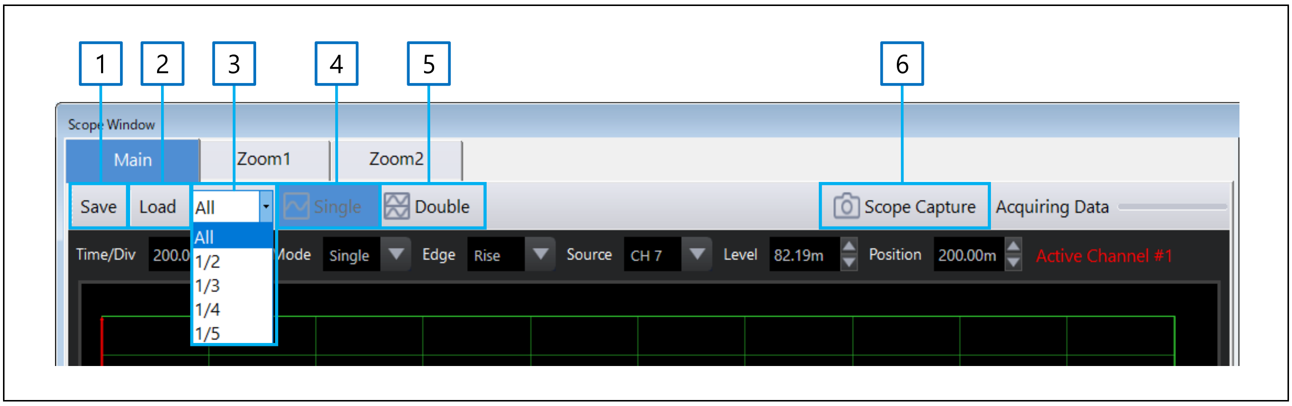


Figure 5-3 Tool Bar

Table 5-3 Explanation of Tool Bar.

No.	Name	Explanation
1	Save button	Saves the displayed waveform information in a CSV file.
2	Load button	Loads the CSV file that was saved with the Save button and displays its waveform.
3	Thinning count	Select the number of data to be thinned from the waveform data from a list box (All, 1/2, 1/3, 1/4, 1/5).
4	Single button	Displays the Main waveform in one screen.
5	Double button	Displays the Main waveform in double screens
6	Scope Capture button	Starts the Image Editor screen. The waveform image and the channel information are displayed in the Image Editor at the time of clicking this button. Another image and information are added each time the button is clicked.

5.3.1.3 Trigger Settings Area

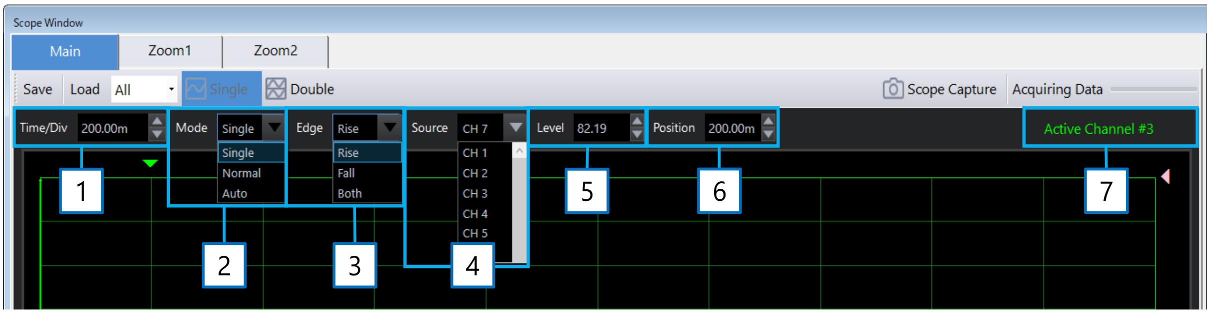


Figure 5-4 Trigger Settings Area

Table 5-4 Explanation of Trigger Settings Area

No.	Name	Explanation
1	Time / Div	Specify the time per division for the horizontal axis for the waveform display.
2	Mode	Select the trigger mode: - Auto : Automatically updates the waveform at a fixed interval - Single : Displays a waveform when a trigger event occurs and stops updating. - Normal : Repeats the waveform display every time a trigger event occurs.
3	Edge	Select the trigger edge: - Rise : Rising edge - Fall : Falling edge - Both : Both edges
4	Source	Select the channel number for which a trigger is to be set.
5	Level	Specify the trigger synchronization level.
6	Position	Specify the trigger display position.
7	Active Channel	Displays the channel number that is set as the Active Channel.

5.3.2 Channel Information Display Area

The structure of the channel information display area is shown below.

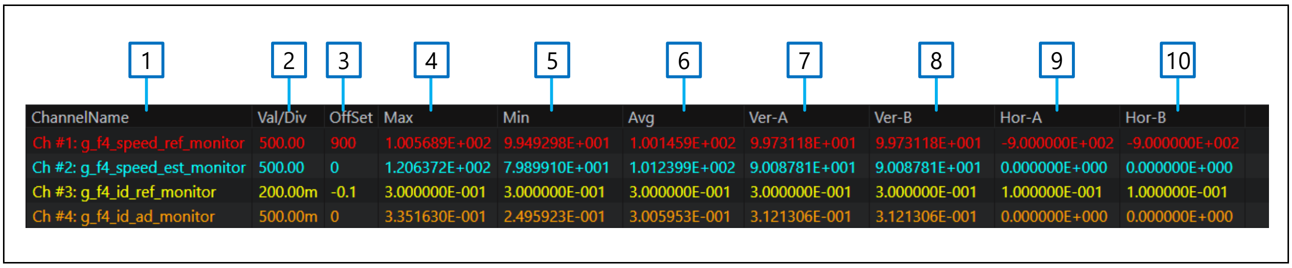


Figure 5-5 Channel Information Display Area

Table 5-5 Explanation of Each Column in the Channel Information Display area

No.	Name	Explanation
1	Ch #n	Displays a variable name for which channel setting was specified.
2	Val / Div	Displays the Val/Div value for which channel setting was specified.
3	Offset	Displays the offset value for which channel setting was specified.
4	Max	Displays the maximum value of the range for which the waveform is to be displayed.
5	Min	Displays the minimum value of the range for which the waveform is to be displayed.
6	Avg	Displays the average of the waveform display values.
7	Ver-A	Displays the value on the vertical axis that was set by Cursor (Ver-A).
8	Ver-B	Displays the value on the vertical axis that was set by Cursor (Ver-B).
9	Hor-A	Displays the value on the horizontal axis that was set by Cursor (Hor-A).
10	Hor-B	Displays the value on the horizontal axis that was set by Cursor (Hor-B).

5.3.3 Measurement Settings Area

The structure of the measurement settings area is shown below.

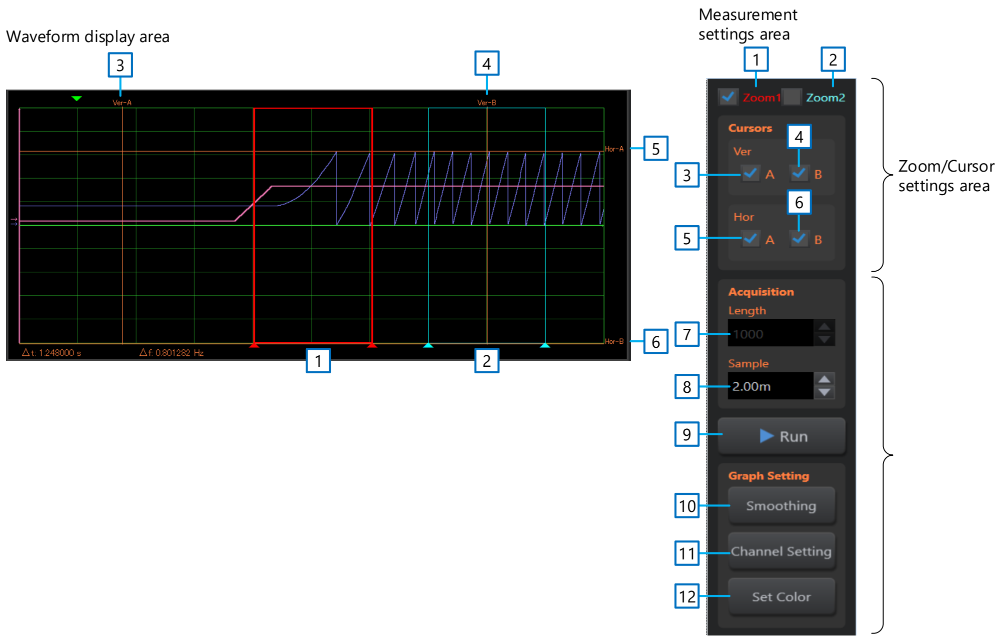


Figure 5-6 Measurement Settings Area

Table 5-6 Explanation of Measurement Settings Area

No.	Group	Name	Explanation
1	Zoom	Zoom1	Displays the range of a waveform to be displayed in Zoom1 in the Main waveform screen as a red frame.
2		Zoom2	Displays the range of a waveform to be displayed in Zoom2 in the Main waveform screen as a blue frame.
3	Cursors	Cursor Ver A	Displays Cursor (Ver-A) in each screen to display waveforms
4		Cursor Ver B	Displays Cursor (Ver-B) in each screen to display waveforms.
5		Cursor Hor A	Displays Cursor (Hor-A) in each screen to display waveforms.
6		Cursor Hor B	Displays Cursor (Hor-B) in each screen to display waveforms.
7	Acquisition	Length	Length changes according to the horizontal axis range and sampling cycle. The value is computed automatically and displayed. (Length = Time/Div ÷ Sample)
8		Sample	Specify the sampling cycle for the data to be acquired from the waveform display.
9	RUN	RUN / STOP	Clicking RUN starts waveform display; clicking STOP stops waveform display. The button is labeled “STOP” while a waveform is being displayed and is labeled “RUN” while waveform display is stopped.
10	Graph Setting	Smoothing	When turned on, this function smoothes the waveform being displayed. (Click to toggle ON/OFF)
11		Channel Setting	Clicking this button displays Window-Channel Settings.
12		Set Color	Click this button to set a background color of the waveform display screen.

5.3.4 Window-Channel Settings

Window-Channel Settings consists of the Channel Settings area and the Math Channel Settings area. The structure of Window-Channel Settings is shown below.

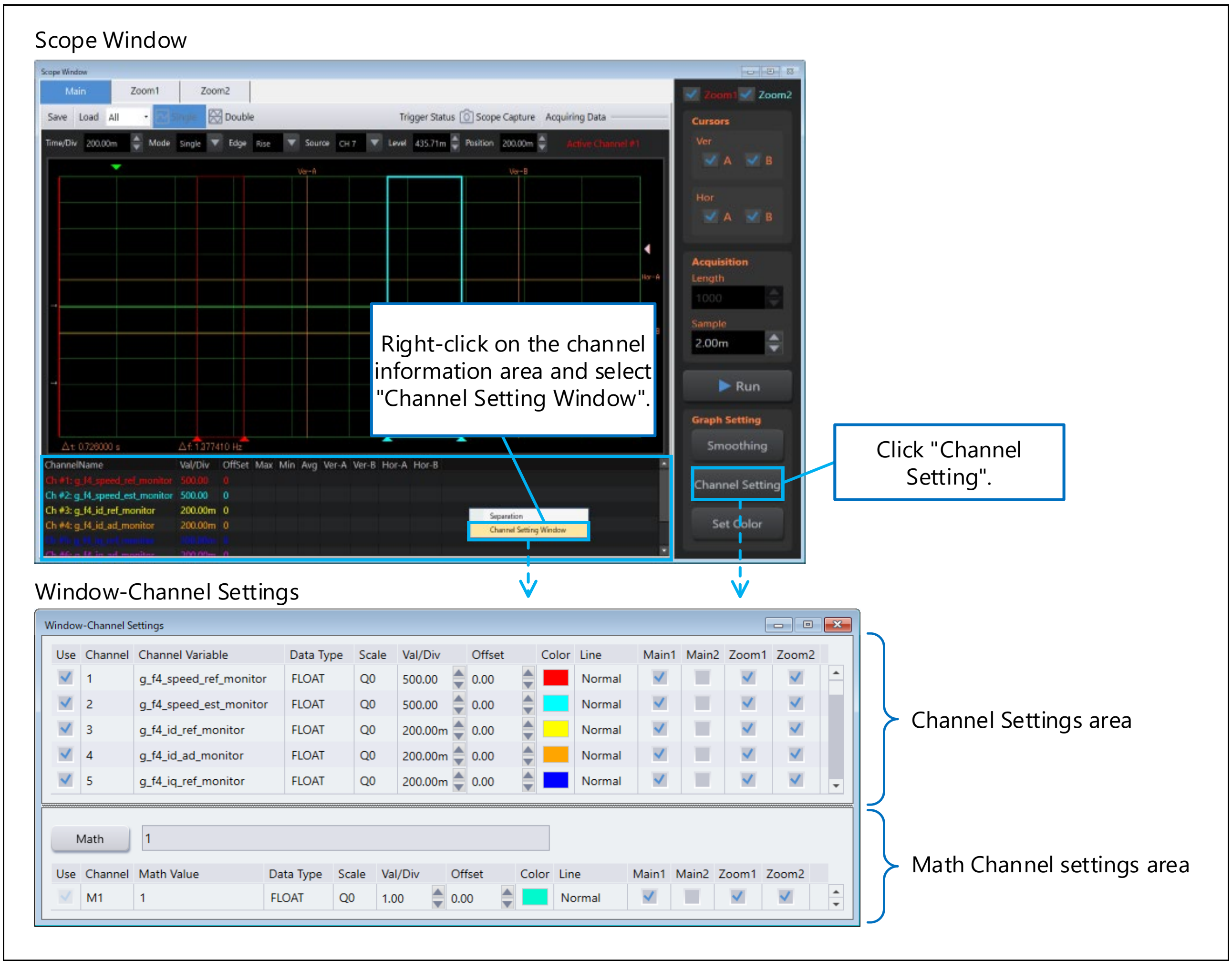


Figure 5-7 Window-Channel Settings

5.3.4.1 Channel Settings Area

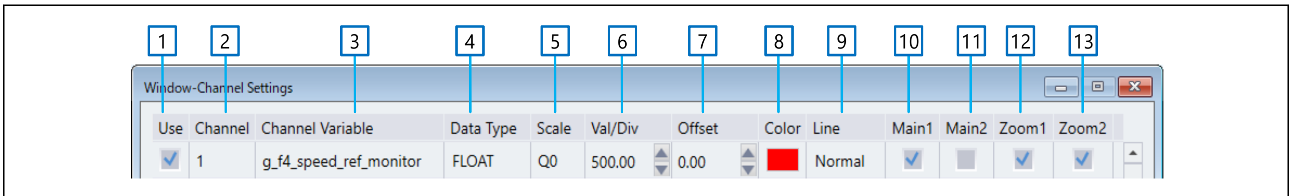


Figure 5-8 Channel Settings Area

Table 5-7 Explanation of Channel Settings Area

No.	Name	Explanation
1	Use	Select the channel for the waveform display. ON/OFF can be operated.
2	Channel	Displays the channel number (channel numbers cannot be edited).
3	Channel Variable	Specify the variable for the waveform display.
4	Data Type	Data type of the variable can be selected.
5	Scale	Vertical axis scale can be entered for the waveform display.
6	Val / Div	Specify the value per division for the vertical axis for the waveform display
7	Offset	Specify the vertical axis offset value for the waveform display
8	Color	Select a color for the waveform display.
9	Line	Select a line width for the waveform display.
10～13	Main1/Main2/Zoom1/Zoom2	Used to select the screens for waveform display.

5.3.4.2 Math Channel Settings Area

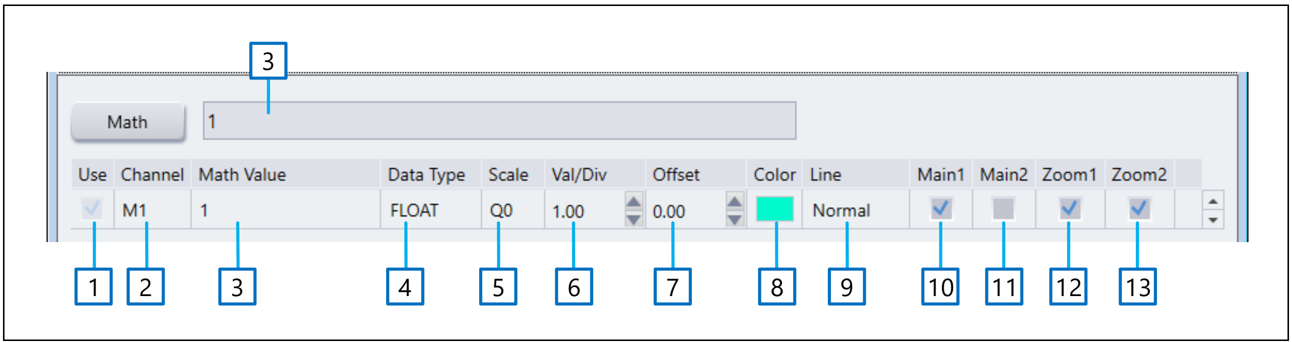


Figure 5-9 Math Channel Settings Area

Table 5-8 Explanation of Math Channel Settings Area

No.	Name	Explanation
1	Use	Display only. ON/OFF cannot be operated.
2	Channel	Displays the math channel number (M1, M2,. . .); math channel numbers are set automatically.
3	Math Value	Specify a formula for math computations.
4	Data Type	Data type can be specified.
5	Scale	Vertical axis scale can be entered for the waveform display.
6	Val / Div	Specify the vertical axis offset values for displaying waveforms.
7	Offset	Specify the vertical axis offset values for displaying waveforms.
8	Color	Select a color for the waveform display.
9	Line	Select a line width for the waveform display. - Thin - Normal - Thick
10～13	Main1/Main2/Zoom1/Zoom2	Used to select the screens for waveform display.

5.3.5 Image Editor Screen

Clicking the “Scope Capture” button at the top right of Scope Window starts Image Editor. The window structure of Image Editor is shown below.

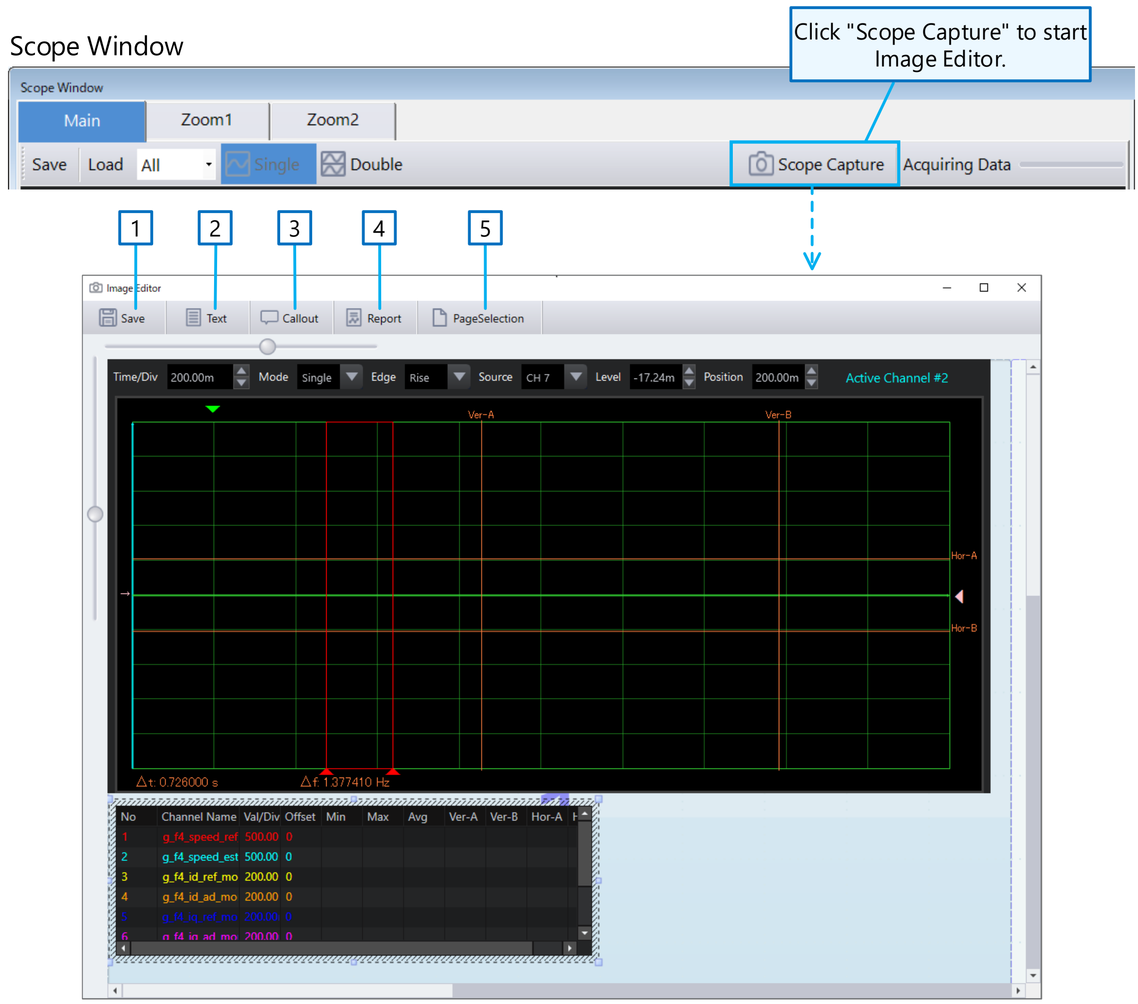


Figure 5-10 Image Editor Screen

Table 5-9 Explanation of Image Editor screen

No.	Name	Explanation
1	Save	Saves the edited image information in a bmp file.
2	Text	Adds autoshape text.
3	Callout	Adds an autoshape callout
4	Report	Saves the edited image information in a pdf file.
5	Page Selection	Click this button to select page to be printed. Tun on the displayed page number by clicking, and the page will become the target for printing.

5.4 Measurement Channel Setting

5.4.1 Displaying Channel Settings Window

To display waveforms in Scope Window, first set up channels in the Channel Settings area of Window-Channel Settings. Window-Channel Settings is displayed in one of the following ways:

- Click the "Channel Setting" button in the measurement settings area of Scope Window
- Right-click on the channel information area at the bottom of Scope Window and select "Channel Settings Window" from the menu.

For details on how to open Window-Channel Settings, see Section 5.3.4.1 Channel Settings Area.

5.4.2 Setting in Channel Settings Area

This section explains how to set the items of the Channel Settings area.

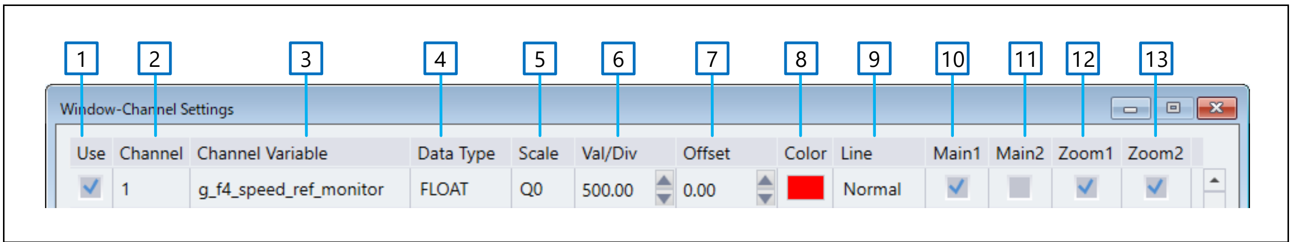


Figure 5-11 Channel Settings Area

(1) Use

Selecting a check box in the Use column enables that channel information. Invalid channel information is grayed out. The maximum number of channels that can be enabled depends on the communication board that is connected.

(2) Channel

The Channel filed shows the assigned channel numbers starting from 1. The channel number cannot be changed in this field. You need to check the checkbox of “Use” to use the channel.

(3) Channel Variable

In the Channel Variable field, specify the variable name for waveform display. The method to enter a variable name is the same as other functions (see Section 4.4.1 Specifying Variable Name.)

(4) Data Type

In the Data Type field, select the data type of the variable from the list.

・Selectable Data Type : UINT8, INT8, UINT16, INT16, UINT32, INT32, FLOAT, BOOL, LOGIC

(5) Scale

In the Scale field, specify the scale value for the variable data (waveform data) set in the channel. The formulas in Table 5-10 are applied to the display data for the waveform data. For handling waveform data, see Section 5.5.2.

Table 5-10 Scale Settings Value

Set value	Formula for calculating data
Qn (n = 0 to 31)	Data × (1 / 2^n)
Integer	Data × (1/integer)
Decimal	Data × (1/decimal)

(6) Val/Div

In the Val/Div field, enter the 1 div value of the vertical axes for displaying the data of the variable with the channel set. You can also use the ▼ and ▲ symbols to the right of the input boxes to change the value.

(7) Offset

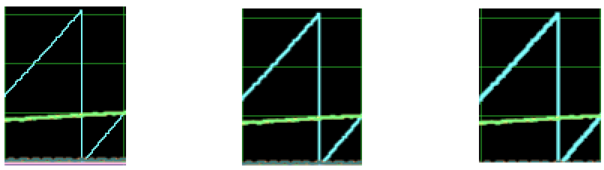
In the Offset filed, enter the offset value for the vertical axes for displaying the data of the variable with the channel set. You can also use the ▼ and ▲ symbols to the right of the input boxes to change the value.

(8) Color

In the Color field, select the waveform display colors for the variable with the channel set.

(9) Line

In the Line filed, select the waveform display line width for the variable with the channel set.



Thin Normal Thick

Figure 5-12 Selectable Line Width for Waveform Display

(10) to (13) Main1 / Main2 / Zoom1 / Zoom2

Fields (10) to (13) are used to select windows to display waveforms of the variable set in the channel. You can only choose one of the Main windows (Main1 or Main2) in the Main tab.

5.5 Math Function

By using Math function, the specified calculation can be performed on the value of the variable with the channel set, and the result can be displayed as a waveform on Scope Window.

5.5.1 Math Channel Settings Area

To display the waveform using the Math function, register a Math channel in the Math Channel Settings area of Window-Channel Settings

For details on how to open Window-Channel Settings, see Section 5.4.1 Displaying Channel Settings Window.

By right-clicking on the Math Channel Settings area and selecting "Math Channel Add" from the menu, you can add rows for Math channels. You can also insert or delete rows for the added rows in the same way.

Math channel numbers are assigned automatically by the system. You cannot sort the display of the Math channels.

。

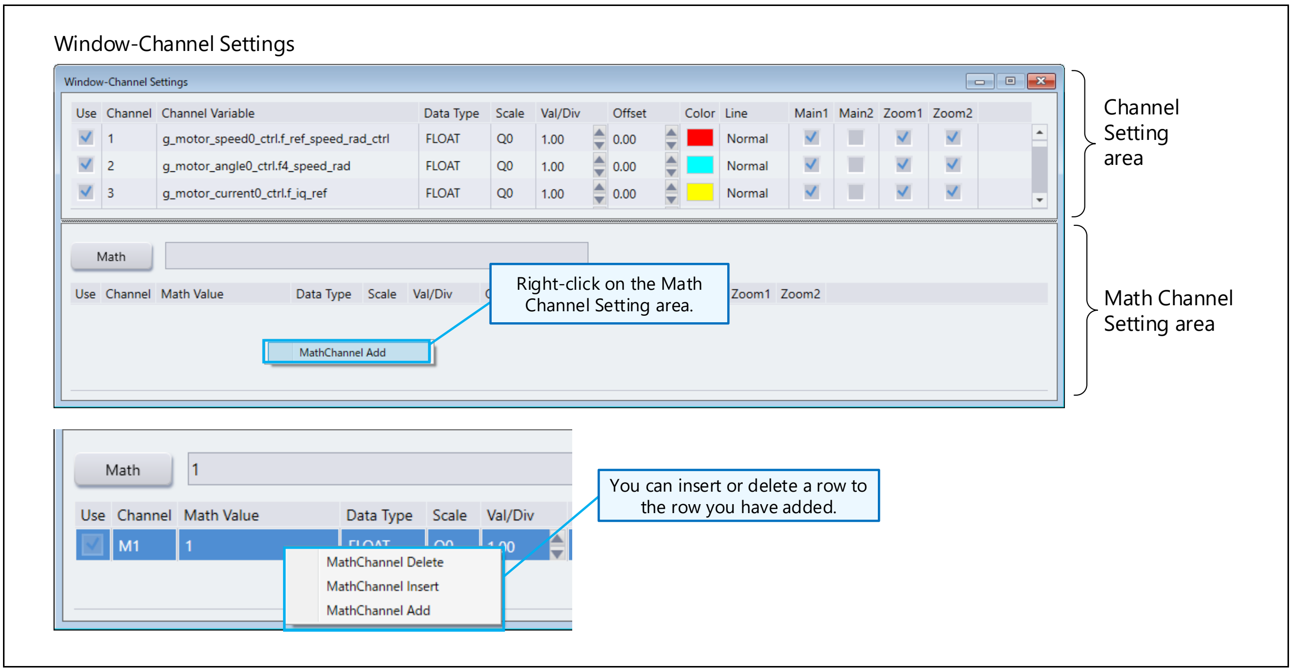


Figure 5-13 Math Channel Setting Settings Area

The following describes how to set the items in the Math Channel Settings area.

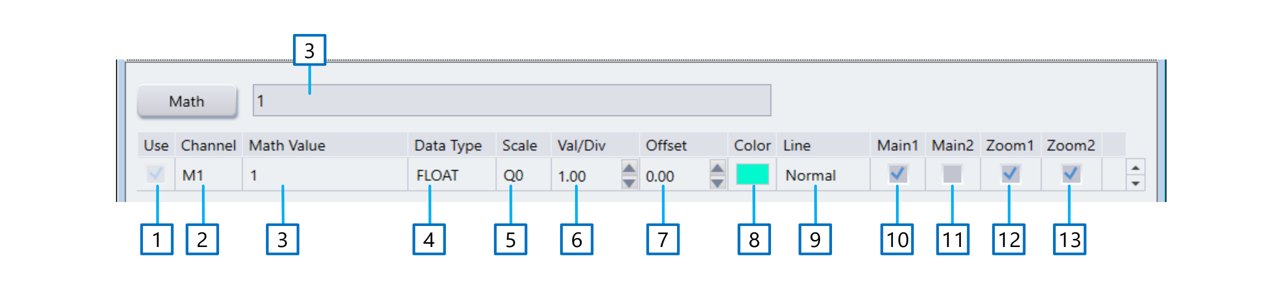


Figure 5-14 Math Channel Settings Area

(1) Use

The checkbox of Use becomes ON when a Math channel is added. You cannot change it.

(2) Channel

The Channel field shows the assigned channel number starting from M1. You cannot change it.

(3) Math Value

In the Math Value field, specify a channel number, value, function, and arithmetic expression. You cannot directly enter a variable name into the Math Value field. In addition, functions and operators have the priority and some restrictions.

In the following cases, the Math Value field turns red, indicating that the entry is not completed or there is a problem.

- During Math Value entry, the arithmetic expression is insufficient or incorrect.
- The specified channel number does not exist.
- The specified Math channel number is on a row below the arithmetic expression.

Table 5-11 Specifiable Values

Setting value		Specification method	Setting example	Remarks
Channel	Channel number	ch(n)	ch1, ch2···	For channel value, the value obtained by applying the Scale value (displayed value) is used.
	Math channel number	M(n)	M1, M2···	For channel value, the value obtained by applying the Scale value (displayed value) is used.
Value	Integer	-	50, −1, 0	—
	Decimal	-	1.5, −2.5	—

Table 5-12 Specifiable Functions and Operators

Type	Operation	Functions/Operators	Setting example	Priority
Function	Delay *Note1	z^n	• z^1(ch1)+ch2 • z^−1(M1)	Priority 1 (high)
	Arc tangent 2 Note2	Atan2(ch(n),ch(n))	• Atan2(ch1,ch2) • Atan2(2,3) (= 0.9827937)	Priority 2
	Sine *Note3	sin()	• sin(ch1) • sin(60) (= −0.3048106)	
	Cosine *Note3	cos()	• cos(ch1) • cos(60) (= −0.952413)	
	Tangent *Note3	tan()	• tan(ch1) • tan(60) (= 0.3200404)	
	Arc tangent *Note3	Atan()	• Atan(ch1) • Atan(60) (= 1.554131)	
	Square root	sqrt()	• sqrt(2) (= 1.414214)	Priority 3
Operator	Power	^	• 2^3 (= 8)	Priority 4
	Multiplication *Note4	*	• ch1*7	Priority 5
	Division	/	• ch1/7	
	Addition	+	• ch1+5	Priority 6 (Low)
	Subtraction	-	• ch1−8	

Note

*1: If the specified data does not exist, 0 is assumed for the computation result; in the case of z^0(ch1), it is assumed to have the same meaning as ch1.

*2: For the Atan2 argument, only channel information (such as ch1 and M1) and numerical values can be specified. No formulas can be specified.

*3: Trigonometric function values are computed in radians (rad).

*4: Implied multiplication cannot be specified (for example, 2(3+4), (1+3)(2+4), 5sin(60), etc.).

(4) Data Type

In the Data Type field, select the data type of the variable from the list.

- Selectable Data Type: UINT8, INT8, UINT16, INT16, UINT32, INT32, FLOAT, BOOL, LOGIC

(5) Scale

In the Scale field, specify the scale value for the waveform data. The formulas in Table 5-13 are applied to the data according to the setting value. For handling waveform data, see the next Section 5.5.2

Table 5-13 Scale Setting Value

Set value	Formula for calculating data
Qn (n = 0 to 31)	Data × (1 / 2^n)
Integer	Data × (1/integer)
Decimal	Data × (1/decimal)

(6) Val / Div

In the Val/Div field, enter the 1 div value of the vertical axes for displaying the data of the variable with the channel set. You can also use the ▼ and ▲ symbols to the right of the input boxes to change the value.

(7) Offset

In the Offset filed, enter the offset value for the vertical axes for displaying the data of the variable with the channel set. You can also use the ▼ and ▲ symbols to the right of the input boxes to change the value.

(8) Color

In the Color field, select the waveform display color for the variable with the channel set.

(9) Line

In the Line filed, select the waveform display line width for the variable with the channel set.

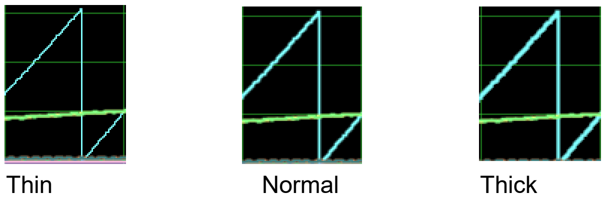


Figure 5-15 Selectable Line Width for Waveform Display

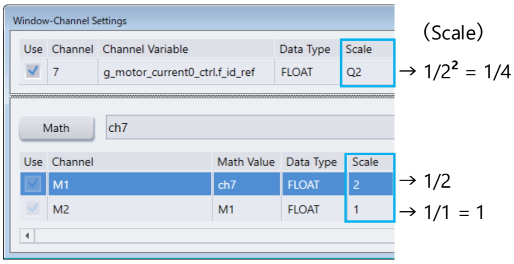
(10) to (13) Main1／Main2／Zoom1／Zoom2

Fields (10) to (13) are used to select windows to display waveforms of the variable set in the channel. You can only choose one of the Main windows (Main 1 or Main 2) in the Main tab.

5.5.2 Handling Wave Data

The sampled “waveform data value” is retained without considering the scale. On the other hand, the waveform is displayed in Scope Window with the scale applied.

When referring a channel number (such as ch7) or Math channel number (such as M1) in the Math Value field of the Math Channel, the value of the waveform display of referring source (value to which the scale is applied) is used as the waveform data value.



Example：When the waveform data of "g_motor_current0_ctrl.f_id_ref" is 4:

Channel	Waveform data value	Scale	Waveform display value
ch7	4	1/4	1
M1	1	1/2	0.5
M2	0.5	1	0.5

Figure 5-16 Waveform Data Handling

5.5.3 Drawing Math Channel Waveform

When you edit the Math channel information on Window-Channel Settings and then click the “Math” button, the waveform for the Math channel is drawn. If it has been already displayed, it is redrawn.

5.6 Waveform Measurement

5.6.1 Trigger Settings Area

Set triggers when displaying a waveform on Scope Window. The following explains the operations for the trigger settings area.

5.6.1.1 Time/Div

In the Time/Div field at the top of Scope Window, specify a time per division for the vertical axis on the waveform display screen.

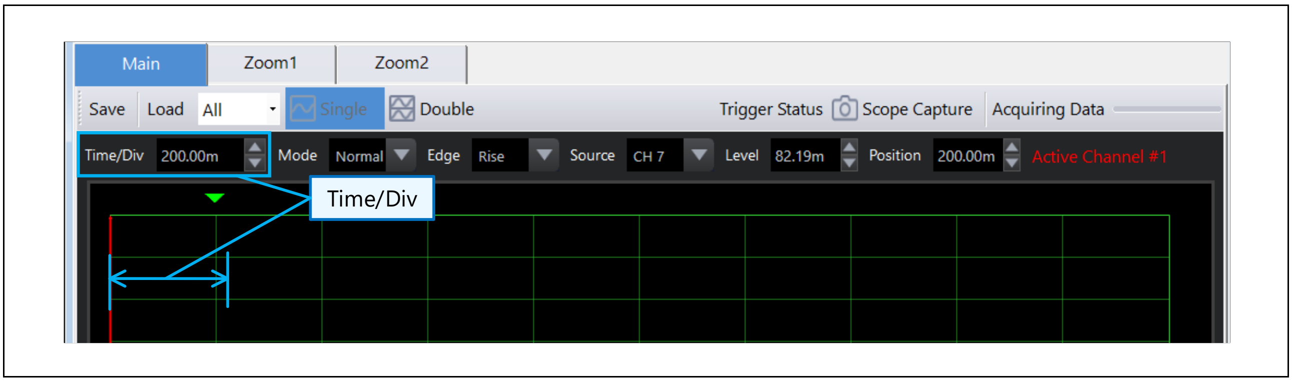


Figure 5-17 Time/Div Selection

5.6.1.2 Mode

In the Mode field at the top of Scope Window, select the operation mode to be in effect when a triggering event occurs following startup of waveform display.

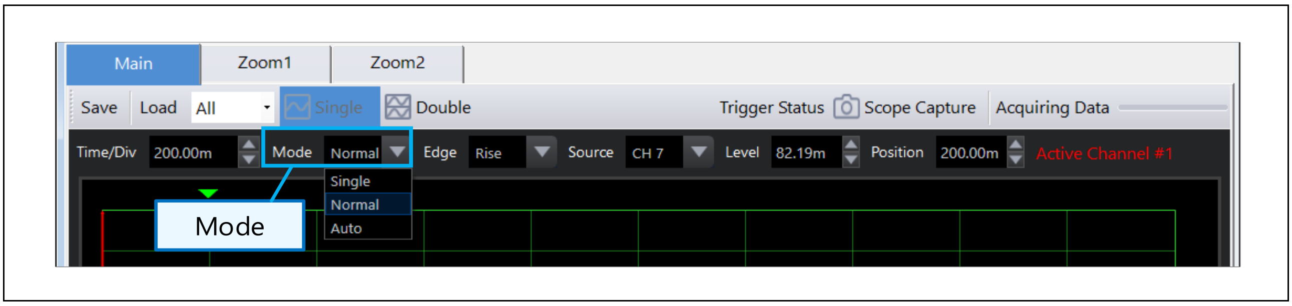


Figure 5-18 Mode Selection

Table 5-14 Mode list

Name	Explanation
Single	Displays the waveform and stops updating the waveform display when a trigger event occurs.
Normal	Updates the waveform display each time a trigger event occurs.
Auto	Repeats automatic updating of the waveform display at a fixed interval.

5.6.1.3 Edge

In the Edge field at the top of Scope Window, set Rise, Fall, or Both of the waveform data as a trigger event following startup of waveform display.

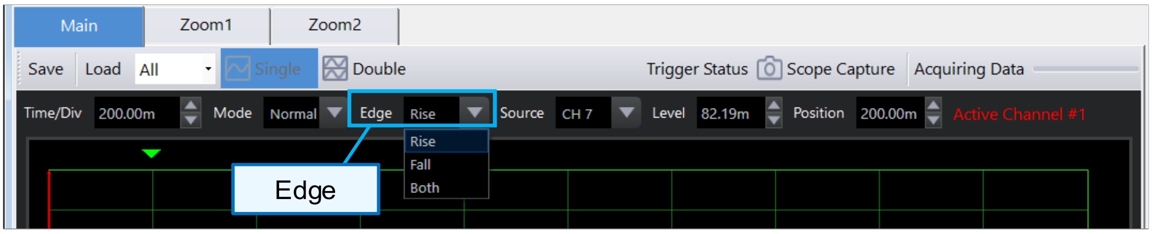


Figure 5-19 Edge Selection

Table 5-15 Edge Function List

Name	Explanation
Rise	Rising edge
Fall	Falling edge
Both	Both edges

5.6.1.4 Source

In the Source field at the top of Scope Window, select a channel number to be targeted by the trigger event following startup of waveform display.

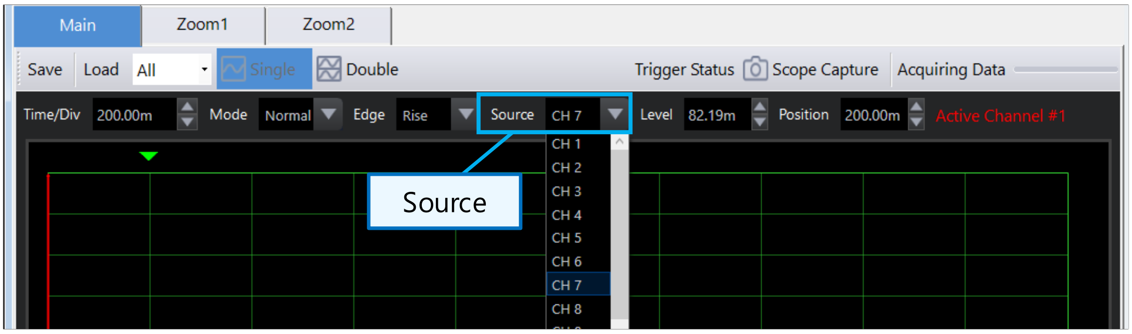


Figure 5-20 Source Selection

5.6.1.5 Level

In the Level field, specify a waveform data value (level) that will become a trigger following startup of waveform display. This setting value will be indicated by the ◂ symbol on the right side of the waveform. You can also adjust the value by dragging the ◂ symbol with a mouse.

5.6.1.6 Position

In the Position field, specify the position (on the horizontal axis) to display a data point at which a trigger event occurs following startup of waveform display. This position will be indicated by the green ▼ symbol along the top of the waveform display screen. You can also adjust the position by dragging the ▼ symbol with a mouse.

Figure 5-21 Level and Position Settings

5.6.2 Acquisition Settings in Measurement Settings Area

5.6.2.1 Length

Length changes according to the horizontal axis range and sampling cycle, and the value is computed automatically and displayed. (Display only)

Length (display width) = Time/Div ÷ × 10 ÷ Sample

(Example) Time/Div = 200, Sample = 2ms ⇒ Length = 200 × 10 ÷ 2 = 1000

5.6.2.2 Sample

Specify the sampling cycle of the data to be acquired for the waveform display.

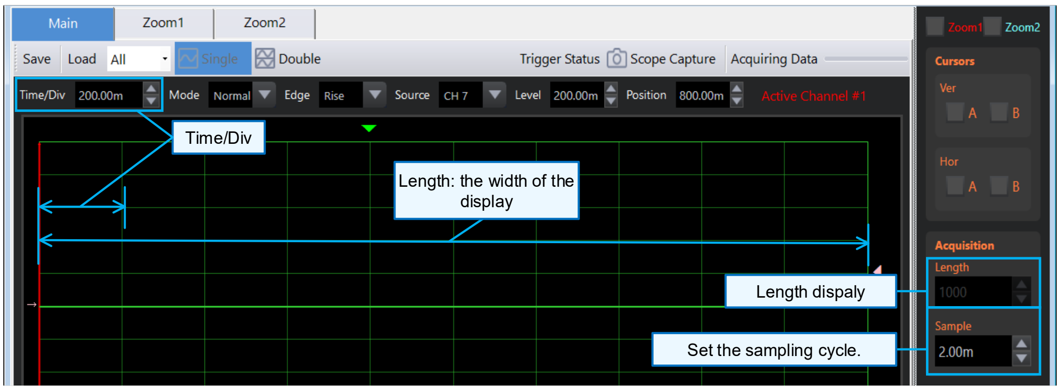


Figure 5-22 Acquisition in Measurement Settings Area (Length and Sample)

5.6.3 Starting and Stopping Waveform Measurement

To start waveform measurement, click the “RUN” button. If the trigger setting mode has been set to Single, waveform measurement is stopped automatically following any trigger event. Otherwise, click the “STOP” button to stop waveform measurement.

If you read a variable value from the Control Window or write a value into a variable while performing waveform measurement, updating of the waveform measurement result display might be delayed in some cases because processing underway in the Control Window has a higher priority.

Figure 5-23 Starting/Stopping Waveform Measurement

Table 5-16 RUN/STOP Button Display

Button label	Action
RUN	Clicking starts waveform measurement (the button's label becomes “STOP” after starting)
STOP	Clicking stops waveform measurement (the button’s label becomes “RUN” after stopping)

5.6.4 Saving and Loading Waveform Data

You can save waveform data with the “Save” button at the top of Scope Window, and load waveform data with the “Load” button.

If you specify “*.csv1” as the file type in the Save operation, two files (*.csv and *.csv1) are saved. If you specify “*.csv”, only a file with *.csv extension is saved.

When you click the “Load” button, the saved waveform data (*.csv1 format) is loaded.

Table 5-17 Waveform Data Saving Format

File type at the time of saving	Saved information and handling
ICS_Wave Data File(*.csv1)	• Saves a file that contains sampled values of the waveform data (*.csv) and a file that contains both the sampled values and the setting information for waveform measurement (*.csv1). • When a saved file (*.csv1) is loaded to RMW, the saved waveform can be displayed.
Comma Separated File(*.csv)	• Saves only sampled values of the waveform data, separated by commas, in CSV format. • Saved files (*.csv) cannot be loaded to RMW.

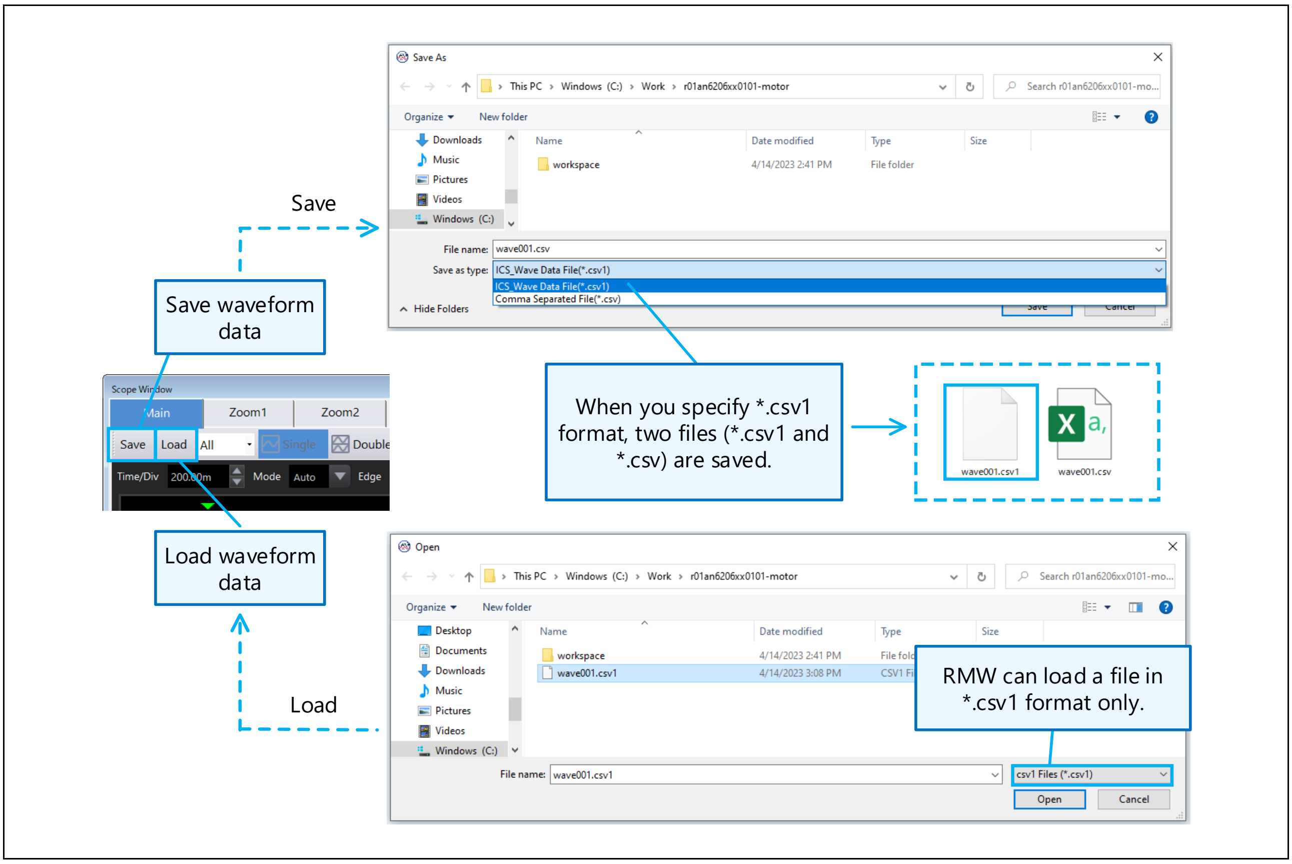


Figure 5-24 Saving and Loading Waveform Data

5.7 Displaying Waveform

5.7.1 Displaying Scope Window

Scope Window is displayed automatically when Analyzer is stared.

5.7.2 Switching Waveform Graph Display

You can switch the waveform graph to be displayed with the tabs at the upper left of Scope Window.

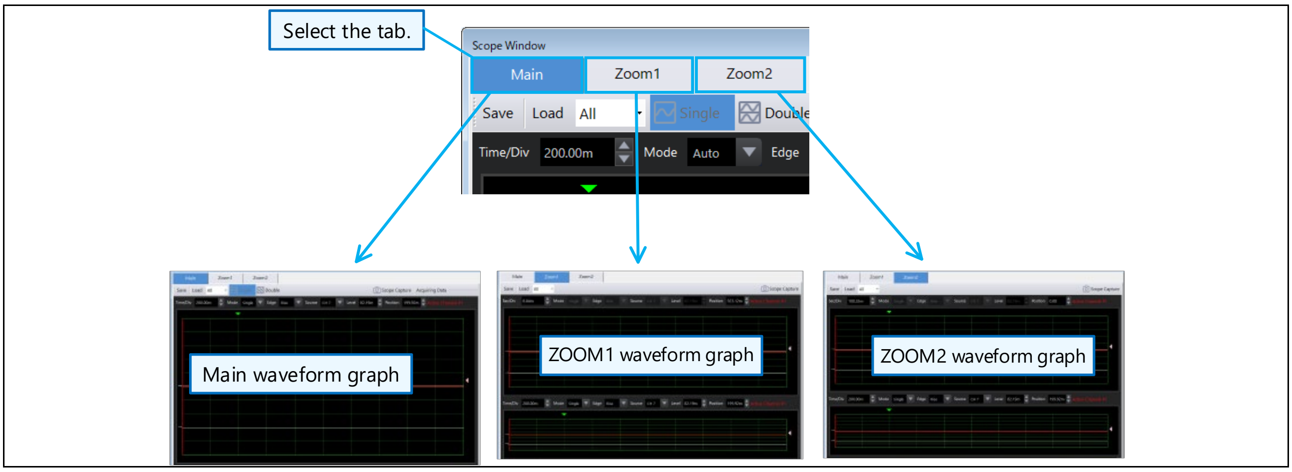


Figure 5-25 Switching Waveform Graph

5.7.3 Switching waveform between Main1 and Main2

The Main waveform graph of Scope Window can be displayed in two windows: Main1 and Main2. While one window is displayed, click the "Double" button to display two windows. While two windows are displayed, click the "Single" button to display one window (only Main 1).

You can specify whether the waveform data of each channel is displayed on Main 1 or Main 2 from Channel Settings Window. For details, refer to Section 5.4.1 Displaying Channel Settings Window.

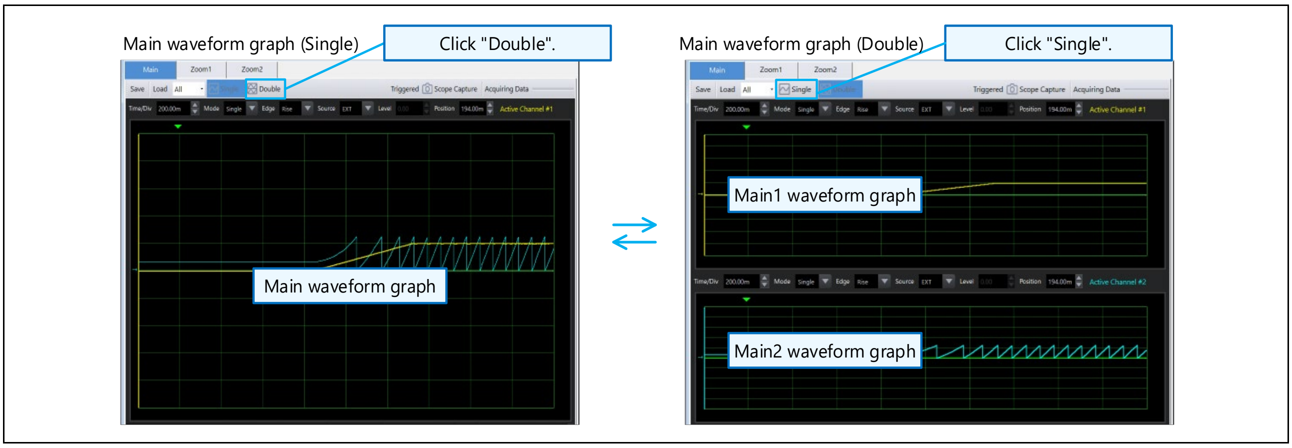


Figure 5-26 Waveform Double Window Display

5.7.4 Specifying Zoom Waveform Display Area

In the Main waveform graph, you can specify the area to be displayed in the Zoom waveform graph.

When you select the check box for Zoom1 and Zoom2 in the measurement settings area of Scope Window, a red frame for Zoom1 and a blue frame for Zoom2 are displayed in the Main waveform display screen.

To adjust the zoom area, first adjust the width by dragging the ▲ symbol at the bottom right corner, then move the start point by dragging the ▲ symbol at the bottom left corner. If the specified zoom area are full of the screen width, you cannot move the ▲ symbol at the bottom left corner.

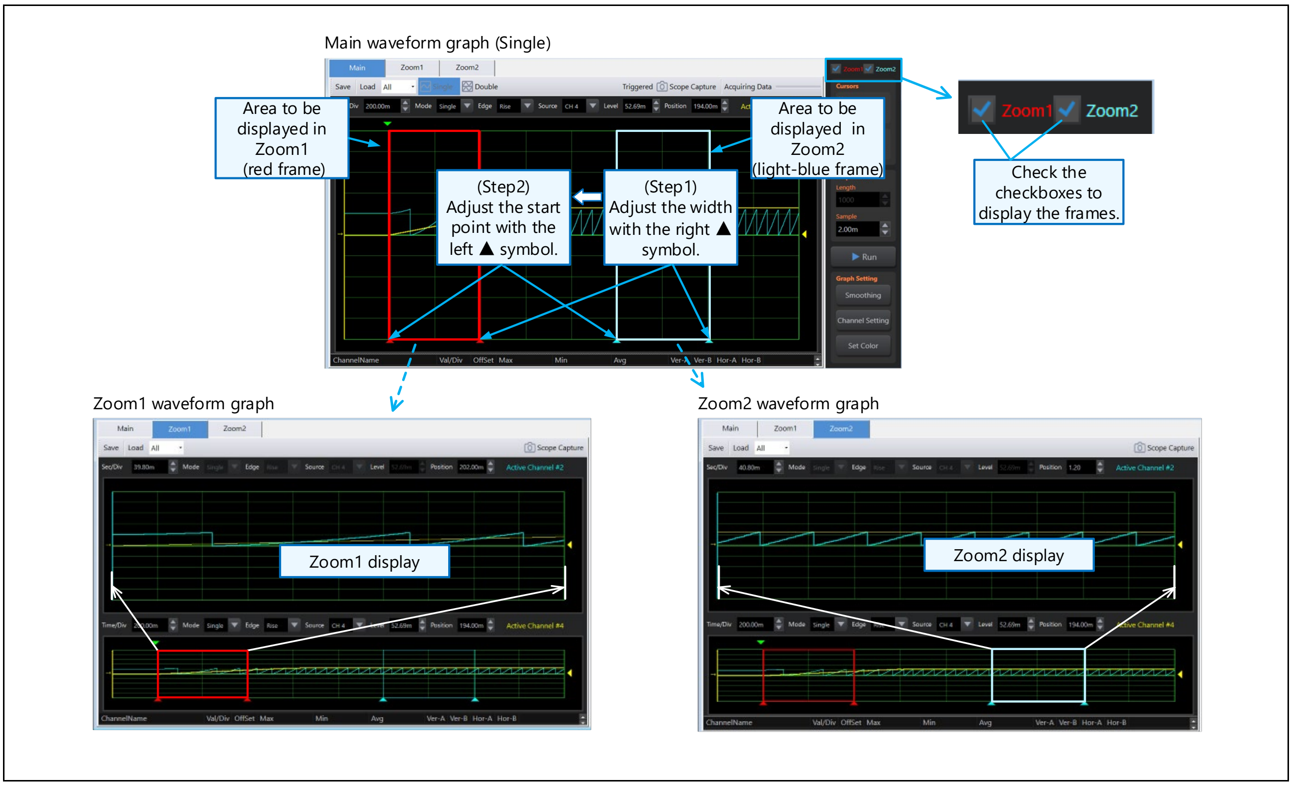


Figure 5-27 Specifying Zoom Area (Single View)

In the Double view, you can also specify the area to be displayed in the Zoom waveform graph. You can move the area between Main1 and Main2 by right-clicking the ▲ symbol at the bottom corner of the area (either left or right).

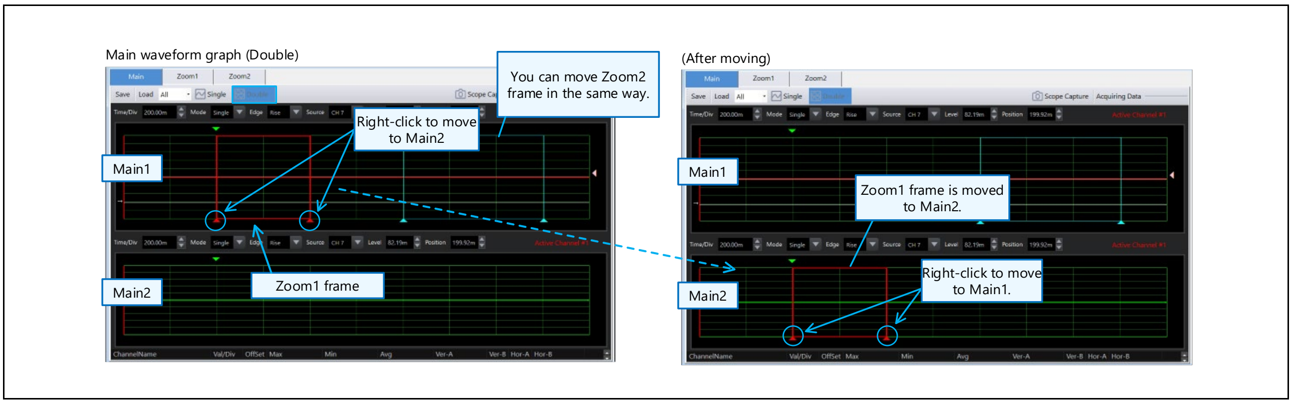


Figure 5-28 Specifying Zoom Area (Double View)

5.7.5 Separating Channel Information Display Area

The channel information display area located at the bottom of the Main, Zoom1, and Zoom2 tabs can be separated to another screen. Right-click on the channel information display area to display the menu, then select “Separation” to separate the area to another screen.

To return the separated channel information display area into the original location, click the Exit button at the upper right corner of the separated screen. If you switch tabs between Main, Zoom1, and Zoom2 while the channel information display area is separated, the area will return to the original location.

The following screen titles are displayed in the separated channel information display area.

Table 5-18 Titles of Separated Channel Information Display Area

Tab before separation	Screen title
Main	MainScope ChannelData
Zoom1	Zoom1Scope ChannelData
Zoom2	Zoom2Scope ChannelData

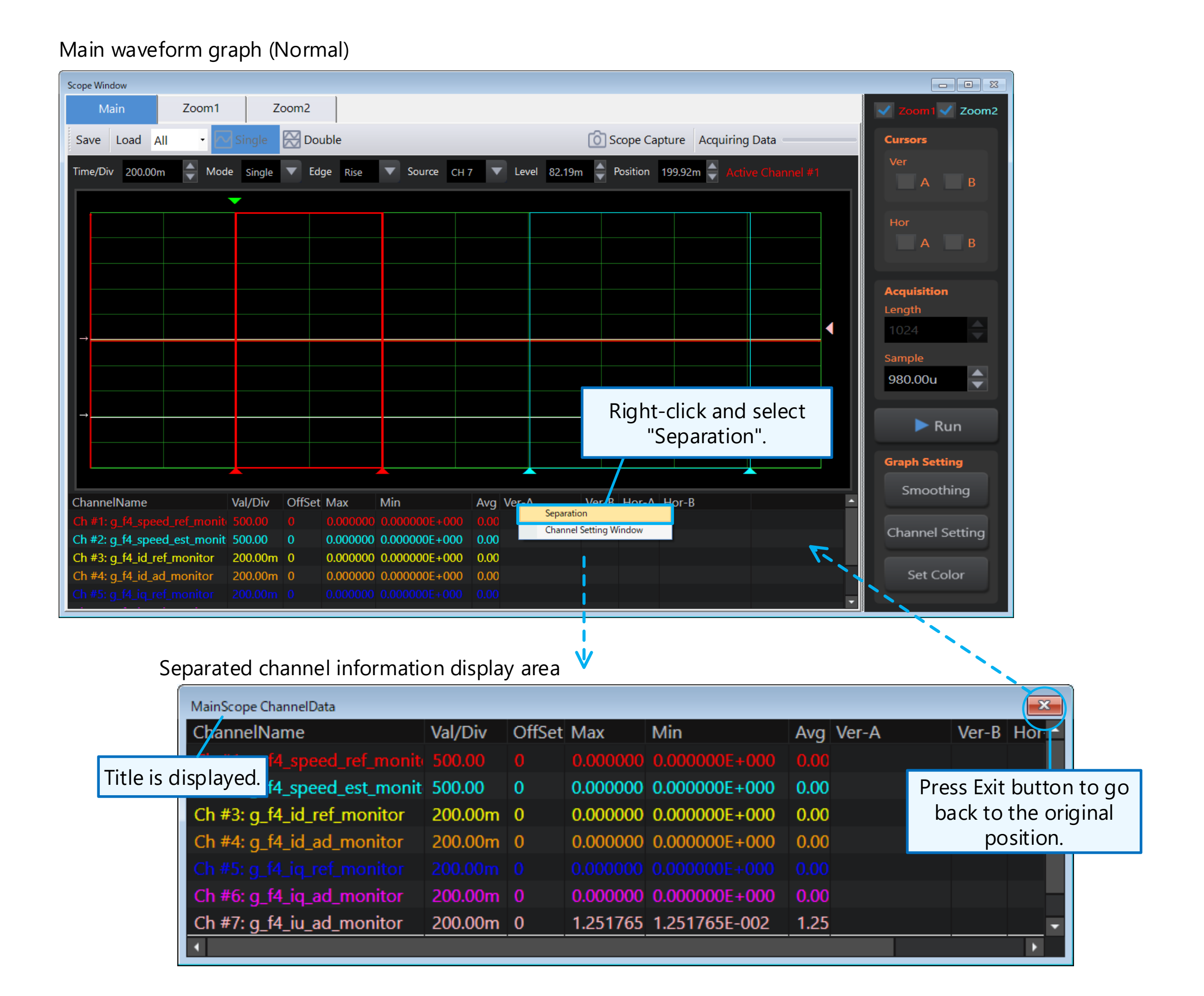


Figure 5-29 Separating Channel Information Display Area

5.7.6 Separating Main Waveform from Zoom Screen

You can separate the Main waveform in the Zoom1 and Zoom2 tabs to another screen.

To return the separated Main waveform graph into the original location, click the Exit button at the upper right corner of the separated screen. If you switch tabs of Main, Zoom1, and Zoom2 while the separated screen is displayed, the Main waveform graph will return to the original location.

The following screen titles are displayed in the separated Main waveform graph.

Table 5-19 Titles of Separated Main Waveform Graph

Tab before separation	Screen title
Main1	Main1Scope Window
Main2	Main2Scope Window

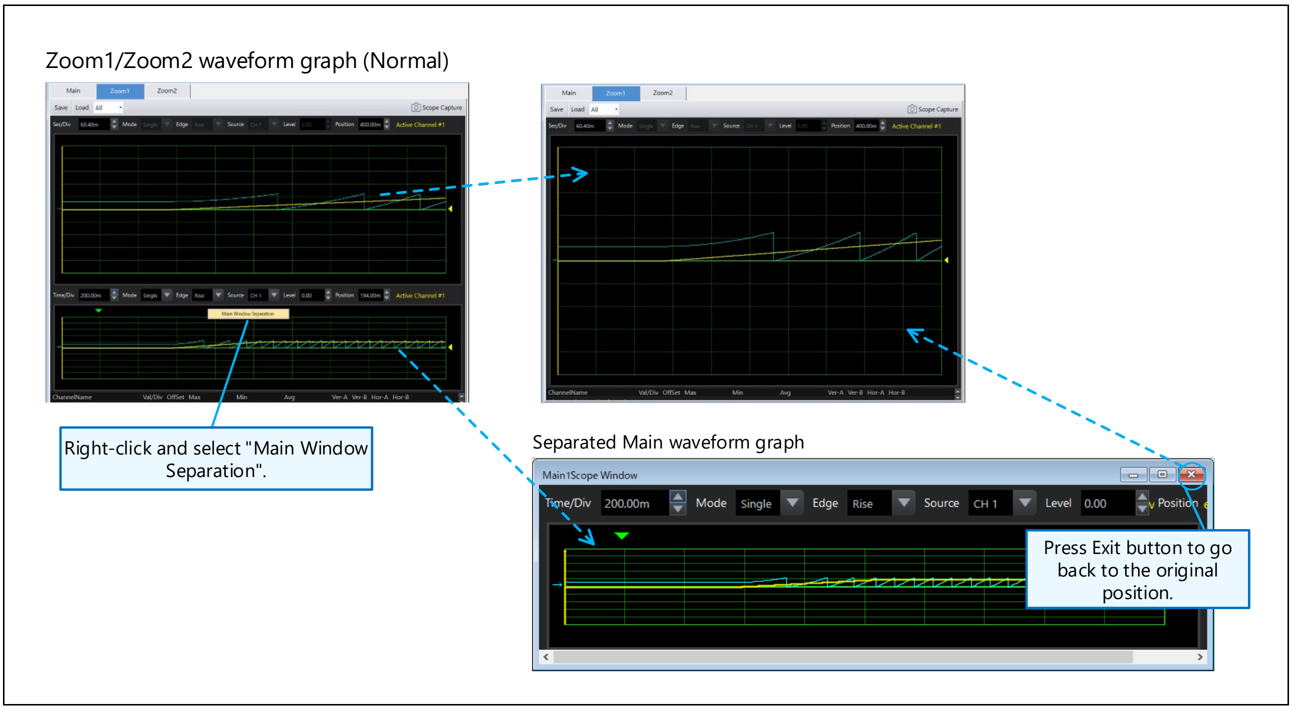


Figure 5-30 Separating Main Waveform Display

5.7.7 Active Channel Display

In the channel information display area of Main and Zoom tabs, you can set a channel as Active Channel. When some waveforms are displayed and overlapped, the waveform of Active Channel is displayed in the frontmost.

In addition, the left vertical axis of the graph turns the same color as the waveform of Active Channel.

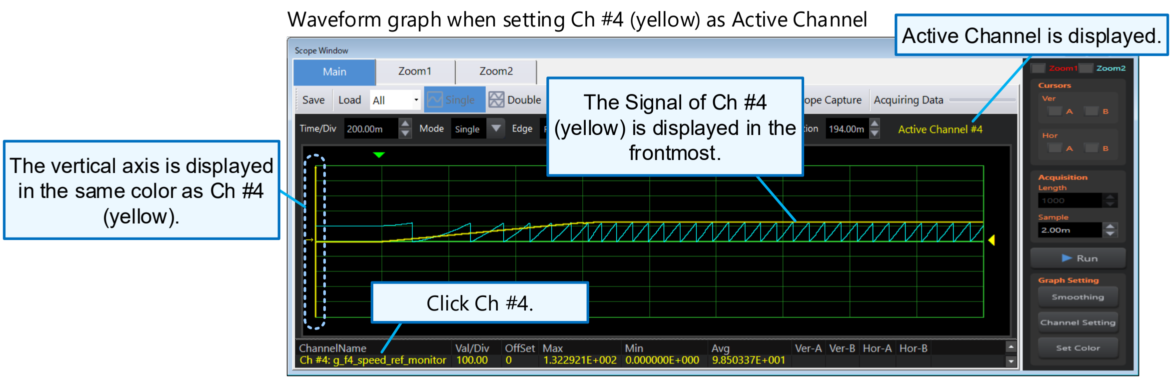


Figure 5-31 Active Channel and Waveform Graph Display

5.7.8 Cursor Display

When the Cursors checkbox of Scope Window measurement settings area is ON, two Ver (vertical) cursors and two Hor (horizontal) cursors can be displayed on the graph. You can adjust the position of each cursor line on the screen by dragging with a mouse.

The values of each waveform at the position of the cursors are displayed in the channel information display area at the bottom of the screen.

When two Ver (vertical) cursors are displayed, the time and frequency between the cursors are displayed.

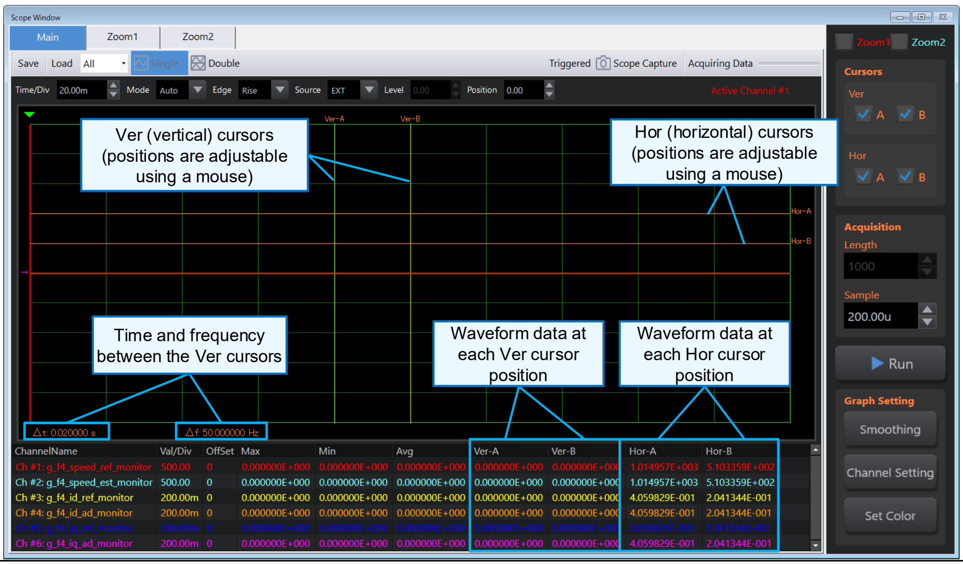


Figure 5-32 Displaying Cursor

5.7.9 Thinning Waveform Display Points

You can switch the point count for displaying waveform data from the list box at the upper left on Scope Window. When “All” is selected, all the data is displayed. Otherwise, thinning is carried out according to the point count setting before displaying the data. For example, if “1/5” is selected, one point is displayed for every five points in the waveform data.

The setting cannot be changed while waveform measurement is being performed.

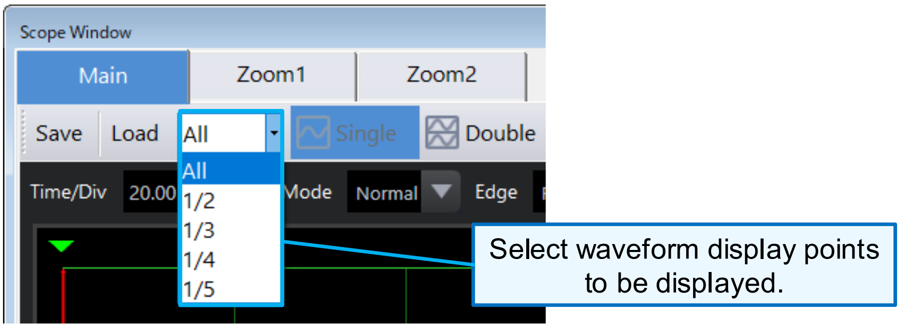


Figure 5-33 Waveform Thinning Settings

5.7.10 Smoothing Waveform

You can smooth the waveform display by clicking the “Smoothing” button in the measurement settings area of Scope Window. Turn the button ON to display a smoothed waveform graph. Turn it OFF to display the graph in the normal state.

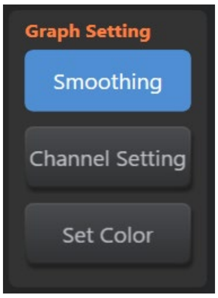


Figure 5-34 Smoothing Button (ON State)

5.7.11 Changing Background Color

You can change the Scope Window's background color by clicking the “Set Color” button at the bottom right of Scope Window.

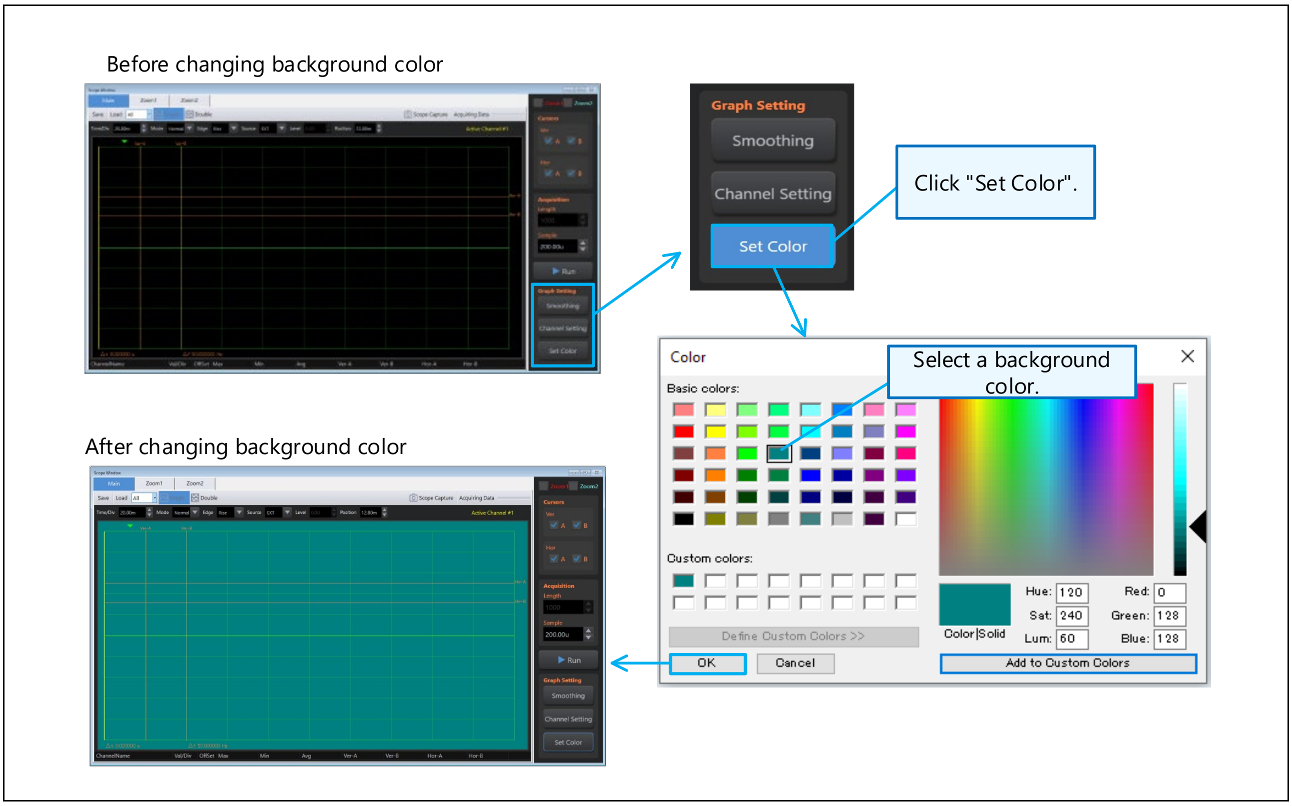


Figure 5-35 Changing Background Color by Set Color

5.7.12 Shortcut Keys

The following keyboard shortcuts are available in Scope Window.

Table 5-20 List of Shortcut Keys

Overview	Action	Shortcut
Displaying the Zoom range frame	Shows or hides Zoom1.	Alt + 1
	Shows or hides Zoom2.	Alt + 2
Editing the Zoom range frame	Moves the starting point (of selected Zoom range frame) to the right.	Alt + right cursor
	Moves the starting point (of selected Zoom range frame) to the left.	Alt + left cursor
	Expands the width (of selected Zoom range frame).	Alt + Shift + right cursor
	Narrows the width (of selected Zoom range frame).	Alt + Shift + left cursor
Measuring waveforms	Starts measurement (= click RUN button)	R
	Stops measurement (= click STOP button)	S
Acquiring screen image	Copies the screen image.	Ctrl + C
Editing the Active Channel	Increases the Val/Div value for Main1.	Up cursor
	Decreases the Val/Div value for Main1.	Down cursor
	Increases the Val/Div value for Main2.	Shift + up cursor
	Decreases the Val/Div value for Main2.	Shift + down cursor
Switching trigger settings	Switches Mode.	O
	Switches Mode (in the opposite direction).	Shift + O
	Switches Edge.	E
	Switches Edge (in the opposite direction).	Shift + E
	Switches Source.	T

5.8 Image Editor Function (Scope Capture Button)

Clicking the “Scope Capture” button at the top of Scope Window starts Image Editor. Image Editor provides a function to capture and edit the waveform images and the channel information that are displayed.

When you click the “Scope Capture” button, the waveform images and the channel information at that time are loaded into Image Editor. While Image Editor is active, additional waveform images and channel information are loaded by every click of “Scope Capture” button.

5.8.1 Starting and Terminating Image Editor

You can start Image Editor from the “Scope Capture” button on the top of Scope Window, and terminate it from the Exit button on the upper right of Image Editor.

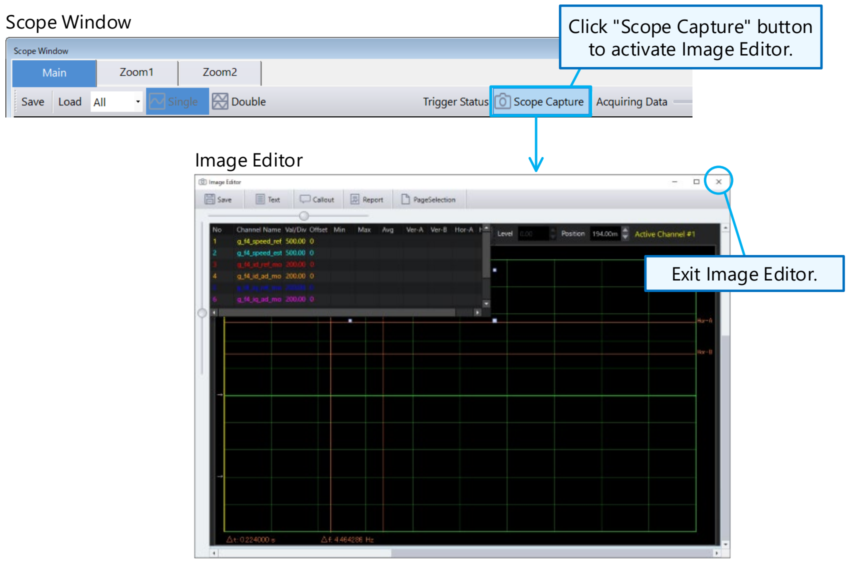


Figure 5-36 Starting/Terminating Image Editor

5.8.2 Changing Display Position and Size

When you click an image displayed in Image Editor to select it, it is surrounded by the shaded area and size-change handles. If you grab a size-change handle with a mouse, you can change the display size. If you grab the shaded area with a mouse, you can move the display position within Image Editor.

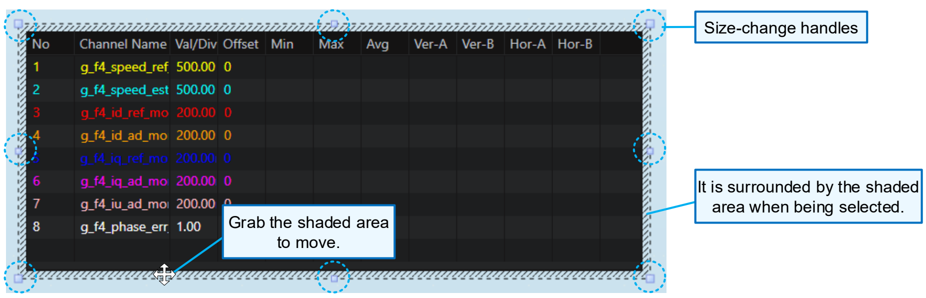


Figure 5-37 Changing Display Position/Size

5.8.3 Image Editor Operation Button

The top of Image Editor contains the following operation buttons.

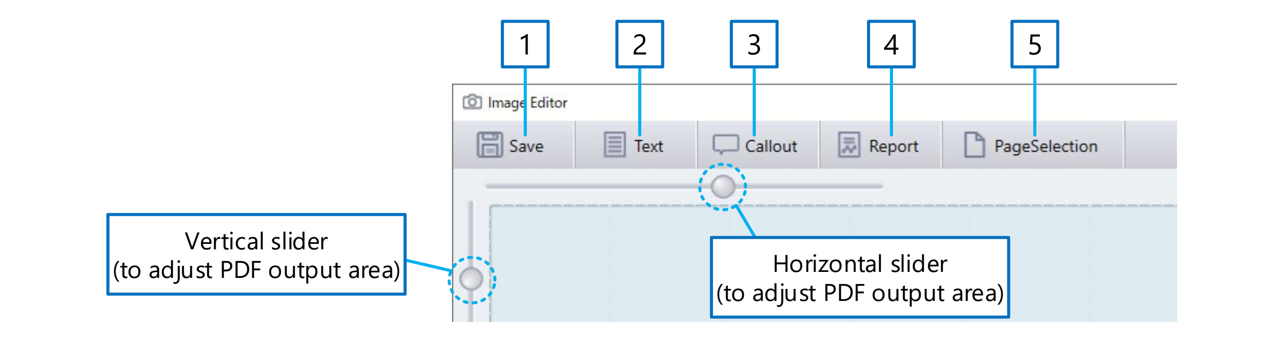


Figure 5-38 Image Editor Operation Buttons

(1) Save button

Click the “Save” button to save the image information edited by Image Editor in bmp/png/jpg format.

(2) Text button

Click the “Text” button to display a text-enabled autoshape within Image Editor. You can move and edit it. Select characters of the displayed text and right-click to display the menu. You can change the color and font.

(3) Callout button

Click the “Callout” button to display the callout menu. When you select one, a callout is displayed in Image Editor, which you can move and edit. You can change the setting from the right-click menu of the displayed callout.

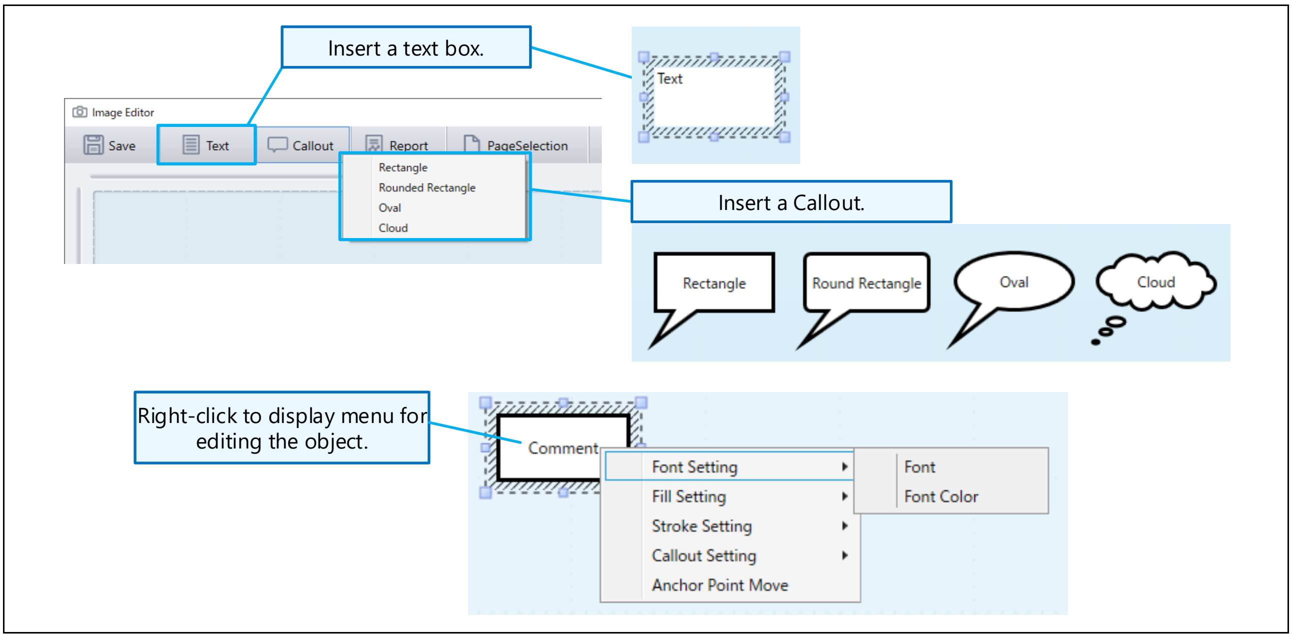


Figure 5-39 Callout Type

Table 5-21 Right-Click Menu Setting for Text and Callout

Text	Callout	Right-click menu	Sub menu	Action
●	●	Font Setting	Font	Selecting a character font
			Font Color	Selecting character color
●	●	Fill Setting	Fill Color	Selecting a color to fill the object
			Fill Color Transparent	Making the object transparent
●	●	Stroke Setting	Stroke Color	Selecting an outer frame color
			Stroke Color Transparent	Making the outer frame transparent
			Stroke Width	Selecting the outer frame thickness (from Thin/Normal/Bold)
–	●	Callout Setting	Selecting the shape (from Rectangle/Round Rectangle/Oval/Cloud).	
–	●	Anchor Point Move	Moving the callout tip.	

(4) Report

Click the Report button to save the image information edited in Image Editor in a pdf file.

(5) Page Selection button

When the Page Selection button is ON, the page numbers of the print area are displayed in the forefront. When a page number is selected, it turns slightly dark blue, indicating the target for output. If a page number is not selected, it turns light blue, indicating that it is not the target for output.

5.8.4 Adjusting PDF Output Area

You can use sliders in Image Editor to adjust the PDF output area: Horizontal slider to adjust the page width. Vertical slider to adjust the page height.

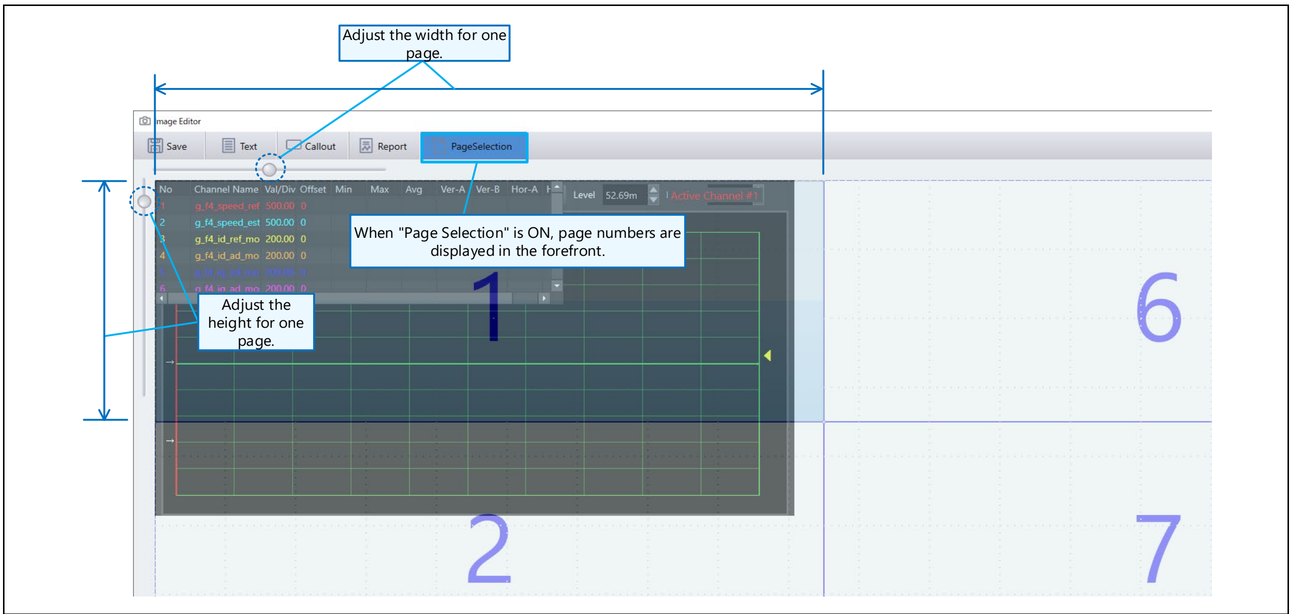


Figure 5-40 Adjusting PDF Output Area

6. [Analyzer] User Button Window

6.1 Overview

User Button Window provides a function to execute preregistered sequences sequentially by the user's instruction (by clicking the created button). User Button Window is displayed when you launch Analyzer tool.

6.2 Features

- Detailed sequences, such as for simultaneous execution, can be created.
- By using internal variables, loaded values can be written as is.

6.3 Screen Structure

Select "Add New" from the menu of User Button Control Window (or select "Show detail Setting" from the right-click menu of an existing button) to open a new User Button editing screen. The screen structure is shown below.

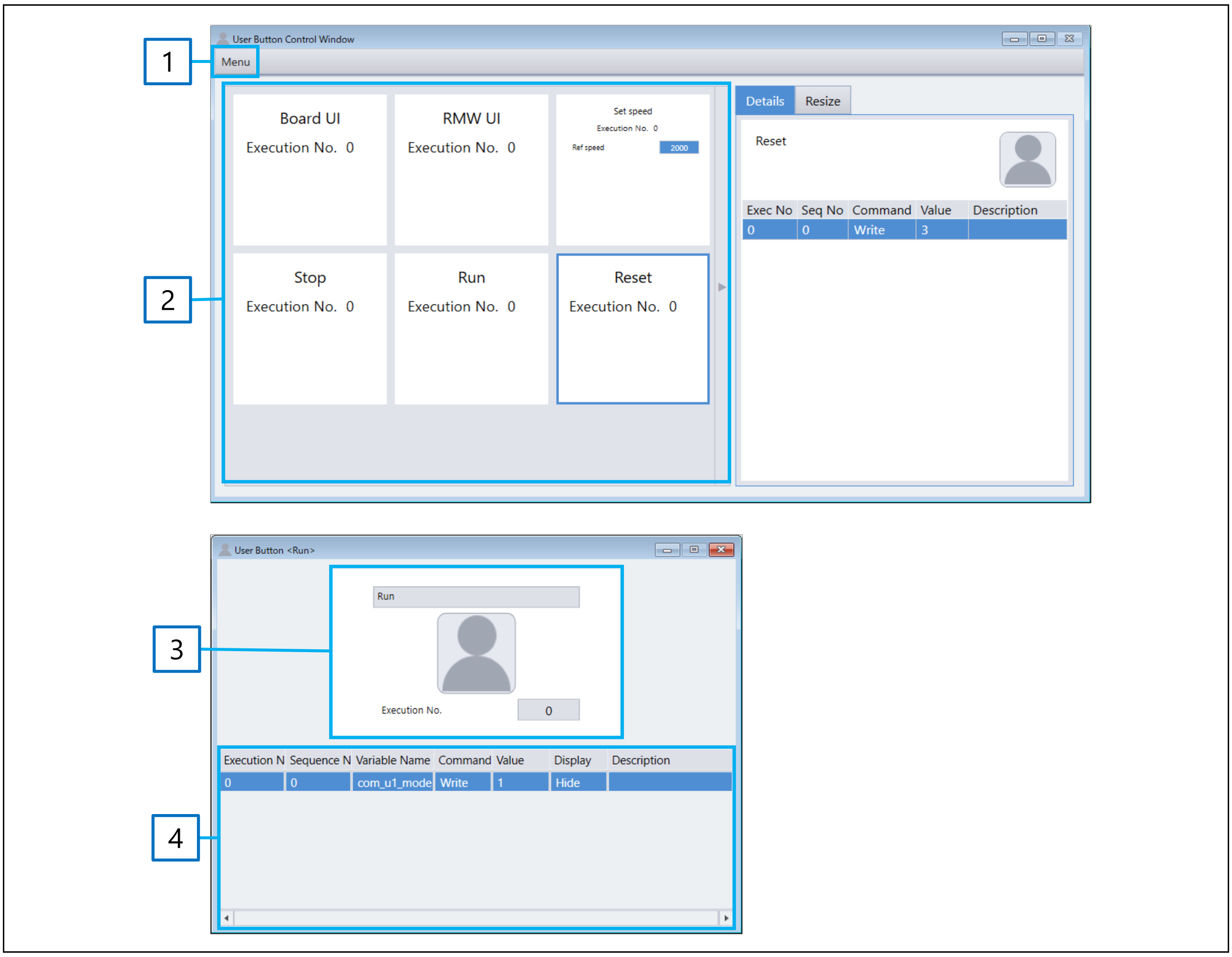


Figure 6-1 User Button Editing Screen

Table 6-1 Functions of User Button Editing Screen

No.	Name	Explanation
1	Menu	Select from Add New / Image Size / Show Control.
2	Button integration area	Multiple buttons are placed in one window.
3	Execution button area	Execution button. You can edit the name, number, and information to be displayed.
4	Sequence-editing area	The sequence to be executed each time the Execution button is clicked. You can edit it here.

6.3.1 Execution Button Area

The names and functions of each part in the execution button area are shown below.

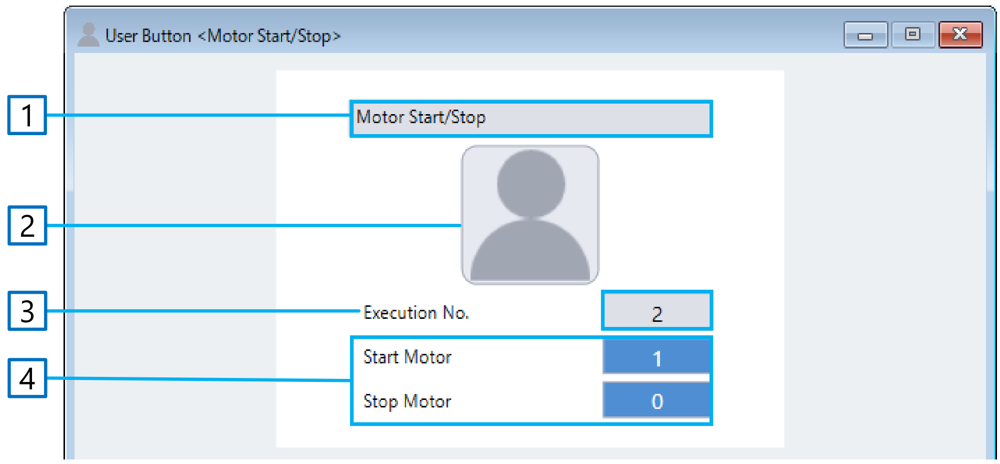


Figure 6-2 Execution Button View

Table 6-2 Function of Execution Button Area

No.	Name	Explanation
1	Button Name	• Any name can be specified. • The name is displayed in the User Button menu on Control Window
2	Execution button	• Clicking this button executes a single step of the sequence. • You can set up an image from the right-click menu.
3	Execution No	• Displays the Execution number to be executed when the Execution button is clicked. • When the Execution button is clicked, the Execution number automatically switches to the next one. When the number reaches the last, it returns to the first Execution number. • Execution numbers can be specified directly.
4	Information	• Displays Description and Value of the sequence information specified as Display=Show in the sequence-editing area. • The displayed Value can be edited.

6.3.2 Sequence-Editing Area

This section explains the names and functions of the various parts of the sequence-editing area.

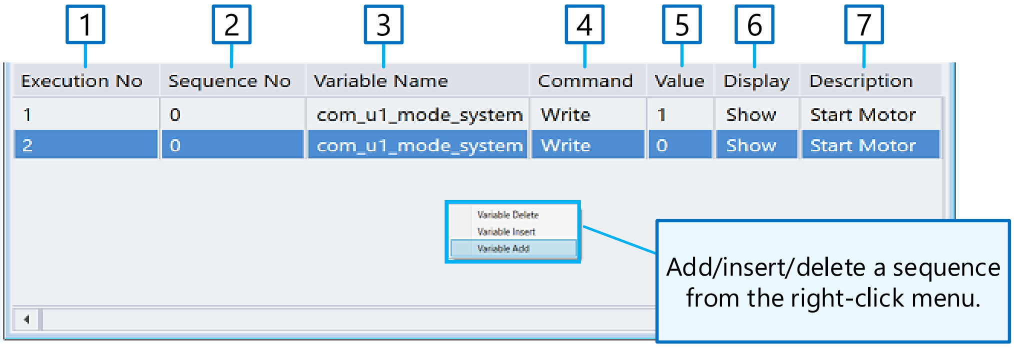


Figure 6-3 Sequence-Editing Area View

Table 6-3 Functions of Sequence-Editing Area

No.	Name	Explanation
1	Execution No	• Specify the processing order as an integer when the execution button is clicked. Execution proceeds sequentially starting with the smallest number.
2	Sequence No	• If there are duplicated Execution numbers, specify the Sequence No. (processing order) in this field as an integer. If there are duplicated Sequence numbers, execution is in the order of top to bottom.
3	Variable Name	• Specify variable names when values are to be loaded or written.
4	Command	Select commands: - Read/Write: Load or write variable values. - Run/Stop: Start or stop waveform display in Scope Window.
5	Value	• When “Commander=Read” is specified: the loaded value is displayed. • When “Commander=Write” is specified: the value to be written is specified. • Loaded values can also be held in an internal area and utilized as values to be written.
6	Display	• Specify whether to display (Show) or not display (Hide) information in the button’s information area.
7	Description	• Anything can be entered. • If “Display=Show” is specified, the entered description is displayed in the button area.

6.3.3 Button Integration Area

This section explains the names and functions of the parts of the button integration area.

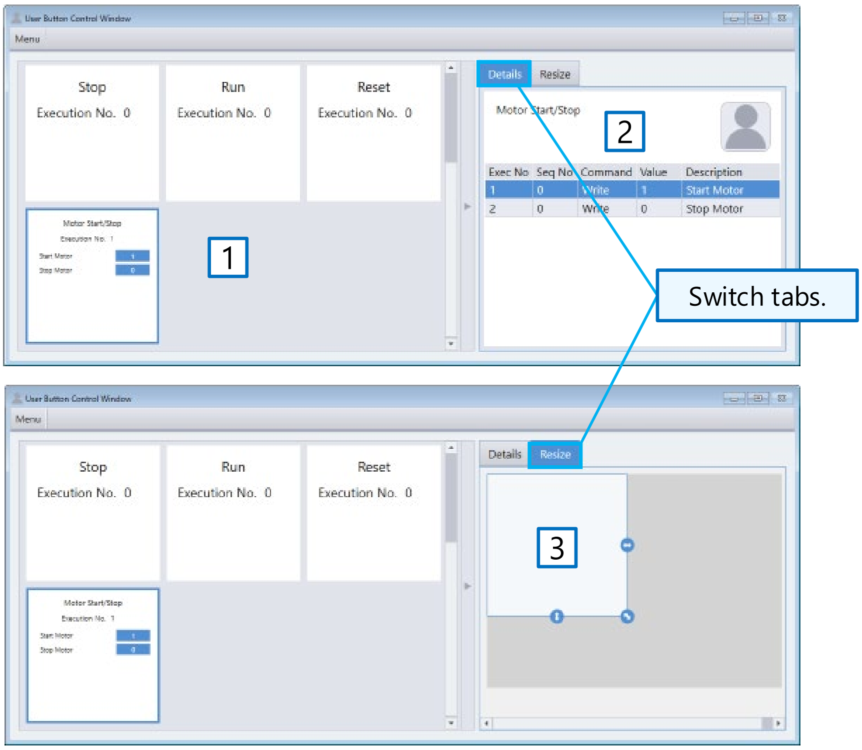


Figure 6-4 Button Integration Area View

Table 6-4 Functions of Button Integration Area

No.	Name	Explanation
1	Button area	• Buttons are arranged side by side. • You can rearrange the buttons freely by mouse-dragging
2	Button details tab	• You can check the sequence set to the button.
3	Button resize tab	• You can adjust the button size.

6.4 Explanation of Operation

6.4.1 Creating New User Button

When you select “Add New” from the Menu on User Button Control Window, a new User Button window will appear. You can create User Buttons up to 16.

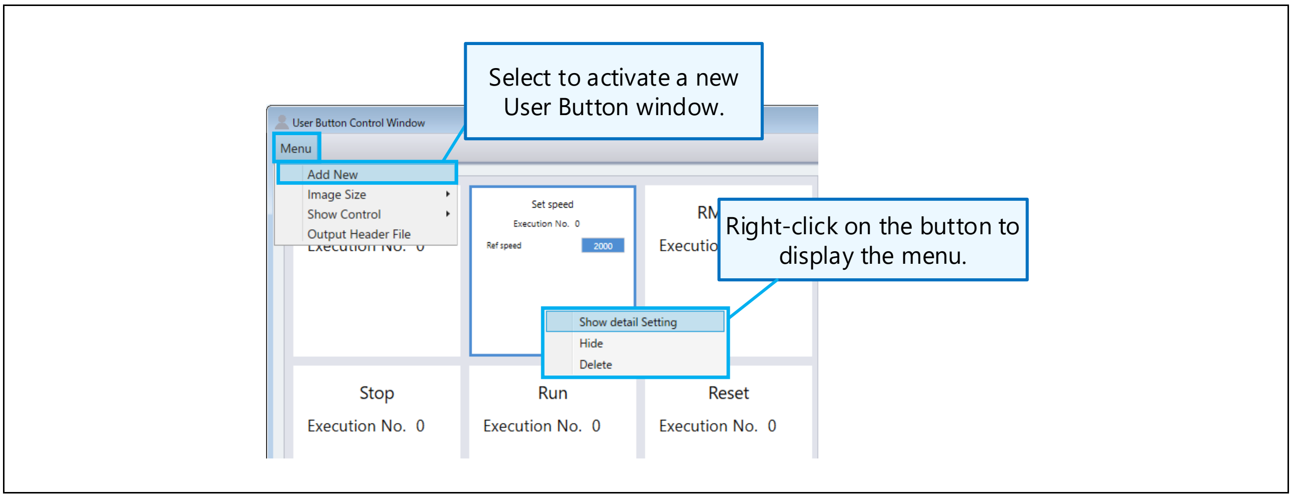


Figure 6-5 Creating New User Button

6.4.2 Delating User Button

Select “Delete” from the right-click menu of each button, and the User Button will be closed.

6.4.3 Showing and Hiding User Button

Select “Hide” from the right-click menu of each button, and the User Button screen will be hidden, but it will not be deleted.

You can also specify Show/Hide for each User Button by the checkbox of the button name list that is displayed when selecting Menu > Show Control of User Button Control Window.

6.4.4 Editing Sequence

6.4.4.1 Setting Up Sequence Rows

You can add, insert, and delete rows from the right-click menu in the sequence-editing area at the bottom of User Button window. However, right-clicking in the Variable Name column displays a menu for specifying variable names.

Table 6-5 Right-Click Menu on Sequence-Editing Area

Menu Item	Action
Variable Add	Adds a row below the last row
Variable Insert	Adds a row above the selected row
Variable Delete	Deletes the selected row

6.4.4.2 Specifying Execution No.

You can specify the execution order as an integer when the Execution button at the top of the User Button window is clicked. Execution proceeds sequentially starting with the smallest number (the specified numbers do not have to be contiguous). To process multiple rows with a single click of the Execution button, specify the same number in multiple rows.

Execution No	Sequence No	Variable Name	Command	Value	Display	Description
0	0	com_u1_mode_system	Write	1	Show	Start Motor 1st Time
1	0	com_u1_mode_system	Write	0	Show	Stopt Motor
2	0	com_u1_mode_system	Write	1	Show	Start Motor 2st Time
3	0	com_u1_mode_system	Write	0	Show	Stopt Motor

Click the Execution button

1st time : Executes the row with “Execution No=0”.

2nd time : Executes the row with “Execution No=1”

3rd time : Executes the row with “Execution No=2”

4th time: Executes the row with “Execution No=3”

5th time : (Repeat from the 1st time)

Figure 6-6 Execution No Setting

6.4.4.3 Specifying Sequence No.

If Execution numbers are duplicated within a sequence, you can specify the execution order within that Execution No. as an integer. Execution proceeds sequentially starting with the smallest number (the specified numbers do not have to be contiguous). You can specify the same number in multiple rows. In the following example, execution proceeds sequentially from top to bottom.

Execution No	Sequence No	Variable Name	Command	Value	Display	Description
0	0	com_u1_mode_system	Write	1	Show	Start Motor 1st Time
0	1	com_u1_mode_system	Write	0	Show	Stopt Motor
2	0	com_u1_mode_system	Write	1	Show	Start Motor 2st Time
2	1	com_u1_mode_system	Write	0	Show	Stopt Motor

Click the Execution button

1st time : Executes the row with “Execution No=0 and Sequence No=0”.

→ then executes the row with “Execution No=0 and Sequence No=1”.

2nd time : Executes the row with “Execution No=2 and Sequence No=0”.

→ then executes the row with “Execution No=2 and Sequence No=1”.

3rd time : (Repeat from the top.)

Figure 6-7 Sequence No Setting

6.4.4.4 Setting Variable

Specify a variable name in the Variable Name field in a sequence by one of the following operations.

(a) Enter variable name directly

When you enter a variable name partway, the potential candidate for the variable is displayed. (Note that only the first candidate in the variable list is displayed, and the list with narrowed candidates is not displayed.)

(b) Select from variable list

When you select a Variable Name cell, “v” is displayed to the right of the cell. Click this “v” to display the variable list, and you can select a variable name from it.

(c) Select using Variable Find function

Click a Variable Name cell once and right-click to display the menu. Select “Variable Find” from the menu, the Variable Find screen will be displayed. For the operation method, see Section 4.4.1.3 Select Variable Using Variable Find Function.

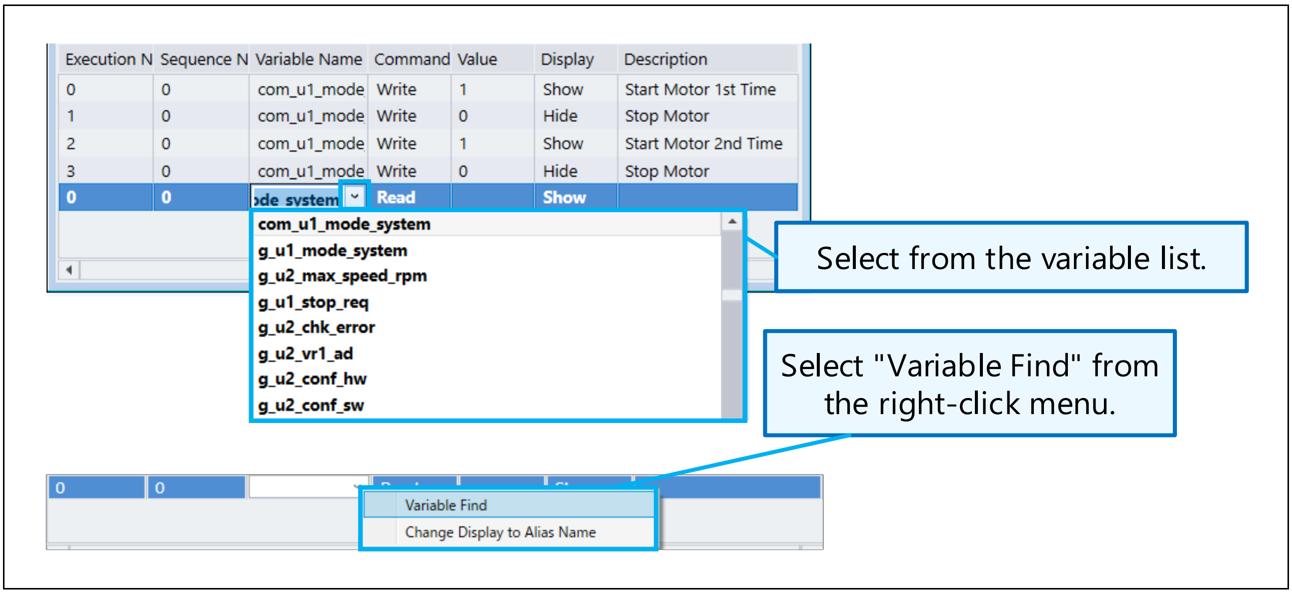


Figure 6-8 How to Input Variable Name

6.4.4.5 Setting Command

Select a Command cell in a sequence and click "v" to display the selectable list. You can select a command from the list.

Table 6-6 Command List

Command	Action
Read	Loads a variable value
Write	Writes a variable value
Run	Starts waveform display in Scope Window
Stop	Stops waveform display in Scope Window

6.4.4.6 Setting Value

When you specify “Read” in the Command field, the value of the loaded variable is displayed in the Value field.

When you specify “Write”, the value set in the Value field is written into the variable.

6.4.4.7 Setting Display

Select a Display cell in a sequence and click “v” to display a list, from which you can select one of the following.

- Show : Displays sequence information (Description and Value) at the top of the screen.
- Hide : Does not display sequence information (Description and Value).at the top of the screen.

6.4.4.8 Setting Description

When you specify “Show” in the Display field in a sequence, the information described in the “Description” field is displayed in the button area.

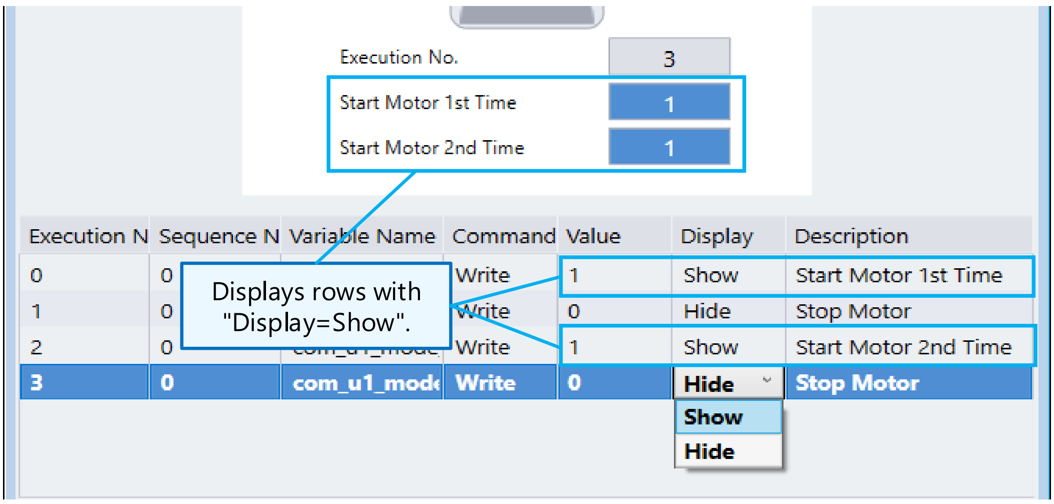


Figure 6-9 Description Setting

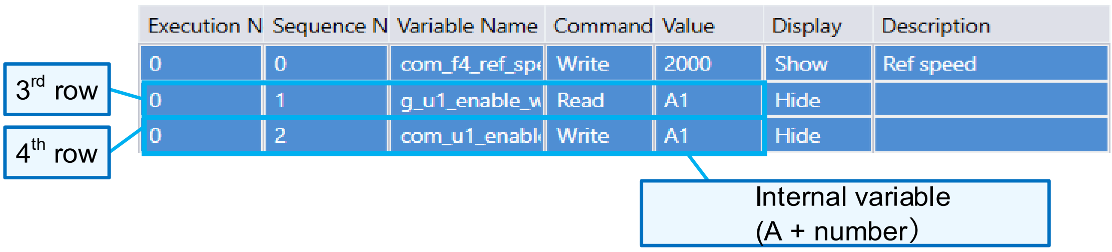
6.4.5 Utilizing Internal Variable

In User Button window, there are internal variables for User Button that can use the read values as values to be written. When a character string that starts with the letter “A” or “a” followed by a number or numbers is described in the Value field of the sequence (e.g., A1, a12345), that variable can be used as an internal variable.

Table 6-7 Command List When Utilizing Internal Variables

Command	Action
Read	Saves the value read from the variable in the internal variable
Write	Writes the value saved in the internal variable

The values of internal variables cannot be directly displayed or referenced. It is also not possible to use an internal variable across multiple User Button windows.



Example of using internal variables
3rd row : Reads the variable value and save it to the internal variable “A1”.
4th row : Writes the value of the internal variable “A1” to the variable.

Figure 6-10 Example of Using Internal Variables

6.4.6 Setting Display

6.4.6.1 Setting Image to Execution Button

Right-click on the button area at the top of the User Button window and select "Image" from the displayed menu. You can set an Image for the Execution button.

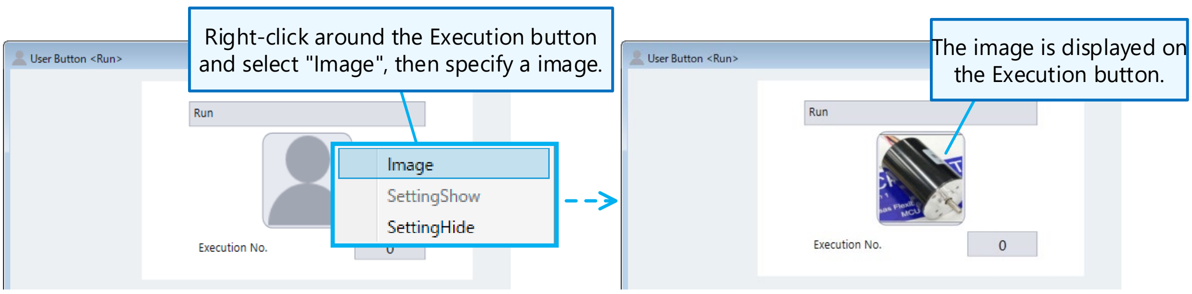


Figure 6-11 How to Set Image to Execution Button

6.4.6.2 Showing and Hiding Sequence-Editing Area

Right-click on the button area of User Button Window to display the menu and select “SettingShow” or “SettingHide” to switch the display of the sequence-editing area. (You must right-click on the button area.)

Table 6-8 Switching Sequence-Editing Area Display

Right-click menu	Action
Setting Show	Shows the sequence-editing area.
Setting Hide	Hides the sequence-editing area.

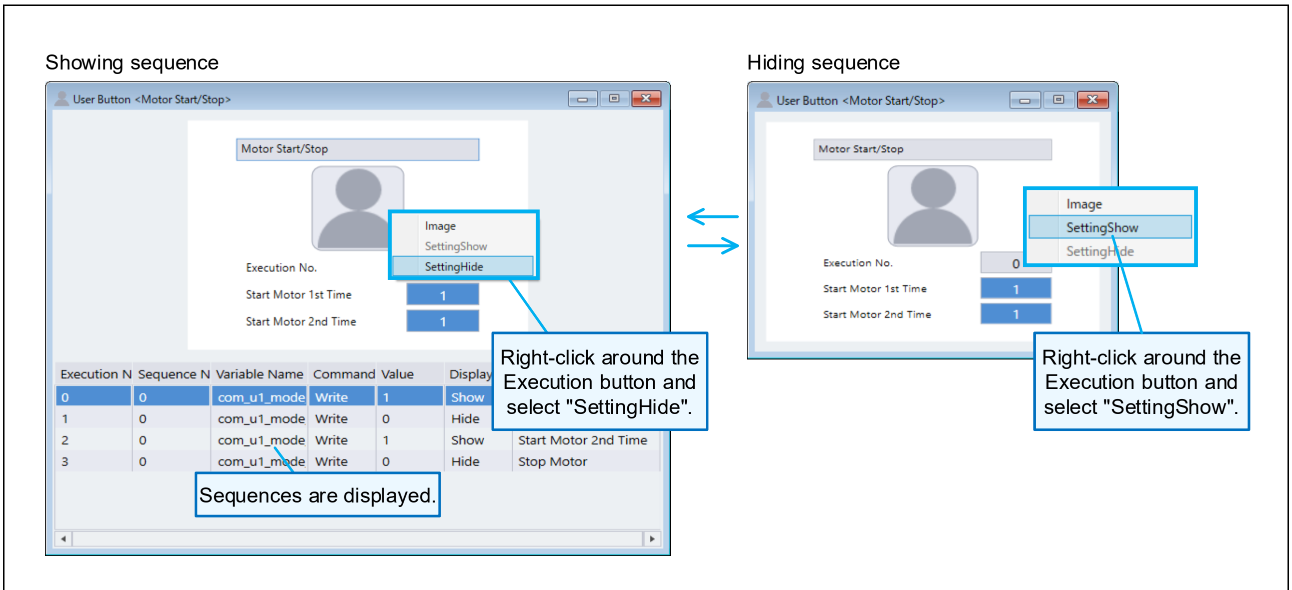


Figure 6-12 Showing/Hiding Sequence-Editing Area

7. [Analyzer] Commander Window

7.1 Overview

Commander is a function to preregister a sequence of instructions for writing data into variables, and then to execute the sequence continuously. Commander is activated with the “Commander” button in Analyzer’s Control Window.

7.2 Features

- Sequentially executes write operations.
- Intervals can be set.
- Repetitive (looped) processing can be performed.

7.3 Window Structure

The structure of Commander window is shown below.

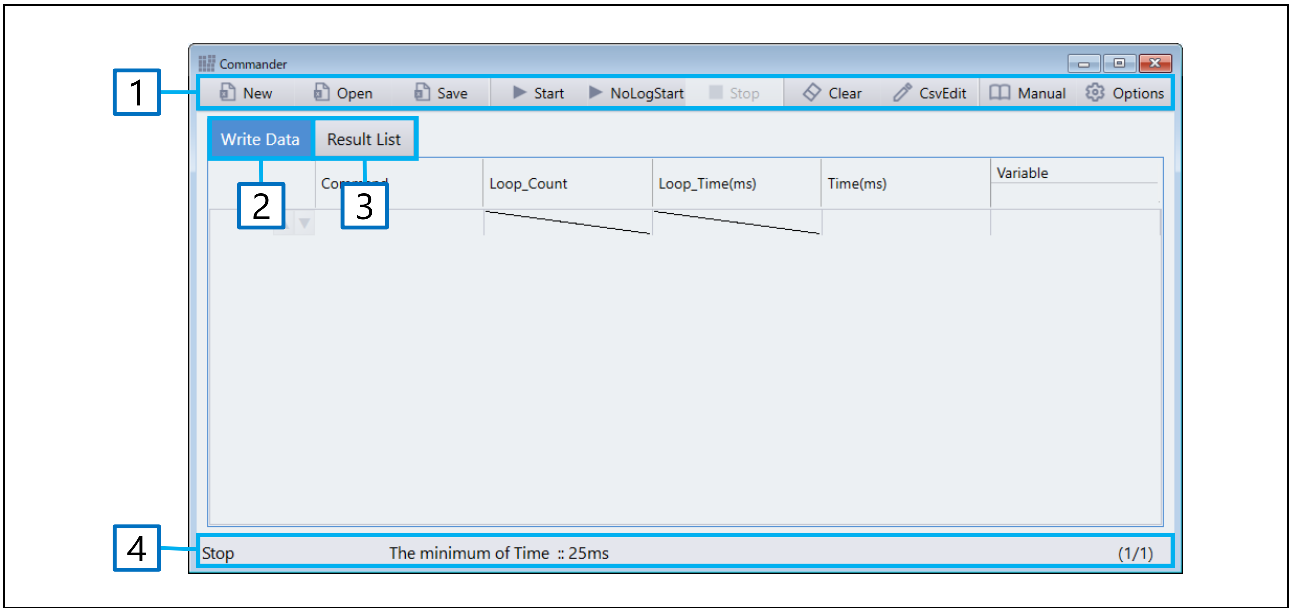


Figure 7-1 Commander Window

Table 7-1 Functions of Commander Window

No.	Name	Explanation
1	Operation button	By Clicking the buttons on the top of the window, you can read/write/edit a CSV file, import edited information, and run/stop the sequence, etc.
2	Write Data tab	Displays sequence information. You cannot edit a sequence on this window. (Use the CSV Edit button to edit a sequence.)
3	Result List tab	Displays a list of the sequence's execution results.
4	Status bar	Displays the sequence's execution status, Send Checker status, the number of commands, etc.

7.3.1 Operation Button

This section explains the operation buttons on the toolbar.

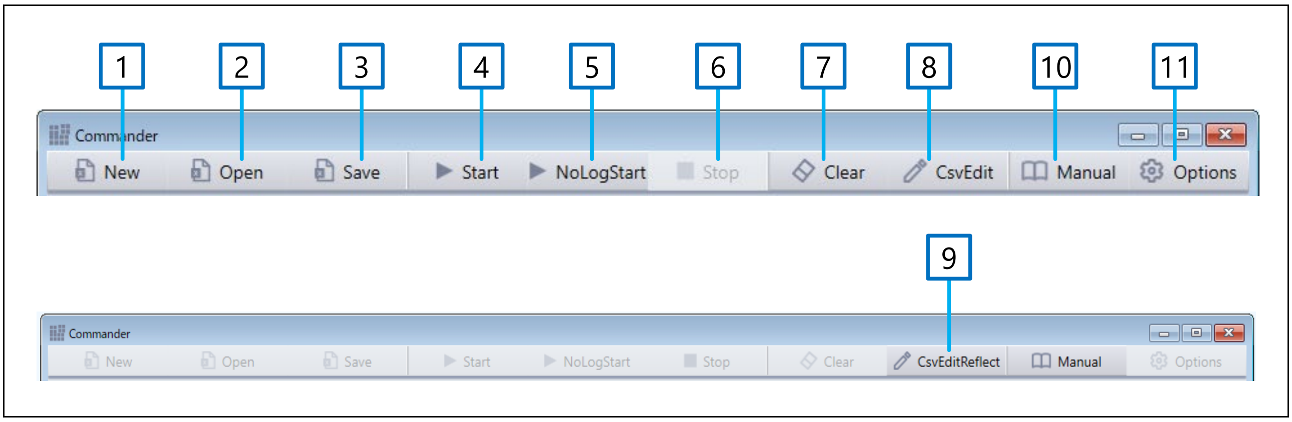


Figure 7-2 Toolbar

Table 7-2 Toolbar Operation Button Functions

No.	Name	Explanation
1	New	Creates a new CSV file and displays the default sequence information.
2	Open	Loads an existing CSV file and displays the sequence information.
3	Save	Saves the CSV file.
4	Start	Executes the sequence.
5	NoLogStart	Executes the sequence without getting the log.
6	Stop	Stops the sequence that is being executed.
7	Clear	Clears the sequence information that is being displayed.
8	CSV Edit	Activates the sequence editing screen and displays information about the sequence (CSV file) that is being displayed.
9	CSV Edit Reflect	The button's label changes when sequence editing is completed. Clicking the button displays information about the edited sequence (CSV file).
10	Manual	Displays the input guide
11	Options	Specify various settings.

7.3.2 Write Data Tab

When a CSV file is loaded, a sequence is displayed in the Write Data tab.

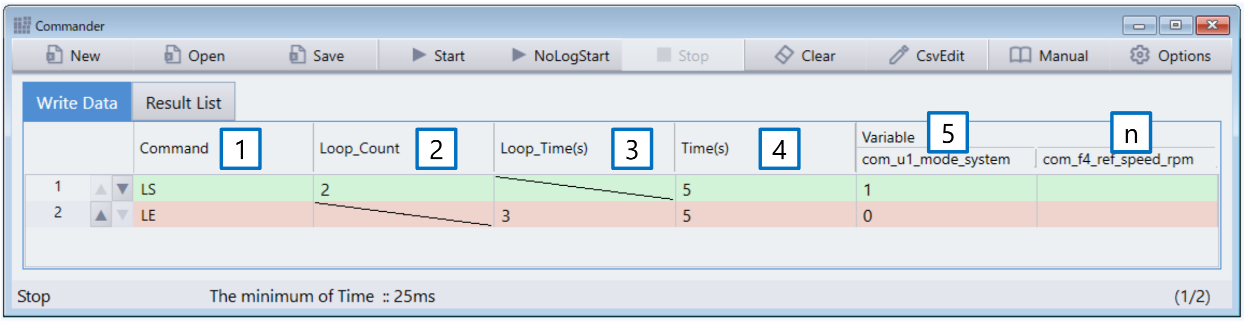


Figure 7-3 Write Data Tab View

Table 7-3 Functions of Write Data Tab

No.	Name	Explanation
1	Command	Displays the commands for looped processing: • LS: Loop start • LE: Loop end
2	Loop Count	Displays the number of loops.
3	Loop Time	Displays the interval between the execution of the command on the last line of looped processing and the return to the command on the first line.
4	Time	Displays each line's execution wait interval.
5	Variable Name	Header line: Displays the name of the variable name to issue a command. Lines below the header: Displays the value to be written into the variable in the header.
n	Variable Name	Displays variable names as the number specified in the sequence.

7.3.3 Result List Tab

When a sequence on the Write Data tab is executed, the results will be displayed on the Result List tab.

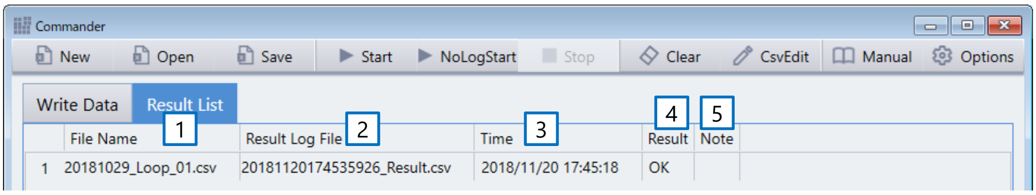


Figure 7-4 Result List Tab View

Table 7-4 Functions of Result List Tab

No.	Name	Explanation
1	File Name	Displays information about the CSV file for which the sequence was executed.
2	Result Log File	Displays information about the log file for the sequence's execution results.
3	Time	Displays the date and time the sequence was executed.
4	Result	Displays the sequence's execution result (OK/NG).
5	Note	Any comments can be entered

7.3.4 Status Bar

The status bar displays the status of Commander.

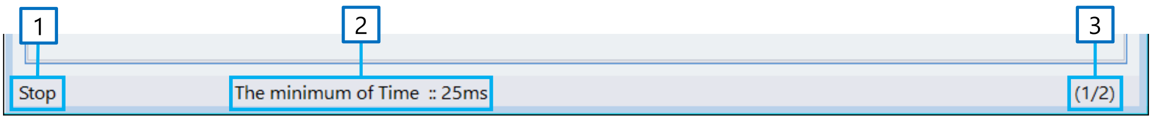


Figure 7-5 Status Bar

Table 7-5 Functions of Status Bar

No.	Name	Explanation
1	Execution status	Displays the execution status of the sequence. • Running : When a sequence is being executed by clicking Start button. • Stop : When the sequence is stopped by clicking Stop button, or the sequence execution is completed.
2	Send Checker information	Displays the Send Checker information. • Send Checker not executed: “Please press the Send Checker Button.” • Send Checker already executed: “The minimum of Time :: XXms” (XX: measured value)
3	Execution count	Displays (line-number/total-number-of-lines) for the sequence. • line-number: Displays the line number that has been selected by clicking, or the line number that is being executed. • total-number-of-lines: Displays the total number of instruction lines in the sequence.

7.4 Explanation of Operation

7.4.1 Starting and Exiting Commander

Click the “Commander” button of Control Window to launch Commander. Click the exit button at the upper-right corner to terminate it.

7.4.2 Executing Send Checker

When you start Commander, the Send Checker function is executed, and the system measures the minimum value of the interval of command value transmission. To operate Commander, it is necessary to execute Send Checker first.

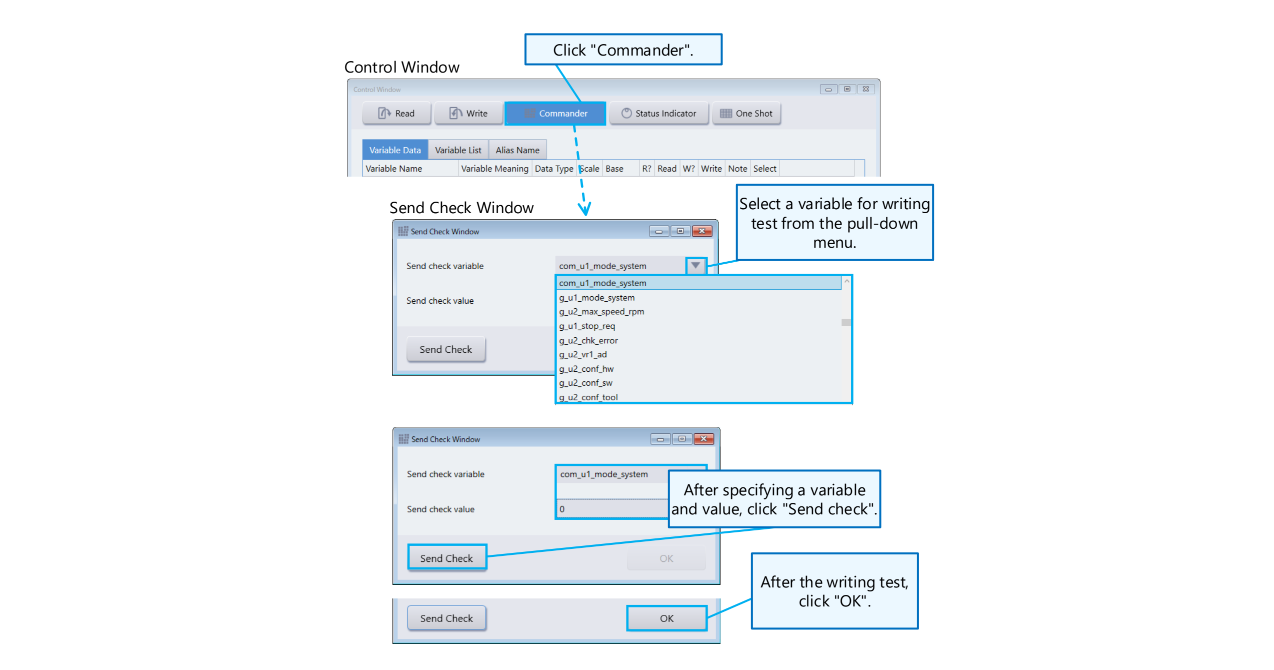


Figure 7-6 Starting Commander and Send Check Window

[Operation]

1. Set the following items on the Send Check Window
   - Send Check Variable : Select a variable that can be used in the communication test from the pull-down menu.
   - Send Check Value : Specify a value that can be used in the communication test.

Note: Since the value is actually written to the selected variable in the measurement, specify the variable name and its value that are safe for writing.

2. Press the "Send check" button to measure.
3. When execution of Send Checker is completed normally, the “OK” button can be pressed.

When Send Checker is completed, click "OK" to open Commander window. The measurement result of Send Checker is displayed in the status bar of Commander window.

- Display after Send Checker execution: “The minimum of Time :: XXms” (XX: measured value)

7.4.3 Specifying CSV File

7.4.3.1 Creating a New CSV File (“New” Button)

Click the “New” button of Commander window to create a new CSV file, and its information is displayed on Commander window.

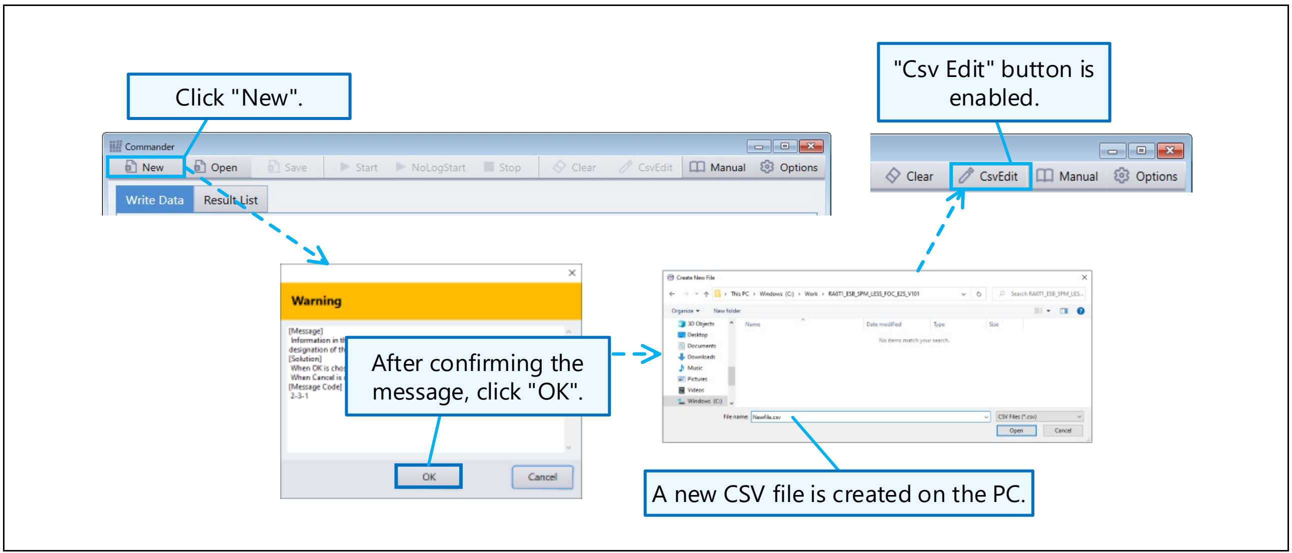


Figure 7-7 Creating New CSV File

［Operation］

1. Click the “New” button, the Warning message screen is displayed.
2. Message: “Emptied file when you open an existing file.” *Note
3. Click “OK” on the Warning message screen, and a screen to specify a new file name will be displayed.
4. Specify a new file name and click the “Open” button. A new CSV file is created, and the “CSV Edit” button is enabled.

Note If you specify an existing CSV file that contains sequence information in the filename specification, that CSV file will be initialized when it is opened. If you want to import an existing CSV file, use the “Open” described in the following.

7.4.3.2 Selecting Existing File (“Open” Button)

Click the “Open” button to load the CSV file that has been created and edited, and the sequence will be displayed on Commander window.

［Operation］

1. Click the “Open” button to display the screen for opening a file.
2. Specify an existing CSV filename and click the “Open” button. The file is loaded, and the sequence information is displayed on Commander window.

7.4.3.3 Clearing Displayed Sequence Information (“Clear” Button)

Click the “Clear” button to clear the sequence being displayed in Commander window.

7.4.4 Editing CSV File (“CSV Edit” Button)

When a sequence is displayed on Commander window, the “CSV Edit” button becomes enabled. Click the “CSV Edit” button to open the editing screen (Excel). The procedure for editing a sequence is explained below.

7.4.4.1 Basic Settings

(a) Specify Variable

For a new CSV file, the E1 cell displays "Please register available variable". Delete this text and specify a variable name.

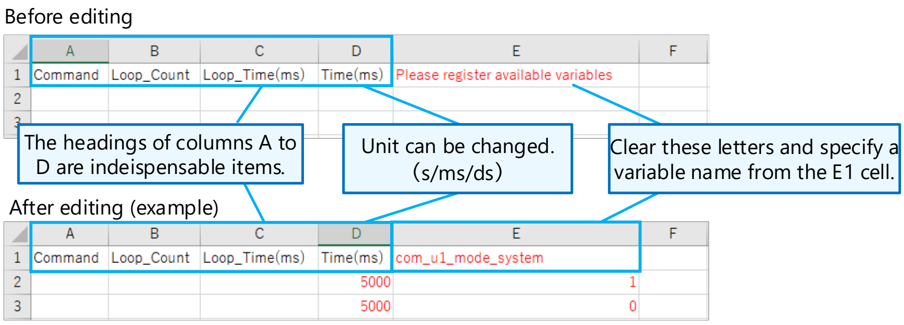
If you want to specify multiple variables per line (execute at the same timing), put these variables in the cell from E1 to the right side (F1, G1, H1...), without skipping a column.

(b) Specify processing interval

In the “Time(xx)” column, you can specify the processing interval. Positive integer/positive decimal values are valid.

(c) Specify command value

To issue commands, specify values in the second and subsequent lines in the column for variable names. Positive integer/positive decimal values are valid. If a command is issued to a large number of variable names in a single line (at the same timing), the next line may be issued before the processing is completed.



Example of editing sequence

(2nd line) Start driving the motor after 5 sec.

(3rd line) Stop driving the motor after 5 sec.

Figure 7-8 Write Data Editing

7.4.4.2 Specify Looping

(a) Specify looping

You can specify a loop operation in the “Command” column of the sequence. Specify "LS" in the loop starting line and "LE" in the ending line in uppercase characters. You can set multiple (nested) loops.

(b) Specify loops count

You can specify the loop count in the “Loop Count” column of the sequence. Specify the count in the line with Command "LS" specified. Positive integer values are valid.

(c) Specify loop interval

In the “Loop time” column of the sequence, you can specify the processing interval for returning from the “LE” (end) line process to the “LS” (start) line process in the loop operation.

Specify Loop time in the line with “LE” (end) specified. Valid values are positive integers and positive decimals.

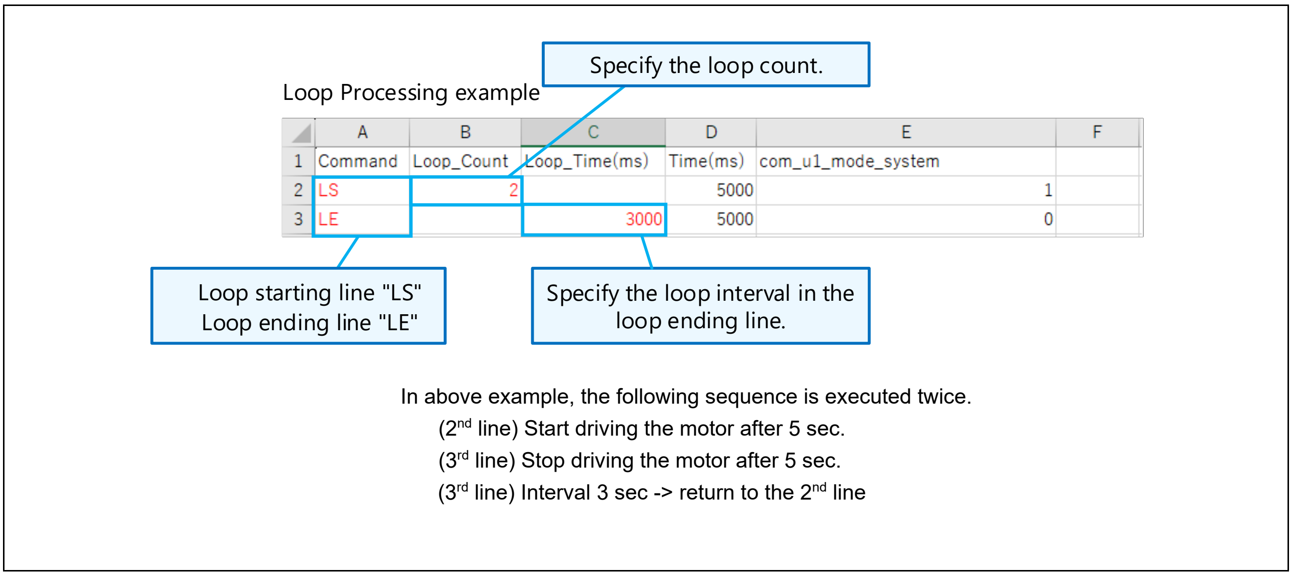


Figure 7-9 Setting Loop Processing Example 1

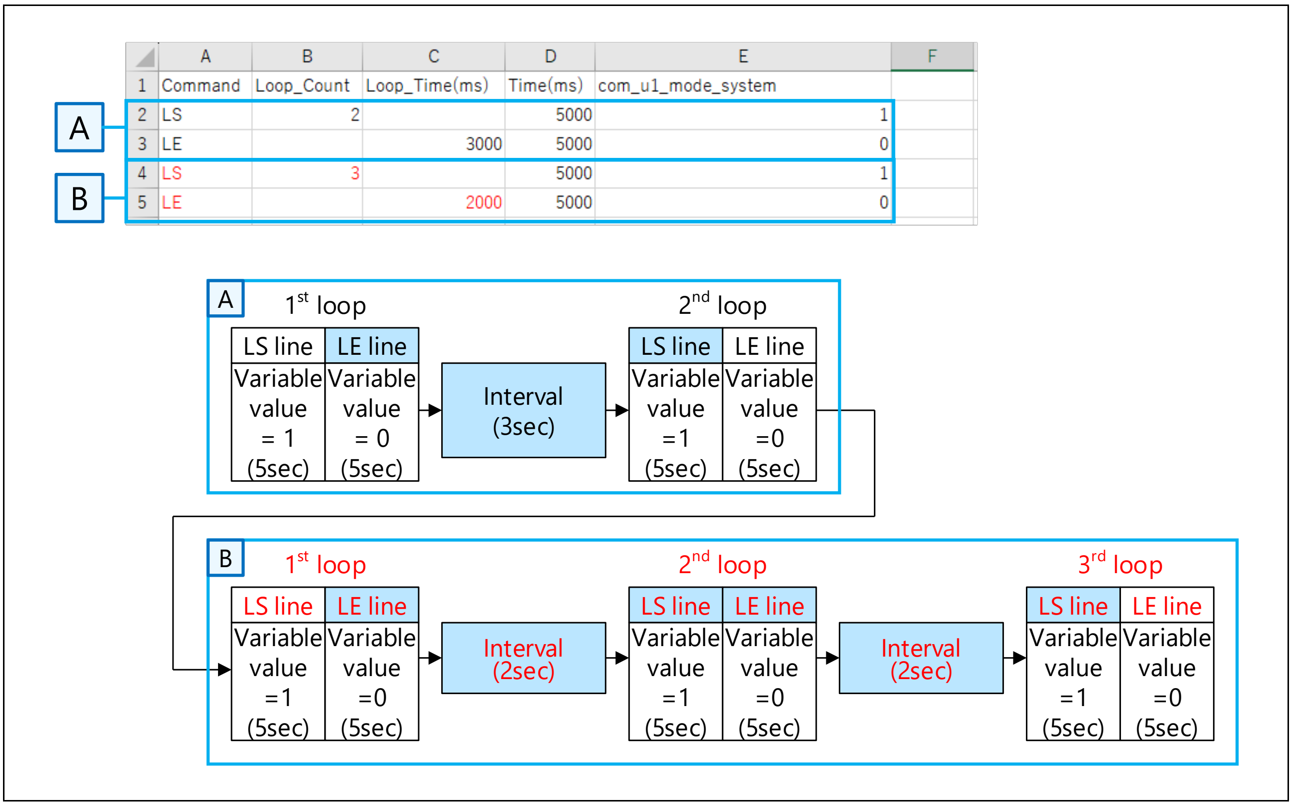


Figure 7-10 Setting Loop Processing Example 2

7.4.4.3 Specifying Time Unit

You can specify a sequence processing interval in the “Time” and “Loop Time” columns in the sequence.

The processing interval unit can be seconds (s), milliseconds (ms), or deciseconds (ds). You set the value for the unit to be used by modifying <unit> in “Time (<unit>)” and “Loop Time (<unit>)” of the header line.

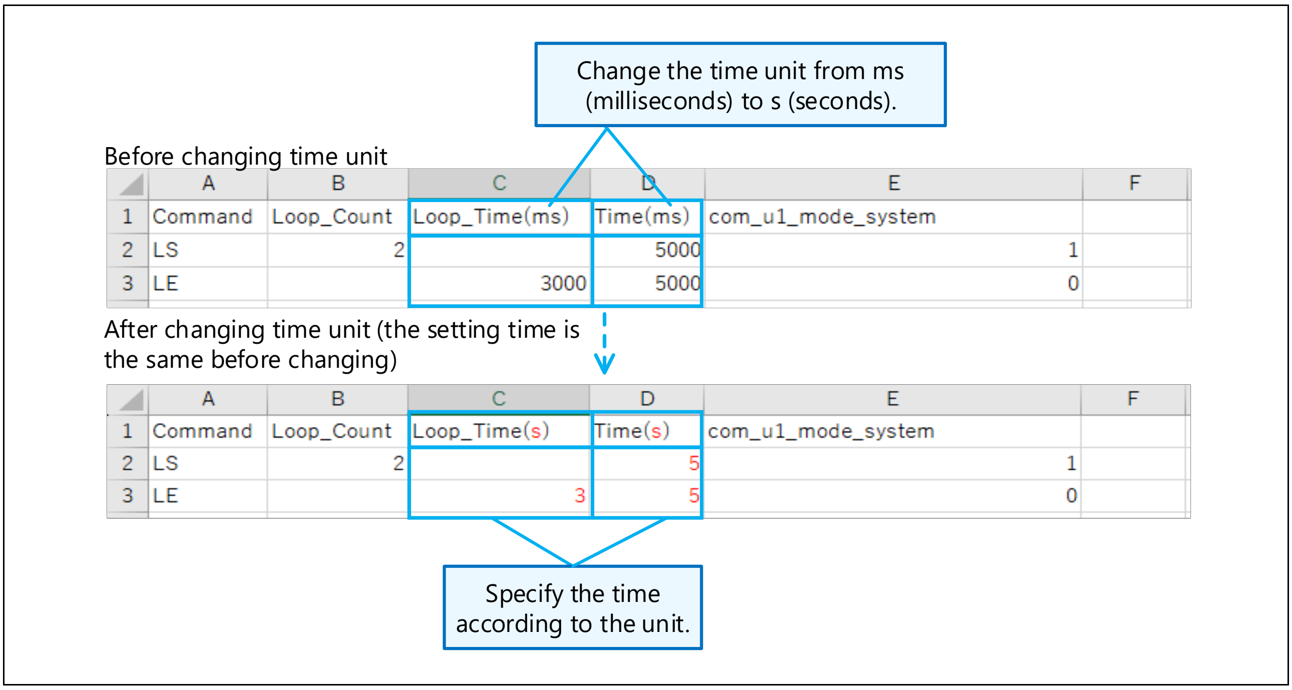


Figure 7-11 Modifying Time Unit

7.4.4.4 Terminating Editing and Reflecting

(a) Exiting editing screen

Save the edited sequence information into the CSV file by overwriting, then close the editing screen (CSV file). The label of the “Csv Edit” button switches to “CSV Edit Reflect”.

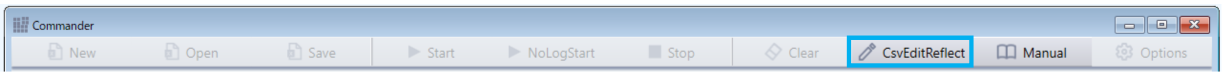


Figure 7-12 CSV Edit Reflect Button

(b) Reflecting the edited sequence information (“CSV Edit Reflect” button)

When you click the "Csv Edit Reflect" button*Note1, the syntax of the edited sequence information (CSV file) is checked, and a message is displayed if there is a problem. If there is no problem, the edited sequence information (CSV file) is reflected in Commander window*Note2, and the label of the button switches to "CSV Edit".

Figure 7-13 CSV Edit Button

Note 1. If you click the “Csv Edit Reflect” button without closing the screen in which you edited the CSV file, an error will result. Be sure to close the editing screen after saving the file.

2. If you finish editing a CSV file and save it with a filename that is different from the name loaded into Commander, the edited sequence information (CSV file) will not be reflected on the Commander window even when you click the “CSV Edit Reflect” button. In such a case, specify the name of the saved CSV file from the “Open” button on the Commander window to reflect the edited sequence information.

7.4.5 Preparing for Sequence Execution

Before executing the sequence, set the log saving of the sequence execution result. Click the "Options" button on the toolbar for setting.

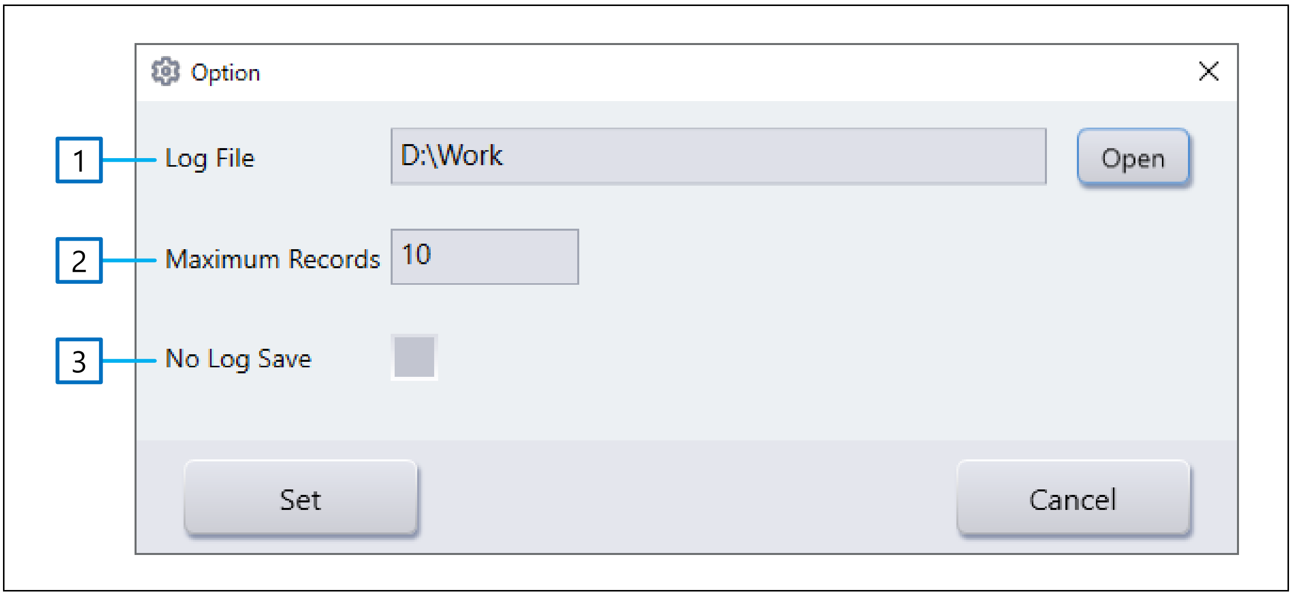


Figure 7-14 Option Settings Screen

Table 7-6 Option Setting

No.	Item name	Settings
1	Log File	Specify the output folder for the sequence execution result log. (Be sure to specify a user-accessible path.)
2	Maximum Records	Specify a maximum retentions count (MAX) for result records. The default value is set to 10.
3	No Log Save	If checked, the log will not be acquired.

The log file is created as shown below (it is not output when "No Log Save" in Option is checked or when sequence execution is started with the "NoLogStart" button).

- Log file output destination : Folders specified in Log File of Option
- Log file name :<CSV filename> + "_YYYYMMDDhhmmssxxx_Result.csv"

7.4.6 Executing Sequence

When sequence information (a CSV file) is loaded onto the Commander window, the “Start” and “NoLogStart” buttons become enabled on the Commander window.

Click the “Start” button to execute the sequence displayed in the Commander window. When you select a line of the displayed sequence and press “Start”, execution will start from the selected line.

If you click the "NoLogStart" button to execute the sequence, the log will not be acquired.

The sequence line being executed is displayed in a blue frame. When the sequence is executed to the end, execution stops automatically. Also, if the "Stop" button is clicked while a sequence is being executed, execution stops.

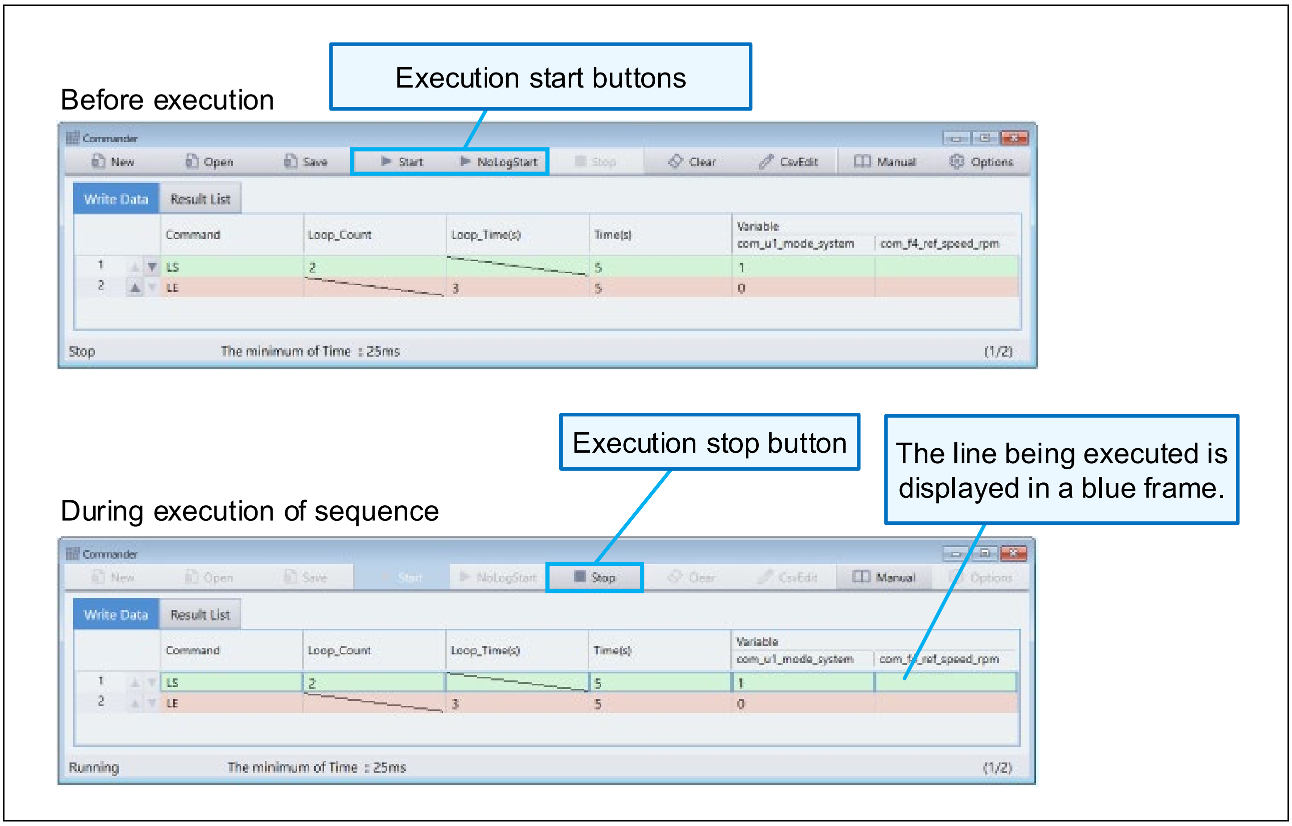


Figure 7-15 Executing Sequence

7.4.7 Result of Sequence Execution

7.4.7.1 Result List Tab

The sequence execution results are displayed in a list of result records on the Result List tab. The results are also output to a log file. (It is not output when the "No Log Save" checkbox in Option is checked, or when sequence execution is started with the "NoLogStart" button.)

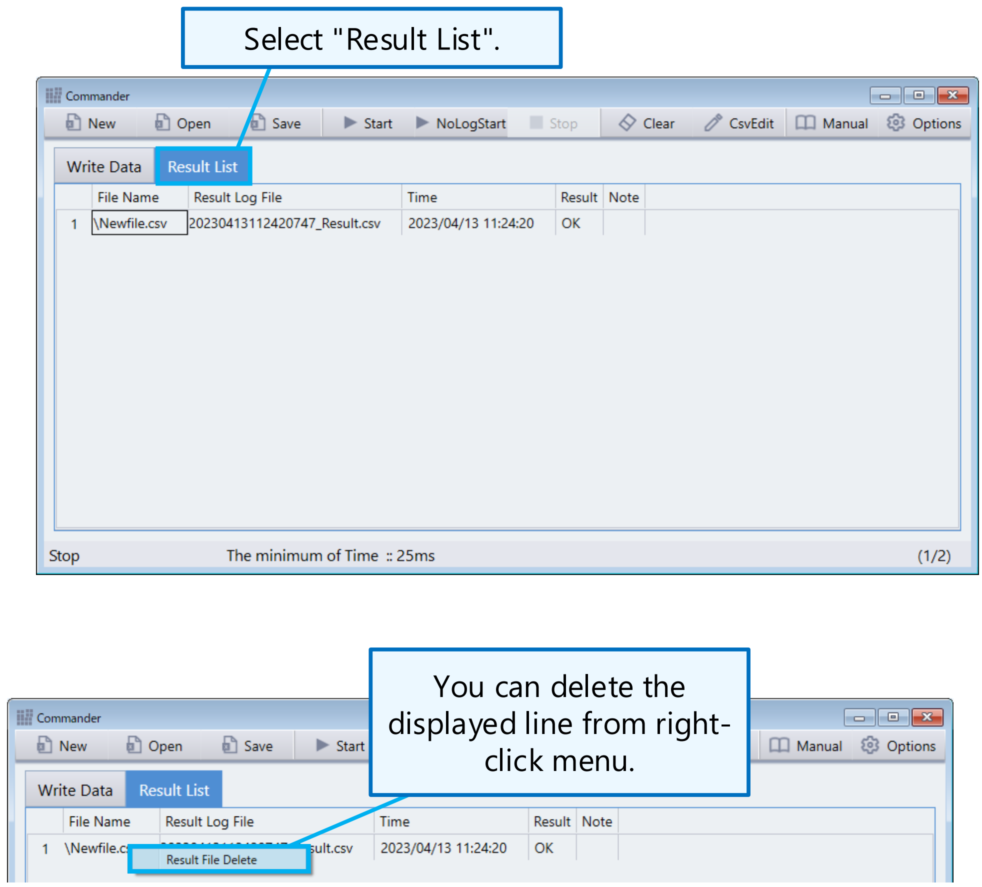


Figure 7-16 Result List View

7.4.7.2 Deleting Result Records

You can delete result records displayed on the Result List tab. Right-click the result record to be deleted and select “Result File Delete” from the displayed menu.

Note When you delete a result record from the Result List tab, the file displayed in Result Log File field (CSV file of the log) is also deleted from the PC.

7.4.7.3 Saving Result Record Information

The result record information displayed in the list on the Result List tab is saved in an RMT file. When an RMT file with saved result records is loaded, the result record information is restored.

8. [Analyzer] Status Indicator Window

8.1 Overview

Status Indicator is a function to monitor the values of the global variables of a user program. If a monitored result matches a preset condition (a threshold is exceeded), an indicator light on the window turns on. Press the "Status Indicator" button on the Analyzer's Control Window to launch Status Indicator.

8.2 Features

- Monitors variable values and indicates that a preset condition is met (a threshold is exceeded) with an indicator light color.
- Multiple monitoring conditions can be set for one variable.
- Multiple variables can be set to be monitored separately or simultaneously.
- When none of the monitoring conditions are met after start of monitoring, the indicator light turns green, and the status becomes "NORMAL".
- A monitoring results record (history of matching setting conditions) is displayed in a list.

8.3 Window Structure

The structure of Status Indicator window is shown below.

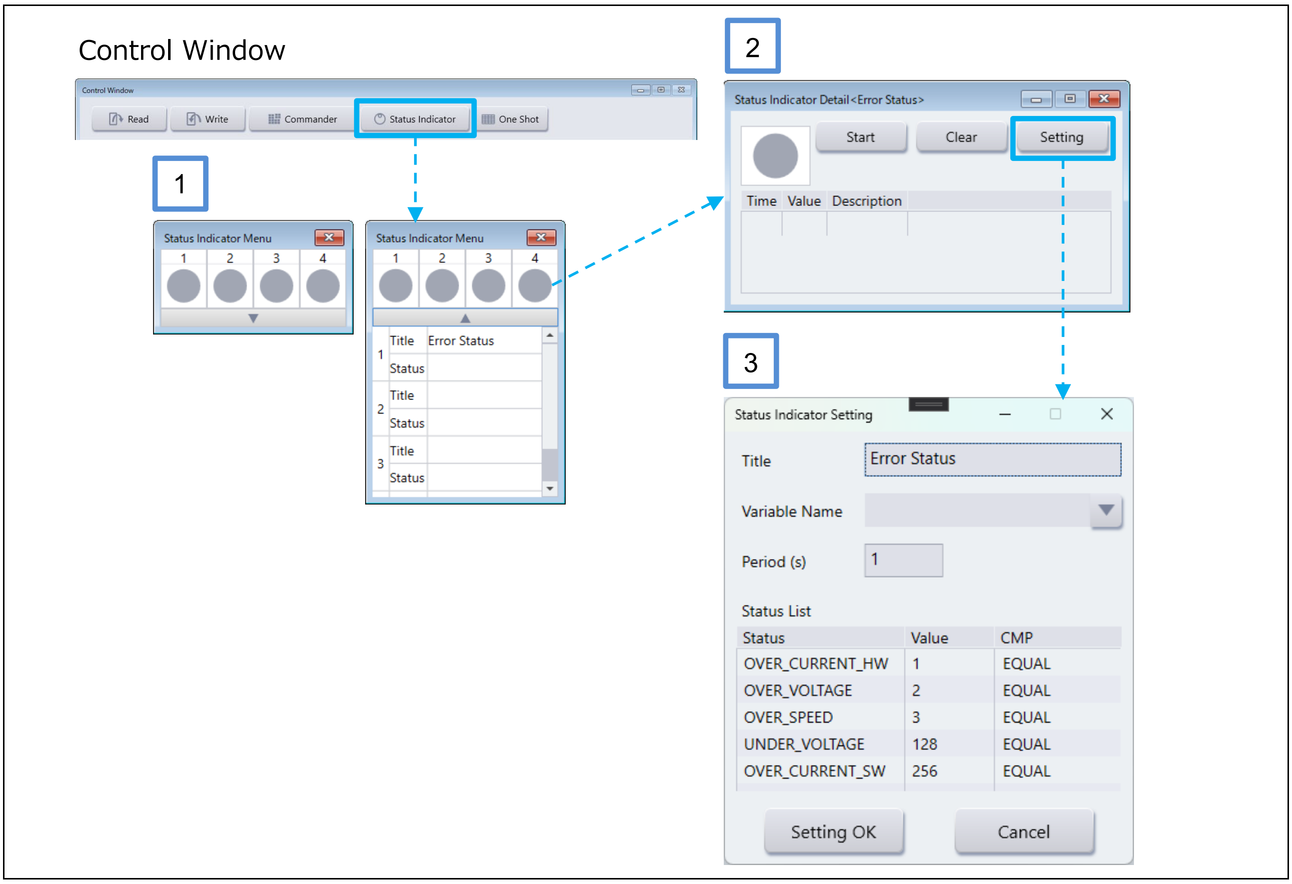


Figure 8-1 Status Indicator Window Structure

Table 8-1 Functions of Status Indicator Window

No.	Name	Explanation
1	Status Indicator Menu	Displays the indicator light that can be set.
2	Status Indicator Detail	Starts or stops monitoring and shows or clears the monitoring result.
3	Status Indicator Setting	Specifies monitoring conditions.

8.3.1 Status Indicator Menu

The Status Indicator Menu screen (hereafter, "Menu screen") starts when the "Status Indicator" button on the Control Window is clicked. You can check each monitoring status from the indicator lights on the Menu screen.

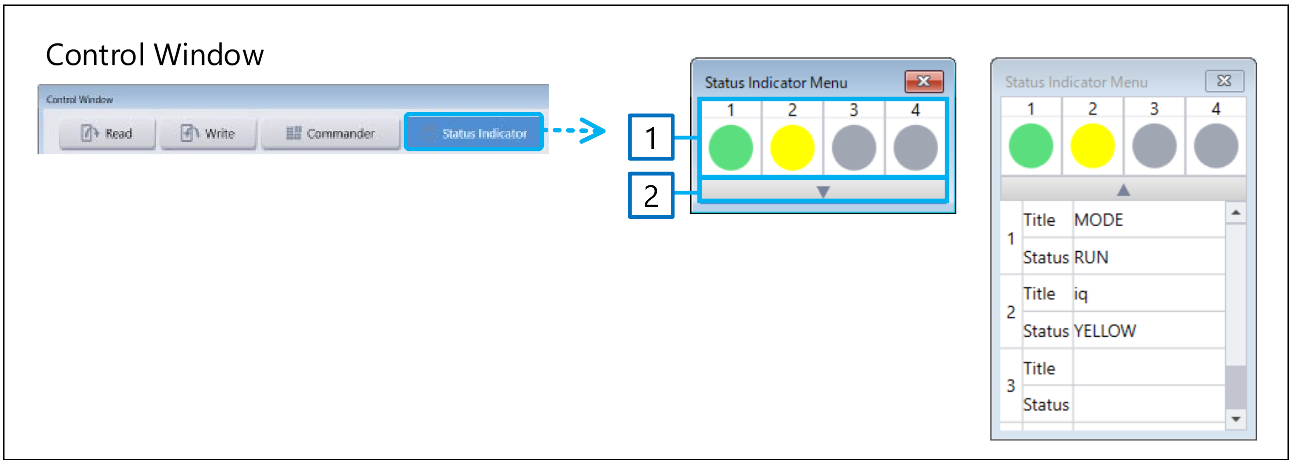


Figure 8-2 Status Indicator Menu Screen

Table 8-2 Functions of Status Indicator Menu Screen

No.	Name	Explanation
1	Indicator light	Indicates the monitoring status with its color.
2	Information display button	Displays the Title and Status of the indicator lights (click ▼ to display, ▲ to hide).

8.3.2 Status Indicator Detail

Click an indicator light on the Menu screen, and a Status Indicator Detail screen (hereinafter referred to as Detail screen) of that indicator will be displayed. In the Detail screen, you can start/stop monitoring and clear the log.

When you click the close button of the screen, the Detail screen will be closed regardless of the monitoring status.

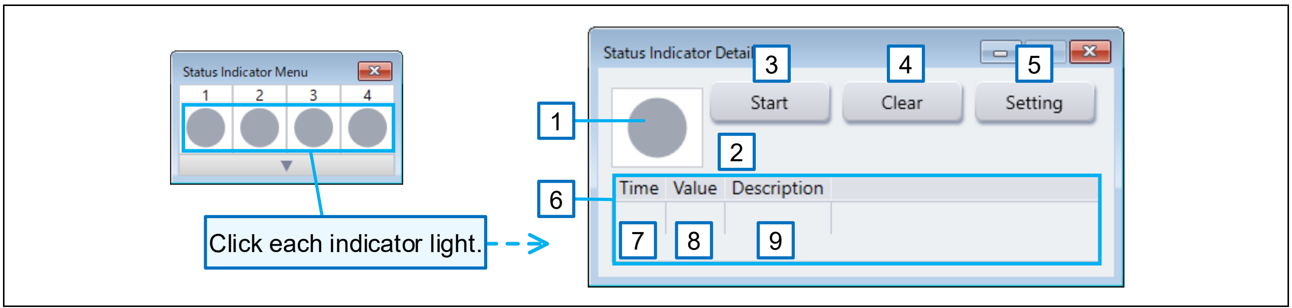


Figure 8-3 Status Indicator Detail Screen

Table 8-3 Functions of Status Indicator Detail Screen

No.	Name	Explanation
1	Indicator light	Indicate the monitoring status with its color
2	Status	Indicates the monitoring status in text.
3	Start / Stop button	Starts or stops monitoring.
4	Clear button	Clears the monitoring result list.
5	Setting button	When this button is clicked while monitoring is stopped, the Status Indicator Settings screen is displayed. (It can not be displayed during monitoring.)
6	Monitoring result list	Lists the information about monitoring conditions that were met (the thresholds were exceeded) during monitoring.
7	Time	Displays the date and time a monitoring condition was met (the threshold was exceeded).
8	Value	Displays the value that matched the monitoring condition (the threshold was exceeded).
9	Description	Displays the status that matched the monitoring condition (the threshold was exceeded).

8.3.3 Status Indicator Setting

Click the “Setting” button on the Detail screen, and the Status Indicator Settings screen (hereafter, “Settings screen”) will be displayed. You can specify detailed monitoring conditions on this screen.

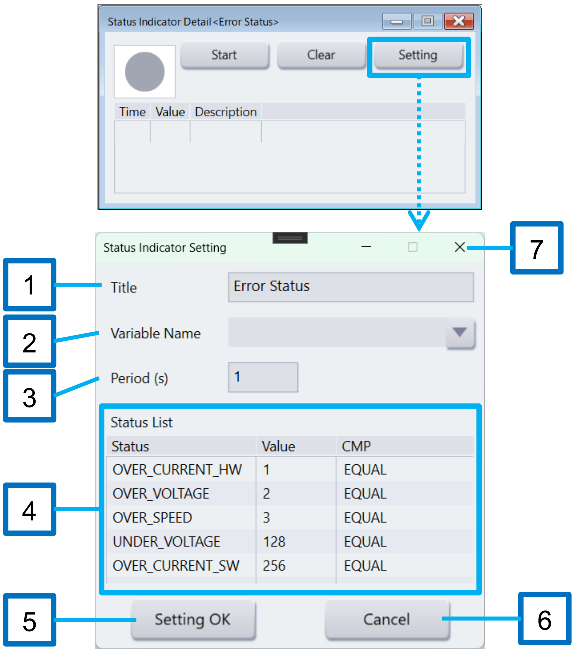


Figure 8-4 Status Indicator Settings Screen

Table 8-4 Functions of Status Indicator Settings Screen

No.	Name	Explanation
1	Title	Specify the name of the monitoring.
2	Variable Name	Specify the name of a variable to be monitored.
3	Periodic	Specify the monitoring interval (the default value is 0 seconds).
4	Status List	Specify the monitoring conditions.
5	Setting OK	Save setting and close window.
6	Cancel	Close window without saving setting.
7	Close button	Close window without saving setting.

8.4 Explanation of Operation

8.4.1 Showing or Hiding Menu Screen

Click the “Status Indicator” button on Control Window, the Menu screen will be displayed.

Click the Exit button at the top right corner of the Menu screen, the Menu screen (and the Detail screen) will be hidden. Even if you click the Exit button on the Menu screen and hide the screen, the monitoring status is retained during monitoring.

8.4.2 Setting Screen (Setting Monitoring Condition)

Select an indicator light from the Menu screen, the Detail screen will be displayed. Then click the Settings button on the Detail screen, and the Settings screen will be displayed. You cannot edit the Settings screen during Status Indicator operation.

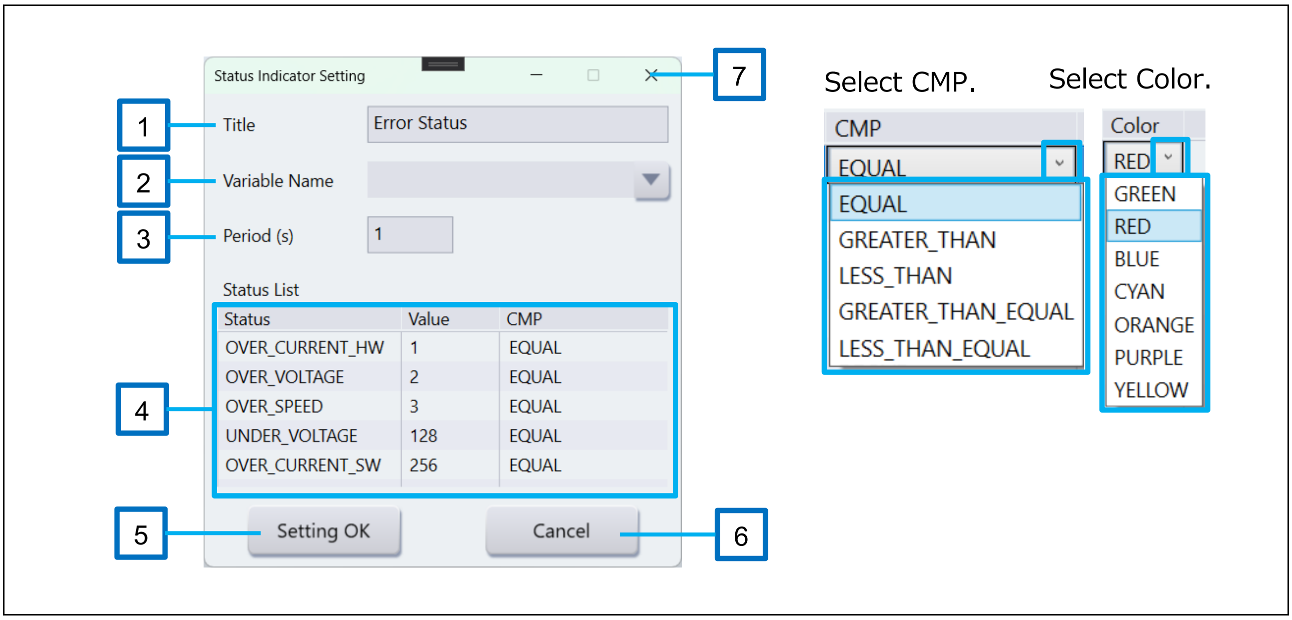


Figure 8-5 Monitoring Settings Screen

(1) Title

Specify a name for the monitoring settings (direct entry).

(2) Variable Name

Specify the name of the variable to be monitored (direct entry, select from the list, or using the Variable Find function).

(3) Periodic

Specify the monitoring interval (direct entry).

(4) Status List

Set monitoring conditions. After editing the last line, another line is added by press the Enter key. You cannot insert a line to the middle of them. To delete a line, press the Delete key when the entire line is selected (turned blue.) (The last line remains blank but does not affect the operation).

- Status : Specify a name for a condition (direct entry).
- Value : Specify a condition value (direct entry).
- CMP : Specify a judgment condition (select from the list).
- Color : Specify a color (select from the list).

Note that when setting a color, consider that the indicator lights is green when none of the monitoring conditions are met.

(5) Setting OK

Save setting and close window.

(6) Cancel

Close window without saving setting.

(7) Close button

Close window without saving setting.

8.4.3 Starting Monitoring

The indicator can start monitoring individually. When the "Start" button is clicked on the Detail screen of the indicator light, monitoring starts. When monitoring starts, the colors of the indicator lights on Menu and Detail screens change to the colors set in the Settings screen according to the conditions. If none of the monitoring conditions are met, the indicator light turns green, and the Status turns NORMAL.

8.4.4 Stopping Monitoring

To stop monitoring, click the “Stop” button on the Detail screen under monitoring. The indicator light turns gray when monitoring stops.

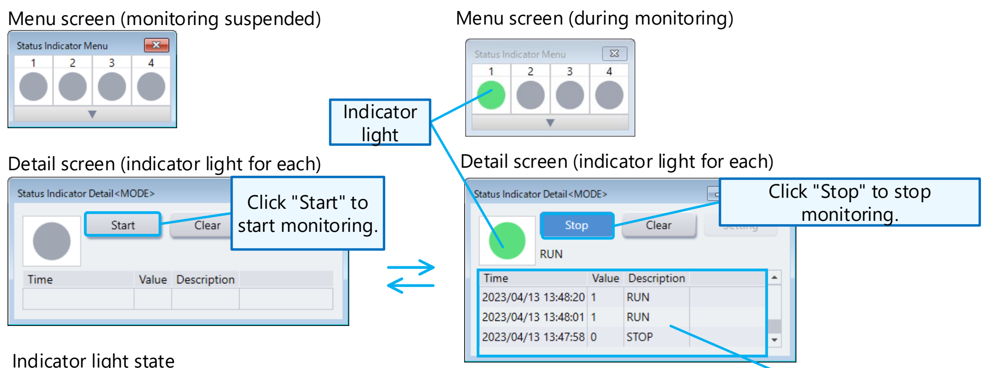
8.4.5 Display During Monitoring

If there are no matching monitoring conditions during monitoring, the indicator light turns green and the Status turns “NORMAL”. When the value of the variable meets the monitoring condition (exceeds the threshold value), the following operation is performed.

- The colors of the indicators on the Menu and Detail screens turn the colors set in the monitoring conditions.
- The information about the matched monitoring condition (threshold exceedance) is added to the monitoring result list on the Detail screen.

In the monitoring result list on the Detail screen, the latest information is added at the top of the list (it does not scroll automatically).

Even if you close the Menu or Detail screen, the monitoring status is retained (continued).



Indicator light state

Monitoring state		Indicator light	Status display
Monitoring suspended		(Gray)	(Blank)
During monitoring	No conditions are met.	Green	NORMAL
	Matching a condition (exceeding a threshold).	Specified color	Specified status

Figure 8-6 Starting/Stopping Monitoring

8.4.6 Clearing Monitoring Results

The monitoring result list in Detail window can be cleared by clicking "Clear" in Detail window. You can click the Clear button regardless of whether monitoring is being performed or has stopped.

8.4.7 Disabled Operations During Monitoring

The operations listed below are disabled during monitoring. To use any of these operations, you must first stop monitoring all.

- Terminating RMW
- Loading an RMT file (Open RMT File)
- Loading a Map file (Load Variable Data)
- Switching tools

If the above operations are not possible, check if an indicator light with the Detail screen closed or an unused indicator light is under monitoring.

8.4.8 Saving to RMT File

Status Indicator setting information is saved in an RMT file. However, the monitoring result list cannot be saved.

9. ［Analyzer］ One Shot Window

9.1 Overview

One Shot is a function to collectively acquire data of consecutive addresses from specified variables and display them in a waveform. One Shot is activated by clicking the “One Shot” button on Analyzer’s Control Window.

9.2 Features

- The buffered data can be displayed as a waveform

9.3 Window Structure

The structure of One Shot window is shown below.

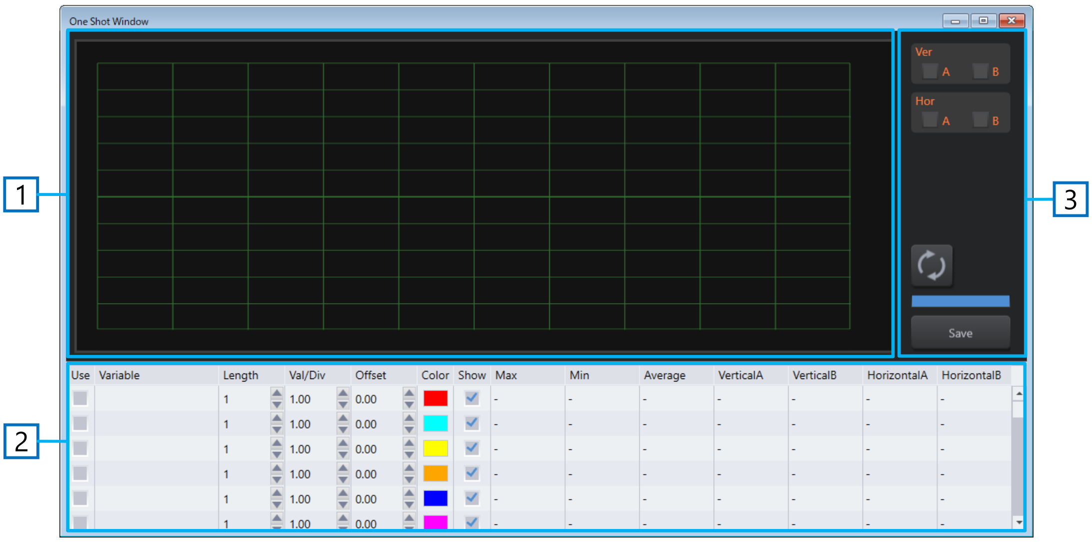


Figure 9-1 One Shot Window

Table 9-1 Functions of One Shot Window

No.	Name	Explanation
1	Waveform display area	Displays the acquired data as a waveform.
2	Channel information area	Sets and displays the channel information
3	Acquisition settings area	Sets the cursor and start acquisition (reload), etc.

9.3.1 Channel Information Area

The structure of the channel information is shown below.

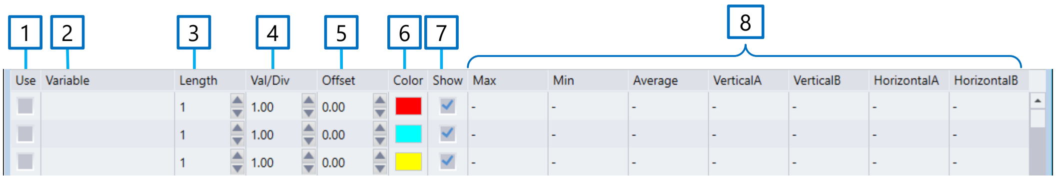


Figure 9-2 Channel Information Area

Table 9-2 Functions of Channel Information Area

No.	Name	Explanation
1	Use	Select the Channel to acquire the waveform
2	Variable	Specify the variable to acquire the waveform.
3	Length	Specify the number of data to acquire the waveform.
4	Val/Div	Specify 1 div of the vertical axis the waveform display.
5	Offset	Specify the vertical axis offset value for the waveform display.
6	Color	Select the color of the waveform to be displayed.
7	Show	If checked, the waveform of that channel will be displayed.
8	Information area	Displays waveform information.

9.3.2 Acquisition Setting Area

The structure of the acquisition settings area is shown below.

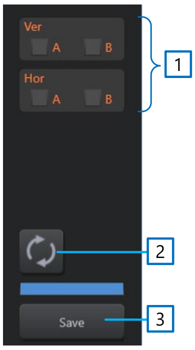


Figure 9-3 Acquisition Settings Area

Table 9-3 Functions of Acquisition Settings Area

No.	Name	Explanation
1	Cursor settings	Sets Show/Hide of the vertical cursor (Ver) and the horizontal cursor (Hor).
2	Graph Reload button	New data is acquired each time you click.
3	Save button	Saves the acquired data.

9.4 Explanation of Operation

9.4.1 Displaying One Shot Window

One Shot Window is activated by clicking the “One Shot” button on Control Window.

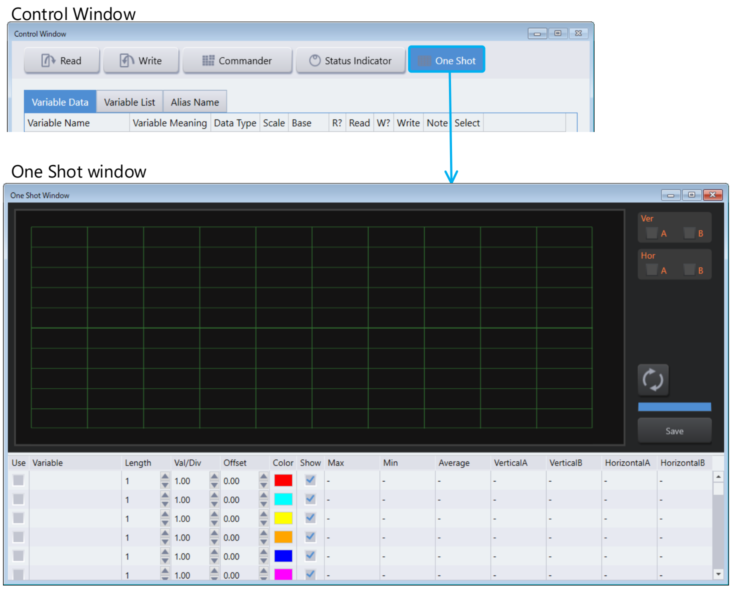


Figure 9-4 Launching One Shot Window

9.4.2 Settings of Channel Settings Area

The following shows the settings of the channel information area on One Shot window.

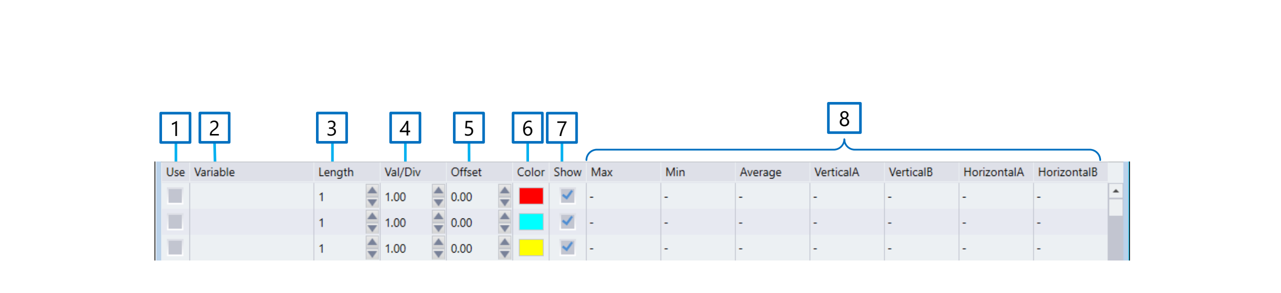


Figure 9-5 Channel Information Area

(1) Use

When this checkbox is ON, information about that channel is acquired.

(2) Variable

Specify the variable name of the start address of the data to be acquired.

(3) Length

Specify the number of data to be acquired.

(4) Val/Div

Enter the 1 dive value of the vertical axis for displaying the data as a waveform. You can also change it with ▼▲ on the right side of the input box.

(5) Offset

Enter the value of the vertical axis offset for displaying the data as a waveform. You can also change it with ▼▲ on the right side of the input box.

(6) Color

Select the color of the waveform display.

(7) Show

When this checkbox is ON, the waveform is displayed.

(8) Information aera

Displays information of the acquired data (maximum, minimum, average, and values at each cursor position).

9.4.3 Data Acquisition

To acquire the data, click the “Graph Reload” button on the acquisition settings area. (The data is updated each time you click.)

9.4.4 Cursor Settings

Turn ON the checkbox in Cursor at the top-right of Scope Window to show two Hor (horizontal) and Ver (vertical) cursors. You can adjust the position of each cursor by dragging it with a mouse in the waveform display area.

The value of the waveform data at the cursor position is displayed in the channel information area.

9.4.5 Saving Acquired Data

The acquired data can be saved by clicking the "Save" button in the acquisition settings area.

10. ［Analyzer］ Parameter Output

10.1 Overview

Parameter Output is a function to output parameters adjusted by Analyzer as a header file of the motor control program. You can launch Parameter Output by pressing the “Output Header” button on Analyzer’s Control Window or selecting the menu item “Output Header File” on User Button Control Window.

10.2 Features

- Outputs the variables adjusted with Control Window or User Button to a header file as micro definitions.
- On the select window, you can select a variable tied to the macro definition to be output to a header file.
- When entering a new macro name on the select window, you can output the variable, adding it to the tail of the header file.

10.3 Window Structure

The structure of Parameter Output window is shown below.

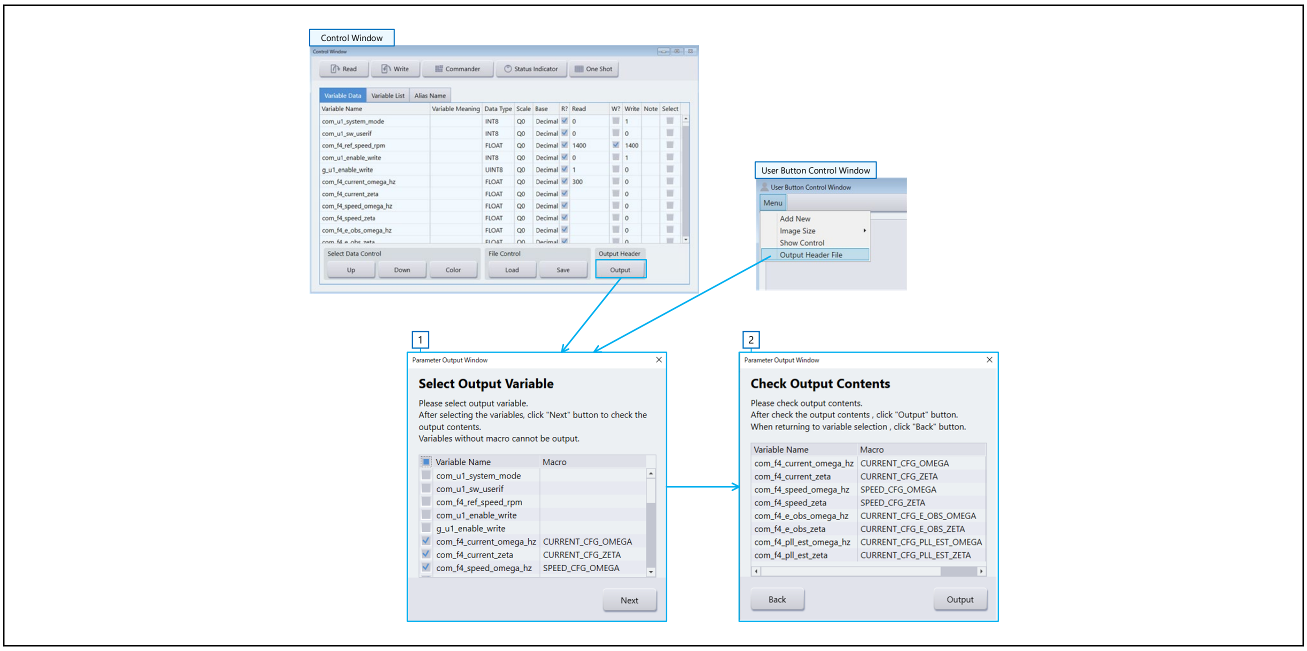


Figure 10-1 Parameter Output Window Structure

Table 10-1 Functions of Parameter Output Window

No.	Name	Explanation
1	Select output variable window	Select variables to be output and enter the macro name.
2	Check output contents window	Check the output contents and output a header file.

10.3.1 Select Output Variable Window

Select Output Variable window is launched from the “Output Header” button on Control Window or the menu item “Output Header File” of User Button Control Window. You can select variables and macro definitions to be output to a header file on this window.

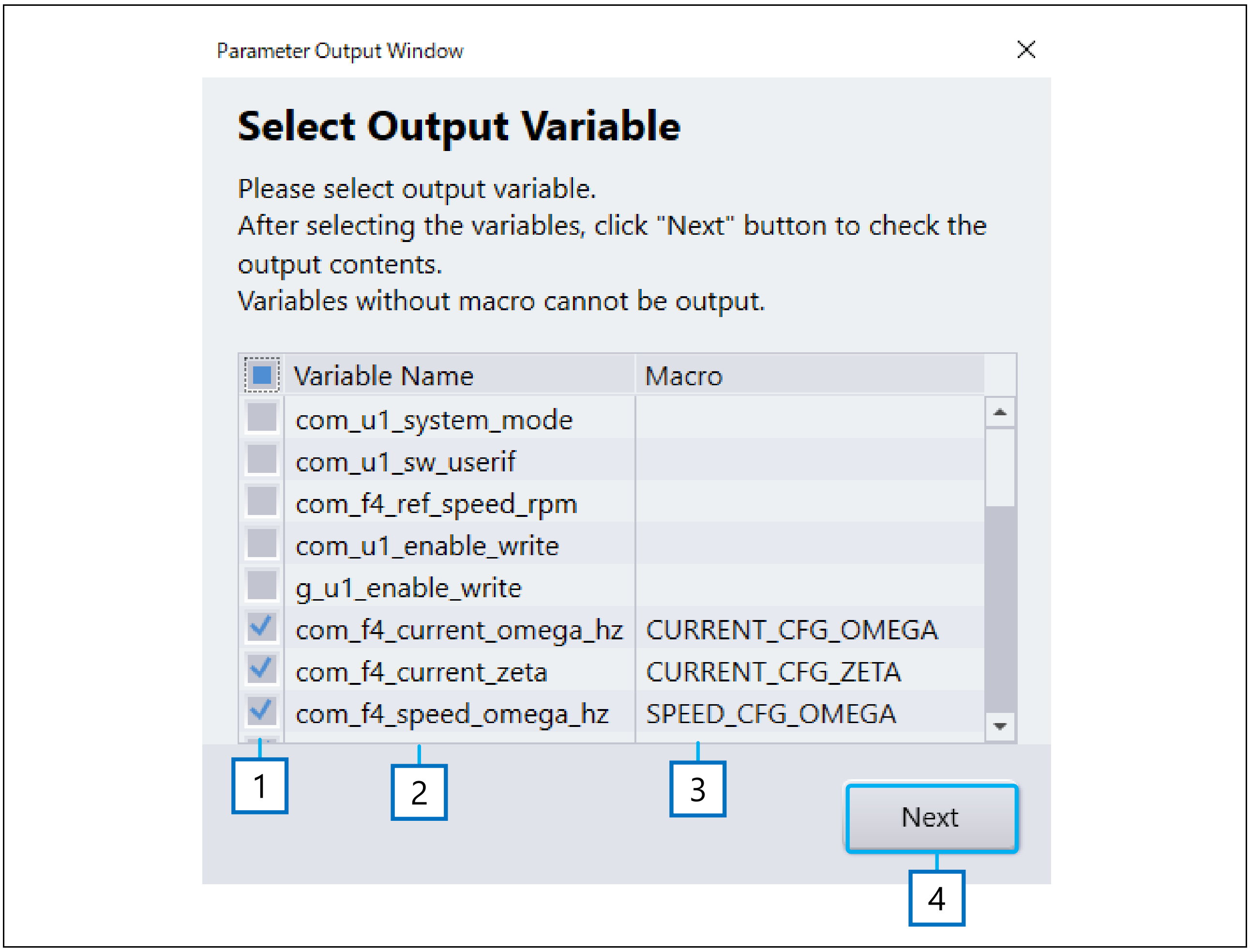


Figure 10-2 Select Output Variable Window

Table 10-2 Functions of Select Output Variable Window

No.	Name	Explanation
1	Select output variable checkbox	Select variables to be output as macro definitions.
2	Variable Name	Displays variables that are subject to be read on Control Window and User Button.
3	Macro	Outputs the macro name. If it is blank, enter a macro name to be output.
4	Next button	Moves to Check Output Contents window.

10.3.2 Check Output Contents Window

Check Output Contents window is displayed by pressing the “Next” button on Select Output Variable window. You can check variables, macro definitions, and values to output to a header file here.

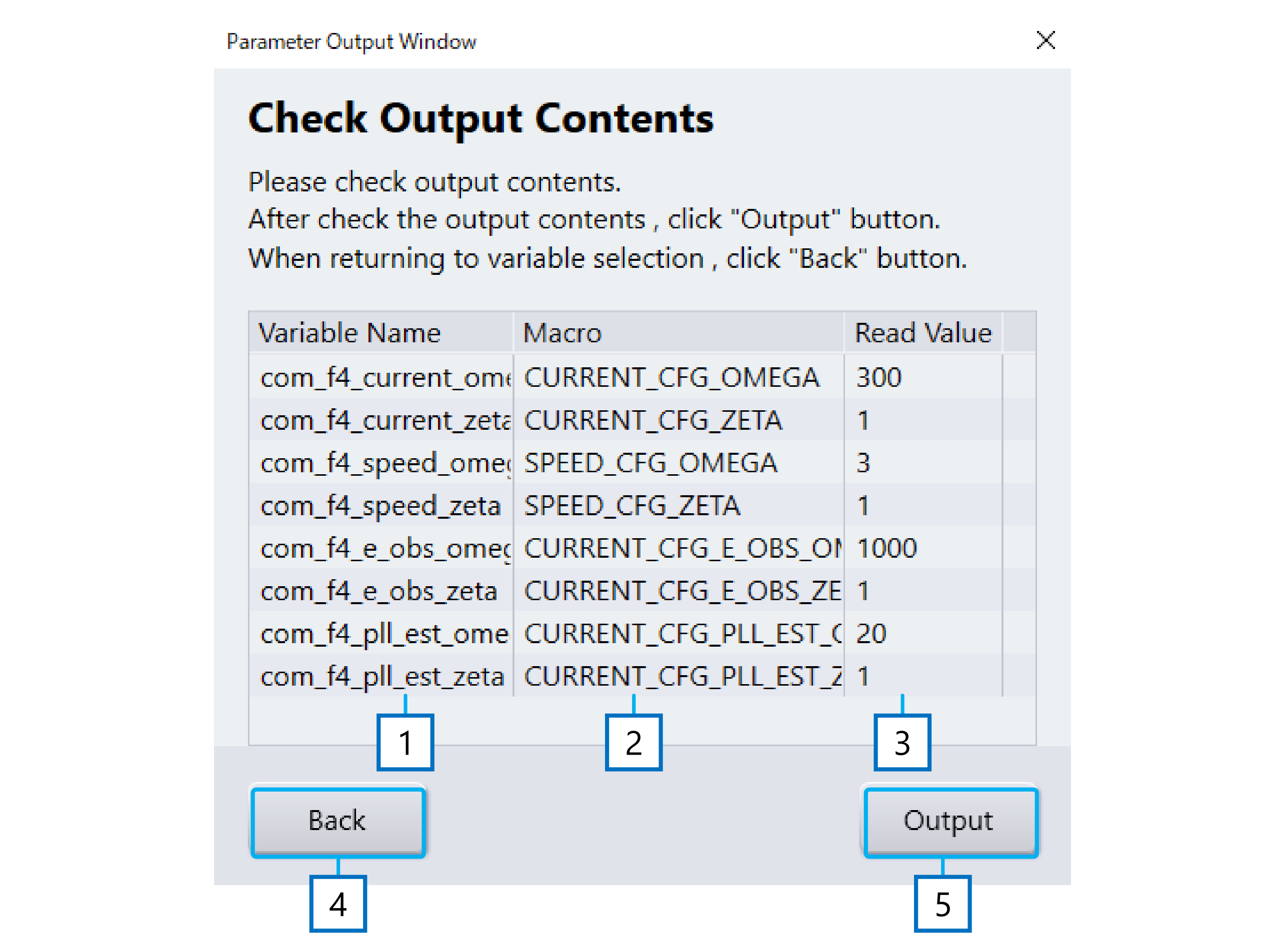


Figure 10-3 Check Output Contents Window

Table 10-3 Functions of Check Output Contents Window

No.	Name	Explanation
1	Variable Name	Displays variables to be output as micro definitions.
2	Macro	Displays macro names to be output.
3	Read Value	Displays setting values to be output as micro definitions. * Displays values that have been written in the motor control program.
4	Back button	Returns to Select Output Variable window.
5	Output button	Select the destination to output a header file.

10.4 Explanation of Operation

10.4.1 Displaying Parameter Output Window

To open Select Output Variable window, press the “Output” button on Control Window or select “Output Header File” from the menu on User Button Control Window.

For the variable included in the header template file information, the corresponding macro name is displayed in the Macro column with the checkbox selected as the output target.

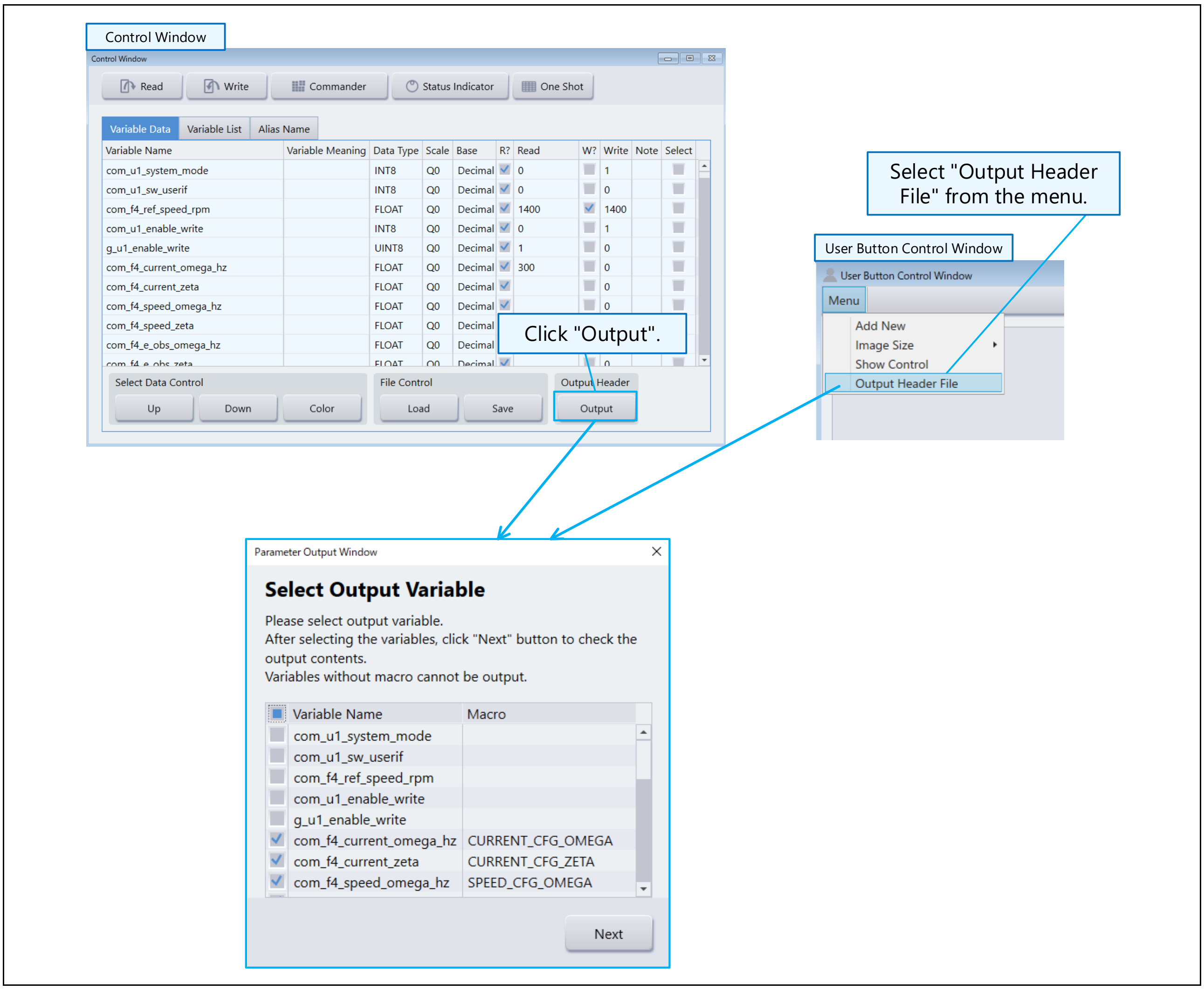


Figure 10-4 Displaying Parameter Output Window

10.4.2 Selecting Output Variable

By checking the checkboxes on Select Output Variable window, you can select the variables and the macro names to be output to a header file.

In addition, you can select or clear all the variables at once by switching the checkbox on the header line.

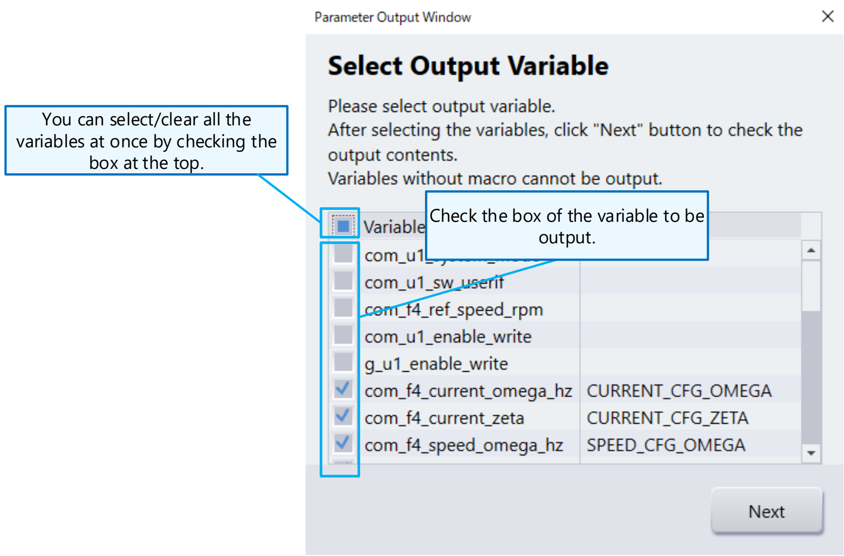


Figure 10-5 Selecting Output Variables

10.4.3 Entering Macro Name

In the Macro column on Select Output Variable window, you can edit a macro name to be output to a header file. However, you cannot edit the cell including the macro name for the header template file information. For the header template file, refer to Section 10.4.6 Setting Header Template File.

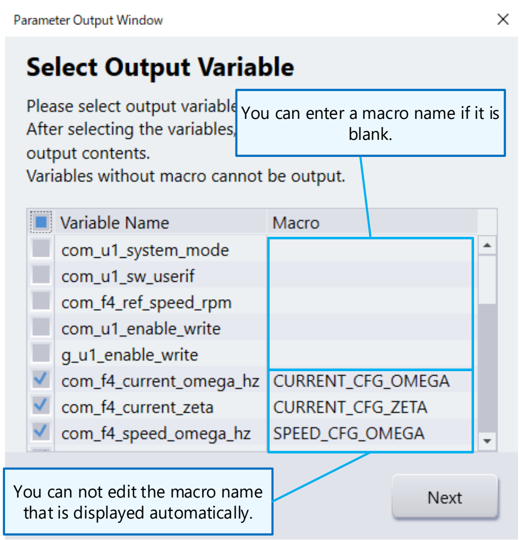


Figure 10-6 Entering Macro Name

10.4.4 Checking Output Contents

Press the “Next” button on Select Output Variable window to move to Check Output Contents window. This window shows variables, macro names, and values set in the motor control program that are to be output.

Note that if the variable without a macro name is selected for output, it is impossible to move to Check Output Contents window.

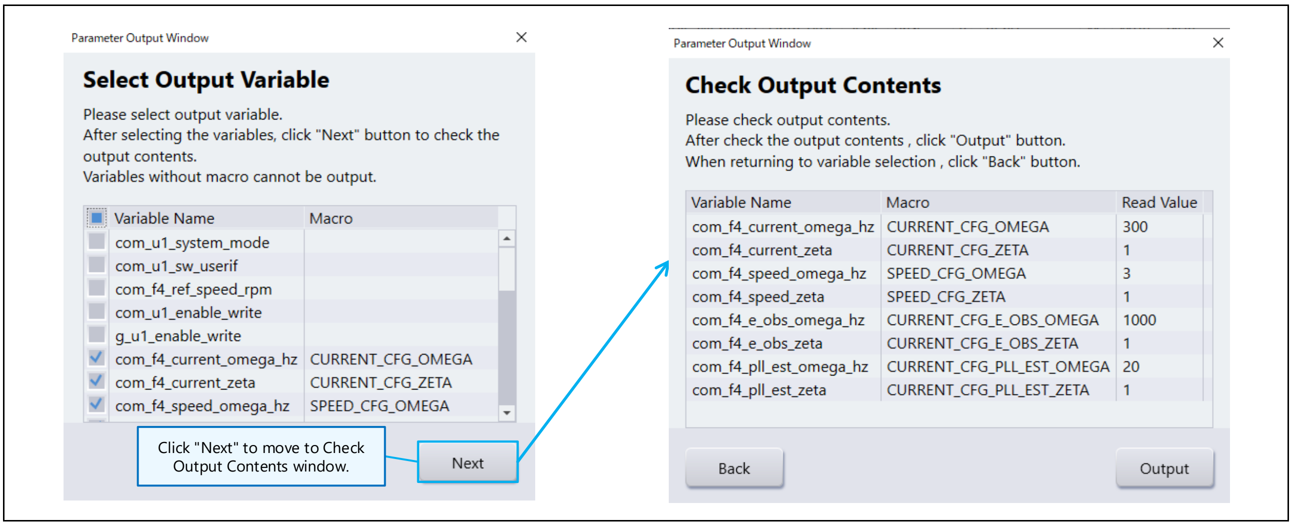


Figure 10-7 Checking Output Contents

10.4.5 Header File Output

Press the “Output” button on Check Output Contents window and select the output destination, and a header file will be output. The contents of header file are generated based on the information of the header template file and the contents of the Check Output Contents window.

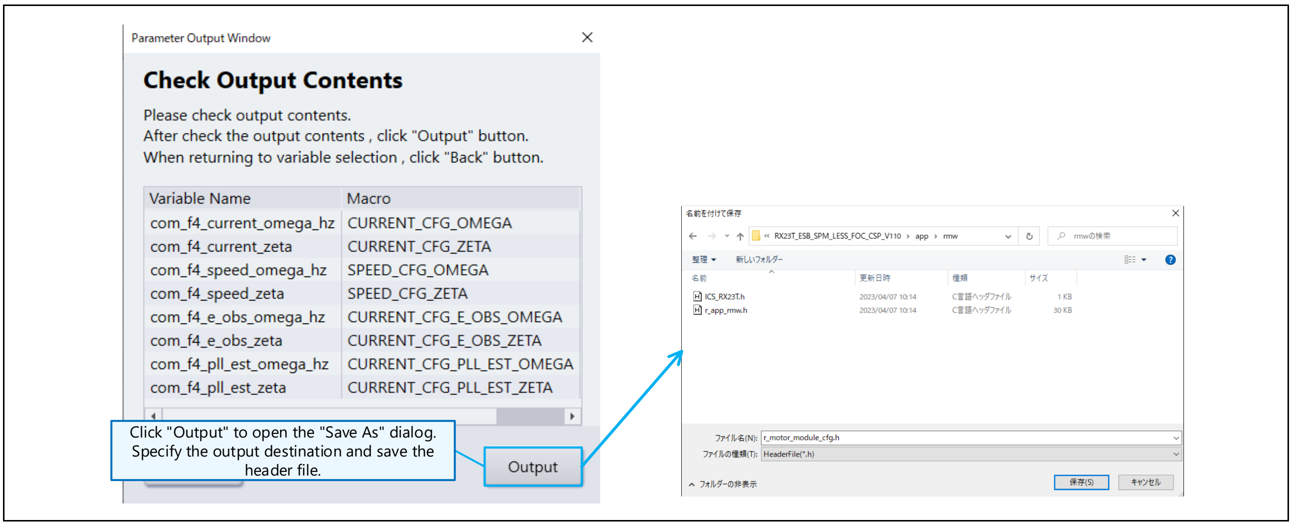


Figure 10-8 Header File Output

10.4.6 Setting Header Template File

Create a template in XML format for a header file to be output.

The created file is applied to by saving it to the same folder as the rmt file.

Name the file as below.

XML file name: OutputHeaderFileInfoForAnalyzer.xml

The XML file consists of the “header part”, “data part” and “footer part”, which are described within the root element “OutputHeaderFileForAnalyzerSetting”.

Figure 10-9 Elements Of Header Template File

Table 10-4 Elements of Header Template File

No.	Name	Element name	Explanation
1	Root element	OutputHeaderFileForAnalyzerSetting	Root element of header template file
2	Header part	Header	Defines the header of the output file.
3	Data part	OutputData	Defines the data of the output file.
4	Footer part	Footer	Defines the footer of the output file.

10.4.6.1 Root Element

Describe the information of the header template within the OutputHeaderFileForAnalyzerSetting tab.

10.4.6.2 Header Part

Outputs the text described with the Header tabs to a file.

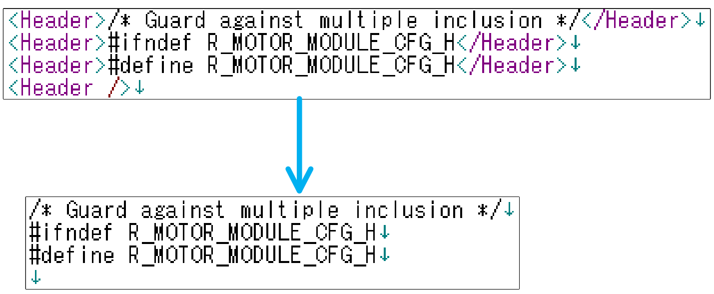


Figure 10-10 Header Part

10.4.6.3 Data Part

The OutputData tab consists of the following attributes.

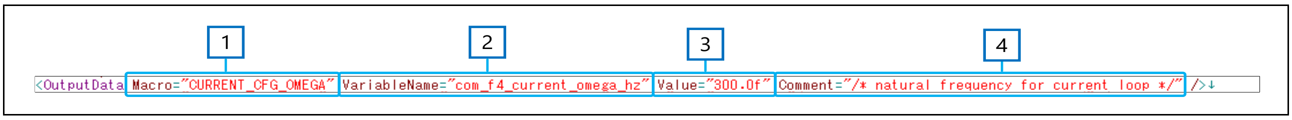


Figure 10-11 Attributes of OutputData Tab

Table 10-5 Attributes of OutputData

No.	Attribute	Explanation
1	Macro	Defines a macro name.
2	VariableName	Defines a variable name for the macro name.
3	Value	Defines the initial value of the macro definition.
4	Comment	Defines a comment for the macro.

Parameter Output combines the contents of each attribute with the OutputData tab and outputs them to a file. There are three output patterns according to the description of each attribute and the selection on Parameter Output Window.

① When a VariableName attribute is described, and when the described VariableName is selected for output, this function replaces the value in Read Value with the value described in the Value attribute and outputs it.

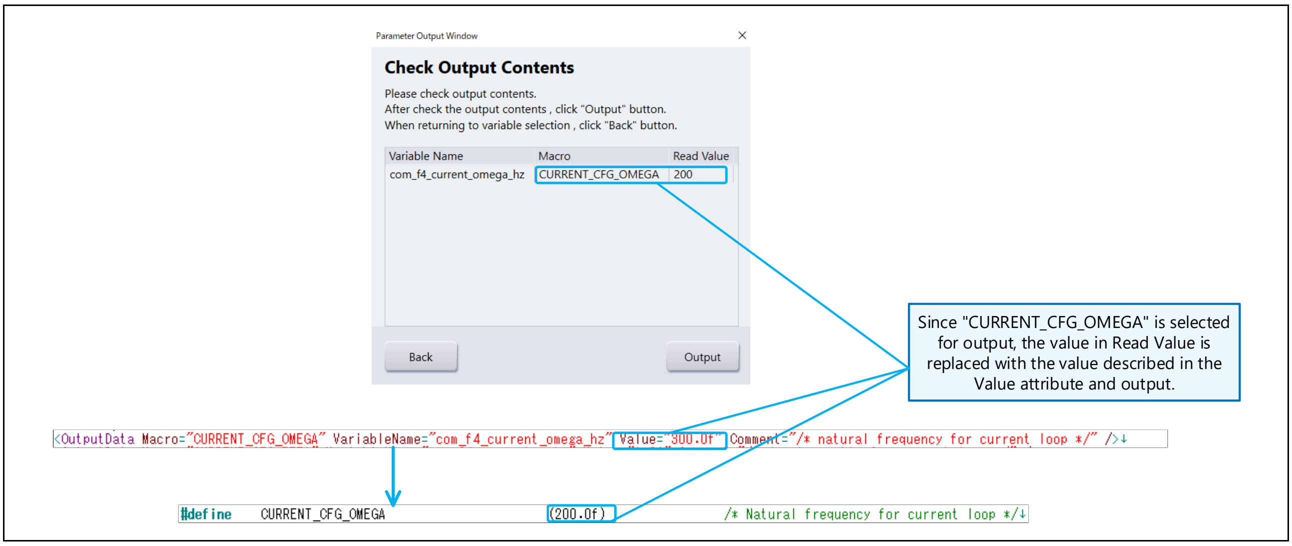


Figure 10-12 Example of Data Part Output

② When a VariableName attribute is not described, or when a described VariableName is not selected for output, this function outputs the description of the Value attribute as it is. When a VariableName attribute is not described, this function combines the macro name described in the Macro attribute with the value described in the Value attribute, and outputs the macro definition.

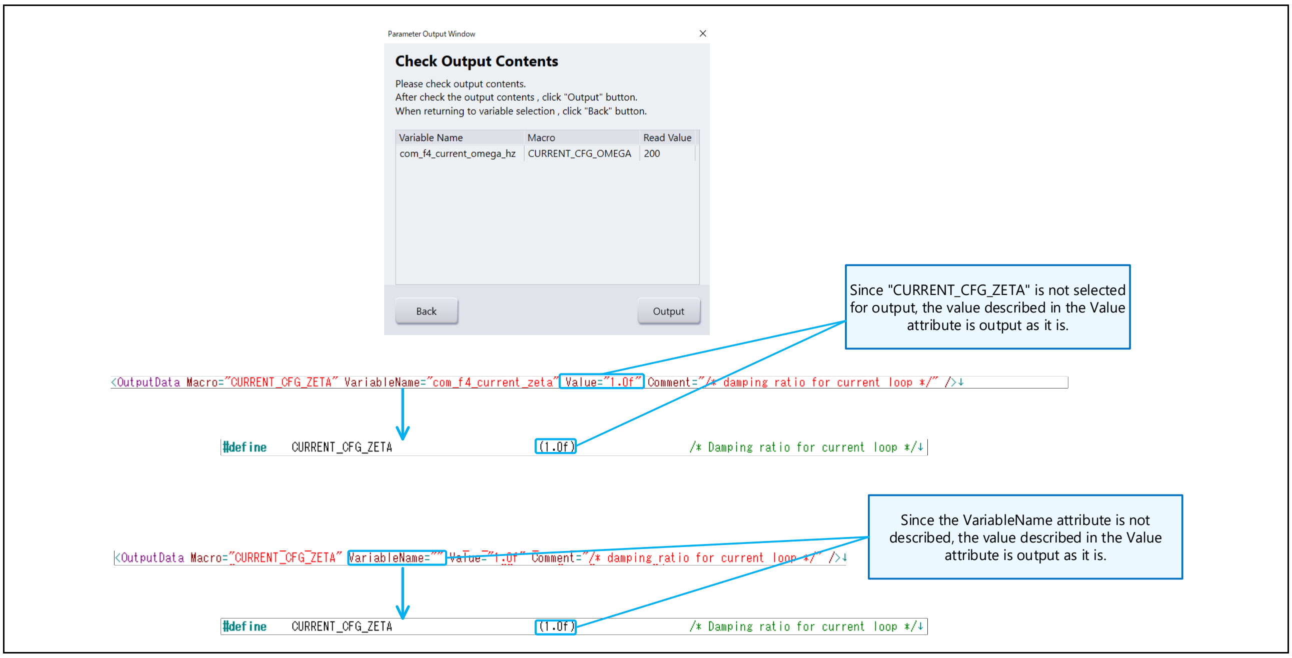


Figure 10-13 Example of Data Part Output

When a Macro attribute is not described, this function outputs only the description of the Comment attribute. This is used to output descriptions other than macro definitions such as compile options and comments.

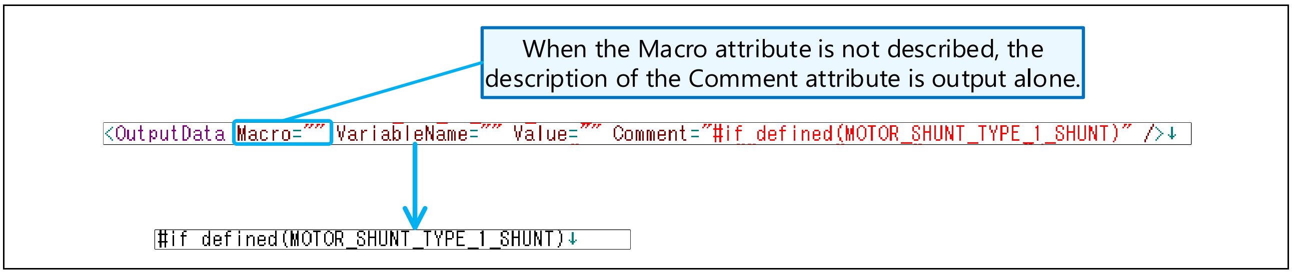


Figure 10-14 Example of Data Part Output

10.4.6.4 Footer Part

Outputs the text described with the Footer tab to a file.

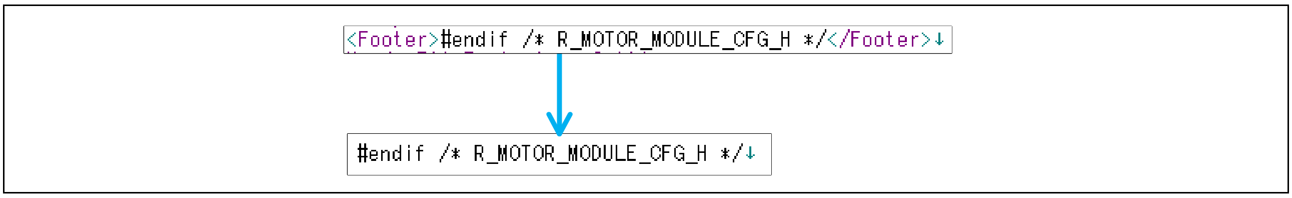


Figure 10-15 Footer Part

11.Navigation

11.1 Overview

Navigation is a function that explains how to operate each function of Renesas Motor Workbench. In the default settings, Navigation is displayed when Analyzer or Tuner is launched.

Since the window structure is same for each tool, this section explains Analyzer's Navigation as an example.

11.2 Features

- Explains the operation procedure with text and images.

11.3 Window Structure

11.3.1 Navigation Window

The structure of Navigation Window is shown below.

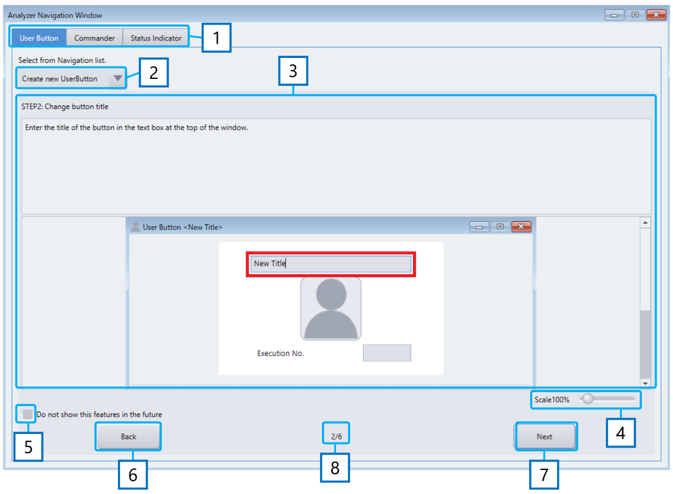


Figure 11-1 Navigation Window

Table 11-1 Functions of Navigation Window

No.	Name	Explanation
1	Function select tab	Select the function for which you want to display Navigation.
2	Operation procedure select box	Select the operation procedure you want to check.
3	Operation procedure display	Displays the selected operation procedure.
4	Image scaling slider	Changes scaling of the displayed image.
5	Disable auto-start checkbox	Enables/disables auto-start of Navigation Window.
6	Back button	Returns to the previous page.
7	Page number display	Displays the numbers of the current page and the last page.
8	Next button	Moves to the next page.

11.4 Explanation of Operation

11.4.1 Switching Displayed Function

When selecting the upper-left tab on Navigation Window, you can switch the function for which Navigation is displayed.

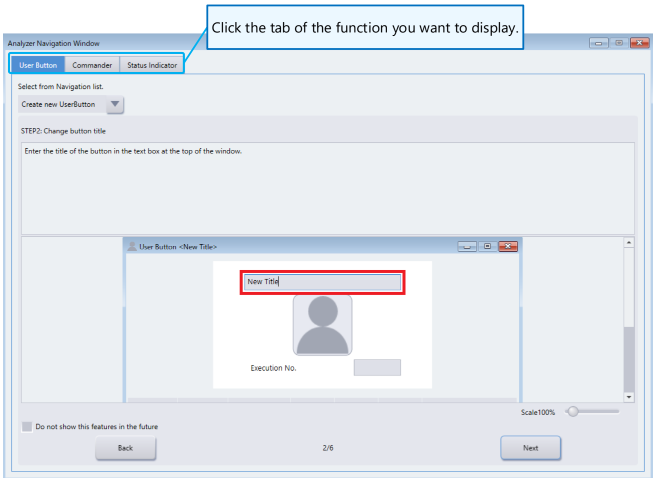


Figure 11-2 Switching Displayed Function

11.4.2 Selecting Operation Procedure

The operation procedure is displayed in the window by selecting it in the operation procedure select box on Navigation window,

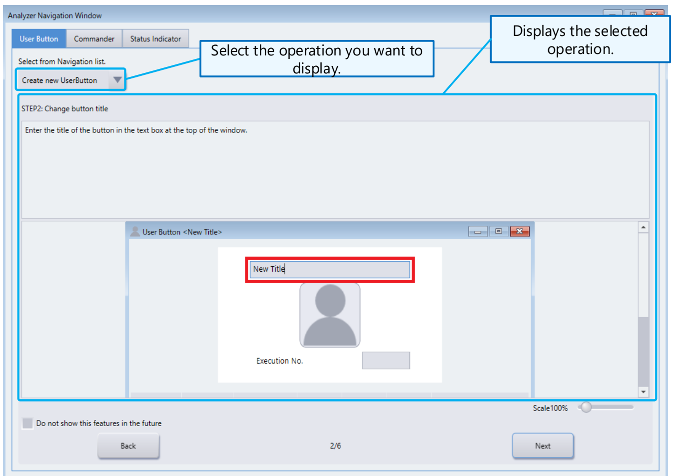


Figure 11-3 Selecting Operation Procedure

11.4.3 Changing Image Scaling

You can change the scaling of the displayed image by using the slider or Ctrl key + mouse wheel.

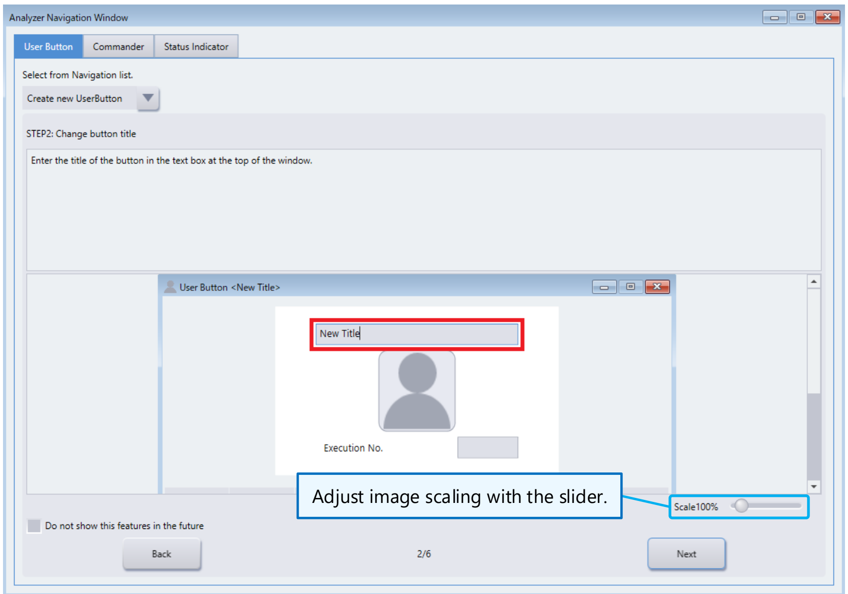


Figure 11-4 Changing Image Scaling

11.4.4 Disabling Navigation Auto-Start

By checking the Disable auto-start checkbox, you can disable auto-start of Navigation when Analyzer or Tuner is launched.

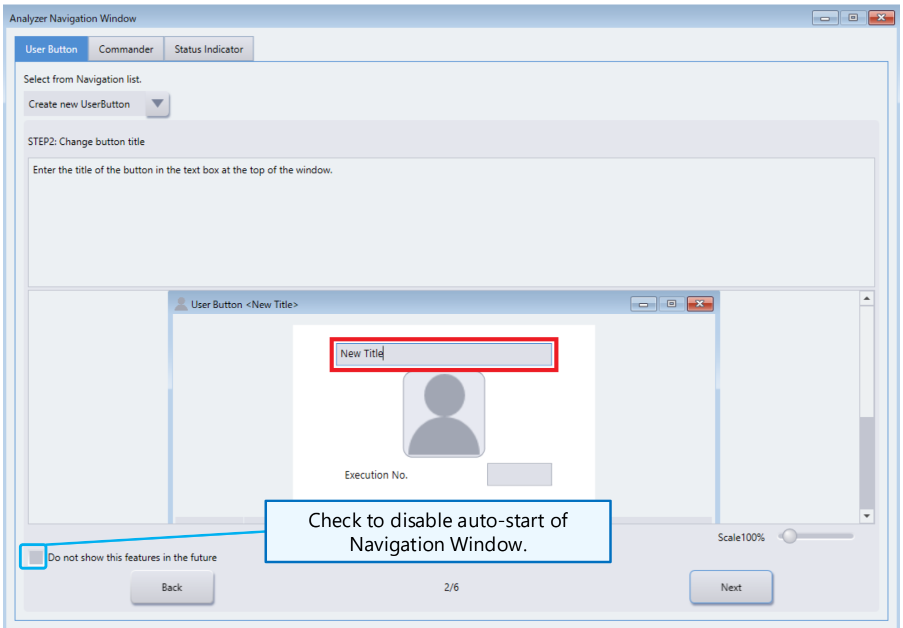


Figure 11-5 Disabling Navigation Auto-Start

If you have disabled Navigation auto-start, you can display Navigation by selecting Analyzer Navigation or Tuner Navigation from Help menu.

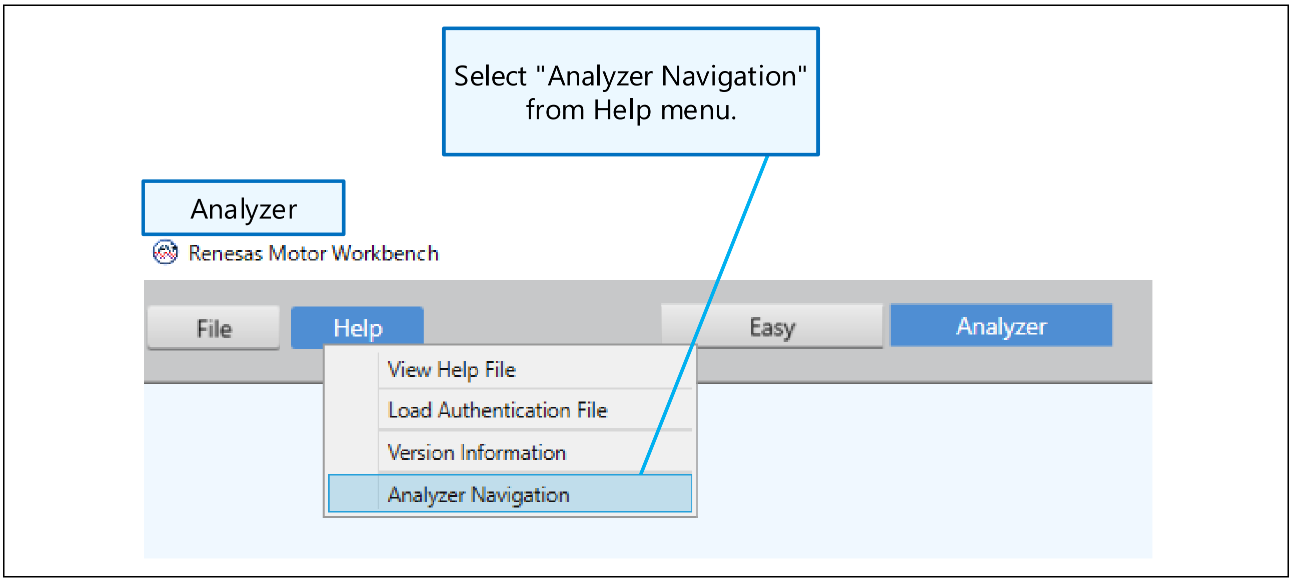


Figure 11-6 Launching Analyzer Navigation

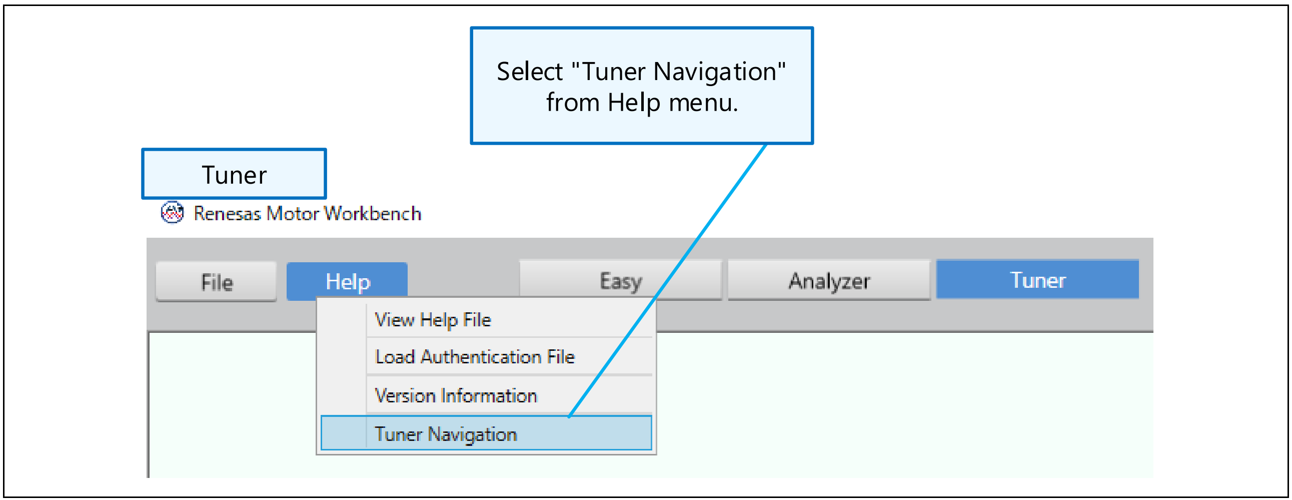


Figure 11-7 Launching Tuner Navigation

11.4.5 Moving to Next/Previous Page

Click the Back/Next button to move to the previous/next Navigation page.

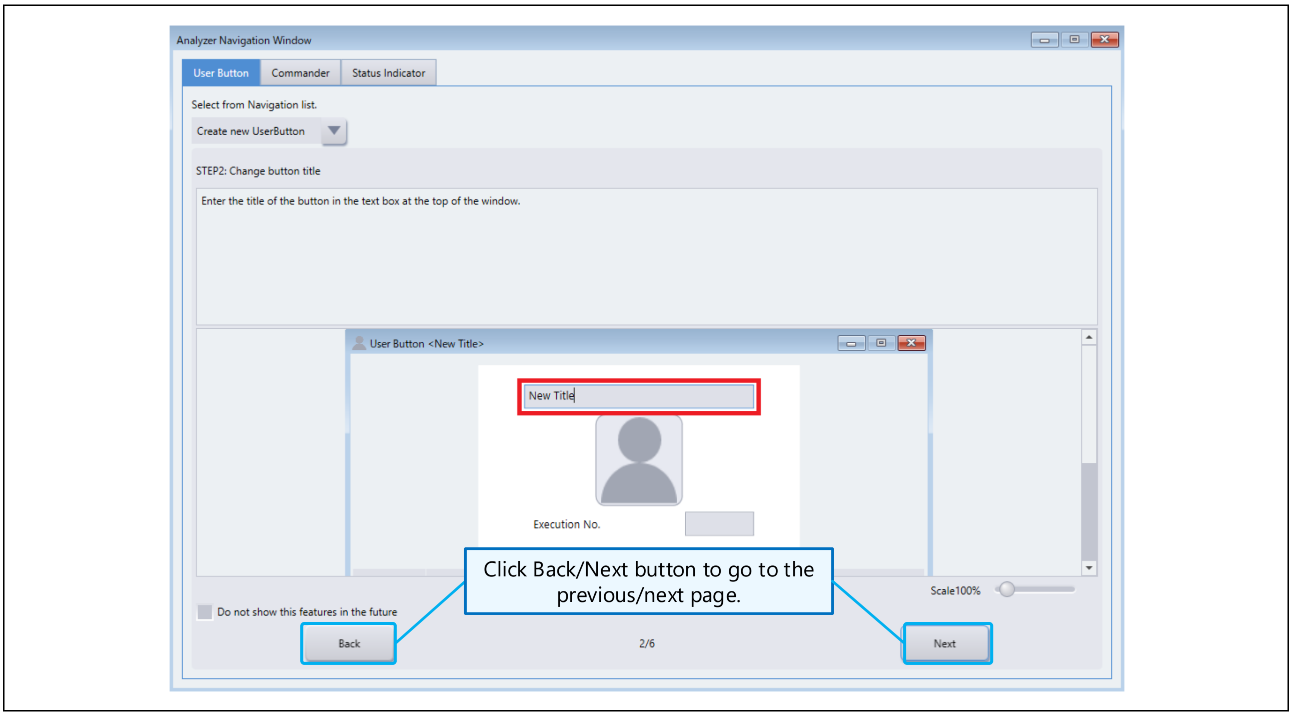


Figure 11-8 Moving to Next/Previous Page

12.Easy GUI

12.1 Overview

Easy GUI is a dedicated GUI tool that allows you to operate and measure the motor driving easily. The GUI configuration is simpler than Analyzer, so you can easily operate a motor even if you are not familiar with it. In addition, the status of the motor can be visualized by GUI, so you can use this tool for demonstrations and other purposes.

12.2 Features

By using Easy GUI, you can easily set the speed and position of the motor through intuitive operations using sliders and sequences, and visually measure the status of the motor with meters and graphs. You can also repeatedly execute sequence operations that change the values of the variables. Variables displayed in GUI can be set optionally according to the user's program.

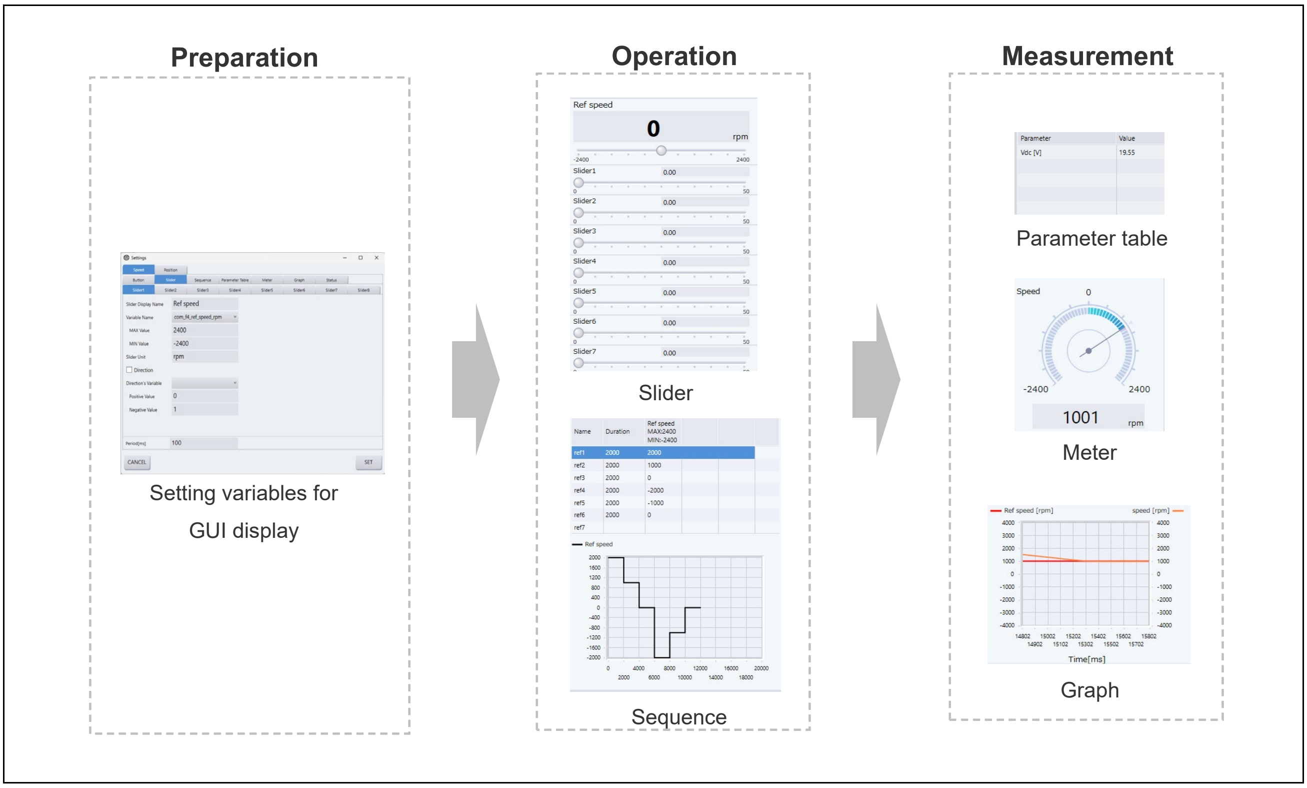


Figure 12-1 Features of Easy GUI

12.3 Explanation of Window

The functions of the Easy GUI window are shown in Figure 12-2 and Table 12-1.

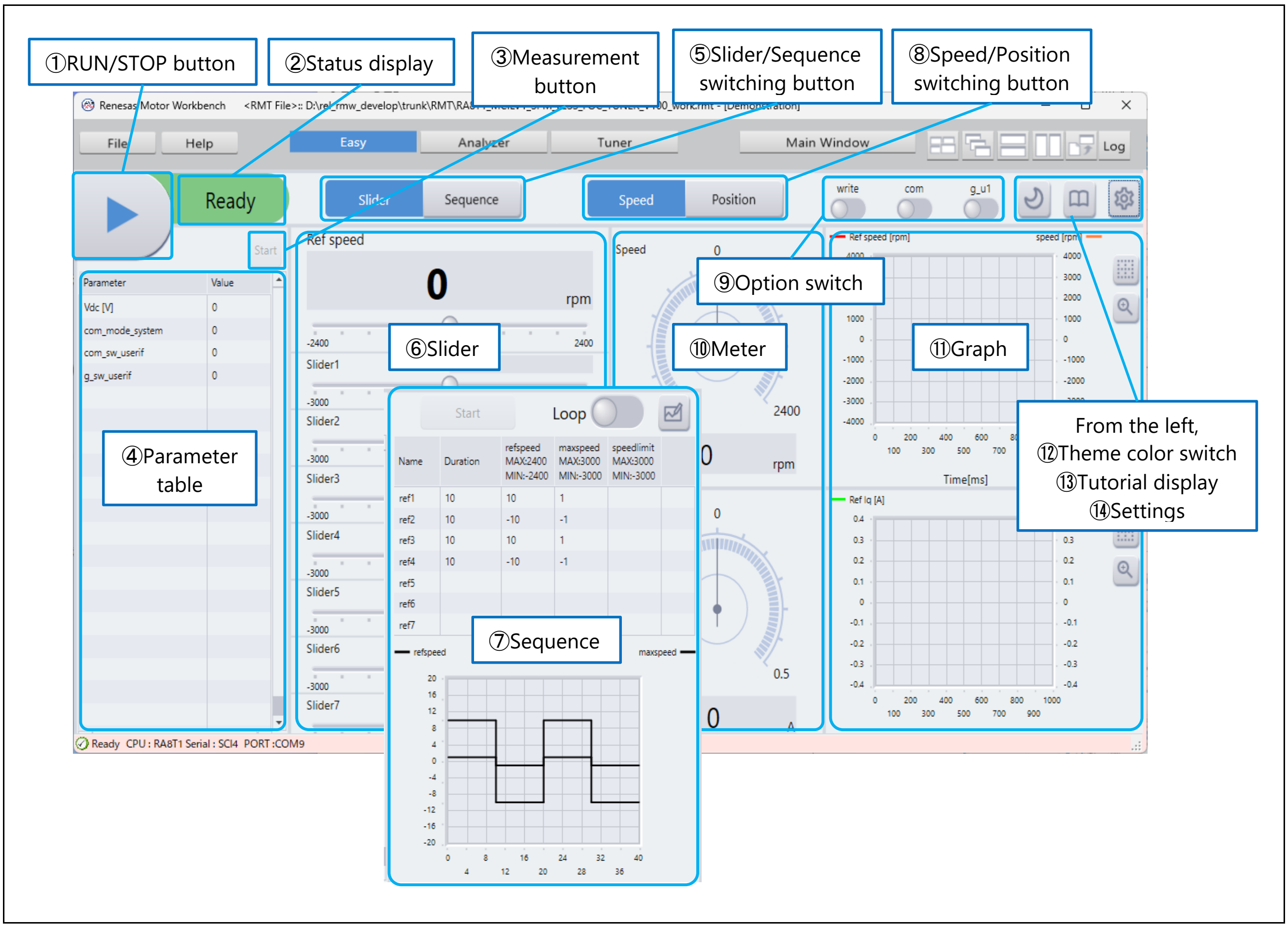


Figure 12-2 Functions of Easy GUI Window

Table 12-1 Functions of Easy GUI Window

No.	Name	Explanation
1	Run/Stop button	Drives or stops the motor.
2	Status display	Displays the drive status.
3	Measure button	Starts measurement
4	Parameter table	Reads parameter values periodically.
5	Slider/sequence switching button	Switches between control parameter input and sequence input.
6	Slider	Specifies command values and control parameters.
7	Sequence	Specifies command value sequences.
8	Speed/position switching button	Switches between speed control mode and position control mode.
9	Option switch	ON/OFF of optional functions.
10	Meter	Displays the drive status in meter view
11	Graph	Displays the drive status in graph view.
12	Theme color switch	Switches the screen theme colors.
13	Tutorial display	Launches the tutorial screen.
14	Settings	Launches the settings screen.

12.4 Preparation

12.4.1 Displaying Easy GUI Window

When Renesas Motor Workbench starts, Main Window is displayed. Press “Easy” in the center of Main Window to display the Easy GUI window. Press “Main Window” in the upper-right of the Easy GUI window to return to Main Window.

Figure 12-3 Displaying Easy GUI Window

12.4.2 Setting Variables to Display in GUI

Before using Easy GUI, you are required to associate the GUI display with variables in the program. For the method for setting variables for GUI display, see Section 12.6.

12.5 Explanation of Operation

This section explains how to operate each function.

12.5.1 Basic Function

① Drive/stop motor: Switches driving or stopping the motor with the RUN/STOP button.

 is displayed when the motor is running. -> Click this button to stop the motor.

 is displayed when the motor is stopped. -> Click this button to drive the motor.

② Status display : Displays the drive status of the motor.

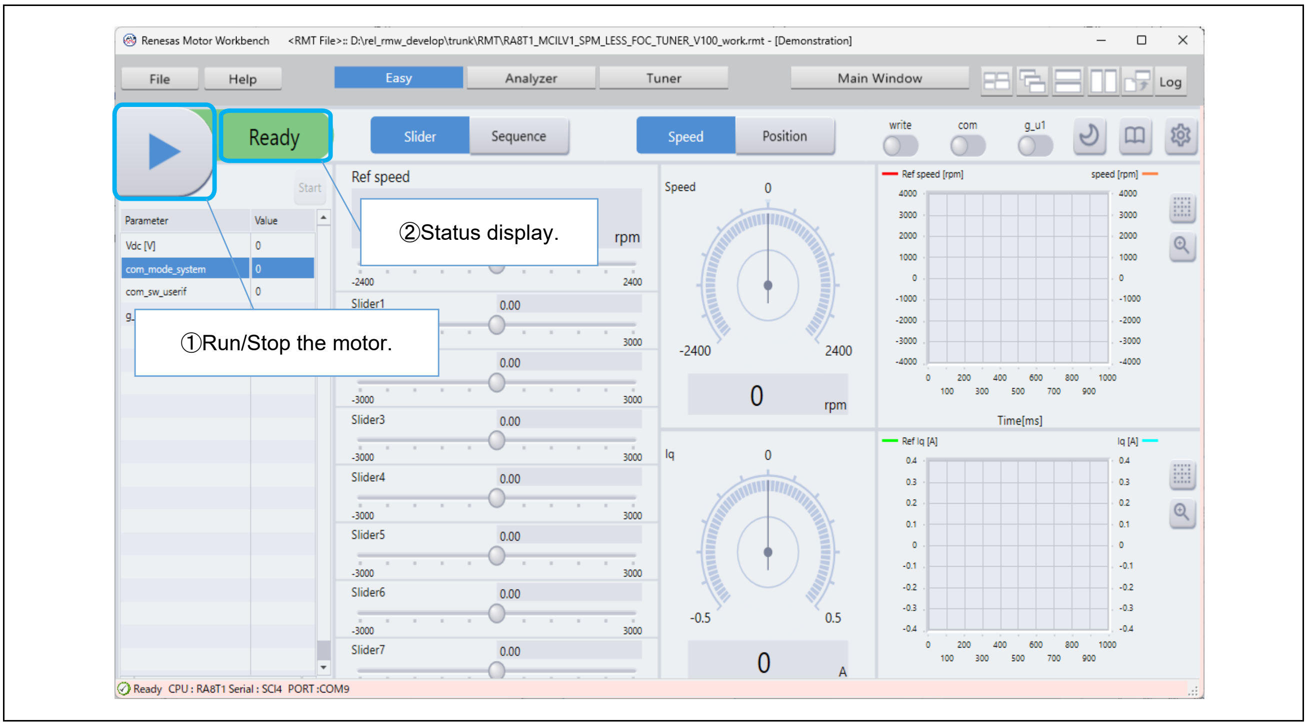


Figure 12-4 Basic Functions

12.5.2 Slider Function

The slider function allows you to change variable values dynamically by moving sliders while the motor is running. The slider function is operated as follows.

① Display the slider view : Press “Slider” (of slider/sequence switching button).
② Drive the motor : If the motor is stopped, press the RUN/STOP button to drive the motor.
③ Move the slider : Move the slider to change the variable value.
You can also enter the value directly without using the slider.
④ Stop the motor : To finish the operation, press the RUN/STOP button to stop the motor.

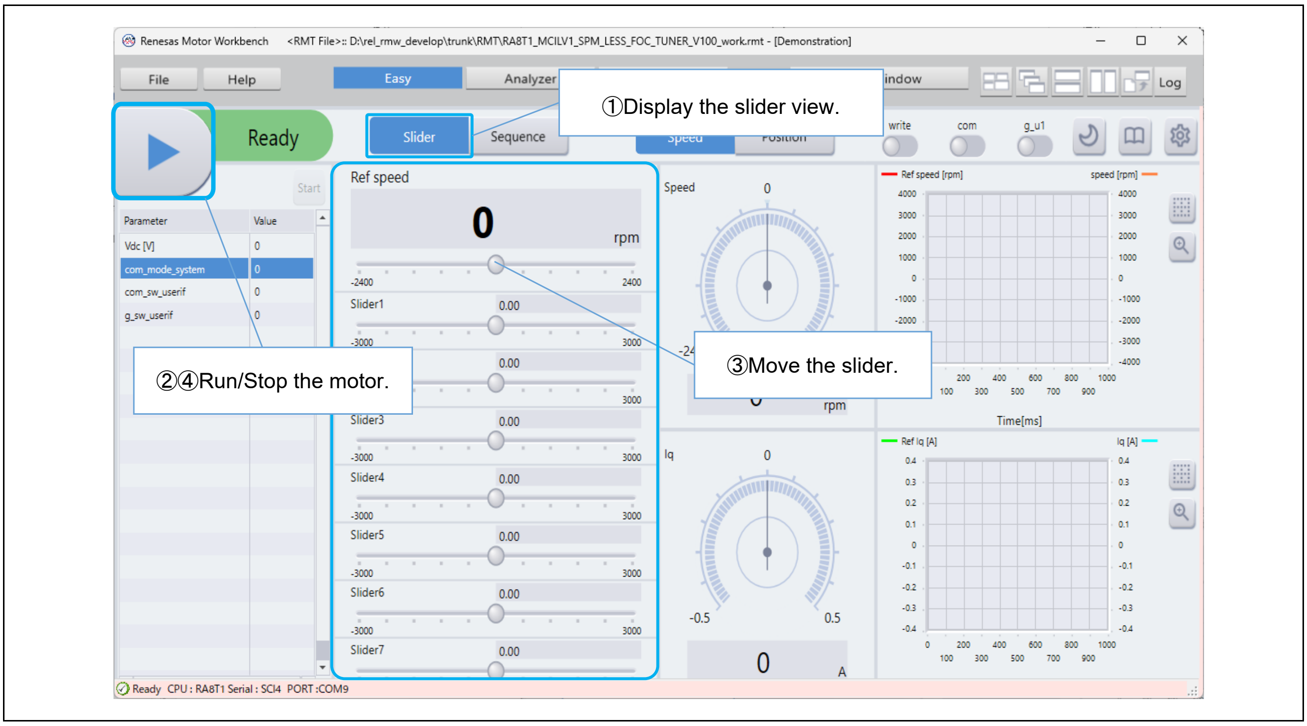


Figure 12-5 Slider Function

12.5.3 Sequence Function

The sequence function allows you to change the variable value sequentially in the order preset in the sequence table while the motor is running.

12.5.3.1 Setting Sequence

The sequence is set by the following operations.

① Display sequence view :Press “Sequence” (of slider/sequence switching button).
② Create a sequence table :Set the variable value to be checked from ref1 in order of the sequence table. Specify the execution time (unit: msec) in the Duration field.
③ Check graph :The execution image of the sequence table is displayed in the graph. If you change the values in the sequence table, the graph is redrawn. The horizontal axis of the graph is the time, and the vertical axis is the value of each variable.

To create a sequence table, you can use the sequence graphical function as well as enter values directly in the sequence table. Refer to Section 12.5.3.2 for the sequence graphical function.

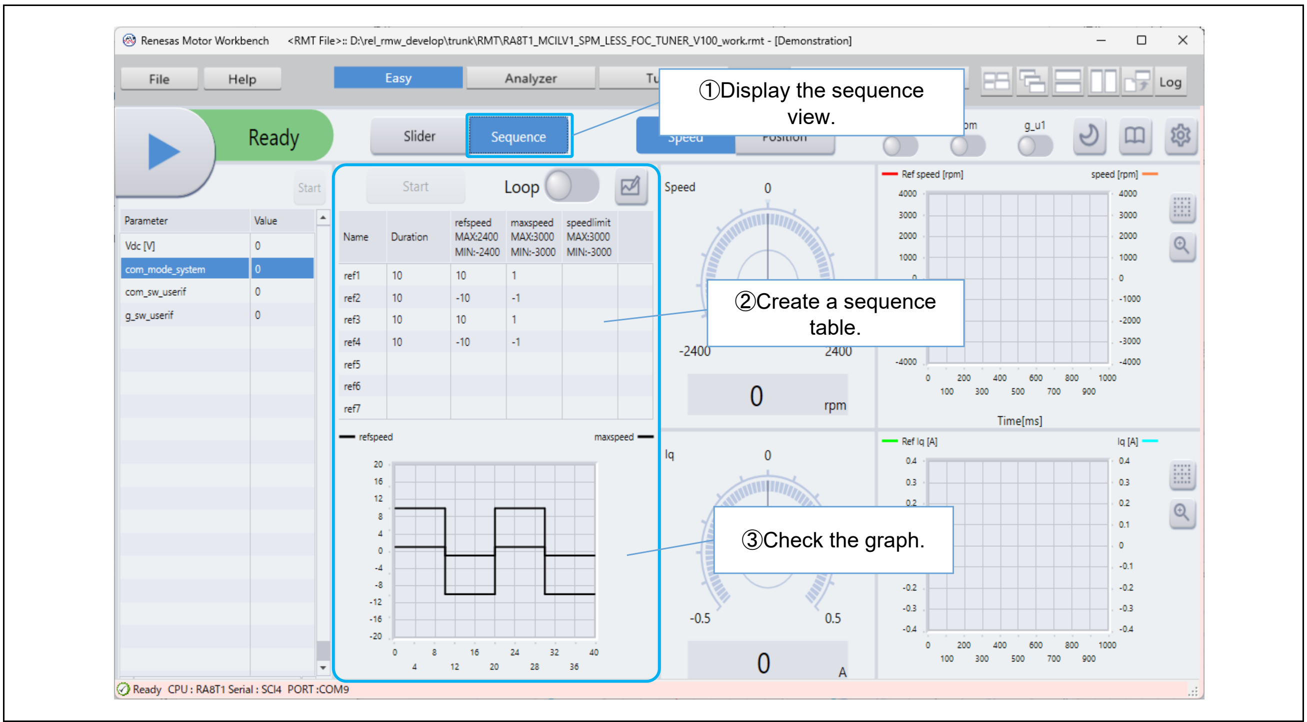


Figure 12-6 Sequence Setting

12.5.3.2 Sequence Graphical Function

If you have created a sequence table by entering values directly into the sequence table, the settings in this section are not required.

The sequence graphical function allows you to set the sequence while checking the setting points in the graph of the template data.

The sequence graphical function is set as follows.

① Display the sequence view : Press "Sequence" (slider/sequence switching button).
② Display the graphical input window : Press the sequence graphical input button.
③ Set the sequence value : Set values in the boxes below while checking the setting points in the graph above. Since this graph is of template data, the set value is not reflected.
④ Reflect to the sequence table and RMT : Press "Set" to close the sequence graphical input window. The set value is reflected in the sequence table and the RMT file.

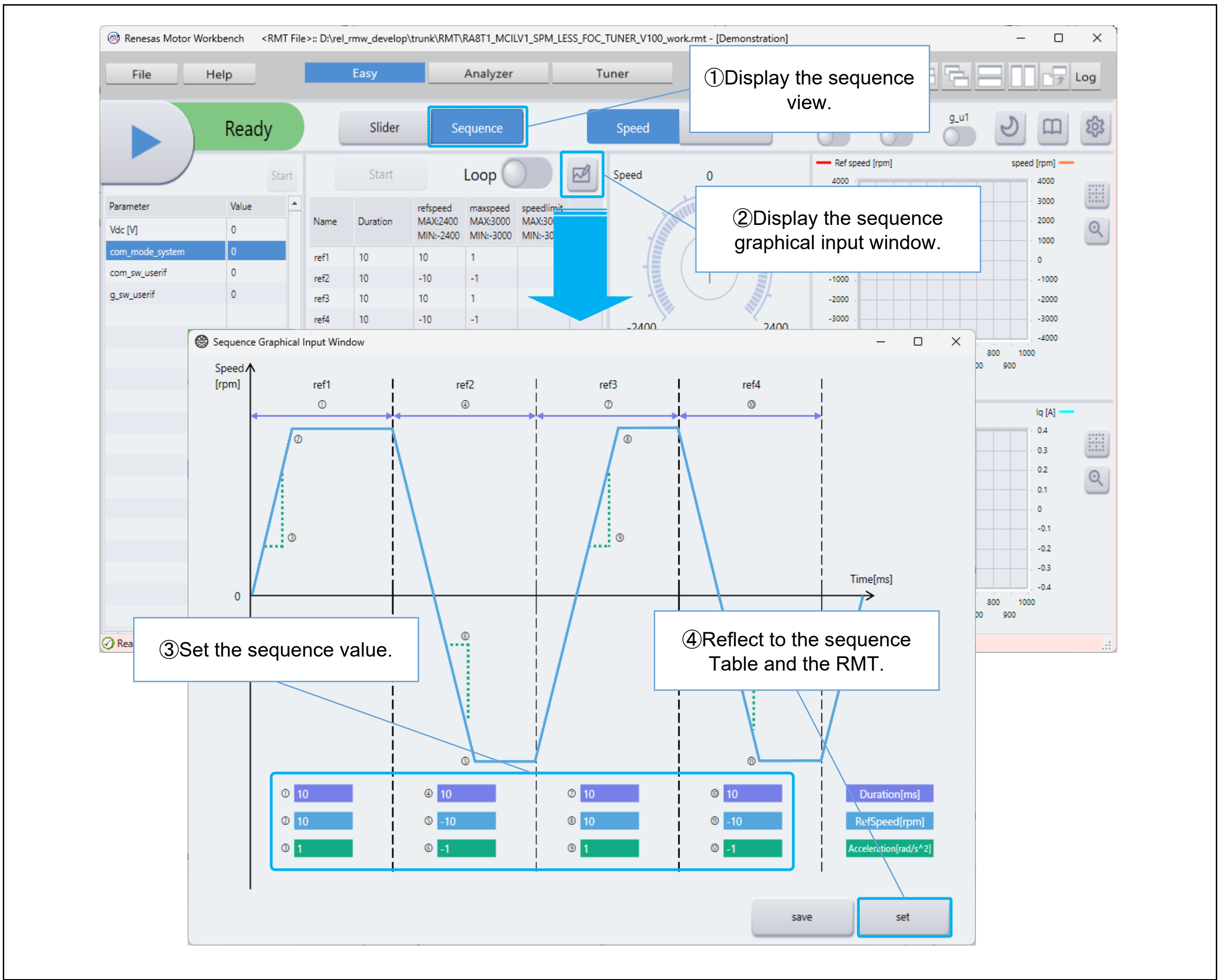


Figure 12-7 Sequence Graphical Function

12.5.3.3 Executing Sequence

Follow the procedure below to execute the sequence for changing the variable value in the order preset in the sequence table while driving a motor.

①	Display sequence view	: Press “Sequence” (slider/sequence switching button)
②	Drive motor	: If the motor is stopped, press the RUN/STOP button to drive the motor.
③	Enable looping	: To repeat the sequence operation, enable the Loop button.
④	Start the sequence	: Press the Start/Stop button.
⑤	Stop the sequence	: Press the Start/Stop button to stop the operation.
⑥	Stop the motor	: Press the RUN/STOP button to stop the motor.

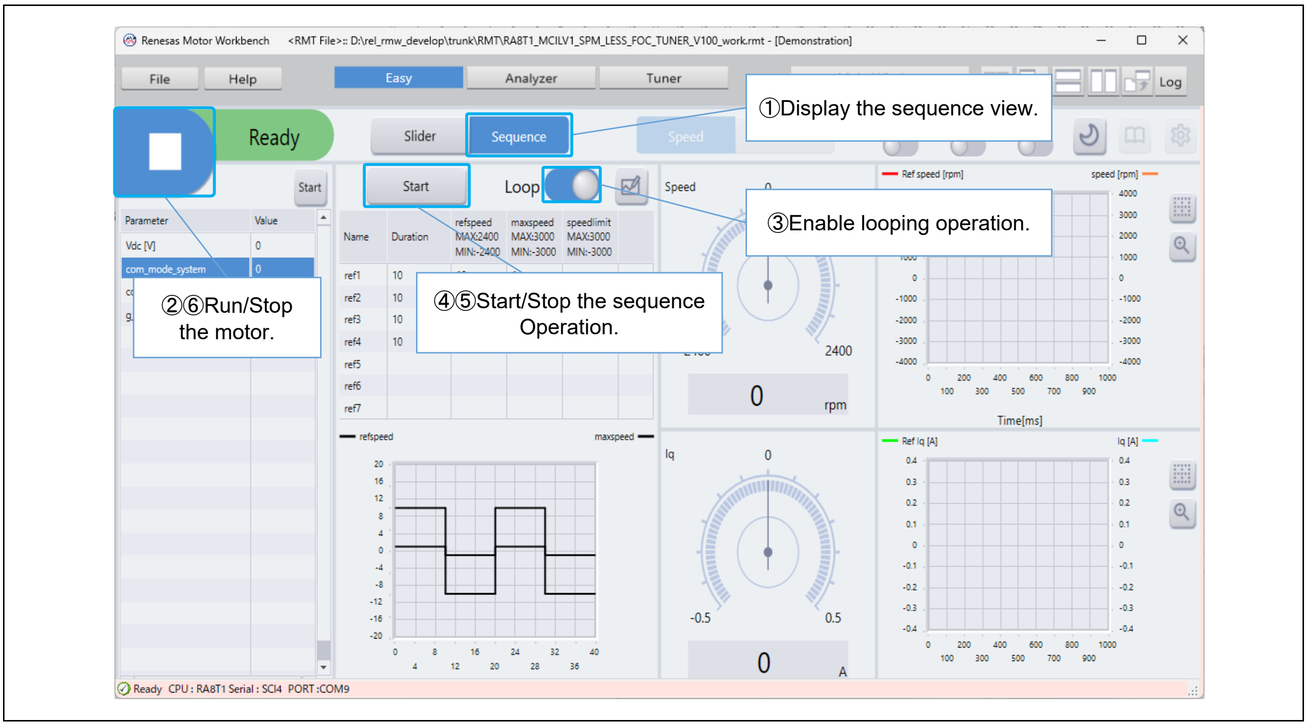


Figure 12-8 Sequence Execution

12.5.4 Measurement Function

The measurement function allows you check the current drive status of a motor in meters and graphs by the following procedures.

①	Drive the motor	: If the motor is stopped, press the RUN/STOP button to drive the motor.
②	Start measurement	: Press the measurement button.
③	Check the measurement result	: The status of the motor is reflected the meter and graph.
④	Zoom the graph	: When zooming the graph, you can copy or save it.
⑤	Stop measurement	: Press the measurement button to stop measurement.
⑥	Stop the motor	: Press the RUN/STOP button to stop the motor.

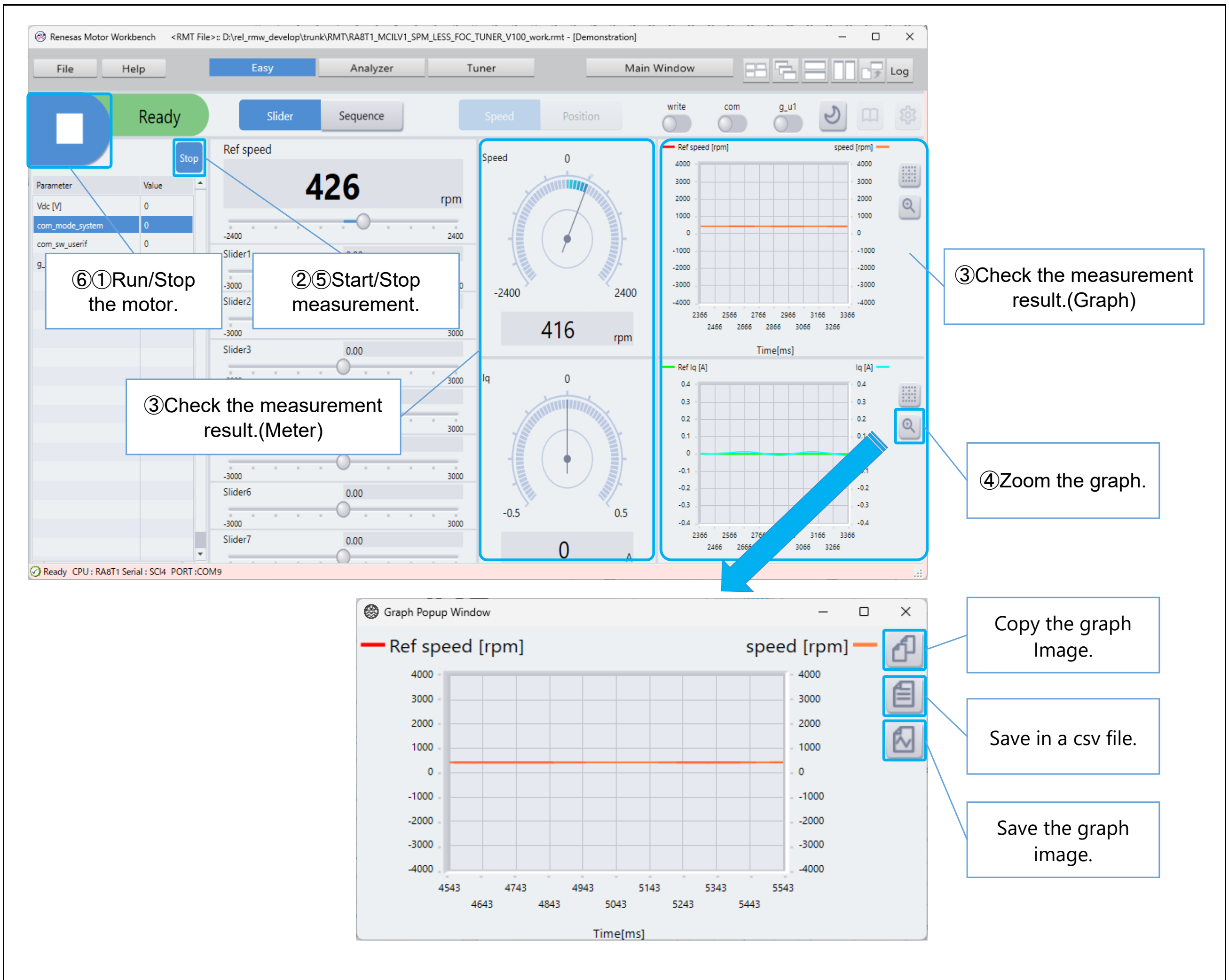


Figure 12-9 Measurement Function

12.6 Setting Variable

You can set the variable to be operated and measured with Easy GUI from the settings button.

When you press the settings button, the settings screen will appear as shown in Figure 12-10 Displaying Setting Window. Select a tab on the settings screen and set the variables to be associated with each GUI.

When specifying a variable name in each settings screen, double-click the corresponding variable name from "Variable Find" that appears when you click the variable name field. You can also search a variable name by entering keywords in the "Find" field of "Variable Find". An example of variable name setting is shown in Figure 12-11.

This section describes examples of settings for each GUI.

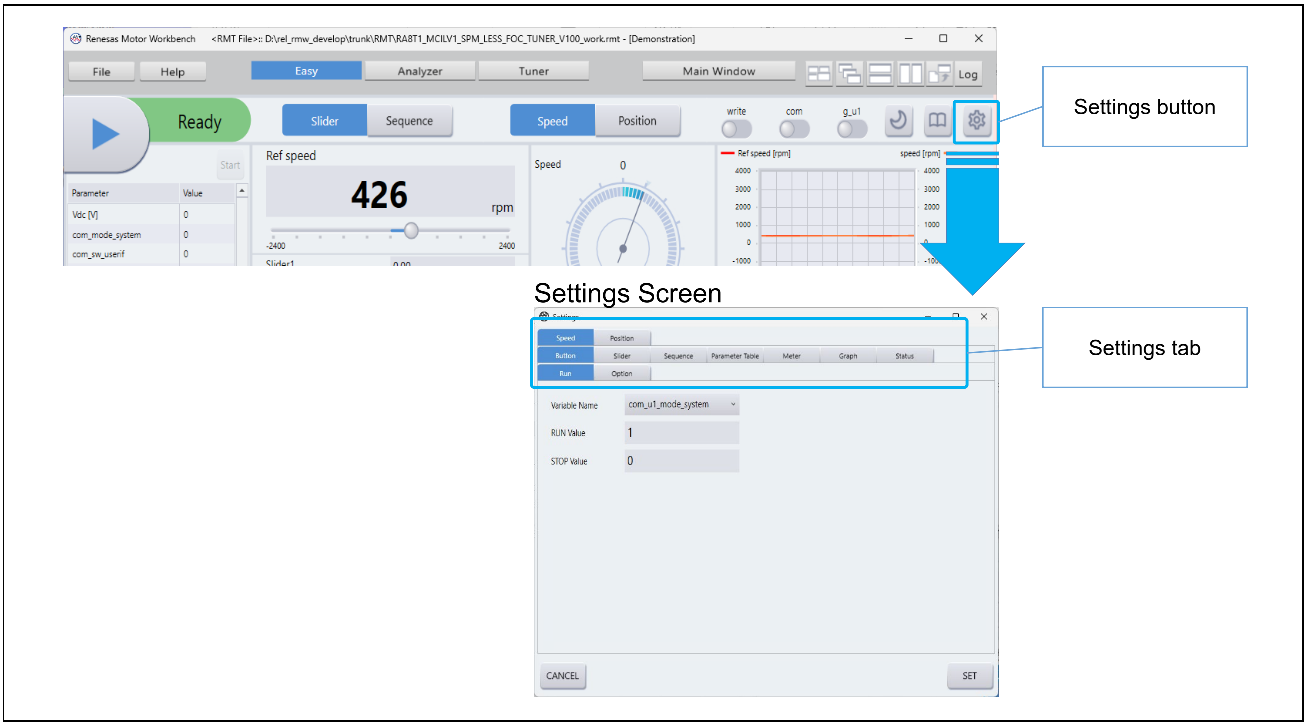


Figure 12-10 Displaying Setting Window

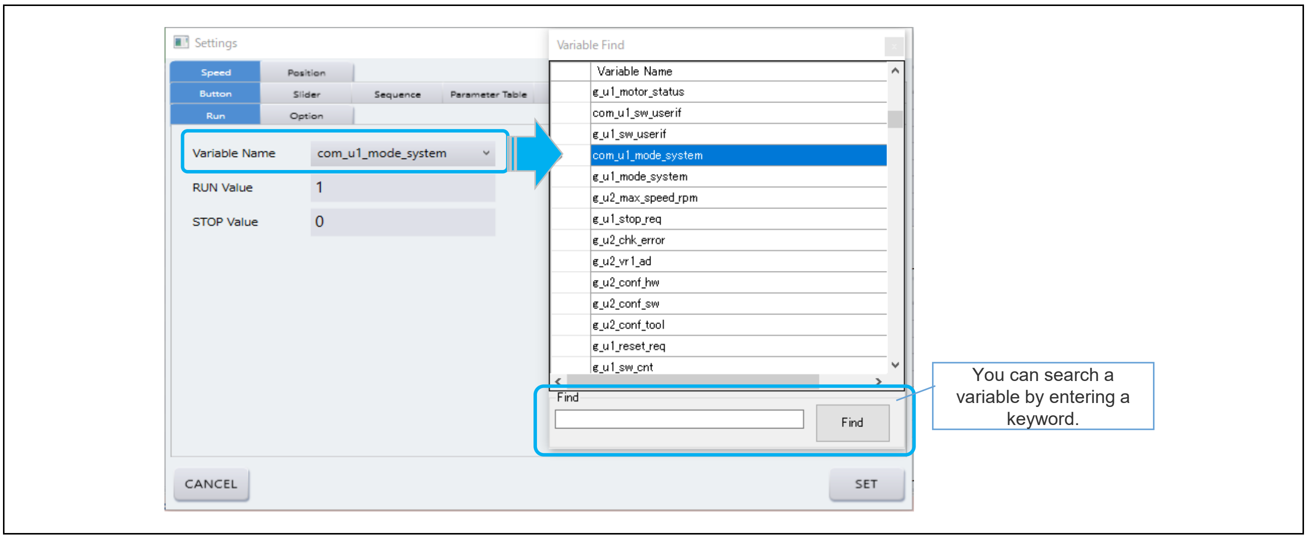


Figure 12-11 Example of Specifying Variable Name

12.6.1 RUN/STOP Button

The value is written to the variable each time you press the RUN/STOP button.

The following is an example of setting variables to associate with the RUN/STOP button.

Table 12-2 Example of Setting RUN/STOP Button

Item		Description
Settings tab	When Speed tab is selected	Speed → Button → Run
	When Position tab is selected	Position → Button → Run
RUN/STOP button		• When “com_u1_mode_system=1 (RUN Value)”, display . • When “com_u1_mode_system=0 (STOP Value)”, display .

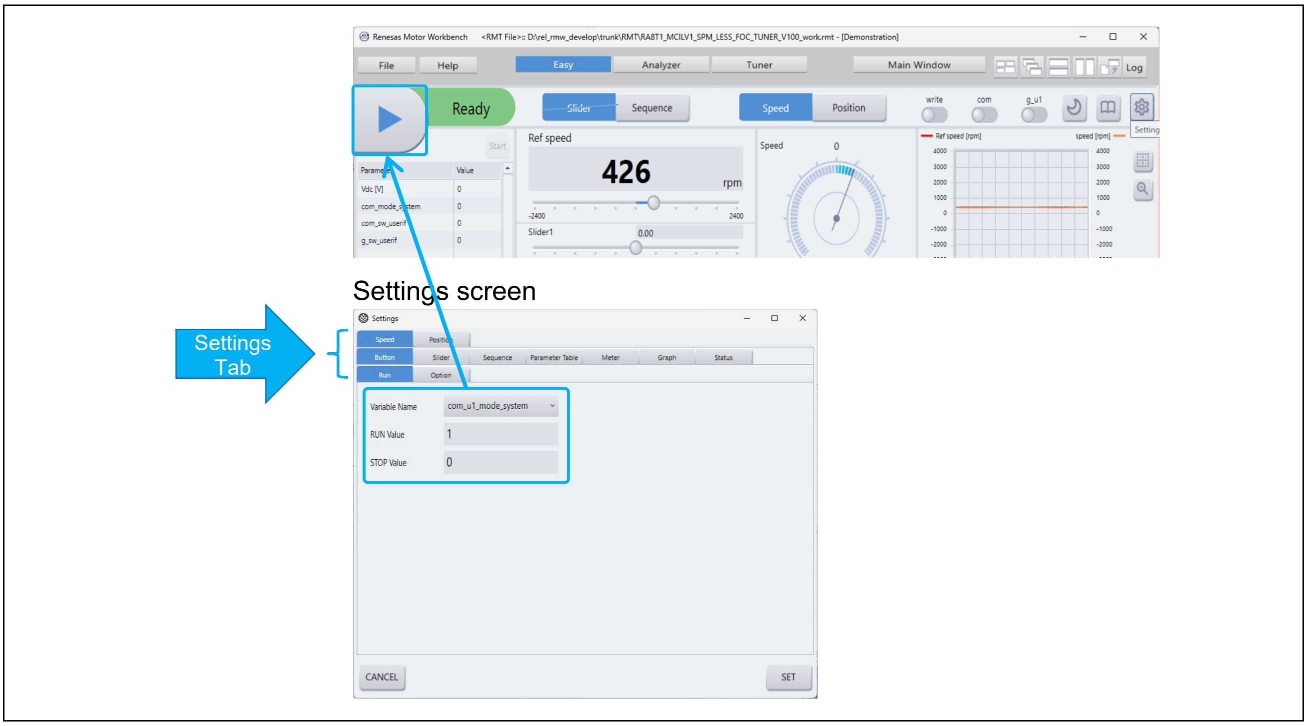


Figure 12-12 Example of Setting RUN/STOP Button

12.6.2 Status Display

If the status is matching the preset condition, the set color and status name will be displayed.

The following is an example of setting variables to associate with the status display.

Table 12-3 Example of Setting RUN/STOP Button

Item		Description
Settings tab	When Speed tab is selected	Speed → Status
	When Position tab is selected	Position → Status
Status display		• When "g_u1_motor_status=0", display Ready . • When "g_u1_motor_status=1", display Run . • When "g_u1_motor_status=2", display Error . • Update the status display at the cycle of 1000ms.

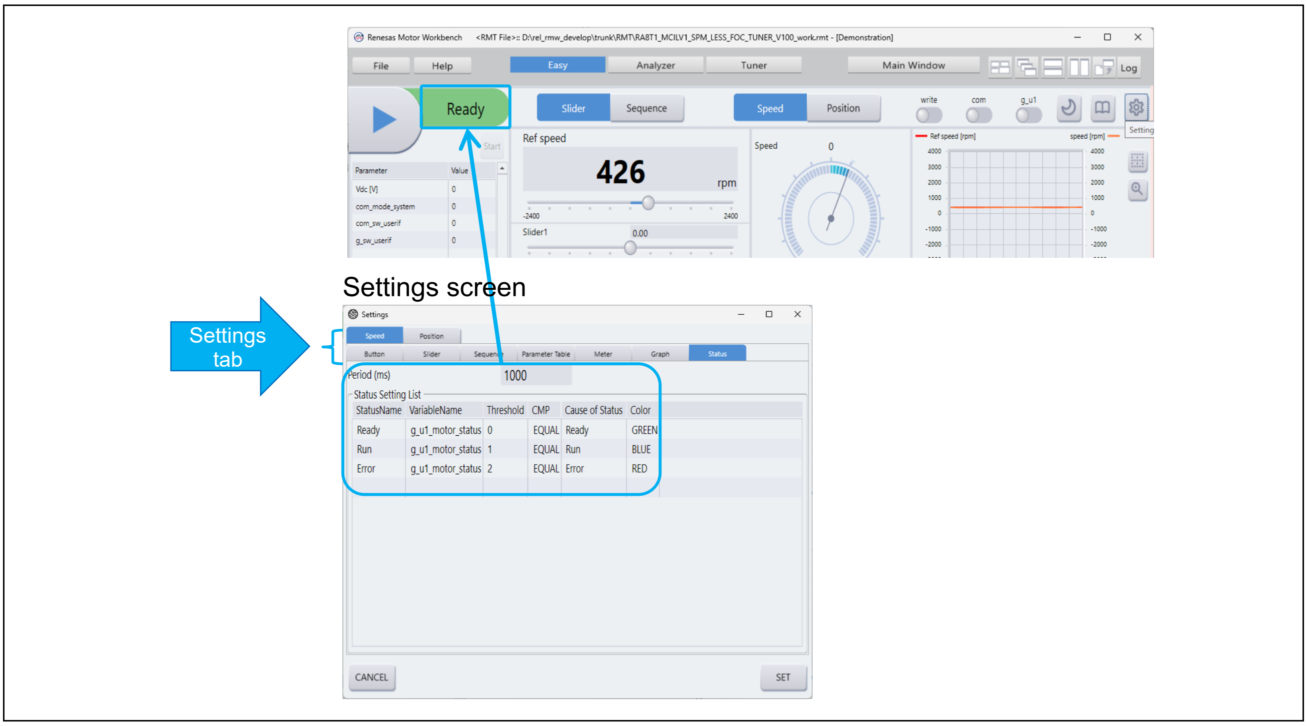


Figure 12-13 Example of Setting Status Display

12.6.3 Parameter Table

The variable value set in the parameter table is periodically read and displayed. Up to 20 parameters can be set in the parameter table.

The following is an example of setting variables to associate with the parameter table.

Table 12-4 Example of Setting Parameter Table

Item		Description
Settings tab	When Speed tab is selected	Speed → Parameter Table
	When Position tab is selected	Position → Parameter Table
Parameter table		• Display the value of the variable “g_f4_vdc_ad_monitor” in the “Vdc [V]” field. • Update the parameter table at the cycle of 1000ms.

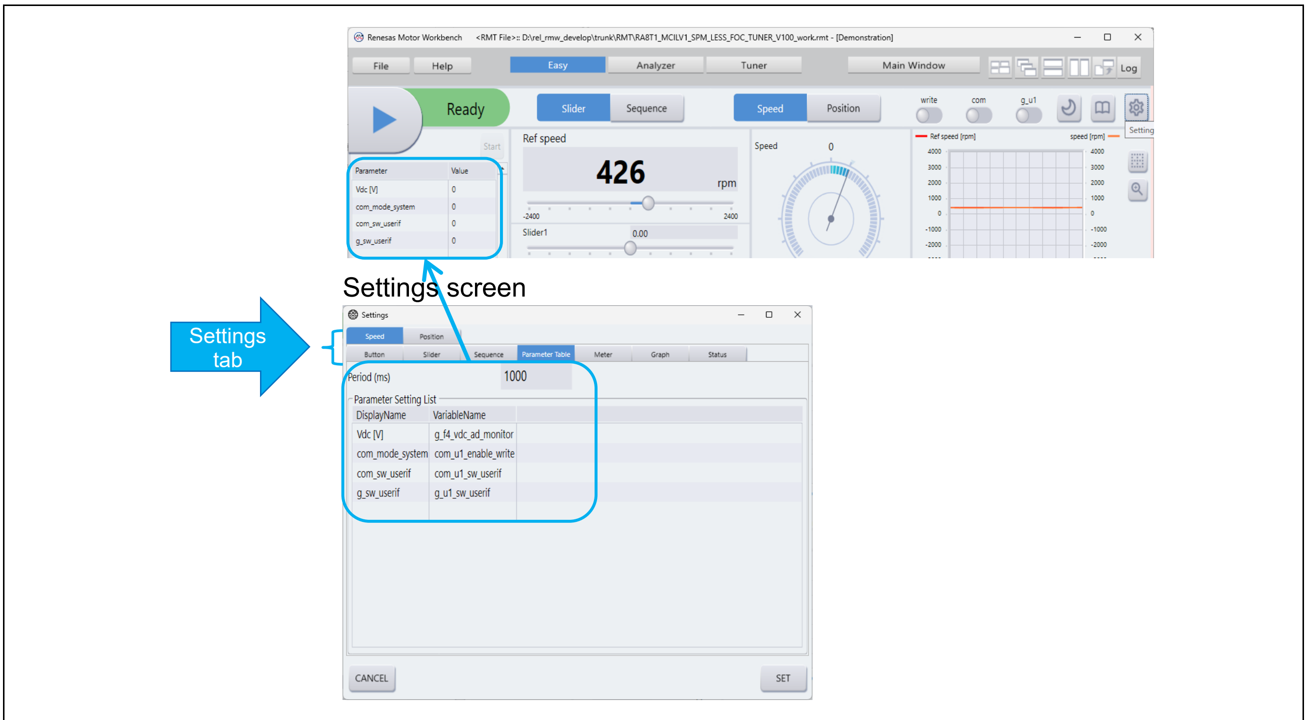


Figure 12-14 Example of Setting Parameter Table

12.6.4 Slider

Writes the value set by the slider operation to the variable. Up to 8 sliders can be set. If there is a variable of the rotation direction, you can set a variable to overwrite its value with a positive or negative slider. The following is an example of setting variables to associate with sliders.

Table 12-5 Example of Setting Slider

Item		Description
Settings tab	When Speed tab is selected	Speed → Slider → Slider1 to 8
	When Position tab is selected	Position → Slider → Slider1 to 8
Slider		• Display “Ref speed” in Slider1 • Set the ragne of ”Ref speed” between -2400rpm to 2400rpm. • Set the value set in “Ref speed” to the variable “com_f4_ref_speed_rpm” • Reflect the value of the slider to the variable at the cycle of 1000ms.

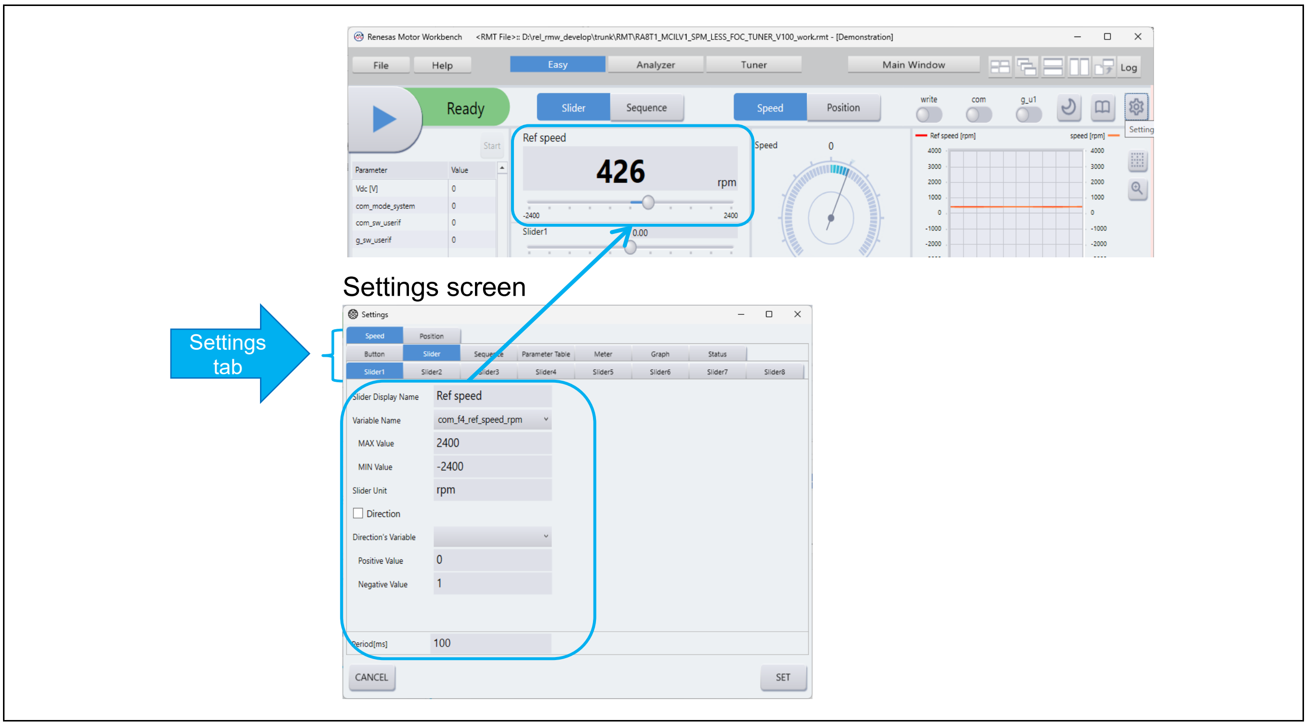


Figure 12-15 Example of Setting Slider

12.6.5 Sequence

Writes the value set in the sequence table to the variable. Up to 3 variables can be set in the sequence table.

The following is an example of setting variables to associate with the sequence.

Table 12-6 Example of Setting Sequence

Item		Description
Settings tab	When Speed tab is selected	Speed → Sequence → Table → Variable1 to 3
	When Position tab is selected	Position → Sequence → Table → Variable1 to 3
Sequence		• Display “Ref speed” in Variable1. • Sets the range of “Ref speed” between -2400 to 2400. • Set the value set in “Ref speed” to the variable “com_f4_ref_speed_rpm” • Display changes in the set value of “Ref speed” with [red] color in the graph.

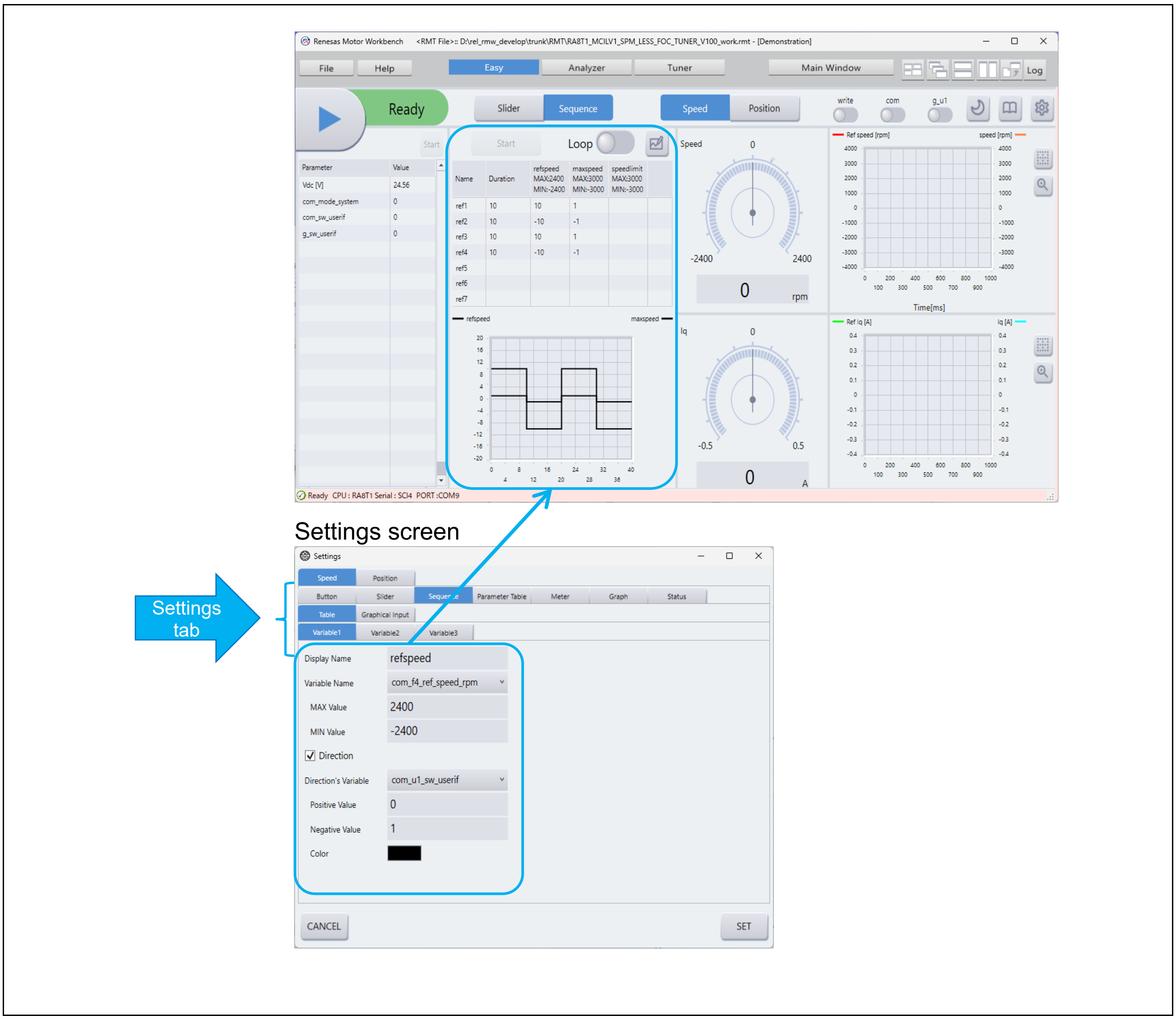


Figure 12-16 Example of Setting Sequence

12.6.6 Sequence Graphical Input

Writes the value set by the sequence graphical input to the variable.

The following is an example of setting variables to associate with the sequence graphical input.

Table 12-7 Example of Setting Sequence Graphical Input

Item		Description
Settings tab	When Speed tab is selected	Speed → Sequence → Graphical Input → Variable1 to 3
	When Position tab is selected	Position → Sequence → Graphical Input → Variable1 to 3
Sequence graphical input		• Display “Ref speed” in Variable1. • Set the range of “Ref speed” between -2400 to 2400. • Set the value set in “Ref speed” to the variable “com_f4_ref_speed_rpm.

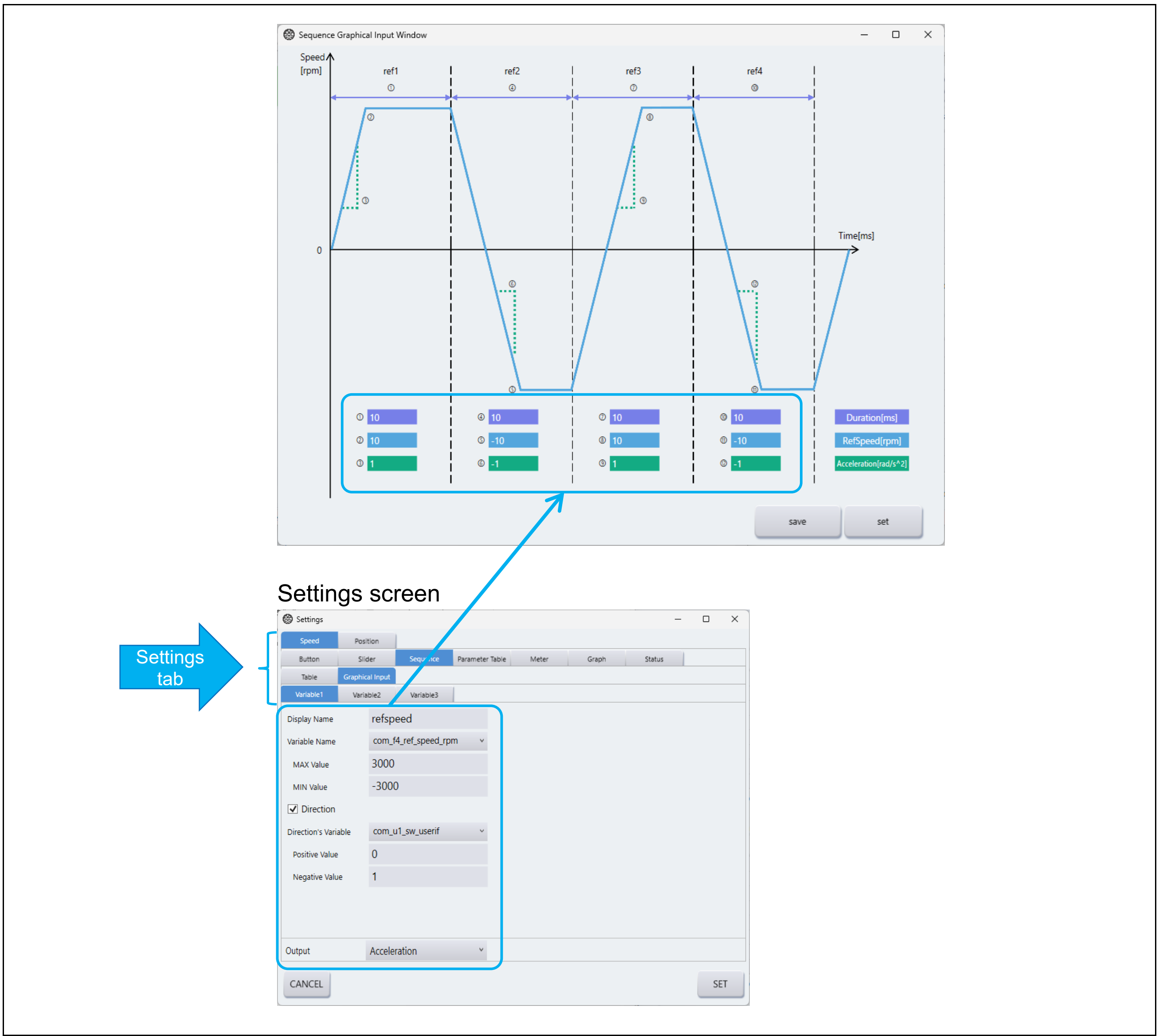


Figure 12-17 Example of Setting Sequence Graphical Input

12.6.7 Option Switch

A value is written to the variable when the Option switch is turned ON / OFF. Up to 3 option switches can be set.

The following is an example of setting variables to associate with the option switches.

Table 12-8 Example of Setting Option Switch

Item		Description
Settings tab	When Speed tab is selected	Speed → Button → Option
	When Position tab is selected	Position → Button → Option
Option switch		• Display "RMW UI" in Option2. • When "com_u1_sw_userif=0 (ON Value), display . • When "com_u1_sw_userif=1 (OFF Value), display .

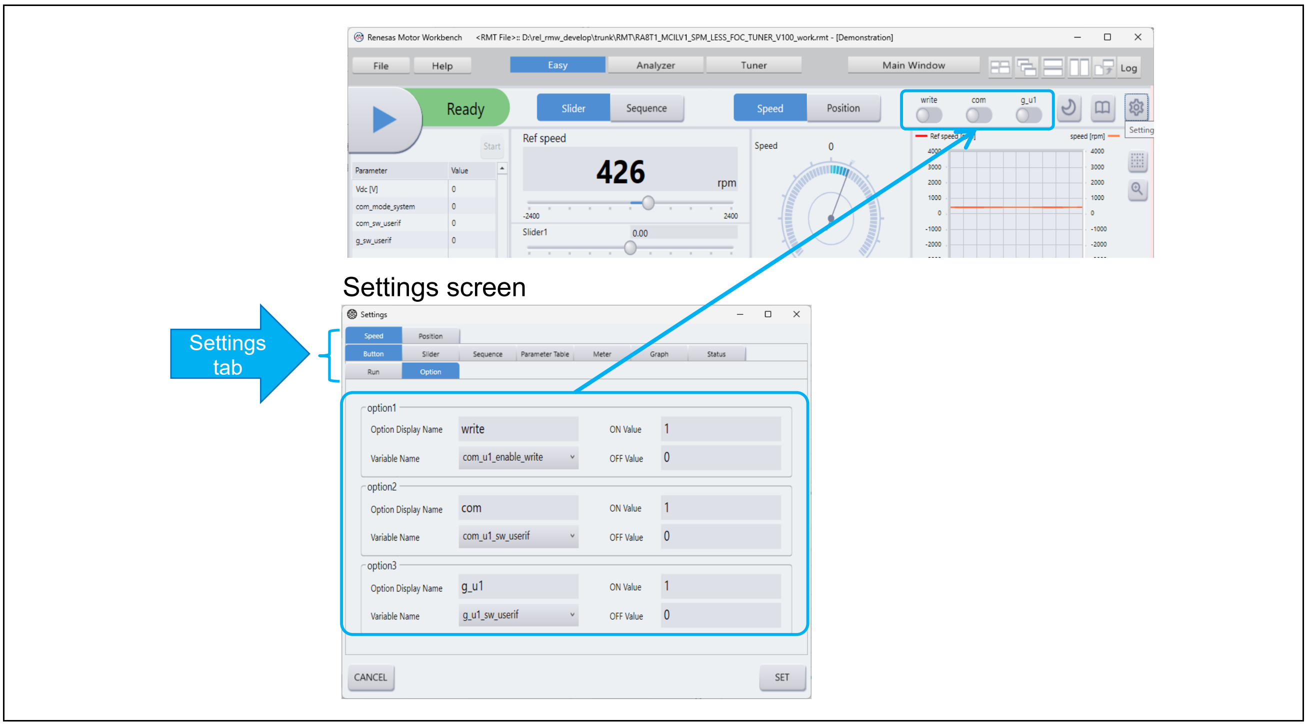


Figure 12-18 Example of Setting Option Switch

12.6.8 Meters

Displays the variable values in meters. Up to 2 meters can be displayed.

When you set the command value variable, the ▲ symbol indicates the position of the command value on the meter. When the meter type is Speed, there is a function that converts the unit from rad/s to rpm. The following is an example of setting variables to associate with the meters.

Table 12-9 Example of Setting Meter

Item		Description
Settings tab	When Speed tab is selected	Speed → Meter → Top/Bottom
	When Position tab is selected	Position → Meter → Top/Bottom
Meter		To set the upper meter, select “Top” tab. • Dispplay the meter with “Speed”. • Select “Speed” for the meter type. • Set the value of “g_f4_speed_rpm_monitor” to “Speed”. • Enable swithcing from rad/s to rpm and set the pole pairs to “com_u2_mtr_pp”. • Enable display of the command value variable and set the command value variable to “com_f4_ref_speed_rpm”. • Set the display range of the meter to -2400 to 2400. • Display the color (cyan) for before changing and the color (blue) for after changing. • Display the variable value on the meter at the cycle of 100ms.

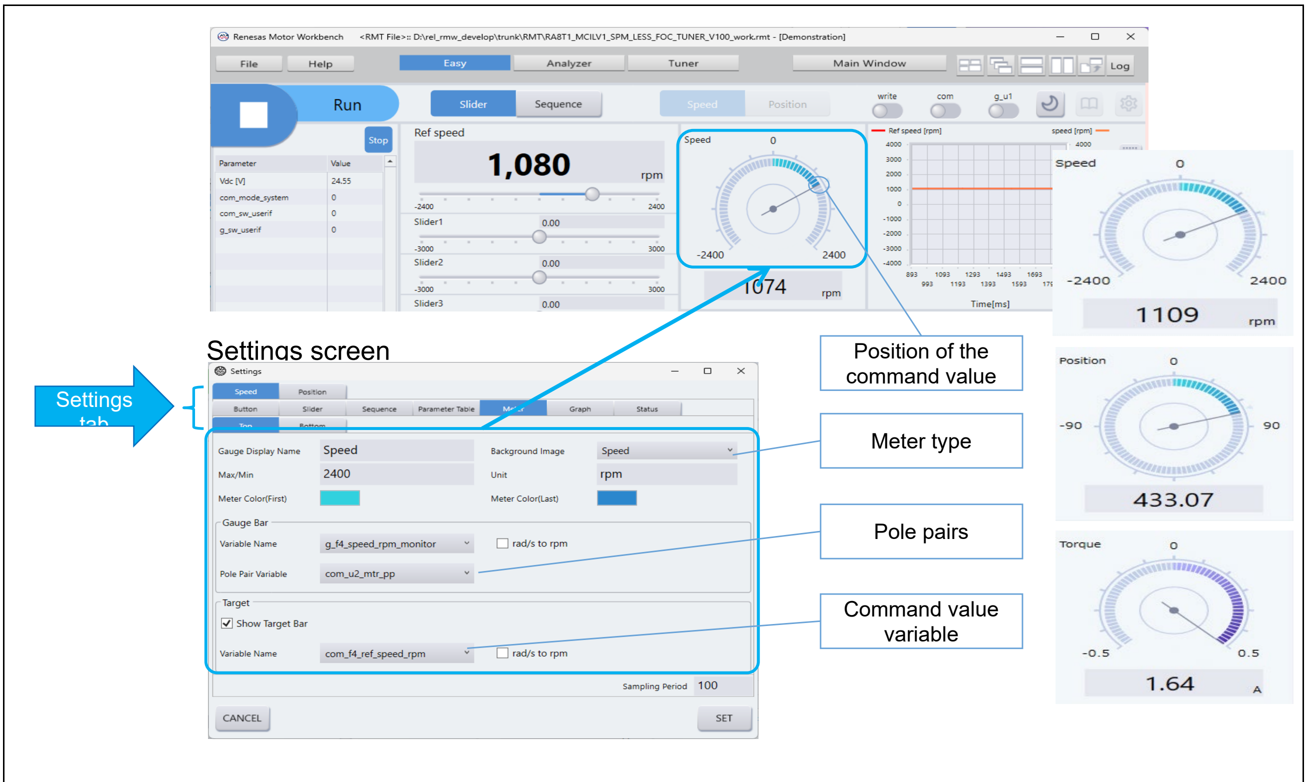


Figure 12-19 Example of Setting Meter

12.6.9 Graph

Displays the variable value in graphs. Up to 2 graphs can be displayed, and up to 2 variables can be displayed in each graph.

The following is an example of setting variables to associate with the graphs.

Table 12-10 Example of Setting Graphs

Item		Description
Settings tab	When Speed tab is selected	Speed → Graph → Top/Bottom
	When Position tab is selected	Position → Graph → Top/Bottom
Graph		To set the upper graph, select "Top" tab. • Set "Ref speed [rpm]" to Parameter1 and display it in the color (red). • Set "speed [rpm]" to Parameter2 and display it in the color (orange). • Set the value of "com_f4_ref_speed_rpm" to "Ref speed [rpm]". • Set the value of "g_f4_speed_rpm_monitor" to "speed [rpm]" • Set the variable type as single precision floating point type (FLOAT). • Set the X-axis scale to 100ms, the range of Y-axis between -1000 to 1000. • Display the variable value at the cycle of 100m.

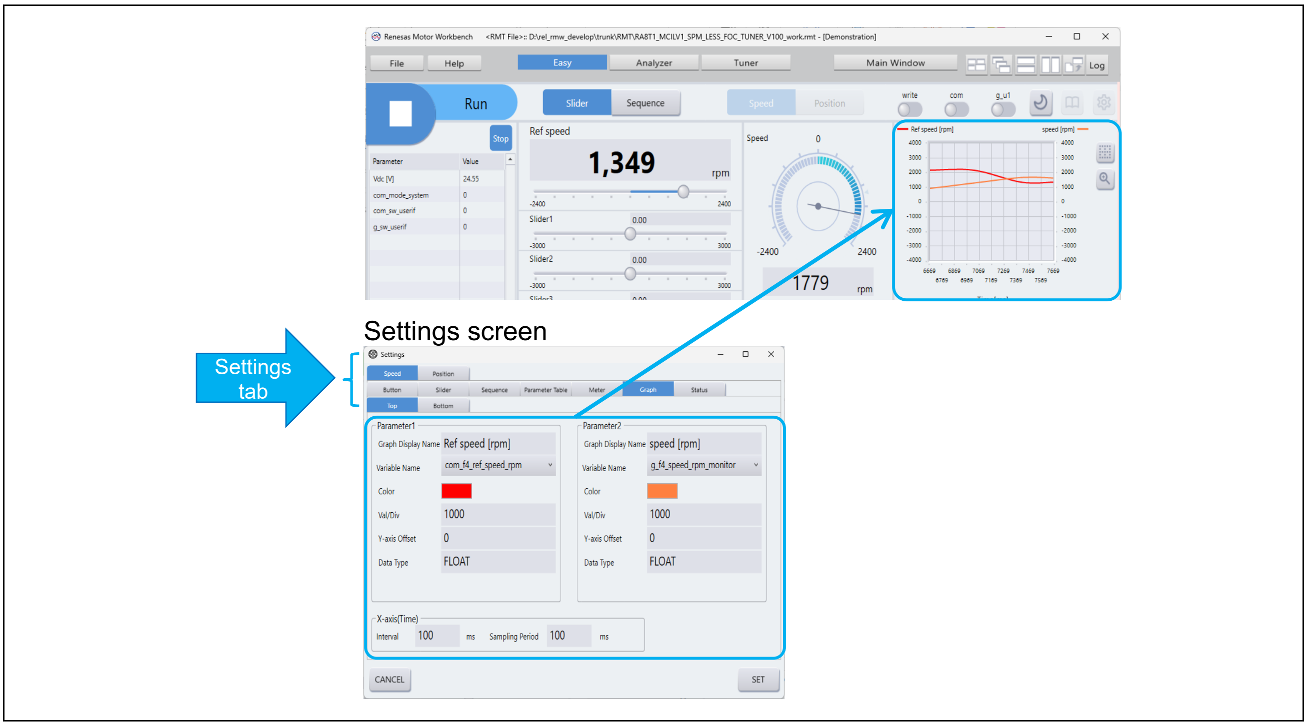


Figure 12-20 Example of Setting Graph

12.7 Switching Theme Color

You can switch the theme color of the screen by pressing on the upper-right of the GUI window.

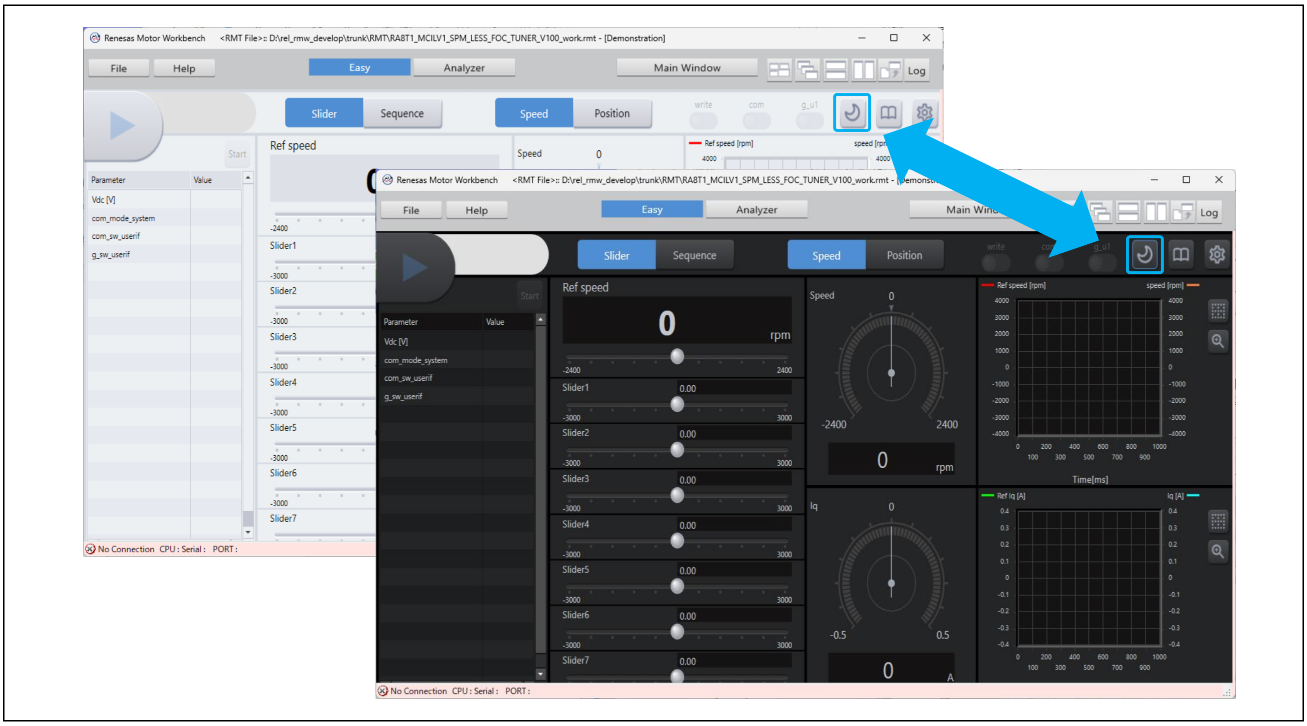


Figure 12-21 Switching Theme Color

12.8 Tutorial

When you click  on the upper-right of the GUI window, the tutorial is displayed.

The explanations are displayed sequentially while the item being explained is highlighted. To exit the tutorial, press “END”.

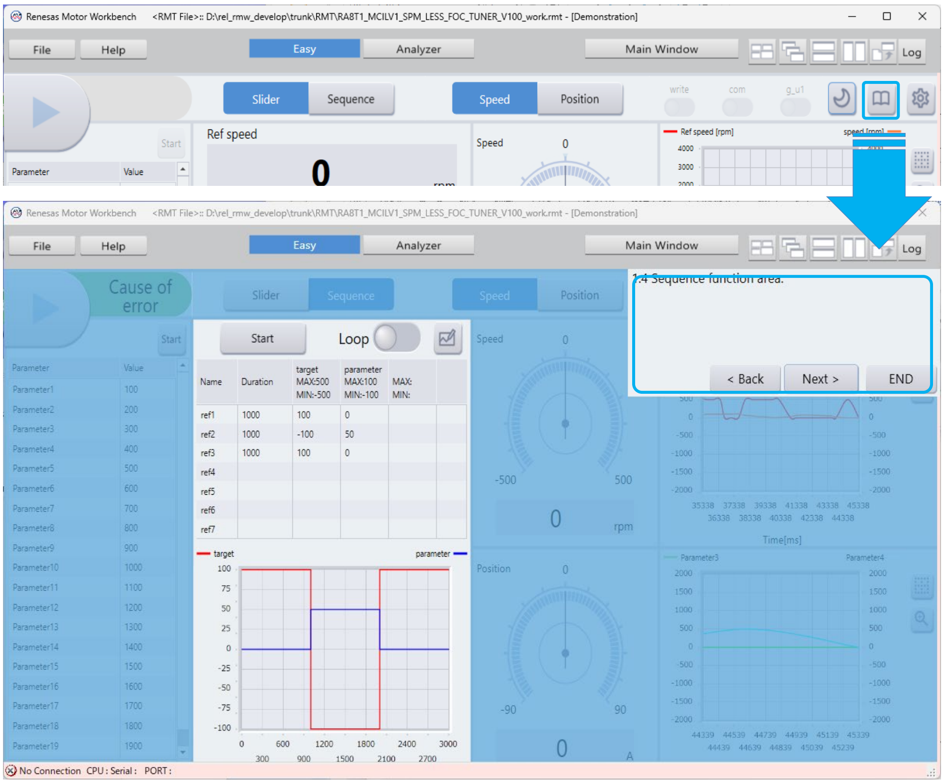


Figure 12-22 Example of Tutorial Display

13.Servo Tool

13.1 Overview

13.1.1 What is Servo Control?

This is a type of feedback control for accurate motor positioning operation. The position sensor built into the motor monitors and controls the motor's position and rotational speed. You can control the motor to match the target value by comparing the target value (position and rotation speed) and sensor measurement value and determining the command value.

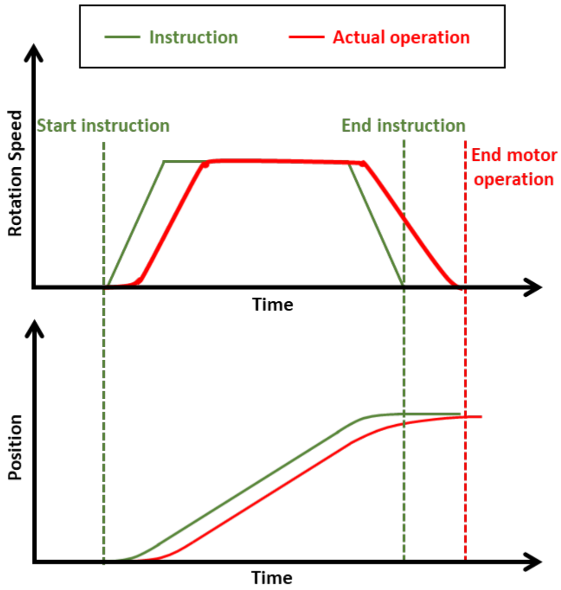


Figure 13-1 Relationship Between Commands and Actual Operations

To operate the motor according to commands, Position Control (control for position deviation) and Speed Control (control for speed deviation) are performed in combination. The difference between the command value and the actual value is called the deviation. A simple block diagram is shown below.

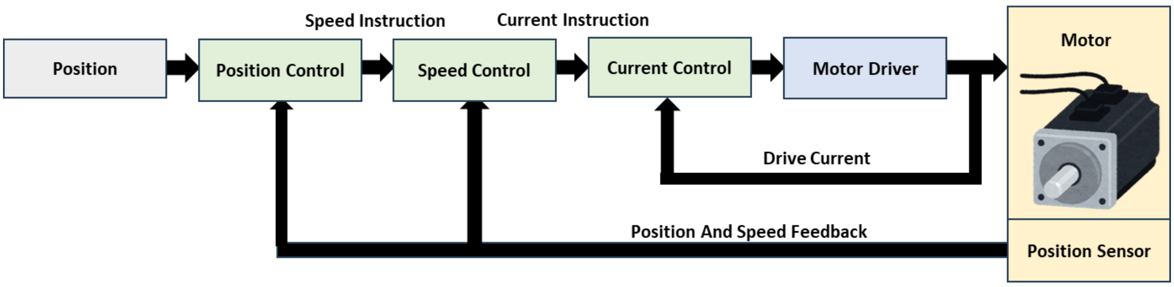


Figure 13-2 Block Diagram of Servo Control

In Position Control, the deviation between the user-specified position command value and the measured value of the position sensor is determined and the speed command value is calculated by proportional control. The "Position Control Frequency" affects the calculation.
In Speed Control, the deviation between the speed command value obtained from Position Control and the speed information calculated from position sensor measurements is determined and the current command value is calculated by PI control. The "Speed Control Frequency" and "Inertia" affect the calculation.
In Current Control, the voltage to be output as PWM is calculated from the current command value obtained from Speed Control to drive the motor driver.
It is important to adjust the responsiveness of Position Control and Speed Control appropriately to match the environment (equipment and mechanism) in which they are used. If the responsiveness is slow, it will take longer to reach the command value. If the responsiveness is excessively fast, the speed will easily fluctuate and operation will become unstable.

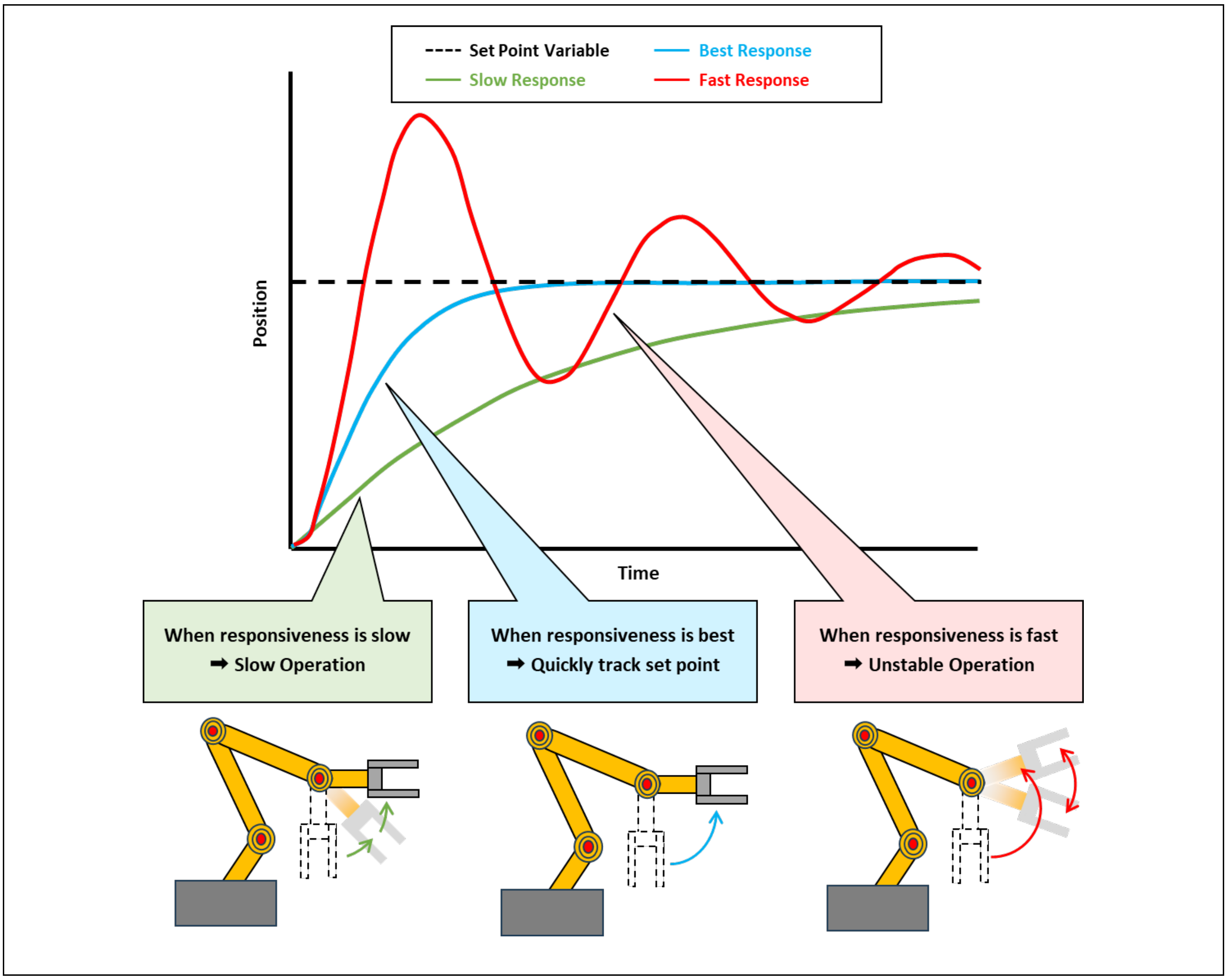


Figure 13-3 Servo Control Responsiveness

13.1.2 What is Servo Tool?

The Servo Tool supports parameter adjustment for Servo Control. It is available only when the servo function is supported by the control program. In that case, "Servo" will be displayed in the "Select Tool" of the Main Window with "Servo" selectable.
After debugging is completed on the motor stand-alone, attach it to the mechanism, it is used Servo Tool. The flow of using the Servo Tool is shown in Figure 13-4.

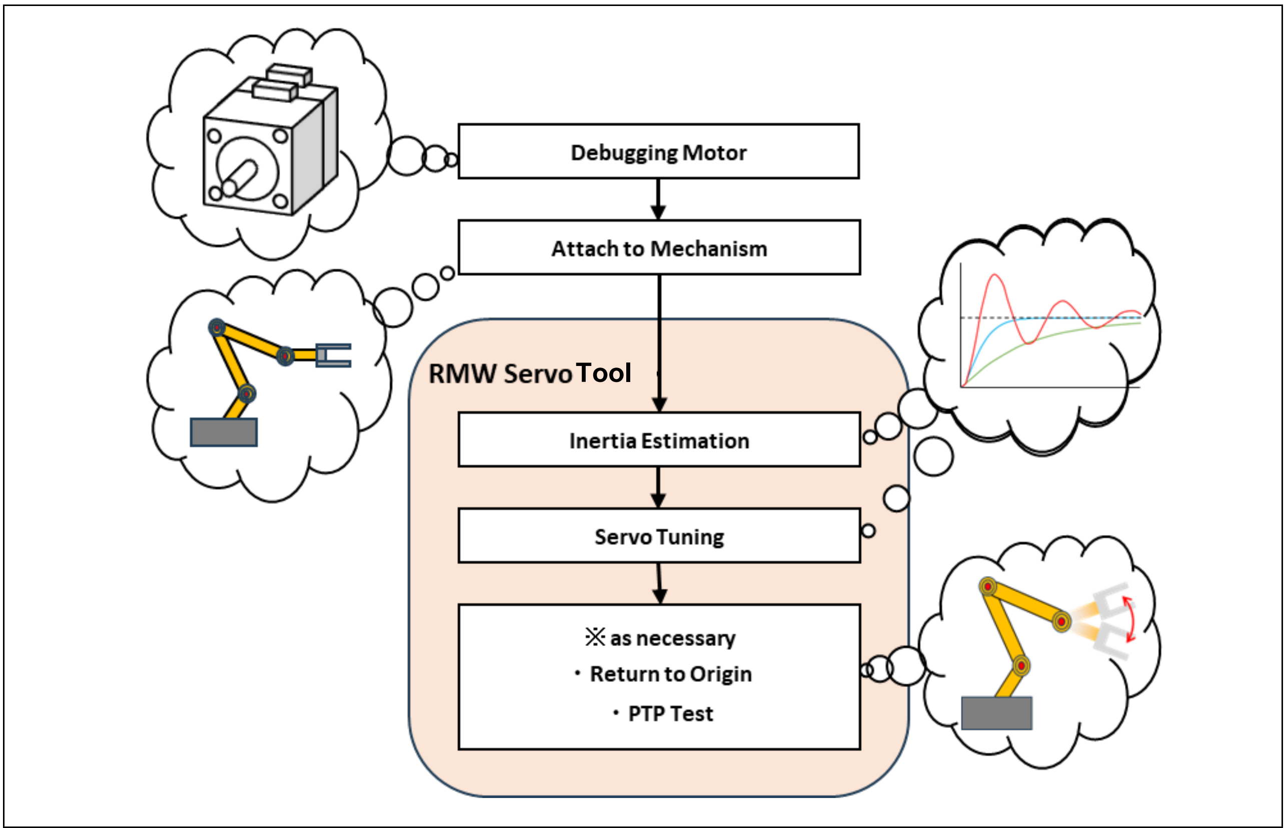


Figure 13-4 Flow of Servo Tool Use

After the motor is attached on the mechanism, "Inertia Estimation" should be performed first. Inertia is an important parameter that is the basis for Servo Control, as it is related to the calculation of acceleration (speed) and required torque (current). Since inertia affects all subsequent motor operations, be sure to perform Inertia Estimation and set as accurate a value as possible.

Finished Inertia Estimation, perform "Servo Adjustment", you should adjust the responsiveness of position control and speed control for Servo Control to match the environment (equipment/mechanism) to be used. After Servo Adjustment, perform a trial run of "Return to Origin" and "PTP operation" on the GUI as necessary to check the validity of the parameters.

13.2 Feature

- In Inertia Estimation, it can estimate the load inertia and the rotor inertia connected with the motor-axis.
- In Servo Adjustment, it can configure Servo Settings such as method and control natural frequencies.
- In Return to Origin, it can set the method for return to origin, the return speed, etc.
- In PTP operation, it can perform PTP (Point to Point) operation for one axis.

13.3 Inertia Estimation

13.3.1 Function Description

13.3.1.1 What is Inertia?

Inertia (moment of inertia) is the size of an object's attempt to maintain its current state. When inertia is small, it is easy for an object to turn and stop. When inertia is large, the object has difficulty turning and stopping.
The moment of inertia of the rotor of a motor is called Rotor Inertia, and the moment of inertia of the mechanism attached to the motor is called Load Inertia.

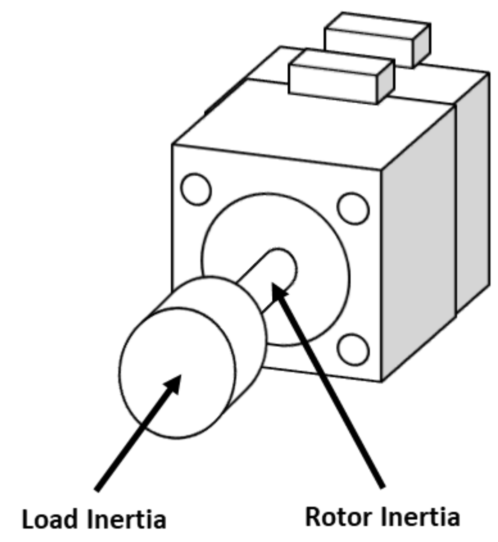


Figure 13-5 Load Inertia and Rotor Inertia

13.3.1.2 What is Inertia Estimation?

Inertia Estimation is the process of estimating the total of the rotor inertia and load inertia connected to the motor axis, determining the ratio to the motor rotor inertia (rotor inertia ratio), and then adjusting the torque. Since the motor rotates during inertia estimation, please be careful not to collide with people or objects around the moving parts.

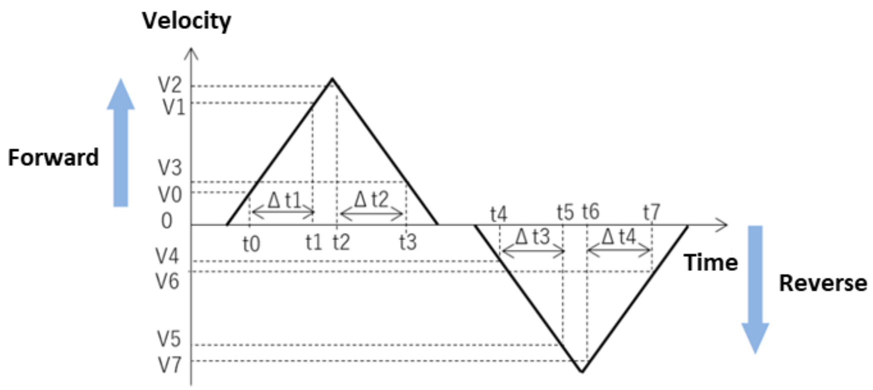


Figure 13-6 Operation During Inertia Estimation

Inertia Estimation uses Position Control to run the motor in the forward and reverse directions with the same amount of rotation. Inertia Estimation is performed using the acceleration time in the forward direction, deceleration time in the forward direction, acceleration time in the reverse direction, and deceleration time in the reverse direction, the average drive current and average acceleration during each time.

13.3.1.3 Importance of Inertia Estimation

Inertia is an important parameter that is relevant to the calculation of acceleration (speed) and required torque (current) and serves as a reference for Servo Control. The range of influence of inertia estimation in Servo Control is shown below.

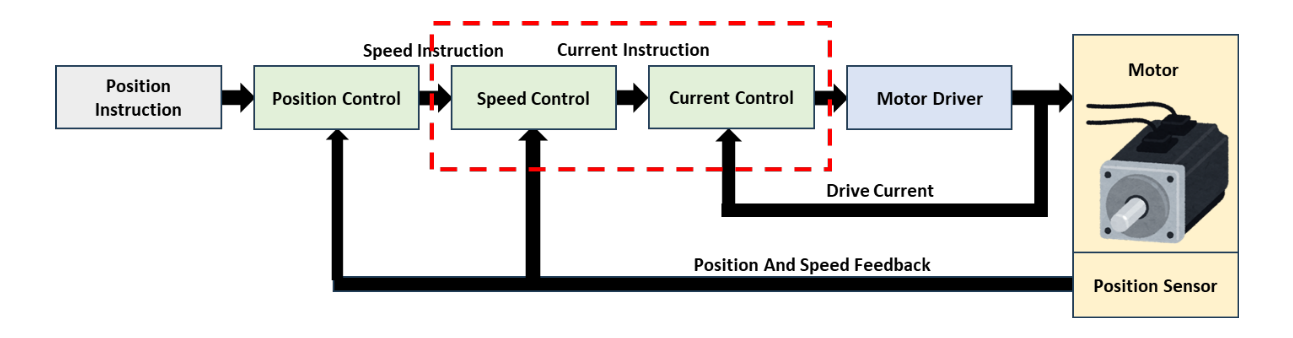


Figure 13-7 Range of Influence of Inertia Estimation in Servo Control

If the estimated inertia is at the correct value, accurate torque can be calculated. If it is lower than the correct value, a lower torque is calculated, resulting in slower response. If it is higher than the correct value, a higher torque is calculated, and the motor is likely to operate erratically.
Since inertia affects all motor operation, it is important to set as accurate a value as possible; not only is it important to set an accurate value in the Servo tool, but it is also important to know the actual value in the program development.

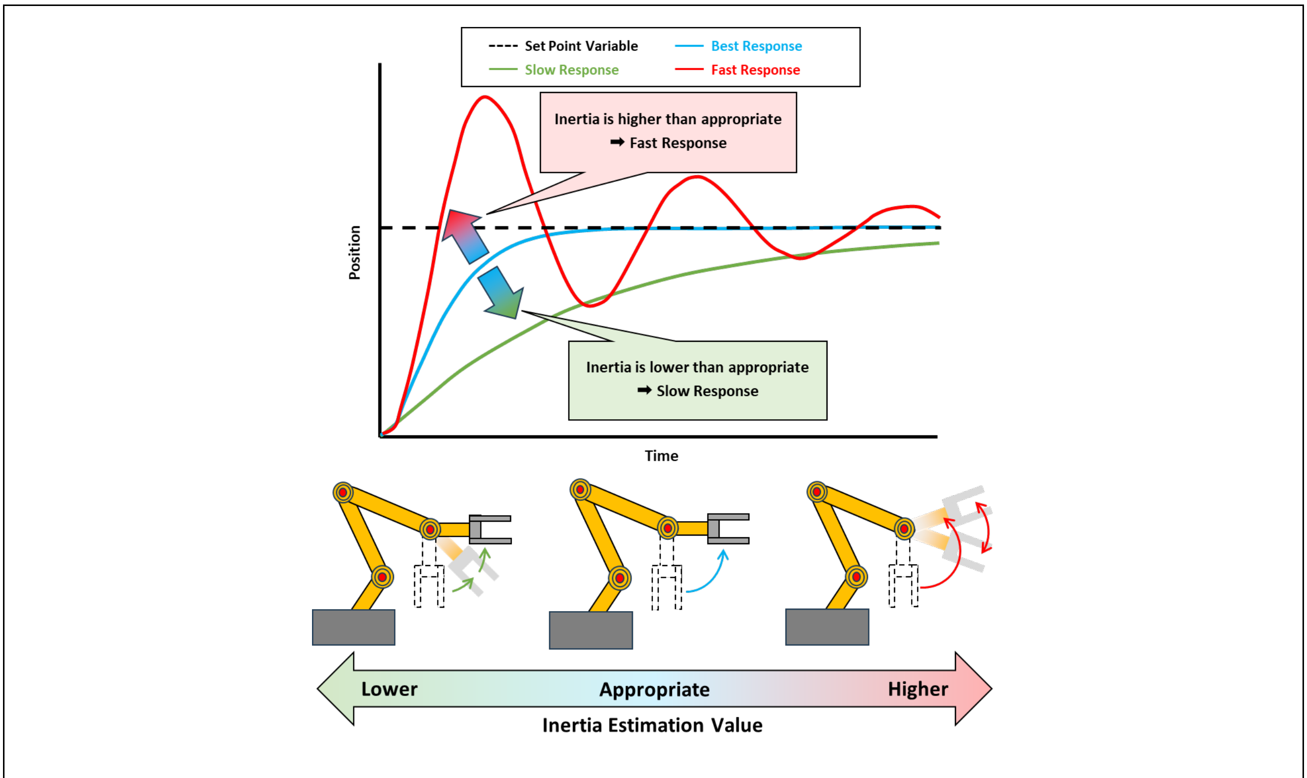


Figure 13-8 Effect of Inertia on Servo Control

13.3.2 Window Structure・Parameter List

This section explains each part of Inertia Estimation Tab window and its function.
In Inertia Estimation tab, the inertia connected to the motor axis can be estimated.

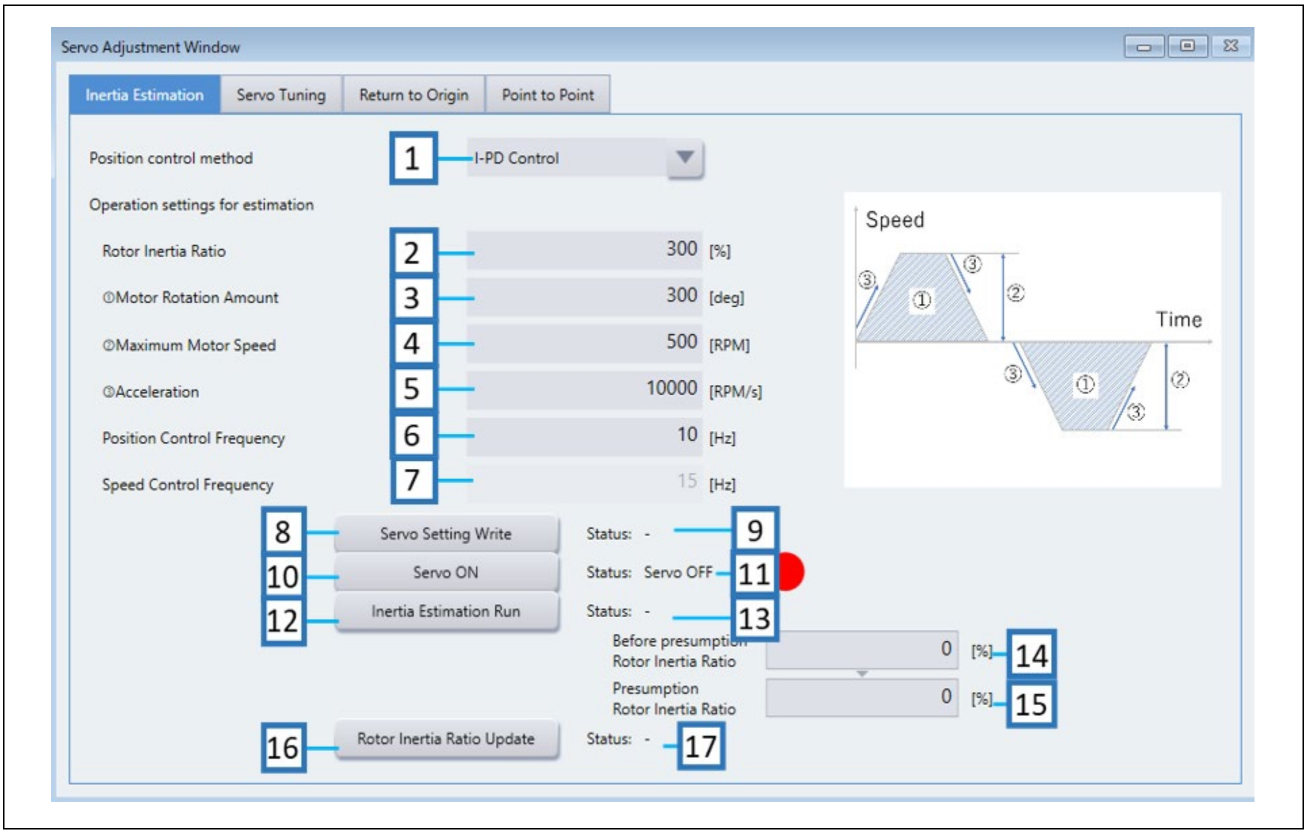


Figure 13-9 Inertia Estimation Tab Window

Table 13-1 Functions of Inertia Estimation Tab Window

No.	Name	Explanation
1	Position control method	Specify the method I-PD Control/PID Control for position control. Basically, there is no problem using the default value "I-PD Control".
2	Rotor Inertia Ratio	Set the rotor inertia ratio. If already calculated, enter that value. If unknown, use the default value (300%) for estimation.
3	Motor Rotation Amount	Set the motor rotation amount. Set the amount of rotation that does not hit surrounding people or objects. If the rotation amount is small, estimation is likely to fail. If the amount of rotation is too small, estimation is more likely to fail
4	Max Motor Speed	Set the maximum motor speed.
5	Acceleration	Set the acceleration.
6	Position Control Frequency	Set the position control natural frequency. Basically, there is no problem if the default value of 10 Hz is used. (Adjustment is made in Section 13.4 Servo Adjustment.)
7	Speed Control Frequency	Set the speed control natural frequency. You can enter it when “Position control method” is “PID Control”. Set the value to 1.5 to 3.0 times the natural frequency of the Position Control system of No. 6. (Adjustment is made in Section 13.4 Servo Adjustment.)
8	Servo Setting Write button	Write the settings from No.1 to 7.
9	Servo Setting Write Status	Displays the writing status after pressing the Servo Setting Write button (No.8).
10	Servo ON/OFF button	Switches ON/OFF of servo. Labels of the button and operations when clicked: Servo ON: Servo control starts when clicked. Servo OFF: Servo control stops when clicked.
11	Servo Status	Displays the servo status. Green: servo is ON. Red: servo is OFF.
12	Inertia Estimation Run/Stop button	Inertia estimation starts when “Run” is clicked. Inertia estimation stops when “Stop” is clicked. Labels of the button: During inertia estimation: “Inertia Estimation Stop” While inertia estimation is stopped: “Inertia Estimation Run”
13	Inertia Estimation Status	Displays the status of inertia estimation.
14	Before presumption Rotor Inertia Ratio	Displays the rotor inertia ratio before presumption.
15	Presumption Rotor Inertia Ratio	Displays the presumption rotor inertia ratio.
16	Rotor Inertia Ratio Update button	Writes the presumption rotor inertia ratio.
17	Rotor Inertia Ratio Update Status	Displays the status of writing the presumption rotor inertia ratio.

13.3.3 Explanation of Operation

The procedure of Inertia Estimation is shown below.

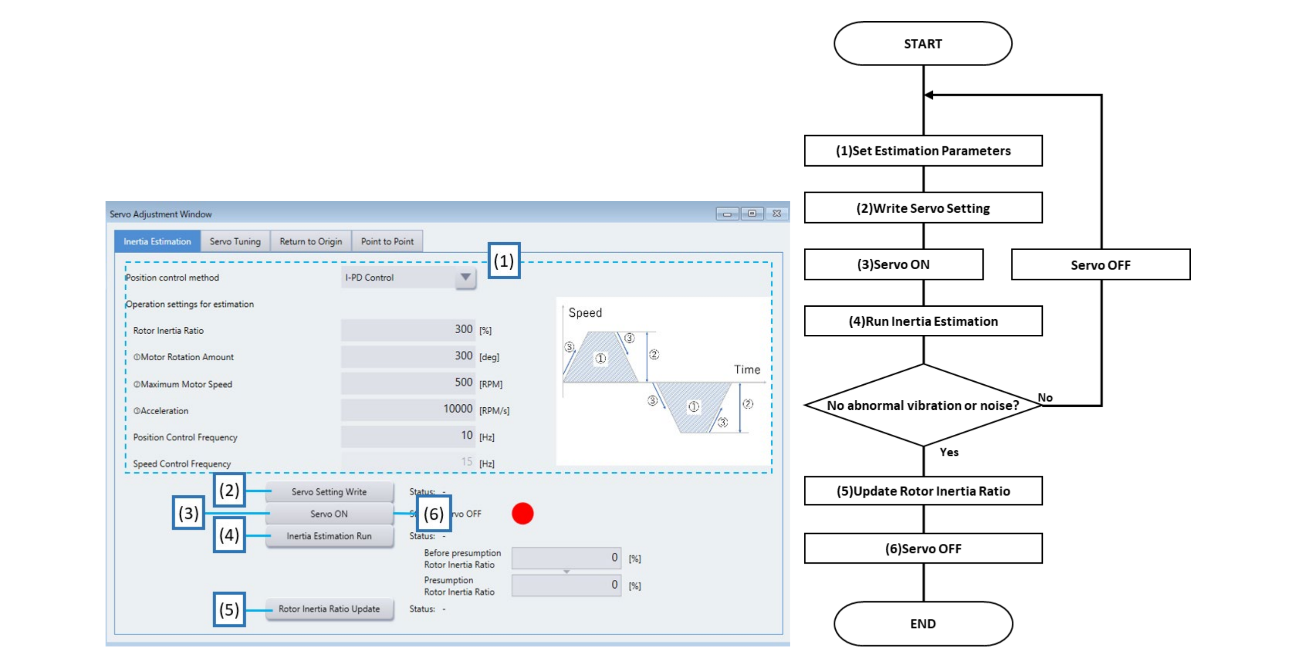


Figure 13-10 Inertia Estimation Operation Procedure

(1) Setting Parameters for Inertia Estimation

Sets the parameters for Inertia Estimation.
(For details, see Section 13.3.2 Window Structure・Parameter List)

(2) Writing Parameters for Inertia Estimation

Clicking the "Servo Setting Write" button writes the parameters set in (1).

(3) Start of Servo Control

Servo Control is started by pressing the "Servo ON" button before the start of Inertia Estimation.

(4) Start/Stop of Inertia Estimation

Inertia Estimation is started when the "Inertia Estimation Run" button is pressed in the "Servo ON" state. After Inertia Estimation is completed, it will automatically stop. Press the "Inertia Estimation Stop" button and Inertia Estimation is stopped.
※If abnormal vibration or noise is heard from the motor, press the "Servo OFF" button to terminate Servo Control. Return to (1) and review the parameter ("Rotor Inertia Ratio").
※If there is no problem, proceed to the next step.

(5) Writing the Estimated Rotor Inertia Ratio

After completing the Inertia Estimation, click the "Rotor Inertia Ratio Update" button to write the estimated rotor inertia ratio.

(6) Stop of Servo Control

Press the "Servo OFF" button to stop Servo Control.

Parameter Setting Guide・Notes

- If abnormal vibration or noise is heard, review the "Rotor Inertia Ratio".
  If the "Rotor Inertia Ratio" does not improve, reduce the "Position Control Frequency". However, set the "Speed Control Frequency" to 1.5 to 3.0 times the "Position Control Frequency.
- If the "Position Control Frequency" is low, the position tracking performance will decrease.
- If the "Speed Control Frequency" is too high, the motor will vibrate and make noise.
- For "Position control method," select "PID Control" for fast response or "I-PD Control" for disturbance suppression while suppressing overshoot.
- Inertia Estimation is based on the drive current during motor operation. Therefore, under conditions where the drive current is low, such as when the amount of movement is small, the operating speed is small, or acceleration is small, it is easy to estimate a smaller inertia. Inertia Estimation should be performed by increasing the amount of movement, operating speed, and acceleration to the extent possible.

13.4 Servo Adjustment

13.4.1 Function Description

13.4.1.1 What is Servo Adjustment?

Servo Adjustment adjusts Position Control responsiveness and Speed Control responsiveness appropriately. to make Servo Control suitable for the environment (equipment/mechanism) in which it is used.
(See Section 13.1 Overview for more information.)

13.4.1.2 What is Control Frequency?

Position Control and Speed Control are controlled in combination, and the response frequency of each control is called Position Speed Control Frequency and Speed Control Frequency.
Setting a higher Position Control frequency will increase the gain of the Position Control loop, thereby minimizing position deviation. Do not set the Position Control frequency higher than necessary.

If the Speed Control Frequency is set higher, the gain of the Speed Control loop is increased, which amplifies speed fluctuations caused by external disturbances and may result in increased vibration and noise. The maximum Speed Control frequency should be about 100 Hz and should not be set higher than necessary.

To ensure the stability of the control loop, the Speed Control frequency should be 1.5 to 3.0 times the Position Control frequency.

13.4.1.3 About IPD Control

IPD Control is a control method in which only the integral works on the deviation, and the proportional and derivative work only on the operating amount (output of the controller). This reduces vibration during positioning even with increased responsiveness, and although responsiveness is slower than with PID Control, positioning accuracy is improved.

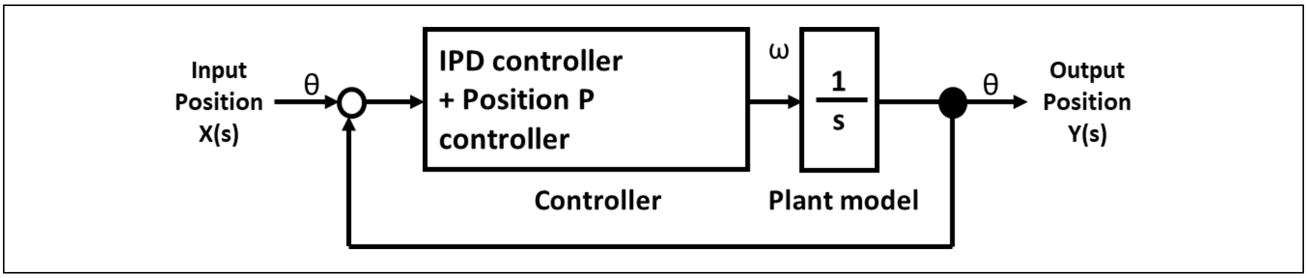


Figure 13-11 Model of IPD Control (Position)

13.4.1.4 Servo Adjustment Example

The waveforms of the detected speed when the Position Control Frequency and Speed Control Frequency are changed are shown below. Speed Control Frequency is set to 1.5 times the Position Control Frequency, and all operating parameters (＝Simplified Test settings, Servo Tuning tab Window,Figure13.13) are the same.

① A low Position Control responsiveness causes large Speed Control fluctuations and position deviations. Increase the Position Control Frequency to reduce Speed Control and Position Deviation.

② A high Speed Control Frequency will cause the motor to vibrate and generate noise. Reduce the Speed Control Frequency to reduce vibration and noise.

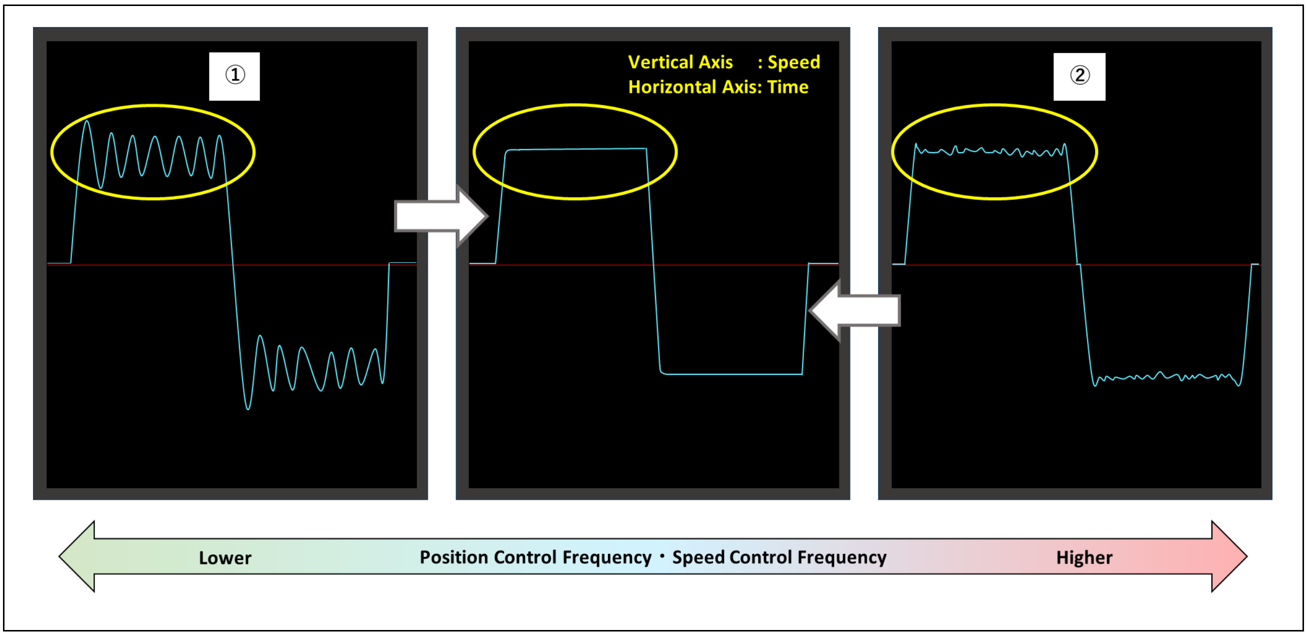


Figure 13-12 Waveforms of Control Frequency and Detection Speed

13.4.2 Window Structure・Parameter List

This section explains each part of “Servo Tuning” tab window and its function.
In Servo Tuning tab, you can perform the servo settings such as the position control method and control natural frequencies.

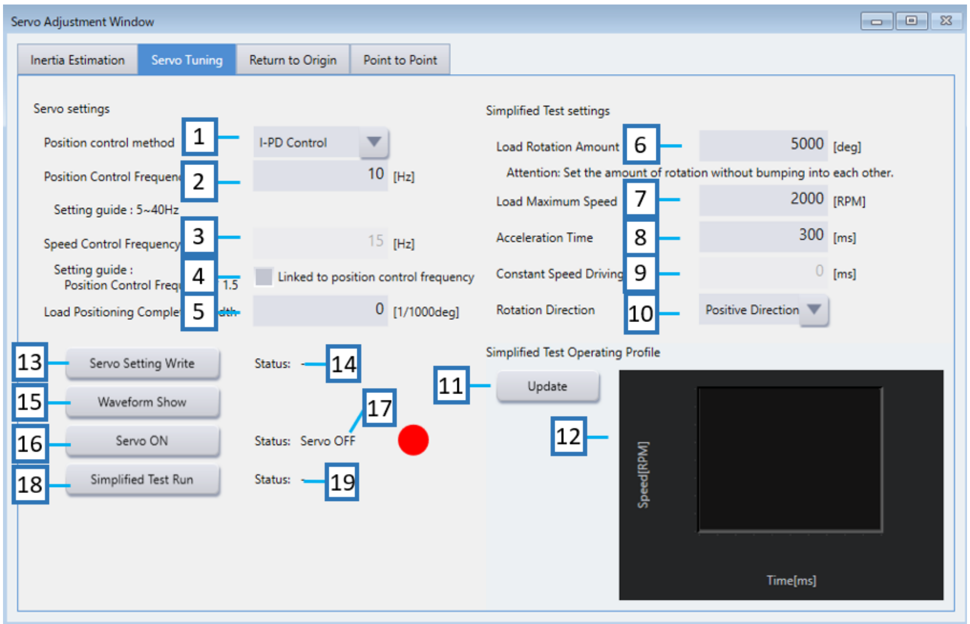


Figure 13-13 Servo Tuning Tab Window

Table 13-2 Functions of Servo Tuning Tab Window

No.	Name	Explanation
1	Position control method	Specify the method I-PD Control/PID Control for position control. Basically, there is no problem using the default value "IPD".
2	Position Control Frequency	Set the position control natural frequency.
3	Speed Control Frequency	Set the speed control natural frequency. You can enter it when “Position control method” is “PID Control”. The value should be 1.5 to 3.0 times the natural frequency of the Position Control system in No. 2.
4	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.
5	Load Positioning Completion	Set the range of the load positioning completion. Set tolerance margin from target position.
6	Load Rotation Amount	Set the load rotation amount.
7	Load Maximum Speed	Set the load maximum speed. Set the speed expected by the system.
8	Acceleration Time	Set the acceleration time. Set the time it takes to reach maximum speed from "STOP" state.
9	Constant Speed Driving Time	Set the constant speed driving time.
10	Rotation Direction	Set the rotation direction.
11	Update button	Updates the display of operation profile.
12	Operation profile	Displays the operation profile of the settings.
13	Servo Setting Write button	Writes settings from No.1 to 10.
14	Servo Setting Write Status	Displays the writing status after pressing the Servo Setting Write button (No.13).
15	Waveform Show button	Displays the waveform window.
16	Servo ON/OFF button	Switches ON/OFF of servo. Labels of the button and operations when clicked: Servo ON: Servo Control starts when clicked Servo OFF: Servo control stops when clicked.
17	Servo Status	Displays the servo status. Green: servo is ON. Red: servo is OFF.
18	Simplified Test Run/Stop button	When “Run” is clicked, the simplified test starts. When “Stop” is clicked, the simplified test stops. Labels of the button: During the simplified test: “Simplified Test Stop” While the simplified test is stopped: “Simplified Test Run”
19	Simplified Test Status	Displays the status of the simplified test execution.

The relationship between parameters and profiles is shown below.

① No. 6 (Load Rotation Amount)
② No. 7 (Load Maximum Speed)
③ No. 8 (Acceleration Time)
④ No. 9 (Constant Speed Driving Time)

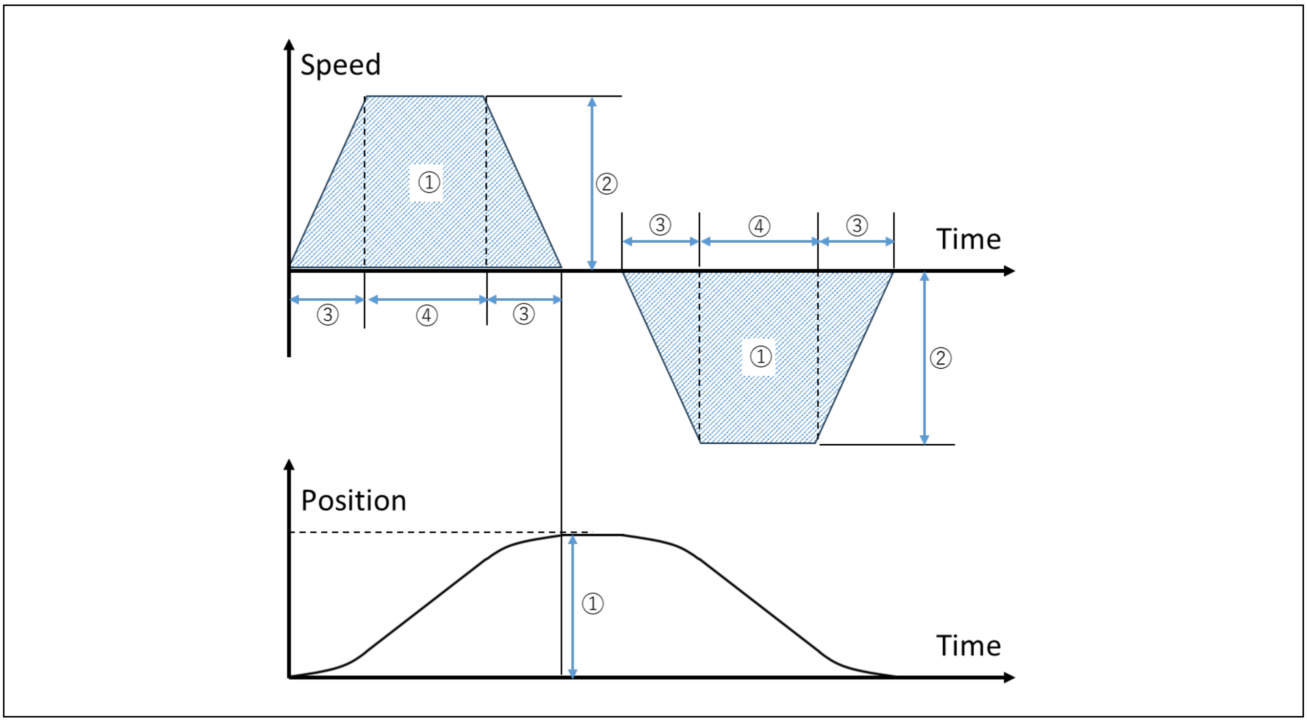


Figure 13-14 Relationship Between Parameters And Profiles

13.4.3 Explanation of Operation

The Servo Adjustment procedure is shown below.

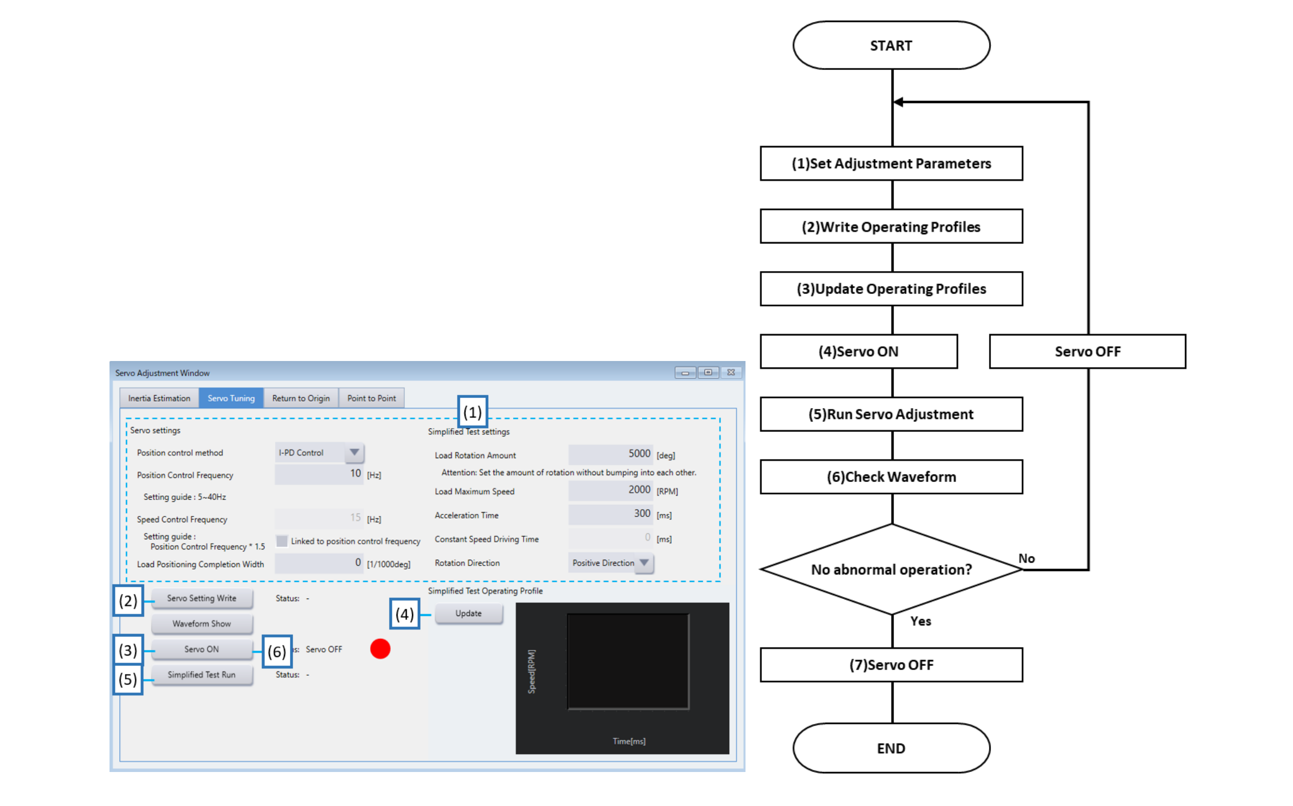


Figure 13-15 Servo Adjustment Operation Procedure

(1) Setting Parameters for Servo Adjustment

Set parameters for servo adjustment.
(For details, refer to Section 13.4.2 Window Structure・Parameter List.)

(2) Writing Servo Adjustment Parameters

When the "Servo Setting Write" button is pressed, the parameters set in (1) are written.

(3) Update of the Operation Profile

Clicking the "Update" button updates the profile.

(4) Start of Servo Control

Before starting servo tuning, press the "Servo ON" button to start Servo Control.

(5) Start/Stop of Servo Adjustment

Press the "Simplified Test Run" button in the "Servo ON" state to start servo adjustment.
After the Servo Adjustment is completed, the servo automatically stops. Clicking the "Simplified Test Stop"

(6) Checking the Waveform

Press the "Waveform Show" button to display the waveform. If you find it necessary to adjust the servo parameters, click the "Servo OFF" button to terminate Servo Control. Go back to (1) and review the parameters ("Position Control Frequency").
If there is no problem, proceed to the next step.
(For details, refer to Section 13.4.1.4 Servo Adjustment Example.) button terminates the Servo Adjustment.

(7) Stop of Servo Control

Press the "Servo OFF" button to stop Servo Control.

Parameter Setting Guide・Notes

- If abnormal vibration or noise is heard, review the "Rotor Inertia Ratio".
  If the "Rotor Inertia Ratio" does not improve, reduce the "Position Control Frequency". However, set the "Speed Control Frequency" to 1.5 to 3.0 times the "Position Control Frequency.
- If the "Position Control Frequency" is low, the position tracking performance will decrease.
- If the "Speed Control Frequency" is too high, the motor will vibrate and make noise.
- For "Position control method," select "PID Control" for fast response or "I-PD Control" for disturbance suppression while suppressing overshoot.
- Since the value of the Speed Control Frequency to be adjusted depends on the operating speed, set the maximum Speed to the speed expected by the system.

13.5 Return to Origin

13.5.1 Function Description

13.5.1.1 What is Return to Origin?

It is function is used to determine the origin of the machine when the equipment is turned on.

If the machine is operated without Return to Origin, it will not be able to execute positioning correctly.
Because the origin of the machine has not been determined and the position is not controlled,
Since the motor rotates during Return to Origin adjustment, be careful that the moving parts do not collide with surrounding people or objects.

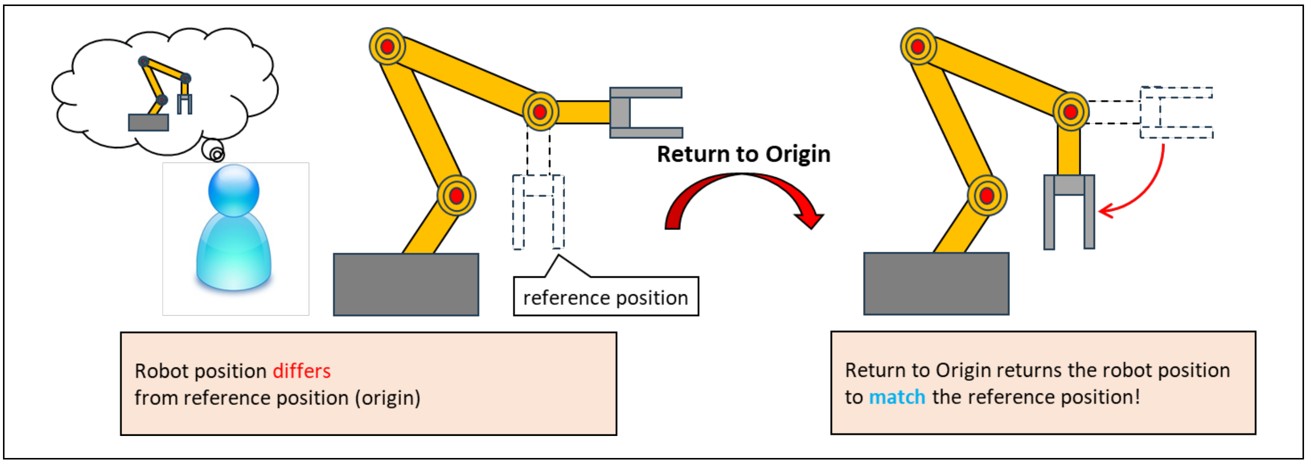


Figure 13-16 What is Return to Origin?

13.5.1.2 What is the Pushdown Origin Return Method?

Pushdown Origin Return method is a method in which the mechanism contacts a stopper or the like, rotates by a specified amount, and then originates at the position where it stops.
Note that if the mechanism contacts an interfering object during Return to Origin, the origin is set to the position rotated by the "Return to Origin amount" from that position.

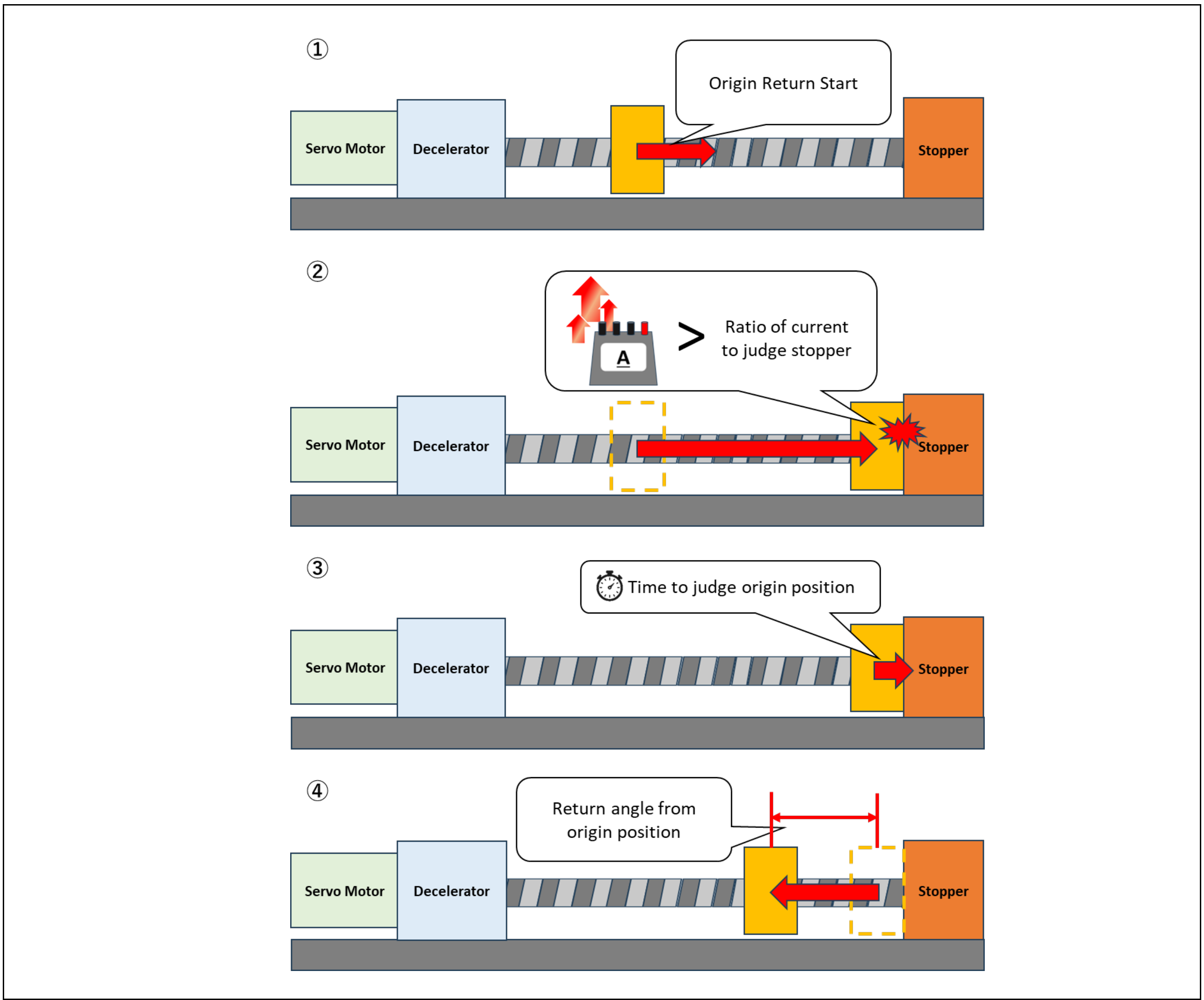


Figure 13-17 Pushdown Origin Return Method

The operation of Pushdown Origin Return method is shown below.

① Return to Origin operation is started.
② When the mechanism contacts the stopper, the torque (current) increases and reaches the "Return to Origin Operation Current".
③ When the time that the current exceeds the "Return to Origin Operation Current" exceeds the "Time to judge origin position" the return operation starts.
④ Reverse the motor and rotate it by the "Return to Origin amount." The origin is set at the position where the motor has rotated.

13.5.2 Window Structure・Parameter List

This section explains the name and function of each part in the tab [Return to Origin] window.
The tab [Return to Origin] allows you to set the return to origin method, return speed, etc.

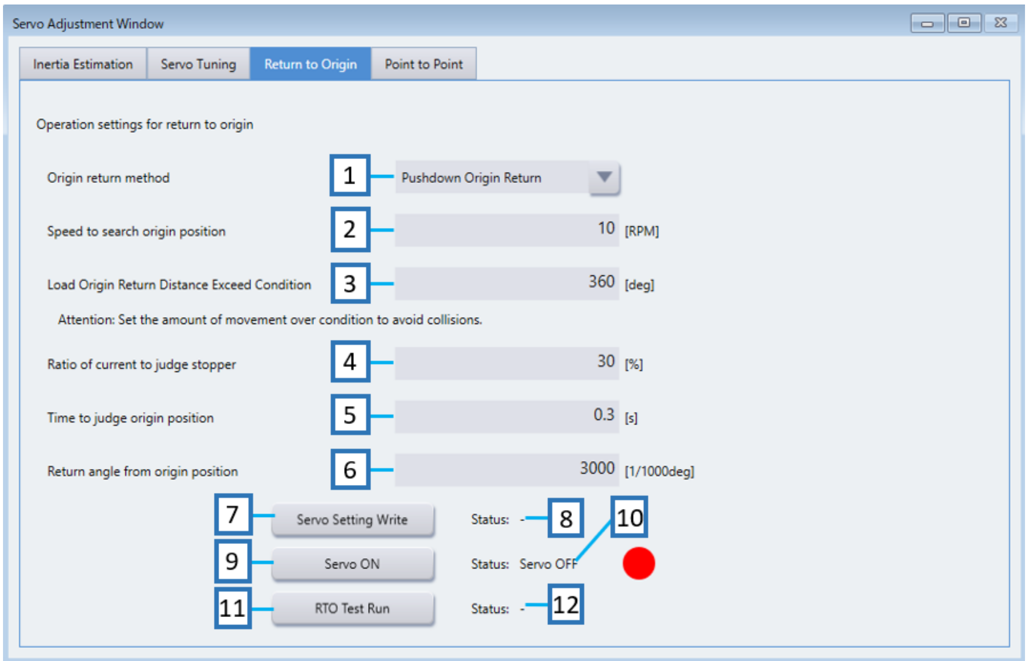


Figure 13-18 Return to Origin Tab Window

Table 13-3 Functions of Return to Origin Tab Window

No.	Name	Explanation
1	Origin return method	Specify the method for return to origin.
2	Load Origin Return Speed	Set the load speed for return to origin.
3	Load Origin Return Distance Exceed Condition	Set the maximum value of amount of load moved when Return to Origin.
4	Ratio of current to judge stopper	Set the Pushdown Origin Return operating current. Set the ratio when the rated current is 100%.
5	Time to judge origin position	Sets the time to push against the machine stopper
6	Return angle from origin position	Sets the position from the machine stopper to be the origin
7	Servo Setting Write button	Writes the settings from No.1 to 6.
8	Servo Setting Write Status	Displays the writing status after pressing the Servo Setting Write button (No.7)
9	Servo ON/OFF button	Switches ON/OFF of servo. Labels of the button and operations when clicked: Servo ON: Servo Control starts when clicked. Servo OFF: Servo control stops when clicked.
10	Servo Status	Displays the servo status. Green: servo is ON. Red: servo is OFF.
11	RTO Test Run/Stop button	When “Run” is clicked, the return to origin test starts. When “Stop” is clicked, the return to origin test stops. Labels of the button: During the return to origin test: “RTO Test Stop” While the return to origin test is stopped: “RTO Test Run”
12	RTO Test Status	Displays the status of the return to origin test execution.

13.5.3 Explanation of Operation

The Return to Origin procedure is shown below.

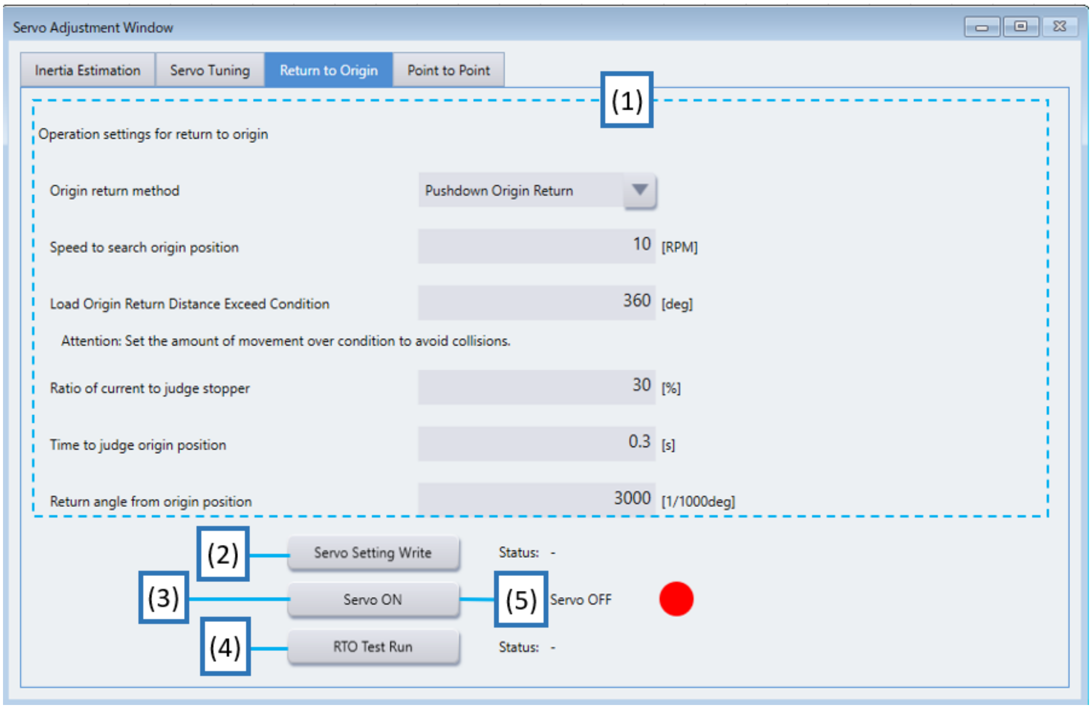


Figure 13-19 Return to Origin Operation Procedure

(1) Setting Parameters for Return to Origin

Set parameters for Return to Origin.
(For details, refer to Section 13.5.2 Window Structure・Parameter List.)

(2) Writing Return to Origin Parameters

When the "Servo Setting Write" button is pressed, the parameters set in (1) are written.

(3) Start of Servo Control

Before starting the Return to Origin test, click the "Servo ON" button to start Servo Control.

(4) Start/Stop of Return to Origin

Pressing the "RTO Test Run" button while the servo is "Servo ON" starts the Return to Origin test.
After the Return to Origin test is completed, the test will stop automatically. Pressing the "RTO Test Stop" button terminates the Return to Origin test.

(5) Stop of Servo Control

Clicking the "Servo OFF" button stops Servo Control.

Parameter Setting Guide・Notes

- The "Speed to search origin position" should be set to a low speed. If the speed is extremely high, the equipment may be damaged when it contacts the stopper.
- If the "Position Control Frequency" is low, the position tracking performance will decrease. The "Load Origin Return Distance Exceed Condition" should be set to a value large enough for movement. If the stopper is not found after moving within the set value range, the Return to Origin will stop at that point.

13.6 Point to Point Operation

13.6.1 Function Description

13.6.1.1 What is PTP Operation?

This is a revolution (Point To Point) operation from one point to another, and is performed by specifying the amount of movement, maximum speed, and acceleration time. It is used for simple positioning control.
It can operate at high speed because it automatically follows a path that is dependent on the robot's posture and does not require complex calculations.

Since the robot follows a free path, care must be taken to ensure that there are no obstacles in the path and that it does not interfere with other machines.

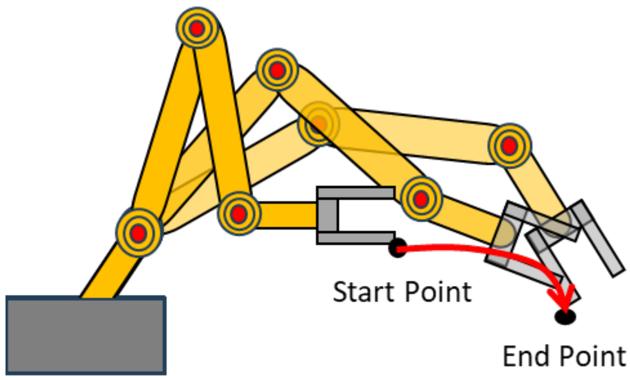


Figure 13-20 PTP Operation Example

13.6.2 Window Structure・Parameter List

This section describes the names and functions of the various parts of the tab [Point to Point] window.

The tab [Point to Point] can perform a Point to Point operation on a single axis.

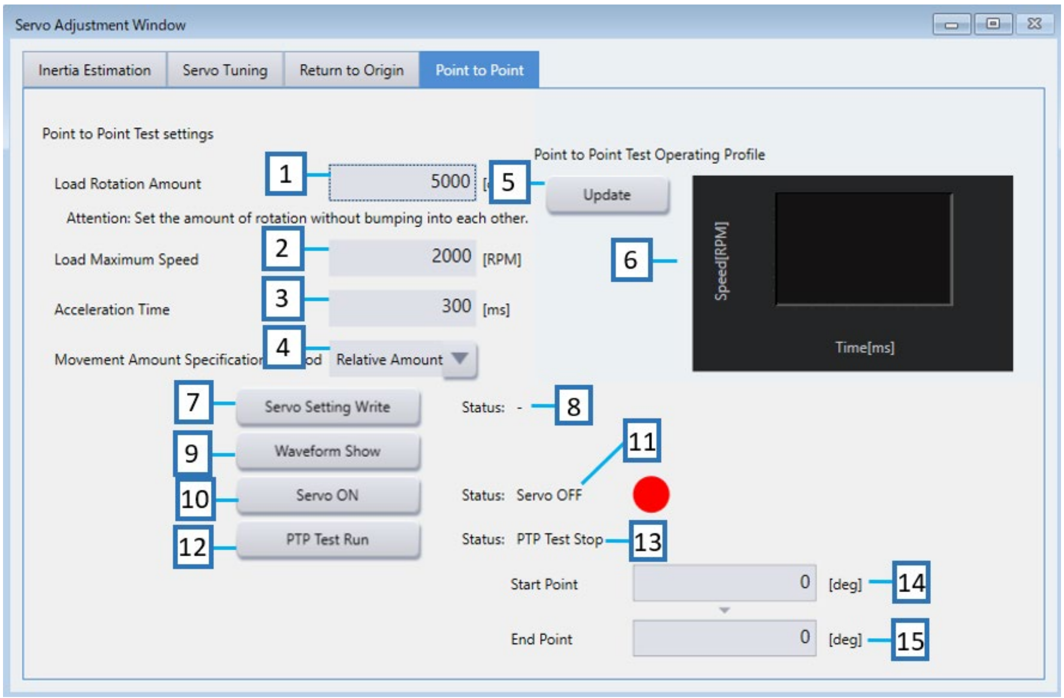


Figure 13-21 Point to Point Tab Window

Table 13-4 Functions of Point to Point Tab Window

No.	Name	Explanation
1	Load Rotation Amount	Set the load rotation amount. If "Relative Amount" is selected for No. 4, set the amount of rotation relative to the current point. If "Absolute Amount" is selected for No. 4, set the position of the target point with respect to the origin.
2	Load Maximum Speed	Set the load maximum speed.
3	Acceleration Time	Set the acceleration time. Set the time it takes to reach the maximum speed from the "STOP" state.
4	Movement Amount Specification Method	Set the method for specifying movement amount.
5	Update button	Updates the display of the operation profile.
6	Operation profile	Displays the operation profile of the settings.
7	Servo Setting Write button	Writes the settings from No.1 to 4.
8	Servo Setting Write Status	Displays the writing status after pressing the Servo Setting Write button (No.7).
9	Waveform Show button	Displays the waveform window.
10	Servo ON/OFF button	Switches ON/OFF of servo. Labels of the button and operations when clicked: Servo ON: Servo Control starts when clicked Servo OFF: Servo control stops when clicked.
11	Servo Status	Displays the servo status. Green: servo is ON. Red: servo is OFF.
12	PTP Test Run/Stop button	When “Run” is clicked, the Point to Point test starts. When “Stop” is clicked, the Point to Point test stops. Labels of the button: During the Point to Point test: “PTP Test Stop” While the Point to Point test is stopped: “PTP Test Run”
13	PTP Test Status	Displays the status of the Point to Point test execution.
14	Start Point	Displays the value of the start point.
15	End Point	Display the value of the end point.

The relationship between parameters and profiles is shown below.

① No.1(Load Rotation Amount)

② No.2(Load Maximum Speed)

③ No.3(Acceleration Time)

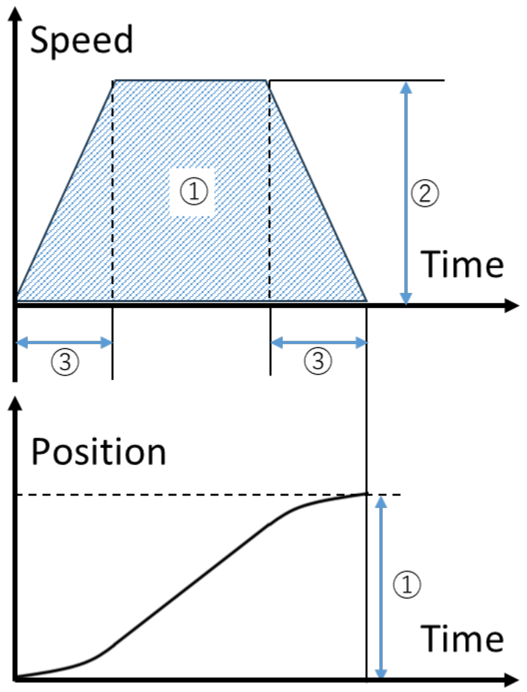


Figure 13-22 Relationship Between Parameters and Profiles

13.6.3 Explanation of Operation

The procedure for Point to Point operation is shown below.

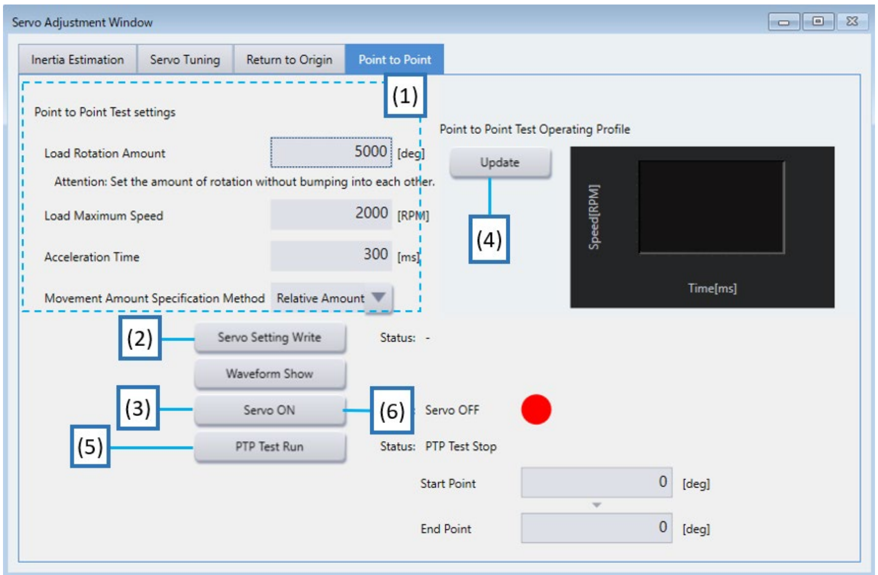


Figure 13-23 Point to Point Operation Procedure

(1) Setting Parameters for Point to Point Operation

Set parameters for Point to Point operation.
(For details, refer to Section 13.6.2 Window Structure・Parameter List.)

(2) Writing Point to Point Operation Parameters

When the "Servo Setting Write" button is pressed, the parameters set in (1) are written.

(3) Start of Servo Control

Before starting the Return to Origin test, click the "Servo ON" button to start Servo Control.

(4) Update of the Operating Profile

Click the "Update" button to update the operating profile.

(5) Start/Stop of Point to Point Operation

When the "PTP Test Run" button is pressed in the "Servo ON" state, Point to Point operation starts.
After the Point to Point operation is completed, it stops automatically. Clicking the "PTP Test Stop" button terminates the Point to Point operation.

(6) Stop of Servo Control

Clicking the "Servo OFF" button stops Servo Control.

Parameter Setting Guide・Notes

- Make sure that there are no obstacles in the path and that there is no interference with other machines.

13.7 Waveform Window

The Waveform window is activated by pressing the "Waveform Show" button on the Servo Tuning tab or the Point to Point tab of the Servo Adjustment window.
Switching between the Servo Tuning and Point to Point tabs displays the last acquired waveform; the waveform is cleared when RMW is closed.

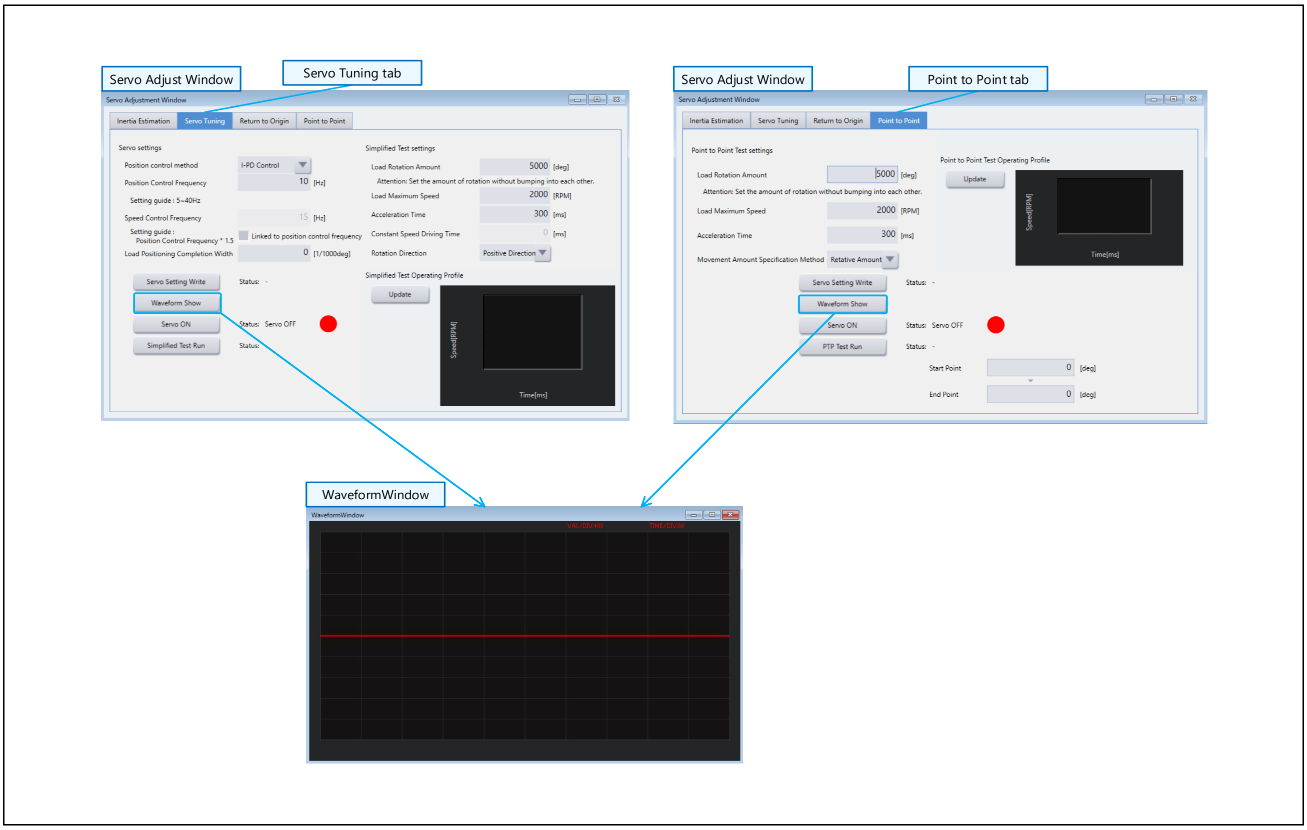


Figure 13-24 Waveform Window

14.Tuner Tool

14.1 Overview

Tuner tool automatically measures specific parameters (e.g resistance, inductance) of a permanent magnet synchronous motor and adjusts various control parameters (PI control gain etc.) required for encoder position control or sensorless speed control.

Tuner tool is available only when the tuning function is supported by the control program. “Tuner” will be displayed as selectable in the “Select Tool” on the Main Window when it is available.

Note : This tool is not guaranteed to work effectively for all motor.

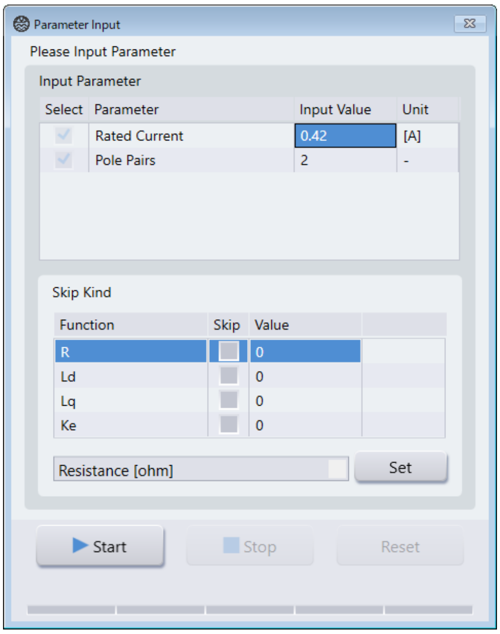
14.2 Features

- Provides two tuning modes: Manual and Easy.
- Tuning result (adjusted parameters) can be output as a PDF report or as a header file in the Renesas control program.

14.3 Window Structure

The window structure of Tuner tool is shown below.

(a) Parameter Input window



(b) Tune Window (Manual tab / Easy tab)

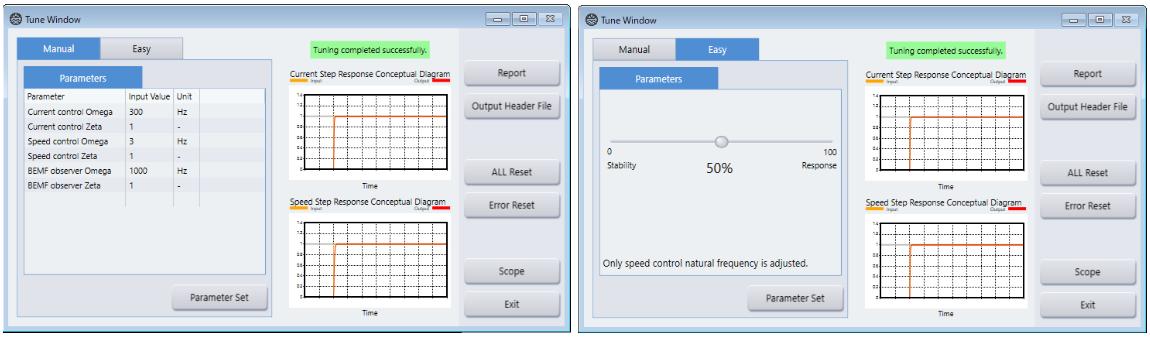


Figure 14-1 Tuner Tool Window

14.4 Explanation of Operation

This section describes the tuning procedure. For details on the functions on each window, see Section 14.5 Function Description.

14.4.1 Preparation

14.4.1.1 Write Execution File

(a) For execution files attached with Renesas Motor Workbench

Write the execution file for Tuner (mot format. hex format for RA MCUs) to the CPU board

- R***_***_SPM_ENCD_FOC_TUNER_***.mot (position control with encoder)
- R***_***_SPM_LESS_FOC_TUNER_***.mot (sensorless speed control)

(b) For sample program (project format) provided by Renesas

Write the built execution file for Tuner to the CPU board.

When including Tuner function to the sample program, refer to Section 14.6 Procedure to Include Tuner Library.

14.4.1.2 Start Renesas Motor Workbench

Click the short-cut icon on the desktop to start-up Renesas Motor Workbench.

14.4.1.3 Load the RMT File

Load the RMT file for Tuner to Renesas Motor Workbench.

- R***_***_SPM_ENCD_FOC_TUNER_***.rmt (position control with encoder)
- R***_***_SPM_LESS_FOC_TUNER_***.rmt (sensorless speed control)

14.4.1.4 Establish Communication

Establish communication between Renesas Motor Workbench and the evaluation board.

14.4.1.5 Start Tuner Tool

Select Tuner tool from Select Tool of Renesas Motor Workbench and start it.

14.4.2 Performing Tuning

14.4.2.1 Input Parameters for Tuning

When you start Tuner tool, the "Parameter Input" window will be displayed. In the Input Parameter section above, enter the rated current and pole pairs of the motor referring to the specifications of the motor.

If you know the properties of the motor in advance and want to skip some tuning, specify those parameters in the Skip Kind field at the bottom (for details, refer to Section 14.5.1 Skip Function (Parameter Input window)).

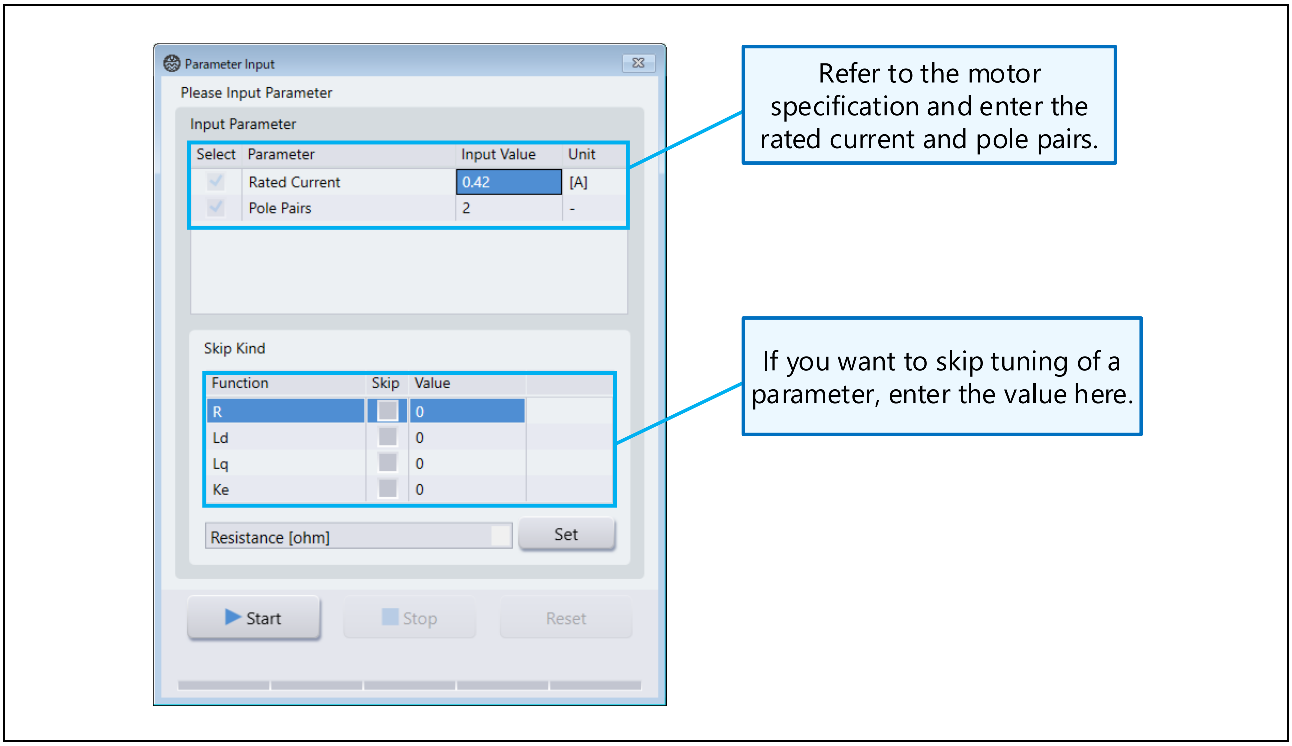


Figure 14-2 Parameter Input Window

14.4.2.2 Perform Tuning

Click the Start button to start tuning, and the progress bar will be displayed while tuning. To stop tuning, click the “Stop” button. When tuning is completed, “Tune Window” will be displayed.

Note : The motor rotates during tuning.

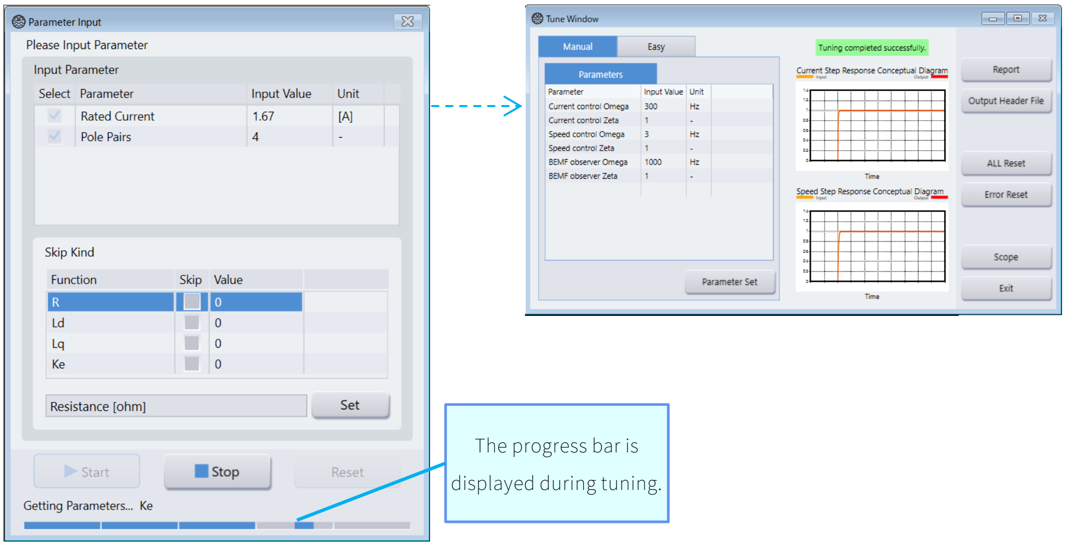


Figure 14-3 Tune Window Display Under Tuning and After Tuning

If an error occurs during tuning, check the error message and click the "Reset" button.

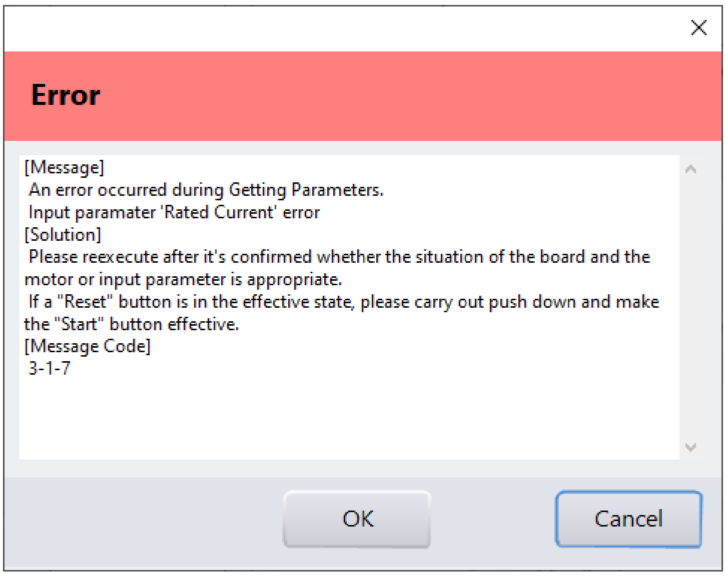


Figure 14-4 Error Message Example

14.4.2.3 Confirm Tuning Result

Click the “Report” button on the “Tune Window” and confirm details about tuning on the “Result Report” screen.

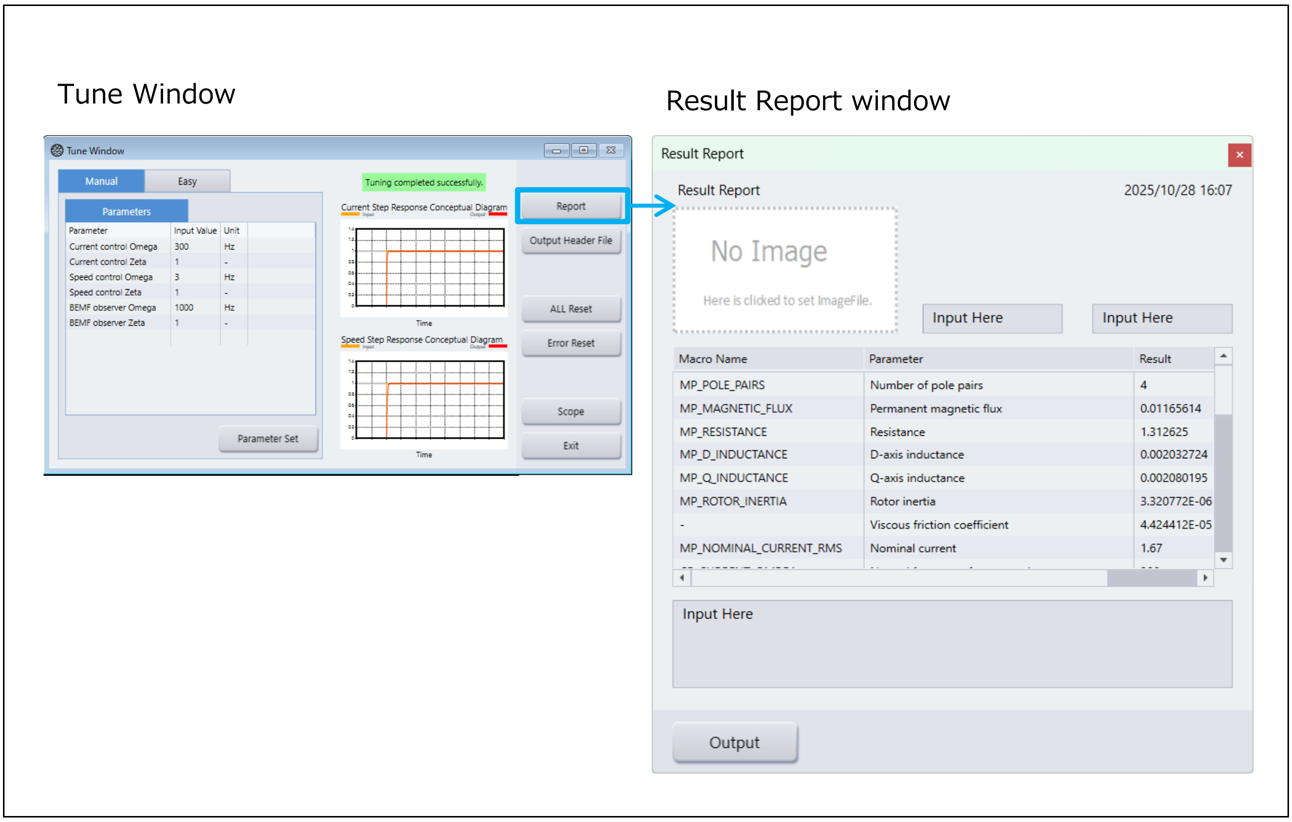


Figure 14-5 Result Report View

14.4.2.4 Terminate Tuning

Click the "Exit" button on "Tune Window" to return to the Parameter Input window.

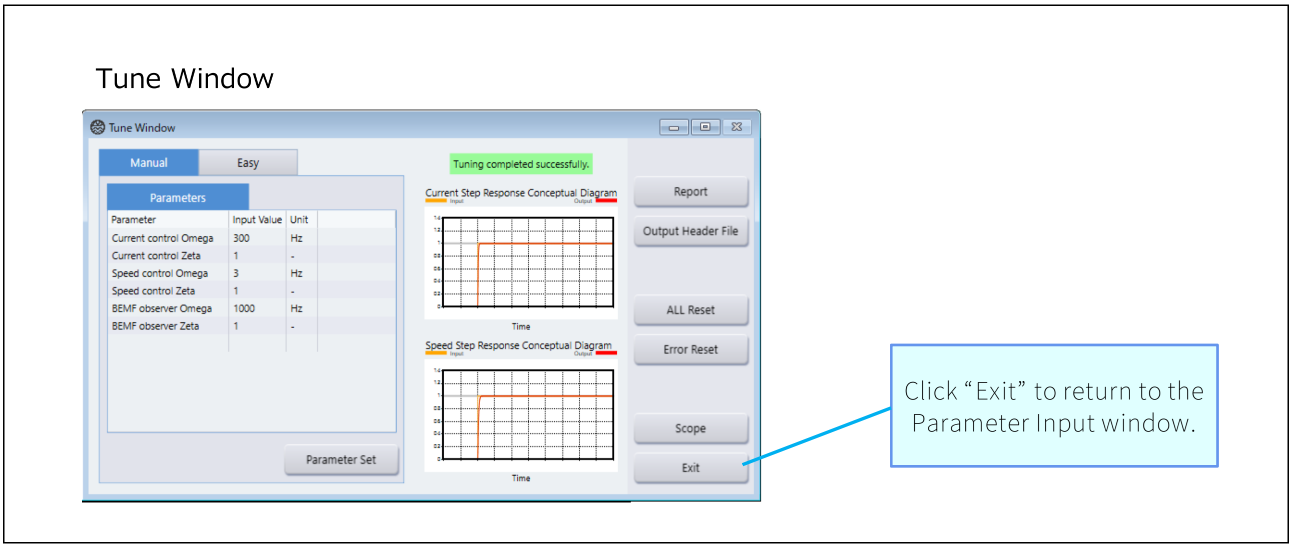


Figure 14-6 Exit Button

14.4.3 Checking Operation

14.4.3.1 [Only for Encoder Program] Input of Encoder Pulse Number per Rotation.

Input of encoder pulse number per rotation on “Tune Window” and click the “Parameter SET” button.

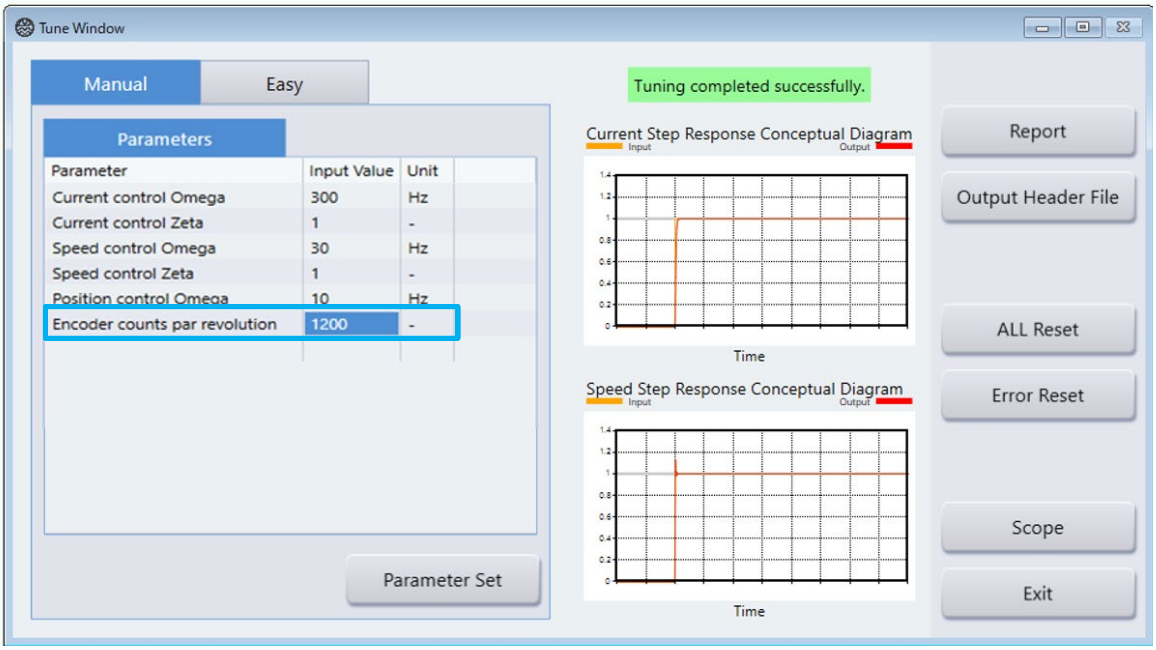


Figure 14-7 Input of Encoder Pulse Number (Encoder Program Only)

14.4.3.2 Test Drive of Motor

Drive the motor with "Drive Test" of Scope window.

Click the "Scope" button on Tune Window, Scope Window will be displayed. Select "Drive Test" in Control's Operation and click the "RUN" button to drive the motor in speed control mode.

Figure 14-8 Drive Test View

14.4.3.3 Other Driving Test

In the Control’s Operation section, you can select Position Control for encoder-position control and Starting for sensorless-speed control.

In Position Control, you can check the operation in the position control mode.

In “Starting, you can check the operation when starting sensorless vector control.

Note In Section 14.4.3.2 and 14.4.3.3, the waveform is displayed automatically. However, the waveform may not be fully displayed depending on the connected motor.

Figure 14-9 Position Control / Starting

14.4.4 Terminating Tuner Tool

If you are testing the motor drive, press the “STOP” button to finish the test. Click the "Exit" button on Tune Window to return to the Parameter Input window. After that, return to Main Window by clicking the “Main Window” button, or operate the tool switch button.

14.5 Function Description

14.5.1 Skip Function (Parameter Input Window)

You can skip measurement of some parameters by entering those tuning parameters such as the resistive, d-axis/q-axis inductance, and magnetic flux before tuning.

In the “Skip Kind” area of the Parameter Input window, select the Skip checkbox of the parameter to be omitted and enter the parameter. When you have finished entry, press the “SET” button.

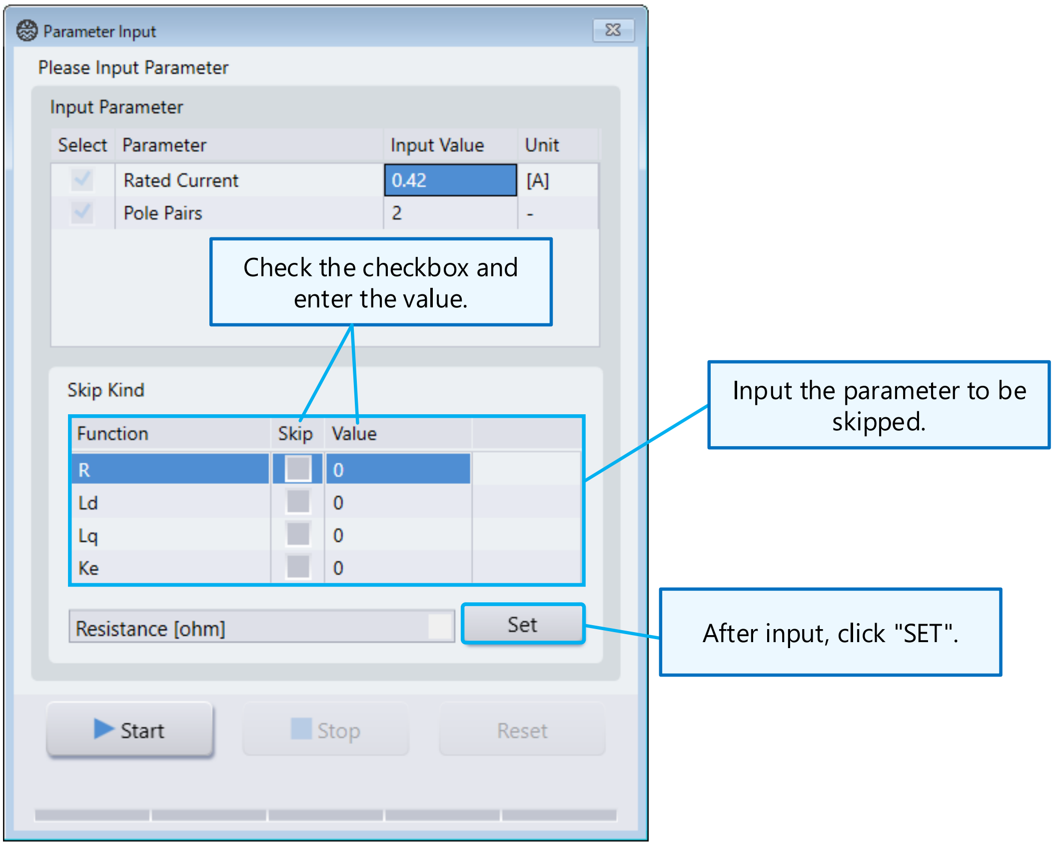


Figure 14-10 Skip Input View

14.5.2 Tuning Control Parameters (Tune Window)

"Tune Window" is a window for control parameter setting. There are two ways to set parameters. You can switch tabs between Manual tab and Easy tab.

On the Manual tab, you can adjust the natural frequency of each feedback loop and damping coefficient to be used and reflect the adjustment to the control.

On the Easy tab, you can adjust parameters that are set to 50% as default by moving the slider from side to side and reflect the adjustment to the control.

To reflect the adjustment to the control, click the "Parameter SET" button.

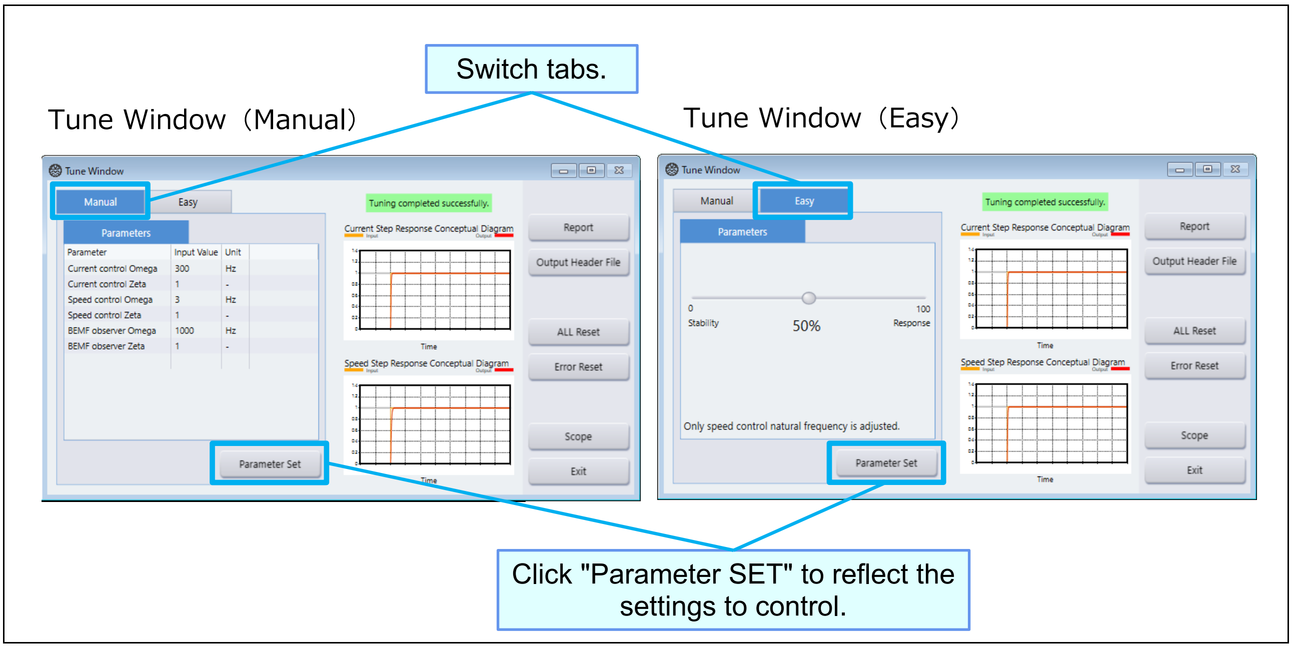


Figure 14-11 Manual And Easy Tab

Table 14-1 Tuning Parameter List (Manual Tab)

Display	Adjustable parameter	Remark
Current control Omega	Natural frequency related to current control [Hz]	
Current control Zeta	Damping coefficient related to current control	
Speed control Omega	Natural frequency related to speed control [Hz]	
Speed control Zeta	Damping coefficient related to speed control	
BEMF observer Omega	Natural frequency related induced-voltage estimation [Hz]	Sensorless speed control only
BEMF observer Zeta	Damping coefficient related induced-voltage estimation	Sensorless speed control only
Position control Omega	Natural frequency related to position control [Hz]	Encoder position control only
Encoder counts per revolution	Encoder pulse counts per revolution	Encoder position control only

14.5.2.1 Step Response Conceptual Diagram

Based on the control parameter values set in the Manual tab and Easy tab, a conceptual diagram of the step response is displayed.
Changing the values of Current control Omega and Speed control Zeta will alter the step response image in the Current Step Response Conceptual Diagram.
Changing the values of Speed control Omega and Speed control Zeta will alter the step response image in the Speed Step Response Conceptual Diagram.
When configuring settings in the Easy tab, only the Speed Step Response Conceptual Diagram will change.

Note : This does not represent the actual response characteristics of the motor you are using.

Figure 14-12 Conceptual Diagram of Step Response Changes Due to Control Parameter Modifications

14.5.3 Reset Function (Tune Window)

If an error occurs in the drive program after tuning, select the type of reset to perform. Click the "ALL Reset" or "Error Reset" button in the Tune Window to display the reset screen.

- ALL Reset

  Resets all adjustment parameters. Parameter Input window will be displayed, so perform tuning again.

- Error Reset

  Resets only the error state and the tuned adjustment parameters are retained.

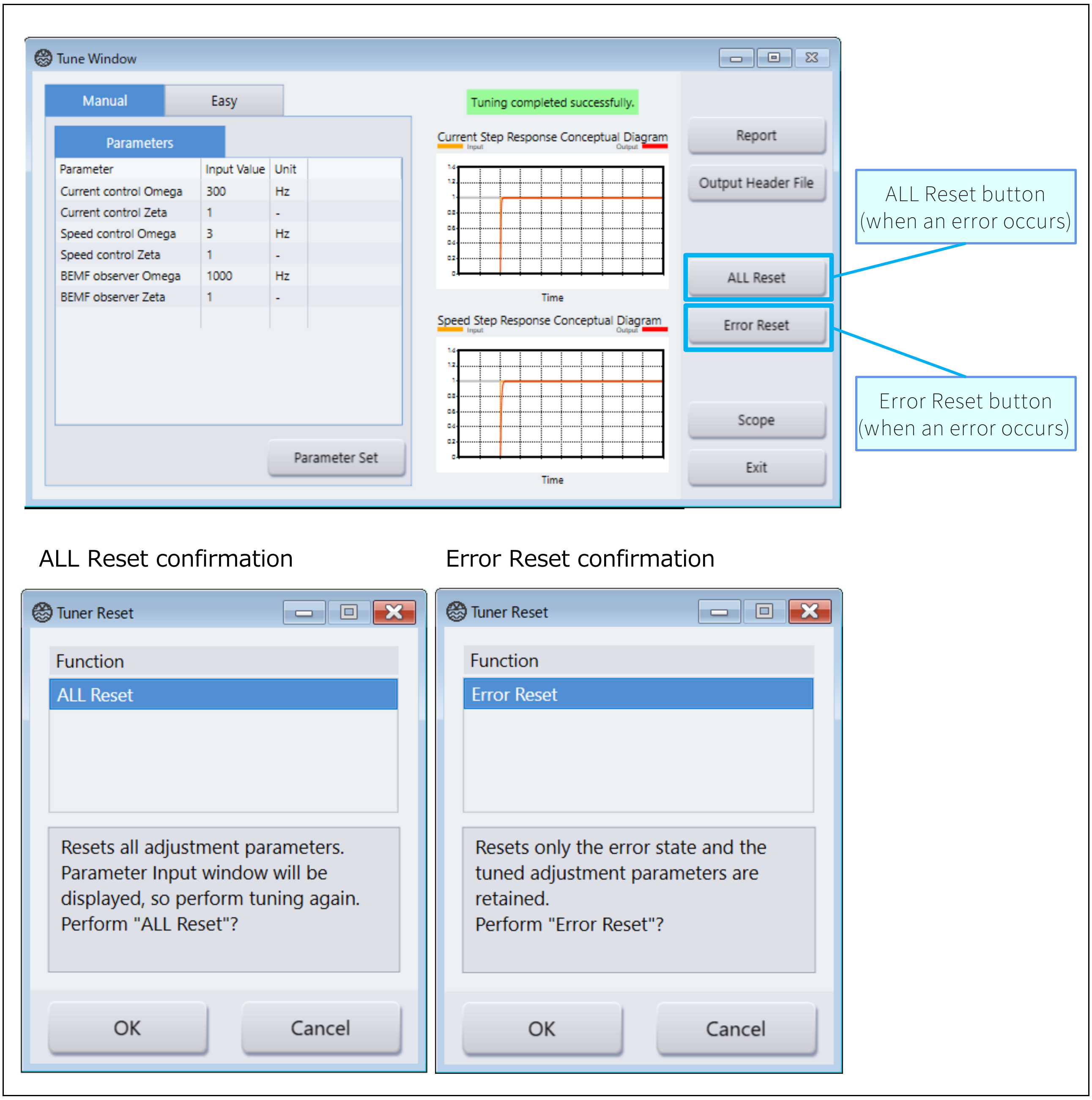


Figure 14-13 Reset Function

14.5.4 Header Output (Tune Window)

The tuning result can be output in the header file format of the Renesas motor control program.

Click the "Output Header File" button on Tune Window, and the saving window for "r_mtr_control_parameter.h" and "r_mtr_motor_parameter.h" will be displayed.

Figure 14-14 Output Header File Button

For the encoder position control program, refer to the application note and specify the encoder pulse count separately when including into the sample program.

14.5.5 PDF Output (Tune Window)

Click the “Report” button on “Tune Window” to display the Result Report window. You can output the tuning result in a PDF file with the “Output” button on this window.

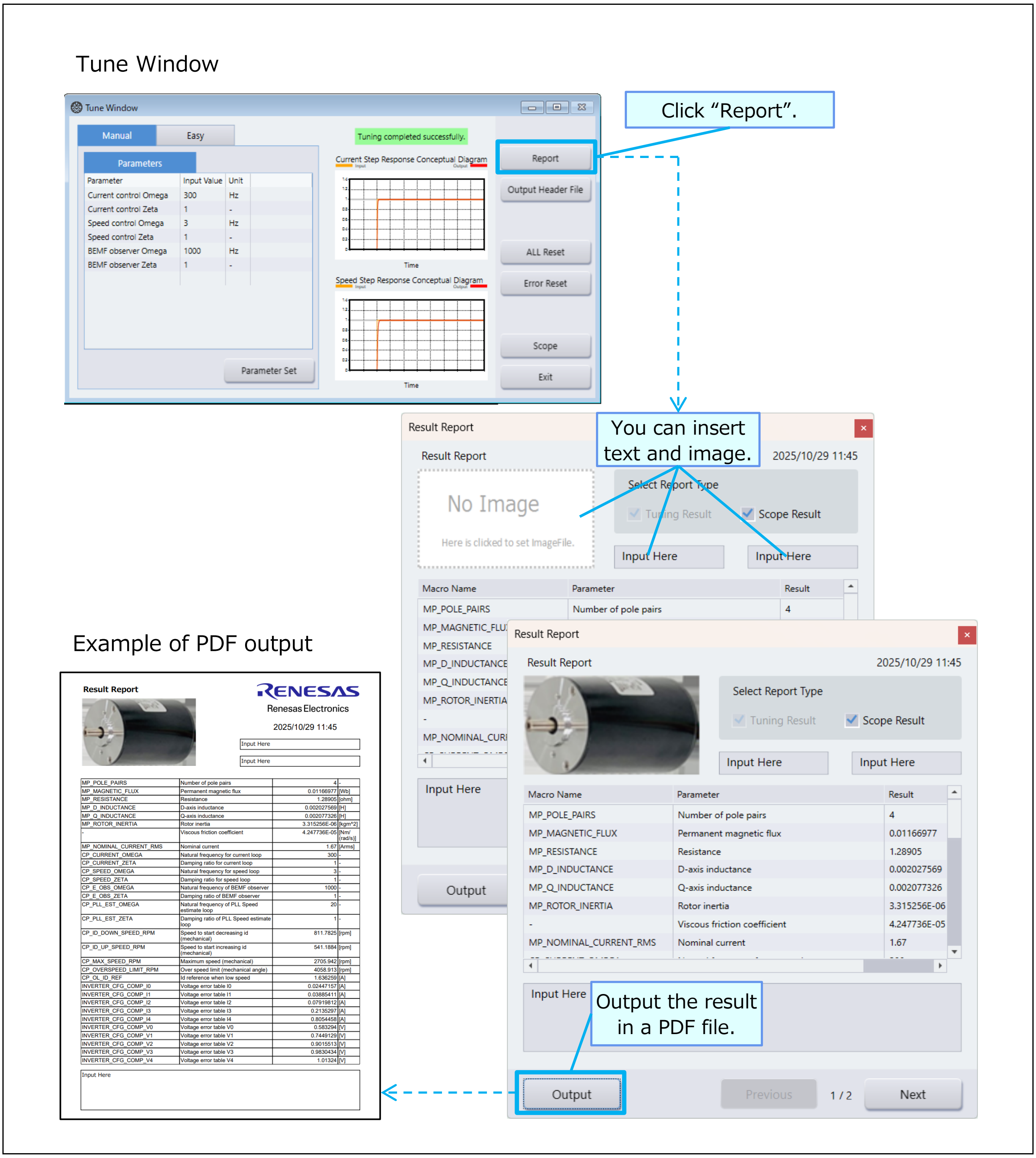


Figure 14-15 Output Button (PDF Output)

The parameter list output as tuning results is shown below.

Table 14-2 Define Directives

Definition name	Description	Unit
MP_POLE_PAIRS	Pole pairs	-
MP_MAGNETIC_FLUX	Magnetic flux	[wb]
MP_RESISTANCE	Resistance	[ohm]
MP_D_INDUCTANCE	Inductance on d-axis	[H]
MP_Q_INDUCTANCE	Inductance on q-axis	[H]
MP_ROTOR_INERTIA	Inertia of rotor	[kg m2]
MP_NOMINAL_CURRENT_RMS	Rated current	[A]
-	Viscous friction coefficient	[Nm/(rad/s)]
CP_CURRENT_OMEGA	Current control system natural frequency	[Hz]
CP_CURRENT_ZETA	Current control system damping factor	-
CP_SPEED_OMEGA	Speed control system natural frequency	[Hz]
CP_SPEED_ZETA	Speed control system damping coefficient	-
CP_E_OBS_OMEGA	Induced voltage estimation system natural frequency (only for sensorless program)	[Hz]
CP_E_OBS_ZETA	Induced voltage estimation system damping coefficient (only for sensorless program)	-
CP_PLL_EST_OMEGA	Position estimation system natural frequency (only for sensorless program)	[Hz]
CP_PLL_EST_ZETA	Position estimation system damping coefficient (only for sensorless program)	-
CP_ID_DOWN_SPEED_RPM	d-axis current command value subtraction start speed (machine angle) (only for sensorless program)	[rpm]
CP_ID_UP_SPEED_RPM	d-axis current command value addition start speed (machine angle) (only for sensorless program)	[rpm]
CP_POS_OMEGA	Position control system natural frequency (only for encoder program)	[Hz]
CP_SOB_OMEGA	Natural frequency of speed observer (only for encoder program)	[Hz]
CP_SOB_ZETA	Attenuation coefficient of speed observer (only for encoder program)	-
CP_MIN_SPEED_RPM	Minimum speed (only for encoder program)	[rpm]
CP_MAX_SPEED_RPM	Maximum speed	[rpm]
CP_OVERSPEED_LIMIT_RPM	Speed limit value	[rpm]
CP_OL_ID_REF	d-axis current command value at open loop	[A]
INVERTER_CFG_COMP_I0	Coefficient for compensation of the voltage error	[A]
INVERTER_CFG_COMP_I1	Coefficient for compensation of the voltage error	[A]
INVERTER_CFG_COMP_I2	Coefficient for compensation of the voltage error	[A]
INVERTER_CFG_COMP_I3	Coefficient for compensation of the voltage error	[A]
INVERTER_CFG_COMP_I4	Coefficient for compensation of the voltage error	[A]
INVERTER_CFG_COMP_V0	Coefficient for compensation of the voltage error	[V]
INVERTER_CFG_COMP_V1	Coefficient for compensation of the voltage error	[V]

INVERTER_CFG_COMP_V2	Coefficient for compensation of the voltage error	[V]
INVERTER_CFG_COMP_V3	Coefficient for compensation of the voltage error	[V]
INVERTER_CFG_COMP_V4	Coefficient for compensation of the voltage error	[V]

14.5.6 Waveform Information Display (Scope Window)

When you click the “View Open” button on Scope Window, you can check the information about the waveform display.

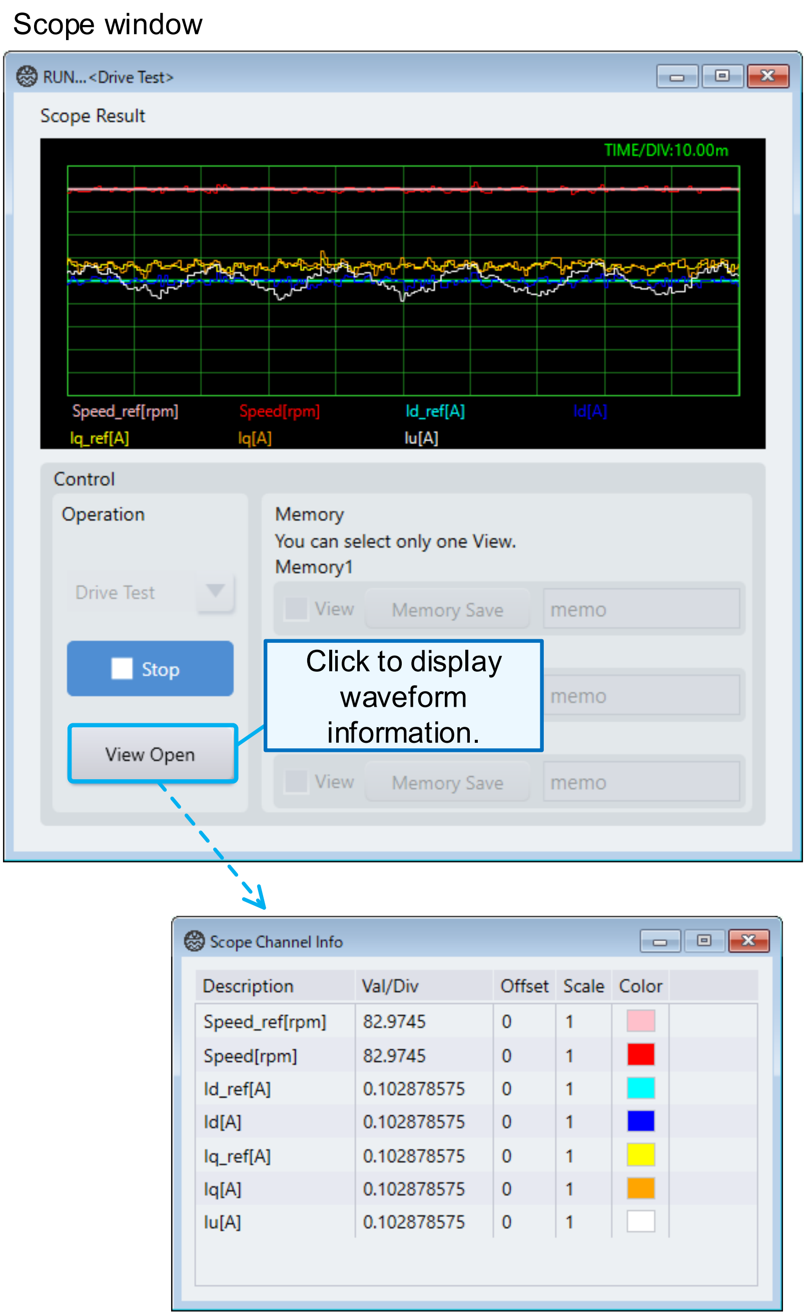


Figure 14-16 View Open Button

14.5.7 Memory Function (Scope Window)

In Scope Window, up to three waveforms of motor driving can be memorized, and individual waveforms can be switched and displayed.

When you click the "Memory Save" button, the waveform is memorized and the label of the button changes to "Memory Clear". Click it again to clear the memory.

When you select the View checkbox, the memorized waveform is displayed.

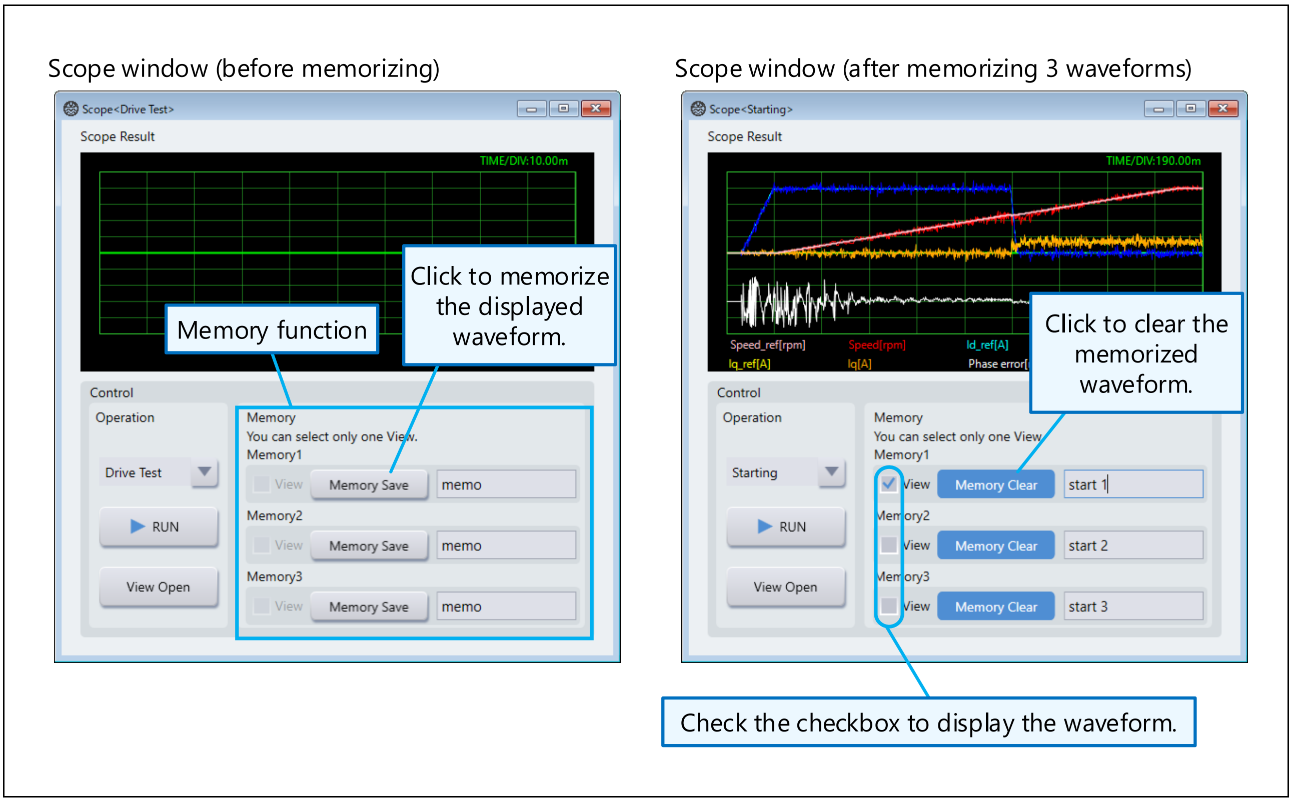


Figure 14-17 Memory Function

14.6 Procedure to Include Tuner Library

14.6.1 Tuner Library Structure

Tuner Library has main four functions: “resistance measurement function”, “magnetic flux measurement function”, “inductance measurement function”, and “inertia and friction measurement function”. It is necessary to configure user- implemented functions so that Tuner Library can access peripheral functions. Tuner Library accesses the peripheral functions from the function pointers of the configured user-implemented functions. For configuration of user-implemented functions, refer to the Section 14.6.4.8.

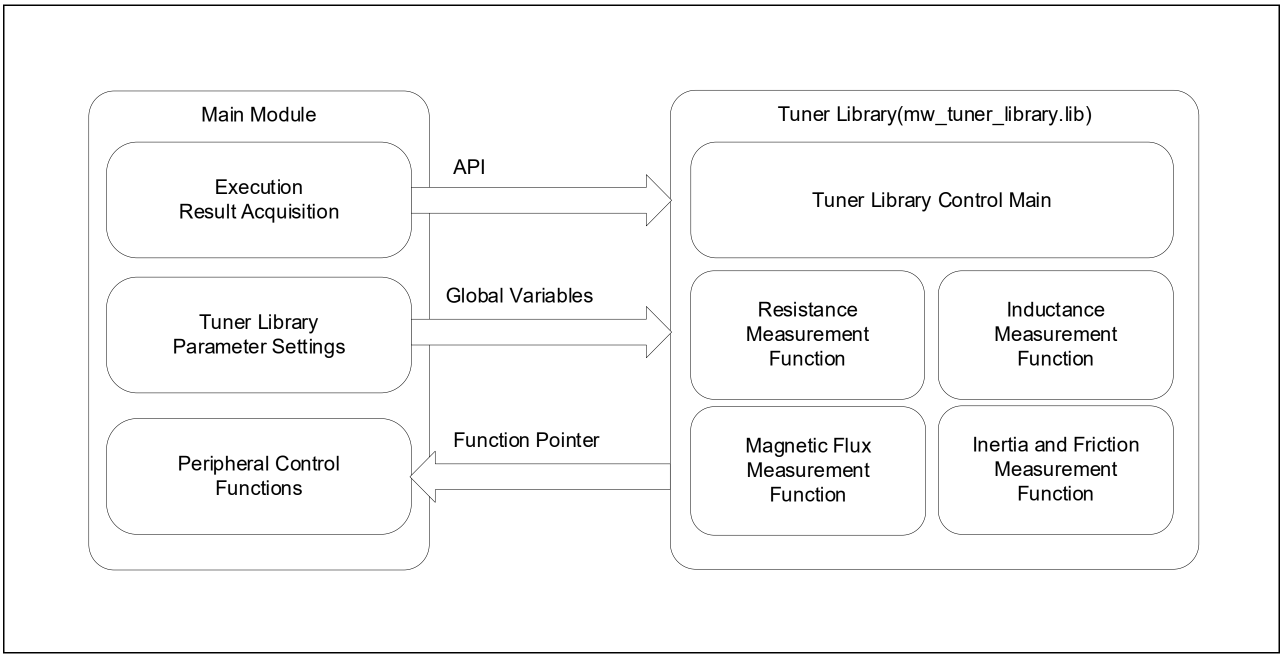


Figure 14-18 Tuner Library Structure

14.6.2 Tuner Library API List

Table 14-3 Tuner Library API [1/3]

Function	Type	Function name	Definition file	Description
Initialize Tuner process	void	R_AID_Init(uint8_t pwm_tick_per_irq, float speed_ctrl_period)	r_aid_auto_identify.h	Calls each time before starting Tuner
Start Tuner process	void	R_AID_CmdStart(void)	r_aid_auto_identify.h	Calls to start Tuner
Stop Tuner process	void	R_AID_CmdStop(void)	r_aid_auto_identify.h	Calls to stop Tuner during the process
Reset Tuner process Initialize required variables	void	R_AID_CmdReset(void)	r_aid_auto_identify.h	Calls to release error status when error occurs
Resume Tuner process Enabled only when the last process is suspended	void	R_AID_CmdResume(void)	r_aid_auto_identify.h	Calls to resume Tuner process if the last process is suspefinded
Set and execute internal state machine event	void	R_AID_CmdByCode(uint16_t cmd_code)	r_aid_auto_identify.h	Calls to set state machine event with argument and execute AID_COMMAND_NONE (0) status that no command is issued AID_COMMAND_START (1) command code that starts tuning AID_COMMAND_STOP (2) command code that stops tuning AID_COMMAND_RESET (3) reset command code AID_COMMAND_RESUME (4) resume command code

Table 14-4 Tuner Library API [2/3]

Function	Type	Function name	Definition file	Description
Call when error occurs Set upper error to Tuner process	void	R_AID_UserError(uint16_t u2_error_code)	r_aid_auto_identify.h	Calls to force the status into error status Arbitrary error codes can be set as argument. TunerLib defines the following four codes: AID_ERROR_INPUT_CURRENT (0x1001) AID_ERROR_INPUT_POLEPAIR (0x1002) AID_ERROR_INPUT_VOLTERR_STEP (0x1003) AID_ERROR_INPUT_INERTIA_RANGE (0x1004)
Set reference current value and number of motor pole pairs	void	R_AID_ConfigMotorPlate (float f4_rated_current, uint16_t u2_num_of_pole_pair)	r_aid_auto_identify.h	Sets Tuner input values of rated current and number of pole pairs as argument and calls Calls before starting Tuner
Set motor electrical characteristics	int32_t	R_AID_SetInitElecParams(float f4_r, float f4_ld, float f4_lq, float f4_ke)	r_aid_auto_identify.h	If any Tuner parameter is already known, and Tuning of the parameter is to be skipped, then sets the value as argument and calls before starting Tuner
Acquire version of Tuner process	void	R_AID_GetVersionInfo(uint16_t *p_major_version, uint16_t *p_minor_version)	r_aid_auto_identify.h	Calls to acquire version information
Acquire current control period [sec]	float	R_AID_GetCurrentCtrlPeriod(void)	r_aid_auto_identify.h	Calls to acquire configured current control period
Acquire speed control period [sec]	float	R_AID_GetSpeedCtrlPeriod(void)	r_aid_auto_identify.h	Calls to acquire configured speed control period
Acquire PWM carrier cycle [sec]	float	R_AID_GetPWMPeriod(void)	r_aid_auto_identify.h	Calls to acquire configured PWM carrier cycle
Acquire Tuner process internal status	uint16_t	R_AID_GetSystemStatus (void)	r_aid_auto_identify.h	Calls to acquire status information AID_STATUS_READY (0) READY status code AID_STATUS_MEASURE (1) MEASURE status code AID_STATUS_ERROR (2) ERROR status code AID_STATUS_RESET (3) RESET status code AID_STATUS_COMPLETED (4) COMPLETED status code

Table 14-5 Tuner Library API [3/3]

Function	Type	Function name	Definition file	Description
Acquire error information	uint16_t	R_AID_GetErrorStatus(void)	r_aid_auto_identify.h	Calls to acquire error information. The value to acquire is value set with R_AID_UserError or error code of TunerLib AID_ERROR_INPUT_CURRENT (0x1001) AID_ERROR_INPUT_POLEPAIR (0x1002) AID_ERROR_INPUT_VOLTERR_STEP (0x1003) AID_ERROR_INPUT_INERTIA_RANGE (0x1004)
Acquire Tuner process progress	float	R_AID_GetProgress(void)	r_aid_auto_identify.h	Calls to acquire progress of Tuner process
Acquire motor resistance value [Ω]	float	R_AID_GetResistance(void)	r_aid_auto_identify.h	Calls to acquire resistance value after Tuning
Acquire d-axis inductance [H]	float	R_AID_GetLd(void)	r_aid_auto_identify.h	Calls to acquire d-axis inductance value after Tuning
Acquire q-axis inductance [H]	float	R_AID_GetLq(void)	r_aid_auto_identify.h	Calls to acquire q-axis inductance value after Tuning
Acquire magnetic flux density [Wb]	float	R_AID_GetKe(void)	r_aid_auto_identify.h	Calls to acquire magnetic flux density after Tuning
Acquire intertia [kgm^2]	float	R_AID_GetInertia(void)	r_aid_auto_identify.h	Calls to acquire inertia value after Tuning
Acquire friction coefficient [Nm/(rad/sec)]	float	R_AID_GetFriction(void)	r_aid_auto_identify.h	Calls to acquire friction coefficient value after Tuning
Acquire Tuner settings	void	R_AID_GetIDSetting(st_aid_id_setting_t *st_id_setting)	r_aid_auto_identify.h	Calls to acquire setting values
For calls at current control period interrupt	void	R_AID_CurrentCtrlISR(void)	r_aid_auto_identify.h	Calls at current control period interrupt Executes TunerLib current control process
For calls at speed control period interrupt	void	R_AID_SpeedCtrlISR(void)	r_aid_auto_identify.h	Calls at speed current control period interrupt Executes TunerLib speed control process

Table 14-6 Tuner Library Variables

Type	Variable name	Definition file	Initial value	Description
VOID_FUNC	g_fp_aid_internal_clear_oc_flag	r_aid_config.h	Function pointer of user-implemented function to access peripheral function from Tuner Library	Overcurrent state release process
MTR_ID_FUNC	g_fp_aid_internal_ctrl_start			PWM output start process
MTR_ID_FUNC	g_fp_aid_internal_ctrl_stop			PWM output stop process
MTR_GET_VDC_FUNC	g_fp_aid_internal_get_vdc			Bus voltage value acquisition process
MTR_GET_CURRENT_IUIW_FUNC	g_fp_aid_internal_get_current_iuiw			U-/W-phase current value acquisition process
MTR_INV_SET_UV_FUNC	g_fp_aid_internal_inv_set_uvw			PWM duty setting process

14.6.3 Tuner Library Macro List

Table 14-7 Define Directives

Definition name	Value	File name	Description
AID_API_MAJOR_VERSION	(1)	r_aid_auto_identify.h	Defines API major version
AID_API_MINOR_VERSION	(1)		Defines API minor version
AID_VOLTERR_TABLE_SIZE	(5)		Table data size
AID_STATUS_READY	(0)		Defines REDY status code
AID_STATUS_MEASURE	(1)		Defines MEASURE status code
AID_STATUS_ERROR	(2)		Defines ERROR status code
AID_STATUS_RESET	(3)		Defines RESET status code
AID_STATUS_COMPLETED	(4)		Defines COMLETED status code
AID_PARAMODE_INIT	(0)		Defines initial identification mode code
AID_PARAMODE_R_DIFF	(1)		Defines RDIFF identification mode code
AID_PARAMODE_RLD_RLS	(2)		Defines code for RLS Ld identification mode
AID_PARAMODE_RLD_DFT	(3)		Defines code for DFT Ld identification mode
AID_PARAMODE_LQ_RLS	(4)		Defines code for RLS Lq identification mode
AID_PARAMODE_LQ_DFT	(5)		Defines code for DFT Lq identification mode
AID_PARAMODE_KE	(6)		Defines rated magnetic flux identification mode code
AID_PARAMODE_JD	(7)		Defines inertia identification mode code
AID_PARAMODE_END	(8)		Defines termination identification mode code
AID_PARAMODE_VOLTERR	(9)		Defines voltage error measurement mode code
AID_ERROR_INPUT_CURRENT	(0x1001)		(4097) Invalid input value of rated current
AID_ERROR_INPUT_POLEPAIR	(0x1002)		(4098) Invalid input number of polar pairs
AID_ERROR_INPUT_VOLTERR_STEP	(0x1003)		(4099) Invalid input value of voltage error current step
AID_ERROR_INPUT_INERTIA_RANGE	(0x1004)		(4100) Invalid input value of inertia range
AID_COMMAND_NONE	(0U)		Defines status that no command is issued
AID_COMMAND_START	(1U)		Defines command code to start Tuning
AID_COMMAND_STOP	(2U)		Defines command code to stop Tuning
AID_COMMAND_RESET	(3U)		Defines command code to reset
AID_COMMAND_RESUME	(4U)		Defines command code to resume
AID_FAULT_PARAM_R	(-1)		Defines fault code for invalid parameter R
AID_FAULT_PARAM_LD	(-2)		Defines fault code for invalid parameter Ld
AID_FAULT_PARAM_LQ	(-3)		Defines fault code for invalid parameter Lq
AID_FAULT_PARAM_KE	(-4)		Defines fault code for invalid parameter Ke
AID_API_MAJOR_VERSION	(1)		Defines API major version

Table 14-8 Structures

Type	Definition name	File name	Description
struct	typedef struct { float f4_rated_current; uint16_t u2_num_pole_pairs ; uint8_t u1_volterr_is_enabled; uint16_t u2_volterr_crnt_step_lsb; float f4_inertia_range; float f4_assumed_inertia; } st_aid_id_setting_t;	r_aid_auto_identify.h	Rated current [A] Number of pole pairs Voltage error measurement enabled or disabled : 1=Enabled, 0=Disabled Current step of voltage error measurement Inertia range: 0 (no load) to 1 (heavy inertia) Inertia to be used for gain design of speed controller

Table 14-9 Function Pointers

Type	Definition name	File name	Description	Remarks
void	(*VOID_FUNC)(void)	r_aid_config.h	Function pointer	typedef void (*VOID_FUNC)(void)
void	(*MTR_ID_FUNC)(void)		Function pointer	typedef void (*MTR_ID_FUNC)(void)
float	(*MTR_GET_VDC_FUNC)(void)		Function pointer	typedef float (*MTR_GET_VDC_FUNC)(void)
void	(*MTR_GET_CURRENT_IUIW_FUNC)(float *f4_iu_ad, float *f4_iw_ad)		Function pointer	typedef void (*MTR_GET_CURRENT_IUIW_FUNC)(float *f4_iu_ad, float *f4_iw_ad)
void	(*MTR_INV_SET_UV_FUNC)(float f4_duty_u, float f4_duty_v, float f4_duty_w)		Function pointer	typedef void (*MTR_INV_SET_UV_FUNC)(float f4_duty_u, float f4_duty_v, float f4_duty_w)

14.6.4 Procedure to Include Tuner Library

As an example, this section explains the procedure to include Tuner Library into the sample code RX26T_MCBA_MCILV1_SPM_LESS_FOC_E2S_V110 and RA6T2_MCILV1_SPM_LESS_FOC_E2S_V110 (without Tuner Library). The integrated development environment is e2 studio.

14.6.4.1 Including Procedure Overview

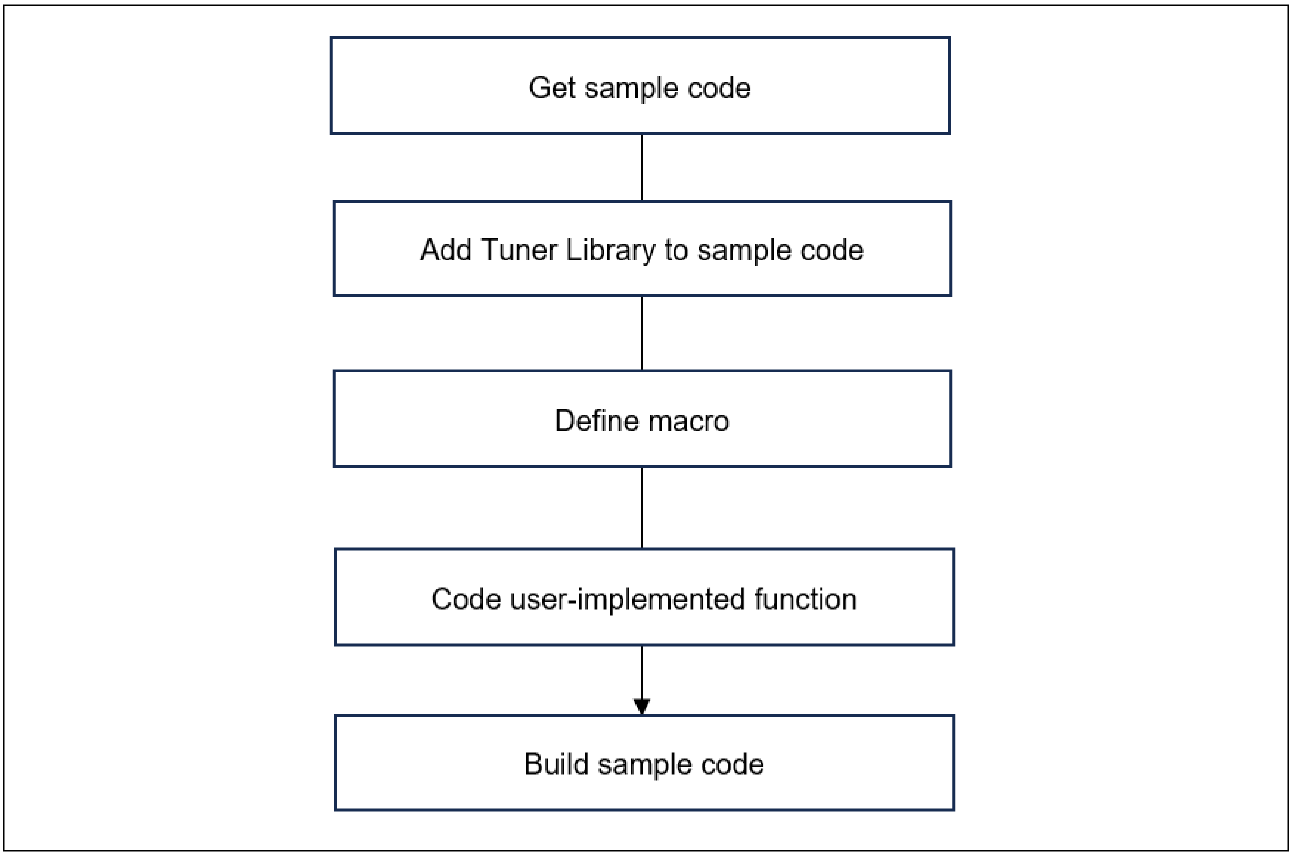


Figure 14-19 Procedure to Include Tuner Library

14.6.4.2 Getting Sample Code

Get the sample code from the following URL:

[RX]

https://www.renesas.com/us/en/document/scd/sensorless-vector-control-permanent-magnet-synchronous-motor-mck-rev110

[RA]

https://www.renesas.com/us/en/document/scd/sensorless-vector-control-permanent-magnet-synchronous-motor-mckmcb-ra-family-sample-code

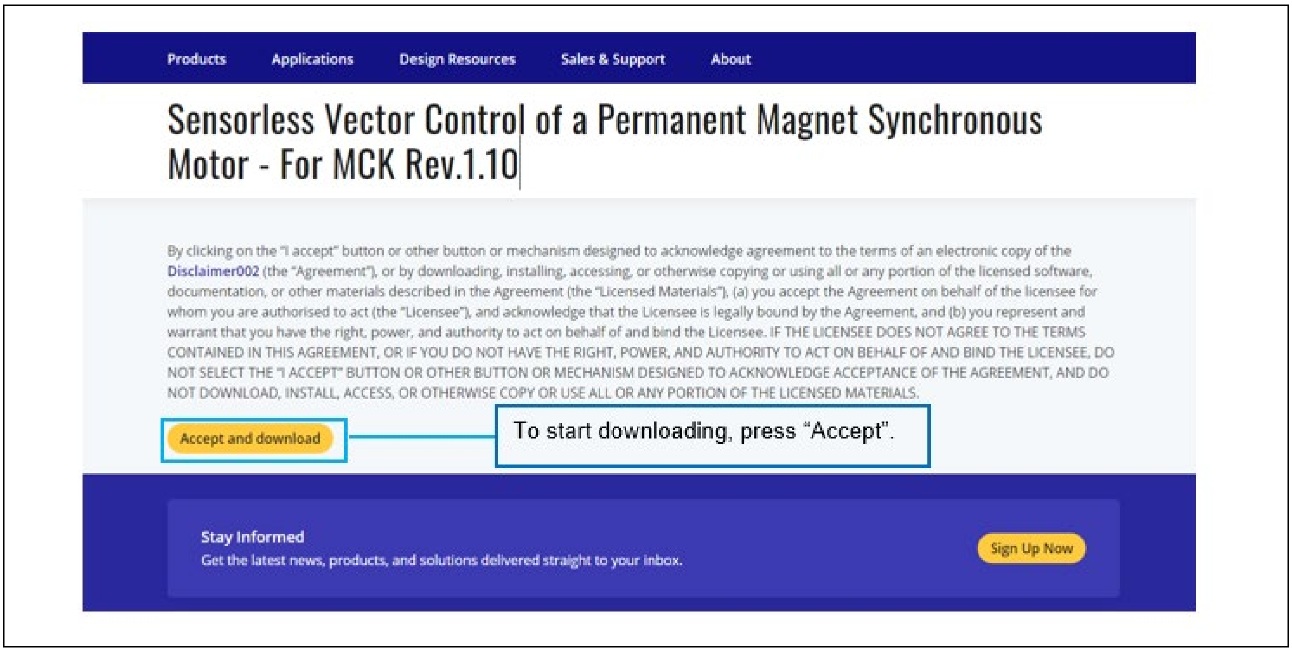


Figure 14-20 Getting Sample Code

When unzipping zip file, you can see several sample codes in the workspace. The sample code to use is RX26T_MCBA_MCILV1_SPM_LESS_FOC_E2S_V110 and RA6T2_MCILV1_SPM_LESS_FOC_E2S_V110.

14.6.4.3 [RX] Adding Tuner Library to Sample Code

Add Tuner Library from the sample code that includes Tuner function. Copy the files from the app/tuner and tuner folders of the downloaded sample code RX26T_MCBA_MCILV1_SPM_LESS_FOC_TUNER_E2S_V110 (Tuner function included).

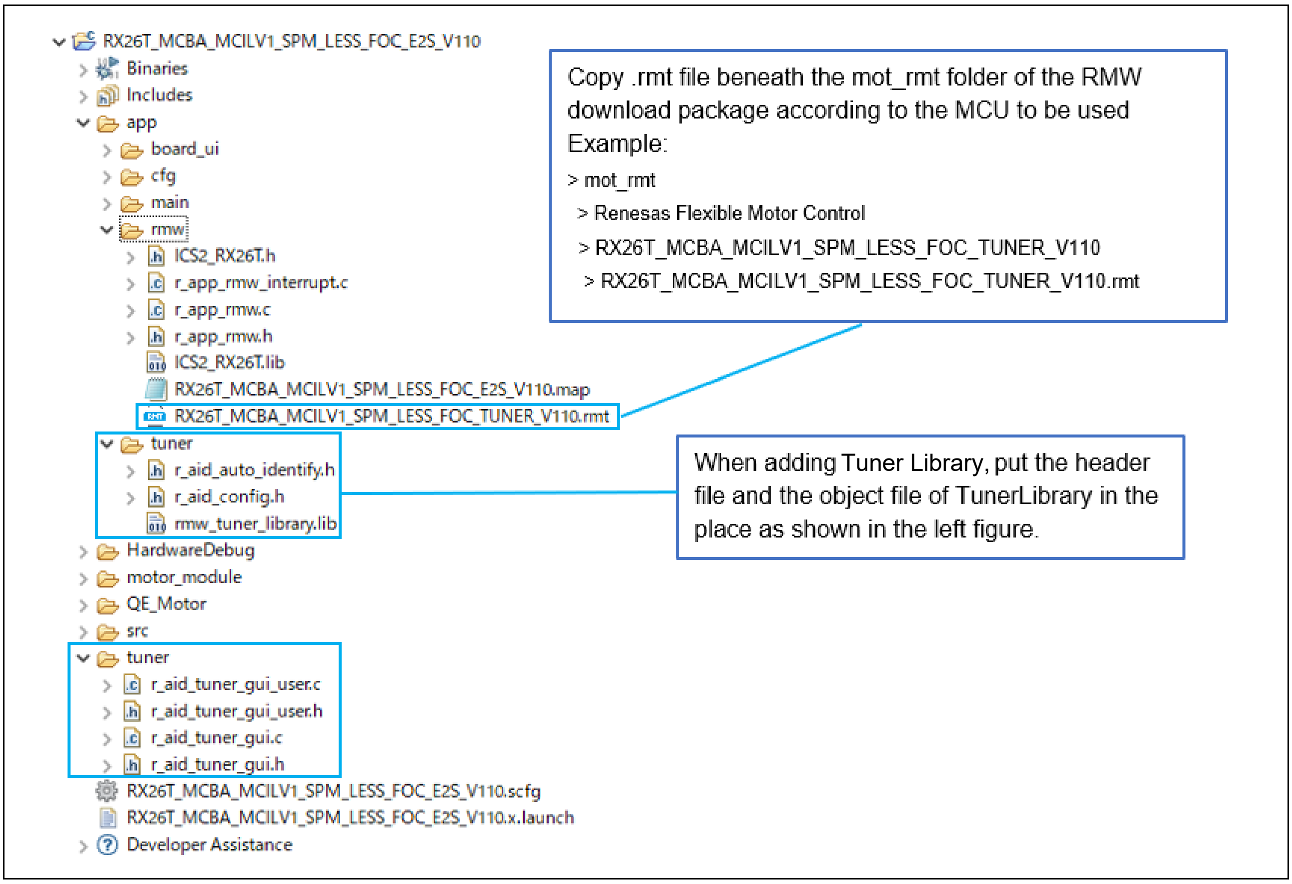


Figure 14-21 Adding Tuner Library to Sample Program

Set the include paths. Add the following two paths of the newly added folders.

RX26T_MCBA_MCILV1_SPM_LESS_FOC_TUNER_E2S_V110\app\tuner

RX26T_MCBA_MCILV1_SPM_LESS_FOC_TUNER_E2S_V110\tuner

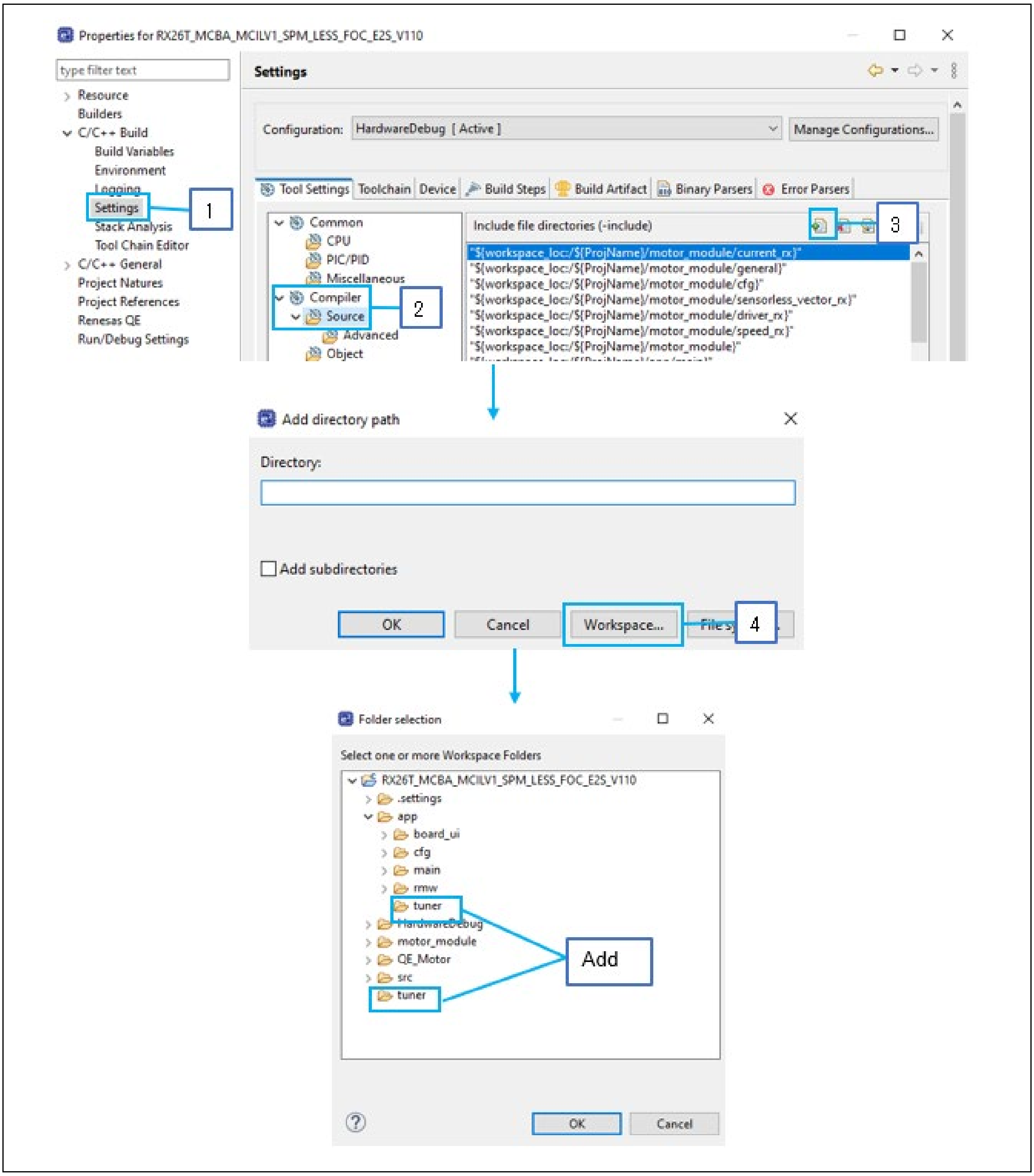


Figure 14-22 Setting Include Path

Specify the Tuner Library to link. Display the property window, and add Tuner Library by following the steps below.

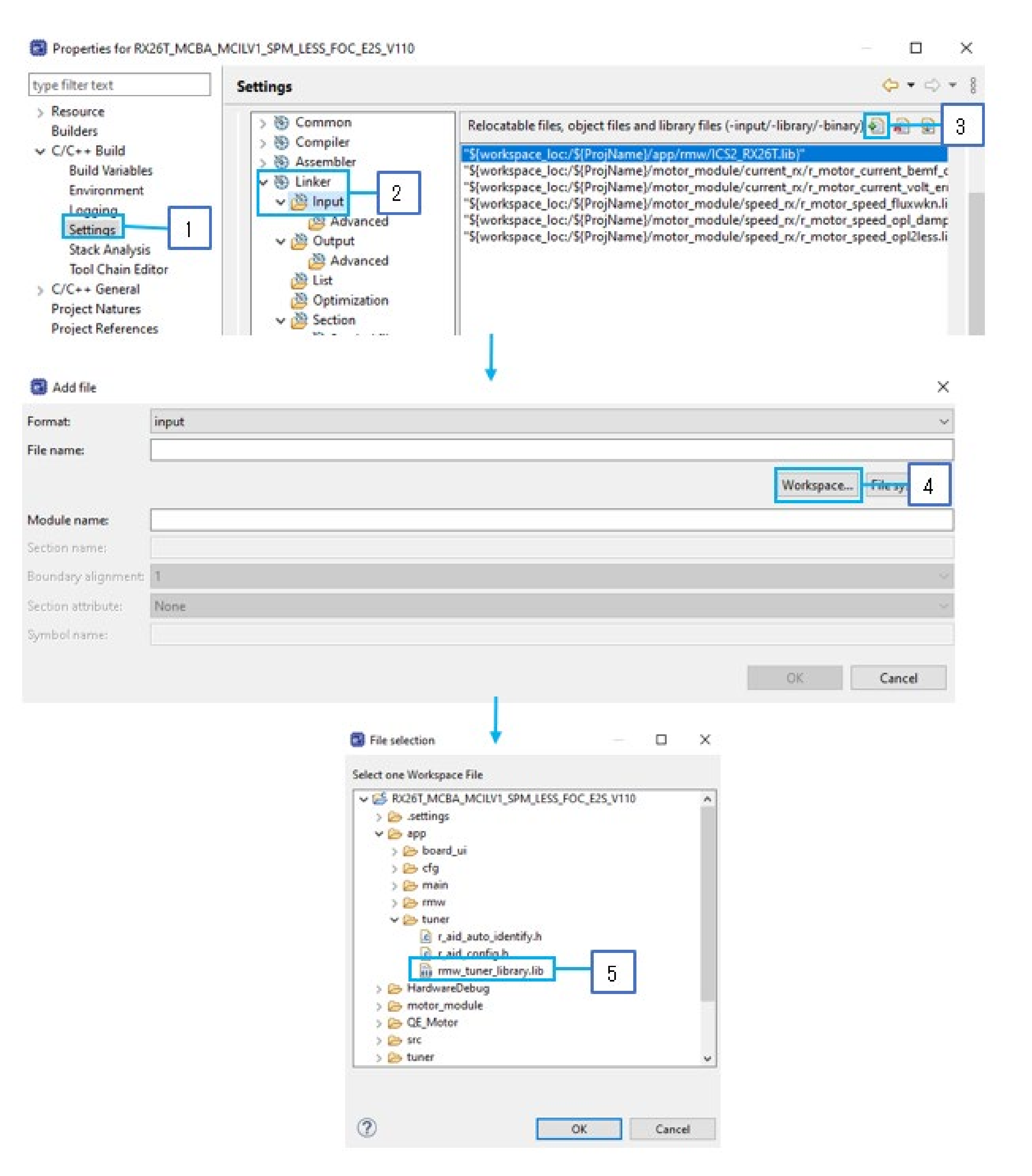


Figure 14-23 Adding Tuner Library

14.6.4.4 [RA] Adding Tuner Library to Sample Code

Add Tuner Library from the sample code that includes Tuner function. Copy the files from the app/tuner and tuner folders of the downloaded sample code RA6T2_MCILV1_SPM_LESS_FOC_TUNER_E2S_V110 (Tuner function included).

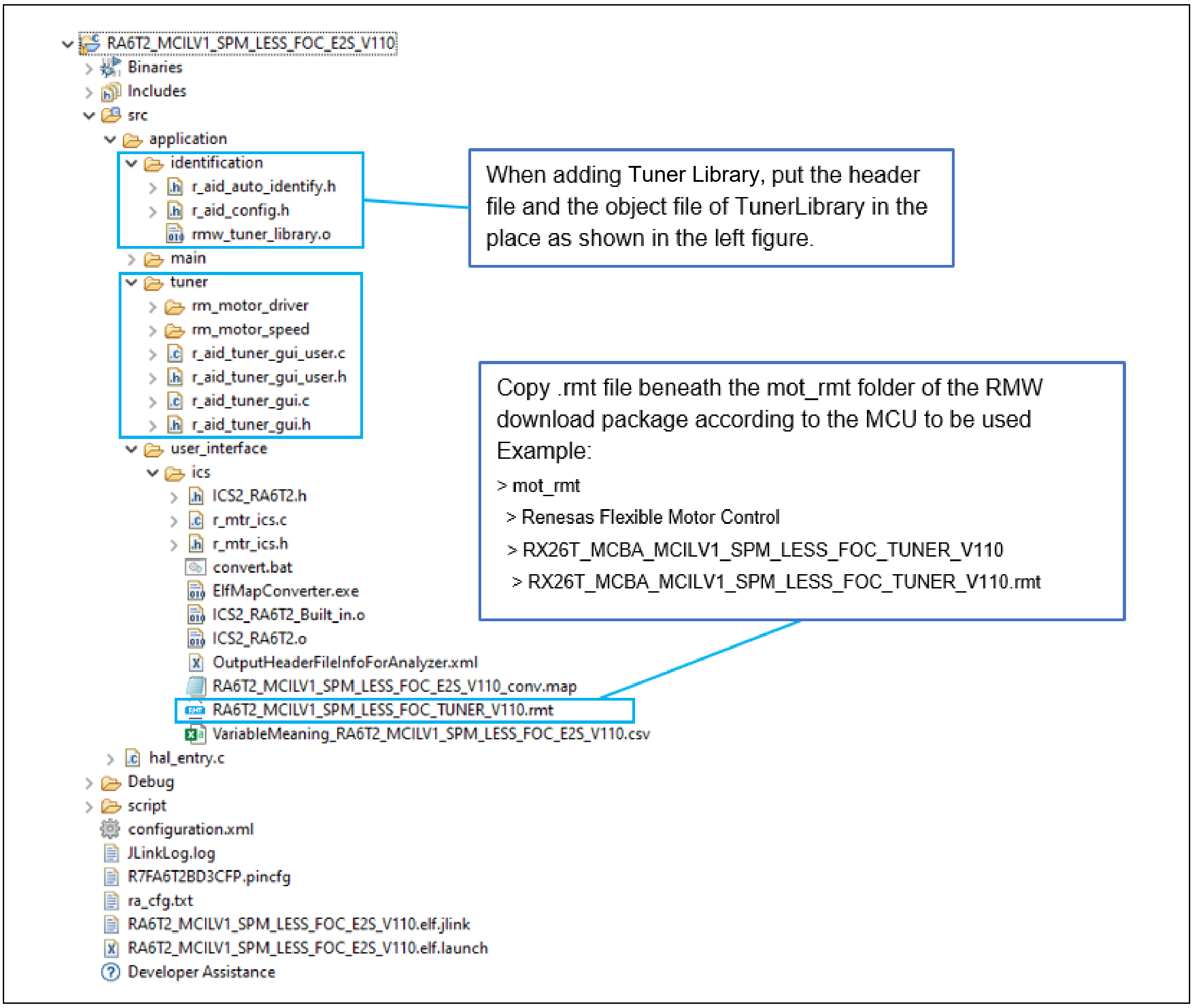


Figure 14-24 Adding Tuner Library to Sample Program

Set the include paths. Add the following two paths of the newly added folders.

RA6T2_MCILV1_SPM_LESS_FOC_TUNER_E2S_V110\src\application\identification

RA6T2_MCILV1_SPM_LESS_FOC_TUNER_E2S_V110\src\application\tuner

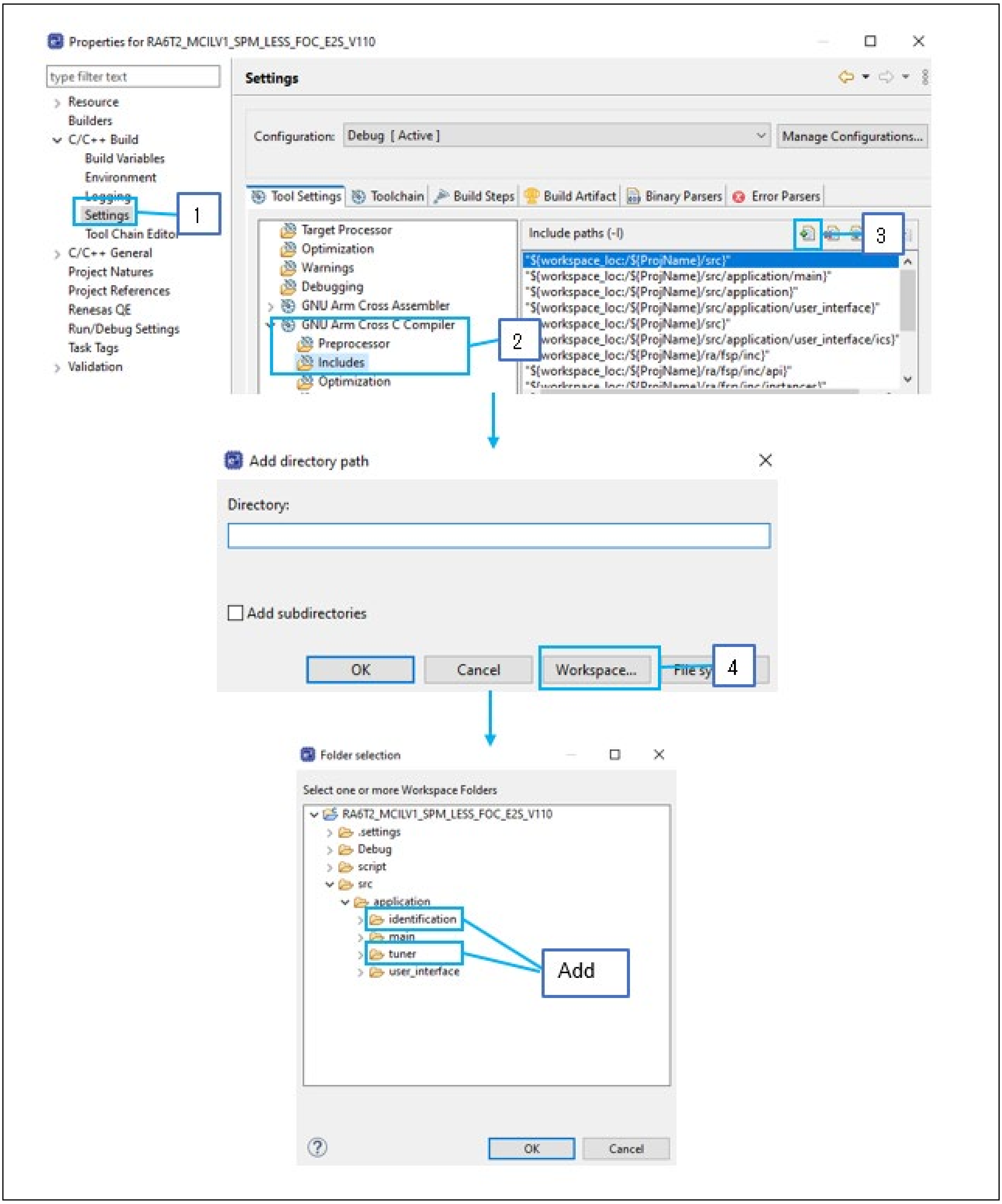


Figure 14-25 Setting Include Path

Specify the Tuner Library to link. Display the property window, and add Tuner Library by following the steps below.

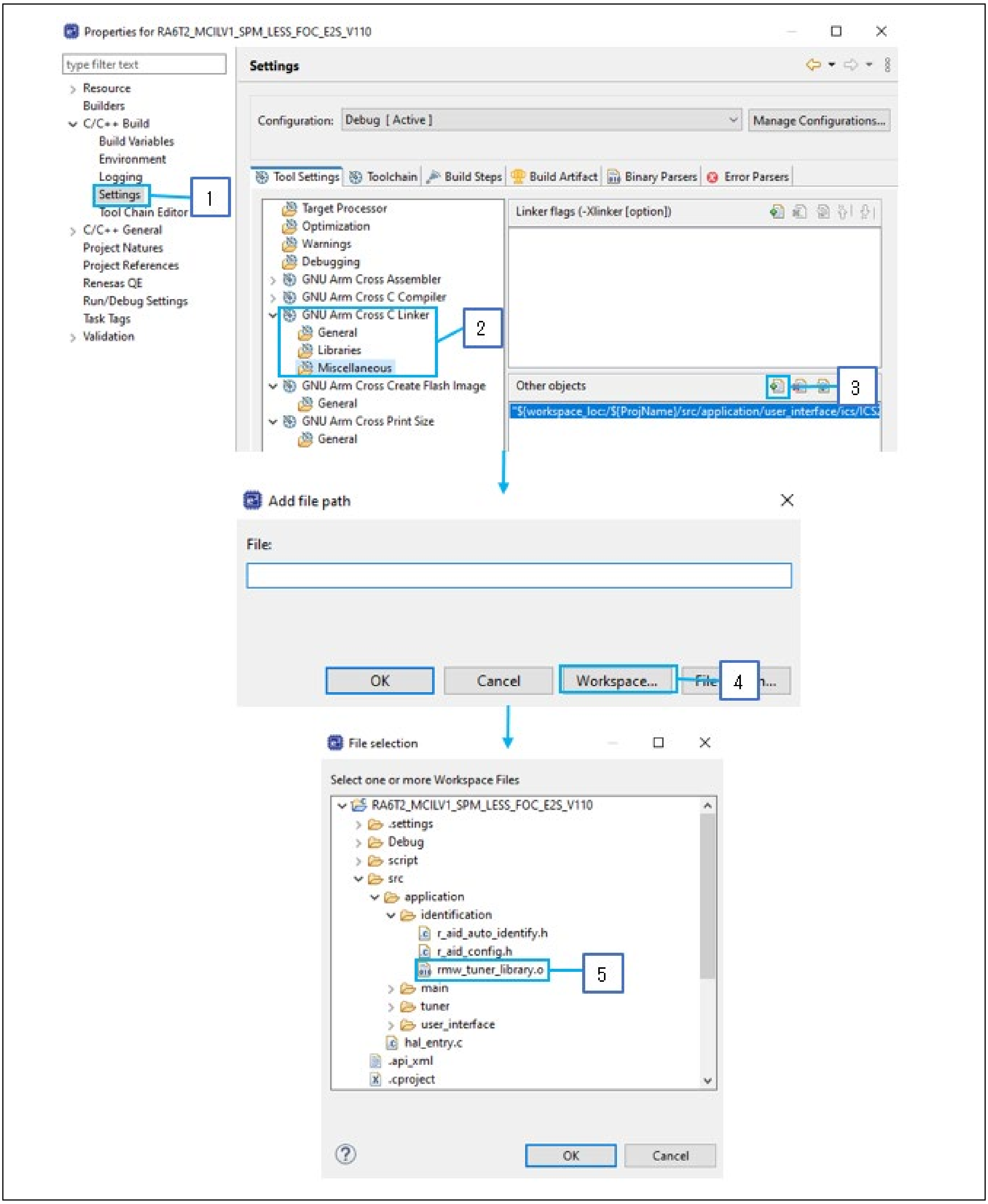


Figure 14-26 Adding Tuner Library

14.6.4.5 Defining Macro

Define macros in [r_aid_tuner_gui_user.h].

Table 14.7 lists macro definitions in the sample code. Change them if necessary.

Table 14-10 Macro List

Macro Definition	Description
AIDU_CURRENT_OMEGA	Natural frequency related to current control
AIDU_CURRENT_ZETA	Damping coefficient related to current control
AIDU_SPEED_OMEGA	Natural frequency related to speed control
AIDU_SPEED_ZETA	Damping coefficient related to speed control
AIDU_E_OBS_OMEGA	Natural frequency related to induced-voltage observer
AIDU_E_OBS_ZETA	Damping coefficient related to induced-voltage observer
AIDU_PLL_EST_OMEGA	Natural frequency related to PLL speed estimation
AIDU_PLL_EST_ZETA	Damping coefficient related to PLL speed estimation
AIDU_INPUT_V	Input voltage
AIDU_SPEED_CTRL_PERIOD	Period of speed control system
AIDU_INT_DECIMATION	Interrupt decimation initial value
AIDU_INV_INFO_OVERVOLTAGE_TH	Overvoltage error threshold [V]
AIDU_INV_INFO_UNDERVOLTAGE_TH	Undervoltage error threshold [V]
AIDU_INV_INFO_PWM_CYCLE_S	PWM cycle [s]
AIDU_INV_INFO_PWM_DEADTIME_S	Dead time [s]
AIDU_CARRIER_SET_BASE	Value to store in PWM carrier cycle setting register (PWM timer cycle [Hz] / Carrier cycle [Hz] / 2)
AIDU_DEADTIME_SET	Value to store in dead time setting register
AIDU_INV_INFO_OVERCURRENT_TH	Overcurrent error threshold [A]
AIDU_INV_INFO_CURRENT_RANGE	Detected current range [A] Sets maximum and minimum values of AD conversion range
AIDU_PWM_TIMER_W_REG	Pointer to W-phase PWM timer count register
AIDU_PWM_TIMER_V_REG	Pointer to V-phase PWM timer count register
AIDU_PWM_TIMER_U_REG	Pointer to U-phase PWM timer count register

14.6.4.6 [RX] Calling Tuner Library Function

− Compile option definition

Add the macro definition to be used as the compile option into the following file:

app\cfg\r_app_control_cfg.h

```
#define APP_CFG_SCI_CH_SELECT      (0x60)

/* Select using Tuner */
#define USE_RMW_TUNER              (1)

#endif /* R_APP_CONTROL_CFG_H */
```

Figure 14-27 Adding to [r_app_control_cfg.h] File

- Implementation of initialization process

Add the initialization process to the following file:

app\main\r_app_main.c

```
#if USE_RWM_TUNER
#include "r_aid_tuner_gui.h"
#include "r_aid_auto_identify.h"
#endif

* Function Name : main
void main(void)
{
clrpsw_i();                                     /* Disable interrupt */

    /* LED off */
    r_app_board_ui_led_control(STATEMACHINE_STATE_STOP);

    /* Initialize ICS */
    ics2_init((void*)g_dtc_table, APP_CFG_SCI_CH_SELECT, ICS_INT_LEVEL, ICS_BRR, ICS_INT_MODE);

    /* Initialize open motor control instance */
    r_app_main_init_motor_ctrl();

    /* Initialize RMW communication support for motor control
     * MUST be called after motor instance being configured */
    r_app_rmw_ui_init();

#if USE_RMW_TUNER
    R_TUNER_InitGUI();
#endif
```

Figure 14-28 Adding to [r_app_main.c] File

− Implementation of Tuner function operation controlling process

Add the Tuner function operation controlling process to the following file:

app\main\r_app_main.c

```
/*==============================*/
/*        Execute event         */
/*==============================*/
if (MAIN_UI_RMW == g_u1_sw_userif)
{
    /* Main process for ICS UI */
    r_app_rmw_ui_mainloop();
#if USE_RMW_TUNER
    R_TUNER_MainLoop();
#endif
}
else if (MAIN_UI_BOARD == g_u1_sw_userif)
{
    /* Main process for board UI */
    r_app_board_ui_mainloop();
}
else
{
    /* Do Nothing */
}
```

Figure 14-29 Adding to [r_app_main.c] File

- Implementation of Tuner result setting process

Add Turning result setting process to the following file:

app\rmw\r_app_rmw.c

```
void r_app_rmw_interrupt_handler(void)
{
    s_u1_cnt_ics++;

    /* Decimation of ICS call */
    if (ICS_DECIMATION < s_u1_cnt_ics)
    {
        s_u1_cnt_ics = 0;

        /* Call ICS */
        ics2_watchpoint();
    }

    /* Update commands and configurations when trigger flag is set */
    if (1 == g_u1_update_param_flag)
    {
        r_app_rmw_update_params();
#if USE_RMW_TUNER
        R_TUNER_SetTuneResult();
#endif
        g_u1_update_param_flag = 0;
    }
    else
    {
        if (MAIN_UI_RMW == g_u1_sw_userif)
        {
            r_app_rmw_update_command();
        }
    }
} /* End of function r_app_rmw_interrupt_handler */

* Function Name : r_app_rmw_system_mode
```

Figure 14-30 Adding to [r_app_rmw.c] File

− Adding interrupt process

Add the interrupt process to the following three files:

src\Config_CMT0\Config_CMT0_user.c

src\Config_MOTOR\Config_MOTOR_user.c

src\Config_POE\Config_POE_user.c

```
#include "r_app_rmw.h"
#if USE_RMW_TUNER
#include "r_aid_tuner_gui.h"
#include "r_aid_auto_identify.h"
#endif

#if FAST_INTERRUPT_VECTOR == VECT_CMT0_CMI0
#pragma interrupt r_Config_CMT0_cmi0_interrupt(vect=VECT(CMT0,CMI0),fint)
#else
#pragma interrupt r_Config_CMT0_cmi0_interrupt(vect=VECT(CMT0,CMI0))
#endif
static void r_Config_CMT0_cmi0_interrupt(void)
{
    /* Start user code for r_Config_CMT0_cmi0_interrupt. Do not edit comment generated here */
#if USE_RMW_TUNER
    if(R_TUNER_IsRunning())
    {
        R_AID_SpeedCtrlISR();
    }
    else
    {
        R_MOTOR_SENSORLESS_VECTOR_SpeedInterrupt(&g_st_sensorless_vector);
    }
#else
    R_MOTOR_SENSORLESS_VECTOR_SpeedInterrupt(&g_st_sensorless_vector);
#endif
    /* End user code. Do not edit comment generated here */
}
```

Figure 14-31 Adding to [Config_CMT0_user.c] File

```
#if USE_RMW_TUNER
#include "r_aid_tuner_gui.h"
#include "r_aid_auto_identify.h"
#endif

#if FAST_INTERRUPT_VECTOR == VECT_S12AD_S12ADI
#pragma interrupt r_Config_MOTOR_ad_interrupt(vect=VECT(S12AD,S12ADI),fint)
#else
#pragma interrupt r_Config_MOTOR_ad_interrupt(vect=VECT(S12AD,S12ADI))
#endif
static void r_Config_MOTOR_ad_interrupt(void)
{
    /* Start user code for r_Config_MOTOR_ad_interrupt-1. Do not edit comment generated here */
#if USE_RMW_TUNER
    if(R_TUNER_IsRunning())
    {
        R_AID_CurrentCtrlISR();
    }
    else
    {
        R_MOTOR_SENSORLESS_VECTOR_CurrentInterrupt(&g_st_sensorless_vector);
    }
#else
    R_MOTOR_SENSORLESS_VECTOR_CurrentInterrupt(&g_st_sensorless_vector);
#endif
    r_app_rmw_interrupt_handler();
    /* End user code. Do not edit comment generated here */

    /* Start user code for r_Config_MOTOR_ad_interrupt-2. Do not edit comment generated here */
    /* End user code. Do not edit comment generated here */
}
```

Figure 14-32 Adding to [Config_MOTOR_user.c] File

```
#include "r_app_rmw.h"
#if USE_RMW_TUNER
#include "r_aid_tuner_gui.h"
#include "r_aid_auto_identify.h"
#endif

void r_Config_POE_oei1_interrupt(void)
{
    /* Start user code for r_Config_POE_oei1_interrupt. Do not edit comment generated here */
#if use_RMW_TUNER
    if (R_TUNER_IsRunning())
    {
        R_AID_UserError(1);
    }
    else
    {
        R_MOTOR_SENSORLESS_VECTOR_OverCurrentInterrupt(&g_st_sensorless_vector);
        R_Config_POE_Stop();
        R_Config_POE_Start();
    }
#else
    R_MOTOR_SENSORLESS_VECTOR_OverCurrentInterrupt(&g_st_sensorless_vector);
    R_Config_POE_Stop();
    R_Config_POE_Start();
#endif

    /* End user code. Do not edit comment generated here */
}
```

Figure 14-33 Adding to [Config_POE_user.c] File

− Changing initial value of variable

Change the initial values of the variables in the following file:

src\app\r_app_rmw.c

For the setting values of “g_u2_conf_tool”, refer to Section 2.3.5.

```
* Global variables
uint16_t     g_u2_conf_hw   = 0x0008;          /* 0000000000001000b */
#if USE_RMW_TUNER
uint16_t     g_u2_conf_sw   = 0x4000;          /* 0100000000000000b */
uint16_t     g_u2_conf_tool = 0x0600;          /* 0000011000000000b */
#else
uint16_t     g_u2_conf_sw   = 0x0000;          /* 0000000000000000b */
uint16_t     g_u2_conf_tool = 0x0200;          /* 0000001000000000b */
#endif
uint8_t      gui_u1_active_gui;
uint16_t     g_u2_conf_sw_ver;
```

Figure 14-34 Adding to [r_app_rmw.c] File

・Adding variables and setting macro definitions in Tuner Library

Configure settings according to the pin assignment on your board in the following file:

tuner\r_aid_tuner_gui_user.h

```
 Macro definitions
#define AIDU_CURRENT_OMEGA               (CURRENT_CFG_OMEGA)           /* Natural frequency of current loop */
#define AIDU_CURRENT_ZETA                (CURRENT_CFG_ZETA)            /* Damping ratio of current loop */
#define AIDU_SPEED_OMEGA                 (SPEED_CFG_OMEGA)             /* Natural frequency of speed loop */
#define AIDU_SPEED_ZETA                  (SPEED_CFG_ZETA)              /* Damping ratio of speed loop */
#define AIDU_E_OBS_OMEGA                 (CURRENT_CFG_E_OBS_OMEGA)     /* Natural frequency of BEMF observer */
#define AIDU_E_OBS_ZETA                  (CURRENT_CFG_E_OBS_ZETA)      /* Damping ratio of BEMF observer */
#define AIDU_PLL_EST_OMEGA               (CURRENT_CFG_PLL_EST_OMEGA)   /* Natural frequency of PLL Speed estimate loop */
#define AIDU_PLL_EST_ZETA                (CURRENT_CFG_PLL_EST_ZETA)    /* Damping ratio of PLL Speed estimate loop */
#define AIDU_INPUT_V                     (INVERTER_CFG_INPUT_V)        /* Damping ratio of PLL Speed estimate loop */
#define AIDU_SPEED_CTRL_PERIOD           (SPEED_CFG_CTRL_PERIOD)       /* The speed control period[s] */
#define AIDU_INT_DECIMATION              (0)
#define AIDU_INV_INFO_OVERVOLTAGE_TH     (INVERTER_CFG_OVERVOLTAGE_LIMIT)
#define AIDU_INV_INFO_UNDERVOLTAGE_TH    (INVERTER_CFG_UNDERVOLTAGE_LIMIT)
#define AIDU_INV_INFO_PWM_CYCLE_S        (MOTOR_COMMON_CTRL_PERIOD)
#define AIDU_INV_INFO_PWM_DEADTIME_S     (INVERTER_CFG_DEADTIME/1000000.0f)
#define AIDU_CARRIER_SET_BASE            (MOTOR_COMMON_CARRIER_SET_BASE)
#define AIDU_DEADTIME_SET                (MOTOR_COMMON_DEADTIME_SET)

#define AIDU_INV_INFO_OVERCURRENT_TH     (5.0f * MTR_SQRT_2 * MOTOR_COMMON_CFG_OVERCURRENT_MARGIN_MULT)
#define AIDU_INV_INFO_CURRENT_RANGE      (INVERTER_CFG_ADC_REF_VOLTAGE / (INVERTER_CFG_CURRENT_AMP_GAIN * INVERTER_CFG_SHUNT_RESIST))

#define AIDU_PWM_TIMER_U_REG             ((MTU3.TGRD))
#define AIDU_PWM_TIMER_V_REG             ((MTU4.TGRC))
#define AIDU_PWM_TIMER_W_REG             ((MTU4.TGRD))

#define AIDU TUNE VOLTERR ENABLE         (true)
```

Figure 14-35 Adding to [r_aid_tuner_gui_user.h] File

14.6.4.7 [RA] Calling Tuner Library Function

・Implementation of initialization process

Add the initialization process to the following file:

src\application\main\mtr_main.c

```
/******************************************************
* Includes <System Includes> , "Project Includes"
******************************************************/
#include <stdint.h>
#include "mtr_main.h"
#include "hal_data.h"
#include "ics2_RA6T2.h"
#include "r_mtr_ics.h"

/* USE_RMW_TUNER */
#include "r_aid_tuner_gui.h"
#include "r_aid_auto_identify.h"
/* USE_RMW_TUNER */
```

Figure 14-36 Adding to [mtr_main.c] File

```
/* USE_RMW_TUNER */
uint16_t g_u2_tune_mode = 0;
uint16_t g_u2_tune_mode_pre = 0;

void rm_motor_driver_cyclic_tuner (adc_callback_args_t * p_args);
void rm_motor_speed_cyclic_tuner (timer_callback_args_t * p_args);
/* USE_RMW_TUNER */

/**********************************************************************
* Function Name : mtr_init
* Description   : Initialization for Motor Control
* Arguments     : None
* Return Value  : None
**********************************************************************
void mtr_init(void)
{
    int i;
    uint8_t u1_conf_motor_type[] = CONF_MOTOR_TYPE;
    uint8_t u1_conf_control[] = CONF_CONTROL;
    uint8_t u1_conf_inverter[] = CONF_INVERTER;
```

Figure 14-37 Adding to [mtr_main.c] File

```
    /* Execute reset event */
    g_motor_sensorless0.p_api->reset(g_motor_sensorless0.p_ctrl);

/* USE_RMW_TUNER */
    R_TUNER_InitGUI();
/* USE_RMW_TUNER */
} /* End of function mtr_init() */

/**********************************************************************
* Function Name : mtr_main
* Description   : Main routine for Motor Control
* Arguments     : None
* Return Value  : None
**********************************************************************
void mtr_main(void)
{
    /*** select user interfaces ***/
    if (g_u1_sw_userif != com_u1_sw_userif)
```

Figure 14-38 Adding to [mtr_main.c] File

・Implementation of Tuner function operation controlling process

Add the Tuner function operation controlling process to the following file:

src\application\main\mtr_main.c

```
void mtr_main(void)
{
    /*** select user interfaces ***/
    if (g_u1_sw_userif != com_u1_sw_userif)
    {
        g_u1_sw_userif = com_u1_sw_userif;
        if (ICS_UI == g_u1_sw_userif)
        {
            g_u1_mode_system = g_u1_motor_status;
        }
    }

    if (BOARD_UI == g_u1_sw_userif)
    {
        board_ui();                     /* User interface control routine */
    }
    else if (ICS_UI == g_u1_sw_userif)
    {
        ics_ui();                       /* User interface using ICS */
/* USE_RMW_TUNER */
        R_TUNER_MainLoop();
        g_u2_tune_mode = R_TUNER_IsRunning();
        if (g_u2_tune_mode != g_u2_tune_mode_pre)
        {

            if (g_u2_tune_mode == 1)
            {
                g_user_motor_driver_cfg.p_adc_instance->p_api->callbackSet(g_user_motor_driver_cfg.p_adc_instance->p_ctrl,
                                                                           rm_motor_driver_cyclic_tuner,
                                                                           &g_motor_driver0_ctrl,
                                                                           &(g_motor_driver0_ctrl.adc_callback_args));
                g_user_motor_speed_cfg.p_timer_instance->p_api->callbackSet(g_user_motor_speed_cfg.p_timer_instance->p_ctrl,
                                                                           rm_motor_speed_cyclic_tuner,
                                                                           &g_motor_speed0,
                                                                           &(g_motor_speed0_ctrl.timer_args));
            }
            else
            {
                g_user_motor_driver_cfg.p_adc_instance->p_api->callbackSet(g_user_motor_driver_cfg.p_adc_instance->p_ctrl,
                                                                           rm_motor_driver_cyclic,
                                                                           &g_motor_driver0_ctrl,
                                                                           &(g_motor_driver0_ctrl.adc_callback_args));
                g_user_motor_speed_cfg.p_timer_instance->p_api->callbackSet(g_user_motor_speed_cfg.p_timer_instance->p_ctrl,
                                                                           rm_motor_speed_cyclic,
                                                                           &g_motor_speed0,
                                                                           &(g_motor_speed0_ctrl.timer_args));
                g_motor_sensorless0.p_api->stop(g_motor_sensorless0.p_ctrl);
            }
        g_u2_tune_mode_pre = g_u2_tune_mode;
        }
/* USE_RMW_TUNER */
    }
    else
    {
```

Figure 14-39 Add to [mtr_main.c] File

・Implementation of over current process
Add the over current process to the following file:

src\application\main\mtr_main.c

```
/*******************************************************************************************************************
* Function Name : g_poe_overcurrent
* Description   : POEG2 Interrupt callback function
* Arguments     : p_args - Callback argument
* Return Value  : None
*******************************************************************************************************************/
void g_poe_overcurrent(poeg_callback_args_t *p_args)
{
/* USE_RMW_TUNER */
    if (R_TUNER_IsRunning())
    {
        R_AID_UserError(1);
    }
    else
    {
        if (p_args != NULL)
        {
            R_POEG_Reset(g_poeg0.p_ctrl);
            g_motor_sensorless0.p_api->errorSet(g_motor_sensorless0.p_ctrl, MOTOR_ERROR_OVER_CURRENT_HW);
            g_u2_chk_error |= MOTOR_ERROR_OVER_CURRENT_HW;
        }
    }
/* USE_RMW_TUNER */
} /* End of function g_poe_overcurrent */
```

Figure 14-40 Add to [mtr_main.c] File

・Changing initial value of variable

Change the initial values of the variables in the following file:

src\application\main\mtr_main.c

For the setting values of “g_u2_conf_tool”, refer to Section 2.3.5.

```
void mtr_init(void)
{
    int i;
    uint8_t u1_conf_motor_type[] = CONF_MOTOR_TYPE;
    uint8_t u1_conf_control[] = CONF_CONTROL;
    uint8_t u1_conf_inverter[] = CONF_INVERTER;
    g_u1_conf_motor_type_len = CONF_MOTOR_TYPE_LEN;
    g_u1_conf_control_len     = CONF_CONTROL_LEN;
    g_u1_conf_inverter_len    = CONF_INVERTER_LEN;
    for (i = 0; i < g_u1_conf_motor_type_len; i++)
    {
        g_u1_conf_motor_type[i] = u1_conf_motor_type[i];
    }
    for (i = 0; i < g_u1_conf_control_len; i++)
    {
        g_u1_conf_control[i] = u1_conf_control[i];
    }
    for (i = 0; i < g_u1_conf_inverter_len; i++)
    {
        g_u1_conf_inverter[i] = u1_conf_inverter[i];
    }
    g_u2_conf_hw = 0x0008;                        /* 0000000000001000b */
/* USE_RMW_TUNER */
    g_u2_conf_sw   = 0x4000;                      /* 0100000000000000b */
    g_u2_conf_tool = 0x0600;                      /* 0000011000000000b */
/* USE_RMW_TUNER */

    motor_fsp_init();
    software_init();                              /* Initialize private global variables */
```

Figure 14-41 Add to [mtr_main.c] File

・Adding variables and setting macro definitions in Tuner Library

Configure settings according to the pin assignment on your board in the following file:

src\application\tuner\r_aid_tuner_gui_user.h

```
 Macro definitions
#define AIDU_CURRENT_OMEGA                 (g_user_motor_current_extended_cfg.p_design_par
#define AIDU_CURRENT_ZETA                  (g_user_motor_current_extended_cfg.p_design_par
#define AIDU_SPEED_OMEGA                   (g_user_motor_speed_extended_cfg.d_param.f_spee
#define AIDU_SPEED_ZETA                    (g_user_motor_speed_extended_cfg.d_param.f_spee
#define AIDU_E_OBS_OMEGA                   (g_user_motor_estimate_extended_cfg.f_e_obs_ome
#define AIDU_E_OBS_ZETA                    (g_user_motor_estimate_extended_cfg.f_e_obs_zet
#define AIDU_PLL_EST_OMEGA                 (g_user_motor_estimate_extended_cfg.f_pll_est_o
#define AIDU_PLL_EST_ZETA                  (g_user_motor_estimate_extended_cfg.f_pll_est_z
#define AIDU_INPUT_V                       (g_user_motor_driver_extended_cfg.mod_param.f4_
#define AIDU_SPEED_CTRL_PERIOD             (g_user_motor_speed_extended_cfg.f_speed_ctrl_p
#define AIDU_INT_DECIMATION                (0)
#define AIDU_INV_INFO_OVERVOLTAGE_TH       (g_user_motor_sensorless_extended_cfg.f_overvol
#define AIDU_INV_INFO_UNDERVOLTAGE_TH      (g_user_motor_sensorless_extended_cfg.f_lowvolt
#define AIDU_INV_INFO_PWM_CYCLE_S          (g_user_motor_current_extended_cfg.f_current_ct
#define AIDU_INV_INFO_PWM_DEADTIME_S       (g_user_motor_driver_extended_cfg.u2_deadtime/1
#define AIDU_CARRIER_SET_BASE              (g_user_motor_driver_extended_cfg.u2_pwm_timer_
#define AIDU_DEADTIME_SET                  (g_user_motor_driver_extended_cfg.u2_pwm_timer_

#define AIDU_INV_INFO_OVERCURRENT_TH       (g_user_motor_sensorless_extended_cfg.f_overcur
#define AIDU_INV_INFO_CURRENT_RANGE        (g_user_motor_driver_extended_cfg.f_current_ran

#define AIDU_PWM_TIMER_U_REG               (&(R_GPT4->GTCCR[2]))
#define AIDU_PWM_TIMER_V_REG               (&(R_GPT5->GTCCR[2]))
#define AIDU_PWM_TIMER_W_REG               (&(R_GPT6->GTCCR[2]))

#define AIDU_TUNE_VOLTERR_ENABLE           (true)
```

Figure 14-42 Adding to [r_aid_tuner_gui_user.h] File

14.6.4.8 Coding User-Implemented Functions

It is necessary to set user-implemented functions so that Tuner Library can access peripheral functions. Tuner Library accesses the peripheral functions from the function pointers of the configured user-implemented functions. Code the process of the user-implemented function in [r_aid_tuner_gui_use.c] file. If the sample code is changed, change the part written in bule in the table according to the change.

Table 14-11 User-Implemented Function List [1/6]

<table>
<tr><th>Function name</th><th>Argument</th><th>Type</th><th>Description</th><th>Function pointer variable name</th></tr>
<tr><td rowspan="2">aid_mtr_inv_set_uvw</td><td>float f4_duty_u : U-phase duty
float f4_duty_v :
V-phase duty
float f4_duty_w : W-phase duty</td><td>void</td><td>Sets PWM duty</td><td rowspan="2">g_fp_aid_internal_inv_set_uvw</td></tr>
<tr><td colspan="3">void aid_mtr_inv_set_uvw(float f4_duty_u, float f4_duty_v, float f4_duty_w)
{
R_MOTOR_DRIVER_BldcDutySet(g_st_sensorless_vector.p_st_driver, f4_duty_u, f4_duty_v, f4_duty_w);
}</td></tr>
<tr><td rowspan="2">aid_mtr_inv_get_uvw</td><td>float *f4_duty_u : U-phase duty
float *f4_duty_v : V-phase duty
float *f4_duty_w : W-phase duty</td><td>void</td><td>Acquires the actual duty cycle for PWM output</td><td rowspan="2">-</td></tr>
<tr><td colspan="3">void aid_mtr_inv_get_uvw(float *f4_duty_u, float *f4_duty_v, float *f4_duty_w)
{
*f4_duty_u = 1.0f - (((float)AIDU_PWM_TIMER_U_REG - ((float)AIDU_DEADTIME_SET * 0.5f) - 1.0f) / (float)AIDU_CARRIER_SET_BASE);
*f4_duty_v = 1.0f - (((float)AIDU_PWM_TIMER_V_REG - ((float)AIDU_DEADTIME_SET * 0.5f) - 1.0f) / (float)AIDU_CARRIER_SET_BASE);
*f4_duty_w = 1.0f - (((float)AIDU_PWM_TIMER_W_REG - ((float)AIDU_DEADTIME_SET * 0.5f) - 1.0f) / (float)AIDU_CARRIER_SET_BASE);
}</td></tr>
</table>

Table 14-12 User-Implemented Function List [2/6]

<table>
<tr><th>Function name</th><th>Argument</th><th>Type</th><th>Description</th><th>Function pointer variable name</th></tr>
<tr><td rowspan="2">aid_mtr_get_current_iuiw</td><td>float *f4_iu_ad : U-phase current
float *f4_iw_ad : W-phase current</td><td>void</td><td>Acquires U-/W-phase current value
Constantly called during identification to acquire the latest value</td><td rowspan="2">g_fp_aid_internal_get_current_iuiw</td></tr>
<tr><td colspan="3">void aid_mtr_get_current_iuiw(float *f4_iu_ad, float *f4_iw_ad)
{
    r_mtr_adc_tb st_ad_data;

    g_st_sensorless_vector.p_st_driver->ADCDataGet(&st_ad_data);
    *f4_iu_ad = (MOTOR_MCU_CFG_ADC_OFFSET - st_ad_data.u2_iu_ad) * g_st_sensorless_vector.p_st_driver->f4_ad_crnt_per_digit;
    *f4_iw_ad = (MOTOR_MCU_CFG_ADC_OFFSET - st_ad_data.u2_iw_ad) * g_st_sensorless_vector.p_st_driver->f4_ad_crnt_per_digit;
}</td></tr>
<tr><td rowspan="2">aid_mtr_get_vdc</td><td>-</td><td>float : Supply voltage</td><td>Acquires supply voltage value.
Constantly called during identification to acquire the latest value</td><td rowspan="2">g_fp_aid_internal_get_vdc</td></tr>
<tr><td colspan="3">float aid_mtr_get_vdc(void)
{
    r_mtr_adc_tb st_ad_data;
    float temp_vdc;

    g_st_sensorless_vector.p_st_driver->ADCDataGet(&st_ad_data);
    temp_vdc = st_ad_data.u2_vdc_ad * g_st_sensorless_vector.p_st_driver->f4_ad_vdc_per_digit;
    return (temp_vdc);
}</td></tr>
</table>

Table 14-13 User-Implemented Function List [3/6]

<table>
<tr><th>Function name</th><th>Argument</th><th>Type</th><th>Description</th><th>Function pointer variable name</th></tr>
<tr><td rowspan="2">aid_mtr_ctrl_start</td><td>-</td><td>void</td><td>Transits PWM output that is controllable with function aid_mtr_inv_set_uvw into DRIVE mode
Called at START ID</td><td rowspan="2">g_fp_aid_internal_ctrl_start</td></tr>
<tr><td colspan="3">void aid_mtr_ctrl_start(void)
{
R_MOTOR_DRIVER_BldcDutySet(g_st_sensorless_vector.p_st_driver, 0.5f, 0.5f, 0.5f);

R_MOTOR_DRIVER_PWMControlStart(g_st_sensorless_vector.p_st_driver);
}</td></tr>
<tr><td rowspan="2">aid_mtr_ctrl_stop</td><td>-</td><td>void</td><td>Transits mode from DRIVE to STOP</td><td rowspan="2">g_fp_aid_internal_ctrl_stop</td></tr>
<tr><td colspan="3">void aid_mtr_ctrl_stop(void)
{
R_MOTOR_DRIVER_PWMControlStop(g_st_sensorless_vector.p_st_driver);

R_MOTOR_DRIVER_BldcDutySet(g_st_sensorless_vector.p_st_driver, 0.5f, 0.5f, 0.5f);
}</td></tr>
<tr><td rowspan="2">aid_mtr_clear_oc_flag</td><td>-</td><td>void</td><td>Clears forced shutdown flag at overcurrent detection</td><td rowspan="2">g_fp_aid_internal_clear_oc_flag</td></tr>
<tr><td colspan="3">void aid_mtr_clear_oc_flag(void)
{
    R_Config_POE_Stop();
    R_Config_POE_Start();
}</td></tr>
</table>

Table 14-14 User-Implemented Function List [4/6]

<table>
<tr><th>Function name</th><th>Argument</th><th>Type</th><th>Description</th><th>Function pointer variable name</th></tr>
<tr><td rowspan="2">rmw_apply_identified_params</td><td>-</td><td>void</td><td>Sets RMW parameters</td><td rowspan="2">-</td></tr>
<tr><td colspan="3">void rmw_apply_identified_params(void)
{
st_aid_id_setting_t id_setting;
float ia_max;
float va_max;

R_AID_GetIDSetting(&id_setting);
com_f4_nominal_current_rms = id_setting.f4_rated_current;
ia_max = com_f4_nominal_current_rms * (AIDU_SQRT_3 / AIDU_SQRT_2);
va_max = AIDU_INPUT_V * (AIDU_SQRT_3 / AIDU_SQRT_2) * AIDU_FLOAT_0_5 * AIDU_FLOAT_0_9;
com_u2_mtr_pp = id_setting.u2_num_pole_pairs;
com_f4_mtr_r = gui_f4_r;
com_f4_mtr_ld = gui_f4_ld;
com_f4_mtr_lq = gui_f4_lq;
com_f4_mtr_m = gui_f4_ke;
com_f4_mtr_j = gui_f4_j;

com_f4_max_speed_rpm = (va_max / gui_f4_ke) / (float)com_u2_mtr_pp / AIDU_TWOPI * AIDU_FLOAT_60_0;
com_f4_overspeed_limit_rpm = com_f4_max_speed_rpm * AIDU_FLOAT_1_5;
com_f4_ol_ref_id = ia_max * AIDU_FLOAT_0_8;
com_f4_id_down_speed_rpm = AIDU_FLOAT_0_3 * com_f4_max_speed_rpm;
com_f4_id_up_speed_rpm = AIDU_FLOAT_0_2 * com_f4_max_speed_rpm;
com_f4_ref_speed_rpm = com_f4_max_speed_rpm / 2;

com_u1_enable_write ^= 1;
}</td></tr>
<tr><td rowspan="2">rmw_apply_reset</td><td>-</td><td>void</td><td>Resets RMW parameters</td><td rowspan="2">-</td></tr>
<tr><td colspan="3">void rmw_apply_reset(void)
{
com_u1_system_mode = STATEMACHINE_EVENT_RESET;
}</td></tr>
</table>

Table 14-15 User-Implemented Function List [5/6]

<table>
<tr><th>Function name</th><th>Argument</th><th>Type</th><th>Description</th><th>Function pointer variable name</th></tr>
<tr><td rowspan="2">mtr_ics_1_parameter</td><td>-</td><td>void</td><td>Reflects settings in Easy tab of Tune window</td><td rowspan="2">-</td></tr>
<tr><td colspan="3"><pre>
void mtr_ics_1_parameter(void)
{
    uint8_t u1_temp;

    /* for 1 parameter */
    u1_temp = gui_u1_active_gui & 0x04;
    if ((0x04 == u1_temp) && (AIDU_1_PARAMETER_SET ==
gui_u1_flag_tune_mode))
    {
        if (gui_u1_flg_1para_init != AIDU_1_PARAMETER_SET)
        {
            gui_f4_1para_speed_omega = AIDU_SPEED_OMEGA;
            gui_u1_flg_1para_init = AIDU_1_PARAMETER_SET;
        }
        if (gui_f4_slide_parameter > 100.0f)
        {
            gui_f4_slide_parameter = 100.0f;
        }
        else if (gui_f4_slide_parameter < 0)
        {
            gui_f4_slide_parameter = 0.0f;
        }
        com_f4_current_omega_hz   =
AIDU_CURRENT_OMEGA;
        com_f4_current_zeta   =
AIDU_CURRENT_ZETA;
        com_f4_speed_omega_hz   = 1.0f +
((gui_f4_1para_speed_omega - 1.0f) * 2) * gui_f4_slide_parameter /
100.0f;
        com_f4_speed_zeta   =
AIDU_SPEED_ZETA;
        com_f4_e_obs_omega_hz   =
AIDU_E_OBS_OMEGA;
        com_f4_e_obs_zeta   =
AIDU_E_OBS_ZETA;
        com_f4_pll_est_omega_hz   =
AIDU_PLL_EST_OMEGA;
        com_f4_pll_est_zeta   =
AIDU_PLL_EST_ZETA;
    }
}
</pre></td></tr>
</table>

Table 14-16 User-Implemented Function List [6/6]

<table>
<tr><th>Function name</th><th>Argument</th><th>Type</th><th>Description</th><th>Function pointer variable name</th></tr>
<tr><td rowspan="2">aid_mtr_ics_interrupt</td><td>-</td><td>void</td><td>RMW interrupt process</td><td rowspan="2">-</td></tr>
<tr><td colspan="3">void aid_mtr_ics_interrupt(void)
{
if (R_TUNER_GetFlugReset() == AIDU_FLUG_RESET_MDOE1)
{
com_u1_system_mode = STATEMACHINE_EVENT_RESET;
R_TUNER_SetFlugReset(AIDU_FLUG_RESET_MDOE2);
}
}</td></tr>
<tr><td rowspan="2">aid_mtr_get_inv_info</td><td>st_aid_inv_info_t* st_inv_info : inv info structure pointer</td><td>void</td><td>Acquires inverter information</td><td rowspan="2">-</td></tr>
<tr><td colspan="3">void aid_mtr_get_inv_info(st_aid_inv_info_t* st_inv_info)
{
st_inv_info->duty_min = AIDU_INV_INFO_DUTY_MIN;
st_inv_info->duty_max = AIDU_INV_INFO_DUTY_MAX;
st_inv_info->overcurrent_th =
AIDU_INV_INFO_OVERCURRENT_TH;
st_inv_info->overvoltage_th =
AIDU_INV_INFO_OVERVOLTAGE_TH;
st_inv_info->undervoltage_th =
AIDU_INV_INFO_UNDERVOLTAGE_TH;
st_inv_info->pwm_cycle_s =
AIDU_INV_INFO_PWM_CYCLE_S;
st_inv_info->pwm_deadtime_s =
AIDU_INV_INFO_PWM_DEADTIME_S;
st_inv_info->pwm_lsb = 1.0f /
(float)AIDU_CARRIER_SET_BASE;
st_inv_info->current_lsb =
AIDU_INV_INFO_CURRENT_RANGE / 4096; /* Full current range /
ADC max digits */
}</td></tr>
</table>

14.6.4.9 Building Sample Code

After completing the change, build the sample code by following the steps below and confirm that no error occurs.

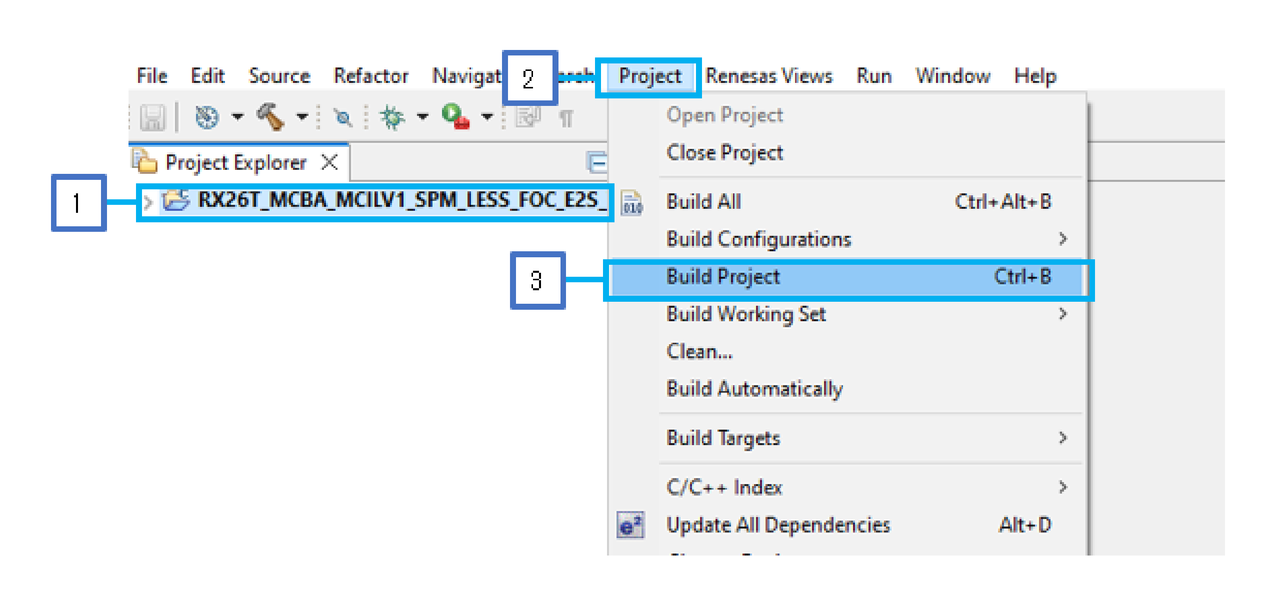


Figure 14-43 Building Sample Code

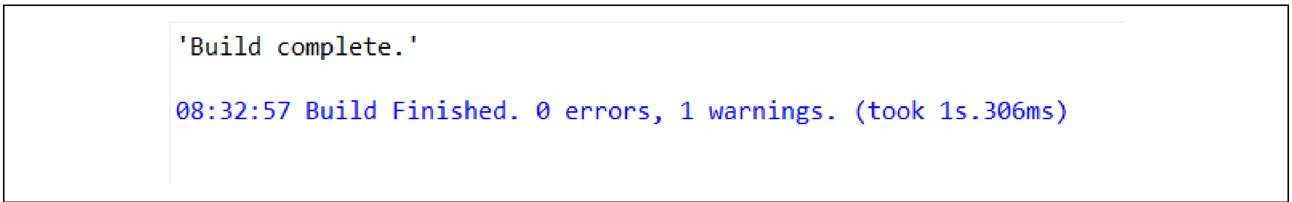


Figure 14-44 Result of Building Sample Code

If an error occurs, check the file that has been added or the code that has been added or changed with the steps in Section 14.6.4.3 and after.

15.Tuner Tool (for RL78)

15.1 Overview

This Section explains the method how to use the tuning function for RL78 Micro Controller. This function automatically measures specific parameters (e.g., resistance, inductance) of a motor (permanent magnet synchronous motor) and adjusts parameters which are required to sensorless vector control.

This function is effective with a target hardware which is downloaded support software in CPU. In that case, when “Renesas Motor Workbench” is connected, “Tuner” icon appears in “Main Window”.

Note : This tool is not guaranteed to work effectively for all motor.

15.2 Features

- Automatic measurement of motor parameters and design internal parameters for motor control according to these measured parameters.
- Tuning results (measured parameters) can be output as a PDF report or as a header file in the Renesas control program.
- User can perform simple and basic moving test of the motor with measured parameters.

15.3 Window Structure

The window structure of Tuner tool is shown below.

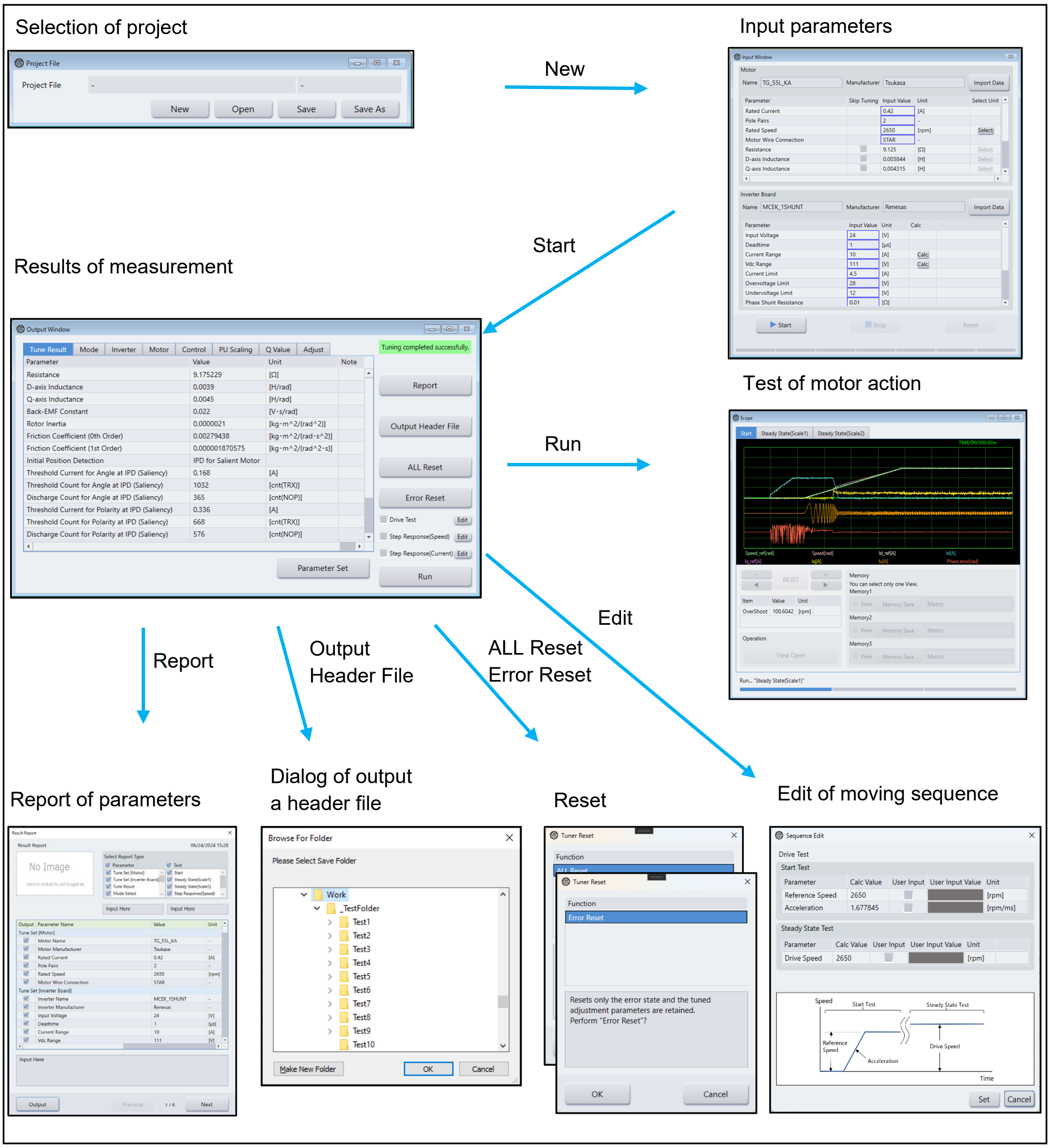


Figure 15-1 Tuner tool window

15.4 Operating procedure

15.4.1 Preparation

i. Download supported software

Please download the MOT file which supports “Tuner function” into the CPU.

・***_MCEK_LESS_FOC_APM_V***.mot

ii. Launch “Renesas Motor Workbench”

Click the short-cut icon on the desktop to start “Renesas Motor Workbench”.

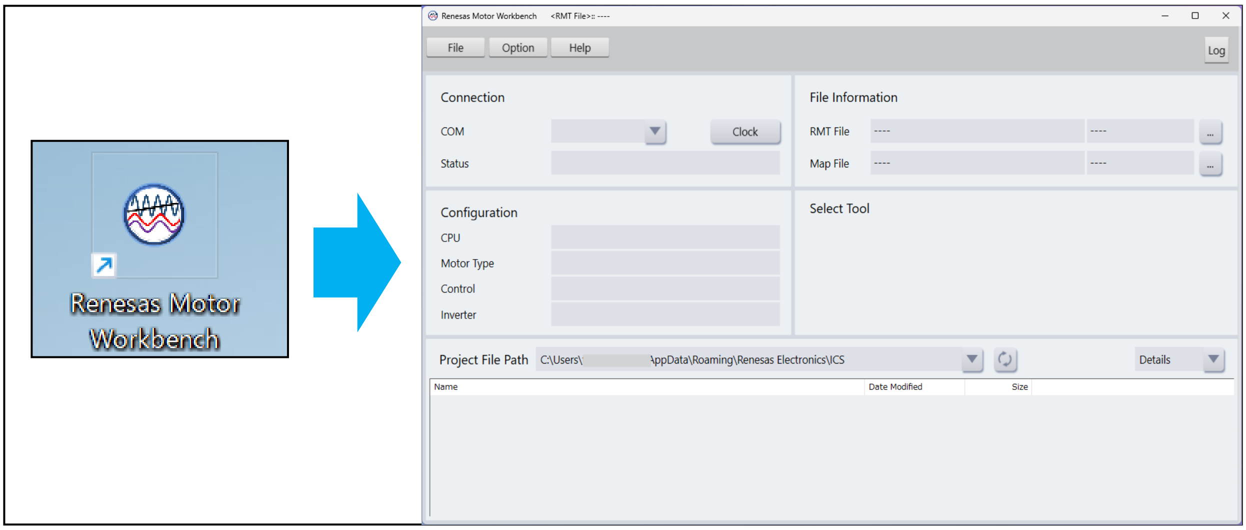


Figure 15-2 Start of “Renesas Motor Workbench”

iii. Read the RMT file

Read the RMT file supported “Tuner function”.

・***_MCEK_LESS_FOC_APM_V***.rmt

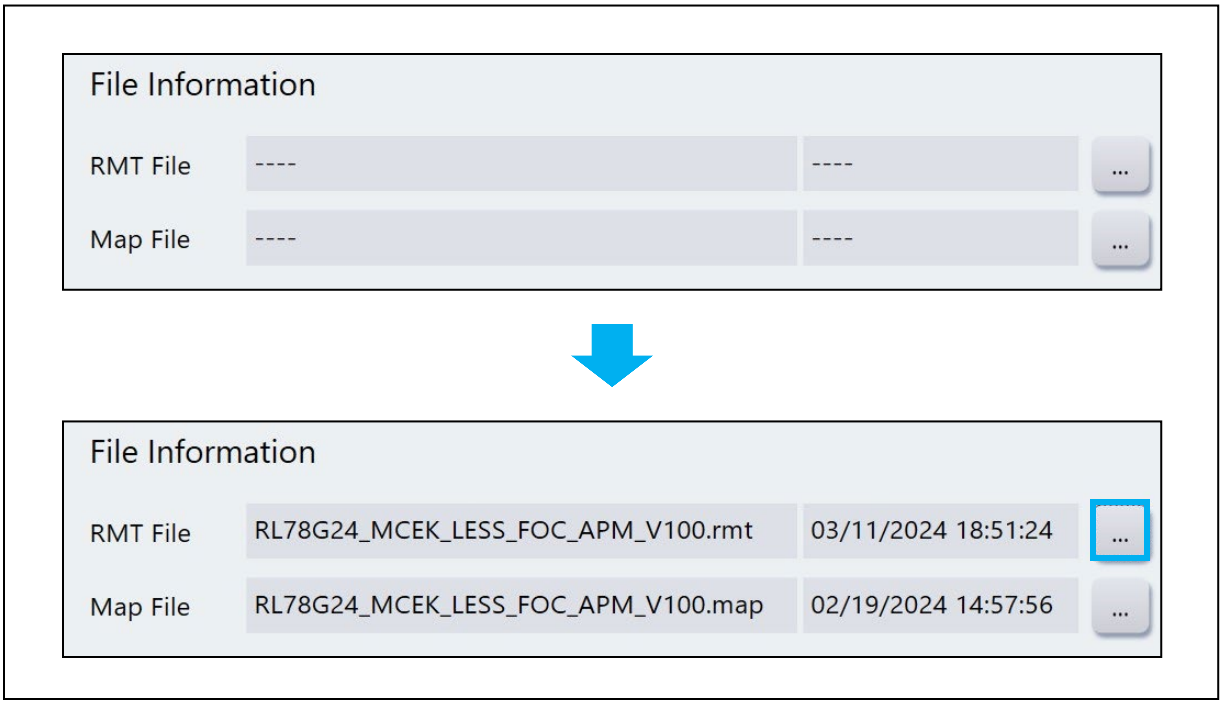


Figure 15-3 Read the RMT file

iv. Select valid connection

Select valid connection (COMx) from pull down menu. If the connection is established, the configuration data are displayed on “Configuration”.

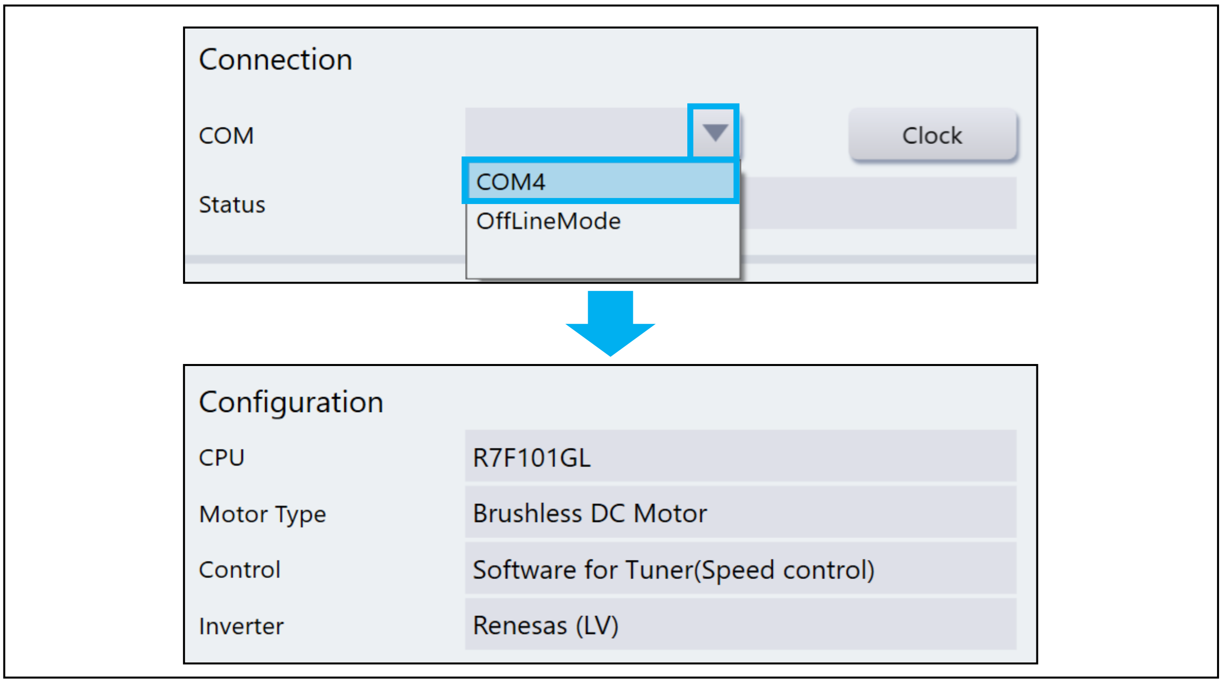


Figure 15-4 Select connection

15.4.2 Automatic measurement

15.4.2.1 Start Tuner function

Select “Tuner” icon and click. Then, the window changes to “Tuner for RL78” display.

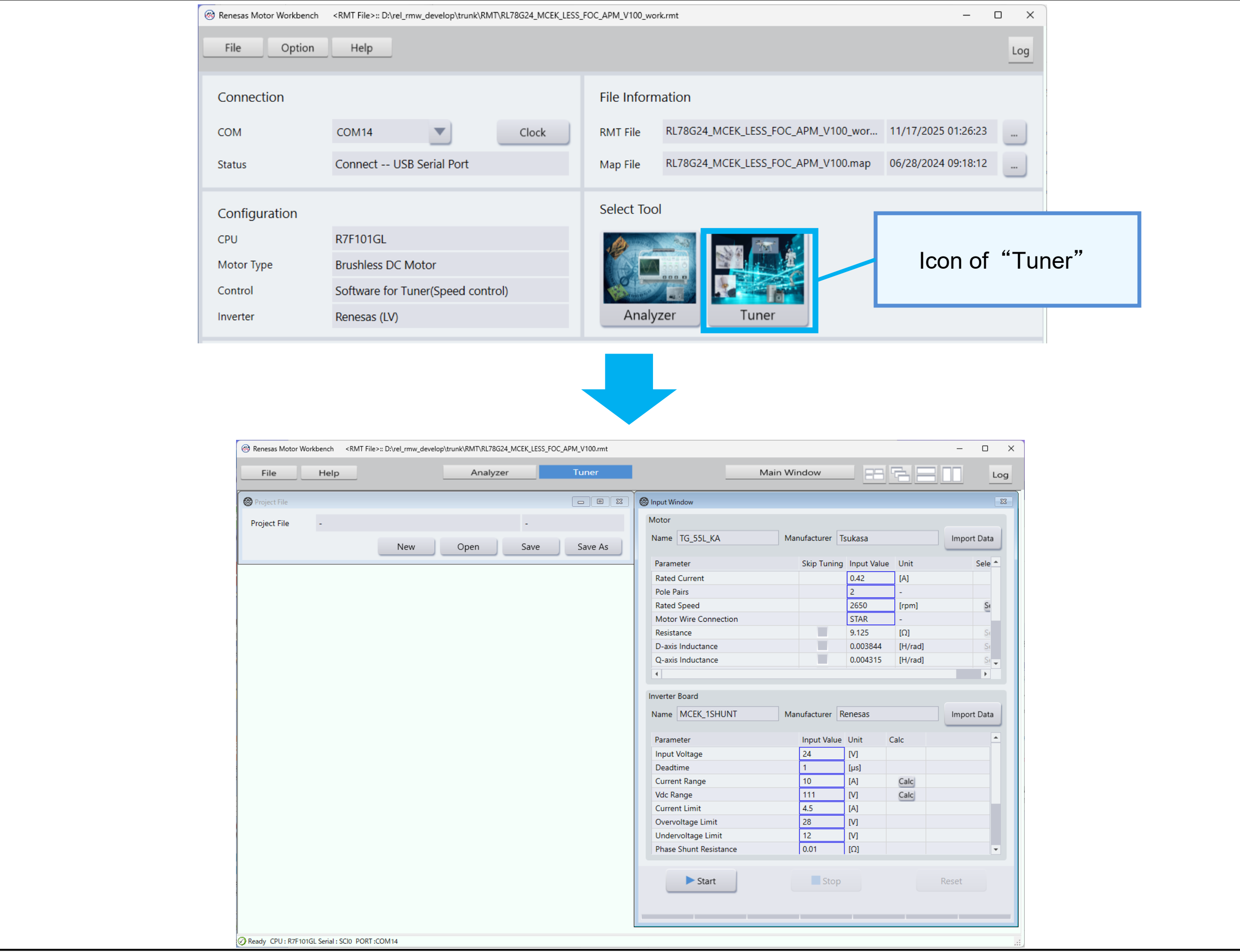


Figure 15-5 Start “Tuner function”

Input Window and Project File window are displayed. Project File window is always displayed. About the function, please refer to “15.5.1 Management of project files”.

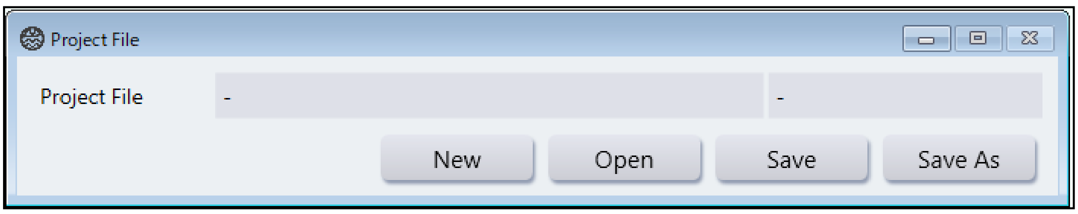


Figure 15-6 Project File window

15.4.2.2 Input parameters for automatic measurement

When you start "Tuner", the Input window will be displayed.

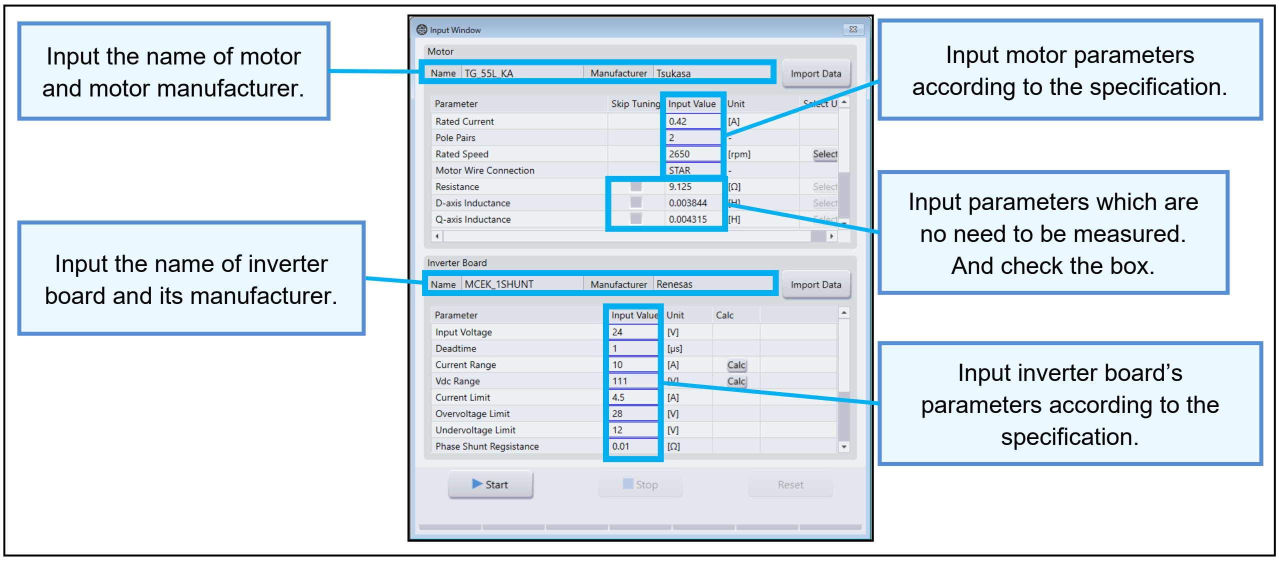


Figure 15-7 Input window

User can input the parameters which are uses to measurement. About input parameters, please refer to Table 15-1 List of required fields in "Specific Motor Parameters" and Table 15-2 List of required fields in "Specific Inverter Parameter" in detail.

Parameters for which "Skip Tuning" is not checked will be measured automatically.

If "Skip Tuning" is checked, the parameter is not measured automatically, and the value entered in "Input Value" is used.

User can use previously saved parameters by loading a project file by the "Import Data" button.

*In the current version, only default values are available for inverter parameters, and changing values is not supported.

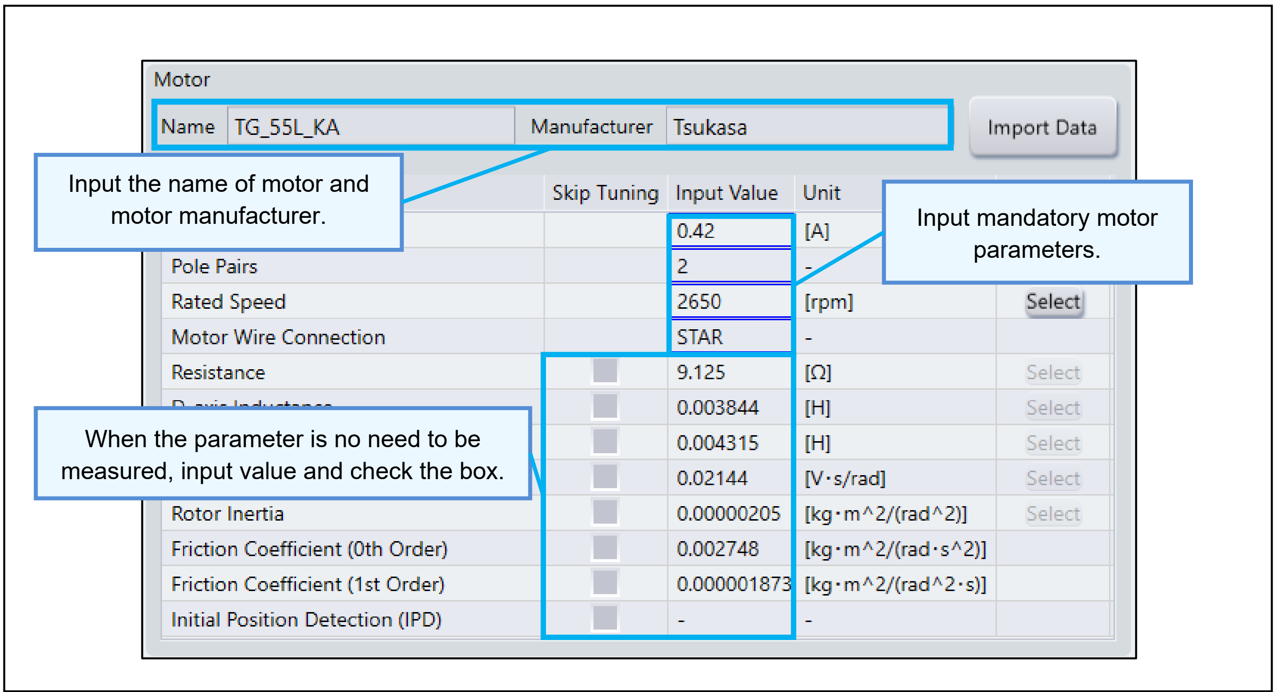


Figure 15-8 Mandatory required fields of "Specific Motor Parameters"

Table 15-1 List of required fields in "Specific Motor Parameters"

Item	Content	Required Input / Measurement
Motor Name	Motor name	Required Input
Motor Manufacturer	Manufacturer name	Required Input
Rated Current	Rated current [A]	Required Input
Pole Pairs	Pole pairs	Required Input
Rated Speed	Rated speed [rpm]	Required Input
Motor Wire Connection	Select from "STAR", "DELTA" and "Not Set"	Required Input
Resistance	Resistance [Ω]	Measurement
d-axis inductance	d-axis inductance [H/rad]	Measurement
q-axis inductance	q-axis inductance [H/rad]	Measurement
Back-EMF Constant	Induced voltage constant [V·s/rad]	Measurement
Rotor Inertia	Rotor inertia [kg·m^2/(rad^2)]	Measurement
Friction Coefficient (0th Order)	Friction coefficient (0th order) [kg·m^2/(rad·s^2)]	Measurement
Friction Coefficient (1st Order)	Friction coefficient (1st order) [kg·m^2/(rad^2·s)]	Measurement
Initial Position Detection (IPD)	Initial position detection	Measurement

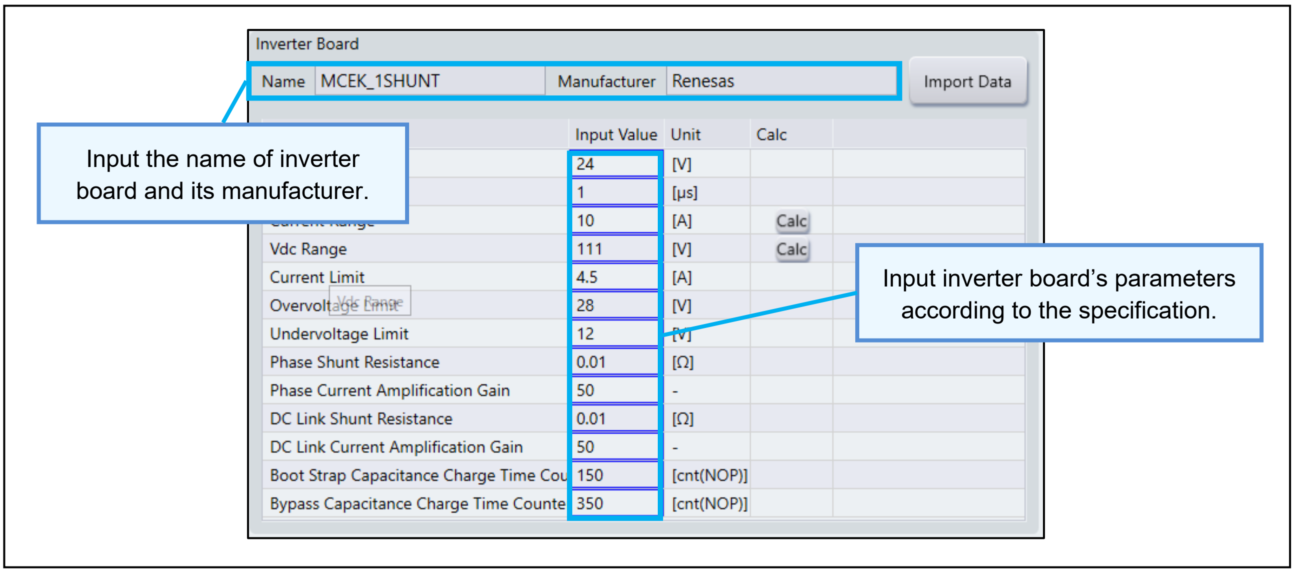


Figure 15-9 Required fields of “Specific Inverter Parameter”

Table 15-2 List of required fields in “Specific Inverter Parameter”

Item	Content
Inverter Name	Inverter board name
Inverter Manufacturer	Manufacturer name
Input Voltage	Main line voltage [V]
Deadtime	Deadtime [μsec]
Current Range	Maximum current of circuit [A]
Vdc Range	Maximum voltage able to detect by the circuit [V]
Current Limit	Limit of current [A]
Overvoltage Limit	Limit of main line voltage [V]
Undervoltage Limit	Limit of low voltage [V]
Phase Shunt Resistance	Resistance of shunt [Ω]
Phase Current Amplification Gain	Amplification gain of phase current
DC Link Shunt Resistance	Resistance of DC link shunt [Ω]
DC Link Current Amplification Gain	Amplification gain of DC link current
Boot Strap Capacitance Charge Time Counter	Counts to wait the charge of boot strap capacitance
Bypass Capacitance Charge Time Counter	Counts to wait the charge of bypass capacitance

15.4.2.3 Perform automatic measurement

Click “Start” button to start automatic measurement. The progress bar will be displayed as the process progress while automatic measurement. After the completion, Output Window will appear.

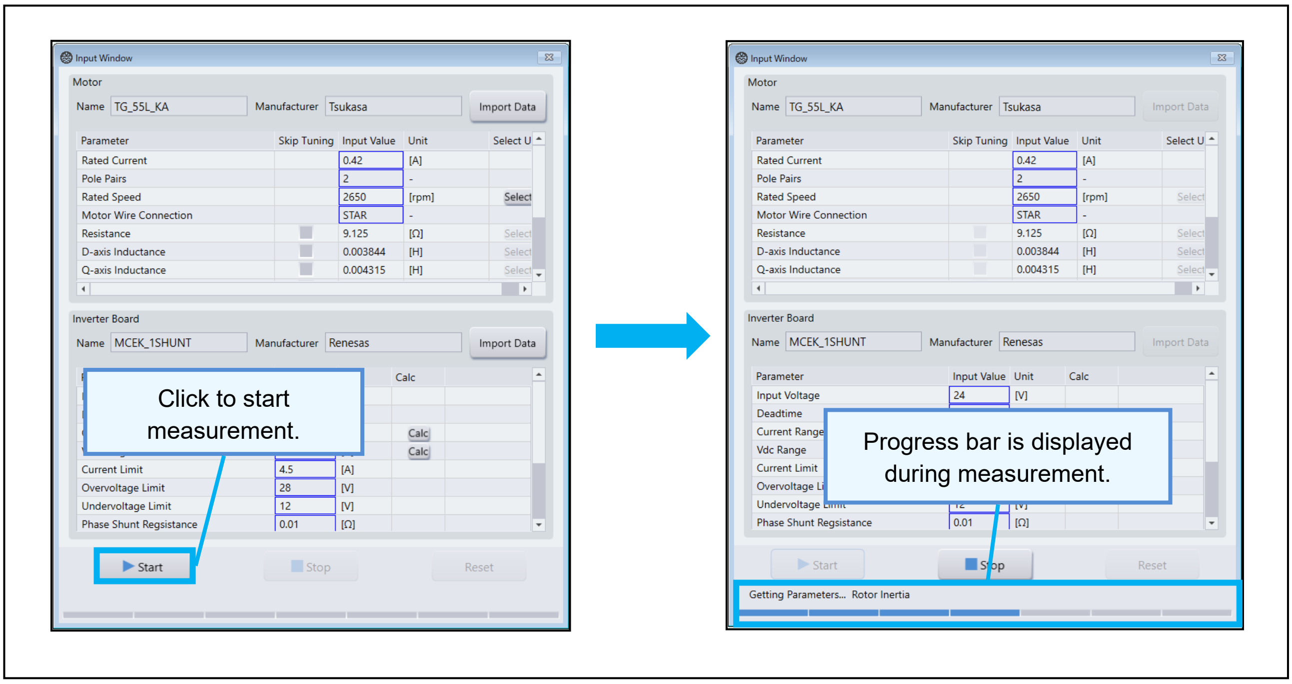


Figure 15-10 Automatic measurement

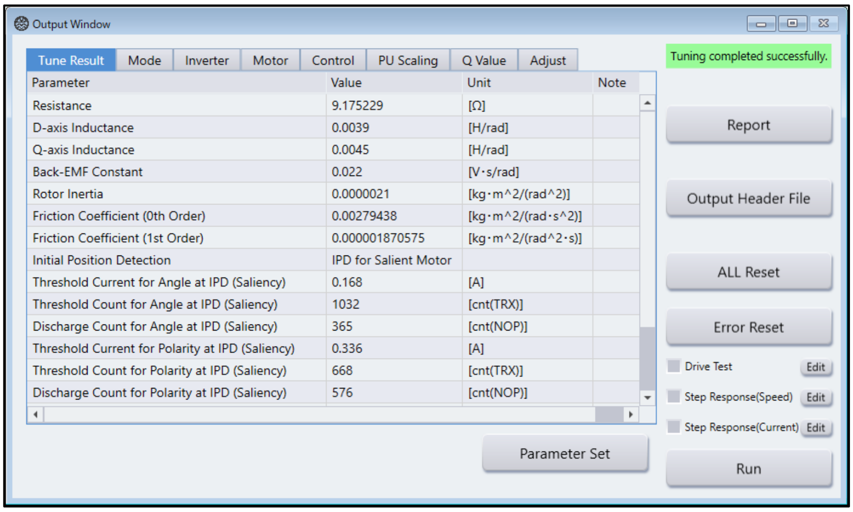


Figure 15-11 Output Window

If an error occurs during automatic measurement, an error window will be displayed. Then, check the error message, and click the “OK” button. When recovering from an error, please click “Reset” button.

If you want to stop the function during automatic measurement, click the “Stop” button.

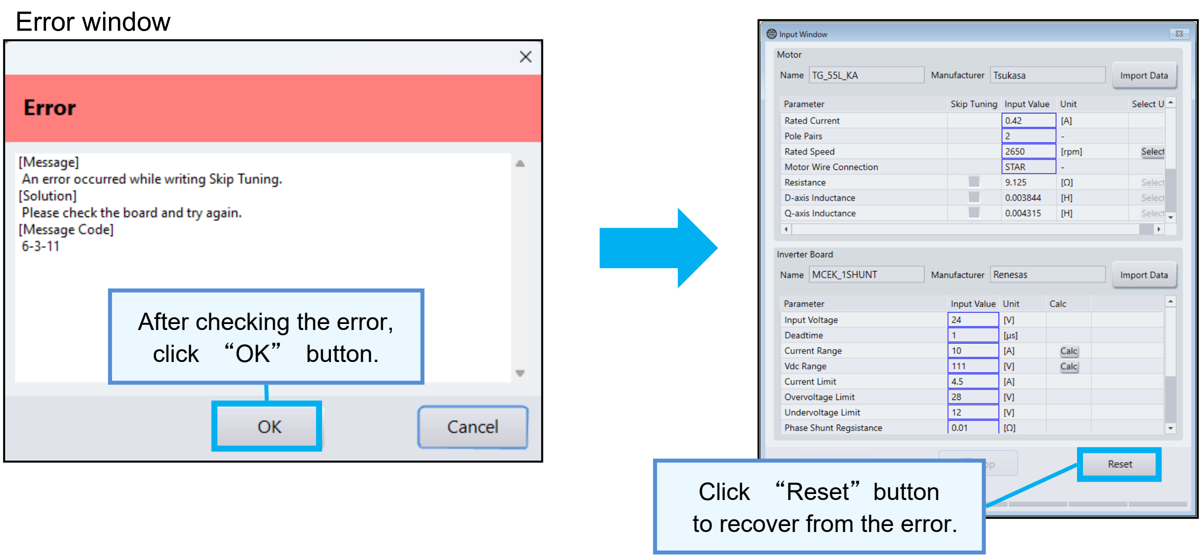


Figure 15-12 Error message (an example)

15.4.3 Design of parameters

15.4.3.1 Confirmation of automatic measurement results

The results of automatic measurement are displayed on Tune Result tab of Output Window.

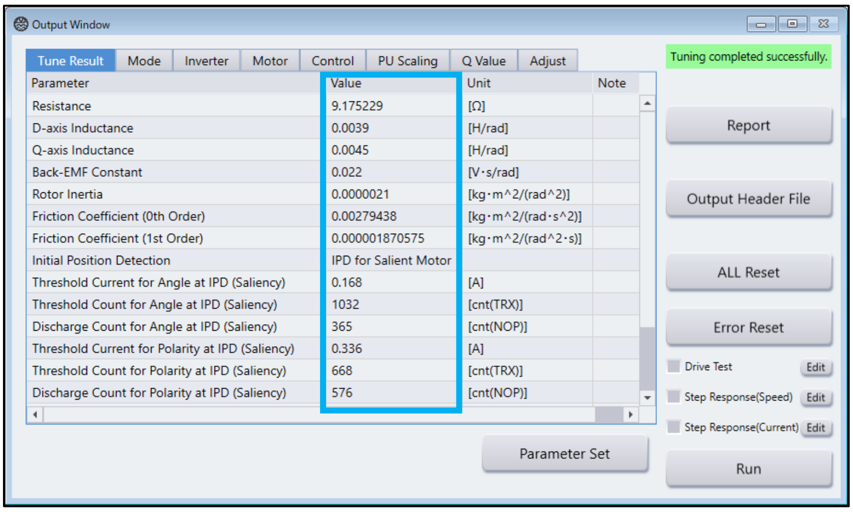


Figure 15-13 Tune Result tab

15.4.3.2 Change parameters

Values which are calculated from automatic measurement are stored in “Value” column of “Inverter”, “Motor”, “Control”, “PU Scaling”, “Q Value”, and “Adjust” tabs. The values of the parameters boxed in blue can be edited by user. By clicking the “Parameter Set” button, they can be reflected to the corresponding variables in the program written to the microcontroller. Some parameters are not reflected in the program, but they will be reflected in the output of the header file.

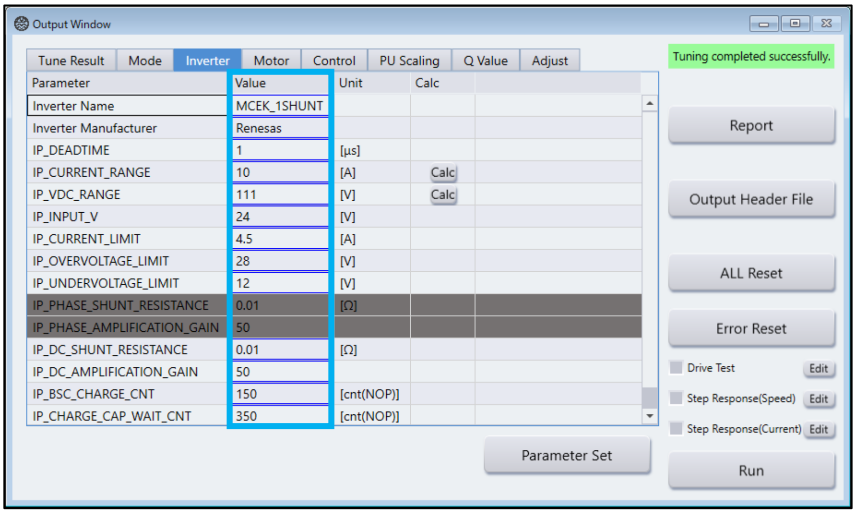


Figure 15-14 Inverter tab

15.4.3.3 Edit control mode

User can edit control mode in Mode tab. Control mode settings are not reflected to program by clicking the "Parameter Set" button. The set value will be output to the header file.

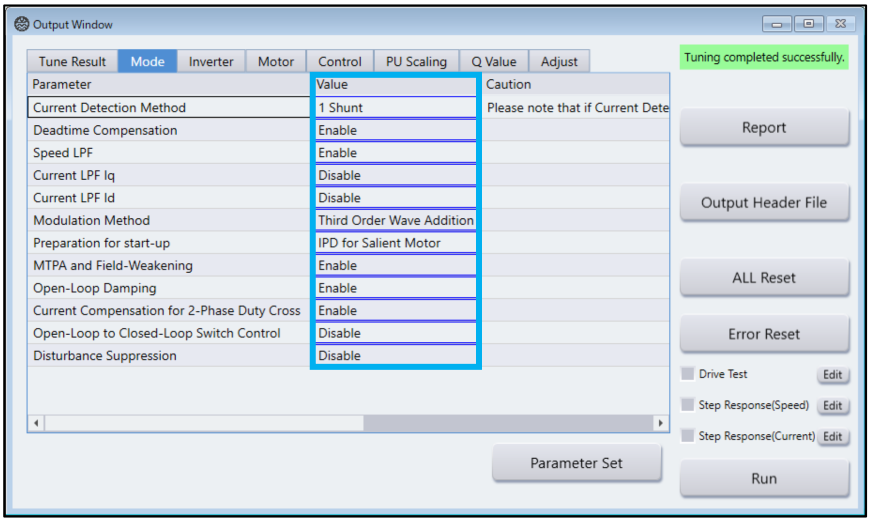


Figure 15-15 Mode tab

15.4.4 Drive Test

Drive Test includes “Startup and steady state drive”, “Step response (speed)”, and “Step response (current)”. User can select which test shall be performed and selected tests are performed continuously.

Wave form is displayed automatically with 15.4.4.1, 15.4.4.2, and 15.4.4.3. Sometimes wave form isn’t displayed completely correspond to connected motor. In that case, user can edit the settings of wave form display with “Scope Channel Info” window. The window can be displayed by clicking “View Open” button.

* The motor may not rotate properly during the Drive Test. If you feel that the motor is driving abnormally, immediately stop the motor by pressing the “Stop” button on the Output Window.

* If the starting time of the motor is long, the waveform of the starting test may not fit on the time scale, and subsequent driving tests may not be executed properly. Also, when a drive test is executed two or more times, the waveform will be displayed with the same vertical axis scale and offset as the first time.

15.4.4.1 Start up and steady state drive

User can confirm the wave form at start up and steady state by perform “Drive Test”. Check the “Drive Test” check box (1) on Output Window. After that, click the “Run” button (2) to display the Scope window.

* If user check the check box other than “Drive Test”, other tests are performed continuously.

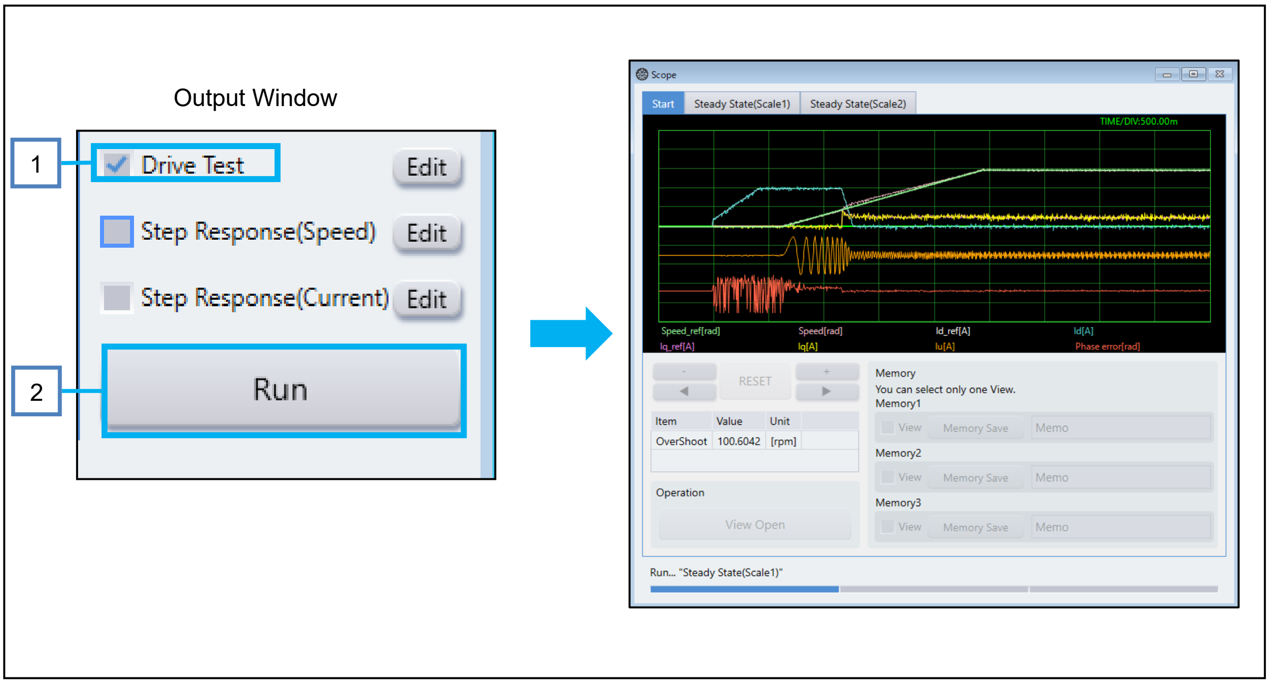


Figure 15-16 Drive Test (start up and steady state drive)

If user want to change test conditions, click “Edit” button. Then, Sequence Edit form window is displayed. Please input desired value and push “Set” to change.

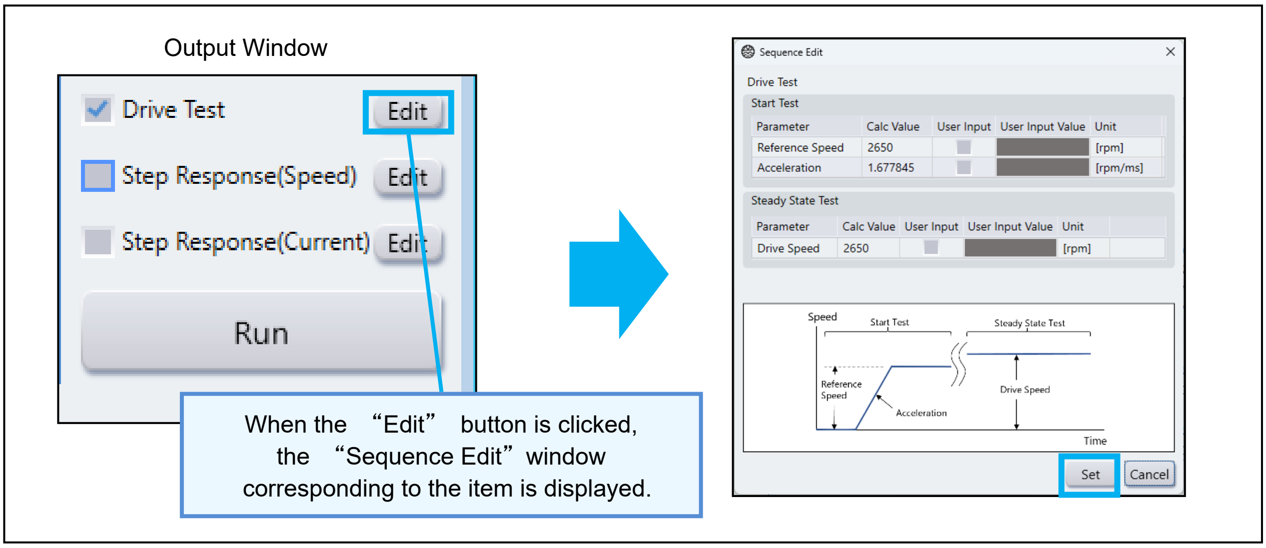


Figure 15-17 Sequence Edit form

15.4.4.2 Step Response (Speed)

User can confirm the step response(speed) by following action. Check the “Step Response(Speed)” check box (1) on Output Window. After that, click the “Run” button (2) to display the Scope window.

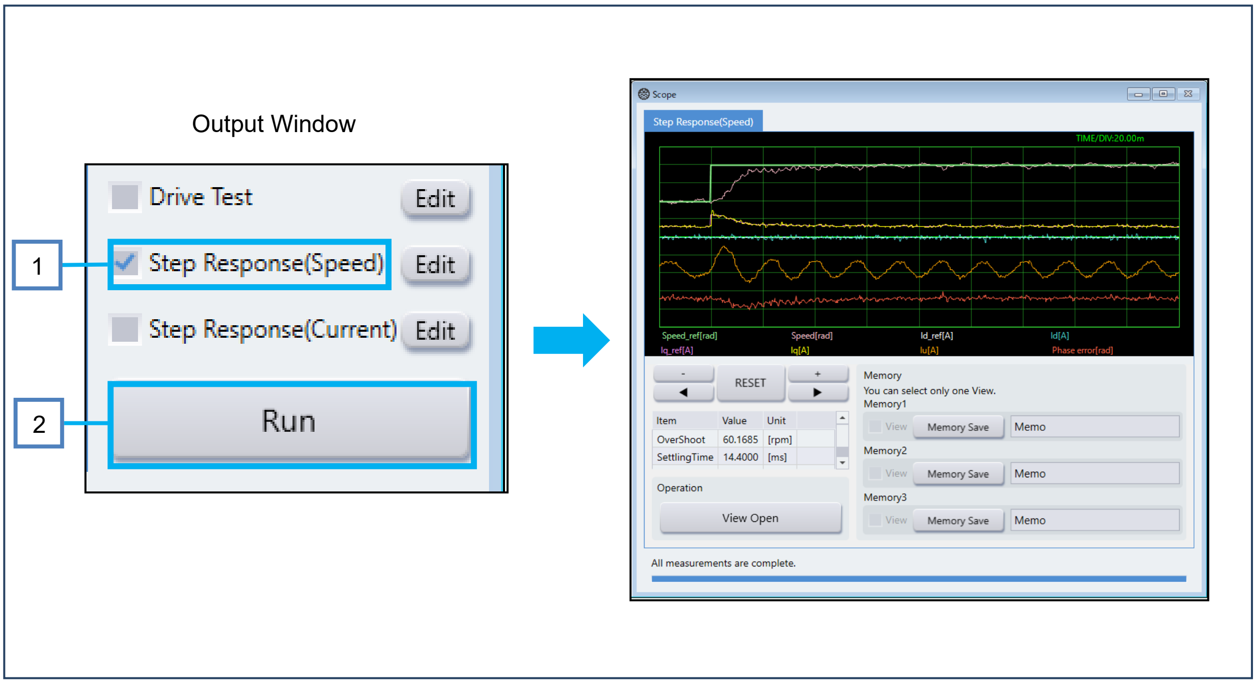


Figure 15-18 Step Response (Speed)

15.4.4.3 Step Response (Current)

User can confirm the step response(current) by following action. Check the “Step Response(Current)” check box (1) on Output Window. After that, click the “Run” button (2) to display the Scope window.

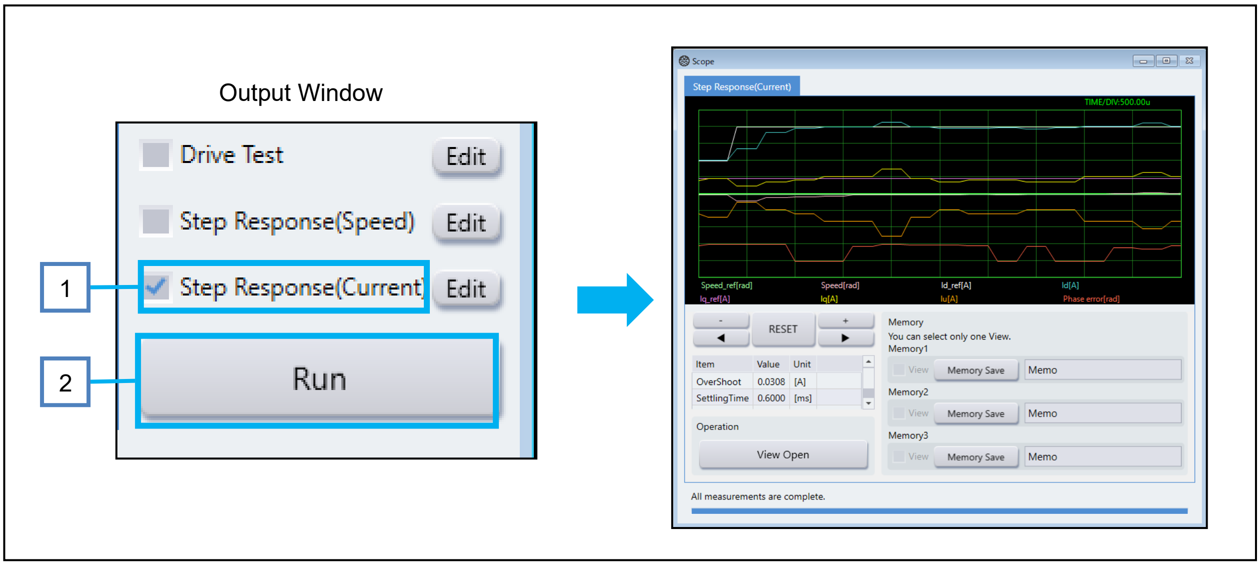


Figure 15-19 Step Response (Current)

15.4.5 Result Report

15.4.5.1 Output of report

User can take set parameters and Drive Test results as a report file. When “Report” button is pushed at each window, Result Report window appears. Push “Output” button to output the report file.

*Parameter values may not be properly reflected in the report file. Please check the Output Window for the correct parameter values.

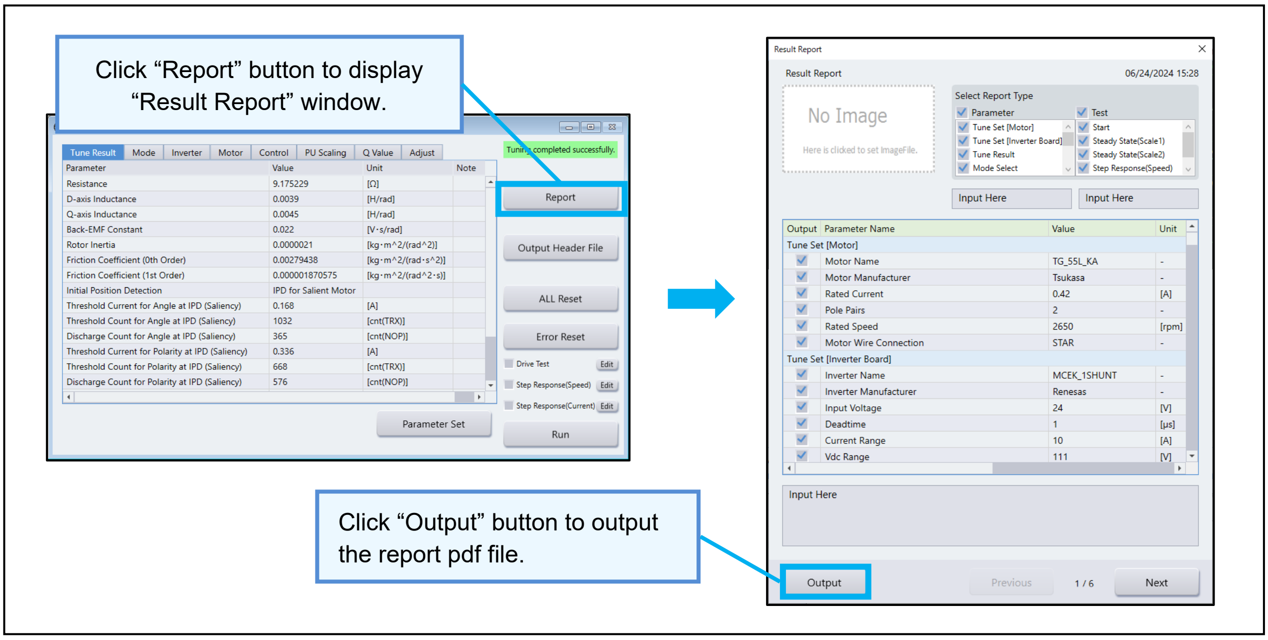


Figure 15-20 Result Report (parameters)

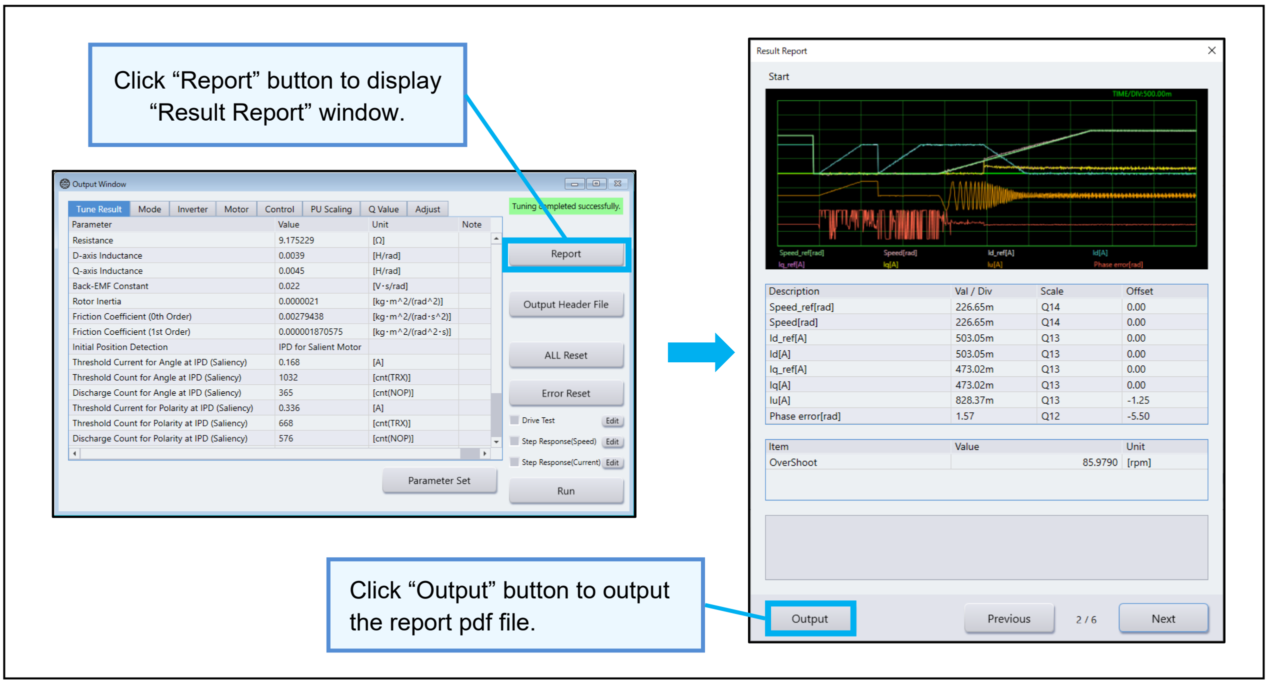


Figure 15-21 Result Report (Drive Test results)

15.4.5.2 Output Header File

Outputs set parameters as header files.

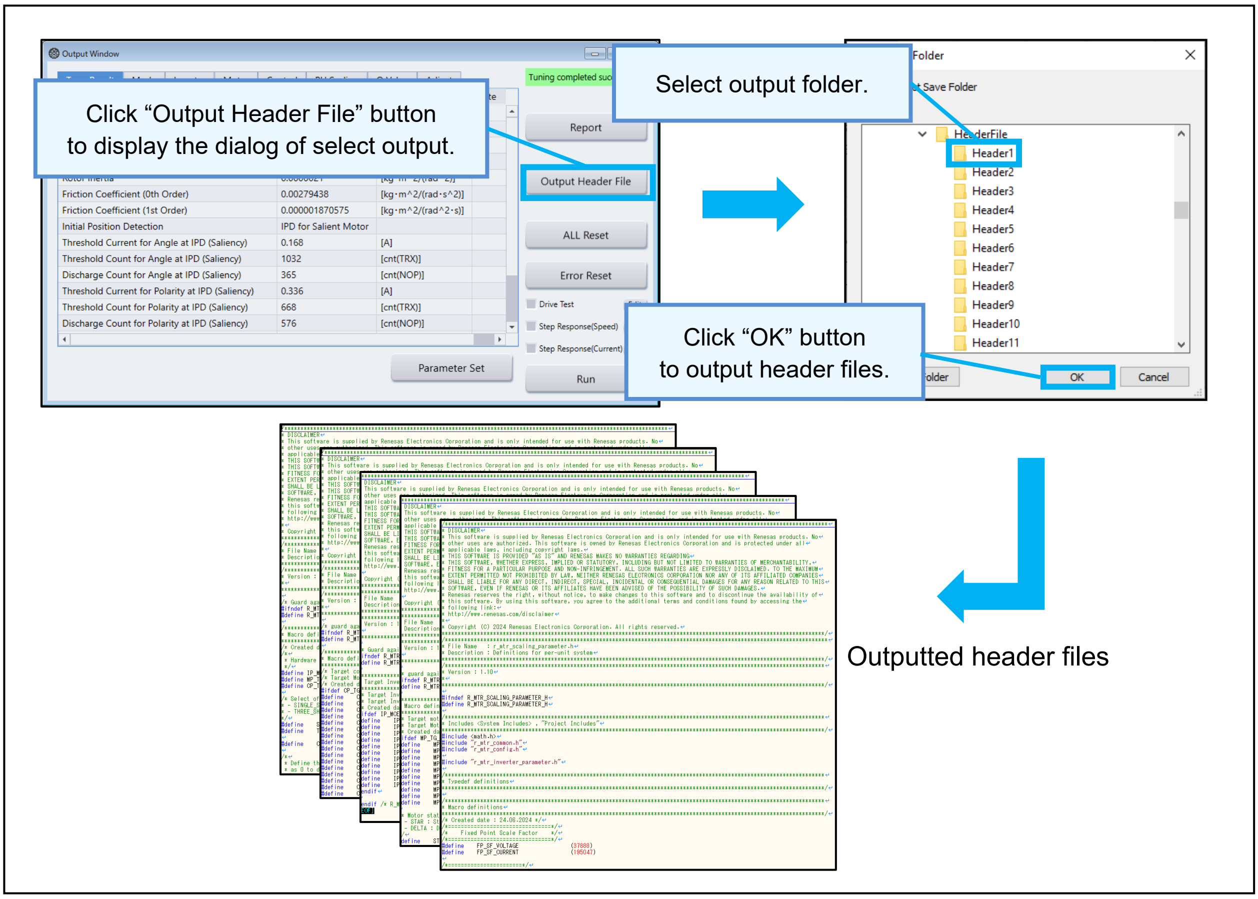


Figure 15-22 Output Header File

15.4.6 Finish of Tuner function

During motor movement as a test, please finish the test by “STOP” button.
After that, return back to the “Main Window” by “Main Window” button or perform the procedure to change the tool.

15.5 Description of function

15.5.1 Management of project files (Project File Selection Window)

In “Project File” window, user can save and load a project file (extension with “apmpj). The project file holds parameters which are displayed in “Input” and “Output” window and measurement results. The project file selection window “Project File” is displayed when “Tuner” is selected from “Select Tool” in the Main Window and is always displayed during function execution.

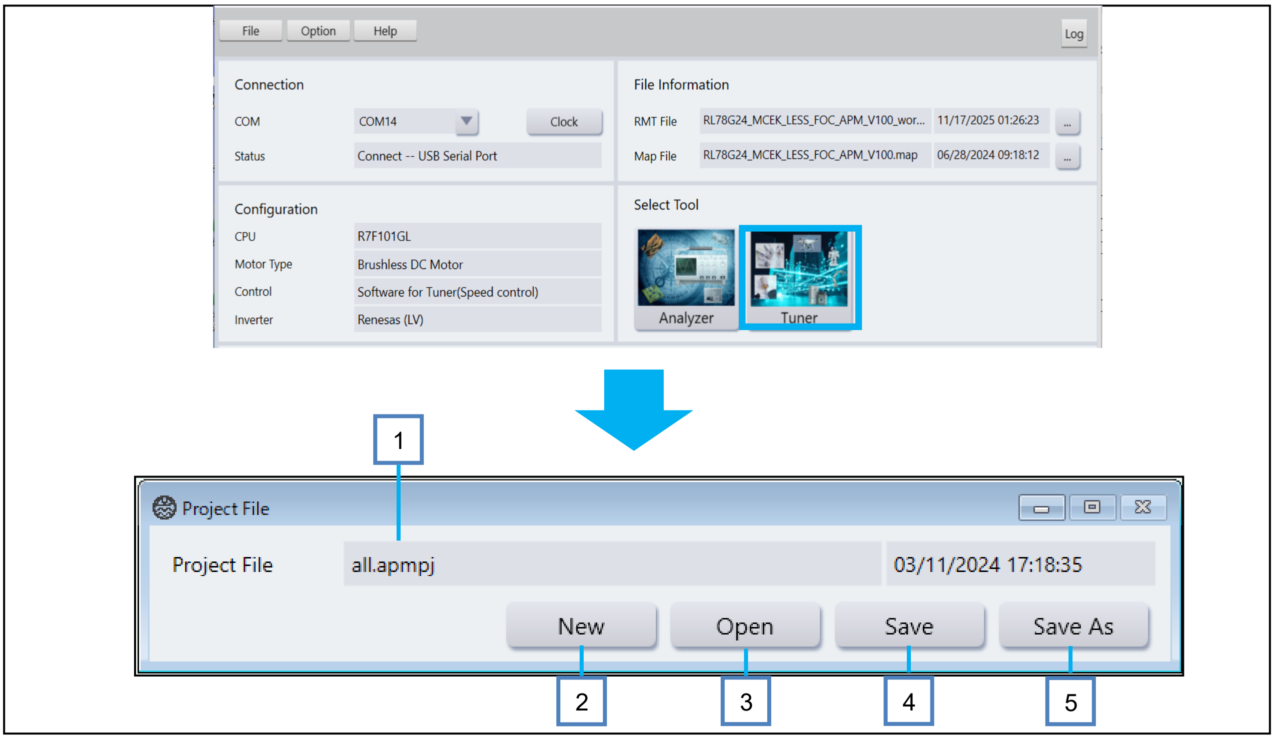


Figure 15-23 Project File selection window

Table 15-3 List of items on the Project File selection window

No	Item	Description
(i)	Project File	Displays the name of the current selected project file and the modification date and time.
(ii)	New	Creates a new project file.
(iii)	Open	Loads a project file.
(iv)	Save	Overwrites the selected project file and saves it.
(v)	Save As	Saves the project file as a new file.

15.5.2 Parameter input function (Input Window)

Input window is displayed when “Tuner” is selected from “Select Tool” on “Main Window”. In this window, user can enter the motor and inverter board parameters. When “Start” button is pressed, automatic measurement is started with entered parameters. The areas boxed in blue can be edited by user.

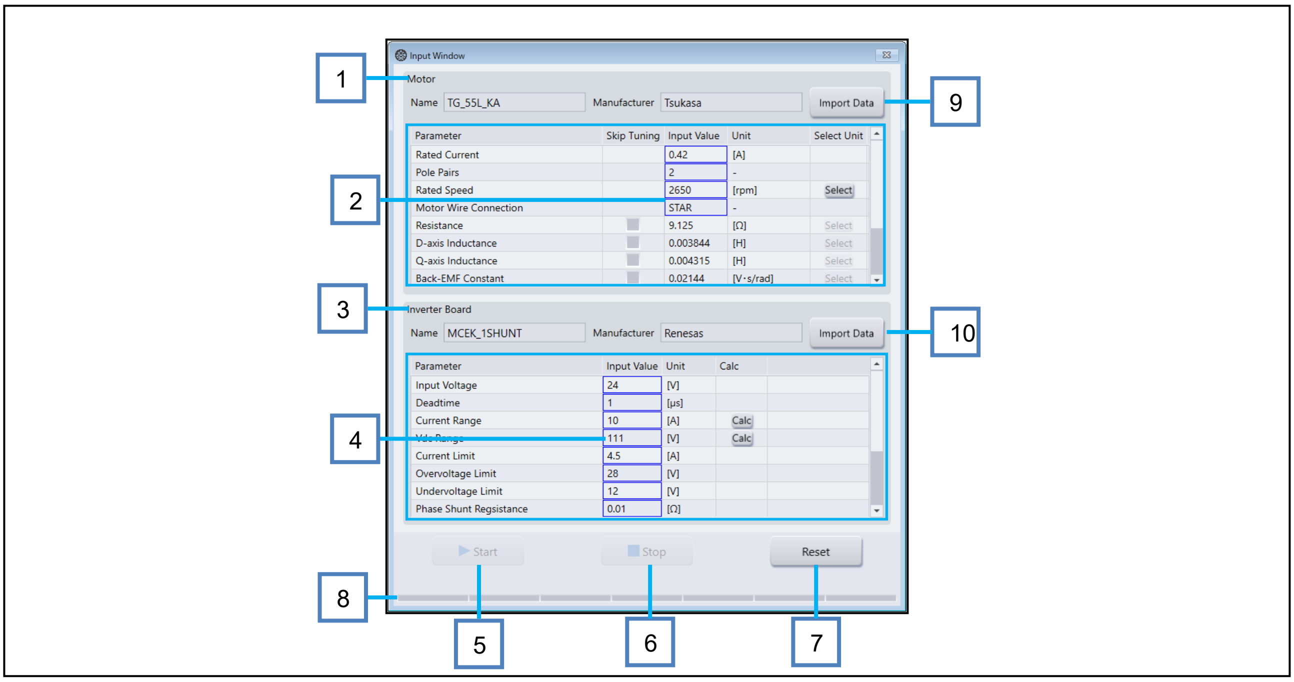


Figure 15-24 Input window

*In the current version, only default values are available for inverter parameters, and changing values is not supported.

Table 15-4 List of items on Input window

<table>
<tr><th>No</th><th>Item</th><th>Description</th></tr>
<tr><td>(i)</td><td>Motor Information</td><td>Enter and display motor information.
<table>
<tr><th>Item</th><th>Description</th><th>Note</th></tr>
<tr><td>Motor Name</td><td>Motor Name</td><td>-</td></tr>
<tr><td>Motor Manufacturer</td><td>Motor Manufacturer Name</td><td>-</td></tr>
</table></td></tr>
<tr><td>(ii)</td><td>Motor Input Parameters</td><td>Please input mandatory motor parameters for automatic measurement.
<table>
<tr><th>Item</th><th>Description</th><th>Note</th></tr>
<tr><td>Parameter</td><td>Parameter name</td><td>-</td></tr>
<tr><td>Skip Tuning</td><td>Check this box if you do not want to measure the target parameters automatically.</td><td>-</td></tr>
<tr><td>Input Value</td><td>Enter the setting value when automatic measurement is not used.</td><td>If the set value exceeds the upper or lower limit values, a red frame will appear as an invalid value.</td></tr>
<tr><td>Unit</td><td>Unit name</td><td>-</td></tr>
<tr><td>Select Unit button</td><td>Pressing the “Select” button to display the unit conversion window.</td><td></td></tr>
</table></td></tr>
<tr><td>(iii)</td><td>Inverter Information</td><td>Enter and display inverter information.
<table>
<tr><th>Item</th><th>Description</th><th>Note</th></tr>
<tr><td>Inverter Name</td><td>Inverter Name</td><td>-</td></tr>
<tr><td>Inverter Manufacturer</td><td>Inverter Manufacturer Name</td><td>-</td></tr>
</table></td></tr>
<tr><td>(iv)</td><td>Inverter Input Parameters</td><td>Please input mandatory inverter board’s parameters for automatic measurement.
<table>
<tr><th>Item</th><th>Description</th><th>Note</th></tr>
<tr><td>Parameter</td><td>Parameter name</td><td>-</td></tr>
<tr><td>Input Value</td><td>Input parameter value</td><td>If the set value exceeds the upper or lower limit values, a red frame will appear as an invalid value.</td></tr>
<tr><td>Unit</td><td>Unit name</td><td>-</td></tr>
<tr><td>Calc button</td><td>Pressing the “Calc” button to display the calculation window.</td><td></td></tr>
</table></td></tr>
<tr><td>(v)</td><td>Start button</td><td>Start Automatic Measurement.</td></tr>
<tr><td>(vi)</td><td>Stop button</td><td>Stop Automatic Measurement.</td></tr>
<tr><td>(vii)</td><td>Reset button</td><td>Pressing the button to send a Reset command to the CPU board.
Obtains the status from the CPU board and displays it in case of an error.</td></tr>
<tr><td>(viii)</td><td>Progress bar</td><td>Progress is displayed during automatic measurement on CPU board.</td></tr>
<tr><td>(ix)</td><td>Import Data(Motor)</td><td>Loads the motor information in the saved project file and reflects it in the set values of the motor input parameters.</td></tr>
<tr><td>(x)</td><td>Import Data(Inverter)</td><td>Loads the inverter board information in the saved project file and reflects it in the set values of the inverter board input parameters.</td></tr>
</table>

15.5.2.1 Loading saved parameters

Click the “Import Data” button to display the file selection dialog and select a saved project file.

When a file is selected in “Import Data” of “Motor”, only parameters in “Motor” are overwritten to the settings in the project file. When a file is selected in “Import Data” of “Inverter Board”, only parameters in “Inverter” are overwritten to the settings in the project file.

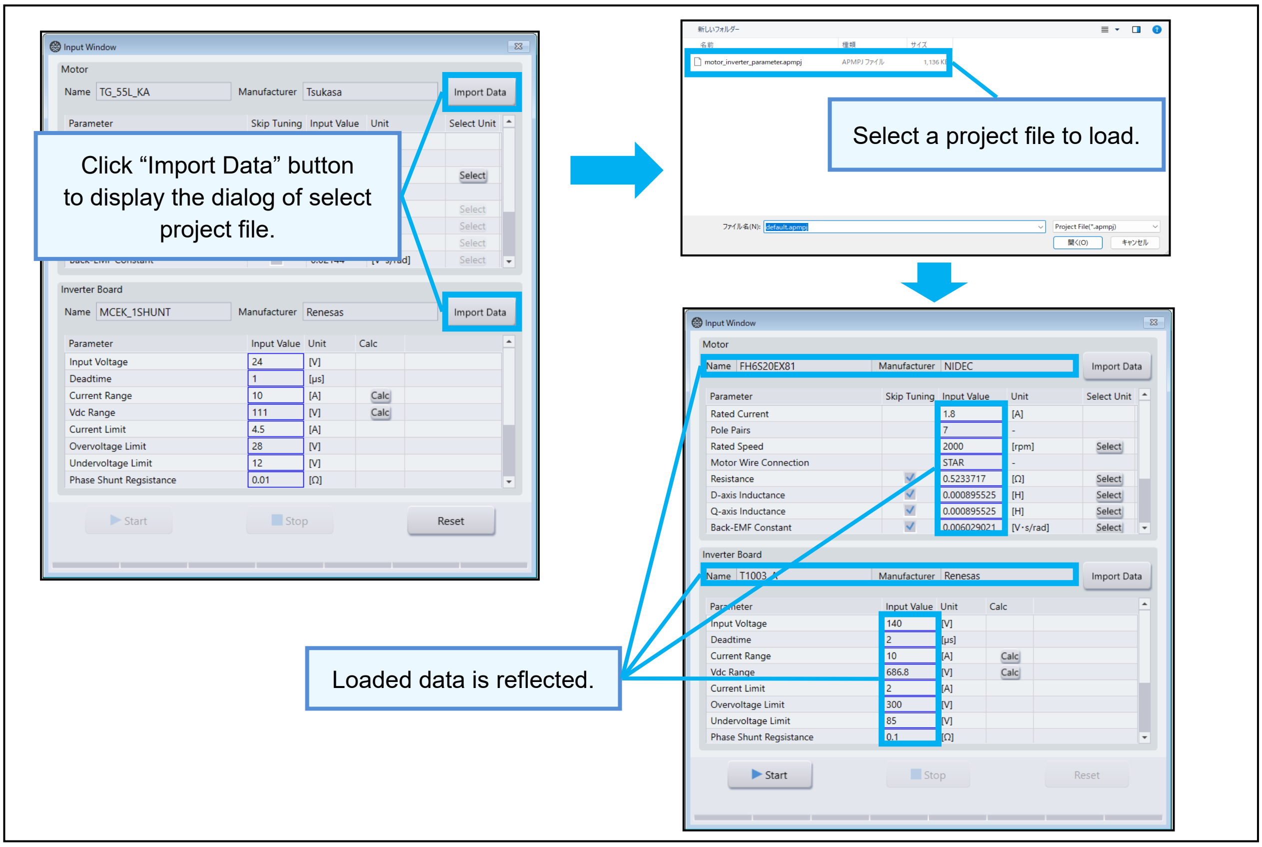


Figure 15-25 Load saved parameters

15.5.2.2 Unit conversion function

“Specific Motor Parameters” can be converted from another unit system to the units which is used in Renesas control program.

On the Unit Conversion window, enter the converted values as parameter values on the Motor tab. The areas boxed in blue can be entered by user. Pressing the "Set" button to reflect the values on each form.

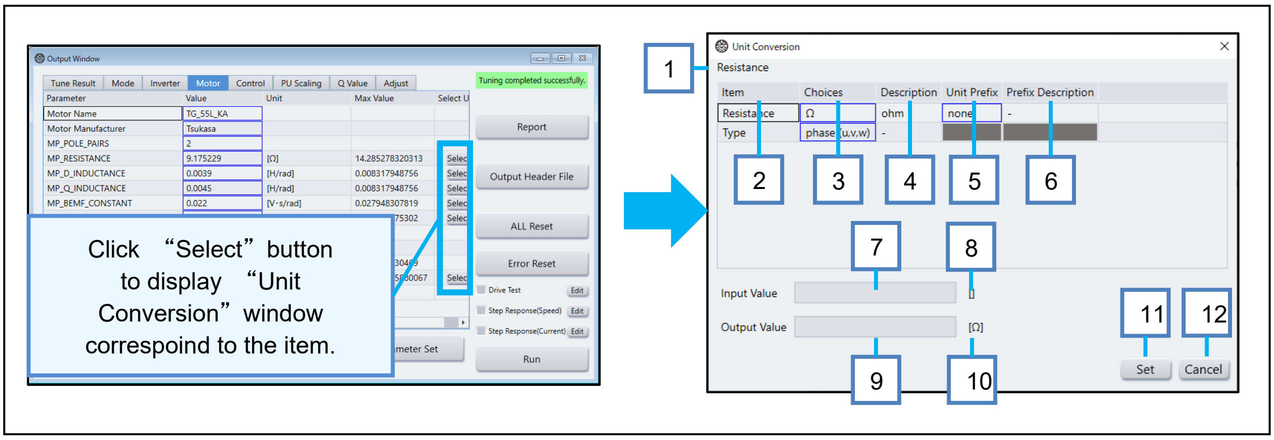


Figure 15-26 Unit Conversion form

Table 15-5 List of items on Unit Conversion form

No	Item	Description	Note
(i)	Input Parameter Name	Displays input parameter names.	-
(ii)	Item	Displays items.	-
(iii)	Choices	Select the unit.	-
(iv)	Description	Displays a description of the selected item.	-
(v)	Unit Prefix	Select the auxiliary unit.	-
(vi)	Prefix Description	Displays a description of the selected auxiliary unit.	-
(vii)	Input Value	Enter the value before unit conversion.	If the value is less than 0, a red frame will appear as an invalid value. *Comma input not allowed.
(viii)	Unit of input Value	Displays the units of the input value.	
(ix)	Output Value	Displays the numerical value after unit conversion.	
(x)	Unit of output Value	Displays the units of the output value.	
(xi)	Set button	Reflect the unit conversion values on each form and closes the unit conversion form.	
(xii)	Cancel button	Closes the unit conversion form.	

15.5.2.3 Scaling value calculation function

The scaling value to current "IP_CURRENT_RANGE" and the scaling value to voltage "IP_VDC_RANGE" can be designed from the detection circuit information on the inverter. (Please refer to below figures.)

(1) Current scaling value calculation function

In Current Range Calculation window, the scaling value is calculated from three items: shunt resistance “Rs”, amplification factor “G”, and voltage reference value, using the following formula (1). Pressing the "Set" button to reflect the values on each form.

Current scaling value [A] = voltage reference value [V] / (shunt resistance “Rs” * amplification factor “G”) ···(1)

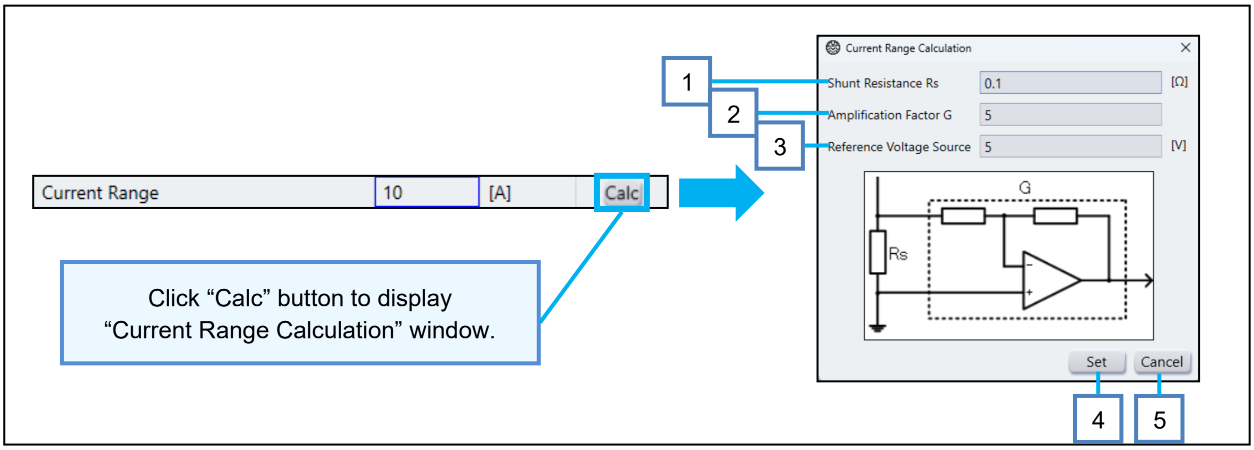


Figure 15-27 Current Range Calculation window

Table 15-6 List of items on Current Range Calculation window

No	Item	Description	Note
1	Shunt Resistance Rs	Please input shunt resistance value of the target inverter.	If the value is 0 or less, a red frame will appear as an invalid value.
2	Amplification Factor G	Please input the current detection amplifier amplification factor.	If the value is 0 or less, a red frame will appear as an invalid value.
3	Reference Voltage Source	Input the reference voltage for A/D detection of MCU.	If the value is 0 or less, a red frame will appear as an invalid value.
4	Set button	Reflect the calculated values on each form and closes the current calculation form.	-
5	Cancel button	Closes the current calculation form.	-

(2) Voltage scaling value calculation function

In Voltage Range Calculation window, the scaling value is calculated from the three items: the voltage divider resistors “R1”, “R2”, and the voltage reference value, using the following formula (2). Pressing the "Set" button to reflect the values on each form.

Voltage scaling value [V] = voltage reference value [V] * (R1+R2) / R2 ···(2)

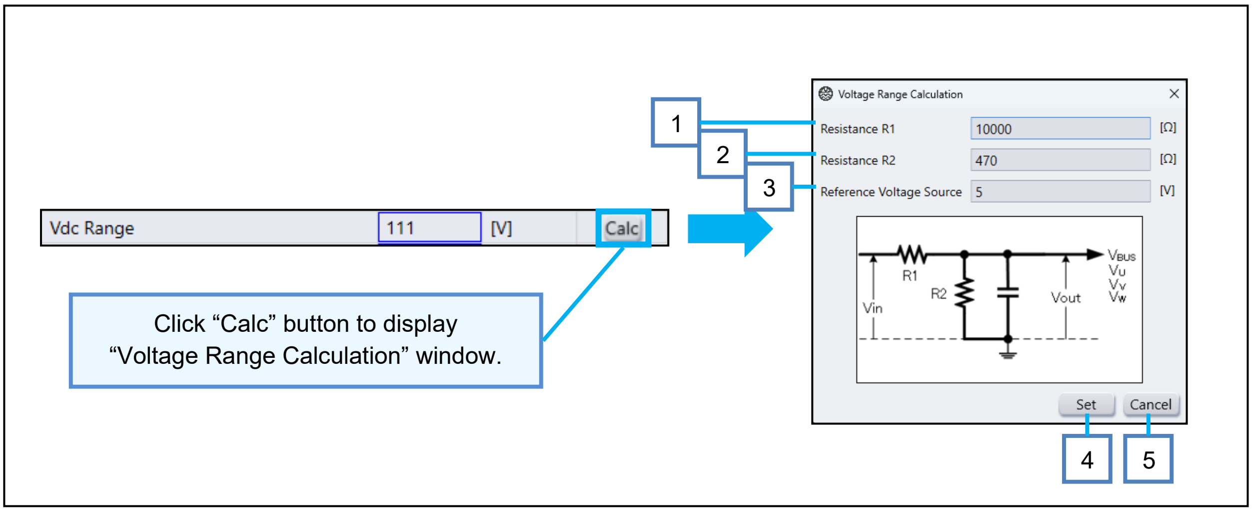


Figure 15-28 Voltage Range Calculation window

Table 15-7 List of items on Voltage Range Calculation window

No	Item	Description	Note
1	Resistance R1	Please input the resistance value for separation of target inverter.	If the value is 0 or less, a red frame will appear as an invalid value.
2	Resistance R2	Please input the resistance value for separation of target inverter.	If the value is 0 or less, a red frame will appear as an invalid value.
3	Reference Voltage Source	Input the reference voltage for A/D detection of MCU.	If the value is 0 or less, a red frame will appear as an invalid value.
4	Set button	Reflect the calculated values on each form and closes the voltage calculation form.	-
5	Cancel button	Closes the voltage calculation form.	-

* If the voltage or current detection circuit of the inverter board is different from the above, set the scaling values according to the target inverter circuit.

15.5.3 Measurement result display function (Output Window)

The results of automatic measurement are displayed and can be edit each parameter. Also, the Drive Test can be started.

When Output Window is displayed, the following parameters are displayed in each tab: “Tune Result” (measurement result), “Mode” (selection of control mode), “Inverter” (specific inverter parameters), “Motor” (specific motor parameters), “Control” (control parameters), “PU Scaling” (scaling value), “Q Value”, “Adjust” (control parameter adjustment value), and these parameters can be edited. Editable parameters are displayed with blue box. And valid or invalid of each TAB’s parameters are changed according to the selection of “Motor” parameters. Invalid parameters are grayed out.

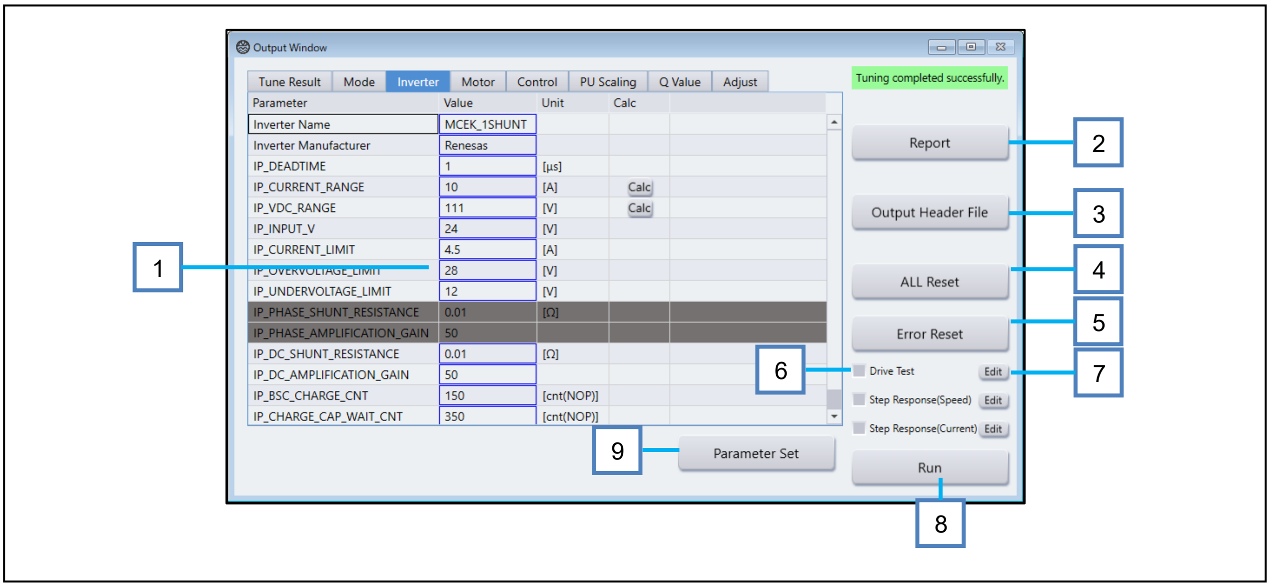


Figure 15-29 Output Window

Table 15-8 List of Items on Output Window

<table>
<tr><th>No</th><th>Item</th><th>Description</th></tr>
<tr><td>1</td><td>parameter</td><td>Various parameters are displayed on the “Tune Result”, “Mode”, “Inverter”, “Motor”, “Control”, “PU Scaling”, “Q Value”, and Adjust tabs to allow parameter editing.
※Editable fields are boxed in blue.</td></tr>
<tr><td>2</td><td>Report button</td><td>Displays the report window and outputs set parameters and Drive Test results.</td></tr>
<tr><td>3</td><td>Output Header File button</td><td>Outputs set parameters as a header file. Display the folder selection dialog and save the header file with extension ".h".</td></tr>
<tr><td>4</td><td>ALL Reset button</td><td>Resets the measurement data of the Tuner function.</td></tr>
<tr><td>5</td><td>Error Reset button</td><td>Resets the error of the Tuner function.</td></tr>
<tr><td>6</td><td>Drive mode</td><td>Select the drive mode in which the Drive Test is to be performed.
The drive mode shall be as follows.
<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>Drive Test</td><td>Startup and Steady-state drive</td></tr>
<tr><td>Step Response(Speed)</td><td>Step Response(Speed)</td></tr>
<tr><td>Step Response(Current)</td><td>Step Response(Current)</td></tr>
</table></td></tr>
<tr><td>7</td><td>Edit button</td><td>Display the “Edit Sequence” window to edit the sequence to be used in the Drive Test. The editable items shall be as follows.
<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>Drive Test</td><td>Target speed, Acceleration, Drive speed</td></tr>
<tr><td>Step Response(Speed)</td><td>Speed before change, Speed after change</td></tr>
<tr><td>Step Response(Current)</td><td>Current before change, Current after change</td></tr>
</table>
For details on the sequence edit function, see “15.5.4 Sequence editing functions”</td></tr>
<tr><td>8</td><td>Drive Test execution/stop button</td><td>When the button is pushed with display “Run”, Drive Test start and Scope window appears.
The display is changed to “Stop”.

When the button is pushed with display “Stop”, Drive Test stops.
The display is changed to “Run”.</td></tr>
<tr><td>9</td><td>Parameter Set button</td><td>Sends the parameters adjusted in each tab to the CPU board.</td></tr>
</table>

15.5.3.1 Tune Result TAB

Tune Result tab displays the automatic measured parameters, and the measurement results used in the Input window. Parameters are not editable, because this tab displays measurement results.

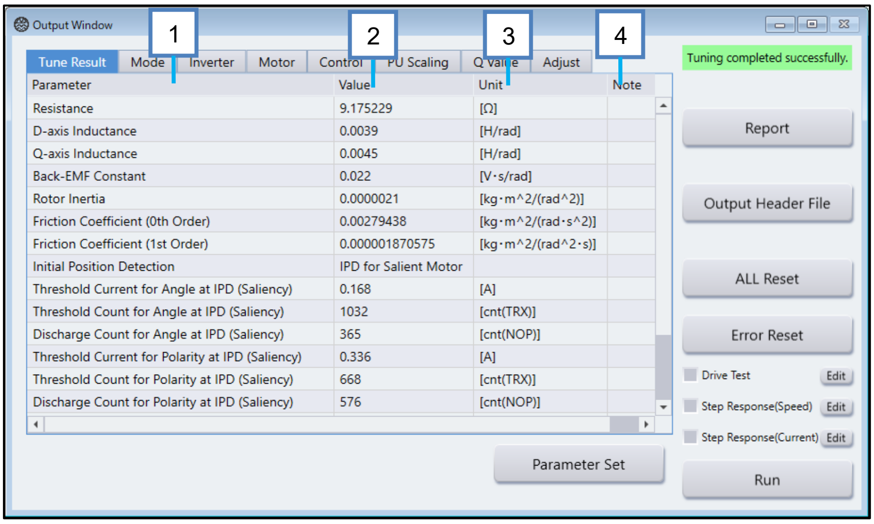


Figure 15-30 Tune Result tab

Table 15-9 List of Items on Tune Result tab

No	Item	Description
1	Parameter	Displays parameter names.
2	Value	Displays parameter values.
3	Unit	Displays the units of each parameter.
4	Note	Displays remarks. Manual Input: Indicates that automatic measurement is skipped or not. Calculated: Indicates that the parameters were designed by calculation.

15.5.3.2 Mode TAB

Mode tab is used to select various control modes. These settings are reflected in the "r_mtr_config.h" output via "Output Header File" button.

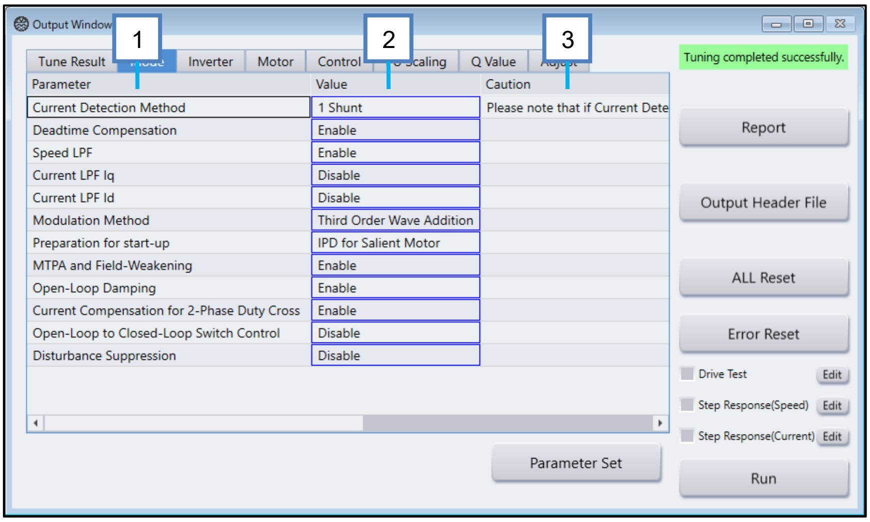


Figure 15-31 Mode tab

Table 15-10 List of items on Mode tab

No	Item	Description
1	Parameter	Displays parameter names.
2	Value	Selected values.
3	Caution	Displays caution.

15.5.3.3 Inverter TAB

Inverter tab is used to set specific parameters related to target inverter board. These settings are reflected in the “r_mtr_inverter_parameter.h” file, which is output via “Output Header File” button.

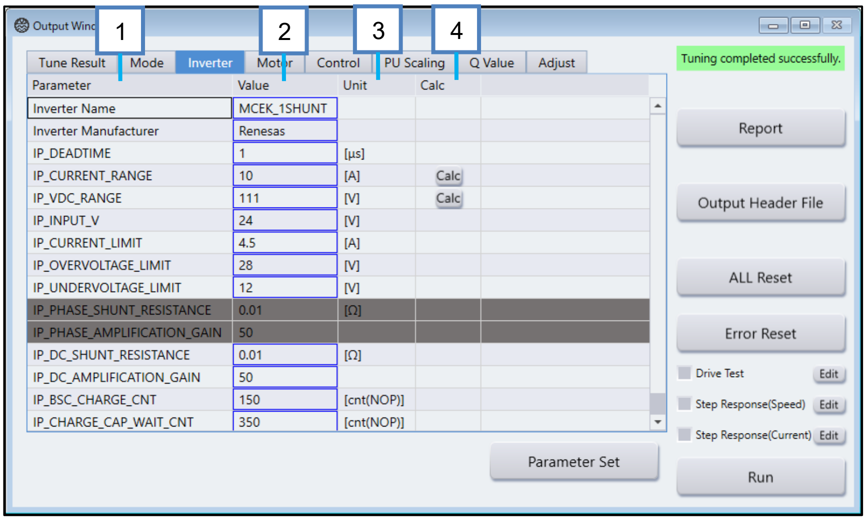


Figure 15-32 Inverter tab

Table 15-11 List of items on Inverter tab

No	Item	Description	Note
1	Parameter	Displays parameter names.	-
2	Value	Entered values.	If the set value exceeds the upper or lower limit values, a red frame will appear as an invalid value. Similarly, a red frame will appear if the setting value does not match the variable type.
3	Unit	Displays the units of each parameter.	
4	Calc button	Pressing the “Select” button to display the calculation window.	

15.5.3.4 Motor TAB

In Motor tab, set specific parameters related to using motor. These settings are reflected in the "r_mtr_motor_parameter.h" file, which is output via "Output Header File" button.

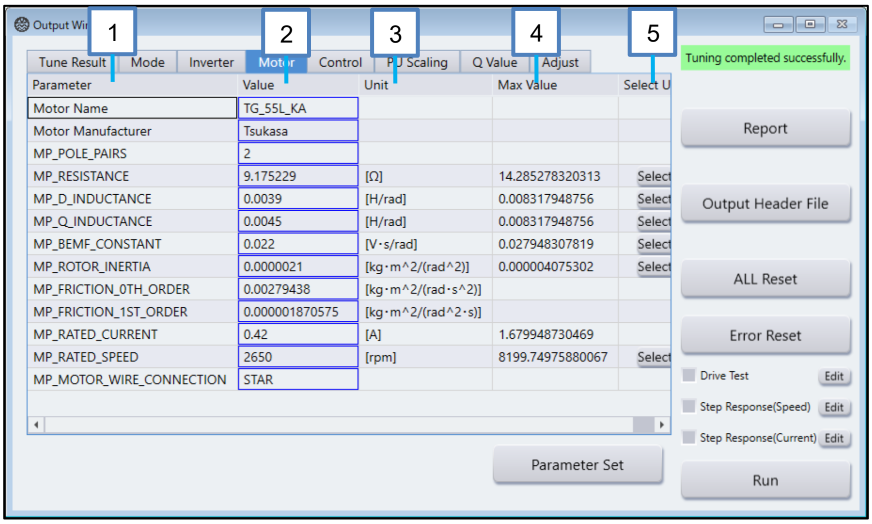


Figure 15-33 Motor tab

Table 15-12 List of items on Motor tab

No	Item	Description	Note
1	Parameter	Parameter names.	-
2	Value	Entered values.	If the set value exceeds the upper or lower limit values, a red frame will appear as an invalid value. Similarly, a red frame will appear if the setting value does not match the variable type.
3	Unit	Displays the units of each parameter.	-
4	Max Value	Displays the maximum value.	-
5	Select Unit button	Pressing the "Select" button to display the unit conversion window.	-

15.5.3.5 Control TAB

In Control tab, set the control parameters. These settings are reflected in the “r_mtr_control_parameter.h” output by “Output Header File” button.

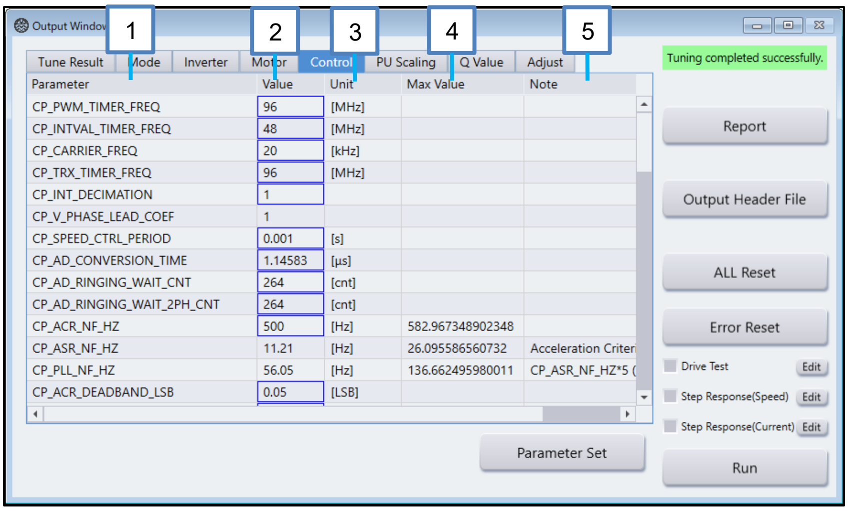


Figure 15-34 Control tab

Table 15-13 List of items on Control tab

No	Item	Description	Note
1	Parameter	Displays parameter names.	-
2	Value	Entered values.	If the set value exceeds the upper or lower limit values, a red frame will appear as an invalid value. Similarly, a red frame will appear if the setting value does not match the variable type.
3	Unit	Displays the units of each parameter.	-
4	Max Value	Displays the maximum value.	If the set value exceeds the upper or lower limit values, a red frame will appear as an invalid value. Similarly, a red frame will appear if the setting value does not match the variable type.
5	Note	Displays formulas for calculating parameter values and related Adjust tab item number.	-

15.5.3.6 PU Scaling TAB

In PU Scaling tab, set the scaling parameters. The scaling parameters are automatically calculated and displayed based on the inverter board and motor characteristic and control parameters set so far. This setting is reflected in the “r_mtr_scaling_parameter.h” file, which is output via “Output Header File” button.

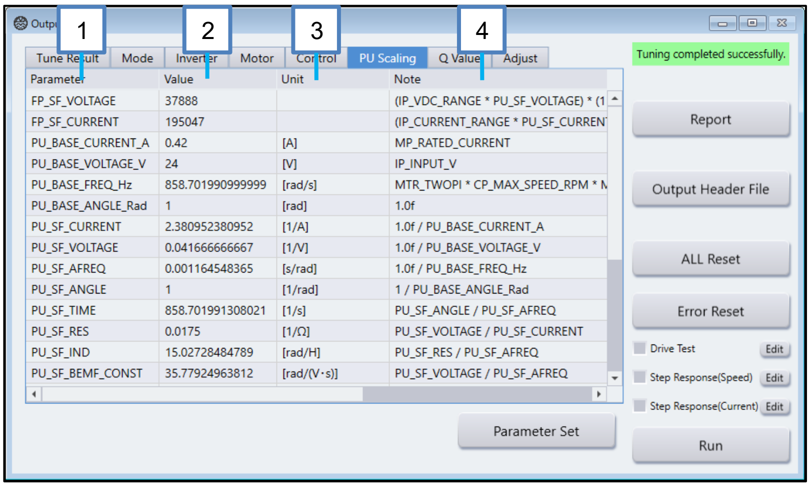


Figure 15-35 PU Scaling tab

Table 15-14 List of items on PU Scaling tab

No	Item	Description	Note
1	Parameter	Displays parameter names.	-
2	Value	Displays parameter values.	If the set value exceeds the upper or lower limit values, a red frame will appear as an invalid value. Similarly, a red frame will appear if the setting value does not match the variable type.
3	Unit	Displays the units of each parameter.	-
4	Note	Displays formulas for calculating parameter values.	-

15.5.3.7 Q Value TAB

Q Value tab is used to set the Q Value, which is automatically set to fit within 16 bits. To change the value, check “User Input” checkbox. This setting is reflected in the “r_mtr_scaling_parameter.h” file, which is output via “Output Header File” button.

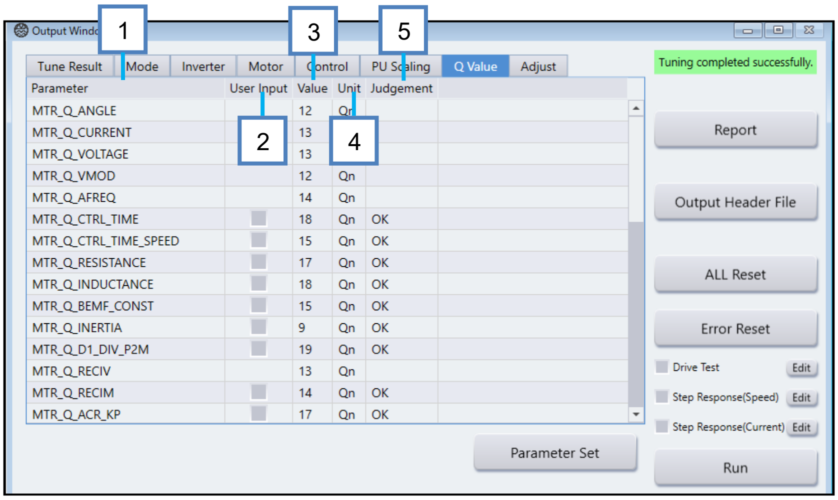


Figure 15-36 Q Value tab

Table 15-15 List of items on Q Value tab

No	Item	Description	Note
1	Parameter	Displays parameter names.	-
2	User Input	Check this box for parameter input.	If the set value exceeds the upper or lower limit values, a red frame will appear as an invalid value. Similarly, a red frame will appear if the setting value does not match the variable type.
3	Value	Entered values.	-
4	Unit	Displays the units of each parameter.	-
5	Judgement	Displays the judgment results about input parameters.	-

15.5.3.8 Adjust TAB

Adjust tab is used to enter adjustment parameters. Each corresponding parameter on the Control tab can be adjusted.

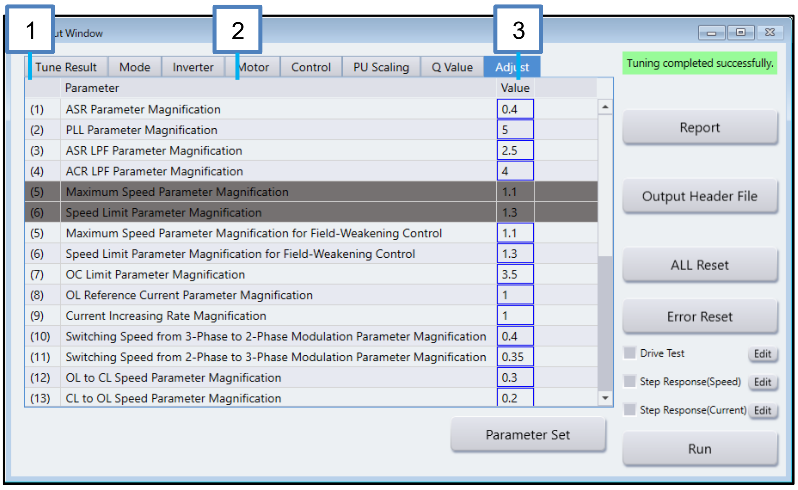


Figure 15-37 Adjust tab

Table 15-16 List of items on Adjust tab

No	Item	Description	Note
1	-	Displays item number.	The item number, correspond to the "Note" on "Control" tab, will be listed.
2	Parameter	Displays parameter names.	-
3	Value	Entered values.	If the set value exceeds the upper or lower limit values, a red frame will appear as an invalid value. Similarly, a red frame will appear if the setting value does not match the variable type.

15.5.4 Sequence editing functions (Sequence Edit Window)

The sequence editing function edits the sequence of parameters used for driving tests. The default setting uses the automatically calculated value (Calc Value). By checking “User Input” box and entering a value in “User Input Value”, you can use required value as a sequence one.

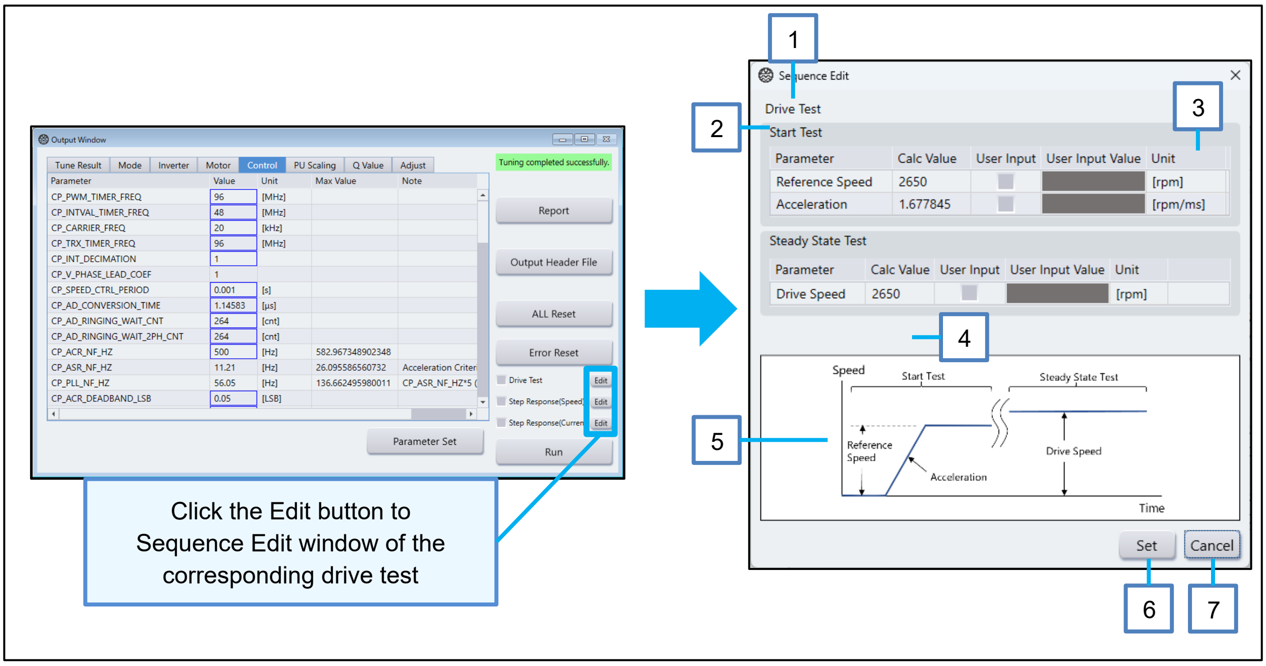


Figure 15-38 Sequence Edit form

Table 15-17 List of items on Sequence Edit form

<table>
<tr><th>No</th><th>Item</th><th>Description</th></tr>
<tr><td>1</td><td>Drive Test Name</td><td>Displays the name of the driving test for which the sequence is to be edited.</td></tr>
<tr><td>2</td><td>Drive Mode</td><td>Displays the name of the target driving test.</td></tr>
<tr><td>3</td><td>Sequence Parameters</td><td>Display and edit sequence parameters.
<table>
<tr><th>Item</th><th>Description</th><th>Note</th></tr>
<tr><td>Parameter</td><td>Displays the name of the parameter to be edited.</td><td>-</td></tr>
<tr><td>Calc Value</td><td>Displays the calculated value of the parameter.
If you do not enter a parameter, the calculated value will be used as the parameter value.</td><td>If the value is invalid, a red frame will appear.</td></tr>
<tr><td>User Input</td><td>Check this box for parameter input.</td><td>-</td></tr>
</table>
</td></tr>
<tr><td>4</td><td>Error Message</td><td>If the parameter value is an invalid value, an error message is displayed.</td></tr>
<tr><td>5</td><td>image</td><td>Displays the corresponding diagram of sequence parameters in the Drive Test waveform.</td></tr>
<tr><td>6</td><td>Set button</td><td>Reflects edited sequence parameters and closes the “Sequence Edit Form”.</td></tr>
<tr><td>7</td><td>Cancel button</td><td>Closes this window.</td></tr>
</table>

15.5.5 Drive Test function (Scope Window)

Perform Drive Tests using set parameters. The kinds of drive tests that can be performed are “starting”, “steady drive”, “speed step response”, and “current step response”.

When the Scope window is activated, the tab for the drive mode selected on the measurement result window is displayed and the Drive Test is performed.

* The motor may not rotate properly during the Drive Test. If you feel that the motor is driving abnormally, immediately stop the motor by pressing the “Stop” button on the Output Window.

*If the starting time of the motor is long, the waveform of the starting test may not fit on the time scale, and subsequent driving tests may not be executed properly. Also, when a drive test is executed two or more times, the waveform will be displayed with the same vertical axis scale and offset as the first time.

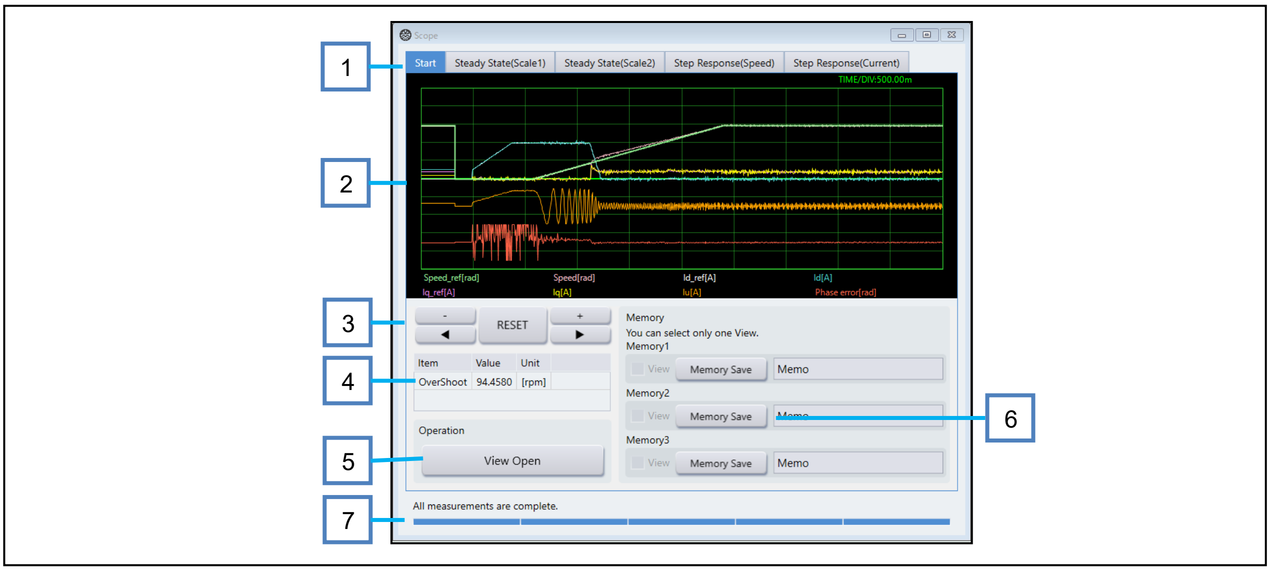


Figure 15-39 Scope window

Table 15-18 List of items on Scope window

No	Item	Description
1	Drive Mode Tab	Displays the tab for the drive mode selected on the measurement result window. The Drive Test result of the applicable drive mode is displayed by selecting the tab. The tabs for the drive mode are as follows. Start: Displays the waveform at motor startup. Steady state(scale1): Displays the waveform in the steady state of the motor, scaled to the natural frequency of the speed control. Steady state(scale2): Displays the motor steady state waveform scaled to the natural frequency of the current control. Step Response(Speed): Displays the waveform of a step change in motor speed. Step Response(Current): Displays the waveform of a step change in motor current.
2	Waveform Graph	Display the waveforms of the Drive Test results. Display with appropriate time scaling according to control parameter values. The scaling of the vertical axis of the graph can be changed with the Zoom function. Wave form is displayed at the finish of each driving test.
3	Zoom Control	Expand/contract the waveform and move the range of the Drive Test result.
4	Calculated value	Display the value calculated based on the data obtained from the Drive Test results. Display items shall be as follows. Start : overshoot Steady state(scale1) : Ripple, Steady state deviation Steady state(scale2) : Ripple, Steady state deviation Step Response(Speed): Overshoot, settling time, rise time Step Response(Current): Overshoot, settling time, rise time scale1: Scale to natural frequency of speed control scale2: Scale to natural frequency of current control
5	View Open button	Displays the Channel Information window.
6	Waveform saving function	Save up to three waveforms that are being displayed.
7	progress bar	Display the progress during the Drive Test.

15.5.5.1 Zoom function

The zoom function allows you to change the scaling of the graph horizontal axis and displayed area.

Figure 15-40 “Zoom” control

Table 15-19 List of items on “Zoom” control

No	Item	Description
1	- button	Zoom out.
2	+ button	Zoom in.
3	◂ button	Move to the left.
4	▸ button	Move to the right.
5	RESET button	Reset scaling to initial state.

15.5.5.2 Channel information edit function

When “View Open” button is pressed, Scope Channel Info window is displayed. In this window, user can check the waveform display information. In addition, the vertical axis scaling of the graph can be changed by editing the setting values.

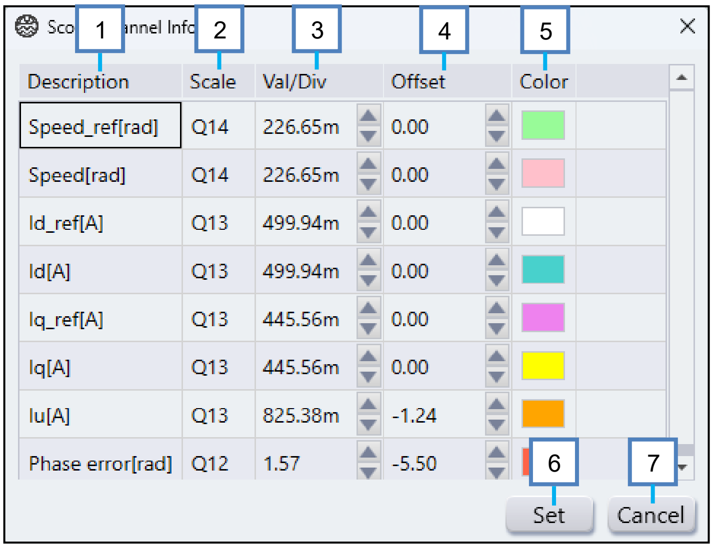


Figure 15-41 Scope Channel Info window

Table 15-20 List of items on Scope Channel Info window

No	Item	Description	Note
1	Description	Display waveform description.	-
2	Scale	Set up the display scale. Allows input in Q Values and scale values.	If the value is other than the Q value or scale value, a red frame will appear as an invalid value.
3	Val/Div	Set up the value per div.	-
4	Offset	Set up display offset values.	-
5	Color	Set up graph line colors.	-
6	Set button	Reflects the edited channel information closes the Scope Channel Info.	-
7	Cancel button	Closes this window.	-

15.5.5.3 Waveform saving function

The waveform saving function can save up to three waveforms that are being displayed, and each waveform can be displayed in turn.

Clicking the “Memory Save” button stores the waveform and the display changes to “Memory Clear”. Click again to clear the memory. Check the “View” checkbox to display the saved waveform.

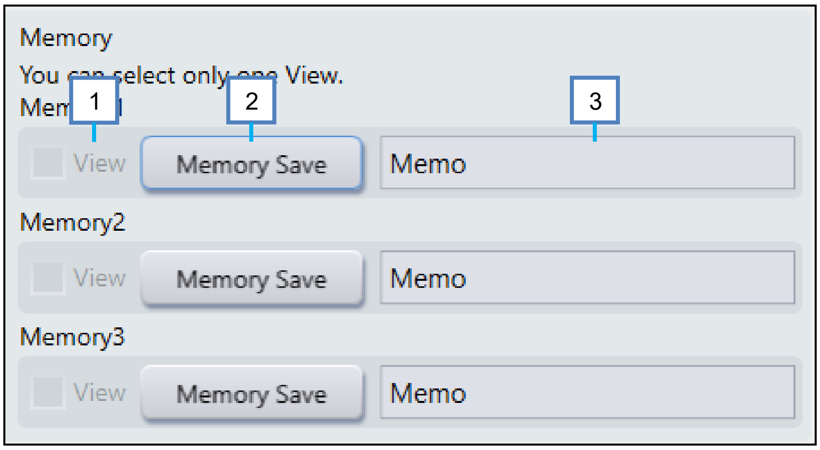


Figure 15-42 Waveform saving function

Table 15-21 List of items on Waveform saving function

No	Item	Description
1	View checkbox	Select whether to display the saved waveform data.
2	Memory Save button	Saves the measured waveform data. Pressing the "Memory Clear" button clears the memory.
3	Memo	Area where notes can be written.

15.5.6 Report function (Output Window)

Outputs set parameters and Drive Test results as a report file.

* Parameter values may not be properly reflected in the report file. Please check the Output Window for the correct parameter values.

15.5.6.1 Displaying output contents and selecting output parameters

The first page of the report window is used to select output contents and output parameters.

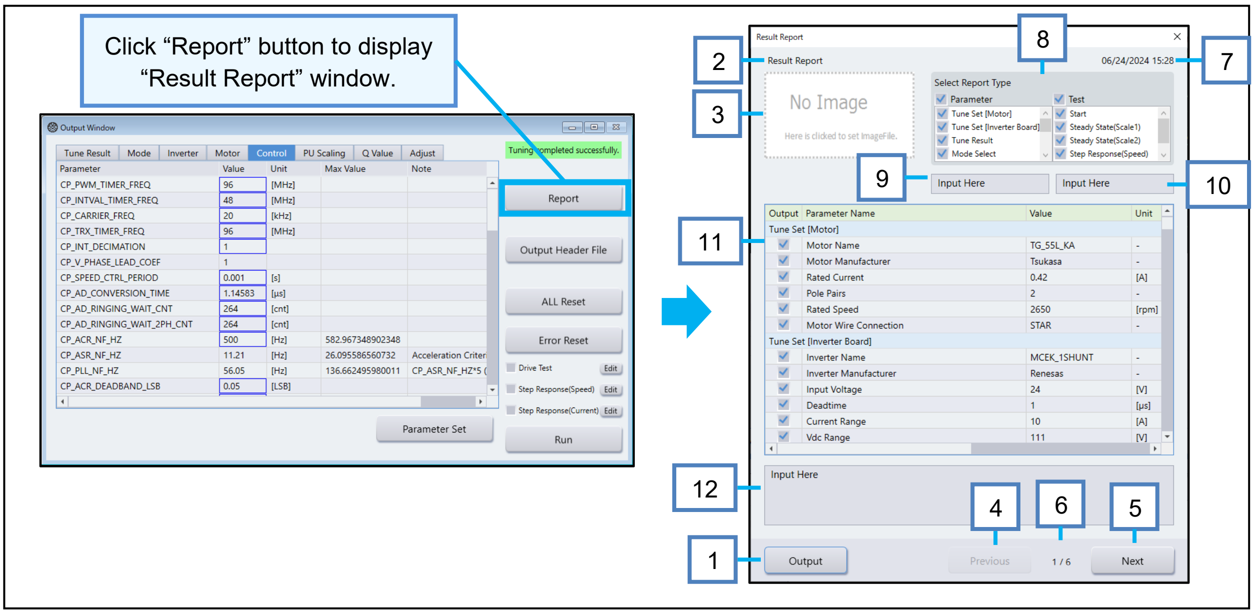


Figure 15-43 Result Report (Page 1)

Table 15-22 List of items on Result Report (Page 1)

No	Item	Description
1	Output button	Create a reports file in PDF format.
2	Page title	Display the page title.
3	Image Selection	Select images to be displayed in the report.
4	Previous button	Moves to the previous page of the report window.
5	Next button	Moves to the next page of the report window.
6	Page	Show current page count/total number of pages.
7	Report Date	Display the date and time when the report was output (when the Report button is clicked).
8	Output Content Selection	Select which information on parameters and Drive Tests to output by using the checkboxes. For parameters, select the following classification. - Tune Set [Motor] - Tune Set [Inverter Board] - Tune Result - Mode Select - Inverter Parameter - Motor Parameter - Control Parameter - Scaling Parameter - Adjust Parameter The Drive Test is selected from the drive results of the following modes and all results stored in Memory. - Drive Test - Step Response(Speed) - Step Response(Current)
9	Free text area 1	Text field that the user can freely enter.
10	Free text area 2	Text field that the user can freely enter.
11	Result display grid	Display parameters. Output availability of each parameter can be selected by checkboxes.
12	Free text area 3	Text field that the user can freely enter. Allows multiple lines of input.

The parameter list output as tuning results is shown below.

Table 15-23 Define Directives

Definition name	Description	Unit
MP_RESISTANCE	Resistance	[Ω]
MP_D_INDUCTANCE	d-axis inductance	[H/rad]
MP_Q_INDUCTANCE	q-axis inductance	[H/rad]
MP_BEMF_CONSTANT	BEMF constant	[V･s/rad]
MP_ROTOR_INERTIA	Inertia	[kg･m^2/(rad^2)]
MP_FRICTION_0TH_ORDER	Static friction coefficient	[kg･m^2/(rad･s^2)]
MP_FRICTION_1ST_ORDER	Kinetic friction coefficient	[kg･m^2/(rad^2･s)]
CP_SAL_ANGLE_CURRENT	Threshold current of angle detection for salient rotor	[A]
CP_SAL_ANGLE_TRX_THRESHOLD	TRX count value differential of angle detection for salient rotor	[cnt(TRX)]
CP_SAL_ANGLE_DISCHARGE	Discharge period of angle detection for salient rotor	[cnt(NOP)]
CP_SAL_POLARITY_CURRENT	Threshold current of polarity detection for salient rotor	[A]
CP_SAL_POLARITY_TRX_THRESHOLD	TRX count value differential of polarity detection for salient rotor	[cnt(TRX)]
CP_SAL_POLARITY_DISCHARGE	Discharge period of polarity detection for salient rotor	[cnt(NOP)]
CP_NON_SAL_CURRENT	Threshold current of angle detection for non-salient rotor	[A]
CP_NON_SAL_TRX_THRESHOLD	RX count value differential of angle detection for non-salient rotor	[cnt(TRX)]
CP_NON_SAL_DISCHARGE	Discharge period of angle detection for non-salient rotor	[cnt(NOP)]
CP_IPD_NOISE_AVOID_CNT	Noise avoidance count at IPD	[cnt(TRX)]

15.5.6.2 Drive Test result display

The second and subsequent pages display the results of the Drive Test.

The Drive Test results are displayed, including the results by drive mode on the Scope window and the contents saved with the “Memory Save” button.

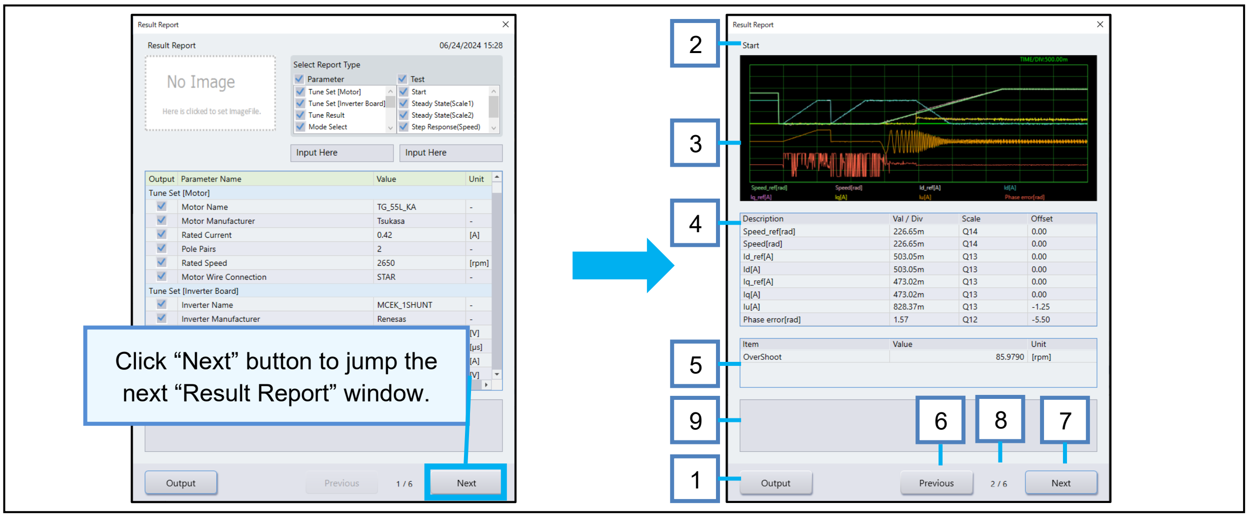


Figure 15-44 Result Report (Page 2)

Table 15-24 List of items on Result Report (Page 2)

No	Item	Description
1	Output button	Create a report file in PDF format.
2	Page title	Display the driving mode and “Memo” of the waveform saving function as the page title.
3	Waveform Graph	Display the waveform of the Drive Test result.
4	Calculated value	Display the value calculated based on the data obtained from the Drive Test results.
5	Previous button	Page transition (Previous) on the report window.
6	Next button	Page transition (Next) on the report window.
7	Page	Show current page count/total number of pages.
8	Free text area	Text field that the user can freely enter. Allows multiple lines of input.

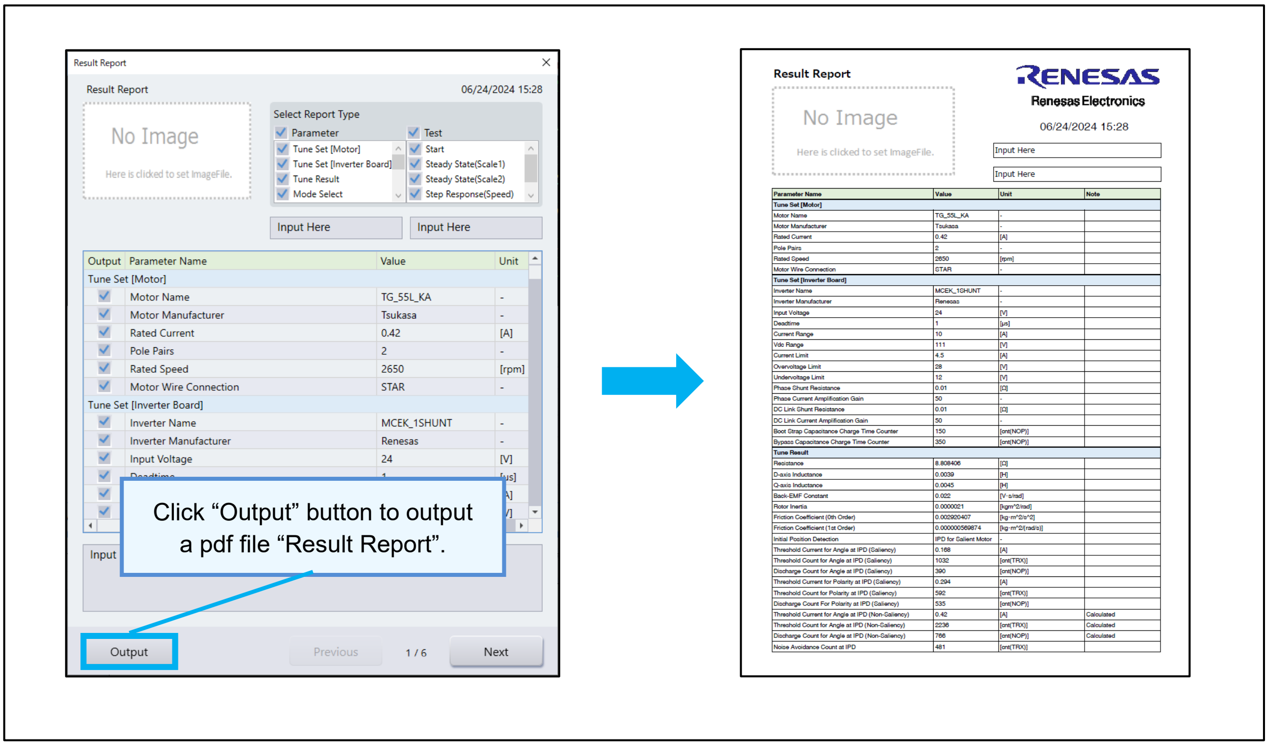


Figure 15-45 PDF output result (Output results of designed parameters)

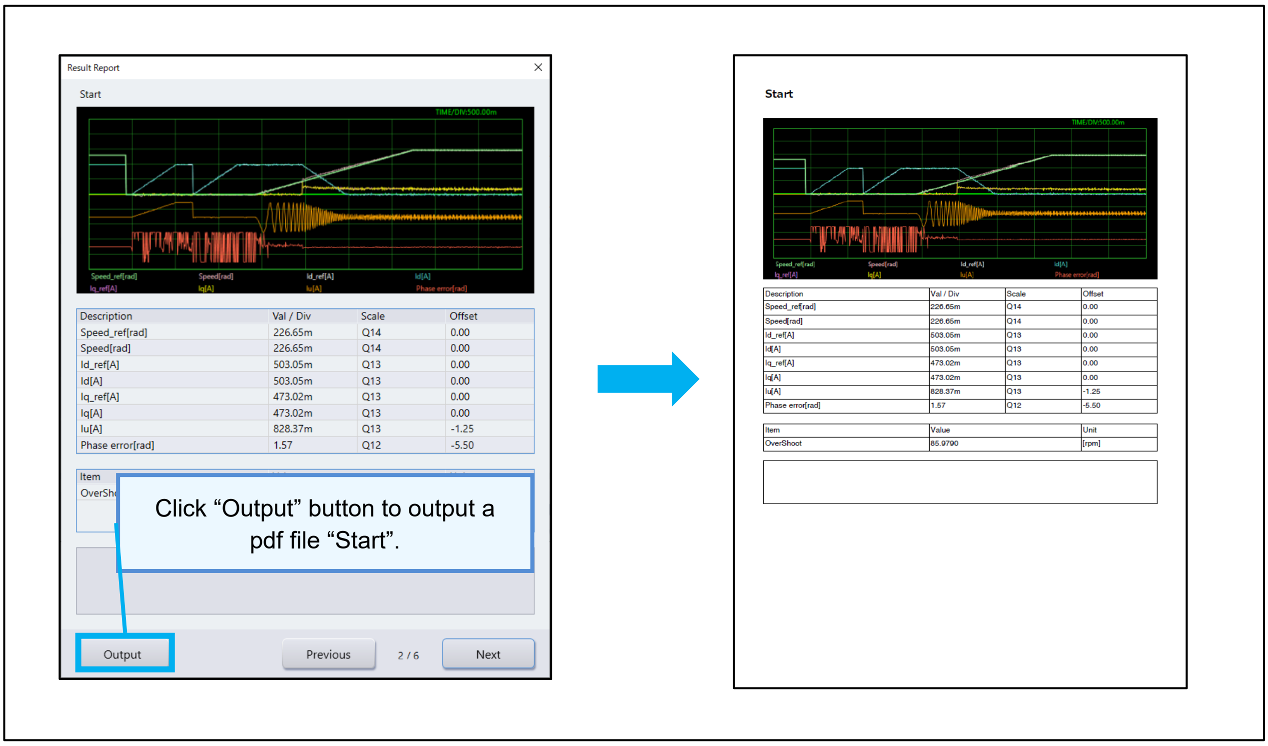


Figure 15-46 PDF output result(Drive Test results)

15.5.7 Header file output function (Output Window)

Outputs set parameters as header files corresponding to the target Renesas sample software.

Click “Output Header File” button on Output Window to display the folder selection dialog and save the header files in the selected folder.

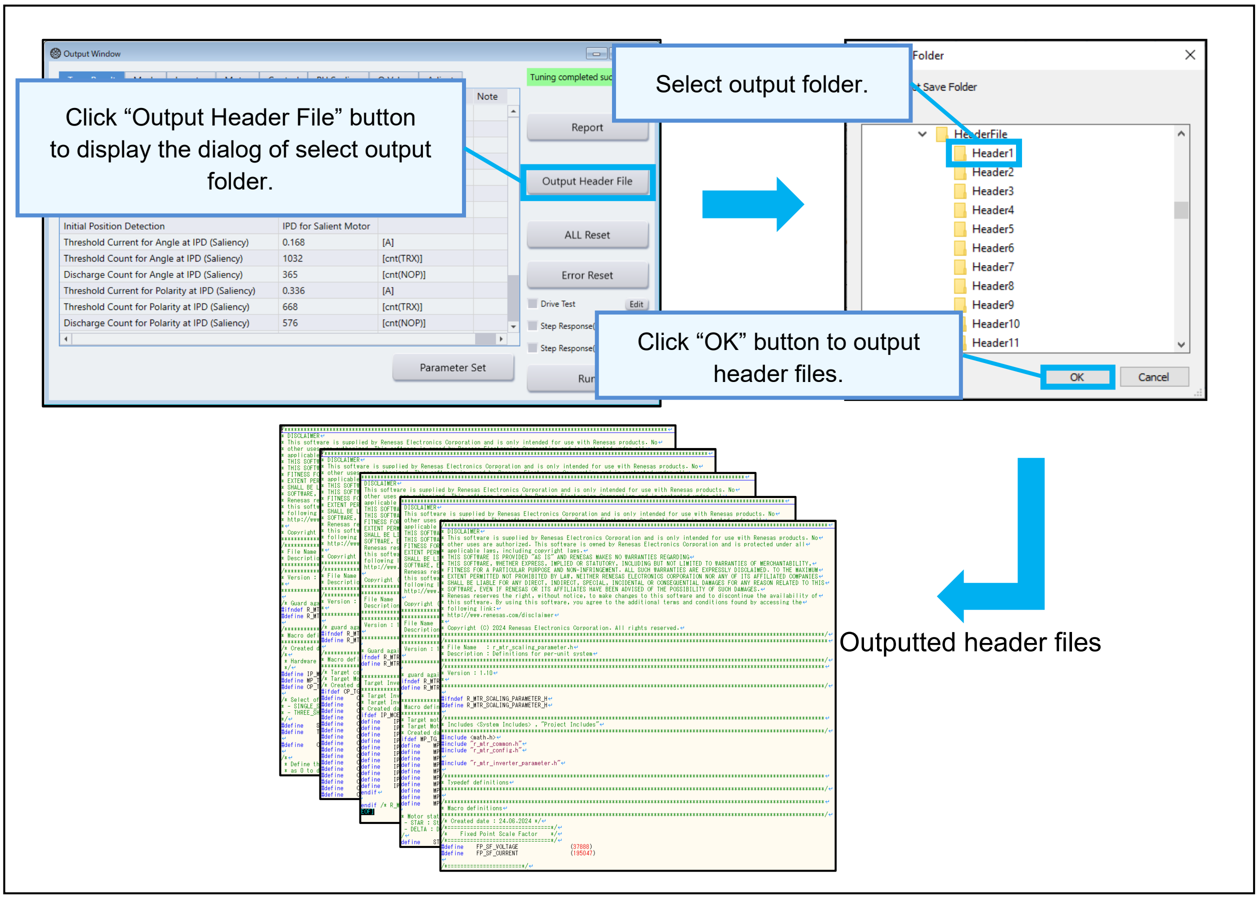


Figure 15-47 Header file output function

Table 15-25 List of created files

No	File	Description
1	r_mtr_config.h	File for Motor Configuration
2	r_mtr_inverter_parameter.h	File for Inverter Parameter
3	r_mtr_motor_parameter.h	File for Motor Parameter
4	r_mtr_control_parameter.h	File for Control related Parameter
5	r_mtr_scaling_parameter.h	File for Scaling Parameter

15.5.8 Reset Function (Output Window)

If an error occurs in the drive program after parameter setting, select the type of reset to execute. Click the "ALL Reset" or "Error Reset" button on the Output Window screen to display the Tuner Reset screen.

- ALL Reset

  Resets all designed parameters. Input window will be displayed, so perform automatic measurement again.

- Error Reset

  Resets only the error state and the designed parameters are retained.

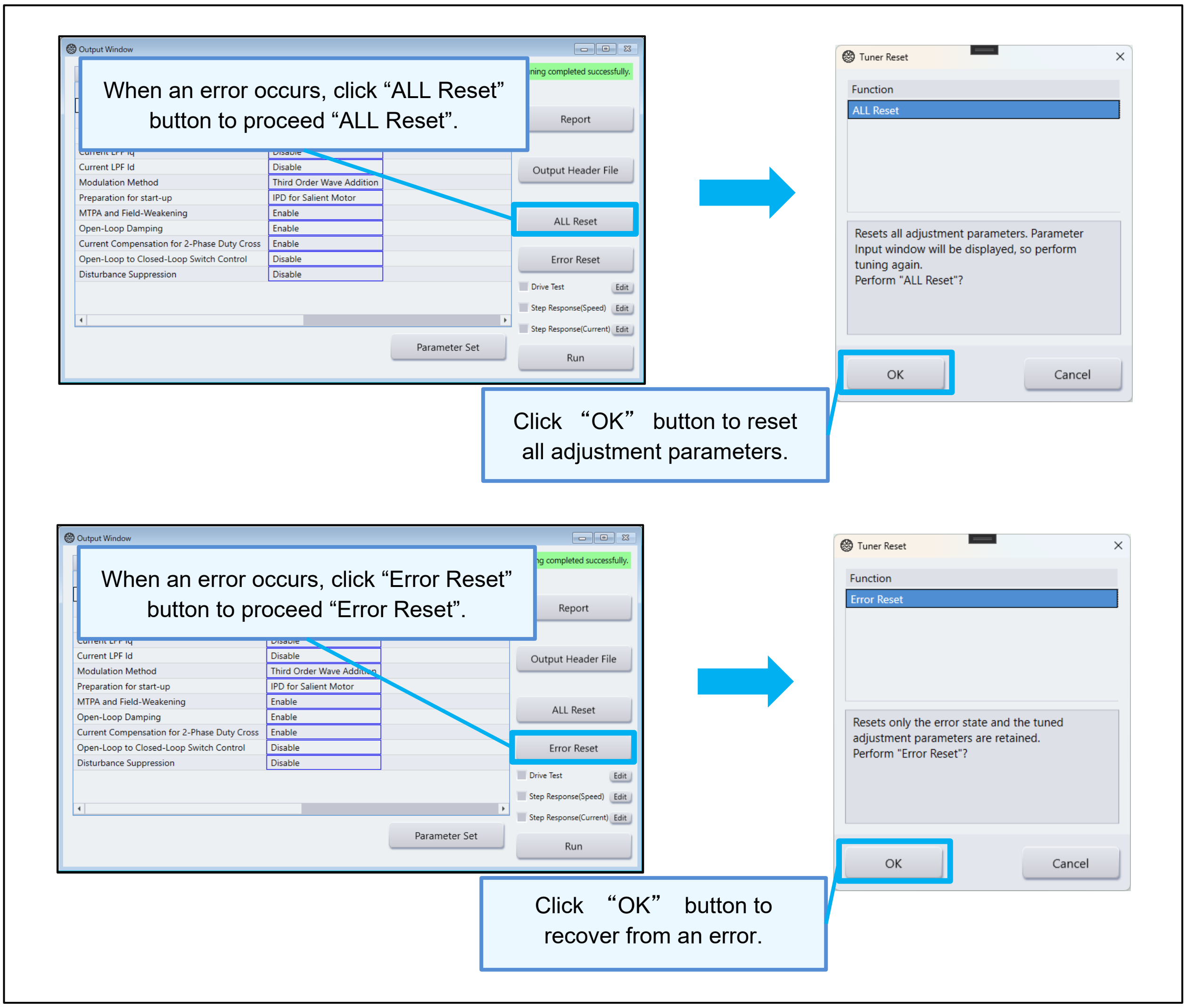


Figure 15-48 Reset function

16.Tuner Tool (for MCI-HV-1)

This section describes how to use the Tuner tool, which is activated by clicking the Tuner icon in the Select。 This section is based on Section 14 of the Renesas Motor Workbench User's Manual (r21uz0004), a motor control development support tool, and has been re-edited for the MCI-HV-1.

16.1 Execution environment preparation

This section describes the preparations required to start the Tuner tool-compatible sample program using the MCI-HV-1.
The sample program used in this document uses MCB-RA6T2 as the CPU board.
There are two input voltages to MCI-HV-1, AC input and DC input. In this sample program, the AC input(AC 100V or 200V) and the DC input (DC 390V) is used.
When DC input an external auxiliary power supply (+15V) input is also required.

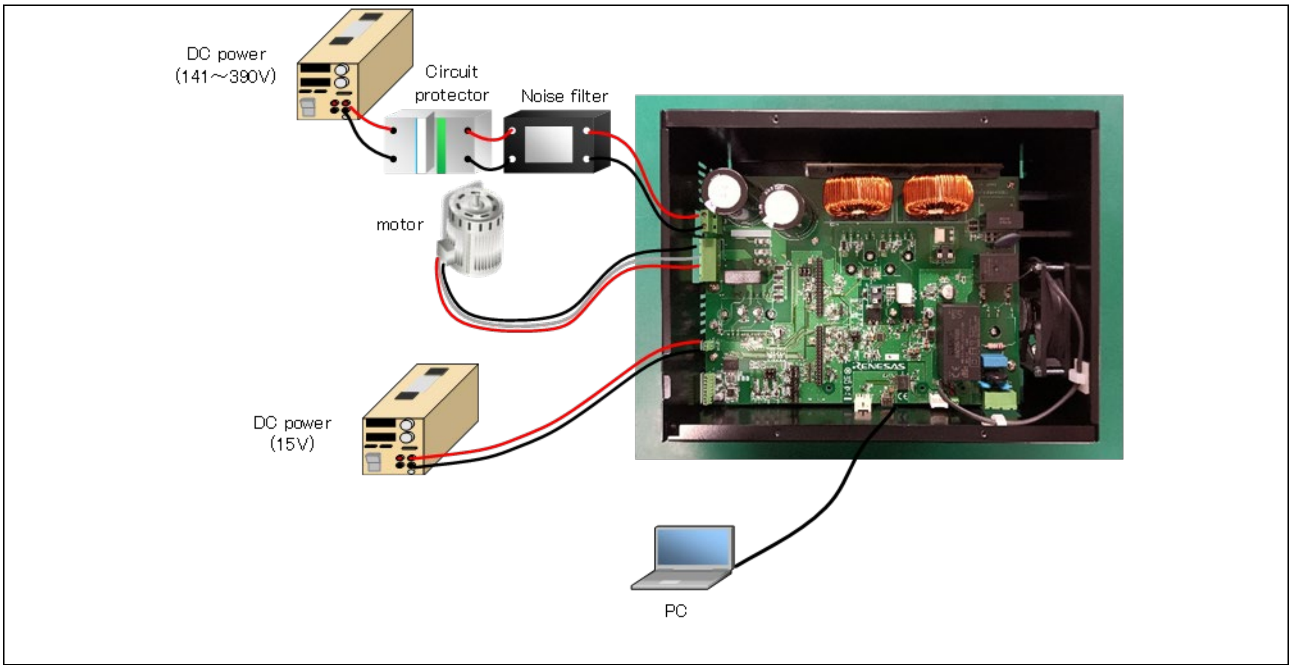


Figure 16-1 Connection image DC (Excerpt from MCI-HV-1 User's Manual)

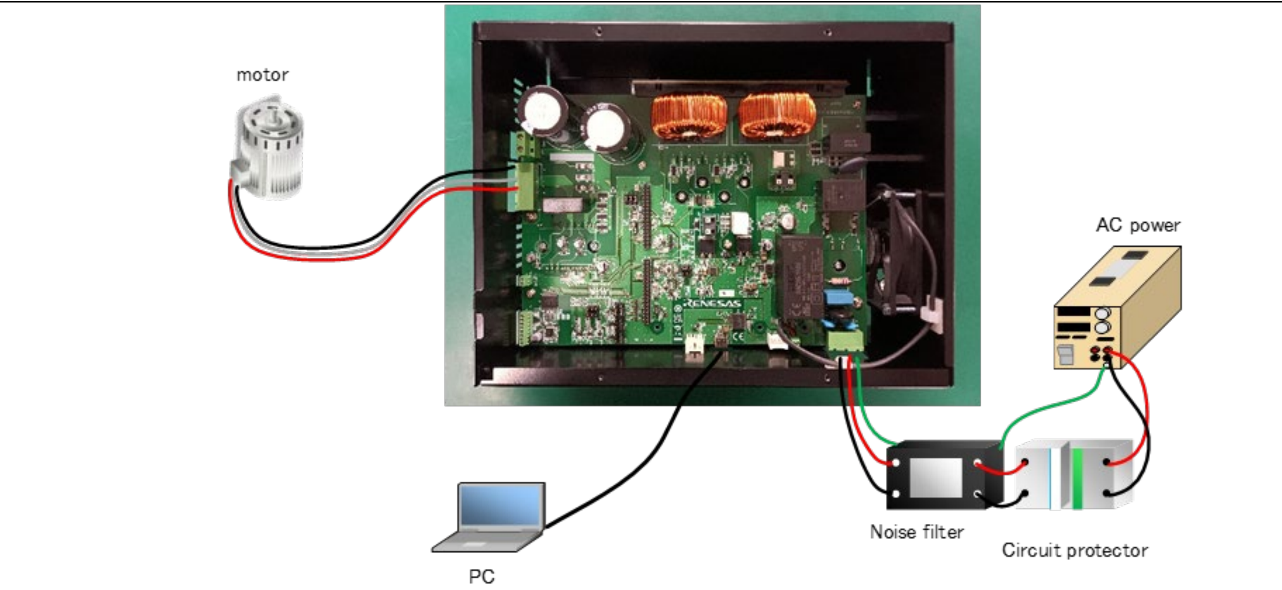


Figure 16-2 Connection image AC (Excerpt from MCI-HV-1 User's Manual)

16.1.1 Advance confirmation

Please read the MCI-HV-1 User's Manual (r12uz0138) and check the precautions when using the MCI-HV-1.

16.1.2 Changing MCI-HV-1 settings

Please read "3.2 Power Supply" and "4.1 Power Supply" in the MCI-HV-1 User's Manual and act as necessary.

16.1.3 Writing Sample Programs

(a) In case of using the executable file which is included in Renesas Motor Workbench, please download the file which supports “Tuner” function into the CPU board.

- R***_MCIHV1_PM_LESS_FOC_TUNER_***.HEX

(b) In case of using the sample program (Project format) which is released by Renesas Electronics, please download built executable file which includes “Tuner” function into the CPU board.

About the method of including “Tuner” function into a sample program, please refer to Section “14.6”.

Please read "3. Setup Guide" in the MCI-HV-1 User's Manual for details on how to connect the CPU board to a PC (Windows 10 or 11).

The USB connector on the CPU board side is USB-Type C.

16.1.4 Connection of various boards and motors

For details on how to connect the three boards (MCI-HV-1, CPU board, and communication board for RMW) and the motor you plan to use, please read "3.1 Hardware setup" in the MCI-HV-1 User's Manual.MCI-HV-1

16.1.5 Power supply connection

Read "3 Setup guide" in the MCI-HV-1 User's Manual before connecting.

For the DC power supply, select and use a device that can output 390 VDC and has the capacity to operate the motor without load. Due to the high voltage, be sure to insulate the wiring and terminals with sufficient care.

When supplying power, be sure to supply power in the following order. Supplying DC power for the main circuit before the auxiliary power may cause the inverter to malfunction.

(1) CN6 inverter board +15V connector

(2) CN3 inverter board VBUS connector(DC input)

When AC input

Connect the AC power supply to the power terminal (CN2) of the inverter board via a circuit protector and noise filter.

16.2 Overview

Tuner tool automatically measures specific parameters (e.g resistance, inductance) of a permanent magnet synchronous motor and adjusts various control parameters (PI control gain etc.) required for encoder position control or sensorless speed control.

Tuner tool is available only when the tuning function is supported by the control program. “Tuner” will be displayed as selectable in the “Select Tool” on the Main Window when it is available.

Note : This tool is not guaranteed to be effectively adjustable for all motors.

16.3 Features

- Provides two tuning modes: Manual and Easy.
- Tuning result (adjusted parameters) can be output as a PDF report or as a header file in the Renesas control program.
  The output in its original format cannot be used in the MCI-HV-1 sample program.

16.4 Precautions for use

Since this tuner tool is used for AC200V motors, please keep safe evaluation with the following points in mind.

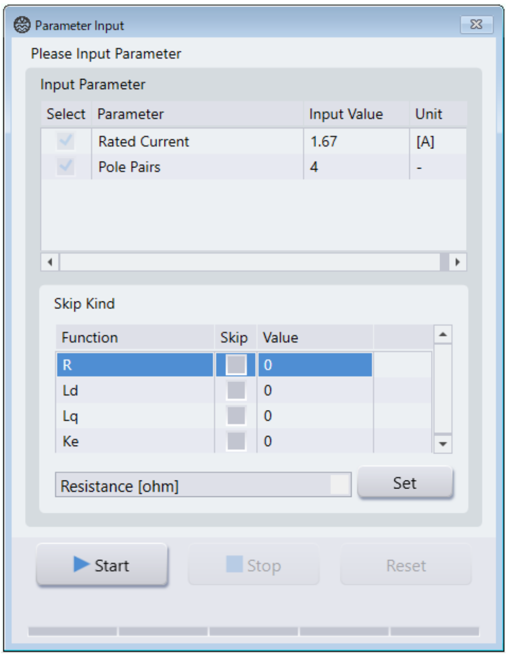
Table 16-1 Tuner Tool / Precautions for use

Points	Detail
Inverter Voltage Protection Level	Overvoltage level 420V Low voltage level 8V
Inverter Current Protection Level	Overcurrent level 21.2A
Applicable Motors	・PM motors (SPM and IPM motors) of 200-VAC type ・Capacity must be less than or equal to MCI-HV-1 and power supply capacity. ・The capacity must be less than the capacity of MCI-HV-1 and the power supply. Do not connect a load.
Other	・Cover the rotating parts of the motor with a metal cover or the like. ・The motor rotates. Please operate in a safe environment. ・The motor will be supplied with electric power. Be careful not to touch the conductive parts to avoid electric shock.

16.5 Window Structure

The window structure of Tuner tool is shown below.

(a) Parameter Input window



(b) Tune Window (Manual tab / Easy tab)

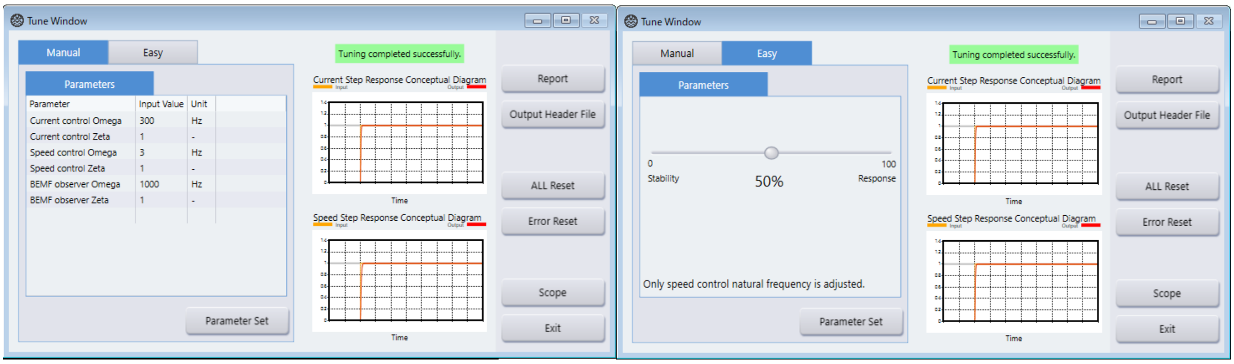


Figure 16-3 Tuner tool window

16.6 Explanation of Operation

This section describes the tuning procedure. For details on the functions on each window, see section 2.4 Main Window Operation.

16.6.1 Preparation

16.6.1.1 Write execution file

Please read "16.1.3 Writing Sample Programs”.

16.6.1.2 Start Renesas Motor Workbench

Click the short-cut icon on the desktop to start-up Renesas Motor Workbench.

16.6.1.3 Load the RMT file

Load the RMT file for Tuner to Renesas Motor Workbench.

- RA6T2_MCIHV1_PM_LESS_FOC_TUNER_E2S_V***.rmt

16.6.1.4 Loading Map File

Loads a map file.

- RA6T2_MCIHV1_PM_LESS_FOC_TUNER_E2S_V***.map

16.6.1.5 Establish communication

Establish communication between Renesas Motor Workbench and the evaluation board.

16.6.1.6 Start Tuner tool

Select Tuner tool from Select Tool of Renesas Motor Workbench and start it.

16.6.2 Performing Tuning

16.6.2.1 Input parameters for tuning

When you start Tuner tool, the "Parameter Input" window will be displayed. In the Input Parameter section above, enter the rated current and pole pairs of the motor referring to the specifications of the motor.

If you know the properties of the motor in advance and want to skip some tuning, specify those parameters in the Skip Kind field at the bottom (for details, refer to "16.7.1 Skip Function (Parameter Input window)").

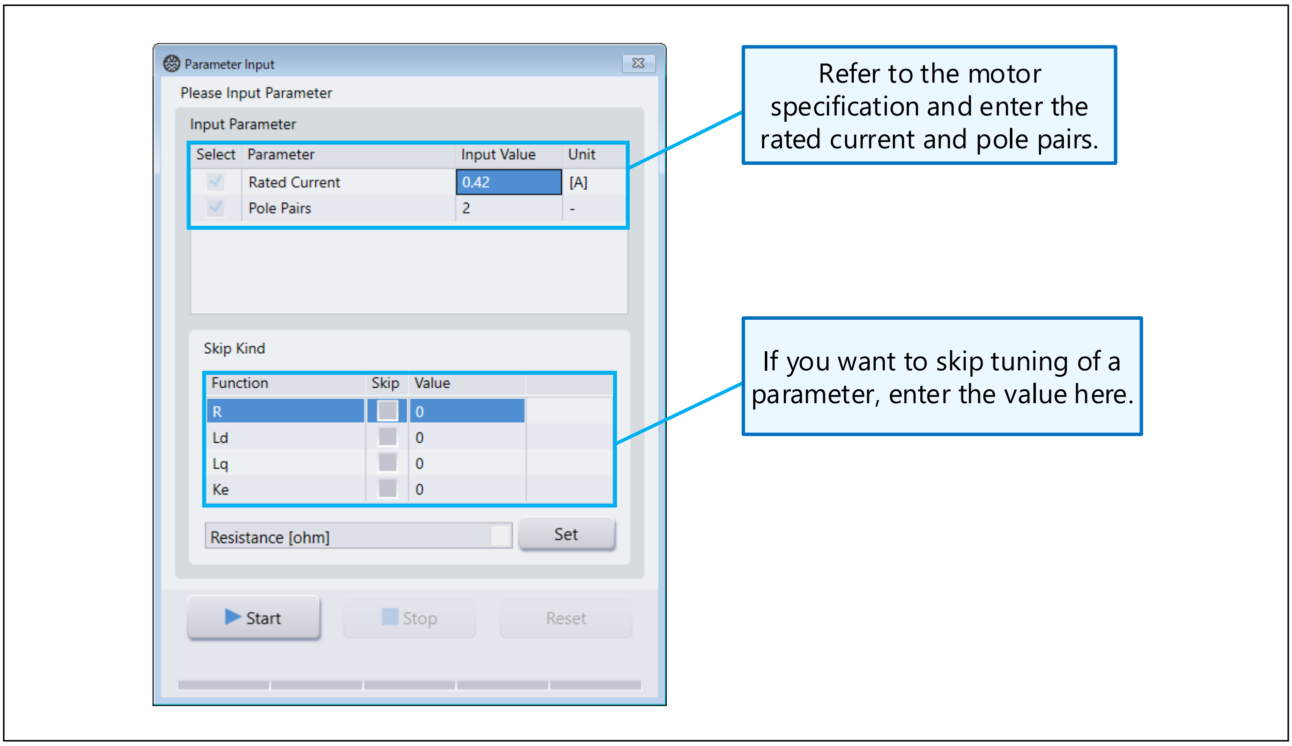


Figure 16-4 Parameter Input window

16.6.2.2 Perform tuning

Click the Start button to start tuning, and the progress bar will be displayed while tuning. To stop tuning, click the “Stop” button. When tuning is completed, “Tune Window” will be displayed.

Note : The motor rotates during tuning. To prevent the coupling attached to the rotating part of the motor from scattering, cover the motor with a metal cover or otherwise execute it in an environment where the motor can rotate.

Read "6. Precautions" in the MCI-HV-1 User's Manual (r12uz0138) again before execution.

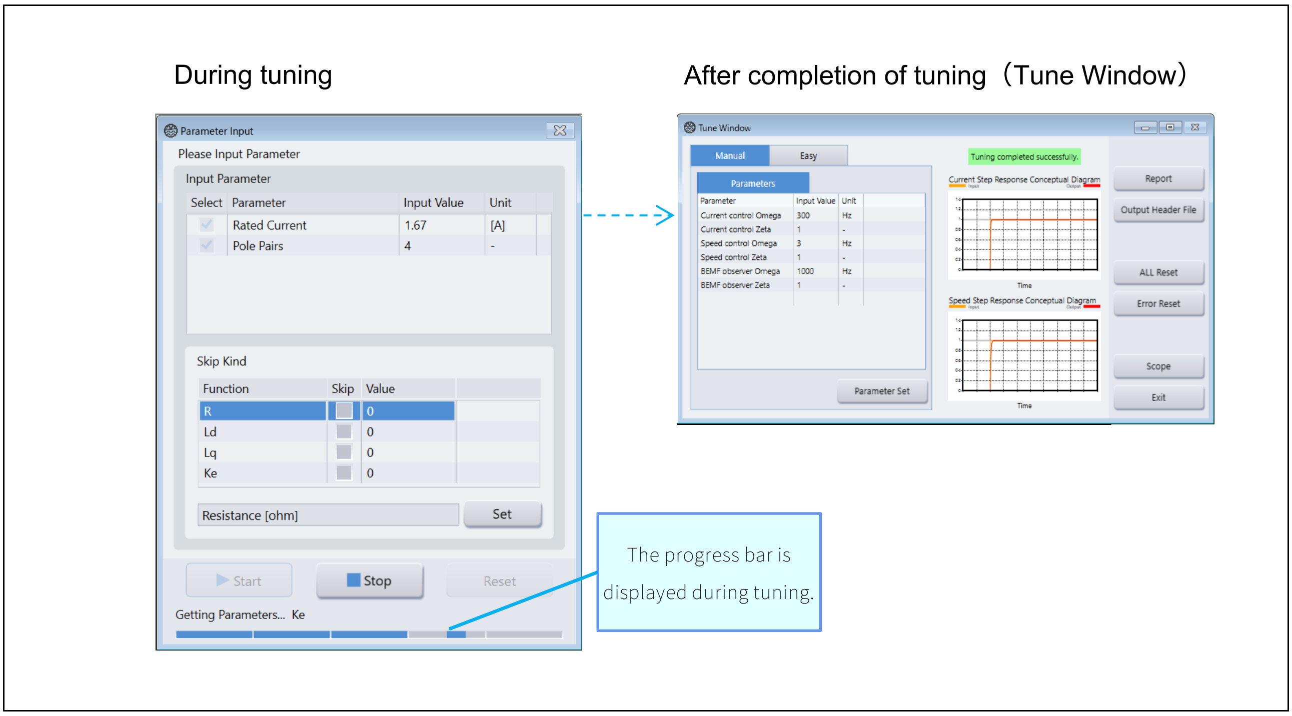


Figure 16-5 Tune Window display under tuning and after tuning

If an error occurs during tuning, check the error message and click the "Reset" button.

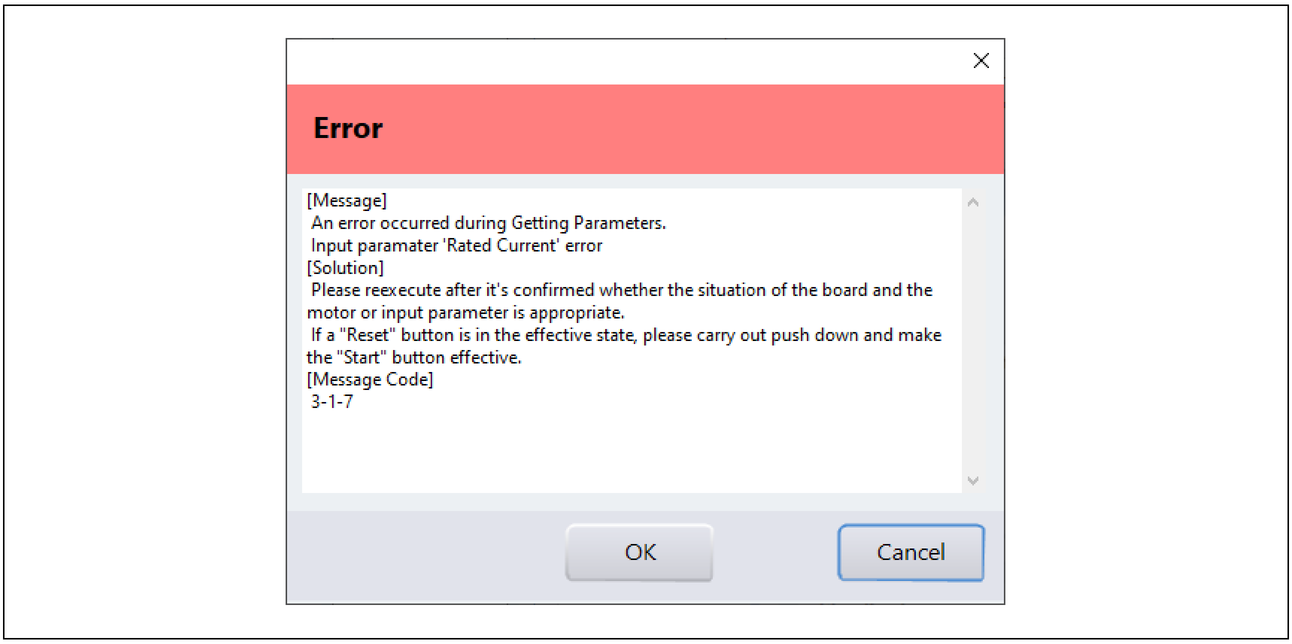


Figure 16-6 Error message example

16.6.2.3 Confirm tuning result

Click the “Report” button on the “Tune Window” and confirm details about tuning on the “Result Report” screen.

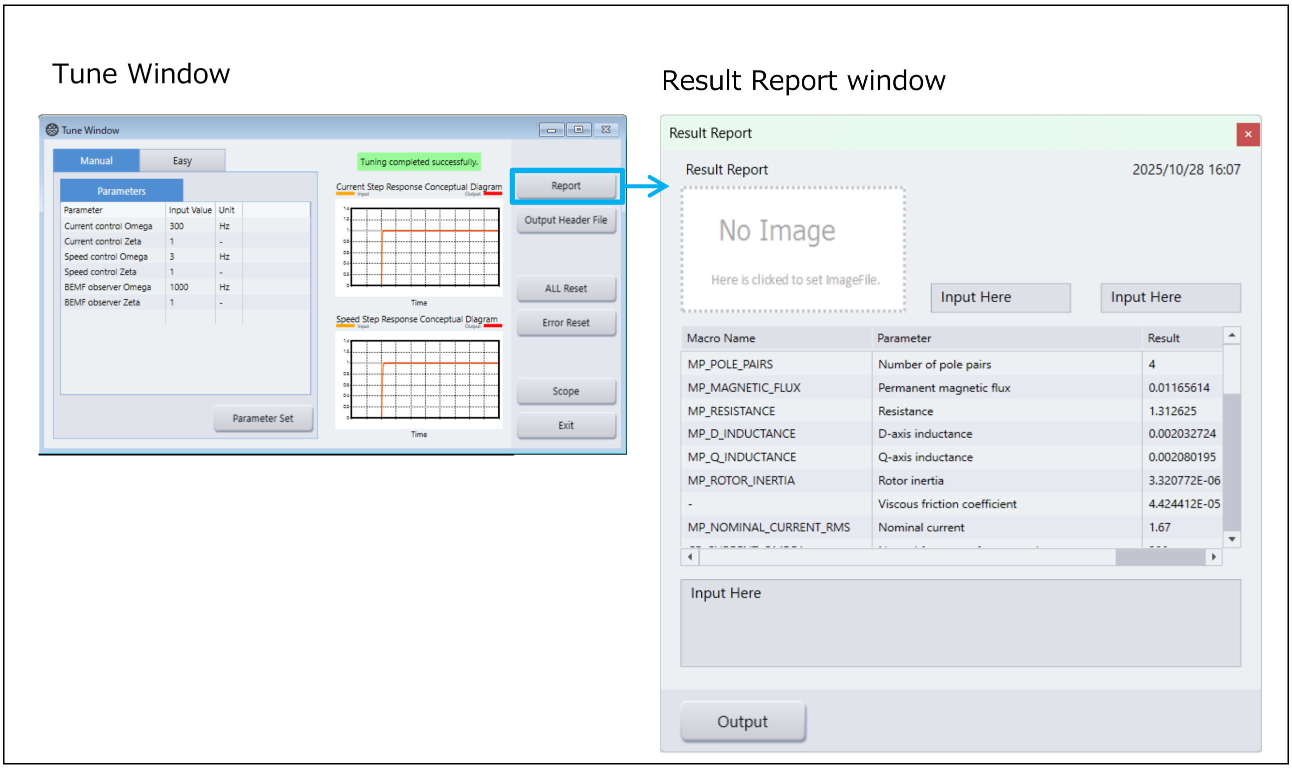


Figure 16-7 Result Report view

16.6.2.4 Terminate tuning

Click the "Exit" button on "Tune Window" to return to the Parameter Input window.

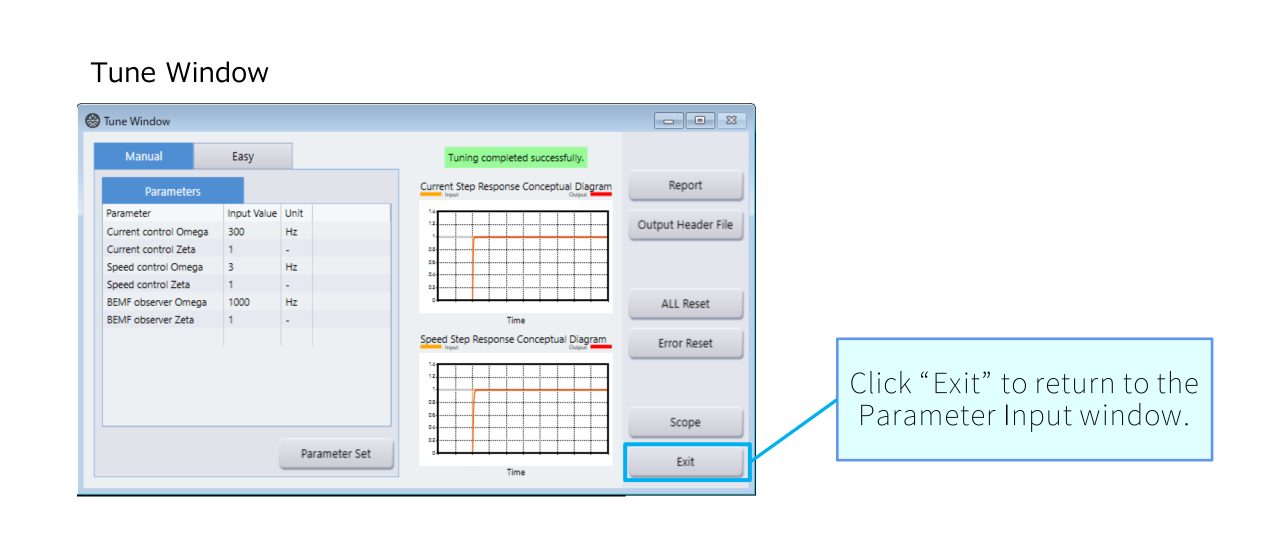


Figure 16-8 Exit button

16.6.3 Checking Operation

Since the tuning results cannot be reflected in the inverter software for MCI-HV-1, motor drive tests cannot be performed. Please reflect the tuning results in the motor parameter header file of the sample program by yourself. See Section 16.7.4(2) for details.

16.6.4 Terminating Tuner tool

If you are testing the motor drive, press the “STOP” button to finish the test. Click the "Exit" button on Tune Window to return to the Parameter Input window. After that, return to Main Window by clicking the “Main Window” button, or operate the tool switch button.

16.7 Function Description

16.7.1 Skip Function (Parameter Input Window)

You can skip measurement of some parameters by entering those tuning parameters such as the resistive, d-axis/q-axis inductance, and magnetic flux before tuning.

In the “Skip Kind” area of the Parameter Input window, select the Skip checkbox of the parameter to be omitted and enter the parameter. When you have finished entry, press the “SET” button.

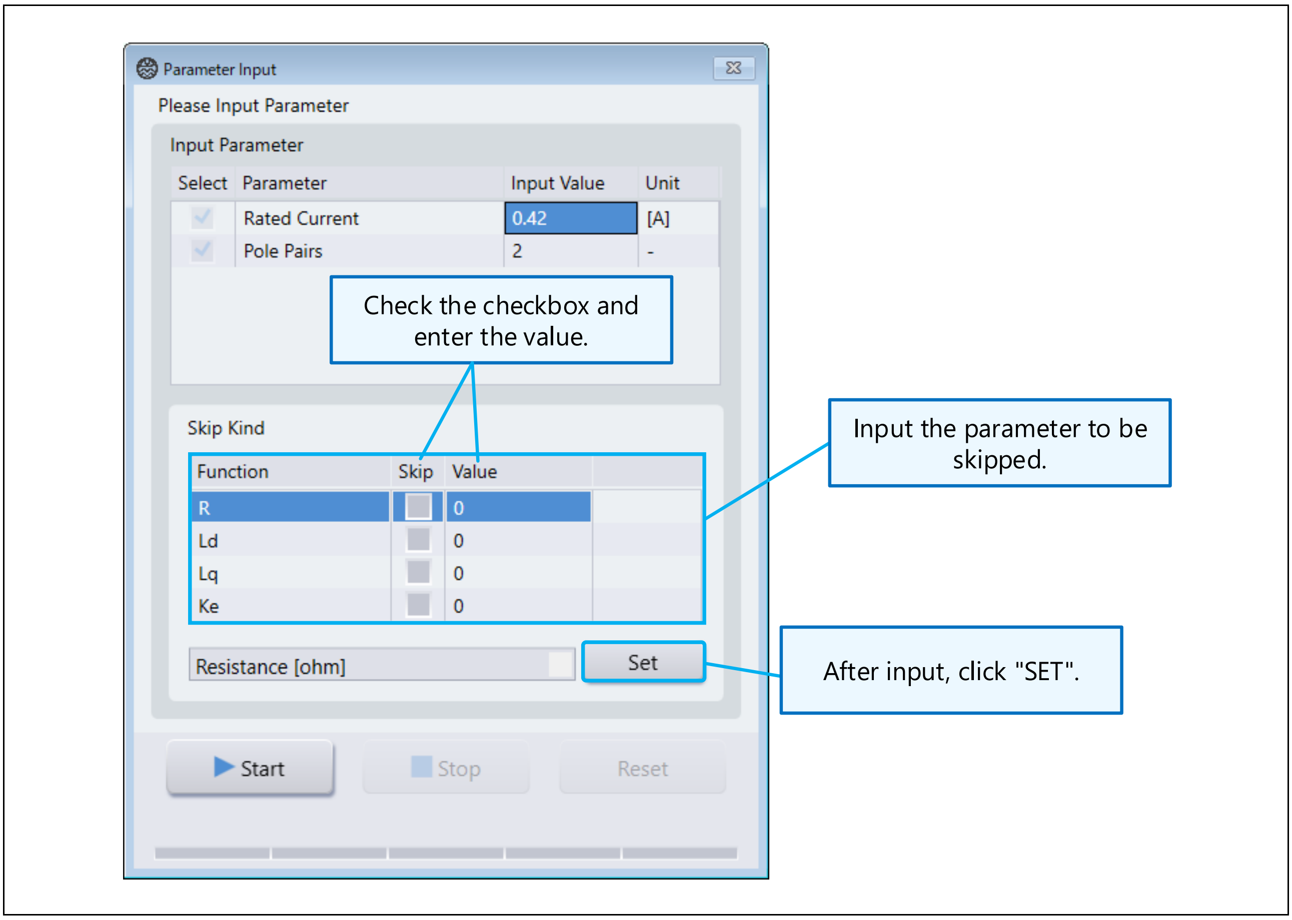


Figure 16-9 Skip Input view

16.7.2 Tuning Control Parameters (Tune Window)

"Tune Window" is a window for control parameter setting. There are two ways to set parameters. You can switch tabs between Manual tab and Easy tab.

On the Manual tab, you can adjust the natural frequency of each feedback loop and damping coefficient to be used and reflect the adjustment to the control.

On the Easy tab, you can adjust parameters that are set to 50% as default by moving the slider from side to side and reflect the adjustment to the control.

To reflect the adjustment to the control, click the "Parameter SET" button.

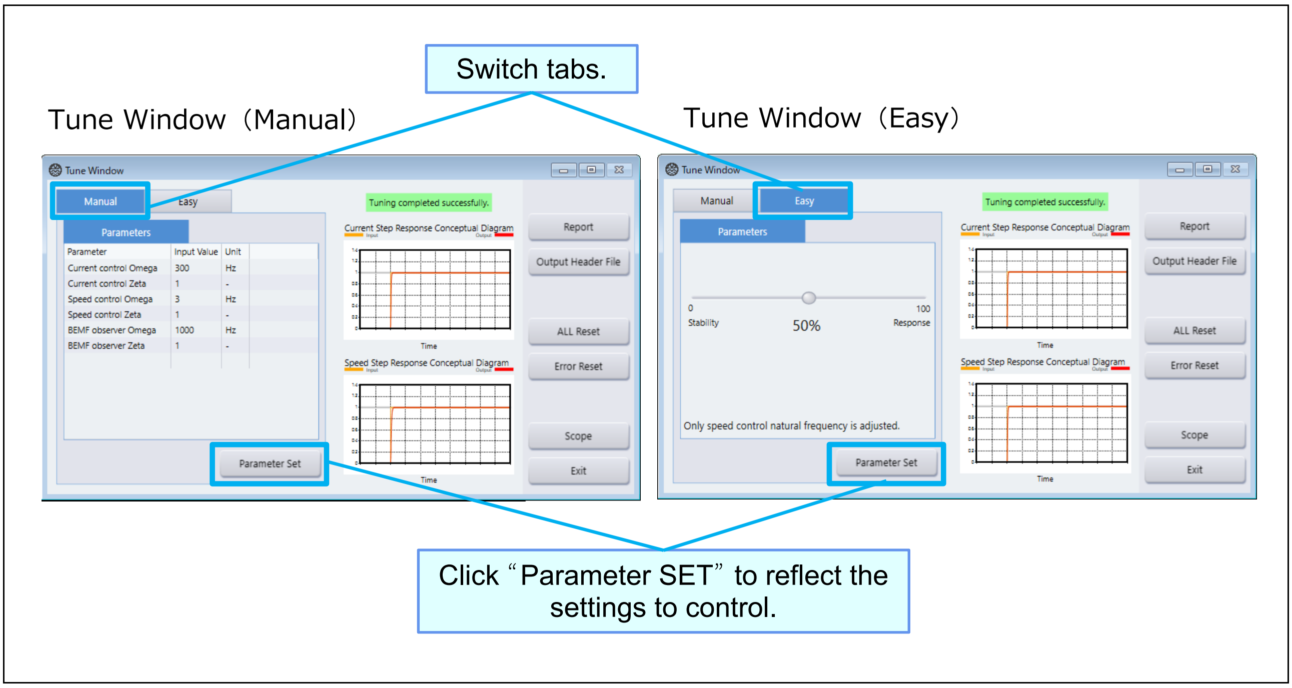


Figure 16-10 Manual and Easy Tab

Table 16-2 Tuning parameter list (Manual tab)

Display	Adjustable parameter	Remark
Current control Omega	Natural frequency related to current control [Hz]	
Current control Zeta	Damping coefficient related to current control	
Speed control Omega	Natural frequency related to speed control [Hz]	
Speed control Zeta	Damping coefficient related to speed control	
BEMF observer Omega	Natural frequency related induced-voltage estimation [Hz]	Sensorless speed control only
BEMF observer Zeta	Damping coefficient related induced-voltage estimation	Sensorless speed control only
Position control Omega	Natural frequency related to position control [Hz]	Encoder position control only
Encoder counts per revolution	Encoder pulse counts per revolution	Encoder position control only

16.7.3 Reset Function (Tune Window)

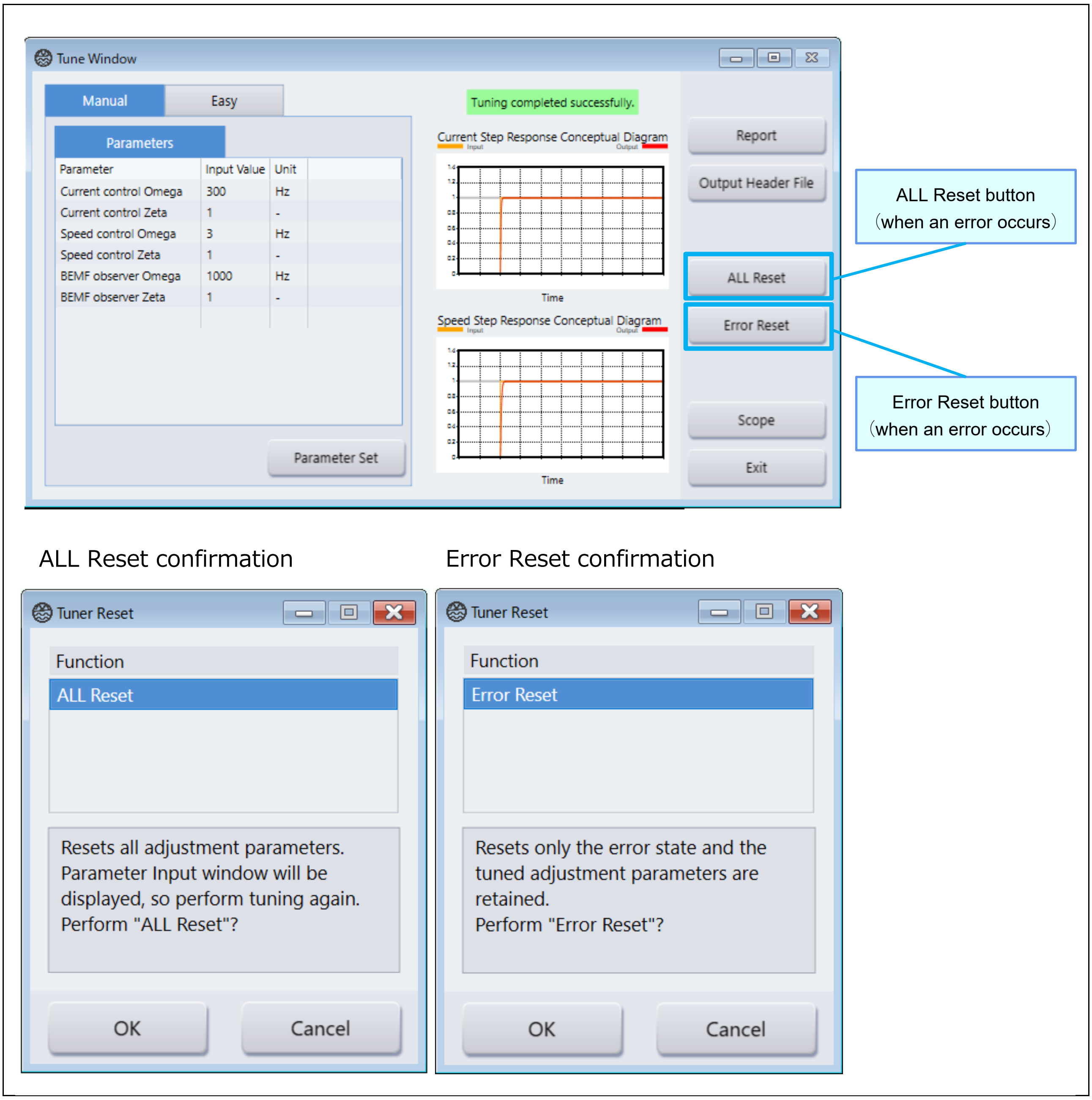
If an error occurs in the drive program after tuning, select the type of reset to perform. Click the "ALL Reset" or "Error Reset" button in the Tune Window to display the reset screen.

- ALL Reset

  Resets all adjustment parameters. Parameter Input window will be displayed, so perform tuning again.

- Error Reset

  Resets only the error state and the tuned adjustment parameters are retained.



Parameter	Input Value	Unit
Current control Omega	300	Hz
Current control Zeta	1	-
Speed control Omega	3	Hz
Speed control Zeta	1	-
BEMF observer Omega	1000	Hz
BEMF observer Zeta	1	-

Figure 16-11 Reset function

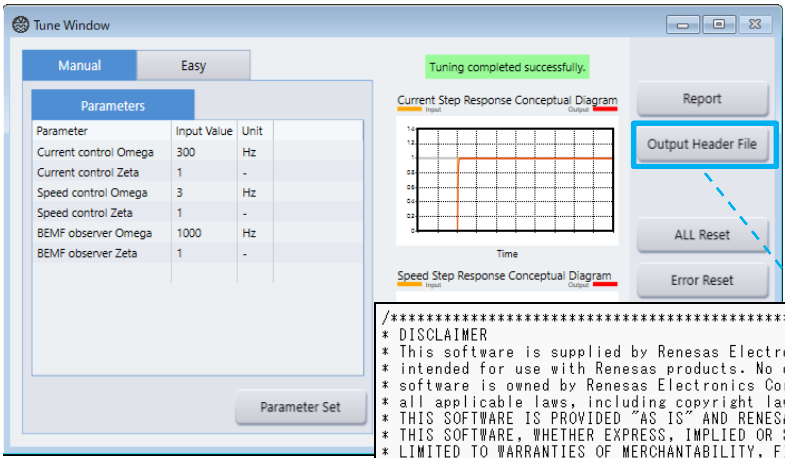
16.7.4 Header Output (Tune Window)

(1) Header file output

The tuning result can be output in the header file format of the Renesas motor control program.

Click the "Output Header File" button on Tune Window, and the saving window for "r_mtr_control_parameter.h" and "r_mtr_motor_parameter.h" will be displayed.

Tune Window



Example of an output header file (r_mtr_motor_parameter.h)

```
/**********************************************************************************
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only
* intended for use with Renesas products. No other uses are authorized. This
* software is owned by Renesas Electronics Corporation and is protected under
* all applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
* THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
* LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
* ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
* BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software
* and to discontinue the availability of this software. By using this software,
* you agree to the additional terms and conditions found by accessing the
* following link:
* http://www.renesas.com/disclaimer
*
* Copyright (C) 2020 Renesas Electronics Corporation. All rights reserved.
**********************************************************************************/

/**********************************************************************************
* File Name   : r_mtr_motor_parameter.h
* Description : Definition of the target motor parameters
*                   (Renesas Motor Workbench Output file)
**********************************************************************************/
/**********************************************************************************
* Date : 2025.10.29 09:59
**********************************************************************************/

/* Guard against multiple inclusion */
#ifndef R_MTR_MOTOR_PARAMETER_H
#define R_MTR_MOTOR_PARAMETER_H

/**********************************************************************************
* Macro definitions
**********************************************************************************/
#define    MTR_MOTOR_PARAMETER    (1)

/* Target motor definitions */
#define    MP_POLE_PAIRS          (4)               /* Number of pole pairs */
#define    MP_RESISTANCE          (1.278677f)       /* Resistance [ohm] */
#define    MP_D_INDUCTANCE        (0.002035681f)    /* D-axis inductance [H] */
#define    MP_Q_INDUCTANCE        (0.002070462f)    /* Q-axis inductance [H] */
#define    MP_MAGNETIC_FLUX       (0.01169281f)     /* Permanent magnetic flux [Wb] */
#define    MP_ROTOR_INERTIA       (3.310325E-06f)   /* Rotor inertia [kgm^2] */
#define    MP_NOMINAL_CURRENT_RMS (1.67f)           /* Nominal current [Arms] */

#endif /* R_MTR_MOTOR_PARAMETER_H */
```

Figure 16-12 Output Header File button

For the encoder position control program, refer to the application note and specify the encoder pulse count separately when including into the sample program.

(2) Reflection on Sample Programs

If the output header file has already been incorporated into the sample program, the output parameters can be reflected by overwriting the header file.
However, the sample program provided for MCI-HV-1 (e.g., r01an7141, RA6T2_MCIHV1_PM_LESS_FOC_PFC_E2S_V***) requires manual reflection.
The table below shows the macro name that manages each parameter. Please change the values on the sample program side with reference to the following.

Table 16-3 Parameter macro comparison table for motor

File/Macro Description	Tuner tool output file	sample program e.g., RA6T2_MCIHV1_PM_LESS_FOC_PFC_E2S_V***
Header file	r_mtr_motor_parameter.h	r_motor_targetmotor_cfg.h
Pole pairs	MP_POLE_PAIRS	MOTOR_CFG_POLE_PAIRS
Resistance [ohm]	MP_RESISTANCE	MOTOR_CFG_RESISTANCE
Inductance on d-axis [H]	MP_D_INDUCTANCE	MOTOR_CFG_D_INDUCTANCE
Inductance on q-axis [H]	MP_Q_INDUCTANCE	MOTOR_CFG_Q_INDUCTANCE
Magnetic flux [wb]	MP_MAGNETIC_FLUX	MOTOR_CFG_MAGNETIC_FLUX
Inertia of rotor [kg m2]	MP_ROTOR_INERTIA	MOTOR_CFG_ROTOR_INERTIA
Rated current [A]	MP_NOMINAL_CURRENT_RMS	MOTOR_CFG_NOMINAL_CURRENT_RMS

Table 16-4 Parameter macro comparison table for motor control

File/Macro Description		Tuner tool output file	sample program RA6T2_MCIHV1_PM_LESS _FOC_PFC_E2S_V***
Header file		r_mtr_control_parameter.h	r_motor_module_cfg.h
	Current control system natural frequency [Hz]	CP_CURRENT_OMEGA	CURRENT_CFG_OMEGA
	Current control system damping factor	CP_CURRENT_ZETA	CURRENT_CFG_ZETA
	Speed control system natural frequency [Hz]	CP_SPEED_OMEGA	SPEED_CFG_OMEGA
	Speed control system damping coefficient	CP_SPEED_ZETA	SPEED_CFG_ZETA
	Induced voltage estimation system natural frequency [Hz]	CP_E_OBS_OMEGA	CURRENT_CFG_E_OBS_OMEGA
	Induced voltage estimation system damping coefficient	CP_E_OBS_ZETA	CURRENT_CFG_E_OBS_ZETA
	Position estimation system natural frequency [Hz]	CP_PLL_EST_OMEGA	CURRENT_CFG_PLL_EST_OMEGA
	Position estimation system damping coefficient	CP_PLL_EST_ZETA	CURRENT_CFG_PLL_EST_ZETA
	d-axis current command value subtraction start speed (machine angle) [rpm]	CP_ID_DOWN_SPEED_RPM	SENSORLESS_VECTOR_ID_DOWN_SPEED_RPM
	d-axis current command value addition start speed (machine angle) [rpm]	CP_ID_UP_SPEED_RPM	SENSORLESS_VECTOR_ID_UP_SPEED_RPM
	Speed limit value [rpm]	CP_OVERSPEED_LIMIT_RPM	SPEED_CFG_SPEED_LIMIT_RPM
	d-axis current command value at open loop [A] d-axis current command value at open loop [A]	CP_OL_ID_REF	CURRENT_CFG_REF_ID_OPENLOOP
Header file		Same as above	r_motor_targetmotor_cfg.h
	Maximum speed [rpm]	CP_MAX_SPEED_RPM	MOTOR_CFG_MAX_SPEED_RPM

16.7.5 PDF Output (Tune Window)

Click the “Report” button on “Tune Window” to display the Result Report window. You can output the tuning result in a PDF file with the “Output” button on this window.

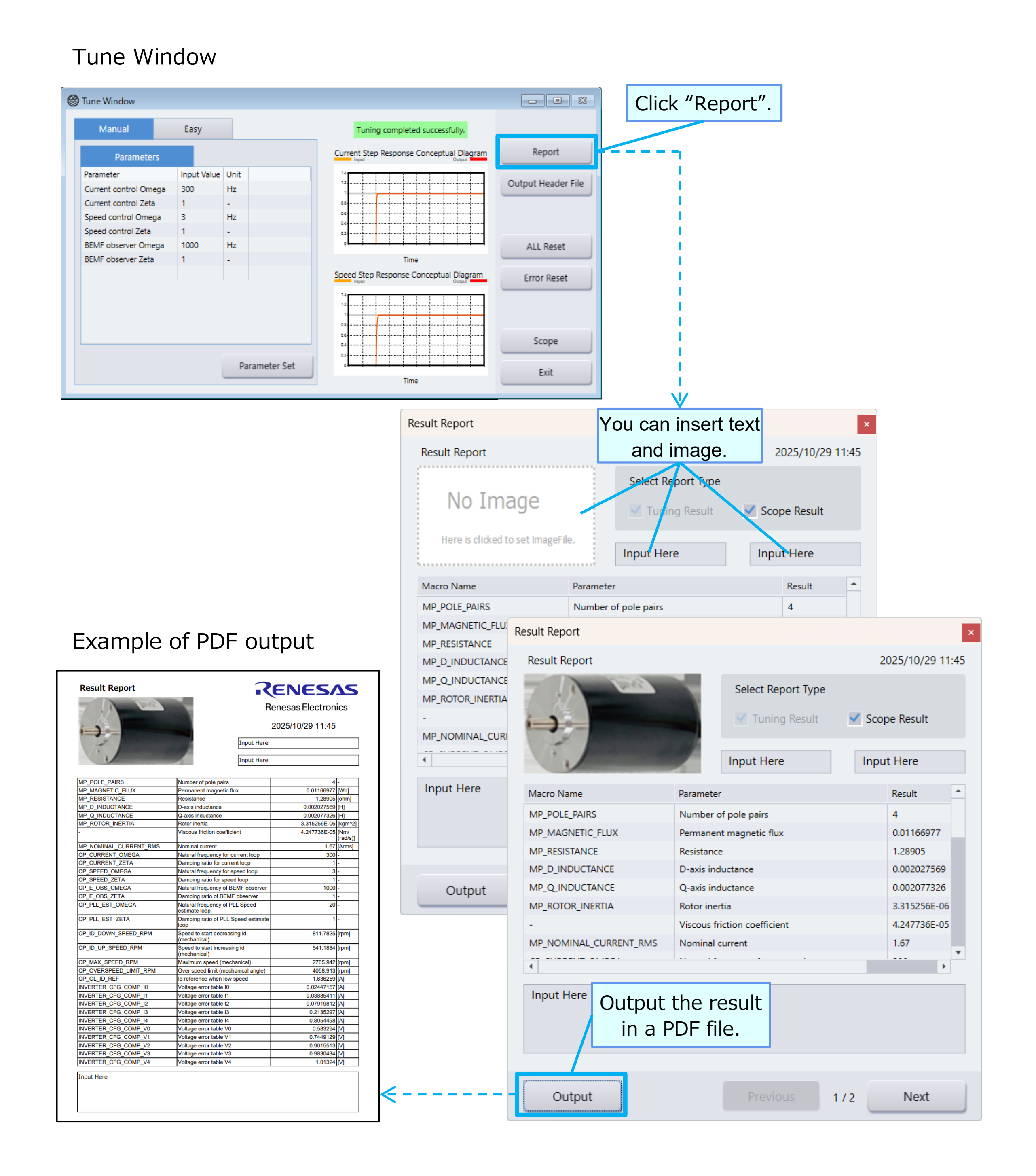


Figure 16-13 Output button (PDF output)

The parameter list output as tuning results is shown below.

Table 16-5 Define Directives

Definition name	Description	Unit
MP_POLE_PAIRS	Pole pairs	-
MP_MAGNETIC_FLUX	Magnetic flux	[wb]
MP_RESISTANCE	Resistance	[ohm]
MP_D_INDUCTANCE	Inductance on d-axis	[H]
MP_Q_INDUCTANCE	Inductance on q-axis	[H]
MP_ROTOR_INERTIA	Inertia of rotor	[kg m2]
MP_NOMINAL_CURRENT_RMS	Rated current	[A]
-	Viscous friction coefficient	[Nm/(rad/s)]
CP_CURRENT_OMEGA	Current control system natural frequency	[Hz]
CP_CURRENT_ZETA	Current control system damping factor	-
CP_SPEED_OMEGA	Speed control system natural frequency	[Hz]
CP_SPEED_ZETA	Speed control system damping coefficient	-
CP_E_OBS_OMEGA	Induced voltage estimation system natural frequency	[Hz]
CP_E_OBS_ZETA	Induced voltage estimation system damping coefficient	-
CP_PLL_EST_OMEGA	Position estimation system natural frequency	[Hz]
CP_PLL_EST_ZETA	Position estimation system damping coefficient	-
CP_ID_DOWN_SPEED_RPM	d-axis current command value subtraction start speed (machine angle)	[rpm]
CP_ID_UP_SPEED_RPM	d-axis current command value addition start speed (machine angle)	[rpm]
CP_MAX_SPEED_RPM	Maximum speed	[rpm]
CP_OVERSPEED_LIMIT_RPM	Speed limit value	[rpm]
CP_OL_ID_REF	d-axis current command value at open loop	[A]
INVERTER_CFG_COMP_I0	Coefficient for compensation of the voltage error	[A]
INVERTER_CFG_COMP_I1	Coefficient for compensation of the voltage error	[A]
INVERTER_CFG_COMP_I2	Coefficient for compensation of the voltage error	[A]
INVERTER_CFG_COMP_I3	Coefficient for compensation of the voltage error	[A]
INVERTER_CFG_COMP_I4	Coefficient for compensation of the voltage error	[A]
INVERTER_CFG_COMP_V0	Coefficient for compensation of the voltage error	[V]
INVERTER_CFG_COMP_V1	Coefficient for compensation of the voltage error	[V]
INVERTER_CFG_COMP_V2	Coefficient for compensation of the voltage error	[V]
INVERTER_CFG_COMP_V3	Coefficient for compensation of the voltage error	[V]
INVERTER_CFG_COMP_V4	Coefficient for compensation of the voltage error	[V]

16.7.6 Waveform Information Display (Scope Window)

When you click the “View Open” button on Scope Window, you can check the information about the waveform display.

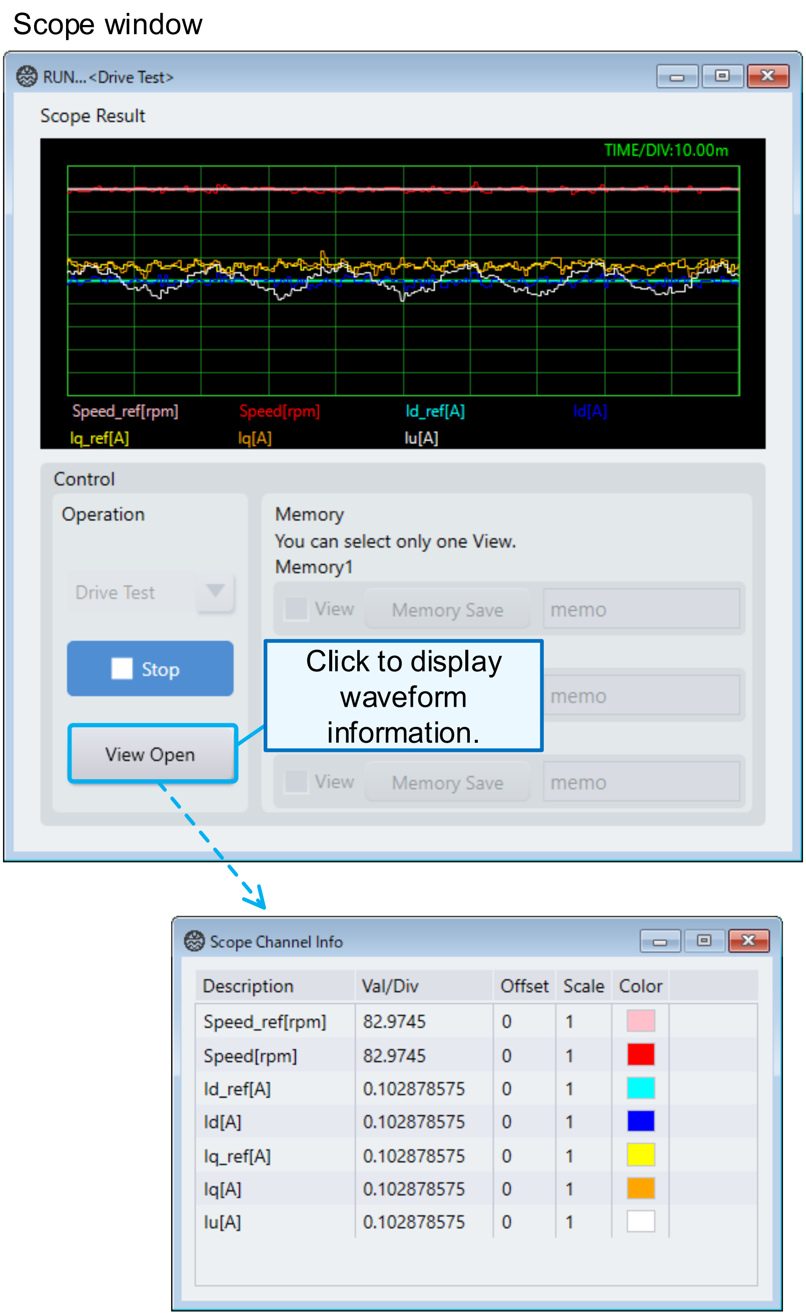


Figure 16-14 View Open button

16.7.7 Memory Function (Scope Window)

In Scope Window, up to three waveforms of motor driving can be memorized, and individual waveforms can be switched and displayed.

When you click the "Memory Save" button, the waveform is memorized and the label of the button changes to "Memory Clear". Click it again to clear the memory.

When you select the View checkbox, the memorized waveform is displayed.

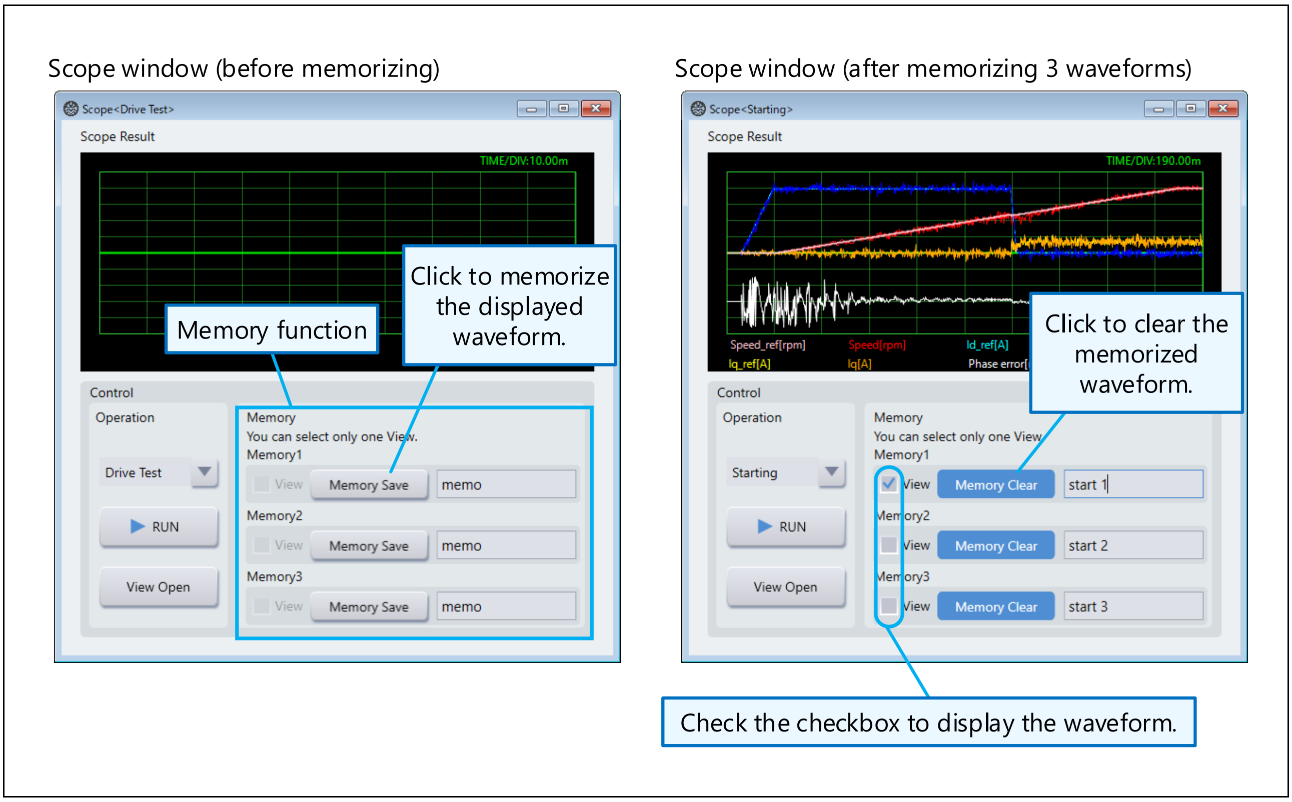


Figure 16-15 Memory function

17.Communication Library

17.1 Overview

A motor control development support tool, Renesas Motor Workbench (hereafter, "RMW"), communicates the variable information with MCUs. RMW and a MCU can easily communicate by including a communication library or built-in type communication library into the MCU program.

The built-in type communication library can perform the same functions as communication libraries, as well as command processing and scope-processing of RMW.

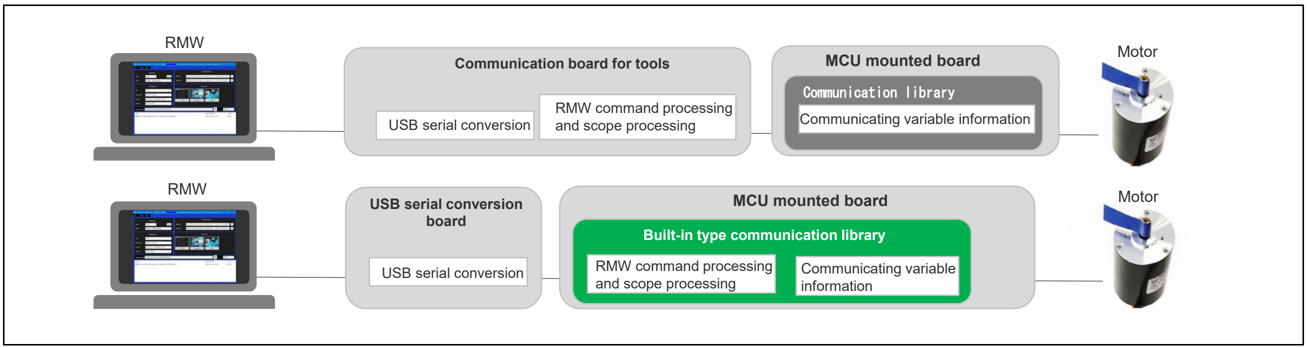


Figure 17-1 Overview of Communication Library

17.2 Features

The communication library enables the communication between RMW and the MCU. Additionally, the built-in type communication library enables the communication between RMW and the MCU using a commercially available USB serial conversion board, instead of a communication board for tools.

Since the built-in type communication library uses the RAM area of the motor control MCU, the number of data displayed in Scope Window is limited. However, it is ideal for simple viewing and debugging. If you want to display faster sampling data, we recommend you to use a communication board for tools and communication libraries.

Table 17-1 Differences Between Communication Library and Built-in Type Communication Library

Item		Communication library	Built in type communication library
Recommended application		Displaying faster sampling data	Simple displaying and debugging
Supported MCU	RX Family	RX13T, RX14T, RX23T, RX24T, RX24U, RX26T, RX66T, RX72T, RX72M	-
	RL78 Family	RL78/G14, RL78/G1F	-
	RA Family	RA6T1, RA6T2, RA6T3, RA4T1, RA8T1, RA8T2, RA2T1	RA6T2, RA6T3, RA4T1, RA8T1, RA8T2
Required communication board		MC-COM	Commercially available serial USB conversion board
Number of data displayed in Scope Window		100,000	• 1,024 (RA6T2, RA8T1, RA8T2) • 512 (RA6T3, RA4T1)
Sampling period		Max. 20us/4ch	Any period
Available function		All	All
RAM capacity used in Scope Window		-	• 32KB (RA6T2, RA8T1, RA8T2) • 8KB (RA6T3, RA4T1)
Library role		Communicates variable information between PC and MCU.	Communicates variable information between PC and MCU. RMW command processing and scope processing.
Library specification		Same specifications for both communication library and built-in type communication library • The argument of the initialization function differs depending on the target MCU. • Available pins differ depending on the target MCU.	

17.3 Communication Library for each MCU

To use Renesas Motor Workbench, it is necessary to include the communication library into a user program. The communication library is provided in the package of Renesas Motor Workbench (in "communication library" folder). Table 4-1 shows the communication libraries provided for each MCU.

Table 17-2 Communication library (for RX) 1/3

Supported CPU	RX23T	RX24T	RX24U
File	ics_RX23T.obj ics_RX23T.h	ics_RX24T.obj ics_RX24T.h	ics_RX24U.obj ics_RX24U.h
Communication rate	0.5 Mbps to 5 Mbps		
Port	SCI1 TXD1:PD3 RXD1:PD5 SCI5 TXD5:PB5 RXD5:PB6 SCI5 TXD5:PB2 RXD5:PB1	SCI1 TXD1:PD3 RXD1:PD5 SCI5 TXD5:PB5 RXD5:PB6 SCI6 TXD6:PB2 RXD6:PB1 SCI6 TXD6:PB0 RXD6:PA5 SCI6 TXD6:P81 RXD6:P80	SCI1 TXD1:PD3 RXD1:PD5 SCI5 TXD5:PB5 RXD5:PB6 SCI6 TXD6:PB2 RXD6:PB1 SCI6 TXD6:PB0 RXD6:PA5 SCI6 TXD6:P81 RXD6:P80
Supported variable type	8bit unsigned integer (number display/setting/waveform display) 8bit signed integer (number display/setting/waveform display) 16bit unsigned integer (number display/setting/waveform display) 16bit signed integer (number display/setting/waveform display) 32bit unsigned integer (number display/setting/waveform display) 32bit signed integer (number display/setting/waveform display) 32bit IEEE754 floating point (number display/setting/waveform display)		
CPU resource to use	SCIx RX, SCIx TX, DTC		

Table 17-3 Communication library (for RX) 2/3

Supported CPU	RX66T	RX72T	RX72M
File	ICS2_RX66T.lib ICS2_RX66T.h	ICS2_RX72T.lib ICS2_RX72T.h	ICS2_RX72M.lib ICS2_RX72M.h
Communication rate	0.5 Mbps to 7.5 Mbps		
Port	SCI1 TXD1:PD3 RXD1:PD5 SCI5 TXD5:PB5 RXD5:PB6 SCI6 TXD6:PB0 RXD6:PB1	SCI5 TXD5:PB5 RXD5:PB6 SCI6 TXD6:PB0 RXD6:PB1 SCI8 TXD8:PC1 RXD8:PC0 SCI12 TXD12:PB5 RXD12:PB6	SCI1 TXD5:PF0 RXD5:PF2 SCI2 TXD6:P50 RXD6:P52 SCI3 TXD8:P23 RXD8:P25 SCI4 TXD8:PB1 RXD8:PB0 SCI5 TXD5:PA4 RXD5:PA3 SCI6 TXD6:P00 RXD6:P01 SCI6 TXD8:PB1 RXD8:PB0 SCI8 TXD12:PJ2 RXD12:PC6 SCI8 TXD12:PJ2 RXD12:PJ1
Supported variable type	8bit unsigned integer (number display/setting/waveform display) 8bit signed integer (number display/setting/waveform display) 16bit unsigned integer (number display/setting/waveform display) 16bit signed integer (number display/setting/waveform display) 32bit unsigned integer (number display/setting/waveform display) 32bit signed integer (number display/setting/waveform display) 32bit IEEE754 floating point (number display/setting/waveform display)		
CPU resource to use	SCIx RX, SCIx TX, DTC		

Table 17-4 Communication library (for RX) 3/3

Supported CPU	RX13T	RX14T	RX26T
File	ICS2_RX13T.lib ICS2_RX13T.h	ICS2_RX14T.lib ICS2_RX14T.h	ICS2_RX26T.lib ICS2_RX26T.h
Communication rate	0.5 Mbps to 4 Mbps	0.5 Mbps to 6 Mbps	0.5 Mbps ～ 7.5 Mbps
Port	SCI1 TXD1:PD3 RXD1:PD5 SCI1 TXD1:PB6 RXD1:PB7 SCI5 TXD5:PB6 RXD5:PB7 SCI5 TXD5:PB2 RXD5:PB1 SCI5 TXD5:P23 RXD5:P24 SCI12 TXD12:PB0 RXD12:P94	SCI1 TXD1:PB5 RXD1:PB6 SCI5 TXD5:P23 RXD5:P24	SCI1 TXD1:PD3 RXD1:PD5 SCI5 TXD5:PD7 RXD5:PE0 SCI5 TXD5:PB5 RXD5:PB6 SCI6 TXD6:P81 RXD6:P80 SCI6 TXD6:PB2 RXD6:PB1 SCI12 TXD12:PD4 RXD12:PD6 SCI12 TXD12:P01 RXD12:P00 SCI12 TXD12:P81 RXD12:P80 SCI12 TXD12:P23 RXD12:P22 SCI12 TXD12:PB5 RXD12:PB6
Supported variable type	8bit unsigned integer (number display/setting/waveform display) 8bit signed integer (number display/setting/waveform display) 16bit unsigned integer (number display/setting/waveform display) 16bit signed integer (number display/setting/waveform display) 32bit unsigned integer (number display/setting/waveform display) 32bit signed integer (number display/setting/waveform display) 32bit IEEE754 floating point (number display/setting/waveform display)		
CPU resource to use	SCIx RX, SCIx TX, DTC		

Table 17-5 Communication library (for RL) 1/2

Supported CPU	RL78/G14	RL78/G1F
File	ics2_RL78G14.Lib ics2_RL78G14.h	ics2_RL78G1F.Lib ics2_RL78G1F.h
Communication rate	0.5 Mbps to 5.33 Mbps	
Port	SCI0 TXD0:P51, RXD0:P50 SCI0 TXD0:P12, RXD0:P11 SCI0 TXD0:P17, RXD0:P16 SCI1 TXD0:P00, RXD0:P01 SCI1 TXD0:P02, RXD0:P03 SCI1 TXD0:P72, RXD0:P73 SCI1 TXD0:P77, RXD0:P76 SCI1 TXD0:P82, RXD0:P81 SCI2 TXD2:P13, RXD2:P14 SCI2 TXD0:P77, RXD0:P76 SCI3 TXD0:P144, RXD0:P143	SCI0 TXD0:P51, RXD0:P50 SCI0 TXD0:P17, RXD0:P16 SCI1 TXD0:P00, RXD0:P01 SCI1 TXD0:P02, RXD0:P03 SCI1 TXD0:P72, RXD0:P73 SCI1 TXD0:P77, RXD0:P76 SCI2 TXD2:P13, RXD2:P14
Supported variable type	8bit unsigned integer (number display/setting/waveform display) 8bit signed integer (number display/setting/waveform display) 16bit unsigned integer (number display/setting/waveform display) 16bit signed integer (number display/setting/waveform display) 32bit unsigned integer (number display/setting) 32bit signed integer (number display/setting) 8bit bool (number display/setting) 8bit logic (number display/setting)	
CPU resource to use	SCIx RX、SCIx TX、DTC	

Table 17-6 Communication library (for RL) 2/2

Supported CPU	RL78/G24
File	ics2_RL78G24.Lib ics2_RL78G24.h
Communication rate	0.5 Mbps to 8 Mbps
Port	SCI0 TXD0:P51，RXD0:P50 SCI0 TXD0:P12，RXD0:P11 SCI0 TXD0:P17，RXD0:P16 SCI1 TXD0:P00，RXD0:P01 SCI1 TXD0:P02，RXD0:P03 SCI1 TXD0:P30，RXD0:P31 SCI1 TXD0:P72，RXD0:P73 SCI2 TXD2:P10，RXD2:P11 SCI2 TXD2:P13，RXD2:P14 SCI2 TXD0:P77，RXD0:P76
Supported variable type	8bit unsigned integer (number display/setting/waveform display) 8bit signed integer (number display/setting/waveform display) 16bit unsigned integer (number display/setting/waveform display) 16bit signed integer (number display/setting/waveform display) 32bit unsigned integer (number display/setting) 32bit signed integer (number display/setting) 8bit bool (number display/setting) 8bit logic (number display/setting)
CPU resource to use	SCIx RX、SCIx TX、DTC

Table 17-7 Communication library (for RA) 1/2

Supported CPU	RA6T1	RA6T2	RA6T3
File	ICS2_RA6T1.o ICS2_RA6T1.h	ICS2_RA6T2.o ICS2_RA6T2.h	ICS2_RA6T3.o ICS2_RA6T3.h
Communication rate	0.5 Mbps to 15.0 Mbps	0.5 Mbps to 20.0 Mbps	0.5 Mbps ～ 16.67 Mbps
Port	SCI0 TXD0:P101 RXD0:P100 SCI4 TXD4:P205 RXD4:P206 SCI9 TXD9:P109 RXD9:P110	SCI9 TXD9:PD05 RXD9:PD06	SCI0 TXD0:P101 RXD0:P100 SCI0 TXD0:P213 RXD0:P212 SCI0 TXD0:P411 RXD0:P410 SCI9 TXD9:P110 RXD9:P109
Supported variable type	8bit unsigned integer (number display/setting/waveform display) 8bit signed integer (number display/setting/waveform display) 16bit unsigned integer (number display/setting/waveform display) 16bit signed integer (number display/setting/waveform display) 32bit unsigned integer (number display/setting/waveform display) 32bit signed integer (number display/setting/waveform display) 32bit IEEE754 floating point (number display/setting/waveform display)		
CPU resource to use	SCIx RX, SCIx TX	SCIx RX, SCIx TX	SCIx RX, SCIx TX

Table 17-8 Communication library (for RA) 2/2

Supported CPU	RA4T1	RA8T1	RA8T2
File	ICS2_RA4T1.o ICS2_RA4T1.h	ICS2_RA8T1.o ICS2_RA8T1.h	ICS2_RA8T2.o ICS2_RA8T2.h
Communication rate	0.5 Mbps ～ 16.67 Mbps	0.5 Mbps ～ 20.0 Mbps	0.5 Mbps ～ 20.0 Mbps
Port	SCI0 TXD0:P101 RXD0:P100 SCI0 TXD0:P213 RXD0:P212 SCI0 TXD0:P411 RXD0:P410 SCI9 TXD9:P110 RXD9:P109	SCI1 TXD1:P213 RXD1:P212 SCI2 TXD2:PA03 RXD2:PA02 SCI3 TXD3:P409 RXD3:P408 SCI4 TXD4:P714 RXD4:P715	SCI0 TXD0:P609 RXD0:P610 SCI1 TXD1:P400 RXD1:P401 SCI1 TXD1:P707 RXD1:P706 SCI4 TXD4:P415 RXD4:P414 SCI6 TXD6:P301 RXD6:P302 SCI6 TXD6:PC14 RXD6:PC13 SCI7 TXD7:P809 RXD7:P808 SCI9 TXD9:P209 RXD9:P208
Supported variable type	8bit unsigned integer (number display/setting/waveform display) 8bit signed integer (number display/setting/waveform display) 16bit unsigned integer (number display/setting/waveform display) 16bit signed integer (number display/setting/waveform display) 32bit unsigned integer (number display/setting/waveform display) 32bit signed integer (number display/setting/waveform display) 32bit IEEE754 floating point (number display/setting/waveform display)		
CPU resource to use	SCIx RX, SCIx TX	SCIx RX, SCIx TX	SCIx RX, SCIx TX

Table 17-9 Communication library (for RA) 2/2

Supported CPU	RA2T1
File	ICS2_RA2T1.o ICS2_RA2T1.h
Communication rate	0.5 Mbps ～ 5.0 Mbps
Port	SCI0 TXD0:P101 RXD0:P100 SCI0 TXD0:P101 RXD0:P104 SCI0 TXD0:P206 RXD0:P100
Supported variable type	8bit unsigned integer (number display/setting/waveform display) 8bit signed integer (number display/setting/waveform display) 16bit unsigned integer (number display/setting/waveform display) 16bit signed integer (number display/setting/waveform display) 32bit unsigned integer (number display/setting/waveform display) 32bit signed integer (number display/setting/waveform display)
CPU resource to use	SCIx RX, SCIx TX

17.4 Built-in Type Communication Library

The RA6T2, RA6T3, RA4T1, RA8T1 and RA8T2 supports built-in type communication libraries. When using the built-in type communication library, you can use a commercially available USB serial conversion module (isolated type).

The built-in type communication library is included in the package of Renesas Motor Workbench (in "RA6T2", "RA6T3", "RA4T1", "RA8T1" and "RA8T2" folder in "communication library").

Table 17-10 Built-in type communication library (for RA) 1/2

Supported CPU	RA6T2	RA6T3	RA4T1
File	ICS2_RA6T2_Built_in.o	ICS2_RA6T3_Built_in.o	ICS2_RA4T1_Built_in.o
Communication rate	-	-	-
Port	SCI9 TXD9:PD05 RXD9:PD06	SCI0 TXD0:P101 RXD0:P100 SCI0 TXD0:P213 RXD0:P212 SCI0 TXD0:P411 RXD0:P410 SCI9 TXD9:P110 RXD9:P109	SCI0 TXD0:P101 RXD0:P100 SCI0 TXD0:P213 RXD0:P212 SCI0 TXD0:P411 RXD0:P410 SCI9 TXD9:P110 RXD9:P109
Supported variable type	8bit unsigned integer (number display/setting/waveform display) 8bit signed integer (number display/setting/waveform display) 16bit unsigned integer (number display/setting/waveform display) 16bit signed integer (number display/setting/waveform display) 32bit unsigned integer (number display/setting/waveform display) 32bit signed integer (number display/setting/waveform display) 32bit IEEE754 floating point (number display/setting/waveform display)		
CPU resource to use	SCIx RX, SCIx TX	SCIx RX, SCIx TX	SCIx RX, SCIx TX

Table 17-11 Built-in type communication library (for RA) 2/2

Supported CPU	RA8T1	RA8T2
File	ICS2_RA8T1_Built_in.o	ICS2_RA8T2_Built_in.o
Communication rate	-	-
Port	SCI1 TXD1:P213 RXD1:P212 SCI2 TXD2:PA03 RXD2:PA02 SCI3 TXD3:P409 RXD3:P408 SCI4 TXD4:P714 RXD4:P715	SCI0 TXD0:P609 RXD0:P610 SCI1 TXD1:P400 RXD1:P401 SCI1 TXD1:P707 RXD1:P706 SCI4 TXD4:P415 RXD4:P414 SCI6 TXD6:P301 RXD6:P302 SCI6 TXD6:PC14 RXD6:PC13 SCI7 TXD7:P809 RXD7:P808 SCI9 TXD9:P209 RXD9:P208 SCI9 TXD9:PA14 RXD9:PA12
Supported variable type	8bit unsigned integer (number display/setting/waveform display) 8bit signed integer (number display/setting/waveform display) 16bit unsigned integer (number display/setting/waveform display) 16bit signed integer (number display/setting/waveform display) 32bit unsigned integer (number display/setting/waveform display) 32bit signed integer (number display/setting/waveform display) 32bit IEEE754 floating point (number display/setting/waveform display)	
CPU resource to use	SCIx RX, SCIx TX	SCIx RX, SCIx TX

17.5 How to Set User Program

17.5.1 DTC

Renesas Motor Workbench operates on a DTC (standard address mode), excluding RA MCUs. Therefore, it is necessary to define the DTC table in a user program.

Reserve the area for the DTC table within a program (within a file to call initialization functions described later).

- For RX microcontrollers
  Assign the DTC table to an address in the RAM so that the lower 12bit becomes 0.
- For RL microcontrollers
  Assign the DTC table to an address so that the lower 8bit becomes 0.

For code sample, refer to 17.5.4 How to Use Library Functions

When using an emulator such as E2, make sure that the user RAM area and the DTC table area do not overlap.

17.5.2 Interruption

Specify the functions (ics_int_sci_eri(), ics_int_sci_rxi()) for the user program interrupt vectors as shown in the code sample below (except for RA MCUs).

When using a project generated by the Renesas standard compiler with RX MCUs, specify them within intprg.c.

```
void Excep_SCI1_ERI1 (void){ ics_int_sci_eri(); }
void Excep_SCI1_RXI1 (void){ ics_int_sci_rxi(); }
void Excep_SCI5_ERI5 (void){ ics_int_sci_eri(); }
void Excep_SCI5_RXI5 (void){ ics_int_sci_rxi(); }
```

Figure 17-2 Code sample for communication interruption functions

17.5.3 Specification of Library Functions

In the communication library, the following two library functions are provided.

- void ics2_init(void* addr, char port, char level, char speed, char mode)
- void ics2_watchpoint(void)

The specifications of each function are shown below.

Table 17-12 Library function (ics2_init) (1/2)

Function name	void ics2_init(void* addr, char port, char level, char speed, char mode)	
Return value	void	None
Argument	void* addr	The start address of a DTC vector table to be used: Users must reserve DTC vector table before calling this function. *No settings of this argument for RA series
	char port	Set SCI port number and pins used for SCI.
	char level	Set SCI interrupt level: Set the appropriate interrupt level for the system, since appx.10 μsec interrupt occurs within minimum 2 msec intervals. *No settings of this argument for RA series
	char speed	Set communication rate: Set the value of the communication rate by the following expression. Communication rate = CLOCK/ (x × (speed + 1)) [Mbps] Communication rate (RA2T1) = CLOCK/ (x × (speed + 1) × (256/ 240)) [Mbps] CLOCK: The frequency of clock source supplied to SCI [MHz] RX: PCLK, RA: SCISPICLK or PCLK *For details, see the MCU's hardware manual. x: The fixed value for each MCU For RX, RA6T1:x = 8 For RL78 :x = 2 For RA6T2, RA4T1, RA6T3, RA8T1, RA8T2, RA2T1:x = 6 Ex.) When setting 1.0 Mbps for RX23T CLOCK = PCLKB = 40 [MHz], x = 8 then 1.0Mbps = 40MHz / (8 * (speed + 1) -> speed : 4 .

Table 17-13 Library function (ics2_init) (2/2)

	char mode	• mode 1 (32bits 8channels Action of 2 times transfer) In this mode, when the function "ics2_watchpoint()" is called once, 4 channels data are transferred with extraction of specified 8 channels data as 8bits, 16bits, and 32bits for wave form display. At next time when "ics2_watchpoint()" is called, other remained 4 channels data are transferred without new extraction. That is to say, in case of "32bits 8channels mode", 8 channels data are transferred with 2 times call of "ics2_watchpoint()". • mode 2 (32bits 4channels Action of 1 time transfer) In this mode, when the function "ics2_watchpoint()" is called, 4 channels data are transferred with extraction of specified 4 channels data as 8bits, 16bits, and 32bits for wave form display. That is to say, in case of "32bits 4channels mode", 4 channels data are transferred with each call of "ics2_watchpoint()". It is impossible to display wave form over 5 channels. • mode 3 (32bits 12channels Action of 3 times transfer) In this mode, when the function "ics2_watchpoint()" is called once, 4 channels data are transferred with extraction of specified 12 channels data as 8bits, 16bits, and 32bits for wave form display. At next time when "ics2_watchpoint()" is called, remained next 4 channels data are transferred without new extraction. And at 3rd time of call "ics2_watchpoint()", last remained 4 channels data are transferred. That is to say, in case of "32bits 12channels mode", 12 channels data are transferred with 3 times call of "ics2_watchpoint()".
Function	Initialization process	

Table 17-14 Library function (ics2_watchpoint)

Function name	void ics2_watchpoint(void)	
Return value	void	None
Argument	void	None
Function	Data transfer: When the communication rate is BR [Mbps], the minimum sampling cycle is 70 + (180/BR) [μsec], and therefore the data transfer function is called at intervals greater than this value. For example, 250 [μsec] intervals or greater are required for communication rate at 1.0 [Mbps]. Note that when using a MC-COM (communication board for tools), the minimum sampling cycle is 10 + (180/BR) [μsec].	

17.5.4 How to Use Library Functions

(1) Calling on RX microcontroller

(a) Calling initialization function “ics2_init ()”

Add the initialization function “ics2_init” to the initialization process part of a program as shown in the sample code below. Specify the argument based on the following rules.

The first argument:	specify the start address (0 for the lower 12 bit) of a DTC table
The second argument:	select the port used from “ics_****.h” (see Table 17-2)
The third argument:	specify the tool interrupt level
The fourth argument:	specify the communication rate
The fifth argument:	specify the transfer mode

```
#include "ics_RX23T.h"

#pragma section DTCTBL
unsigned long dtc_table[256];     /* caution alignment 0x000 */
#pragma section

void main(void)
{
        ics2_init((void*) dtc_table, port, level, speed, mode);
}
```

Figure 17-3 Calling initialization function ics2_init() (RX23T sample code)

(b) Calling the data transfer function “ics2_watchpoint()”

Call the data transfer function “ics2_watchpoint()” at intervals longer than the minimum sampling cycle as shown in the sample code below.

```
void int_TM100u(void)                           /* 100 μsec interval */
{
        if (3 <= g_u1_cnt_decimation)    /* decimation of ICS call */
        {
                g_u1_cnt_decimation = 0;
                ics2_watchpoint();               /* data transfer */
        }
        g_u1_cnt_decimation++;
}
```

Figure 17-4 Calling data transfer function ics2_watchpoint() (RX23T sample code)

(2) Calling on RL microcontroller

(a) Calling initialization function “ics2_init()”

Add the initialization function” ics2_init” to the initialization process part of a program as shown in the sample code below. Specify the argument based on the following rules.

The first argument:	specify the start address (0 for the lower 12 bit) of a DTC table
The second argument:	select the port used from “ics_****.h” (see Table 17-5)
The third argument:	specify the tool interrupt level
The foruth argument:	specify the communication rate
The fifth argument:	specify the transfer mode

```
#include "ics_RL78G1F_ca.h"

#pragma address dtc_tbl = 0xFFE00
char dtc_tbl[0xD0];

void main (void)
{
        ics2_init((void*) dtc_table, port, level, speed, mode);
}
```

Figure 17-5 Calling initialization function ics2_init() (RL78/G1F sample code)

(b) Calling data transfer function “ics2 watchpoint()”

Call the data transfer function “ics2_watchpoint ()” at intervals longer than the minimum sampling cycle as shown in the sample code below.

```
__interrupt void int_TM50u(void)          /* 50 μsec */
{
        if (4 <= g_u1_cnt_decimation)     /* decimation of ICS call */
        {
                g_u1_cnt_decimation = 0;
                ics2_watchpoint();                /* data transfer */
        }
        g_u1_cnt_decimation++;
}
```

Figure 17-6 Calling data transfer function ics2_watchpoint() (RL78/G1F sample code)

(3) Calling on RA microcontroller

(a) Calling initialization function “ics2_init()”

Add the initialization function” ics2_init()” to the initialization process part of a program as shown in the sample code below. Specify the argument based on the following rules.

The first argument:	select the port used from “ics_****.h” (see Table 17-7)
The second argument:	specify the communication rate.
The third argument:	specify the transfer mode.

```
#include "ics2_RA6T1.h"

void main(void)
{
        ics2_init(port, speed, mode);
}
```

Figure 17-7 Calling initialization function ics2_init() (RA6T1 sample code)

(b) Calling data transfer function “ics2_watchpoint()”

Call the data transfer function “ics2_watchpoint ()” at intervals longer than the minimum sampling cycle as shown in the sample code below.

```
void int_TM100u(void)                          /* 100 μsec interval */
{
        if (3 <= g_u1_cnt_decimation)    /* decimation of ICS call */
        {
                g_u1_cnt_decimation = 0;
                ics2_watchpoint();             /* data transfer */
        }
        g_u1_cnt_decimation++;
}
```

Figure 17-8 Calling data transfer function ics2_watchpoint() (RA6T1 sample code)

17.6 HW Configuration

This section explains the HW configuration when including a library and using RMW.

17.6.1 When Using MCK Motor Control Evaluation Kit

Figure 17-9 shows an example of HW configuration when RMW is used with the MCK motor control evaluation kit.

- Example of MCK motor control evaluation kit:
  MCK-RA6T2

- Example of HW configuration when using a communication library
  Download the program that includes the communication library to the MCU on the PC.
  Connect a communication board for tools between the PC with RMW installed and the CPU board.

- Example of HW configuration when using the built-in type communication library.
  Download the program that includes the built-in type communication library to the MCU on the PC.
  Connect a commercially available serial USB conversion board between the PC with RMW installed and the CPU board.

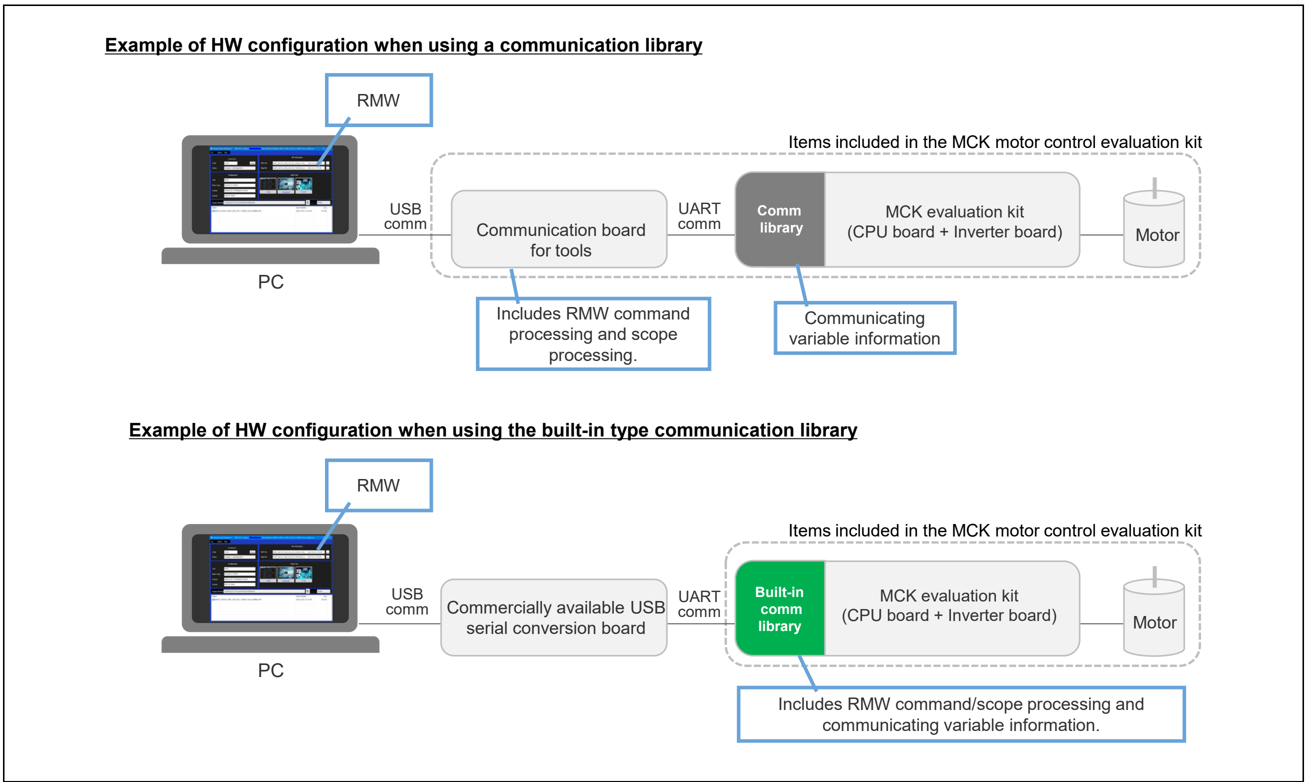


Figure 17-9 Example of HW Configuration When Using MCK Motor Control Evaluation Kit

17.6.2 When Using Evaluation System

Figure 17-10 shows an example of HW configuration when RMW is used with an evaluation system.

- Examples of evaluation systems
  - Evaluation System for Stepping Motor with Resolver
  - Evaluation System for BLDC Motor
  - Motor Control Evaluation System for RA Family - RA6T1 Group

- Example of HW configuration when using a communication library

  Download the program that includes the communication library to the MCU on the CPU card.

  Connect the PC with RMW installed to the evaluation system.

- Example of HW configuration when using the built-in type communication library.

  Download the program that includes the built-in type communication library to the MCU on the CPU card.

  Connect a commercially available serial USB conversion board between the PC with RMW installed and the CPU card.

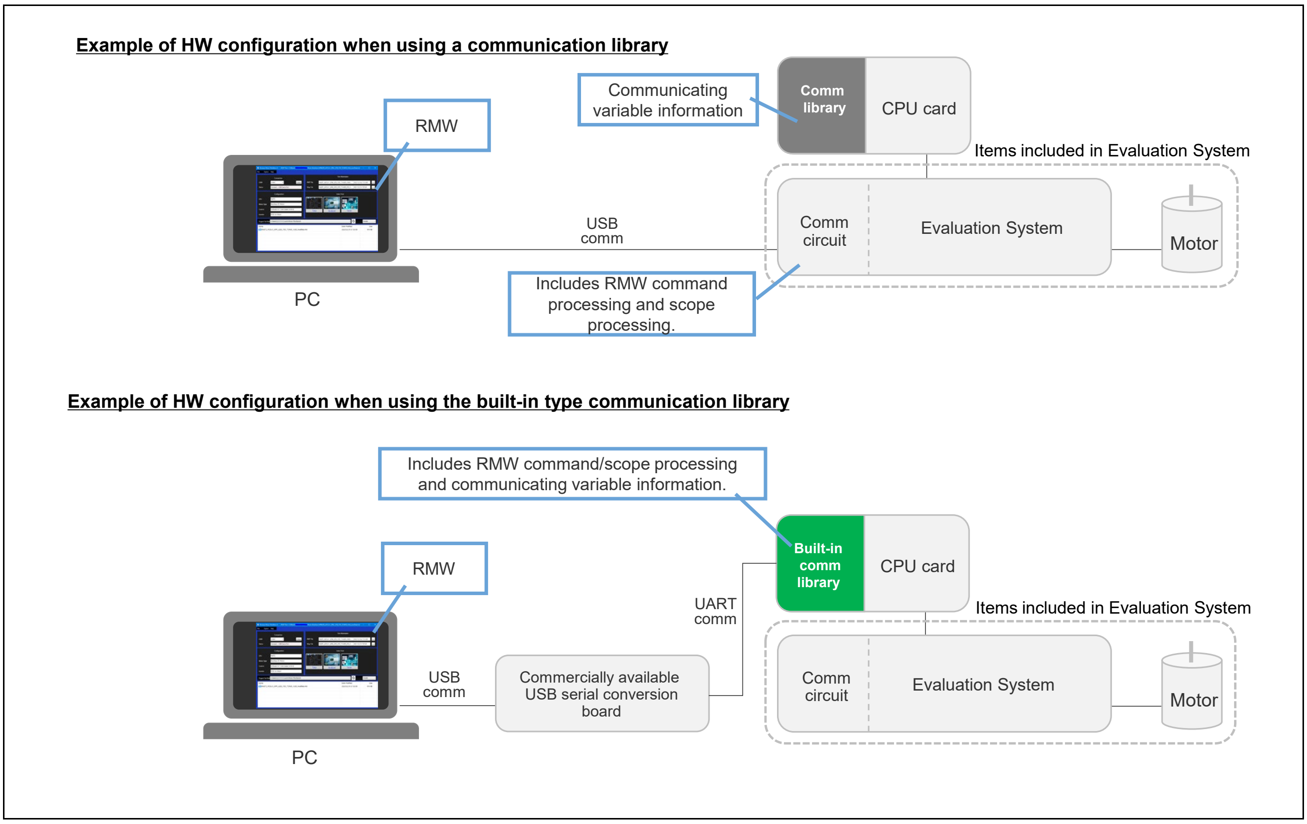


Figure 17-10 Example of HW Configuration When Using Evaluation System

17.6.3 When Using User Board

Figure 17-11 shows an example of HW configuration when RMW is used with a user board.

- Example of user board
  A board developed by a user.

- Example of HW configuration when using a communication library
  Download the program that includes the communication library to the MCU on a user board.
  Connect a communication board for tools between the PC with RMW installed and the user board.

- Example of HW configuration when using the built-in type communication library.
  Download the program that includes the built-in type communication library to the MCU on the user board.

  Connect a commercially available serial USB conversion board between the PC with RMW installed and the user board.

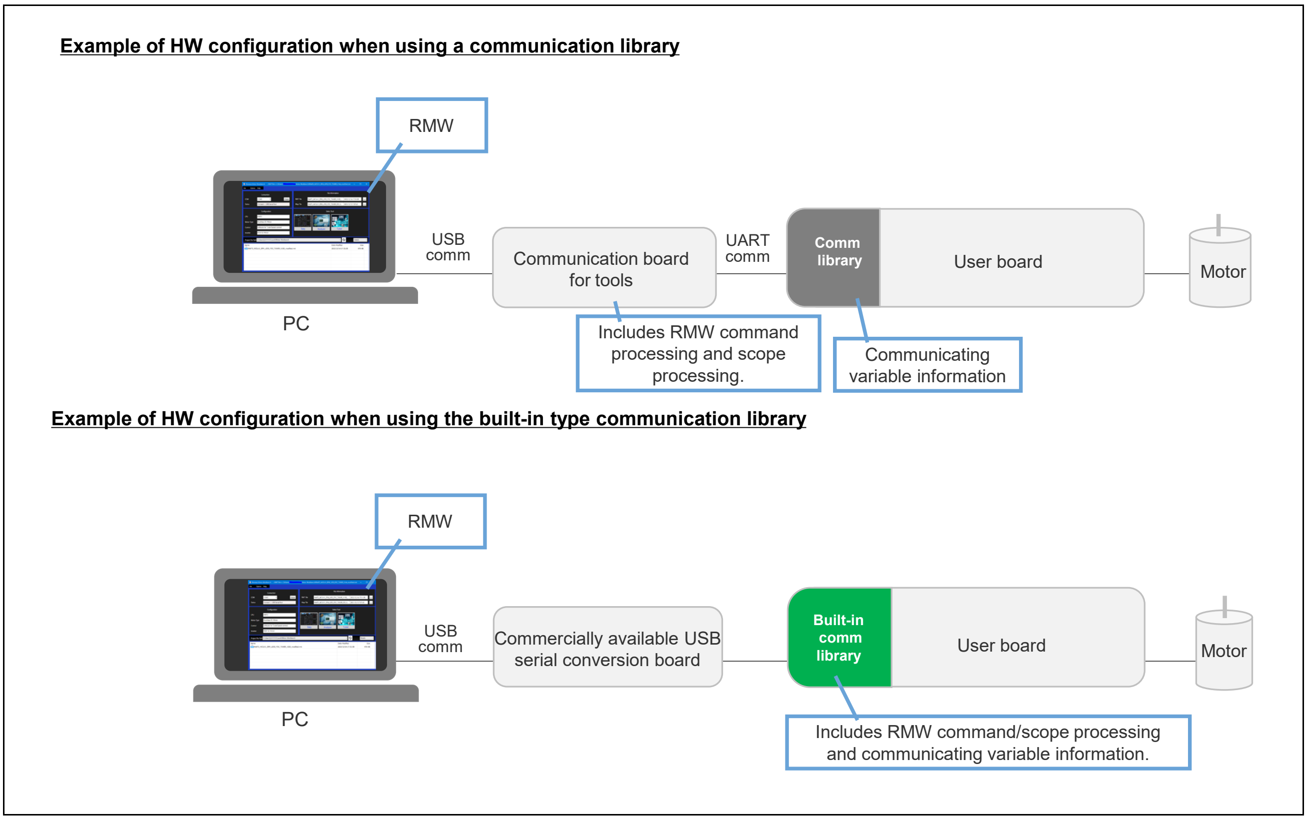


Figure 17-11 Example of HW Configuration When Using User Board

17.7 Example of Using Communication Library

The procedure for incorporating the communication library into the motor control program and using RMW is as follows.

- Determining the terminals to use: Decide which terminals to use for connecting the communication board to the board equipped with the MCU.
- Determining the communication speed: Set the communication speed between the RMW and the board.
- Creating the motor control program
  - ➢ Copy the communication library files.
  - ➢ Add the communication library files.
  - ➢ Call the communication library functions.
- Download the motor control program, which incorporates the communication library, to the MCU.
- Connect with RMW

This chapter describes examples of using RMW with the MCK-RA6T2 MCK Motor Control Evaluation Kit.

The procedure for using RMW on the user board is equivalent to that when using the MCK Motor Control Evaluation Kit.

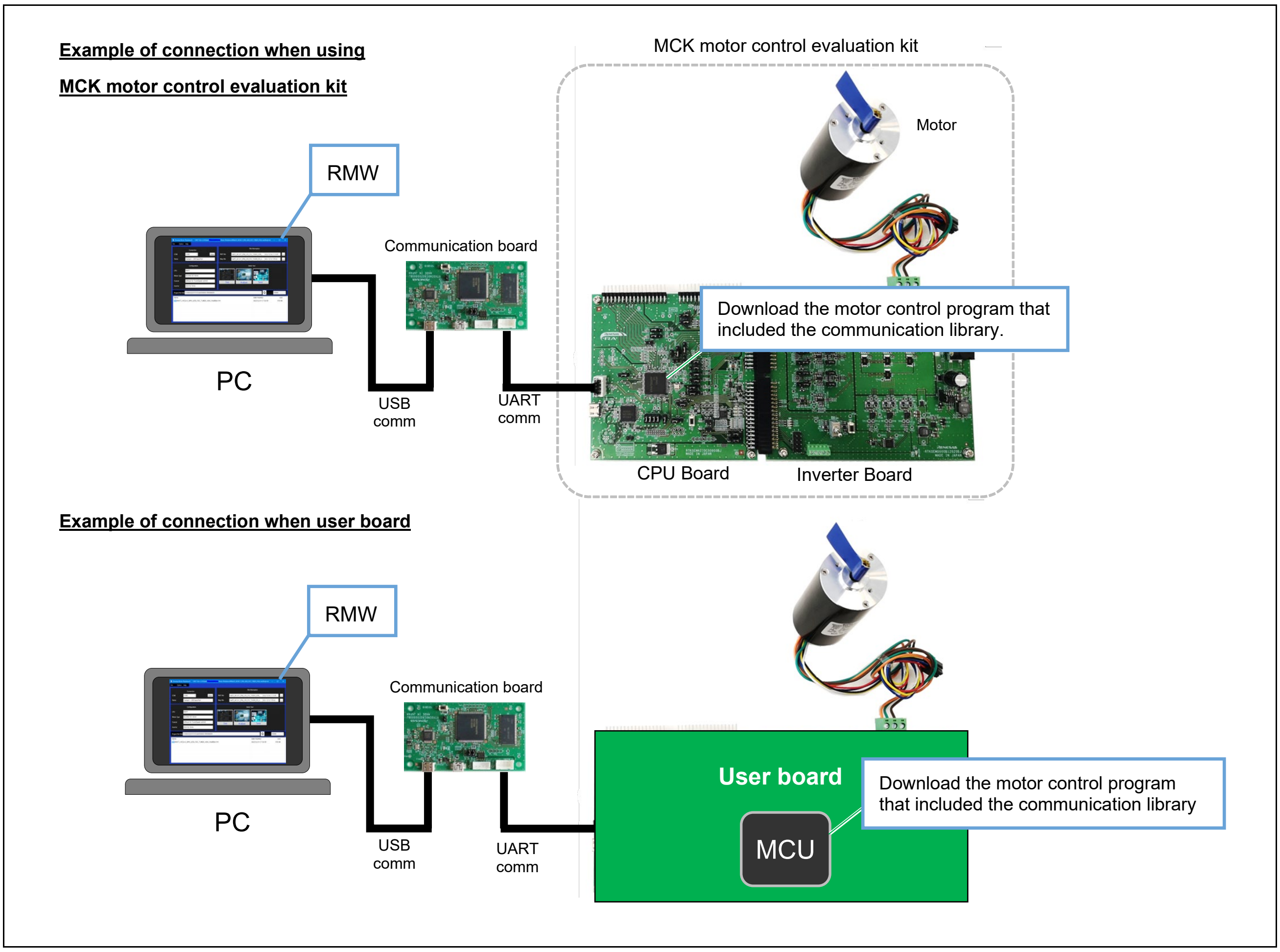


Figure 17-12 Example of Connection When Including Communication Library and Using RMW

17.7.1 HW Preparation

Use the MCK-RA6T2 and the included communication board (MC-COM).

For details on the MCK-RA6T2, refer to the following documents:

- R12QS0047 MCK-RA6T2 Quick Start Guide
- R12UZ0091 MCK-RA6T2 User's Manual

For details on the MC-COM, refer to the following document:

- R12UZ0093 MC-COM User's Manual

Items included in MCK-RA6T2

Figure 17-13 Items of be Used

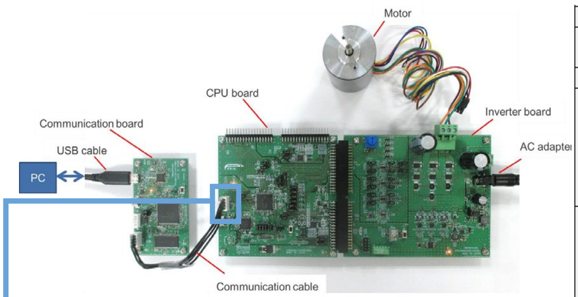
17.7.2 Determine Which Pin to Use

Connect the SCI terminals between the CPU board and the communication board using a communication cable. For details on the SCI terminals to use, refer to the MCK-RA6T2 User's Manual.

For the MCK-RA6T2, use SCI9 and PD05/TXD9_A and PD06/RXD9_A.

The pins determined in this chapter will be used in the following chapters:

- 17.7.4.3 Call Library Function : Setting the port argument for ics2_init()
- 17.7.6 Connect with RMW : Connecting to the USB-to-Serial Conversion Board



Supported CPU	RA6T1	RA6T2
File	ICS2_RA6T1.o ICS2_RA6T1.h	ICS2_RA6T2.o ICS2_RA6T2.h
Communication rate	0.5 Mbps to 15.0 Mbps	0.5 Mbps to 20.0 Mbps
Port	SCI0 TXD0:P101 RXD0:P100 SCI4 TXD4:P205 RXD4:P206 SCI9 TXD9:P109 RXD9:P110	SCI9 TXD9:PD05 RXD9:PD06
Supported variable type	8bit unsigned integer (nun 8bit signed integer (number display/setting/waveform display) 16bit unsigned integer (number display/setting/waveform display) 16bit signed integer (number display/setting/waveform display) 32bit unsigned integer (number display/setting/waveform display) 32bit signed integer (number display/setting/waveform display) 32bit IEEE754 floating point (number display/setting/waveform display)	
CPU resource to use	SCIx RX, SCIx TX	SCIx RX, SCIx TX

Pins that can be used in the built-in Type communication library.

Table 6-9 SCI connector (CN10) pin assignment

Pin No.	Pin Function	RA6T2 Connection Pin
1	GND	-
2	MCU RXD	PD06/RXD9_A
3	MCU TXD	PD05/TXD9_A
4	VCC	-

SCI9-A

Terminals that can be connected to the communication board.

Except from "MCK-RA6T2 User's Manual"

Figure 17-14 Determining Terminals to be Used

17.7.3 Determine Communication Speed

Determine the communication speed between the PC and the communication board, as well as the CPU board. The communication speed must satisfy the following conditions:

- It must be a communication speed supported by the library.
- It must be a communication speed supported by the MC-COM board

The communication speed determined in this chapter will be used in the following chapters:

- 17.7.4.3 Call Library Function : Set in the speed argument of ics2_init()
- 17.7.6 Connect with RMW : Set the communication speed in RMW

When using a MC-COM (Renesas communication board for tools), you can select the following communication rates (clock frequencies). Set it based on the jumper (JP2) of the MC-COM.

Table 17-15 Settings of MC-COM JP2 and selectable clock frequencies

JP2	Selectable clock frequency
Short	1 Mbps (8 MHz), 5 Mbps (40 MHz), 7.5 Mbps (60 MHz), 10 Mbps (80 MHz), 15 Mbps (120 MHz)
Open	6.25 Mbps (50 MHz), 8.33 Mbps (66666666 Hz), 12.5 Mbps (10 MHz), 16.66 Mbps (133333333 Hz)

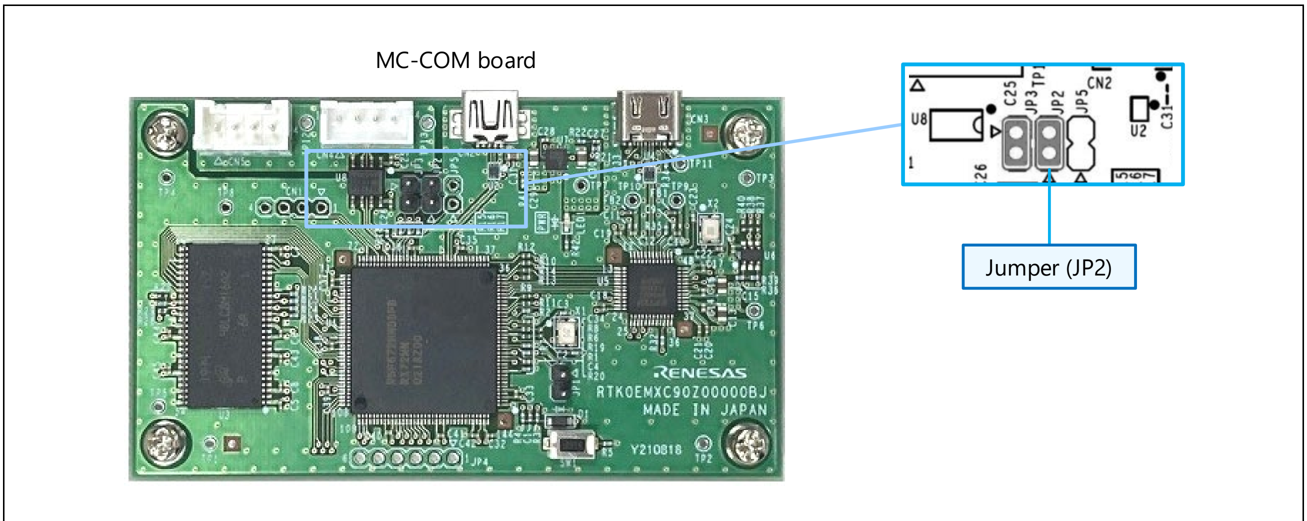


Figure 17-15 JP2 of MC-COM (communication board for tools)

17.7.4 Create Program

Create the motor control program to download to the RA6T2 mounted on the MCK-RA6T2 CPU board. This chapter explains how to incorporate the communication library into a motor control program already created in e^2 studio.

17.7.4.1 Copy Library File

The header files and object files for the communication library are included in the downloaded RMW installer files. Copy the communication library header files and object files to any location within your motor control program.

Figure 17-16 shows the folder structure after copying the library files, using the sample code as an example.

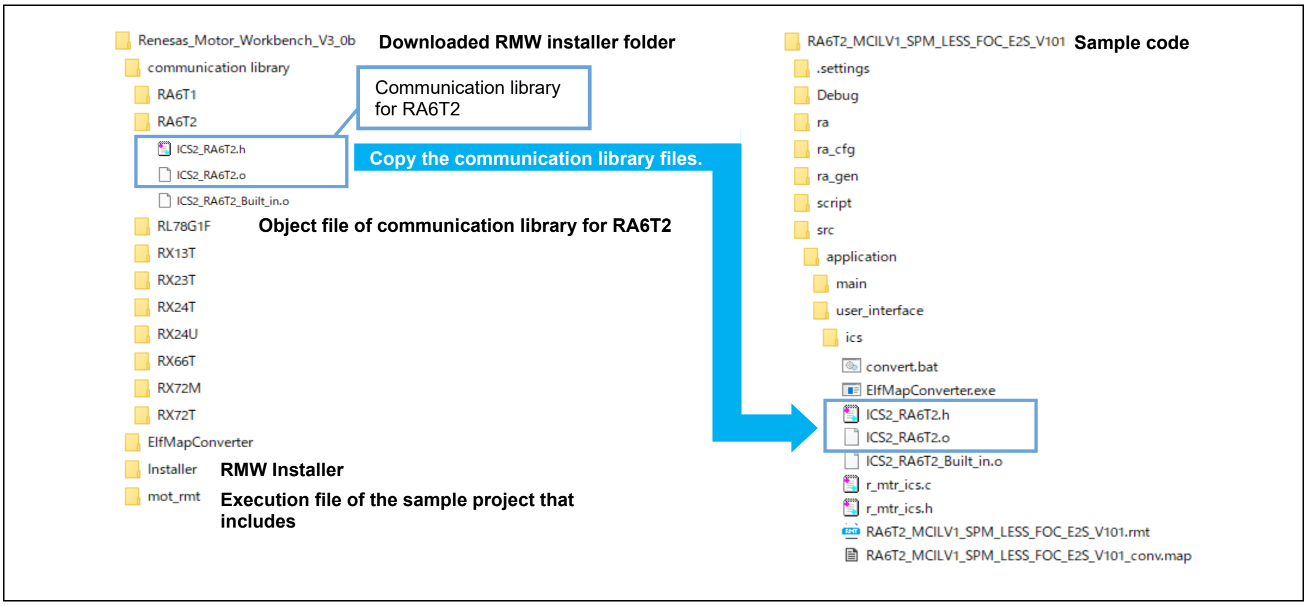


Figure 17-16 Folder Structure When Library Files are Copied

17.7.4.2 Add Library File

Add the copied library files in e^2 studio. Open the project properties in e2 studio and add the object file path as shown in Figure 17-17 and the header file path as shown in Figure 17-18.

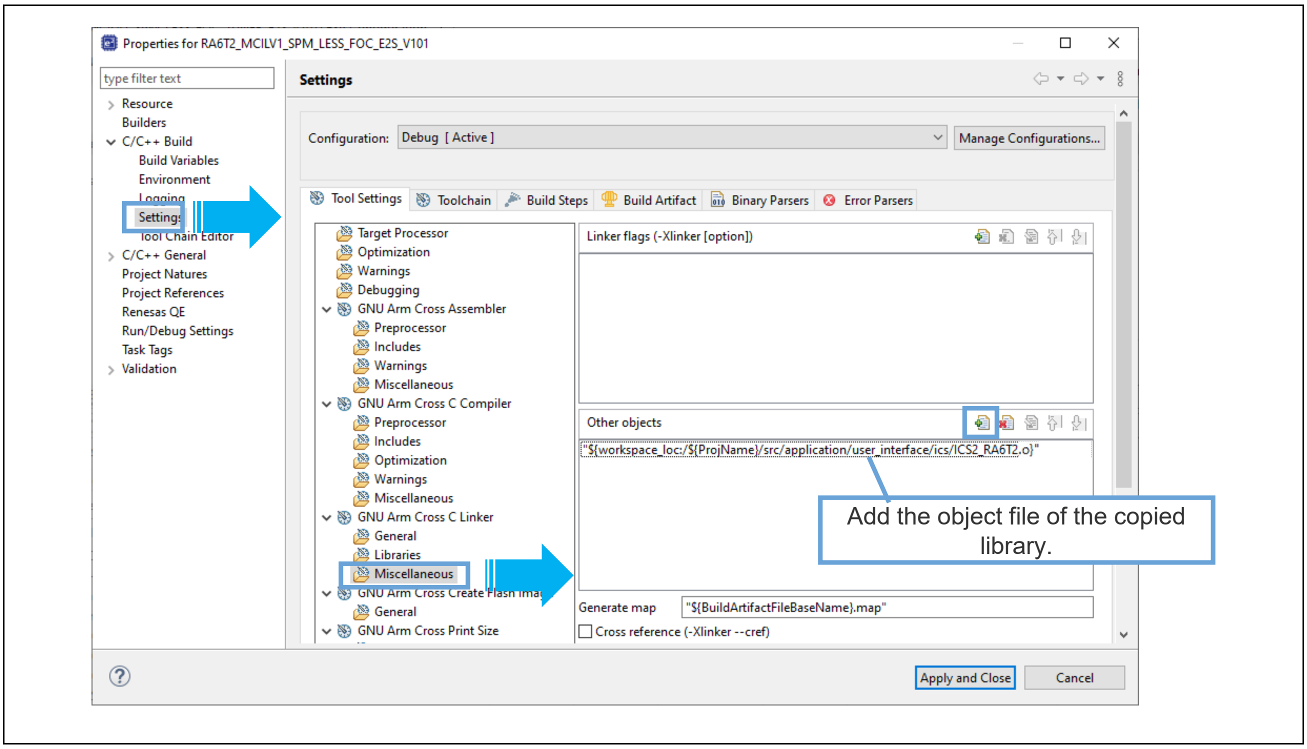


Figure 17-17 Adding Library Object File on e^2 Studio

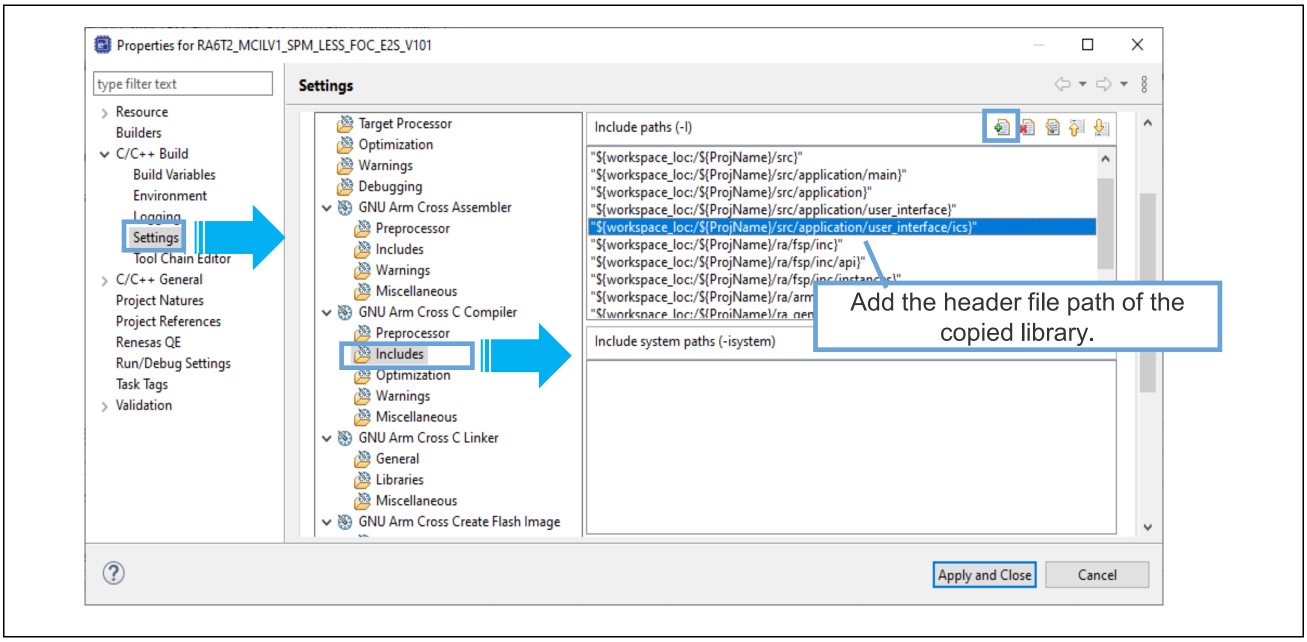


Figure 17-18 Adding Library Header File Path on e2 Studio

17.7.4.3 Call Library Function

The library initialization function uses the pins determined in Section 17.7.2 and the communication speed determined in Section 17.7.3.

The communication library for RA6T2 provides the following two types of functions:
- void ics2_init(uint8_t port, uint8_t speed, uint8_t mode) : Initialization function
- void ics2_watchpoint(void) : Data transfer function to be called periodically

The arguments for the ics2_init() function differ depending on the target microcontroller. The pin to set for the port argument in ics2_init() is defined in the header file ICS2_RA6T2.h.
An example of calling the communication library function from the RA6T2 motor control program is as follows.

(1) Calling initialization function “ics2_init()”

Add the initialization function” ics2_init()” to the initialization process part of a program as shown in the sample code below. Specify the argument based on the following rules.

The first argument: select the port used from “ics_****.h”
The second argument: specify the communication rate.
The third argument: specify the transfer mode.

```
#include "ics2_RA6T2.h"

void main(void)
{
      /* port  = SCI9_A (PD05/TXD9 and PD06/RXD9): defined in ICS2_RA6T2.h     */
      /* Rate  = (SCI Clock / (6 x (speed + 1)))[Mbps] :                       */
      /*           Rate = 1Mbps, SCISPICLK = 120MHz -> speed = 19              */
      /* mode  = 32bit x 8ch                                                   */
      ics2_init(ICS_SCI9_PD05_PD06, 19, 1);
}
```

Figure 17-19 Calling initialization function ics2_init() (RA6T2 sample code)

(2) Calling data transfer function “ics2_watchpoint()”

Call the data transfer function “ics2_watchpoint ()” at intervals longer than the minimum sampling cycle as shown in the sample code below.

```
/* Call the data transfer function cyclically                              */
/* If the transfer rate is Rate[Mbps], the minimum sampling cycle is       */
/* 70+(180/Rate) [μsec], and it must be called at a cycle greater than this.  */

void int_TM50u(void)                   /* 50 μsec interval */
{
      g_u1_cnt_decimation++;
      if (5 <= g_u1_cnt_decimation)      /* decimation of ICS call */
      {
```

```
            g_u1_cnt_decimation = 0;
            ics2_watchpoint();          /* data transfer */
        }
    }
```

Figure 17-20 Calling data transfer function ics2_watchpoint() (RA6T2 sample code)

17.7.5 Download Program

Connect the PC to the CPU board and download the motor control program to the RA6T2 on the CPU board. When downloading the motor control program to the CPU board, connections to the inverter board and motors are not required.

For details on the program download method, refer to the Quick Start Guide for the evaluation kit you are using or the programming tool manual.

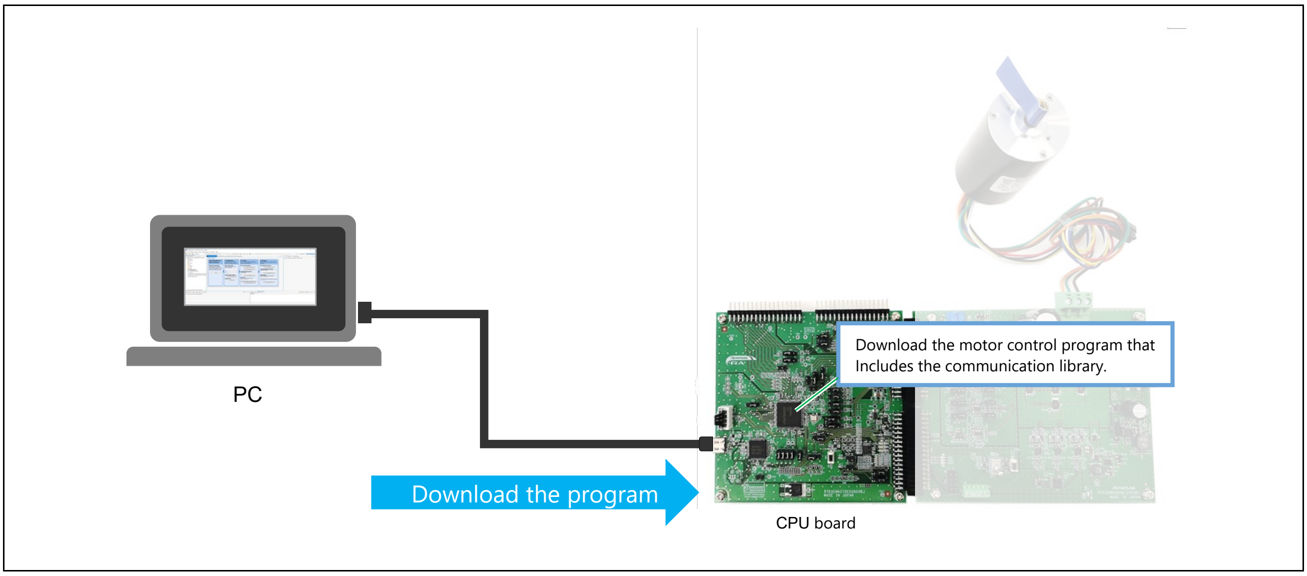


Figure 17-21 Connection When the Program is Being Downloaded

17.7.6 Connect with RMW

Connect the PC with RMW installed to each board as shown in Figure 17-22. Use the pins determined in Section 17.7.2 for the connection between the CPU board and the Communication board.
Set the RMW communication speed as shown in Figure 17-23. In the 'Connection' section of the RMW Main Window screen, click the 'Clock' button and set the communication speed determined in Section 17.7.3. The communication speed set in RMW must be identical to the speed set in the library's initialization function ics2_init().
To set the clock frequency, obtain the value by multiplying the communication rate by 8.

Example:

When the communication rate is 1 Mbps, set the clock frequency to 8 MHz (8,000,000 Hz).

When the communication rate is 5 Mbps, set the clock frequency to 40 MHz (40,000,000 Hz).

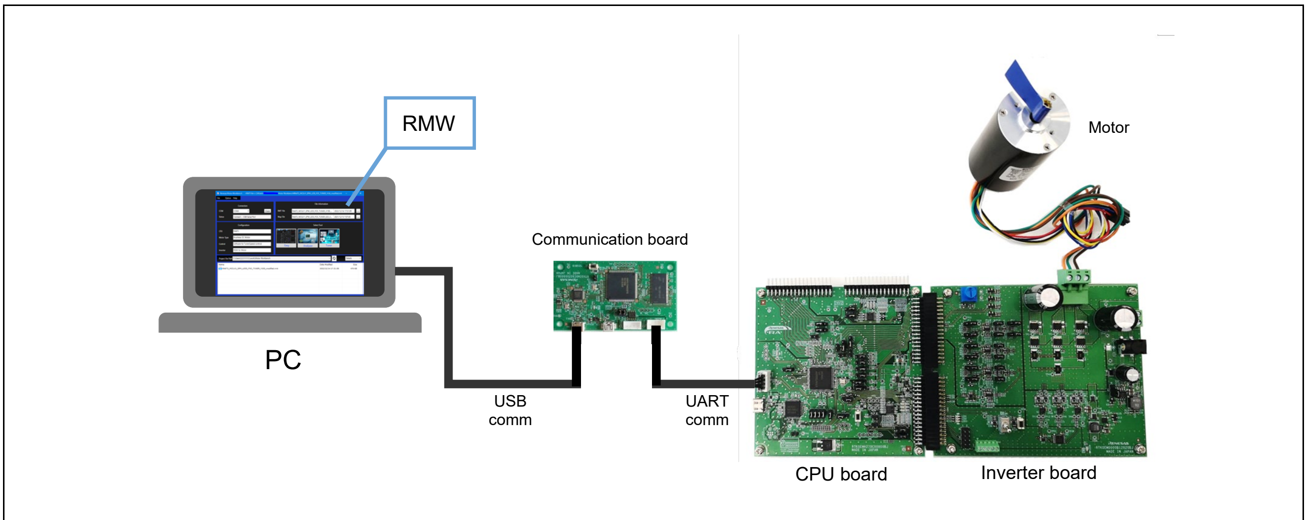


Figure 17-22 Connection When RMW is Used

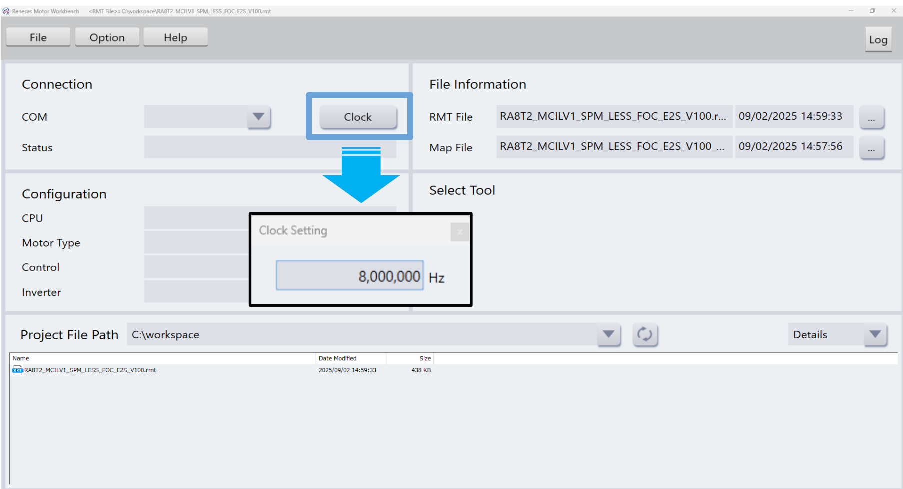


Figure 17-23 Setting RMW Communication Speed

17.8 Example of Using Built-in Communication Library

To include the built-in type communication library to the motor control program and use RMW, follow the steps below.

- Determine which pins to use : determine the pin used to connect the USB serial conversion board to the MCU mounted board.
- Determine the communication speed: determine the communication speed between RMW and the board.
- Create a motor control program
  - Copy the built-in type communication library file.
  - Add the built-in type communication library file.
  - Call the built-in type communication library function.
- Download the motor control program that includes the built-in type communication library to the MCU.
- Connect with RMW.

This section describes an example of using RMW with a MCK Motor Control Evaluation Kit, MCK-RA6T2.

The procedure for using RMW with a user board is the same as the procedure for using RMW with a MCK motor control evaluation kit.

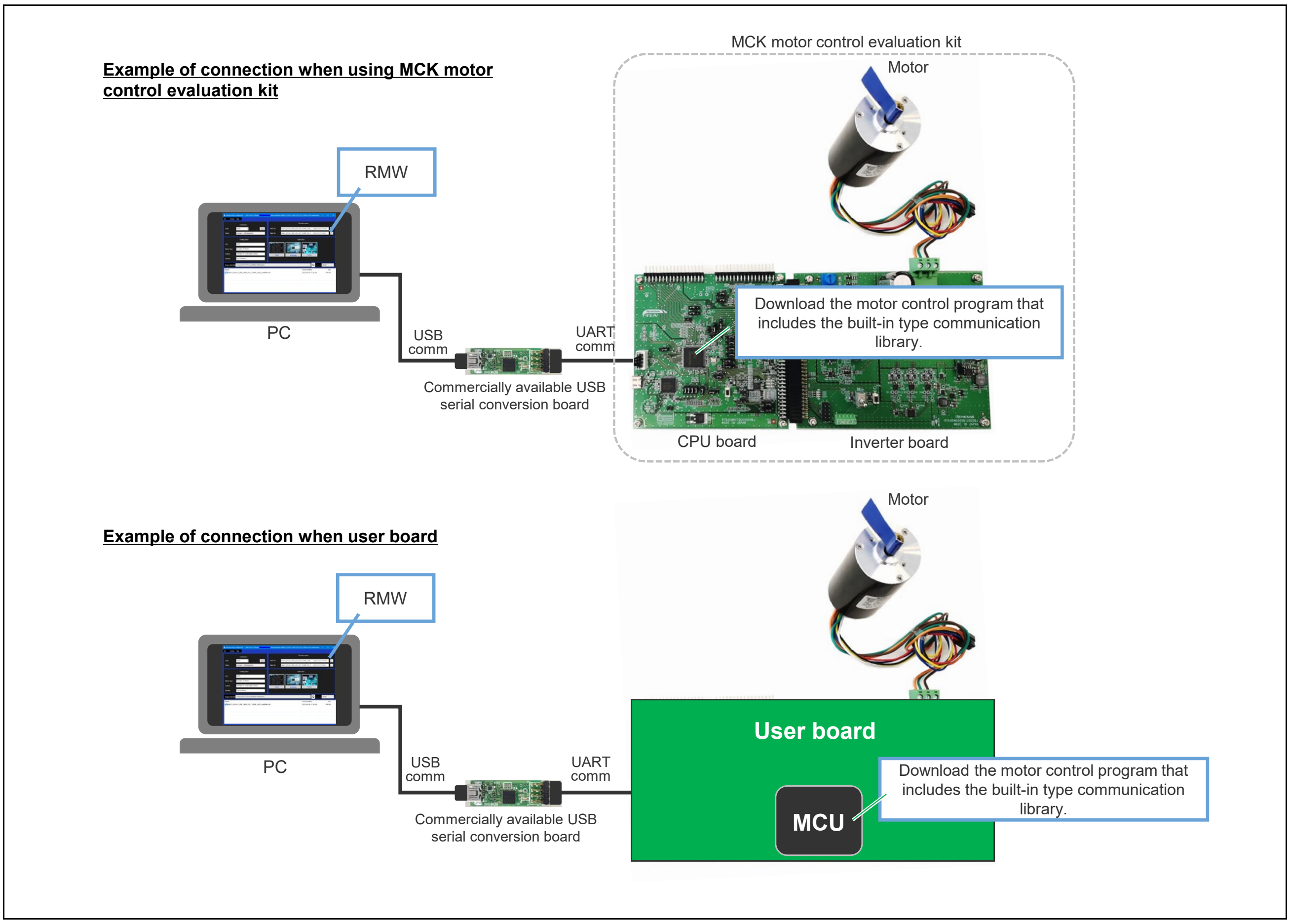


Figure 17-24 Example of Connection When Including Built-In Type Communication Library and Using RMW

17.8.1 HW Preparation

Use an MCK-RA6T2 and a commercially available USB serial conversion board.

Since the built-in type communication library is used, the communication board included in the MCK-RA6T2 is not used.

For details on MCK-RA6T2, refer to the following documents.

- R12QS0047 MCK-RA6T2 Quick Start Guide
- R12UZ0091 MCK-RA6T2 User's Manual

Items included in MCK-RA6T2

Commercially available USB serial conversion board

Figure 17-25 Items to be Used

17.8.2 Determine Which Pin to Use

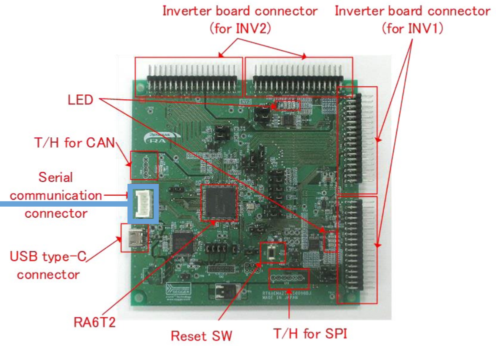
Determine the pins to connect the CPU board to the USB serial conversion board. The pins must satisfy the following conditions:

- Pins that can be used in the library
- The corresponding pins of the CPU board can be connected to the USB serial converion board.

For MCK-RA6T2, use SCI9, PD05/TXD9_A, and PD06/RXD9_A as shown in Figure 17-26.

The pins to use determined in this section are used in the following sections:

- 17.8.4.3 Call the library function : set the argument port of ics2_init()
- 17.8.6 Connecting with RMW : connect to the USB serial conversion board



Supported CPU	RA6T2	RA6T3	RA4T1
File	ICS2_RA6T2_Built_in.o	ICS2_RA6T3_Built_in.o	ICS2_RA4T1_Built_in.o
Communication rate	-	-	-
Port	SCI9 TXD9:PD05 RXD9:PD06	SCI0 TXD0:P101 RXD0:P100 SCI0 TXD0:P213 RXD0:P212 0:P411 0:P410 9:P110 RXD9:P109	SCI0 TXD0:P101 RXD0:P100 SCI0 TXD0:P213 RXD0:P212 SCI0 TXD0:P411 RXD0:P410 SCI9 TXD9:P110 RXD9:P109
Supported variable type	8bit unsigned integer (number display/setting/waveform display) 8bit signed integer (number display/setting/waveform display) 16bit unsigned integer (number display/setting/waveform display) 16bit signed integer (number display/setting/waveform display) 32bit unsigned integer (number display/setting/waveform display) 32bit signed integer (number display/setting/waveform display) 32bit IEEE754 floating point (number display/setting/waveform display)		
CPU resource to use	SCIx RX, SCIx TX	SCIx RX, SCIx TX	SCIx RX, SCIx TX

Pins that can be used in the built-in type communication library

Table 6-9 SCI connector (CN10) pin assignment

Pin No.	Pin Function	RA6T2 Connection Pin
1	GND	-
2	MCU RXD	PD06/RXD9_A
3	MCU TXD	PD05/TXD9_A
4	VCC	-

SCI9-A

Except from "MCK-RA6T2 User's Manual"

Pins that can be connected to the USB serial conversion board.

Figure 17-26 Determining Pins to be Used

17.8.3 Determine Communication Speed

Determine the communication speed between the PC and the USB serial conversion board/CPU board. The communication speed must satisfy the following conditions.

- The communication speed that can be used in the library.
- The communication speed can be set for the USB serial conversion board.

You can check the communication speed that can be set for the USB serial conversion board from the properties of the PC’s device manager, as shown in Figure 17-27.

The communication speed determined in this section is used in the following section:

- 17.8.4.3 Call the library function : set the argument speed of ics2_init()
- 17.8.6 Connecting to RMW : set the communication speed to RMW

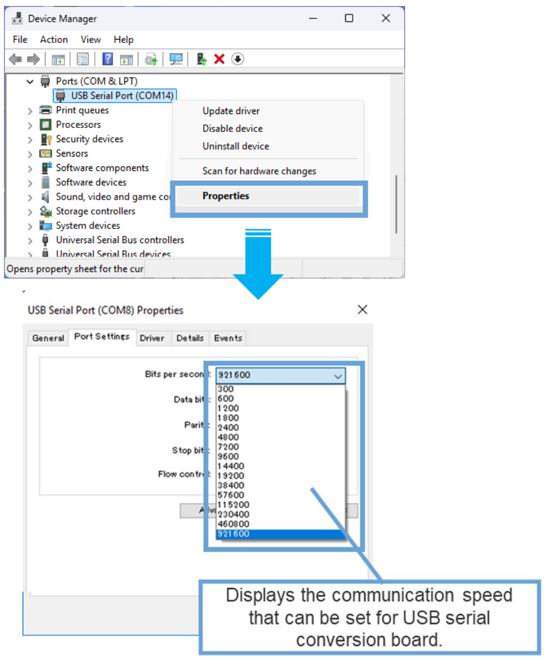


Figure 17-27 Checking the Communication Speed That Can Be Set for RMW

17.8.4 Create Program

Create a motor control program to be downloaded to the RA6T2 mounted on the CPU board of MCK-RA6T2.

This section describes how to include the built-in type communication library into a motor control program that has already been created with e^2 studio.

17.8.4.1 Copy Library File

The header and object files of the built-in type communication library are included in the file where you downloaded the RMW installer. Copy the header and object files of the built-in type communication library to any locations in the motor control program.

Figure 17-28 shows the folder structure in which the library file was copied to the sample code folder.

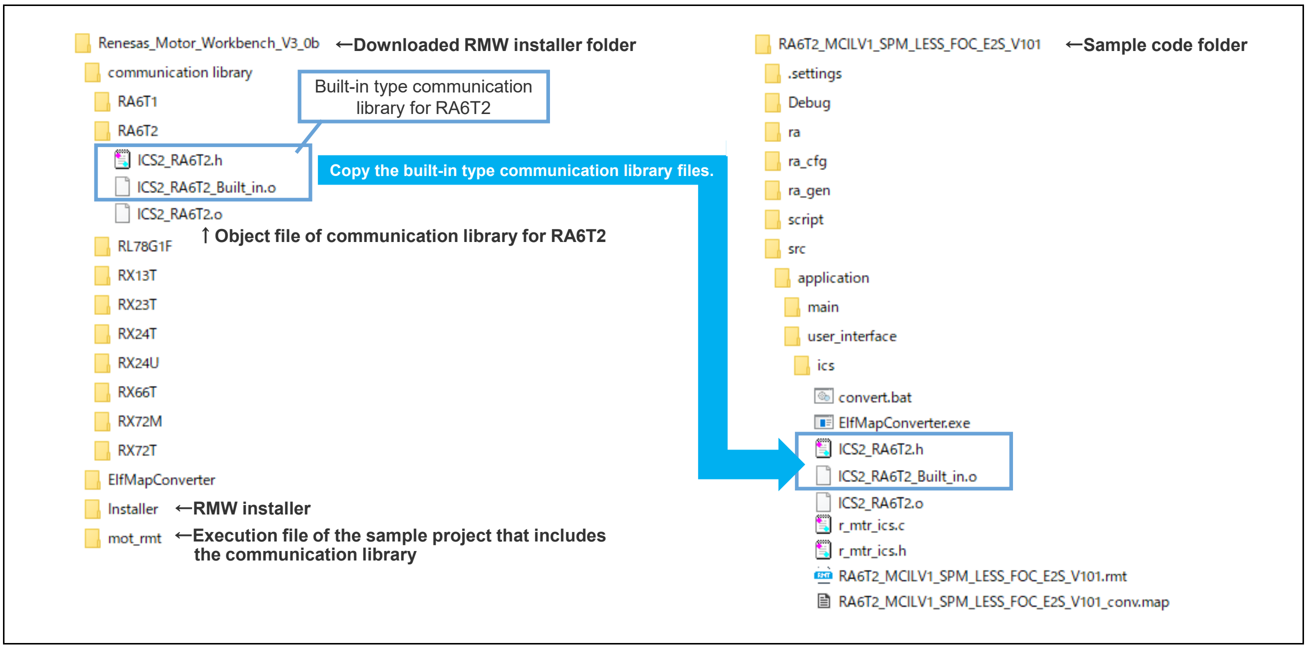


Figure 17-28 Folder Structure When Library Files are Copied

17.8.4.2 Add Library File

Add the copied library file on e[2] studio. Open the project's properties on e[2] studio, and add the object file as shown in Figure 17-29, and add the header file path as shown in Figure 17-30.

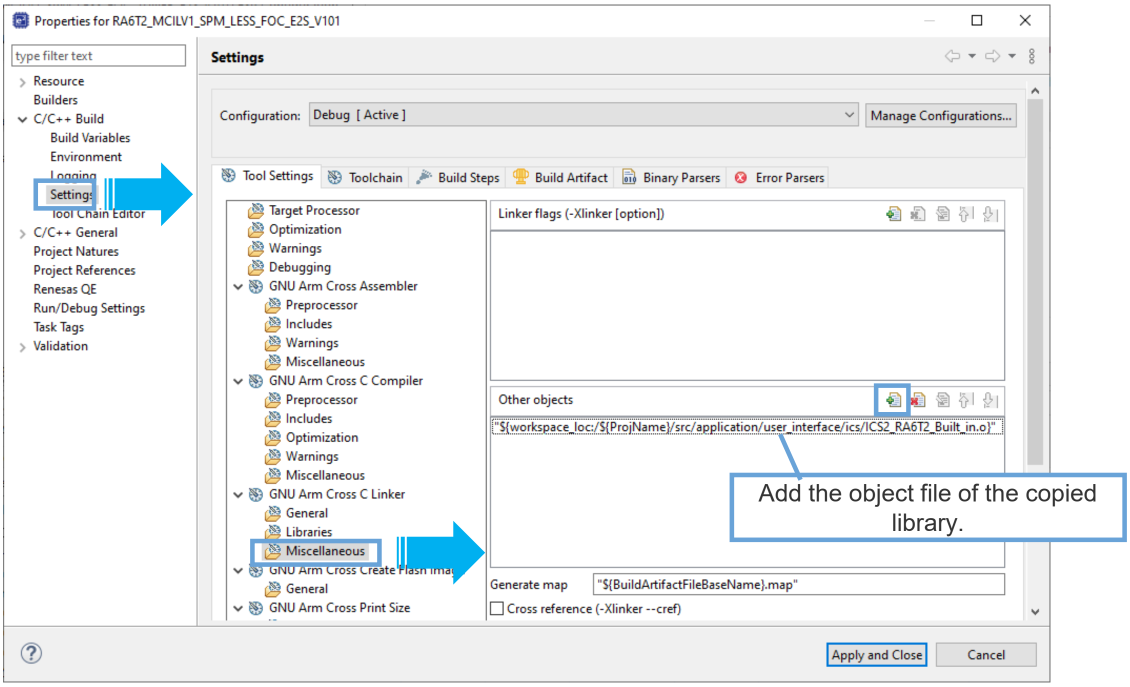


Figure 17-29 Adding Library Object File on e[2] Studio

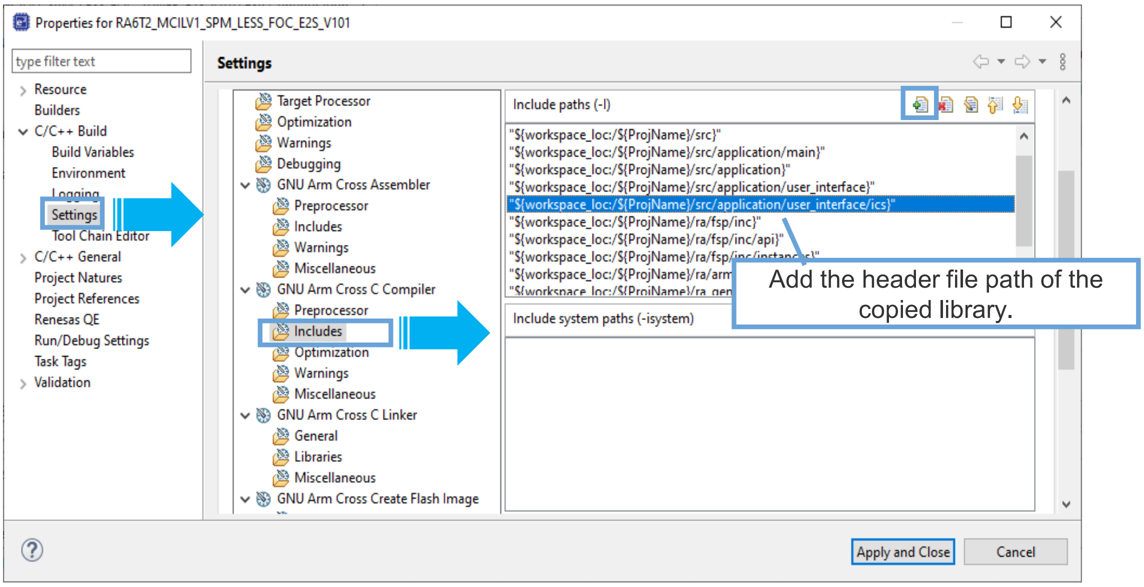


Figure 17-30 Adding Library Header File Path on e[2] Studio

17.8.4.3 Call Library Function

The library initialization function uses the pins to be used as determined in Section 17.8.2 and the communication speed as determined in Section 17.8.3.

The built-in type communication library for RA6T2 provides the following two types of functions:

- void ics2_init(uint8_t port, uint8_t speed, uint8_t mode) : initialization function
- void ics2_watchpoint(void) : data transfer function called periodically

The argument of the ics2_init() function differs depending on the target MCU. The pins set to the argument port of the ics2_init() function are defined in the header file ICS2_RA6T2.h.

The following shows an example of calling the built-in type communication library function from the motor control program of RA6T2.

```
#include "ICS2_RA6T2.h"
void main(void)
{
    /* port  = SCI9_A (PD05/TXD9 and PD06/RXD9): defined in ICS2_RA6T2.h      */
    /* Rate  = (SCI Clock / (6 x (speed + 1)))[Mbps] :                        */
    /*          Rate = 921600bps, SCISPICLK = 120MHz -> speed = 21  */
    /* mode  = 32bit x 8ch                                                    */
    ics2_init(ICS_SCI9_PD05_PD06, 21, 1);
}

/* Call the data transfer function cyclically                                */
void timer_isr(void)
{
    ics2_watchpoint();
}
```

17.8.5 Download Program

Connect the PC to the CPU board and download the motor control program to the RA6T2 on the CPU board. When downloading the motor control program to the CPU board, you are not required to connect to the inverter board and the motor.

For more information on how to download the program, see the Quick Start Guide of your evaluation kit or the documentation of your writing tool.

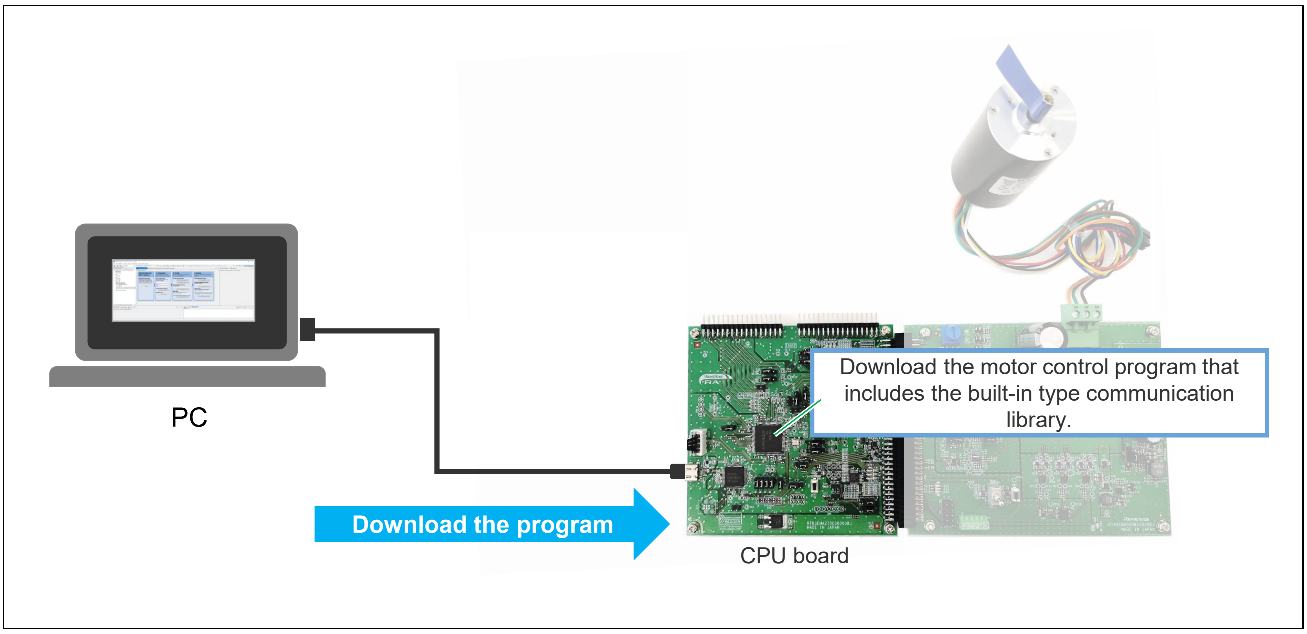


Figure 17-31 Connection When The Program is Being Downloaded

17.8.6 Connect with RMW

Connect the PC with RMW installed to each board as shown in Figure 17-32. To connect the CPU board to the USB serial conversion board, use the pins determined in Section 17.8.2.

Set RMW communication speed as shown in Figure 17-33. Select "Baud rate Dialog" from the "Option" menu in Main Window of RMW, and set the communication speed determined in Section 17.8.3. The communication speed set for RMW must be the same as the communication speed set in the library initialization function ics2_init().

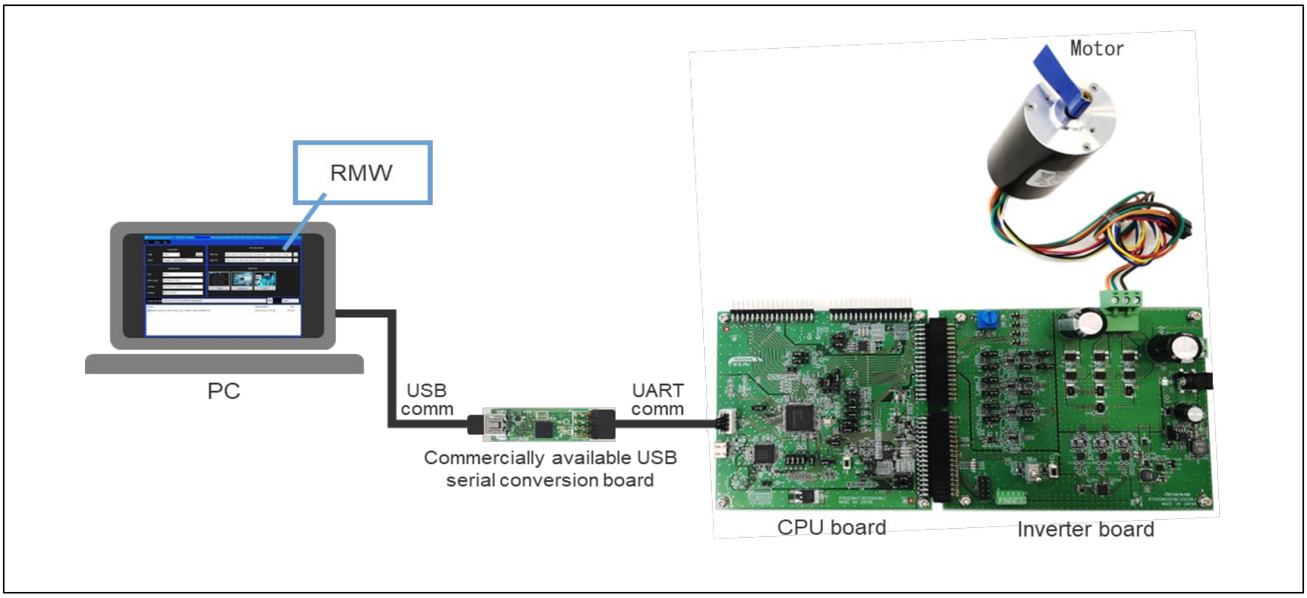


Figure 17-32 Connection When RMW is Used

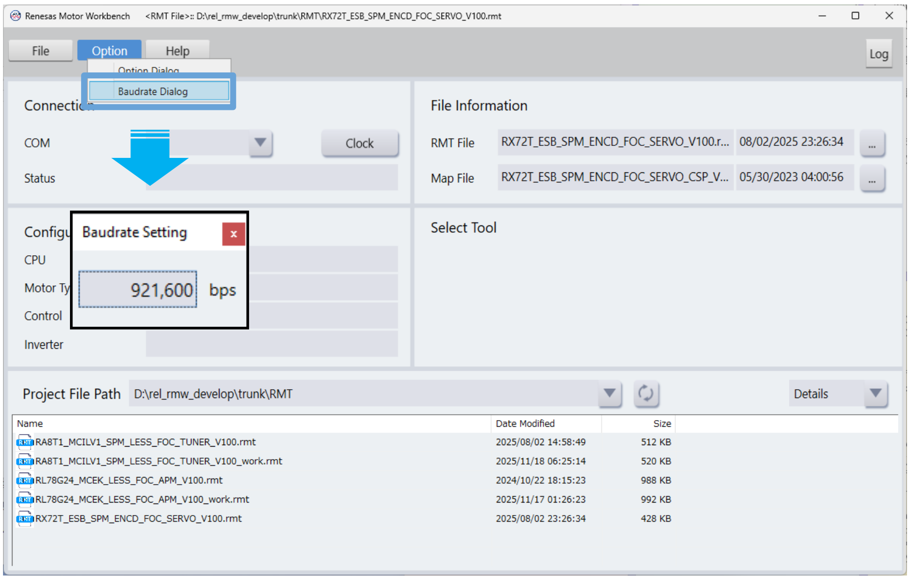


Figure 17-33 Setting RMW Communication Speed

Website and Support

Renesas Electronics Website

https://www.renesas.com/

Inquiries

https://www.renesas.com/contact

Revision History

Rev.	Date	Revision	
		Page	Contents
1.00	Apr.05.17	-	Issued
2.00	Nov.27.18	-	Supported version 2.0
2.01	May.16.19	-	Supported RX72T
2.02	Nov.29.19	-	Supported RX13T
2.03	Oct.28.20	-	Supported RA6T1
3.00	Dec.09.21	-	Supported version 3.0
4.00	May.30.23	-	• Renesas Motor Workbench 3.0 User's Manual has changed to Renesas Motor Workbench 3.0 Quick Start Guide(R21QS0011) • Revised new edition issued. • Supported Section 10, 11, 13 for Renesas Motor Workbench 3.1 • Modified Section 1, 2, 3, 4 for Renesas Motor Workbench 3.1
4.01	Oct.27.2023	-	• Added 1.2 Displaying and entering decimal points for Renesas Motor Workbench 3.1.1 • Fixed 2.4.1.1 Acquiring New Authentication File Changed the URL for obtaining the authentication file to the RMW website.
4.02	Jan.30.24	-	• Supported RA8T1 • Revised 13. Servo • Added 14.6 Procedure to Include Tuner Library
4.10	Feb.07.25	-	• Removed 2.4.1: Load Authentication file • Revised 4.1: Updated communication rate • Added 15: Tuner Tool (for RL78) • Added 16: Tuner Tool (for MCI-LV-1)
4.11	Nov.10.25	-	Supported RA8T2 and RA2T1
4.20	Dec.15.25	-	• Added 2.3.7: Display Log • Revised 8.3.3 Status Indicator Setting • Revised 14: Tuner Tool • Revised 15: Tuner Tool (for RL78) • Revised 16: Tuner Tool (MCI-HV-1)
4.21	Mar.31.26	-	• Supported RX14T • Added RA8T2 supported port

- This document is issued as Renesas Motor Workbench 3.0 User's Manual by re-editing the following two existing functional descriptions (application notes):
  - Renesas Motor Workbench Function Description(R20AN0527)
  - Renesas Motor Workbench Tuner Function Description (R20AN0528)

Ref: Revision history of old version of RMW functional description (R20AN0527)

Rev.	Date	Revision	
		Page	Contents
1.00	Nov.27.18	-	First edition released.
2.00	Dec.09.21	-	Revised with Renesas Motor Workbench3.0 release

Ref: Revision history of old version of RMW Tuner functional description (R20AN0528)

Rev.	Date	Revision	
		Page	Contents
1.00	Nov.27.18	-	First edition released.
2.00	Dec.09.21	-	Revised with Renesas Motor Workbench3.0 release

- The existing Renesas Motor Workbench User's Manual (R21UZ0004) has changed to Renesas Motor Workbench Quick Start Guide (R21QS0011). Please refer it together with this document.

Renesas Motor Workbench 3.3.1 User's Manual

Publication Date: Rev.4.21 Mar.31.26

Published by: Renesas Electronics Corporation

Motor Control Development Support Tool
Renesas Motor Workbench 3.3.1

Renesas Electronics Corporation

R21UZ0004EJ0421