

RA/RX/RL78 ファミリ

モータ制御開発支援ツール Renesas Motor Workbench3.3.0 クイックスタートガイド

はじめに

Renesas Motor Workbench は、ルネサスのモータ制御用 MCU（対象 MCU は「1.3 仕様」を参照）を使用した、モータ制御プログラムのデバッグやチューニングを行うための開発支援ツールです。内部のツールとして、Analyzer、Easy、Tuner の各機能を提供します。

本クイックスタートガイドは、Renesas Motor Workbench のインストールから実行までの手順を説明したものです。Renesas Motor Workbench の各機能については、以下のマニュアルを参照してください。

- Renesas Motor Workbench ユーザーズマニュアル（R21UZ0004）

目次

1. 概要	2
1.1 Renesas Motor Workbench とは	2
1.2 システム要件	3
1.3 仕様	5
1.4 サポート情報（参考）	6
2. インストールと実行	7
2.1 ダウンロード	8
2.2 パッケージ内容	8
2.3 インストール	9
2.4 ソフトウェアの起動	10
2.5 実行プログラムの書き込み	12
2.6 変数情報の読み込み	18
2.7 ホスト PC と評価ボードの接続	23
2.8 操作説明	29
2.9 RMT ファイルの保存と Renesas Motor Workbench の終了	34
3. Map ファイルの生成方法	35
3.1 [RX] CS+（CC コンパイラ）環境の場合	35
3.2 [RX] e ² studio 環境の場合	37
3.3 [RA] e ² studio 環境の場合	40
4. 通信ライブラリ	42
4.1 MCU 別 通信ライブラリ	42
4.2 ビルトイン型通信ライブラリ	49
4.3 ユーザプログラムの設定方法	51
改訂記録	57

1. 概要

1.1 Renesas Motor Workbench とは

Renesas Motor Workbench は、USB で PC と接続したモータ制御評価ボード上で実行されるプログラムのグローバル変数に対して、リアルタイムに値の読み出しや書き込みができるツールです。

表 1-1 Renesas Motor Workbench の主な機能

機能	概要
Analyzer	<ul style="list-style-type: none"> Main Window の Select Tool から「Analyzer」アイコンをクリックすると起動します。 USB 接続したボード上で、実行するプログラムのグローバル変数に対して「変数値の読み込み」「変数値の書き込み」「変数値をリアルタイム波形表示」などができます。
Easy GUI	<ul style="list-style-type: none"> Main Window の Select Tool から「Easy」アイコンをクリックすると起動します。 USB 接続したボードで動作するプログラムに対して、駆動や指令値の指示を GUI で簡単に行うことができます。
Servo	<ul style="list-style-type: none"> Main Window の Select Tool から「Servo」アイコンをクリックすると起動します。 USB 接続したボードで動作するプログラムに対して、サーボ動作の指示を GUI で簡単に行うことができます。
Tuner	<ul style="list-style-type: none"> Main Window の Select Tool から「Tuner」アイコンをクリックすると起動します。 USB 接続したボードで動作するプログラムに対して、モータの「固有パラメータの測定」や「制御に必要なパラメータの自動調整」※ができます。

注 ボード上で実行するプログラムに、自動調整機能が組み込まれている必要があります。

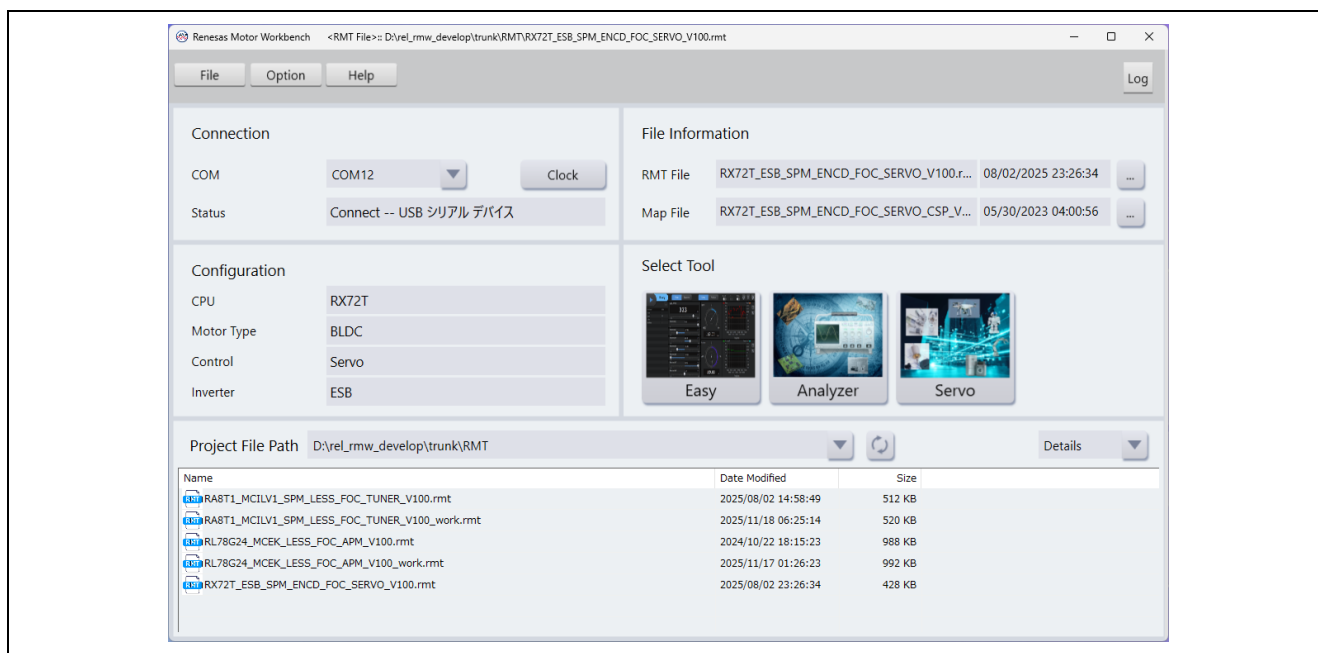


図 1-1 Renesas Motor Workbench の Main Window

Renesas Motor Workbench の利用対象として、以下のモータ制御プログラムを想定しています。

- 本パッケージ内にあるサンプルプログラム
- ルネサスが提供するサンプルプログラム（ルネサス WEB から取得できます）
- ユーザプログラム

1.2 システム要件

1.2.1 ターゲットシステム

Renesas Motor Workbench は、PC と USB で接続したモータ制御ボードで使用できます。PC との接続方法により以下のタイプがあります。

- タイプ1：PC との通信機能（絶縁タイプ）を搭載したモータ制御インバータボード
（例）Renesas Solution Starter Kit - Evaluation System for BLDC Motor（Motor RSSK）
- タイプ2：ツール用通信ボード[※]で PC と接続したモータ制御インバータボード
（例）Renesas Flexible Motor Control Kit - MCK-RA6T2（ツール用通信ボードを同梱）
- タイプ3：市販の USB シリアル変換モジュール（絶縁タイプ）で PC と接続したモータ制御インバータボード（この場合は、ビルトイン型通信ライブラリ（4 章を参照）をご使用ください）

【注】 ツール用通信ボード：MC-COM（ルネサスエレクトロニクス製：RTK0EMXC90S00000BJ）または W2002（デスクトップラボ社製）

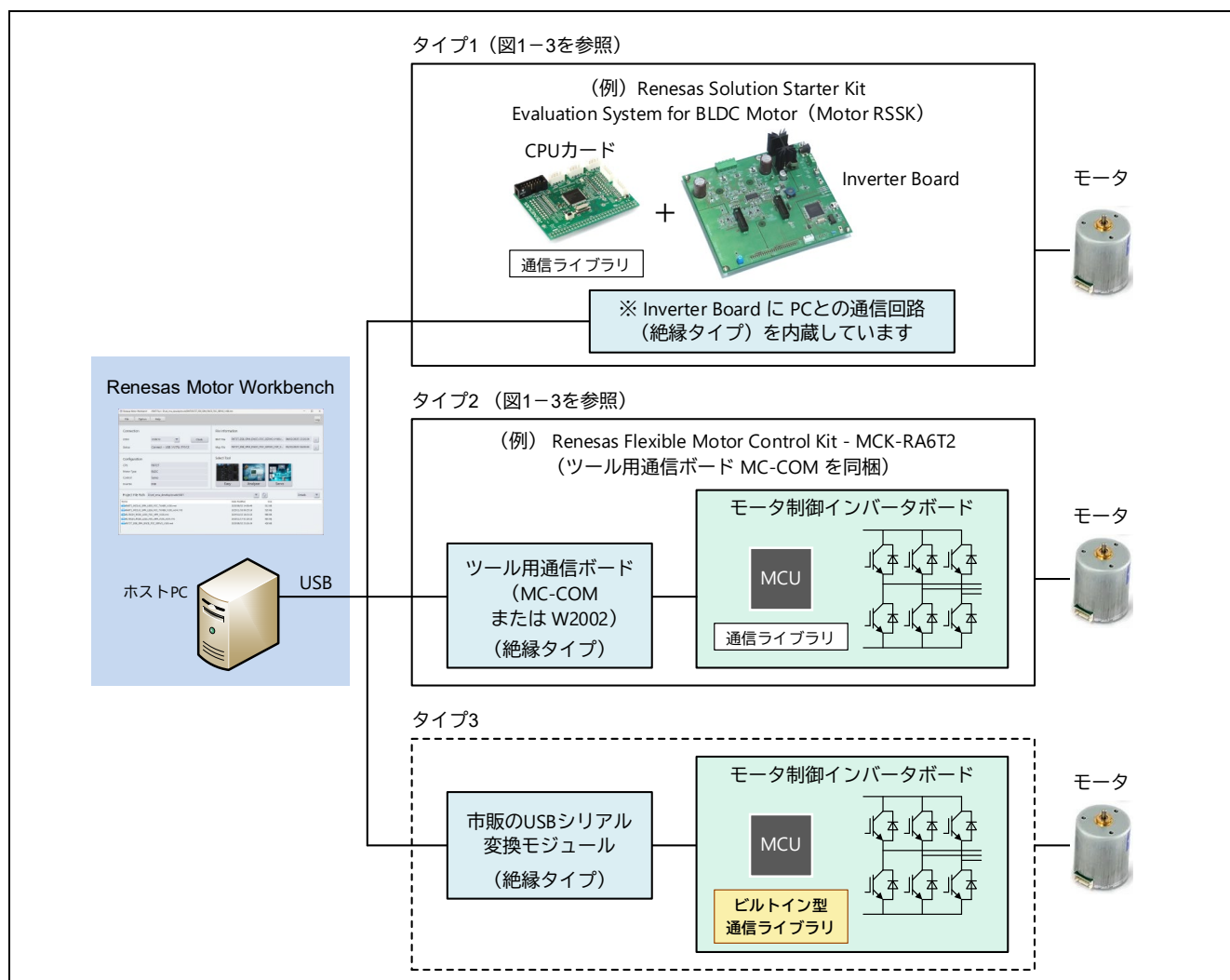
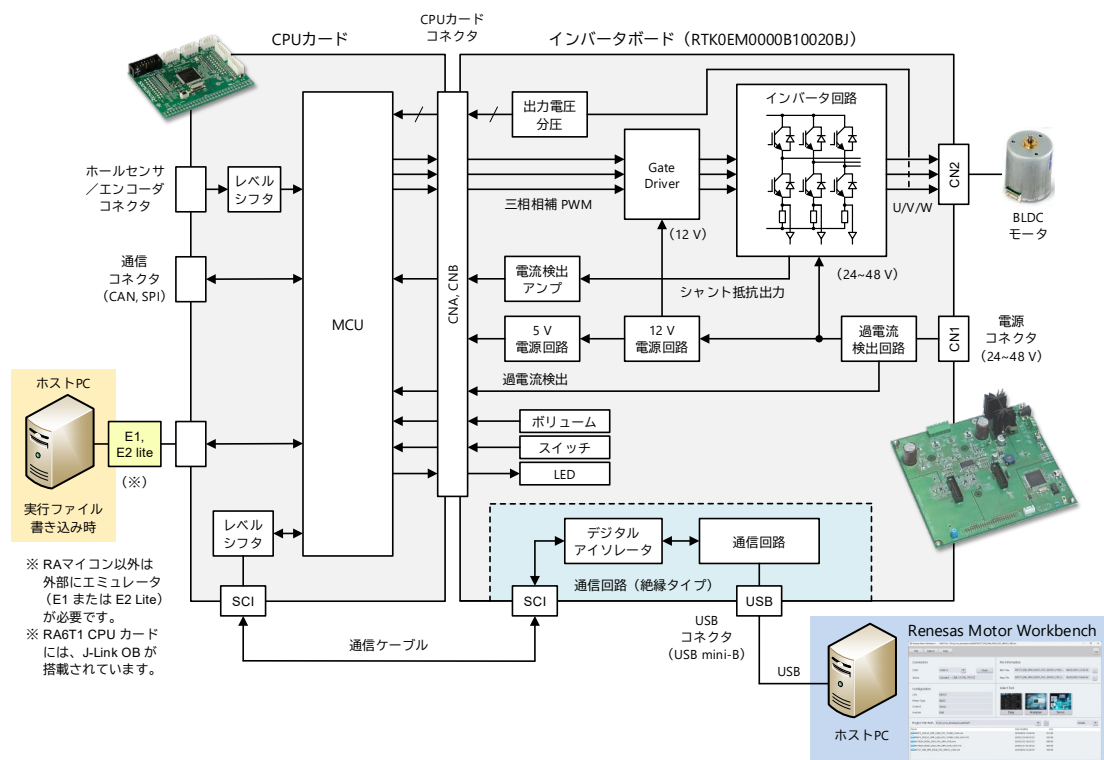
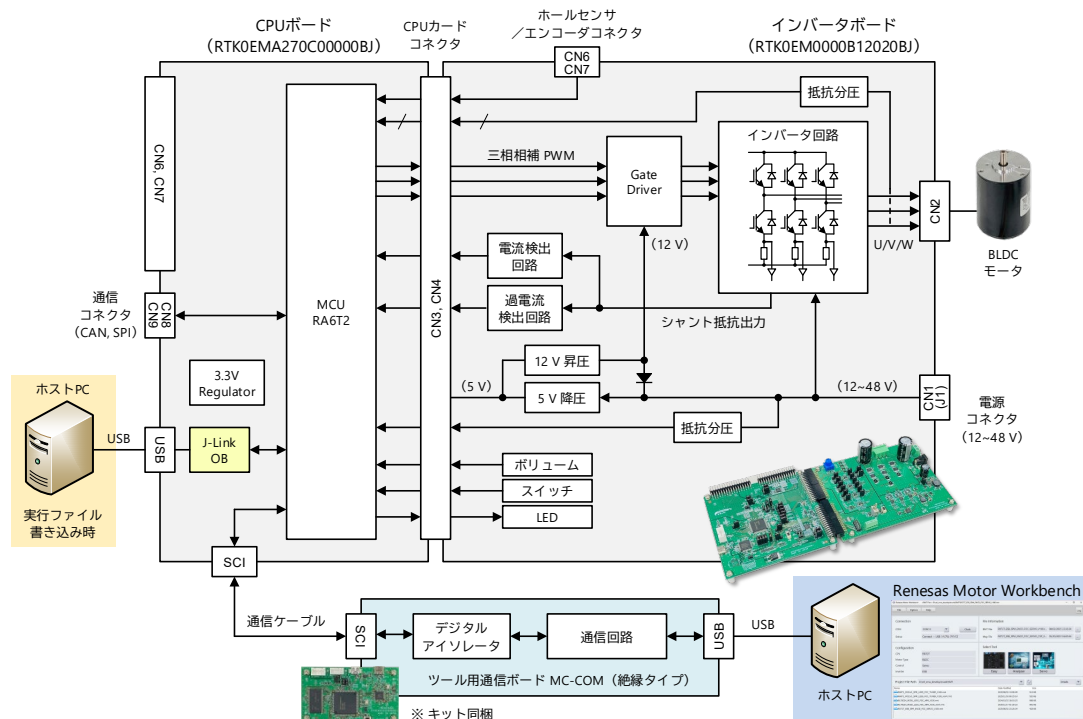


図 1-2 Renesas Motor Workbench 使用時のシステム構成

<タイプ 1 の例> Renesas Solution Starter Kit - Evaluation System for BLDC Motor (Motor RSSK)



<タイプ 2 の例> Renesas Flexible Motor Control Kit - MCK-RA6T2



備考 ハードウェア構成はマイコンごとに異なります。本図はシステム構成を説明するためのもので、一部を簡略化しています。詳細は各評価システムのドキュメントを参照してください。

図 1-3 ルネサスモータ評価システムのハードウェア構成図 (図 1-2 のタイプ 1 とタイプ 2 の例)

1.2.2 ホスト PC のシステム要件

表 1-2 ホスト PC のシステム要件

項目	仕様
PC	USB 端子を装備した Windows パソコン <ul style="list-style-type: none"> • 空きディスク容量：150 MB 以上 • RAM：4GB 以上 • USB ポート：1 ch (USB 2.0)
OS	Windows10, 11
.NET	Microsoft .NET Framework 4.6.1 以降
アプリ	Adobe Reader (PDF ドキュメントと Report 機能を利用する場合に必要) Microsoft Excel (Commander 画面で CsvEdit 機能を利用する場合に必要)

1.3 仕様

表 1-3 Renesas Motor Workbench の仕様

項目		仕様
ターゲットシステム	対応 MCU	RL78/G14、RL78/G1F、RL78/G24 RX13T、RX23T、RX24T、RX24U、RX26T、RX66T、RX72T、RX72M RA4T1、RA6T1、RA6T2、RA6T3、RA8T1、RA8T2、RA2T1
	周辺機能	UART 1 ch、DTC (RA マイコンは除く)、ポート：TXD、RXD、
通信機能	ターゲットシステム	<ul style="list-style-type: none"> • 通信回路を内蔵したモーター制御インバータボード (Motor RSSK など) • ツール用通信ボードで接続されたモーター制御インバータボード (MCK-RA6T2 など) [ツール用通信ボード] <ul style="list-style-type: none"> ・ MC-COM (RTK0EMXC90S00000BJ：ルネサスエレクトロニクス製) ・ W2002 (デスクトップラボ社製[※]) • 市販の USB シリアル変換モジュール (絶縁タイプ) で接続されたモーター制御インバータボード (ビルトイン型通信ライブラリが必要)
	PC	USB2.0
内部機能	変数読み書き	最大同時選択：255 個
	波形表示	<ul style="list-style-type: none"> • 最大同時表示：12ch • 最大データ数： <ul style="list-style-type: none"> — 1024 (Motor RSSK、ビルトイン型通信ライブラリ使用時) — 100000 (ツール用通信ボード MC-COM 使用時)

注 株式会社デスクトップラボ / Desk Top Laboratories Inc (<http://www.desktoplab.co.jp/>)

1.4 サポート情報（参考）

ルネサス WEB からは、モータ制御に関連する情報を豊富に提供しています（以下に例を示します）。
「Renesas Motor Workbench」のページでは、Renesas Motor Workbench の使用方法を動画でご紹介しています。

- モータ制御

<https://www.renesas.com/ja/key-technologies/motor-control>

- Renesas Motor Workbench

<https://www.renesas.com/ja/software-tool/renesas-motor-workbench>



図 1-4 ルネサス WEB「Renesas Motor Workbench」ページ

2. インストールと実行

Renesas Motor Workbench のインストールから実行までの作業の流れを以下に示します。

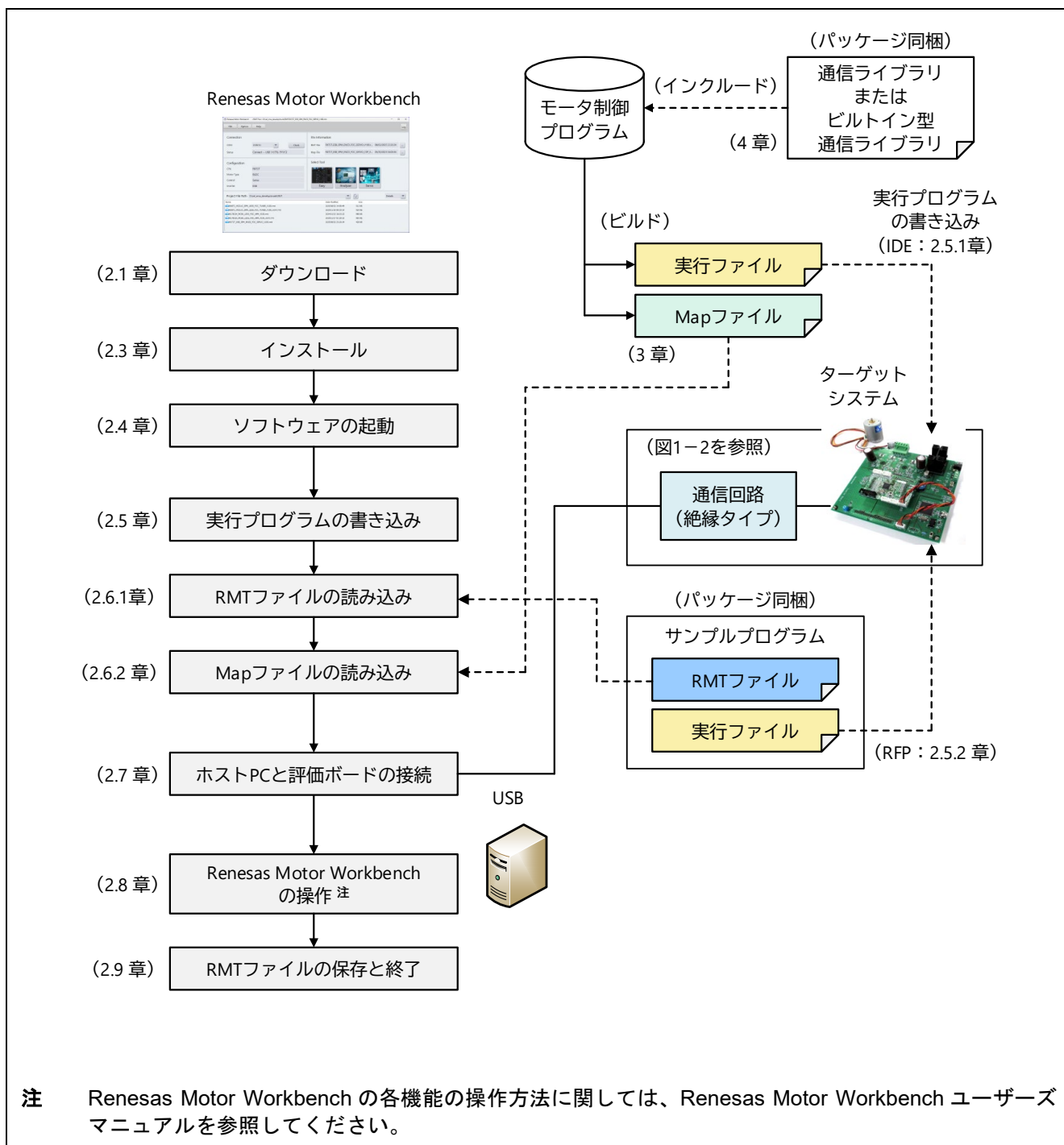


図 2-1 Renesas Motor Workbench のインストールから実行までの流れ

2.1 ダウンロード

Renesas Motor Workbench は、ルネサス WEB よりダウンロードしてください。ファイルは zip 形式で圧縮されていますので、あらかじめ解凍してください。

2.2 パッケージ内容

ダウンロードしたパッケージのフォルダ構成と、各フォルダ内のファイルを以下に示します。

communication library.....	(1) 通信用ライブラリ（マイコン別にフォルダがあります）
ElfMapConverter	(2) Map ファイル変換ツール
Installer	(3) インストーラ
mot_mmt	(4) 実行ファイルと環境ファイル（評価ボード別にフォルダがあります）
r20an0527ej0200-motor.pdf	(5) ドキュメント（同梱するドキュメント名・種類は、変更になる場合があります）
r20an0527jj0200-motor.pdf	
r20an0528ej0200-motor.pdf	
r20an0528jj0200-motor.pdf	
r21uz0004ej0300-motor.pdf	
r21uz0004jj0300-motor.pdf	

図 2-2 パッケージのフォルダ構成（zip 解凍後）

表 2-1 各フォルダ内のファイル

No.	フォルダ	ファイル	内容
(1)	Map ファイル生成ツール	ElfMapConverter.exe	RA マイコンの elf ファイルから map ファイルを生成するツール
(2)	通信用ライブラリ（マイコン別にフォルダがあります）	ICS_RX***.h ICS_RX***.obj	RX マイコン組み込み用ファイル
		ICS2_RX***.h ICS2_RX***.lib	
		ICS2_RL78***.h ICS2_RL78***.lib	RL78 マイコン組み込み用ファイル
		ICS2_RA***.o ICS2_RA***_Built_in.o ICS2_RA***.h	RA マイコン組み込み用ファイル（ICS2_RA***_Built_in.o は、ビルトイン型通信ライブラリ）
(3)	インストーラ	renesas_motor_workbench_****.msi	Renesas Motor Workbench インストーラ
(4)	実行ファイルと RMT ファイル（評価ボード別にフォルダがあります）	****.mot (RX, RL) ****.hex (RA)	Motor RSSK に書き込む実行ファイル
		****.rmt	Renesas Motor Workbench の RMT ファイル（環境設定を保存したファイル）
(5)	ドキュメント	r21uz0004ej****-rmw-um.pdf r21uz0004jj****-rmw-um.pdf	Renesas Motor Workbench ユーザーズマニュアル
		r21qs0011ej****-rmw-qsg.pdf r21qs0011jj****-rmw-qsg.pdf	クイックスタートガイド（本書）

2.3 インストール

Renesas Motor Workbench のインストールは、パッケージ内の「Installer」フォルダにあるインストーラ
「renesas_motor_workbench_***.msi」を実行します。

インストーラが起動したら、画面の指示に従ってインストールを進めてください。

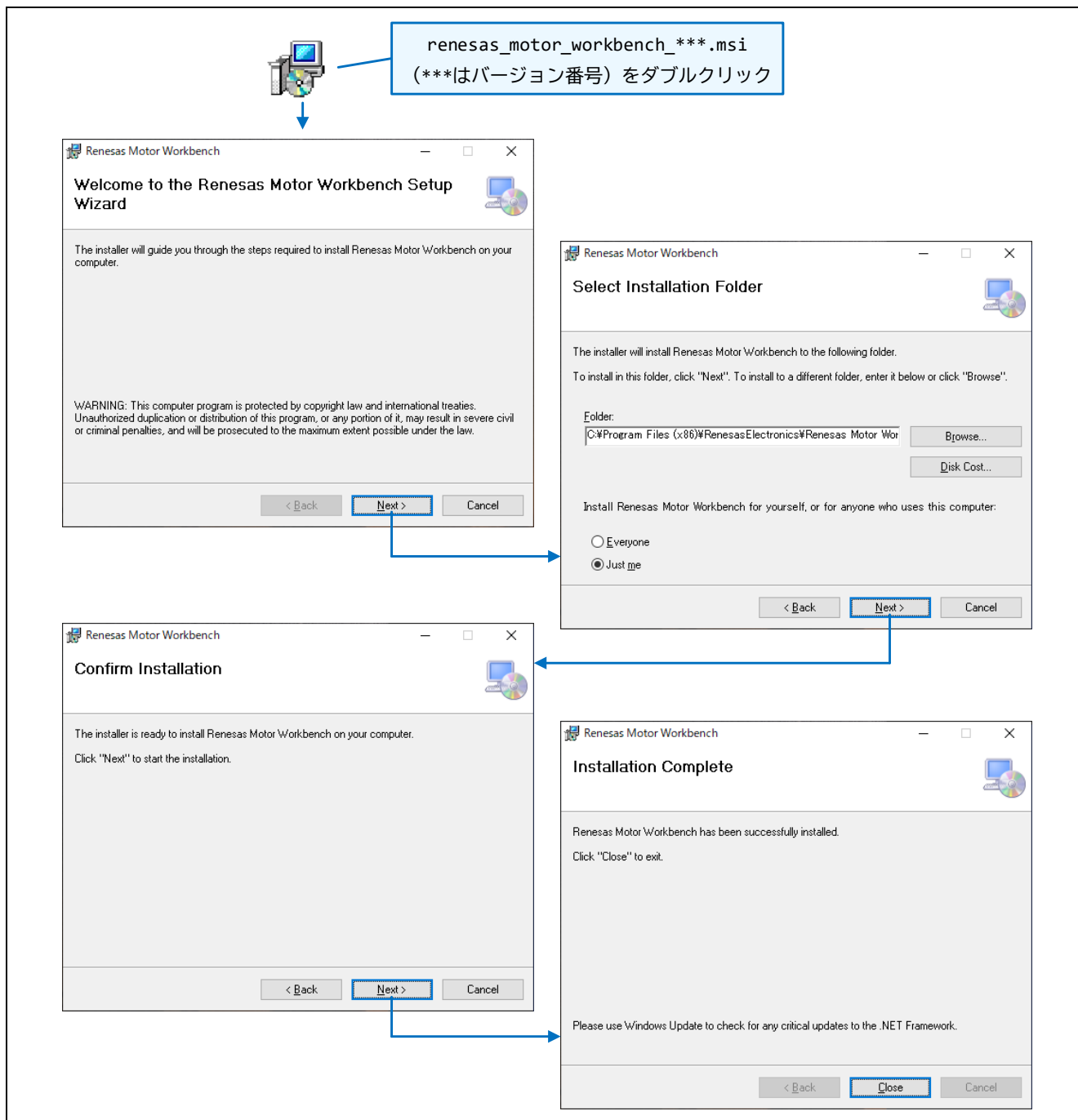


図 2-3 インストール手順

2.4 ソフトウェアの起動

Renesas Motor Workbench がインストールされると、デスクトップに下記ショートカットアイコンが作成されます。ショートカットアイコンをダブルクリックして「Renesas Motor Workbench」を起動します。

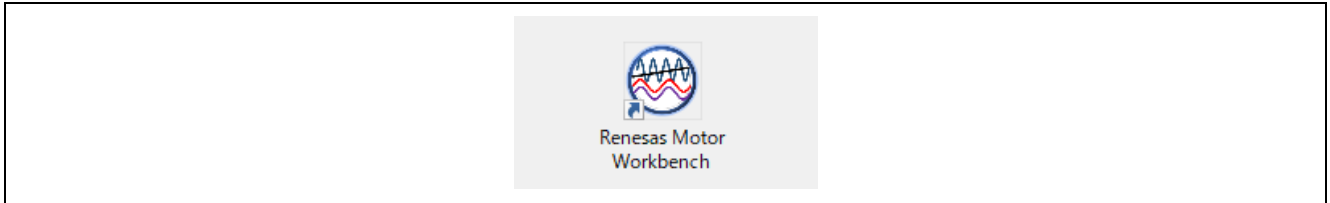


図 2-4 起動用ショートカットアイコン

Windows のスタートメニューから起動する場合は、「Renesas Electronics」 > 「Renesas Motor Workbench」を選択してください。



図 2-5 スタートメニューからの起動

Windows のエクスプローラから RMT ファイルをダブルクリックして Renesas Motor Workbench を起動することもできます。この場合、選択した RMT ファイルが読み込まれた状態で起動します。

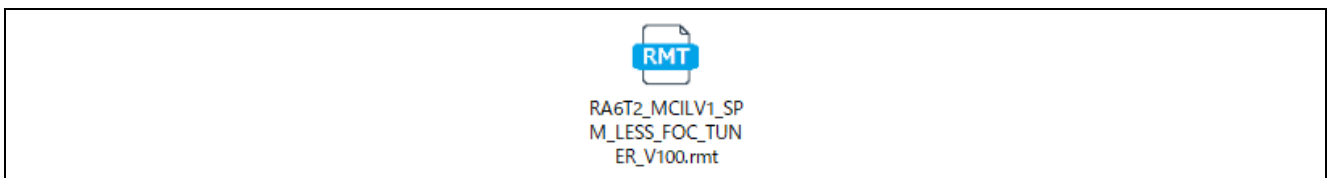


図 2-6 RMT ファイルをダブルクリックして起動

Renesas Motor Workbench が起動しない場合は、インストールに失敗している可能性があります。一度アンインストールしてから、前章の手順で再度インストールしてください。

2.4.1 Main Window の表示確認

Renesas Motor Workbench が起動すると Main Window が開き、同時に初期設定を補助する Select Navigation のダイアログが表示されます（Select Navigation での設定は、後からできるため、ここでは「Finish」ボタンで Select Navigation のダイアログを閉じて構いません）。

備考 Select Navigation ダイアログは、Help メニューからも起動できます。

Renesas Motor Workbench が起動することを確認できたら、次に評価ボードを接続するための準備を行います。

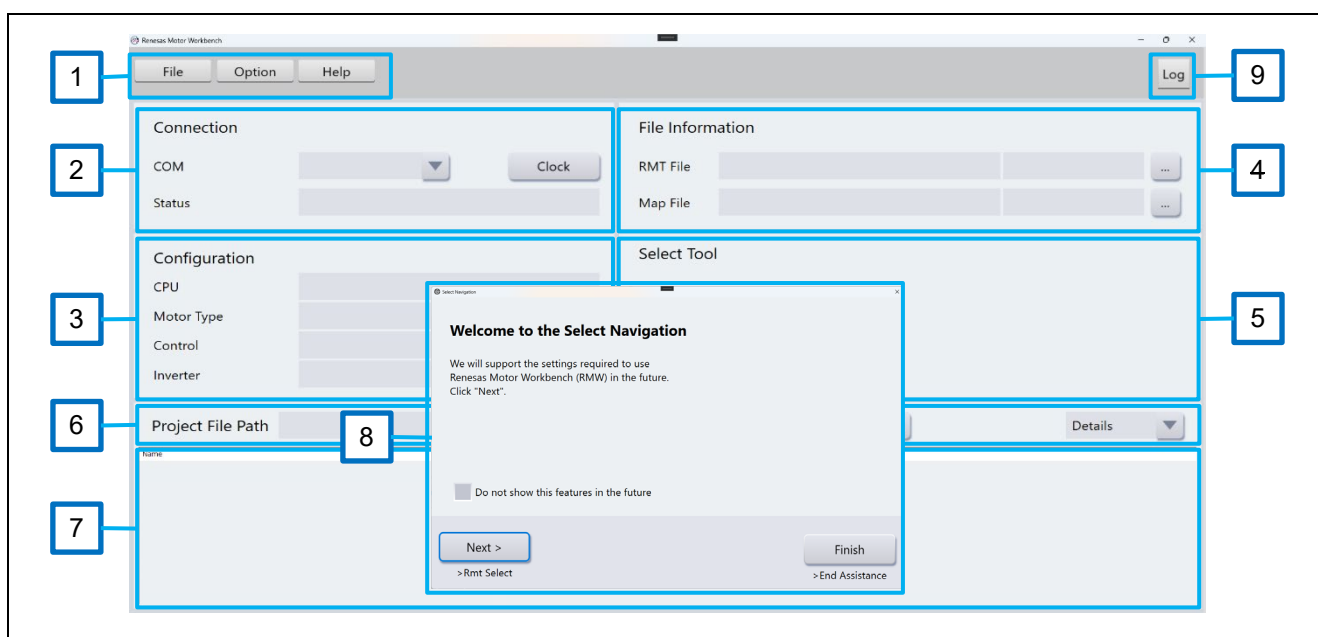


図 2-7 起動直後の Main Window 表示

表 2-2 Main Window 各部の名称（No.は上の図と対応します）

No.	名称	機能概要
1	メニュー	File／Option／Help の各メニュー
2	Connection	通信設定（COM 接続の選択、通信クロック、接続状況の表示）
3	Configuration	接続先の情報（CPU 情報、モータ種別、制御方法、インバータ名）（※ルネサス提供の実行ファイルのみの機能です）
4	File Information	<ul style="list-style-type: none"> 読み込んだ RMT ファイルと Map ファイルの情報表示。 右側のボタンで、RMT ファイルまたは Map ファイルを読み込み可能
5	Select Tool	利用できるツールアイコンの表示。アイコンのクリックでツールを起動
6	Project File Path	<ul style="list-style-type: none"> 過去に読み込んだ RMT ファイルへのパスをプルダウン表示（最大 10 件） 選択したフォルダ内の RMT ファイルを下の File List に表示
7	File List	<ul style="list-style-type: none"> Project File Path で示されたフォルダにある RMT ファイルを表示 ダブルクリックで Renesas Motor Workbench に RMT ファイルを読み込み
8	Select Navigation	Renesas Motor Workbench の各種設定をダイアログ形式で行います
9	Log 表示	操作ログの表示

2.5 実行プログラムの書き込み

モータ制御プログラムのプロジェクトをビルドして生成された実行ファイルや、本パッケージに含まれる実行ファイル（.mot または .hex 形式）をターゲットマイコンに書き込みます。書き込みは IDE（CS+または e2 studio）や「Renesas Flash Programmer（RFP）」などのツールを利用します。

【注意】 PC と CPU ボードの接続方法、および電源の供給方法は、実行プログラムの書き込み時と Renesas Motor Workbench 使用時で異なります。詳細は、各評価システムのマニュアルを参照してください。

2.5.1 IDE でサンプルプログラムをビルドして書き込む場合

ルネサスが WEB から提供するサンプルプログラムを統合開発環境 e2 studio を用いて書き込む方法を説明します（ここでは例として RA マイコンを使用しています）。

ビルドの詳細については、サンプルプログラム付属のアプリケーションノートを参照してください。

(1) プロジェクトのインポート

サンプルプログラムをルネサス WEB より入手して、以下の手順で e² studio にインポートします。

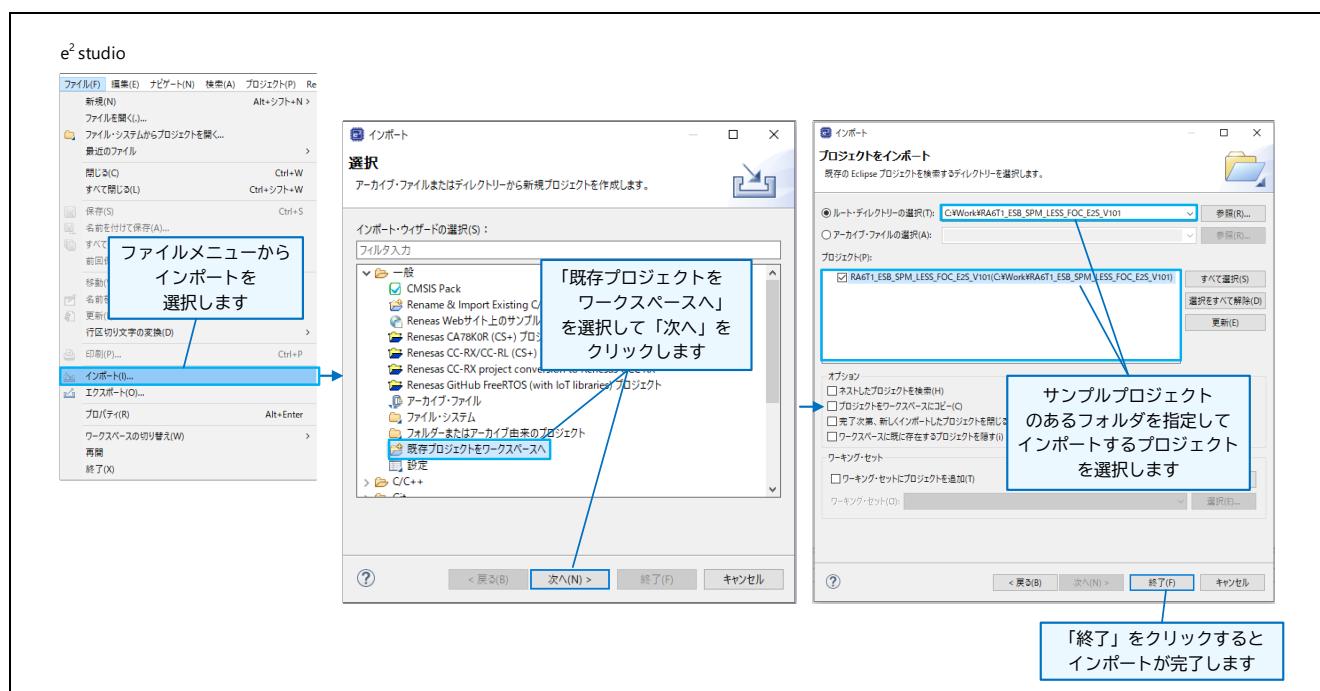


図 2-8 プロジェクトのインポート (e² studio)

(2) ツールチェーンの設定

プロジェクトのプロパティで、環境に合わせてツール構成を設定します。

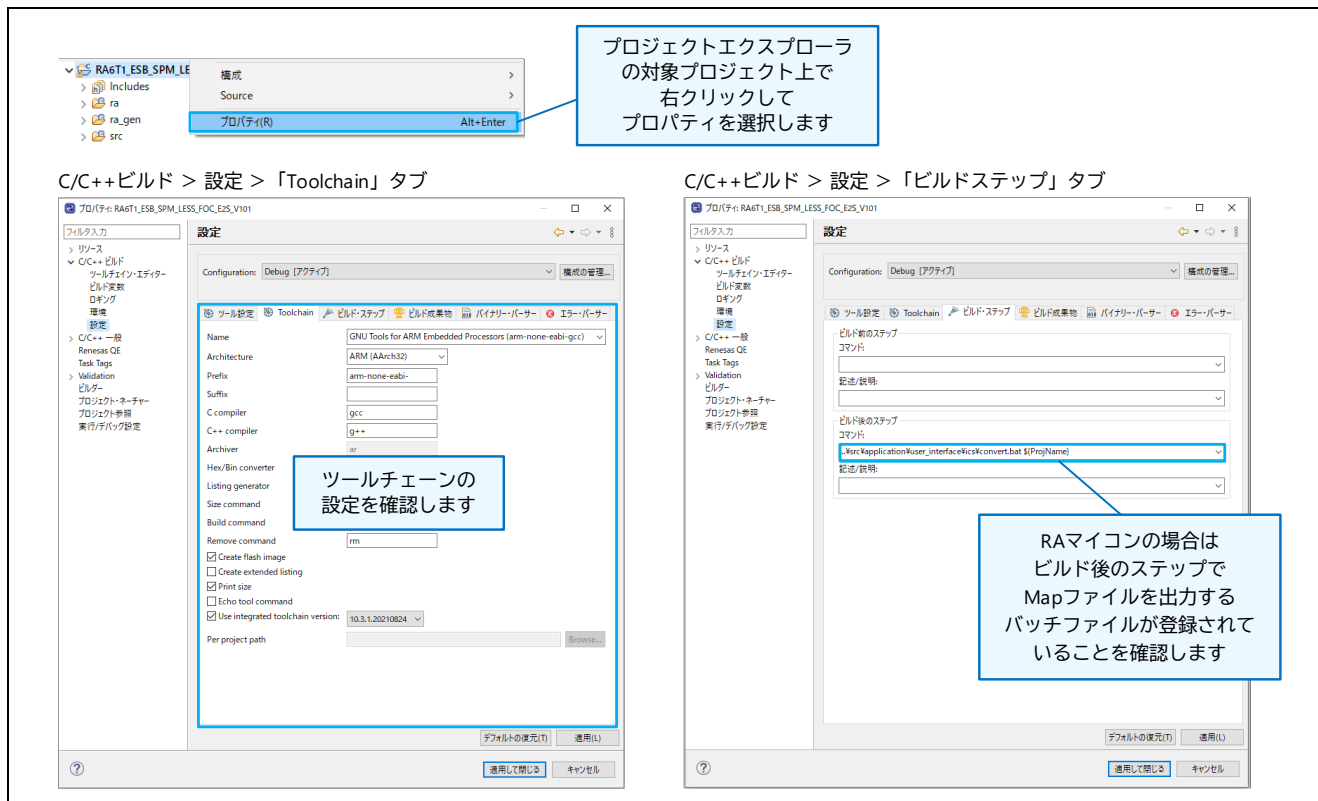


図 2-9 ツールチェーンの設定

(3) ビルド

プロジェクトをビルドします。ビルド結果でエラーが無かったことを確認してください。

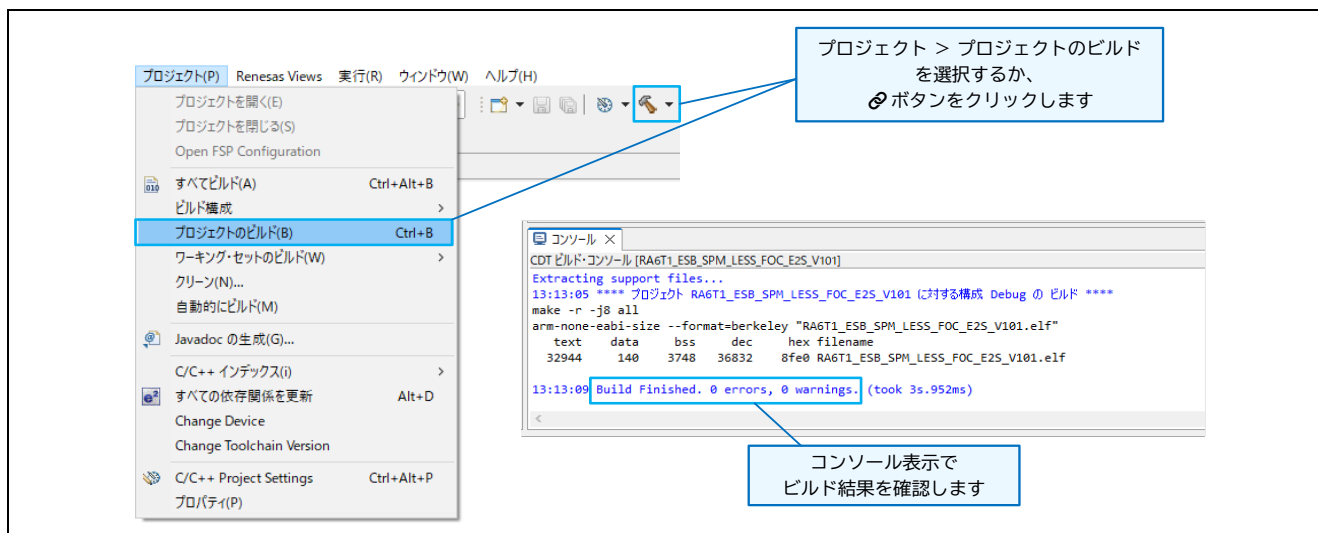



図 2-10 プロジェクトのビルド

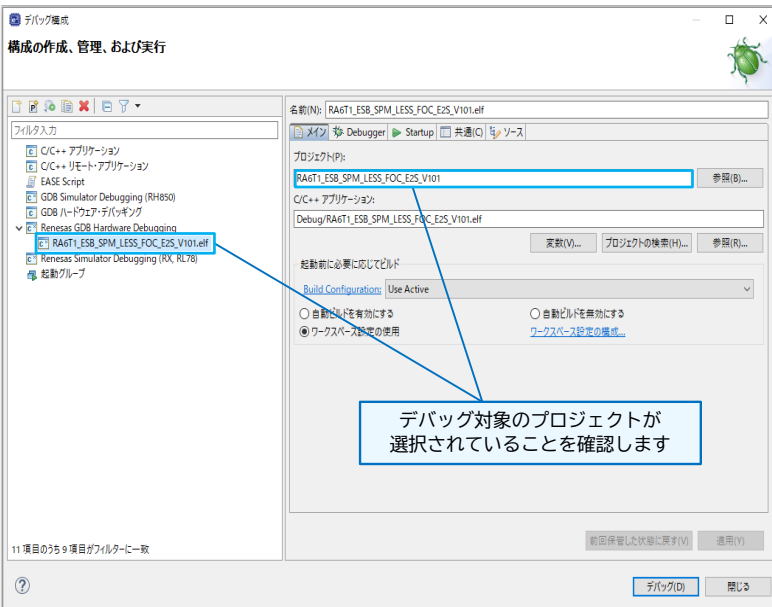
(4) 評価ボードへの書き込み

PC と評価ボードを USB ケーブルで接続して（CPU ボードに J-Link インタフェースが実装されている場合）、プログラムを評価ボードに書き込んでください。



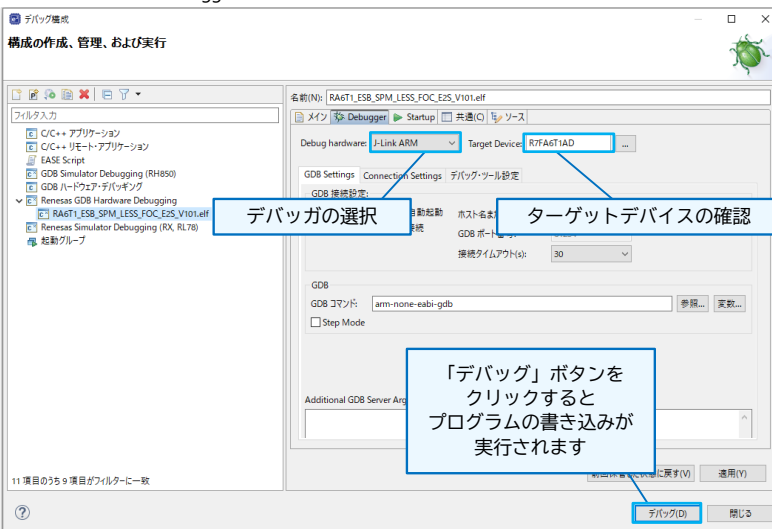
実行 > デバッグの構成を開きます

デバッグ構成 > 「メイン」タブ



デバッグ対象のプロジェクトが選択されていることを確認します

デバッグ構成 > 「Debugger」タブ



デバッガの選択

ターゲットデバイスの確認

「デバッグ」ボタンをクリックするとプログラムの書き込みが実行されます

↓

コンソール画面で「ダウンロード終了」を確認します

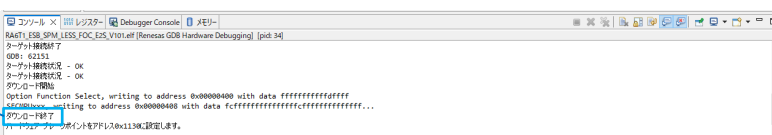


図 2-11 評価ボードへのプログラム書き込み

(5) 評価ボードの取り外し

プログラムが書き込めたら、実行 > 切断で、評価ボードとの接続を切断して USB ケーブルを取り外してください。

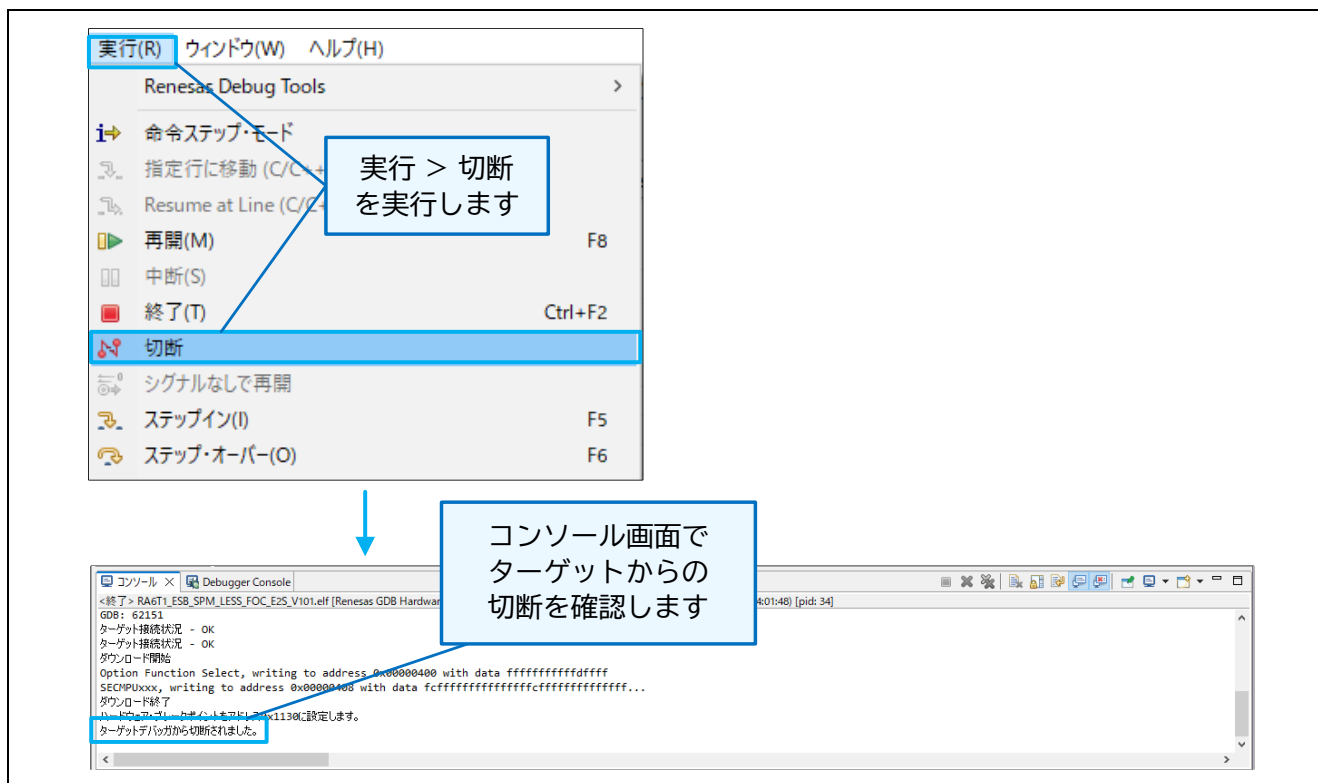


図 2-12 評価ボードの取り外し

なお、再ビルドを行った場合は、最新の実行ファイルと同時に生成した Map ファイルを Renesas Motor Workbench で読み込んでください。Map ファイルの生成方法は 3 章を、Renesas Motor Workbench の Map ファイルの読み込みについては 2.6.2 章を参照してください。

【注意】プログラムの書き込み後、Renesas Motor Workbench を使用してプログラムを実行（モータを駆動）する場合、ターゲットボードとの接続は、専用の通信回路（絶縁タイプ）を使用してください。

2.5.2 Renesas Flash Programmer (RFP) で書き込む場合

本パッケージに同梱されている実行ファイルを Renesas Flash Programmer (RFP) ツールでターゲットボードに書き込む方法を以下に示します (RFP は、ルネサス WEB から入手できます)。CPU ボードとの通信は、使用するシステムに合わせて設定してください。

なお、RMT ファイルも同梱されているものを読み込んでください (RMT ファイルの読み込み方法は、2.6.1 章を参照してください)。

- (1) RFP ツールを立ち上げて、新しいプロジェクトを作成します。
- (2) 通信方法を指定して、PC とターゲットボードを接続します。

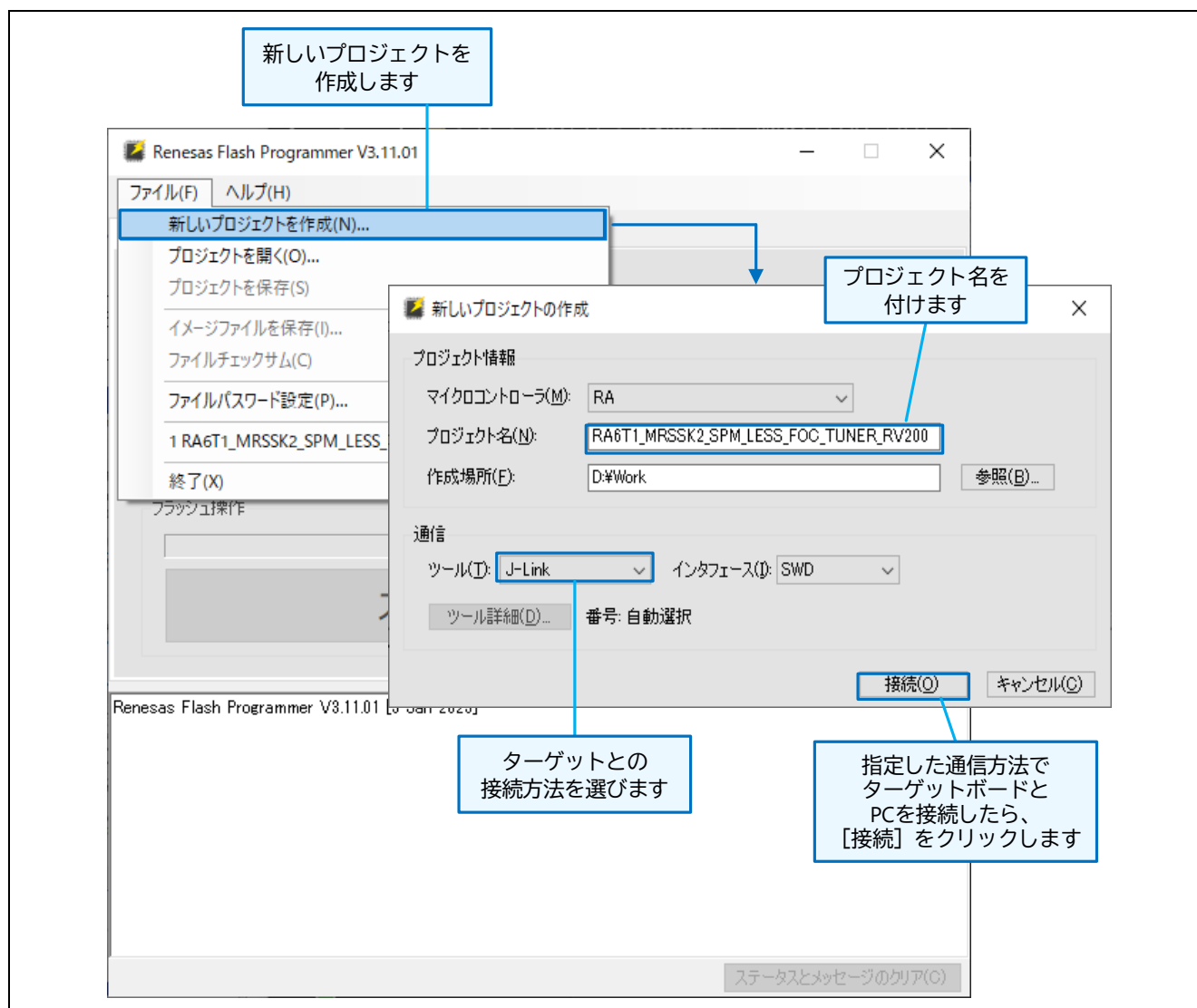
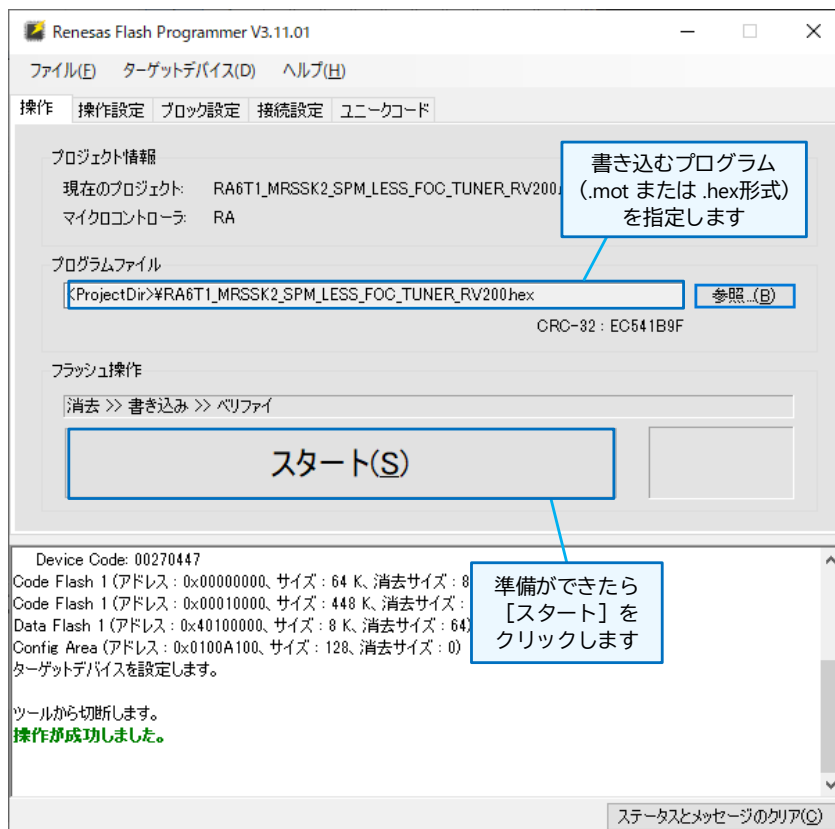
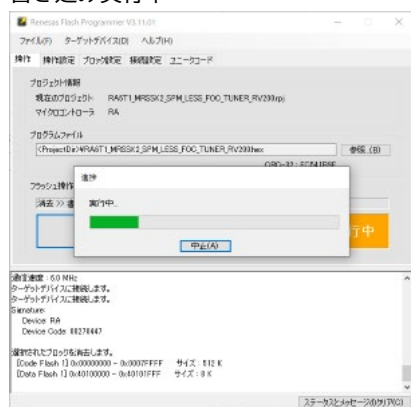


図 2-13 Renesas Flash Programmer による実行プログラムの書き込み (1/2)

- (3) 書き込むプログラムを指定して、書き込みをスタートします。
- (4) 書き込みが正常に終了したら、RFP を終了してターゲットボードを PC から取り外します。



書き込み実行中



書き込み完了（正常終了）

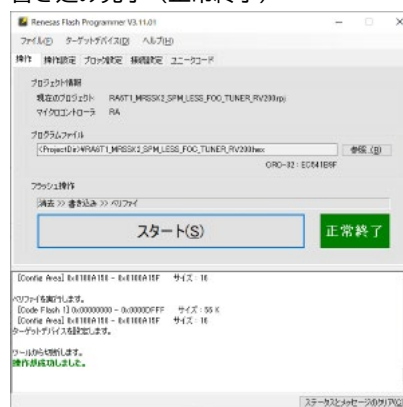


図 2-14 Renesas Flash Programmer による実行プログラムの書き込み (2/2)

【注意】プログラムの書き込み後、Renesas Motor Workbench を使用してプログラムを実行（モータを駆動）する場合、ターゲットボードとの接続は、専用の通信回路（絶縁タイプ）を使用してください。

2.6 変数情報の読み込み

Renesas Motor Workbench は、実行プログラム生成時に出力した Map ファイルを読み込んで、グローバル変数にアクセスします。プログラムを変更して再ビルドした場合は、Map ファイルを再度読み込む必要があります。

Renesas Motor Workbench は、変数情報とツールの設定情報を含んだ「環境情報」を、RMT ファイル（RMW のプロジェクトファイル）として保存したり、読み込むことができます。

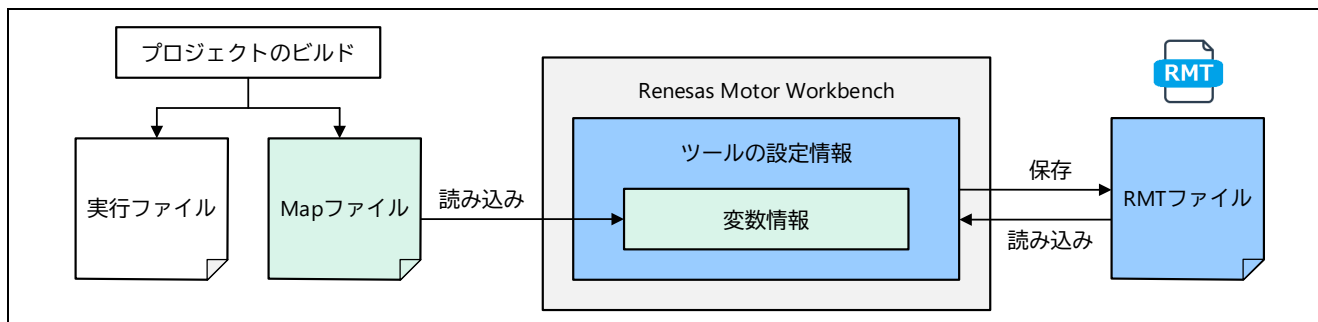


図 2-15 Map ファイルと RMT ファイル（RMW のプロジェクトファイル）

2.6.1 RMT ファイルの読み込み

本パッケージ同梱のサンプルプログラムやルネサス提供のサンプルプログラムなどで、RMT ファイル（変数情報とツールの各種設定情報の両方を含む）が提供されている場合は、下記のいずれかの操作により RMT ファイルを読み込みます。

- Select Navigation 機能（Help メニューから表示可能）を使って読み込む。
- Main Window の File メニューから「Open RMT File」で RMT ファイルを読み込む。
- Main Window の File Information の RMT File 横の「...」ボタンをクリックして RMT ファイルを読み込む。
- Main Window の Project File Path でプロジェクトフォルダを選択し、File List に表示された RMT ファイルをダブルクリックして読み込む。
- PC 上の RMT ファイルを Main Window の File List へドラッグ&ドロップして、そのファイルをダブルクリックして読み込む（この場合、PC 上のプロジェクトフォルダにも RMT ファイルがコピーされます）。

ただし、サンプルプログラムを変更してビルドした場合は、ビルドで生成された Map ファイルを再度 Renesas Motor Workbench に読み込む必要があります。

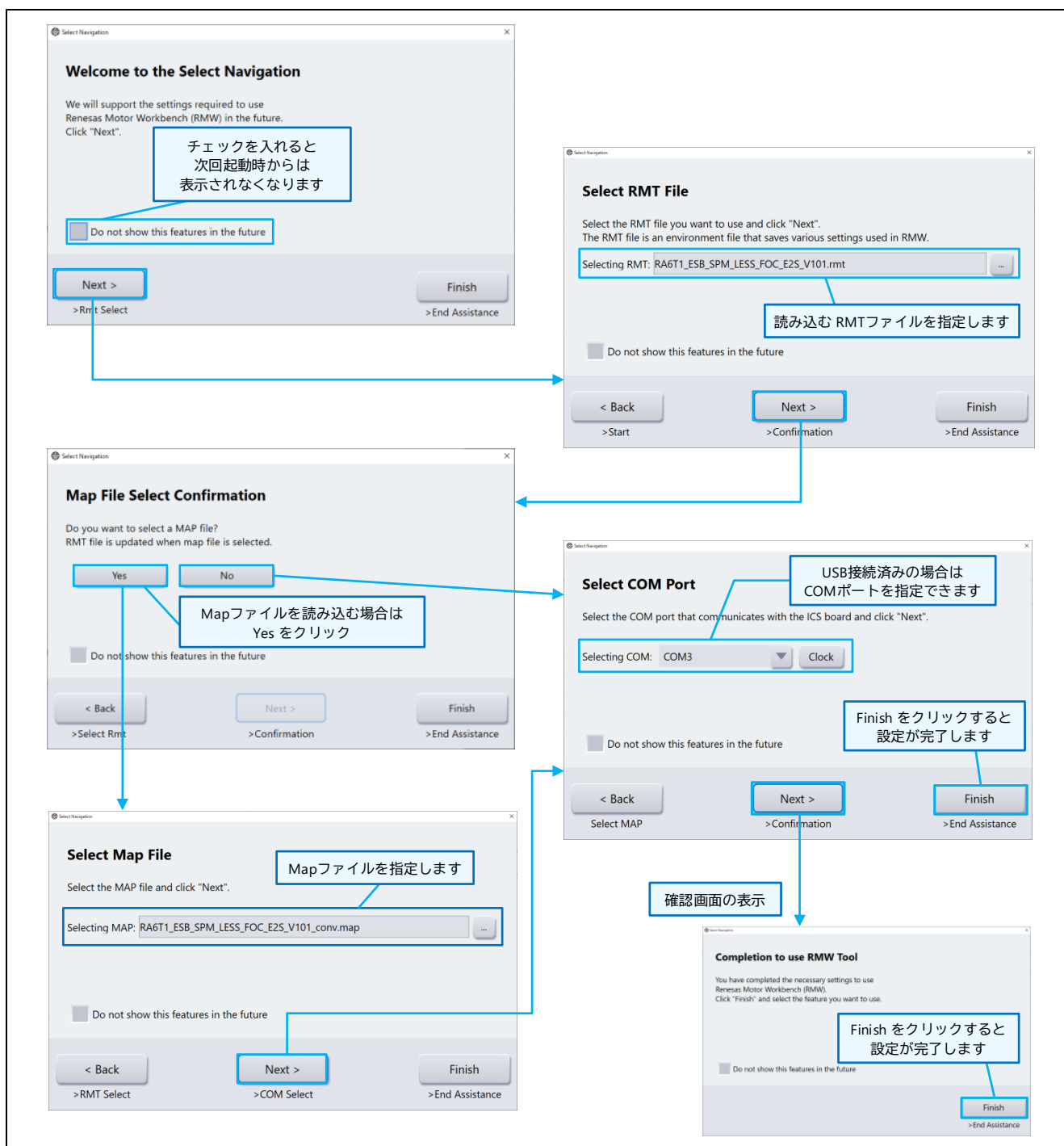


図 2-16 (a) Select Navigation 機能による RMT ファイル読み込み

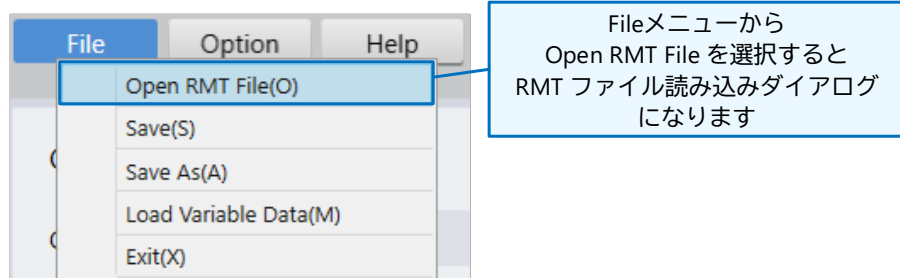


図 2-17 (b) File メニューの「Open RMT File」による RMT ファイル読み込み

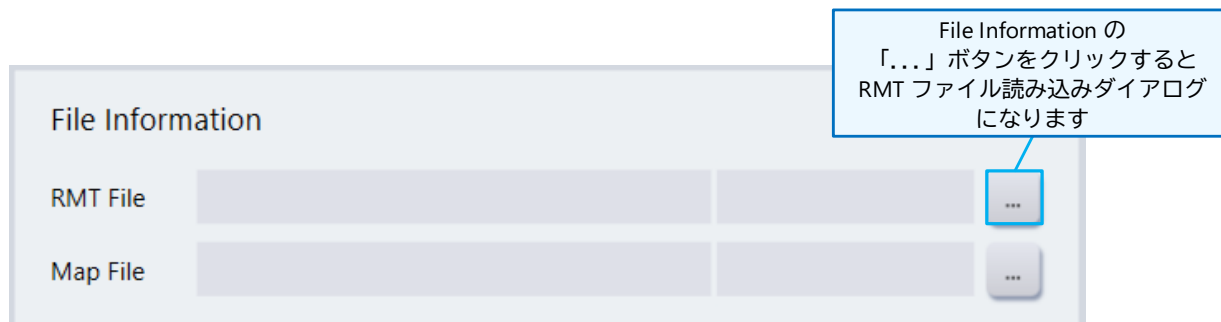


図 2-18 (C) Main Window の File Information からの RMT ファイル読み込み

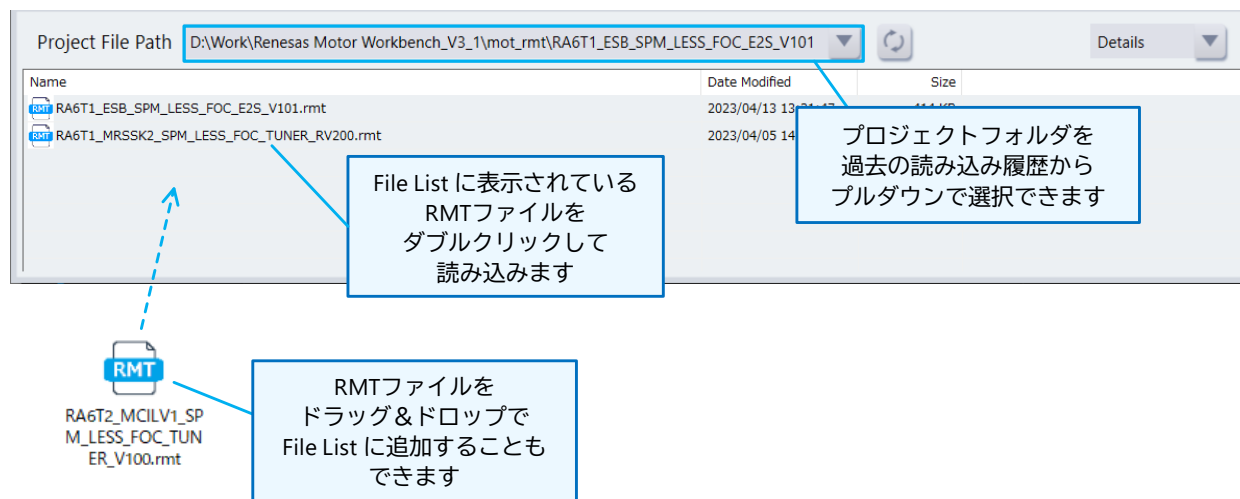


図 2-19 (d)(e) Main Window の File List からの RMT ファイル読み込み

2.6.2 Map ファイルの読み込み

ユーザプログラムを使用する場合（RMT ファイルが存在しない場合）は、まず Renesas Motor Workbench で Map ファイルを読み込む必要があります。

Map ファイルは、プロジェクトのビルドで実行プログラムと同時に生成したものを使用しなければなりません。プログラムを変更して再ビルドした場合は、Map ファイルも再度読み込む必要があります。Map ファイルの生成方法は、「3 Map ファイルの生成方法」を参照してください。

Map ファイル読み込みの手順：

- (1) 以下のいずれかの方法で Map ファイルを読み込みます（操作方法は「2.6.1 RMT ファイルの読み込み」の図を参考にしてください）。
 - (a) Select Navigation 機能（Help メニューから表示可能）を使って読み込む。
 - (b) Main Window の File メニューから「Load Variable Data」を選択し、Map ファイルを指定して読み込む。
 - (c) Main Window の File Information の Map File 横の「...」ボタンをクリックして Map ファイルを読み込む。

RX の場合：

拡張子*.map のファイルを読み込んでください。

RA の場合：

ファイル選択ダイアログボックスのファイル種類「All Files (*.*)」を選択し、拡張子が*.rmap のファイルを読み込んでください。ダイアログボックスのファイル種類「MAP Files (*.map)」を選択したままでは、拡張子*.rmap のファイルが表示されません。

ただし、「3.3 [RA] e2 studio 環境の場合」の Tips に記載した方法や拡張子名を直接変更することにより、拡張子*.rmap を*.map に変更することも可能です。この場合は、ダイアログボックスのファイル種類を変更せずに、変更後の*.map ファイルを読み込んでください。

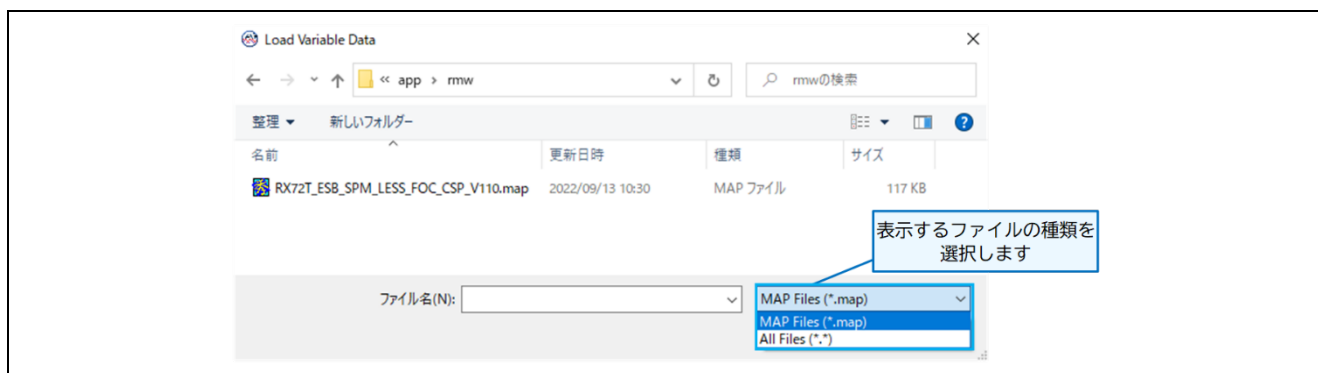


図 2-20 Map ファイル選択ダイアログボックス

- (2) Map ファイルが読み込まれると、User Setting Form 画面が表示されます。読み込みの際、Option Dialog（Option メニューの「Option Dialog」で表示）で、「データ型の変更」または「配列へ変更」の設定がある場合は、その設定を反映した結果がリストへ表示されます。

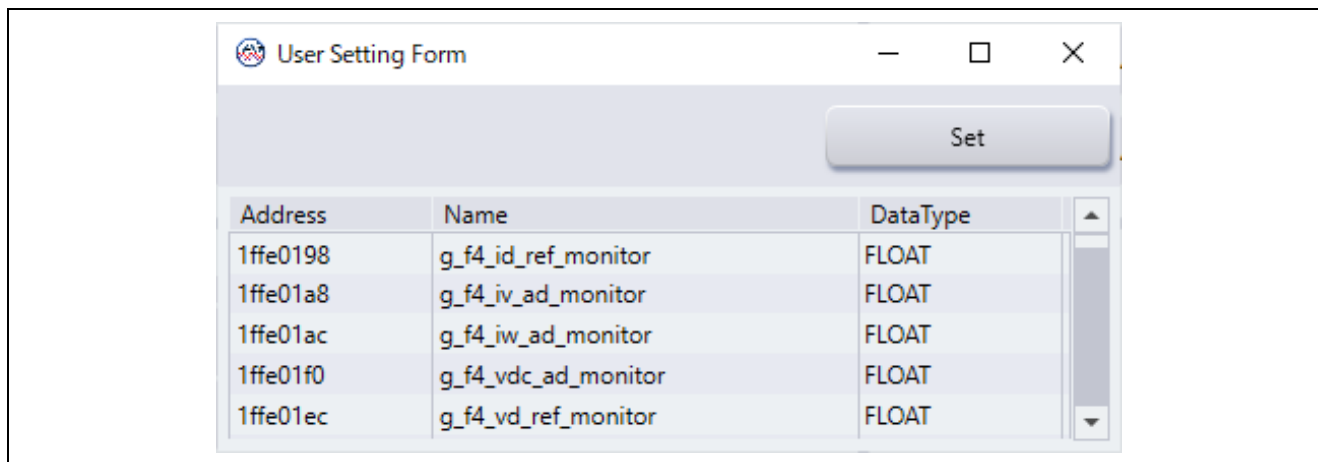


図 2-21 User Setting Form 画面

また、表示された User Setting Form 画面上で、Data Type の設定変更ができます。

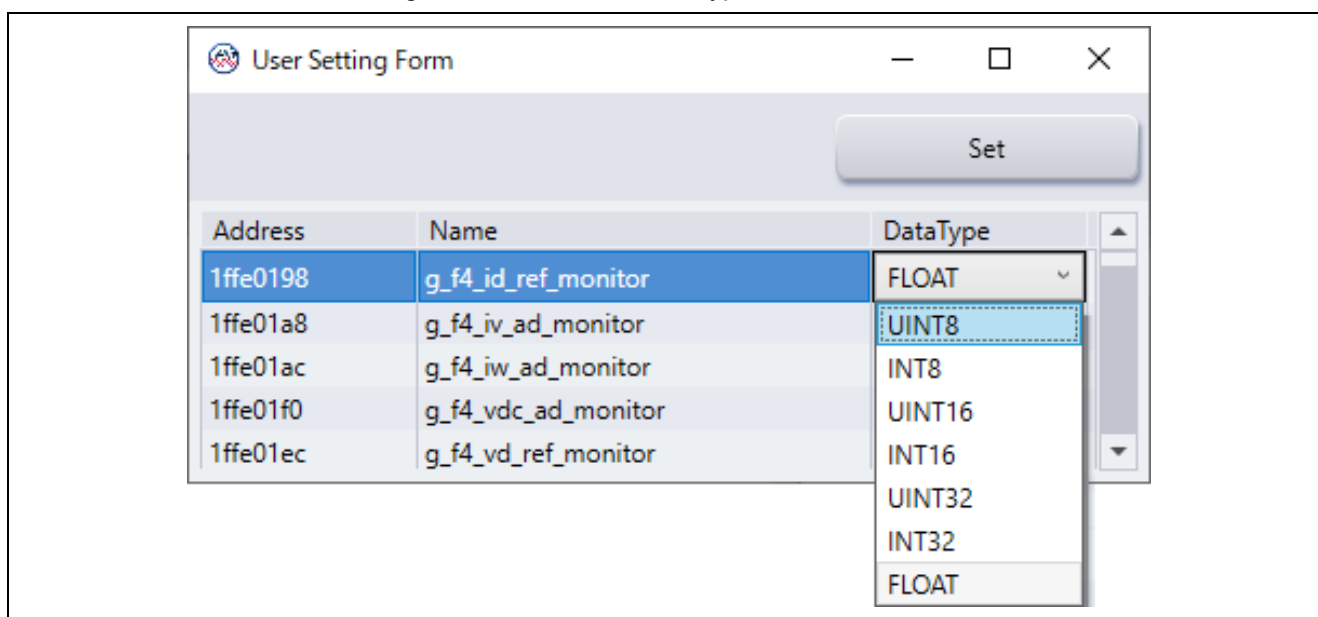


図 2-22 User Setting Form 画面（Data Type の設定変更）

- (3) 「SET」 ボタンをクリックすると、Renesas Motor Workbench へ変数情報が取り込まれます。

なお、「SET」 ボタンで変数情報が取り込まれると、Renesas Motor Workbench の機能で Data Type を変更することはできません。変更したい場合は、再度「Load Variable Data」機能による変数情報の読み込みを行ってください。

2.7 ホスト PC と評価ボードの接続

2.7.1 USB の接続

Renesas Motor Workbench 使用時のシステム要件は、1.2 章を参照してください。

プログラムを書き込んだ評価ボードに電源を供給して、PC と評価ボードを USB ケーブルで接続しま
す。

正常に接続すると、Main Window の Connection の「COM」選択リストに、接続した「COM**」（**は
ポート番号）が表示されます。これは、パソコンとボードの COM 通信が確立し、通信を開始できる状態を
示しています。

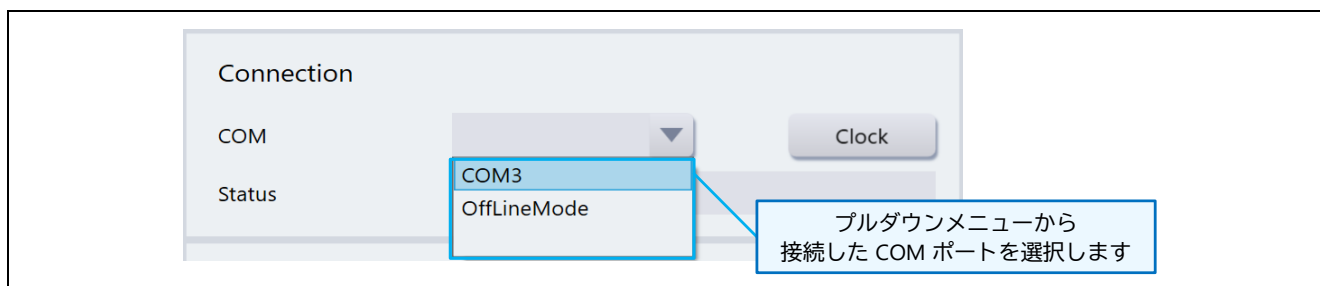


図 2-23 Main Window 「Connection」 の COM 選択リスト

2.7.2 USB 通信の開始

COM 選択リストから「COM**」を選択すると、通信接続処理が開始されます。接続処理の状態は、
Status に表示されます。接続処理が正常に終了すると Status に「Connect -- USB Serial Port」と表示され
ます。この状態になるとツールが利用可能な状態になります。



図 2-24 接続後の Connection 画面 Status 表示

COM 選択リストから「COM**」を選択した際にエラーメッセージが表示された場合は、USB 接続に問題があるか、ターゲットの評価ボードの電源が入っていない、あるいは通信クロックの設定が誤っている可能性があります。

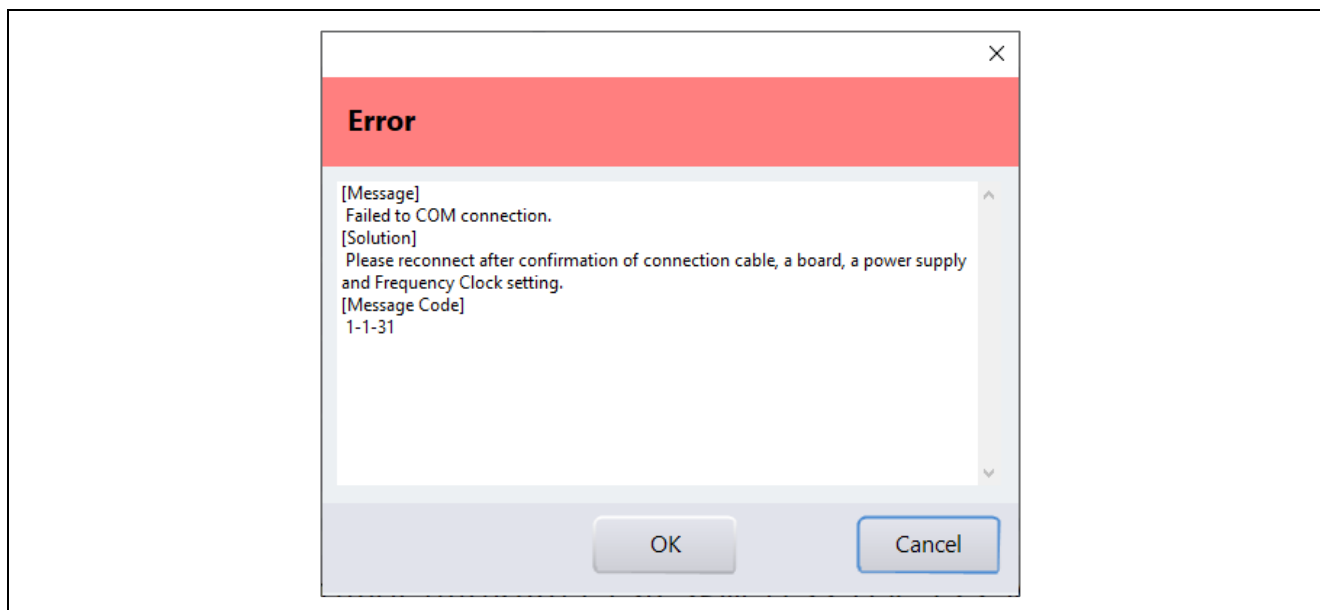


図 2-25 COM 接続時のエラーメッセージ

2.7.3 ツール用通信ボードを使用する場合（クロック周波数の設定）

ツール用通信ボード（ルネサスエレクトロニクス製 MC-COM、またはデスクトップラボ社製 W2002）を
使用する場合は、下記の操作で通信速度の設定を行ってください。

Main Window の「Connection」で「Clock」ボタンをクリックすると、Clock Setting ダイアログが表示さ
れます。ここには、現在設定されているクロック周波数が表示されていますので、その値を変更します。変
更したクロック周波数は、COM 接続時に設定が有効になります。

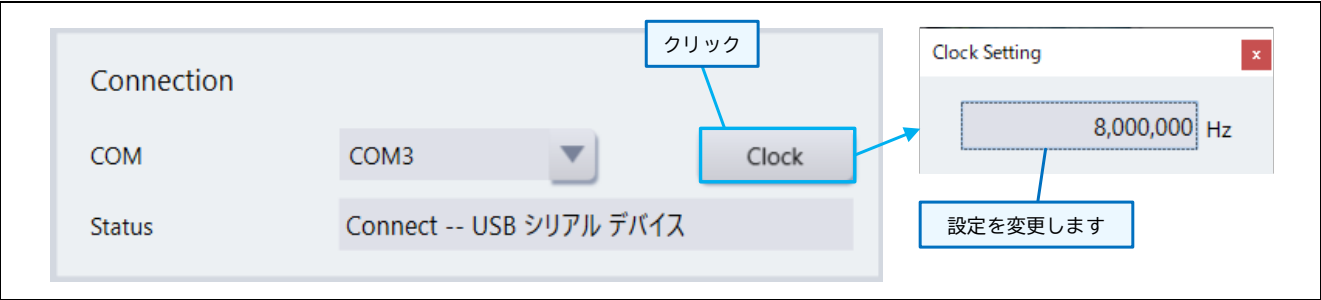


図 2-26 Clock Setting ダイアログ

設定するクロック周波数は、通信レートを 8 倍した値を設定してください。

例： 通信レート 1 Mbps の場合、クロック周波数 8 MHz（8,000,000 Hz）を設定

通信レート 5 Mbps の場合、クロック周波数 40 MHz（40,000,000 Hz）を設定

ルネサスエレクトロニクス製ツール用通信ボード MC-COM を使用する場合は、以下の通信レート（ク
ロック周波数）が選択できます。クロック周波数は、MC-COM のジャンパ（JP2）に合わせて設定してくだ
さい。

表 2-3 ツール用通信ボード MC-COM の JP2 設定と選択可能なクロック周波数

ジャンパ JP2	選択可能なクロック周波数
ショート	1 Mbps（8 MHz）、5 Mbps（40 MHz）、7.5 Mbps（60 MHz）、 10 Mbps（80 MHz）、15 Mbps（120 MHz）
オープン	6.25 Mbps（50 MHz）、8.33 Mbps（6666666 Hz）、12.5 Mbps（10 MHz）、 16.66 Mbps（13333333 Hz）

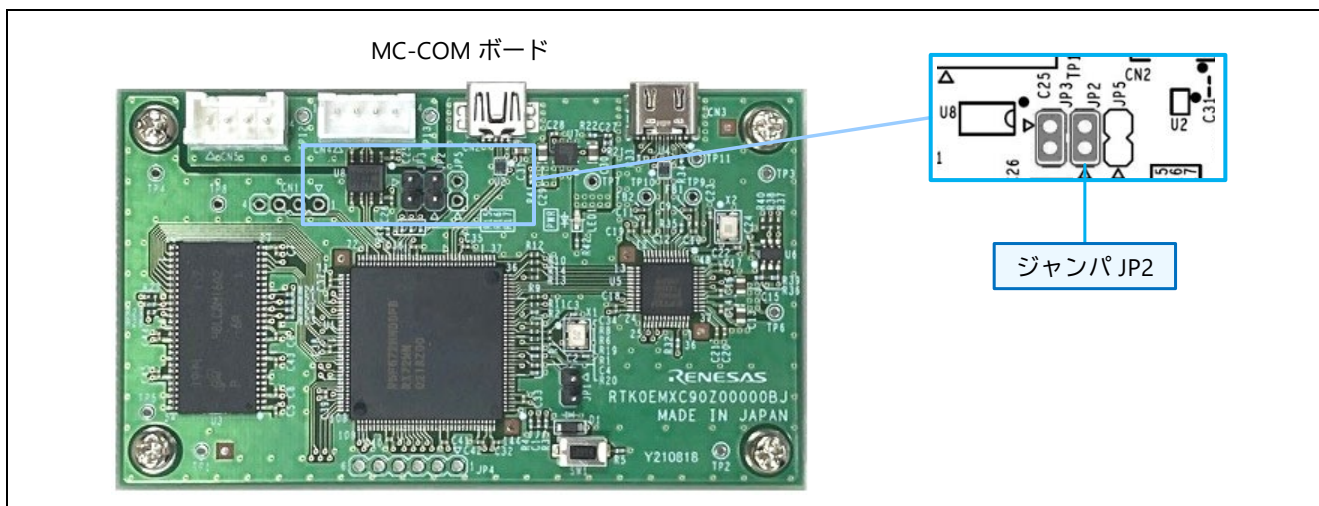


図 2-27 ツール用通信ボード MC-COM のジャンパ JP2

各 MCU において、以下の条件における最大通信レートを表 2-4, 2-5 に示します。

条件

- ・ CPU クロック周波数を最大設定
- ・ MC-COM が対応するボーレート

表 2-4 最大通信レート (RX マイコン用)

製品名	CPU クロック	供給クロック (UART)	ボーレート	サンプリング周期	Lib API(ics2_init) 設定値	Lib API(ics2_init) 設定値
					第 4 引数 char speed	第 5 引数 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

表 2-5 最大通信レート（RA マイコン用）

製品名	CPU ク ロック	供給ク ロック (UART)	ボーレート	サンプリ ング周期	Lib API(ics2_init) 設定値	Lib API(ics2_init)設 定値
					第 3 引数 char speed	第 4 引数 char mode
RA8T2	1GHz	120MHz	10Mbps	50us/4ch	1	2
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
RA2T1	64MHz	32MHz	5Mbps	50us/4ch	0	2

2.7.4 ビルトイン型通信ライブラリを使用する場合（ボーレートの設定）

ビルトイン型通信ライブラリを使用する場合、下記の操作でボーレートの設定を行ってください。

Main Window で「Option」メニューの「Baudrate Dialog」を選択すると、Baudrate Setting ダイアログが表示されます。ここには、現在設定されているボーレートが表示されていますので、その値を変更します。変更したボーレートは、COM 接続時に設定が有効になります。

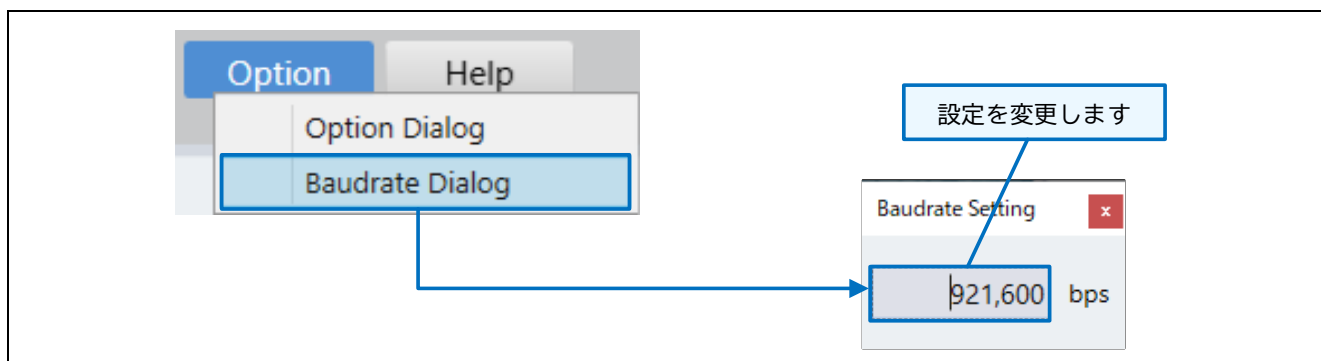


図 2-28 ボーレートの設定

設定するボーレートは、ビルトイン型通信ライブラリを組み込んだプログラムで設定した値にしてください。また、設定値は接続している USB シリアル変換基板等に PC から設定可能な値にする必要があります。設定可能な値は、Windows のコントロールパネルにあるデバイスマネージャから、COM ポートのプロパティ（ポートの設定）で確認できます。

2.7.5 USB 接続で問題が発生した場合

USB 接続において何らかの問題が発生した場合は、次のように対応してください。

- 評価ボードとの接続方法、評価ボードへの電源供給方法を確認してください。
- 評価ボードの実行プログラムと Renesas Motor Workbench に読み込んだ変数リストの内容が、同一ビルドで生成されたものであることを確認してください。
- 評価ボードをリセットする、USB ケーブルをつなぎ直す、別の USB ポートに変えてみる、などを試してください。

2.8 操作説明

2.8.1 内部ツールの起動

Map ファイルまたは RMT ファイルを読み込み、ターゲットが正しく接続されると、Main Window の Select Tool に利用可能なツールのアイコンが表示されます（Main Window の表示や利用できるツールはサンプルプログラムにより異なります）。

Select Tool のツールアイコンをクリックするとツールが起動し、各ツールのウィンドウが開きます。

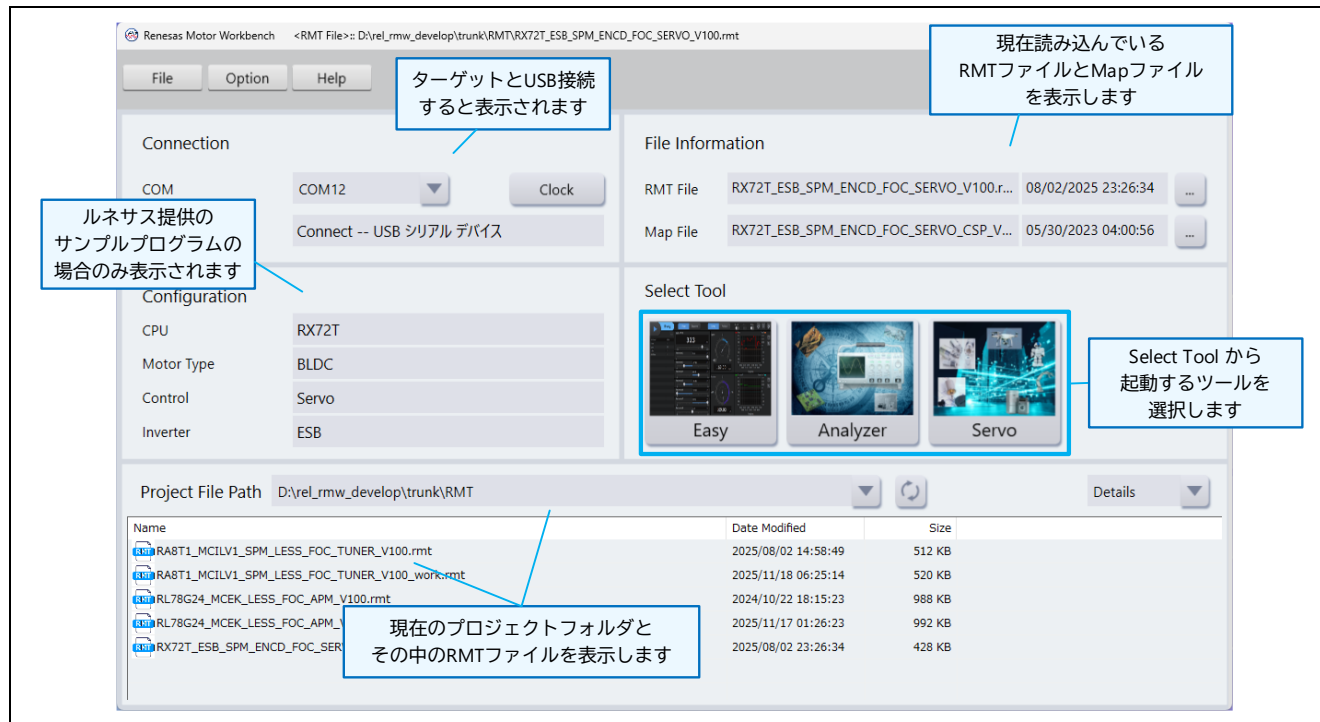


図 2-29 Main Window（ターゲットに接続後）

2.8.2 GUI 操作の基本

ここでは、Renesas Motor Workbench の Main Window と内部ツール（Easy、Analyzer、Servo、Tuner）に関わる基本的な GUI 操作方法を説明します。

各機能の詳細については、Renesas Motor Workbench のユーザーズマニュアルを参照してください。

(1) ツール切り替えボタン

Main Window の Select Tool からツールを起動します。各ツールの画面上部中央に「ツール切り替えボタン」が表示され、クリックすると他のツールへ切り替えができます。

ただし、動作中（モータ駆動中）のツール切り替えはできません。

(2) Main Window 切り替えボタン

各ツールの画面上部右側に「Main Window ボタン」が表示されます。クリックすると Main Window へ戻ります。

ただし、動作中（モータ駆動中）の Main Window への切り替えはできません。

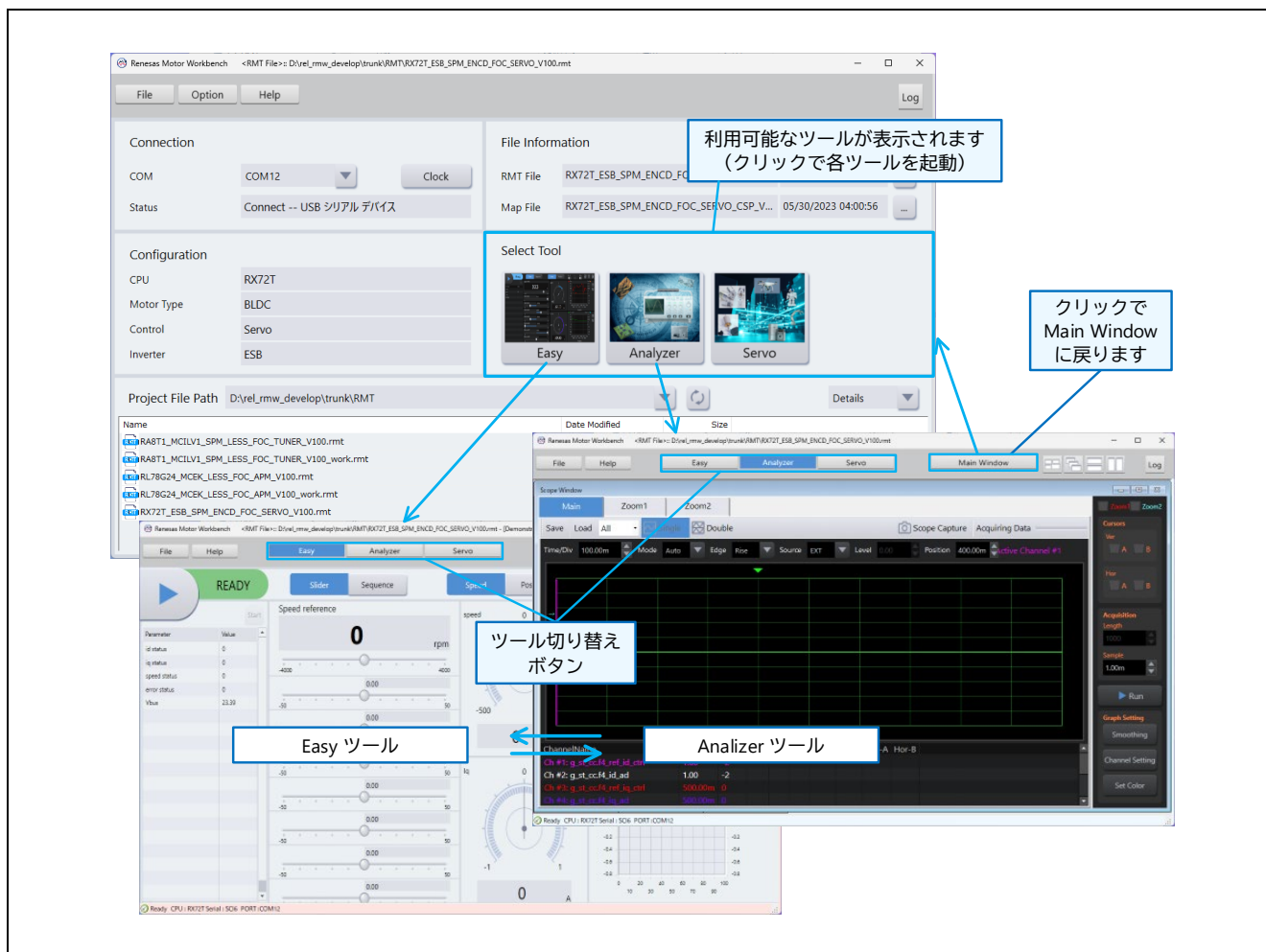


図 2-30 ツール切り替えボタンと Main Window 切り替えボタン

(3) ウィンドウ表示切り替えボタン

各ツール（Analyzer、Servo、Tuner）画面では、ウィンドウの表示形態を下記ボタンから変更することができます。また、最大化したウィンドウの解除もできます。

ウィンドウ一覧ボタン（下図 (a) のボタン）で最前面に表示する画面を選択できます。ただし、ウィンドウをツール外にフレームアウトしている場合は、起動中画面のリストには表示されません（その場合は、Windows のタスクバーで画面を選択してください）。

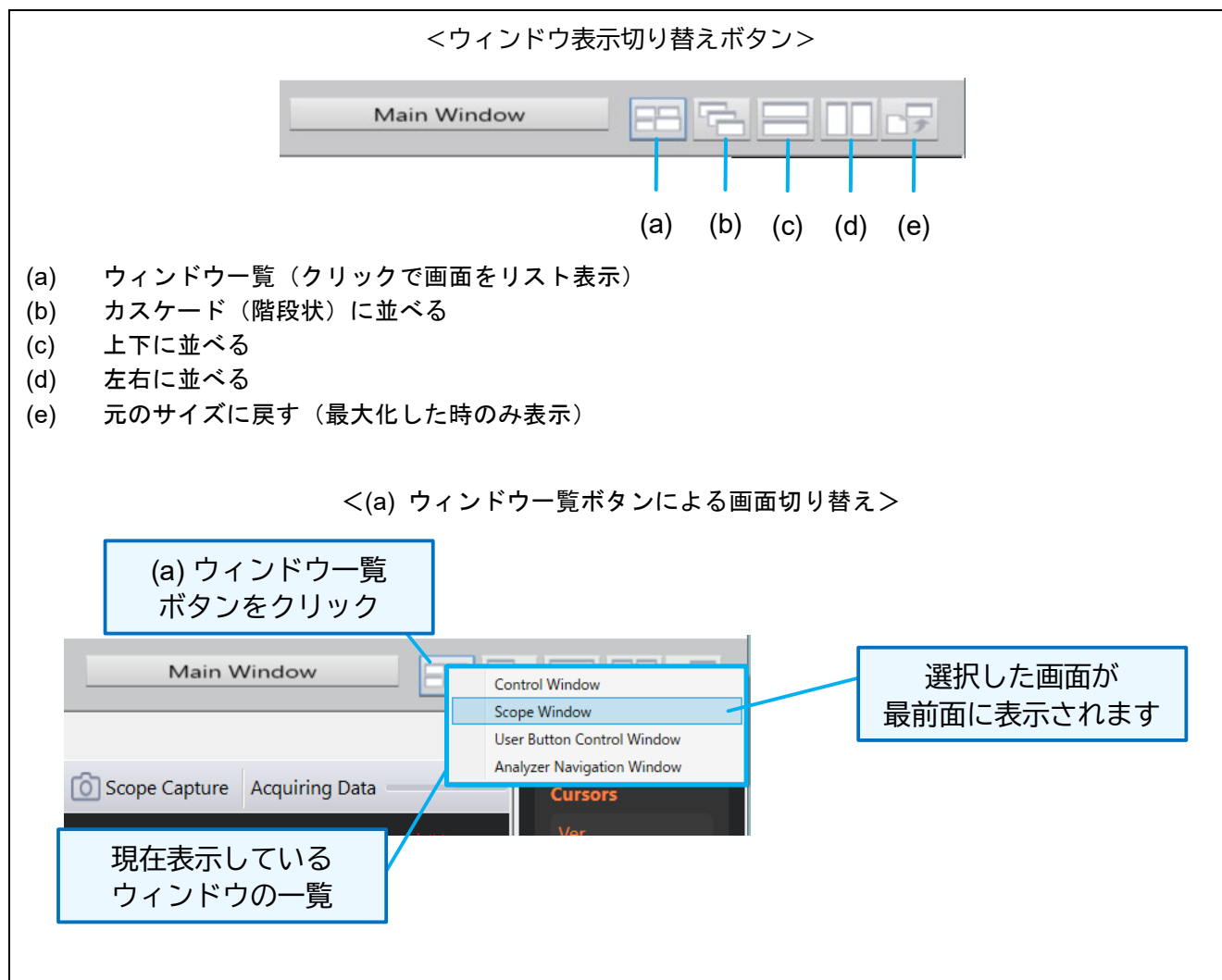


図 2-31 各ツール画面のウィンドウ表示切り替えボタン

(4) ウィンドウのフレームアウトとフレームイン

ツール内のウィンドウ（一部を除く）は、タイトル部をドラッグしてツール枠外へ移動（フレームアウト）できます。また、逆にウィンドウをドラッグして元のツール枠内へ戻すこと（フレームイン）もできます。

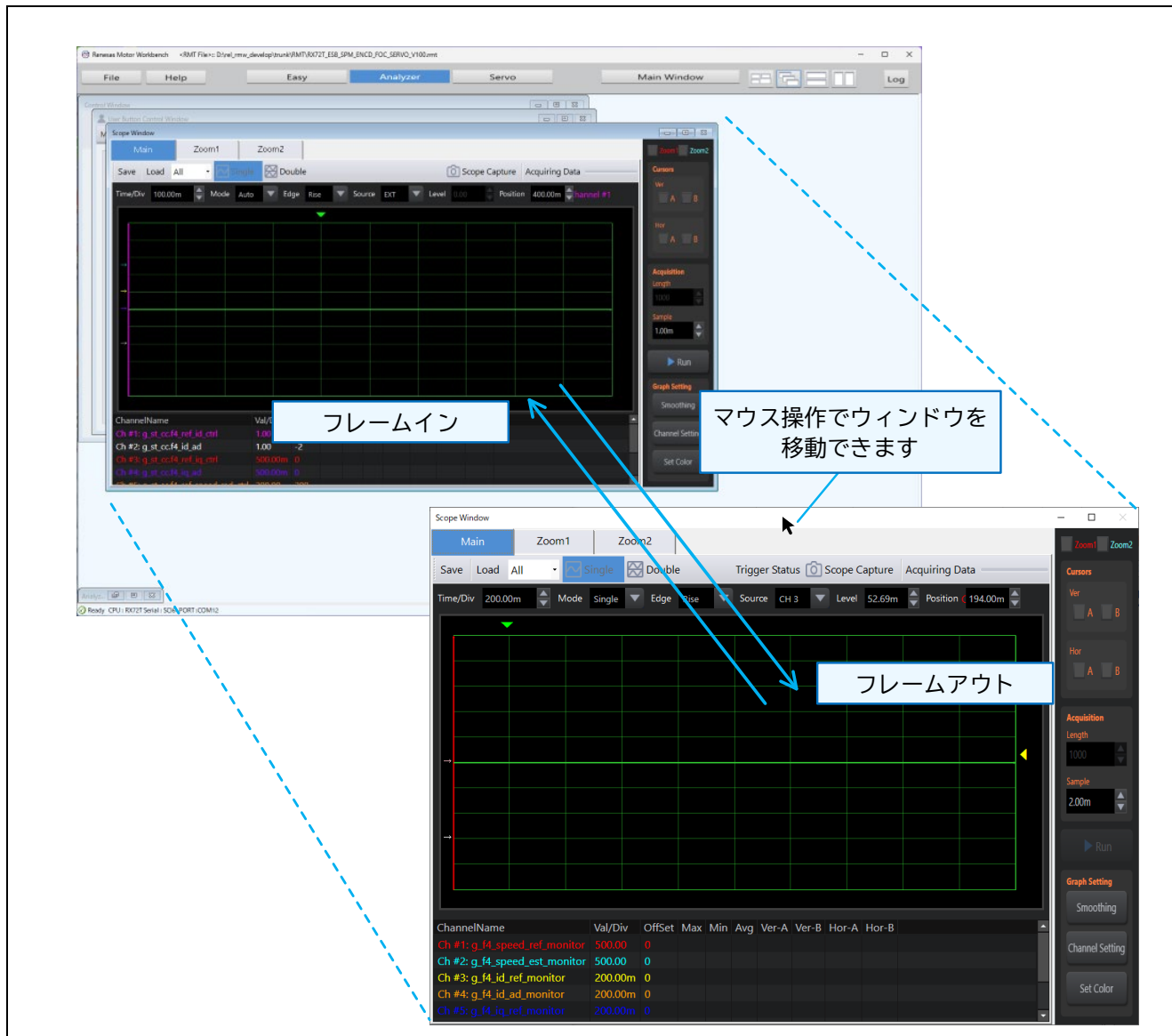


図 2-32 ウィンドウのフレームアウトとフレームイン

(5) ログコンソールの表示切り替えボタン

Main Window 及び、各ツール（Easy GUI、Analyzer、Servo、Tuner）画面では一部の操作でログを記録します。ログの表示は上部メニューにあるログ表示切り替えボタンから行います。ボタンを押下するとウィンドウの右側にログコンソールを表示します。表示中にボタンを押下すると、ログコンソールを閉じます。

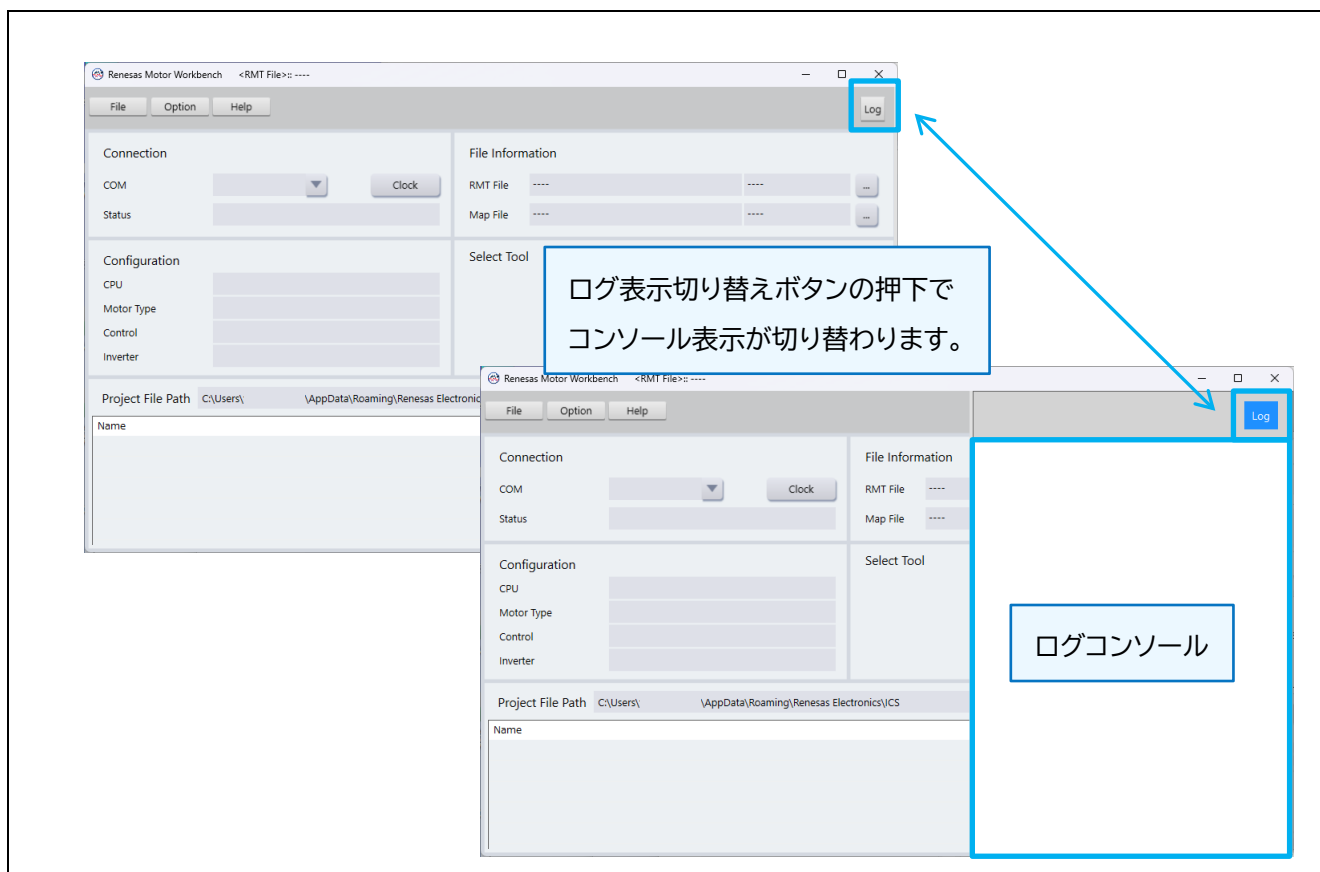


図 2-33 ログコンソールの表示切替え

2.9 RMT ファイルの保存と Renesas Motor Workbench の終了

RMT ファイルは、Main Window の「File」メニューから「Save」または「Save As」で保存できます。

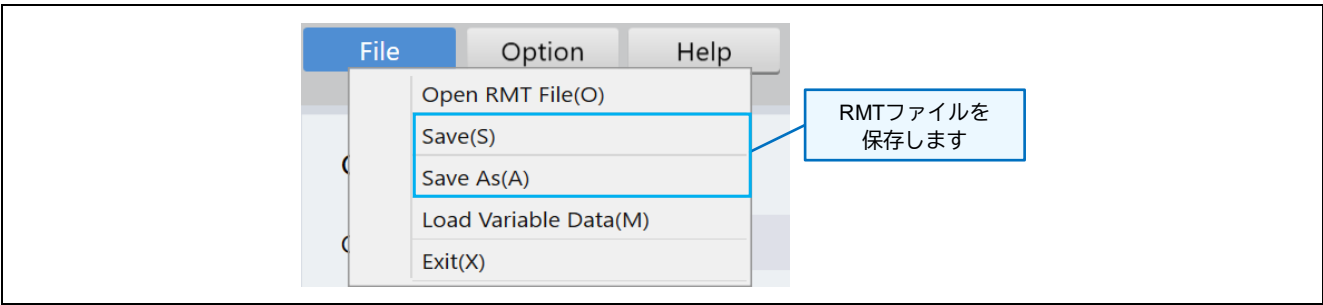


図 2-34 RMT ファイルの保存

Renesas Motor Workbench を終了するには、Main Window の「File」メニューから「Exit」を選択するか、画面右上の「閉じる」ボタンをクリックします。
終了時には、以下のように終了画面が表示されます。

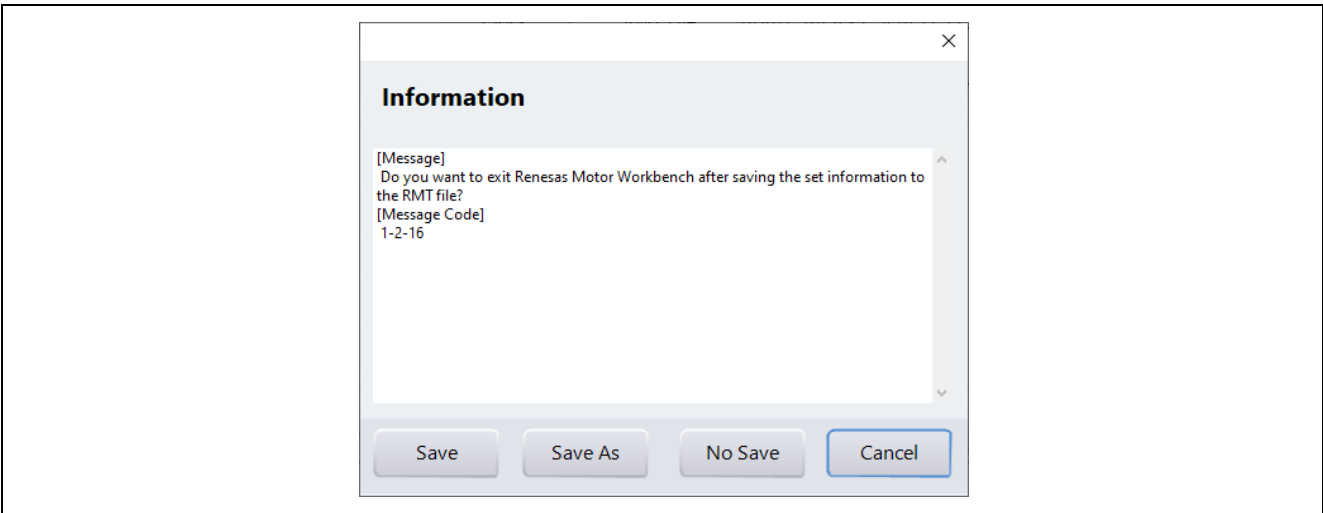


図 2-35 Renesas Motor Workbench 終了画面

各ボタンの機能は以下の通りです。

表 2-6 終了画面のボタン機能

ボタン	機能
Save	Renesas Motor Workbench の環境情報を読み込み中の RMT ファイルに上書き保存して終了
Save As	Renesas Motor Workbench の環境情報を、ファイル名を指定した RMT ファイルに保存して終了
No Save	Renesas Motor Workbench の環境情報を保存しないで終了
Cancel	終了処理自体をキャンセルする

3. Map ファイルの生成方法

Renesas Motor Workbench は、評価ボードのマイコンに書き込んだプログラムのグローバル変数の情報（Map ファイル）を読み込むと、グローバル変数へアクセスが可能となります。この Map ファイルは、プログラムをビルドした時に、実行ファイル（制御ソフト）と共に生成することができます。

統合開発環境「CS+（CC コンパイラ）」、「e²studio」、「e²studio（RA）」において、Map ファイルを生成するための設定手順を以下に示します。

3.1 [RX] CS+（CC コンパイラ）環境の場合

設定手順：

- ① CS+を起動して該当プロジェクトを表示し、「プロジェクト・ツリー」から、該当プロジェクト情報を展開します。
- ② 「プロジェクト・ツリー」の「CC-RX（ビルド・ツール）」を右クリックして、メニューから「プロパティ」を選択すると、画面右側に「プロパティ」画面が表示されます。
- ③ 「プロパティ」内の「リンク・オプション」をクリック。
- ④ 「リスト」と「その他」の情報を展開します。
- ⑤ 「リスト」内の「リンケージ・リスト・ファイルを出力」に「はい(リスト内容=すべて)(-LIST - SHow=ALL)」が設定されている場合は、そのまま「⑨」へ進んでください。それ以外は「⑥」へ進んでください。
- ⑥ 「リンク」の「リンケージ・リスト・ファイルを出力する」で「はい(リスト内容=選択)(-LIST)」を選択します。
- ⑦ 「リンク」の「シンボル情報を出力する」で「いいえ」を選択します。
- ⑧ 「その他」の「その他の追加オプション」に「-Show=Symbol,struct」（区切りはカンマ）を追記します。
- ⑨ 上記の設定をしてビルドすると、「変数情報」を含んだ Map ファイルが生成されます。

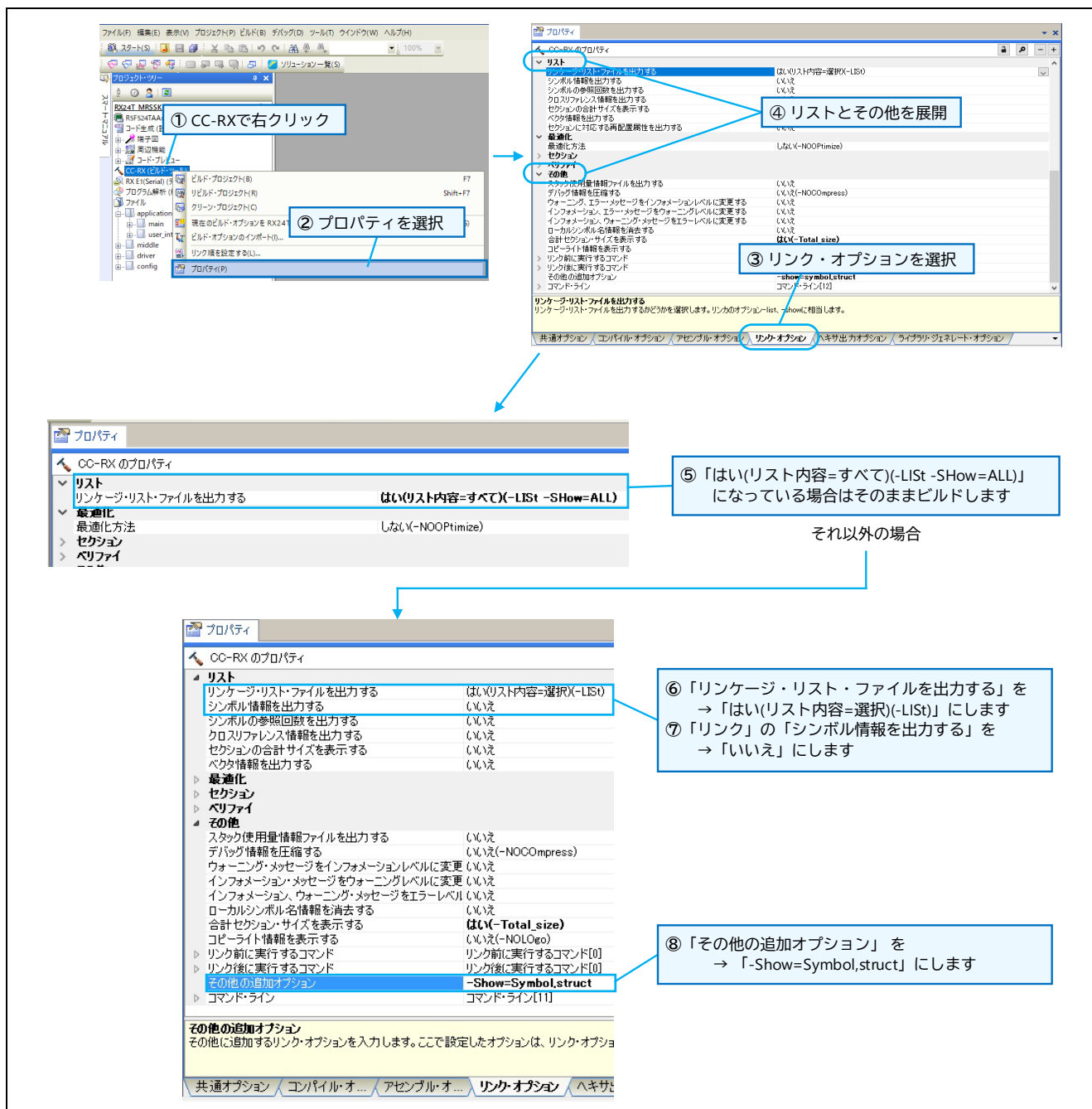


図 3-1 CS+（CC コンパイラ）プロパティ画面の設定

Tips : ビルド完了後、RMT ファイルの保存先に Map ファイルをコピーする

RMT ファイルを読み込んだ後に Map ファイルを読み込むと、読み込んだ RMT ファイルのディレクトリパスが指定された状態でダイアログボックスが開きます。これは、直前にファイル指定ダイアログボックスで操作したディレクトリが開くためです。そのため、Map ファイルを RMT ファイルのディレクトリにあらかじめコピーしておく、ダイアログボックスでディレクトリパスを変更することなく RMT/Map ファイルを選択することができます。

以下のように、リンク・オプション「その他」の「リンク後に実行するコマンド」に Map ファイルを RMT ファイルの保存ディレクトリにコピーするコマンドを入力しておく、ビルド実行後に自動でコピーすることができます。

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

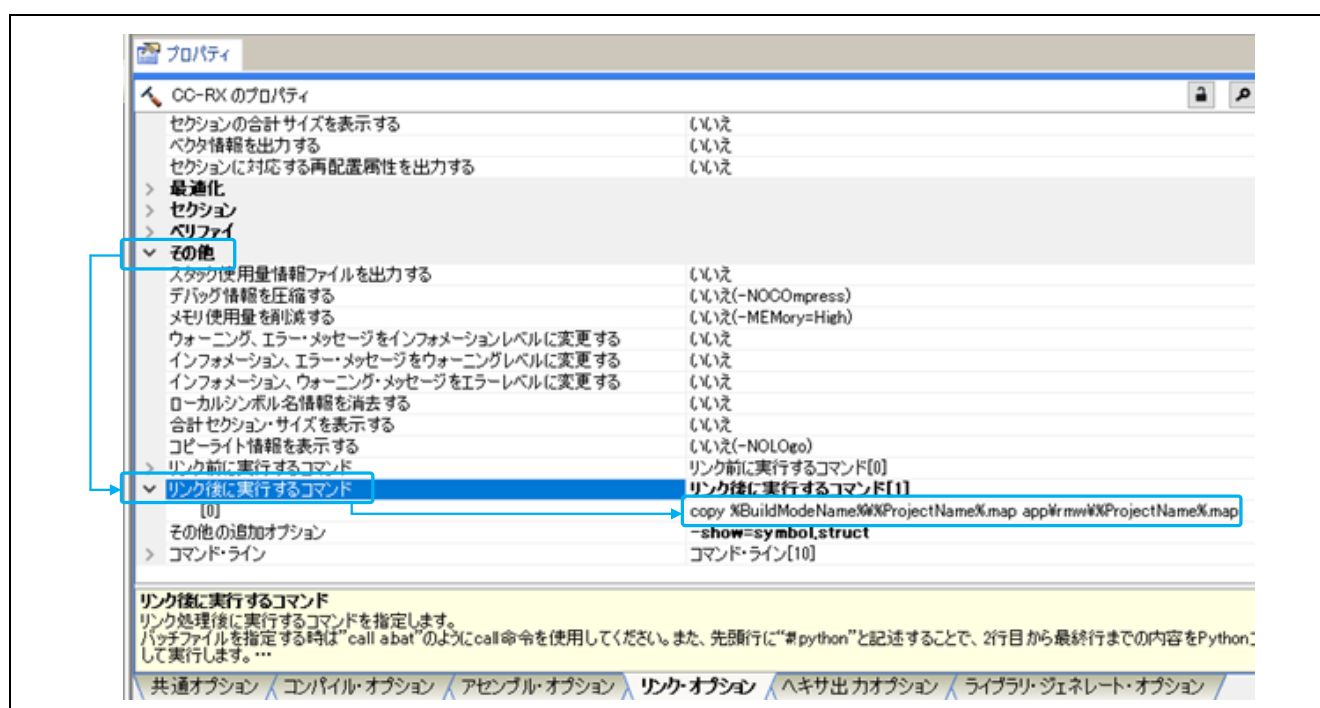


図 3-2 CS+ (CC コンパイラ) Map ファイルコピーの設定

3.2 [RX] e² studio 環境の場合

設定手順 :

- ① e² studio を起動して該当プロジェクトを表示します。
- ② 「プロジェクト・エクスプローラ」より該当プロジェクトを右クリックしてメニューを表示して、「プロパティ」をクリックするとプロパティ画面が表示されます。
- ③ 画面左側「C/C++ ビルド」を展開して「設定」を選択。
- ④ 画面右側「設定」内のタブ「ツール設定」タブを選択。
- ⑤ 「Linker」の中の「ユーザー」を選択。
- ⑥ 画面右下の「ユーザー定義オプション (コマンドラインの後ろへ追加)」で「+」ボタンを押して、「-Show=Symbol,struct」(区切りはカンマ)を追記します。
- ⑦ 上記の設定をしてビルドすると、「変数情報」を含んだ Map ファイルが生成されます。

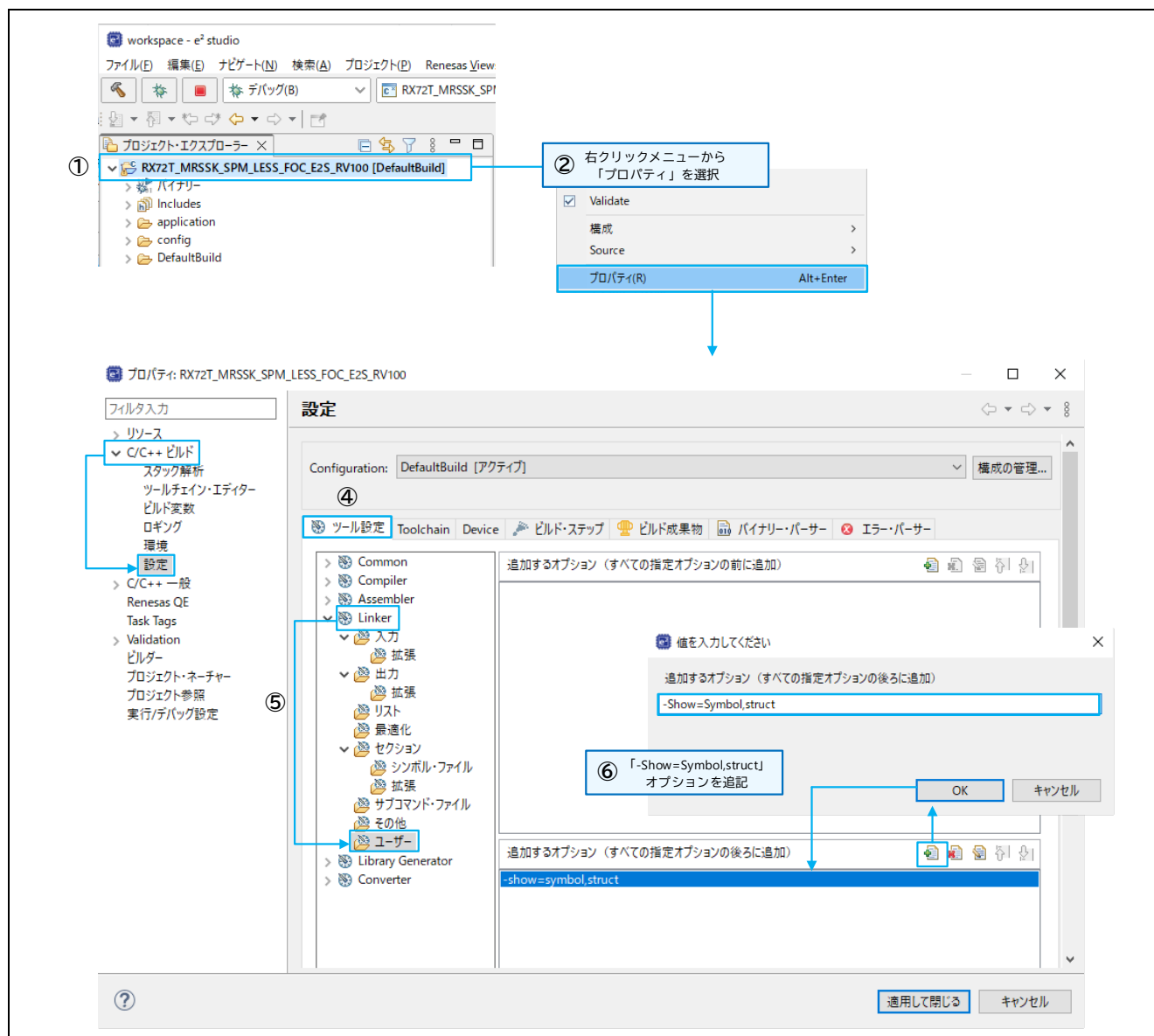


図 3-3 e2studio のプロパティ設定

Tips : RMT ファイルの保存先に Map ファイルをコピーする

RMT ファイルを読み込んだ後に Map ファイルを読み込むと、読み込んだ RMT ファイルのディレクトリパスが指定された状態でダイアログボックスが開きます。これは、直前にファイル指定ダイアログボックスで操作したディレクトリが開くためです。そのため、Map ファイルを RMT ファイルのディレクトリにあらかじめコピーしておく、ダイアログボックスでディレクトリパスを変更することなく RMT/Map ファイルを選択することができます。

以下のように、「C/C++ビルド」-「設定」内の「ビルド・ステップ」タブを選択し、「ビルド後のステップ」に Map ファイルを RMT ファイルの保存ディレクトリにコピーするコマンドを入力しておく、ビルド実行後に自動でコピーすることができます。

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

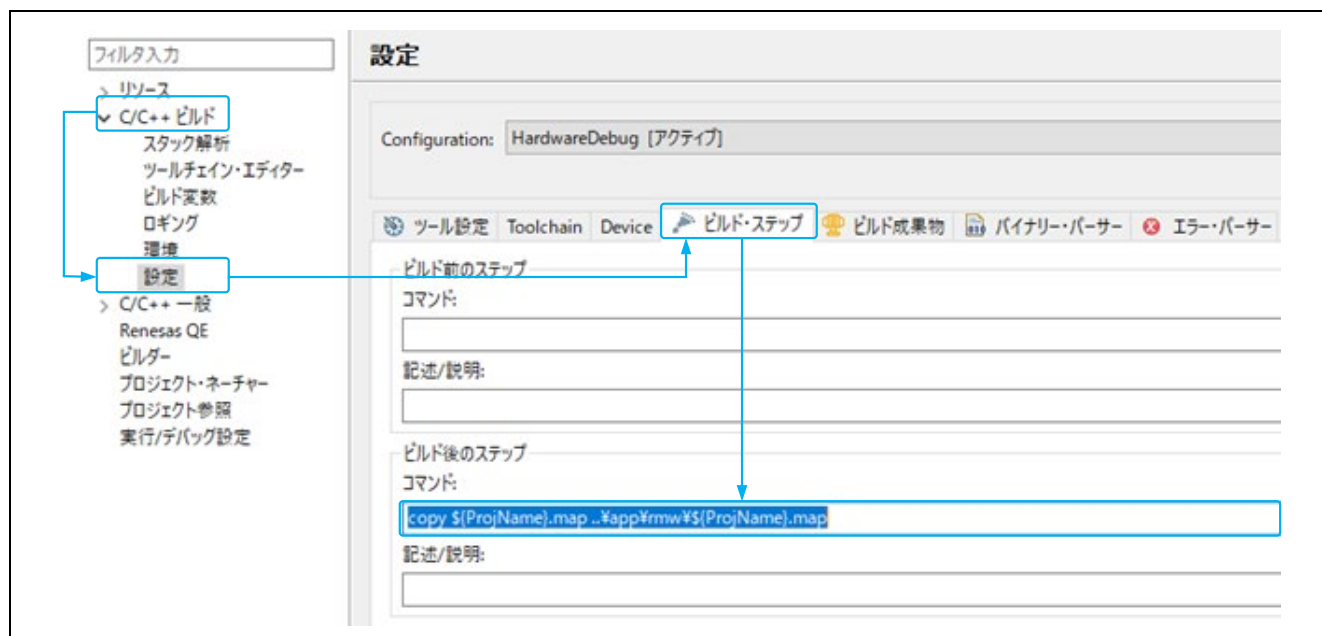


図 3-4 e2studio Map ファイルコピーの設定

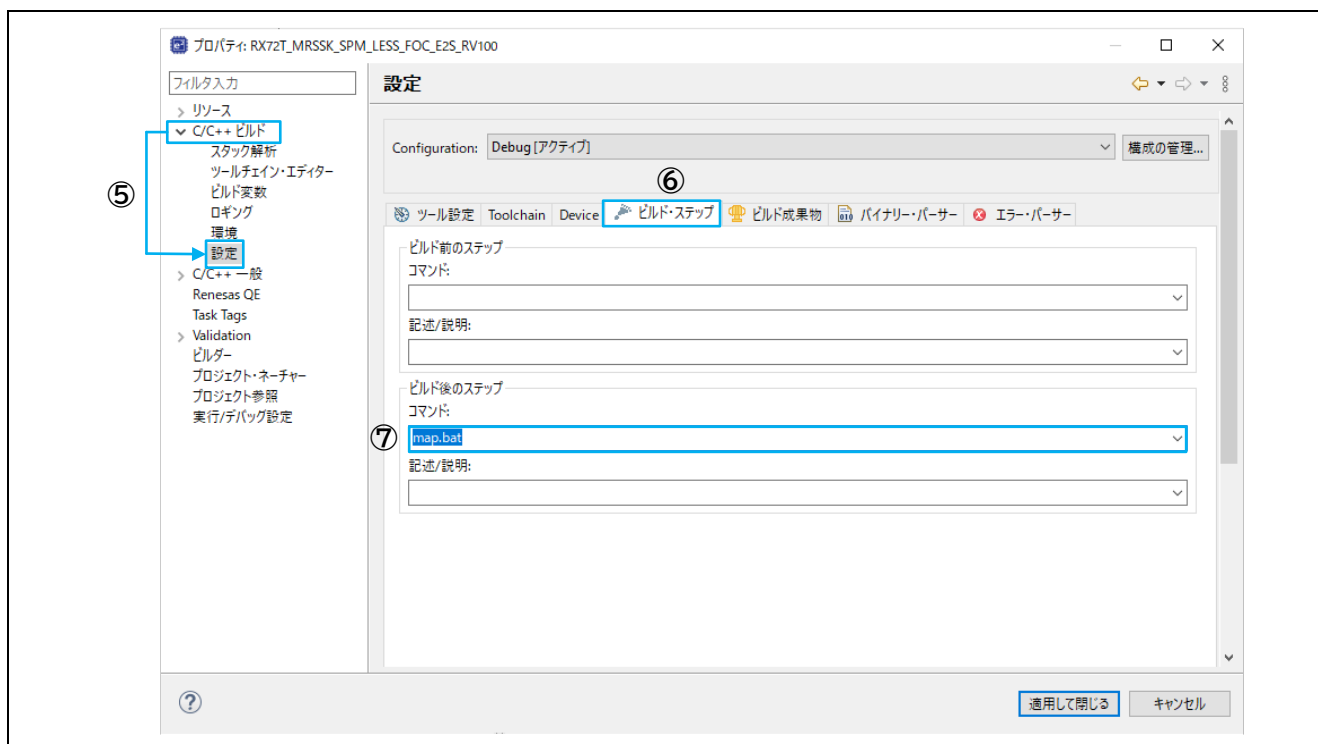
3.3 [RA] e² studio 環境の場合

設定手順：

- ① Renesas Motor Workbench に付属の「ElfMapConverter.exe」をプロジェクトの出力フォルダにコピーします。
- ② 以下のコマンドを記述した bat ファイルを作成します（ファイル名は自由です）。

```
arm-none-eabi-objdump.exe -W プロジェクト名.elf>プロジェクト名.txt
ElfMapConverter.exe プロジェクト名.txt
```

- ③ 「e2studio」を起動して、プロジェクトを表示します。
- ④ 「プロジェクト・エクスプローラ」より該当プロジェクトを右クリックしてメニューを表示して、「プロパティ」をクリックするとプロパティ画面が表示されます。
- ⑤ 画面左側「C/C++ ビルド」を展開して「設定」を選択。
- ⑥ 画面右側「設定」内のタブ「ビルド・ステップ」を選択。
- ⑦ 「ビルド後のステップ」のコマンド欄に②で作成した bat ファイルを追記します（ここでは、ファイル名を「map.bat」としています）。
- ⑧ 上記の設定でビルドすると、変数情報を含んだ Map ファイル（.rmap）が生成されます。

図 3-5 e²studio (RA) のプロパティ設定

Tips : RMT ファイルの保存先に Map ファイルをコピーする

RMT ファイルを読み込んだ後に Map ファイルを読み込むと、読み込んだ RMT ファイルのディレクトリパスが指定された状態でダイアログボックスが開きます。これは、直前にファイル指定ダイアログボックスで操作したディレクトリが開くためです。そのため、Map ファイルを RMT ファイルのディレクトリにあらかじめコピーしておく、ダイアログボックスでディレクトリパスを変更することなく RMT/Map ファイルを選択することができます。

また、本章で生成した Map ファイル(*.rmap)の拡張子を*.map に変更してコピーすると、Map ファイルを読み込むダイアログボックスで、表示するファイルの種類を変更する必要がなくなります。

本章の設定手順②で説明した bat ファイルへ以下のようなコマンドを追記すると、ビルド実行後に拡張子を変更した Map ファイルを自動でコピーすることができます。追記コマンドの例では、拡張子を *.rmap から *.map に変更したことが分かるように、コピー後の Map ファイル名に”_conv”を付加しています。

```
copy プロジェクト名.rmap <RMT ファイル保存ディレクトリ>%プロジェクト名_conv.map
```

4. 通信ライブラリ

4.1 MCU 別 通信ライブラリ

Renesas Motor Workbench を利用するには、ユーザプログラムに通信ライブラリをインクルードする必
要があります。通信ライブラリは Renesas Motor Workbench パッケージに同梱されています

(「communication library」フォルダ)。

各マイコンに提供される通信ライブラリを以下に示します。

表 4-1 通信ライブラリ (RX マイコン用) (1/3)

対応 CPU	RX23T	RX24T	RX24U
ファイル	ics_RX23T.obj ics_RX23T.h	ics_RX24T.obj ics_RX24T.h	ics_RX24U.obj ics_RX24U.h
通信レート	0.5 Mbps ~ 5 Mbps		
ポート	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
サポート変数タイプ	8bit 符号なし 整数型 (数値表示・設定・波形表示) 8bit 符号あり 整数型 (数値表示・設定・波形表示) 16bit 符号なし 整数型 (数値表示・設定・波形表示) 16bit 符号あり 整数型 (数値表示・設定・波形表示) 32bit 符号なし 整数型 (数値表示・設定・波形表示) 32bit 符号あり 整数型 (数値表示・設定・波形表示) 32bit IEEE754 浮動小数点 (数値表示・設定・波形表示)		
CPU 使用リソース	SCIx RX、SCIx TX、DTC		

表 4-2 通信ライブラリ（RX マイコン用）（2/3）

対応 CPU	RX66T	RX72T	RX72M
ファイル	ICS2_RX66T.lib ICS2_RX66T.h	ICS2_RX72T.lib ICS2_RX72T.h	ICS2_RX72M.lib ICS2_RX72M.h
通信レート	0.5 Mbps ～ 7.5 Mbps		
ポート	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
サポート変数タイプ	8bit 符号なし 整数型（数値表示・設定・波形表示） 8bit 符号あり 整数型（数値表示・設定・波形表示） 16bit 符号なし 整数型（数値表示・設定・波形表示） 16bit 符号あり 整数型（数値表示・設定・波形表示） 32bit 符号なし 整数型（数値表示・設定・波形表示） 32bit 符号あり 整数型（数値表示・設定・波形表示） 32bit IEEE754 浮動小数点（数値表示・設定・波形表示）		
CPU 使用リソース	SCIx RX、SCIx TX、DTC		

表 4-3 通信ライブラリ（RX マイコン用）（3/3）

対応 CPU	RX13T	RX26T
ファイル	ICS2_RX13T.lib ICS2_RX13T.h	ICS2_RX26T.lib ICS2_RX26T.h
通信レート	0.5 Mbps ～ 4 Mbps	0.5 Mbps ～ 7.5 Mbps
ポート	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
サポート変数タイプ	8bit 符号なし 整数型（数値表示・設定・波形表示） 8bit 符号あり 整数型（数値表示・設定・波形表示） 16bit 符号なし 整数型（数値表示・設定・波形表示） 16bit 符号あり 整数型（数値表示・設定・波形表示） 32bit 符号なし 整数型（数値表示・設定・波形表示） 32bit 符号あり 整数型（数値表示・設定・波形表示） 32bit IEEE754 浮動小数点（数値表示・設定・波形表示）	
CPU 使用リソース	SCIx RX、SCIx TX、DTC	

表 4-4 通信ライブラリ（RL マイコン用）（1/2）

対応 CPU	RL78/G14	RL78/G1F
ファイル	ics2_RL78G14.Lib ics2_RL78G14.h	ics2_RL78G1F.Lib ics2_RL78G1F.h
通信レート	0.5 Mbps ～ 5.33 Mbps	
ポート	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
サポート変換タイプ	8bit 符号なし 整数型（数値表示・設定・波形表示） 8bit 符号あり 整数型（数値表示・設定・波形表示） 16bit 符号なし 整数型（数値表示・設定・波形表示） 16bit 符号あり 整数型（数値表示・設定・波形表示） 32bit 符号なし 整数型（数値表示・設定） 32bit 符号あり 整数型（数値表示・設定） 8bit BOOL 型（数値表示・設定） 8bit LOGIC 型（数値表示・設定）	
CPU 使用 リソース	SCIx RX、SCIx TX、DTC	

表 4-5 通信ライブラリ (RL マイコン用) (2/2)

対応 CPU	RL78/G24
ファイル	ics2_RL78G24.Lib ics2_RL78G24.h
通信レート	0.5 Mbps ~ 8 Mbps
ポート	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
サポート変換 タイプ	8bit 符号なし 整数型 (数値表示・設定・波形表示) 8bit 符号あり 整数型 (数値表示・設定・波形表示) 16bit 符号なし 整数型 (数値表示・設定・波形表示) 16bit 符号あり 整数型 (数値表示・設定・波形表示) 32bit 符号なし 整数型 (数値表示・設定) 32bit 符号あり 整数型 (数値表示・設定) 8bit BOOL 型 (数値表示・設定) 8bit LOGIC 型 (数値表示・設定)
CPU 使用リ ソース	SCIx RX、SCIx TX、DTC

表 4-6 通信ライブラリ (RA マイコン用) (1/3)

対応 CPU	RA6T1	RA6T2	RA6T3
ファイル	ICS2_RA6T1.o ICS2_RA6T1.h	ICS2_RA6T2.o ICS2_RA6T2.h	ICS2_RA6T3.o ICS2_RA6T3.h
通信レート	0.5 Mbps ~ 15.0 Mbps	0.5 Mbps ~ 20.0 Mbps	0.5 Mbps ~ 16.67 Mbps
サポートポート	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
サポート変数タイプ	8bit 符号なし 整数型 (数値表示・設定・波形表示) 8bit 符号あり 整数型 (数値表示・設定・波形表示) 16bit 符号なし 整数型 (数値表示・設定・波形表示) 16bit 符号あり 整数型 (数値表示・設定・波形表示) 32bit 符号なし 整数型 (数値表示・設定・波形表示) 32bit 符号あり 整数型 (数値表示・設定・波形表示) 32bit IEEE754 浮動小数点 (数値表示・設定・波形表示)		
CPU 使用リソース	SCIx RX、SCIx TX	SCIx RX、SCIx TX	SCIx RX、SCIx TX

表 4-7 通信ライブラリ (RA マイコン用) (2/3)

対応 CPU	RA4T1	RA8T1	RA8T2
ファイル	ICS2_RA4T1.o ICS2_RA4T1.h	ICS2_RA8T1.o ICS2_RA8T1.h	ICS2_RA8T2.o ICS2_RA8T2.h
通信レート	0.5 Mbps ~ 16.67 Mbps	0.5 Mbps ~ 20.0 Mbps	0.5 Mbps ~ 20.0 Mbps
サポートポート	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	SCI1 TXD1:P400 RXD1:P401
サポート変数タイプ	8bit 符号なし 整数型 (数値表示・設定・波形表示) 8bit 符号あり 整数型 (数値表示・設定・波形表示) 16bit 符号なし 整数型 (数値表示・設定・波形表示) 16bit 符号あり 整数型 (数値表示・設定・波形表示) 32bit 符号なし 整数型 (数値表示・設定・波形表示) 32bit 符号あり 整数型 (数値表示・設定・波形表示) 32bit IEEE754 浮動小数点 (数値表示・設定・波形表示)		
CPU 使用リソース	SCIx RX、SCIx TX	SCIx RX、SCIx TX	SCIx RX、SCIx TX

表 4-8 通信ライブラリ（RA マイコン用）（3/3）

対応 CPU	RA2T1
ファイル	ICS2_RA2T1.o ICS2_RA2T1.h
通信レート	0.5 Mbps ~ 5.0 Mbps
サポートポート	SCI0 TXD0:P101 RXD0:P100 SCI0 TXD0:P101 RXD0:P104 SCI0 TXD0:P206 RXD0:P100
サポート変数タイプ	8bit 符号なし 整数型（数値表示・設定・波形表示） 8bit 符号あり 整数型（数値表示・設定・波形表示） 16bit 符号なし 整数型（数値表示・設定・波形表示） 16bit 符号あり 整数型（数値表示・設定・波形表示） 32bit 符号なし 整数型（数値表示・設定・波形表示） 32bit 符号あり 整数型（数値表示・設定・波形表示）
CPU 使用リソース	SCIx RX、SCIx TX

4.2 ビルトイン型通信ライブラリ

RA6T2、RA6T3、RA4T1、RA8T1、RA8T2 は、ビルトイン型通信ライブラリをサポートしています。ビルトイン型通信ライブラリを使用すると、市販の USB シリアル変換モジュール（絶縁タイプ）を使用することができます。

ビルトイン型通信ライブラリは、Renesas Motor Workbench パッケージに同梱されています（「communication library」フォルダ内の「RA6T2」、「RA6T3」、「RA4T1」、「RA8T1」、「RA8T2」フォルダ）。

表 4-9 ビルトイン型通信ライブラリ（RA マイコン用）（1/2）

対応 CPU	RA6T2	RA6T3	RA4T1
ファイル	ICS2_RA6T2_Built_in.o	ICS2_RA6T3_Built_in.o	ICS2_RA4T1_Built_in.o
通信レート	—	—	—
サポートポート	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
サポート変数タイプ	8bit 符号なし 整数型（数値表示・設定・波形表示） 8bit 符号あり 整数型（数値表示・設定・波形表示） 16bit 符号なし 整数型（数値表示・設定・波形表示） 16bit 符号あり 整数型（数値表示・設定・波形表示） 32bit 符号なし 整数型（数値表示・設定・波形表示） 32bit 符号あり 整数型（数値表示・設定・波形表示） 32bit IEEE754 浮動小数点（数値表示・設定・波形表示）		
CPU 使用リソース	SCIx RX、SCIx TX		

表 4-10 ビルトイン型通信ライブラリ (RA マイコン用) (2/2)

対応 CPU	RA8T1	RA8T2
ファイル	ICS2_RA8T1_Built_in.o	ICS2_RA8T2_Built_in.o
通信レート	—	—
サポートポート	SCI1 TXD1:P213 RXD1:P212 SCI2 TXD2:PA03 RXD2:PA02 SCI3 TXD3:P409 RXD3:P408 SCI4 TXD4:P714 RXD4:P715	SCI9 TXD9:PA14 RXD9:PA12
サポート変数タイプ	8bit 符号なし 整数型 (数値表示・設定・波形表示) 8bit 符号あり 整数型 (数値表示・設定・波形表示) 16bit 符号なし 整数型 (数値表示・設定・波形表示) 16bit 符号あり 整数型 (数値表示・設定・波形表示) 32bit 符号なし 整数型 (数値表示・設定・波形表示) 32bit 符号あり 整数型 (数値表示・設定・波形表示) 32bit IEEE754 浮動小数点 (数値表示・設定・波形表示)	
CPU 使用リソース	SCIx RX、SCIx TX	SCIx RX、SCIx TX

4.3 ユーザプログラムの設定方法

4.3.1 DTC

Renesas Motor Workbench は DTC（標準アドレスモード）を使用します（RA マイコンは除く）。そのため、ユーザプログラム内に DTC テーブルを定義する必要があります。

プログラム内（後述の初期化関数を呼ぶファイル内）に DTC テーブルの領域を確保します。

- RX マイコンの場合
DTC テーブルのセクションを下位 12bit が 0 になる様に RAM 上のアドレスに割り当てます。
- RL マイコンの場合
DTC テーブルのセクションを下位 8bit が 0 になる様にアドレスを割り当てます。

記述例は、「4.3.4 ライブラリ関数の使用方法」を参照してください。

E2 等のエミュレータを使用する場合は、ユーザプログラムの RAM 領域と DTC テーブルの領域が重ならないようにして下さい。

4.3.2 割り込み

ユーザプログラムの割り込みベクタへ、記述例のように関数（ics_int_sci_eri()、ics_int_sci_rxi()）を記述します（RA マイコンは除く）。

RX マイコンにおいてルネサス製標準コンパイラで自動生成されるプロジェクトを使用する場合は、intprg.c ファイルの中に記述します。

```
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(); }
```

図 4-1 通信用割り込み関数記述例

4.3.3 ライブラリ関数の仕様

通信ライブラリでは、以下の2つのライブラリ関数を提供します。

- void ics2_init(void* addr, char port, char level, char speed, char mode)
- void ics2_watchpoint(void)

各関数の仕様を以下に示します。

表 4-11 ライブラリ関数 (ics2_init) (1/2)

関数名	void ics2_init(void* addr, char port, char level, char speed, char mode)	
戻り値	void	なし
引数	void* addr	使用する DTC テーブルの先頭アドレス : ユーザは本関数を呼び出す前に DTC テーブルを確保する必要があります。 ※RA マイコンは、本引数の設定はありません
	char port	SCI のポート番号や、SCI の使用するピンを設定します。
	char level	SCI の割り込みレベルを指定 : 最小 2 msec 間隔で 10 μsec 程度の割り込み処理が発生する可能性があるため、システムとして適切な割り込みレベルを設定します。 ※RA マイコンは、本引数の設定はありません
	char speed	通信レートの設定 : 以下の計算式を用いて、使用する通信レートとなる値を設定します。 <ul style="list-style-type: none"> • 通信レート = $CLOCK / (x \times (speed + 1))$ [Mbps] 通信レート(RA2T1) = $CLOCK / (x \times (speed + 1) \times (256/240))$ [Mbps] — CLOCK : SCI に供給されるクロックソースの周波数[MHz] RX : PCLK、RA : SCISPICK or PCLK ※詳細は使用する MCU のハードウェアマニュアルを参照してください。 — x : 各 MCU による固定値 RX、RA6T1 : x = 8 RL78 : x = 2 RA6T2、RA4T1、RA6T3、RA8T1、RA8T2、RA2T1 : x = 6 例 : RX23T で 1.0Mbps を設定する場合 CLOCK = PCLKB = 40 [MHz], x = 8 なので 1.0Mbps = 40MHz / (8 * (speed + 1)) から、speed : 4

表 4-12 ライブラリ関数 (ics2_init) (2/2)

引数	char mode	<ul style="list-style-type: none"> mode 1 (32 ビット 8 チャンネル 2 回転送モード動作) ics2_watchpoint()関数が呼ばれると、波形表示に関しては、一度に指定された 8ch 分の 8 ビット、16 ビット、32 ビットデータを取り込み、4ch 分のデータを転送します。次に ics2_watchpoint()関数が呼ばれた時、データを取り込まず、未転送の残り 4ch 分のデータを転送します。 つまり、32 ビット 8 チャンネルモードの場合には、2 回の ics2_watchpoint()関数により、1 回の 8ch 分の転送が行われます。 mode 2 (32 ビット 4 チャンネル 1 回転送モード動作) ics2_watchpoint()関数が呼ばれると、波形表示に関しては、毎回、指定された 4ch 分の 8 ビット、16 ビット、32 ビットデータを取り込みます。そして、その 4ch 分のデータを転送します。 つまり、32 ビット 4 チャンネルモードの場合には、1 回の ics2_watchpoint()関数呼び出しにより、1 回の 4ch 分の転送が行われます。5 チャンネル以上の波形表示の機能はありません。 mode 3 (32 ビット 12 チャンネル 3 回転送モード動作) ics2_watchpoint()関数が呼ばれると、波形表示に関しては、一度に指定された 12ch 分の 8 ビット、16 ビット、32 ビットデータを取り込み、4ch 分のデータを転送します。次に ics2_watchpoint()関数が呼ばれた時、データを取り込まず、未転送のデータの内の残り 4ch 分のデータを転送します。さらに次に ics2_watchpoint()関数が呼ばれた時に、最後の 4ch 分のデータを転送します。つまり、32 ビット 12 チャンネルモードの場合には、3 回の ics2_watchpoint()関数により、1 回の 12ch 分の転送が行われます。
機能	初期化処理	

表 4-13 ライブラリ関数 (ics2_watchpoint)

関数名	void ics2_watchpoint(void)	
戻り値	void	なし
引数	void	なし
機能	データ転送 : 通信レートを BR [Mbps]とした時、 $70 + (180/BR)$ [μsec] が最小のサンプリング周期になりますので、この値以上の間隔でデータ転送関数を呼び出します。 例えば、通信レートが 1.0 Mbps の場合 250 μsec 以上の間隔が必要です。 なお、ツール用通信ボード MC-COM を使用する場合は、 $10 + (180/BR)$ [μsec] が最小のサンプリング周期になります。	

4.3.4 ライブラリ関数の使用方法

(1) RX マイコンの場合

(a) 初期化関数 ics2_init() の呼び出し

記述例を参考に、初期化関数 ics2_init() をプログラムの初期化処理部に追記します。各引数の設定は、以下のとおりです。

第 1 引数：DTC テーブルの先頭アドレス（下位 12bit は 0 に）を設定

第 2 引数：ics_xxxx.h から使用するポートを選択（表 4-1 を参照）

第 3 引数：ツールの割り込みレベルを設定

第 4 引数：通信レートを設定

第 5 引数：転送モードを設定

```
#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);
}
```

図 4-2 初期化関数 ics2_init() の呼び出し（RX23T での記述例）

(b) データ転送関数 ics2_watchpoint() の呼び出し

記述例のように、最小サンプリング周期より長い周期で、データ転送関数 ics2_watchpoint() を呼び出します。

```
void int_TM100u(void)          /* 100 μsec 間隔 */
{
    if (3 <= g_u1_cnt_decimation) /* decimation of ICS call */
    {
        g_u1_cnt_decimation = 0;
        ics2_watchpoint();      /* data transfer */
    }
    g_u1_cnt_decimation++;
}
```

図 4-3 データ転送関数 ics2_watchpoint() の呼び出し（RX23T での記述例）

(2) RL マイコンの場合

(a) 初期化関数 ics2_init() の呼び出し

記述例を参考に、初期化関数 ics2_init() をプログラムの初期化処理部に追記します。各引数の設定は、以下のとおりです。

第 1 引数：DTC テーブルの先頭アドレス（下位 12bit は 0 に）を設定

第 2 引数：ics_xxxx.h から使用するポートを選択（表 4-4 を参照）

第 3 引数：ツールの割り込みレベルを設定

第 4 引数：通信レートを設定

第 5 引数：転送モードを設定

```
#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);
}
```

図 4-4 初期化関数 ics2_init() の呼び出し（RL78/G1F での記述例）

(b) データ転送関数 ics2_watchpoint() の呼び出し

記述例のように、最小サンプリング周期より長い周期で、データ転送関数 ics2_watchpoint() を呼び出します。

```
__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++;
}
```

図 4-5 データ転送関数 ics2_watchpoint() の呼び出し（RL78/G1F での記述例）

(3) RA マイコンの場合

(a) 初期化関数 ics2_init() の呼び出し

記述例を参考に、初期化関数 ics2_init() をプログラムの初期化処理部に追記します。各引数の設定は、以下のとおりです。

第 1 引数 : ics_xxxx.h から使用するポートを選択（表 4-6 を参照）

第 2 引数 : 通信レートを設定

第 3 引数 : 転送モードを設定

```
#include "ics2_RA6T1.h"

void main(void)
{
    ics2_init(port, speed, mode);
}
```

図 4-6 初期化関数 ics2_init() の呼び出し（RA6T1 での記述例）

(b) データ転送関数 ics2_watchpoint() の呼び出し

記述例のように、最小サンプリング周期より長い周期で、データ転送関数 ics2_watchpoint() を呼び出します。

```
void int_TM100u(void) /* 100 μsec 間隔 */
{
    if (3 <= g_u1_cnt_decimation) /* decimation of ICS call */
    {
        g_u1_cnt_decimation = 0;
        ics2_watchpoint(); /* data transfer */
    }
    g_u1_cnt_decimation++;
}
```

図 4-7 データ転送関数 ics2_watchpoint() の呼び出し（RA6T1 での記述例）

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	May.30.23	—	クイックスタートガイドとして初版発行
1.01	Oct.27.23	—	「2.5.3 新しい認証ファイルの入手」を修正 認証ファイルの入手先 URL を Renesas Motor Workbench の web サイトに変更
1.02	Jan.30.24	—	RA8T1 を追加
1.10	Feb.07.25	—	「1.1 Renesas Motor Workbench とは」を修正 Servo を追加 「1.2.2 ホスト PC のシステム要件」を修正 ディスク容量、OS を改訂 「1.3 仕様」を修正 対応 RL78 マイコンを修正 「2.2 パッケージ内容」を修正 RL78 マイコン通信ライブラリについて修正 「2.5 認証」を削除 「2.7.3 ツール用通信ボードを使用する場合」を修正 MCU ごとの最大通信速度を定義 「4.1 MCU 別通信ライブラリ」を修正 通信レートを改訂 RL78 マイコン通信ライブラリについて修正 「4.3.3 ライブラリ関数の仕様」を修正 ics2_init の char mode の引数説明を追加
1.11	Jun.30.25	6	1.4 サポート情報を更新
1.12	Nov.10.25	—	RA8T2、RA2T1 を追加
1.20	Dec.15.25	11 33	2.4.1 ログボタン表記を追加 2.8.2(5) ログコンソールの表示切り替え機能を追加

参考：旧版ユーザーズマニュアル（R21UZ0004）改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Apr.05.17	—	1.0 版リリース
2.00	Nov.27.18	—	2.0 版リリース
2.01	May.16.19	—	RX72T を追加
2.02	Nov.29.19	—	RX13T を追加
2.03	Oct.28.20	—	RA6T1 を追加
3.00	Dec.09.21	—	3.0 版リリース
—	May.30.23	—	クイックスタートガイドに変更

備考

本書は、モータ制御開発支援ツール Renesas Motor Workbench ユーザーズマニュアル（R21UZ0004）を「クイックスタートガイド」として再編集したものです。

なお、従来の機能説明書（アプリケーションノート）2冊

- Renesas Motor Workbench 機能説明書（R20AN0527）
- Renesas Motor Workbench Tuner 機能説明書（R20AN0528）

を再編集して、

- Renesas Motor Workbench ユーザーズマニュアル（R21UZ0004）

としました。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限られません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものとしします。
13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。