RA/RX/RL78 Family

Motor Control Development Support Tool Renesas Motor Workbench 3.2.0 Quick Start Guide

Introduction

Renesas Motor Workbench is a development support tool for Renesas MCUs for motor control (for target MCUs, see '1.3 Specifications') that performs debugging and tuning of motor control programs. It provides internal tools, Analyzer, Easy, and Tuner. This Quick Start Guide describes the procedure for Renesas Motor Workbench from installation to execution. For each function of Renesas Motor Workbench, refer to the following manual.

- Renesas Motor Workbench User's Manual (R21UZ0004)

Table of Contents

1. Overview

1.1 About Renesas Motor Workbench

Renesas Motor Workbench is a tool that enables real-time reading/writing values of global variables of the program executed on the motor control evaluation board connected to a PC via USB.

Table 1-1 Main functions of Renesas Motor Workbench

Function	Overview
Analyzer	• Starts when the “Analyzer” icon is clicked from Select Tool in Main Window. • For global variables of the program executed on a board connected via USB, it can “load variable values”, “write variable values”, and “display variable values in waveforms in real time”.
Easy GUI	• Starts when the “Easy” icon is clicked from Select Tool in Main Window. • The GUI allows you to easily instruct drive and command values for programs executed on a board connected via USB.
Servo	• Starts when the “Servo” icon is clicked from Select Tool in Main Window. • The GUI allows you to easily instruct servo for programs executed on a board connected via USB.
Tuner	• Starts when the Tuner icon is clicked from Select Tool in Main Window. • Allows you to operate “measurement of peculiar parameter” and “auto-adjustment of control parameter Note of the motor for the program executed on a board connected via USB.

Note: The program executed on the board must have the auto-adjustment function.

Figure 1-1 Main Window of Renesas Motor Workbench

Renesas Motor Workbench assumes the following sample programs as the target.

- The Sample program included in this package
- The Sample programs provided by Renesas (downloadable from Renesas Web site)
- User programs

1.2 System Requirement

1.2.1 Target System

Renesas Motor Workbench can be used on the motor control board connected to a PC via USB. There are three types of boards depending on connection to the PC.

Type 1: Motor control inverter board that incorporates the communication function to the PC (isolated type)
(ex.) Renesas Solution Starter Kit - Evaluation System for BLDC Motor (Motor RSSK)

Type 2: Motor control inverter board that connects to the PC with a communication board for tools Note
(ex.) Renesas Flexible Motor Control Kit - MCK-RA6T2 (including a communication board for tools)

Type 3: Motor control inverter board that connects to the PC with a commercially available USB serial conversion module (isolated type). In this case, use the built-in type communication library (see 4. Communication Library.)

Note: Communication board for tools: MC-COM (Renesas, RTK0EMXC90S00000BJ) or W2002 (Desk Top Laboratories Inc.)

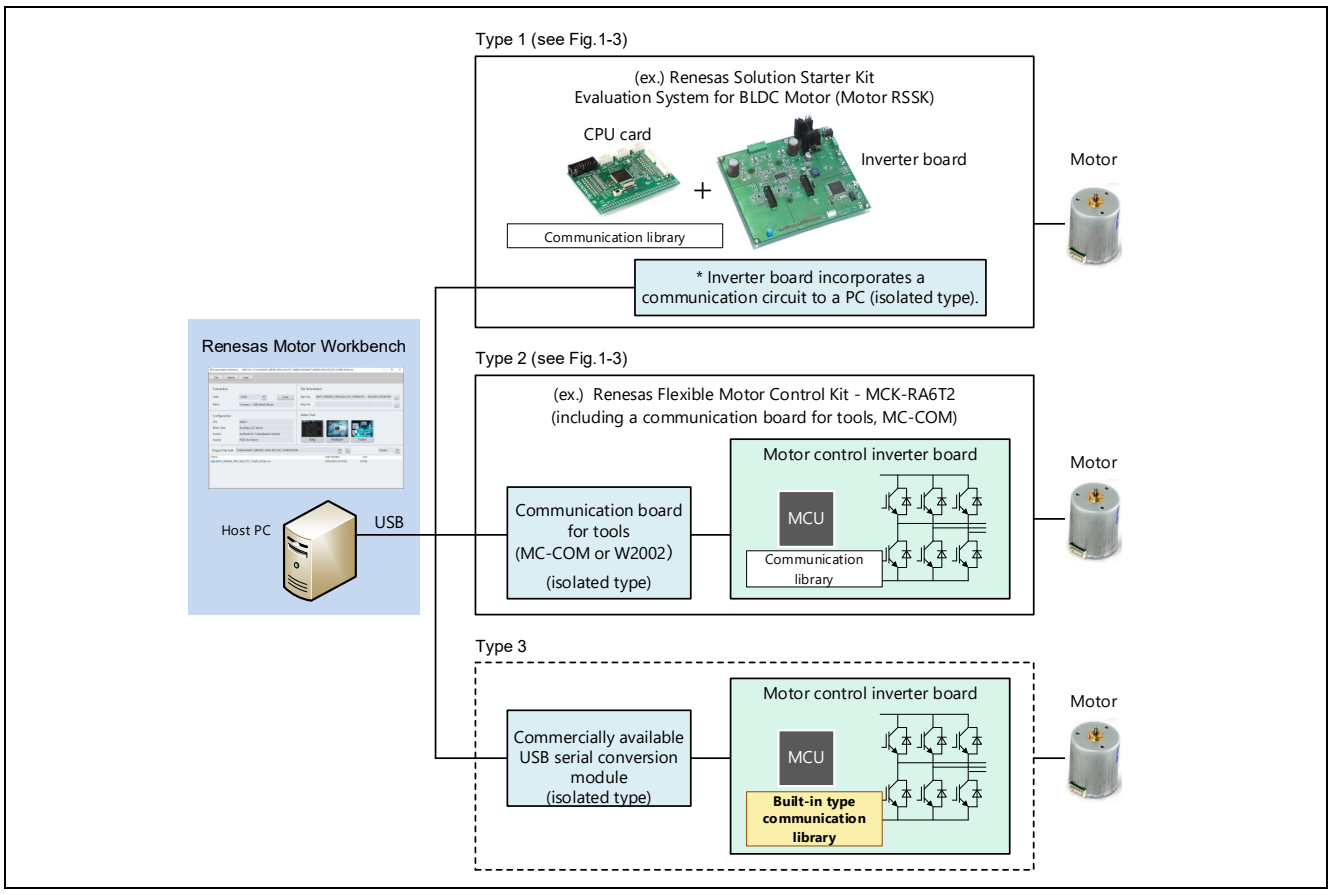
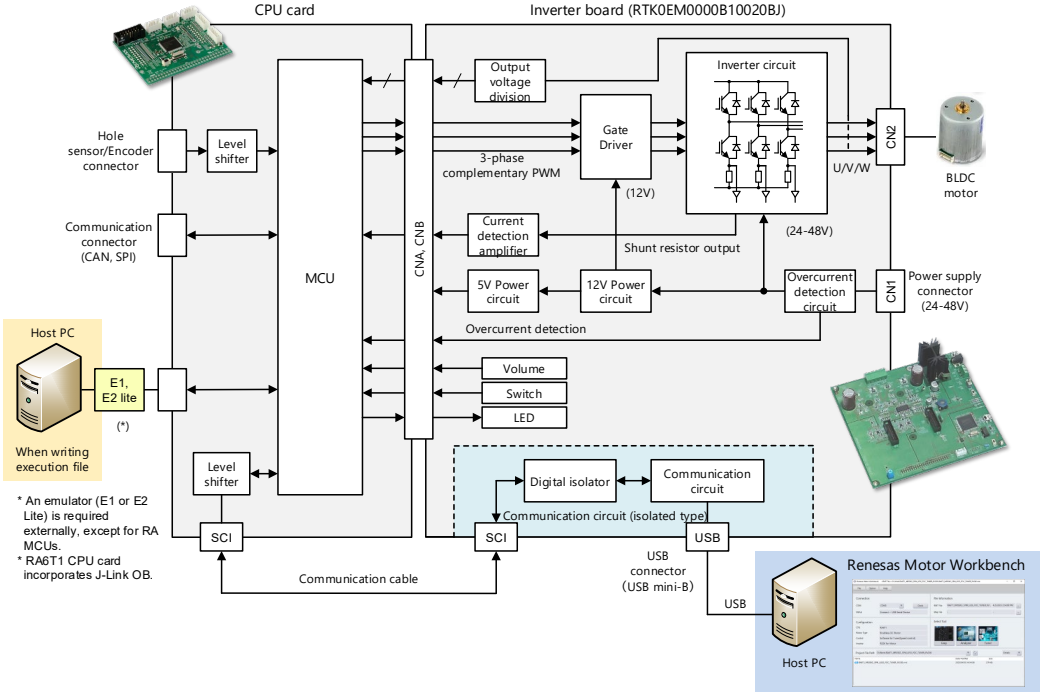
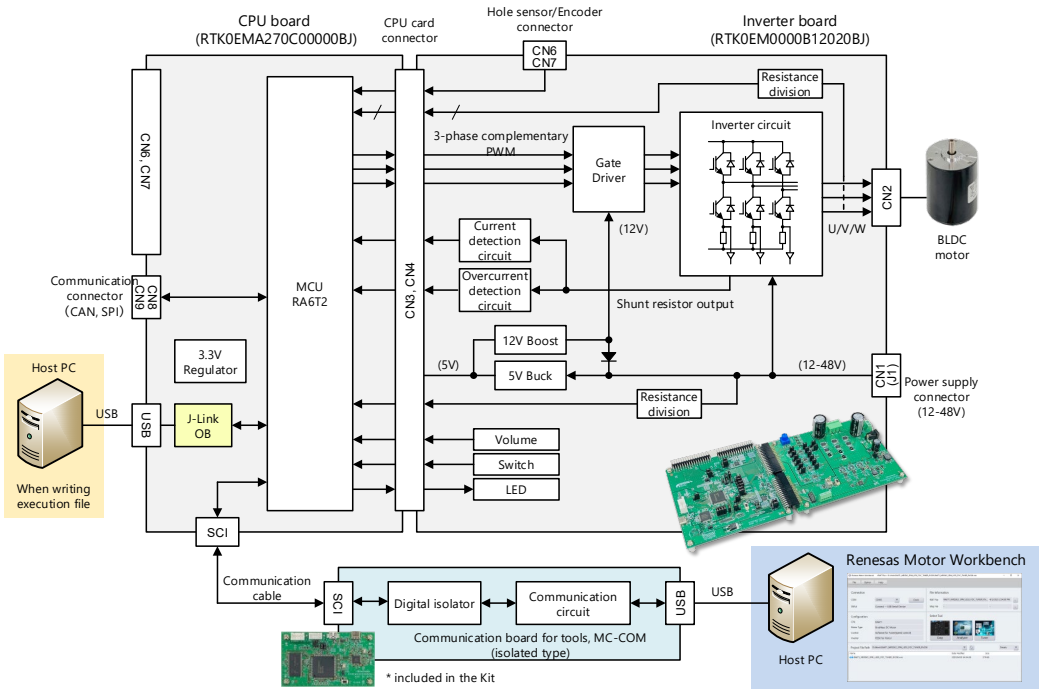


Figure 1-2 System configuration when using Renesas Motor Workbench

< Example of Type 1 > Renesas Solution Starter Kit - Evaluation System for BLDC Motor (Motor RSSK)



< Example of Type 2 > Renesas Flexible Motor Control Kit - MCK-RA6T2



Note: The hardware configuration varies for each MCU. This figure is for explaining the system configuration and is simplified in some parts. For details, refer to the documentation for each evaluation system.

Figure 1-3 Hardware configuration of Renesas Motor Evaluation System (example of Type 1 and 2 in Figure 1-2)

1.2.2 Host PC

Table 1-2 System requirement for Host PC

Item	Specification
PC	Windows PC with a USB terminal • Available disk capacity: 150 MB and more • RAM: 4GB and more • USB ports: 1 ch (USB 2.0)
OS	Windows10, 11
.NET	Microsoft .NET Framework 4.6.1 or higher
Application	Adobe Reader (required when using PDF documents and Report function) Microsoft Excel (required when using CsvEdit function in Commander window)

1.3 Specifications

Table 1-3 Renesas Motor Workbench specifications

Item		Specification
Target system	Supported MCU	RL78/G14, RL78/G1F, RL78/G24 RX13T, RX23T, RX24T, RX24U, RX26T, RX66T, RX72T, RX72M RA4T1, RA6T1, RA6T2, RA6T3, RA8T1
	Peripheral function	UART 1 ch, DTC (excluding RA MCUs), Port: TXD, RXD
Communication function	Target system	• Motor control inverter board with built-in type communication circuits (ex. Motor RSSK) • Motor control inverter board connected with a communication board for tools (ex. MCK-RA6T2) [Communication board for tools] — MC-COM (RTK0EMXC90S00000BJ, by Renesas) — W2002 (by Desk Top Laboratories Inc. Note) • Motor control inverter board connected with a commercially available USB serial conversion module (isolated type) (built-in type communication library is required.)
	PC	USB2.0
Internal function	Variable read/write	Maximum simultaneous selection: 255
	Waveform display	• Maximum simultaneous display: 12ch • Maximum number of data — 1024 (Motor RSSK, when using a built-in type communication library) — 100000 (when using a communication board for tools 'MC-COM')

Note: Desk Top Laboratories Inc (http://www.desktoplab.co.jp/)

1.4 Supportive Information (Reference)

Renesas web pages provide colorful information on motor control as follows. The page “Renesas Motor Workbench” introduces how to use Renesas Motor Workbench through videos.

- Motor Control
  https://www.renesas.com/en/key-technologies/motor-control

- Renesas Motor Workbench
  https://www.renesas.com/en/software-tool/renesas-motor-workbench

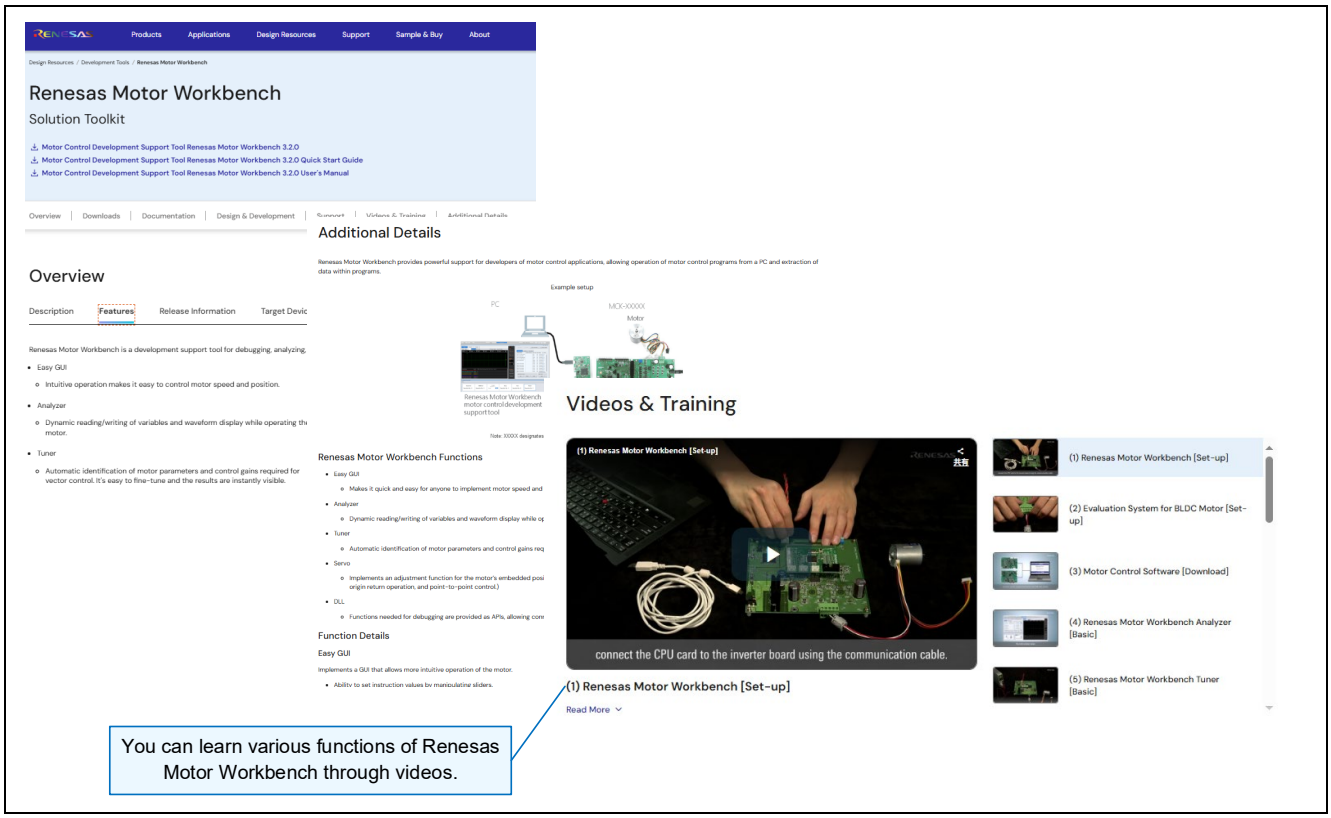
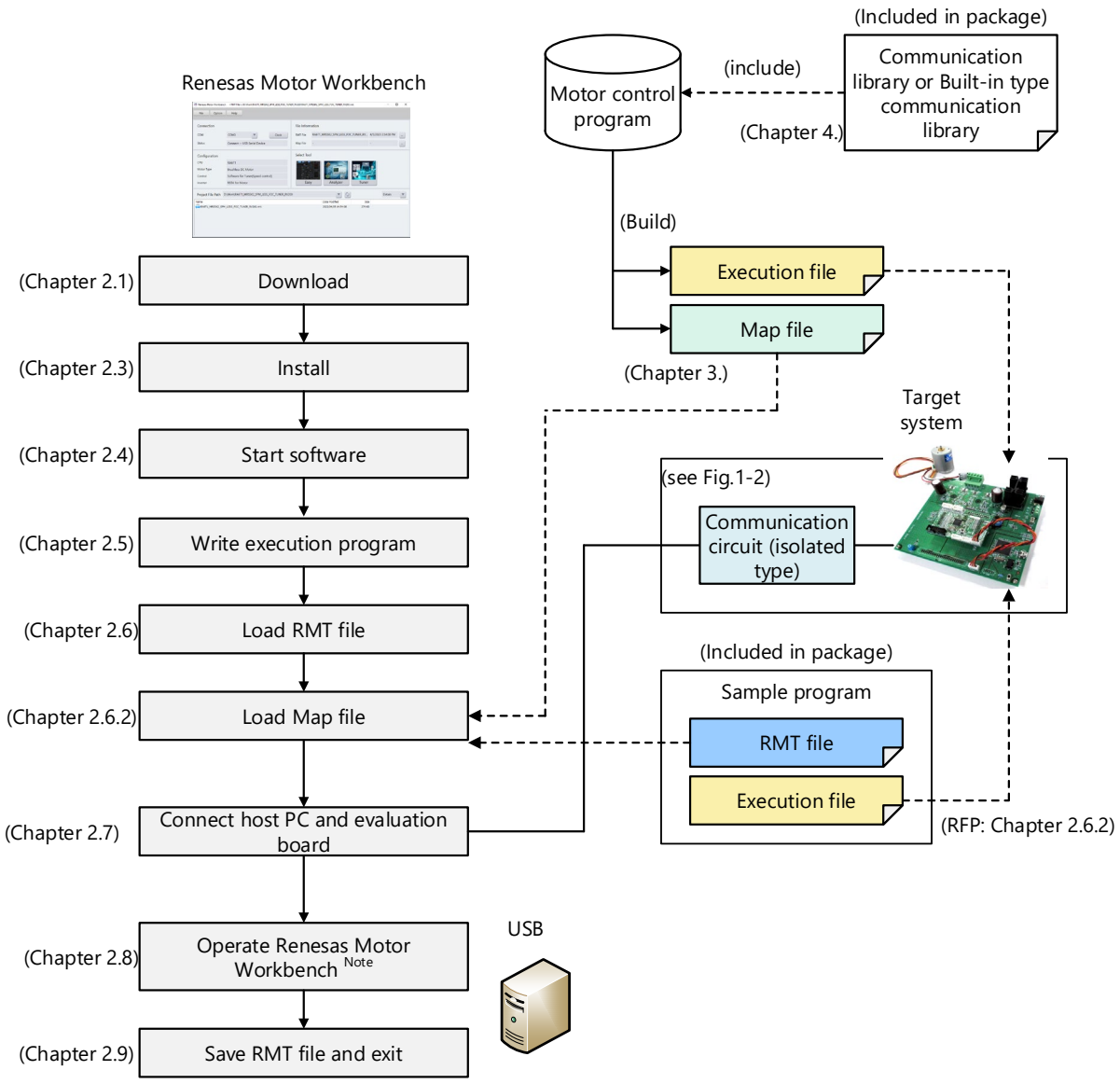


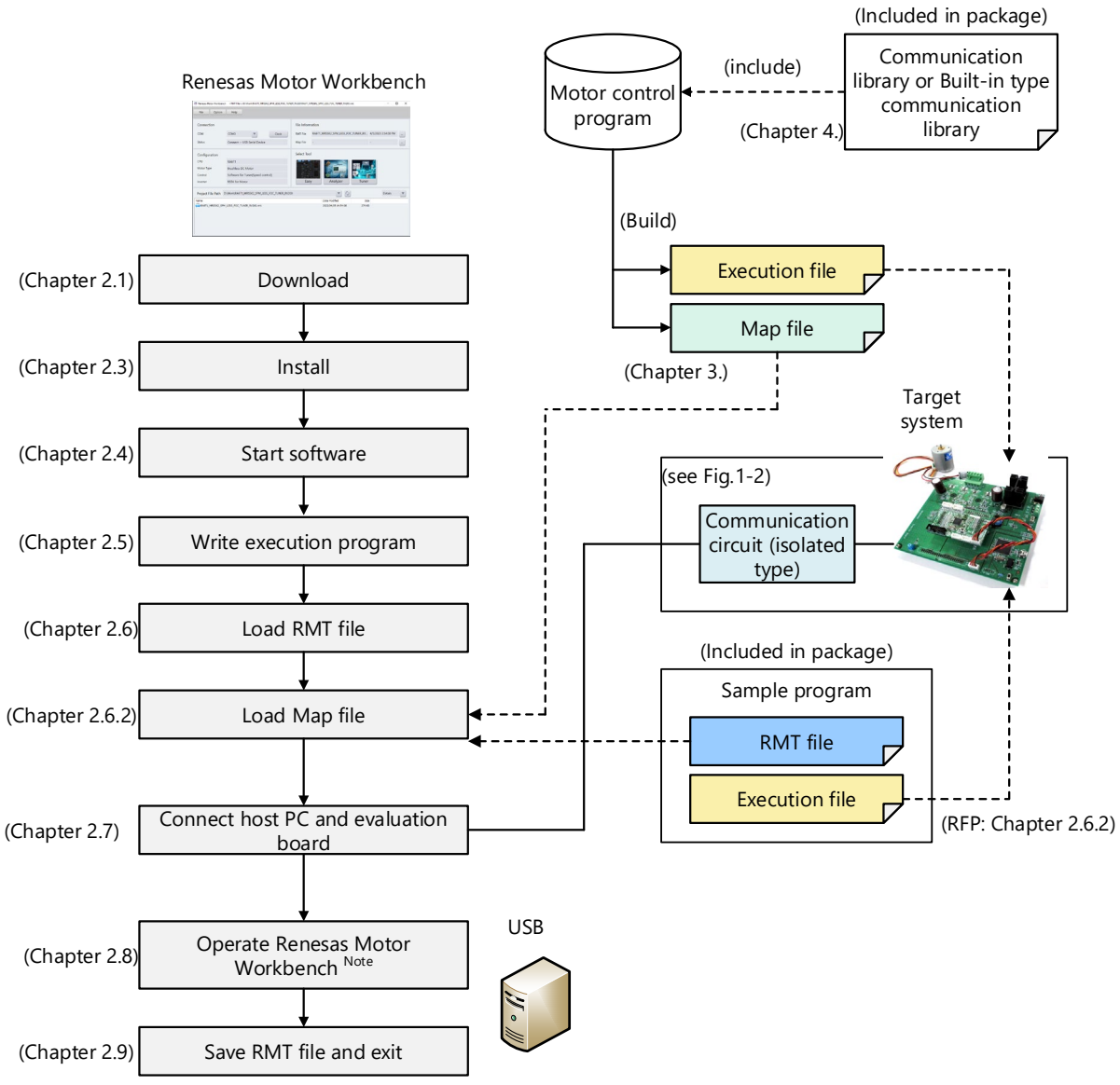
Figure 1-4 Renesas web page “Renesas Motor Workbench”

2. Installation and Execution



Note: For how to operate each function of Renesas Motor Workbench, see Renesas Motor Workbench User's Manual

Figure 2-1 shows the flow of the operation of Renesas Motor Workbench from installation to execution.



Note: For how to operate each function of Renesas Motor Workbench, see Renesas Motor Workbench User's Manual

Figure 2-1 Flow from installation to execution of Renesas Motor Workbench

2.1 Download

Download Renesas Motor Workbench from the Renesas WEB. The downloaded file is compressed in .zip file, so unzip it in advance.

2.2 Package Contents

The folder construction of the downloaded package and the file list of each folder are shown below.

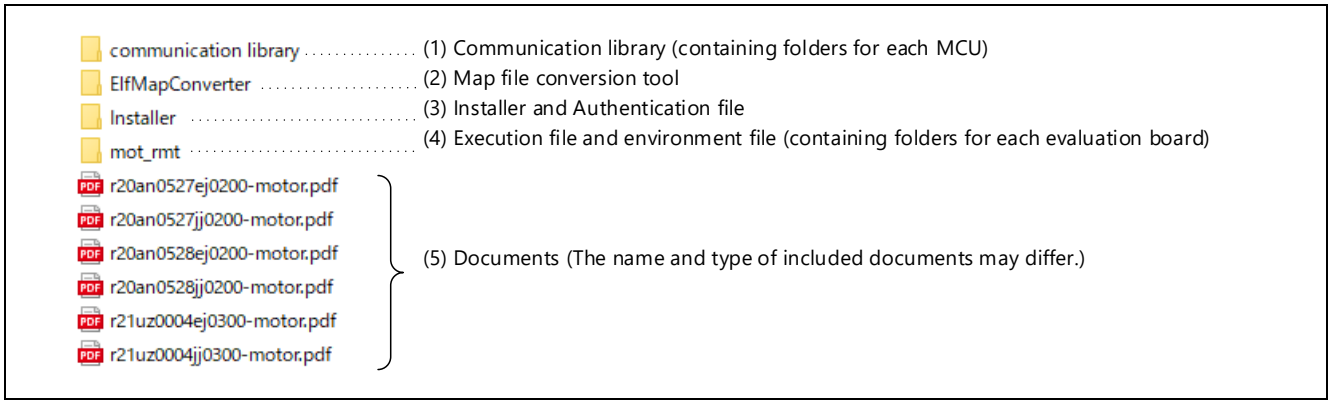


Figure 2-2 Folder construction of package (after unzipped)

Table 2-1 File list in each folder

No.	Folder	File	Description
(1)	Map file generation tool	ElfMapConverter.exe	Tool to generate a map file from elf file of RA MCU.
(2)	Communication library (containing folders for each MCU)	ICS_RX***.h ICS_RX***.obj	File for RX MCU building
		ICS2_RX***.h ICS2_RX***.lib	
		ICS2_RL78***.h ICS2_RL78***.lib	File for RL78 MCU building
		ICS2_RA***.o ICS2_RA***_Built_in.o ICS2_RA***.h	File for RA MCU building (ICS2_RA***_Built_in.o is a built-in type communication library)
(3)	Installer	renesas_motor_workbench_****.msi	Renesas Motor Workbench installer
(4)	Execution file and RMT file (containing folders for each evaluation board)	****.mot (RX, RL) ****.hex (RA)	Execution file to be written into Motor RSSK
		****.rmt	RMT file of Renesas Motor Workbench (file that saves environment settings)
(5)	Document	r21uz0004ej****-rmw-um.pdf r21uz0004jj****-rmw-um.pdf	Renesas Motor Workbench User's Manual
		r21qs0011ej****-rmw-qsg.pdf r21qs0011jj****-rmw-qsg.pdf	Quick Start Guide (this document)

2.3 Installation

To install Renesas Motor Workbench, execute installer “renesas_motor_workbench_***.msi” in the “Installer” folder.

After the installer starts, follow the instructions on the screen and install it.

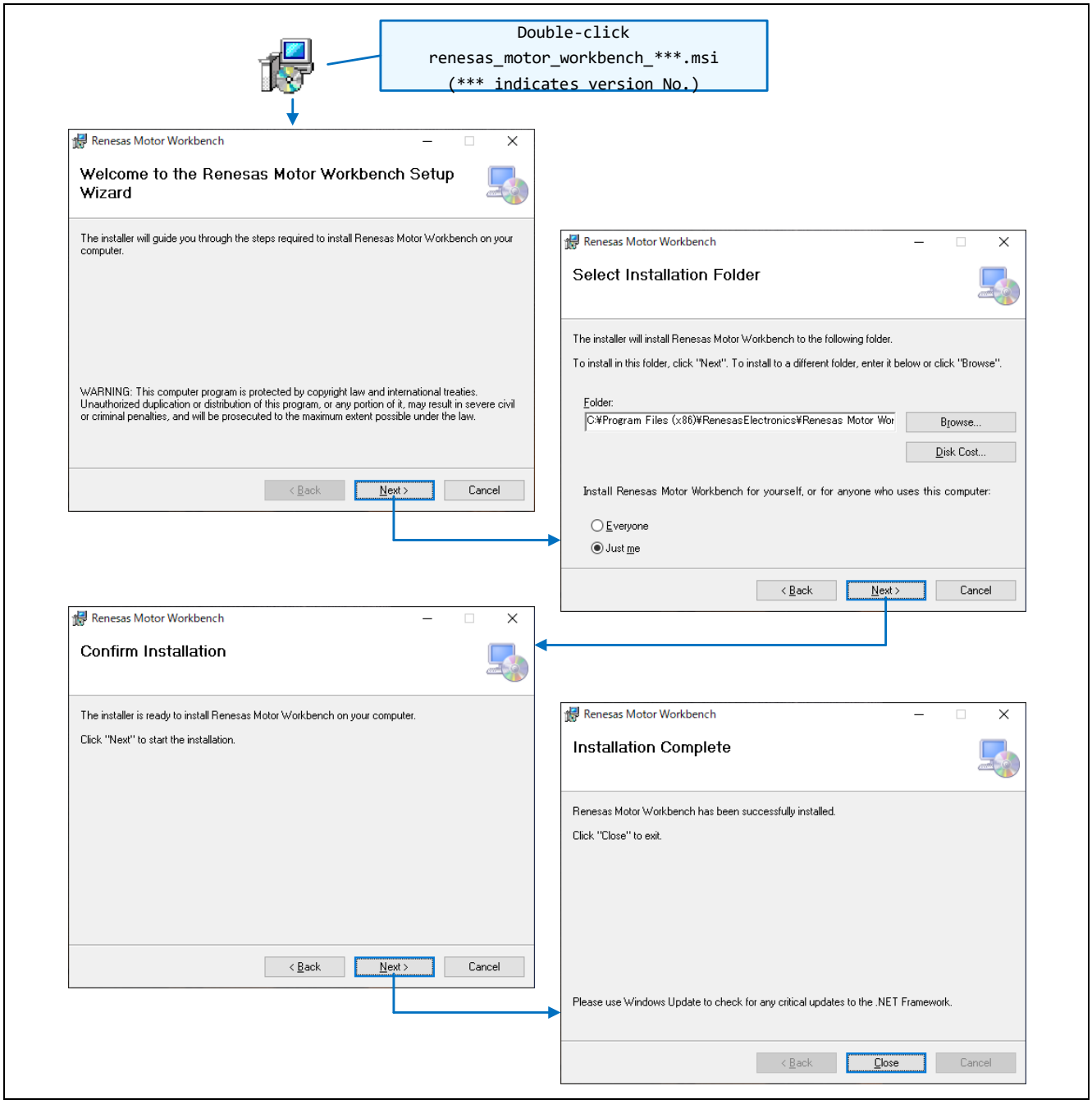


Figure 2-3 Installation steps

2.4 Starting Software

When Renesas Motor Workbench is installed, a shortcut icon will be shown to a desktop. Double-click the icon for “Renesas Motor Workbench” to start it.

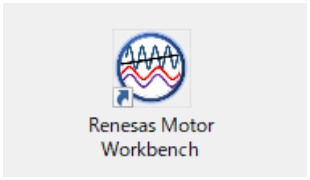


Figure 2-4 Shortcut icon for starting

To start Renesas Motor Workbench from the Windows start menu, select “Renesas Electronics” > “Renesas Motor Workbench”.

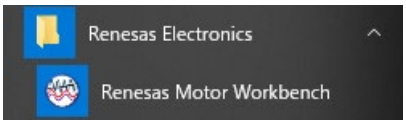


Figure 2-5 Starting from start menu

You can also start Renesas Motor Workbench by double-clicking a RMT file in the Windows Explorer. In this case, it starts with the selected RMT file already loaded.

Figure 2-6 Starting by double-clicking RMT file

If you cannot start Renesas Motor Workbench, installation might be failed. Uninstall and install it again following the procedure described in the previous section.

2.5 Writing Execution Program

Write an execution file generated by building the project of the motor control program or the execution file included in this package (.mot or .hex) into the target MCU. To write the file, use the tool such as IDE (CS+ or e^2 studio) or “Renesas Flash Programmer (RFP)”.

Note: The methods of connecting a PC to a CPU board and of supplying power differ between when the execution program is being written and when Renesas Motor Workbench is being used. For details, refer to the manuals for each evaluation system.

2.5.1 When building and writing sample program with IDE

This section describes how to write the sample program provided by Renesas Web by using Integrated Development Environment (IDE), e^2 studio. (The RA MCU is used as an example here.)

For details on building, refer to the Application Note attached with the sample program.

(1) Importing project

Get the sample program from Renesas Web and import it into e^2 studio as the following steps.

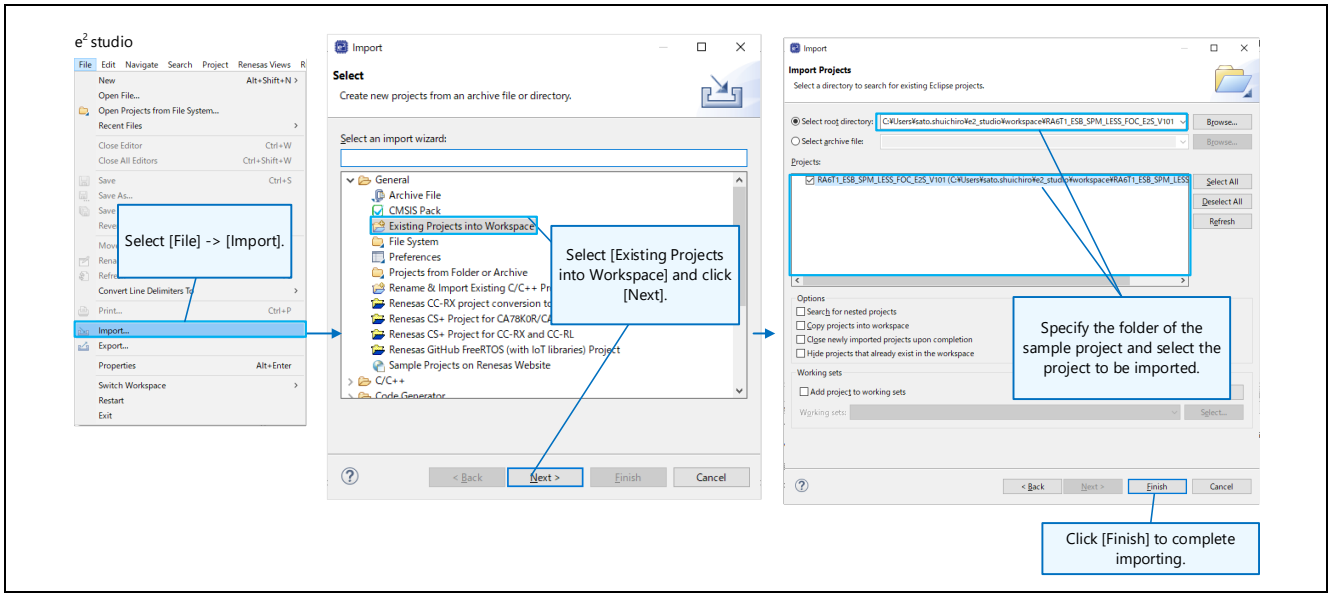


Figure 2-7 Importing project (e^2 studio)

(2) Setting toolchain

In the properties of the project, set the tool configuration for your environment.

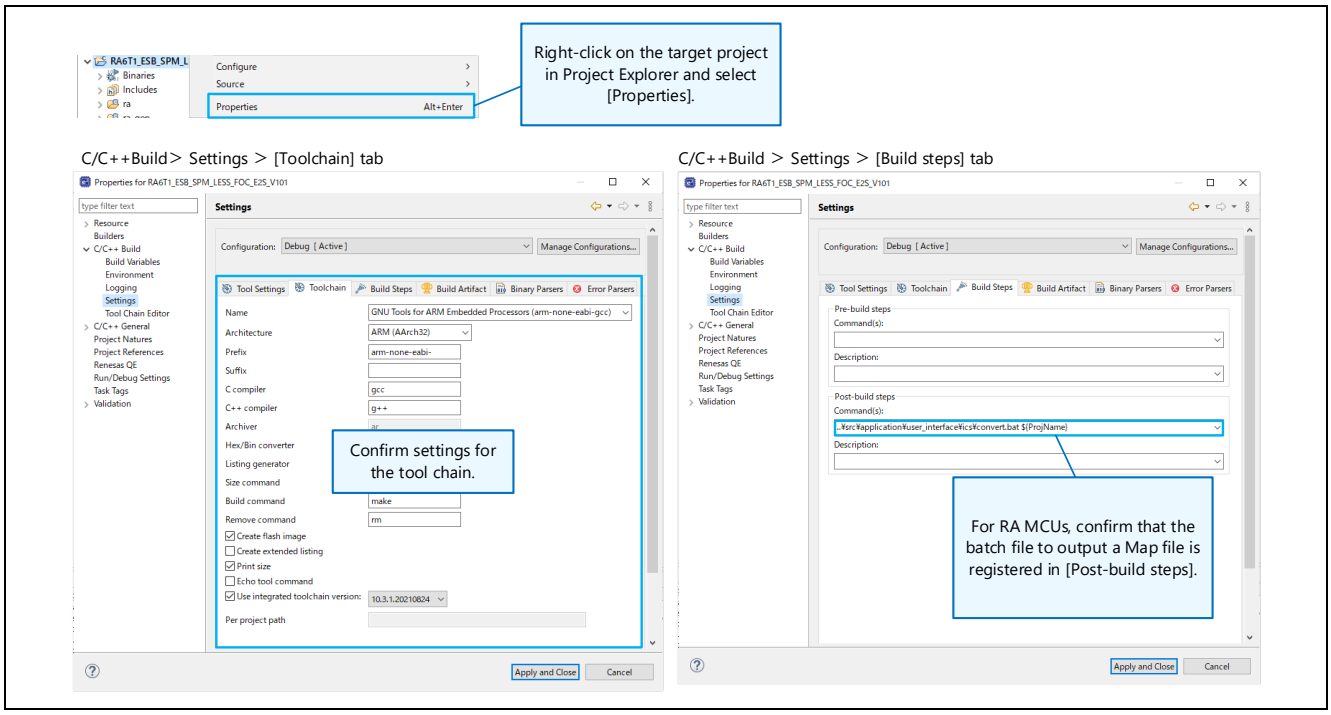


Figure 2-8 Setting toolchain

(3) Building

Build the project. Confirm that there are no errors in the build-result.

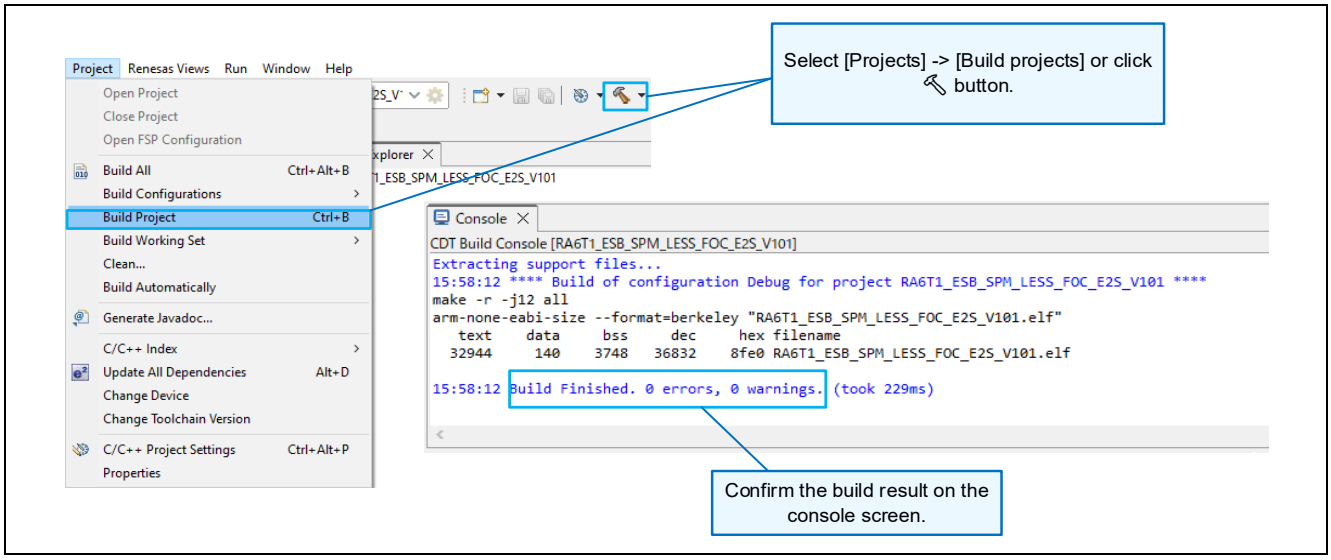


Figure 2-9 Building project

(4) Writing into evaluation board

Connect a PC to the evaluation board with USB cable (when the CPU board has a J-Link interface), write the program into the evaluation board.

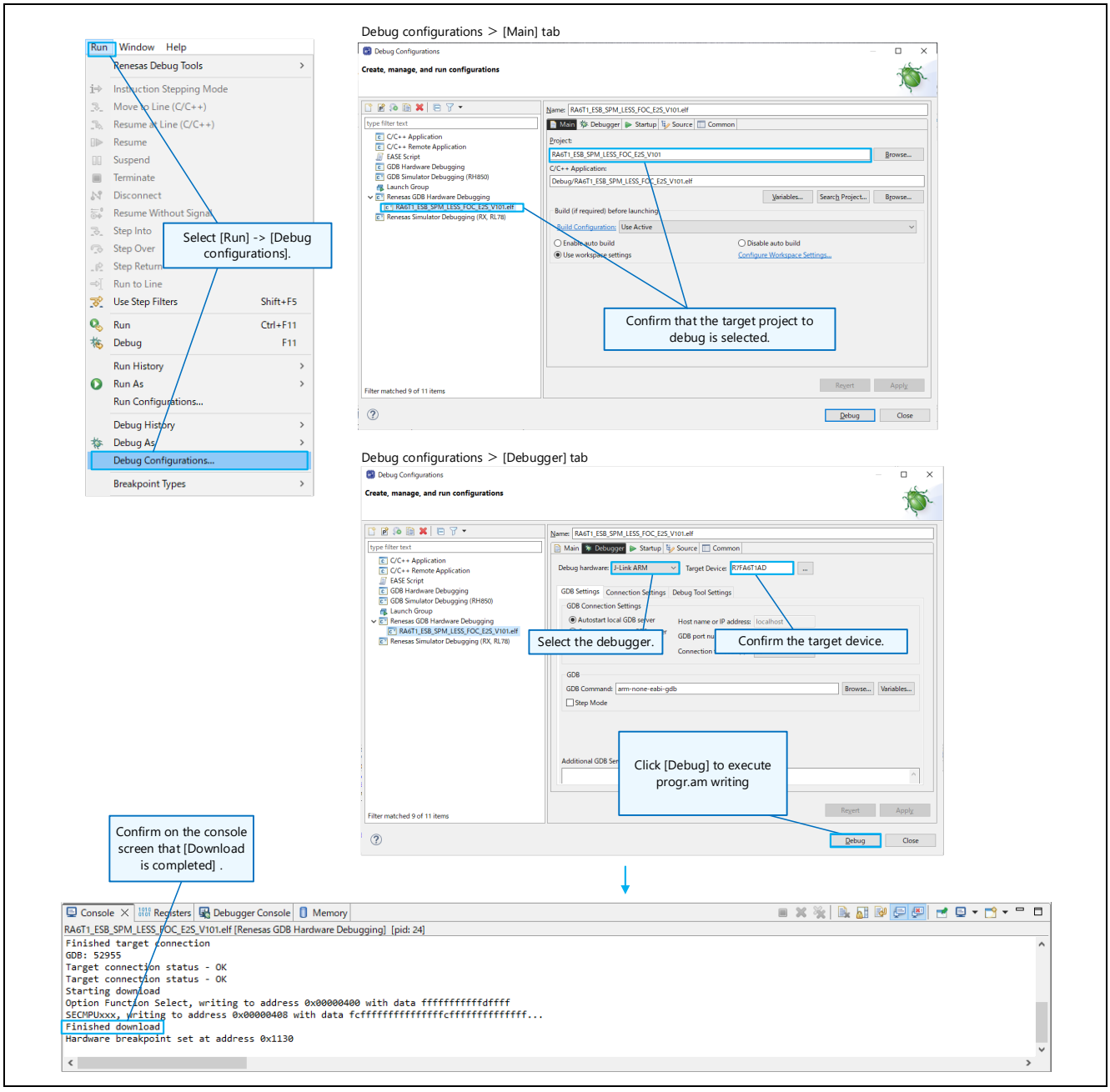


Figure 2-10 Writing program into evaluation board

(5) Disconnecting evaluation board

After writing the program, select “Run” > “Disconnect” to disconnect the evaluation board and remove the USB cable.

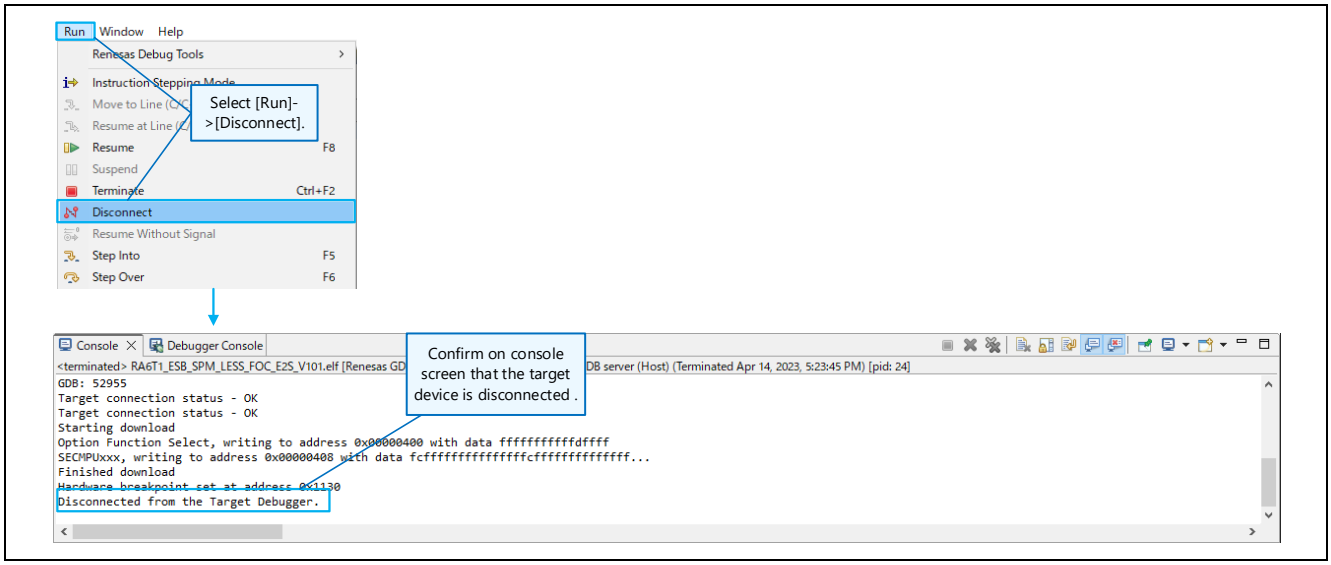


Figure 2-11 Disconnecting evaluation board

If the project has been rebuilt, load the Map file which was generated at the same time as the latest execution file into Renesas Motor Workbench. For how to generate a Map file, see chapter 3, and for how to load the file, see chapter 2.6.2.

Note: If executing the program (driving a motor) using Renesas Motor Workbench after writing the program, use a communication dedicated circuit (isolated type) to connect to the target board.

2.5.2 When writing with Renesas Flash Programmer (RFP)

This section describes how to write the execution file included in this package into the target board using Renesas Flash Programmer (RFP). (You can download RFP from Renesas Web.) Configure settings for communication to the CPU board according to the system to be used.

Load the RMT file that is included as well. (For how to load RMT files, see chapter 2.6.1.)

(1) Launch the RFP tool and create a new project.

(2) Specify the communication method and connect a PC and the target board.

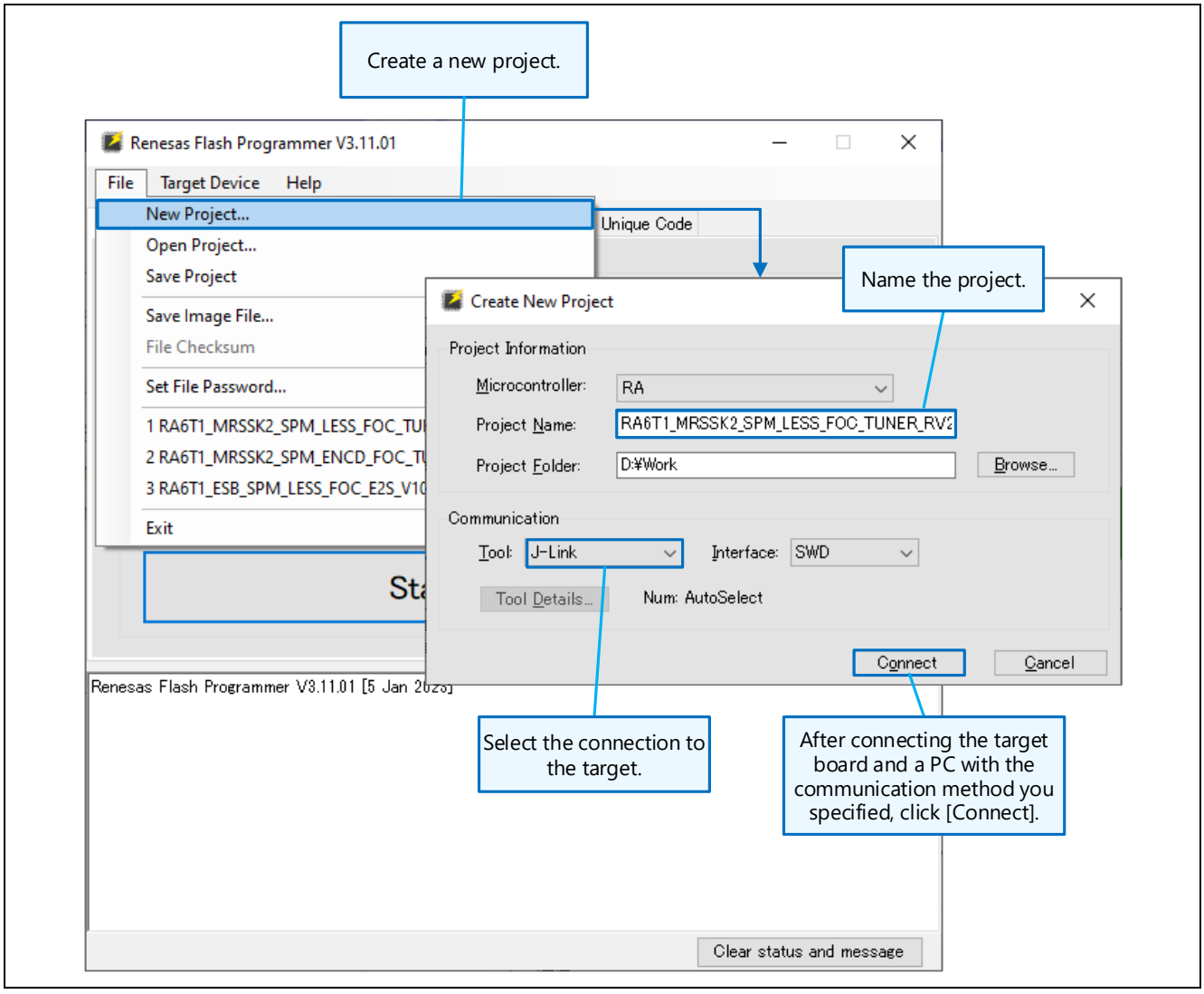


Figure 2-12 Writing execution program with Renesas Flash Programmer (1/2)

(3) Specify the program to write and start writing.

(4) After writing is normally completed, terminate RFP and remove the target board from the PC.

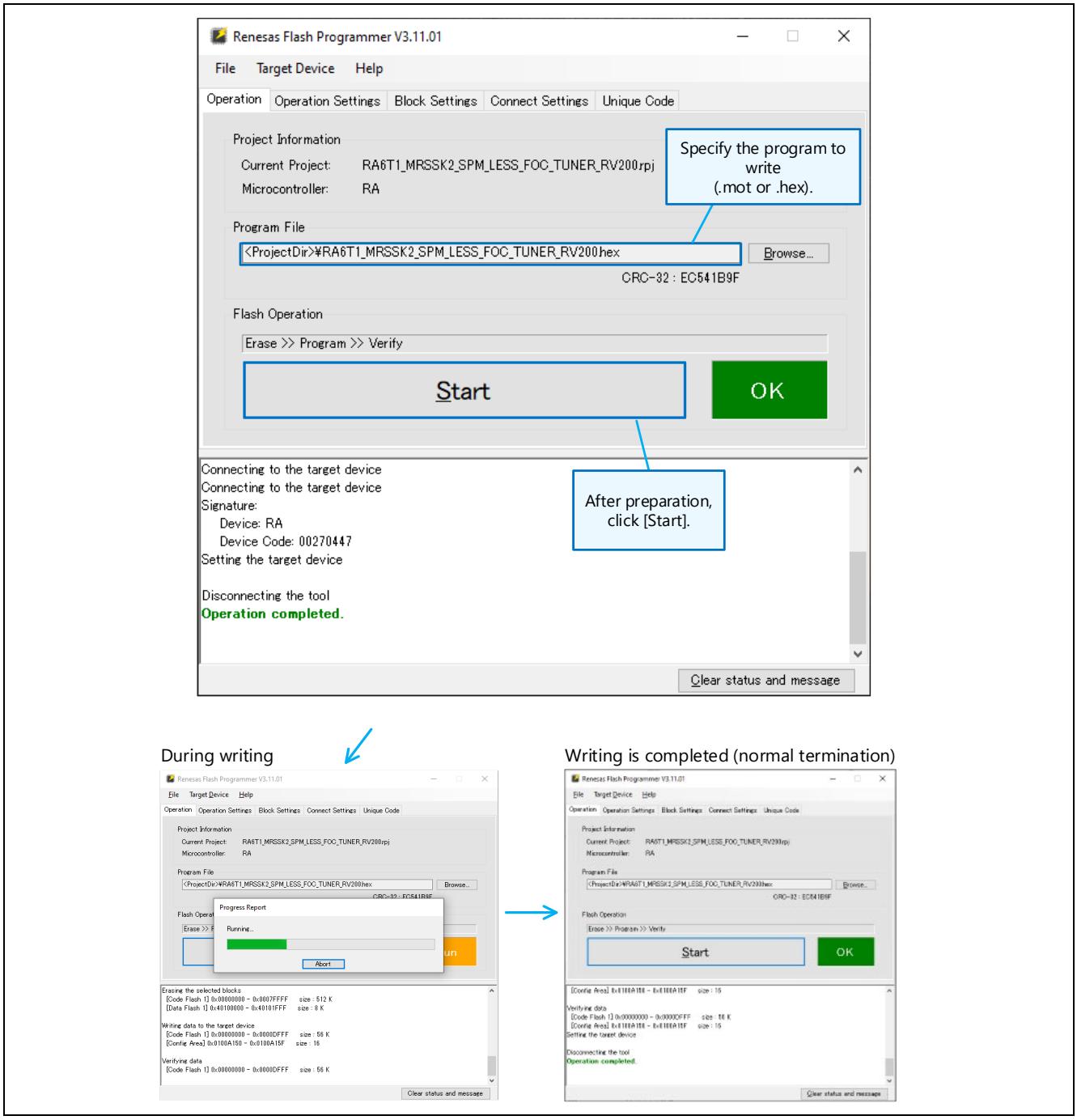


Figure 2-13 Writing execution program with Renesas Flash Programmer (2/2)

Note: If executing the program (driving a motor) using Renesas Motor Workbench after writing the program, use a communication dedicated circuit (isolated type) to connect to the target board.

2.6 Loading Variable Information

Renesas Motor Workbench loads the Map file which is outputted in generation of the execution program and gains access to the global variables. If the program has been changed and rebuilt, it is necessary to load the Map file again.

Renesas Motor Workbench can save and load the variable information and environment information including settings for tools as an RMT file (An RMW project file).

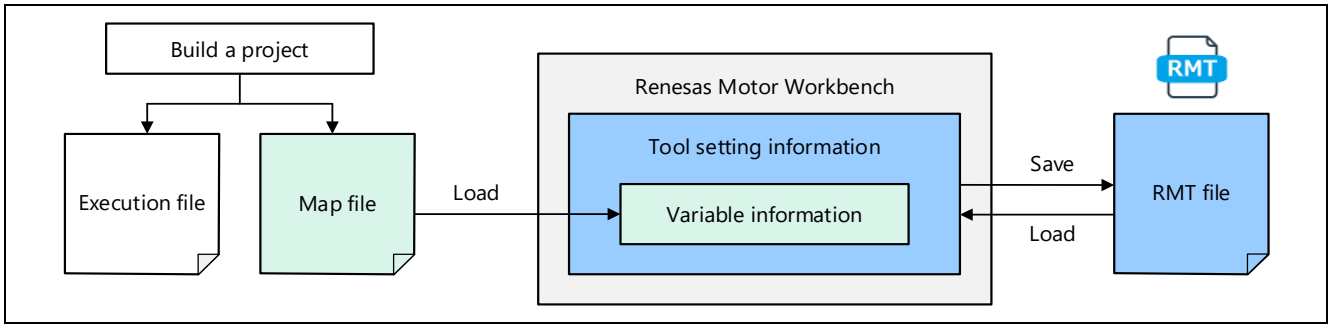


Figure 2-14 Map file and RMT file (An RMW project file)

2.6.1 Loading RMT File

When an RMT file (storing both variable information and tool setting information) is provided in the sample program included in this package or in the sample program provided by Renesas, load the RMT file by following one of the operations below.

(a) Use Select Navigation function (displayed from Help menu) and load the RMT file.
(b) Select “Open RMT File” of the Main Window’s File menu and load the RMT file.
(c) Click the “…” button next to the RMT File field in Main Window’s File Information and load the RMT file.
(d) Specify the project folder in Main Window’s Project File Path and double-click the RMT file displayed on the File List to load it.
(e) Drug & drop an RMT file on a PC to Main Window’s File List and doble-click to load it. (In this case, the RMT file is copied to the project folder on the PC.)

Note that if the sample program has been changed and built, it is necessary to load the Map file generated in building into Renesas Motor Workbench again.

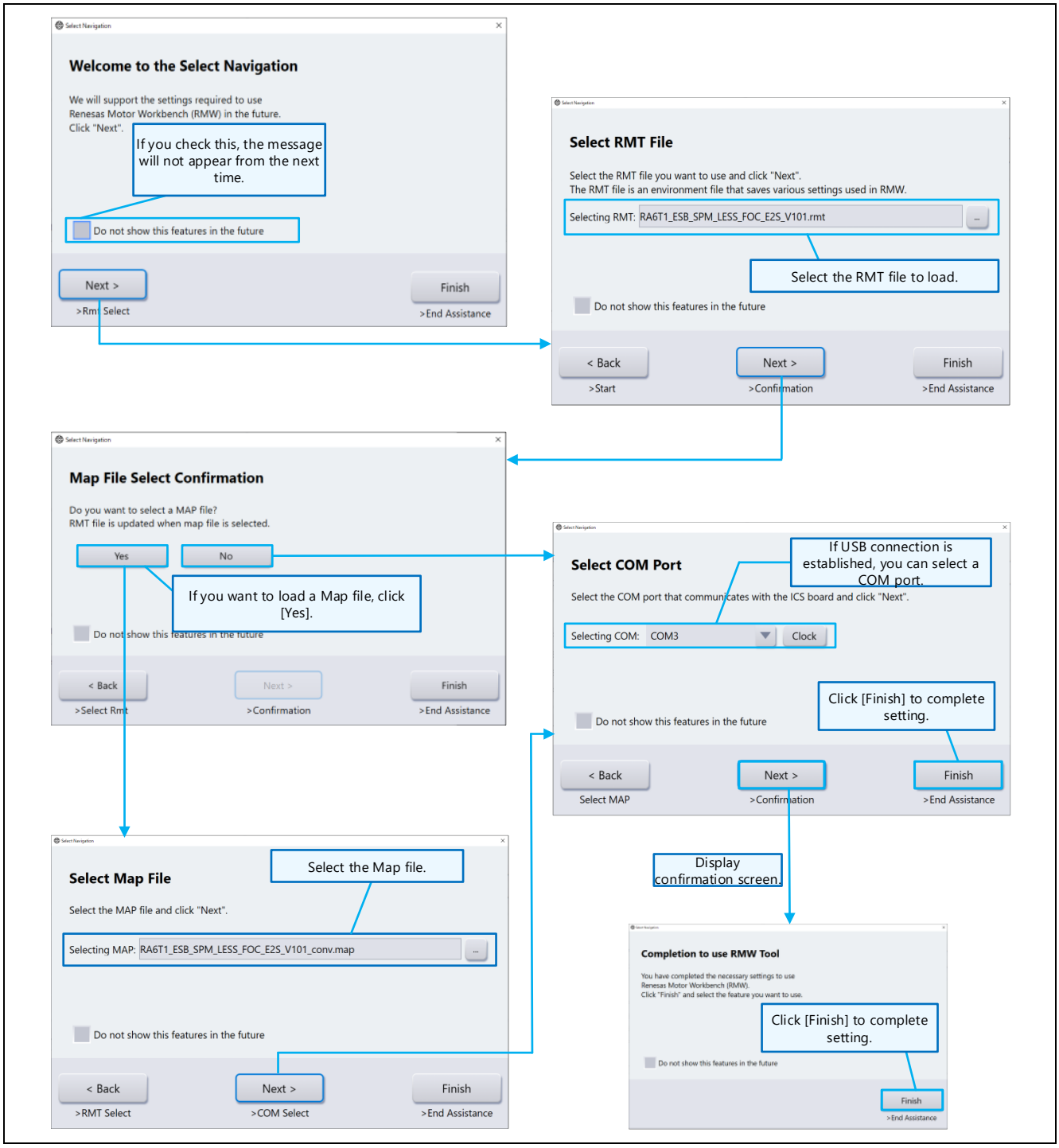


Figure 2-15 ① Loading RMT file using Select Navigation function

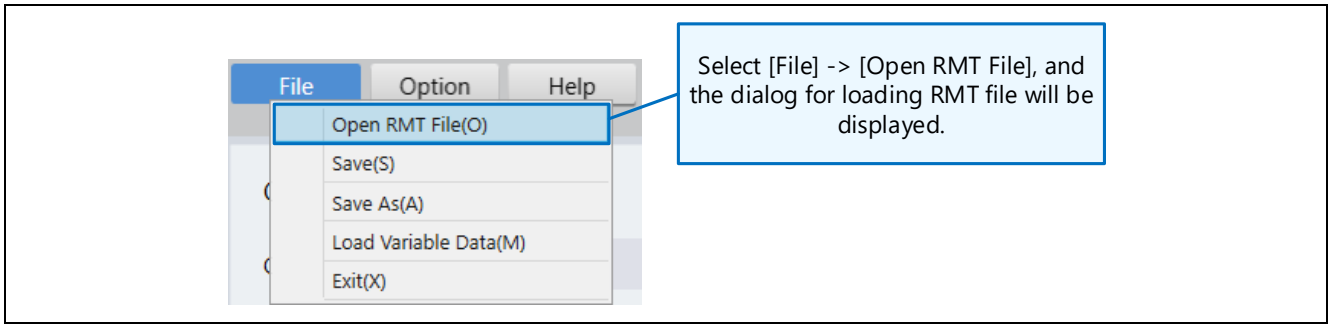


Figure 2-16 ② Loading RMT file from "Open RMT File" of File menu

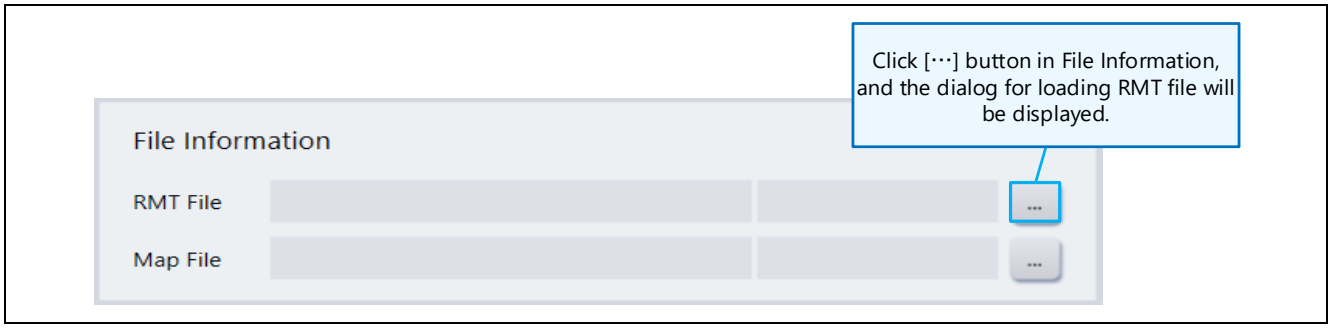


Figure 2-17 ③ Loading RMT file from Main Window's File Information

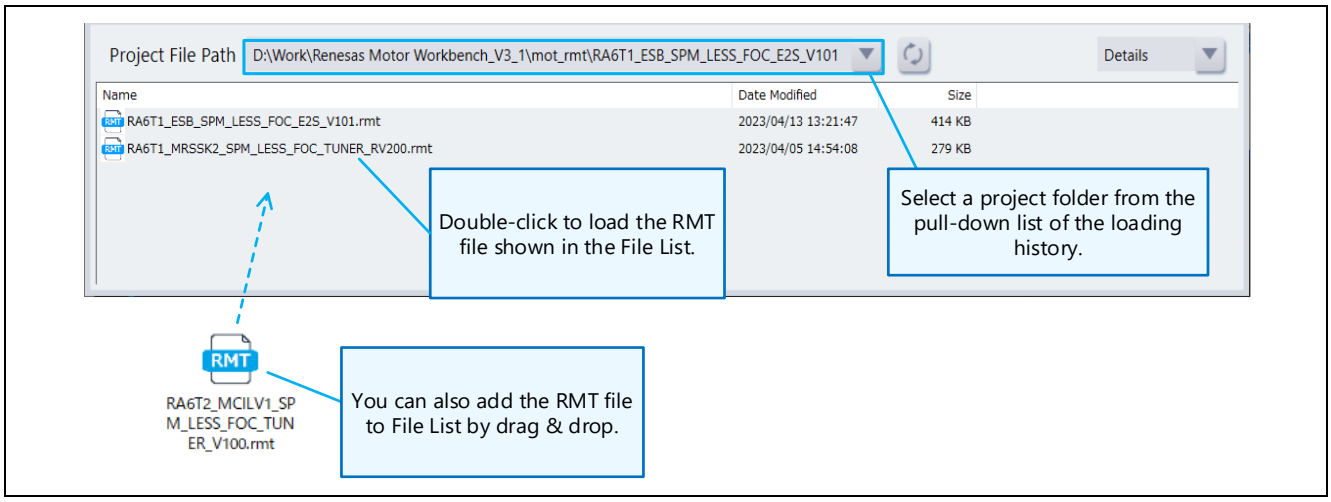


Figure 2-18 ④⑤ Loading RMT file from Main Window's File List

2.6.2 Loading Map File

When using a user program (when an RMT file doesn't exist), you need to load a Map file into Renesas Motor Workbench first.

You must use the Map file that was created at the same time as the execution file in building. If the program has been changed and rebuilt, it is necessary to load the Map file again as well. For generating a Map file, refer to "3. How to Generate Map File".

Steps for loading Map file:

(1) Load a Map file by one of the operations below. (For how to operate, refer to the figures in 2.6.1 Loading RMT File.)

(a) Use Select Navigation function (displayed from Help menu) to load the Map file.

(b) Select "Load Variable Data" from the Main Window's File menu and specify the Map file to load.

(c) Click the "…" button next to the Map File filed in Main Window's File Information to load it.

For RX:
Load a file with a *.map extension.

For RA:
In the file selection dialog, select "All Files (*.*)" for file type, and load a file with a *.rmap extension. If "MAP Files (*.map)" is selected for file type, a file with *.rmap extension is not displayed.

However, you can change the extension from *.rmap to *.map by the method described in the Tips of "3.3 [RA] In e2 studio Environment" or by modifying the extension directly. In this case, load the file with *.map extension that you have modified without changing the file type in the file selection dialog.

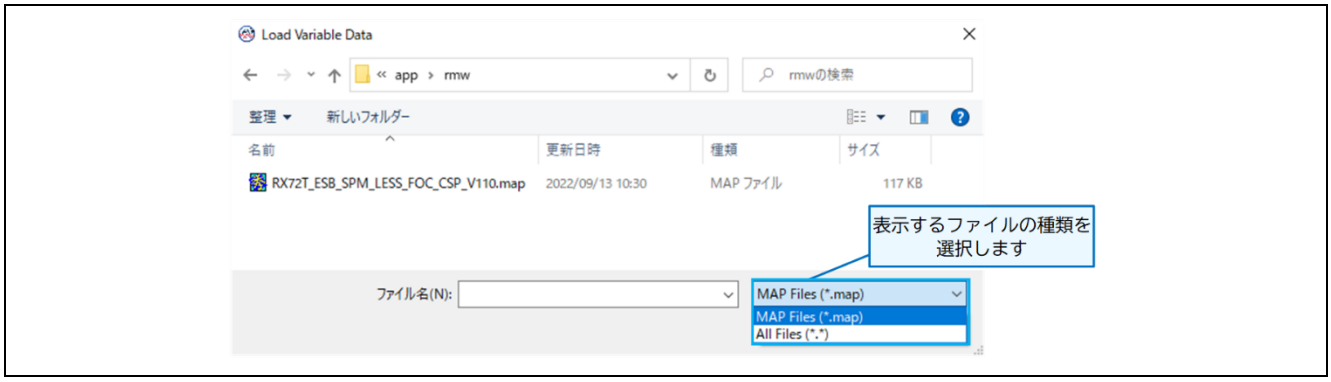


Figure 2-19 Map file selection dialog box

(2) After loading the Map file, "Use Setting Form" screen is displayed. If "Change in Data Type" or "Change to Array" has been set in the Option Dialog (displayed by selecting "Option Dialog" from Option menu) in loading, these settings are reflected to the list.

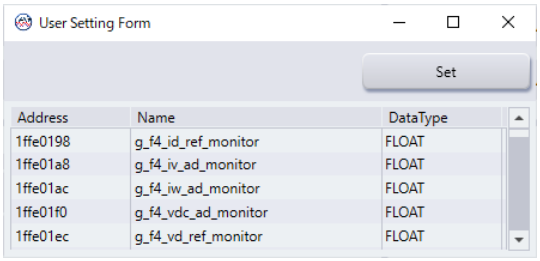


Figure 2-20 User Setting Form screen

You can also change the Data Type settings on the User Setting Form screen.

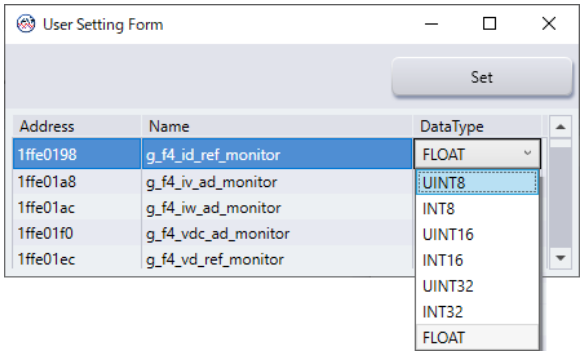


Figure 2-21 User Setting Form screen (change Data Type settings)

(3) When you press “SET” button, the variable information is loaded into Renesas Motor Workbench.

Note that after loading variable information by pressing “SET” button, you cannot change the Data Type using the function of Renesas Motor Workbench. To make a change, load the variable information with “Load Variable Data” function again.

2.7 Connecting Host PC and Evaluation Board

2.7.1 Connecting via USB

For system requirements when Renesas Motor Workbench is used, refer to chapter 1.2.

Supply power to the evaluation board with the program written and connect between a PC and the evaluation board with a USB cable.

When they are connected normally, the connected “COM**” (** indicates the port number) is displayed in the pull-down list for COM selection in the Main Window’s Connection. This means the status where COM communication between the PC and the board is established and a communication can start.

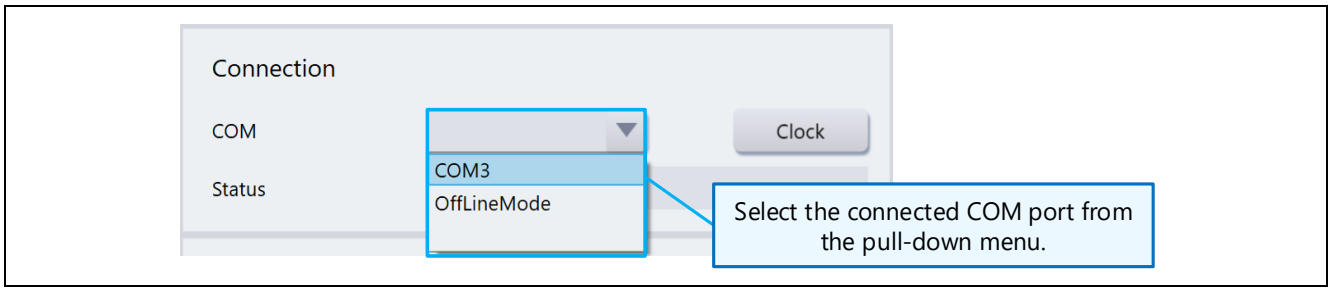


Figure 2-22 COM selection list of Main Window’s Connection

2.7.2 Starting USB Communication

When you select “COM**” from the COM selection list, the communication connecting process starts. The connecting process status is displayed in “Status”. When the connecting process is normally completed, “Connect – USB Serial Port” will be displayed in “Status”. In this state, tools are enabled.

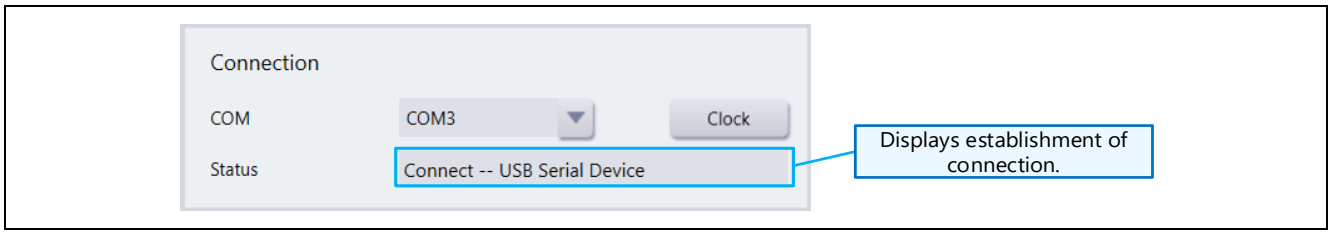


Figure 2-23 Connection status view after connection establishment

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.

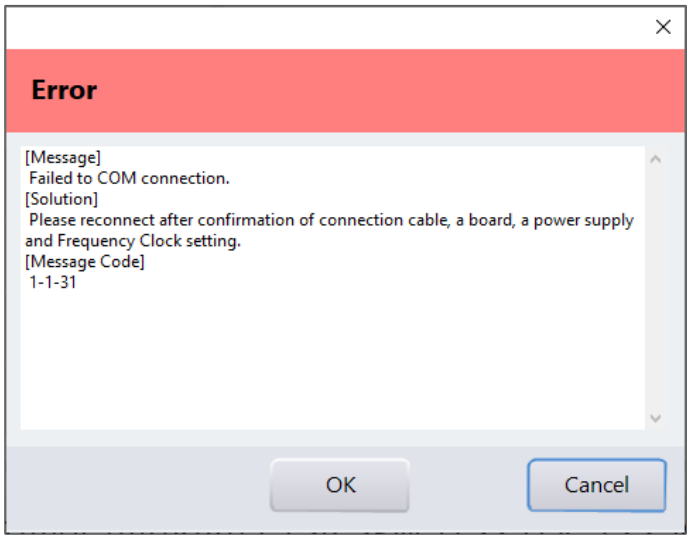


Figure 2-24 Error message under COM connection

2.7.3 When using Communication Board for Tools (setting clock frequency)

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 the "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.

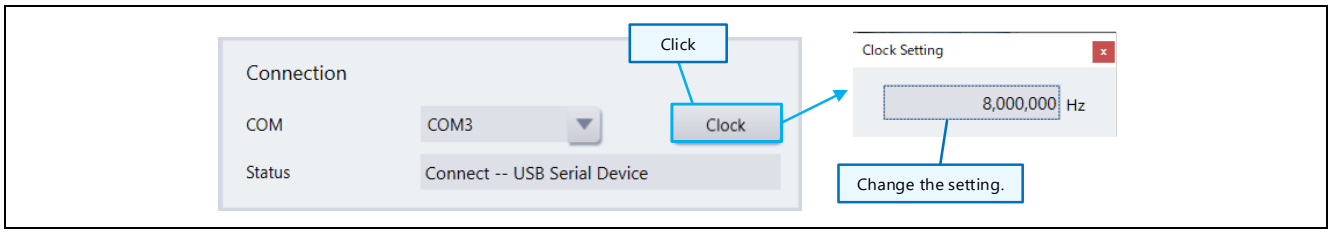


Figure 2-25 Clock Setting dialog

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 2-2 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)

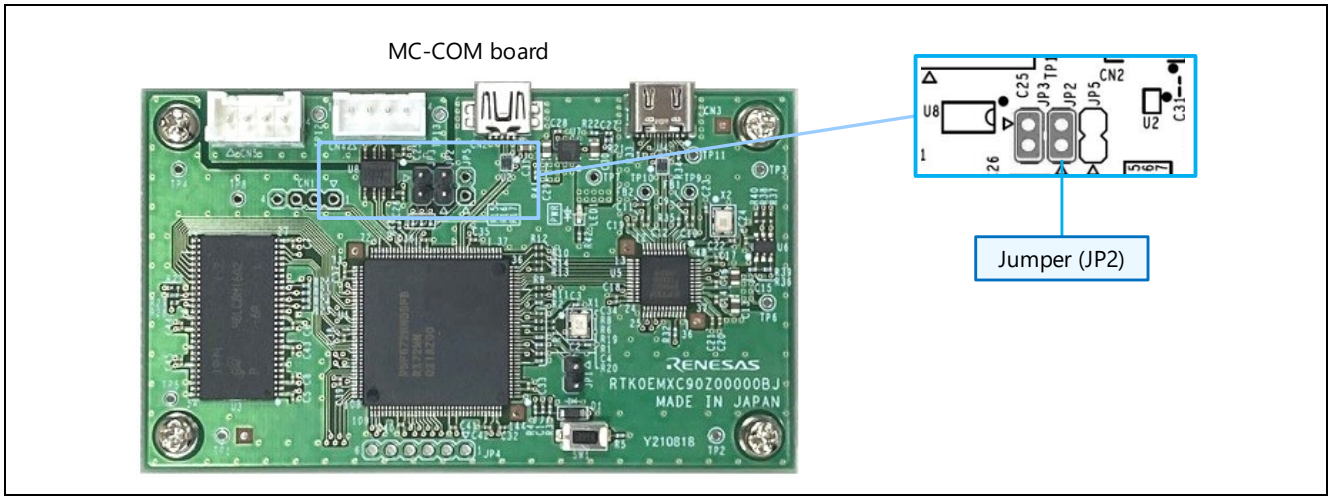


Figure 2-26 JP2 of MC-COM (communication board for tools)

Table 2-4 and 2-5 shows maximum communication rate on each MCU under conditions shown in this table.

Conditions

- Configure CPU clock frequency at maximum
- Applicable baud rate for MC-COM

Table 2-1 Maximum communication rate (for RX microcontroller)

Product name	CPU clock frequency	Supplied clock frequency (UART)	Baud rate	Sampling interval	Lib API(ics2_init) configured value	Lib API(ics2_init) configured value
					4th argument char speed	5th argument char mode
RX72T	200MHz	50MHz	6.25Mbps	50us/4ch	0	2
RX66T	160MHz	40MHz	5Mbps	50us/4ch	0	2
RX26T	120MHz	60MHz	7.5Mbps	50us/4ch	0	2
RX24T/U	80MHz	40MHz	5Mbps	50us/4ch	0	2
RX23T	40MHz	40MHz	5Mbps	50us/4ch	0	2
RX13T	32MHz	32MHz	1Mbps	200us/4ch	3	2

Table 2-2 Maximum communication rate (for RA microcontroller)

Product name	CPU clock frequency	Supplied clock frequency (UART)	Baud rate	Sampling interval	Lib API(ics2_init) configured value	Lib API(ics2_init) configured value
					3rd argument char speed	4th argument char mode
RA8T1	480MHz	120MHz	10Mbps	50us/4ch	1	2
RA6T3	200MHz	100MHz	16.6Mbps	50us/4ch	0	2
RA4T1	100MHz	100MHz	16.6Mbps	50us/4ch	0	2
RA6T2	240MHz	120MHz	10Mbps	50us/4ch	1	2
RA6T1	120MHz	120MHz	15Mbps	50us/4ch	0	2

2.7.4 When using Built-in type Communication Library (setting baud rate)

When using the built-in type communication library, set the baud rate by the following operation.

Select "Baud rate Dialog" from Main Window's "Option" menu to display the BaudrateSetting 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.

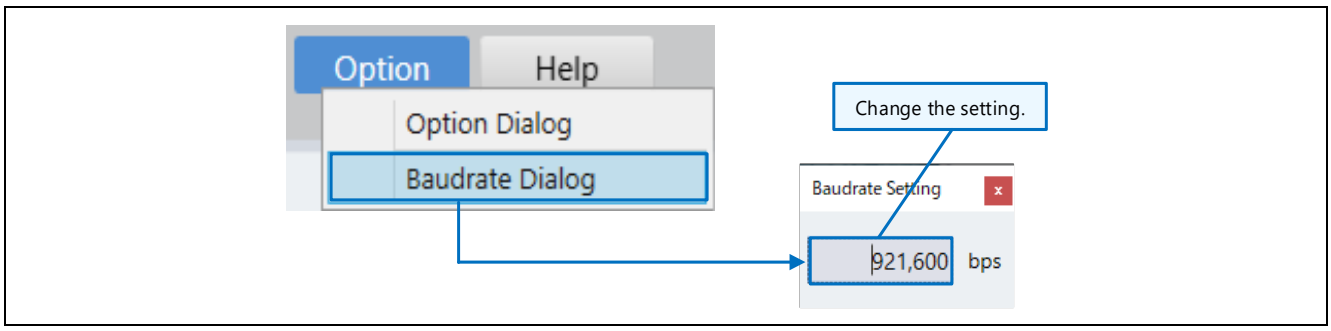


Figure 2-27 Setting baud rate

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 to the connected USB serial conversion board from the PC. You can check the configurable values in the properties of the COM port (port setting) from Device Manager of Windows control panel.

2.7.5 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.
- If no connectable COM is displayed, check the authentication status. (Select “Load Authentication File” from “Help” menu and check the message on the authentication screen.)
- 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.

2.8 Description of Operation

2.8.1 Launching Internal Tool

When a Map or RMT file is loaded and the target is connected normally, icons for available tools will be shown in Main Window's Select Tool. (The view of Main Window and available tools vary depending on the sample program.)

Click an icon in Select Tool to start the tool, and the window for the tool will open.

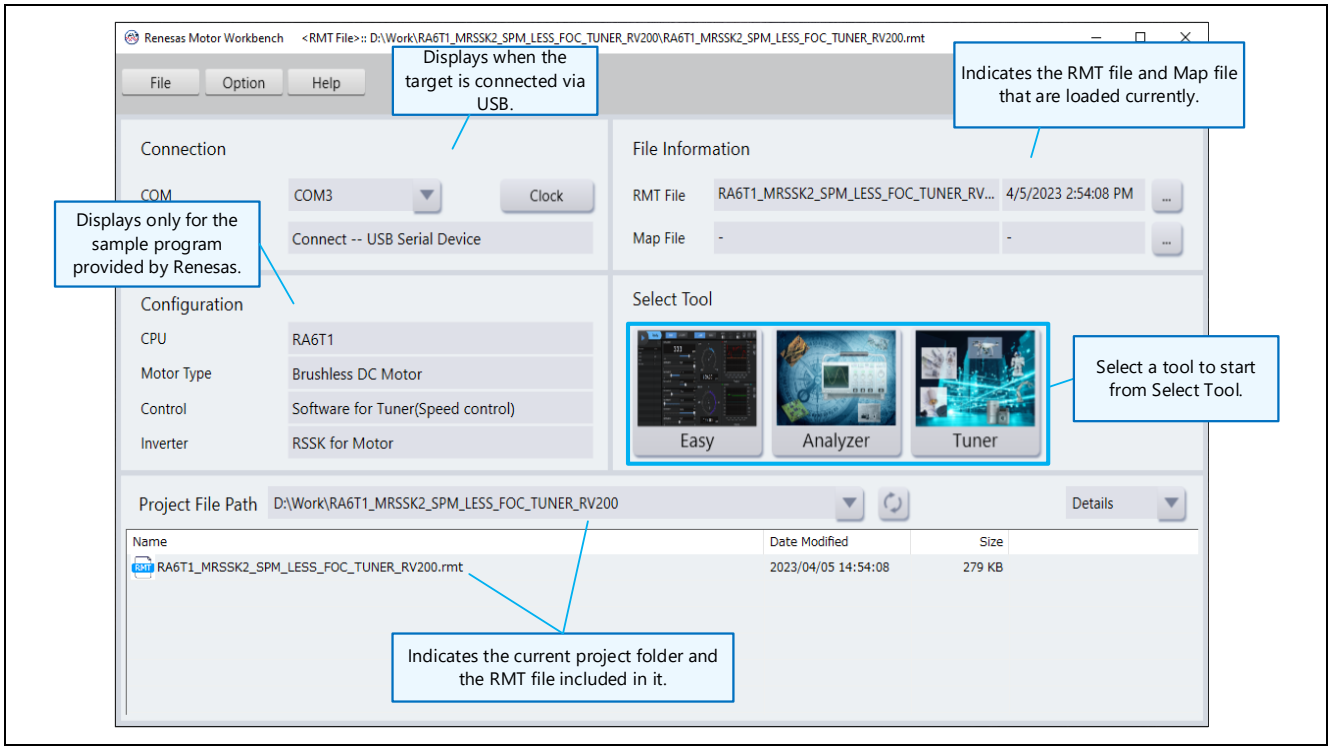


Figure 2-28 Main Window (after the target is connected)

2.8.2 Basic GUI Operation

This section describes the basic GUI operations for Main Window and the internal tools (Easy, Analyzer, Servo, and Tuner) of Renesas Motor Workbench.

For details on each function, refer to the Renesas Motor Workbench User's Manual.

(1) Tool switch button

When you launch a tool from Select Tool on Main Window, "Tool switch button" will be displayed on the top of the screen of each tool. Click it to switch to another tool.

Note that you cannot switch tools during operating (driving a motor).

(2) Main Window switch button

"Main Window button" is shown on the upper right of the screen of each tool. Click it to go back to Main Window.

Note that you cannot switch to Main Window during operating (driving a motor).

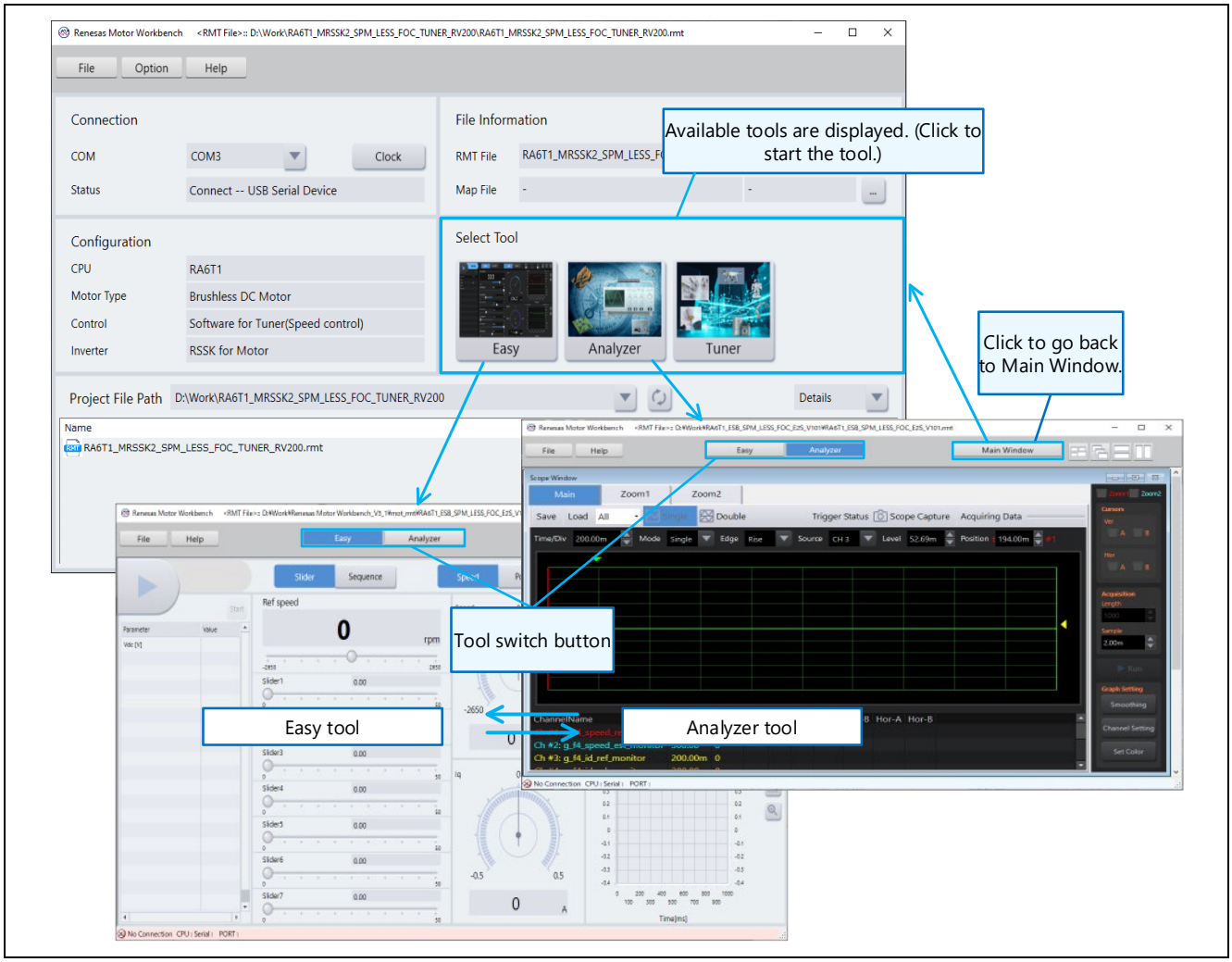


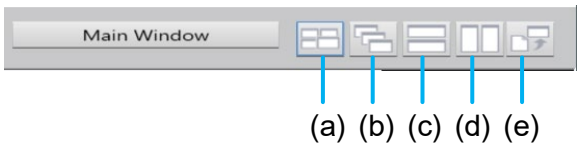
Figure 2-29 Tool switch button and Main Window button

(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 (click to list windows)
(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)＞

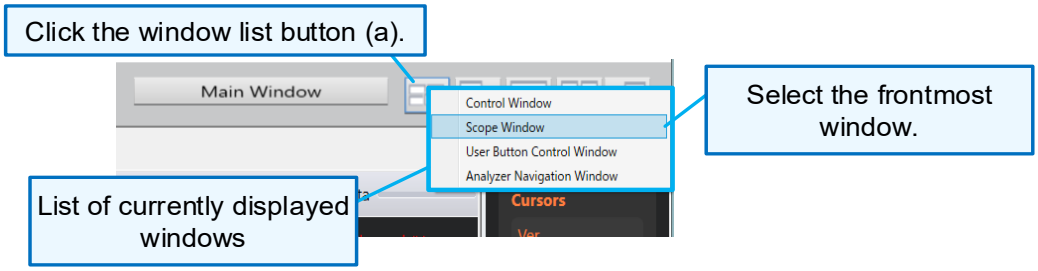


Figure 2-30 Window view switch button in each tool window

(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 return (frame in) the window to the original tool frame by dragging it.

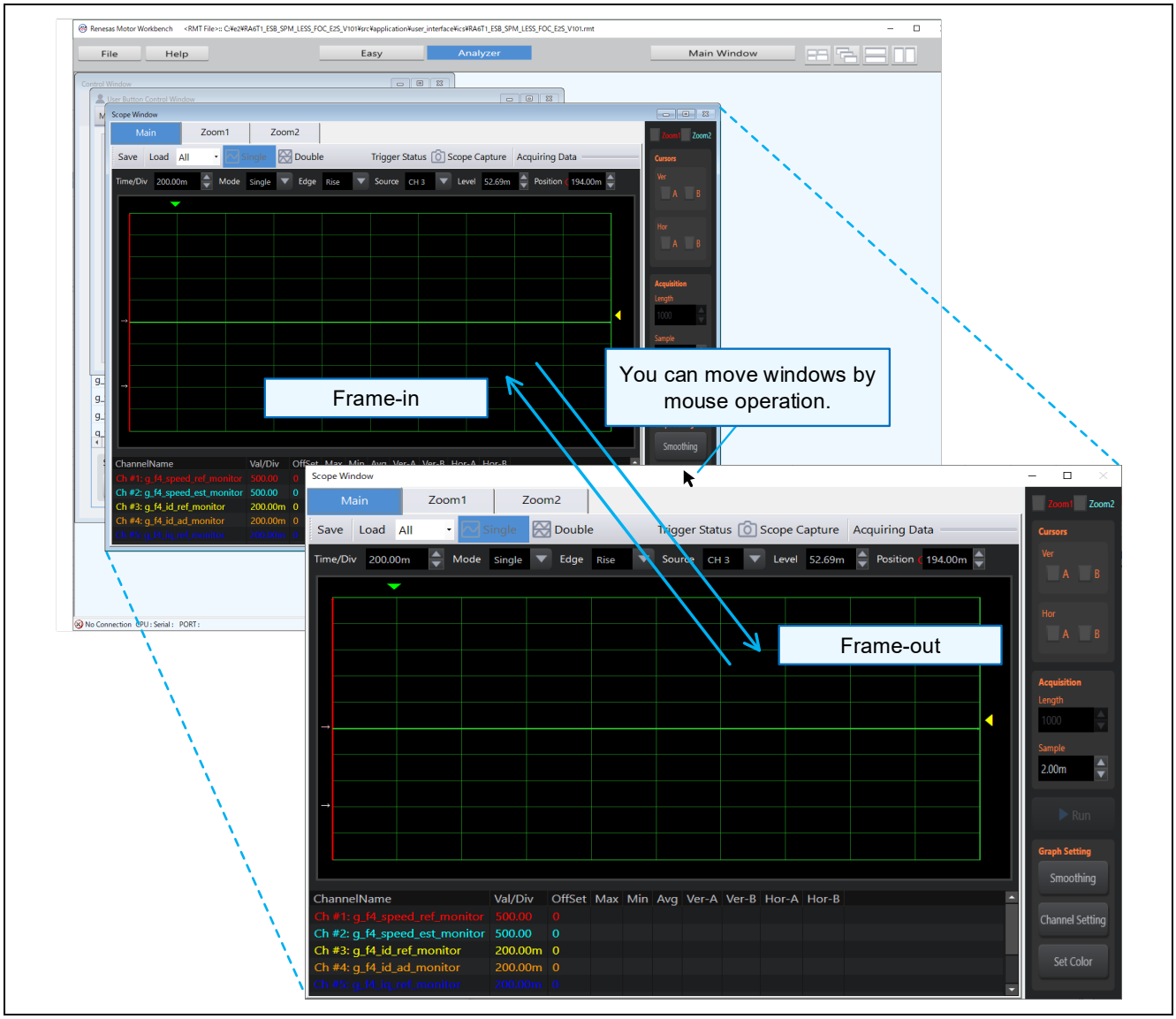


Figure 2-31 Framing out and in window

2.9 Saving RMT file and terminating Renesas Motor Workbench

The RMT file can be saved from “Save” or “Save As” of Main Window’s “File” menu.

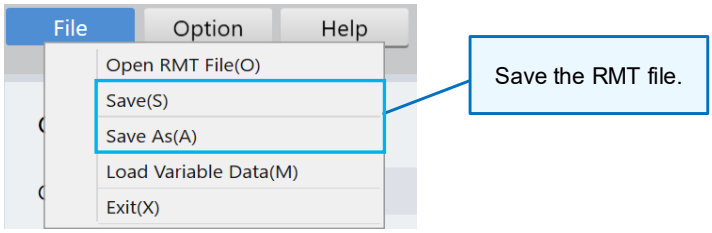


Figure 2-32 Saving RMT file

To terminate Renesas Motor Workbench, select “Exit” from Main Window’s “File” menu or press the close button on the upper right corner of the screen.

On termination, the exit screen is shown as below.

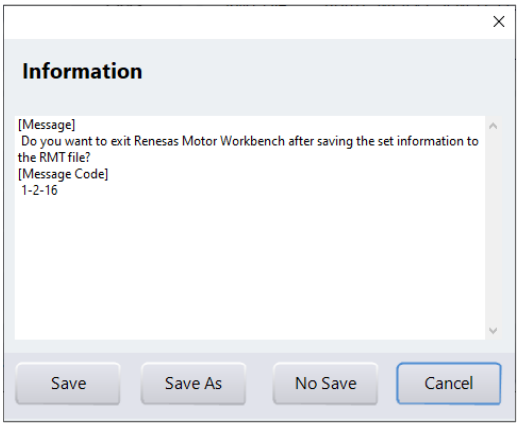


Figure 2-33 Renesas Motor Workbench exit screen

The functions of each button on the exit screen are as follows.

Table 2-3 Functions of buttons on exit screen

Button	Function
Save	Saves the environment information of Renesas Motor Workbench into the loaded RMT file by overwriting it and exits.
Save As	Saves the environment information of Renesas Motor Workbench into a new RMT file with a new name specified and exits.
No Save	Exits without saving the environment information of Renesas Motor Workbench.
Cancel	Cancel the termination process.

3. How to Generate Map File

When Renesas Motor Workbench loads the global variable information (Map file) of the program written into the MCU on the evaluation board, it gains access to the global variables. This Map file can be generated together with the execution file (control software) in the program building.

The followings are setting procedures to generate a Map file in the Integrated Development Environment "CS+ (CC compiler)", "e^2 studio", and "e^2 studio (RA)".

3.1 [RX] In CS+ (CC Compiler) Environment

Setting procedure:

① Launch CS+ and display the target project. Open the project information from "Project Tree".

② Right-click on "CC-RX (Build Tool)" in "Project Tree" and select "Property". The "Property" window will be displayed on the right side of the screen.

③ Click "Link Options" in "Property".

④ Open "List" and "Others".

⑤ See "Outputs the linkage list file" under "List" and if it is set to "Yes (List contents=ALL)(-LISt -SHow=ALL)", skip to ⑨ . If not, proceed to ⑥.

⑥ Select "Yes (List contents=specify) (-LISt)" to "Outputs the linkage list file" under "List".

⑦ Select "No" to "Outputs a symbol name list in a module" under "List".

⑧ Add "-Show=Symbol,struct" (comma-delimited) to "Other additional options" under "Others".

⑨ Build with the above settings, and a Map file will be generated including "variable information".

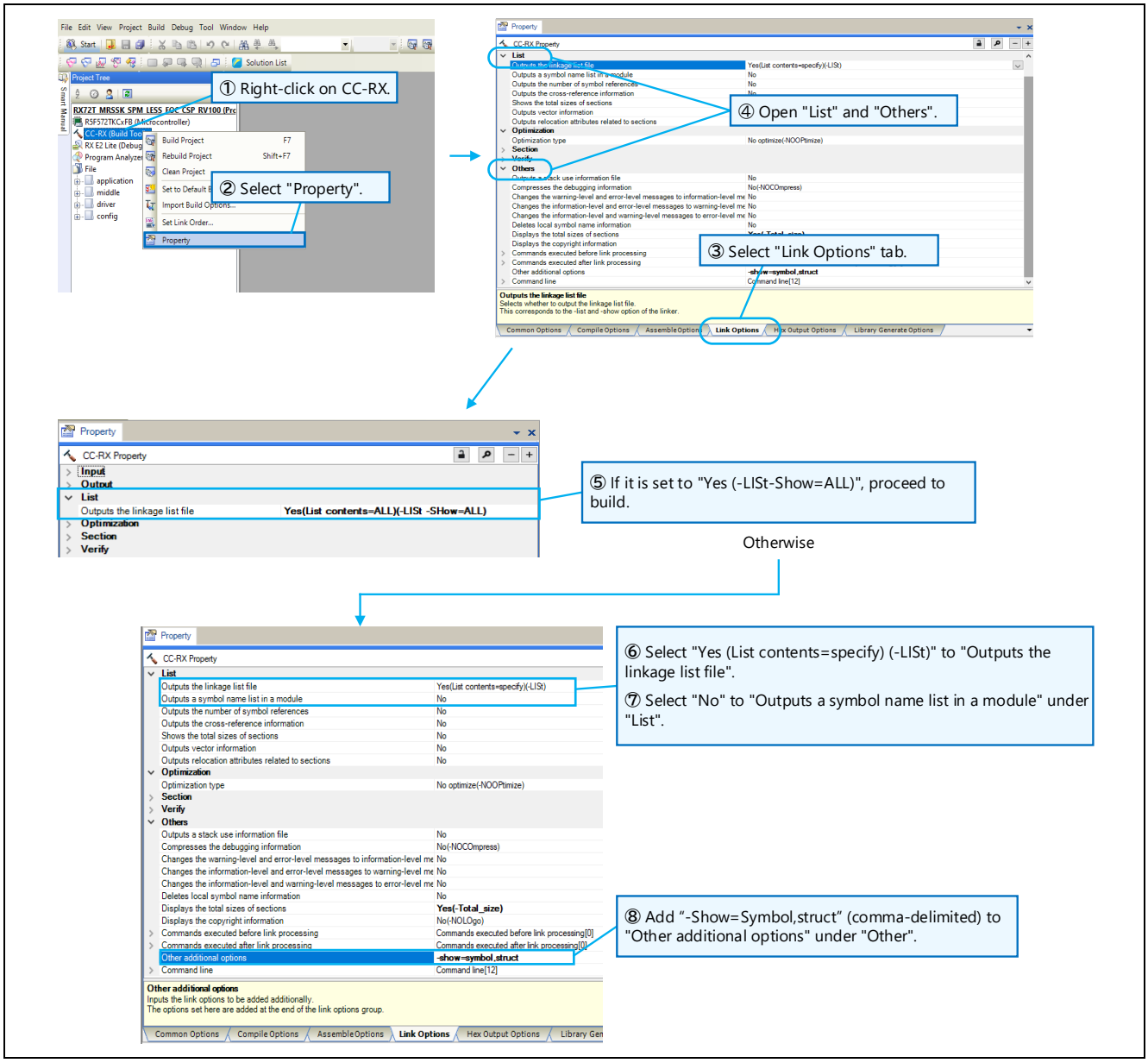


Figure 3-1 CS+ (CC Compiler) property setting

Tips : Copy a Map file to the location where a RMT file is saved after the build.

When you load a Map file after loading a RMT file, the dialog box is opened with the directory path of the loaded RMT file specified. This is because the directory is opened that has been the most recently operated in the file selection dialog box. Therefore, if you have copied a Map file to a RMT file directory in advance, you can select the RMT/Map file without changing the directory path in the dialog box.

In [Link Options] > [Others] > [Command executed after the build], enter the command as below to copy the Map file to the directory where the RMT file is saved, and the Map file will be copied automatically after the build.

```
copy %BuildModeName%\%ProjectName%.map app\rmw\%ProjectName%.map
```

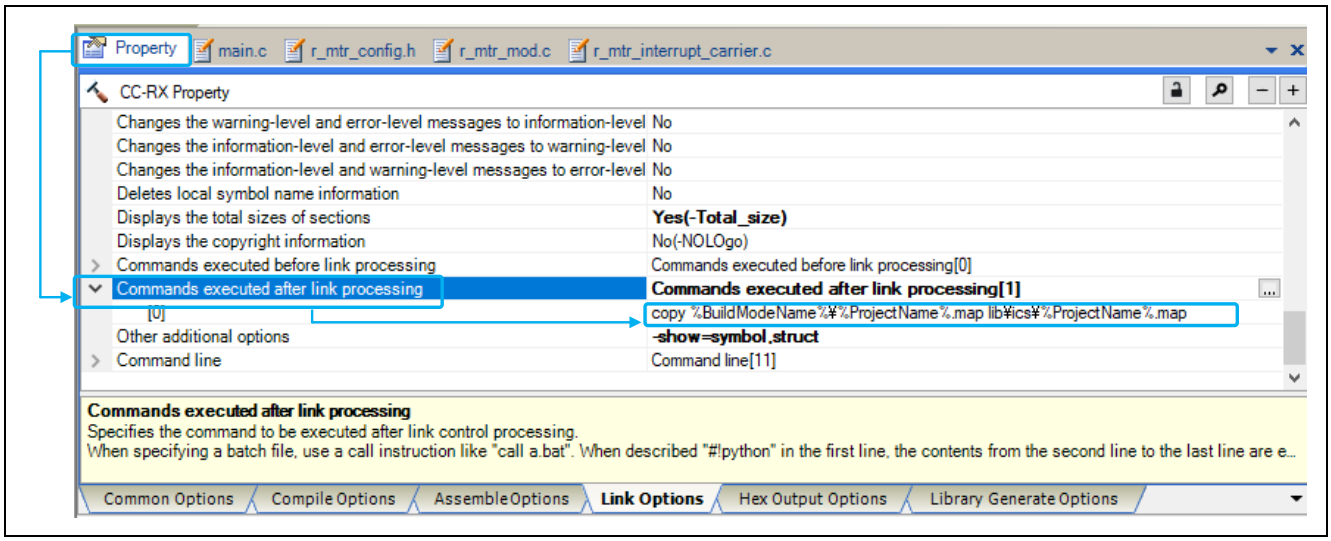


Figure 3-2 Settings for copying Map file in CS+ (CC Compiler)

3.2 [RX] In e² studio Environment

Setting procedure:

① Launch e² studio and display the target project.

② Right-click on the target project from “Project Explorer” and select “Properties”. The “Properties” window will be displayed.

③ Open “C/C++ Build” on the left and select “Settings”.

④ Select “Tool Settings” tab on the right from “Settings”.

⑤ Select “User” under “Linker”.

⑥ In “User-defined options (add after command line)” on the right bottom, press “+” button and add “-Show=Symbol,struct” (comma-delimited)

⑦ Build with the above settings, and a Map file will be generated including “variable information”.

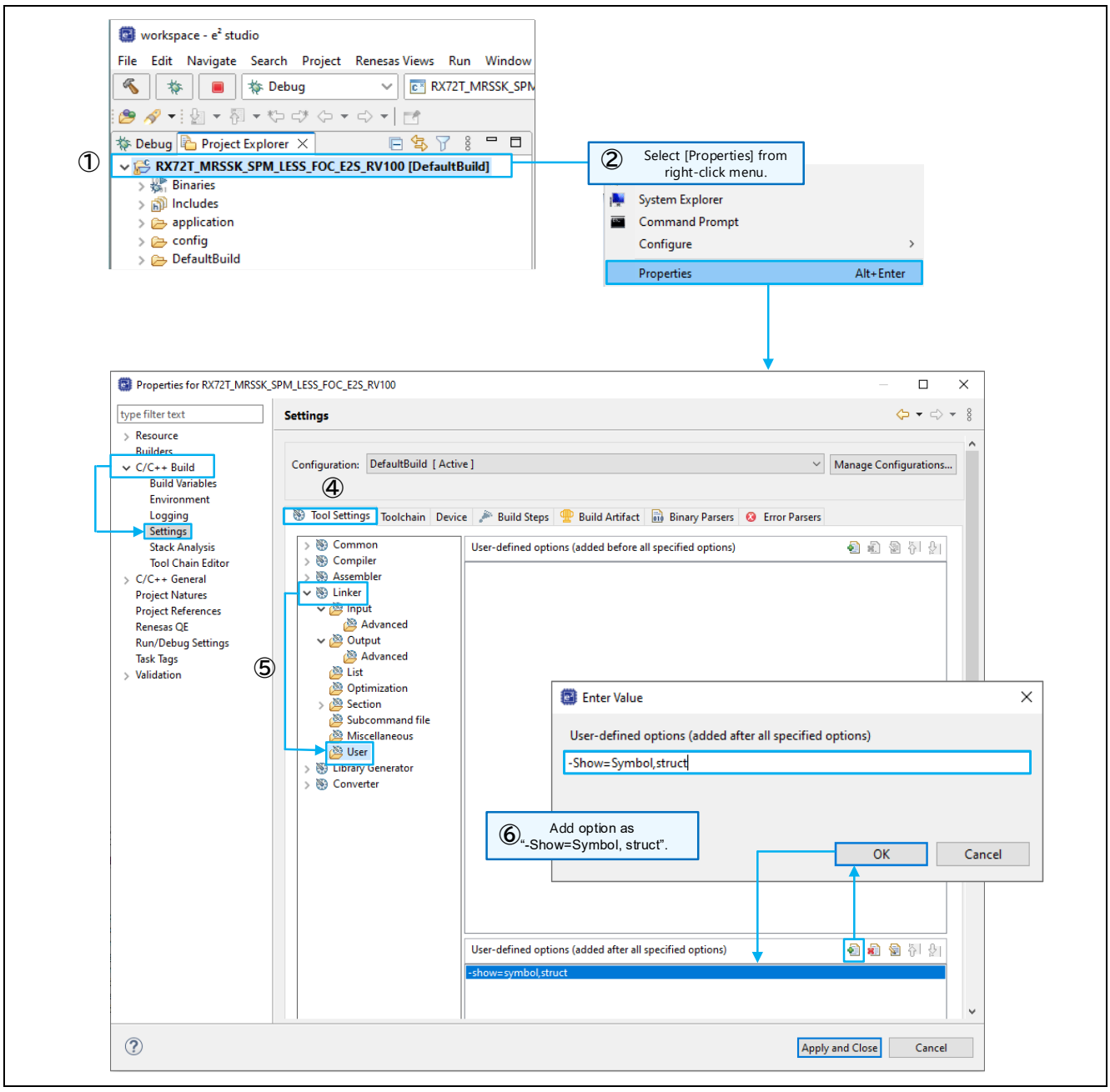


Figure 3-3 e² studio property setting

Tips: Copy a Map file to the location where a RMT file is saved.

When you load a Map file after loading a RMT file, the dialog box is opened with the directory path of the loaded RMT file specified. This is because the directory is opened that has been the most recently operated in the file selection dialog box. Therefore, if you have copied a Map file to a RMT file directory in advance, you can select the RMT/Map file without changing the directory path in the dialog box.

In [C/C++Build] > [Settings] > [Build Step] tab > [Post-build steps] , enter the command as below to copy the Map file to the directory where the RMT file is saved, and the Map file will be copied automatically after the build.

```
copy ${ProjName}.map ..\app\rmw\${ProjName}.map
```

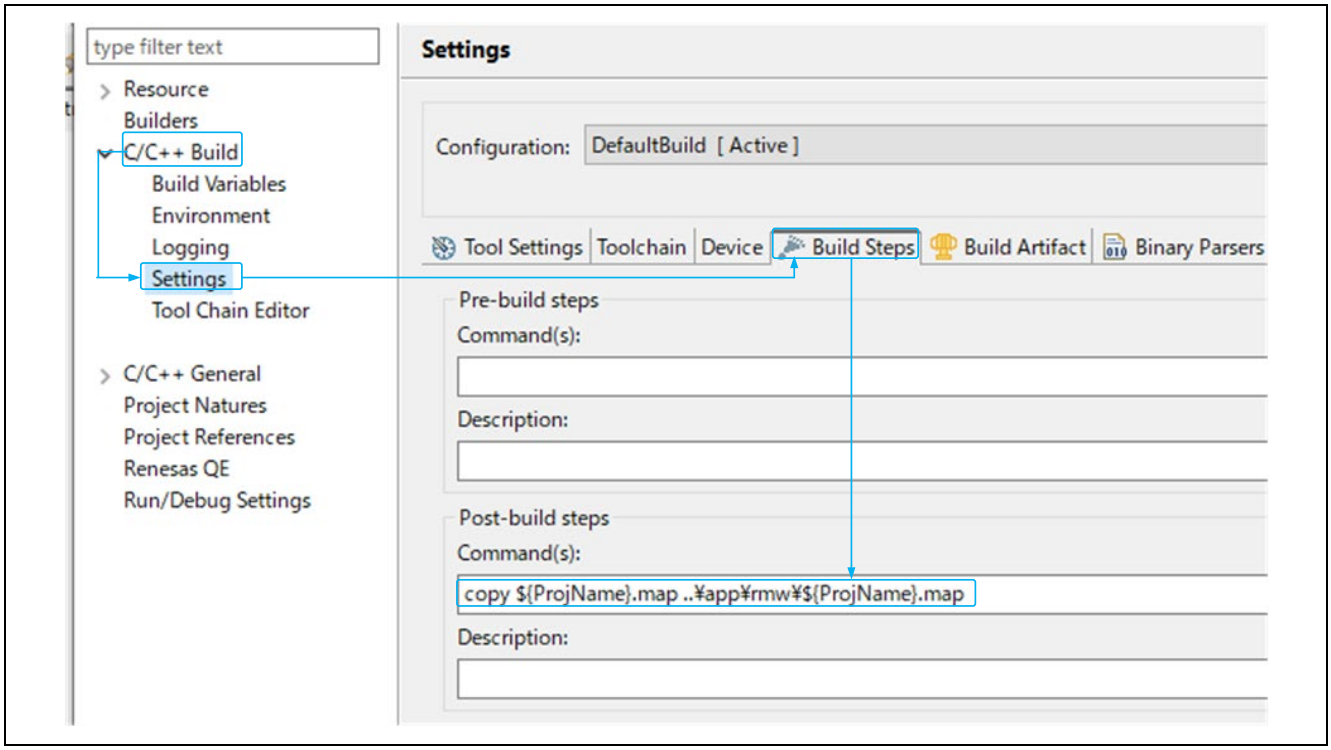


Figure 3-4 Settings for copying Map file in e2studio

3.3 [RA] In e² studio Environment

Setting procedure:

① Copy “ElfMapConverter.exe” attached in Renesas Motor Workbench to the output folder of the project.

② Create a bat file that describes the following command. (The file can be named as any name.)

```
arm-none-eabi-objdump.exe -W project-name.elf>project-name.txt
ElfMapConverter.exe project-name.txt
```

③ Launch “e² studio” and display the project.

④ Right-click on the target project from “Project Explorer” and select “Properties”. The “Properties” window will be displayed.

⑤ Open “C/C++ Build” on the left and click “Settings”.

⑥ Select “Build Steps” tab on the right in “Settings”.

⑦ Add the bat file created in ② to the command field of "Post-build steps". (The file is named “map.bat” for example here.)

⑧ Build with the above settings, and a Map file (.rmap) will be generated including the “variable information”.

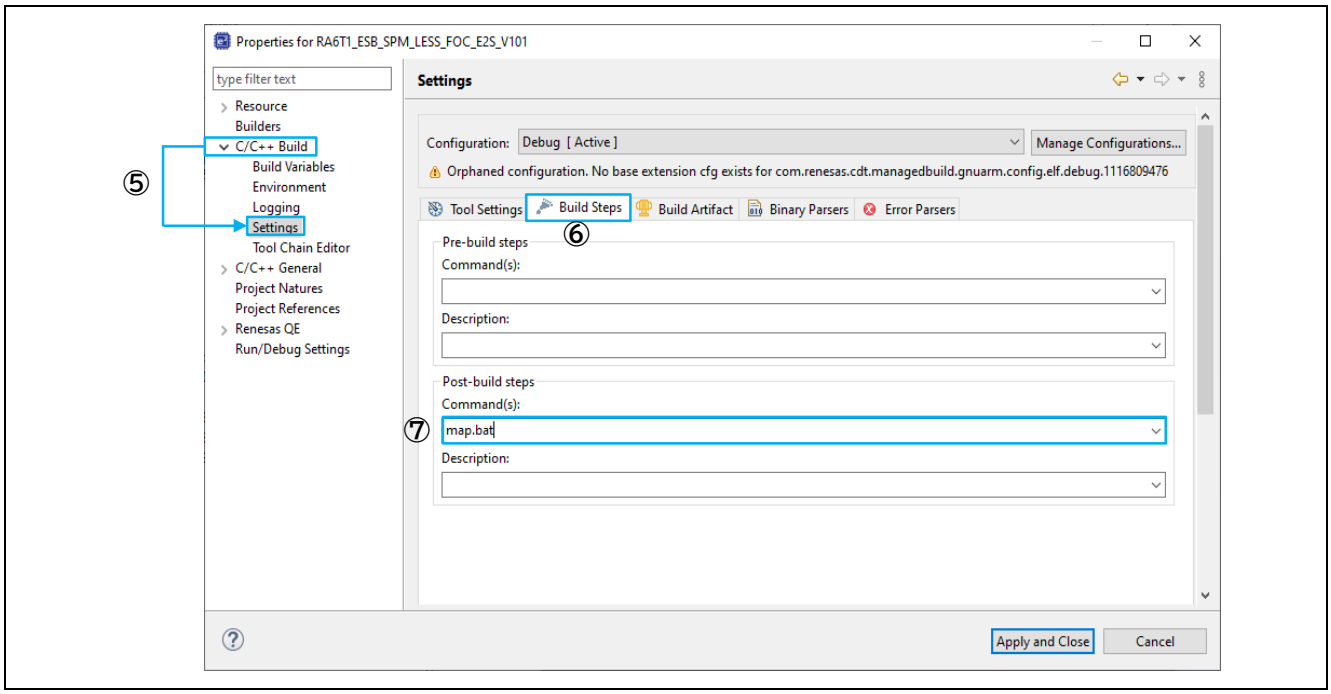


Figure 3-5 e² studio (RA) property setting

Tips: Copy a Map file to the location where a RMT file is saved.

When you load a Map file after loading a RMT file, the dialog box is opened with the directory path of the loaded RMT file specified. This is because the directory is opened that has been the most recently operated in the file selection dialog box. Therefore, if you have copied a Map file to a RMT file directory in advance, you can select the RMT/Map file without changing the directory path in the dialog box.

In addition, if you have modified the extension of the Map file generated in this section from *.rmap to *.map and copy the file, you do not need to change the file type in the dialog box for loading Map files.

If you add the command as below in the bat file described in the above step ②, the Map file whose extension has been modified will be copied automatically after the build. In the following example, “_conv” is added to the copied Map file name so that you can identify the extension changed from *.rmap to *.map.

```
copy ProjectName.rmap <RMT file save directory>\ProjectName_conv.map
```

4. Communication Library

4.1 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 4-1 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 4-2 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 4-3 Communication library (for RX) 3/3

Supported CPU	**RX13T**	**RX26T**
File	ICS2_RX13T.lib ICS2_RX13T.h	ICS2_RX26T.lib ICS2_RX26T.h
Communication rate	0.5 Mbps to 4 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: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 4-4 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 4-5 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 4-6 Communication library (for RA) 1/2

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 (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

Table 4-7 Communication library (for RA) 2/2

Supported CPU	RA6T3	RA4T1	RA8T1
File	ICS2_RA6T3.o ICS2_RA6T3.h	ICS2_RA4T1.o ICS2_RA4T1.h	ICS2_RA8T1.o ICS2_RA8T1.h
Communication rate	0.5 Mbps ～ 16.67 Mbps	0.5 Mbps ～ 16.67 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	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
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

4.2 Built-in Type Communication Library

The RA6T2, RA6T3 ,RA4T1 and RA8T1 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” and “RA8T1”folder in “communication library”).

Table 4-8 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 4-9 Built-in type communication library (for RA) 2/2

Supported CPU	RA8T1
File	ICS2_RA8T1_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
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

4.3 How to Set User Program

4.3.1 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 4.3.4 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.

4.3.2 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 4-1 Code sample for communication interruption functions

4.3.3 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 4-10 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] — 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: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 4-11 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 4-12 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].	

4.3.4 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 4-1)
The third argument: specify the tool interrupt level
The forth 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 4-2 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 4-3 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 4-2)
The third argument: specify the tool interrupt level
The forth 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 4-4 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 4-5 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 4-3)
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 4-6 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 4-7 Calling data transfer function ics2_watchpoint() (RA6T1 sample code)

Revision History

Rev.	Date	Revision	
		Page	Contents
1.00	May.30.23	-	Issued as Quick Start Guide
1.01	Oct.27.23	-	Fixed 2.5.3 Acquiring New Authentication File Changed the URL for obtaining the authentication file to the RMW website.
1.02	Jan.30.24	-	Supported RA8T1
1.10	Feb.07.25	-	Fixed 1.1 Added Servo feature Fixed 1.2.2 System Requirement: Revised disk capacity, OS Fixed 1.3: Updated RL78 MCU table Fixed 2.2: Updated RL78 communication library Fixed 2.7.3: Added maximum communication rate table Fixed 4.1: Updated communication library for each MCU Fixed 4.3.3: Updated specification of library functions
1.11	Jun.30.25	6	1.4 Updated Supportive Information

Ref: Revision history of old version of User's Manual (R21UZ0004)

Rev.	Date	Revision	
		Page	Contents
1.00	Apr.05.17	-	Released version 1.0
2.00	Nov.27.18	-	Released 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	-	Released version 3.0
-	May.30.23	-	Changed to Quick Start Guide

Remarks

- This document is issued as "Quick Start Guide" by re-editing "Motor Control Development Support Tool Renesas Motor Workbench User's Manual (R21UZ0004)".

- In addition, these two existing Function Manuals (Application Notes)
  — Renesas Motor Workbench Functional Description (R20AN0527)
  — Renesas Motor Workbench Tuner Functional Description (R20AN0528)
  are re-edited into:
  — Renesas Motor Workbench User's Manual (R21UZ0004)

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/.