RENESAS

User's Manual

改訂新版

モータ制御開発支援ツール
Renesas Motor Workbench 3.3.0

ユーザーズマニュアル

RA/RX/RL78 ファミリ

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、
予告なしに、本資料に記載した製品または仕様を変更することがあります。
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

ルネサス エレクトロニクス
www.renesas.com

Rev 4.20 Dec 2025

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、ノイズ幅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、これに限りません。）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いたします。
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

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。

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/

目次

モータ制御開発支援ツール Renesas Motor Workbench 3.3.0

RA/RX/RL78 ファミリ

1. Renesas Motor Workbench の概要

本資料は、モータ制御開発支援ツール Renesas Motor Workbench（以下、RMW と略）の各機能と操作方法を説明したものです。安全にご使用いただくために、本資料をよくお読みください。

なお、本資料記載の機能説明や操作説明は、ターゲットが正しく接続されていることを前提としています。オフラインモードあるいは接続に何らかの問題がある場合、一部の機能は使用できないか、または制限されることがあります。

1.1 RMW の機能

RMW は、モータ制御プログラムのデバッグ、解析、チューニングを行うための開発支援ツールです。
RMW は、以下のような機能を提供します。

- Analyzer ツール
  - マイコン内部の変数の動的な読み書き
  - 変数の値の変化をリアルタイムに波形で表示することが可能
  - トリガ設定やズームの解析機能
- Easy ツール
  - 直感的な操作でモータの速度制御や位置制御を簡単に実行
  - メータやグラフで駆動状況を一目で確認
- Tuner ツール
  - ベクトル制御に必要なパラメータを自動で抽出、調整
  - 手動調整機能による微調整も可能
  - 調整結果の出力機能（ヘッダファイル、PDF）
- Servo ツール
  - Inertia Estimation
    モータのロータやモータ軸に接続されているシャフト・負荷イナーシャを、実際にモータを動かして推定する機能
  - Servo Tuning
    位置制御方式や制御パラメータなどサーボ動作に関する設定を行う機能
  - Return to Origin
    原点復帰方法、復帰速度などを設定する機能
  - Point to Point
    1 軸に対して、PTP 動作(Point to Point)を行う機能

1.2 小数点の表示と入力

RMW で数値を表示する画面、及び数値を入力する箇所において、小数を扱う場合は小数点にピリオドを使用しています。OS の言語設定に依らず固定の設定のため、小数点にカンマを使用する言語の場合でも、RMW の数値表示及び数値入力にはピリオドを使用しますのでご注意ください。

1.3 関連資料

RMW のインストール方法、利用のための環境設定、準備などについては、以下のクイックスタートガイドを参照してください。

- Renesas Motor Workbench クイックスタートガイド（R21QS0011）

RMW が持つ機能の一部（ターゲットマイコンとの通信機能等）を DLL の形で公開しています。ユーザが PC 上でオリジナルのアプリを開発する場合は、下記を参照してください。

- Renesas Motor Workbench 通信用 DLL 機能説明書（R20AN0683）

2. Main Window

2.1 概要

Main Window は RMW を起動すると表示される画面です。通信ボードとの接続、プロジェクト情報の読み込み、各ツールの起動など、RMW 全般の操作を行います。

2.2 ウィンドウ構成

Main Window の構成を以下に示します。

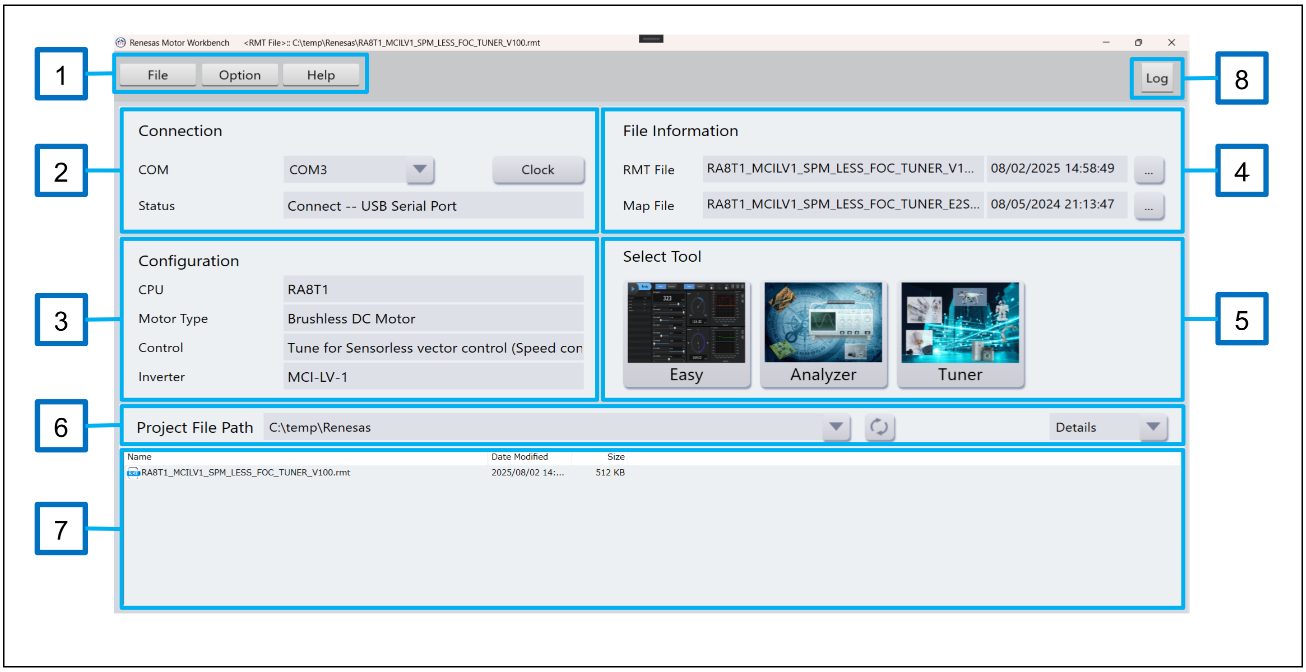


図 2-1 Main Window の画面構成

表 2-1 Main Window 各部の名称（No.は上図と対応）

No.	名称	概要	参照章
1	Menu バー	メニューバー（File、Option、Help）	2.3.1
2	Connection	ターゲットシステムとの通信設定	2.3.2
3	Configuration	ターゲットシステムの情報（**ルネサス提供の実行ファイルのみ表示**）	2.3.3
4	File Information	現在有効になっている RMT ファイルと Map ファイルの表示	2.3.4
5	Select Tool	利用できるツールのアイコン表示	2.3.5
6	Project File Path	過去に読み込んだことのある RMT ファイルへのパスをプルダウンメニューで表示（最大 10 件） 選択したフォルダにある RMT ファイルを下の File List に表示（この時点では、まだ RMT ファイルの情報は読み込まれていません。）	2.3.6
7	File List	Project File Path で示されたフォルダにある RMT ファイルのリスト表示 ダブルクリックで RMT ファイルを読み込み RMT ファイルを直接ドラッグ＆ドロップでここに追加することも可能	
8	Log 表示	操作ログの表示	2.3.7

2.3 Main Window 各部の説明

2.3.1 メニューバー

2.3.1.1 File メニュー

File メニューの機能は以下の通りです。

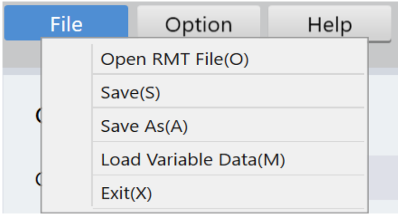


図 2-2 File メニュー

表 2-2 File メニューの機能

名称	説明
Open RMT File	RMT ファイル注 を読み込みます。
Save	読み込んだ変数情報と RMW の設定情報を、RMT ファイルへ上書き保存します。
Save As	読み込んだ変数情報と RMW の設定情報を、RMT ファイルへ名前を付けて保存します。
Load Variable Data	ユーザプログラムで保持するグローバル変数のリスト（変数情報）を Map ファイルから読み込みます。
Exit	RMW を終了します（終了画面が表示されます）。

注 RMW のプロジェクトファイル（変数情報と RMW の各ツールで設定した情報の両方を含む）

2.3.1.2 Option メニュー

Option の機能は以下の通りです。

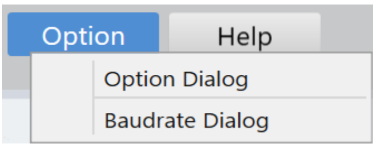


図 2-3 Option メニュー

表 2-3 Option メニューの機能

名称	説明
Option Dialog	RMW の各種設定を行います。Option Dialog 画面を表示します。
Baudrate Dialog	ビルトイン型通信ライブラリ使用時のボーレート設定を行います。

2.3.1.3 Help メニュー

Help の機能は以下の通りです。

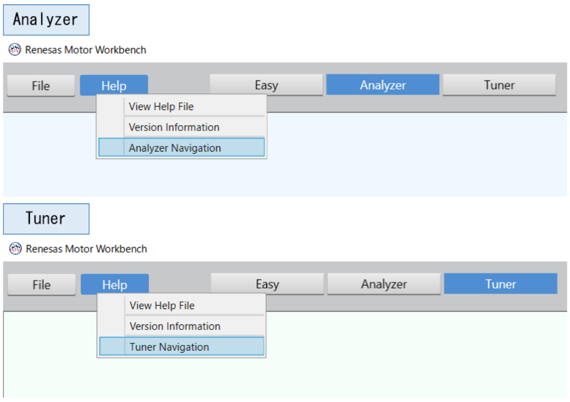


図 2-4 Help メニュー

表 2-4 Help メニューの機能

名称	説明
View Help File	RMW のヘルプ情報を表示します。
Version Information	RMW のバージョン情報を確認できます。
Select Navigation	Main Window 表示中にメニューから選択可能です 。 RMT ファイルまたは Map ファイルを読み込むためのダイアログを表示します。Select Navigation は、デフォルトで RMT 起動時に表示されます（表示させないことも可能）。
Analyzer Navigation	Analyzer ツール表示中にメニューから選択可能です。 Analyzer 内機能の操作手順を確認するダイアログを表示します。Analyzer Navigation は、デフォルトで Analyzer 起動時に表示されます（表示させないことも可能）。
Tuner Navigation	Tuner ツール表示中にメニューから選択可能です。 Tuner 機能の操作手順を確認するダイアログを表示します。Tuner Navigation は、デフォルトで Tuner 起動時に表示されます（表示させないことも可能）。

2.3.2 Connection 画面

Connection 画面の機能は、以下の通りです。

xz

図 2-5　Connection 画面

表 2-5　Connection の機能

名称	説明
COM	接続可能な COM 番号、または Offline Mode がプルダウンで表示されます。 COM 番号を選択すると COM 接続を試みます。
Status	COM 番号を選択して COM 接続を試みた時に、接続状態を表示します。 正常に接続すると「Connect – USB Serial Port」と表示します。
Clock ボタン	ボタンのクリックで通信する周波数の設定ができます。 Motor RSSK の場合は、8 MHz 固定となります。

2.3.3 Configuration

Configuration の機能 注 は以下の通りです。Configuration には、制御プログラムで定義されている内容が表示されます。

注 ルネサスが提供するサンプルコードを MCU にダウンロードした場合に表示します。

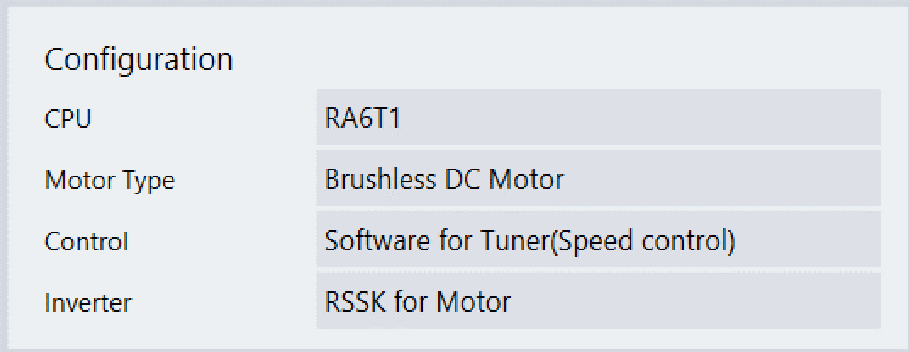


図 2-6 Configuration 画面

表 2-6 Configuration の機能

名称	説明
CPU	接続しているターゲットの CPU 名を表示します。
Motor Type	制御プログラムが対象としているモータの種類を表示します。
Control	制御プログラムの制御方法を表示します。
Inverter	制御プログラムが対象としているインバータボード情報を表示します。

2.3.4 File Information

File Information の機能は以下の通りです。ファイル名にマウスを合わせるとファイル情報を表示、右のボタンをクリックすると読み込みができます。

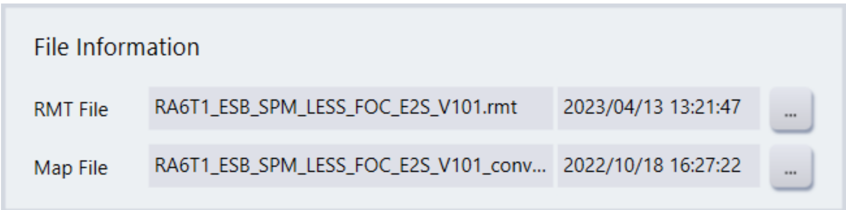


図 2-7 File Information 画面

表 2-7 File Information の機能

名称	説明
RMT File	読み込み中の RMT ファイル名とそのファイルの更新日時を表示します。
Map File	読み込み中の Map ファイル名とそのファイルの更新日時を表示します。

2.3.5 Select Tool

Select Tool は、対象のプログラムで利用可能なツールのアイコンを表示します。表示されたアイコンをクリックするとそのツールが起動します。

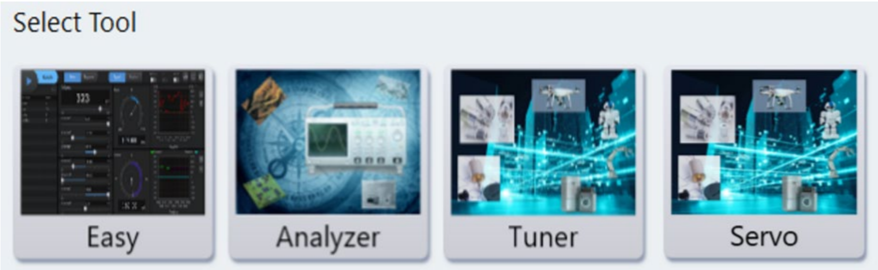


図 2-8　Select Tool 画面

表 2-8　Select Tool の機能

名称	説明
Select Tool （アイコン表示）	プログラムに応じて、利用可能なツール（Easy、Analyzer、Tuner、Servo）のアイコンを表示します。 クリックすると各ツールが起動し、ウィンドウが切り替わります。

利用可能なツールは、モータ制御プログラムに定義した変数「g_u2_conf_tool」の値によって決まります。COM 接続時に変数値を読み込み、その値によって利用可能なツールを表示します。
以下に、ツールの利用可能条件を示します。

表 2-9　利用可能ツールの条件

名称	利用可能条件
Easy	常時利用可能です。
Analyzer	常時利用可能です。
Tuner	変数 g_u2_conf_tool の値が 0x600 のとき利用可能です。
Servo	変数 g_u2_conf_tool の値が 0xA00 のとき利用可能です。

2.3.6 Project File Path と File List

Project File Path と File List は、RMW のプロジェクトフォルダを管理します。

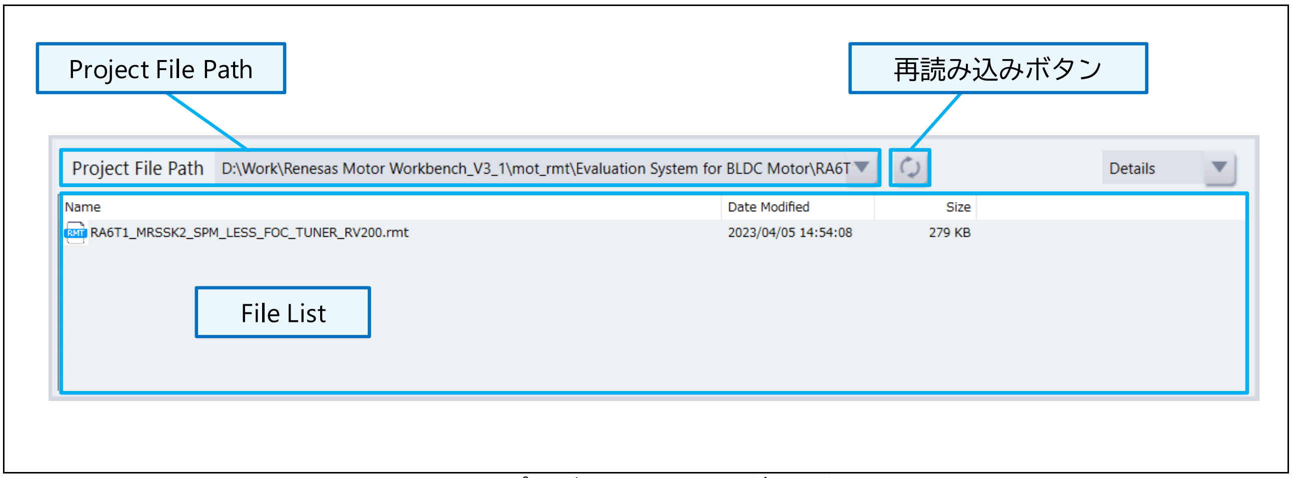


図 2-9　Project File Path、File List 画面とプロジェクトフォルダの関係

表 2-10　Project File Path と File List の機能

名称	説明
Project File Path	• 過去に読み込んだプロジェクトフォルダ（読み込んだ RMT ファイルのあるフォルダ）のパスを最大 10 件まで、プルダウンメニューで表示します。 • 直接パスを入力することはできません。 • 選択したパスのプロジェクトフォルダ内にある RMT ファイルが、下の File List に表示されます（File List に表示した時点では、まだ RMW には読み込まれていません）。 • Project File Path 横の「再読み込み」ボタンで、下の File List（プロジェクトフォルダ内の RMT ファイルのリスト表示）が更新されます。
File List	• Project File Path で指定されたプロジェクトフォルダ内にある RMT ファイルをリスト表示します。 • 表示された RMT ファイルをダブルクリックすると、RMW に読み込まれます。 • Project File Path 指定後に、PC 上のプロジェクトフォルダで RMT ファイルの追加／削除を行った場合は、Project File Path 横の「再読み込み」ボタンをクリックして表示を更新します（自動では反映されません）。 • **ファイル操作上の注意点**： - PC 上の RMT ファイルを直接ドラッグ＆ドロップで File List に追加すると、File List に表示されるとともに、PC 上のプロジェクトフォルダにもその RMT ファイルがコピーされます。 - File List 上の右クリックメニューで RMT ファイルを削除（Delete）した場合、PC 上の RMT ファイルも削除されます。

2.3.7 ログ表示

メニューバーの Log ボタンを押下することで、操作ログの表示タブをウィンドウの右側に表示します。もう一度 Log ボタンを押下すると操作ログの表示を閉じます。

操作ログは Main Window の他、各ツールにおいても表示が可能です。

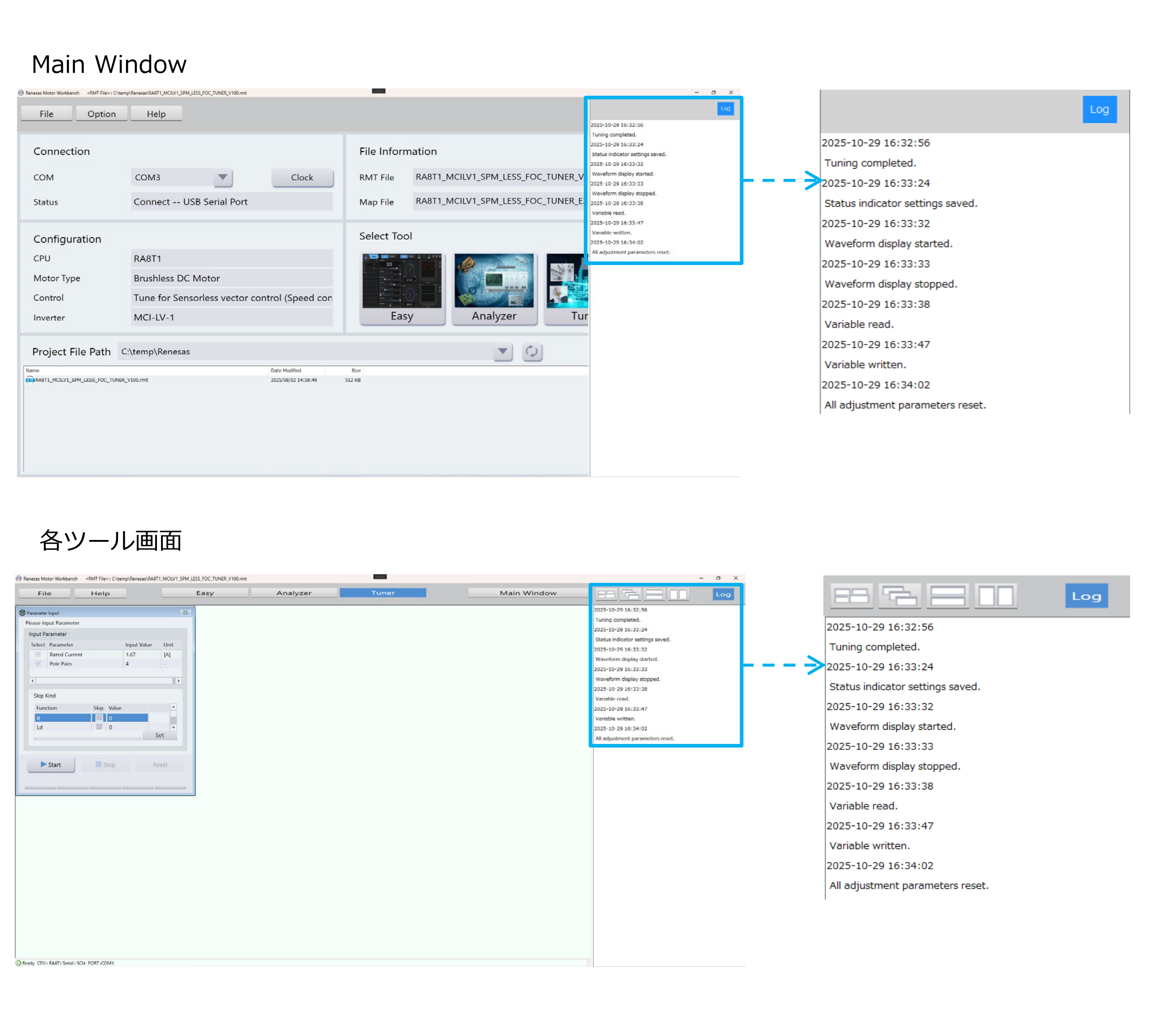


図 2-10 操作ログの表示

2.4 Main Window の操作

2.4.1 RMT ファイルの読み込み

RMW のプロジェクトファイル（RMT ファイル）には、Map ファイルから取得した変数情報と RMW 各ツールの設定情報の両方が保存されています。

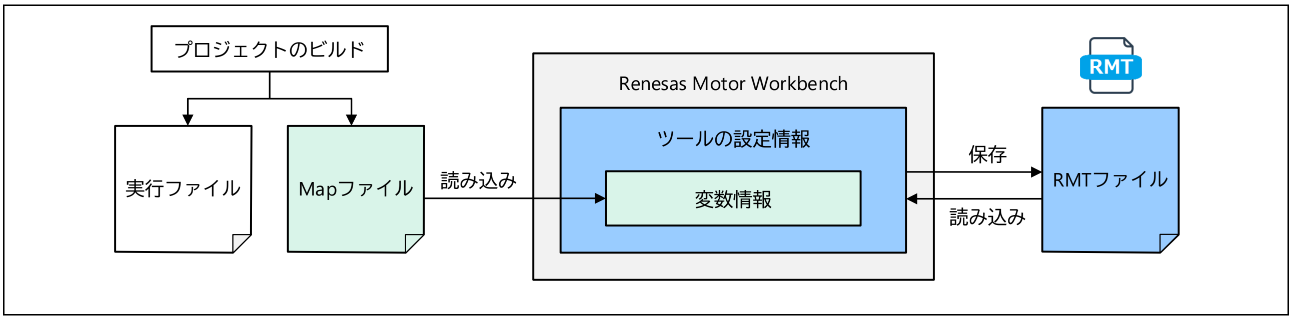


図 2-11　RMT ファイル

プロジェクトファイルの読み込みとは、RMT ファイルを RMW へ読み込むことです。RMT ファイルを読み込むと、それまで読み込んでいた情報はクリアされます。RMT ファイルを読み込む方法を以下に示します。

2.4.1.1 File メニューまたは「Open RMT file」ボタンによる読み込み

File メニューで「Open RMT file」を選択するか、File Information の RMT File 横の「Open RMT file」ボタンをクリックすると、RMT ファイルを開くダイアログが表示されます。ファイルを指定して「開く」ボタンをクリックすると、RMT ファイルが読み込まれます。

RMT ファイルが読み込まれると、読み込んだ RMT ファイルのパスとファイル名が Main Window のタイトルバーに表示され、File Information、Project File Path および File List の情報が更新されます。

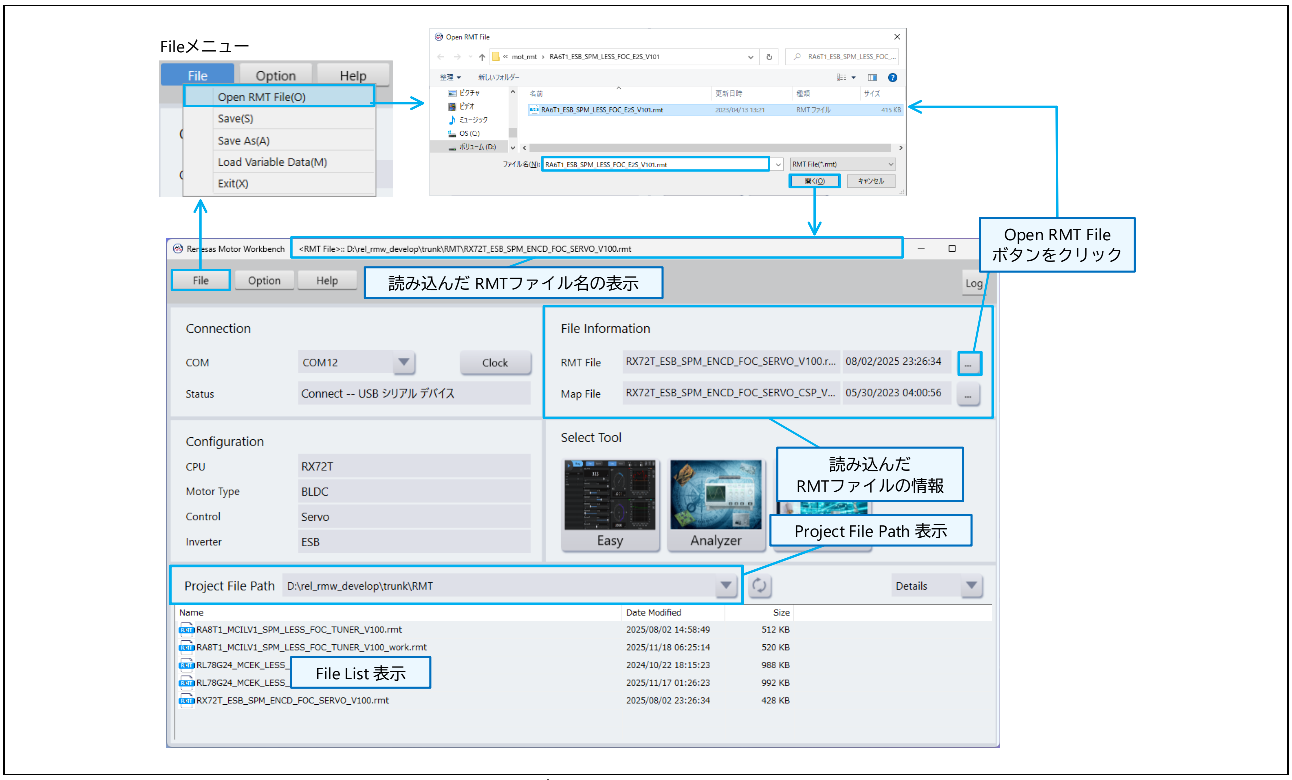


図 2-12 File メニューからの RMT ファイル読み込み

2.4.1.2 Project File Path と File List を使った読み込み

Project File Path のプルダウンメニューには、過去に読み込んだ RMT ファイルのパスが最大 10 件表示されます。パスを選択すると、選択したプロジェクトフォルダ内の RMT ファイルが、File List に表示されます。パスの直接入力指定はできません。

File List に表示された RMT ファイルをダブルクリックすると RMT ファイルが読み込まれます。RMT ファイルが読み込まれると、Main Window のタイトルバーと、File Information の情報が更新されます。

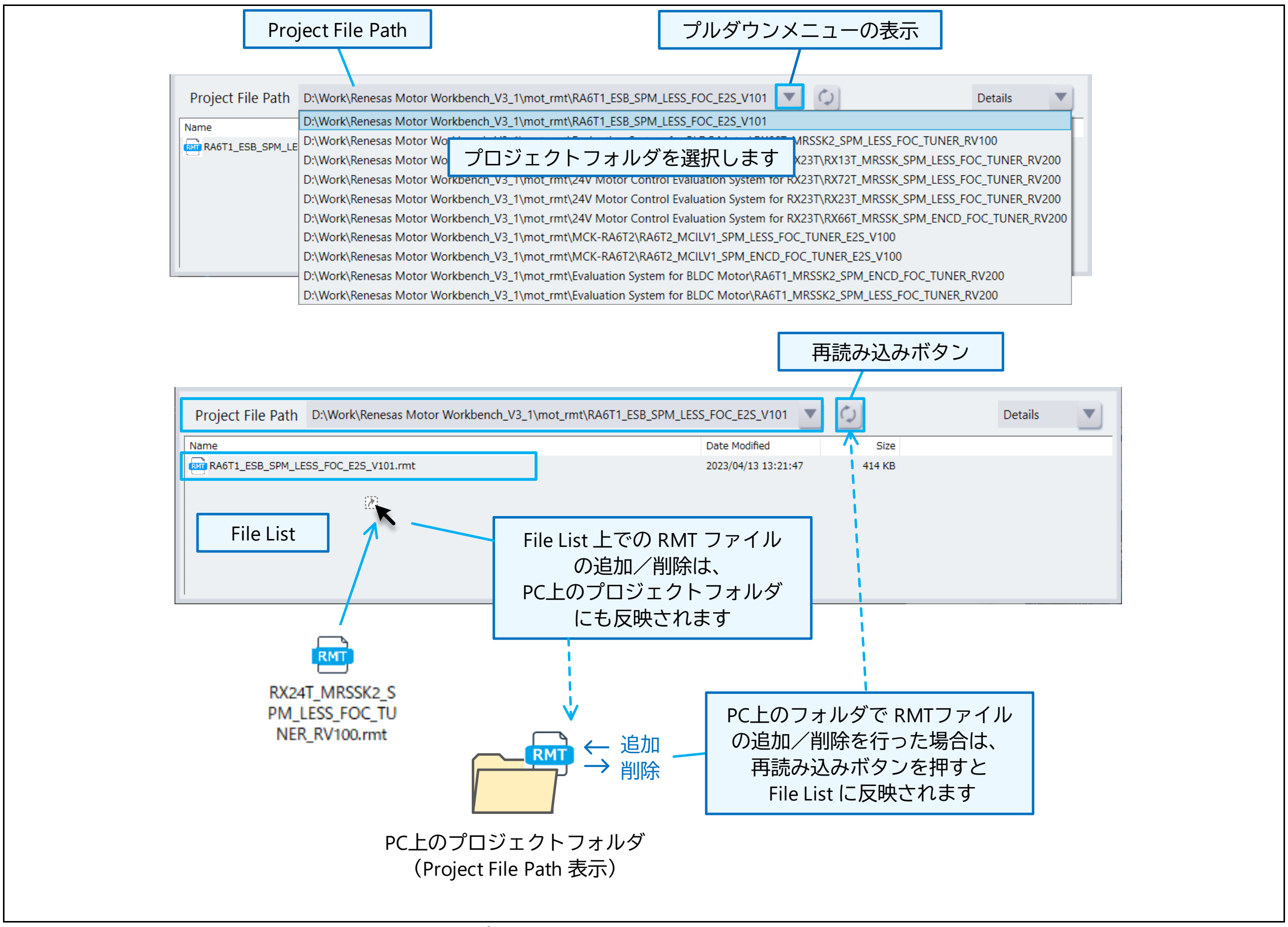


図 2-13 File List からの RMT ファイル読み込み

File List には、PC 上の RMT ファイルをドラッグ＆ドロップで追加することができます。この操作により、**PC 上のプロジェクトフォルダ（Project File Path で指定）にもそのファイルがコピーされます。**

また、File List 上の右クリックメニューで、RMT ファイルを削除（Delete）することもできますが、その場合は、**PC 上のプロジェクトフォルダ内の RMT ファイルも削除されます。**

Project File Path 指定後に、PC 上のプロジェクトフォルダで RMT ファイルの追加／削除を行った場合は、Project File Path 横の「再読み込み」ボタンをクリックして、File List 表示を更新してください（自動では反映されません）。

2.4.1.3 Select Navigation による読み込み

Select Navigation 機能を使って、RMT ファイルを読み込むこともできます。Select Navigation については、2.5 章を参照してください。

2.4.2 RMT ファイルの保存

RMT ファイルの保存の方法は、以下の通りです。

2.4.2.1 File メニューの「Save」または「Save As」で保存

Main Window の File メニューから「Save」を選択すると、読み込み中の RMT ファイルに上書き保存します。また、「Save As」を選択すると「名前を付けて保存」画面が表示されます。RMT ファイル名を指定して保存します。

2.4.2.2 Renesas Motor Workbench 終了時の保存

Renesas Motor Workbench 終了時の終了画面で、RMT ファイルの保存を選択することができます。手順は以下の通りです。

1. File メニューから「Exit」を選択、または Main Window の右上「閉じる」ボタンをクリックすると、終了画面が表示されます。
2. RMT ファイルを保存する場合は、終了画面で「Save」または「Save As」を選択します。

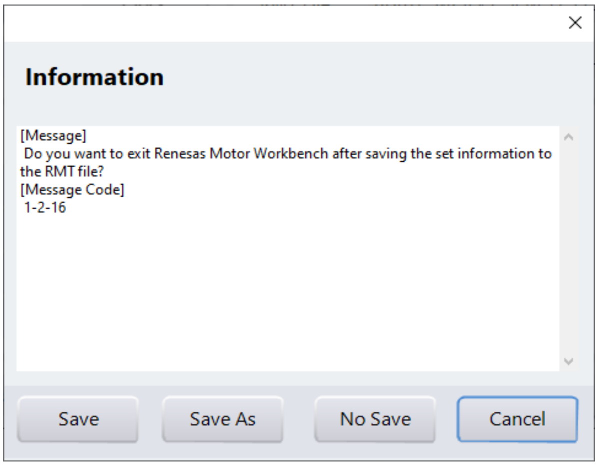


図 2-14 Renesas Motor Workbench の終了画面

表 2-11 終了画面のボタン機能

ボタン	機能
Save	RMW のプロジェクトファイルを読み込み中の RMT ファイルに上書き保存して終了します。
Save As	RMW のプロジェクトファイルをファイル名を指定した RMT ファイルに保存して終了します。
No Save	RMW のプロジェクトファイルを保存しないで終了します。
Cancel	終了処理自体をキャンセルします。

2.4.3 Map ファイル（変数リスト）の読み込み

ユーザプログラムのビルド時に生成した Map ファイルを RMW へ読み込みます。これによりグローバル変数の変数リスト（変数情報）が更新されます。ユーザプログラムを変更して再度ビルドした場合は、Map ファイルを再読み込みする必要があります。

以下に操作手順を示します。

2.4.3.1 File メニュー「Load Variable Data」からの読み込み

1. Main Window の File メニューから「Load Variable Data」を選択すると、Map ファイルの選択画面が表示されます。
2. Map ファイルを選択すると「User Setting Form」画面が表示されます（Option メニュー ＞ Option Dialog の「Load Variable Data」に「Data Type」設定がある場合）。
3. その設定が反映された変数リストが「User Setting Form」画面に表示されます。
4. User Setting Form 画面の情報を確認後、「Set」ボタンをクリックすると、変数情報が読み込まれます。

RMW のこれ以外の機能で、変数リストの「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 機能を実行して、変数情報を設定してください。

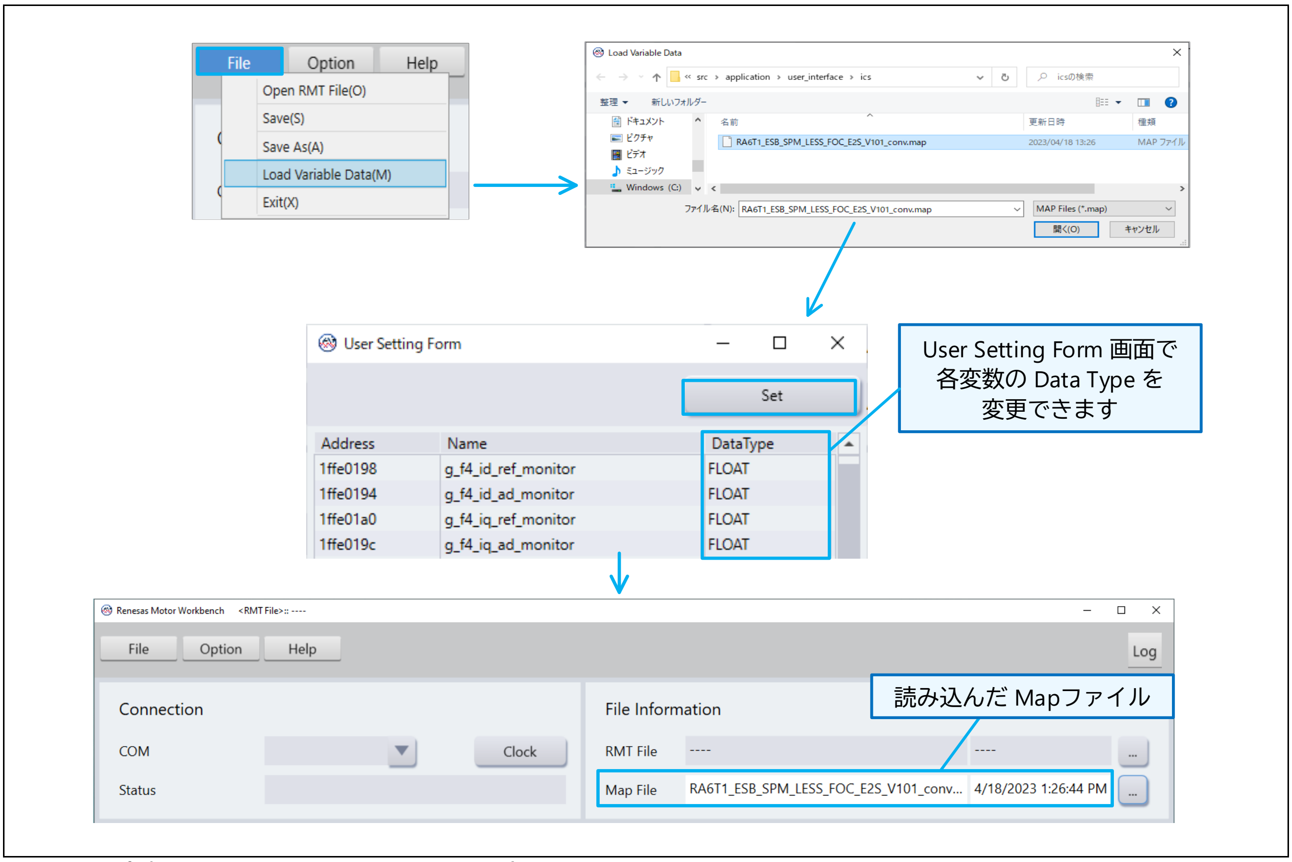


図 2-15 変数リスト（Map ファイル）の読み込み

2.4.3.2 Select Navigation による読み込み

Select Navigation 機能を使って、Map ファイルを読み込むこともできます。Select Navigation については、2.5 章を参照してください。

2.4.4 Option メニューの「Option Dialog」

Option メニューで「Option Dialog」を選択すると、Map ファイル読み込み時の設定と Analyzer 機能の切り替え抑止の設定ができます。

次回起動時も RMW を同じ設定にするには、RMT ファイルの保存が必要です。

表 2-12　Option Dialog メニューの一覧

タブ	機能	説明
Main Window	Load Variable Data	• ユーザプログラムのグローバル変数を Map ファイルからの読み込み（Load Variable Data 機能実行）時の Data Type を指定します。 • Prefix Setting - 変数の Data Type を変更したい場合に変数名の接頭語を指定します。 - 接頭語はカンマ区切りで最大４つまで指定できます。
	Load Variable Meaning Data	• 変数の用途情報を記載した CSV ファイルのパスを指定します 。
Analyzer	Control Window	• 安全のために、モータ駆動中の他ツールへの切り替えを抑止することができます。 - Variable Name：切り替え抑止の確認時に参照する変数名を指定します。 - Value：切り替え抑止の確認時に参照する変数の値を指定します（指定値が抑止する値）

2.4.4.1 Main Window タブの Load Variable Data

各 Data Type（UINT8、INT8 など）に変数の接頭語を指定すると、変数リスト（Map ファイル）読み込み（Load Variable data 機能実行）時に、接頭語が合致した変数はここで指定した Data Type が設定されます。

配列として扱う変数は「Array of <Data Type>」、配列以外は「<Data Type>」で Data Type を指定します。

各 Data Type には、カンマ区切りで最大４つまで指定できます。

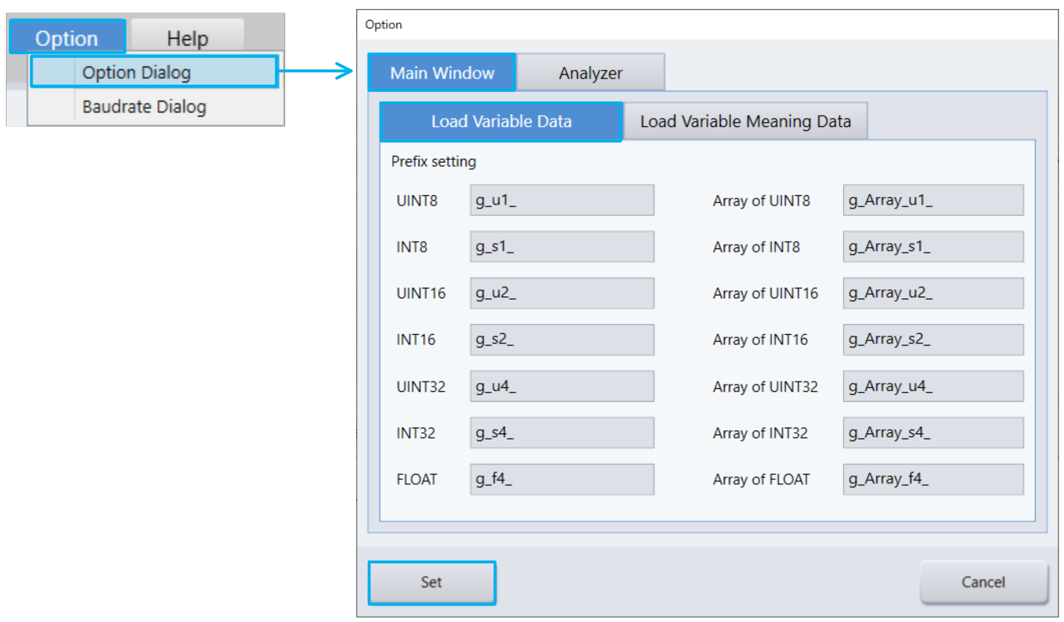


図 2-16　Main Window タブの Load Variable Data 設定

2.4.4.2 Load Variable Meaning Data（変数の用途情報ファイルの設定 ）

変数の用途情報を CSV ファイル形式で読み込み、各変数の用途と単位を表示することができます。
CSV ファイルは、以下の作成例のように、

変数名<Variable name>,用途<Meaning>,単位<Unit>

の形式で作成してください。

読み込んだ CSV ファイルの情報は、2.4.2RMT ファイルの保存に記載した手順で RMT ファイルに保存することができます。CSV ファイルの情報が保存されている RMT ファイルを読み込んだ場合、保存済みの変数の用途情報と単位を表示します。

変数名 用途 単位

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

図 2-17 変数の用途情報ファイル（csv 形式）の作成例

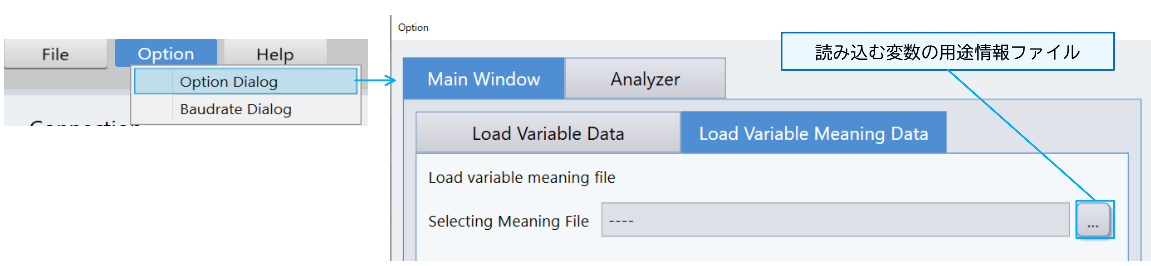


図 2-18 変数の用途情報ファイルの設定例

2.4.4.3　Analyzer タブの Control Window

ユーザプログラムでモータ駆動中に他ツールへの画面遷移を抑止することができます。
「Variable Name」に変数名、「Value」に駆動中（処理中）と判断できる値を指定すると、ツール切り替え時に指定の変数名の値が抑止する値である場合にツール切り替えを抑止します。ツール切り替えが抑止されると、「Cannot changeover TAB of tools while Motor is in.」のメッセージが表示されます。
初期設定は、空欄で設定されています。「Variable Name」が空欄の場合、ツール切り替えの抑止は行われません（この状態でも RMW は操作可能です）。

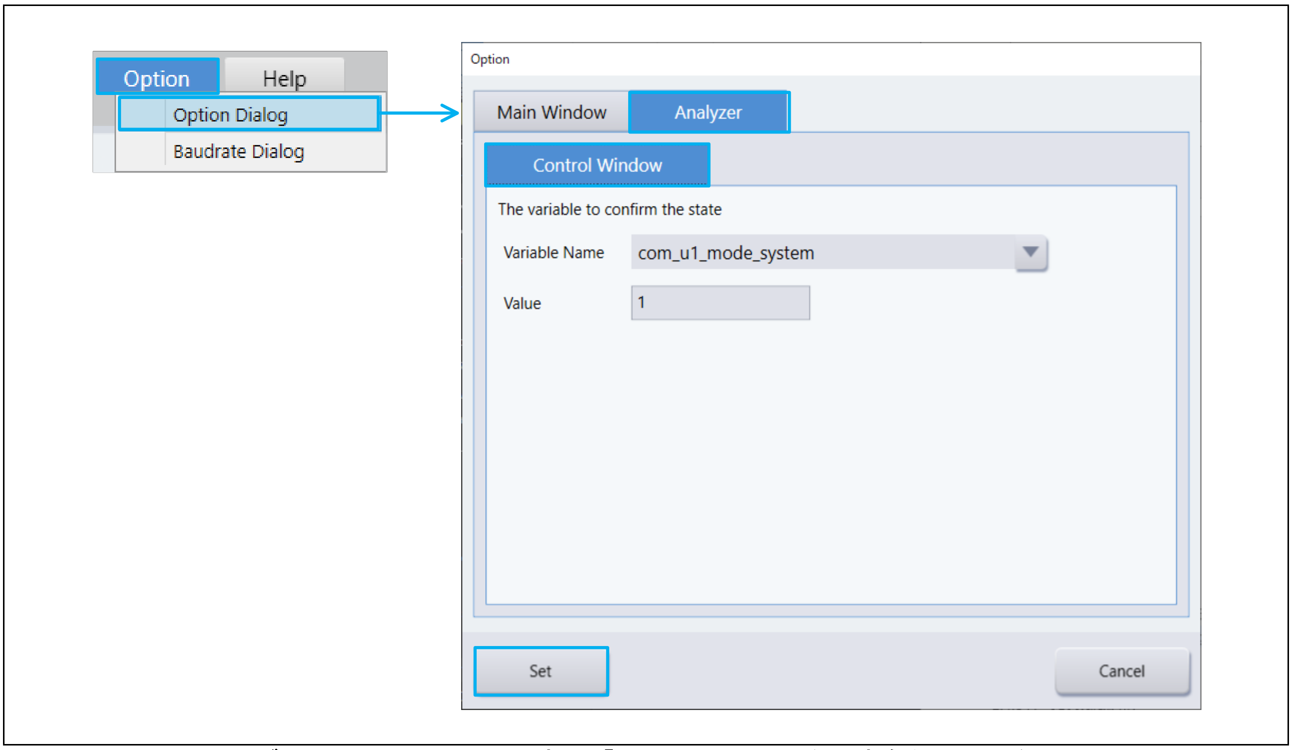


図 2-19　Analyzer タブの Control Window 設定（「モータ駆動中」を判定条件とした例）

2.4.5 USB の接続

2.4.5.1 USB 通信の開始

Main Window の「Connection」の COM 情報のリストを展開すると、COM 番号または「Offline Mode」がプルダウンメニューで表示されます。

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。接続処理が正常終了すると「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。

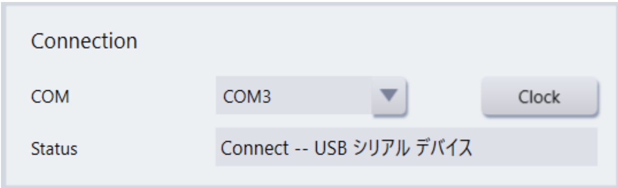


図 2-20 接続後の Connection 画面 Status 表示

表 2-13 接続後の Main Window 表示

画面	表示
Connection Status	「Connect」と表示します。
Configuration	接続環境の情報を表示します（ルネサス提供の実行ファイルのみの機能です）。
Select Tool	• 利用可能なツールアイコンを表示します。 • COM 選択リストで「Offline Mode」を選ぶとオフラインモードとなり、使用できるツール機能は制限されます。

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。

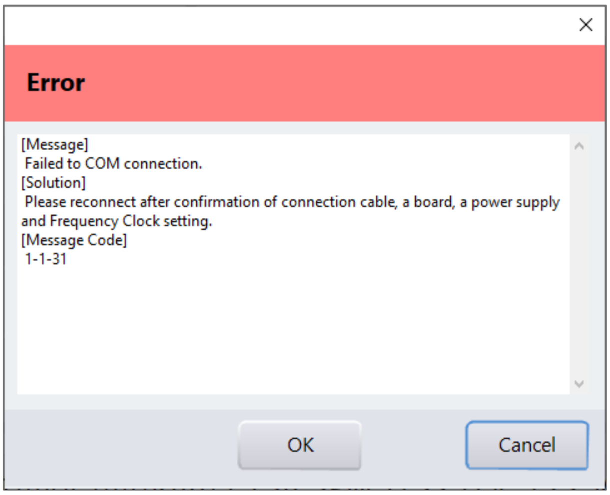


図 2-21 COM 接続時のエラーメッセージ

2.4.5.2 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。

2.4.5.3 クロック周波数の設定（ツール用通信ボードを使用する場合）

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。

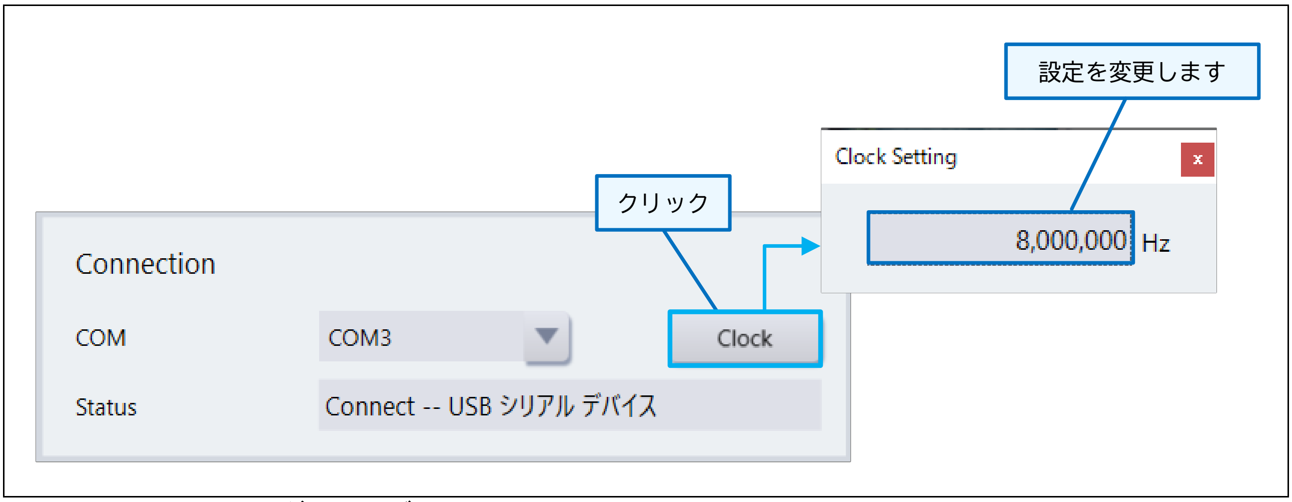


図 2-22 Clock Setting ダイアログ

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）に合わせてして設定してください。

表 2-14 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（66666666 Hz）、12.5 Mbps（10 MHz）、16.66 Mbps（133333333 Hz）

2.4.5.4 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）

ユーザプログラムで、ビルトイン型通信ライブラリを使用（市販の USB シリアル変換モジュールを使用）している場合、下記の操作でボーレートの設定を行ってください。

備考 ビルトイン型通信ライブラリについては、17 章ビルトイン型通信ライブラリを参照してください。

Main Window で「Option」メニューの「Baudrate Dialog」を選択すると、BaudrateSetting ダイアログが表示されます。表示は、現在設定されているボーレートになっているので、その値を変更します。変更したボーレートは、COM 接続時に有効になります。

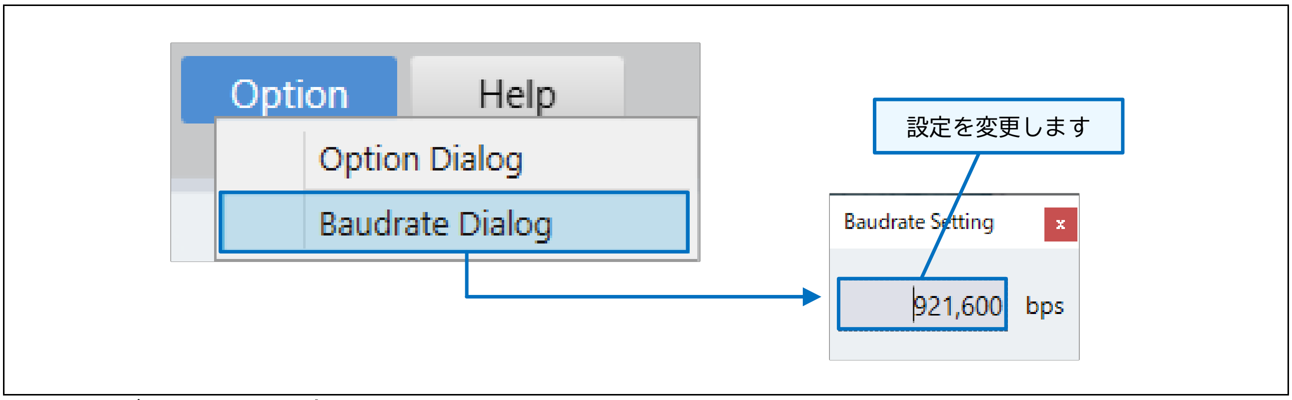


図 2-23 ボーレートの設定

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 シリアル変換モジュールに 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。

2.4.6 ウィンドウの基本操作

Main Window では、Select Tool に利用可能なツールのアイコンが表示されます。これをクリックすると各ツールへ切り替わります。利用可能なツールは、ユーザプログラムと COM 接続の状況により異なります。

各ツール（Analyzer、Tuner、Eazy）の画面では、画面上部にツール切り替えボタンと Main Window に戻るためのボタンが表示されます。ただし、メニューバーの Option ＞ Option Dialon ＞ Analyzer タブの「Control Window」で設定した条件に当てはまる場合、モータ駆動中のツール切り替えが抑止されます（詳しくは、2.4.4 章を参照してください）。

2.4.6.1 ツール切り替えボタン

Main Window より各ツールを起動すると各ツールのフレーム上部中央に「ツール切り替え」ボタンが表示されます。クリックすると別のツールへ切り替えができます。

なお、2.4.4.2 で切り替え抑止の設定をした場合、設定条件を満たす間はツール切り替えを行えません。

2.4.6.2 Main Window 切り替えボタン

Main Window より各ツールを起動すると各ツールのフレーム上部右側に「Main Window」ボタンが表示されます。クリックすると Main Window へ切り替えができます。

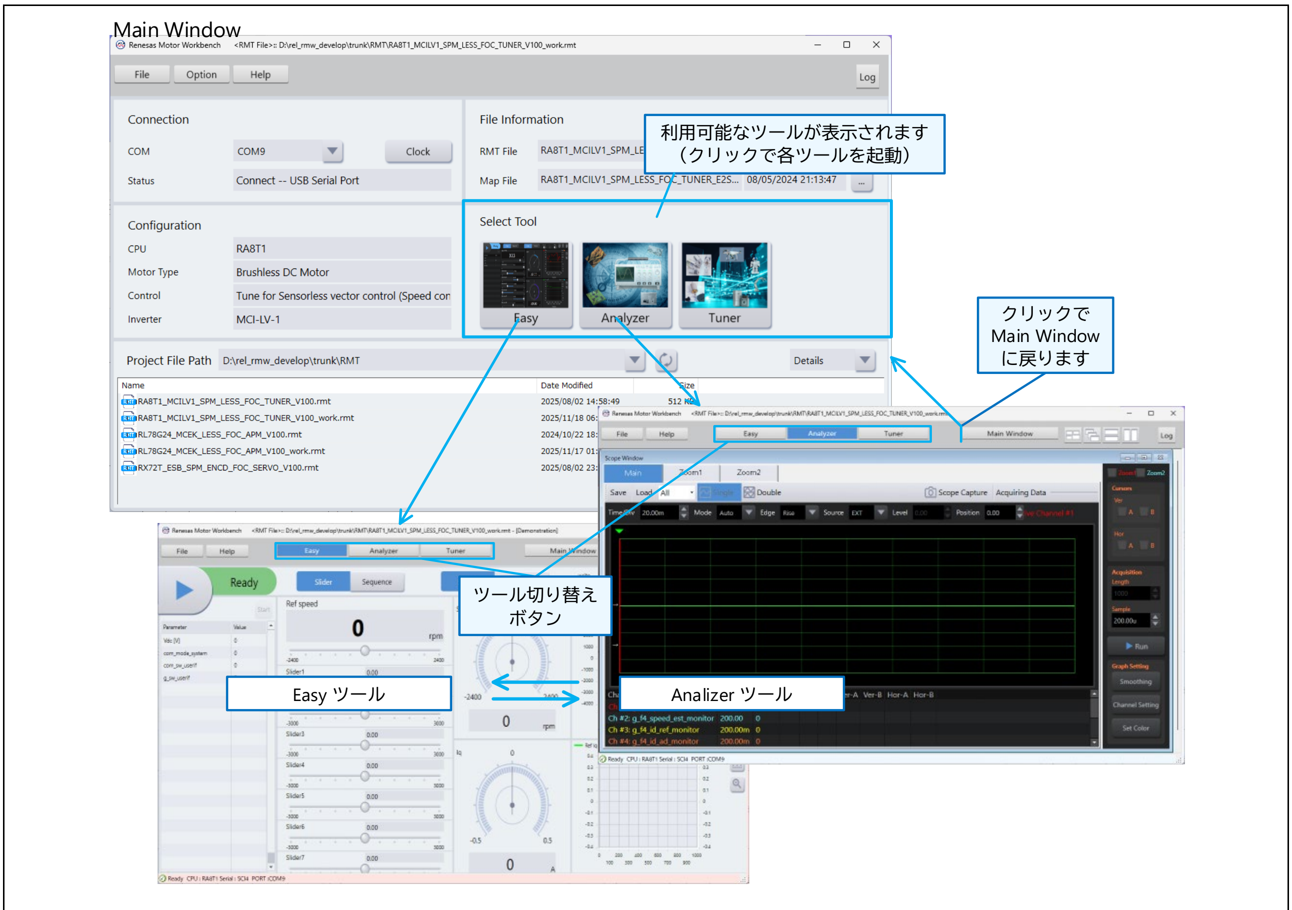


図 2-24 ツール切り替えボタンと Main Window ボタン

ただし、モータ駆動中などの状態で Main Window へ切り替えることは出来ません。

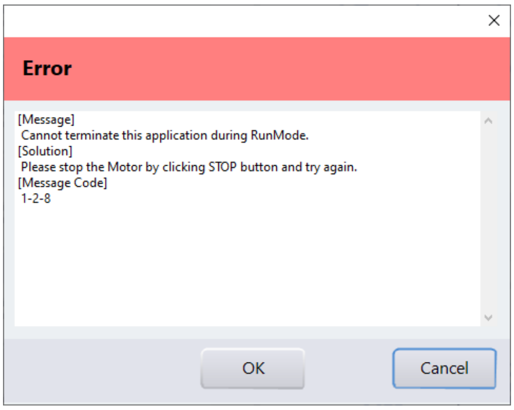


図 2-25　処理実行中のツール切り替えエラーメッセージ

2.4.6.3　　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）。

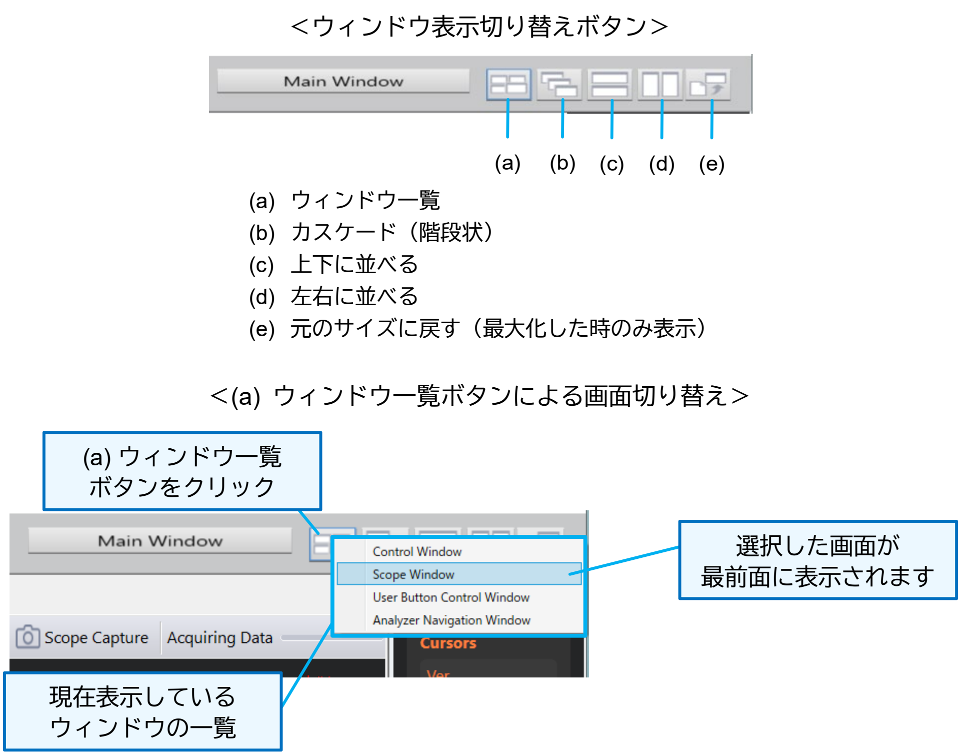


図 2-26　ウィンドウ切り替えボタン

2.4.6.4 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移動（フレームイン）もできます。

図 2-27　ウィンドウのフレームアウトとフレームイン

2.4.7 バージョンの確認方法

RMW のバージョンは、Help メニューの「Version Information」で確認できます。

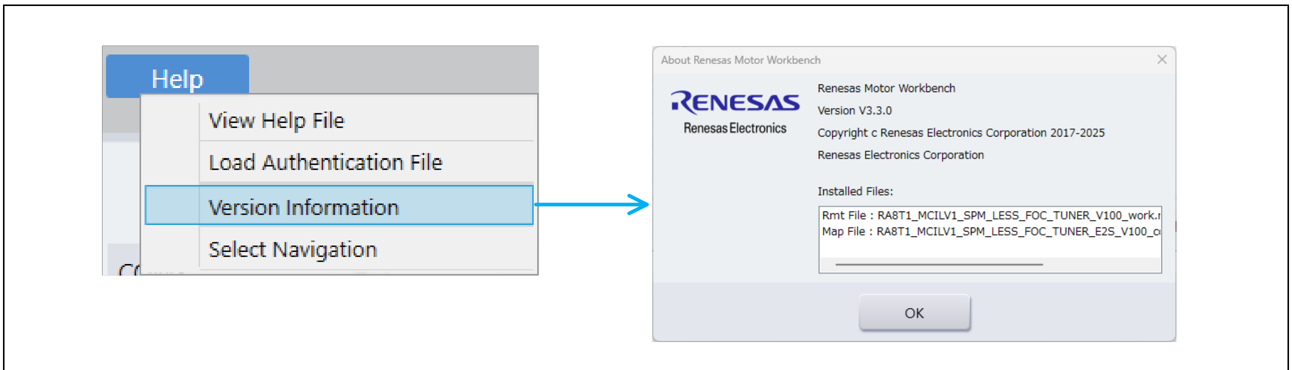


図 2-28　バージョン情報の表示

2.5 Select Navigation 機能

Select Navigation は、RMW の基本設定を補助する機能で、RMT ファイルや Map ファイルの読み込み、COM ポートの設定などをダイアログ形式でサポートします。

Select Navigation は、RMW の起動時に表示されます（起動時の画面で「Do not show this featuers in the future」にチェックを入れると次回から表示されません）。また、Main Window の Help メニューから「Select Navigation」を選択することでも表示できます。

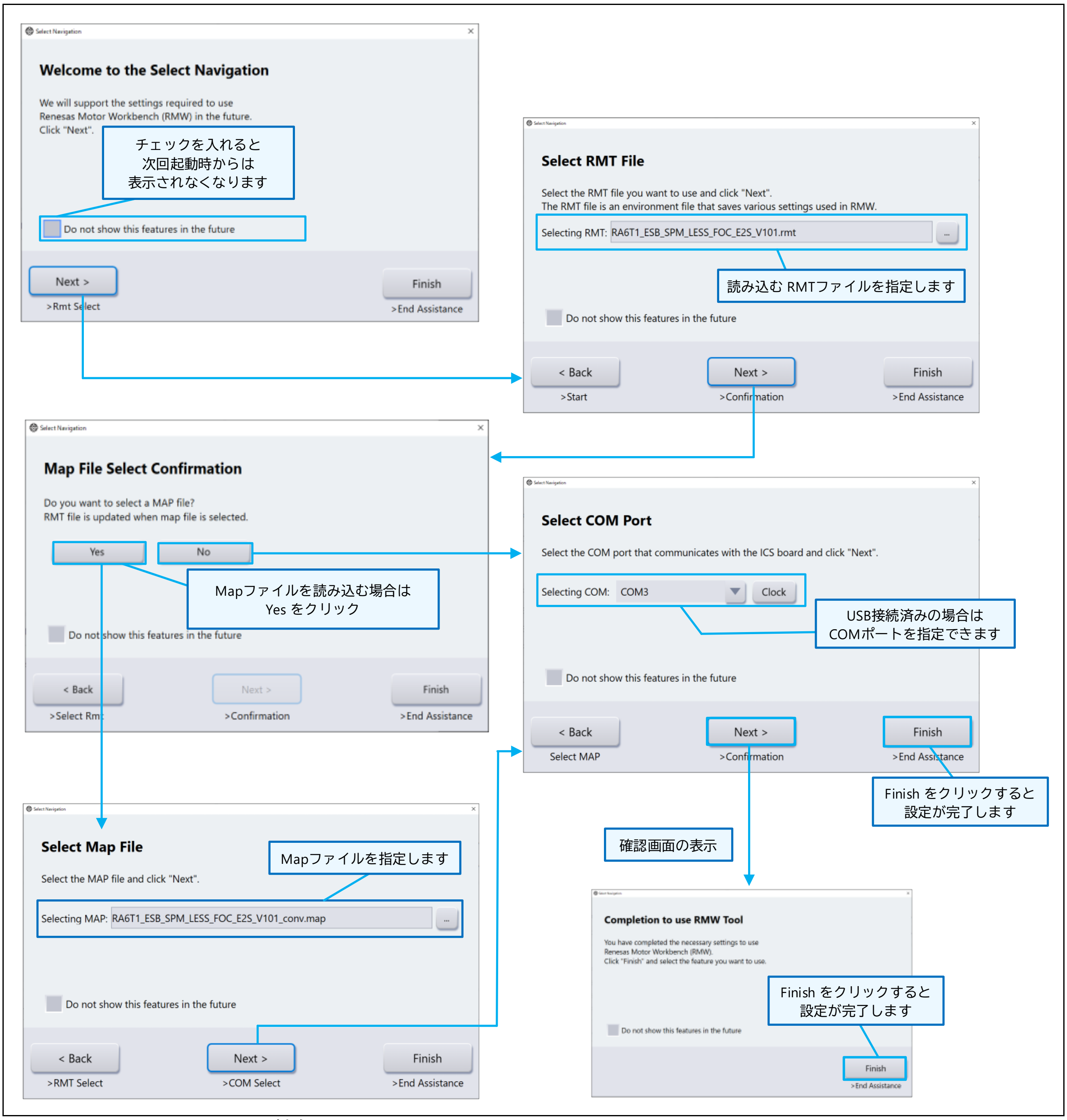


図 2-29 Select Navigation 機能

3. Analyzer ツール

3.1 概要

Analyzer は、モータを回転させたまま、プログラムを止めずに変数の読み書きや、波形の表示ができるツールです。

モータ制御では、プログラムの実行を急に停止すると、PWM 出力の状態によっては大電流が流れてインバータボードが壊れてしまう可能性があるため、一般のアプリケーションのようにプログラムにブレークを張ってマイコン内部の変数を確認することができません。

また、モータ制御、特にベクトル制御では「d 軸」や「q 軸」の電流値のように、プログラムで演算した電流値を制御に使用しているため、オシロスコープ等では値を確認できません。本ツールを使えば、その電流値を直接確認できることも大きな特徴といえます。

そのほか、Analyzer ツールはシーケンスに従ってトリガをかけたり、波形の一部分をズームして表示するなど、多彩な機能を備えています。

Analyzer ツールの機能を有効に活用することで DA コンバータや外部バスを使用してデータを出力したり、メモリに保存して後から解析したりするよりも、はるかに効率よく開発を進めることができます。

3.2 Analyzer ツールの構成

Analyzer ツールには、3 つの基本機能（ウィンドウ）

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

と、上記 Control Window から起動できる以下の 3 つのサブツール（画面）があります。

- Commander
- Status Indicator
- One Shot
- Parameter Output

Analyzer の基本機能は Analyzer ツールを起動すると表示され、Analyzer ツール起動中は、基本機能のウィンドウを閉じることは出来ません。

Analyzer ツール内の各ウィンドウは、Analyzer ツールのウィンドウ外にフレームアウトすることもできます（詳しくは、「2.4.6　ウィンドウの基本操作」を参照してください）。

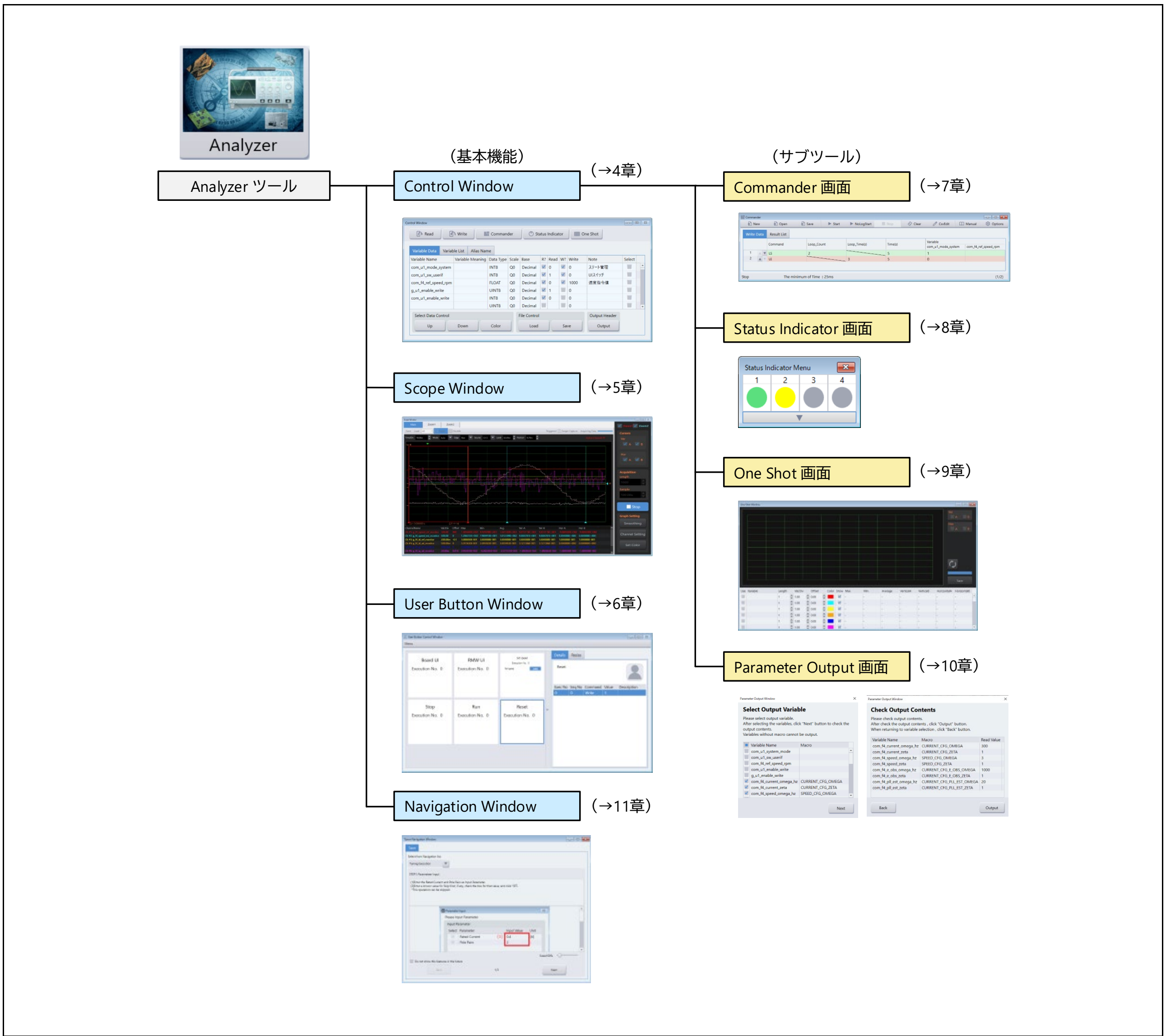


図 3-1　Analyzer ツールの構成

3.3 Analyzer ツールの外観

Analyzer ツールの外観を以下に示します（ウィンドウを上下に並べて表示しています）。

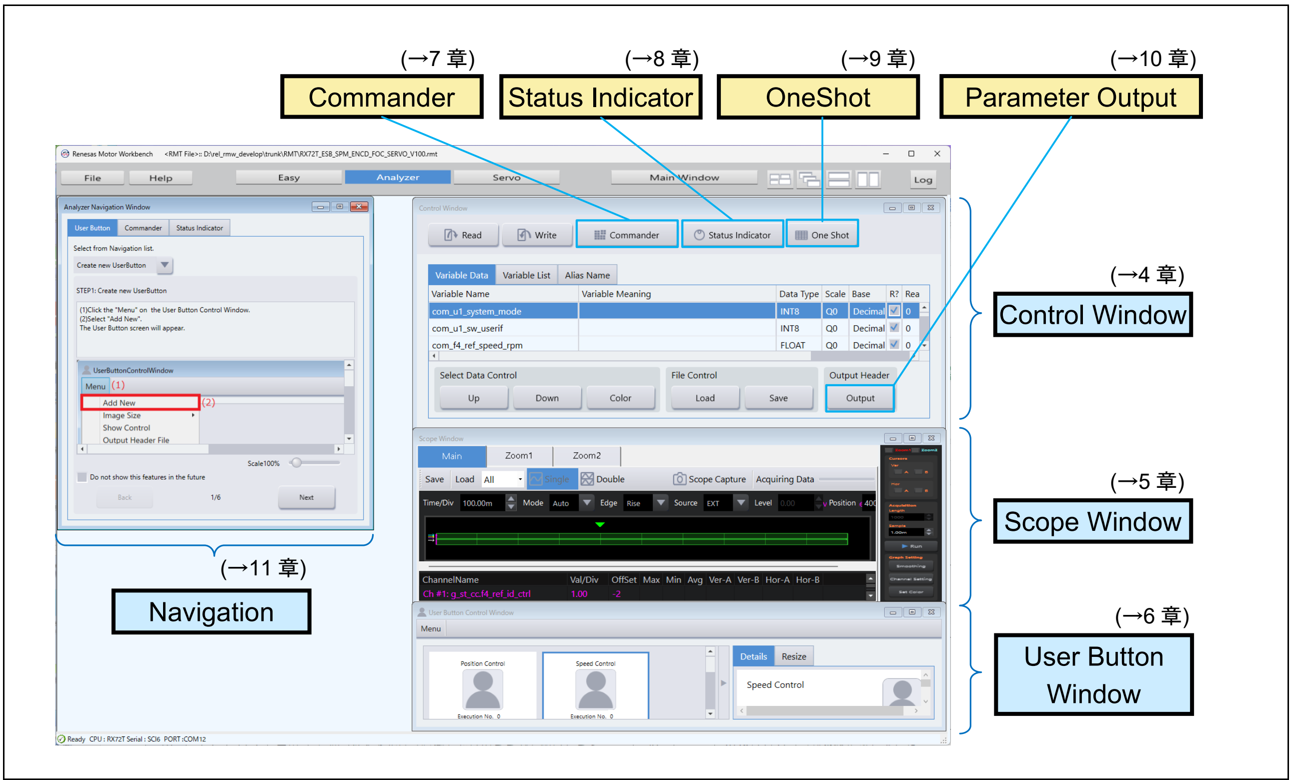


図 3-2 Analyzer ツールの外観

表 3-1 Analyzer ツールのウィンドウ

ウィンドウ名		説明	参照章
基本機能	Control Window	変数の読み書きがリアルタイムにできます。Map ファイルから読み込んだ変数の中から、対象の変数を指定して操作します。	4 章
	Scope Window	選択した変数をオシロスコープのように波形で表示します。ズーム表示や画面のキャプチャも可能です。	5 章
	User Button Window	予め登録したシーケンス処理を、ユーザの指示（ボタンのクリック）で順次実行します。	6 章
	Navigation Window	Renesas Motor Workbench の各機能の操作手順を説明します。	11 章
サブツール	Commander 画面	変数への書き込み指示のシーケンスを予め登録し、実行することができます。インターバル間隔の指定やループ処理なども可能です。	7 章
	Status Indicator 画面	変数の値を監視し、設定した条件に合致（閾値超え）すると表示灯が点灯します。	8 章
	One Shot 画面	バッファリングされたデータ（指定した変数から連続したアドレスのデータ）を波形で表示します。	9 章
	Parameter Output	Analyzer で調整したパラメータを、モータ制御プログラムに組み込むためのヘッダファイルをとして出力します。	10 章

4. ［Analyzer］Control Window

4.1 概要

Control Window で Map ファイルから読み込んだ変数に対して、値の読み書きをリアルタイムに行うことができます。Control Window は Analyzer ツールを起動すると表示されます。

Control Window には、Variable Data タブ、Variable List タブ、Alias Name タブの 3 つのタブがあります。また、Control Window 内のボタンから、サブウィンドウ（Commander 画面、Status Indicator 画面、One Shot 画面）を起動できます。

4.2 特徴

- リアルタイムな変数読み込みで、瞬時値の取得に適しています。
- 操作する変数を選ぶ際に、入力した文字から候補のリストが自動で表示されます。

4.3 画面構成

Control Window の画面構成と機能を以下に示します。

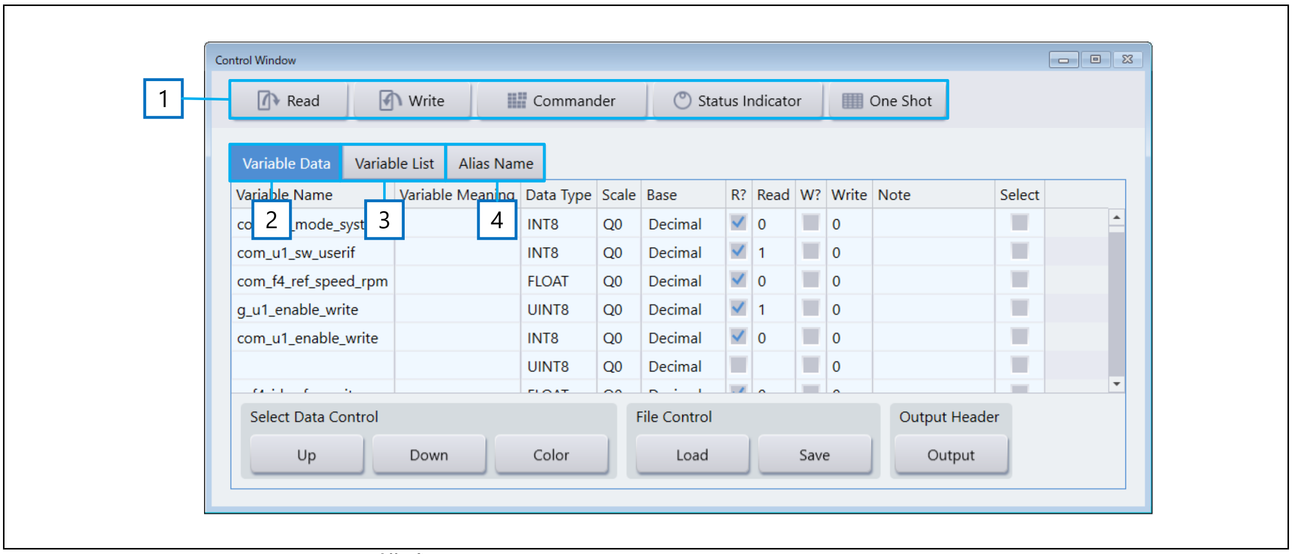


図 4-1 Control Window の画面構成

表 4-1 Control Window 各部の説明

No.	名称	説明
1	操作ボタン	Read ボタンと Write ボタンのクリックで、変数の値の読み込み／書き込みを行います（ターゲットが接続されていない状態では操作できません）。 Commander、Status Indicator、OneShot ボタンをクリックすると、各機能を起動します。
2	Variable Data タブ	Map ファイルから読み込んだ変数情報から、変数名を指定して値の読み込みと書き込みができます。
3	Variable List タブ	Map ファイルから読み込んだ変数情報の一覧を表示します。
4	Alias Name タブ	Map ファイルから読み込んだ変数情報の変数名に対して Alias（別名）を付けることができます。

4.3.1 操作ボタン

操作ボタンの名称と機能を説明します。

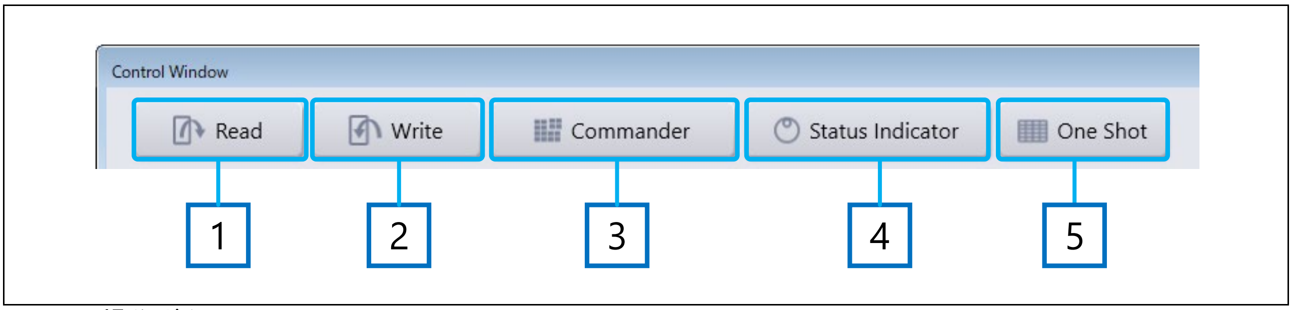


図 4-2　操作ボタン

表 4-2　操作ボタンの機能

No.	名称	説明
1	Read	変数の値を読み込みます。
2	Write	変数に値を書き込みます。
3	Commander	Commander 画面を起動します。 （詳細は「7 章［Analyzer］Commander 画面」を参照してください）
4	Status Indicator	Status Indicator Menu 画面を起動します。 （詳細は「8 章［Analyzer］Status Indicator 画面」を参照してください）
5	One Shot	One Shot 画面を起動します。 （詳細は「9 章［Analyzer］One Shot 画面」を参照してください）

4.3.2 Variable Date タブ

Variable Date タブでは、対象となる変数を登録して、変数の値の読み込みと書き込みを行います。
Variable Date タブの各部の名称と機能を以下に示します。

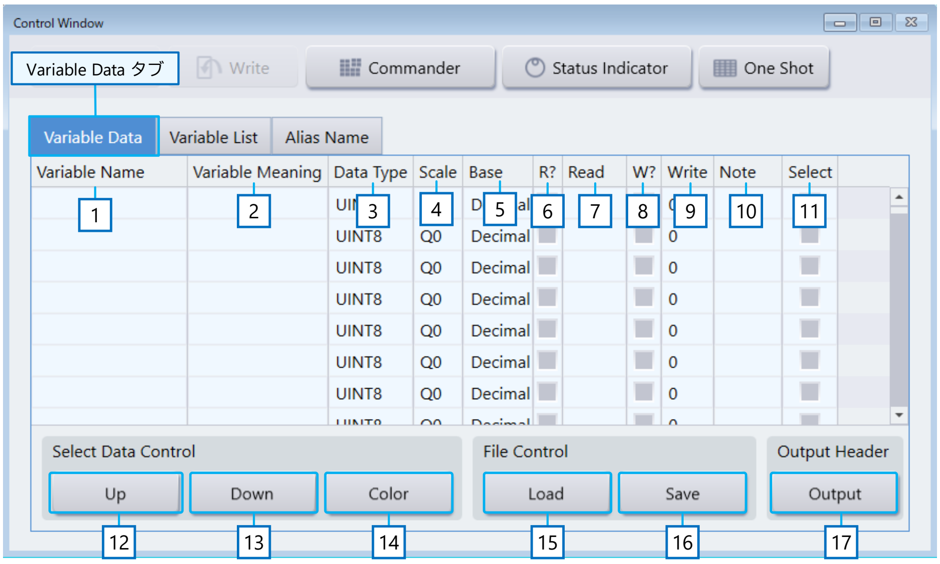


図 4-3 Variable Date タブ画面

表 4-3 Variable Date タブの機能

No.	名称	説明
1	Variable Name	読み書きする変数の名前を指定します。 指定は、直接入力／リスト選択／Variable Find 機能 からできます。 （指定方法の詳細は「4.4.1 Variable Name の設定」を参照してください。）
2	Variable Meaning	Variable Name に対応する用途と単位を表示します 。
3	Data Type	Map ファイルから読み込んだ変数のデータ型を表示します。 データ型は変更が可能ですがこの画面でのみ有効です（Variable List には反映されません）。
4	Scale	変数のスケール値を設定します。
5	Base	表示する進数を設定します。
6	R?	Read ボタンクリック時の値の読み込み ON/OFF を指定します。
7	Read	読み込んだ値を表示します。
8	W?	Write ボタンクリック時の値の書き込み ON/OFF を指定します
9	Write	書き込む値を直接入力します。
10	Note	自由に記入できます（記入した情報は RMT ファイルに保存されます）。
11	Select	選択すると、行の移動や背景色の変更が一括でできます（複数選択が可能）。
12	Up ボタン	Select で選択した行を 1 行上に移動します。
13	Down ボタン	Select で選択した行を 1 行下に移動します。
14	Color ボタン	Select で選択した行の背景色を変更します。
15	Load ボタン	読み書きする変数の設定情報を CSV ファイルから読み込みます。
16	Save ボタン	このタブで設定した情報を CSV ファイルへ保存します。
17	Output ボタン	Parameter Output 画面を起動します 。

4.3.3 Variable List タブ

Variable List タブには、Map ファイルから読み込んだ、すべての変数情報が表示されます。変数情報は表示のみで、Description 欄以外は、この画面で編集はできません。

Variable List タブの各部の名称と機能を以下に示します。

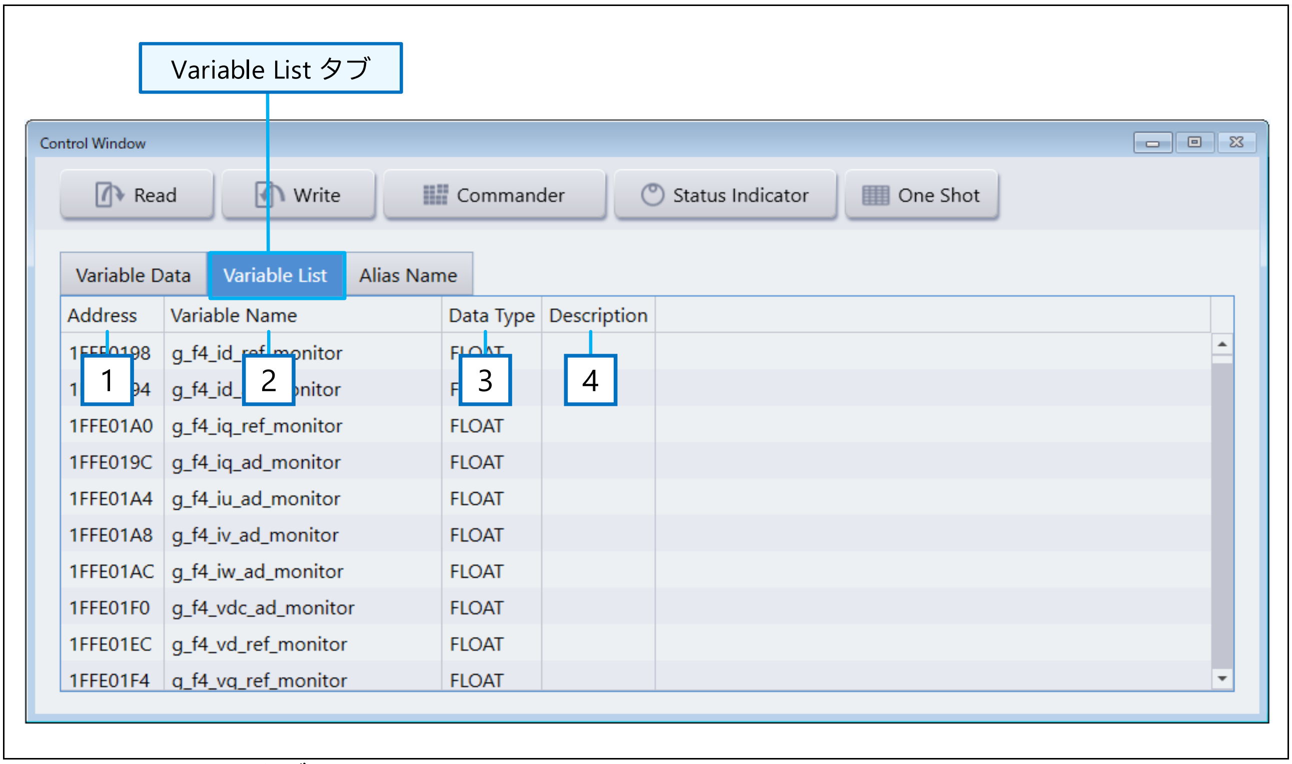


図 4-4 Variable List タブ画面

表 4-4 Variable List タブの機能

No.	名称	説明
1	Address	変数のアドレスを表示します（編集はできません）。
2	Variable Name	変数名を表示します（編集はできません）。
3	Data Type	変数のデータ型を表示します（編集はできません）。
4	Description	説明を自由に記入できます（記入した情報は RMT ファイルに保存されます）。

4.3.4 Alias Name タブ

Alias Name タブでは、Map ファイルから読み込んだ変数に Alias Name（別名）を設定できます。設定した内容は、Alias Name 表示として以下で利用できます。

- Scope Window の Channel Setting 画面
- User Button Window
- Commander 画面
- Variable Find 画面（変数検索）

Alias Name タブの各部の名称と機能を以下に示します。

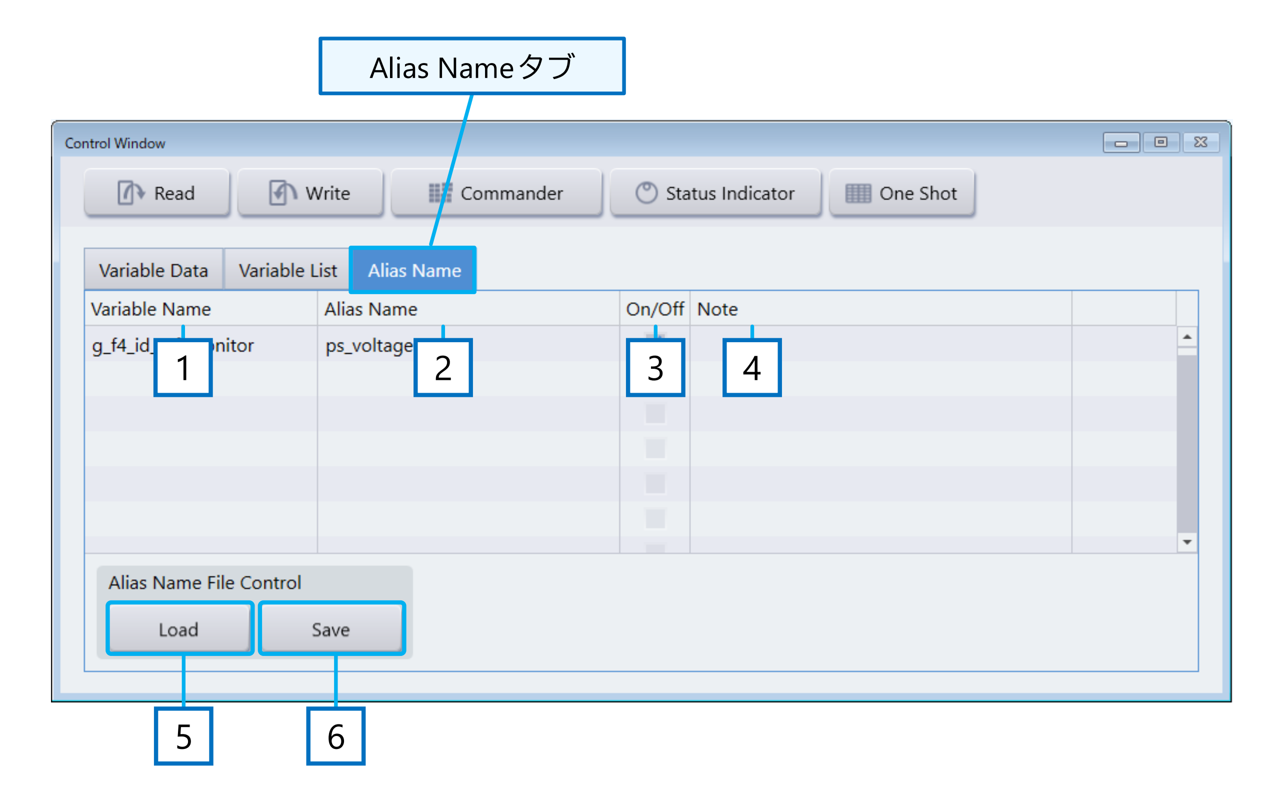


図 4-5 Alias Name タブ画面

表 4-5 Alias Name タブの機能

No.	名称	説明
1	Variable Name	Alias Name（別名）を設定する変数名を指定します。 指定は、直接入力／リスト選択／Variable Find からできます。 （詳細は「4.4.1 Variable Name の設定」を参照してください。）
2	Alias Name	Alias Name（別名）を設定します。
3	On/Off	Alias Name の有効／無効を設定します（チェックボックスにチェックを入れると設定が有効になります）。
4	Note	自由に記入できます（記入した内容は RMT ファイルに保存されます）。
5	Load	Alias Name の設定情報を CSV ファイルから読み込みます。
6	Save	設定した Alias Name 情報を CSV ファイルへ保存します。

4.4 変数の設定（Variable Data タブ）

4.4.1 Variable Name の設定

変数の値の読み込みや変数へ値の書き込みをするためには、Control Window の Variable Data タブで操作対象の変数名を指定します。この変数名の指定は、下記の (1)~(3) のいずれかの操作で行います。

4.4.1.1 Variable Name セルへの直接入力

Variable Data タブの Variable Name セルをクリックして選択し（セルの色が変わる）、さらにもう 1 回クリックすると、編集可能な状態になります。この状態から変数名を直接入力することができます。

文字を入力し始めると、読み込んだ変数情報から一致する可能性のある変数の候補が自動でリスト表示されるので、その中から変数を選ぶこともできます。

ただし、文字を途中まで入力した状態で、変数を指定しないままセルの選択を解除すると、それまでの入力がクリアされます。

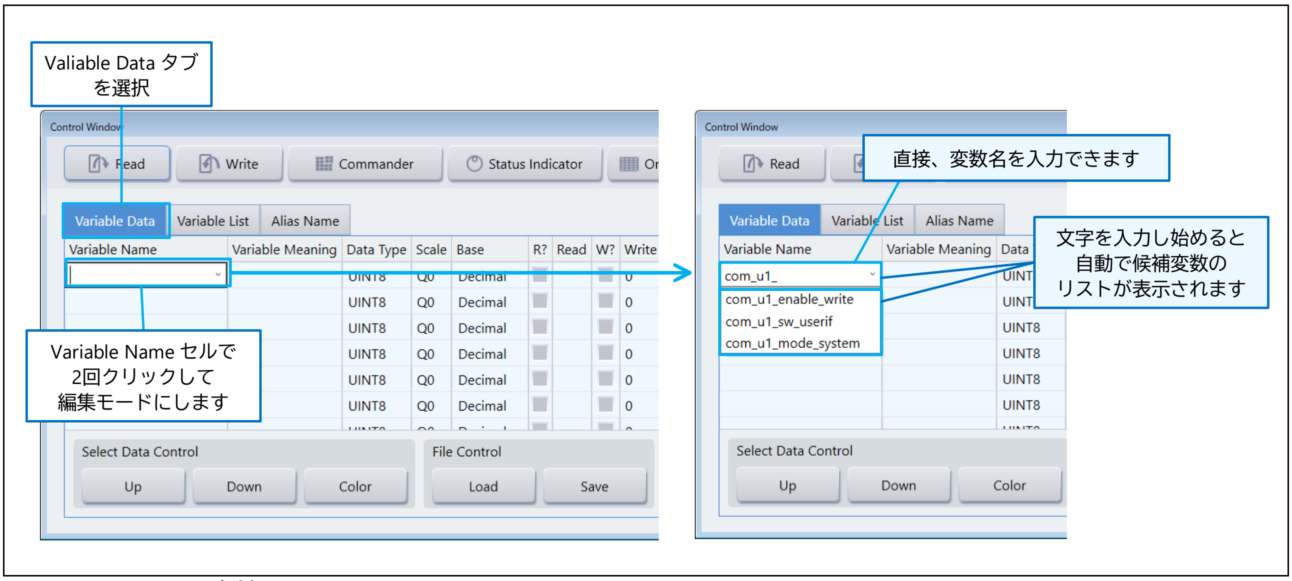


図 4-6 セルへの直接入力

4.4.1.2 一覧リストからの変数選択

Variable Name セルをクリックして選択し（セルの色が変わります）、さらにもう一度クリックすると、編集可能な状態になります。この状態で、セルの右側に「v」が表示されます。この「v」をクリックすると、読み込んだ変数情報の一覧リストが表示されるので、その中から変数を選択します。

また、変数名を途中まで入力した状態で「v」をクリックすると、合致する可能性のある変数の候補が自動でリスト表示されるので、その中から変数を選ぶこともできます（ (1)と同じ動作）。

ただし、この場合も文字を途中まで入力した状態で、変数を指定しないままセルの選択を解除すると、それまでの入力がクリアされます。

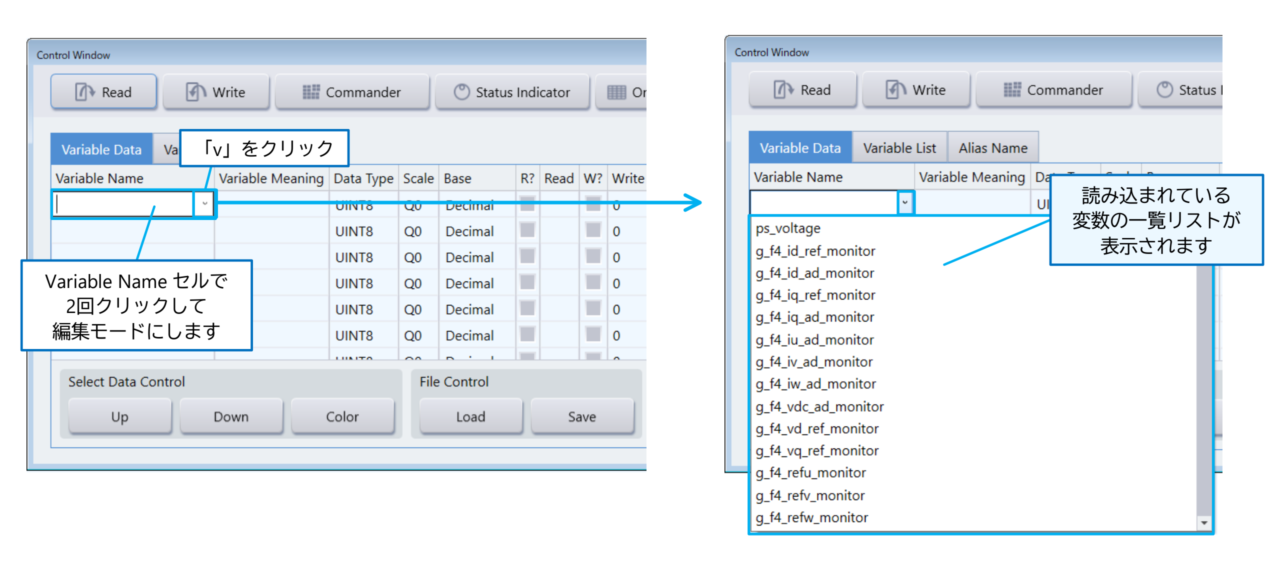


図 4-7 一覧リストからの変数選択

4.4.1.3 Variable Find 機能を使った変数選択

Variable Name セルを右クリックするとメニューが表示されます。このメニューから「Variable Find」を選択すると Variable Find 画面が表示されます。この画面からキーワードによる変数名または用途情報を絞り込み、変数を選択できます。

Variable Find 画面のテキスト入力欄にキーワードを入力後、「Find」ボタンをクリックします。複数キーワードを「スペース区切り」で入力すれば、AND 検索ができます。また、Variable Find 画面下部のチェックボックスにチェックマークを付けると、用途情報を読み込んだ変数のみ検索対象に絞り込むことが可能です。

Variable Find 機能の操作を以下に示します。

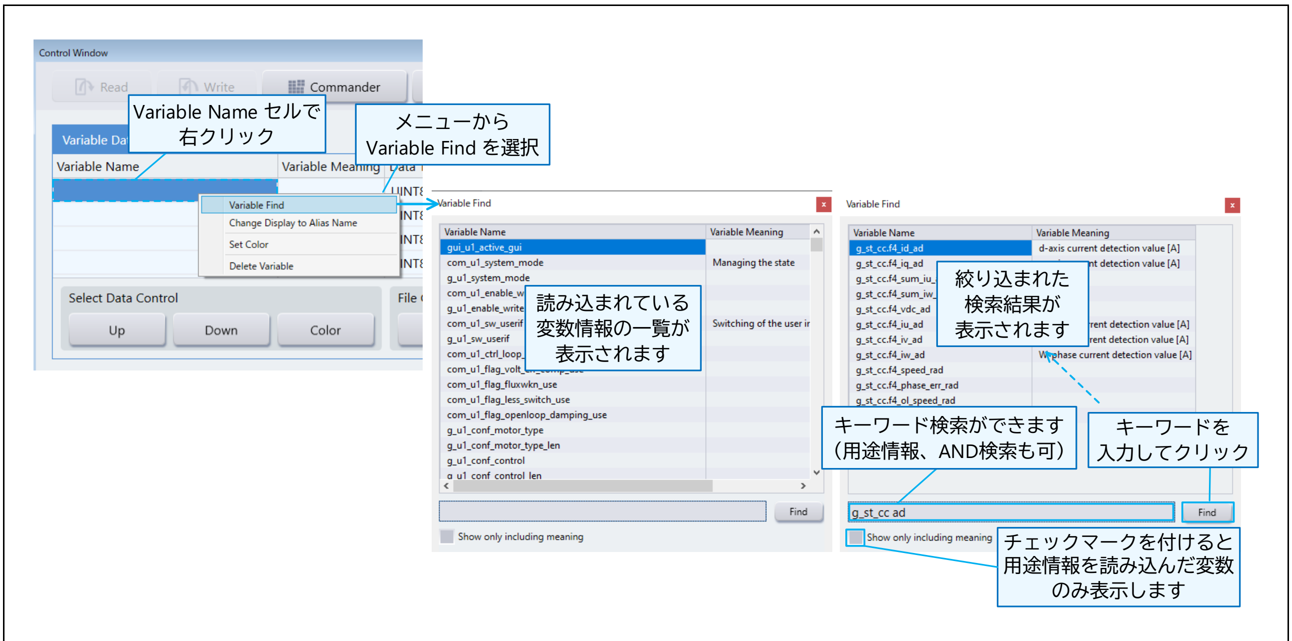


図 4-8 Variable Find 機能を使った変数選択

4.4.2 Data Type の設定

Control Window の Variable Data タブには Data Type 欄があり、Variable Name に変数名が設定されると、Map ファイルから読み込んだ「変数情報」の Data Type が自動で表示されます。

Data Type 欄は、Control Window での Read/Write 操作で使用する変数の Data Type の変更ができます。Data Type 欄の右側「v」をクリックすると選択リストが表示され、選択すると変更ができます。

表 4-6 Data Type 一覧

Data Type	説明
UINT8	8bit 符号なし 整数型
INT8	8bit 符号あり 整数型
UINT16	16bit 符号なし 整数型
INT16	16bit 符号あり 整数型
UINT32	32bit 符号なし 整数型
INT32	32bit 符号あり 整数型
FLOAT	32bit 浮動小数点型
BOOL	True と False の 2 値をとるデータ型
LOGIC	Bit Field をビット表示

4.4.3 Scale の設定

Control Window の Variable Data タブには Scale 欄があり、Variable Name に変数名が設定されると初期値として「Q0」が自動で表示されます。

Scale 欄は、Control Window での Read/Write 操作で使用するデータに対する Scale を変更できます。Scale 欄は直接入力が可能です。入力可能な値は Qn（n = 0～31）、正の整数、正の小数です

表 4-7 Scale 入力値

入力値	データの計算式
Qn（n = 0～31）	データ × (1÷(2 の n 乗))
正の整数	データ × (1÷整数)
正の小数	データ × (1÷小数)

Scale 欄の直接入力では、入力に誤りがあるとエラー表示になります。

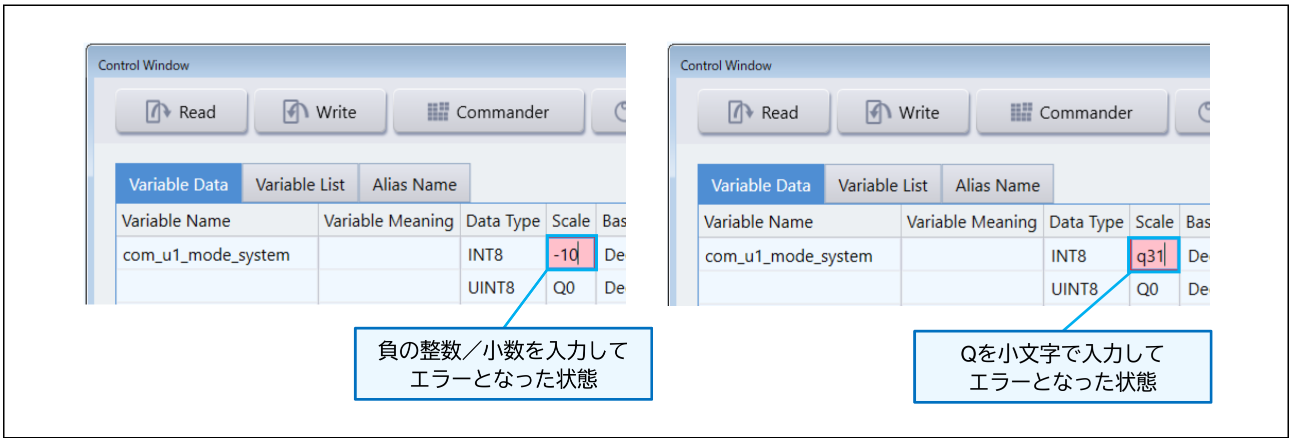


図 4-9 Scale 入力時のエラー表示

4.4.4 Base の設定

Control Window の Variable Data タブには Base 欄があり、初期値として Decimal（10 進数）が設定されています。

Base 欄は、Control Window での Read/Write 操作で使用するデータに対する Base を変更できます。Decimal（10 進数）、Binary（2 進数）、Octal（8 進数）、Hex（16 進数）から選択してください。

4.4.5 変数情報の Note 欄

Control Window の Variable Data タブには Note 欄があり、変数情報のメモなど、ユーザが自由に記入できます。

Note 欄へ記載した情報は、RMT ファイルに保存されます（RMT ファイルを保存して終了した場合、次回起動時に情報が表示されます）。

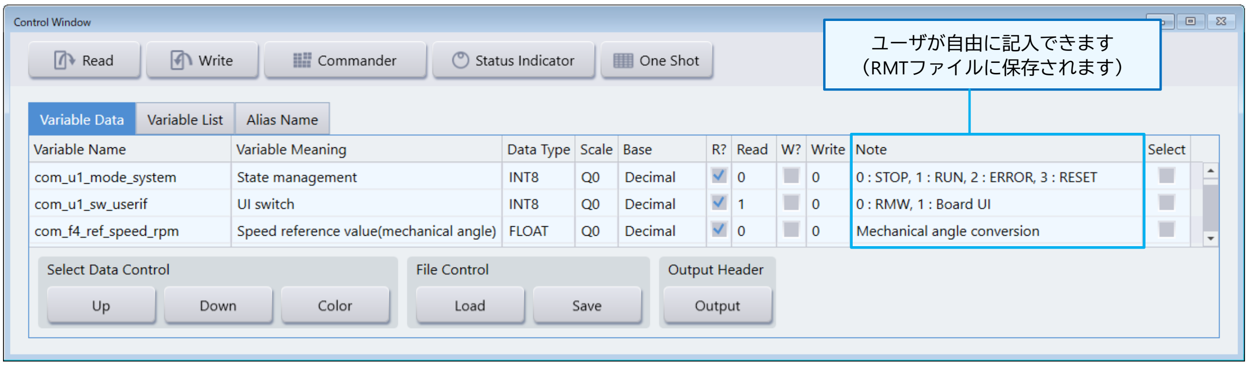


図 4-10 Variable Data タブの Note 欄

4.4.6 変数値の読み込み

Control Window の Variable Data タブで、変数値の読み込みができます。

「R?」欄のチェックボックスを ON にして、「Read」ボタンをクリックすると、読み込んだ値が Read 欄に表示されます（ターゲットを接続していない場合、「Read」ボタンは無効になっています）。

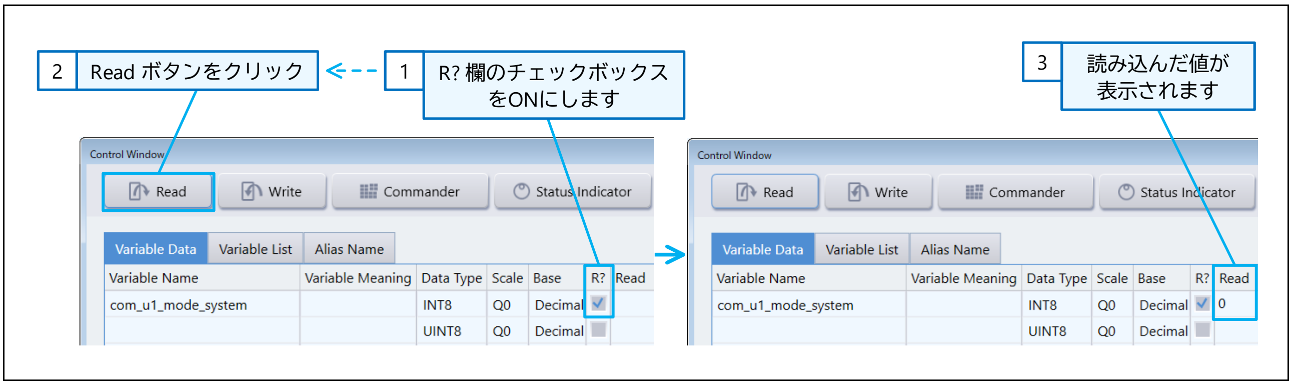


図 4-11 変数値の読み込み

「R?」の列にあるセルのチェックボックスは、一括で ON/OFF の切り替え操作ができます。

4.4.6.1 複数セルを選択して切り替え操作

複数セルを選択した（セルが青色になった）状態から右クリックして表示されるメニューの「Select Set」「Select Clear」を選択すると、選択セルのチェックボックスを一括で切り替えられます。

4.4.6.2 全てのセルの切り替え操作

「R?」セルの右クリックメニューで「All Set」「All Clear」を選択すると、「R?」の全セルのチェックボックスを一括で切り替えられます。

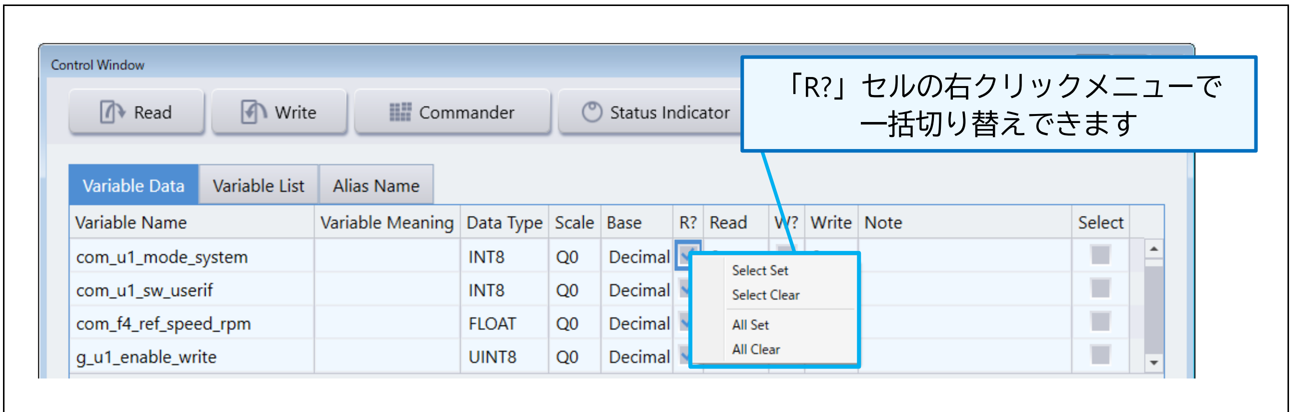


図 4-12 「R?」チェックボックスの一括切り替え

4.4.7 変数値の書き込み

Control Window の Variable Data タブから、変数値の書き込みができます。

「W?」欄のチェックボックスを ON 状態にします。次に「Write」欄に変数へ書き込む値を入力します。その後、「Write」ボタンをクリックすると、変数への書き込みが実行されます（ターゲットを接続していない場合、「Write」ボタンは無効になっています）。

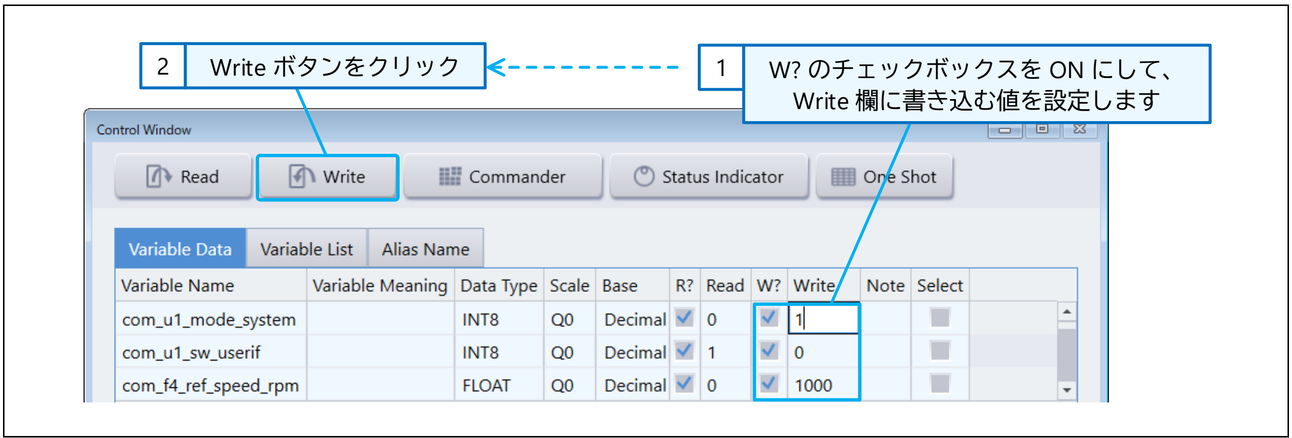


図 4-13 変数値の書き込み

書き込み結果は、「4.4.6 変数値の読み込み」の操作で確認できます。

「W?」の列にあるセルのチェックボックスは、一括で ON/OFF の切り替え操作ができます。

4.4.7.1 複数セルを選択して切り替え操作

複数セルを選択した（セルが青色になった）状態から右クリックして表示されるメニューの「Select Set」「Select Clear」を選択すると、選択セルのチェックボックスを一括で切り替えられます。

4.4.7.2 全てのセルの切り替え操作

「W?」セルの右クリックメニューで「All Set」「All Clear」を選択すると、「W?」の全セルのチェックボックスを一括で切り替えられます。

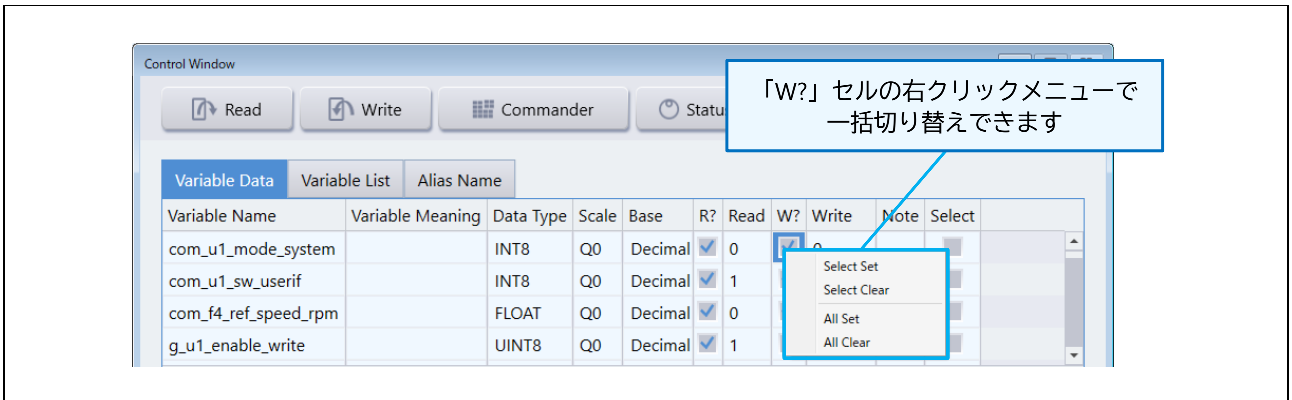


図 4-14 「W?」チェックボックスの一括切り替え

4.4.8 変数表示の並び替え

Control Window の Variable Data タブに設定した変数名の表示の並び替えができます。移動する行の Select 欄のチェックボックスを ON 状態にします。その後、画面下の「Up」「Down」ボタンをクリックすると入れ替えができます。複数行を選択して操作することもできます。

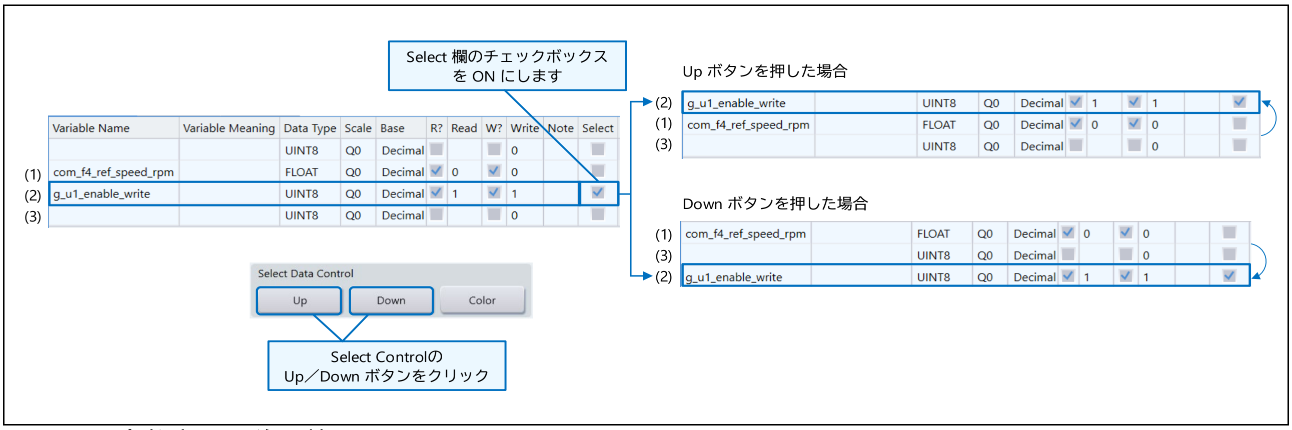


図 4-15 変数表示の並べ替え

4.4.9 変数表示の背景色変更

Control Window の Variable Data タブの行の背景色を変更して、変数を目立たせることができます。変更方法は、下記の 2 つがあります。

4.4.9.1 右クリックメニューからの変更

背景色を変更したい行で右クリックすると表示されるメニューの「Set Color」を選択すると、色の設定画面が表示され、色選択すると背景色の変更ができます。

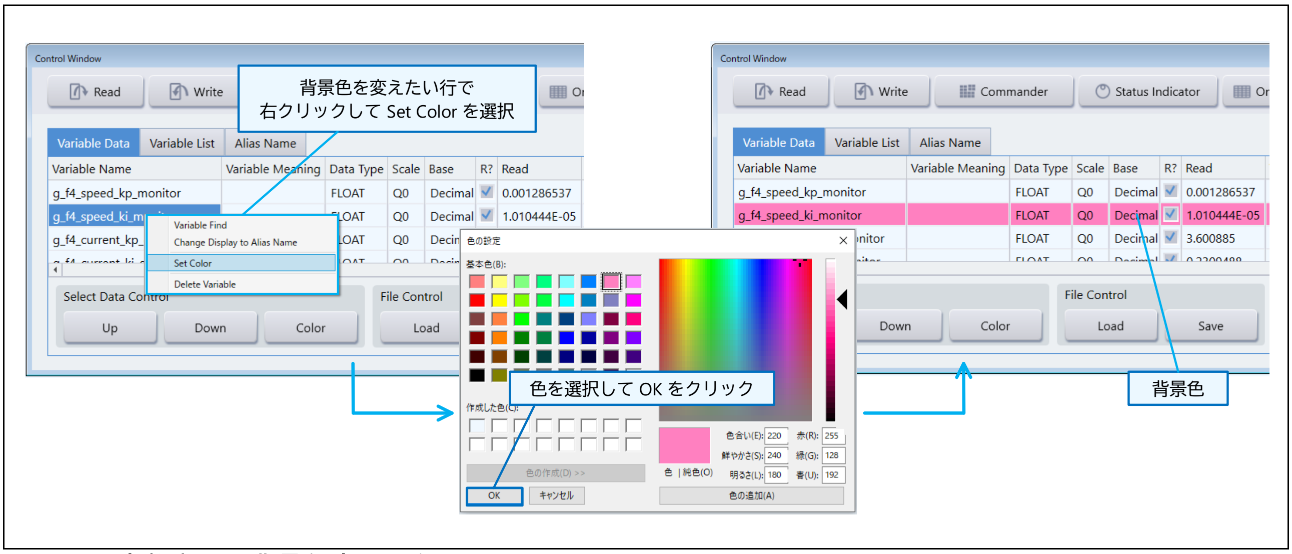


図 4-16 変数表示の背景色変更（右クリックメニュー）

4.4.9.2 Select 欄使用による行選択

背景色を変更したい行の Select 欄のチェックボックスを ON 状態にします。その後、画面下部の「Color」ボタンをクリックすると、色の設定画面が表示され、色選択をすると背景色の変更ができます。複数行を選択して操作することもできます。

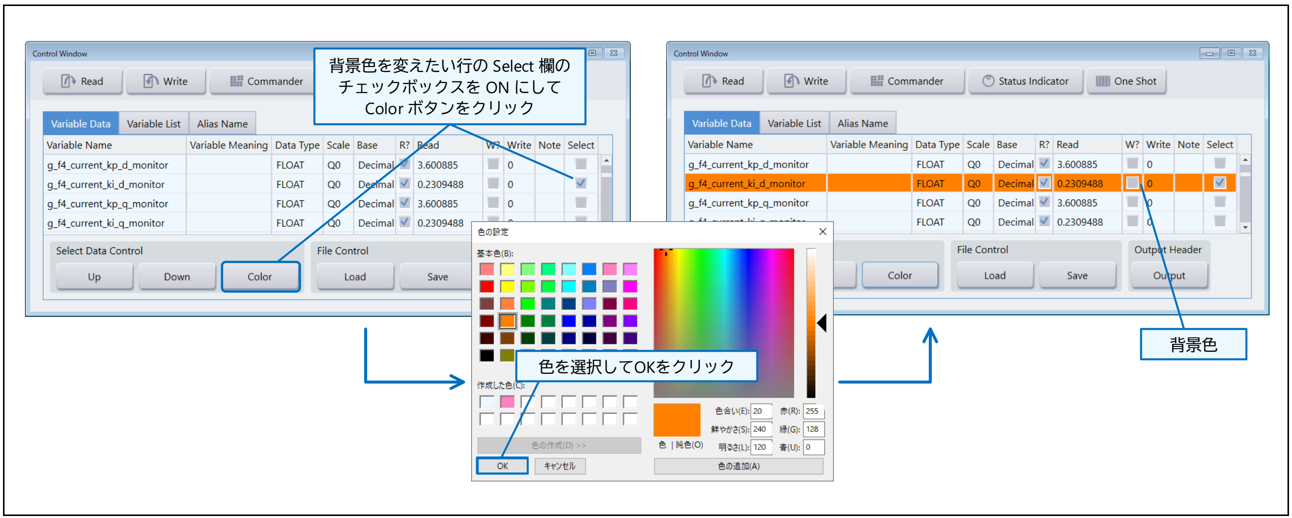


図 4-17 変数表示の背景色変更（Select 欄使用による行選択）

4.4.10 変数の削除

Control Window の Variable Data タブの設定中の変数名を削除するには、下記の 2 つの方法があります。変数名を削除すると、削除した行の情報はリセットされます。

4.4.10.1 Delete キーによる削除

削除する変数名のセルを選択状態にして、Delete キーをクリックすると削除ができます。

4.4.10.2 右クリックメニューから削除

削除する変数名の行（但し、R? / W? / Select 欄の上は除く）で右クリックすると表示されるメニューから「Delete Variable」を選択すると削除ができます。

変数名が表示されているセルを複数選択した状態から上記 (1) (2) の操作をすると、複数行の削除ができます。変数名が削除されると、その行に設定した情報は初期状態にリセットされます。
なお、Select 欄のチェックを ON にしても削除はされません。

4.4.11 変数情報の保存／読み込み

Variable Data タブで設定した変数情報は、CSV ファイル形式で保存（Save）／読み込み（Load）ができます。

4.4.11.1 保存（Save）の操作

Variable Data タブの「Save」ボタンをクリックすると、設定情報を CSV ファイル形式で保存ができます。

4.4.11.2 読み込み（Load）の操作

Variable Data タブの「Load」ボタンをクリックすると、CSV ファイルから設定情報を読み込み、Variable Name タブのリストへ反映（上書き）します。

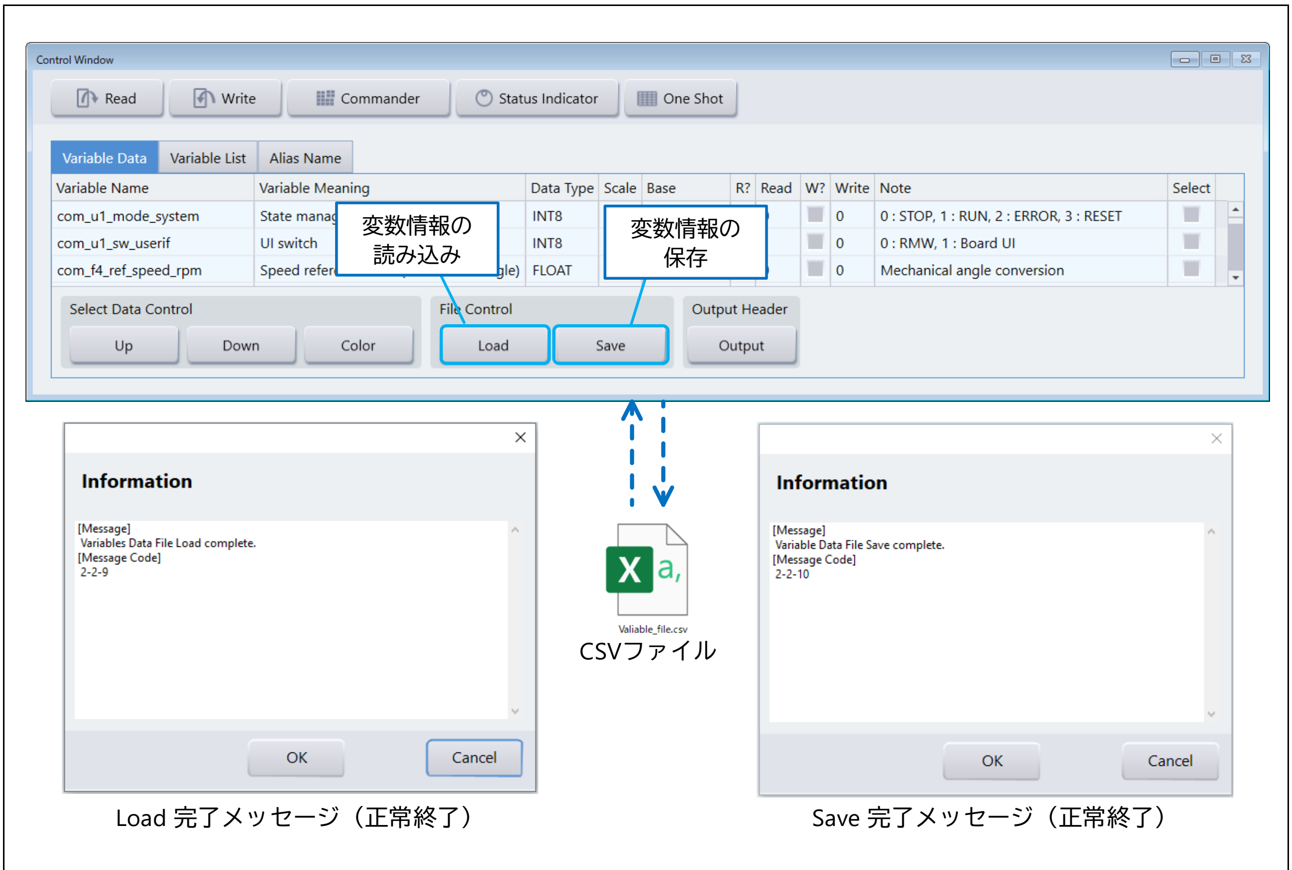


図 4-18 変数情報の保存／読み込み（CSV ファイル形式）

なお、RMT ファイルの保存を行うと、Control Window の各タブ（Variable Data タブ、Variable List タブ、Alias Name タブ）に設定した情報も保存されます。また、保存した RMT ファイルを読み込むことで、情報が復元されます。

4.5　変数の一覧表示（Variable List タブ）

Variable List タブには、Map ファイルから読み込んだ、すべての変数情報が表示されます。変数情報は表示のみで、Description 欄以外は、この画面で編集はできません。

画面構成については、4.3.3 章を参照してください。

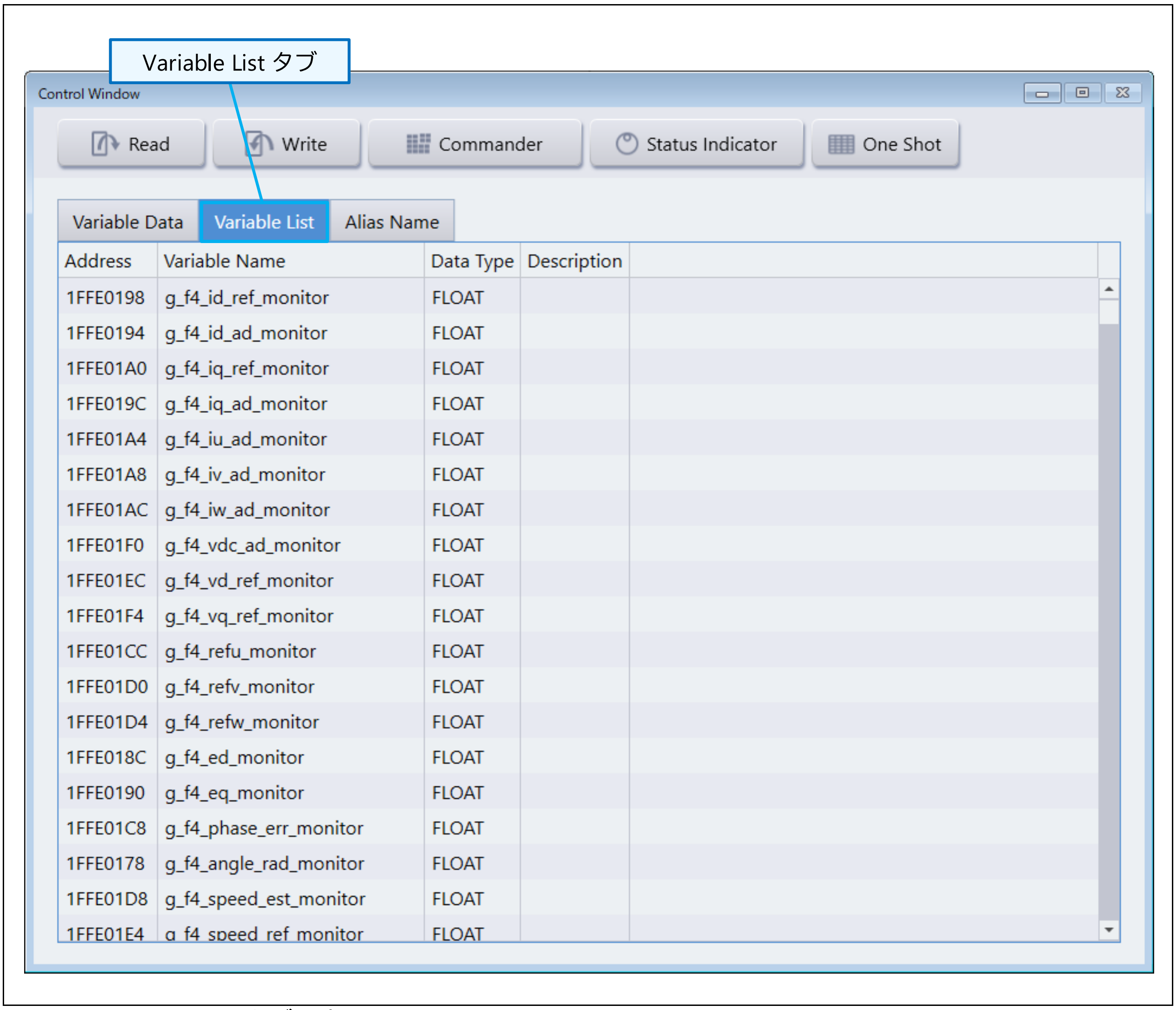


図 4-19　Variable List タブの表示

4.6 別名の設定（Alias Name タブ）

Control Window の Alias Name タブで、変数名に別名を指定することができます。別名は通常の変数名と同様に各機能の変数リストに含まれて表示されます。

Alias Name 表示が利用できる画面は、Control Window の Valiable Data タブ、変数検索のための Variable Find 画面、Commander 画面、User Button Window、Scope Window の Channel Setting 画面があります。

4.6.1.1 対象となる変数の指定

Alias Name を設定するには、まず対象となる変数を Alias Name タブの Variable Name 欄で指定します。

変数名の入力方法は、Variable Data タブの Variable Name 欄と同様の操作ができます（詳しくは「4.4.1 Variable Name の設定」を参照してください）。

4.6.1.2 Alias Name（別名）の設定

Alias Name タブの Alias Name 欄でその変数の別名を付けます。

4.6.1.3 Alias Name（別名）の有効／無効の設定

Alias Name タブの On/Off 欄で、別名設定の有効無効を指定します。

チェックボックスを On にすると有効、Off にすると無効になります。リスト内には同じ Variable Name や Alias Name を複数設定することもできますが、その場合は 1 つのみ Alias Name を On にできます。

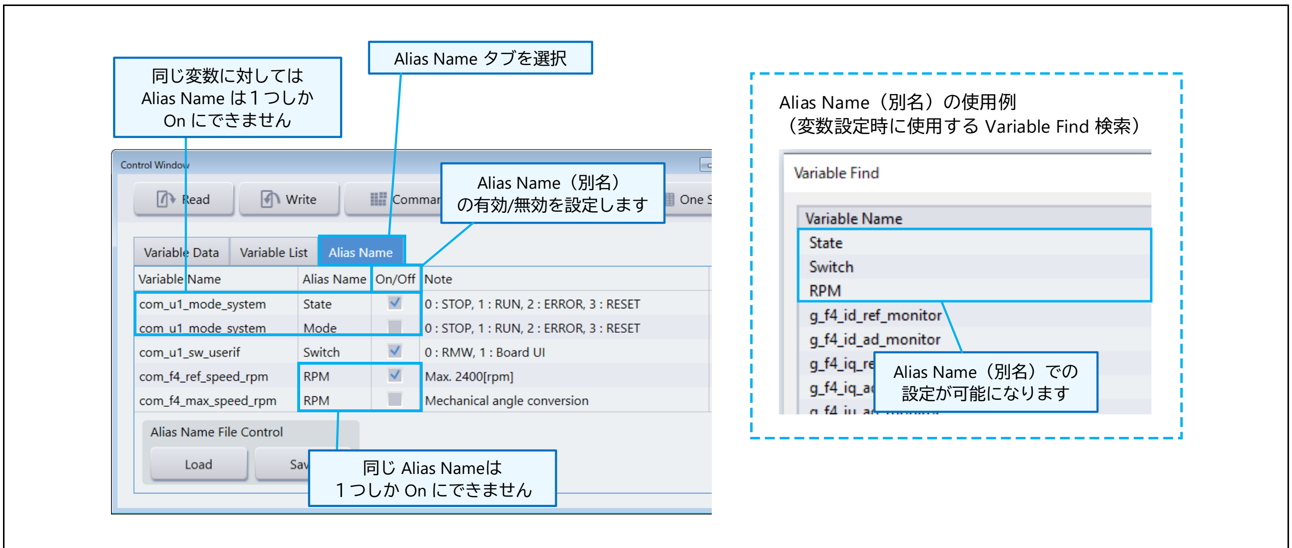


図 4-20 Alias Name（別名）の有効/無効の設定

備考 他の機能で Alias Name（別名）の指定がある状態において、Alias Name の On/Off 設定を On から Off にすると、Alias Name は Variable Name（変数名）に自動で切り替わります。ただし、Off から On にした場合、自動では Alias Name に切り替わらないため、再度使用する機能側で Alias Name を指定してください。

4.6.1.4 Note の設定

Alias Name タブの Note 欄は、メモとして自由に記入できます。

4.6.1.5 別名リストの保存／読み込み

Alias Name タブの設定情報は、カンマ区切りの CSV ファイル形式で保存（Save）／読み込み（Load）ができます。CSV ファイルを読み込むと、Alias Name タブの情報は上書きされます。

なお、RMT ファイルの保存を行うと Alias Name タブの情報も保存されます。

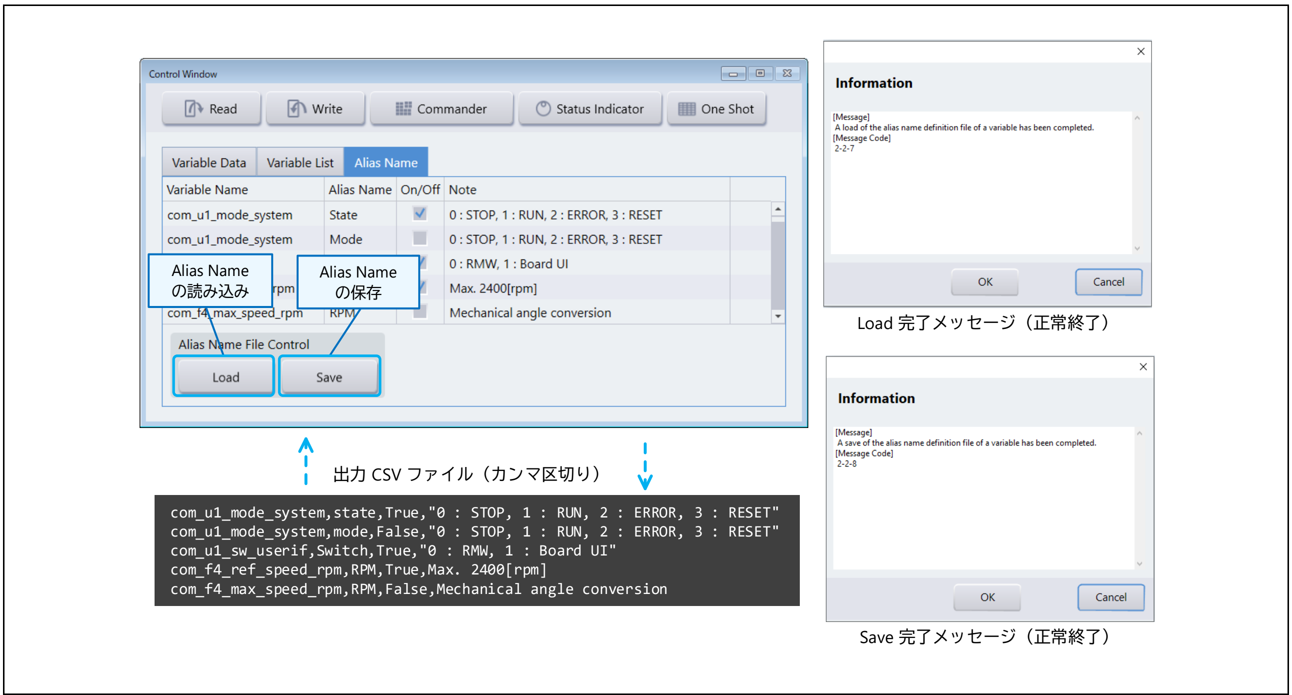


図 4-21 別名リストの保存/読み込み（CSV ファイル形式）

4.6.1.6 Alias Name（別名）と Variable Name（変数名）の表示切り替え

Alias Name を設定すると、他の機能で変数名を指定する際に Alias Name を利用できるようになります。

Alias Name 表示が利用できる画面（変数検索の Variable Find 画面は除く）では、変数名を右クリックすると、メニューに「Change Display to Variable Name」や「Change Display to Alias Name」が表示され、それらを選択すると表示を切り替えることができます。

- Change Display to Variable Name ：Alias Name から Variable Name へ表示を切り替えます。
- Change Display to Alias Name ：Variable Name から Alias Name へ表示を切り替えます。

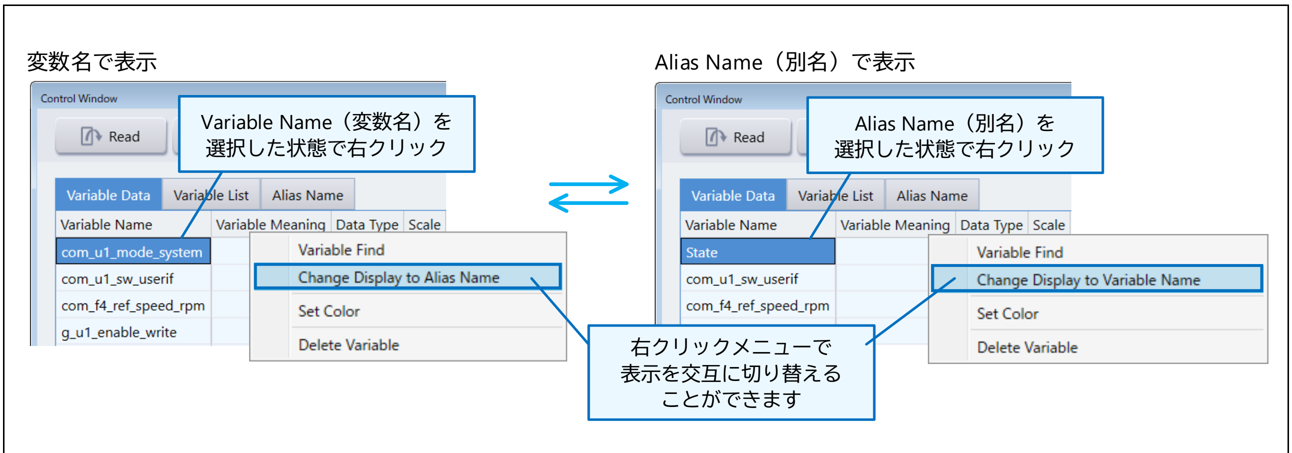


図 4-22 Alias Name（別名）と Variable Name（変数名）の表示切り替え

5. ［Analyzer］Scope Window

5.1 概要

Analyzer ツールの Scope Window は、選択した変数の値の変化をオシロスコープのように波形で表示します。Scope Window は、Analyzer ツールを起動すると表示されます。

5.2 特徴

- オシロスコープ感覚で操作が可能です（トリガ、オフセット、ズーム、カーソルなど）。
- 各チャネルのサンプリングタイミングが同期できます。
- 計測した波形情報の Save/Load や Report 出力が可能です。

5.3　　画面構成

Scope Window の画面構成と機能を以下に示します。

図 5-1　Scope Window の構成

表 5-1　Scope Window 各部の説明

No.	名称	説明
1	波形表示部	変数の値を波形で表示します。
2	チャネル情報表示部	チャネル情報を表示します。
3	計測設定部	波形の計測開始／停止、各種設定を行います。
4	トリガ設定部	波形表示画面の横軸とトリガイベント関連の設定を行います。
5	Window-Channel Setting 画面	Channel Setting ボタンをクリックすると表示され、チャネルの登録と詳細な設定ができます。
6	Image Editor 画面	Scope Capture ボタンをクリックすると表示され、波形表示のイメージを調整して pdf または画像形式で出力します。

5.3.1 波形表示部

波形表示部の構成を以下に示します。

5.3.1.1 画面表示切り替えタブ

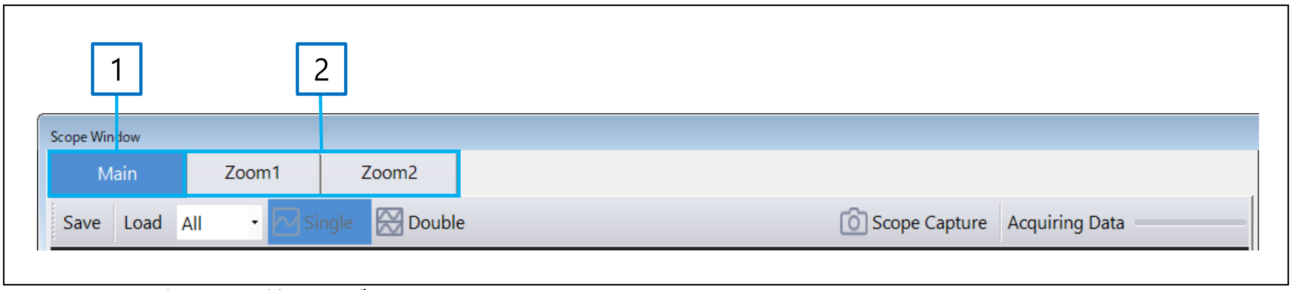


図 5-2 画面表示切り替えタブ

表 5-2 画面表示切り替えタブの説明

No.	名称	説明
1	Main タブ	Main の波形を表示します。
2	Zoom1 タブ／Zoom2 タブ	Zoom 波形を表示します。

5.3.1.2 ツールバー

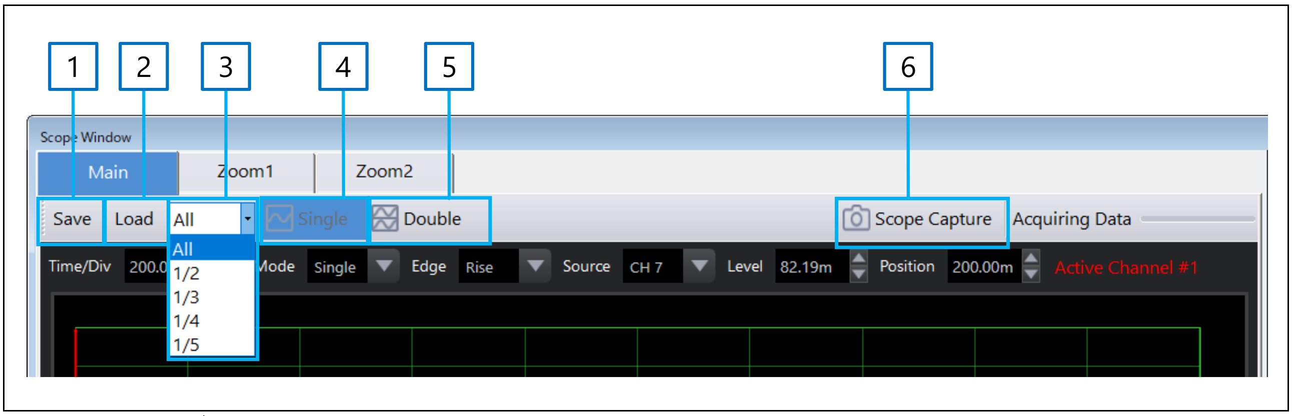


図 5-3 ツールバー

表 5-3 ツールバーの説明

No.	名称	説明
1	Save ボタン	表示されている波形情報を CSV ファイルに保存します。
2	Load ボタン	Save で保存した CSV ファイルを読み込み、波形を表示します。
3	間引き数	表示する波形データの間引き数を、リストボックス（All, 1/2, 1/3, 1/4, 1/5）から選択します。
4	Single ボタン	Main の波形を 1 画面で表示します
5	Double ボタン	Main の波形を 2 画面で表示します
6	Scope Capture ボタン	Image Editor 画面が起動します。ボタンクリック時の波形画像とチャネル情報が Image Editor へ表示され、ボタンをクリックするたびに追加します。

5.3.1.3 トリガ設定部

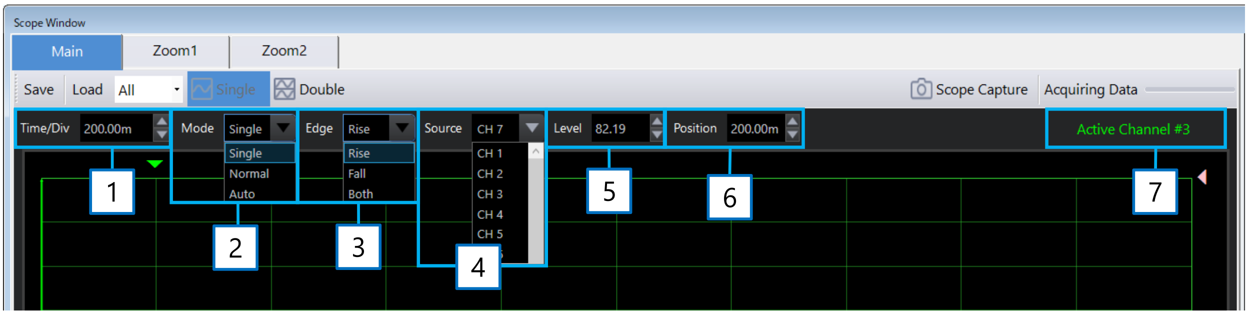


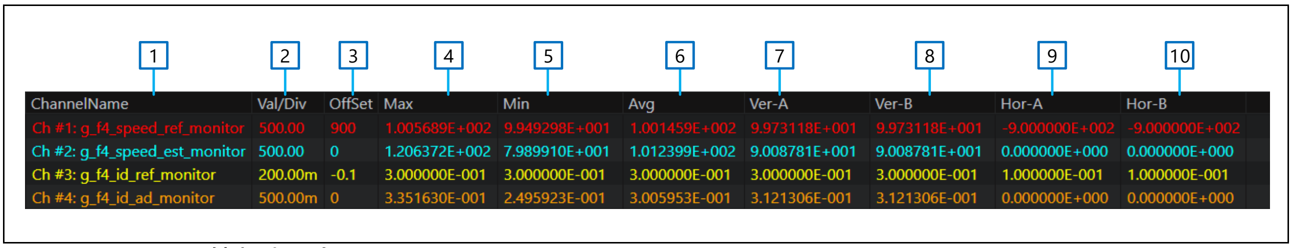
図 5-4 トリガ設定部

表 5-4 トリガ設定部の説明

No.	名称	説明
1	Time / Div	波形の横軸 1 div の時間を設定します。
2	Mode	トリガのモードを選択します。 - Auto ：一定間隔で波形を自動更新する - Single ：トリガイベント発生時に波形表示して更新を停止する - Normal ：トリガイベントが発生する度に波形表示を繰り返す
3	Edge	トリガのエッジを選択します。 - Rise ：立ち上がりエッジ - Fall ：立ち下がりエッジ - Both ：両エッジ
4	Source	トリガ設定をするチャネル番号を選択します。
5	Level	トリガの同期レベルを指定します。
6	Position	トリガの表示位置を指定します。
7	Active Channel	Active Channel に設定中のチャネル番号を表示します。

5.3.2 チャネル情報表示部

チャネル情報表示部の構成を以下に示します。



ChannelName	Val/Div	OffSet	Max	Min	Avg	Ver-A	Ver-B	Hor-A	Hor-B
Ch #1: g_f4_speed_ref_monitor	500.00	900	1.005689E+002	9.949298E+001	1.001459E+002	9.973118E+001	9.973118E+001	-9.000000E+002	-9.000000E+002
Ch #2: g_f4_speed_est_monitor	500.00	0	1.206372E+002	7.989910E+001	1.012399E+002	9.008781E+001	9.008781E+001	0.000000E+000	0.000000E+000
Ch #3: g_f4_id_ref_monitor	200.00m	-0.1	3.000000E-001	3.000000E-001	3.000000E-001	3.000000E-001	3.000000E-001	1.000000E-001	1.000000E-001
Ch #4: g_f4_id_ad_monitor	500.00m	0	3.351630E-001	2.495923E-001	3.005953E-001	3.121306E-001	3.121306E-001	0.000000E+000	0.000000E+000

図 5-5　チャネル情報表示部

表 5-5　チャネル情報表示部の各列の説明

No.	名称	説明
1	Ch #n	チャネル設定した変数名が表示されます。
2	Val / Div	チャネル設定した Val/Div 値が表示されます。
3	Offset	チャネル設定した Offset 値が表示されます。
4	Max	波形表示している範囲の最大値が表示されます。
5	Min	波形表示している範囲の最小値が表示されます。
6	Avg	波形表示している値の平均値が表示されます。
7	Ver-A	Cursor（Ver-A）で設定した縦軸の値が表示されます。
8	Ver-B	Cursor（Ver-B）で設定した縦軸の値が表示されます。
9	Hor-A	Cursor（Hor-A）で設定した横軸の値が表示されます。
10	Hor-B	Cursor（Hor-A）で設定した横軸の値が表示されます。

5.3.3 計測設定部

計測設定部の構成を以下に示します。

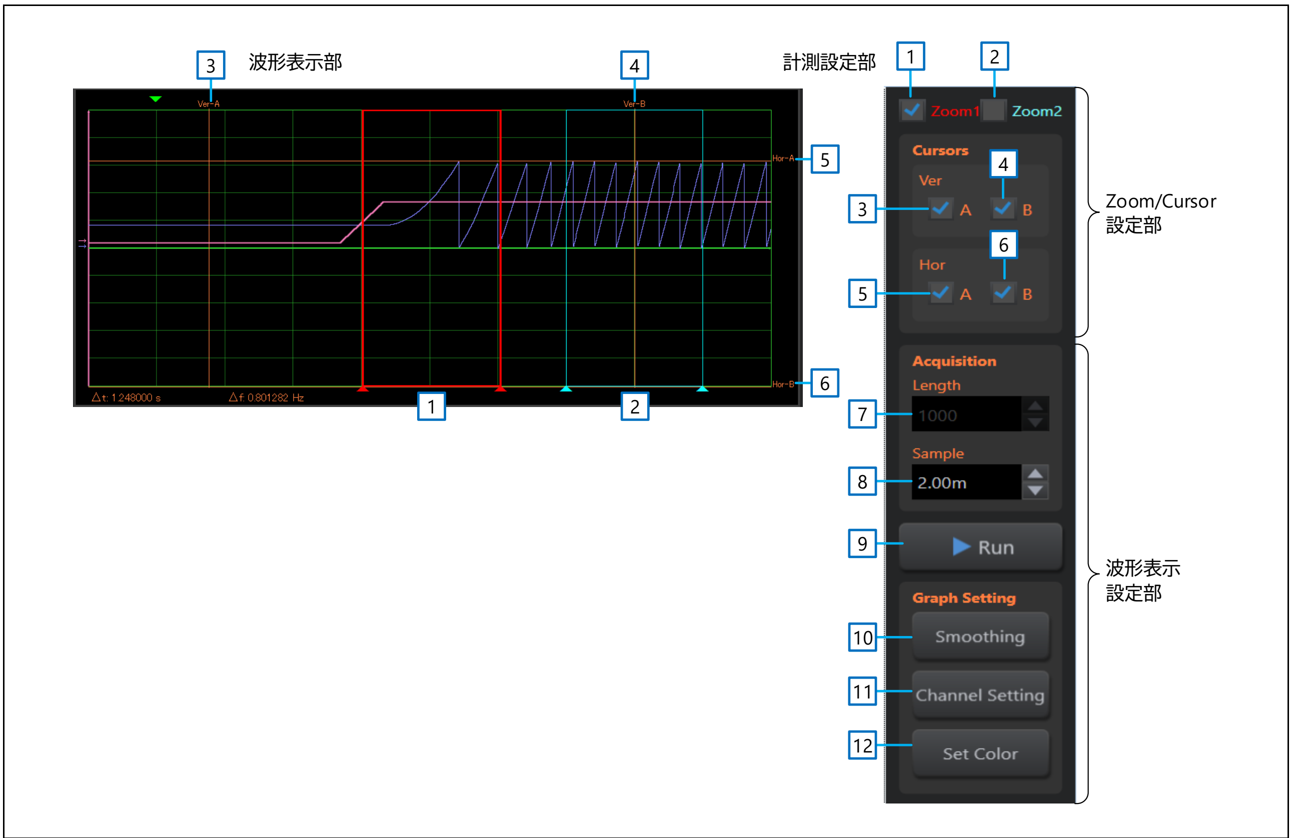


図 5-6 計測設定部

表 5-6 計測設定部の説明

No.	グループ	名称	説明
1	Zoom	Zoom1	Zoom1 に波形表示する範囲の枠を Main 波形画面に赤枠で表示します。
2		Zoom2	Zoom2 に波形表示する範囲の枠を Main 波形画面に青枠で表示します。
3	Cursors	Cursor Ver A	Cursor（Ver-A）を波形表示する各画面に表示します。
4		Cursor Ver B	Cursor（Ver-B）を波形表示する各画面に表示します。
5		Cursor Hor A	Cursor（Hor-A）を波形表示する各画面に表示します。
6		Cursor Hor B	Cursor（Hor-B）を波形表示する各画面に表示します。
7	Acquisition	Length	横軸レンジとサンプリング周期の値で変化し、値は自動計算して表示します。（Length ＝ Time/Div ÷ Sample）
8		Sample	波形表示で取得するデータのサンプリング周期を指定します。
9	RUN	RUN / STOP	RUN 表示をクリックすると波形表示を開始し、STOP 表示をクリックすると波形表示を停止します。ボタンの表示は波形表示中は「STOP」、波形表示停止中は「RUN」となります。
10	Graph Setting	Smoothing	ON にすると波形を滑らかに表示します（クリックで ON/OFF 切り替え）
11		Channel Setting	クリックで Window-Channel Setting 画面を表示します。
12		Set Color	クリックで波形表示画面の背景色を設定できます。

5.3.4 Window-Channel Settings 画面

Window-Channel Settings 画面は、Channel Setting 部と Math Channel Setting 部で構成されます。Window-Channel Settings 画面の構成を以下に示します。

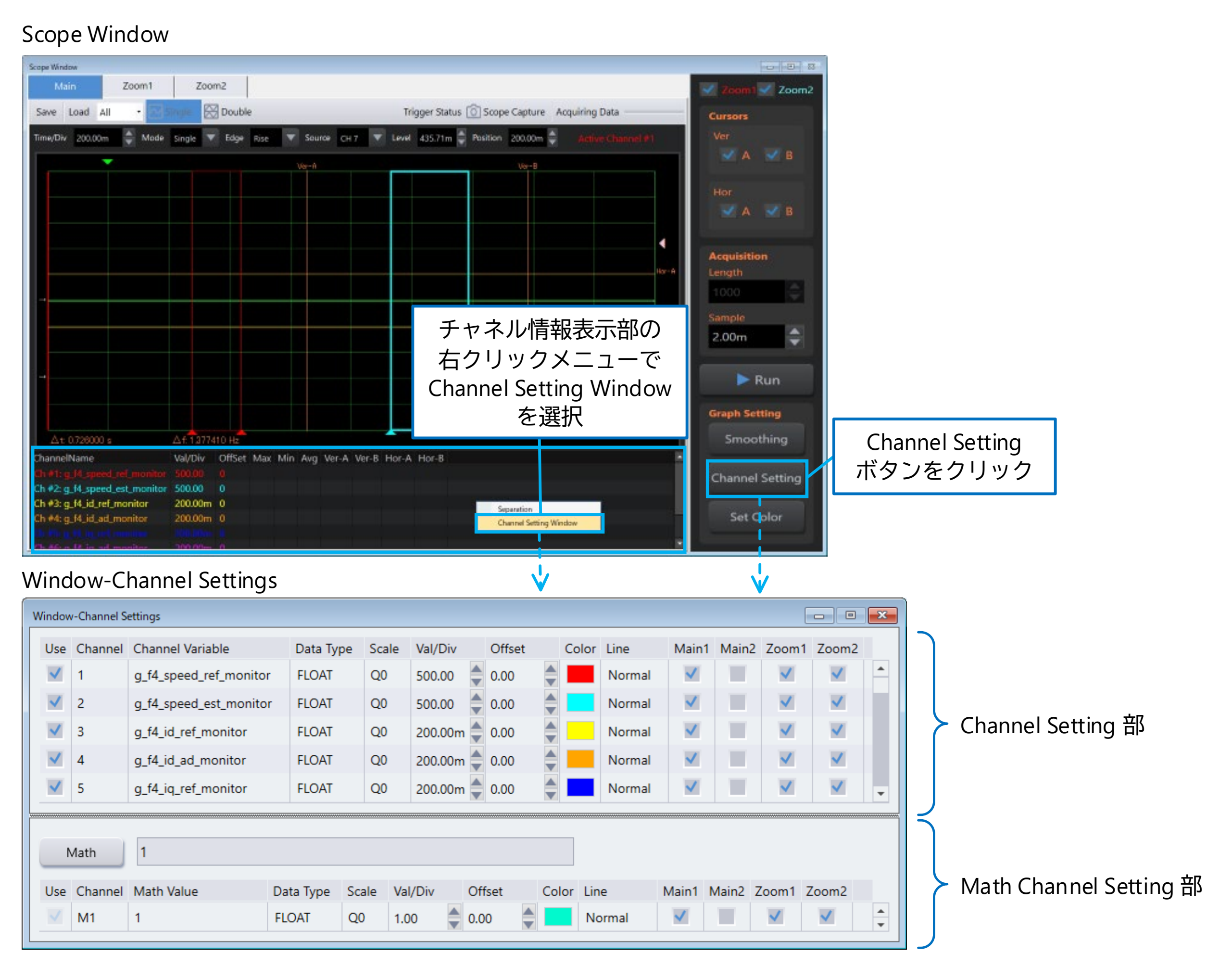


図 5-7 Window-Channel Settings 画面

5.3.4.1 Channel Setting 部

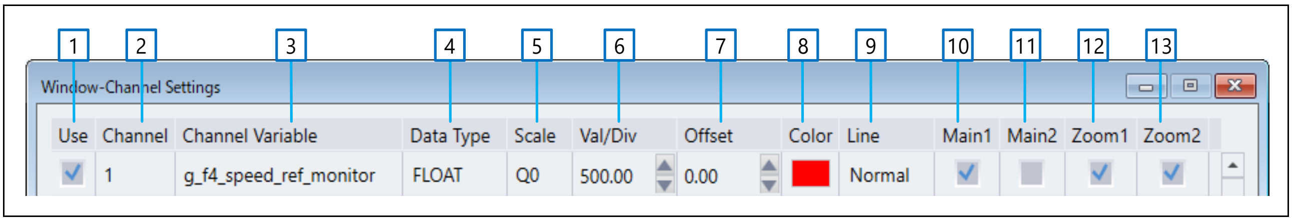


図 5-8 Channel Setting 部

表 5-7 Channel Setting 部の説明

No.	名称	説明
1	Use	波形表示するチャネルを選択します。ON/OFF 操作ができます。
2	Channel	チャネル番号を表示します。編集はできません。
3	Channel Variable	波形表示する変数を指定します。
4	Data Type	変数のデータ型を選択できます。
5	Scale	表示波形の縦軸スケールを入力できます。
6	Val / Div	波形表示時の縦軸 1div の値を指定します。
7	Offset	波形表示時の縦軸オフセット値を指定します。
8	Color	表示する波形の色を選択できます。
9	Line	表示する波形の太さを選択できます。
10～13	Main1/Main2/Zoom1/Zoom2	チェックを入れた画面に、変数の値の変化を波形で表示します。

5.3.4.2 Math Channel Setting 部

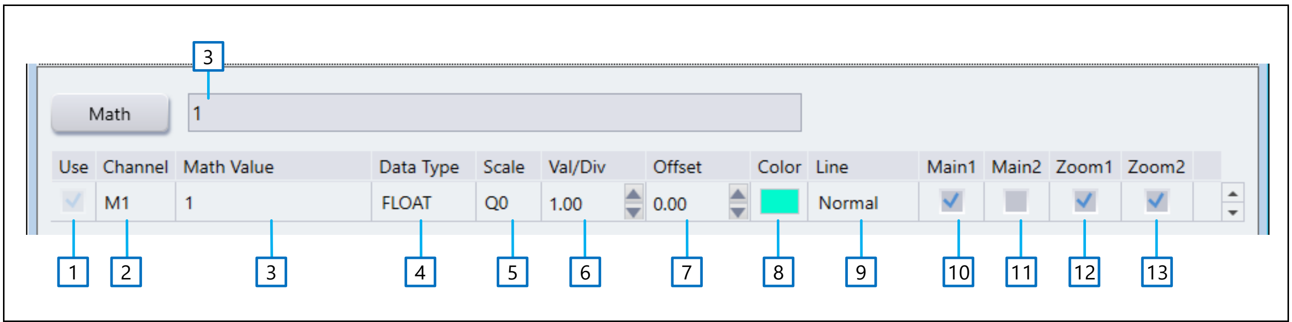


図 5-9 Math Channel Setting 部

表 5-8 Math Channel Setting 部の説明

No.	名称	説明
1	Use	表示のみで ON/OFF 操作はできません。
2	Channel	Math 用のチャネル番号（M1, M2,･･･）。番号は自動設定されます。
3	Math Value	Math 演算する式を指定します。
4	Data Type	データ型の指定ができます。
5	Scale	表示波形の縦軸スケールを入力できます。
6	Val / Div	波形表示時の縦軸 1 div の値を指定します。
7	Offset	波形表示時の縦軸オフセット値を指定します。
8	Color	表示する波形の色を選択できます。
9	Line	表示する波形の線の太さを選択できます。 - thin：細線 - Normal：標準 - thick：太線
10～13	Main1/Main2/Zoom1/Zoom2	チェックを入れた画面に、計算値の変化を波形で表示します。

5.3.5　Image Editor 画面

Scope Window 右上の「Scope Capture」ボタンをクリックすると Image Editor が起動します。Image Editor 画面の構成を以下に示します。

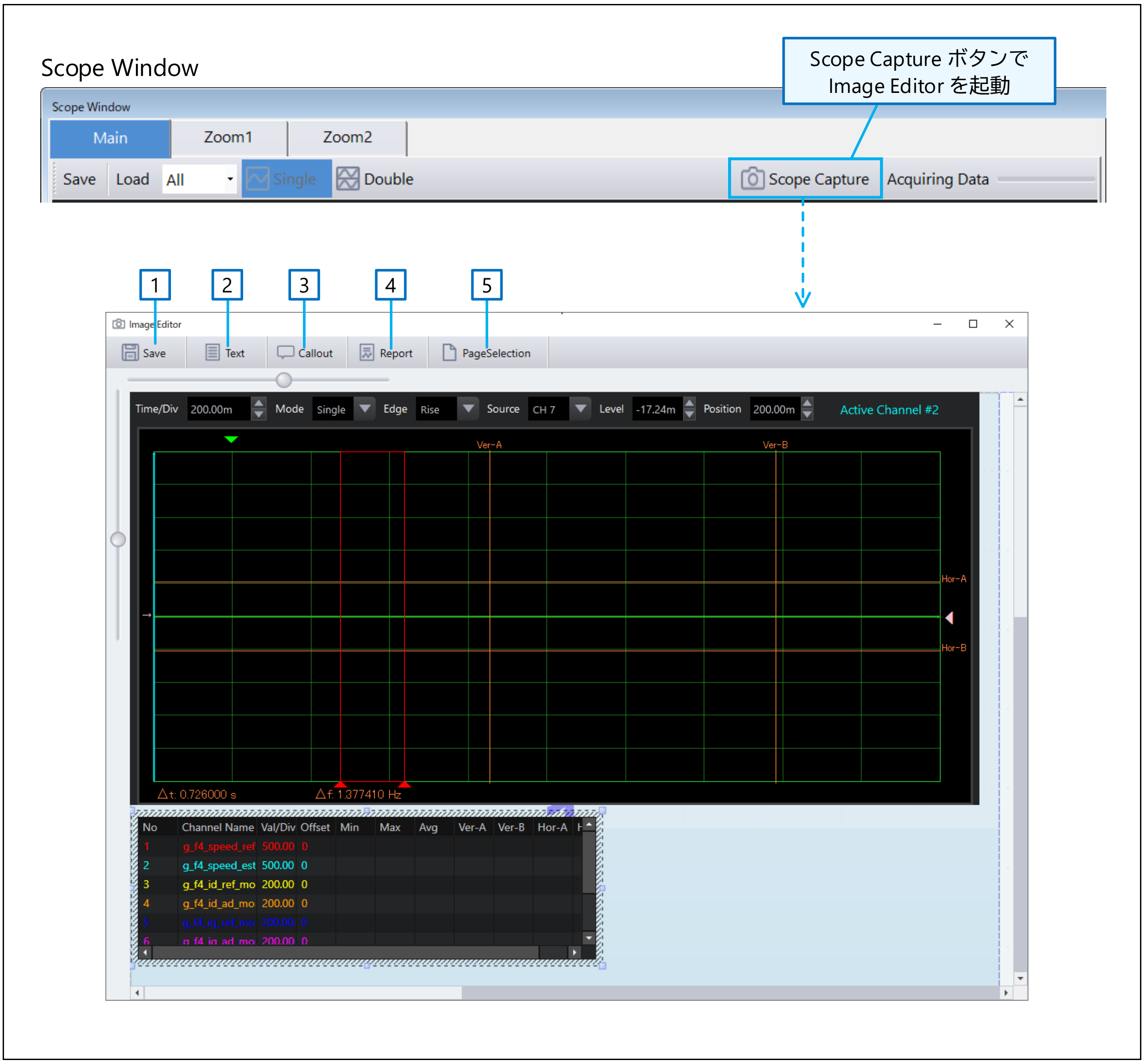


図 5-10　Image Editor 画面

表 5-9　Image Editor 画面の説明

No.	名称	説明
1	Save	編集したイメージ情報を bmp ファイルに保存します。
2	Text	オートシェイプのテキストを追加します。
3	Callout	オートシェイプの吹き出しを追加します。
4	Report	編集したイメージ情報を PDF ファイルに保存します。
5	Page Selection	このボタンをクリックすると印刷対象ページを選択するモードになります。表示されたページ番号をクリックして ON にすると印刷対象ページになります。

5.4 測定チャネルの設定

5.4.1 Window-Channel Settings の表示

Scope Window に波形を表示するには、まず Window-Channel Settings 画面の Channel Setting 部で、チャネル設定を行います。Window-Channel Settings 画面は、次のいずれかの操作で表示されます。

- Scope Window の計測設定部にある「Channel Setting」ボタンをクリックする
- Scope Window 下部のチャネル情報表示部で右クリックして、メニューから「Channel Setting Window」を選択する

Window-Channel Settings 画面の表示方法は、「5.3.4 (1) Window-Channel Settings 画面」を参照してください。

5.4.2 Channel Settings 部の設定

Channel Setting 部の各項目の設定方法を以下に示します。

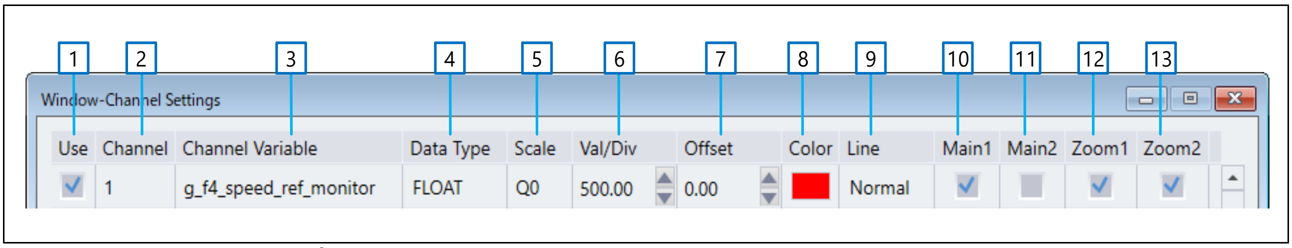


図 5-11 Channel Setting 部

(1) Use

Use は、チェックボックスを ON にするとチャネル情報が有効となります。無効な Channel 情報はグレーアウトされます。有効にできる最大チャネル数は接続する通信ボードによって変化します。

(2) Channel

Channel には、1 から始まるチャネルが割り当てられています。この欄でチャネル番号を変更することはできません。実際に利用するチャネルは、Use のチェックボックスを ON にする必要があります。

(3) Channel Variable

Channel Variable で波形表示する変数名を設定します。変数名の入力方法は、他の機能と同じです（「4.4.1 Variable Name の設定」を参照してください。）

(4) Data Type

Data Type は、変数の Data Type をリストから選択します。
・選択可能な Data Type： UINT8, INT8, UINT16, INT16, UINT32, INT32, FLOAT, BOOL, LOGIC

(5) Scale

Scale は、チャネルに設定した変数のデータ（波形データ）に対するスケール値を設定します。波形データに対して表示データには、表 5-10 の計算式が適用されます。波形データの扱いについては、5.5.2 章を参照してください。

表 5-10 Scale の入力値

設定値	データの計算式
Qn（n = 0～31）	データ × （1 / 2^n）
整数	データ × （1 / 整数）
小数	データ × （1 / 小数）

(6) Val/Div

Val/Div は、チャネル設定した変数をデータ表示する際の縦軸 1 div の値を入力します。入力ボックス右側の▼▲でも変更できます。

(7) Offset

Offset は、チャネル設定した変数をデータ表示する際の縦軸オフセットの値を入力します。入力ボックス右側の▼▲でも変更できます。

(8) Color

Color は、チャネル設定した変数の波形表示の色を選択します。

(9) Line

Line は、チャネル設定した変数の波形表示の線の太さを選択します。

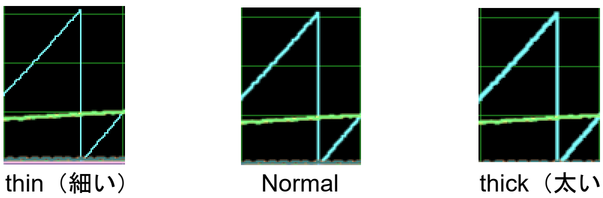


図 5-12 Line で指定できる波形表示の線の太さ

(10)～(13) Main1／Main2／Zoom1／Zoom2

(10)～(13) の欄は、チャネル設定した変数の波形表示を、どの波形表示画面に表示するのかを選択します。Main タブで表示される Main 画面（Main1 と Main2）は、どちらか一方のみ選択できます。

5.5 Math 機能

Math 機能を使用すると、チャネル設定した変数の値に対して指定した演算を行い、その結果を Scope Window に波形として表示させることができます。

5.5.1 Math Channel Setting 部の設定

Math 機能を使った波形を表示するには、Channel Setting 画面の Math Channel Setting 部で Math 用チャネルを登録します。

Channel Setting 画面の表示方法は、「5.4.1 Window-Channel Settings の表示」を参照してください。

Math Channel Setting 部の右クリックメニューから「MathChannel Add」を選択すると、Math 用 Channel の行追加ができます。追加した行では、同様の操作で行の挿入や削除もできます。

チャネル番号はシステムが自動採番します。Math 用チャネルの表示の並び替えはできません。

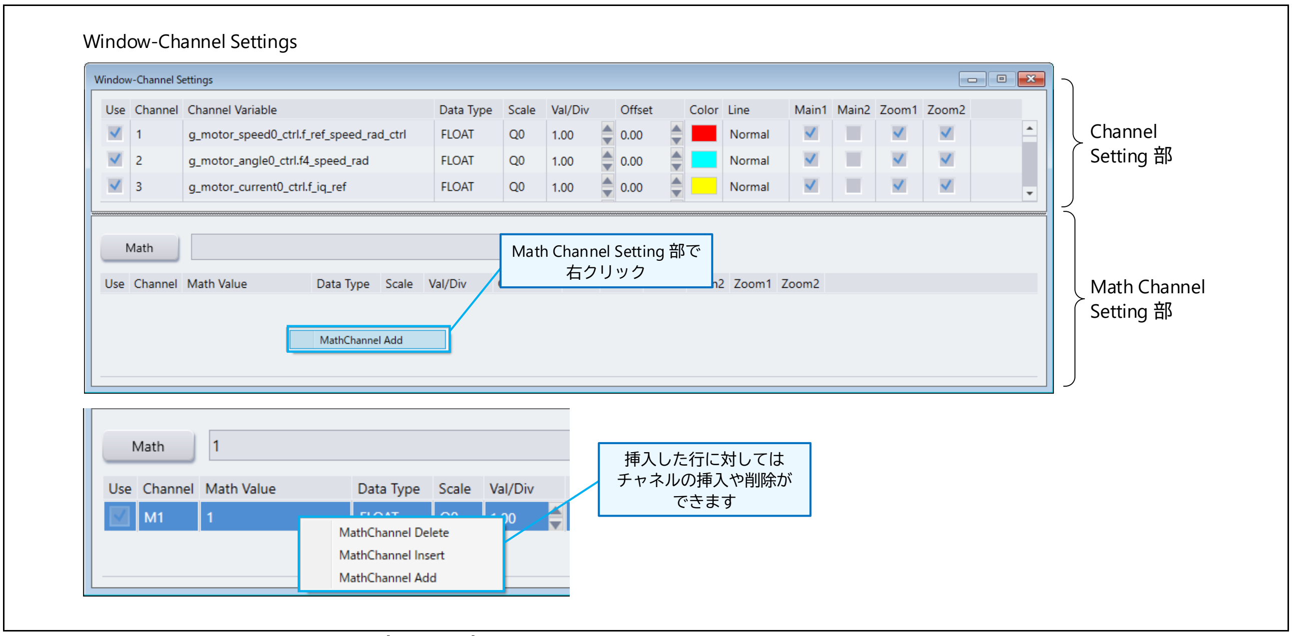


図 5-13 Math Channel Setting 部の設定

Math Channel Setting 部の各項目の設定方法を以下に示します。

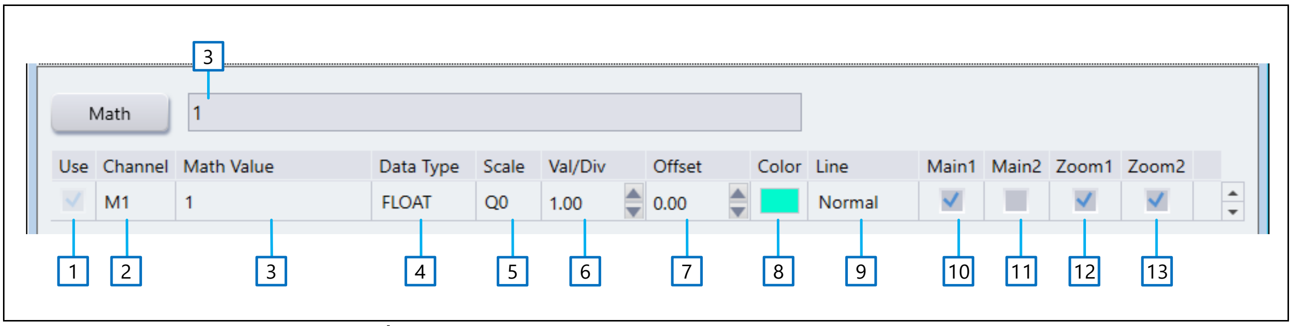


図 5-14　Math Channel Setting 部

(1)　Use

Use は、Math 用チャネルを追加した時点でチェックボックスが ON になります。ユーザが操作することは出来ません。

(2)　Channel

Channel には、M1 から始まるチャネルが割り当てられます。チャネル番号を変更することはできません。

(3)　Math Value

Math Value には、チャネル番号、値、関数、演算式を設定します。Math Value に変数名を直接入力することはできません。また、関数と演算子には、優先順位といくつかの制約事項があります。

なお、以下の場合は、Math Value 欄が赤くなり、入力が未完または何らかの問題があることを示します。

- Math Value の入力途中、演算式が不十分または正しくない場合
- 指定チャネル番号が存在しない場合
- 指定した Math チャンネル番号が演算式より下の行にある場合

表 5-11　指定可能な値

設定値		指定方法	設定例	備考
チャネル	チャネル番号	ch(n)	ch1, ch2･･･	チャネルの値は、Scale 値を適用した値（表示している値）を使用します。
	Math チャネル番号	M(n)	M1, M2･･･	チャネルの値は、Scale 値を適用した値（表示している値）を使用します。
値	整数	-	50, −1, 0	—
	小数	-	1.5, −2.5	—

表 5-12　指定可能な関数／演算子

種別	演算	関数／演算子	設定例	優先順位
関数	遅延 注1	z^n	• z^1(ch1)+ch2 • z^−1(M1)	優先度 1 （高）
	逆正接 2 注2	Atan2(ch(n),ch(n))	• Atan2(ch1,ch2) • Atan2(2,3)　(= 0.9827937)	優先度 2
	正弦 注3	sin()	• sin(ch1) • sin(60)　(= −0.3048106)	
	余弦 注3	cos()	• cos(ch1) • cos(60)　(= −0.952413)	
	正接 注3	tan()	• tan(ch1) • tan(60)　(= 0.3200404)	
	逆正接 注3	Atan()	• Atan(ch1) • Atan(60)　(= 1.554131)	
	平方根	sqrt()	• sqrt(2)　(= 1.414214)	優先度 3
演算子	べき乗	^	• 2^3　(= 8)	優先度 4
	乗算 注4	*	• ch1*7	優先度 5
	除算	/	• ch1/7	
	加算	+	• ch1+5	優先度 6 （低）
	減算	-	• ch1−8	

注 1. 指定データが存在しない場合は計算結果 0 とし、z^0(ch1) の場合は ch1 と同義とします。
2. Atan2 の引数はチャネル情報（ch1, M1 など）と数値のみで、計算式は指定できません。
3. 三角関数はラジアン（rad）で算出します。
4. 暗黙の乗算は指定できません（例：2(3+4), (1+3)(2+4), 5sin(60) など）

(4)　Data Type

Data Type は、変数の Data Type をリストから選択します。
・選択可能な Data Type：UINT8, INT8, UINT16, INT16, UINT32, INT32, FLOAT, BOOL, LOGIC

(5)　Scale

Scale は、波形データに対するスケール値を設定します。設定値により、データに対して以下の計算式が適用されます。波形データの扱いについては、次の 5.5.2 章を参照してください。

表 5-13　Scale の設定値

設定値	データの計算式
Qn（n = 0～31）	データ × （1 / 2^n）
整数	データ × （1 / 整数）
小数	データ × （1 / 小数）

(6) Val / Div

Val/Div は、チャネル設定した変数をデータ表示する際の縦軸 1 div の値を入力します。入力ボックス右側の▼▲でも変更できます。

(7) Offset

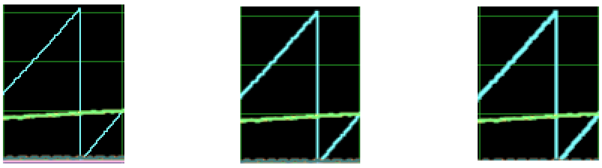
Offset は、チャネル設定した変数をデータ表示する際の縦軸オフセットの値を入力します。入力ボックス右側の▼▲でも変更できます。

(8) Color

Color は、チャネル設定した変数の波形表示の色を選択します。

(9) Line

Line は、チャネル設定した変数の波形表示の線の太さを選択します。



thin（細い） Normal thick（太い）

図 5-15 Line で指定できる波形表示の線の太さ

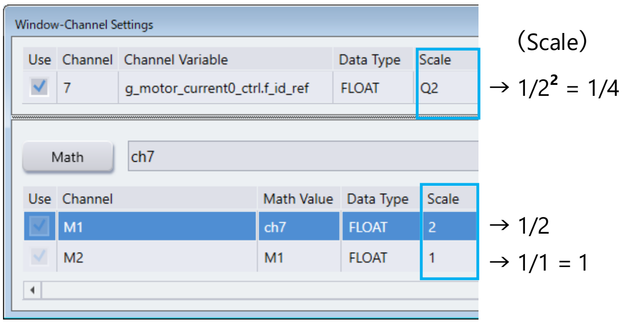
(10)～(13) Main1／Main2／Zoom1／Zoom2

(10)～(13) の欄は、チャネル設定した変数の波形表示を、どの波形表示画面に表示するのかを選択します。Main タブで表示される Main 画面（Main1 と Main2）は、どちらか一方のみ選択できます。

5.5.2 波形データの扱い

サンプル取得した「波形データの値」は、Scale を考慮していない値で保持します。一方、Scope Window に表示する波形は、Scale を適用した値で表示します。

Math 用チャネルの Math Value 欄でチャネル番号（ch7 など）または Math チャネル番号（M1 など）を参照した場合は、参照元の波形表示の値（Scale を適用した値）を波形データの値として使用します。



例：g_motor_current0_ctrl.f_id_refの 波形データが 4 の場合

Channel	波形データの値	Scale	波形表示の値
ch7	4	1/4	1
M1	1	1/2	0.5
M2	0.5	1	0.5

図 5-16 波形データの扱い

5.5.3 Math チャンネル波形の描画

Channel Setting 画面から Math 用チャネル情報を編集し、「Math」ボタンをクリックすると Math 用チャネルの波形を描画します。すでに表示していた場合は再描画します。

5.6 波形の計測

5.6.1 トリガ設定部

Scope Window に波形表示する際にトリガ設定を行います。トリガ設定部の操作方法を以下に示します。

5.6.1.1 Time/Div

Scope Window の上部の Time/Div は、波形表示画面の横軸 1 div の時間を指定します。

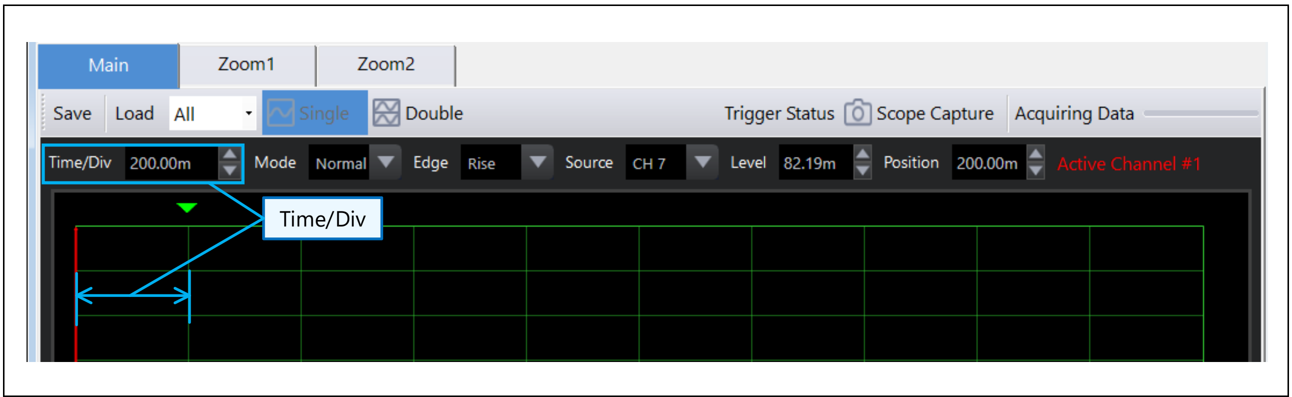


図 5-17 Time/Div 選択

5.6.1.2 Mode

Scope Window の上部の Mode は、波形表示を開始後、トリガとなるイベント発生時の動作モードの選択ができます。

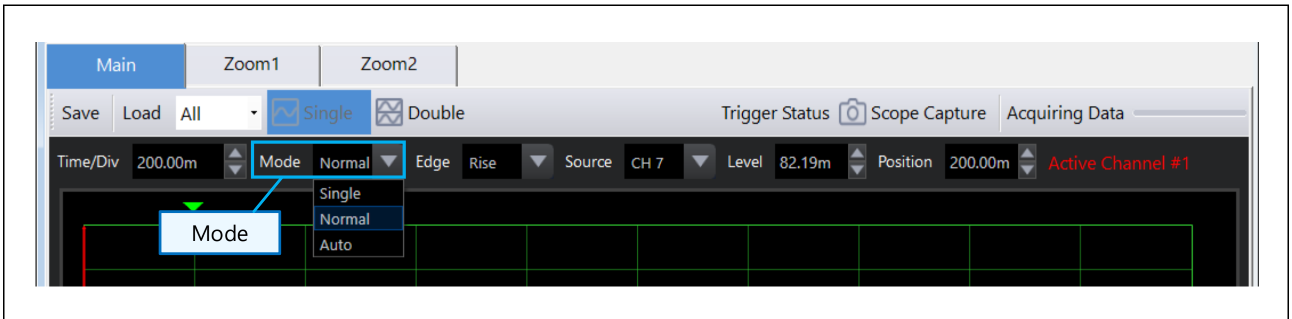


図 5-18 Mode 選択

表 5-14 Mode 一覧

名称	説明
Single	トリガイベントが発生すると波形を表示した後、波形表示の更新を停止します。
Normal	トリガイベントが発生する度に、波形表示の更新を繰り返します。
Auto	一定間隔で波形表示の自動更新を繰り返します。

5.6.1.3 Edge

Scope Window の上部の Edge は、波形表示を開始後、波形データの立ち上がり／立ち下がり／両方のいずれかをトリガイベントに設定できます。

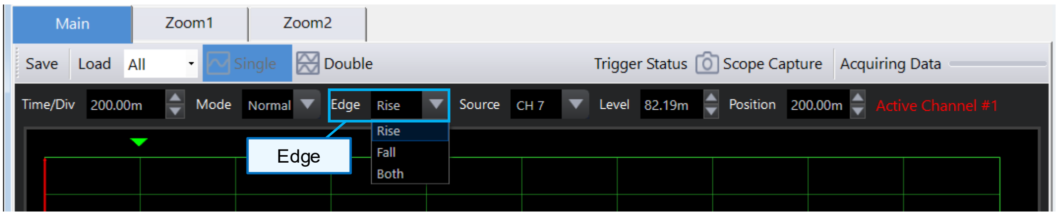


図 5-19　Edge 選択

表 5-15　Edge 機能一覧

名称	説明
Rise	立ち上がりエッジ
Fall	立ち下がりエッジ
Both	両エッジ

5.6.1.4 Source

Scope Window の上部の Source は、波形表示を開始後、トリガイベントの対象とする Channel 番号を設定します。

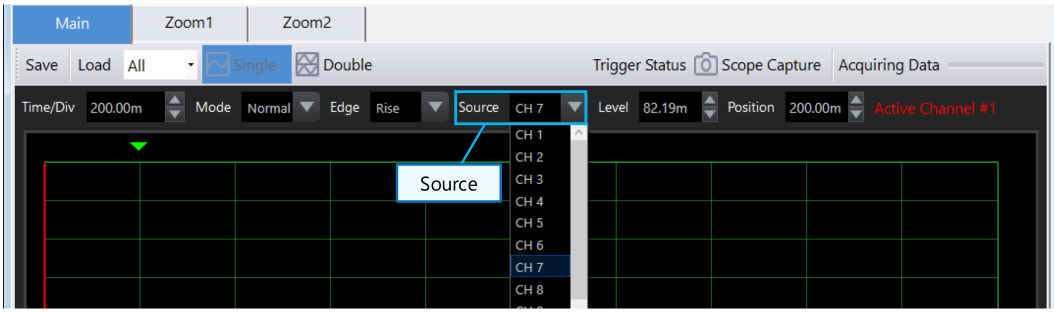


図 5-20　Source 選択

5.6.1.5 Level

Level は、波形表示開始後、トリガとなる波形データの値（レベル）を設定します。設定値は波形の右側に「◀」で表示します。位置の調整は「◀」をマウスでドラッグすることでも行えます。

5.6.1.6 Position

Position は、波形表示開始後、トリガイベントが発生したデータポイントの表示位置（横軸位置）を設定します。設定した表示位置は波形表示画面の上側に緑の「▼」で表示されます。位置の調整は「▼」をマウスでドラッグすることでも行えます。

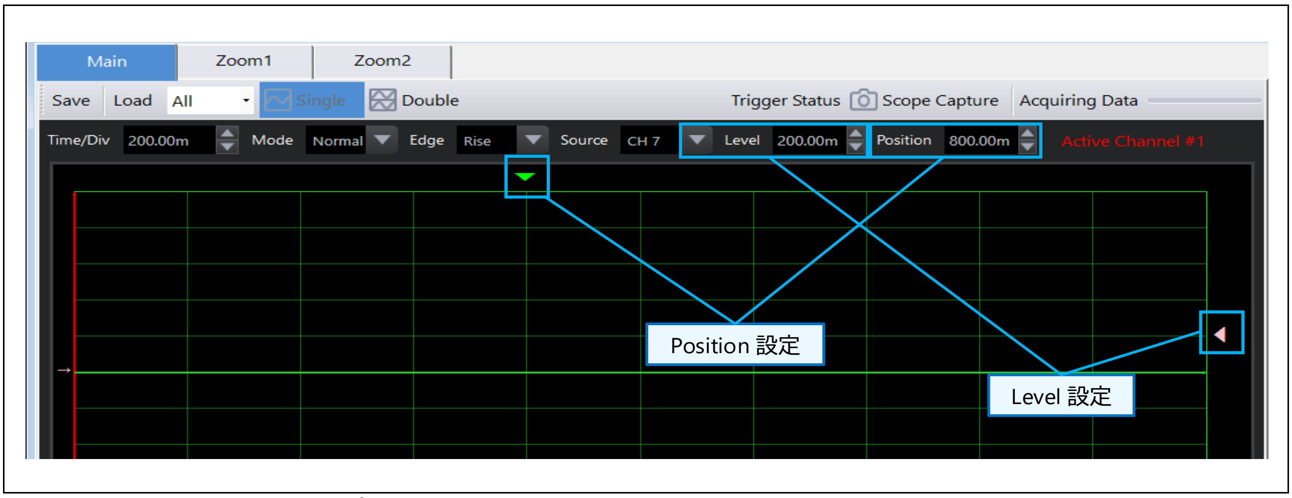


図 5-21　Level と Position 設定

5.6.2 計測設定部の Acquisition 設定

5.6.2.1 Length

横軸レンジとサンプリング周期の値で変化し、値は自動計算して表示します（表示のみ）。

Length（表示画面の横幅）= Time/Div × 10 ÷ Sample

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

5.6.2.2 Sample

波形表示で取得するデータのサンプリング周期を指定します。

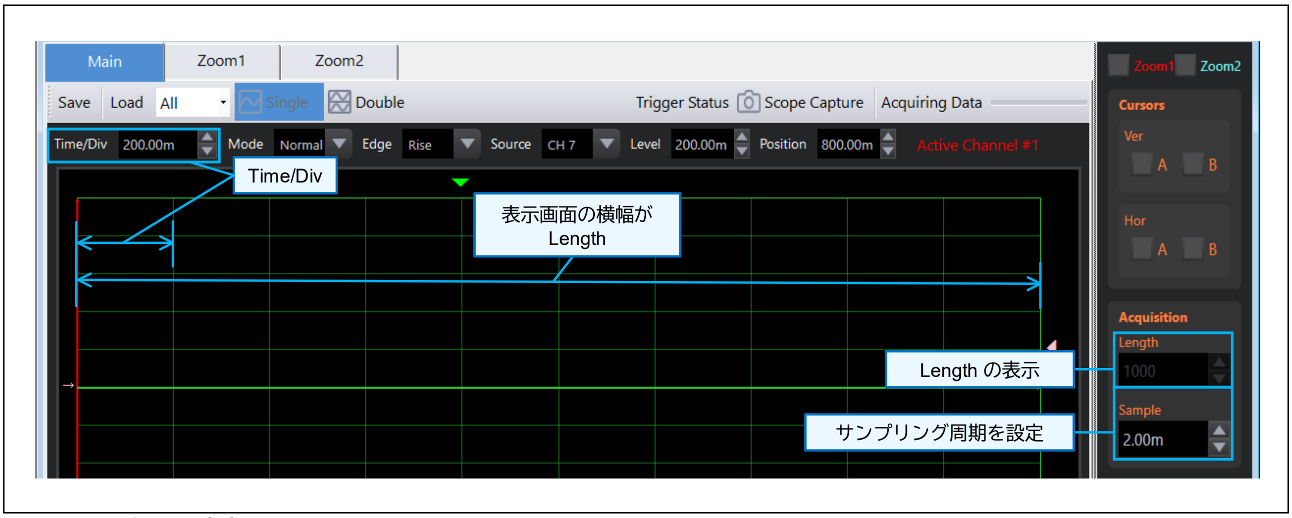


図 5-22　計測設定部の Acquisition（Length と Sample）

5.6.3 波形計測の開始と停止

波形計測の開始は「RUN」ボタンをクリックします。トリガ設定の Mode が Single 設定時は、トリガイベント後に波形計測は自動的に停止します。それ以外は「STOP」ボタンをクリックして、波形計測を停止します。

なお、波形計測を実行しながら Control Window から変数の値の読み込み、または変数へ値の書き込みを行うと、Control Window の処理が優先されるため、波形計測した結果の表示更新が遅れることがあります。

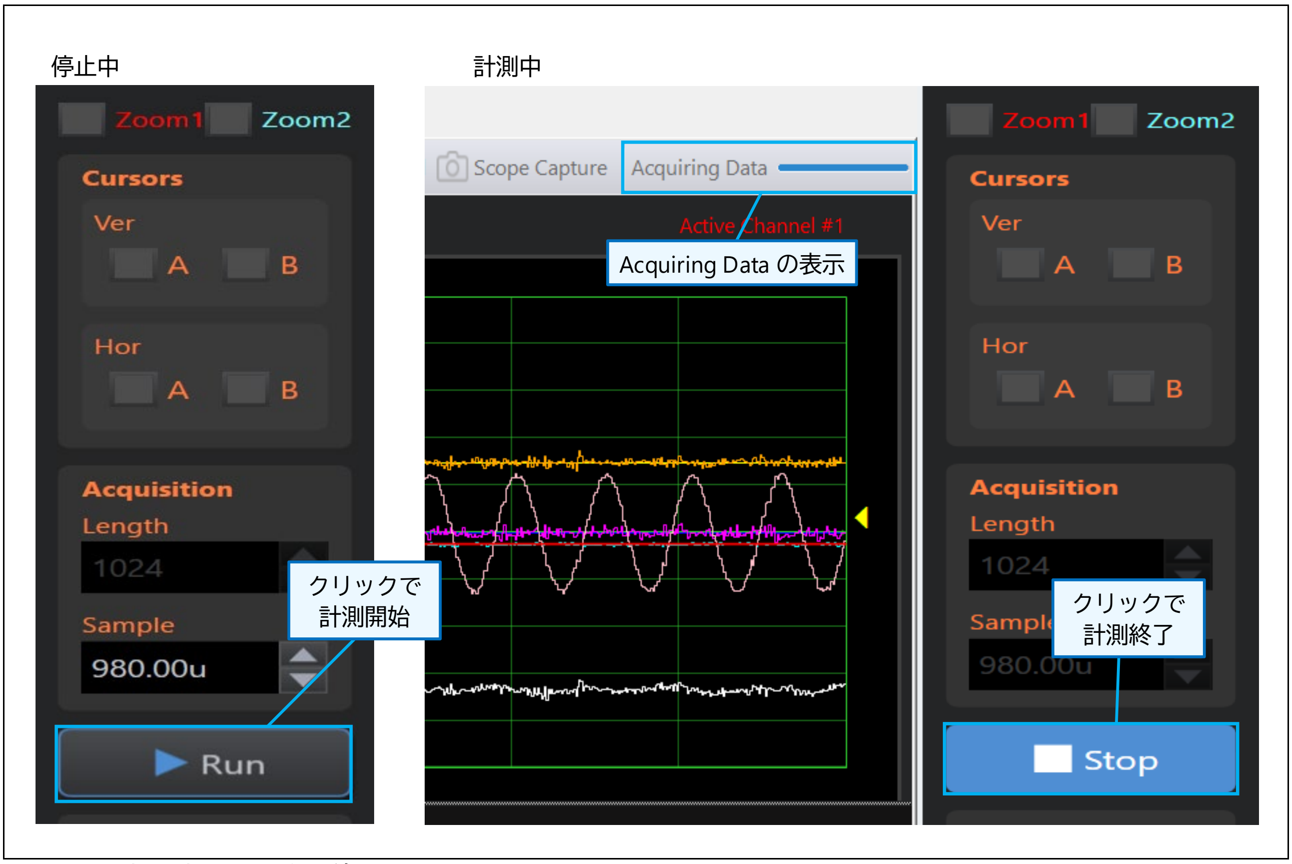


図 5-23 波形計測の開始と停止

表 5-16 RUN/STOP ボタンの表示

ボタン表示	動作
RUN	クリックで波形計測を開始します（波形計測を開始すると、ボタン表示は「STOP」になります）。
STOP	クリックで波形計測を停止します（波形計測を停止すると、ボタン表示は「RUN」になります）。

5.6.4 波形データの保存と読み込み

Scope Window 上部の「Save」ボタンで波形データの保存が、「Load」ボタンで波形データの読み込みができます。

保存時にファイルの種類で「*.csv1」を指定すると「*.csv」と「*.csv1」の 2 つのファイルを保存し、「*.csv」を指定すると「*.csv」のみを保存します。

「Load」ボタンをクリックすると、保存した波形データ（*.csv1 形式）の読み込みができます。

表 5-17 波形データの保存形式

保存時のファイルの種類	保存情報と扱い方
ICS_Wave Data File(*.csv1)	• 波形データのサンプリング値（*.csv）と、サンプリング値と波形計測時の設定情報の両方を含んだファイル（*.csv1）を保存します。 • 保存したファイル（*.csv1）は、RMW に Load して、保存した波形を表示することができます。
Comma Separated File(*.csv)	• 波形データのサンプリング値のみをカンマ区切りの CSV 形式（*.csv）で保存します。 • 保存したファイル（*.csv）を RMW に Load することはできません。

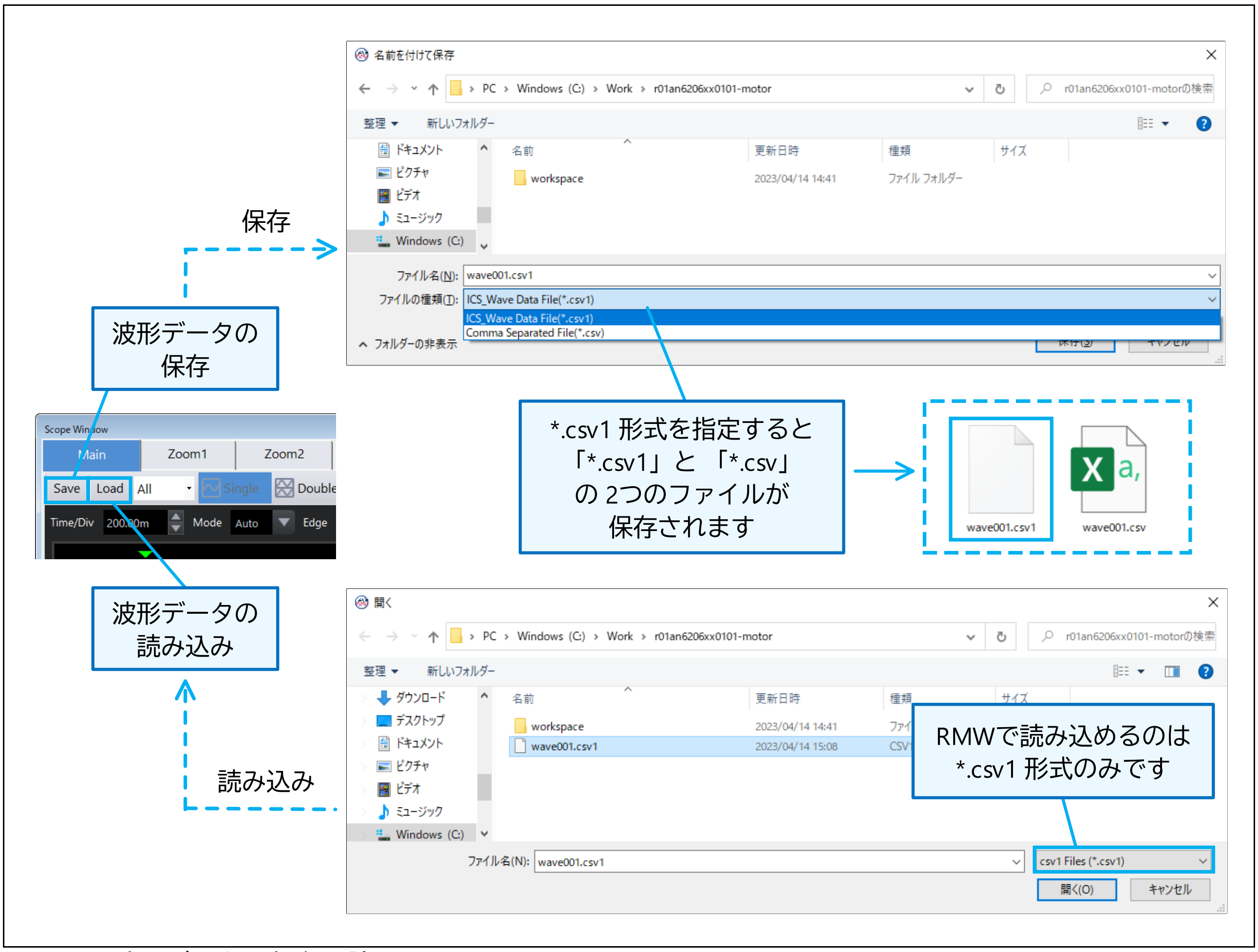


図 5-24 波形データの保存と読み込み

5.7 波形の表示

5.7.1 Scope Window の表示

Scope Window は、Analyzer ツールを起動すると自動で表示されます。

5.7.2 波形グラフの表示切り替え

Scope Window 左上のタブで表示する波形グラフの切り替えができます。

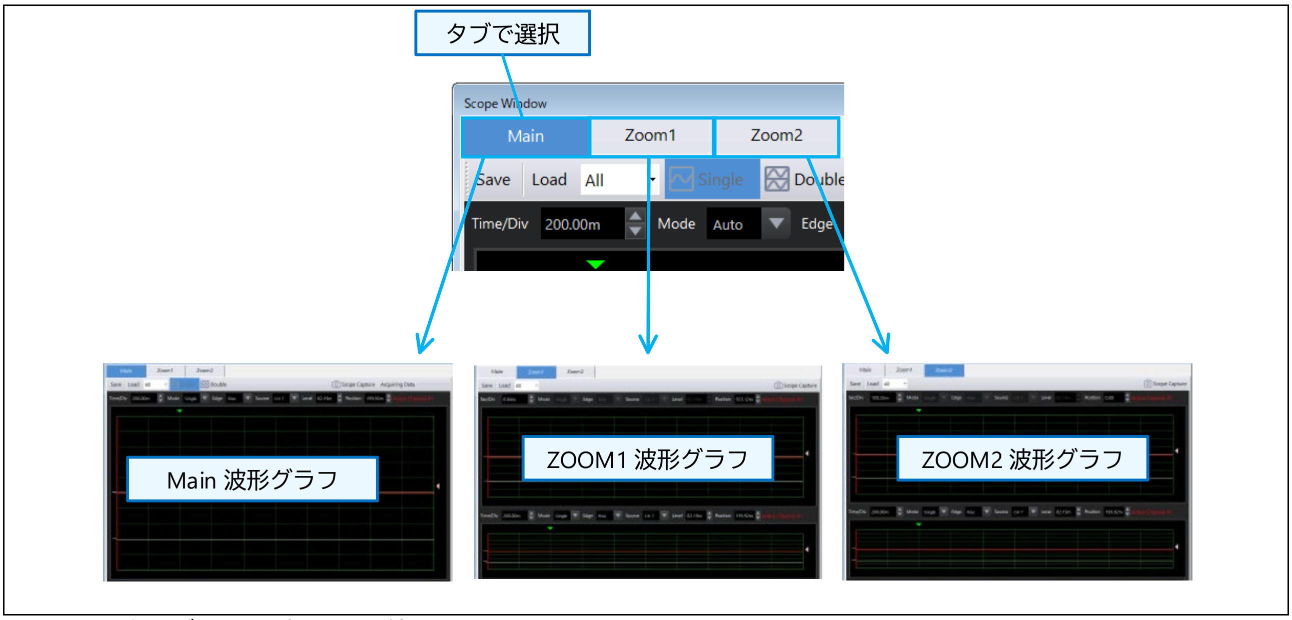


図 5-25 波形グラフの表示切り替え

5.7.3 Main1 と Main2 波形グラフの切り替え

Scope Window の Main 波形グラフは、Main1 と Main2 の 2 画面表示にすることができます。1 画面が表示された状態で、「Double」ボタンをクリックすると 2 画面に、また 2 画面が表示された状態で「Single」ボタンをクリックすると 1 画面（Main1 のみ）の表示になります。

各チャネルの波形データを Main1 と Main2 のどちらに表示するかは、Window-Channel Settings 画面で設定します。詳しくは、「5.4.1 Window-Channel Settings の表示」を参照してください。

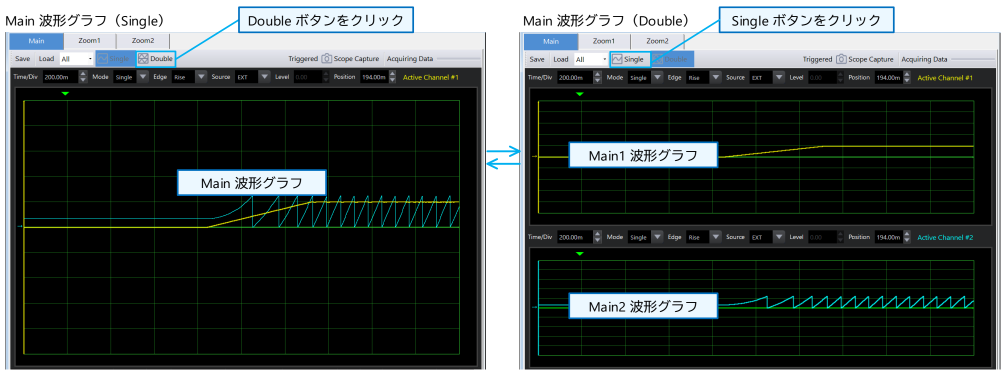


図 5-26 波形 2 画面表示

5.7.4 Zoom 波形グラフの範囲指定

Main 波形グラフで、Zoom 波形グラフに表示する範囲の指定ができます。

Scope Window 計測設定部の Zoom1、Zoom2 のチェックボックスを ON にすると「Main 波形表示画面」に Zoom1 は赤線、Zoom2 は青線で範囲枠が表示されます。

ズーム範囲の調整は、まず右下の▲のドラッグで幅を調整し、その後、左下▲のドラッグで始点を動かしてください。ズーム範囲指定が画面幅いっぱいの時は、左下の▲は動きません。

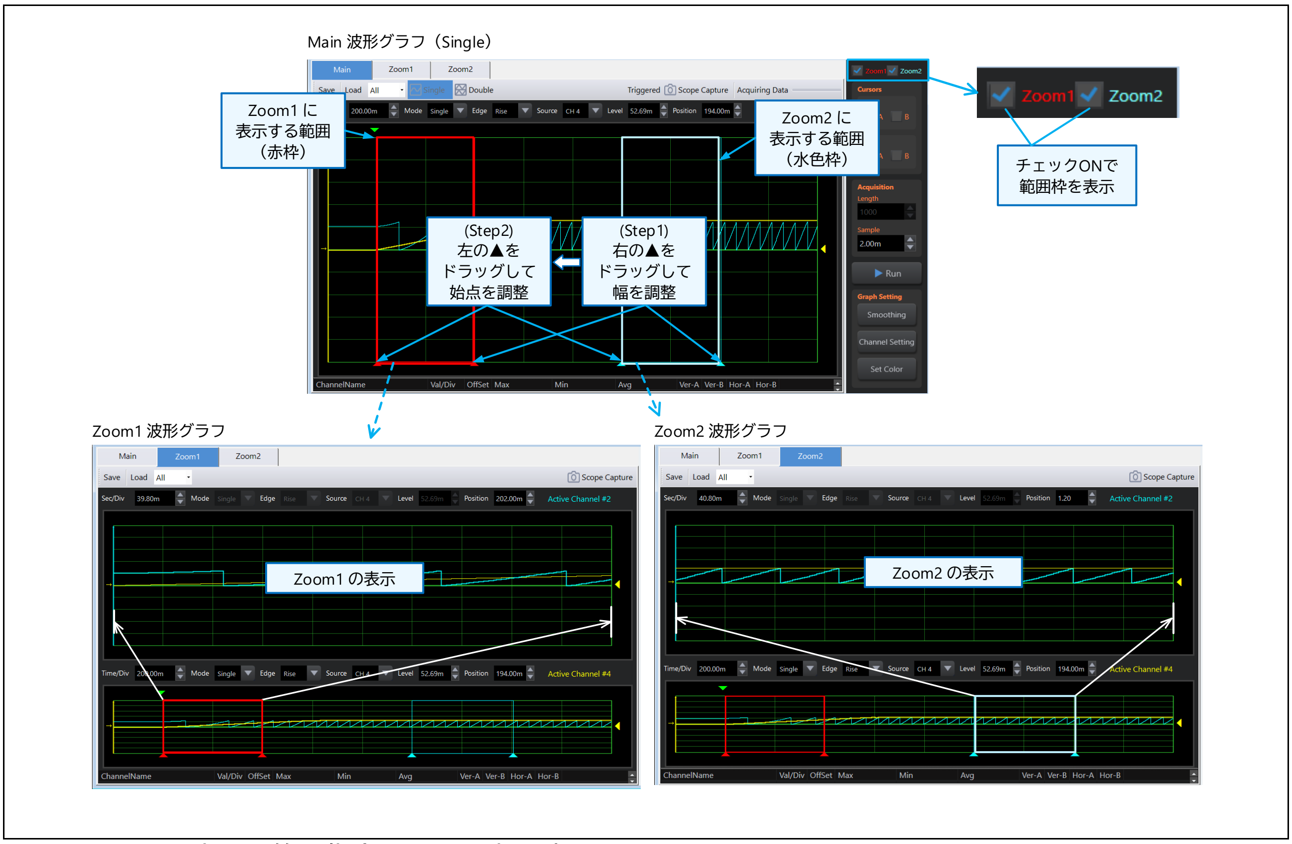


図 5-27 Zoom 表示の範囲指定（Single 表示時）

Double 表示の場合も、Zoom 波形グラフに表示する範囲を指定することができます。範囲枠下の▲（左右どちらでもよい）を右クリックすると、Main1 と Main2 の間で範囲枠を移動することができます。

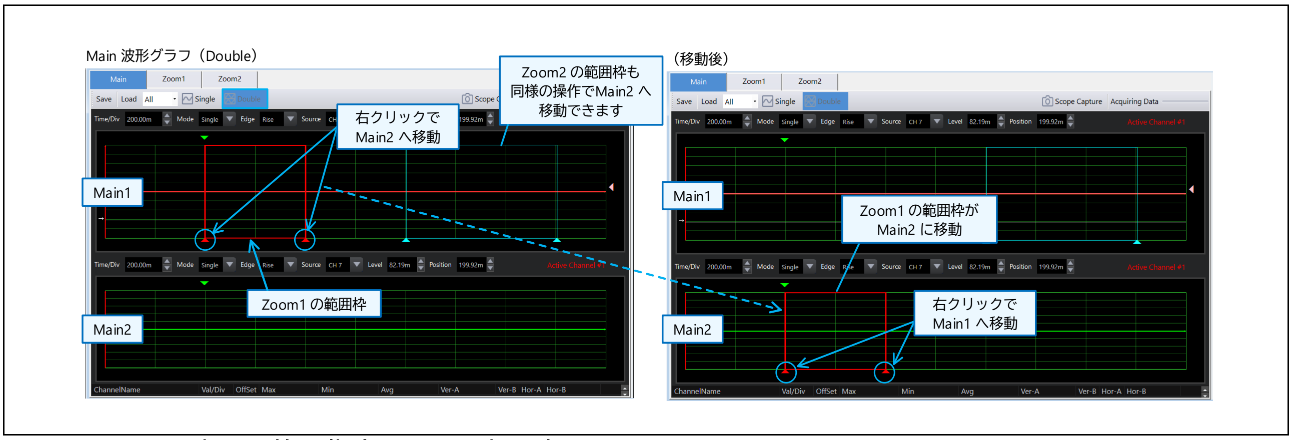


図 5-28 Zoom 表示の範囲指定(Double 表示時)

5.7.5 チャネル情報表示部の分離

Main/Zoom1/Zoom2 の各タブ画面の下部にあるチャネル情報表示部は別画面に分離することができます。チャネル情報表示部で右クリックして表示されるメニューから「Separation」を選択すると別画面に分離します。

分離したチャネル情報表示部を元に戻すには、別画面右上の「閉じる」ボタンをクリックします。また、別画面に分離しているときに、Main/Zoom1/Zoom2 のタブを切り替えると、チャネル情報表示部は元に戻ります。

分離したチャネル情報表示部には、以下の画面タイトルが表示されます。

表 5-18 分離したチャネル情報表示部のタイトル

分離前のタブ	画面タイトル
Main	MainScope ChannelData
Zoom1	Zoom1Scope ChannelData
Zoom2	Zoom2Scope ChannelData

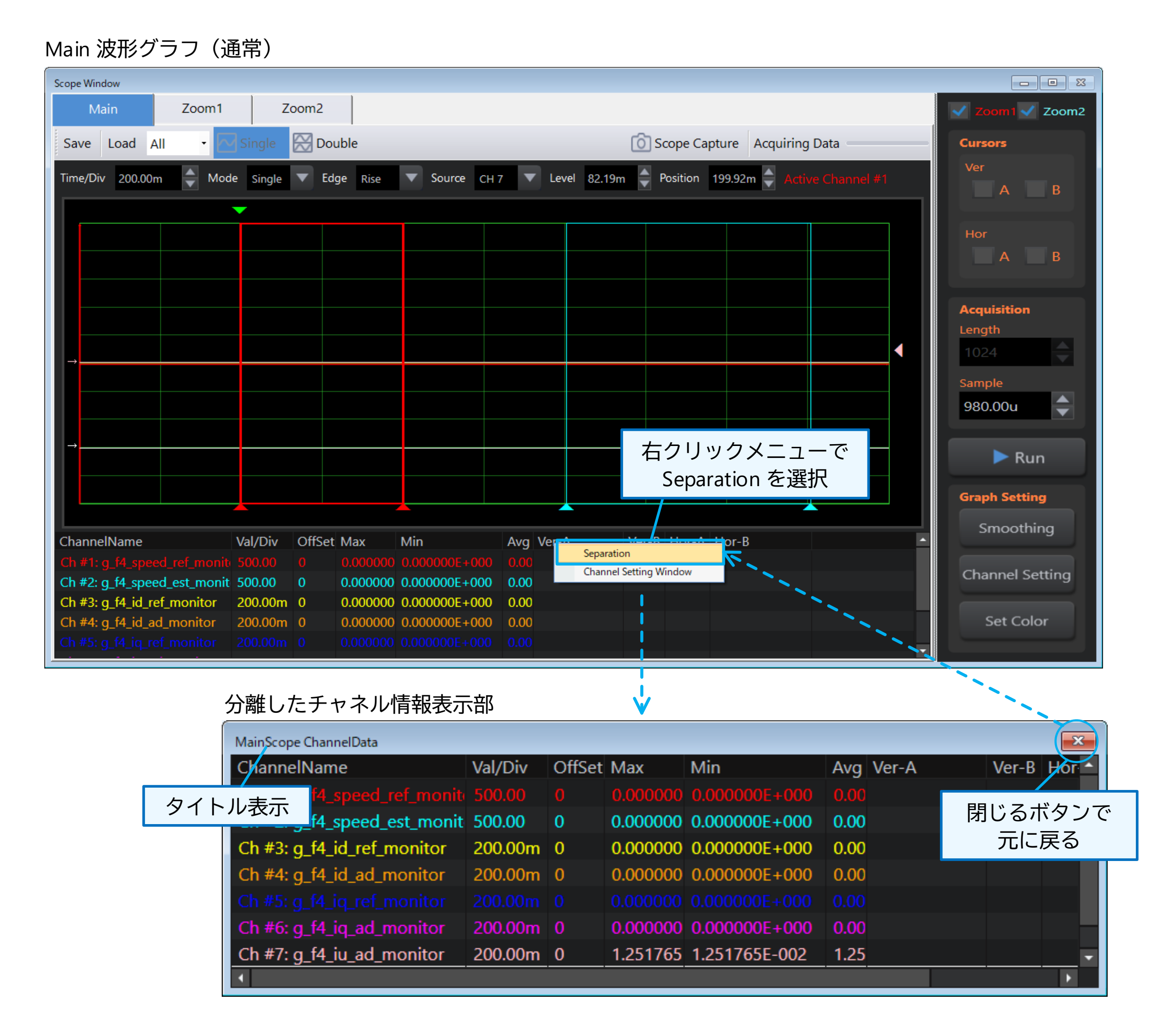


図 5-29 Channel 情報画面の分離

5.7.6　Zoom 画面からの Main 波形グラフの分離

Zoom1/Zoom2 タブ画面の Main 波形グラフを別画面に分離することができます。

分離した Main 波形グラフを元に戻すには、別画面右上の「閉じる」ボタンをクリックします。また、別画面に分離しているときに、Main/Zoom1/Zoom2 のタブを切り替えると、Main 波形グラフは元に戻ります。

分離した Main 波形グラフには、以下の画面タイトルが表示されます。

表 5-19　分離した Main 波形グラフのタイトル

分離前のタブ	画面タイトル
Main1	Main1Scope Window
Main2	Main2Scope Window

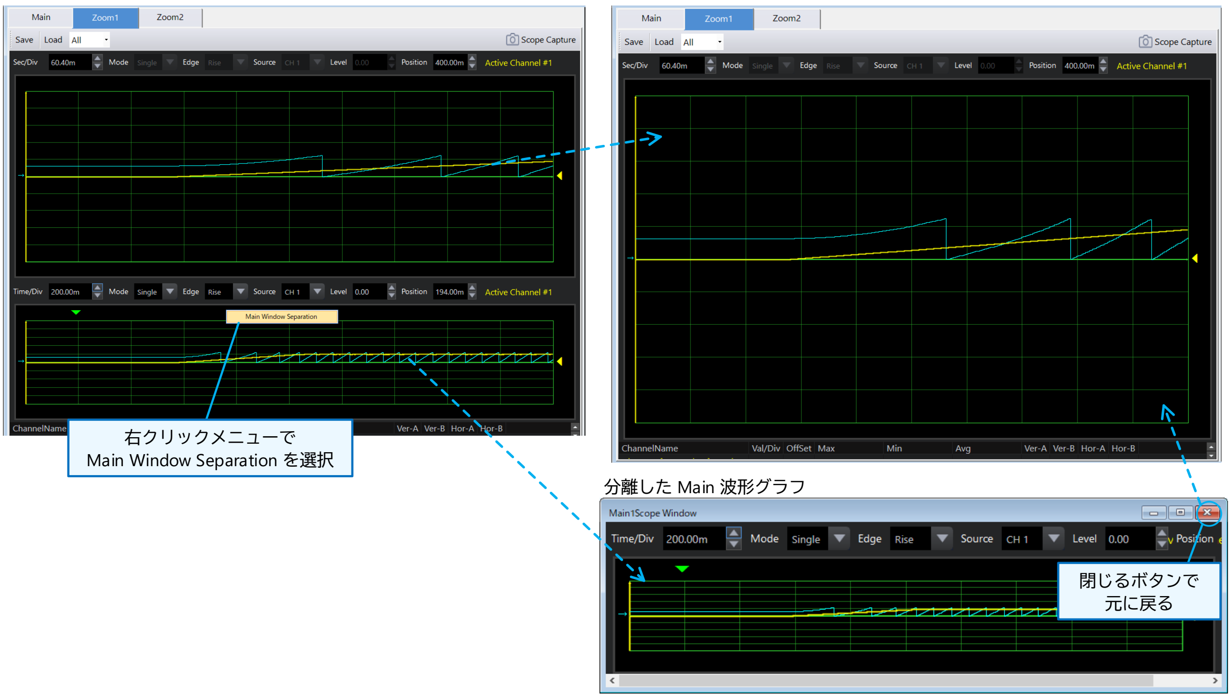


図 5-30　Main 波形画面の分離

5.7.7　Active Channel 表示

Main や Zoom のチャネル情報表示部で、チャネルを選択すると Active Channel に設定できます。波形が重なって表示されている場合、Active Channel の波形は最前面で優先表示されます。

また、グラフ左側の縦軸が、Active Channel の波形と同色になります。

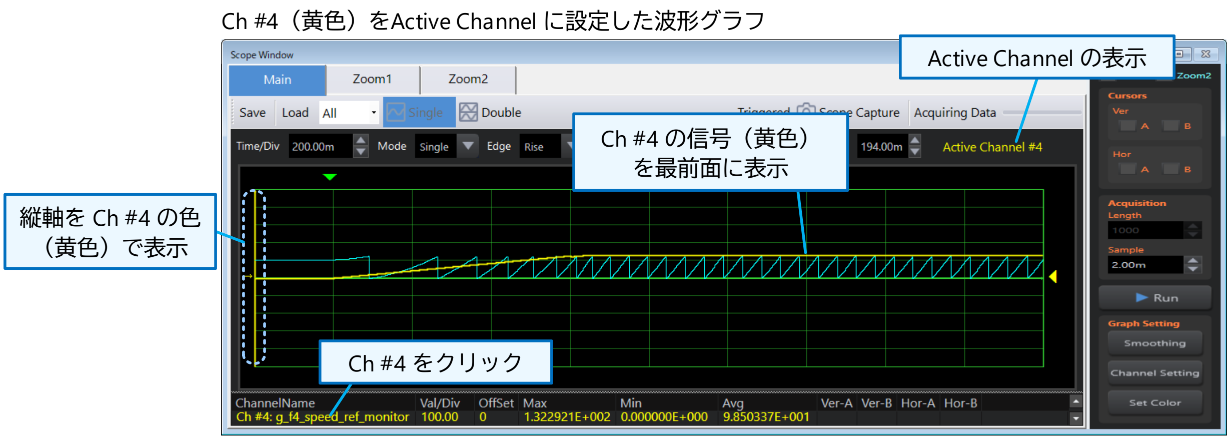


図 5-31　Active Channel と波形グラフの表示

5.7.8　カーソル表示

Scope Window 計測設定部の Cursors チェックボックスを ON にすると、Ver（垂直）、Hor（水平）各 2 本ずつグラフ上にカーソルを表示できます。画面内の各カーソル線の位置は、マウスのドラッグで調整できます。

カーソルを置いた位置の各波形の値が、画面下部のチャネル情報部に表示されます。

また、Ver（垂直）カーソルを 2 本表示すると、カーソル点間の時間と周波数が表示されます。

図 5-32　カーソル表示

5.7.9 波形表示ポイント数の間引き

Scope Window の左上リストボックスで、波形データの表示ポイント数の切り替えができます。「All」は全データを使って表示、それ以外は設定による間引き表示をします。例えば「1/5」を選択した場合、波形データの 5 ポイントのうちの 1 ポイントに間引いて波形を表示します。

ただし、波形計測中は、設定を変更することはできません。

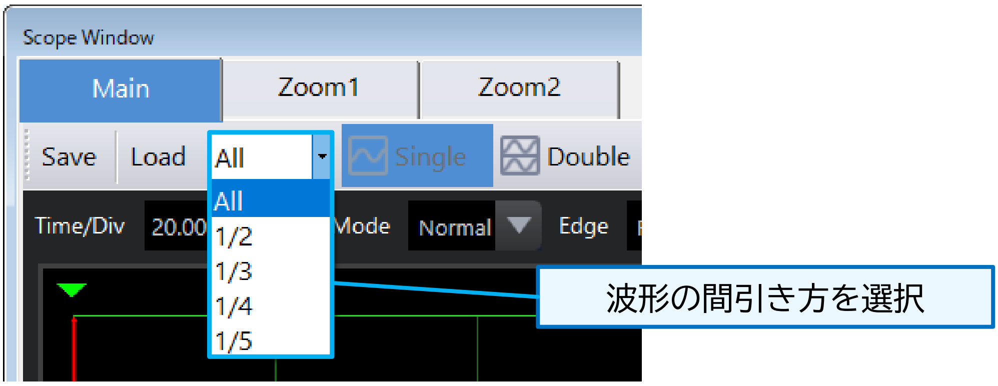


図 5-33　波形表示ポイントの設定

5.7.10 波形のスムージング

Scope Window 計測設定部の「Smoothing」ボタンで波形表示を滑らかに表示することができます。ON にすると波形を滑らかにしたグラフを表示し、OFF にすると通常の表示状態になります。

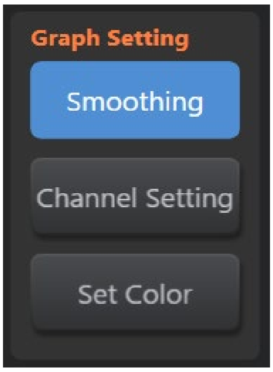


図 5-34　Smoothing ボタン（ON の状態）

5.7.11 画面背景色の変更

Scope Window の右下の「Set Color」ボタンをクリックすると、Scope Window の背景色を変更できます。

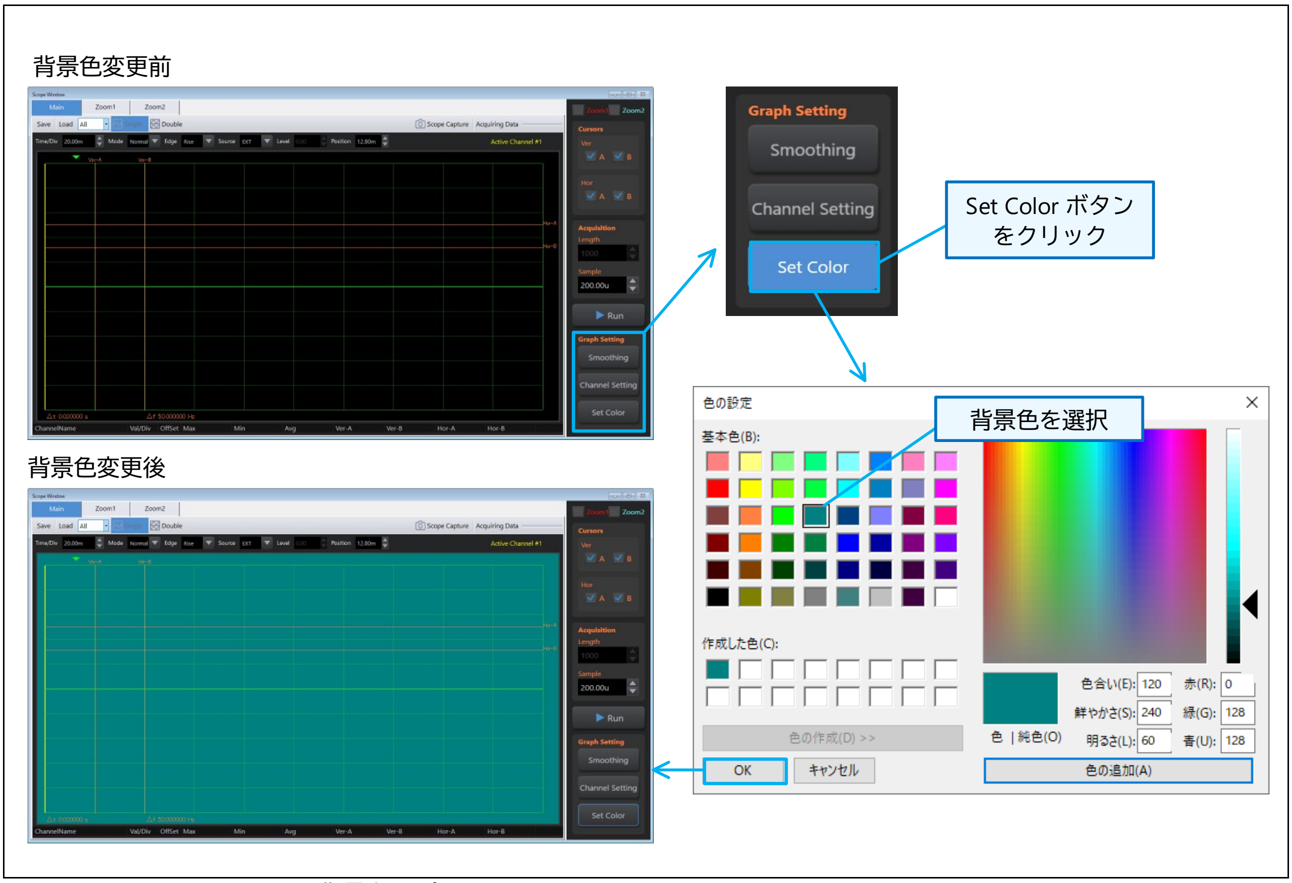


図 5-35 Set Color による背景色の変更

5.7.12 ショートカットキー

Scope Window では、下記のショートカットキー操作ができます。

表 5-20 ショートカットキー一覧

概要	動作	ショートカット
Zoom 範囲枠の表示	Zoom1 の表示／非表示	Alt + 1
	Zoom2 の表示／非表示	Alt + 2
Zoom 範囲枠の編集	（選択した Zoom 範囲枠の）始点を右へ移動	Alt + 右カーソル
	（選択した Zoom 範囲枠の）始点を左へ移動	Alt + 左カーソル
	（選択した Zoom 範囲枠の）幅を広げる	Alt + Shift + 右カーソル
	（選択した Zoom 範囲枠の）幅を縮める	Alt + Shift + 左カーソル
波形の計測	計測を開始（＝RUN ボタンをクリック）	R
	計測を停止（＝STOP ボタンをクリック）	S
画面イメージの取得	画面イメージをコピー	Ctrl + C
Active Channel の編集	Main1 の「Val/Div」値を大きくする	上カーソル
	Main1 の「Val/Div」値を小さくする	下カーソル
	Main2 の「Val/Div」値を大きくする	Shift + 上カーソル
	Main2 の「Val/Div」値を小さくする	Shift + 下カーソル
トリガ設定の切り替え	「Mode」の切り替え	O
	「Mode」の切り替え（逆順）	Shift + O
	「Edge」の切り替え	E
	「Edge」の切り替え（逆順）	Shift + E
	「Source」の切り替え	T

5.8　Image Editor 機能（Scope Capture ボタン）

Scope Window 上部の「Scope Capture」ボタンをクリックすると Image Editor が起動します。Image Editor は、表示中の波形画像とチャネル情報を取り込んで編集するための機能を提供します。

「Scope Capture」ボタンをクリックすると、その時点の波形画像とチャネル情報が Image Editor に取り込まれます。Image Editor の起動中は、「Scope Capture」ボタンをクリックするたびに波形画像とチャネル情報が追加で取り込まれます。

5.8.1　Image Editor の起動／終了

Scope Window の上部にある「Scope Capture」ボタンから起動し、Image Editor の右上「閉じる」ボタンから終了できます。

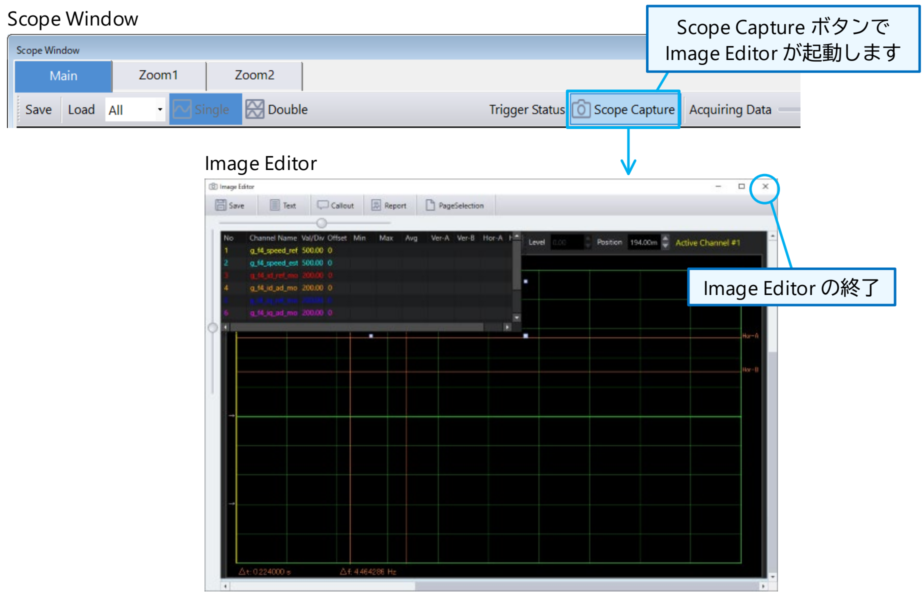


図 5-36　Image Editor の起動と終了

5.8.2　表示位置とサイズの変更

Image Editor に表示されたイメージをクリックすると選択状態となり、周囲に斜線とサイズ変更ハンドルが表示されます。サイズ変更ハンドルをマウスでつかむと、表示サイズの変更ができます。また、斜線表示の部分をマウスでつかんで、Image Editor 内で表示位置を移動できます。

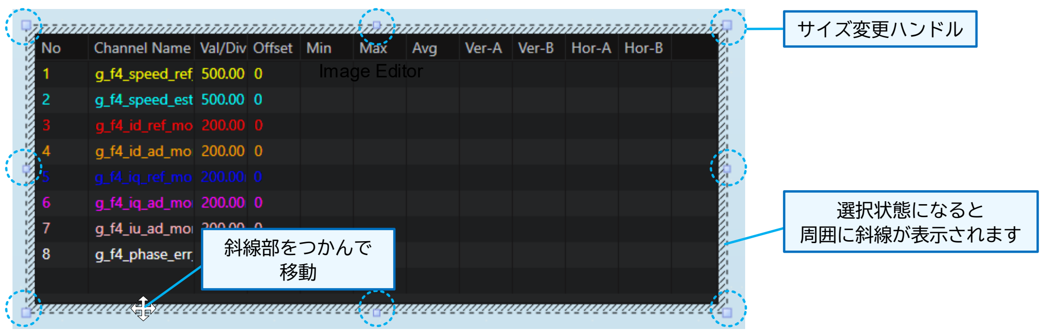


図 5-37　表示位置／サイズの変更

5.8.3 Image Editor の操作ボタン

Image Editor 上部には、以下の操作ボタンがあります。

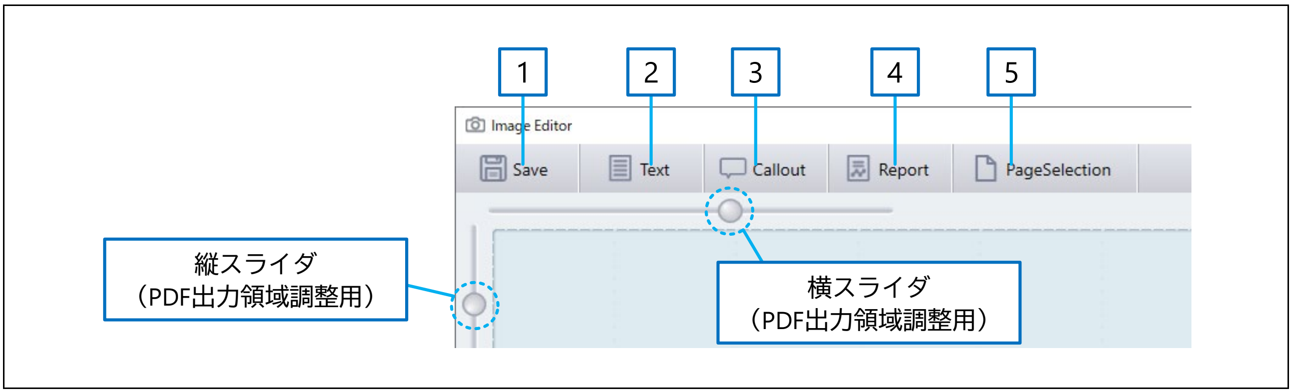


図 5-38 Image Editor の操作ボタン

(1) Save ボタン

「Save」ボタンをクリックすると Image Editor で編集したイメージ情報を bmp/png/jpg 形式で保存します。

(2) Text ボタン

「Text」ボタンをクリックするとテキスト入力可能なオートシェイプが Image Editor 内に表示され、移動や編集ができます。表示中のテキスト内の文字を選択して右クリックするとメニューが表示され、フォントや文字色の変更ができます。

(3) Callout ボタン

「Callout」ボタンをクリックすると吹き出しのメニューが表示されます。選択すると Image Editor に吹き出しが表示され、移動や編集ができます。表示中の吹き出しを右クリックするとメニュー表示され、設定の変更ができます。

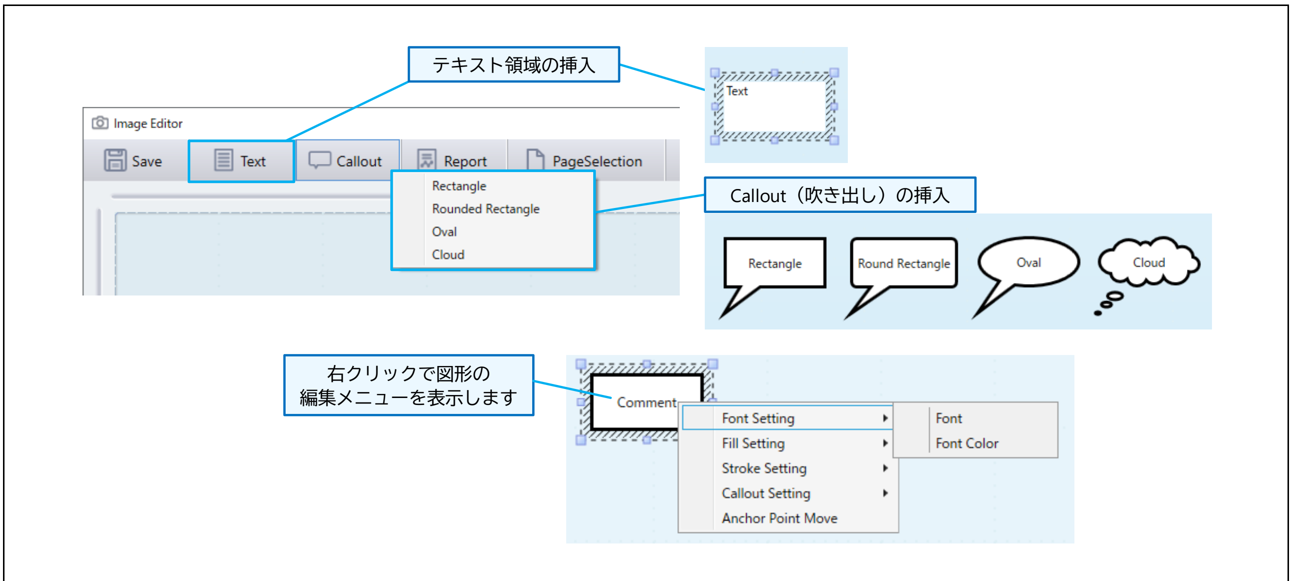


図 5-39 Callout(吹き出し)の種類

表 5-21　テキストと Callout（吹き出し）の右クリックメニュー設定

テキスト	吹き出し	右クリックメニュー	サブメニュー	内容
●	●	Font Setting	Font	文字フォントの選択
			Font Color	文字の色
●	●	Fill Setting	Fill Color	図形内部の塗りつぶし色
			Fill Color Transparent	図形内部を透明にする
●	●	Stroke Setting	Stroke Color	外枠の色
			Stroke Color Transparent	外枠を透明にする
			Stroke Width	外枠の太さ（Thin/Normal/Bold から選択）
－	●	Callout Setting	Rectangle/Round Rectangle/Oval/Cloud から選択	
－	●	Anchor Point Move	吹き出しの先端の移動	

(4)　Report

Report ボタンをクリックすると、Image Editor で編集した情報を PDF ファイルとして保存ができます。

(5)　Page Selection ボタン

Page Selection ボタンを ON 状態にすると、印刷範囲のページ番号が最前面に表示されます。最前面に表示されたページ番号を選択するとやや濃い青に変わり、出力の対象となります。非選択にするとやや薄い青になり、出力の対象外となります。

5.8.4　PDF 出力領域の調整

Image Editor 内のスライダで PDF 出力範囲を調整できます。横スライダで 1 ページの横幅を、縦スライダで縦幅が調整します。

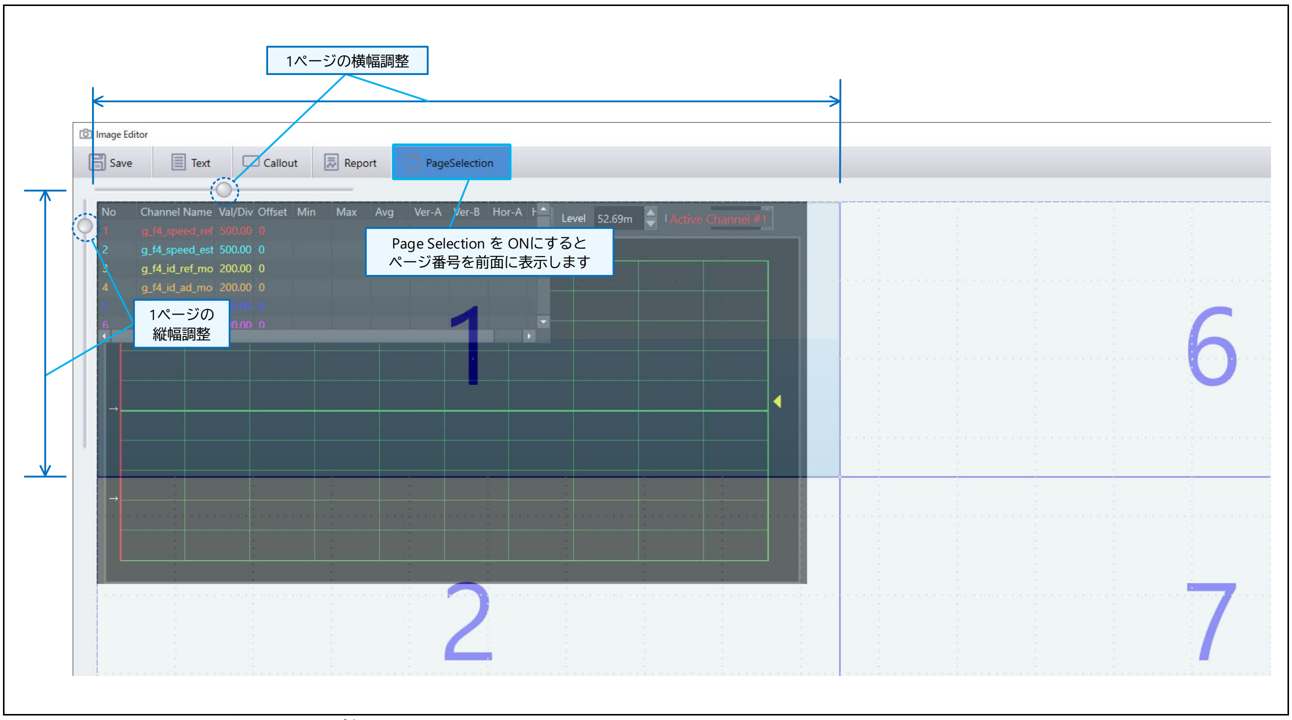


図 5-40　PDF 出力領域の調整

6. ［Analyzer］User Button Window

6.1 概要

User Button Window は、予め登録したシーケンス処理をユーザの指示（作成したボタンのクリック）で順次実行する機能を提供します。User Button Window は、Analizer ツールを起動すると表示されます。

6.2 特徴

- 同時実行など細かいシーケンスの指示が可能です。
- 内部変数を使うことで、取得した値をそのまま書き込めます。

6.3 画面構成

User Button Control Window のメニューから「Add New」を選択（または、既存のボタンの右クリックメニューで「Show detail Setting」を選択）すると、User Button 編集画面が表示されます。画面の構成を以下に示します。

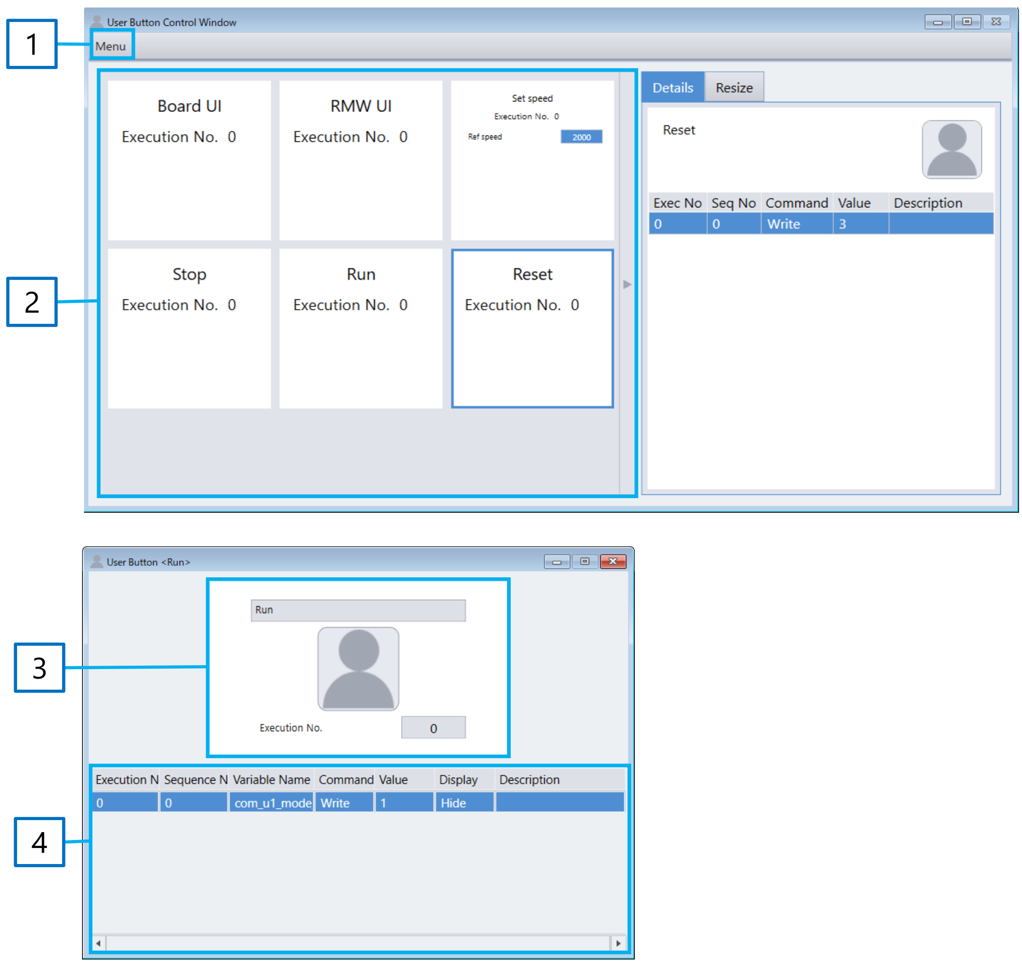


図 6-1　User Button 編集画面

表 6-1　User Button 編集画面の機能

No.	名称	説明
1	メニュー	Add New／Image Size／Show Control を選択できます。
2	ボタン統合部	複数のボタンが 1 つの画面に配置されます。
3	実行ボタン部	実行ボタン。ボタン名／Execution No／表示情報を編集できます。
4	シーケンス編集部	実行ボタンをクリックするたびに実行されるシーケンス。ここで編集できます。

6.3.1　実行ボタン部

ボタン部の各部の名称と機能を以下に示します。

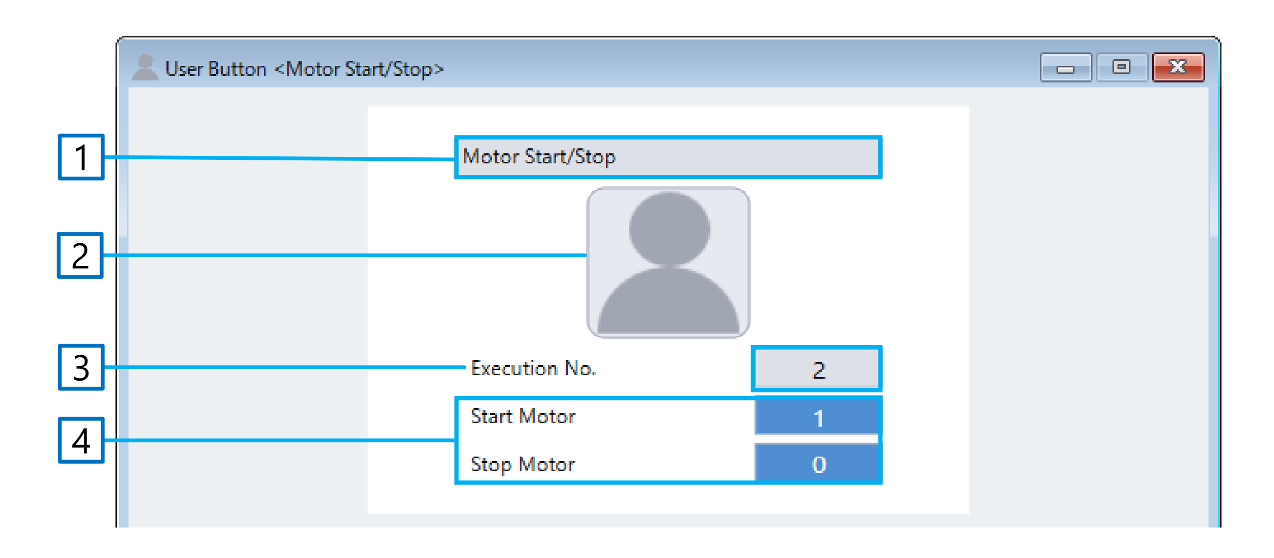


図 6-2　ボタン部画面

表 6-2　ボタン部の機能

No.	名称	説明
1	ボタン名	• 名称を自由に指定できます。 • 名称は Control Window の User Button メニューに表示されます。
2	実行ボタン	• クリックするとシーケンスが 1 ステップ実行します。 • 右クリックメニューからイメージ画像の設定ができます。
3	Execution No	• 実行ボタンクリック時に実行する Execution No を表示します。 • 実行ボタンクリックすると、次の Execution No に自動で切り替わり、最終番号まで行くと最初の Execution No に戻ります。 • Execution No は直接指定ができます。
4	情報表示	• シーケンス部位で Display=Show 指定としたシーケンス情報の Description と Value を表示します。 • 表示した Value 値は編集ができます

6.3.2 シーケンス編集部

シーケンス編集部の各部の名称と機能を説明します。

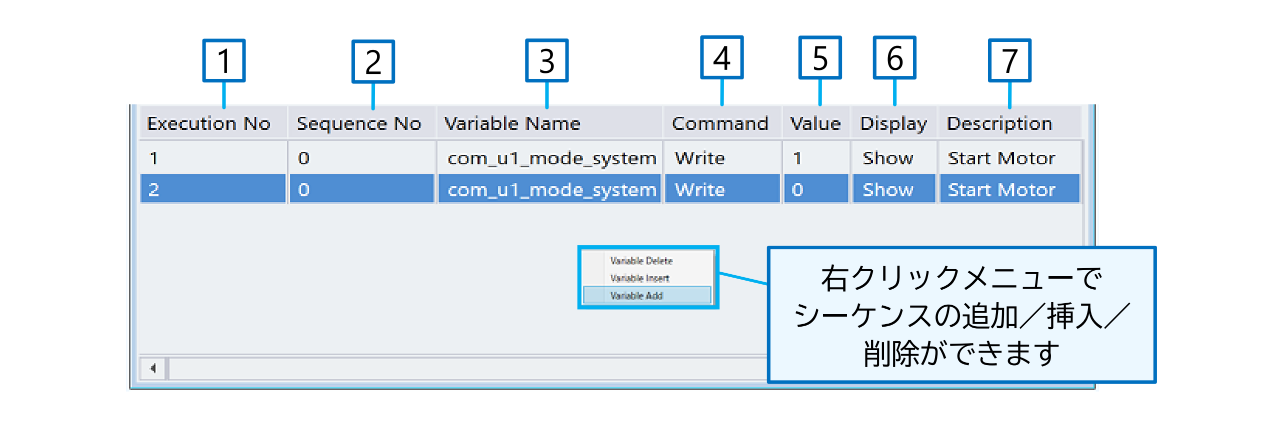


図 6-3　シーケンス編集部画面

表 6-3　シーケンス編集部の機能

No.	名称	説明
1	Execution No	• 実行ボタンを押したときの処理順を整数で指定します。小さい数値から順に実行します。
2	Sequence No	• 同一の「Execution No」がある場合、この Sequence No（処理順）を整数で指定して処理順を指定します。Sequence No が同じ場合は、上から実行します。
3	Variable Name	• 読み込み／書き込みする場合は、変数名を指定します。
4	Command	• 指令を選択できます。 - Read / Write　：変数値の読み込み／書き込み - Run / Stop　　：Scope Window の波形表示の開始／停止
5	Value	• 「Commander=Read」指定時：読み込んだ値を表示します。 • 「Commander=Write」指定時：書き込む値を指定します。 • 読み込み値を内部領域に保持し、書き込み値として利用することもできます。
6	Display	• ボタン部の「情報表示」に表示する（Show）／しない（Hide）を設定します。
7	Description	• 自由に記入できます。 • Display 欄で Show を設定すると、記載内容をボタン部の「情報表示」に表示します。

6.3.3 ボタン統合部

ボタン統合部の各部の名称と機能を説明します。

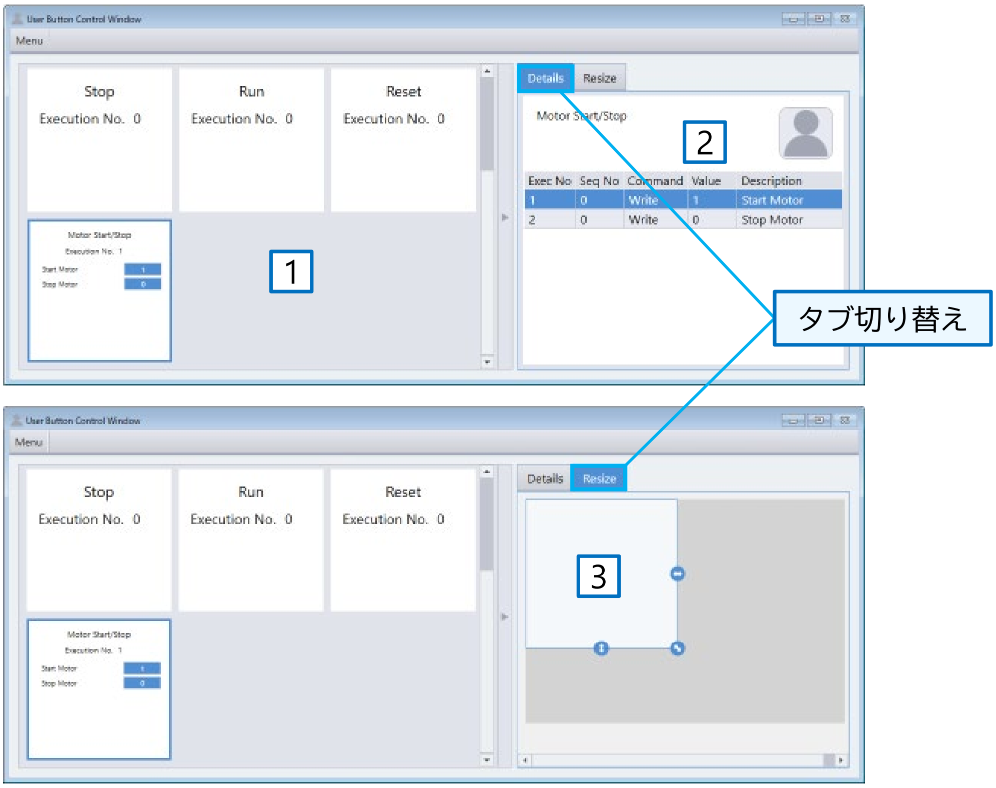


図 6-4　ボタン部画面

表 6-4　ボタン部の機能

No.	名称	説明
1	ボタン一覧	• ボタンが整列して配置されます。 • ボタンの並びはマウスのドラッグで自由に変更できます。
2	ボタン詳細タブ	• ボタンに設定されたシーケンスを確認できます。
3	ボタンサイズタブ	• ボタンのサイズを調整できます

6.4 操作説明

6.4.1 User Button の新規作成

User Button Control Window の Menu ＞ Add New を選択すると、新規 User Button 画面が起動します。User Button は最大 16 個まで作成できます。

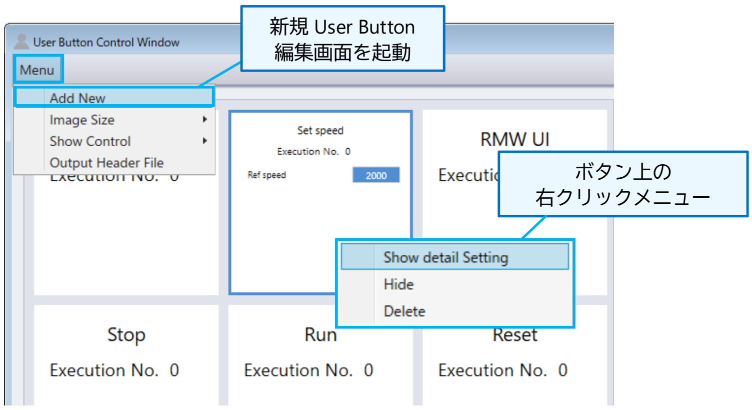


図 6-5 User Button の新規作成

6.4.2 User Button の削除

各ボタンを右クリックして表示されるメニューから「Delete」を選択すると、その User Button 画面が終了します。

6.4.3 User Button の表示／非表示

各ボタンを右クリックして表示されるメニューから「Hide」を選択すると、その User Button 画面を非表示にできます。User Button 画面は削除されません。

User Button Control Window の Menu ＞ Show Control に表示されるボタン名リストのチェックボックスで、対象の User Button 表示／非表示を設定できます。

6.4.4 シーケンスの編集

6.4.4.1 シーケンス行の設定

User Button 画面下部のシーケンス編集部で右クリックして表示するメニューから行の追加／挿入／削除ができます。ただし Variable Name では、右クリックすると Variable Name 設定用メニューが表示されます。

表 6-5 シーケンス編集部の右クリックメニュー

メニュー項目	動作
Variable Add	最終行に 1 行を追加します
Variable Insert	選択行の前に 1 行を追加します
Variable Delete	選択した行を削除します

6.4.4.2 Execution No の設定

User Button 画面上部の実行ボタンをクリックした時に実行する順番を整数で指定できます。番号が小さいものから順に実行します（飛び番号も指定可能）。1 回の実行ボタンクリックで複数行を処理する場合は、同一番号を複数行に指定してください。

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

実行ボタンのクリック
1 回目：Execution No=0 の行を実行
2 回目：Execution No=1 の行を実行
3 回目：Execution No=2 の行を実行
4 回目：Execution No=3 の行を実行
5 回目：（上記 1 回目からくりかえし）

図 6-6　Execution No 設定

6.4.4.3 Sequence No の設定

シーケンス内の Execution No の番号が同一のものがある場合に、その中での実行の順番を整数で指定できます。番号が小さいものから順に実行します（飛び番号も指定可能）。同一番号を複数行に指定もできます、下記の例では、上から順番に実行します。

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

実行ボタンのクリック
1 回目 ：Execution No=0、Sequence No=0 の行を実行
→ Execution No=0、Sequence No=1 の行を続けて実行
2 回目 ：Execution No=2、Sequence No=0 の行を実行
→ Execution No=2、Sequence No=1 の行を続けて実行
3 回目 ：（上記 1 回目からくりかえし）

図 6-7　Sequence No 設定

6.4.4.4 変数の設定

シーケンス内の Variable Name には、下記のいずれかの操作で変数名の設定ができます。

(a) 変数名の直接入力

途中まで入力すると候補の変数名が表示されます（ただし、変数リストの中の候補の最初の 1 つが表示されるだけで、候補を絞り込んだリスト表示はありません）。

(b) 変数リストから選択

Variable Name セルを選択すると、セル右側に「v」が表示されます。「v」をクリックすると変数リストが表示され、変数名を選択できます。

(c) Variable Find 機能を使って選択

Variable Name セルを 1 回だけクリックして、右クリックで表示されるメニューから「Variable Find」を選択すると Variable Find 画面が表示されます。操作方法は、「4.4.1.3 Variable Find 機能を使った変数選択」を参照してください。

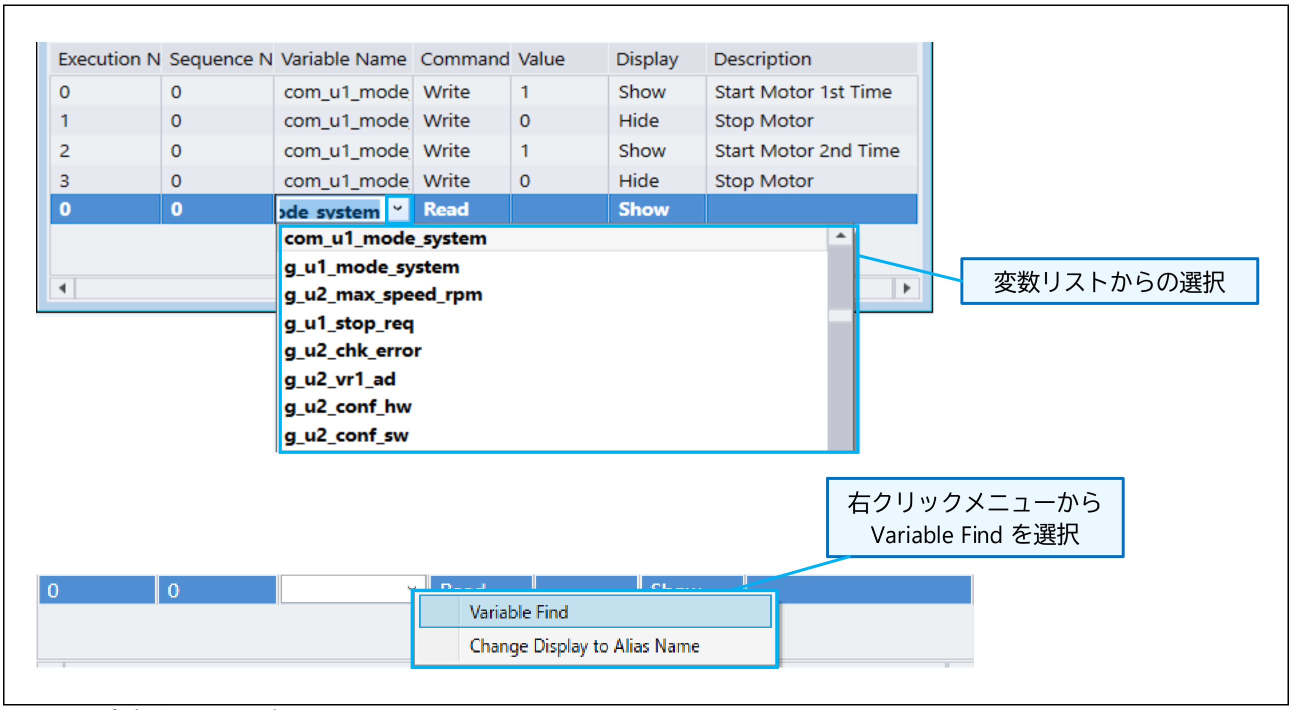


図 6-8 変数名の入力方法

6.4.4.5 Command の設定

シーケンス内の Command セルを選択して表示する「v」をクリックすると、選択可能なリストが表示されて選択ができます。

表 6-6 Command 一覧

Command 欄	動作
Read	変数値を読み込みます。
Write	変数値を書き込みます。
Run	Scope Window の波形表示を開始します。
Stop	Scope Window の波形表示を停止します。

6.4.4.6 Value の設定

Command 欄で「Read」を指定すると、読み込んだ変数の値をその行の Value に表示します。
Command 欄で「Write」を指定すると、Value に設定した値を変数に書き込みます。

6.4.4.7 Display の設定

シーケンス内の Display セルを選択して表示する「v」をクリックするとリストが表示され、選択すると下記の設定ができます。

- Show : 画面上部にシーケンス情報（Description, Value）を表示する。
- Hide : 画面上部にシーケンス情報（Description, Value）は表示しない。

6.4.4.8 Description の設定

シーケンスの Display 欄を「Show」にした場合、Description 欄に記入した内容がボタン部に表示されます。

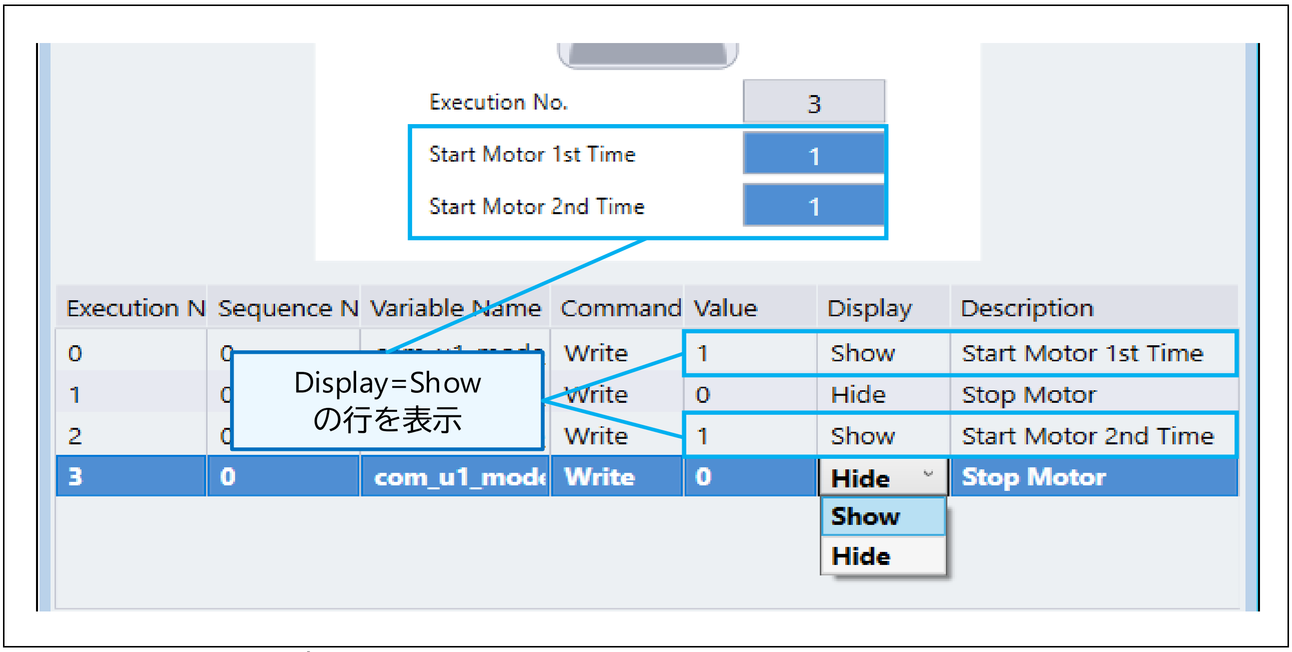


図 6-9 Description の設定

6.4.5 内部変数の利用

User Button 画面では、読み込んだ値を書き込み値として利用できる User Button 用の内部変数が準備されています。User Button 用の内部変数は、シーケンスの Value 欄に「先頭文字が A または a で、それに続けて数字列」を記述すると内部変数として利用できます（内部変数例：A1, a12345 など）。

表 6-7 内部変数利用時の Command 一覧

Command 欄	動作
Read	変数から読み込んだ値を内部変数に保存します。
Write	内部変数に保存した値を書き込みます。

内部変数の値を直接表示や参照はできません。また、複数の User Button 画面をまたがって内部変数を利用することはできません。

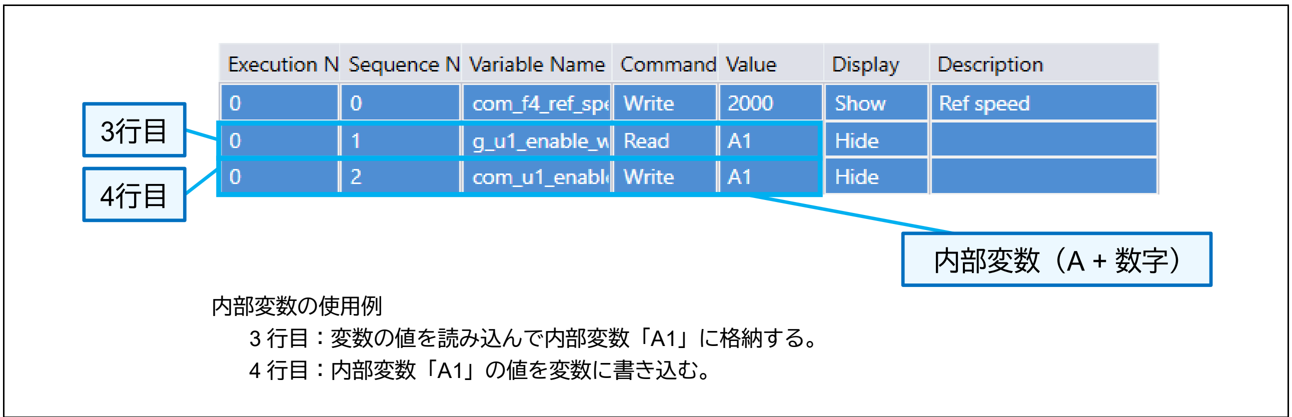


図 6-10 内部変数の使用例

6.4.6 画面の表示設定

6.4.6.1 実行ボタンへの画像設定

User Button 画面上部のボタン部を右クリックして表示されるメニューから「Image」を選択すると、実行ボタンに画像を設定できます。

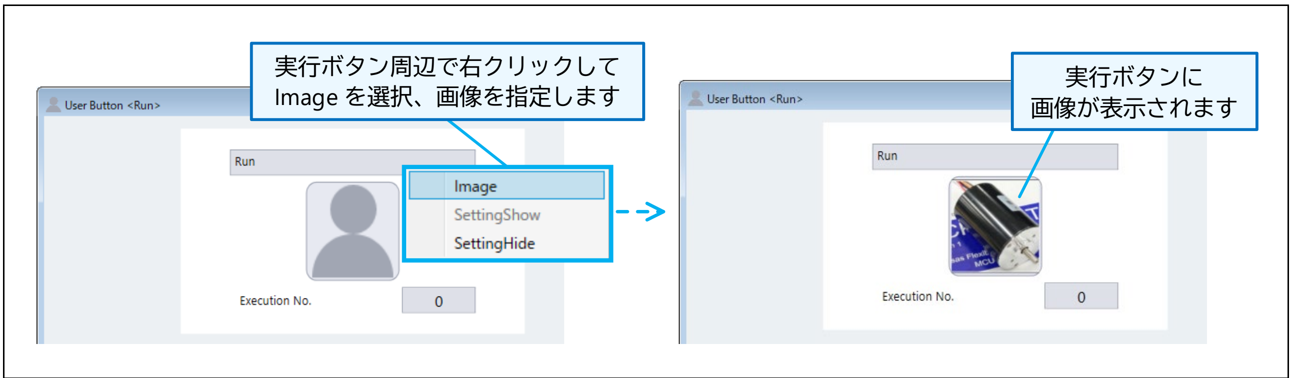


図 6-11 実行ボタンへの画像設定する方法

6.4.6.2 シーケンス編集部の表示／非表示

User Button 画面のボタン部で右クリックすると表示されるメニューから「Setting Show」または「Setting Hide」を選択することで、シーケンス編集部の表示／非表示を切り替えられます（右クリックは、必ずボタン部の上で行ってください）。

表 6-8 シーケンス編集部の表示切り替え

右クリックメニュー	操作
Setting Show	シーケンス編集部を表示する
Setting Hide	シーケンス編集部を非表示にする

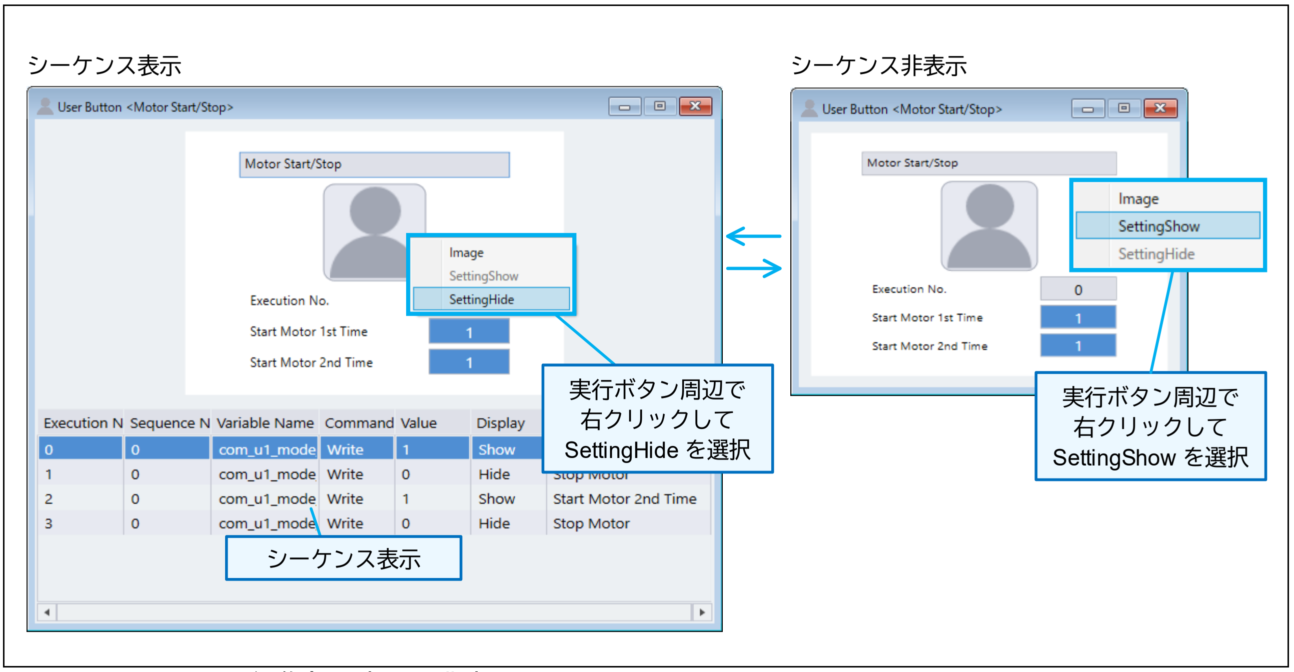


図 6-12 シーケンス編集部の表示／非表示

7. ［Analyzer］Commander 画面

7.1 概要

Commander は、変数への書き込み指示のシーケンスを予め登録し、シーケンスを連続的に実行するための機能です。Commander は、Analyzer ツールの Control Window にある「Commander」ボタンで起動します。

7.2 特徴

- シーケンシャルに書き込みを実行できます。
- インターバル間隔の設定が可能です。
- 繰り返し（ループ）処理を実行できます。

7.3 画面構成

Commander の画面構成を以下に示します。

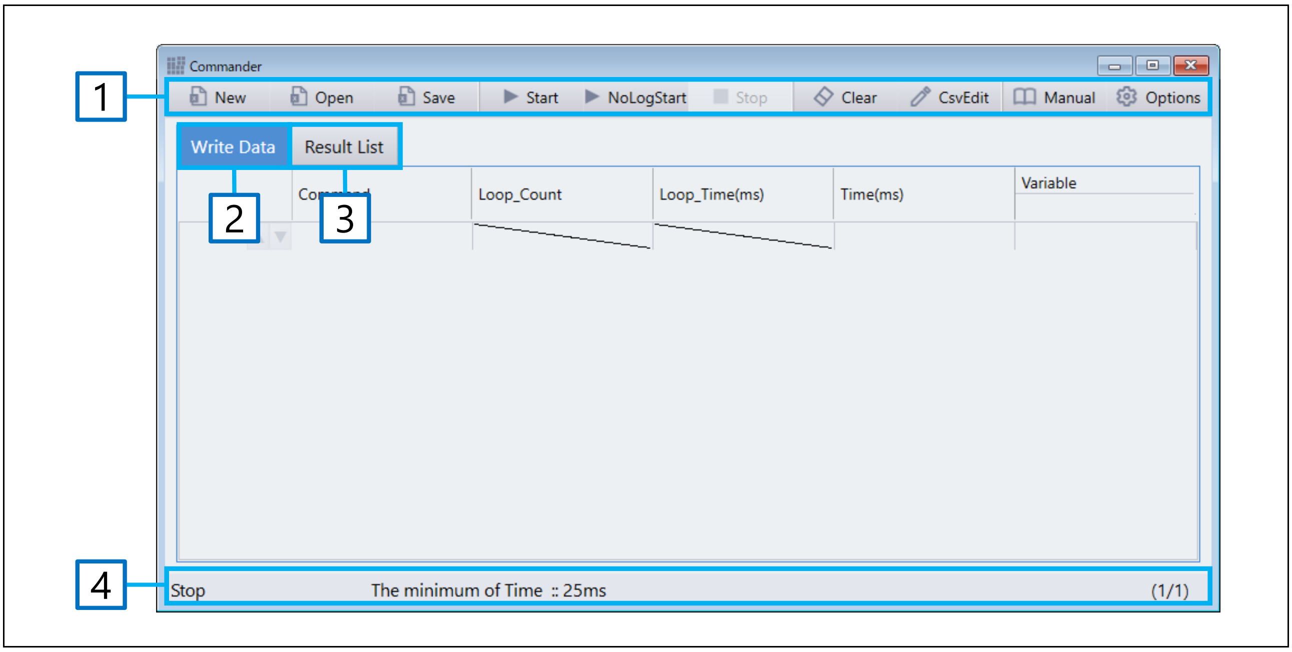


図 7-1　Commander 画面

表 7-1　Commander 画面の機能

No.	名称	説明
1	操作ボタン	画面上部のボタンクリックで、CSV ファイルの読み込み／書き込み、CSV ファイルの編集／編集情報の反映、シーケンスの実行／停止などができます。
2	Write Data タブ	シーケンス情報を表示します。この画面ではシーケンス編集はできません（シーケンス編集は CSV Edit ボタンから行います）。
3	Result Data タブ	シーケンスの実行結果を一覧で表示します。
4	ステータスバー	シーケンス実行状態、Send Checker 状態、コマンド数などを表示します。

7.3.1 操作ボタン

ツールバーの操作ボタンについて説明します。

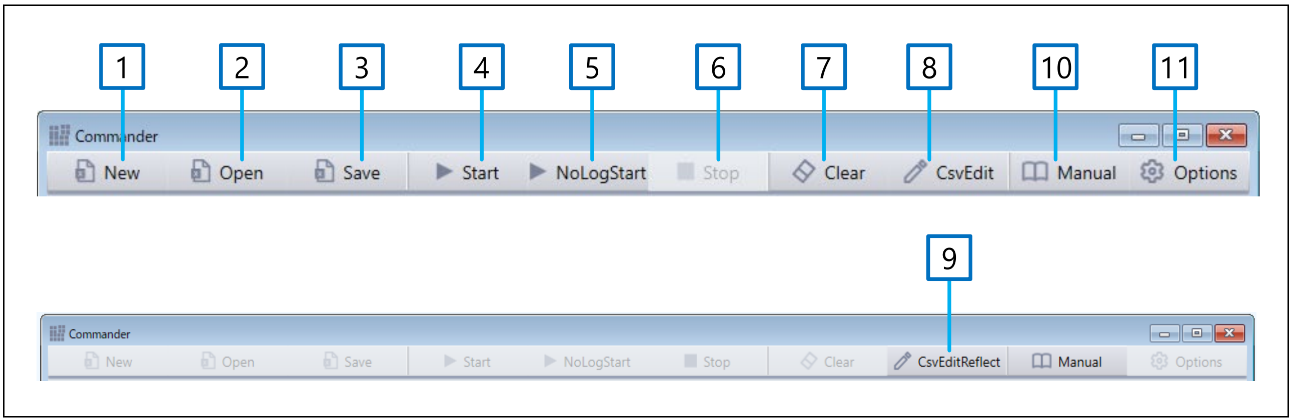


図 7-2　ツールバー

表 7-2　ツールバーの操作ボタン機能

No.	名称	説明
1	New	新規 CSV ファイルが生成され、初期状態のシーケンス情報を表示します。
2	Open	既存 CSV ファイルを読み込み、シーケンス情報を表示します。
3	Save	CSV ファイルを保存します。
4	Start	シーケンスを実行します。
5	NoLogStart	ログ取得を行わずにシーケンスを実行します。
6	Stop	実行中シーケンスを停止します。
7	Clear	表示中のシーケンス情報をクリアします。
8	CSV Edit	シーケンス編集画面が起動して、表示中のシーケンス（CSV ファイル）情報を表示します。
9	CSV Edit Reflect	シーケンス編集が終了するとボタン表示が切替ります。ボタンをクリックすると編集したシーケンス（CSV ファイル）情報を表示します。
10	Manual	入力ガイドを表示します。
11	Options	各種設定を行います。

7.3.2 Write Data タブ

CSV ファイルを読み込むと、Write Data タブ画面にシーケンスが表示されます。

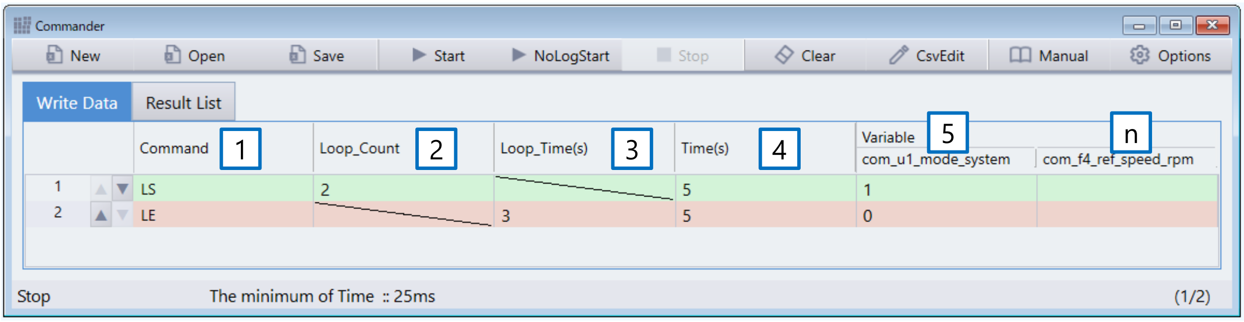


図 7-3 Write Data タブ画面

表 7-3 Write Data タブ画面の機能

No.	名称	説明
1	Command	繰り返し処理コマンドを表示します。 • LS：繰り返し開始 • LE：繰り返し終了
2	Loop Count	繰り返し回数を表示します。
3	Loop Time	繰り返し処理の最終行のコマンド実行後、最初の行のコマンドに戻って実行するまでの間隔を表示します。
4	Time	各行の実行待ち間隔を表示します。
5	変数名	見出し行　　　：コマンド発行する変数名を表示します。 見出し行以下　：見出しの変数へ書き込みする値を表示します。
n	変数名	シーケンスに設定されている変数の数だけ表示します。

7.3.3　Result List タブ

Write Data タブのシーケンスを実行すると、結果が Result List タブ画面に表示されます。

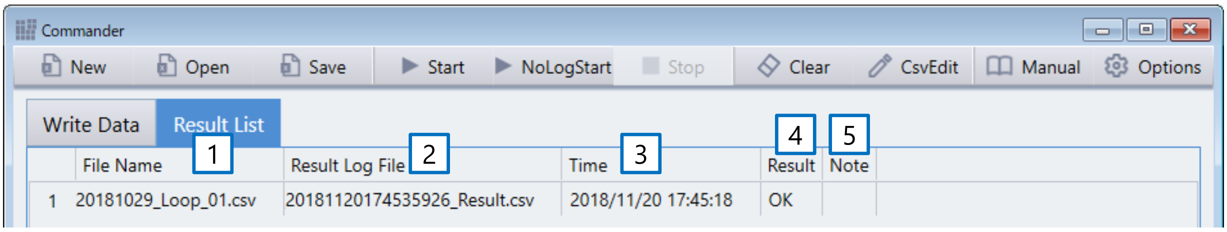


図 7-4　Result タブ画面

表 7-4　Result タブ画面の機能

No.	名称	説明
1	File Name	シーケンス実行した CSV ファイル情報を表示します。
2	Result Log File	シーケンス実行した結果の Log ファイルの情報を表示します。
3	Time	シーケンス実行した日時を表示します。
4	Result	シーケンス実行した結果（OK/NG）を表示します。
5	Note	メモ欄として自由に記入できます

7.3.4　ステータスバー

ステータスバーには Commander の状態を表示します。

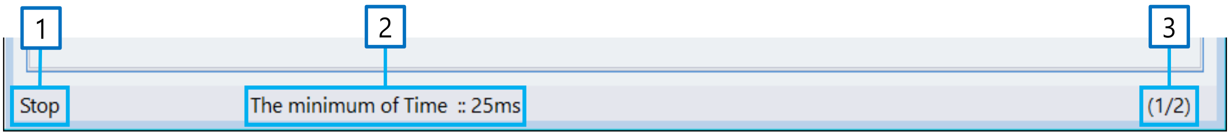


図 7-5　ステータスバー

表 7-5　ステータスバー機能

No.	名称	説明
1	実行状態	シーケンスの実行状態を表示します。 • Running　：Start ボタンクリックでシーケンスを実行している状態 • Stop　　：Stop ボタンクリックでシーケンスが停止している状態、 または シーケンス実行が終了した状態
2	Send Checker 情報	Send Checker の情報を表示します。 • Send Checker 未実行　：「Please press the Send Checker Button.」 • Send Checker 実行済み　：「The minimum of Time :: XXms」(XX：計測値)
3	実行カウント	シーケンスの（行番号／総行数）を表示します。 • 行番号：クリック選択中の行番号、または実行中の行番号を表示 • 総行数：シーケンスの総命令行数

7.4　操作説明

7.4.1　Commander の起動／終了

Control Window の「Commander」ボタンをクリックすると、Commander が起動します。Commander は、Commander 画面の右上の「閉じる」ボタンで終了します。

7.4.2　Send Checker の実行

Commander を起動すると Send Checker 機能が実行され、ご使用の環境の指令値送信する間隔の最小値をシステムで計測します。Commander を操作するには、最初に Send Checker の実行が必要です。

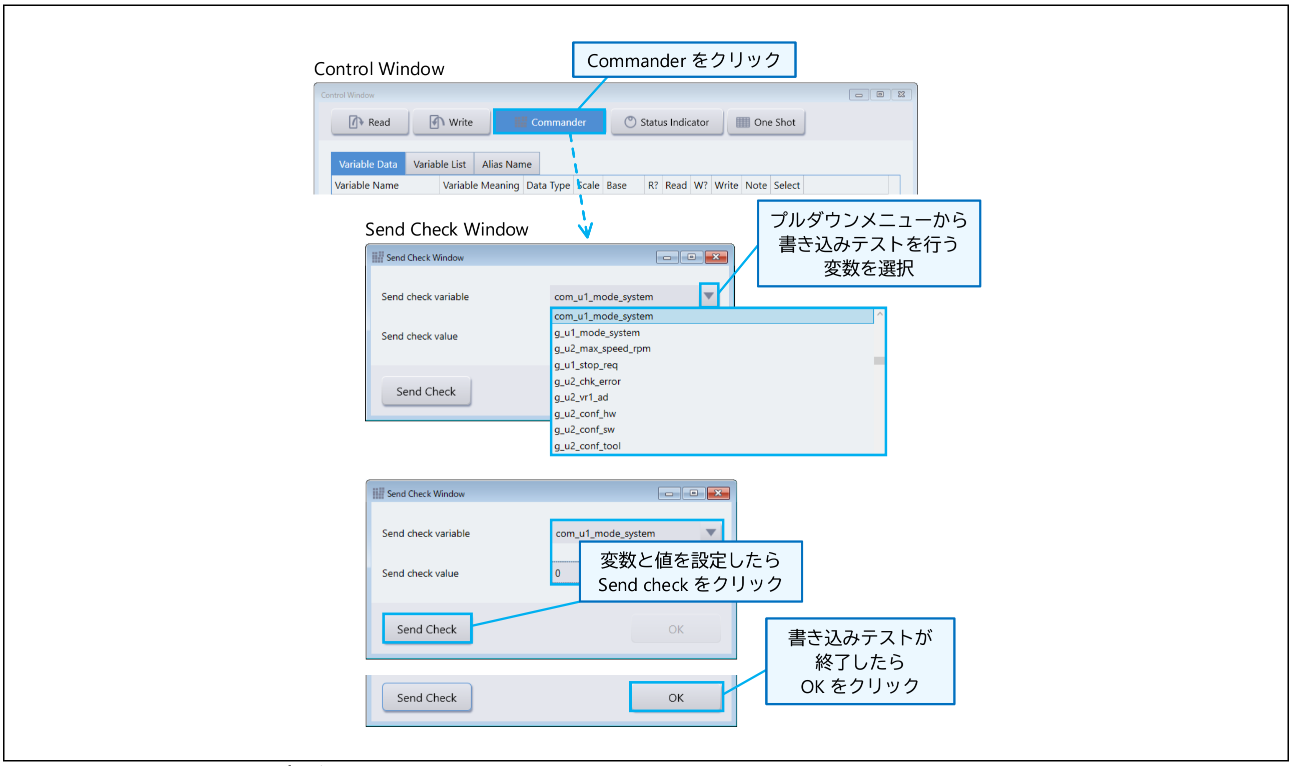


図 7-6　Commander の起動と Send Check Window

［操作］

1. Send Check Window で下記項目を設定します。
   - Send Check Variable：通信テストで使用可能な変数をプルダウンメニューから選択します
   - Send Check Value　：通信テストで使用可能な値を指定します

注意　計測は、実際に選択した変数に値を書き込むため、書き込みを行っても安全な変数名とその値を指定してください。

2. 「Send check」ボタンを押して計測します。
3. Send Checker が正常に終了すると、「OK」ボタンを押せる状態になります。

Send Checker が終わったら、「OK」ボタンをクリックして、Commander 画面を開きます。Send Checker の計測結果は、Commander 画面のステータスバーに表示されます。

- Send Checker 実行後の表示：「The minimum of Time :: XX ms」（XX：計測値）

7.4.3　CSV ファイルの指定

7.4.3.1　CSV ファイルの新規作成（「New」ボタン）

Commander 画面の「New」ボタンをクリックすると、新規で CSV ファイルを生成して、その情報を Commander 画面に表示します。

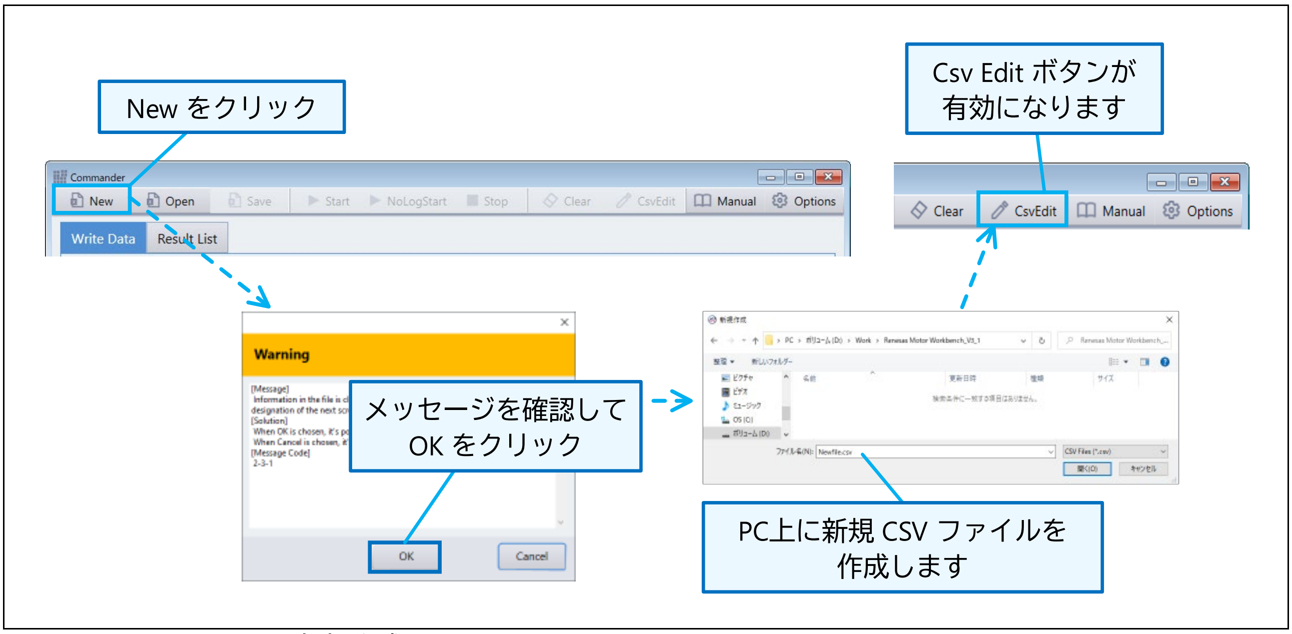


図 7-7　CSV ファイルの新規作成

［操作］

1. 「New」ボタンをクリックすると、Warning メッセージ画面が表示されます。
2. Message：既存ファイルを選択した場合は、そのファイルを初期化します 注。
3. Warning メッセージ画面の OK をクリックすると、ファイル名を指定する画面が表示されます。
4. ファイル名を入力して「開く」ボタンをクリックすると、新しい CSV ファイル生成され、「CSV Edit」ボタンが有効になります。

注　ファイル名の指定で、シーケンス情報を含んだ既存の CSV ファイルを指定すると、その CSV ファイルは、開いた時点で初期化されます。既存の CSV ファイルを読み込みたい場合は、次の「Open」ボタンを使用してください。

7.4.3.2　既存ファイルを選択（「Open」ボタン）

「Open」ボタンをクリックすると、作成／編集済みの CSV ファイルを読み込み、Commander 画面へシーケンスを表示します。

［操作］

1. 「Open」ボタンをクリックすると、ファイルを開く画面が表示されます。
2. 既存の CSV ファイル名を指定して「開く」ボタンをクリックすると、ファイルが読み込まれて、Commander 画面にシーケンス情報が表示されます。

7.4.3.3　表示中のシーケンス情報のクリア（「Clear」ボタン）

「Clear」ボタンをクリックすると、Commander 画面に表示中のシーケンスをクリアします。

7.4.4 CSV ファイルの編集（「CSV Edit」ボタン）

Commander 画面にシーケンスを表示した状態になると、「CSV Edit」ボタンが有効になります。「CSV Edit」ボタンをクリックすると編集画面（エクセル）が起動します。ここではシーケンスの編集方法を説明します。

7.4.4.1 基本設定

(a) 変数の指定

新規 CSV ファイル時は、E1 セルには「Please register available variable」が表示されますが、これは消去して変数名を指定します。

1 行に複数の変数を指定（同じタイミングで実行）する場合は、E1 セルから右（F1, G1, H1 ...）へ,

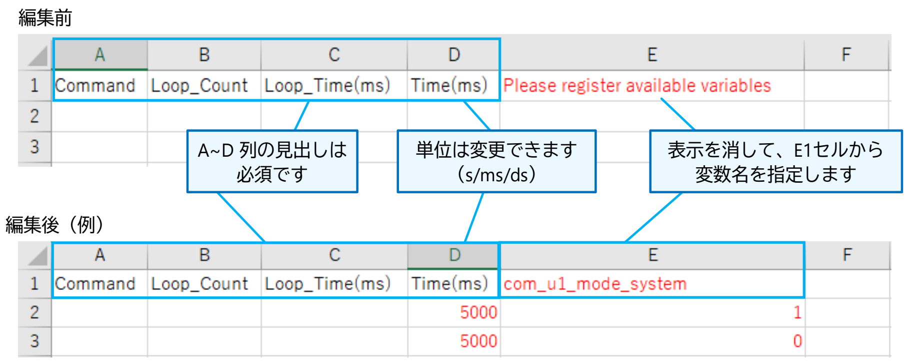
1 列に 1 変数を間隔を空けずに詰めて記載します。

(b) 処理間隔の指定

「Time(xx)」は、処理間隔（インターバル）を指定できます。値は正の整数／正の小数の指定が有効です。

(c) 指令値の設定

変数名の記載がある列の 2 行目以下に値を設定すると指令値発行ができます。値は、整数／小数の指定が有効です。1 行（同じタイミング）で多数の変数名へ指令を発行すると、その処理が完了する前に次の行の指令が発行される可能性があります。



編集例のシーケンス

（2 行目）5 秒後モータ駆動開始

（3 行目）5 秒後モータ駆動停止

図 7-8 Write データの編集

7.4.4.2 LOOP 指定

(a) 繰り返しの指定

シーケンスの「Command」の列は、繰り返し（Loop）処理の指定ができます。開始行に「LS」、終了行に「LE」を大文字で指定します。繰返しは複数（入れ子）設定ができます。

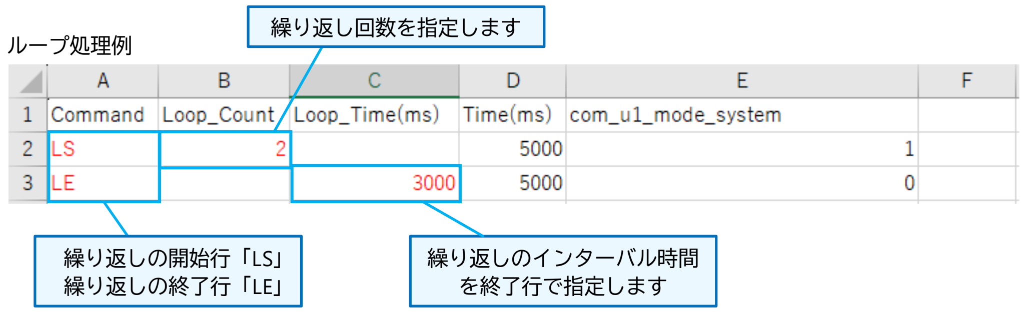
(b) 繰り返し回数の指定

シーケンスの「Loop_Count」の列は、繰り返し回数の指定ができます。回数は Command の「LS」指定行に指定します。値は、正の整数が有効です。

(c)　繰り返し間隔の指定

シーケンスの「Loop_time」の列は、繰り返し（Loop）処理の「LE」（終了）指定行から「LS」（開始）指定行へ戻る時の処理間隔（インターバル）を指定します。

Loop_time は「LE」（終了）を指定した行に記述します。値は、正の整数／正の小数が有効です。



編集例では以下のシーケンスを 2 回実行
（2 行目）5 秒後モータ駆動開始
（3 行目）5 秒後モータ駆動停止
（3 行目）3 秒間のインターバル → （2 行目）に戻る

図 7-9　繰り返し処理の設定例 1

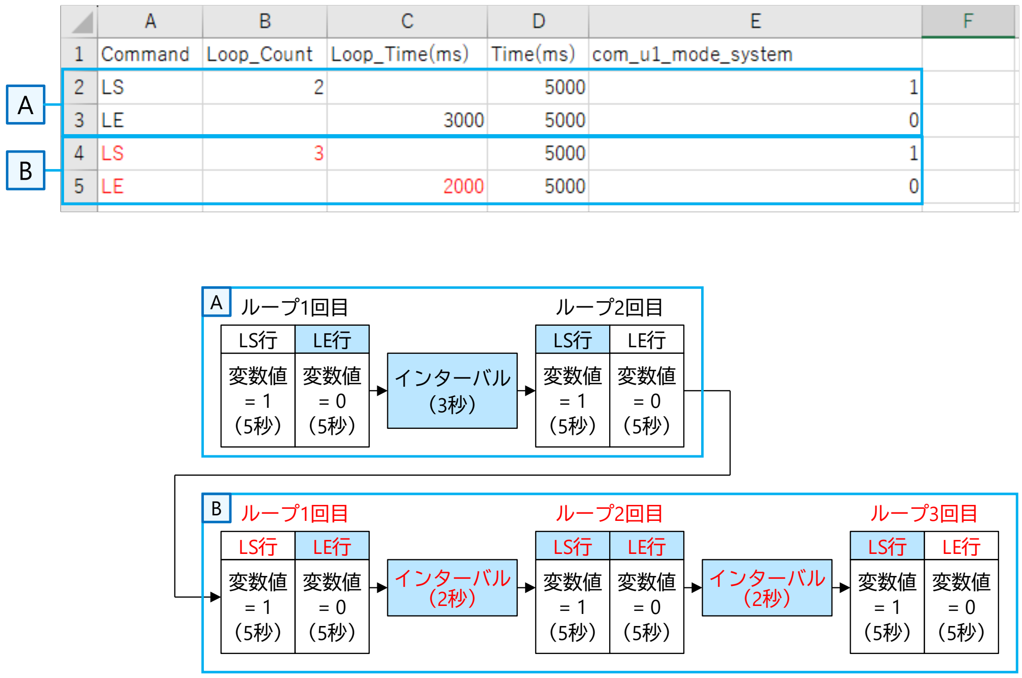


図 7-10　繰り返し処理の設定例 2

7.4.4.3 時間単位指定

シーケンスの「Time」「Loop_Time」の列は、シーケンスの処理間隔の指定ができます。

処理間隔の単位は「s/ms/ds」が指定できます。設定は見出し行の「Time(単位)」「Loop_Time(単位)」のカッコ内の単位を修正します。

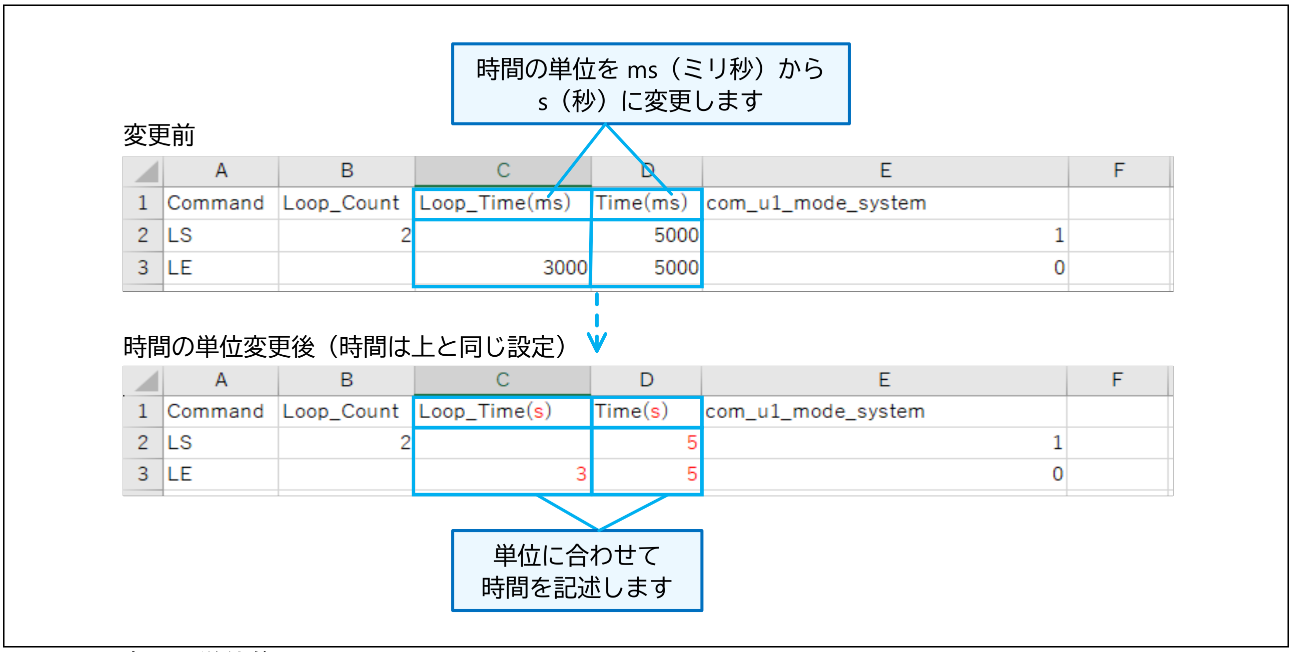


図 7-11　時間の単位修正

7.4.4.4 編集の終了と反映

(a) 編集画面の終了

編集したシーケンス情報は CSV ファイルへ上書き保存して、編集画面（CSV ファイル）を閉じてください。「Csv Edit」ボタンは「Csv Edit Reflect」ボタンに表示が切り替わります。

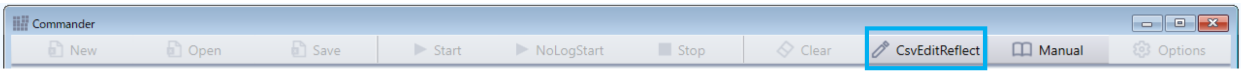


図 7-12　CSV Edit Reflect ボタン

(b) 編集したシーケンス情報反映（「CSV Edit Reflect」ボタン）

「Csv Edit Reflect」ボタンをクリック[注1]すると、編集したシーケンス（CSV ファイル）情報の文法チェックが行われて、問題がある場合はメッセージが表示されます。問題がない場合は編集したシーケンス（CSV ファイル）情報が Commander 画面に反映されて[注2]、ボタン表示が「CSV Edit」に切り替わります。

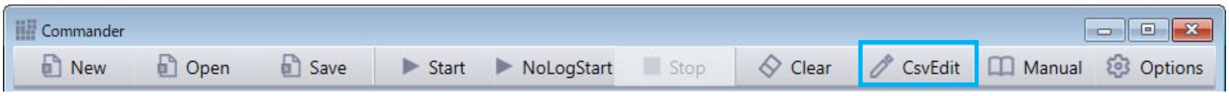


図 7-13　CSV Edit ボタン

注 1. CSV ファイルを編集した画面を閉じずに「CSV Edit Reflect」ボタンをクリックとエラーになります。保存後は編集画面を必ず閉じてください。

2. CSV ファイルの編集終了後、Commander に読み込んだ CSV ファイル名と異なるファイル名で保存終了すると、「CSV Edit Reflect」ボタンをクリックしても編集したシーケンス（CSV ファイル）情報は Commander 画面へ反映されません。この場合、Commander 画面の「Open」ボタンから保存した CSV ファイル名を指定して反映してください。

7.4.5　シーケンス実行の準備

シーケンスを実行する前に、シーケンス実行結果のログ保存に関する設定を行います。設定は、ツールバーの「Options」ボタンで行います。

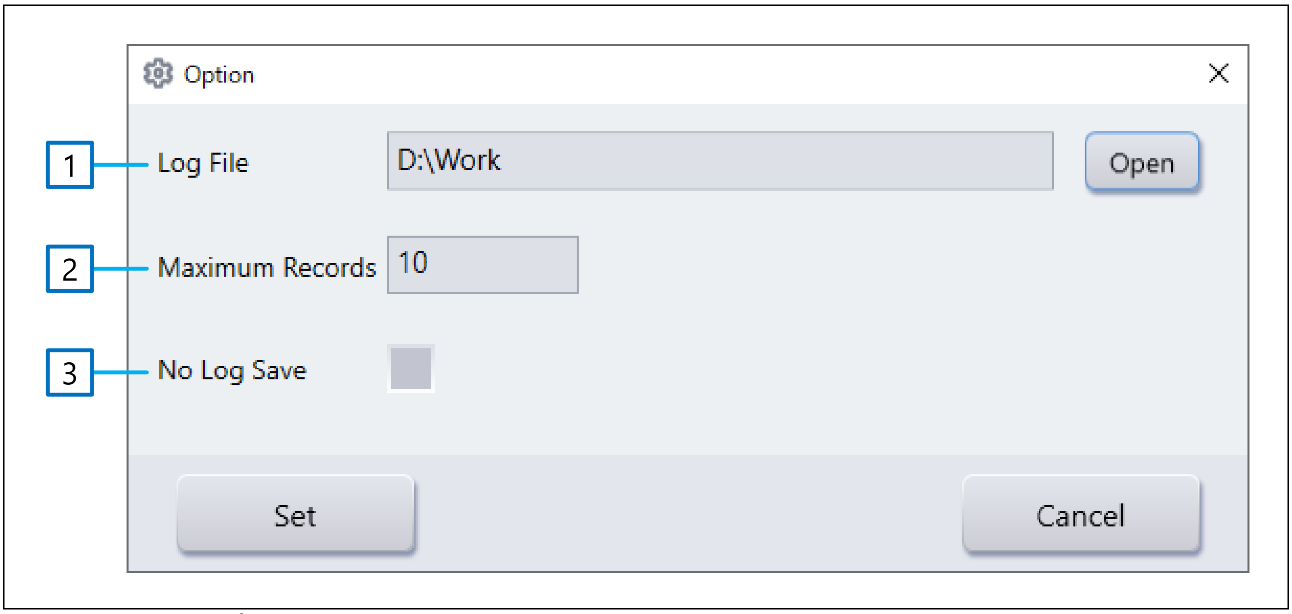


図 7-14　Option 設定画面

表 7-6　Option の設定

No.	項目名	設定
1	Log File	シーケンス実行結果の Log ファイルの出力フォルダを指定します（ユーザがアクセス可能なパスを指定してください）。
2	Maximum Records	結果レコードの最大保持数（MAX）を指定します。初期値は 10 件に設定されています。
3	No Log Save	チェックを入れるとログの取得を行いません。

ログファイルは下記のように作成されます（Option の「No Log Save」にチェックを入れた場合、および「NoLogStart」ボタンでシーケンス実行を開始した場合は出力されません）。

- ログファイル出力先　: Option の Log File で指定したフォルダ
- ログファイル名　　　: <CSV ファイル名> + “_YYYYMMDDhhmmssxxx_Result.csv”

7.4.6 シーケンスの実行

Commander にシーケンス（CSV ファイル）情報が読み込まれると、Commander 画面の「Start」ボタンと「NoLogStart」ボタンが有効となります。

「Start」ボタンをクリックすると、Commander 画面に表示中のシーケンスを実行します。表示中のシーケンス行を選択した状態から「Start」ボタンをすると、選択行から実行します。

「NoLogStart」ボタンでシーケンスを実行した場合は、ログの取得を行いません。

実行中のシーケンス行は、青枠で表示されます。シーケンスを最後まで実行すると自動で停止します。また、シーケンス実行中に「Stop」ボタンをクリックすると停止します。

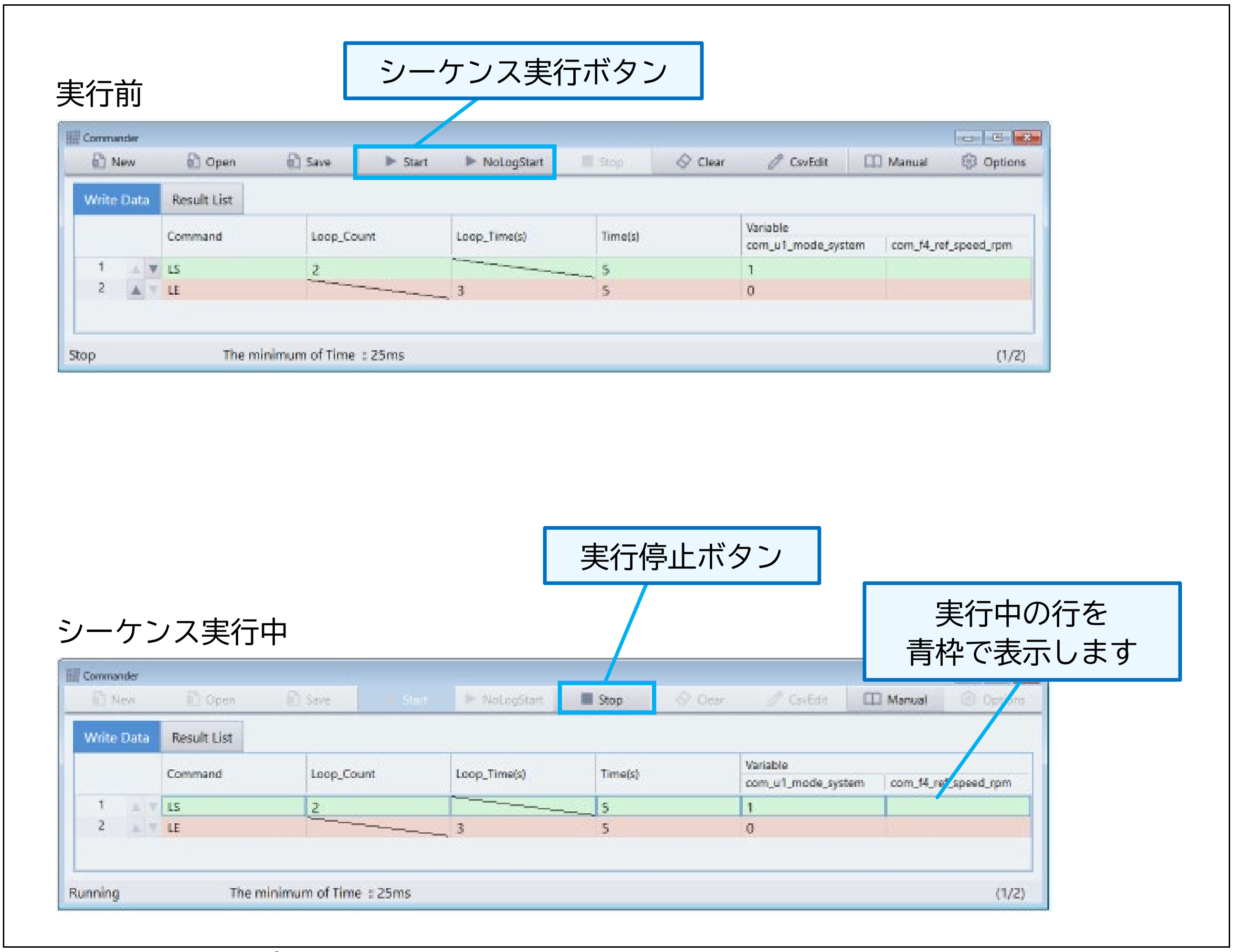


図 7-15 シーケンスの実行

7.4.7　シーケンス実行の結果

7.4.7.1　Result List タブ

シーケンス実行の結果は、Result List タブに結果レコード一覧で表示します。また、結果は Log ファイルにも出力されます（Option の「No Log Save」にチェックを入れた場合、および「NoLogStart」ボタンでシーケンス実行を開始した場合は出力されません）。

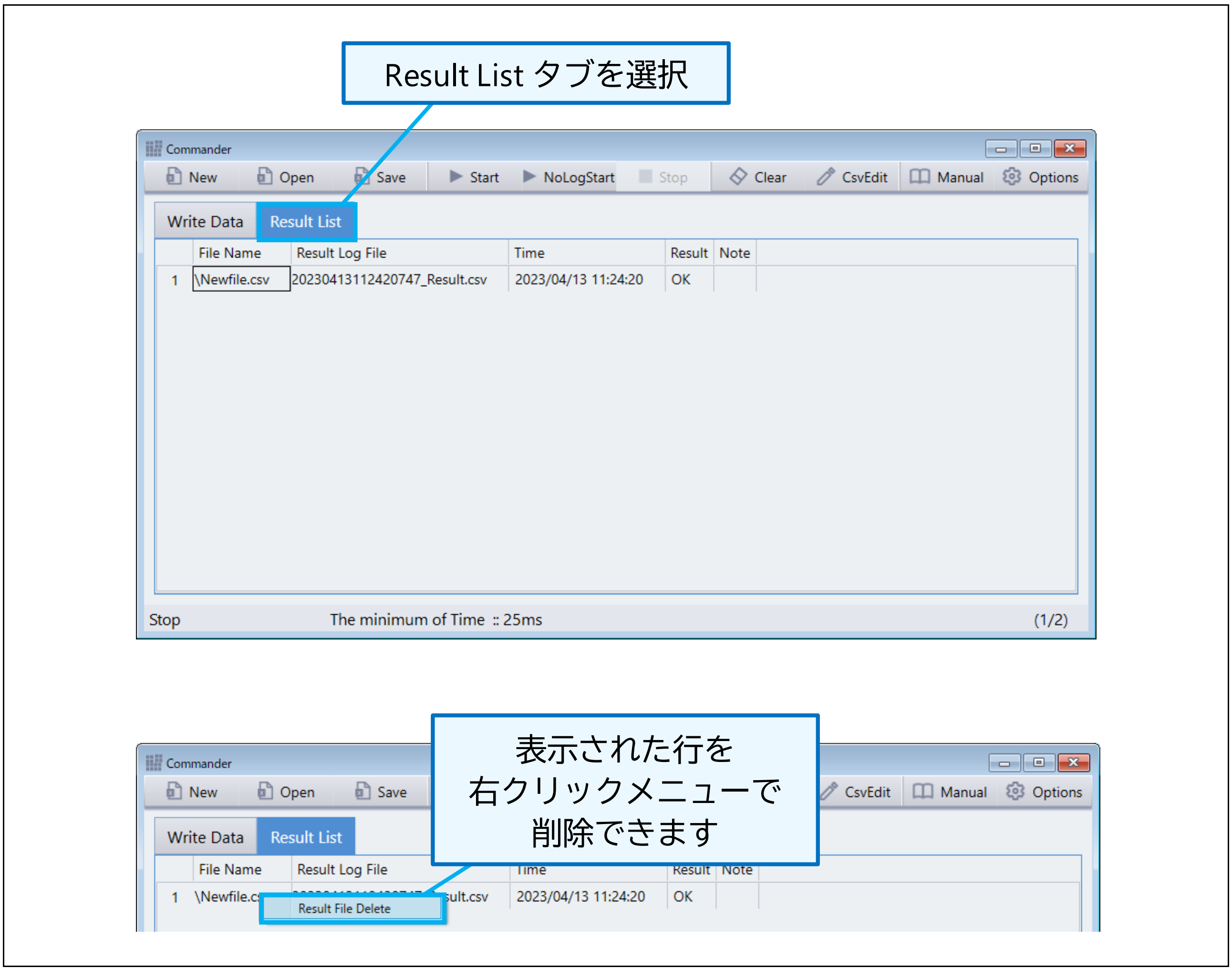


図 7-16　Result List 画面

7.4.7.2　結果レコードの削除

Result List タブに表示された結果レコードは削除できます。削除する結果レコードを右クリックすると表示するメニューから「Result File Delete」を選択すると削除します。

注意　Result List タブから結果レコードを削除すると、Result Log File 欄に表示されたファイル（ログの CSV ファイル）も同時に PC 上から削除されます。

7.4.7.3　結果レコード情報の保存

Result List タブのリストに表示された結果レコードの情報は、RMT ファイルに保存されます。結果レコードを保存した RMT ファイルを読み込んだ場合、結果レコードの情報が復元されます。

8. ［Analyzer］Status Indicator 画面

8.1 概要

Status Indicator は、ユーザプログラムのグローバル変数の値を監視する機能で、監視した結果が設定した条件に合致（閾値越え）すると、画面上で表示灯が点灯して知らせます。Status Indicator は、Analyzer ツールの Control Window にある「Status Indicator」ボタンで起動します。

8.2 特徴

- 変数の値を監視して、条件に合致（閾値越え）したことを表示灯の色で知らせます。
- 1 つの変数に対して、複数の監視条件を設定できます。
- 複数の変数を別々に監視設定することが可能で、複数を同時に監視することも可能です。
- 監視スタート後、いずれの監視条件にも合致しない場合、表示灯は緑色に点灯し、Status は「NORMAL」になります。
- 監視結果の記録（設定条件への適合履歴）をリスト表示します。

8.3 画面構成

Status Indicator の画面構成を以下に示します。

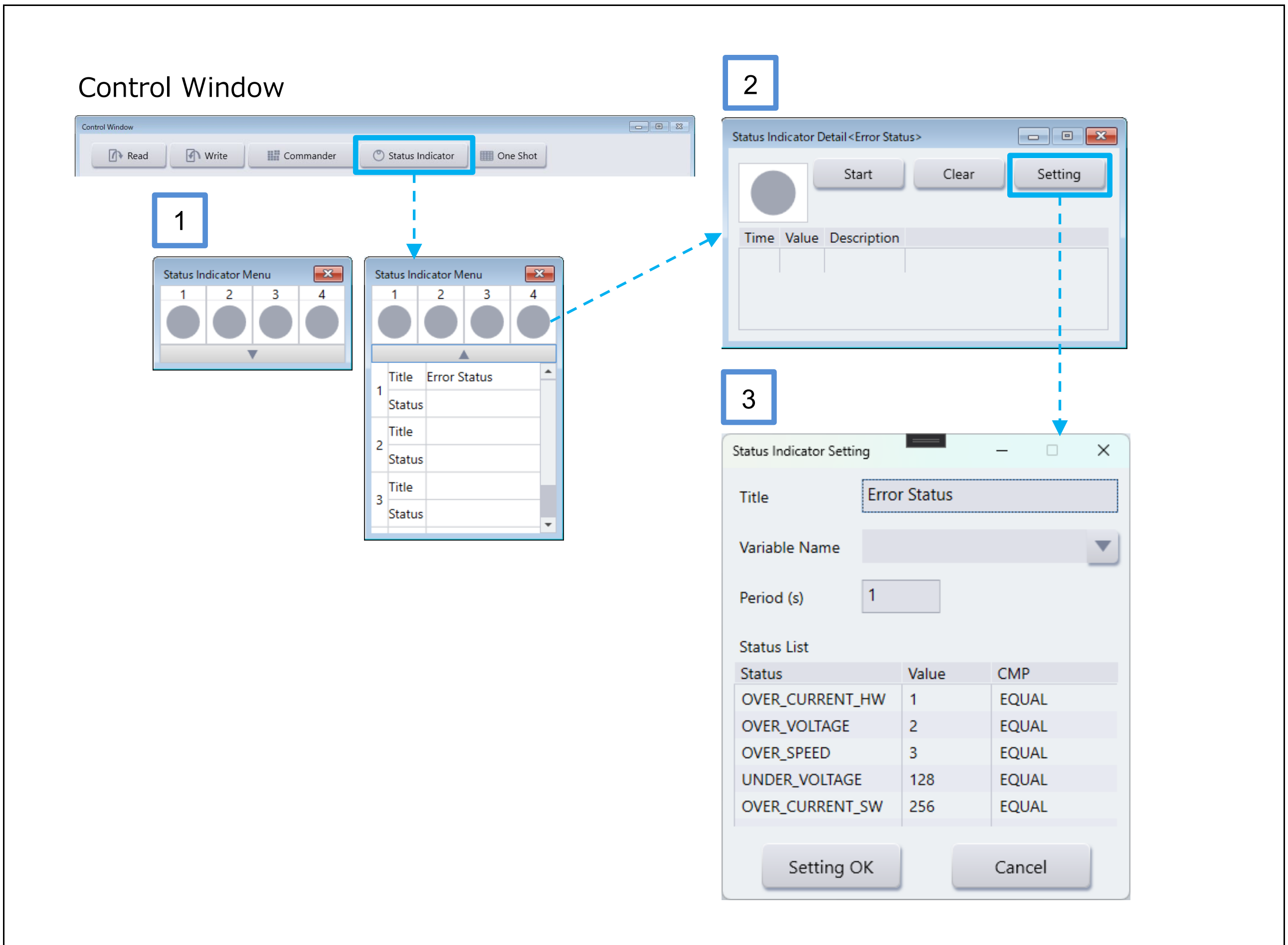


図 8-1 Status Indicator の画面構成

表 8-1 Status Indicator 画面の機能

No.	名称	説明
1	Status Indicator Menu	設定可能な表示灯が表示されます。
2	Status Indicator Detail	監視の開始／停止、監視結果の表示／クリアを行います。
3	Status Indicator Setting	監視する条件の設定をします。

8.3.1 Status Indicator Menu

Status Indicator Menu 画面（以下、Menu 画面）は、Control Window の「Status Indicator」ボタンをクリックすると起動します。Menu 画面から、各監視の状態を表示灯の色で確認することができます。

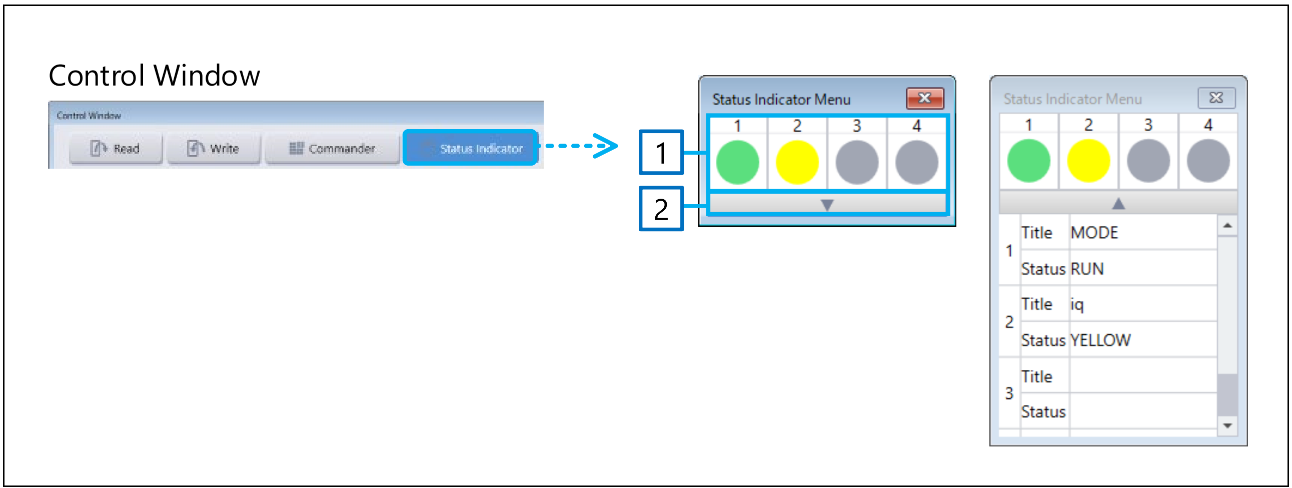


図 8-2 Status Indicator Menu 画面

表 8-2 Status Indicator Menu 画面の機能

No.	名称	説明
1	表示灯	監視状態を色で表示します。
2	情報表示ボタン	表示灯の Title と Status を表示します（▼で表示、▲で隠す）

8.3.2 Status Indicator Detail

Menu 画面の表示灯をクリックすると、その表示灯の Status Indicator Detail 画面（以下、Detail 画面）が開きます。Detail 画面では、監視の開始／停止とログのクリアができます。

画面の「閉じる」ボタンを押したときは、監視の状態に関係なく Detail 画面を閉じます。

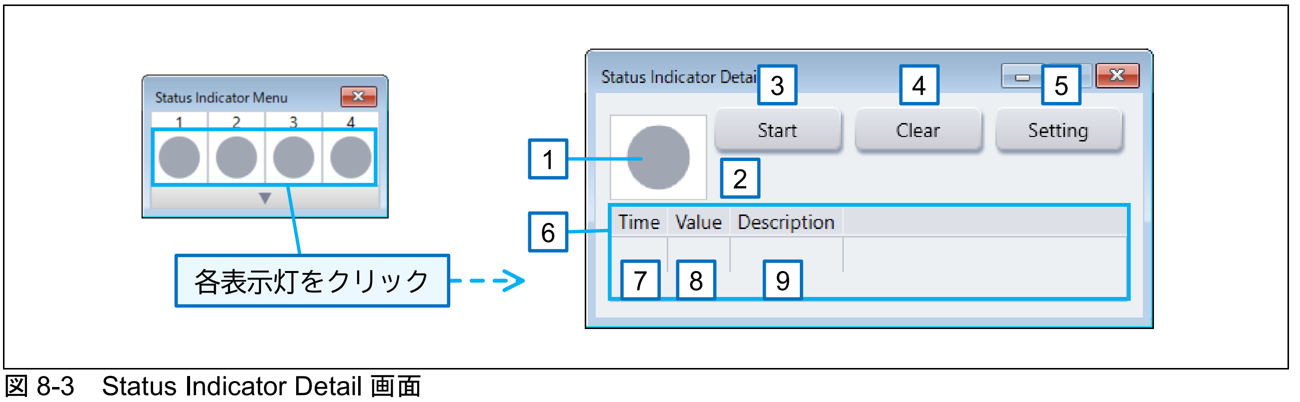


図 8-3 Status Indicator Detail 画面

表 8-3 Status Indicator Detail 画面の機能

No.	名称	説明
1	表示灯	監視状態を色で表示します。
2	Status	監視状態をテキストで表示します。
3	Start／Stop ボタン	ボタンをクリックすると、監視の開始／停止をします。
4	Clear ボタン	監視結果リストをクリアします。
5	Setting ボタン	監視停止中にこのボタンをクリックすると Status Indicator Setting 画面が表示されます（監視中は表示できません）。
6	監視結果リスト	監視中に監視設定に合致（閾値越え）した情報をリスト表示します。
7	Time	監視条件に合致（閾値越え）した日時を表示します。
8	Value	監視条件に合致（閾値越え）した値を表示します。
9	Description	監視条件に合致（閾値越え）した Status を表示します。

8.3.3 Status Indicator Setting

Detail 画面の「Setting」ボタンクリックすると、Status Indicator Setting 画面（以下、Setting 画面）が表示されます。Setting 画面では、監視条件の詳細を設定できます。

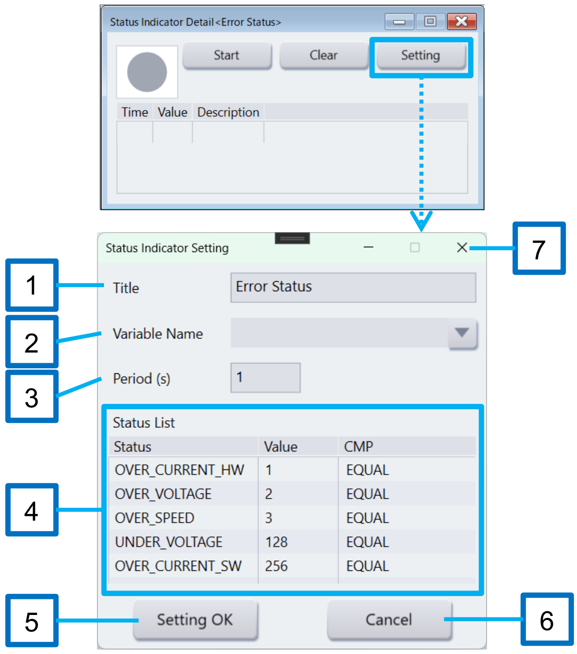


図 8-4 Status Indicator Setting 画面

表 8-4 Status Indicator Setting 画面の機能

No.	名称	説明
1	Title	監視の名称を設定します。
2	Variable Name	監視する変数名を設定します。
3	Periodic	監視する間隔を設定します（初期値は 0 秒）。
4	Status List	監視条件を設定します。
5	Setting OK	設定を保存して画面を閉じます。
6	Cancel	設定を保存せずに画面を閉じます。
7	閉じるボタン	設定を保存せずに画面を閉じます。

8.4　操作説明

8.4.1　Menu 画面の表示／非表示

Control Window の「Status Indicator」ボタンをクリックすると、Menu 画面が表示されます。

Menu 画面右上の「閉じる」ボタンをクリックすると、Menu 画面（およびは Detail 画面）は非表示になります。監視中状態で Menu 画面の「閉じる」ボタンを押して画面を非表示にしても、監視中状態は保持されます。

8.4.2　Setting 画面（監視条件の設定）

Menu 画面で表示灯をクリックすると Detail 画面が表示され、Detail 画面の「Setting」ボタンのクリックで Setting 画面が表示されます。Status Indicator 動作中は、Setting 画面の編集はできません。

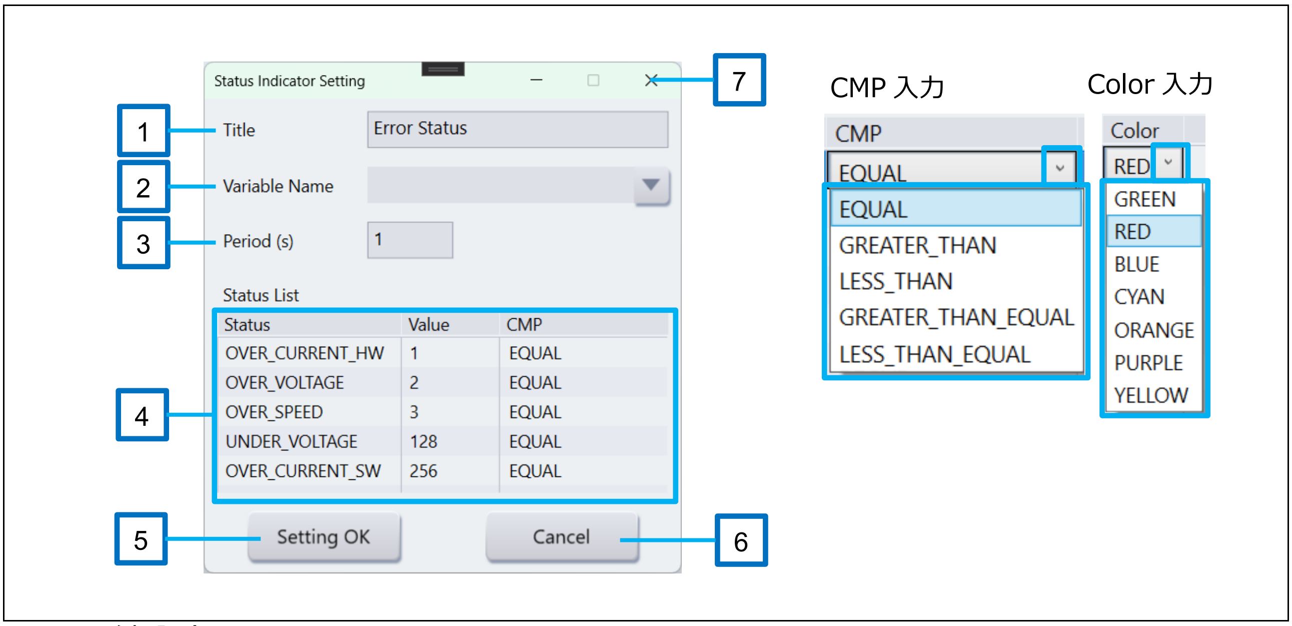


図 8-5　監視設定画面

(1)　Title

監視設定の名称を設定します（直接入力）。

(2)　Variable Name

監視する変数名を設定します（直接入力、またはリスト選択、Variable Find が使用できます）。

(3)　Periodic

監視の間隔を設定します（直接入力）。

(4)　Status List

監視条件を入力します。最終行を編集後、Enter キーを押すことで下に 1 行追加されます。途中行への挿入はできません。行を削除する場合は、行全体が選択された（青くなった）状態で Delete キーを押します（最終行には空白行が残りますが、動作に影響ありません）。

- Status　：条件の名称を設定します（直接入力）。
- Value　：条件値を設定します（直接入力）。
- CMP　：判定条件を設定します（リストから選択）。
- Color　：配色を設定します（リストから選択）。

なお、Color の設定にあたっては、いずれの監視条件にも合致しない場合に表示灯が緑色に点灯することを考慮してください。

(5) Setting OK

設定を保存して画面を閉じます。

(6) Cancel

設定を保存せずに画面を閉じます。

(7) 閉じるボタン

設定を保存せずに画面を閉じます。

8.4.3 監視の開始

表示灯は個別に監視を開始できます。表示灯の Detail 画面で「Start」ボタンをクリックすると監視を開始します。監視を開始すると、Menu 画面と Detail 画面の表示灯の色が、条件に応じて Setting 画面で設定した色に切替ります。いずれの監視条件にも合致しない場合、表示灯は緑色に点灯し、Status は「NORMAL」になります。

8.4.4 監視の停止

監視中の Detail 画面から「Stop」ボタンをクリックすると監視を停止します。監視を停止すると表示灯は灰色になります。

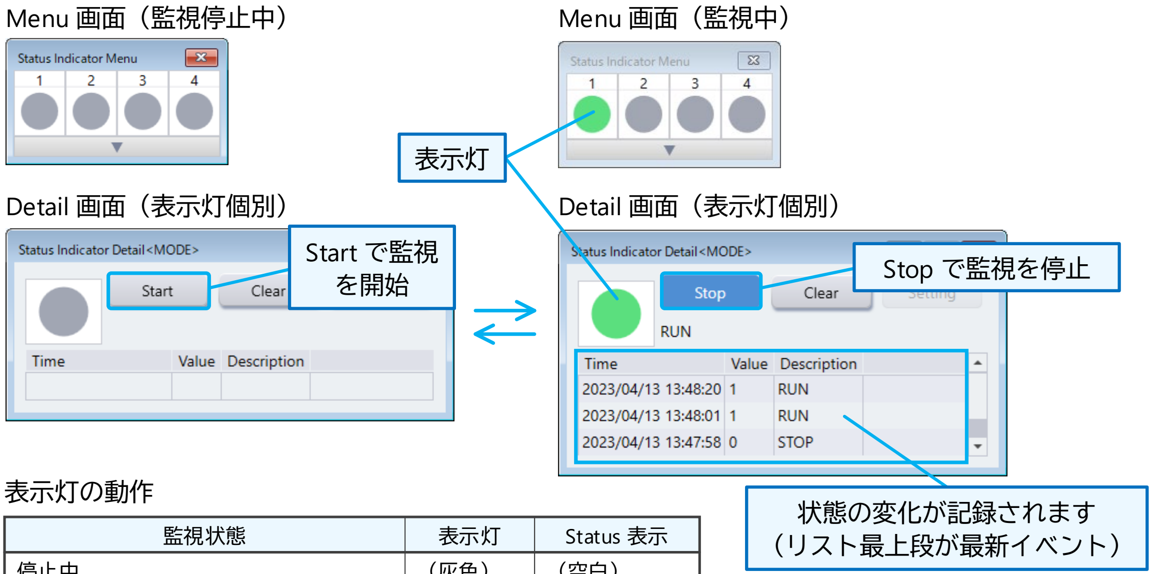
8.4.5 監視中の表示

監視中に合致する監視条件がない場合、表示灯は緑色に点灯し、Status は「NORMAL」になります。変数の値が監視条件に合致（閾値越え）すると次のように動作します。

- Menu 画面と Detail 画面の表示灯の色が、監視条件で設定した色になります。
- Detail 画面の監視結果リストに監視条件に合致（閾値越え）した情報を追記します。

Detail 画面の監視結果リストは、リスト最上部に最新情報が追加されます（自動ではスクロールしません）。

Menu 画面や Detail 画面を閉じても、監視中の状態は保持（継続）されます。



表示灯の動作

監視状態		表示灯	Status 表示
停止中		（灰色）	（空白）
監視中	いずれの条件にも合致しない	緑色	NORMAL
	条件に合致（閾値越え）	設定した色	設定した Status

図 8-6 監視の開始／停止

8.4.6　監視結果のクリア

Detail 画面の「Clear」ボタンをクリックすると、Detail 画面の監視結果リストがクリアされます。
「Clear」ボタンは、監視中／監視停止中の状態とは関係なく操作できます。

8.4.7　監視中にできない操作

監視中は、下記の操作は抑止されます。操作するには、すべての監視を停止してください。

- RMW の終了
- RMT ファイルの読み込み（Open RMT File）
- Map ファイルの読み込み（Load Variable Data）
- ツールの切り替え

上記の操作ができない場合、Detail 画面を閉じている表示灯、または使用していない表示灯が、監視中になっていないか確認してください。

8.4.8　RMT ファイルへの保存

Status Indicator の設定情報は RMT ファイルに保存されますが、監視結果リストの内容を保存することは出来ません。

9. ［Analyzer］One Shot 画面

9.1 概要

One Shot は、指定した変数から連続したアドレスのデータを一括取得して波形で表示する機能です。
One Shot は、Analyzer ツールの Control Window にある「One Shot」ボタンで起動します。

9.2 特徴

- バッファリングされたデータを波形で表示することが可能です。

9.3 画面構成

One Shot の画面構成を以下に示します。

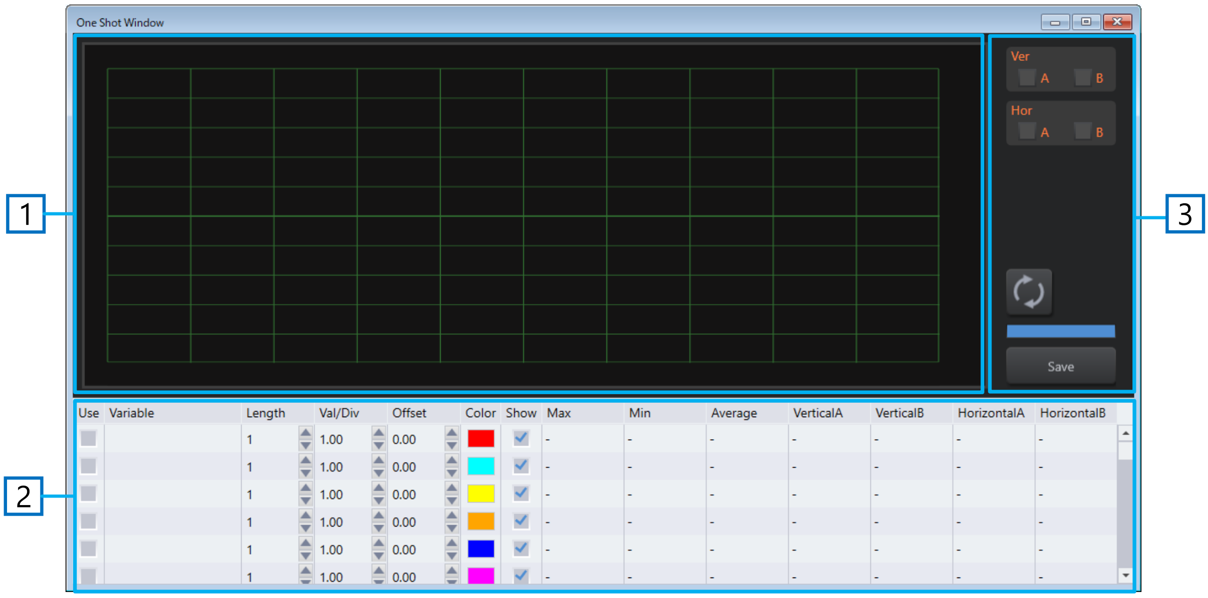


図 9-1 One Shot 画面

表 9-1 One Shot 画面の機能

No.	名称	説明
1	波形表示部	取得したデータを波形で表示します。
2	チャネル情報部	チャネル情報の設定と表示を行います。
3	取得設定部	カーソルの設定、取得開始（リロード）等を行います。

9.3.1 チャネル情報部

チャネル情報部の構成を以下に示します。

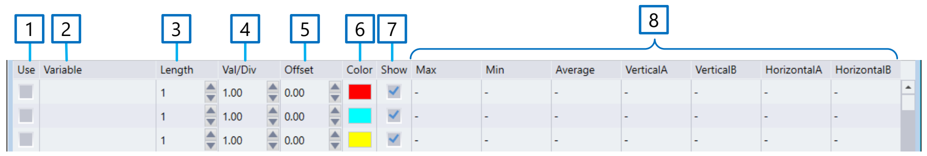


図 9-2 チャネル情報部

表 9-2 チャネル情報部の機能

No.	名称	説明
1	Use	波形取得するチャネルを選択します。
2	Variable	波形取得する変数を指定します。
3	Length	波形取得するデータ数を指定します。
4	Val/Div	波形表示時の縦軸 1 div の値を指定します。
5	Offset	波形表示時の縦軸オフセット値を指定します。
6	Color	表示する波形の色を選択できます。
7	Show	波形を表示するチャネルを選択します。
8	情報部	波形の情報を表示します。

9.3.2 取得設定部

取得設定部の構成を以下に示します。

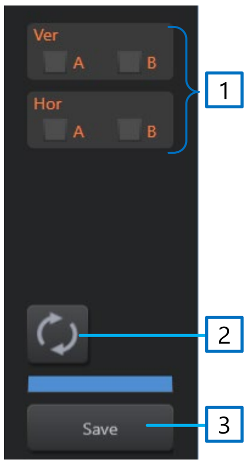


図 9-3 取得設定部

表 9-3 取得設定部の機能

No.	名称	説明
1	カーソル設定	垂直（Ver）カーソルと水平（Hor）カーソルの表示／非表示を設定します。
2	Graph Reload ボタン	クリックするたびに新たにデータを取得します。
3	Save ボタン	取得したデータを保存します。

9.4 操作説明

9.4.1 One Shot 画面の表示

One Shot 画面は、Control Window の「One Shot」ボタンをクリックすると起動します。

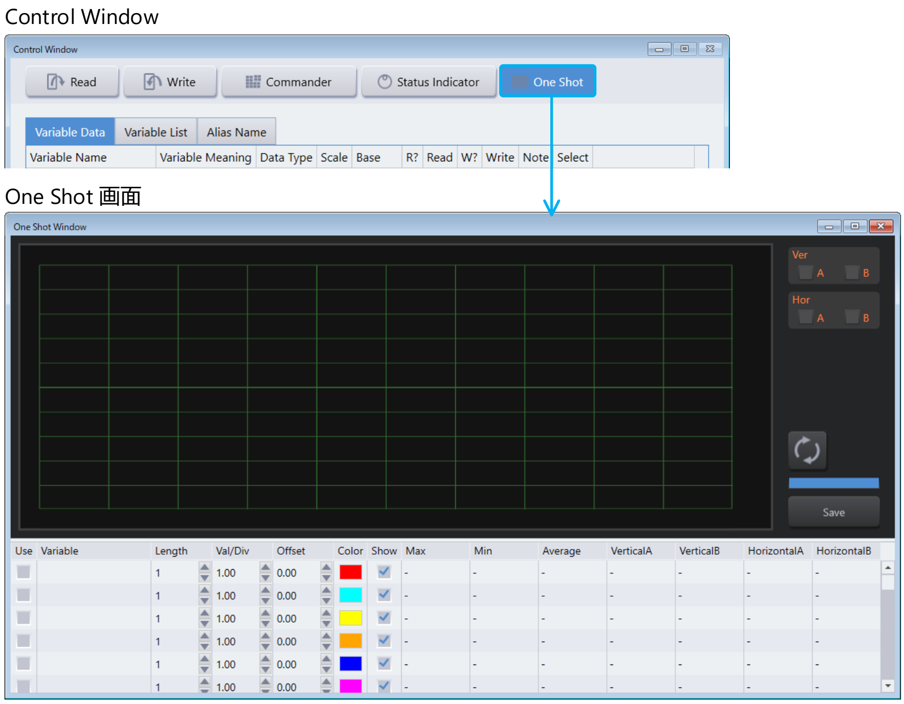


図 9-4 One Shot 画面の起動

9.4.2 チャネル情報部の設定

One Shot 画面のチャネル情報部の設定内容を以下に示します。

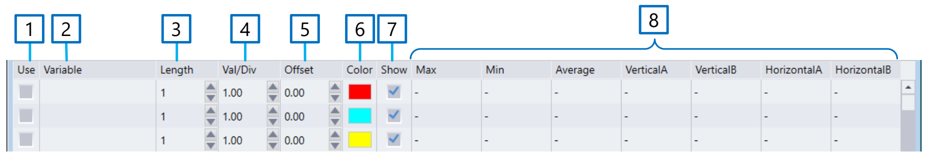


図 9-5 チャネル情報部

(1) Use

チェックボックスを ON にすると、そのチャネルの情報を取得します。

(2) Variable

取得するデータの先頭アドレスの変数名を設定します。

(3) Length

取得するデータ数を設定します。

(4) Val/Div

データを波形表示する際の、縦軸 1 div の値を入力します。入力ボックス右側▼▲でも変更できます。

(5) Offset

データを波形表示する際の、縦軸オフセットの値を入力します。入力ボックス右側▼▲でも変更できます。

(6) Color

波形表示の色を選択します。

(7) Show

チェックボックスを ON にすると波形を表示します。

(8) 情報部

取得したデータの情報（最大、最小、平均、各カーソル位置の値）が表示されます。

9.4.3 データの取得

データの取得は、取得設定部の「Graph Reload」ボタンをクリックします（クリックするたびにデータが更新されます）。

9.4.4 カーソルの設定

Scope Window 右上の Cursor のチェックボックスを ON にすると、Hor（水平）と Ver（垂直）のカーソルを各 2 本表示できます。波形表示部で各カーソルをマウスでドラッグして、位置の調整ができます。

なお、チャネル情報部にカーソル位置の波形データの値が表示されます。

9.4.5 取得データの保存

取得設定部の「Save」ボタンをクリックすると、取得したデータを保存することができます。

10. ［Analyzer］Parameter Output

10.1 概要

Parameter Output は、Analyzer で調整したパラメータを、モータ制御プログラムのヘッダファイルとして出力する機能です。Parameter Output は、Analyzer ツールの Control Window にある「Output Header」ボタン、または User Button Control Window のメニュー「Output Header File」で起動します。

10.2 特徴

- Control Window または User Button で調整した変数を、マクロ定義としてヘッダファイルへ出力します。
- 選択画面上で、ヘッダファイルへ出力するマクロ定義に紐づける変数を選択する事が可能です。
- 選択画面上で新規マクロ名称を入力すれば、ヘッダファイルの末尾に追加して出力可能です。

10.3 画面構成

Parameter Output の画面構成を以下に示します。

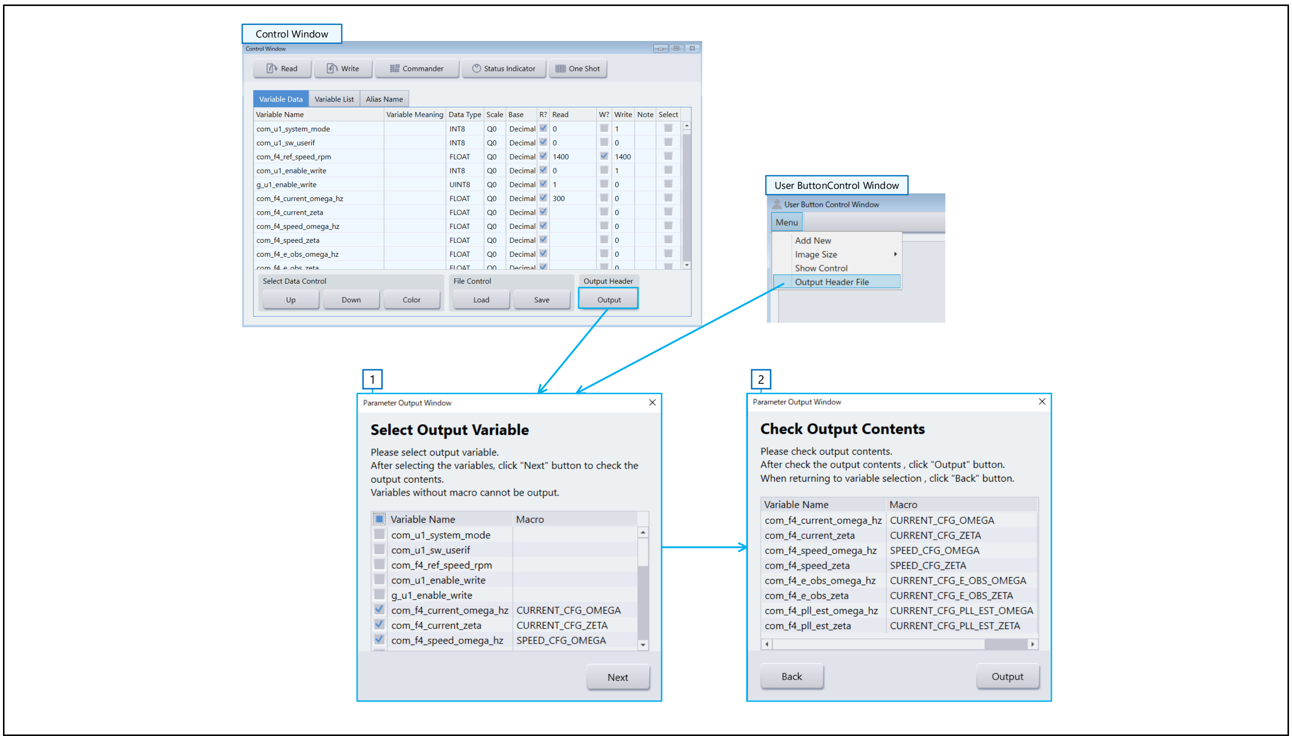


図 10-1 Parameter Output の画面構成

表 10-1 Parameter Output 画面の機能

No.	名称	説明
1	出力変数選択画面	出力する変数の選択と出力するマクロ名の入力を行います。
2	出力内容確認画面	出力する内容の確認とヘッダファイルの出力を行います。

10.3.1 出力変数選択画面

出力変数選択画面は、Control Window にある「Output Header」ボタン、または User Button Control Window のメニュー「Output Header File」から起動します。出力変数選択画面から、ヘッダファイルに出力するマクロ定義と変数を選択することができます。

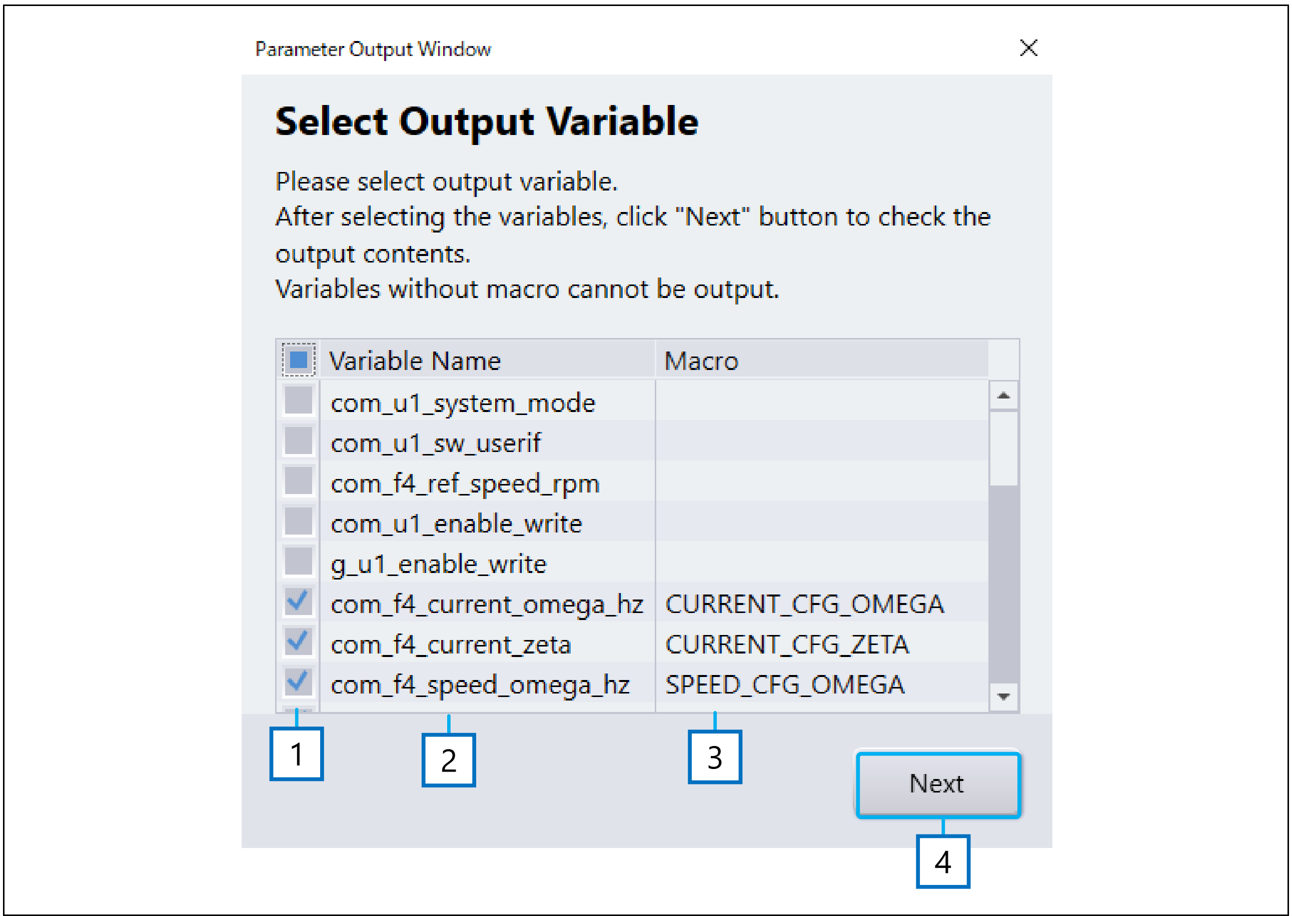


図 10-2　出力変数選択画面

表 10-2　出力変数選択画面の機能

No.	名称	説明
1	出力変数選択チェックボックス	マクロ定義として出力する変数を選択します。
2	Variable Name	Control Window 及び User Button で読み込み対象になっている変数を表示します。
3	Macro	マクロ名を出力します。空欄の場合は出力するマクロ名を入力します。
4	Next ボタン	出力内容確認画面に遷移します。

10.3.2 出力内容確認画面

出力内容確認画面は、出力変数選択画面で「Next」ボタンを押下することで表示されます。出力内容確認画面では、ヘッダファイルに出力する変数、マクロ名、値を確認できます。

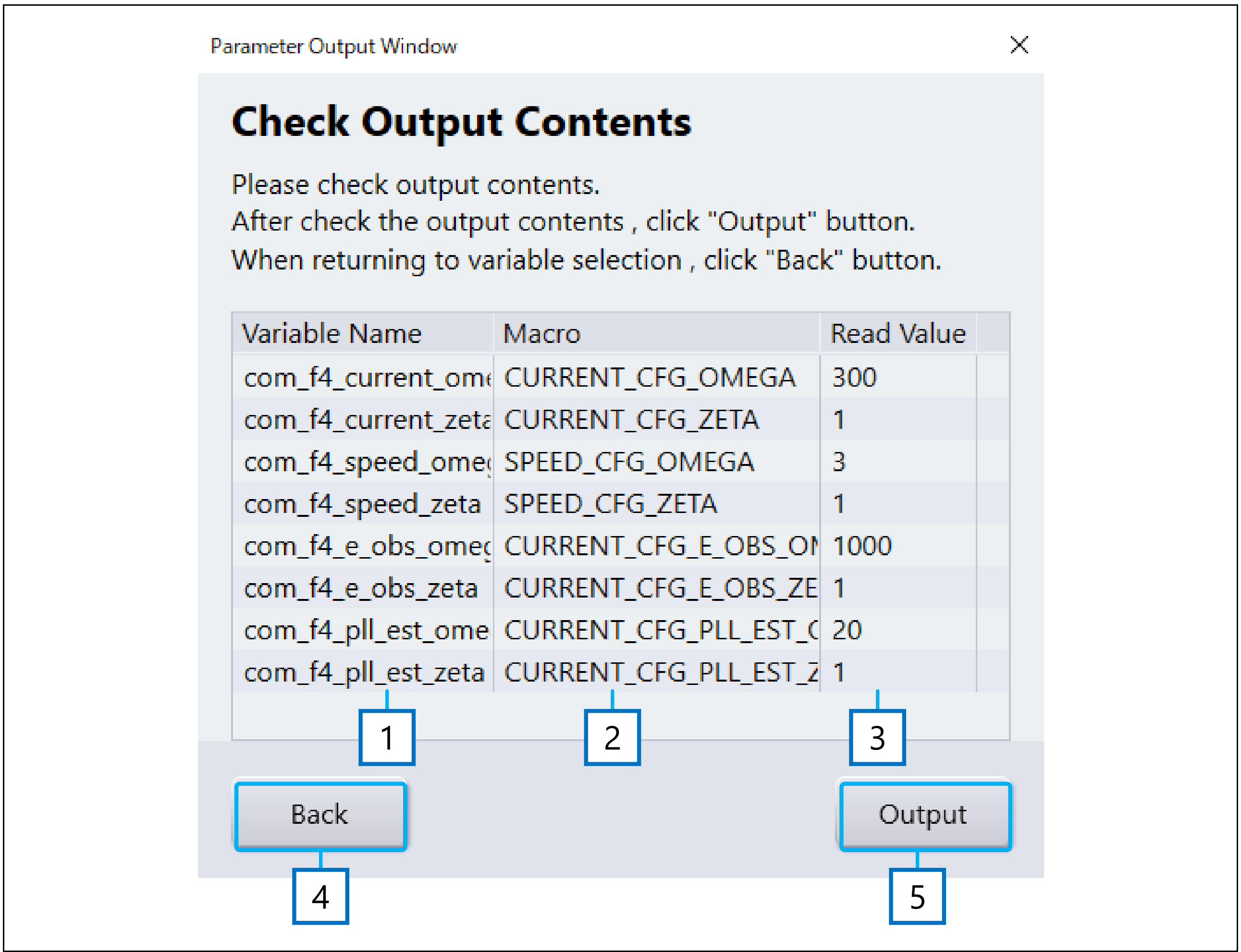


図 10-3 出力内容確認画面

表 10-3 出力内容確認画面の機能

No.	名称	説明
1	Variable Name	マクロ定義として出力する変数を表示します。
2	Macro	出力するマクロ名を表示します。
3	Read Value	マクロ定義として出力する設定値を表示します。 ※モータ制御プログラムに書き込み済みの値を表示します。
4	Back ボタン	出力変数選択画面に戻ります。
5	Output ボタン	出力先を選択してヘッダファイルを出力します。

10.4 操作説明

10.4.1 Parameter Output 画面の表示

Control Window 画面の「Output」ボタン押下、もしくは User Button Control Window 画面のメニューから「Output Header File」を選択することで出力変数選択画面を表示します。

ヘッダテンプレートファイルの情報に含まれる変数の Macro 列には、対応するマクロ名が表示され、出力対象としてチェックボックスがチェックされた状態で表示されます。

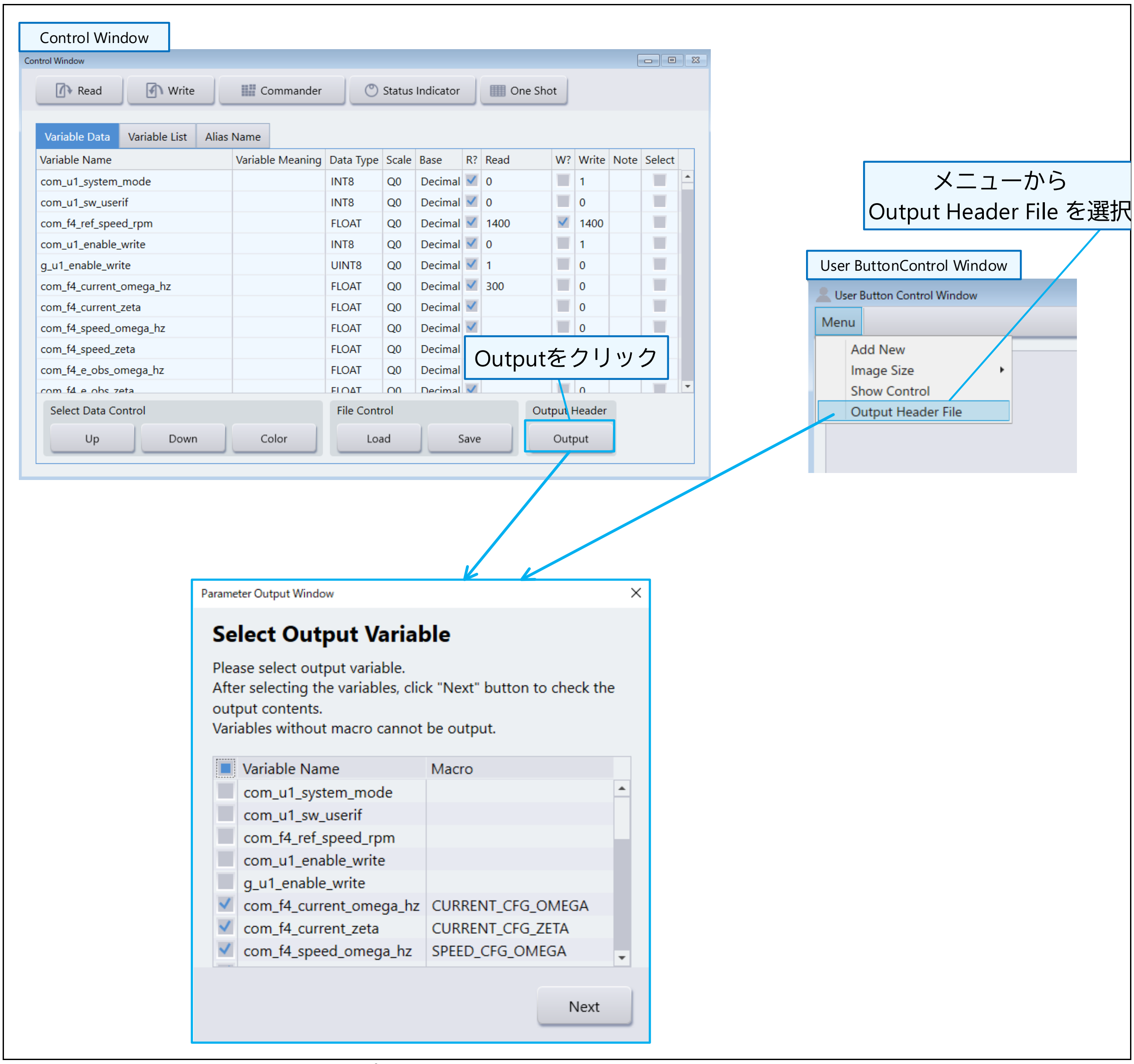


図 10-4 Parameter Output 画面の表示

10.4.2 出力する変数の選択

選択画面に表示されるチェックボックスを選択することにより、ヘッダファイルへ出力する変数とマクロ名称を選択します。

また、ヘッダー行のチェックボックスを切り替えることで、変数の一括選択及び解除をすることができます。

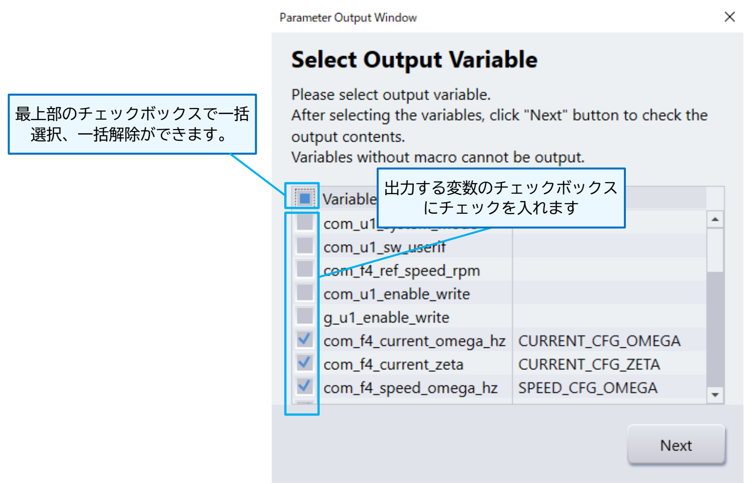


図 10-5 出力する変数の選択

10.4.3 マクロ名の入力

ヘッダファイルへ出力するマクロ名称は、選択画面の Macro 列から編集することができます。

ただし、ヘッダテンプレートファイルの情報に対応するマクロ名を表示しているセルは編集できません。

ヘッダテンプレートファイルについては「10.4.6 ヘッダテンプレートファイルの設定」を参照してください。

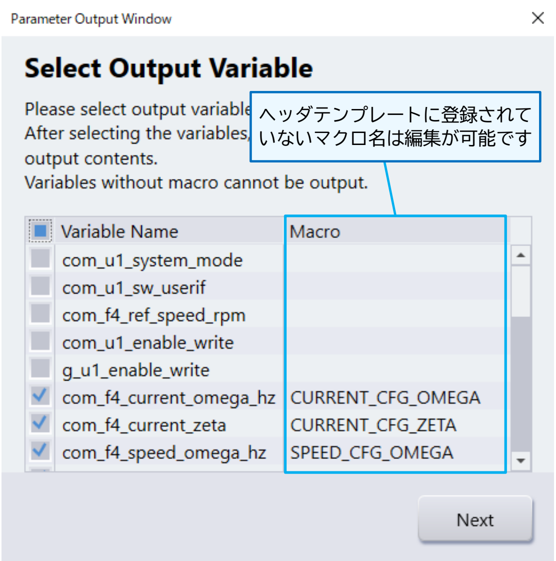


図 10-6 マクロ名の入力

10.4.4 出力内容の確認

出力変数選択画面の「Next」ボタン押下により、出力内容確認画面に遷移することができます。出力内容確認画面には、出力する変数、マクロ名及びモータ制御プログラムに設定された値を表示します。

ただし、マクロ名が入力されていない変数が出力対象に選択されている場合、出力内容確認画面に遷移することはできません。

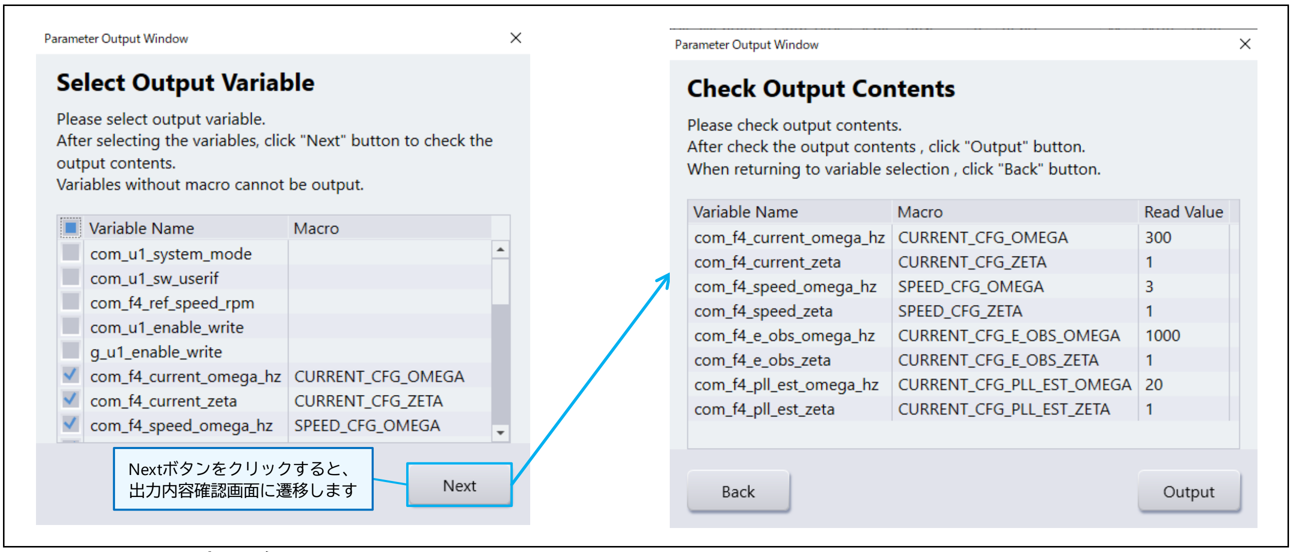


図 10-7 出力内容の確認

10.4.5 ヘッダファイルの出力

出力内容確認画面の「Output」ボタンを押下し、出力先を選択することでヘッダファイルを出力します。ヘッダファイルの内容はヘッダテンプレートファイルの情報と出力内容確認画面の内容を基に生成されます。

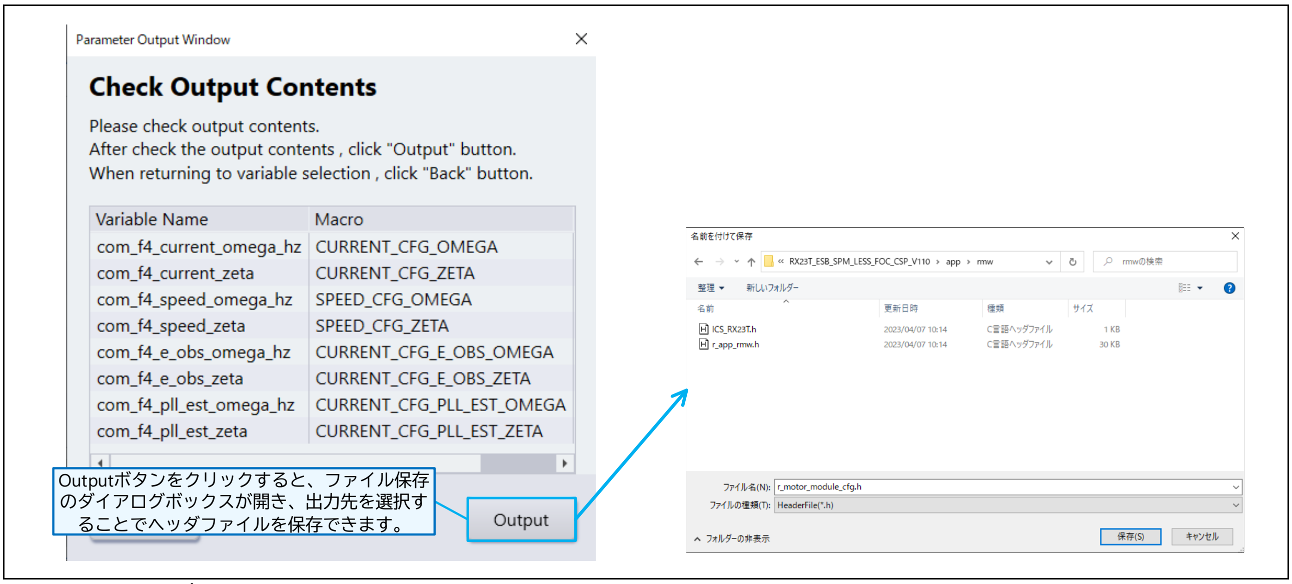


図 10-8 ヘッダファイルの出力

10.4.6 ヘッダテンプレートファイルの設定

出力するヘッダファイルのテンプレートを XML 形式のファイルで作成します。
作成したファイルは、rmt ファイルと同じフォルダ内に保存することで適用されます。
また、ファイル名は以下のように作成してください。
XML ファイル名：OutputHeaderFileInfoForAnalyzer.xml

XML ファイルは、「ヘッダ部」、「データ部」、「フッタ部」で構成され、ルート要素「OutputHeaderFileForAnalyzerSetting」内に記述します。

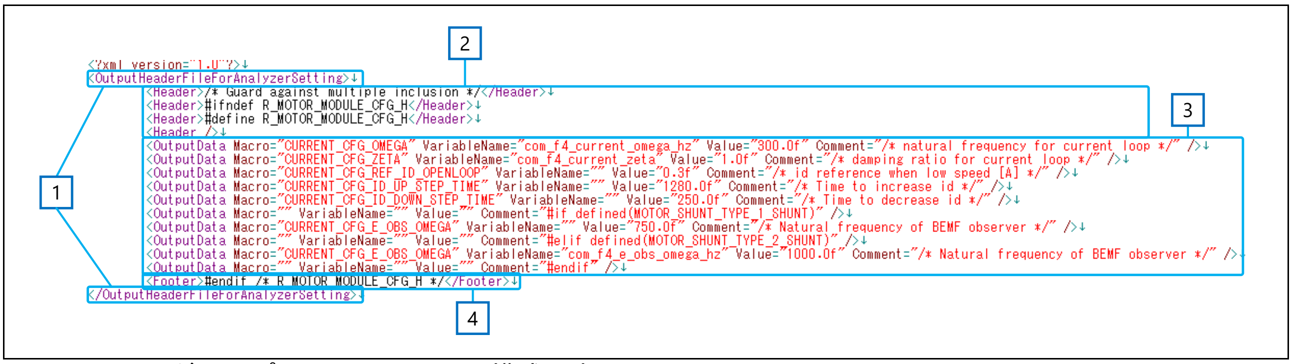


図 10-9 ヘッダテンプレートファイルの構成要素

表 10-4 ヘッダテンプレートファイルの構成要素

No.	名称	要素名	説明
1	ルート要素	OutputHeaderFileForAnalyzerSetting	ヘッダテンプレートファイルのルート要素です。
2	ヘッダ部	Header	出力ファイルのヘッダ部の内容を定義します。
3	データ部	OutputData	出力ファイルのデータ部の内容を定義します。
4	フッタ部	Footer	出力ファイルのフッタ部の内容を定義します。

10.4.6.1 ルート要素

OutputHeaderFileForAnalyzerSetting タブ内にヘッダテンプレートの情報を記述します。

10.4.6.2 ヘッダ部

Header タブに記載されたテキストをファイルに出力します。

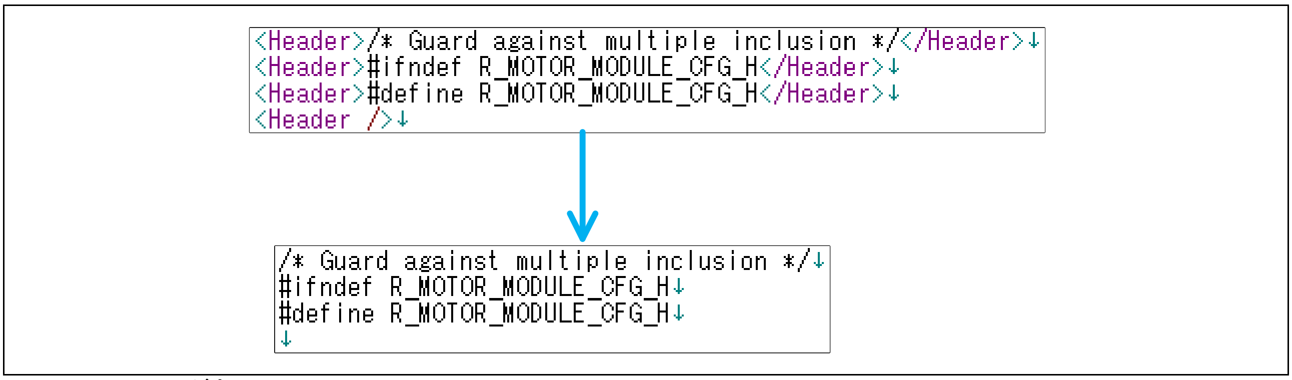


図 10-10 ヘッダ部

10.4.6.3 データ部

OutputData タブは以下の属性で構成します。

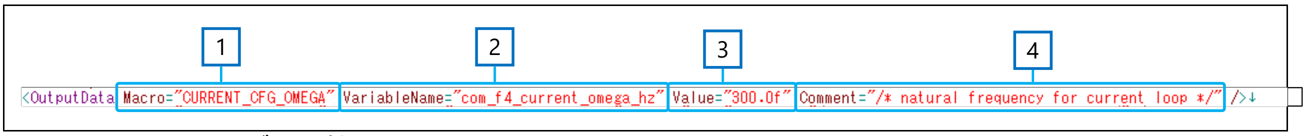


図 10-11 Output タブの属性

表 10-5 OutputDate の属性

No.	属性	説明
1	Macro	マクロ名を定義します。
2	VariableName	マクロ名に対応する変数名を定義します。
3	Value	マクロ定義の初期値を定義します。
4	Comment	マクロのコメントを定義します。

Parameter Output は、OutputData タブに記述した各属性の内容を結合して、ファイルに出力します。
各属性の記述有無と、Parameter Output 画面の選択内容により、3 種類の出力パターンがあります。

① VariableName属性が記述されていて、かつ記述されているVariableNameが出力対象に選択されている場合、Value属性に記述した数値をRead Valueの値に置き換えて出力します。

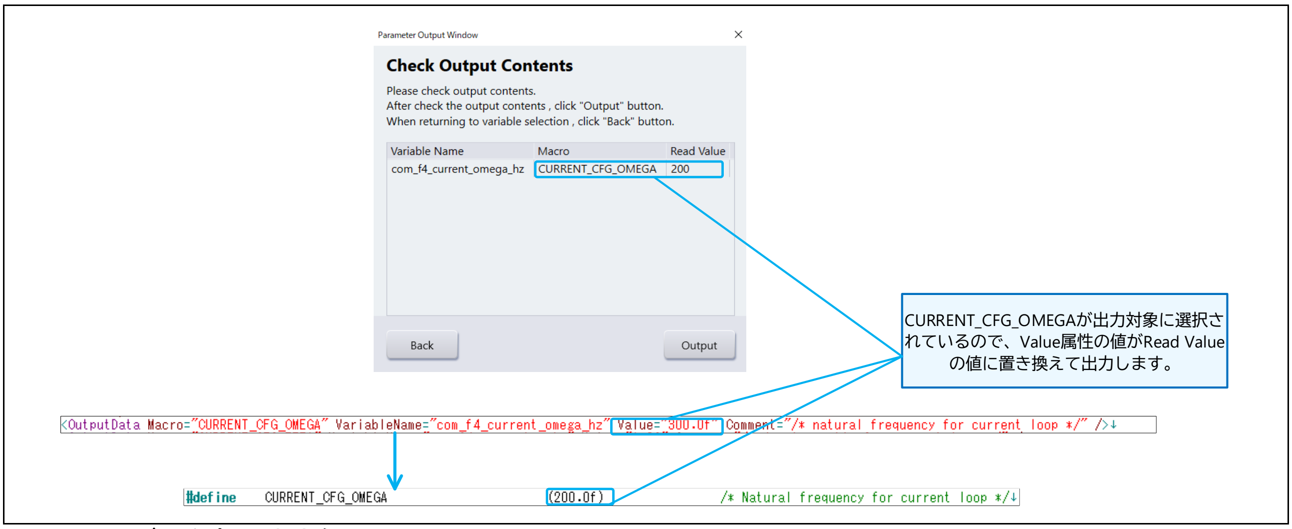


図 10-12 データ部の出力例

② VariableName属性が記述されていない、もしくは記述されているVariableNameが出力対象に選択されていない場合、Value属性の記述をそのまま出力します。
VariableName 属性を記述しない場合、Macro 属性に記述したマクロ名と Value 属性に記述した値を結合して、マクロ定義を出力します。

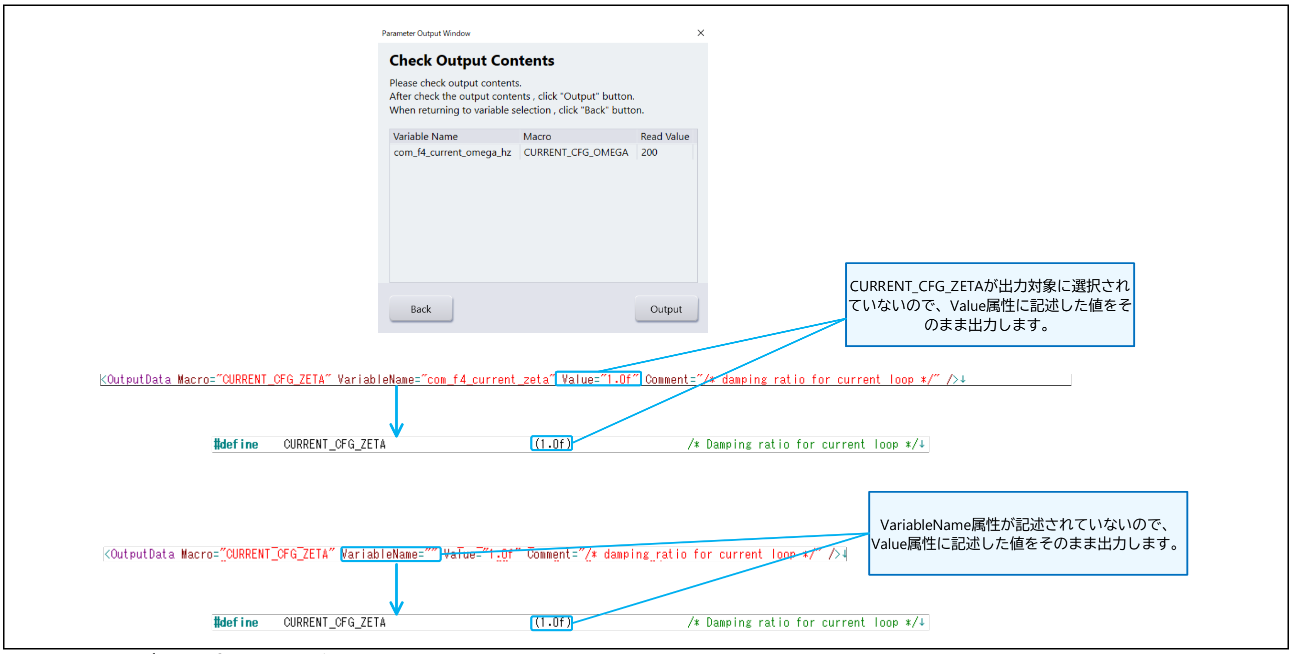


図 10-13　データ部の出力例

③ Macro属性が記述されていない場合、Comment属性の記述のみを出力します。
コンパイルオプションやコメント行などマクロ定義以外の記述を出力する際に使用します。

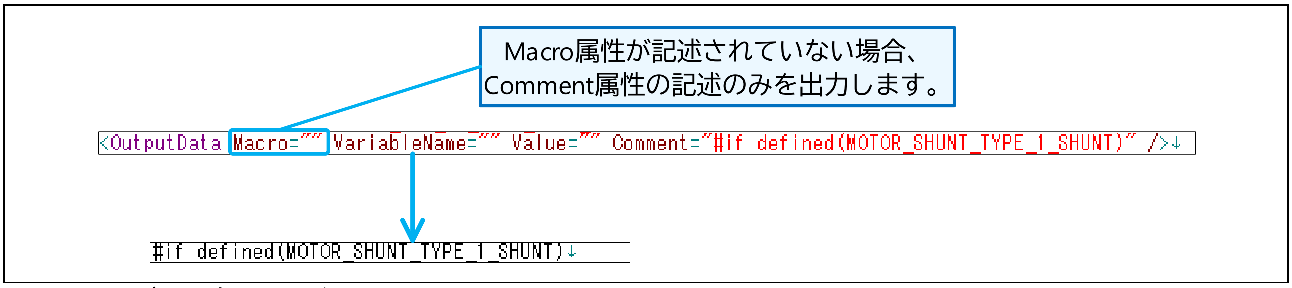


図 10-14　データ部の出力例

10.4.6.4　フッタ部

Footer タブに記述されたテキストをファイルに出力します。

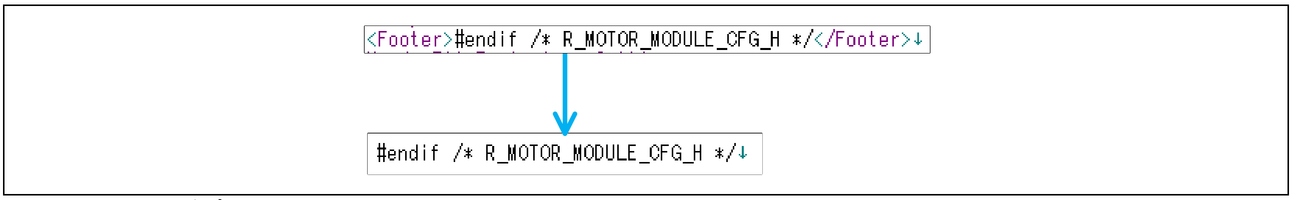


図 10-15　フッタ部

11. Navigation

11.1 概要

Navigation は、Renesas Motor Workbench の各機能の操作手順を説明する機能です。Navigation は、デフォルト設定では Analyzer および Tuner 起動時に表示されます。

画面構成は共通なので、本章では Analyzer の Navigation を例に説明します。

11.2 特徴

- 説明文と画像で操作手順を表示します。

11.3 画面構成

11.3.1 Navigation 画面

Navigation の画面構成を以下に示します。

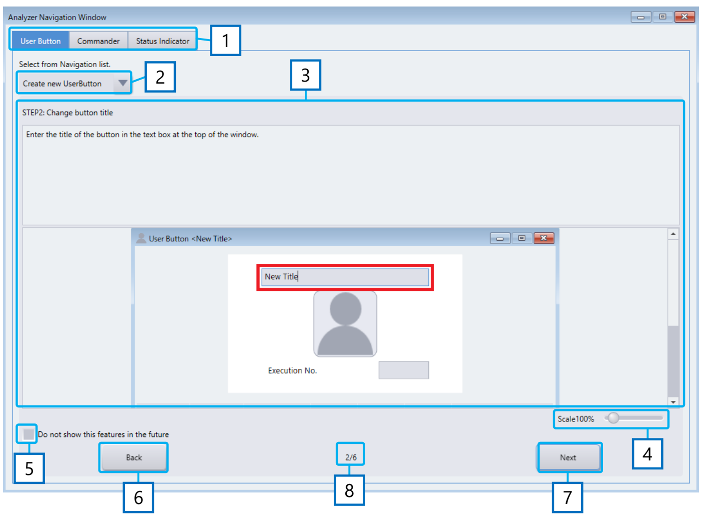


図 11-1 Navigation 画面

表 11-1 Navigation 画面の機能

No.	名称	説明
1	機能選択タブ	ナビゲーションを表示する機能を選択します。
2	操作手順選択ボックス	確認したい操作手順を選択します。
3	操作手順表示	選択した機能の操作手順を表示します。
4	画像拡大率スライダ	表示中の画像の拡大率を変更します。
5	自動起動無効化チェックボックス	Navigation 画面の自動起動の有効・無効を変更します。
6	Back ボタン	前のページに移動します。
7	ページ番号表示	現在のページ番号と最大ページ番号を表示します。
8	Next ボタン	後のページに移動します。

11.4 操作説明

11.4.1 表示機能の切り替え

Navigation 画面の左上に表示されるタブから、ナビゲーション対象とする機能を切り替えることができます。

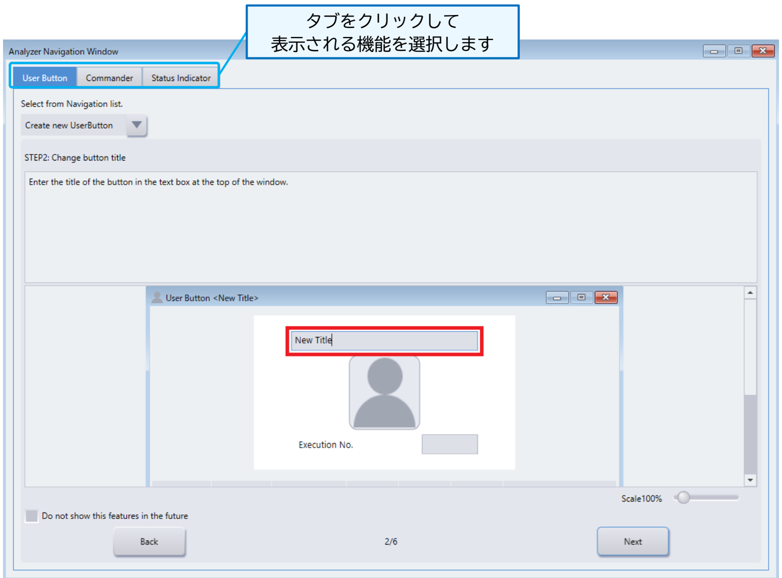


図 11-2 表示機能の切り替え

11.4.2 操作手順の選択

Navigation 画面の操作手順選択ボックスで選択した操作手順を、画面に表示することができます。

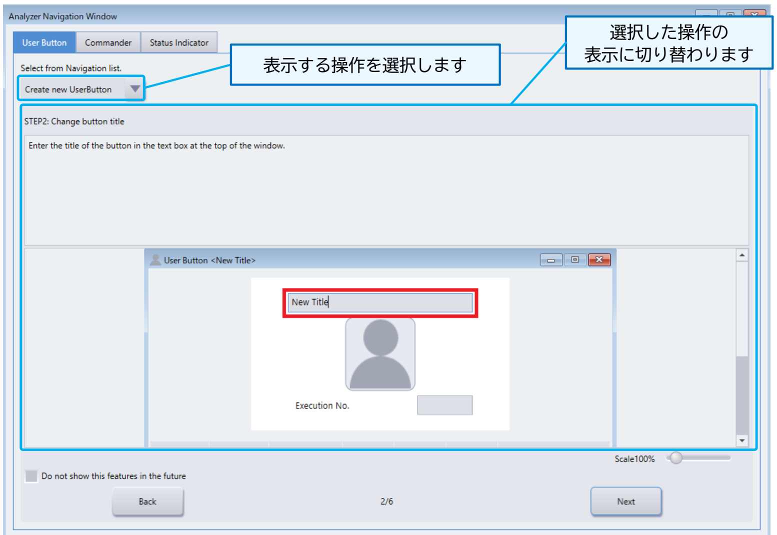


図 11-3 操作手順の選択

11.4.3 画像拡大率の変更

画面に表示された画像は、スライダまたは Ctrl キー+マウスホイールにより、表示サイズを変更することができます。

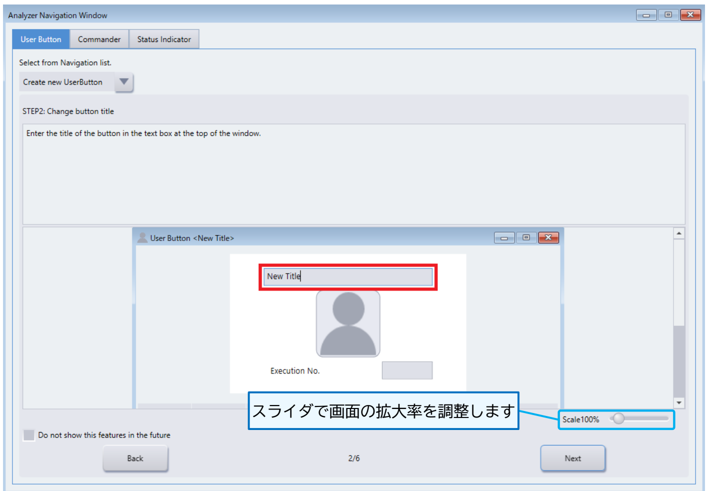


図 11-4 画像拡大率の変更

11.4.4 自動起動の変更

自動起動無効化チェックボックスにチェックすると、Analyzer および Tuner 起動時の Navigation 自動起動を無効化できます。

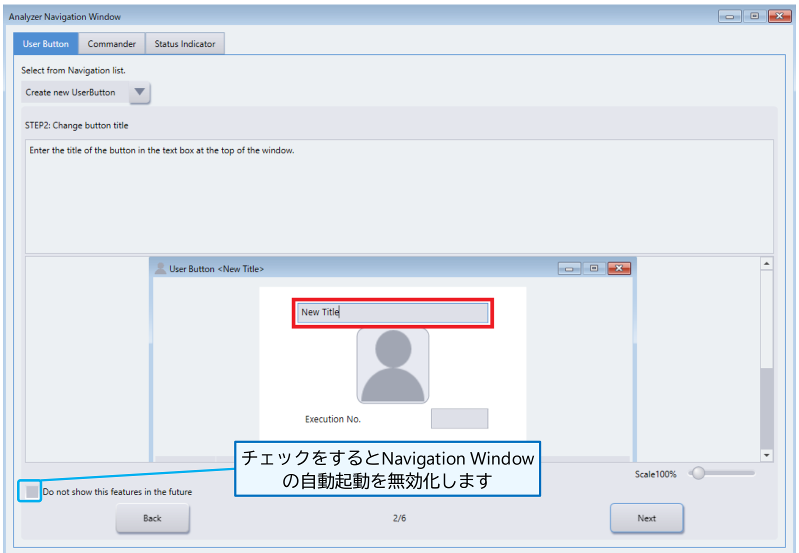


図 11-5 Navigation 自動起動の変更

自動起動を無効にした場合は、Help メニューの Analyzer Navigation または Tuner Navigation を選択すると表示することができます。

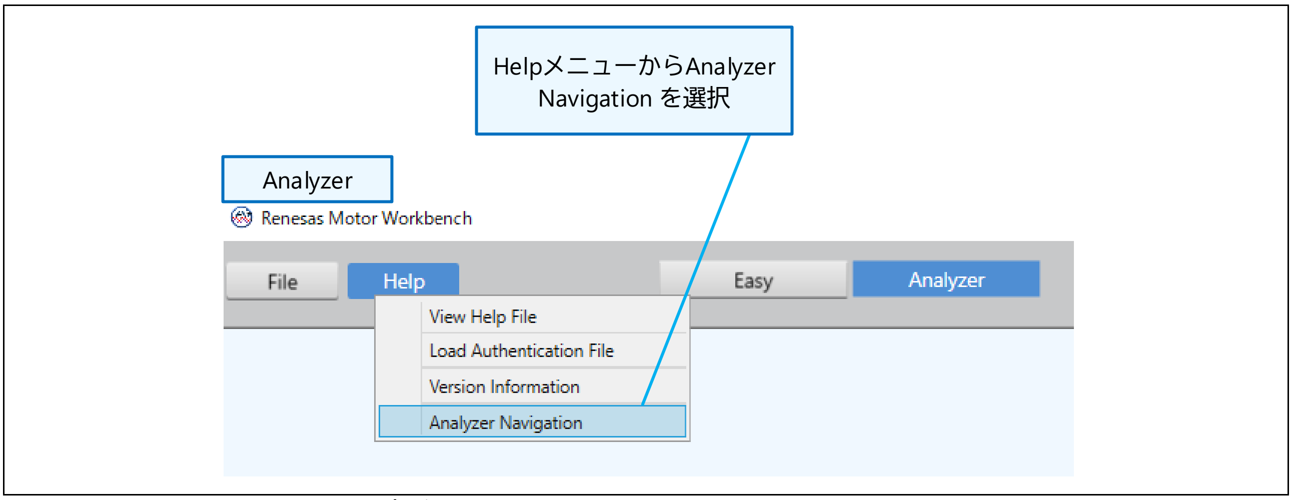


図 11-6 Analyzer Navigation の起動

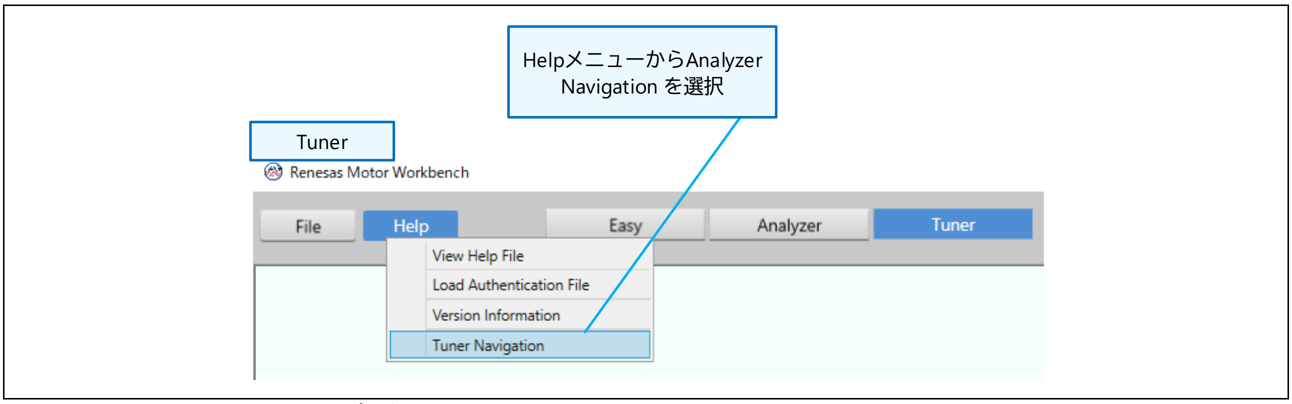


図 11-7 Tuner Navigation の起動

11.4.5 ページの移動

Back ボタンもしくは Next ボタンを押下すると次のナビゲーションページへ移動します。

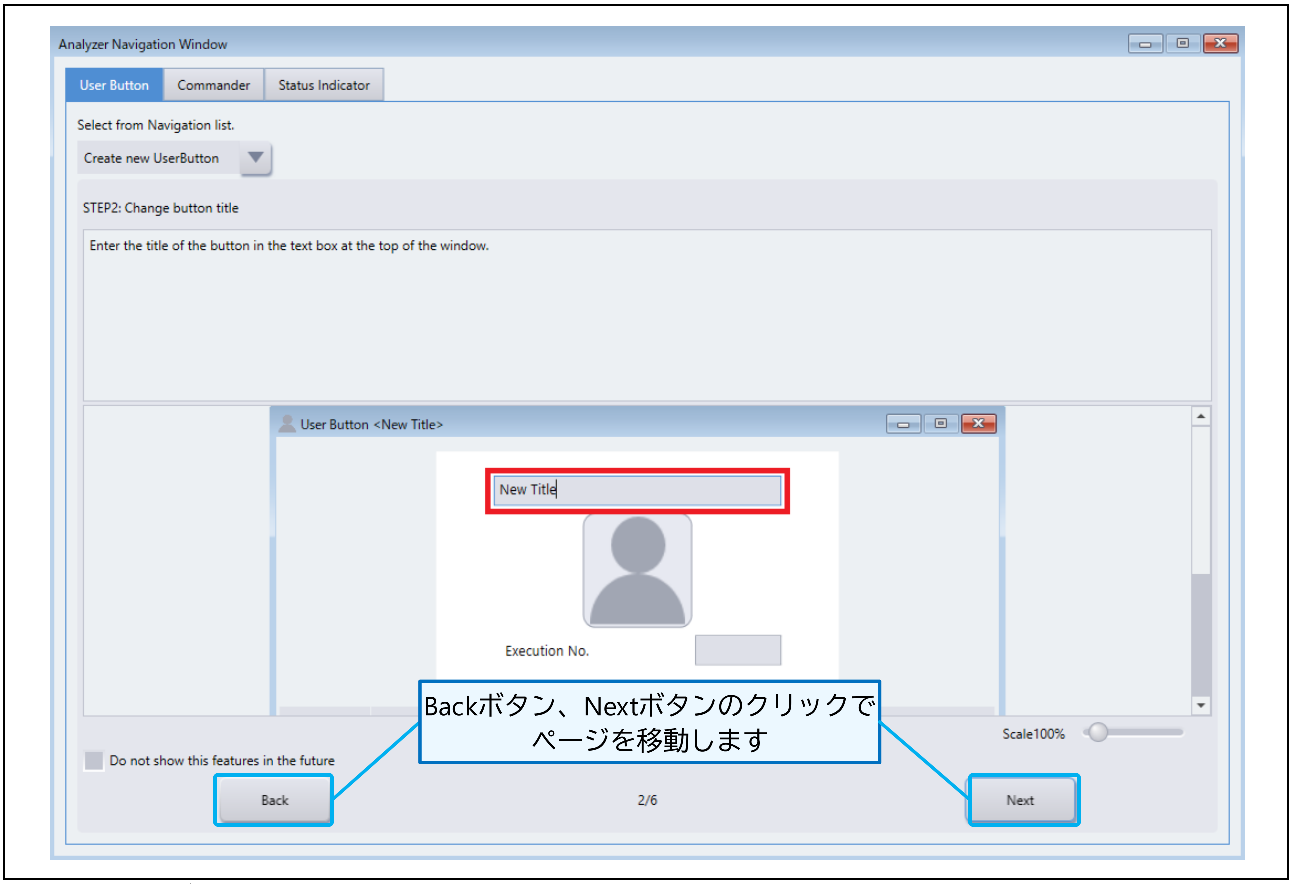


図 11-8 ページの移動

12. Easy GUI

12.1 概要

Easy GUI は、モータの駆動を簡単に操作し計測することができる専用の GUI ツールです。

アナライザよりもシンプルな GUI 構成のため、モータに不慣れな場合も簡単にモータを操作することができます。また、モータの状態を GUI で視覚的にみることができるため、デモンストレーション等の用途にも活用することができます。

12.2 特徴

Easy GUI を使用することで、スライダやシーケンスを使って直感的な操作でモータの速度や位置を簡単に設定でき、モータの状態をメータやグラフで視覚的に計測することができます。また、順次変数の値を変更するシーケンス動作を繰り返し実行することもできます。

GUI に表示する変数は、ユーザのプログラムに応じて任意に設定することができます。

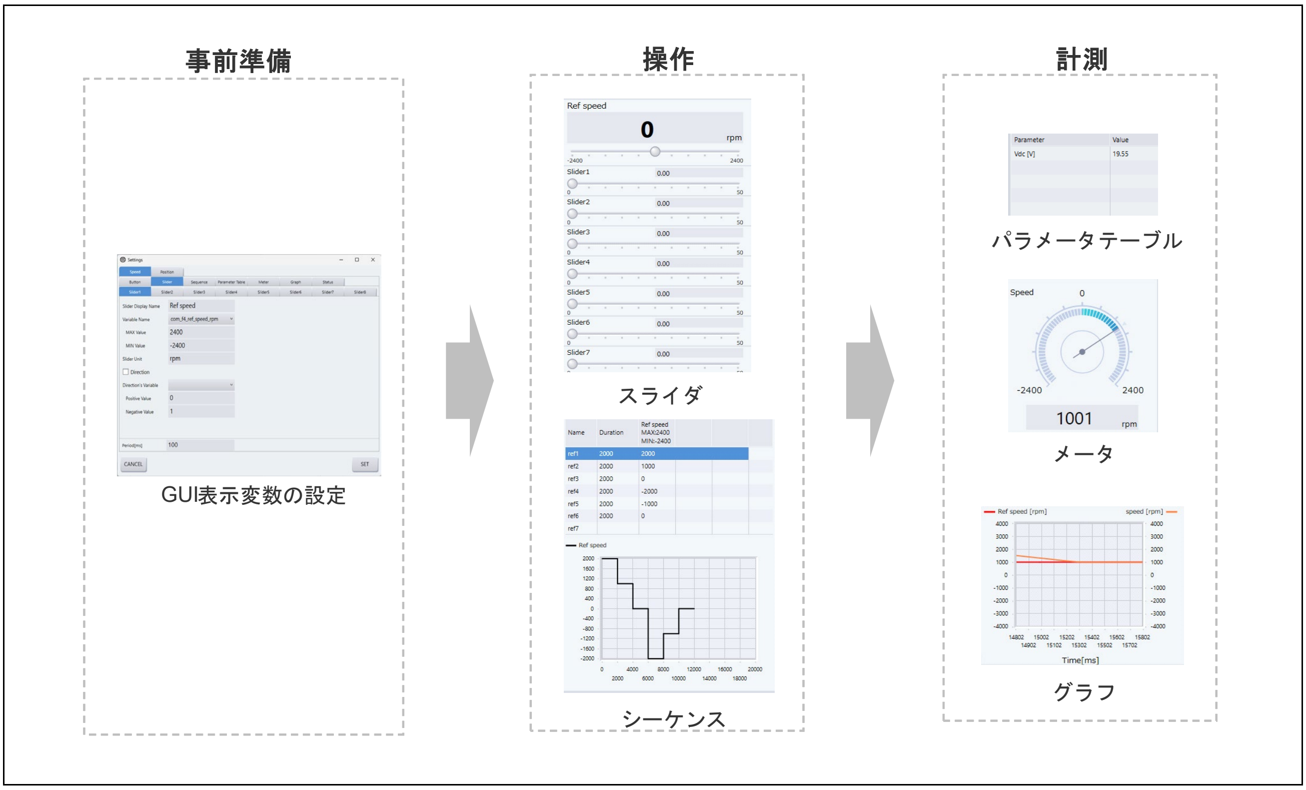


図 12-1 Easy GUI の特徴

12.3 画面説明

Easy GUI 画面の機能を図 12-2 と表 12-1 に示します。

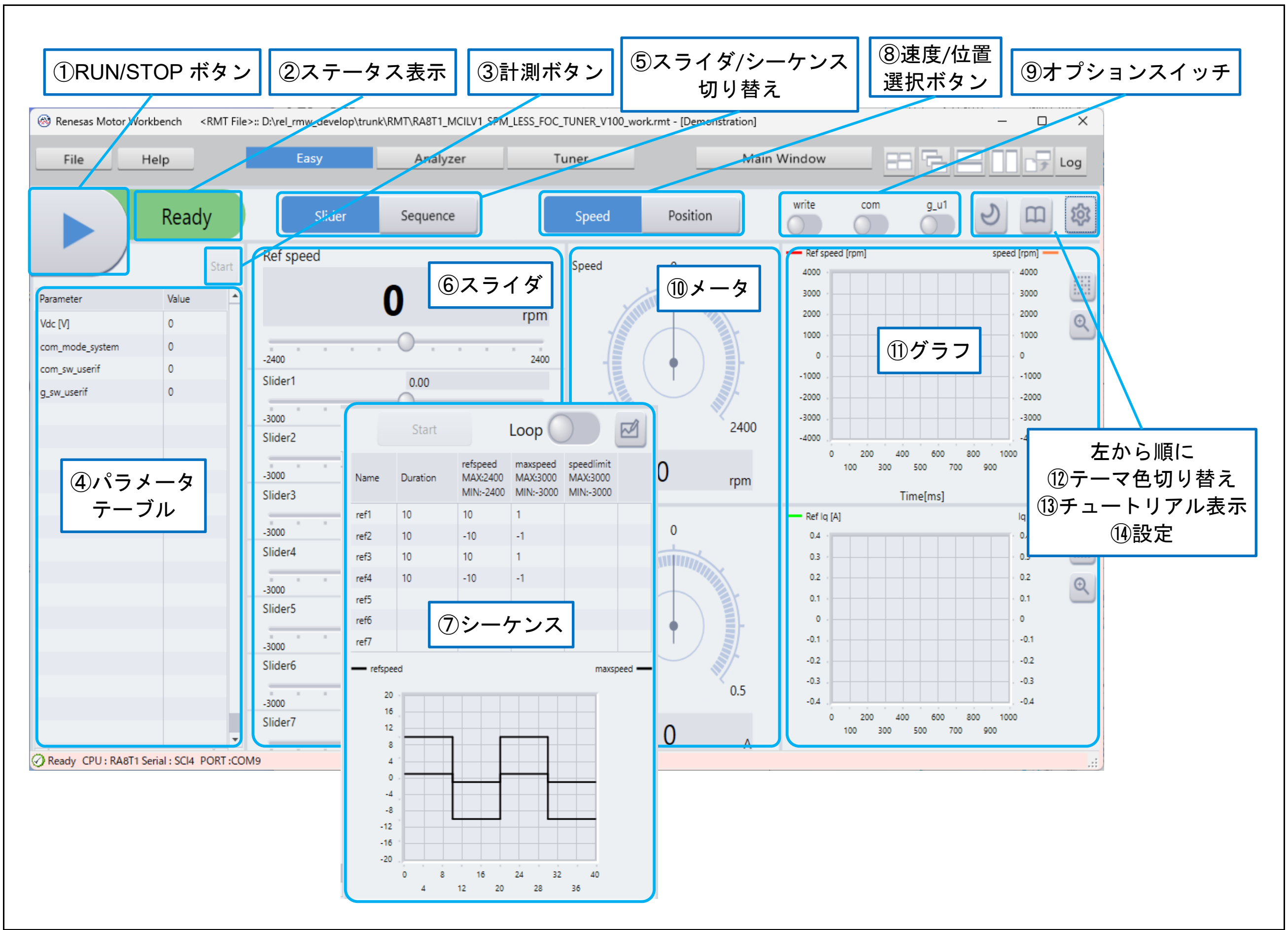


図 12-2 Easy GUI 画面の機能

表 12-1 Easy GUI 画面の機能

No.	名称	説明
1	RUN/STOP ボタン	モータの駆動/停止
2	ステータス表示	駆動状態の表示
3	計測ボタン	計測開始/停止
4	パラメータテーブル	パラメータ値を周期読み込み
5	スライダ/シーケンス切替え	制御パラメータ入力、シーケンス入力の切替え
6	スライダ	指令値入力、制御パラメータ入力
7	シーケンス	指令値シーケンス入力
8	速度/位置切替え	速度制御モード、位置制御モードの切替え
9	オプションスイッチ	オプション機能の ON/OFF スイッチ
10	メータ	駆動状態をメータ表示
11	グラフ	駆動状態をグラフ表示
12	テーマ色切替え	画面テーマ色の切替え
13	チュートリアル表示	チュートリアル画面の起動
14	設定	設定画面の起動

12.4 事前準備

12.4.1 Easy GUI 画面表示

Renesas Motor Workbench を起動すると Main Window が表示されます。Main Window の中央にある「Easy」を押すと、Easy GUI の画面が表示されます。Easy GUI 右上の「Main Window」を押すと、Main Window に戻ります。

図 12-3 Easy GUI 画面の表示

12.4.2 GUI 表示変数の設定

Easy GUI を使用する前に、GUI の表示とプログラムの変数を関連付ける必要があります。GUI 表示変数の設定方法は、12.6 章を参照してください。

12.5 操作説明

本章では各機能の操作方法を説明します。

12.5.1 基本機能

① モータの駆動/停止：RUN/STOPボタンでモータの駆動と停止を切り替えます
はモータが駆動している状態 → ボタンを押すとモータが停止
はモータが停止している状態 → ボタンを押すとモータが駆動

② ステータス表示 ：モータの駆動状態が表示されます

図 12-4 基本機能

12.5.2 スライダ機能

スライダ機能では、モータを駆動しながらスライダの操作で変数の値を動的に変更することができます。
スライダ機能は以下の手順で操作します。

①　スライダ画面表示：スライダ/シーケンス切換えボタンの「Slider」を押します
②　モータを駆動　　：モータが停止している場合は、RUN/STOP ボタンを押してモータを駆動します
③　スライダを操作　：スライダを動かし変数の値を変更します
　　　　　　　　　　　スライダを使用せずに直接値を入力することもできます
④　モータを停止　　：動作を終了する場合は、RUN/STOP ボタンを押してモータを停止します

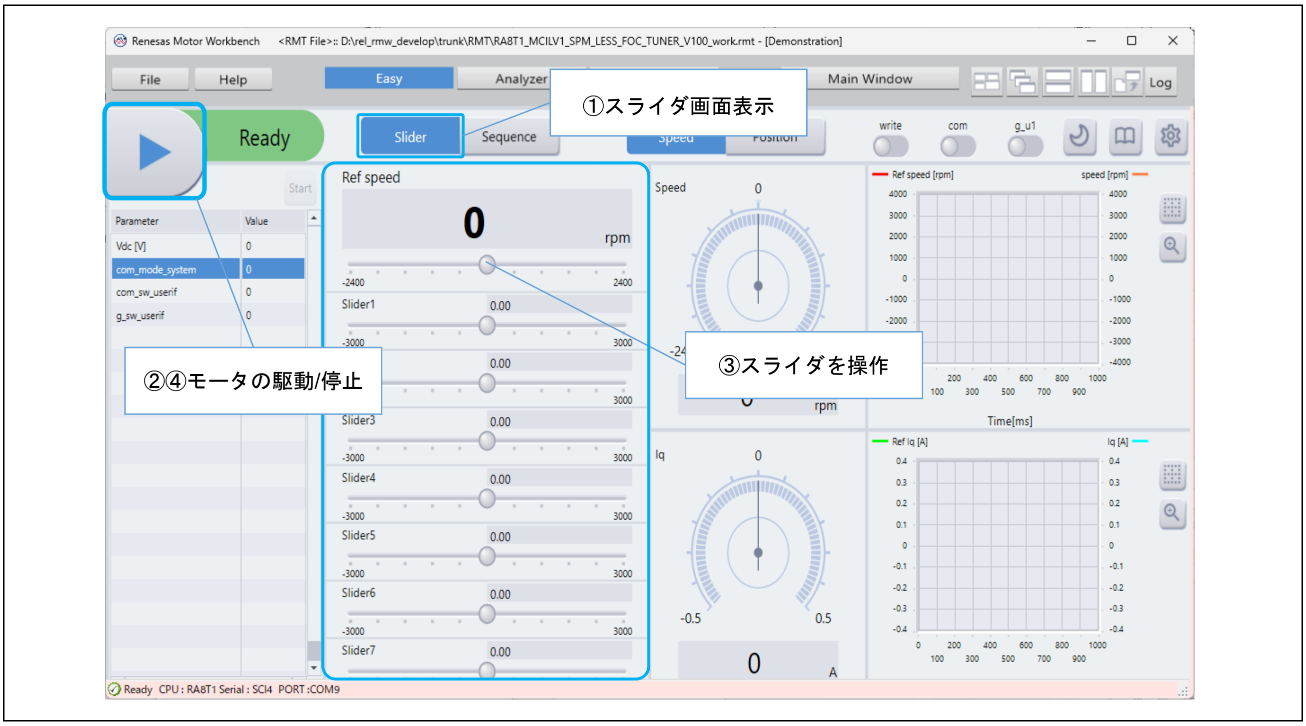


図 12-5　スライダ機能

12.5.3 シーケンス機能

シーケンス機能では、モータを駆動しながらあらかじめシーケンス表に設定した順に変数の値を変更することができます。

12.5.3.1 シーケンスの設定

シーケンスは以下の手順で設定します。

① シーケンス画面表示：スライダ/シーケンス切換えボタンの「Sequence」を押します
② シーケンス表の作成：シーケンス表の ref1 から順番に、動作を確認したい変数値を設定します
　　Duration には、実行時間(単位：ミリ秒)を設定します
③ グラフの確認　：シーケンス表の実行イメージがグラフで表示されます
　　シーケンス表の値を変更すると、グラフが再描画されます
　　グラフの横軸は時間、縦軸は各変数の値です

シーケンス表の作成は、シーケンス表に直接値を入力する他に、シーケンスグラフィカル機能を使用して設定することもできます。シーケンスグラフィカル機能については 12.5.3.2 章を参照してください。

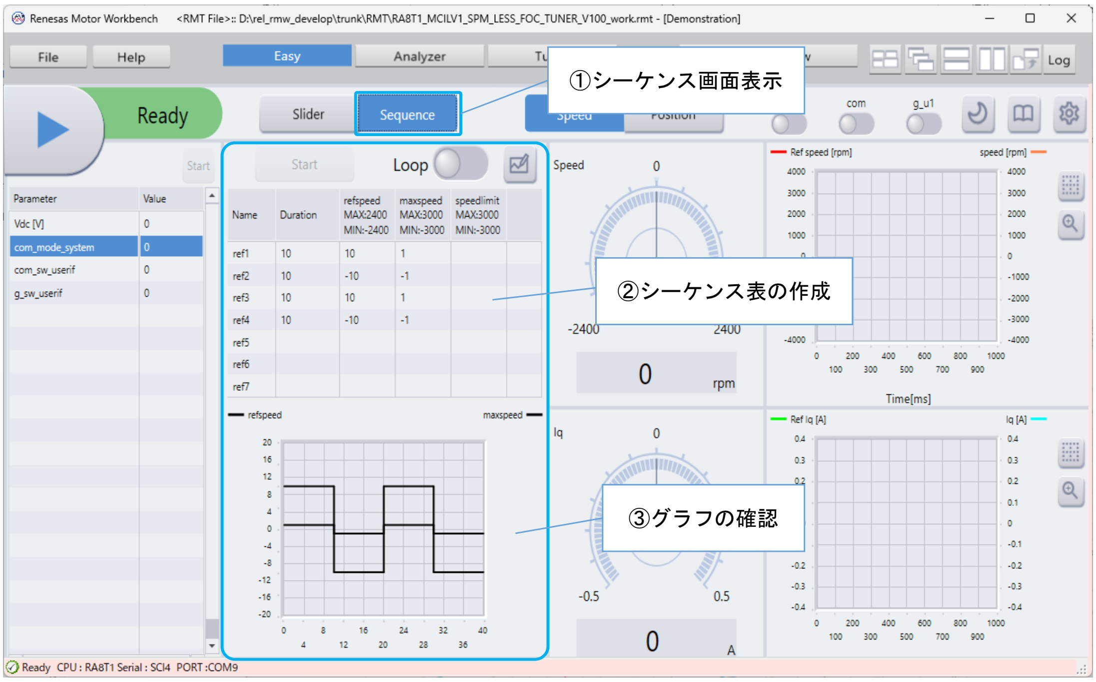


図 12-6 シーケンスの設定

12.5.3.2　シーケンスグラフィカル機能

シーケンス表に直接値を入力してシーケンス表を作成した場合は、本章の設定は必要ありません。

シーケンスグラフィカル機能を使うことで、テンプレートデータのグラフで設定箇所を確認しながらシーケンスの設定を行うことができます。

シーケンスグラフィカル機能は以下の手順で設定します。

① シーケンス画面表示　　　：スライダ/シーケンス切換えボタンの「Sequence」を押します
② グラフィカル入力画面表示：シーケンスグラフィカル入力ボタンを押します
③ シーケンス値の設定　　　：上のグラフで設定箇所を確認しながら、下の各ボックスに値を設定します
　　グラフはテンプレートデータのため、設定した値は反映されません
④ シーケンス表と RMT に反映：「Set」を押して、シーケンスグラフィカル入力画面を閉じます
　　設定した値が、シーケンス表と RMT ファイルに反映されます

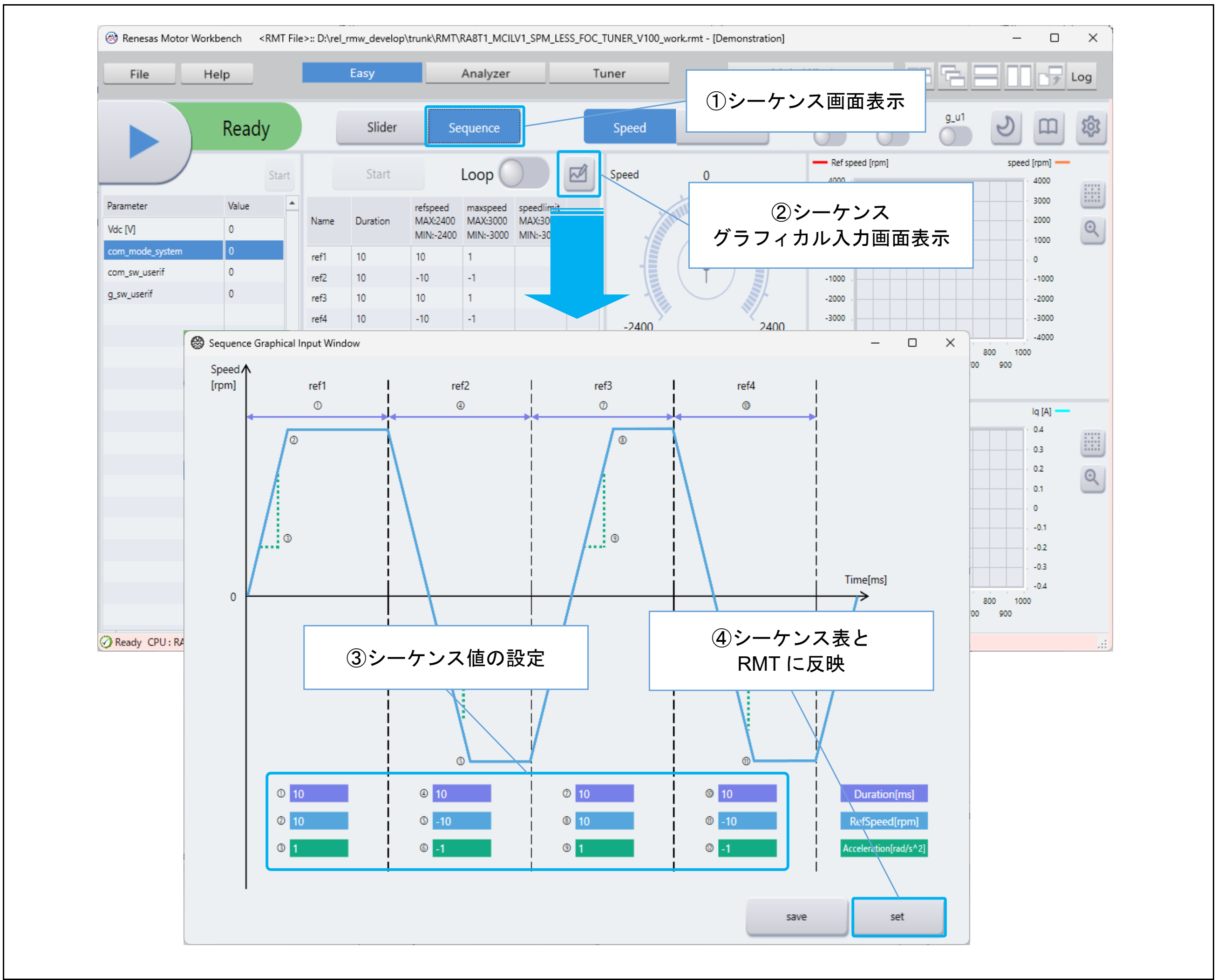


図 12-7　シーケンスグラフィカル機能

12.5.3.3 シーケンスの実行

モータを駆動しながら、シーケンス表にあらかじめ設定した順に変数の値を変更するシーケンス動作を以下の手順で実行します。

① シーケンス画面表示 ：スライダ/シーケンス切換えボタンの「Sequence」を押します
② モータを駆動 ：モータが停止している場合は、RUN/STOP ボタンを押してモータを駆動します
③ 繰り返し動作有効化 ：シーケンス動作を繰り返し実行する場合は LOOP ボタンを有効にします
④ シーケンス動作開始 ：Start/Stop ボタンを押します
⑤ シーケンス動作停止 ：動作を終了する場合は、Start/Stop ボタンを押します
⑥ モータを停止 ：RUN/STOP ボタンを押してモータを停止します

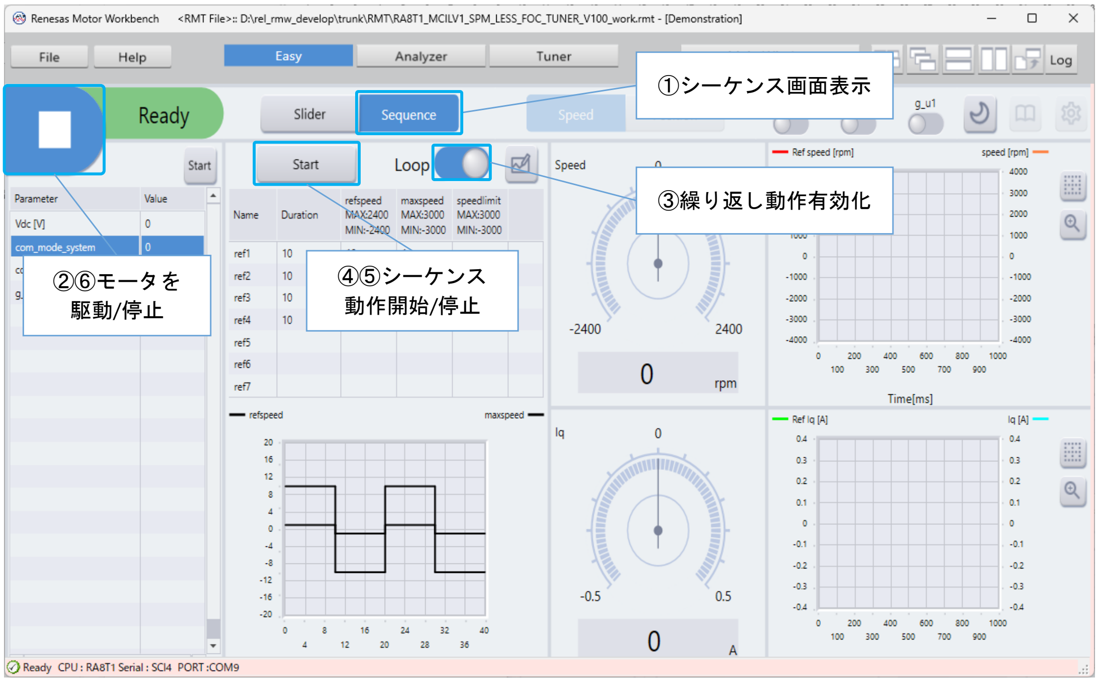


図 12-8 シーケンスの実行

12.5.4 計測機能

計測機能では、以下の手順で現在のモータの状態をメータおよびグラフで確認することができます。

① モータを駆動 : モータが停止している場合は、RUN/STOP ボタンを押してモータを駆動します
② 計測開始 : 計測ボタンを押します
③ 計測結果を確認 : モータの状態がメータとグラフに反映されます
④ グラフを拡大 : グラフを拡大すると、グラフのコピーや保存をすることができます
⑤ 計測停止 : 計測を終了する場合は、計測ボタンを押します
⑥ モータを停止 : RUN/STOP ボタンを押してモータを停止します

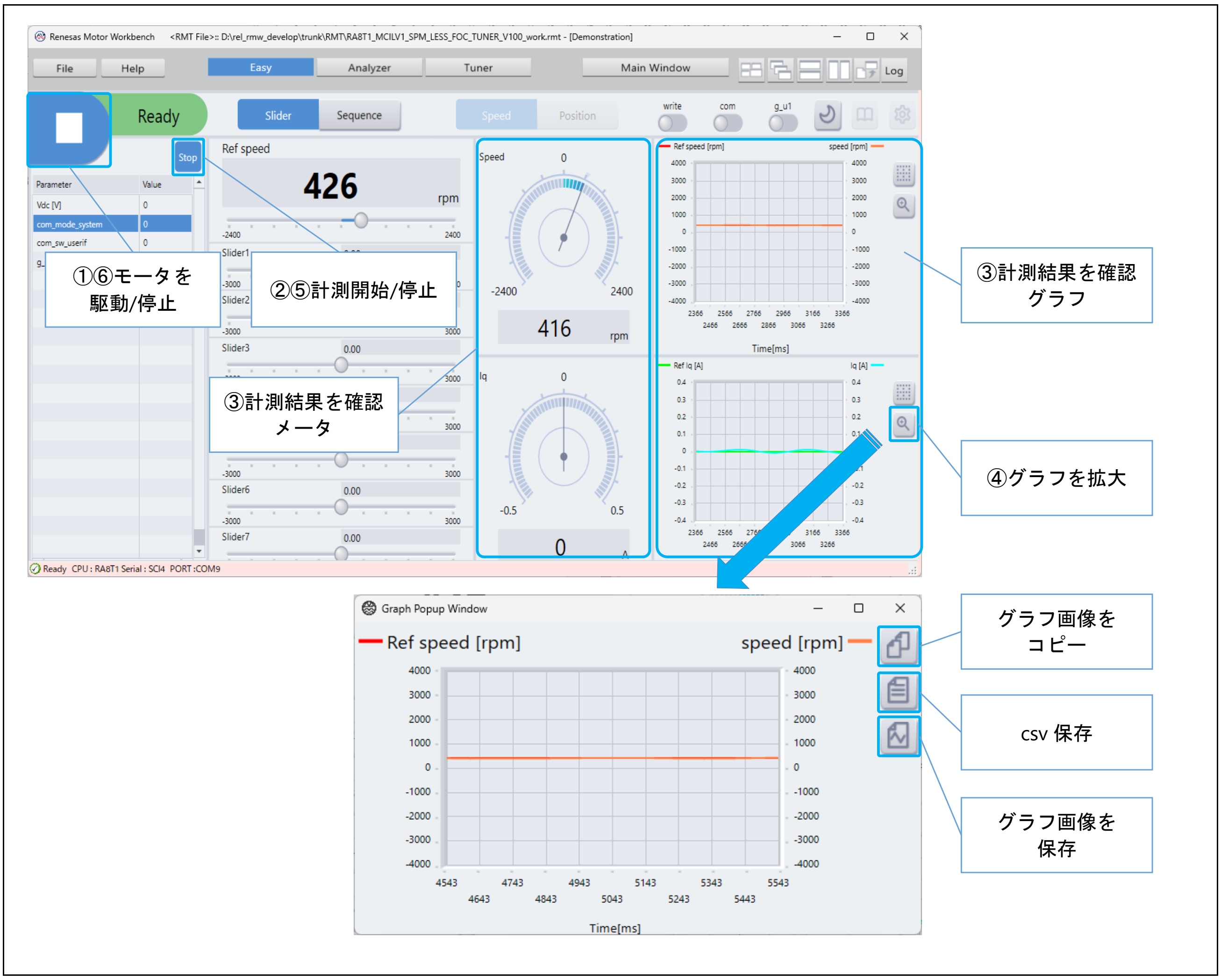


図 12-9 計測機能

12.6 変数の設定

Easy GUI で操作および計測する変数は、設定ボタンで設定することができます。

設定ボタンを押すと、図 12-10 のとおり設定画面が表示されます。設定画面のタブを選択し、各 GUI に関連付ける変数の設定を行います。

各設定画面で変数名を設定する場合は、変数名の欄をクリックすると表示される「Variable Find」の中から該当する変数名をダブルクリックします。「Variable Find」の Find 欄にキーワードを入力し検索することも可能です。変数名の設定例を図 12-11 に示します。

本章では、各 GUI に対する設定例を説明します。

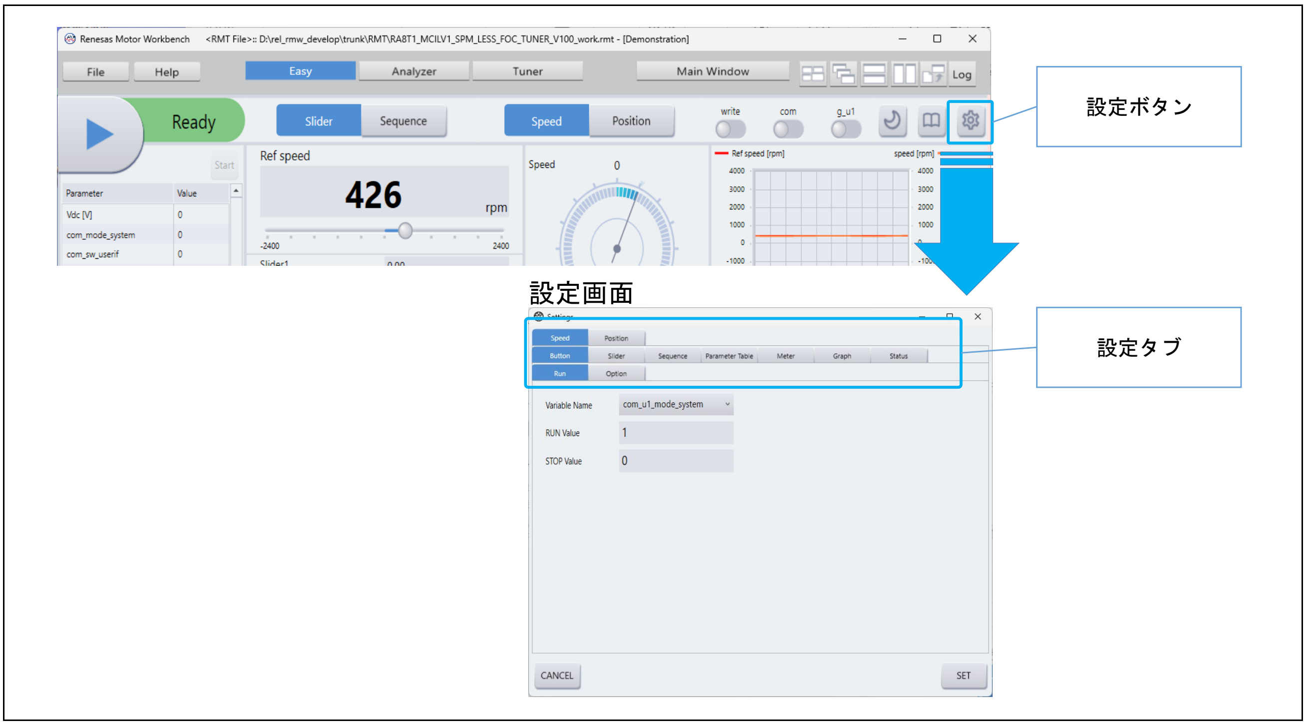


図 12-10 設定画面の表示

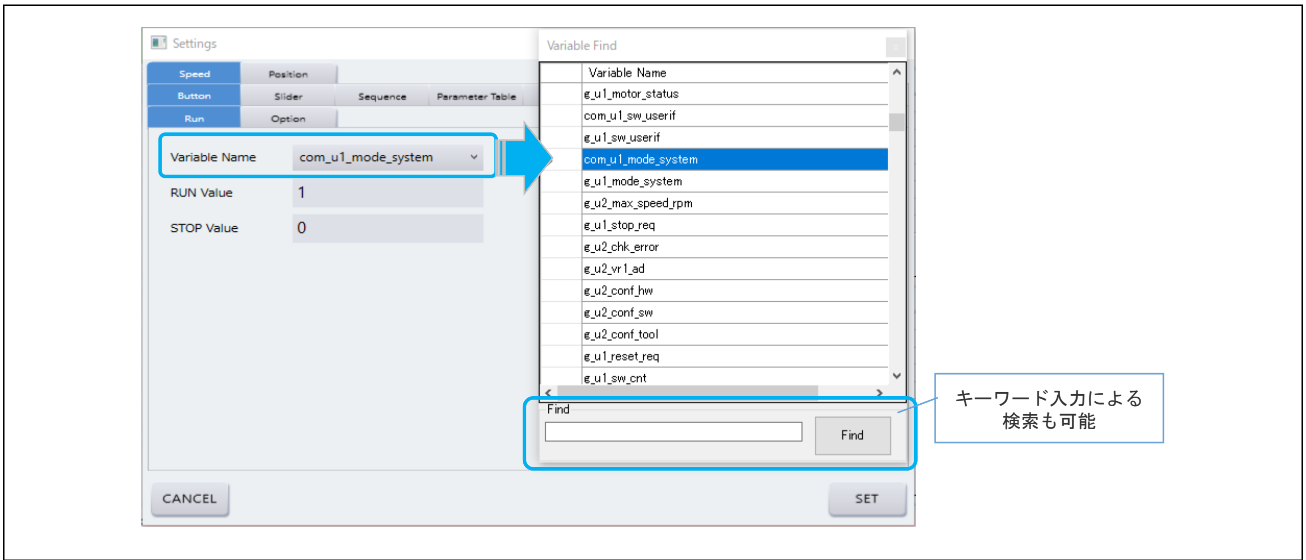


図 12-11 変数名の設定例

12.6.1 RUN/STOP ボタン

RUN/STOP ボタンを押す毎に変数に値を書き込みます。
RUN/STOP ボタンに関連付ける変数の設定例は以下のとおりです。

表 12-2 RUN/STOP ボタン設定例

項目		内容
設定タブ	Speed 選択時	Speed → Button → Run
	Position 選択時	Position → Button → Run
RUN/STOP ボタン		• 変数 com_u1_mode_system=1（RUN Value）の場合、を表示 • 変数 com_u1_mode_system=0（STOP Value）の場合、を表示

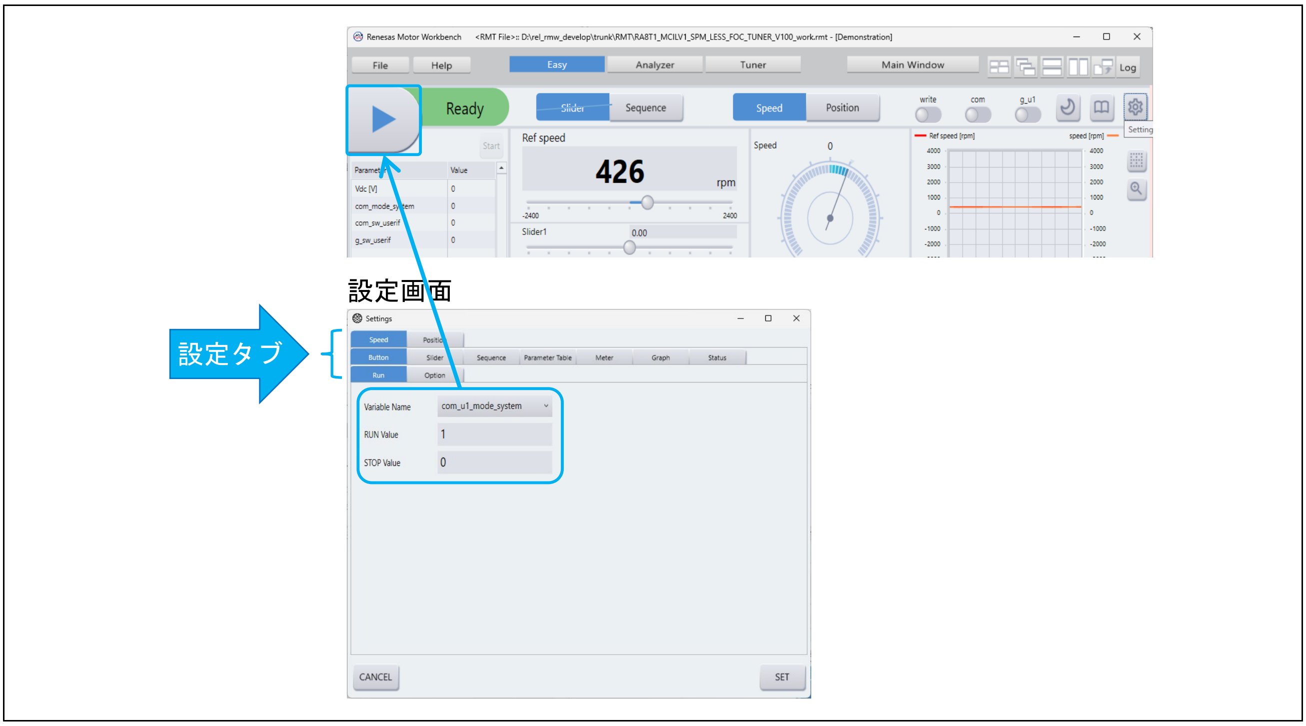


図 12-12 RUN/STOP ボタン設定例

12.6.2　ステータス表示

ステータスに設定した条件に一致した場合、設定した色とステータス名を表示します。
ステータス表示に関連付ける変数の設定例は以下のとおりです。

表 12-3　ステータス表示設定例

項目		内容
設定タブ	Speed 選択時	Speed　→　Status
	Position 選択時	Position　→　Status
ステータス表示		• 変数 g_u1_motor_status=0 の場合、Ready を表示 • 変数 g_u1_motor_status=1 の場合、Run を表示 • 変数 g_u1_motor_status=2 の場合、Error を表示 • 1000ms 周期でステータス表示を更新

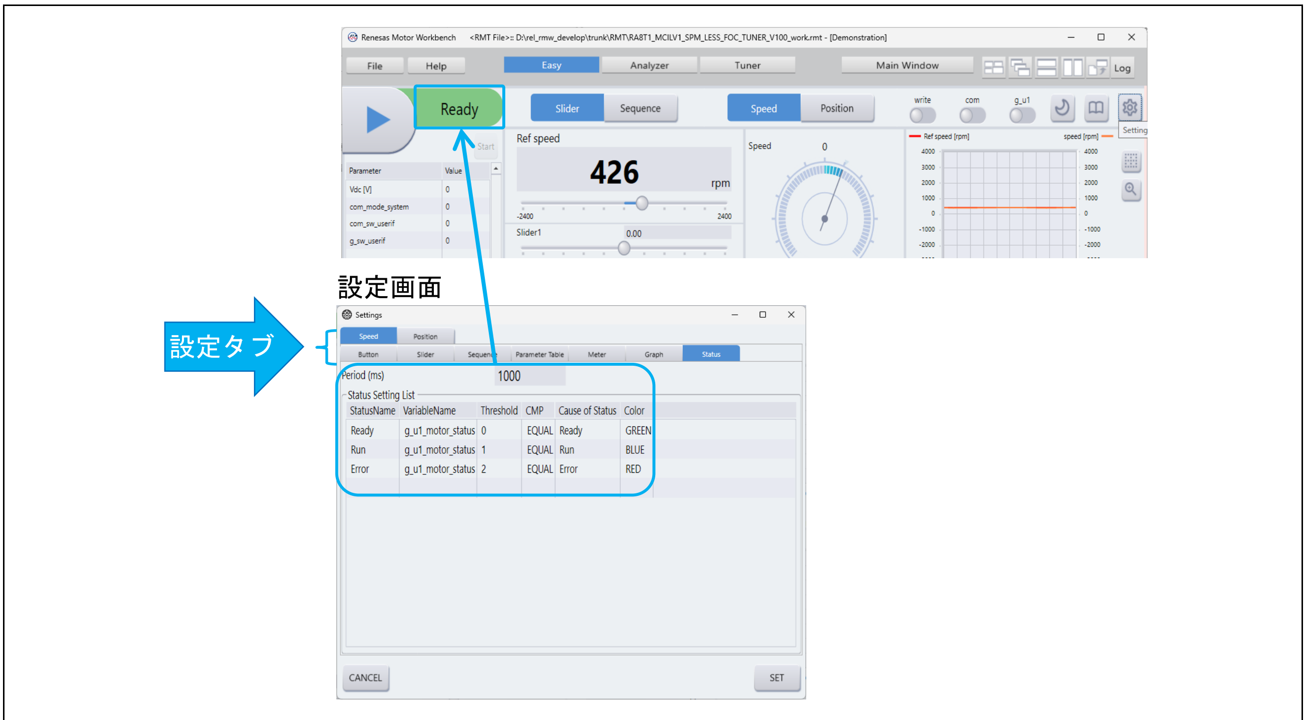


図 12-13　ステータス表示設定例

12.6.3 パラメータテーブル

パラメータテーブルに設定した変数値を周期的に読み込んで表示します。パラメータテーブルには最大 20 個のパラメータを設定することができます。

パラメータテーブルに関連付ける変数の設定例は以下のとおりです。

表 12-4 パラメータテーブル設定例

項目		内容
設定タブ	Speed 選択時	Speed → Parameter Table
	Position 選択時	Position → Parameter Table
パラメータテーブル		• 「Vdc [V]」の項目に、変数 g_f4_vdc_ad_monitor の値を表示 • 1000ms 周期でパラメータテーブルを更新

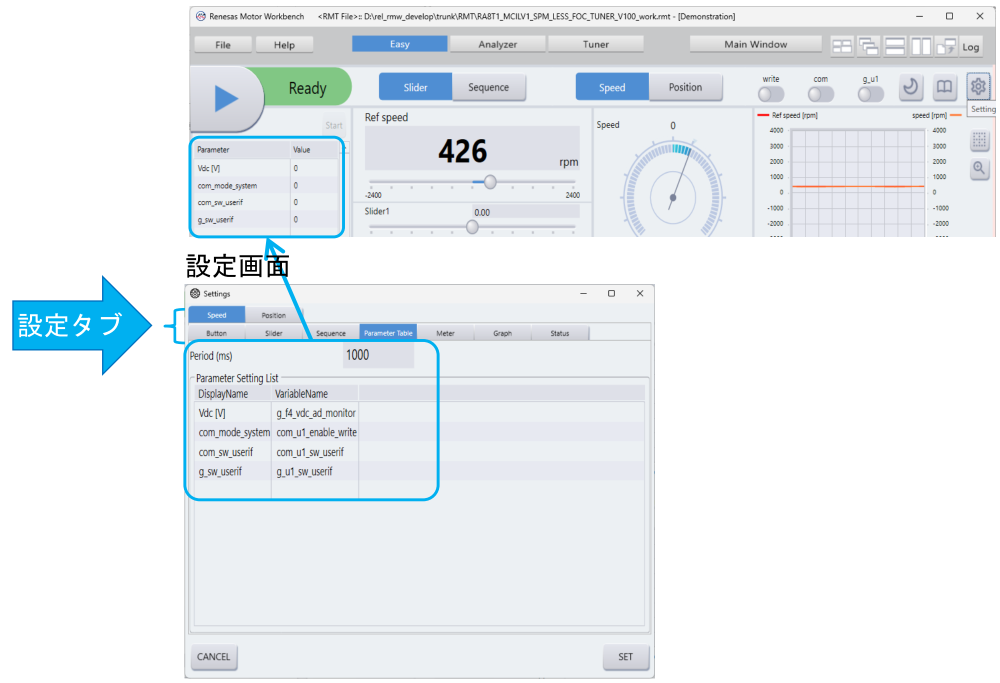


図 12-14 パラメータテーブル設定例

12.6.4 スライダ

スライダ操作した値を変数に書き込みます。最大 8 個のスライダを設定することができます。回転方向の変数を持っている場合は、スライダの正負によって値を書き換える変数を設定することができます。

スライダに関連付ける変数の設定例は以下のとおりです。

表 12-5 スライダ設定例

項目		内容
設定タブ	Speed 選択時	Speed → Slider → Slider1～8
	Position 選択時	Position → Slider → Slider1～8
スライダ		• Slider1 に「Ref speed」を表示 • 「Ref speed」の設定範囲を、-2400rpm～2400rpm に設定 • 「Ref speed」に設定した値を、変数 com_f4_ref_speed_rpm に設定 • 1000ms 周期でスライダの値を変数に反映

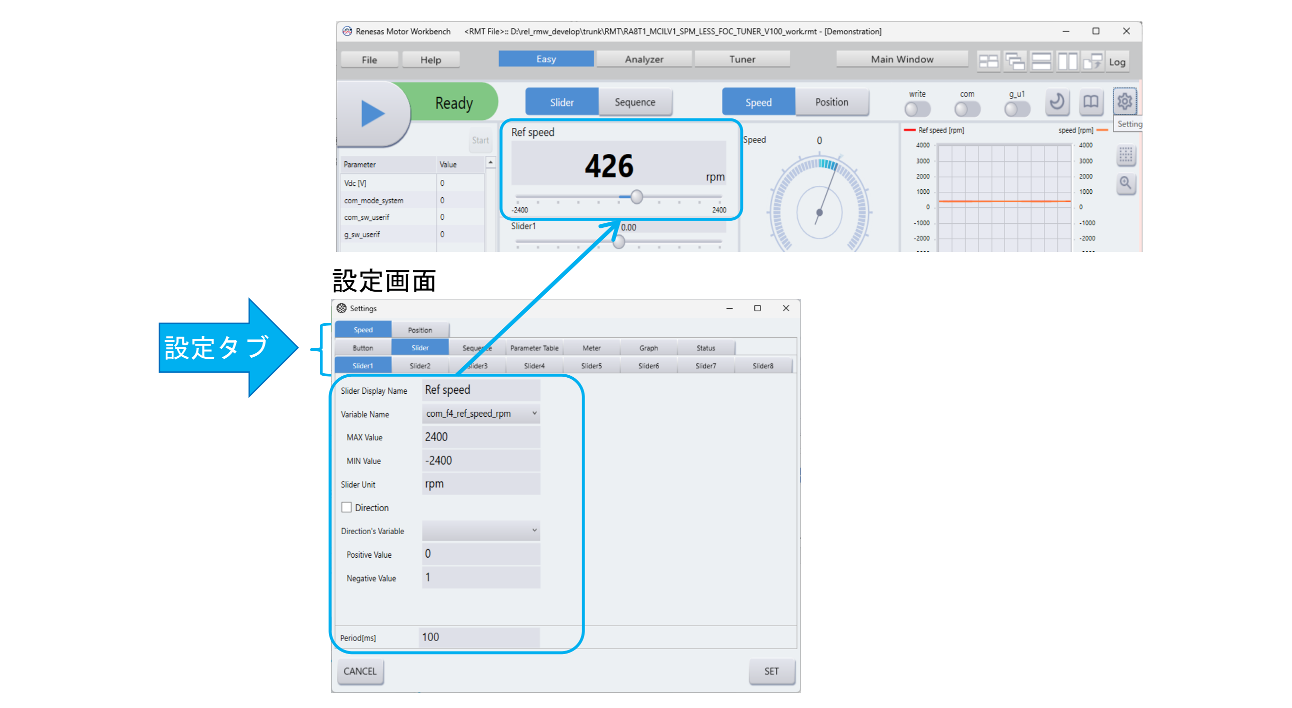


図 12-15 スライダ設定例

12.6.5 シーケンス

シーケンス表に設定した値を変数に書き込みます。シーケンス表には最大 3 個の変数を設定することができます。

シーケンスに関連付ける変数の設定例は以下のとおりです。

表 12-6 シーケンス設定例

項目		内容
設定タブ	Speed 選択時	Speed → Sequence → Table → Variable1～3
	Position 選択時	Position → Sequence → Table → Variable1～3
シーケンス		• Variable1 に「Ref speed」を表示 • 「Ref speed」の設定範囲を、-2400～2400 に設定 • 「Ref speed」に設定した値を、変数 com_f4_ref_speed_rpm に設定 • 「Ref speed」の設定値の変化をグラフに の色で表示

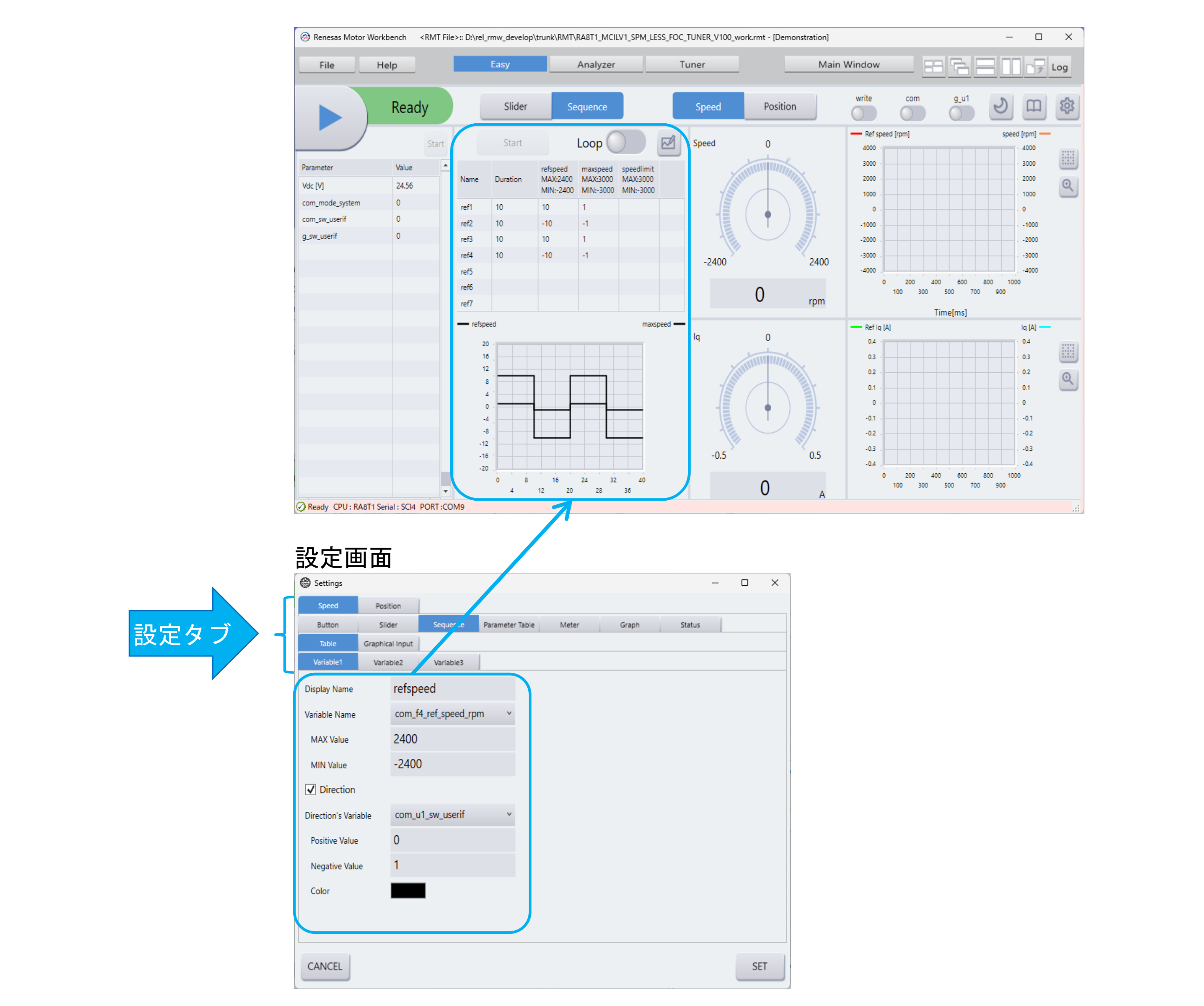


図 12-16 シーケンス設定例

12.6.6 シーケンスグラフィカル入力

シーケンスグラフィカル入力で設定した値を変数に書き込みます。
シーケンスグラフィカル入力に関連付ける変数の設定例は以下のとおりです。

表 12-7 シーケンスグラフィカル入力設定例

項目		内容
設定タブ	Speed 選択時	Speed → Sequence → Graphical Input → Variable1～3
	Position 選択時	Position → Sequence → Graphical Input → Variable1～3
シーケンスグラフィカル入力		• Variable1 に「Ref speed」を表示 • 「Ref speed」の設定範囲を、-2400～2400 に設定 • 「Ref speed」に設定した値を、変数 com_f4_ref_speed_rpm に設定

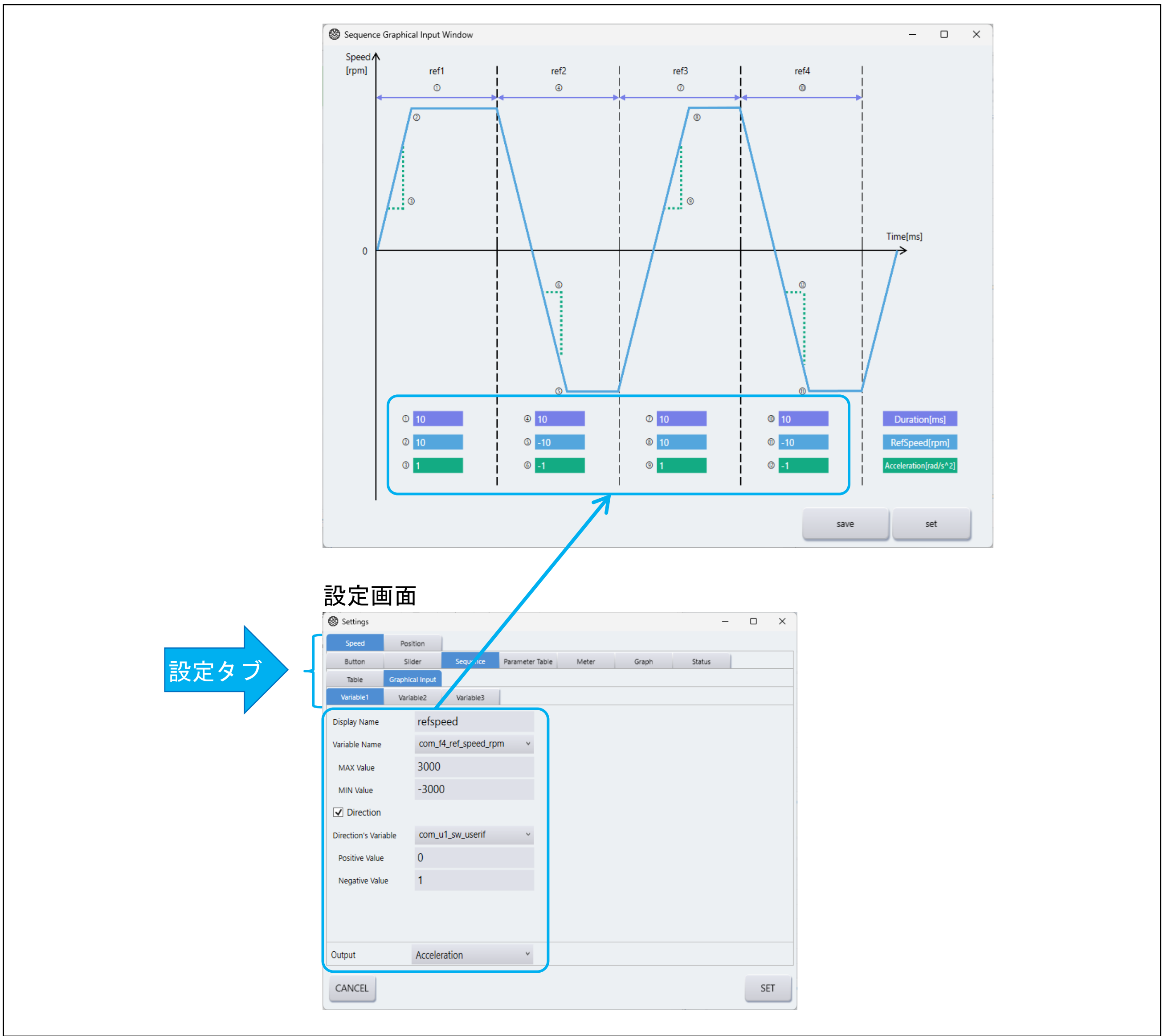


図 12-17 シーケンスグラフィカル入力設定例

12.6.7 オプションスイッチ

オプションスイッチを ON/OFF した場合、変数に任意の値を書き込みます。最大 3 個のオプションスイッチを設定することができます。

オプションスイッチに関連付ける変数の設定例は以下のとおりです。

表 12-8 オプションスイッチ設定例

項目		内容
設定タブ	Speed 選択時	Speed → Button → Option
	Position 選択時	Position → Button → Option
オプションスイッチ		• Option2 に「RMW UI」を表示 • 変数 com_u1_sw_userif=0（ON Value）の場合、を表示 • 変数 com_u1_sw_userif=1（OFF Value）の場合、を表示

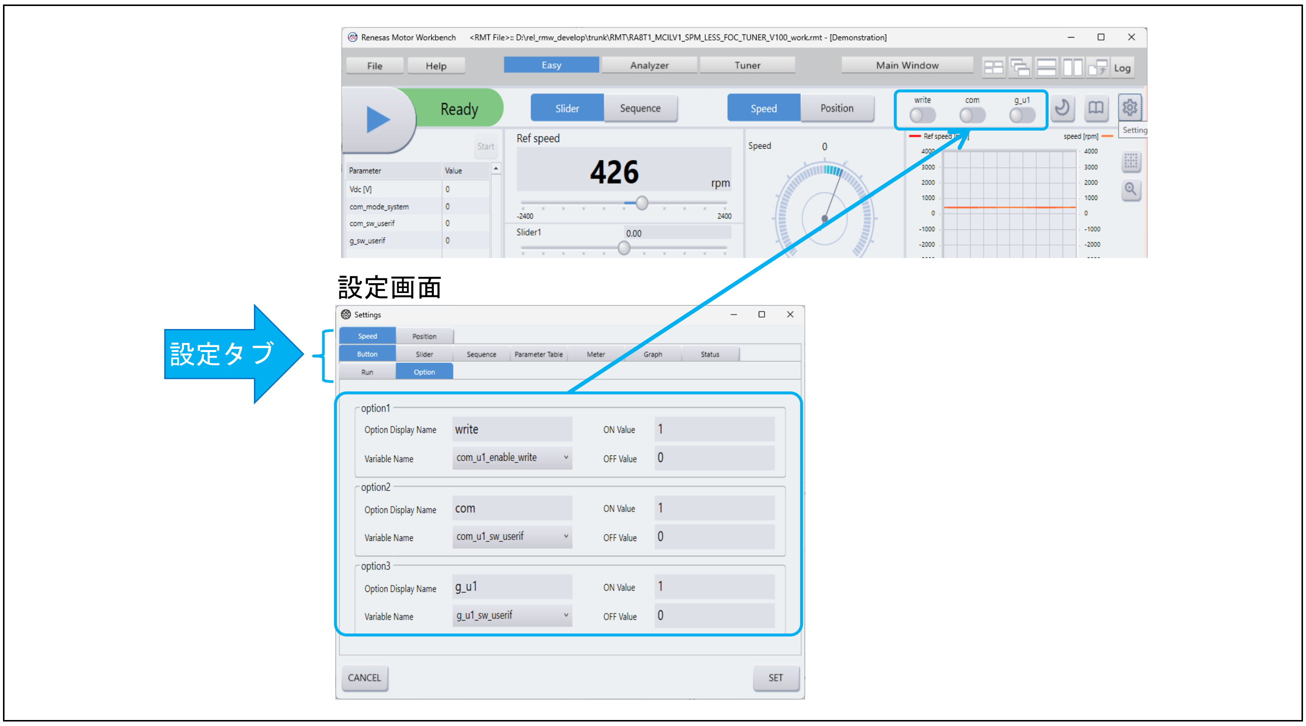


図 12-18 オプションスイッチ設定例

12.6.8 メータ

変数の値をメータに表示します。最大 2 個のメータを表示することができます。

指令値変数を設定することでメータ上に▲マークで指令値の位置を表示することができます。メータタイプが Speed の場合は rad/s から rpm へ単位変換する機能があります。

メータに関連付ける変数の設定例は以下のとおりです。

表 12-9 メータ設定例

項目		内容
設定タブ	Speed 選択時	Speed → Meter → Top/Bottom
	Position 選択時	Position → Meter → Top/Bottom
メータ		上側のメータの設定をする場合は「Top」タブを選択 • メータを「Speed」と表示 • メータタイプ「Speed」を選択 • 変数 g_f4_speed_rpm_monitor の値を「Speed」に設定 • rad/s から rpm への変換を有効にし、極対数に変数 com_u2_mtr_pp を設定 • 指令値変数の表示を有効にし、指令値変数に com_f4_ref_speed_rpm を設定 • メータの表示範囲を-2400～2400 に設定 • 変化前を（水色）の色で表示し、変化後を（青色）の色で表示 • 100ms 周期で変数の値をメータに表示

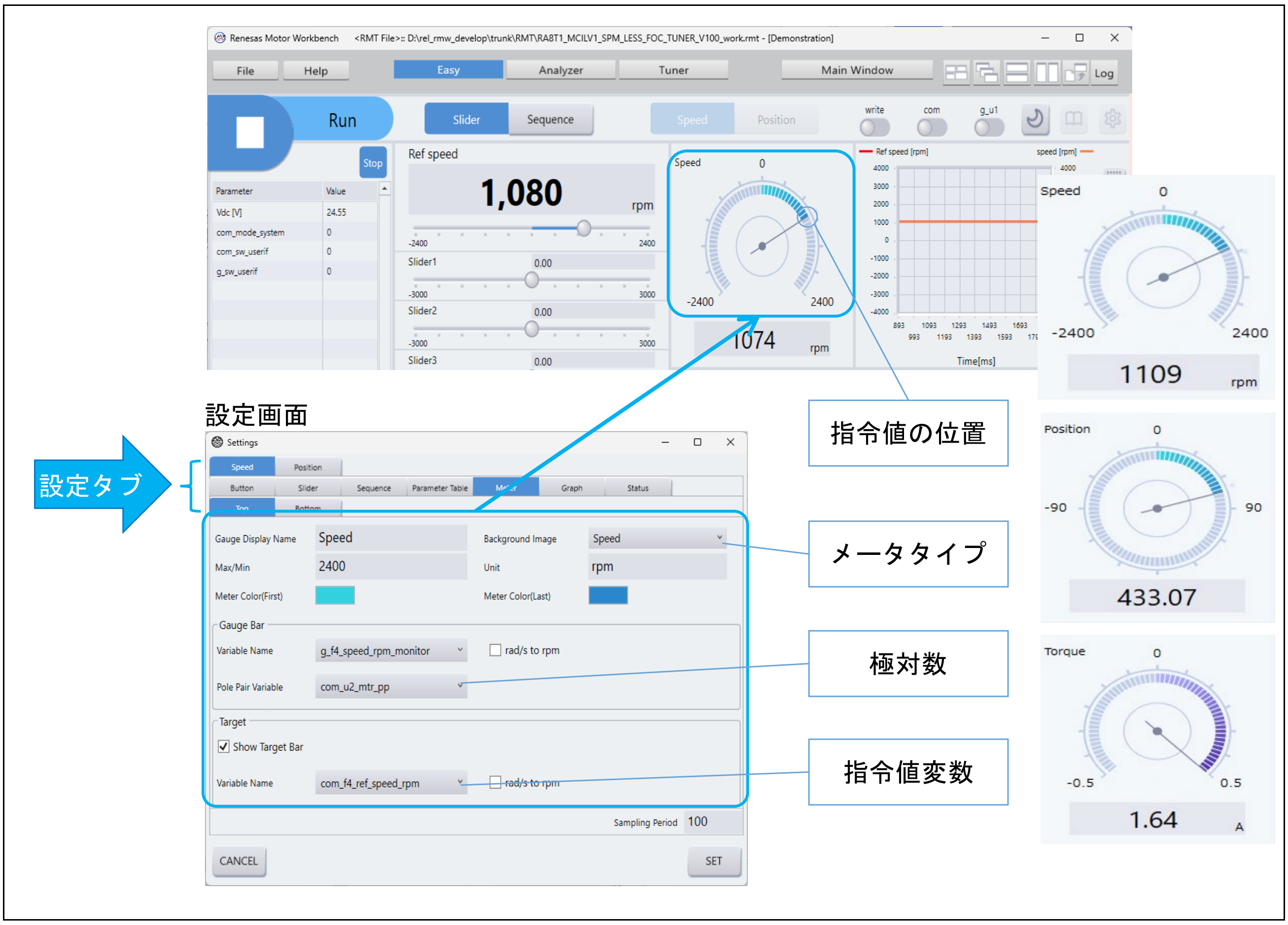


図 12-19 メータ設定例

12.6.9 グラフ

変数の値をグラフに表示します。最大 2 個のグラフを表示することができ、1 個のグラフに最大 2 個の変数の値を表示することができます。

グラフに関連付ける変数の設定例は以下のとおりです。

表 12-10 グラフ設定例

項目		内容
設定タブ	Speed 選択時	Speed → Graph → Top/Bottom
	Position 選択時	Position → Graph → Top/Bottom
グラフ		上側のグラフの設定をする場合は「Top」タブを選択 • Parameter1 を「Ref speed [rpm]」とし、■の色で表示 • Parameter2 を「speed [rpm]」とし、■の色で表示 • 変数 com_f4_ref_speed_rpm の値を「Ref speed [rpm]」に設定 • 変数 g_f4_speed_rpm_monitor の値を「speed [rpm]」に設定 • 変数の型を単精度浮動小数点型(FLOAT)に設定 • X 軸の目盛間隔を 100ms に、Y 軸範囲を-1000～1000 に設定 • 100ms 周期で変数の値をグラフに表示

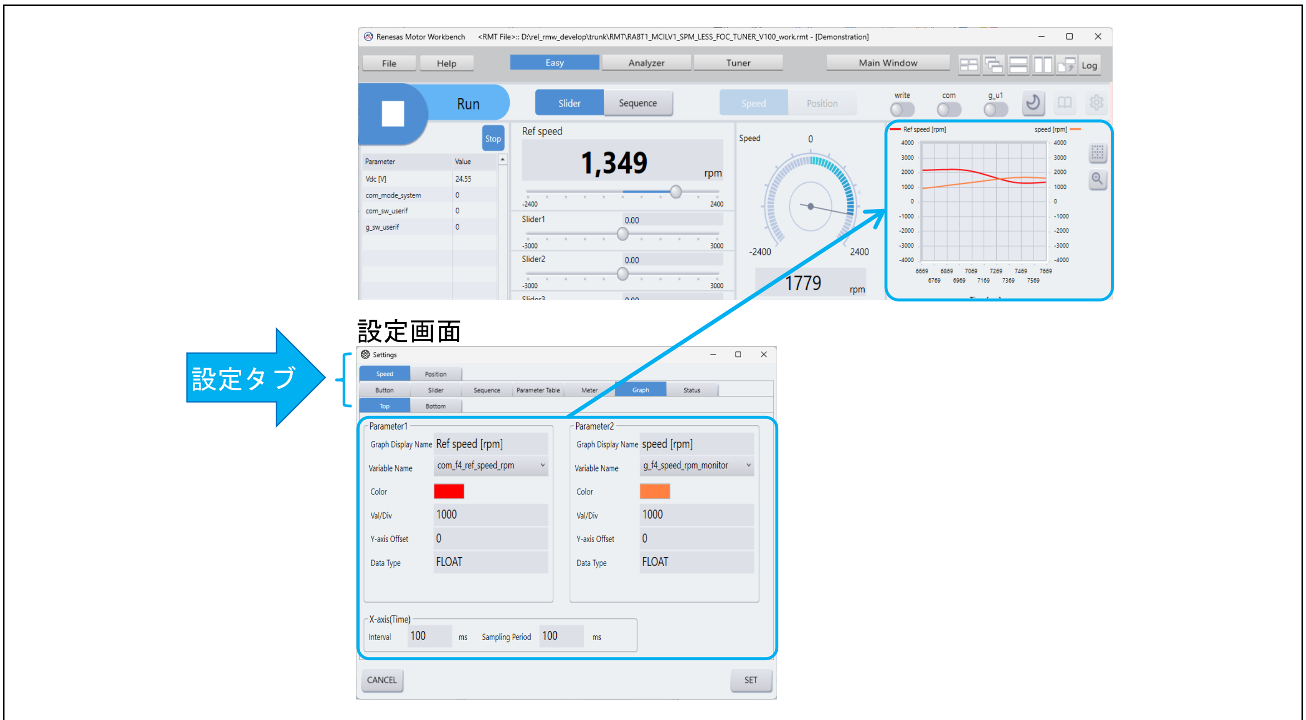


図 12-20 グラフ設定例

12.7 テーマ色切替え

GUI 画面右上のを押すと、画面のテーマ色を切替えることができます。

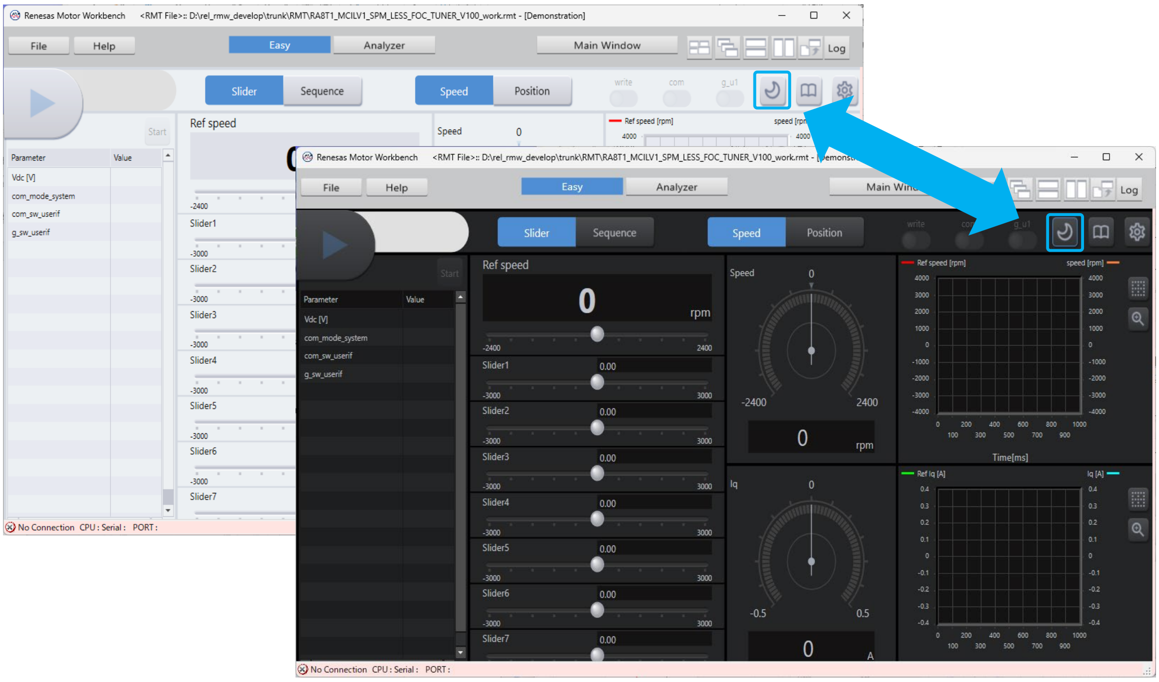


図 12-21 テーマ色切り替え

12.8 チュートリアル

GUI 画面右上の を押すと、チュートリアルを表示することができます。
GUI の説明箇所をハイライトにしながら、説明文が順次表示されます。チュートリアルを終了する場合は「END」を押してください。

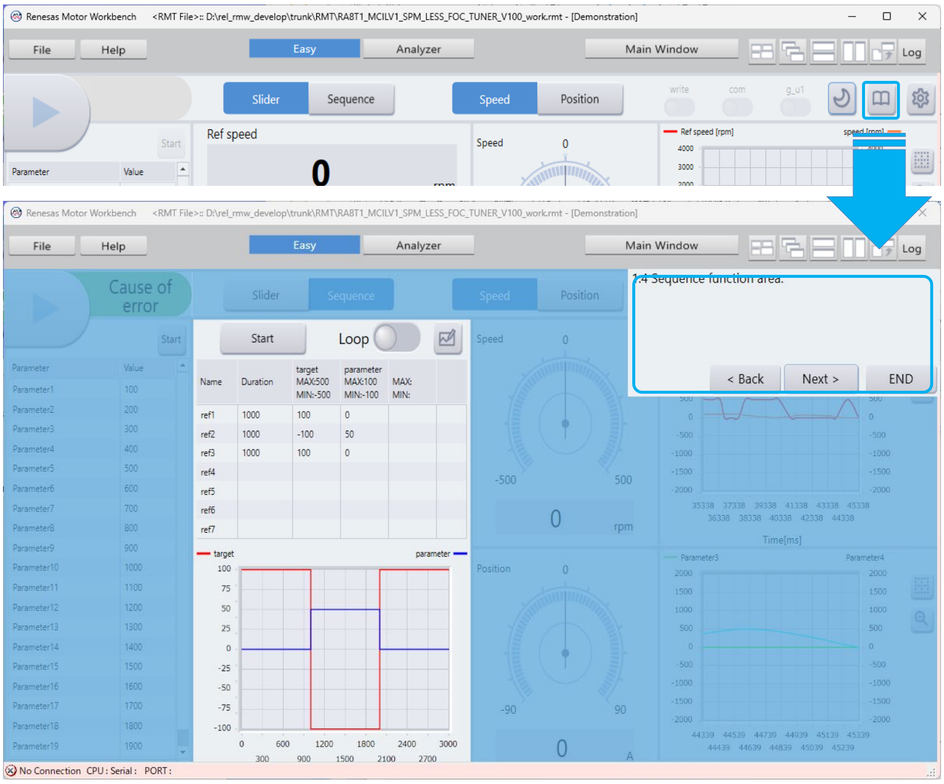


図 12-22 チュートリアル表示例

13. Servo ツール

13.1 概要

13.1.1 サーボ制御とは

モータの位置決め運転を正確に行うためのフィードバック制御の一つです。モータに内蔵する位置センサにより、モータの位置、回転速度をモニタし、制御します。目標値(位置や回転数)とセンサの測定値を比較し指令値を決めるため、目標値と一致するようにモータを正確に制御することができます。

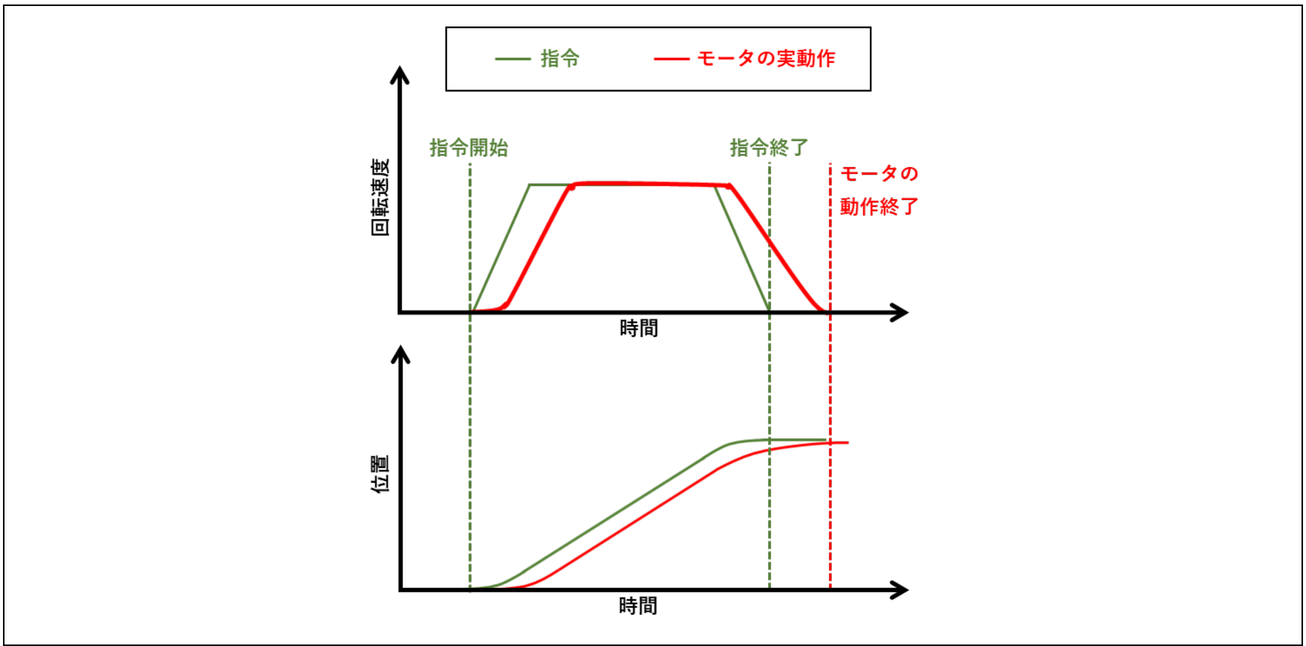


図 13-1　指令と実際の動作

モータを指令通りに運転するために、位置制御（位置偏差に対する制御）と速度制御（速度偏差に対する制御）を組み合わせて制御を行います。指令値と実際の値の差を偏差と呼びます。以下に簡易的なブロック線図を示します。

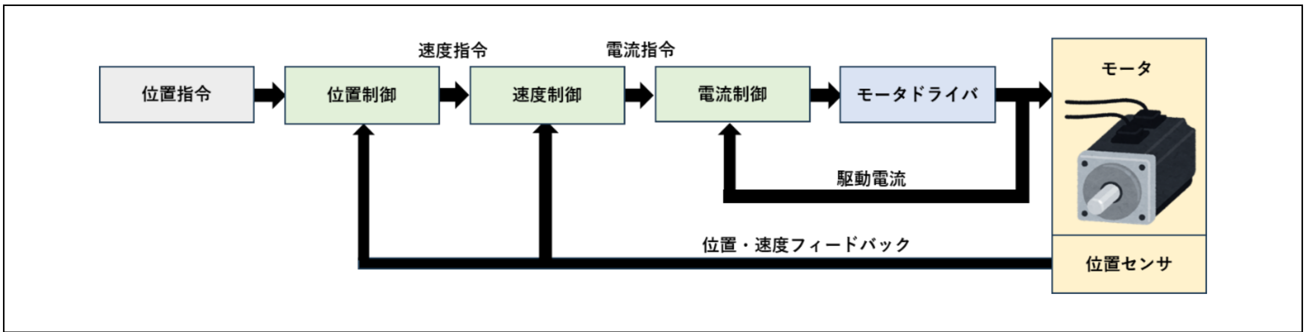


図 13-2　サーボ制御のブロック線図

位置制御では、ユーザが指定した位置指令値と位置センサの測定値の偏差を求め、比例制御によって速度指令値を算出します。算出時に、「位置制御周波数」が影響します。

速度制御では、位置制御から求めた速度指令値と位置センサの測定値から計算した速度情報から偏差を求め、PI 制御によって電流指令値を算出します。算出時に、「速度制御周波数」と「イナーシャ」が影響します。

電流制御では、速度制御から求めた電流指令値より PWM として出力する電圧を算出し、モータドライバを駆動します。

使用する環境（装置・機構）に合わせて、位置制御の応答性、速度制御の応答性を適切に調整することが重要となります。応答性が遅いと、指令値に到達するまでの時間が長くなります。応答性を過度に速くすると、速度が変動しやすくなり、動作が不安定になります。

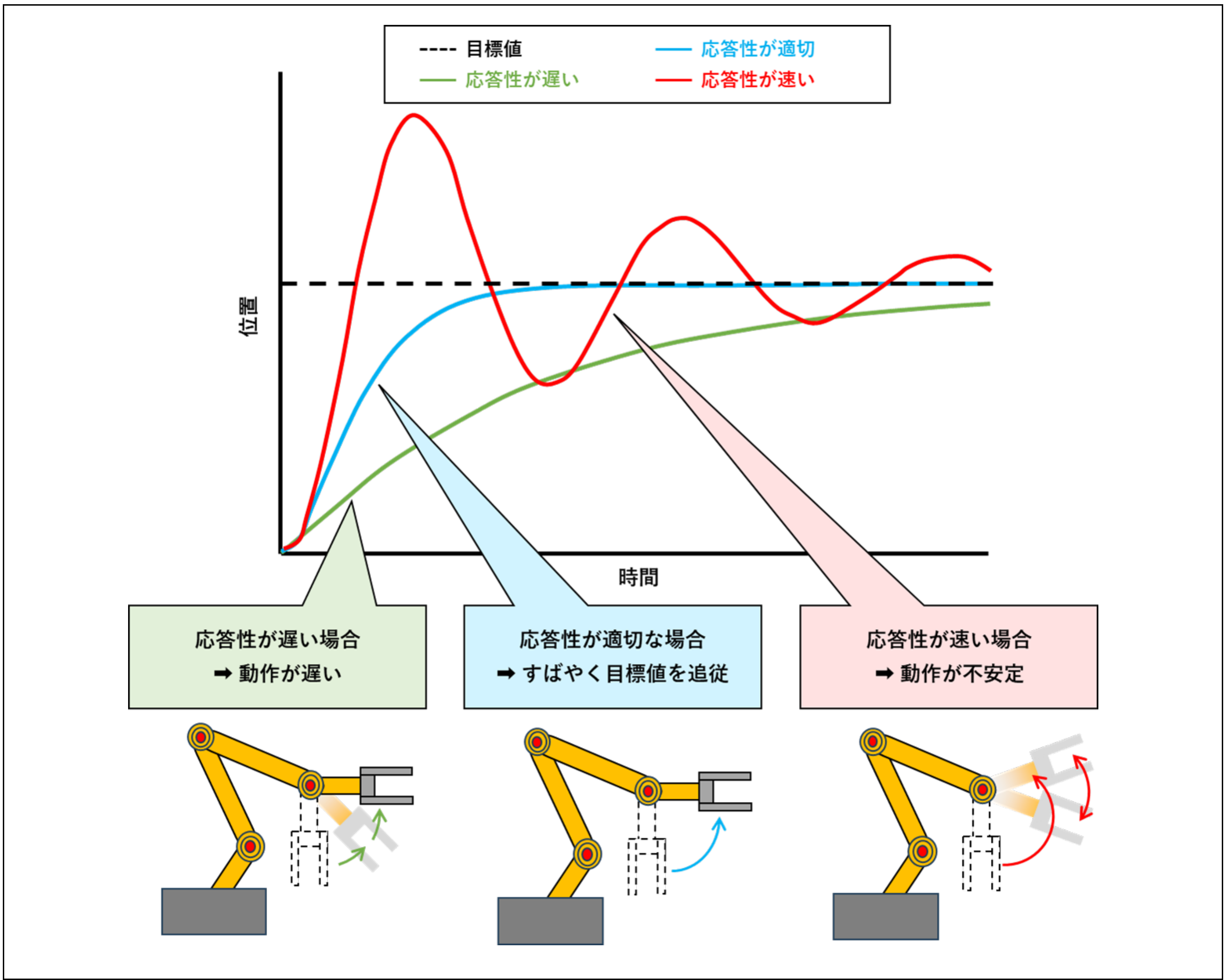


図 13-3 サーボ制御の応答性

13.1.2　　Servo ツールとは

Servo ツールは、サーボ制御を行う際のパラメータ調整をサポートするツールです。制御プログラムでServo 機能がサポートされている場合に限り利用可能となります。その場合、Main Window の「Select Tool」に「Servo」が選択可能な状態で表示されます。

モータ単体でのデバッグが完了した後、機構に取り付けた後に使用します。Servo ツールを使用する流れを図 13-4 に示します。

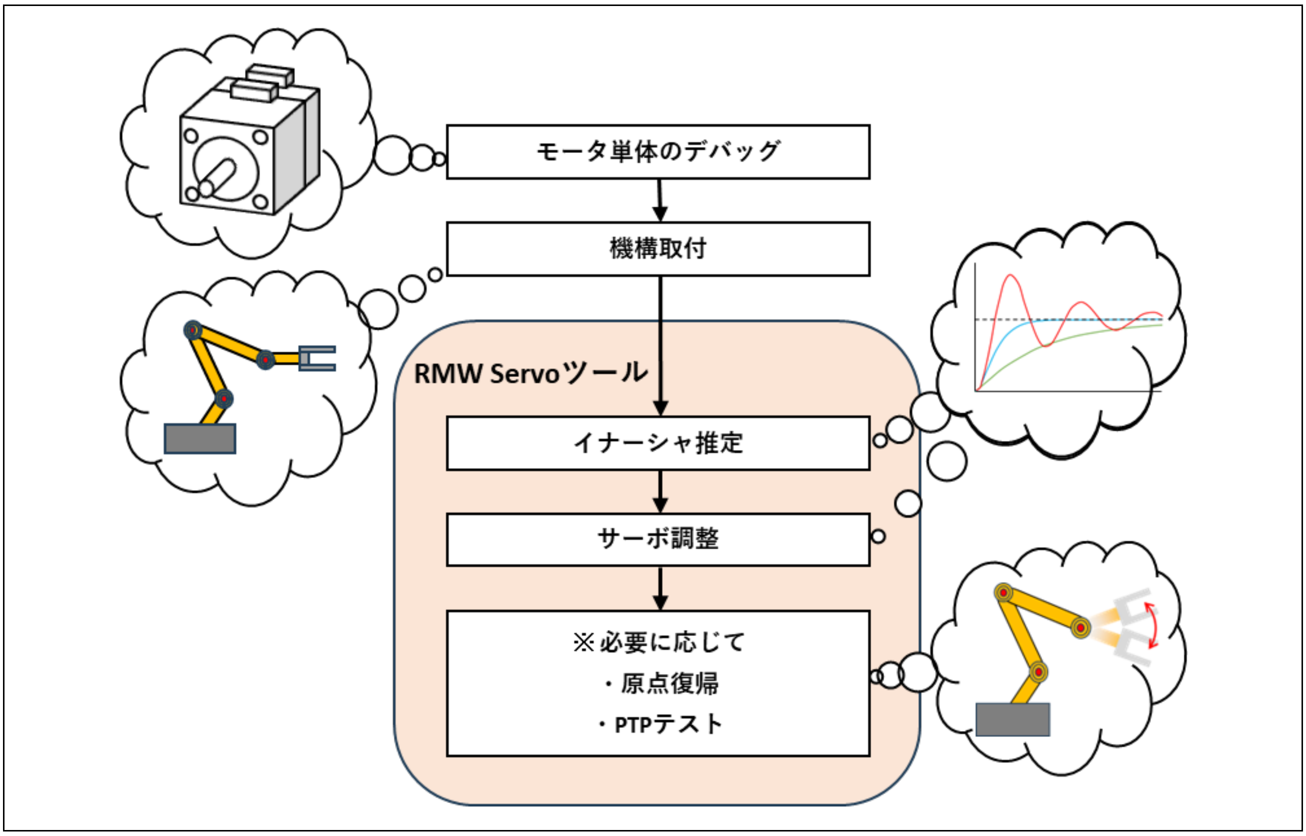


図 13-4　Servo ツール使用の流れ

モータを機構に取り付けた後は、初めに「イナーシャ推定」を実施してください。イナーシャは、加速度(速度)と必要トルク(電流)の計算に関連し、サーボ制御を行ううえで基準となる重要なパラメータです。イナーシャは以降のモータの動作すべてに影響があるため、必ずイナーシャ推定を実施し、できるだけ正確な値を設定してください。

イナーシャ推定後、「サーボ調整」を実施してください。使用する環境（装置・機構）に合わせてサーボ制御を行うために、位置制御の応答性、速度制御の応答性を調整します。

サーボ調整後、必要に応じて「原点復帰」「PTP 動作」を GUI 上で試運転することで、パラメータの妥当性を確認することができます。

13.2 特徴

- イナーシャ推定では、モータ軸に接続されているロータ+負荷イナーシャを推定することが可能です。
- サーボ調整では、位置制御方式や各制御系固有周波数などのサーボ設定を行うことが可能です。
- 原点復帰では、原点復帰方法、復帰速度などを設定することが可能です。
- PTP 動作では、1 軸に対して、PTP 動作(Point to Point)を行うことが可能です。

13.3 イナーシャ推定

13.3.1 機能説明

13.3.1.1 イナーシャとは

慣性(慣性モーメント)のことで、物体がその時の状態を維持しようとする大きさを表します。イナーシャが小さい時、物体は回りやすく止まりやすくなります。イナーシャが大きい時、物体は回りにくく止まりにくくなります。

モータの回転子の慣性モーメントをロータイナーシャ、モータに取り付けた機構側の慣性モーメントを負荷イナーシャといいます。

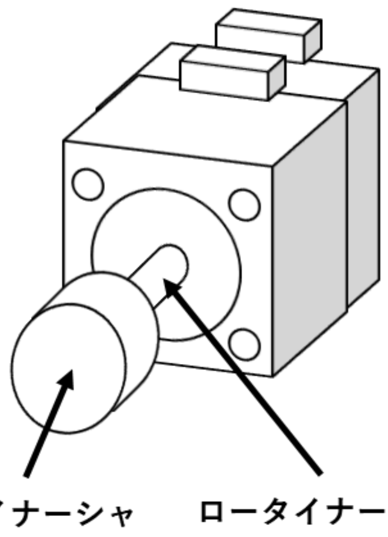


図 13-5 モータと機構のイナーシャ

13.3.1.2 イナーシャ推定とは

モータ軸に接続されているロータイナーシャと負荷イナーシャの合計を推定し、モータのロータイナーシャとの比(ロータイナーシャ比)を決定後、トルク調整を行います。

イナーシャ推定時、モータが回転しますので、可動部が周囲の人や物に衝突しないよう十分にご注意ください。

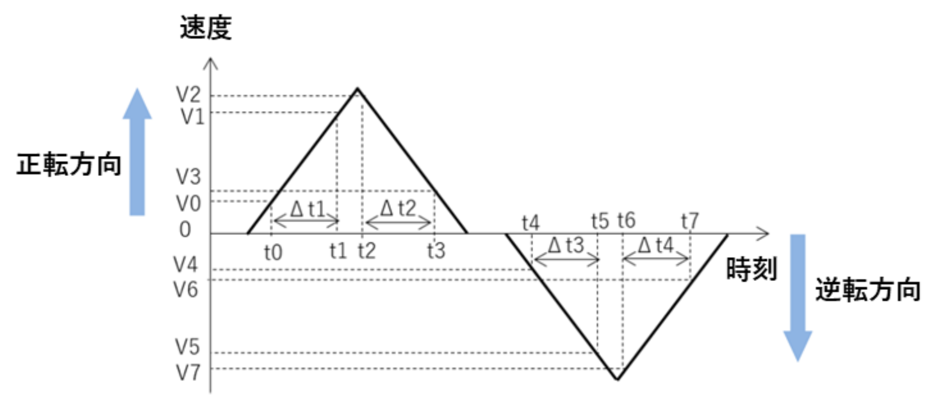


図 13-6 イナーシャ推定時動作

イナーシャの推定は、位置制御を用いてモータを正転方向と逆転方向に同じ移動量で動作させます。この時の正転方向加速時間、正転方向減速時間、逆転方向加速時間、及び逆転方向減速時間、各時間の駆動電流平均値及び平均加速度を用いてイナーシャを推定します。

13.3.1.3 イナーシャ推定の重要性

イナーシャは、加速度(速度)と必要トルク(電流)の計算に関連し、サーボ制御を行ううえで基準となる重要なパラメータです。サーボ制御におけるイナーシャ推定の影響範囲を以下に示します。

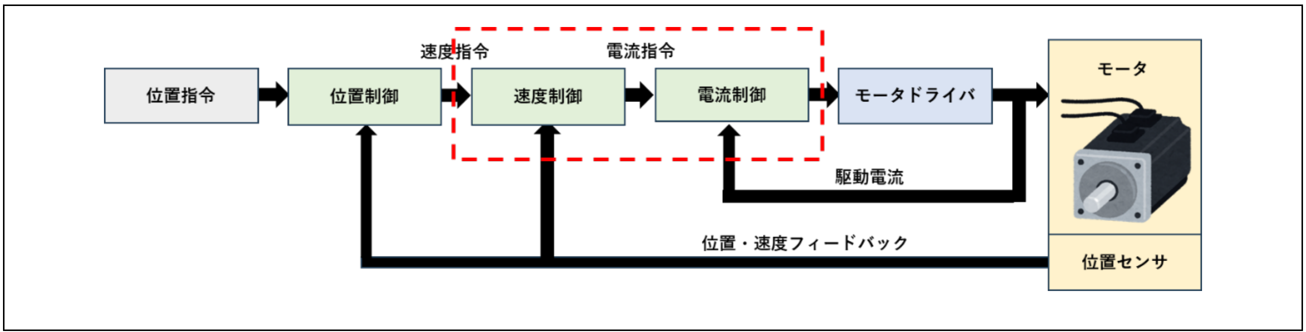


図 13-7 サーボ制御におけるイナーシャ推定の影響範囲

推定したイナーシャが適正値の場合、正確なトルクを算出することができます。適正値より低い場合、低いトルクが算出され、応答性が遅くなります。適正値より高い場合、高いトルクが算出され、モータが不安定な動作をしやすくなります。

イナーシャは以降のモータの動作すべてに影響があるため、できるだけ正確な値を設定する必要があります。Servo ツールにて正確な値を設定するだけでなく、実際の値を把握しておくことがプログラム開発において重要となります。

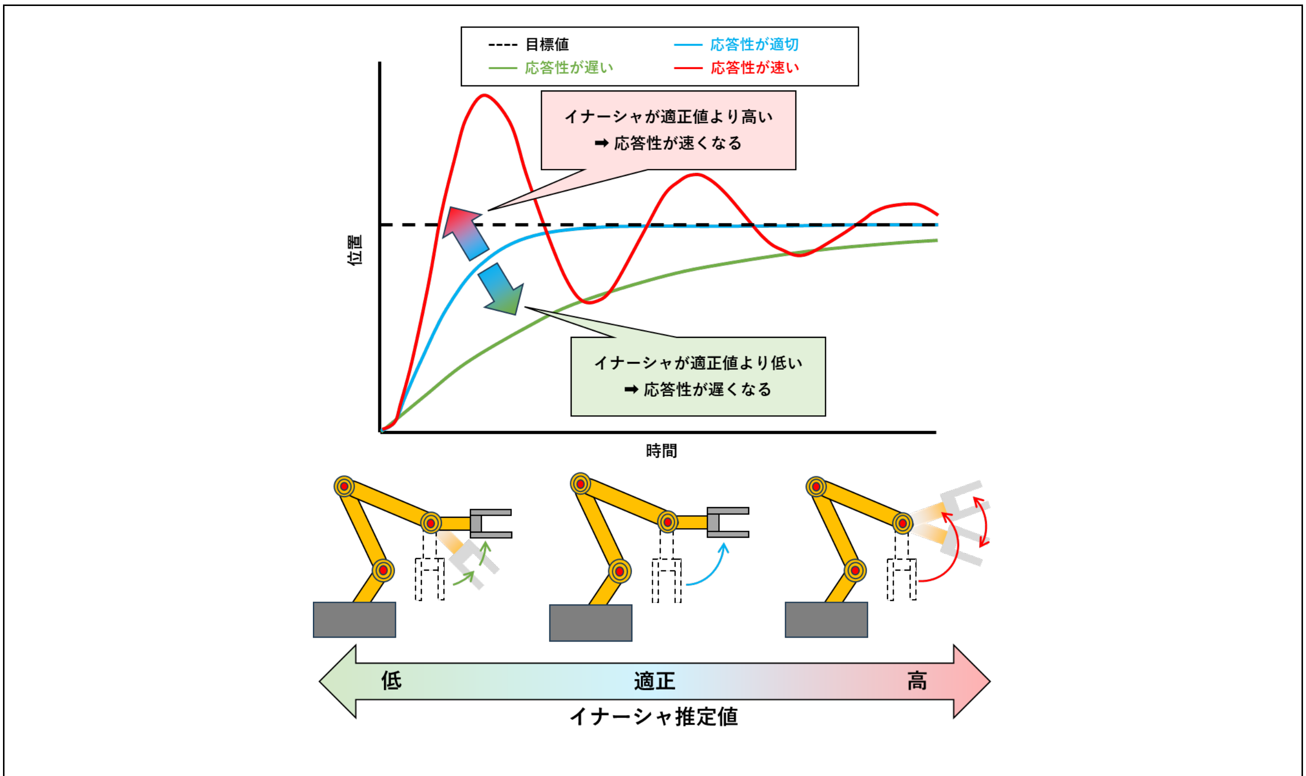


図 13-8 サーボ制御におけるイナーシャの影響

13.3.2 画面構成・パラメーター覧

タブ[Inertia Estimation]画面内の各部の名称と機能を説明します。
タブ[Inertia Estimation]は、モータ軸に接続されているイナーシャを推定できます。

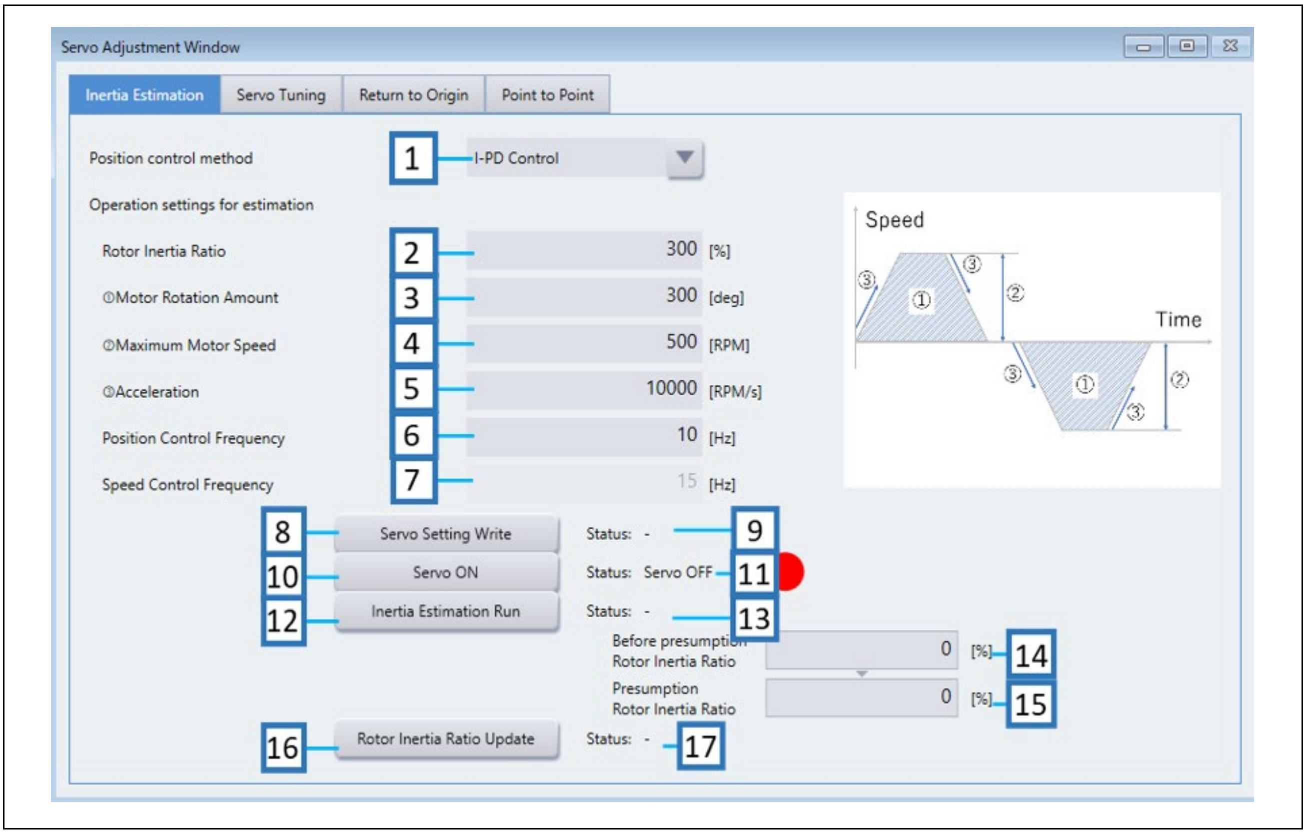


図 13-9 Inertia Estimation タブ画面

表 13-1　Inertia Estimation タブ画面の機能

No.	名称	説明
1	Position control method	位置制御方式を指定します。 I-PD Control / PID Control 基本的に初期値「I-PD Control」のままで問題ありません。
2	Rotor Inertia Ratio	ロータイナーシャと負荷イナーシャの比を設定します。 計算済みの場合はその値を入力してください。 不明な場合は初期値(300%)で推定を行ってください。
3	Motor Rotation Amount	モータの移動量を設定します。 周囲の人や物にぶつからない移動量を設定してください。 移動量が小さいと推定に失敗しやすくなります。
4	Maximum Motor Speed	モータの最大速度を設定します。
5	Acceleration	加速度を設定します。
6	Position Control Frequency	位置制御系固有周波数を設定します。 基本的に初期値 10Hz のままで問題ありません。 （「13.4 サーボ調整」で調整します）
7	Speed Control Frequency	速度制御系固有周波数を設定します。 「Position control method」が「PID Control」の場合に入力できます。 No.6 の位置制御系固有周波数の 1.5～3.0 倍の値にしてください。 （「13.4 サーボ調整」で調整します）
8	Servo Setting Write ボタン	No.1～No.7 の設定を書き込みます。
9	Servo Setting Write Status	Servo Setting Write ボタン (No.8)クリック後の設定書き込み状況を表示します。
10	Servo ON/OFF ボタン	サーボの実行/停止を切り替えます。 ボタンの表示文字とクリック時の動作 Servo ON：クリックするとサーボ制御を開始します。 Servo OFF：クリックするとサーボ制御を停止します。
11	Servo Status	サーボの実行状態を表示します。 「ON」の場合はランプを緑色に、「OFF」の場合ランプを赤色に表示します。
12	Inertia Estimation 開始/終了ボタン	Run 表示をクリックするとイナーシャ推定を開始し、Stop 表示をクリックするとイナーシャ推定を停止します。 ボタンの表示文字は、イナーシャ推定実行中は「Inertia Estimation Stop」、イナーシャ推定停止中は「Inertia Estimation Run」となります。
13	Inertia Estimation Status	イナーシャ推定の実行状態を表示します。
14	Before presumption Rotor Inertia Ratio	推定前ロータイナーシャ比を表示します。
15	Presumption Rotor Inertia Ratio	推定ロータイナーシャ比を表示します。
16	Rotor Inertia Ratio Update ボタン	推定ロータイナーシャ比を書き込みます。
17	Rotor Inertia Ratio Update Status	推定ロータイナーシャ比の書き込み状況を表示します。

13.3.3 操作説明

イナーシャ推定の手順を以下に示します。

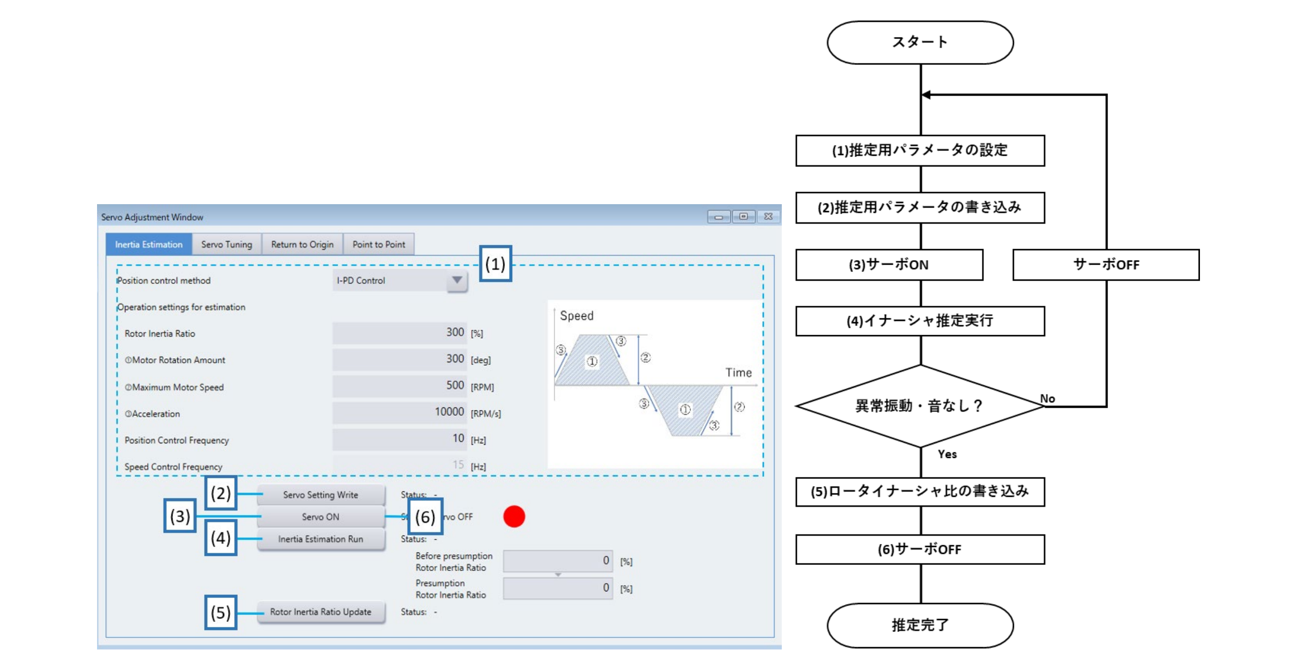


図 13-10 イナーシャ推定操作手順

(1) イナーシャ推定用パラメータの設定

イナーシャ推定のパラメータを設定します。
（詳しくは、「13.3.2 画面構成・パラメータ一覧」を参照してください）

(2) イナーシャ推定用パラメータの書き込み

「Servo Setting Write」ボタンを押下すると(1)で設定したパラメータを書き込みます。

(3) サーボ制御の開始

イナーシャ推定開始前に「Servo ON」ボタンを押下しサーボ制御を開始します。

(4) イナーシャ推定の開始／停止

「Servo ON」状態で「Inertia Estimation Run」ボタンを押下するとイナーシャ推定を開始します。イナーシャ推定完了後は自動的に停止します。「Inertia Estimation Stop」ボタンを押下するとイナーシャ推定を停止します。

※モータから異常な振動や音がする場合、「Servo OFF」ボタンを押下しサーボ制御を終了します。(1)へ戻りパラメータ(「Rotor Inertia Ratio」)を見直してください。

※問題ない場合、次の工程へ進んでください。

(5) 推定ロータイナーシャ比の書き込み

イナーシャ推定完了後、「Rotor Inertia Ratio Update」ボタンを押下すると推定ロータイナーシャ比を書き込みます。

(6) サーボ制御の停止

「Servo OFF」ボタンを押下するとサーボ制御を停止します。

パラメータ設定ガイド・注意事項

- 異常な振動・音がする場合、「Rotor Inertia Ratio」を見直してください。
  「Rotor Inertia Ratio」の見直しで改善されない場合、「Position Control Frequency」を小さくしてください。ただし、「Speed Control Frequency」を「Position Control Frequency」の 1.5 倍～3.0 倍にしてください。
- 「Position Control Frequency」が小さいと位置の追従性が下がります。
- 「Speed Control Frequency」が大きいとモータが振動し騒音を発します。
- 「Position control method」の選び方について、応答を速くしたい場合は「PID Control」、オーバーシュートを抑えつつ外乱抑制を行いたい場合は「I-PD Control」を選択してください。
- モータ運転時の駆動電流よりイナーシャを推定しますので、移動量が小さい、運転速度が小さい、加速度が小さいなど駆動電流が少なくなる条件では、イナーシャを小さく推定しやすくなります。可能な範囲で移動量・運転速度・加速度を大きくしてイナーシャ推定を実行してください。

13.4 サーボ調整

13.4.1 機能説明

13.4.1.1 サーボ調整とは

使用する環境（装置・機構）に合わせてサーボ制御を行うために、位置制御の応答性、速度制御の応答性を適切に調整します。
（詳しくは、「13.1 概要」を参照してください）

13.4.1.2 制御周波数とは

位置制御と速度制御を組み合わせて制御し、それぞれの制御の応答周波数を位置制御周波数・速度制御周波数といいます。

位置制御周波数を高く設定すると、位置制御ループのゲインが上がるため、位置偏差を小さく抑えることが可能となります。位置制御周波数は必要以上に大きく設定しないようにしてください。

速度制御周波数を高く設定すると、速度制御ループのゲインが上がるため外乱による速度変動を増幅し、振動・騒音が大きくなる場合があります。速度制御周波数は最大でも 100Hz 程度を目安とし、必要以上に大きく設定しないようにしてください。

制御ループの安定性確保のため、速度制御周波数は位置制御周波数の 1.5～3.0 倍を目安としてください。

13.4.1.3 制御方式について

PID 制御は、位置偏差に対して比例制御を行い、速度偏差に対して比例制御と積分制御を行う制御方式です。これにより応答性を速くすることができます。

IPD 制御は、位置偏差に対して積分制御のみを行う制御方式です。これにより、応答性を上げても位置決め時の振動を低減させることができます。PID 制御より応答性が遅くなりますが、位置決めの精度が上がります。

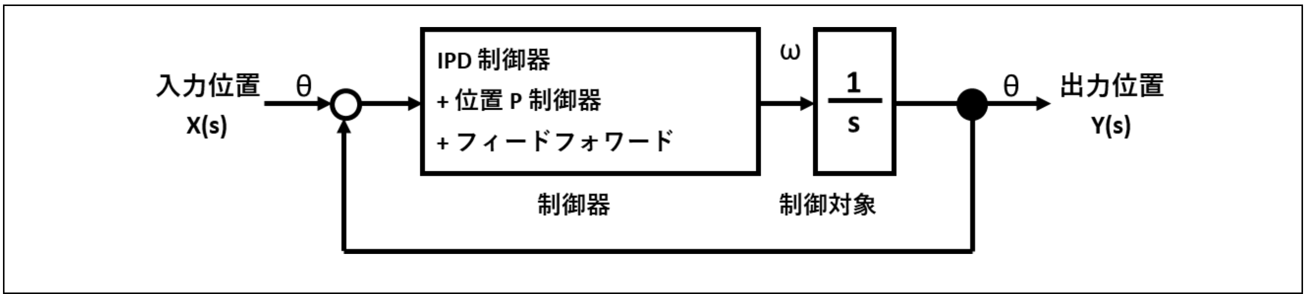


図 13-11　IPD 制御のモデル(位置)

13.4.1.4 サーボ調整例

以下に位置制御周波数と速度制御周波数を変更した時の検出速度の波形を示します。

※速度制御周波数は位置制御周波数の 1.5 倍とし、動作パラメータはすべて同じとします。

(ア) 位置制御周波数が低いと応答性が悪くなり、速度変動・位置偏差が大きくなります。位置制御周波数を大きくし、速度変動・位置偏差を低減してください。

(イ) 速度制御周波数が大きいとモータが振動し騒音を発しやすくなります。速度制御周波数を小さくし、振動・騒音を低減してください。

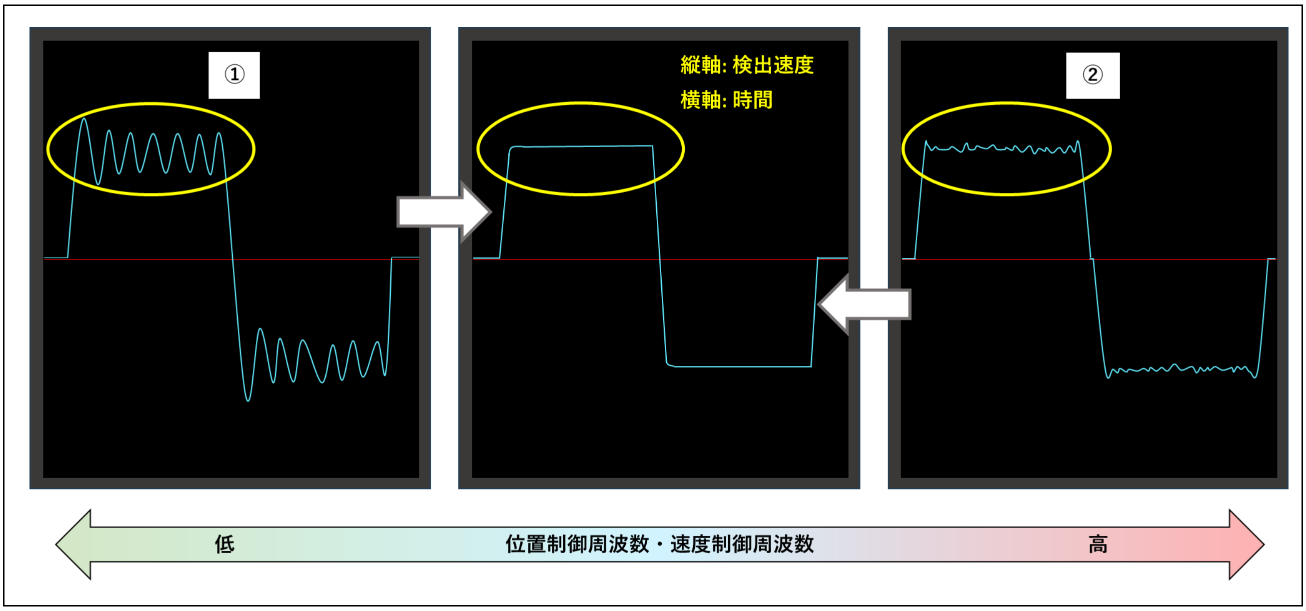


図 13-12 制御周波数と検出速度の波形

13.4.2 画面構成・パラメータ一覧

タブ[Servo Tuning]画面内の各部の名称と機能を説明します。
タブ[Servo Tuning]は、位置制御方式や各制御系固有周波数などのサーボ設定を実行できます。

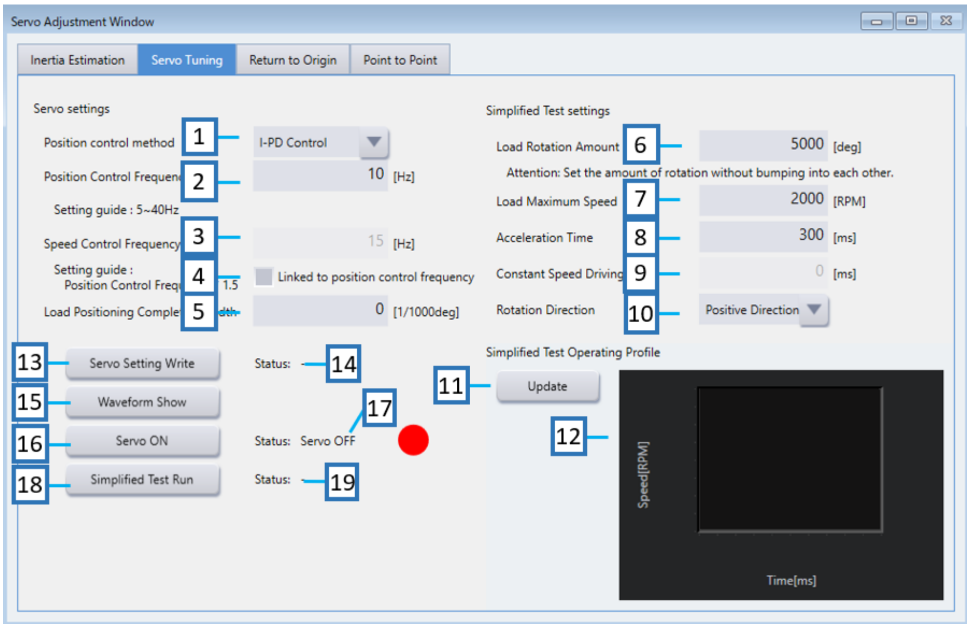


図 13-13 Servo Tuning タブ画面

表 13-2　Servo Tuning タブ画面の機能

No.	名称	説明
1	Position control method	位置制御方式を指定します。 I-PD Control / PID Control
2	Position Control Frequency	位置制御系固有周波数を設定します。
3	Speed Control Frequency	速度制御系固有周波数を設定します。 「Position control method」が「PID Control」の場合に入力ができます。 No.2 の位置制御系固有周波数の 1.5～3.0 倍の値にしてください。
4	Linked to position control frequency	チェックを入れると位置制御系固有周波数を 1.5 倍にした値を速度制御系固有周波数に自動設定します。No.3 に設定した値は上書きされます。
5	Load Positioning Completion Width	負荷の位置決め完了幅を設定します。 目標位置からの許容誤差を設定してください。
6	Load Rotation Amount	簡易試運転時の負荷の移動量を設定します。
7	Load Maximum Speed	簡易試運転時の負荷の最大速度を設定します。 システムで想定している速度を設定してください。
8	Acceleration Time	簡易試運転時の加速時間を設定します。 停止状態から最大速度に達するまでの時間を設定してください。
9	Constant Speed Driving Time	簡易試運転時の等速運転時間を表示します。
10	Rotation Direction	簡易試運転時の回転方向を指定します。
11	Update ボタン	動作プロファイルの表示を更新します。
12	動作プロファイル	設定した内容の動作プロファイルを表示します。
13	Servo Setting Write ボタン	No.1～No.10 の設定を書き込みます。
14	Servo Setting Write Status	Servo Setting Write ボタン(No.13)クリック後の設定書き込み状況を表示します。
15	Waveform Show ボタン	Waveform 画面を表示します。
16	Servo ON/OFF ボタン	サーボの実行/停止を切り替えます。 ボタンの表示文字とクリック時の動作 Servo ON：クリックするとサーボ制御を開始します。 Servo OFF：クリックするとサーボ制御を停止します。
17	Servo Status	サーボの実行状態を表示します。 「ON」の場合はランプを緑色に、「OFF」の場合ランプを赤色に表示します。
18	Simplified Test 開始/終了ボタン	Run 表示をクリックすると簡易試運転を開始し、Stop 表示をクリックすると簡易試運転を停止します。 ボタンの表示文字は、簡易試運転実行中は「Simplified Test Stop」、簡易試運転停止中は「Simplified Test Run」となります。
19	Simplified Test Status	簡易試運転の実行状態を表示します。

パラメータとプロファイルの関係を以下に示します。

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

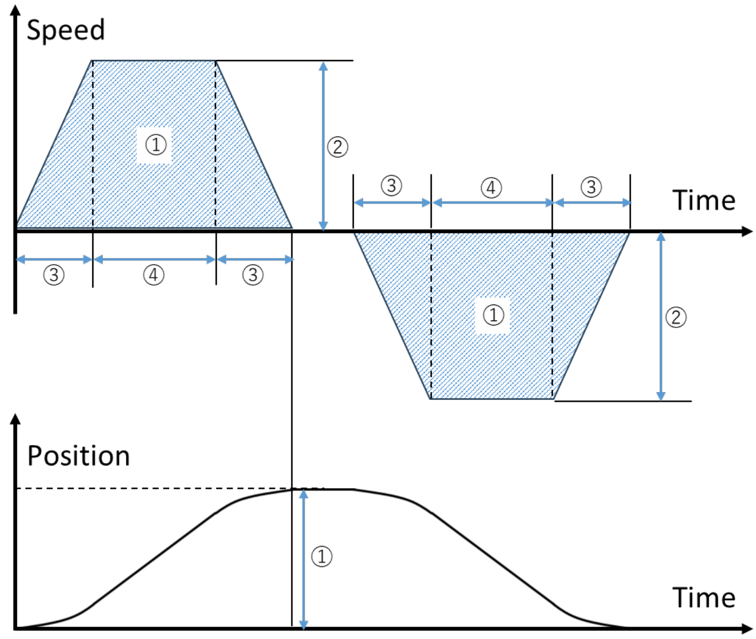


図 13-14 パラメータとプロファイルの関係

13.4.3 操作説明

サーボ調整の手順を以下に示します。

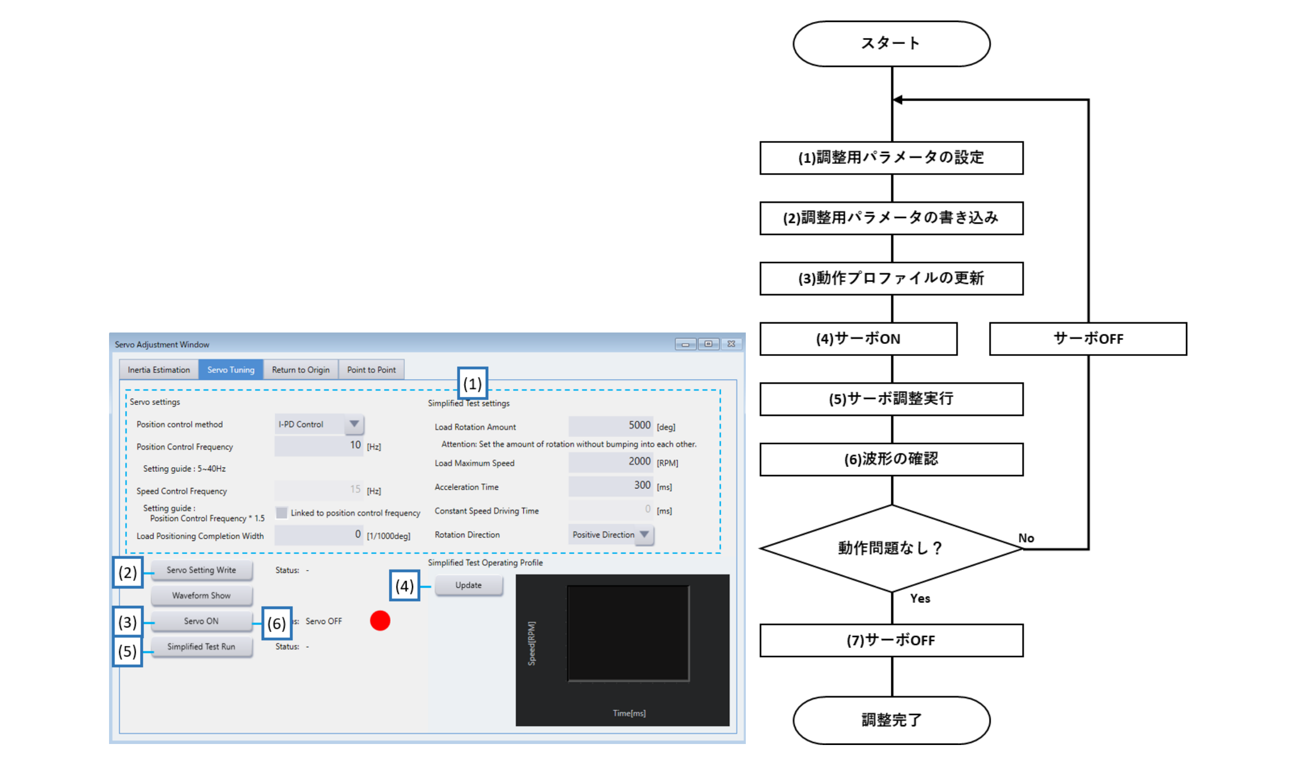


図 13-15 サーボ調整操作手順

(1) サーボ調整用パラメータの設定

サーボ調整のパラメータを設定します。
（詳しくは、「13.4.2 画面構成・パラメータ一覧」を参照してください）

(2) サーボ調整用パラメータの書き込み

「Servo Setting Write」ボタンを押下すると(1)で設定したパラメータを書き込みます。

(3) 動作プロファイルの更新

「Update」ボタンを押下すると動作プロファイルを更新します。

(4) サーボ制御の開始

サーボ調整開始前に「Servo ON」ボタンを押下しサーボ制御を開始します。

(5) サーボ調整の開始／停止

「サーボ ON」状態で「Simplified Test Run」ボタンを押下するとサーボ調整を開始します。
サーボ調整完了後は自動的に停止します。「Simplified Test Stop」ボタンを押下するとサーボ調整を終了します。

(6)　波形の確認

「Waveform Show」ボタンを押下し、波形を表示します。波形を確認しサーボパラメータの調整が必要と判断した場合、「Servo OFF」ボタンを押下しサーボ制御を終了してください。(1)へ戻りパラメータ(「Position Control Frequency」)を見直してください。

問題ない場合、次の工程へ進んでください。

（詳しくは、「13.4.1.4 サーボ調整例」を参照してください）

(7)　サーボ制御の停止

「Servo OFF」ボタンを押下するとサーボ制御を停止します。

パラメータ設定ガイド・注意事項

- 異常な振動・音がする場合、「Position Control Frequency」を小さくしてください。ただし、「Speed Control Frequency」を「Position Control Frequency」の 1.5 倍～3.0 倍にしてください。
- 「Position Control Frequency」が小さいと位置の追従性が下がります。
- 「Speed Control Frequency」が大きいとモータが振動し騒音を発します。
- 「Position control method」の選び方について、応答を速くしたい場合は「PID Control」、オーバーシュートを抑えつつ外乱抑制を行いたい場合は「I-PD Control」を選択してください。
- 動作速度によって調整する制御周波数の値が変わるため、最大速度にはシステムで想定している速度を設定してください。

13.5 原点復帰

13.5.1 機能説明

13.5.1.1 原点復帰とは

装置の電源を投入した際に、機械の原点を決めるための機能です。

原点復帰を行わずに運転を行うと、機械の原点が決まっておらず位置の管理ができていないため、位置決めを正しく実行できなくなります。

原点復帰調整時はモータが回転するため、可動部が周囲の人や物に衝突しないよう十分にご注意ください。

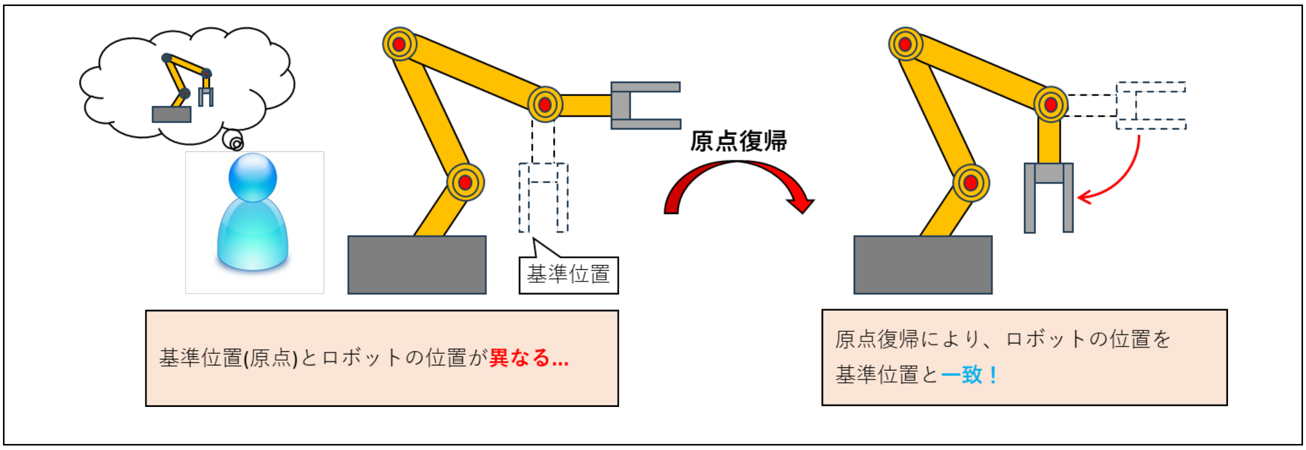


図 13-16　原点復帰とは

13.5.1.2 押し当て方式とは

押し当て方式は、メカをストッパなどに接触させ、指定量だけ移動し停止した位置を原点とする方式です。
原点復帰中に干渉物に接触した場合、その位置から「原点復帰戻り量」だけ移動した位置を原点としてしまうのでご注意ください。

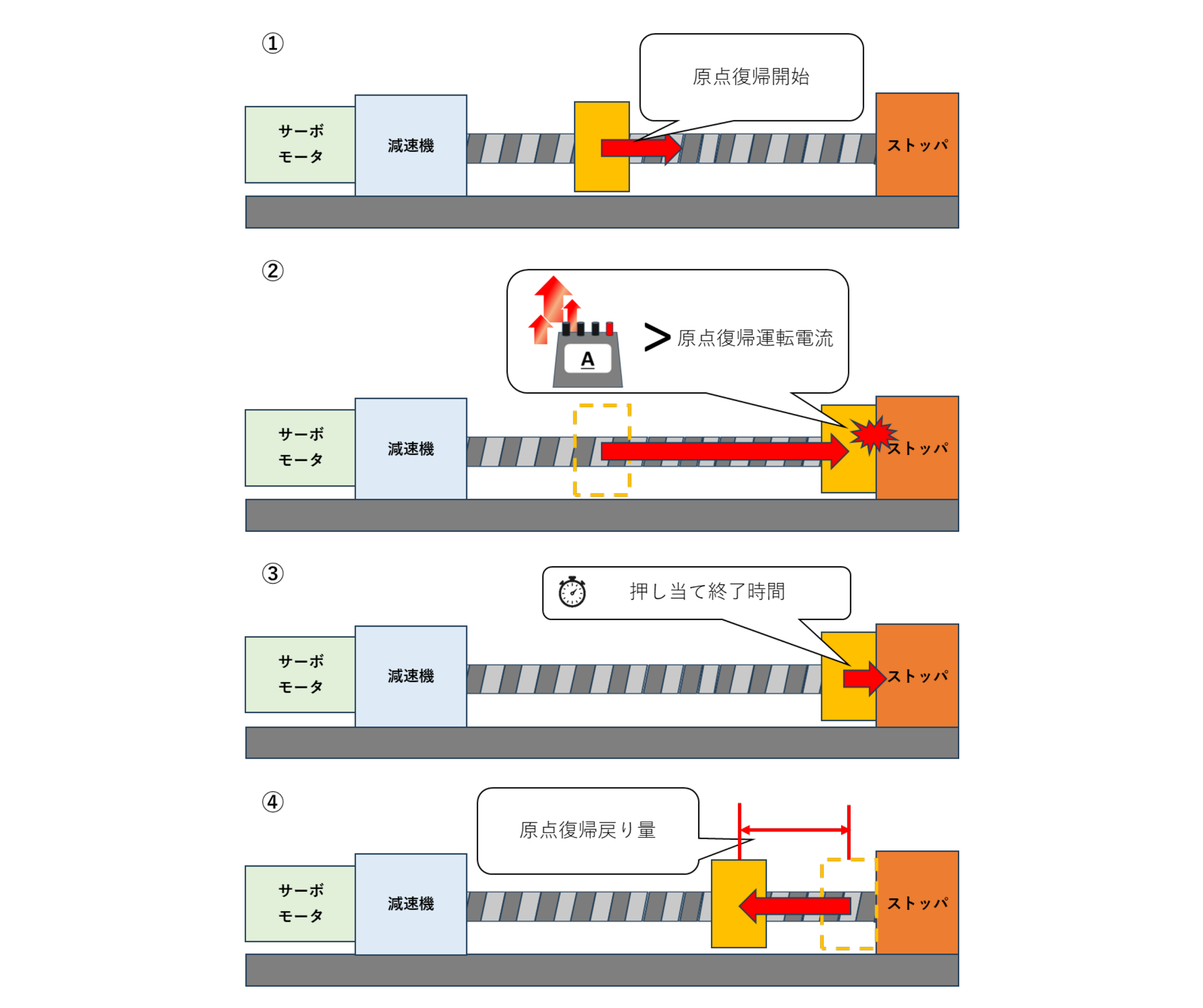


図 13-17 原点復帰動作押し当て方式

押し当て方式の動作を以下に示します。

① 原点復帰動作を開始します。
② 機構がストッパに接触することでトルク(電流)が大きくなり、「原点復帰運転電流」に達します。
③ 電流が「原点復帰運転電流」を超えている時間が「押し当て終了時間」を超えると戻り動作を開始します。
④ モータを反転し、「原点復帰戻り量」だけ移動します。移動した位置を原点とします。

13.5.2 画面構成・パラメータ一覧

タブ[Return to Origin]画面内の各部の名称と機能を説明します。
タブ[Return to Origin]は、原点復帰方法、復帰速度などの設定を実行できます。

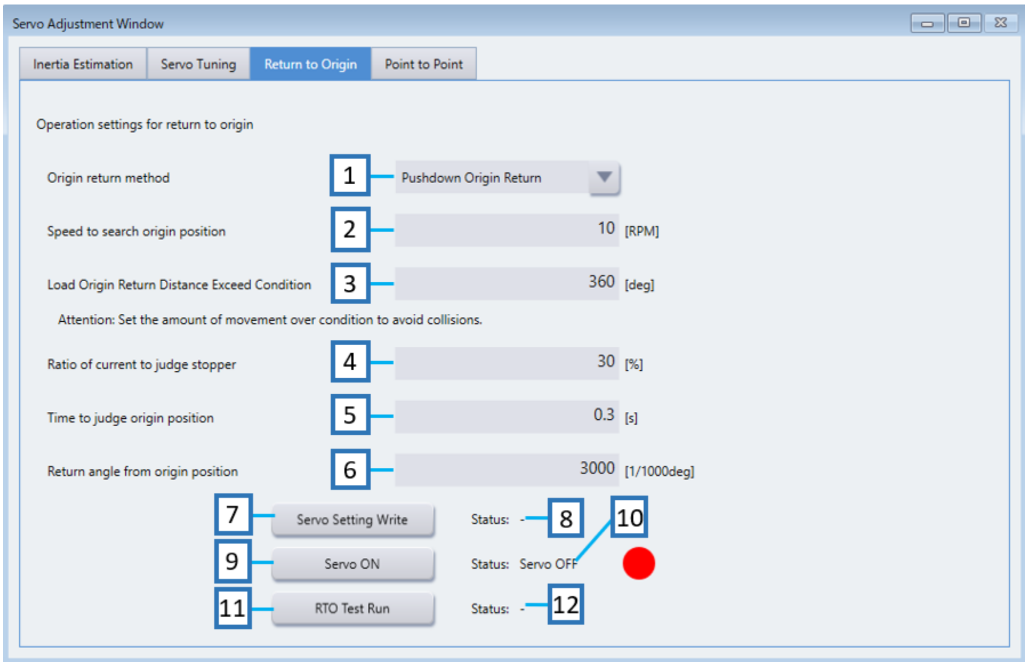


図 13-18 Return to Origin タブ画面

表 13-3　Return to Origin タブ画面の機能

No.	名称	説明
1	Origin return method	原点復帰方法を指定します。
2	Speed to search origin position	負荷の原点復帰速度を設定します。
3	Load Origin Return Distance Exceed Condition	原点復帰時の負荷の移動量の最大値を設定します。
4	Ratio of current to judge stopper	押し当て原点復帰運転電流を設定します。 定格電流を 100%として設定してください。
5	Time to judge origin position	機械ストッパを押し当てる時間を設定します。
6	Return angle from origin position	原点にする機械ストッパからの位置を設定します。
7	Servo Setting Write ボタン	No.1～No.6 の設定を書き込みます。
8	Servo Setting Write Status	Servo Setting Write ボタン(No.7)クリック後の設定書き込み状況を表示します。
9	Servo ON/OFF ボタン	サーボの実行/停止を切り替えます。 ボタンの表示文字とクリック時の動作 Servo ON：クリックするとサーボ制御を開始します。 Servo OFF：クリックするとサーボ制御を停止します。
10	Servo Status	サーボの実行状態を表示します。 「ON」の場合はランプを緑色に、「OFF」の場合ランプを赤色に表示します。
11	RTO Test 開始/終了 ボタン	Run 表示をクリックすると原点復帰テストを開始し、Stop 表示をクリックすると原点復帰テストを停止します。 ボタンの表示文字は、原点復帰テスト実行中は「RTO Test Stop」、原点復帰テスト停止中は「RTO Test Run」となります。
12	RTO Test Status	原点復帰テストの実行状態を表示します。

13.5.3 操作説明

原点復帰の手順を以下に示します。

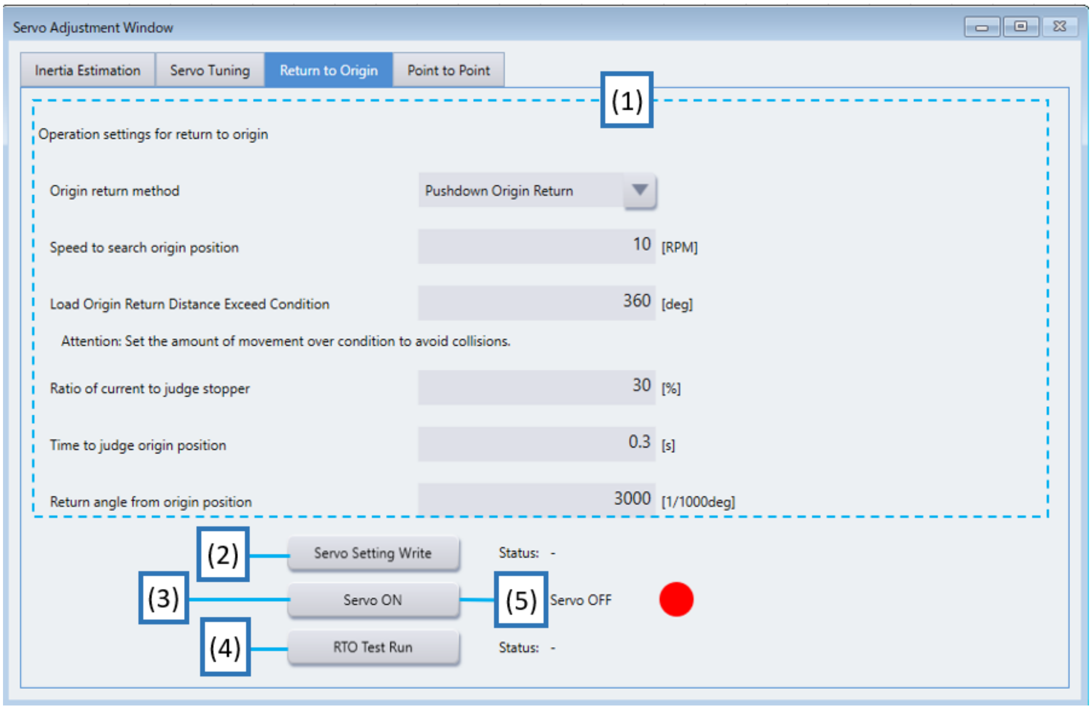


図 13-19　原点復帰操作手順

(1)　原点復帰用パラメータの設定

原点復帰のパラメータを設定します。
（詳しくは、「13.5.2 画面構成・パラメータ一覧」を参照してください）

(2)　原点復帰用パラメータの書き込み

「Servo Setting Write」ボタンを押下すると(1)で入力したパラメータを書き込みます。

(3)　サーボ制御の開始

原点復帰テスト開始前に「Servo ON」ボタンを押下しサーボ制御を開始します。

(4)　原点復帰の開始／停止

「サーボ ON」状態で「RTO Test Run」ボタンを押下すると原点復帰テストを開始します。
原点復帰テスト完了後は自動的に停止します。「RTO Test Stop」ボタンを押下すると原点復帰テストを終了します。

(5)　サーボ制御の停止

「Servo OFF」ボタンを押下するとサーボ制御を停止します。

パラメータ設定ガイド・注意事項

- 「Speed to search origin position」は低い速度を設定してください。速度が極端に高いとストッパに接触した際に、機器が破損する恐れがあります。
- 「Load Origin Return Distance Exceed Condition」は移動量に対し十分大きい値を設定してください。設定した値の範囲を移動してストッパが見つからない場合、その時点で原点復帰は停止します。

13.6 Point to Point 動作

13.6.1 機能説明

13.6.1.1 PTP 動作とは

ある地点から別の地点までの移動(Point To Point)動作で、移動量・最大速度・加速時間を指定することで動作します。簡単な位置決め制御に利用します。

ロボットの姿勢に依存し動きやすい経路を自動で辿ることと、複雑な計算を必要としないことから高速で動作することが可能となります。

自由な経路を辿るため経路に障害物がないか、他の機械と干渉してしまわないか注意を払う必要があります。

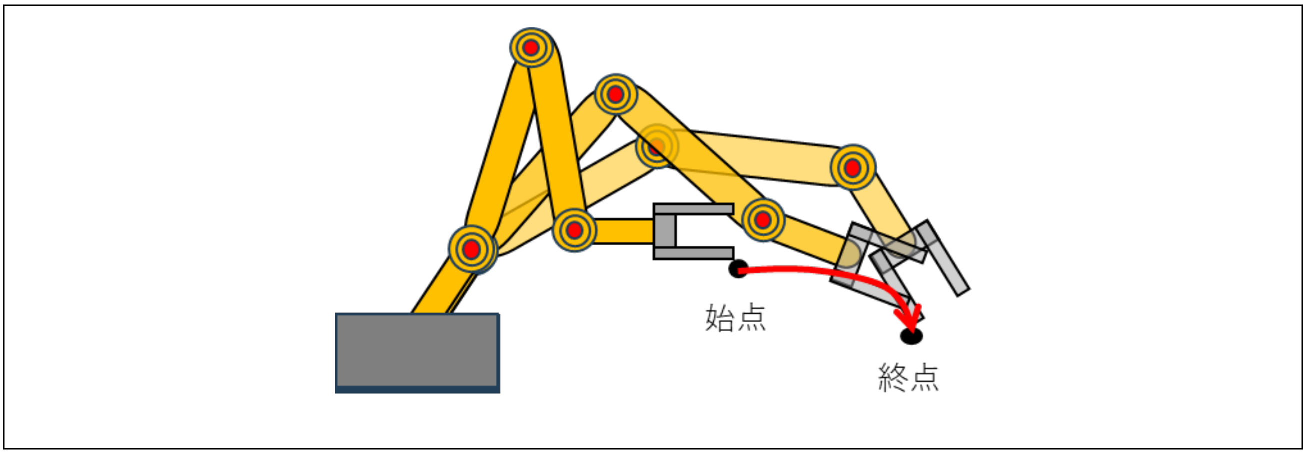


図 13-20 PTP 動作例

13.6.2 画面構成・パラメーター覧

タブ[Point to Point]画面内の各部の名称と機能を説明します。
タブ[Point to Point]は、1 軸に対して Point to Point 動作を実行できます。

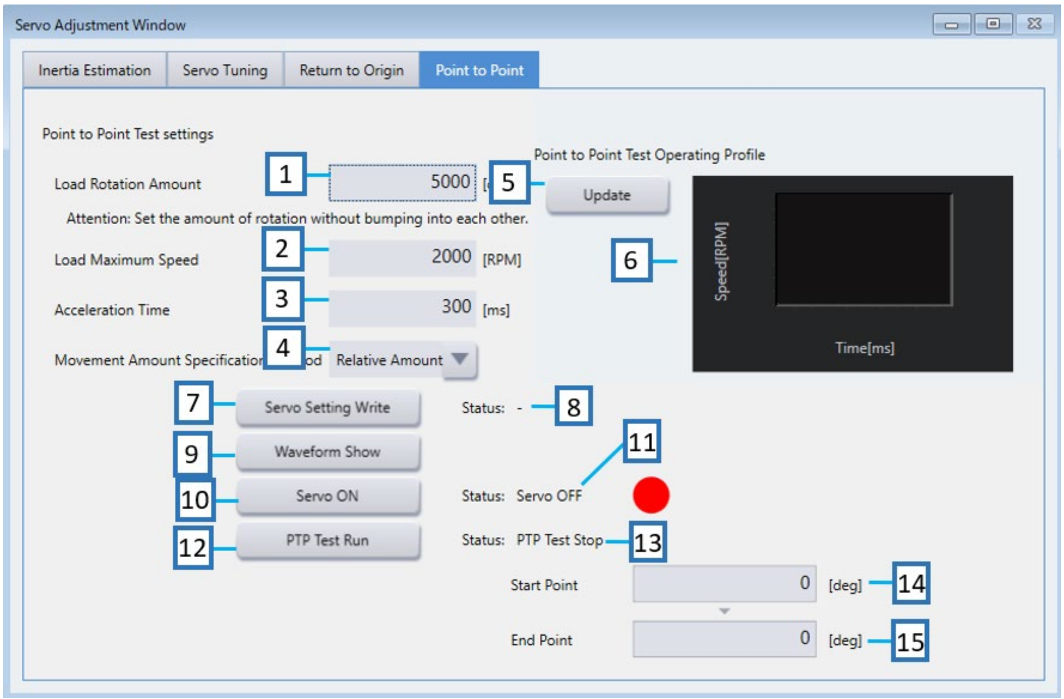


図 13-21 Point to Point タブ画面

表 13-4　Point to Point タブ画面の機能

No.	名称	説明
1	Load Rotation Amount	負荷の移動量を設定します。 No.4 で「Relative Amount」を選択した場合は、現在地点からの相対的な移動量を設定してください。 No.4 で「Absolute Amount」を選択した場合は、原点を基準とし目標地点の位置を設定してください。
2	Load Maximum Speed	負荷の最大速度を設定します。
3	Acceleration Time	加速時間を設定します。 停止状態から最大速度に達するまでの時間を設定してください。
4	Movement Amount Specification Method	移動量指定方法を指定します。 Relative Amount：現在地からの相対的な移動量を設定する場合に選択してください。 Absolute Amount：原点を基準とした移動量を設定する場合に選択してください。
5	Update ボタン	動作プロファイルの表示を更新します。
6	動作プロファイル	設定した内容の動作プロファイルを表示します。
7	Servo Setting Write ボタン	No.1～No.4 の設定を書き込みます。
8	Servo Setting Write Status	Servo Setting Write ボタン(No.7)クリック後の設定書き込み状況を表示します。
9	Waveform Show ボタン	Waveform 画面を表示します。
10	Servo ON/OFF ボタン	サーボの実行/停止を切り替えます。 ボタンの表示文字とクリック時の動作 Servo ON：クリックするとサーボ制御を開始します。 Servo OFF：クリックするとサーボ制御を停止します。
11	Servo Status	サーボの実行状態を表示します。 「ON」の場合はランプを緑色に、「OFF」の場合ランプを赤色に表示します。
12	PTP Test 開始/終了ボタン	Run 表示をクリックすると Point to Point テストを開始し、Stop 表示をクリックすると Point to Point テストを停止します。 ボタンの表示文字は、Point to Point テスト実行中は「PTP Test Stop」、Point to Point テスト停止中は「PTP Test Run」となります。
13	PTP Test Status	Point to Point テストの実行状態を表示します。
14	Start Point	開始位置の値を表示します。
15	End Point	終了位置の値を表示します。

パラメータとプロファイルの関係を以下に示します。

① No.1(Load Rotation Amount)

② No.2(Load Maximum Speed)

③ No.3(Acceleration Time)

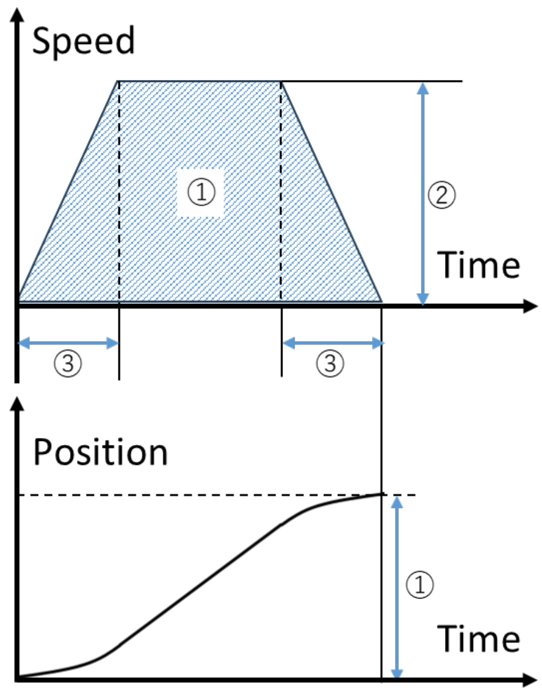


図 13-22 パラメータとプロファイルの関係

13.6.3 操作説明

Point to Point 動作の手順を以下に示します。

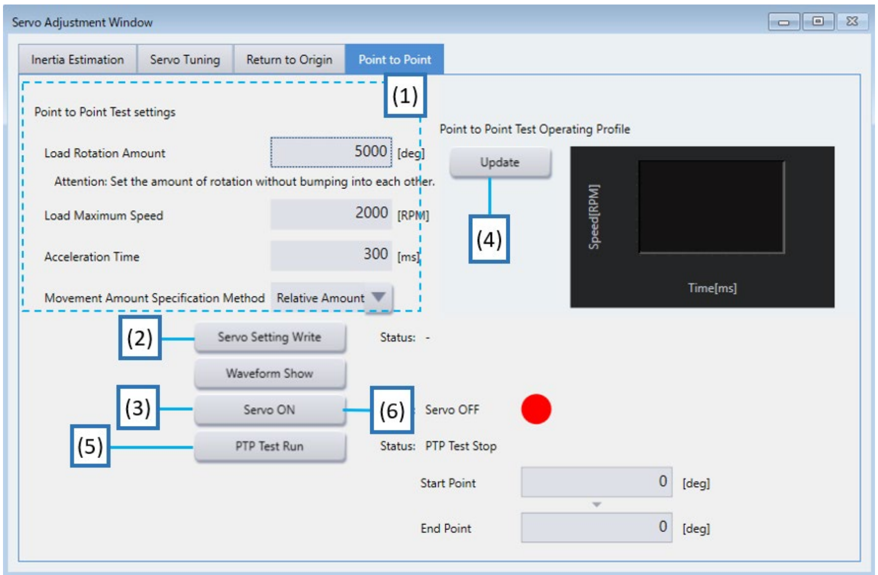


図 13-23 Point to Point 動作操作手順

(1) Point to Point 動作用パラメータの設定

Point to Point 動作用パラメータを設定します。
（詳しくは、「13.6.2 画面構成・パラメータ一覧」を参照してください）

(2) Point to Point 動作用パラメータの書き込み

「Servo Setting Write」ボタンを押下すると(1)で入力したパラメータを書き込みます。

(3) サーボ制御の開始

Point to Point 動作開始前に「Servo ON」ボタンを押下しサーボ制御を開始します。

(4) 動作プロファイルの更新

「Update」ボタンを押下すると動作プロファイルを更新します。

(5) Point to Point 動作の開始／停止

「サーボ ON」状態で「PTP Test Run」ボタンを押下すると Point to Point 動作を開始します。
Point to Point 動作完了後は自動的に停止します。「PTP Test Stop」ボタンを押下すると Point to Point 動作を終了します。

(6) サーボ制御の停止

「Servo OFF」ボタンを押下するとサーボ制御を停止します。

パラメータ設定ガイド・注意事項

- 経路に障害物がないか、他の機械と干渉しないよう注意してください。

13.7 Waveform 画面

Waveform 画面は、Servo Adjustment 画面の Servo Tuning タブまたは Point to Point タブにある「Waveform Show」ボタンを押下すると起動します。

Servo Tuning タブ、Point to Point タブを切り替えると最後に取得した波形を表示します。RMW を終了すると波形はクリアされます。

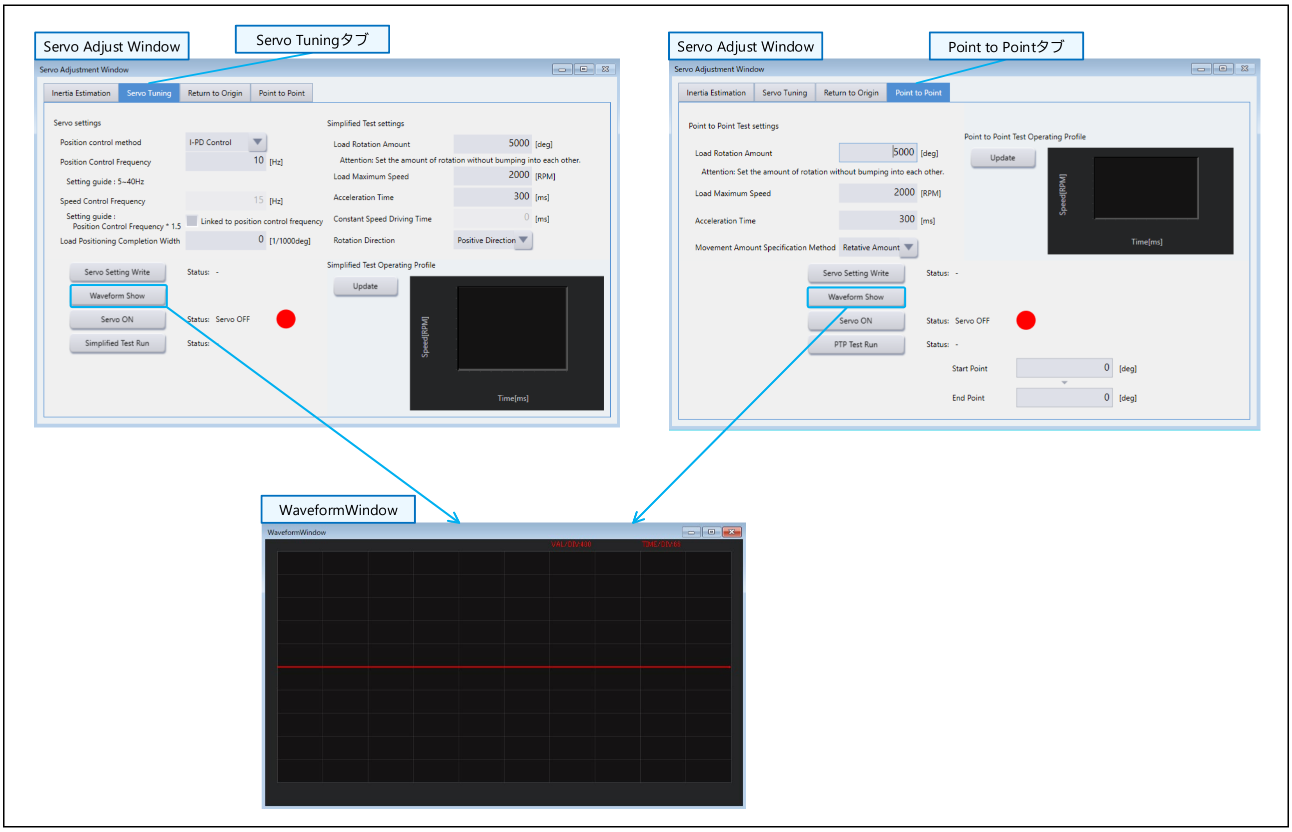


図 13-24 Waveform 画面

14. Tuner ツール

14.1 概要

Tuner ツールは、永久磁石同期モータ固有のパラメータ（抵抗、インダクタンス等）を自動測定し、エンコーダ位置制御やセンサレス速度制御をする際に必要な各種制御パラメータ（PI 制御のゲイン等）を自動調整するツールです。

Tuner ツールは、制御プログラムで Tuning 機能がサポートされている場合に限り利用可能となります。その場合、Main Window の「Select Tool」に「Tuner」が選択可能な状態で表示されます。

注意　本ツールは、すべてのモータに対して有効に機能することを保証するものではありません。

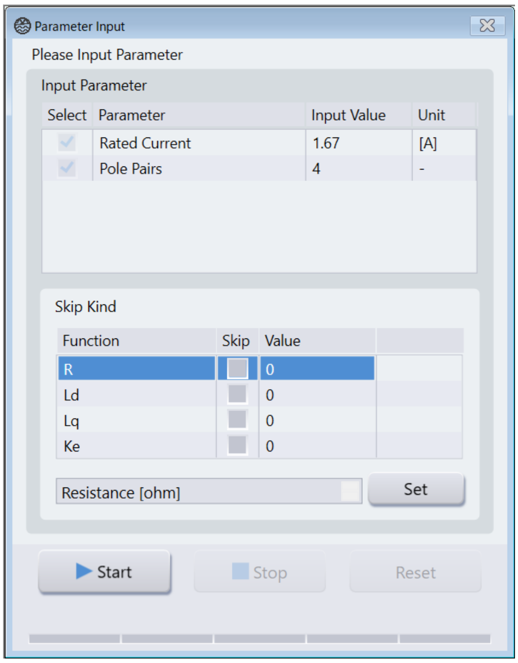
14.2 特徴

- Manual と Easy の 2 つの調整モードを提供します。
- Tuning 結果（調整済みのパラメータ）を PDF 形式のレポートや、ルネサス製制御プログラムのヘッダファイル形式で出力可能です。

14.3 画面構成

Tuner ツールの構成を以下に示します。

(a) Parameter Input 画面



(b) Tune Window（Manual タブ／Easy タブ）

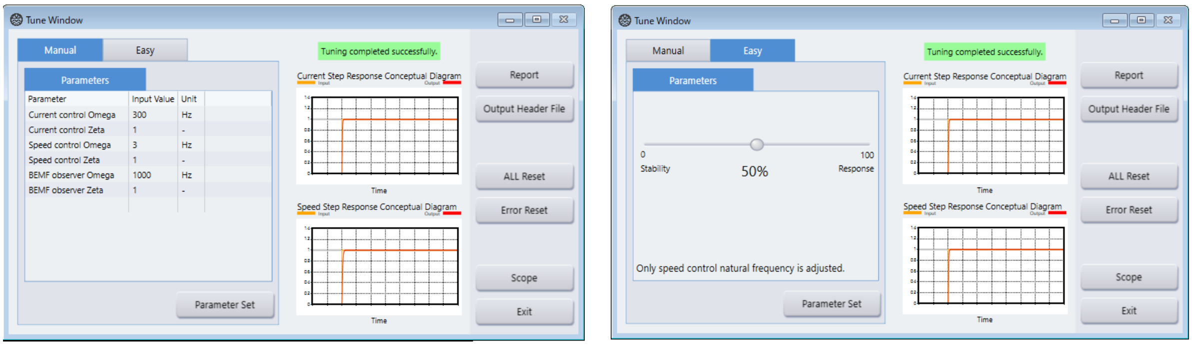


図 14-1 Tuner ツール画面

14.4 操作説明

この章では、チューニング作業の手順について説明します。各画面にある機能の詳細については、「14.5 機能説明」を参照してください。

14.4.1 事前準備

14.4.1.1 実行ファイルの書き込み

(a) Renesas Motor Workbench に同梱されている実行ファイルの場合

Tuner 用の実行ファイル（mot 形式、RA マイコンは hex 形式）を CPU ボードに書き込みます。

- R***_***_SPM_ENCD_FOC_TUNER_***.mot （エンコーダ利用位置制御）
- R***_***_SPM_LESS_FOC_TUNER_***.mot （センサレス速度制御）

(b) ルネサス提供のサンプルプログラム（プロジェクト形式）の場合

ビルドした Tuner 用の実行ファイルを CPU ボードに書き込みます。
Tuner 機能をサンプルプログラムに組み込む手順は、14.6 章を参照ください。

14.4.1.2 Renesas Motor Workbench の起動

デスクトップにあるショートカットアイコンで Renesas Motor Workbench を起動します。

14.4.1.3 RMT ファイルの読み込み

Renesas Motor Workbench に Tuner 用 RMT ファイルを読み込みます。

- R***_***_SPM_ENCD_FOC_TUNER_***.rmt （エンコーダ利用位置制御）
- R***_***_SPM_LESS_FOC_TUNER_***.rmt （センサレス速度制御）

14.4.1.4 通信の確立

Renesas Motor Workbench と評価ボードの通信を確立します。

14.4.1.5 Tuner ツールの起動

Renesas Motor Workbench の Select Tool から Tuner ツールを選択して起動します。

14.4.2 チューニングの実行

14.4.2.1 チューニング用パラメータの入力

Tuner ツールを起動すると、Parameter Input 画面が表示されます。上の Input Parameter 部に、モータの仕様を参照して、Rated Current（定格電流）と Pole Pairs（極対数）を入力します。

あらかじめモータの特性が分かっていて、チューニングをスキップしたい項目がある場合は、下の Skip Kind 部にスキップするパラメータを設定します（詳しくは、「14.5.1 Skip 機能（Parameter Input 画面）」を参照してください）。

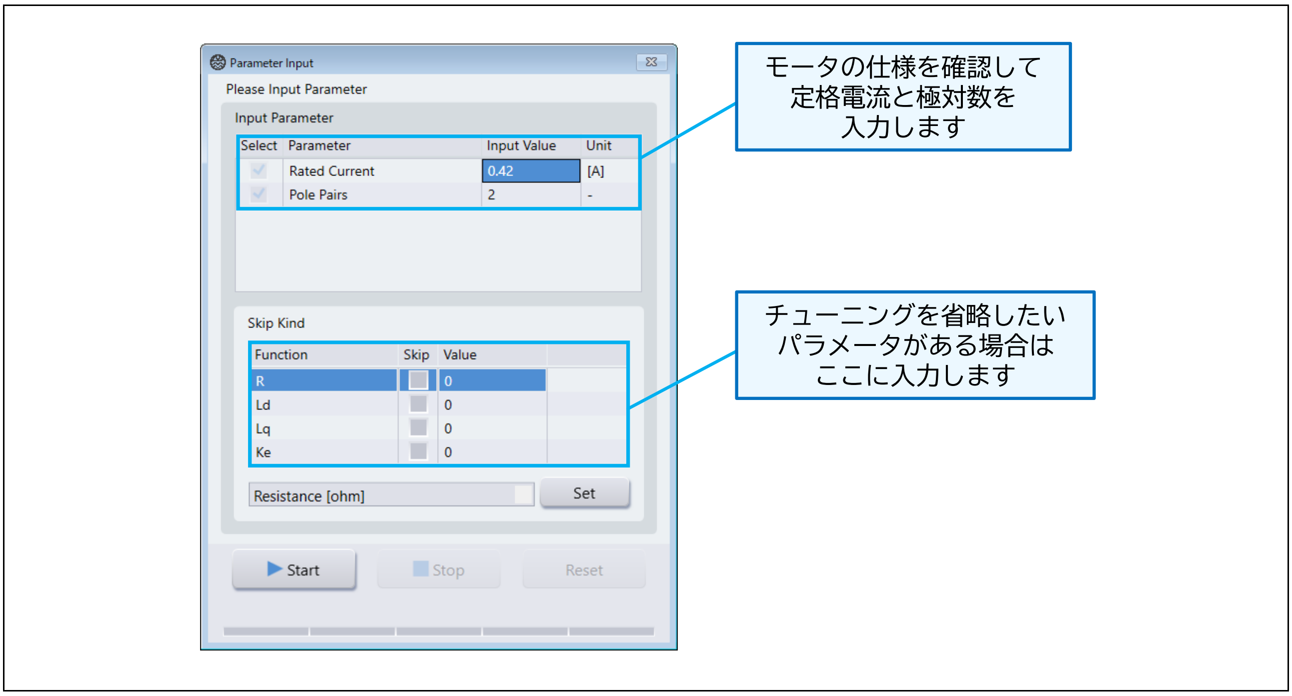


図 14-2 Parameter Input 画面

14.4.2.2　　チューニングの実行

「Start」ボタンをクリックするとチューニングを開始し、チューニング中はプログレスバーが表示されます。チューニングを途中で停止したい場合は、「Stop」ボタンをクリックしてください。チューニングが完了すると Tune Window が表示されます。

注意　チューニングの実行中にモータが回転します。

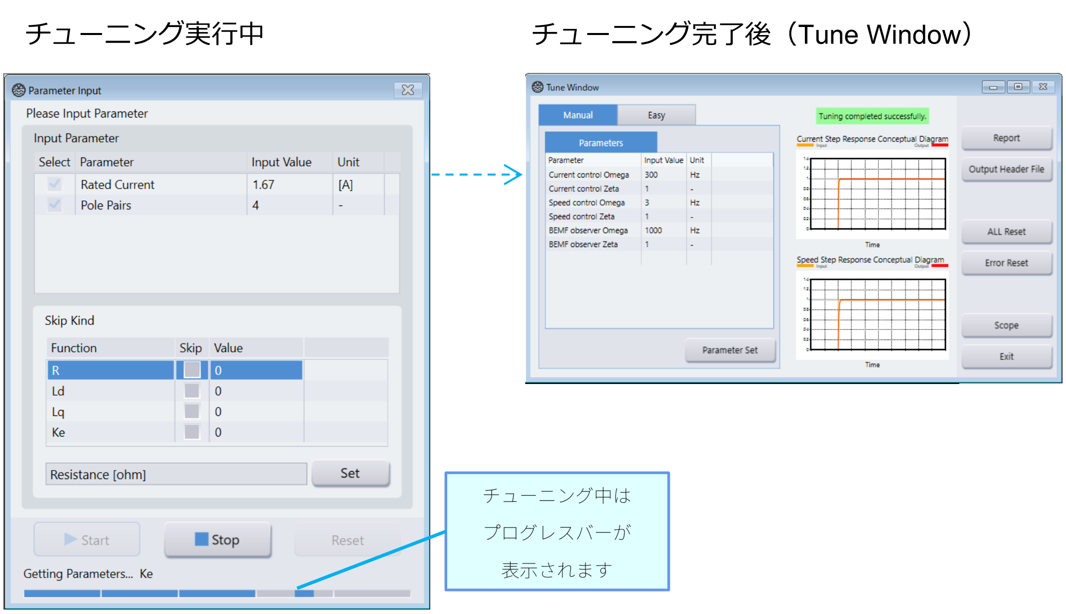


図 14-3　チューニング実行中とチューニング完了後の Tune Window 表示

Tuning 時にエラーが発生した場合は、エラーメッセージを確認して「Reset」ボタンをクリックしてください。

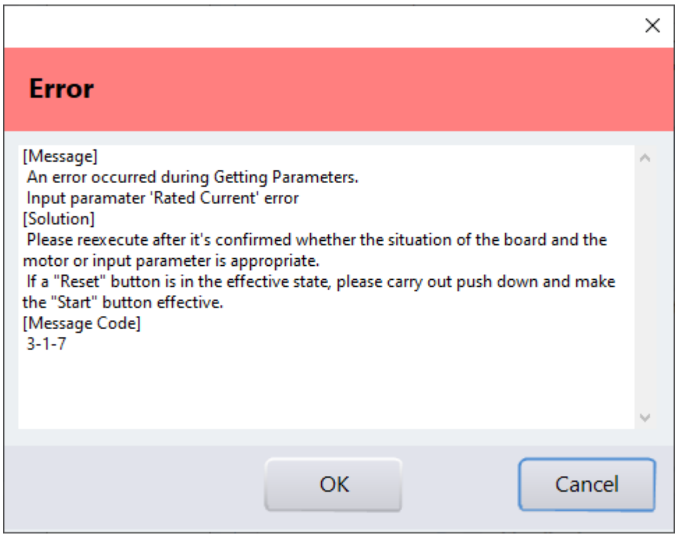


図 14-4　エラーメッセージ例

14.4.2.3 チューニング結果の確認

Tuning Window の「Report」ボタンをクリックして、Result Report 画面でチューニングの詳細を確認します。

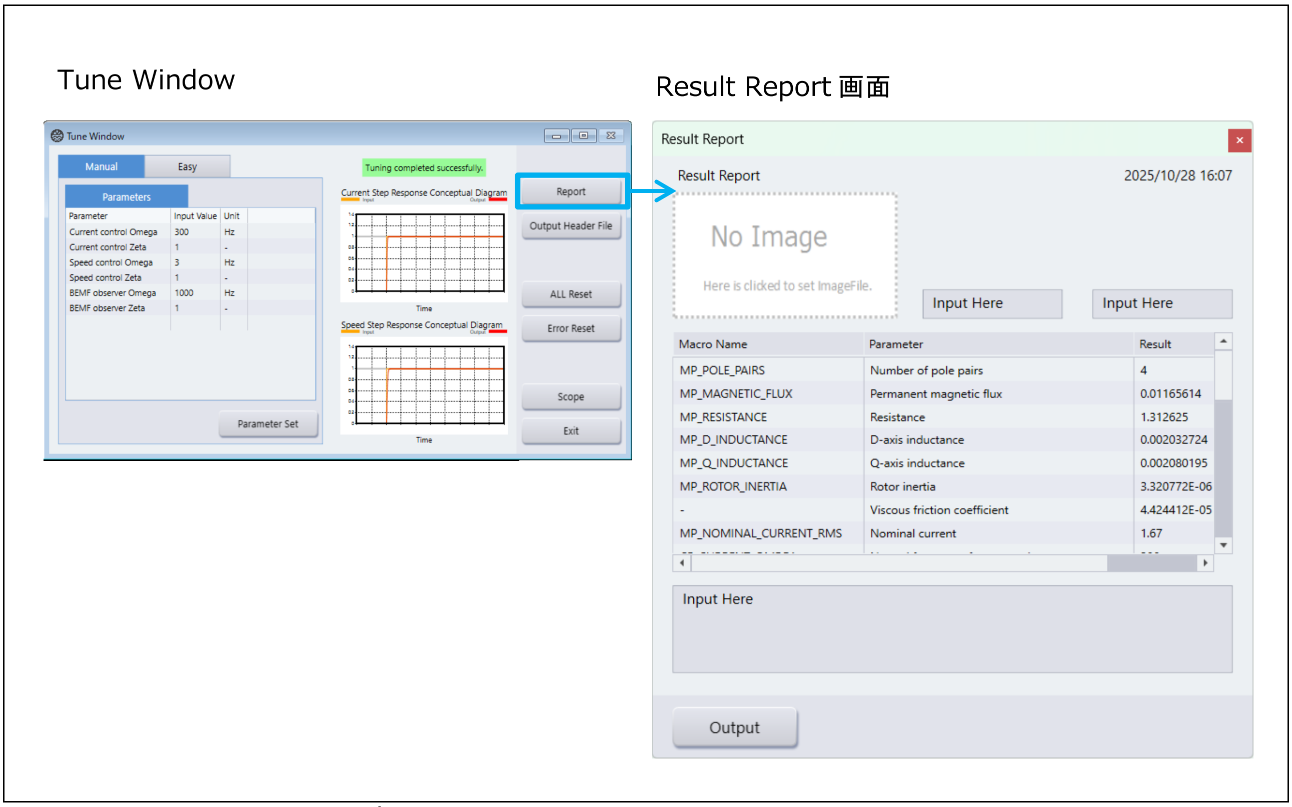


図 14-5 Result Report 画面の表示

14.4.2.4 Tuning の終了

Tuning Window の「Exit」ボタンをクリックすると、Parameter Input 画面に戻ります。

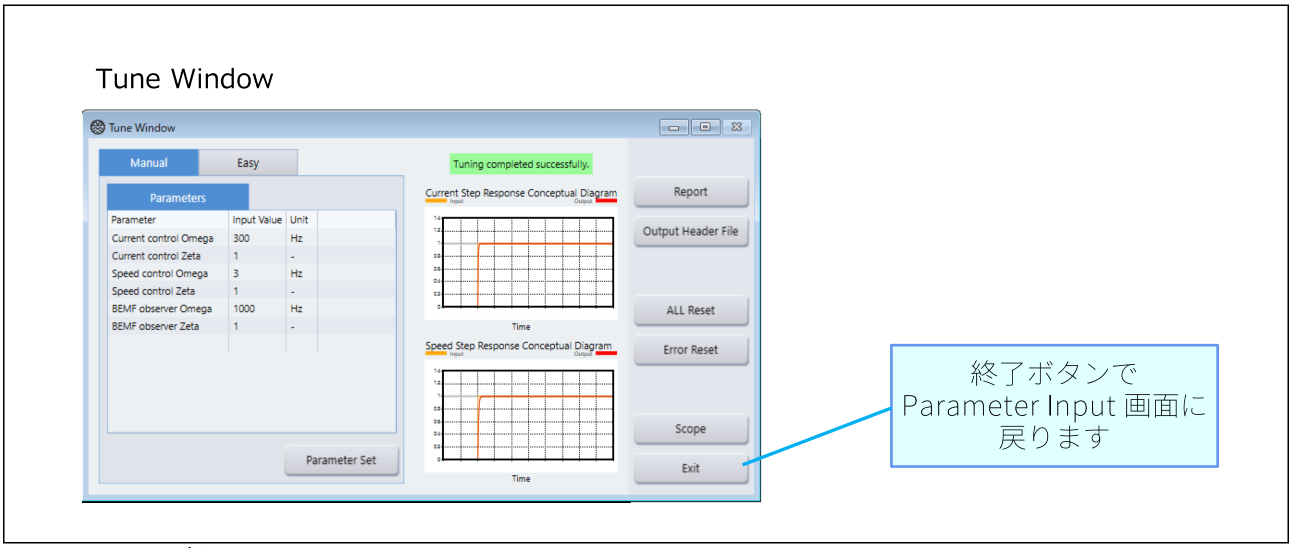


図 14-6 Exit ボタン

14.4.3 動作の確認

14.4.3.1 【エンコーダ用プログラムのみ】 1 回転あたりのエンコーダパルス数の入力

Tune Window で 1 回転あたりのエンコーダパルス数を入力し、「Parameter SET」ボタンをクリックしてください。

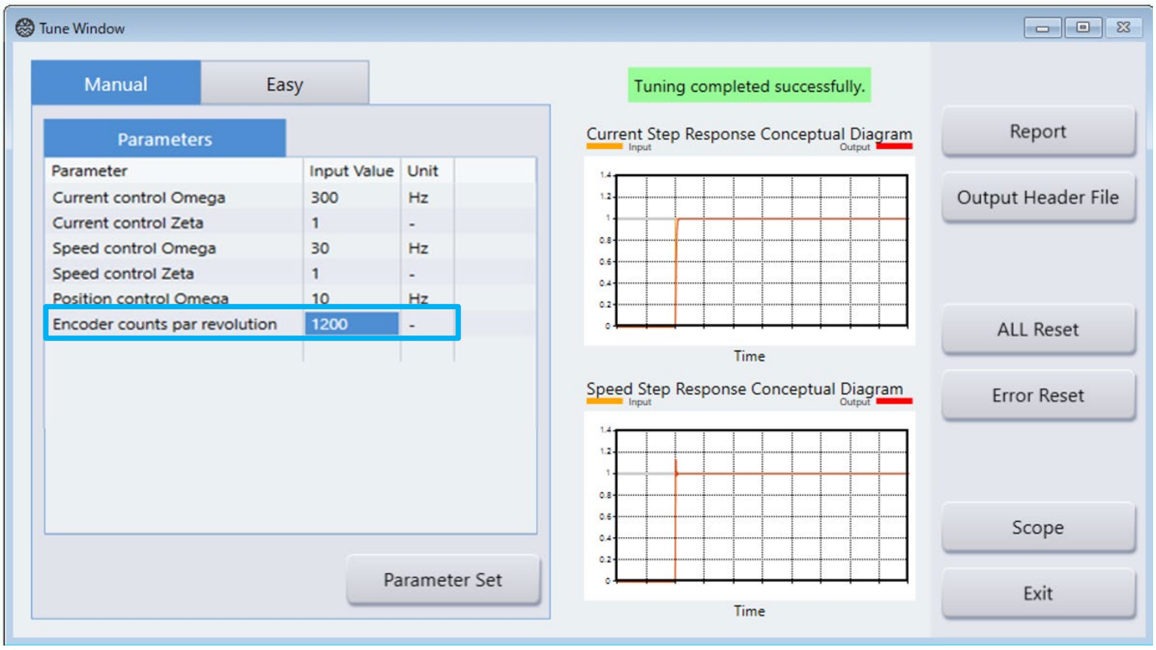


図 14-7 エンコーダパルス数の入力（**エンコーダ用プログラムのみ**）

14.4.3.2　　モータの駆動テスト

Scope 画面の Drive Test でモータを駆動します。

Tune Window の「Scope」ボタンをクリックすると、Scope 画面が表示されます。Control 部の Operation で「Drive Test」を選択し、「RUN」ボタンをクリックすると、モータが速度制御モードで駆動されます。

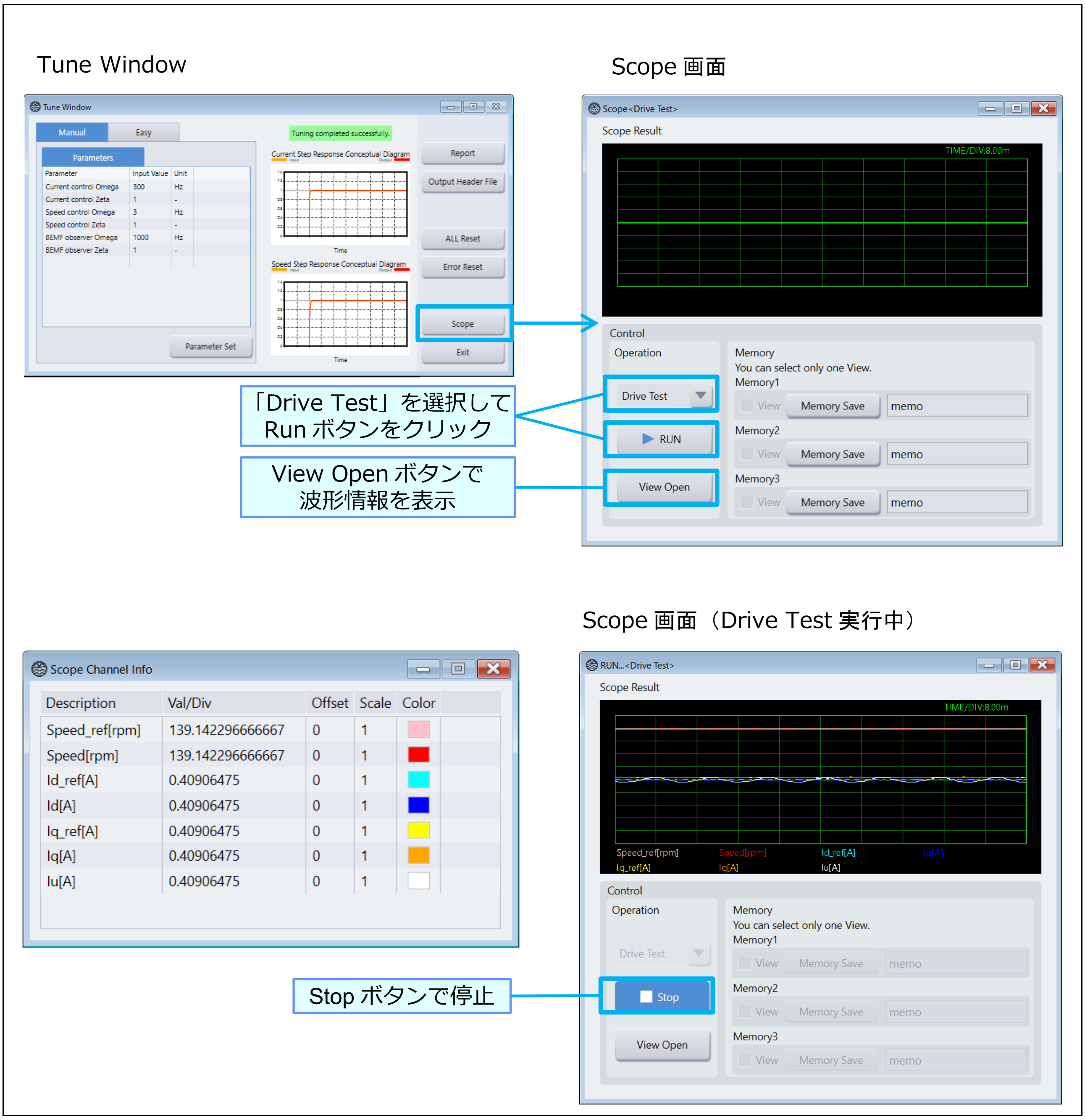


図 14-8　Drive Test 画面

14.4.3.3 その他の駆動テスト

Control 部の Operation で、エンコーダ位置制御では Position Control、センサレス速度制御では Starting を選択することができます。

Position Control では、位置制御モードでの動作を確認することができます。

Starting では、センサレスベクトル制御の始動時の動作を確認することができます。

注意 (2) と (3) で波形は自動で表示されますが、接続されているモータによっては波形が完全に表示されない場合があります。

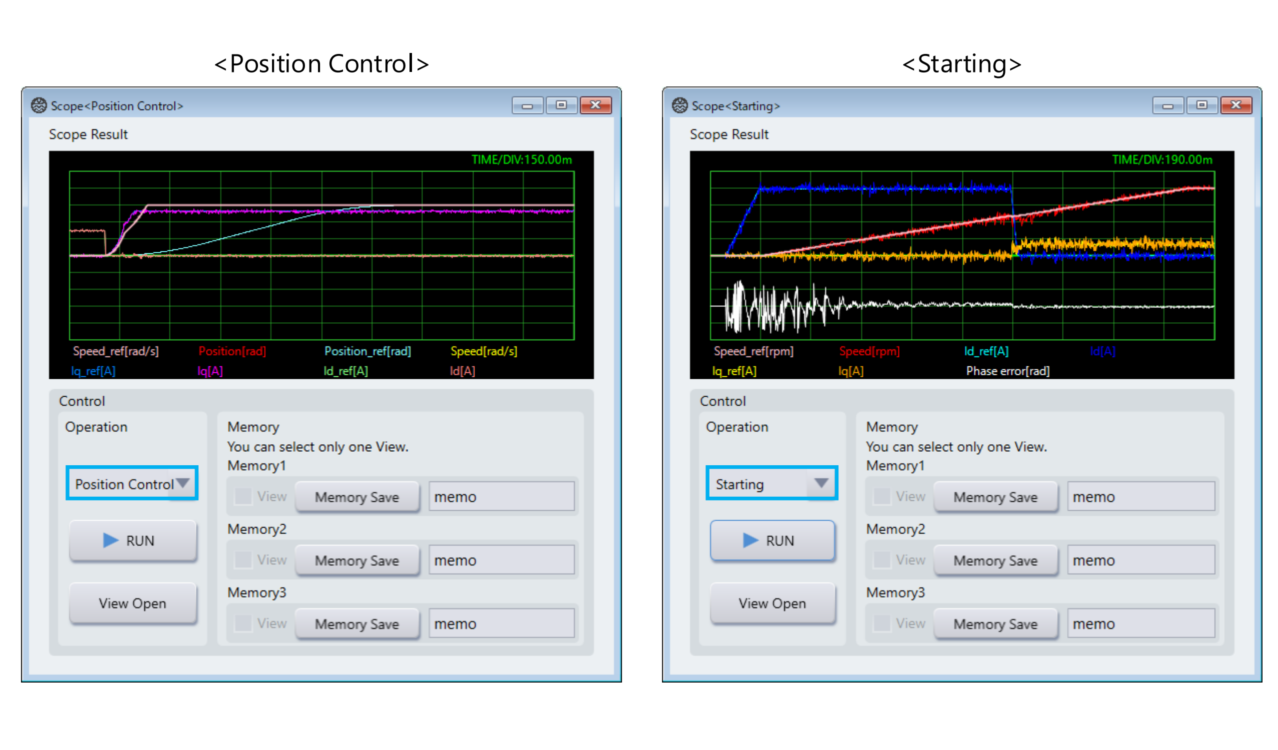


図 14-9 Position Control / Starting

14.4.4 Tuner ツールの終了

モータをテスト駆動している場合は、「STOP」ボタンでテストを終了します。Tune Window の「Exit」ボタンをクリックすると、Parameter Input 画面に戻ります。

その後「Main Window」ボタンで Main Window に戻るか、またはツール切り替えボタンの操作を行ってください。

14.5　機能説明

14.5.1　Skip 機能（Parameter Input 画面）

チューニングパラメータのうち、抵抗値、d 軸/q 軸インダクタンス、磁石磁束をチューニング前に入力することで、入力したパラメータの測定を省略できます。

Parameter Input 画面の Skip Kind 部で、省略するパラメータの Skip チェックボックスにチェックを入れてパラメータ値を入力してください。

入力が終わったら、「SET」ボタンをクリックします。

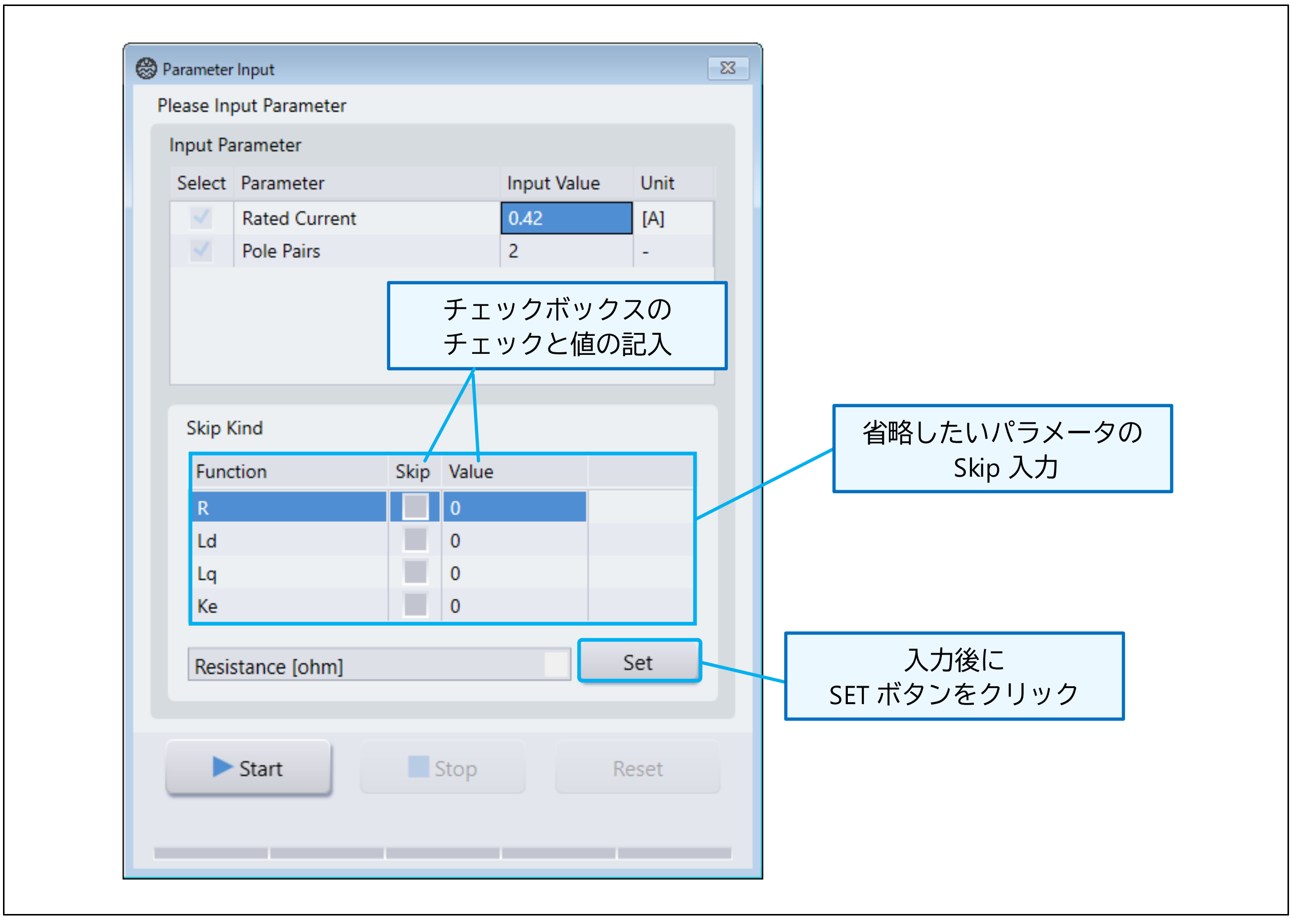


図 14-10　Skip 入力画面

14.5.2 制御パラメータの調整（Tune Window）

Tune Window は、制御パラメータの設定用 Window で、設定方法は 2 種類の方法があり、タブによって切り替えます。タブには、Manual タブと Easy タブがあります。

Manual タブでは、制御の中で使用する各フィードバックループの固有周波数と減衰係数を調整し制御に反映することができます。

Easy タブでは、初期値を 50%として、スライダを左右へ操作する事で調整し制御へ反映する事ができます。

制御へ反映するには、「Parameter SET」ボタンをクリックしてください。

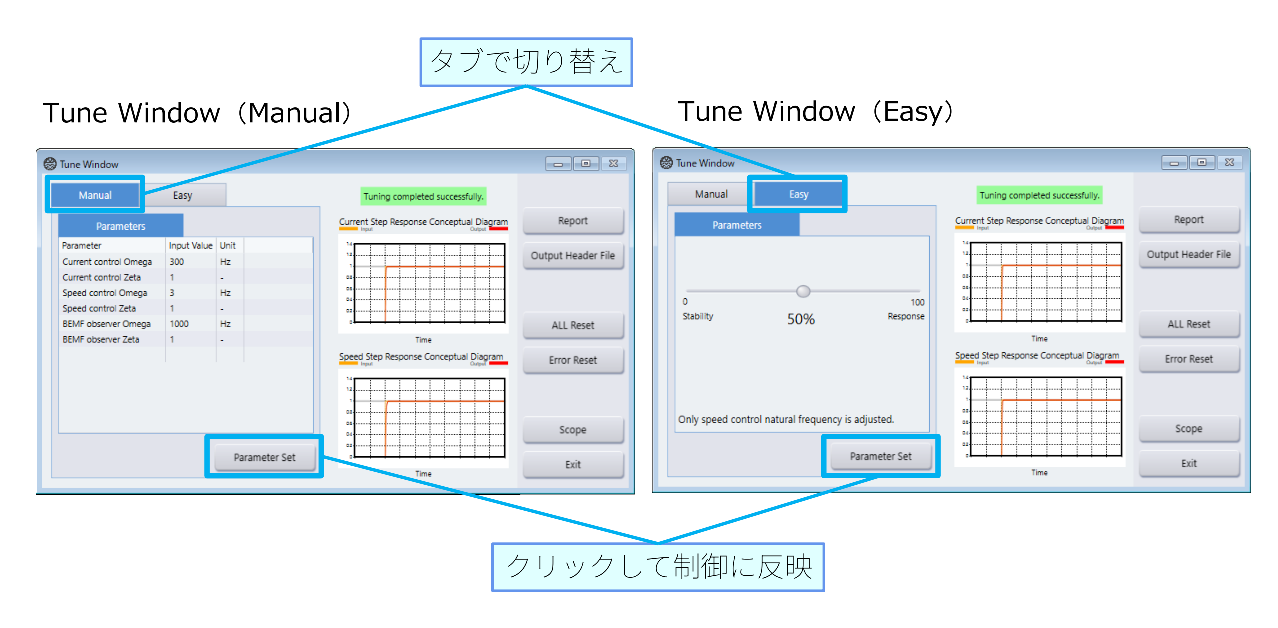


図 14-11　Manual タブと Easy タブ

表 14-1　調整パラメータ一覧（Manual タブ）

画面表示	調整パラメータ	備考
Current control Omega	電流制御系固有周波数 [Hz]	
Current control Zeta	電流制御系減衰係数	
Speed control Omega	速度制御系固有周波数 [Hz]	
Speed control Zeta	速度制御系減衰係数	
BEMF observer Omega	誘起電圧推定系固有周波数 [Hz]	センサレス速度制御のみ
BEMF observer Zeta	誘起電圧推定系減衰係数	センサレス速度制御のみ
Position control Omega	位置制御系固有周波数 [Hz]	エンコーダ位置制御のみ
Encoder counts per revolution	1 回転あたりエンコーダパルス数	エンコーダ位置制御のみ

14.5.2.1 ステップ応答イメージ図

Manual タブと Easy タブで設定された制御パラメータ値に応じて、ステップ応答のイメージを描画します。

Current control Omega および Speed control Zeta の値を変更した場合は Current Step Response Conceptual Diagram を、Speed control Omega および Speed control Zeta の値を変更した場合は Speed Step Response Conceptual Diagram のステップ応答イメージが変化します。

Easy タブでの設定においては、Speed Step Response Conceptual Diagram のみ変化します。

注意　ご使用のモータの実際の応答特性を示すわけではありません。

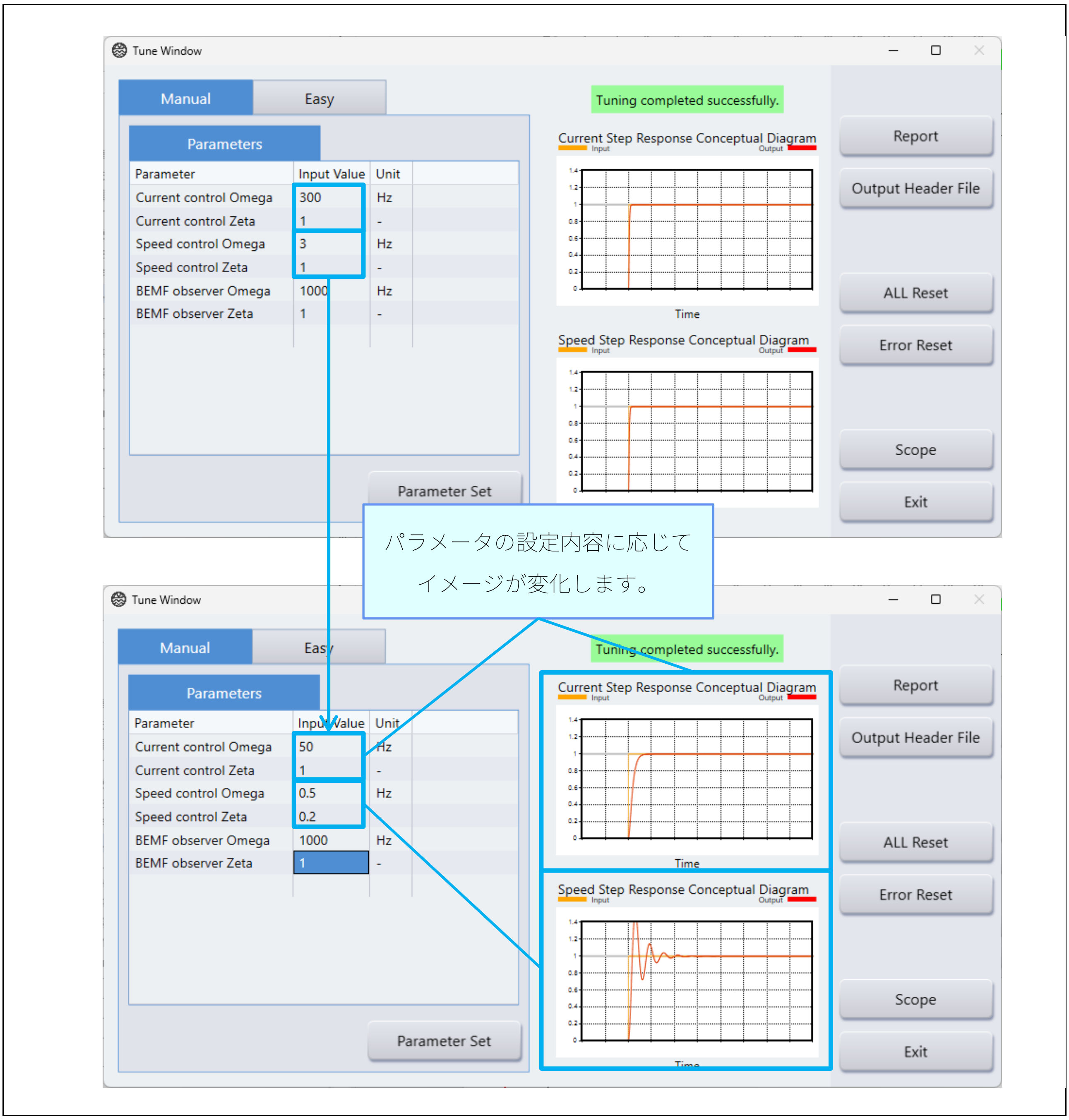


図 14-12 制御パラメータ変更によるステップイメージの変化

14.5.3 リセット機能（Tune Window）

チューニング後に駆動プログラムにエラーが発生した場合は、リセットの種類を選択して実行します。Tune Window で「ALL Reset」「Error Reset」ボタンをクリックするとリセット画面を表示できます。

- ALL Reset
  調整パラメータすべてをリセットします。Parameter Input 画面に遷移しますので、チューニングを再度実行してください。

- Error Reset
  エラー状態のみをリセットし、チューニングした調整パラメータは保持されます。

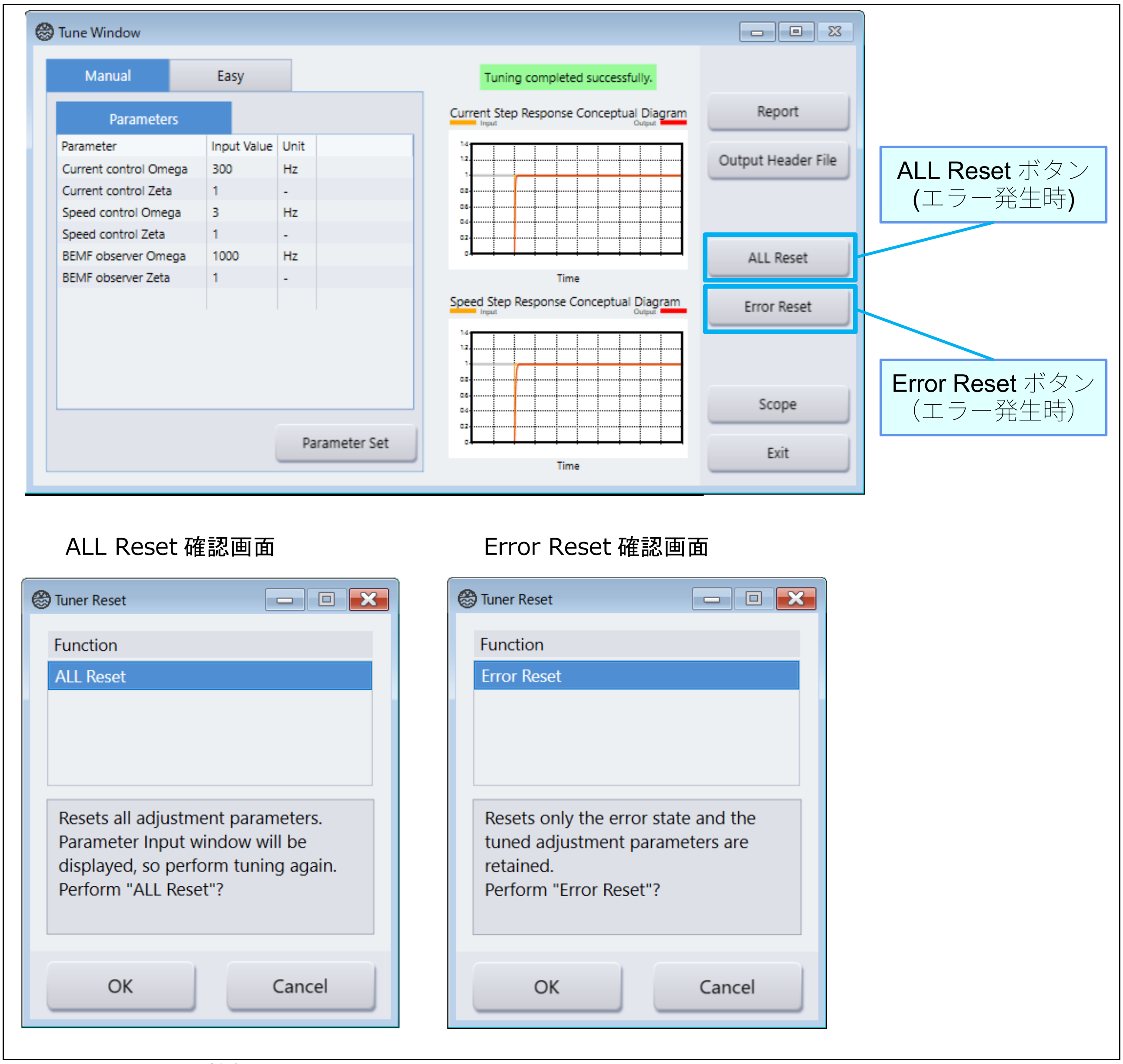


図 14-13 リセット機能

14.5.4　　Header 出力（Tune Window）

Tuning 結果をルネサス製モータ制御プログラムのヘッダファイル形式で出力することができます。

Tune Window の「Output Header File」ボタンをクリックすると、「r_mtr_control_parameter.h」と「r_mtr_motor_parameter.h」の保存画面が表示されます。


```
/**********************************************************************************
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only
* intended for use with Renesas products. No other uses are authorized. This
* software is owned by Renesas Electronics Corporation and is protected under
* all applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
* THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
* LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
* ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
* BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software
* and to discontinue the availability of this software. By using this software,
* you agree to the additional terms and conditions found by accessing the
* following link:
* http://www.renesas.com/disclaimer
*
* Copyright (C) 2020 Renesas Electronics Corporation. All rights reserved.
**********************************************************************************/

/*********************************************************************************
* File Name   : r_mtr_motor_parameter.h
* Description : Definition of the target motor parameters
*                  (Renesas Motor Workbench Output file)
*********************************************************************************/
/*********************************************************************************
* Date : 2025.10.29 09:59
*********************************************************************************/

/* Guard against multiple inclusion */
#ifndef R_MTR_MOTOR_PARAMETER_H
#define R_MTR_MOTOR_PARAMETER_H

/*********************************************************************************
* Macro definitions
*********************************************************************************/
#define     MTR_MOTOR_PARAMETER     (1)

/* Target motor definitions */
#define     MP_POLE_PAIRS           (4)                  /* Number of pole pairs */
#define     MP_RESISTANCE           (1.278677f)          /* Resistance [ohm] */
#define     MP_D_INDUCTANCE         (0.002035681f)       /* D-axis inductance [H] */
#define     MP_Q_INDUCTANCE         (0.002070462f)       /* Q-axis inductance [H] */
#define     MP_MAGNETIC_FLUX        (0.01169281f)        /* Permanent magnetic flux [Wb] */
#define     MP_ROTOR_INERTIA        (3.310325E-06f)      /* Rotor inertia [kgm^2] */
#define     MP_NOMINAL_CURRENT_RMS  (1.67f)              /* Nominal current [Arms] */

#endif /* R_MTR_MOTOR_PARAMETER_H */
```

図 14-14　Output Header File ボタン

エンコーダ位置制御プログラムでは、サンプルプログラムに組み込む際、アプリケーションノートを参照してエンコーダパルス数の設定を別途行ってください。

14.5.5 PDF 出力（Tune Window）

Tune Window の「Report」ボタンをクリックすると、Result Report 画面が表示されます。この画面の「Output」ボタンで、チューニング結果を PDF ファイルとして出力できます。

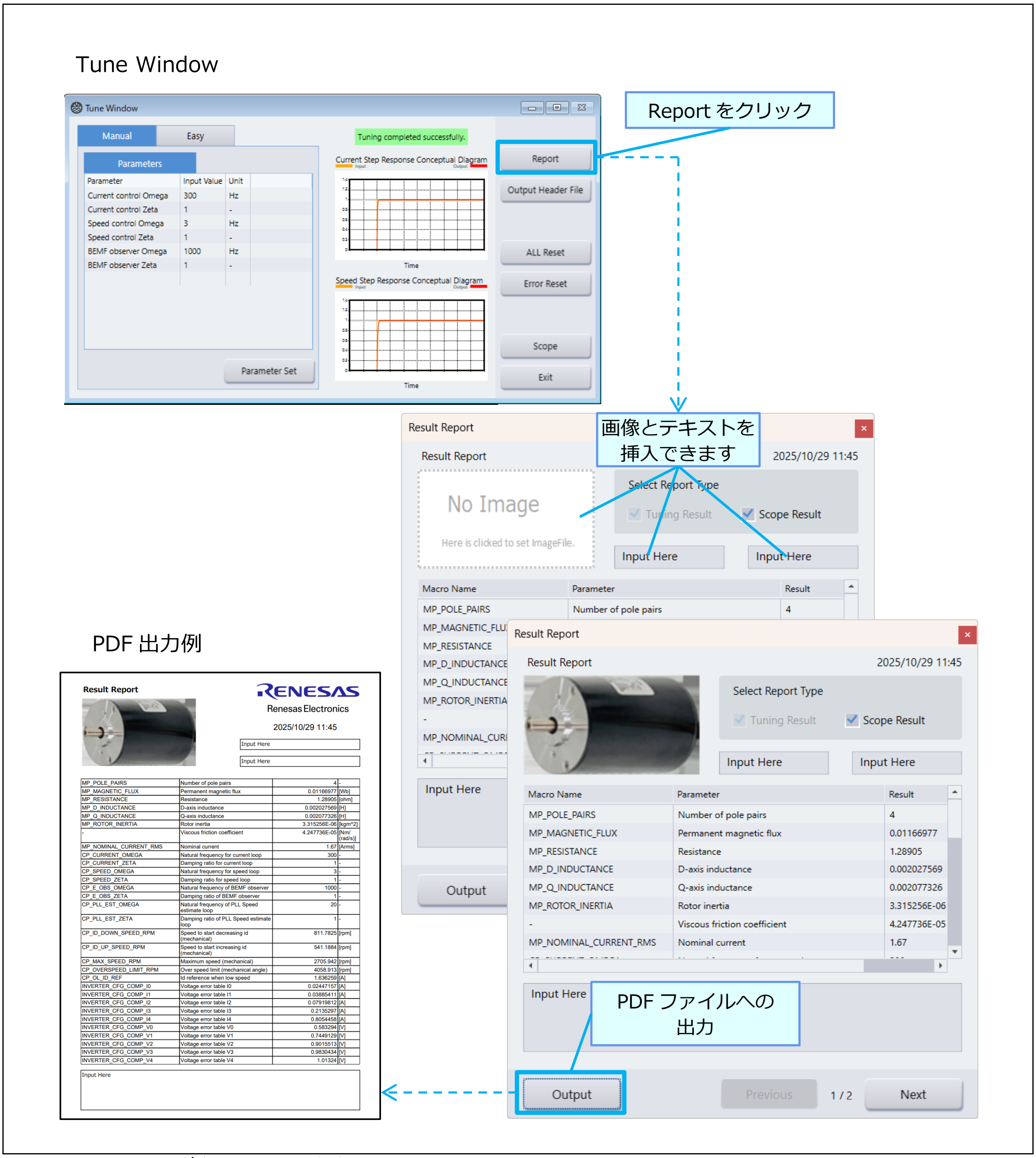


図 14-15 Output ボタン（PDF 出力）

チューニング結果として出力されるパラメーター覧を以下に示します。

表 14-2　出力パラメーター覧

マクロ名	概要	単位
MP_POLE_PAIRS	極対数	-
MP_MAGNETIC_FLUX	磁束	[wb]
MP_RESISTANCE	抵抗	[ohm]
MP_D_INDUCTANCE	d 軸のインダクタンス	[H]
MP_Q_INDUCTANCE	q 軸のインダクタンス	[H]
MP_ROTOR_INERTIA	ロータのイナーシャ	[kg m2]
MP_NOMINAL_CURRENT_RMS	定格電流	[A]
-	粘性摩擦係数	[Nm/(rad/s)]
CP_CURRENT_OMEGA	電流制御系固有周波数	[Hz]
CP_CURRENT_ZETA	電流制御系減衰係数	-
CP_SPEED_OMEGA	速度制御系固有周波数	[Hz]
CP_SPEED_ZETA	速度制御系減衰係数	-
CP_E_OBS_OMEGA	誘起電圧推定系固有周波数 (センサレス用プログラのみ)	[Hz]
CP_E_OBS_ZETA	誘起電圧推定系減衰係数 (センサレス用プログラのみ)	-
CP_PLL_EST_OMEGA	位置推定系固有周波数 (センサレス用プログラのみ)	[Hz]
CP_PLL_EST_ZETA	位置推定系減衰係数 (センサレス用プログラのみ)	-
CP_ID_DOWN_SPEED_RPM	d 軸電流指令値減算開始速度（機械角） (センサレス用プログラのみ)	[rpm]
CP_ID_UP_SPEED_RPM	d 軸電流指令値加算開始速度（機械角） (センサレス用プログラのみ)	[rpm]
CP_POS_OMEGA	位置制御系固有周波数 (エンコーダ用プログラのみ)	[Hz]
CP_SOB_OMEGA	速度オブザーバ固有周波数 (エンコーダ用プログラのみ)	[Hz]
CP_SOB_ZETA	速度オブザーバ減衰係数 (エンコーダ用プログラのみ)	-
CP_MIN_SPEED_RPM	最低速度 (エンコーダ用プログラのみ)	[rpm]
CP_MAX_SPEED_RPM	最大速度	[rpm]
CP_OVERSPEED_LIMIT_RPM	速度制限値	[rpm]
CP_OL_ID_REF	オープンループ時の d 軸電流指令値	[A]
INVERTER_CFG_COMP_I0	電圧誤差補償用係数	[A]
INVERTER_CFG_COMP_I1	電圧誤差補償用係数	[A]
INVERTER_CFG_COMP_I2	電圧誤差補償用係数	[A]
INVERTER_CFG_COMP_I3	電圧誤差補償用係数	[A]
INVERTER_CFG_COMP_I4	電圧誤差補償用係数	[A]
INVERTER_CFG_COMP_V0	電圧誤差補償用係数	[V]
INVERTER_CFG_COMP_V1	電圧誤差補償用係数	[V]
INVERTER_CFG_COMP_V2	電圧誤差補償用係数	[V]

INVERTER_CFG_COMP_V3	電圧誤差補償用係数	[V]
INVERTER_CFG_COMP_V4	電圧誤差補償用係数	[V]

14.5.6　波形情報の表示（Scope 画面）

Scope 画面の「View Open」ボタンをクリックすると、波形表示に関する情報を確認できます。

Scope 画面

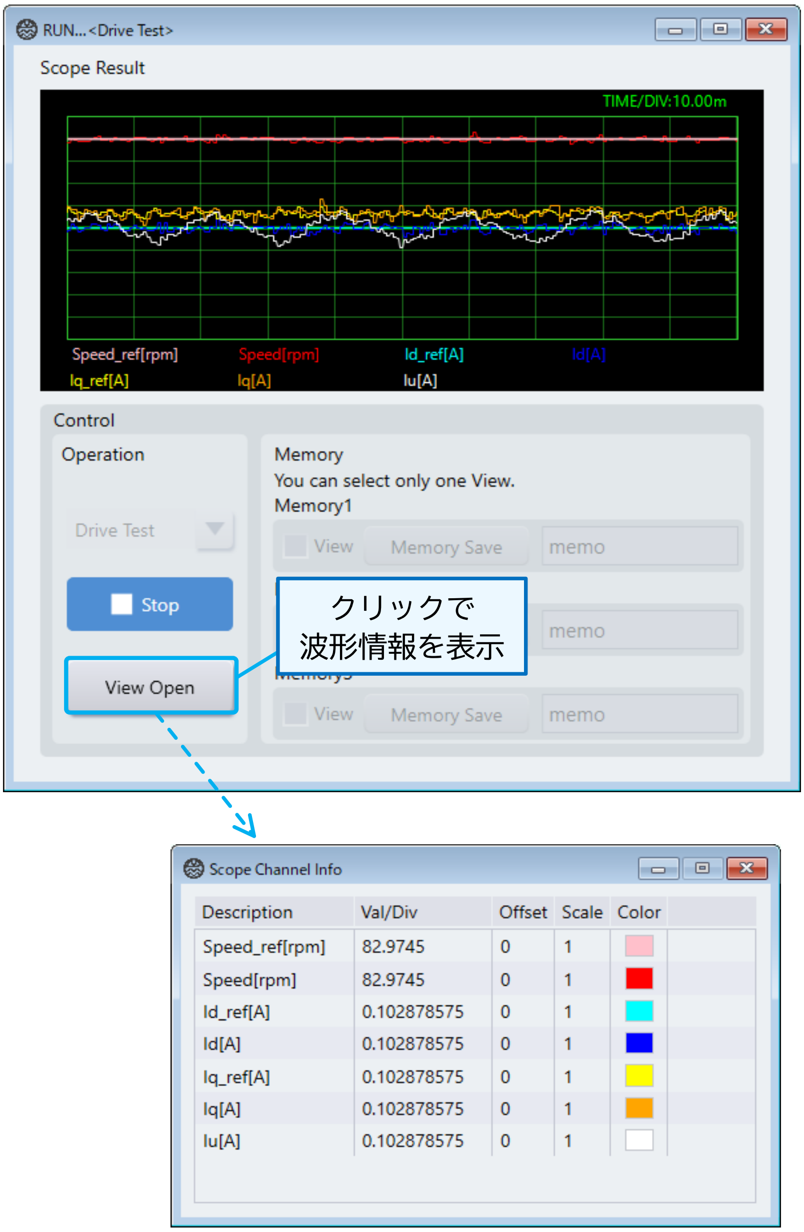


図 14-16　View Open ボタン

14.5.7 Memory 機能（Scope 画面）

Scope 画面では、モータ駆動時の波形を最大 3 つまでメモリし、各波形を切り替えて表示することができます。

「Memory Save」ボタンをクリックすると波形をメモリし、ボタンの表示は Memory Clear に変わります。再度クリックするとメモリをクリアします。

View チェックボックスにチェックをすると、メモリした波形が表示されます。

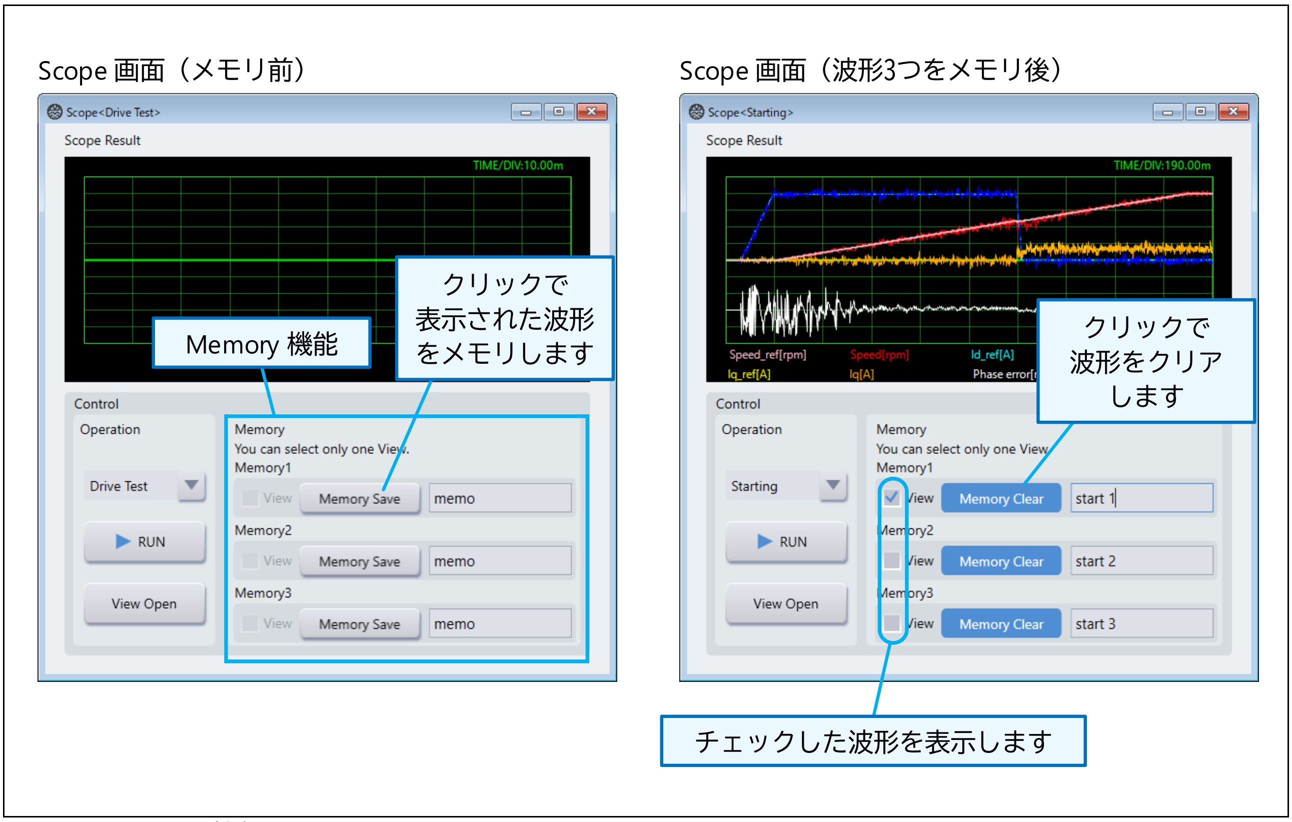


図 14-17 メモリ機能

14.6 Tuner Library 組み込み手順

14.6.1 Tuner Library 全体構成

Tuner Library は主に、「抵抗測定機能」、「磁束測定機能」、「インダクタンス測定機能」、「慣性・摩擦測定機能」の４つの機能を持っています。 Tuner Library が周辺機能にアクセスするために、ユーザ実装関数を設定する必要があります。Tuner Library は設定されたユーザ実装関数の関数ポインタから周辺機能にアクセスします。ユーザ実装関数の設定については 14.6.4.8 章を参照ください。

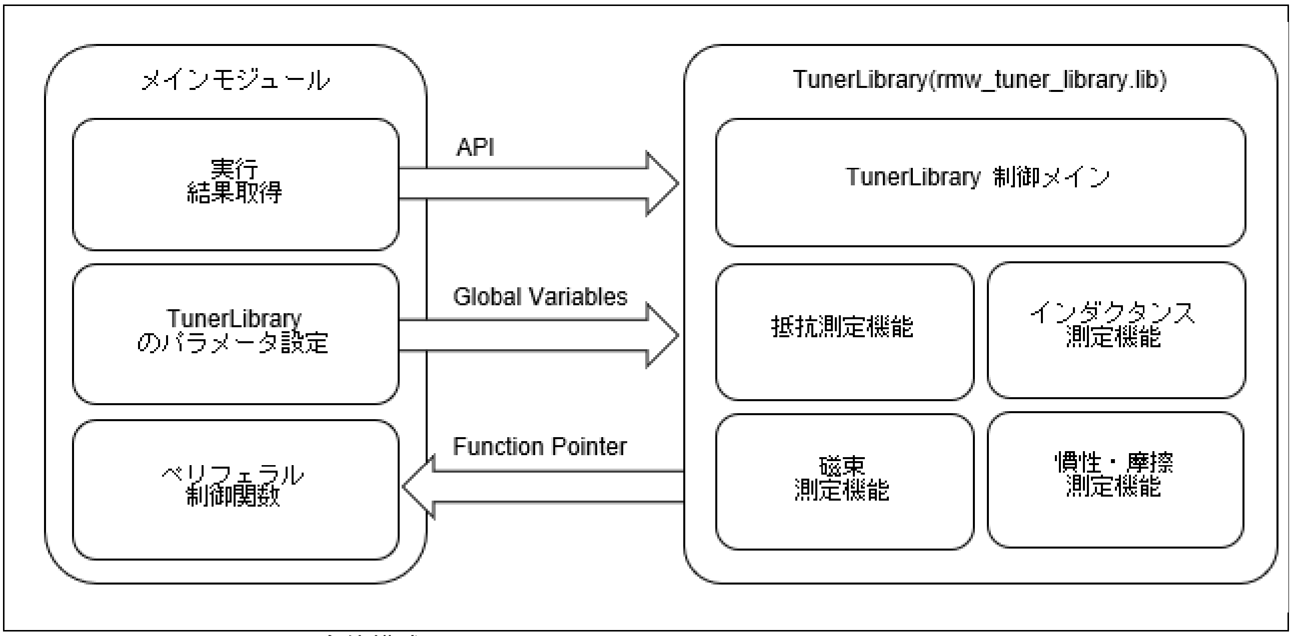


図 14-18 Tuner Library 全体構成

14.6.2 Tuner Library API 一覧

表 14-3 Tuner Library API (1/3)

機能	型	関数名	定義ファイル	概要
Tuner 処理初期化	void	R_AID_Init(uint8_t pwm_tick_per_irq, float speed_ctrl_period)	r_aid_auto_identify.h	Tuner 開始前に毎回コールします。
Tuner 処理開始	void	R_AID_CmdStart(void)	r_aid_auto_identify.h	Tuner 開始時にコールします。
Tuner 処理停止	void	R_AID_CmdStop(void)	r_aid_auto_identify.h	Tuner 中に停止させる際にコールします。
Tuner 処理リセット 必要変数初期化	void	R_AID_CmdReset(void)	r_aid_auto_identify.h	エラーが発生した際、エラー状態を解除する際にコールします。
Tuner 処理再起動 前回中断時のみ有効	void	R_AID_CmdResume(void)	r_aid_auto_identify.h	前回 Tuner 処理を中断している場合、再開させる際にコールします。
内部ステートマシンイベント指定実行	void	R_AID_CmdByCode(uint16_t cmd_code)	r_aid_auto_identify.h	ステートマシンイベントを引数で指定して実行する際にコールします。 AID_COMMAND_NONE (0) status that no command is issued AID_COMMAND_START (1) command code that starts tuning AID_COMMAND_STOP (2) command code that stops tuning AID_COMMAND_RESET (3) reset command code AID_COMMAND_RESUME (4) resume command code
エラー発生時呼び出し 上位エラーを Tuner 処理に設定する。	void	R_AID_UserError(uint16_t u2_error_code)	r_aid_auto_identify.h	強制的にエラー状態にする場合にコールします。 引数には任意のエラーコードを設定できます。 TunerLib で定義しているのは以下の 4 つです。 AID_ERROR_INPUT_CURRENT (0x1001) AID_ERROR_INPUT_POLEPAIR (0x1002) AID_ERROR_INPUT_VOLTERR_STEP (0x1003) AID_ERROR_INPUT_INERTIA_RANGE (0x1004)
基準電流値、及びモータ極対数設定	void	R_AID_ConfigMotorPlate(float f4_rated_current, uint16_t u2_num_of_pole_pair)	r_aid_auto_identify.h	Tuner 用の入力値、定格電流と極対数を引数に設定してコールします。 Tuner 開始前にコールします。

表 14-4 Tuner Library API (2/3)

機能	型	関数名	定義ファイル	概要
モータ電気特性設定	int32_t	R_AID_SetInitElecParams(float f4_r, float f4_ld, float f4_lq, float f4_ke)	r_aid_auto_identify.h	Tuner パラメータに既知のものがあり、そのパラメータの Tuner をスキップする場合、引数に値を設定して Tuner 開始前にコールします。
Tuner 処理バージョン取得	void	R_AID_GetVersionInfo(uint16_t *p_major_version, uint16_t *p_minor_version)	r_aid_auto_identify.h	バージョン情報を取得する場合にコールします。
電流制御周期取得 [秒]	float	R_AID_GetCurrentCtrlPeriod(void)	r_aid_auto_identify.h	設定されている電流制御周期を取得する場合にコールします。
速度制御周期取得 [秒]	float	R_AID_GetSpeedCtrlPeriod(void)	r_aid_auto_identify.h	設定されている速度制御周期を取得する場合にコールします。
PWM キャリア周期取得 [秒]	float	R_AID_GetPWMPeriod (void)	r_aid_auto_identify.h	設定されている PWM キャリア周期を取得する場合にコールします。
Tuner 処理内部ステート取得	uint16_t	R_AID_GetSystemStatus(void)	r_aid_auto_identify.h	status 情報を取得する場合にコールします。 AID_STATUS_READY (0) READY status code AID_STATUS_MEASURE (1) MEASURE status code AID_STATUS_ERROR (2) ERROR status code AID_STATUS_RESET (3) RESET status code AID_STATUS_COMPLETED (4) COMPLETED status code
エラー情報取得	uint16_t	R_AID_GetErrorStatus(void)	r_aid_auto_identify.h	エラー情報を取得する場合にコールします。取得する値は R_AID_UserError で設定した値、または TunerLib のエラーコードです。 AID_ERROR_INPUT_CURRENT (0x1001) AID_ERROR_INPUT_POLEPAIR (0x1002) AID_ERROR_INPUT_VOLTERR_STEP (0x1003) AID_ERROR_INPUT_INERTIA_RANGE (0x1004)
Tuner 処理進捗状況取得	float	R_AID_GetProgress(void)	r_aid_auto_identify.h	Tuner 処理の進捗状況を取得する場合にコールします。
モータ抵抗値取得 [Ω]	float	R_AID_GetResistance(void)	r_aid_auto_identify.h	Tuner 完了後、抵抗値を取得する場合にコールします。
d 軸インダクタンス取得 [H]	float	R_AID_GetLd(void)	r_aid_auto_identify.h	Tuner 完了後、d 軸インダクタンス値を取得する場合にコールします。

表 14-5 Tuner Library API (3/3)

機能	型	関数名	定義ファイル	概要
q 軸インダクタンス取得 [H]	float	R_AID_GetLq(void)	r_aid_auto_identify.h	Tuner 完了後、q 軸インダクタンス値を取得する場合にコールします。
磁束密度取得 [Wb]	float	R_AID_GetKe(void)	r_aid_auto_identify.h	Tuner 完了後、磁束密度値を取得する場合にコールします。
イナーシャ取得 [kgm^2]	float	R_AID_GetInertia(void)	r_aid_auto_identify.h	Tuner 完了後、イナーシャ値を取得する場合にコールします。
摩擦係数取得 [Nm/(rad/sec)]	float	R_AID_GetFriction(void)	r_aid_auto_identify.h	Tuner 完了後、摩擦係数値を取得する場合にコールします。
Tuner 用設定値取得	void	R_AID_GetIDSetting(st_aid_id_setting_t *st_id_setting)	r_aid_auto_identify.h	設定されている各設定値を取得する場合にコールします。
電流制御周期割り込み内呼び出し用	void	R_AID_CurrentCtrlISR(void)	r_aid_auto_identify.h	電流制御周期割り込みでコールします。 TunerLib の電流制御処理を実行します。
速度制御周期割り込み内呼び出し用	void	R_AID_SpeedCtrlISR(void)	r_aid_auto_identify.h	速度制御周期割り込みでコールします。 TunerLib の速度制御処理を実行します。

表 14-6 Tuner Library 変数

型	変数名	定義ファイル	初期値	概要
VOID_FUNC	g_fp_aid_internal_clear_oc_flag	r_aid_config.h	Tuner Library から周辺機能にアクセスするためのユーザ実装関数の関数ポインタ	過電流状態解除処理
MTR_ID_FUNC	g_fp_aid_internal_ctrl_start			PWM 出力開始処理
MTR_ID_FUNC	g_fp_aid_internal_ctrl_stop			PWM 出力停止処理
MTR_GET_VDC_FUNC	g_fp_aid_internal_get_vdc			母線電圧値取得処理
MTR_GET_CURRENT_IUIW_FUNC	g_fp_aid_internal_get_current_iuiw			U/W 相電流値取得処理
MTR_INV_SET_UV_FUNC	g_fp_aid_internal_inv_set_uvw			PWM デューティ設定処理

14.6.3 Tuner Library マクロ一覧

表 14-7 define 定義

定義名	値	ファイル名	概要
AID_API_MAJOR_VERSION	(1)	r_aid_auto_identify.h	API のメジャーバージョンを定義
AID_API_MINOR_VERSION	(1)		API のマイナーバージョンを定義
AID_VOLTERR_TABLE_SIZE	(5)		テーブルのデータサイズ
AID_STATUS_READY	(0)		READY ステータスコードを定義
AID_STATUS_MEASURE	(1)		MEASURE ステータスコードを定義
AID_STATUS_ERROR	(2)		エラー（ERROR）ステータスコードの定義
AID_STATUS_RESET	(3)		RESET ステータスコードの定義
AID_STATUS_COMPLETED	(4)		COMPLETED ステータスコードの定義
AID_PARAMODE_INIT	(0)		初期識別モードコードの定義
AID_PARAMODE_R_DIFF	(1)		RDIFF 識別モードコードを定義
AID_PARAMODE_RLD_RLS	(2)		RLS 法 Ld 識別モードコードを定義
AID_PARAMODE_RLD_DFT	(3)		DFT 方式 Ld 識別モードコードを定義
AID_PARAMODE_LQ_RLS	(4)		RLS 法の Lq 識別モードコードを定義
AID_PARAMODE_LQ_DFT	(5)		DFT 法 Lq 同定モードコードを定義
AID_PARAMODE_KE	(6)		定格磁束同定モードコードを定義
AID_PARAMODE_JD	(7)		イナーシャ同定モードコードを定義
AID_PARAMODE_END	(8)		終端識別モードコードを定義
AID_PARAMODE_VOLTERR	(9)		電圧誤差測定モードコードを定義
AID_ERROR_INPUT_CURRENT	(0x1001)		(4097) 定格電流入力値無効
AID_ERROR_INPUT_POLEPAIR	(0x1002)		(4098) 入力極数異常
AID_ERROR_INPUT_VOLTERR_STEP	(0x1003)		(4099) 電圧異常電流ステップ入力値無効
AID_ERROR_INPUT_INERTIA_RANGE	(0x1004)		(4100) 慣性範囲入力無効
AID_COMMAND_NONE	(0U)		コマンドを発行しないステータスを定義
AID_COMMAND_START	(1U)		チューニングを開始するコマンドコードを定義
AID_COMMAND_STOP	(2U)		同調停止命令コード
AID_COMMAND_RESET	(3U)		リセット命令コード
AID_COMMAND_RESUME	(4U)		リジュームコマンドコード
AID_FAULT_PARAM_R	(-1)		パラメータ R が無効であるフォルトコードを定義
AID_FAULT_PARAM_LD	(-2)		パラメータ Ld が無効であるというフォルトコードを定義
AID_FAULT_PARAM_LQ	(-3)		パラメータ Lq が無効であるフォルトコードを定義
AID_FAULT_PARAM_KE	(-4)		パラメータ Ke が無効であるというフォルトコードを定義
AID_API_MAJOR_VERSION	(1)		API のメジャーバージョンを定義

表 14-8　構造体

型	定義名	ファイル名	概要
struct	typedef struct { float　f4_rated_current; uint16_t　u2_num_pole_pairs ; uint8_t　u1_volterr_is_enabled; uint16_t　u2_volterr_crnt_step_lsb; float　f4_inertia_range; float　f4_assumed_inertia; } st_aid_id_setting_t;	r_aid_auto_identify.h	定格電流[A] 極対の数 電圧誤差測定は有効か 1=有効, 0=無効 電圧誤差測定の電流ステップ イナーシャの範囲, 0(無負荷) ~ 1(重いイナーシャ) 速度制御器のゲイン設計に使用するイナーシャ

表 14-9　関数ポインタ

型	定義名	ファイル名	概要	備考
void	(*VOID_FUNC)(void)	r_aid_config.h	関数ポインタ	typedef void (*VOID_FUNC)(void)
void	(*MTR_ID_FUNC)(void)		関数ポインタ	typedef void (*MTR_ID_FUNC)(void)
float	(*MTR_GET_VDC_FUNC)(void)		関数ポインタ	typedef float (*MTR_GET_VDC_FUNC)(void)
void	(*MTR_GET_CURRENT_IUIW_FUNC)(float *f4_iu_ad, float *f4_iw_ad)		関数ポインタ	typedef void (*MTR_GET_CURRENT_IUIW_FUNC)(float *f4_iu_ad, float *f4_iw_ad)
void	(*MTR_INV_SET_UV_FUNC)(float f4_duty_u, float f4_duty_v, float f4_duty_w)		関数ポインタ	typedef void (*MTR_INV_SET_UV_FUNC)(float f4_duty_u, float f4_duty_v, float f4_duty_w)

14.6.4 Tuner Library 組み込み手順

RX マイコンの例として、サンプルプコード RX26T_MCBA_MCILV1_SPM_LESS_FOC_E2S_V110（Tuner Library 無し）、

RA マイコンの例として、サンプルプコード RA6T2_MCILV1_SPM_LESS_FOC_E2S_V110（Tuner Library 無し）に Tuner Library を組み込む場合の手順を説明します。統合開発環境は e2 studio です。

14.6.4.1 組み込み手順の概要

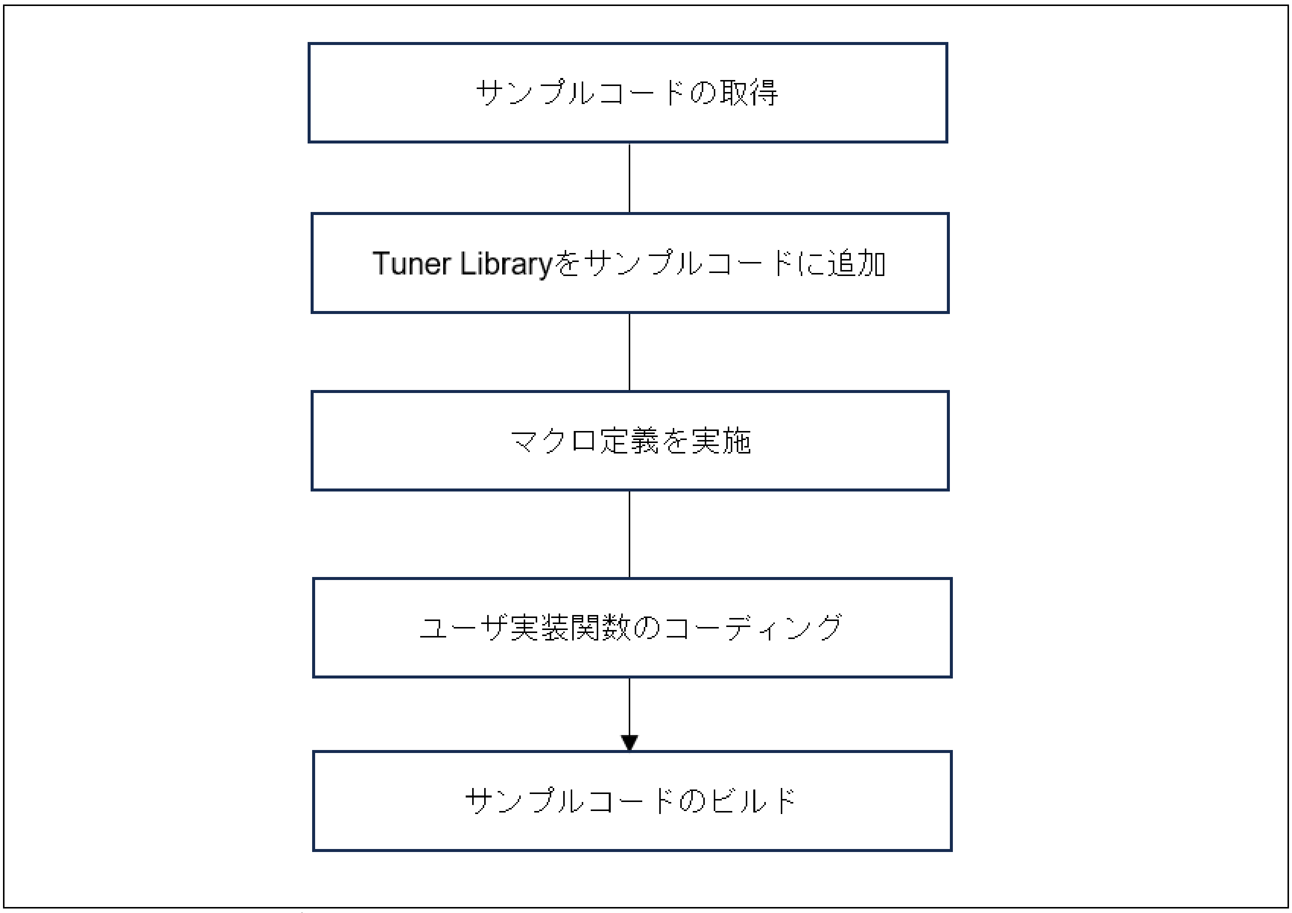


図 14-19 Tuner Library 組み込み手順

14.6.4.2　　サンプルコードの取得

下記の URL からサンプルコードを取得します。

・[RX]
https://www.renesas.com/jp/ja/document/scd/sensorless-vector-control-permanent-magnet-synchronous-motor-mck-rev110

・[RA]
https://www.renesas.com/jp/ja/document/scd/sensorless-vector-control-permanent-magnet-synchronous-motor-mckmcb-ra-family-sample-code

図 14-20　サンプルコード取得

zip ファイルを解凍すると、workspace の中に複数のサンプルコードがあります。その中の RX26T_MCBA_MCILV1_SPM_LESS_FOC_E2S_V110、RA6T2_MCILV1_SPM_LESS_FOC_E2S_V110 が今回使用するサンプルコードです。

14.6.4.3 [RX] Tuner Library をサンプルコードに追加する

Tuner 機能が組み込まれているサンプルコードから Tuner Library を追加します。
ダウンロードしたサンプルコードの RX26T_MCBA_MCILV1_SPM_LESS_FOC_TUNER_E2S_V110（Tuner 機能入り）の app¥tuner フォルダと tuner フォルダからファイルをコピーします。

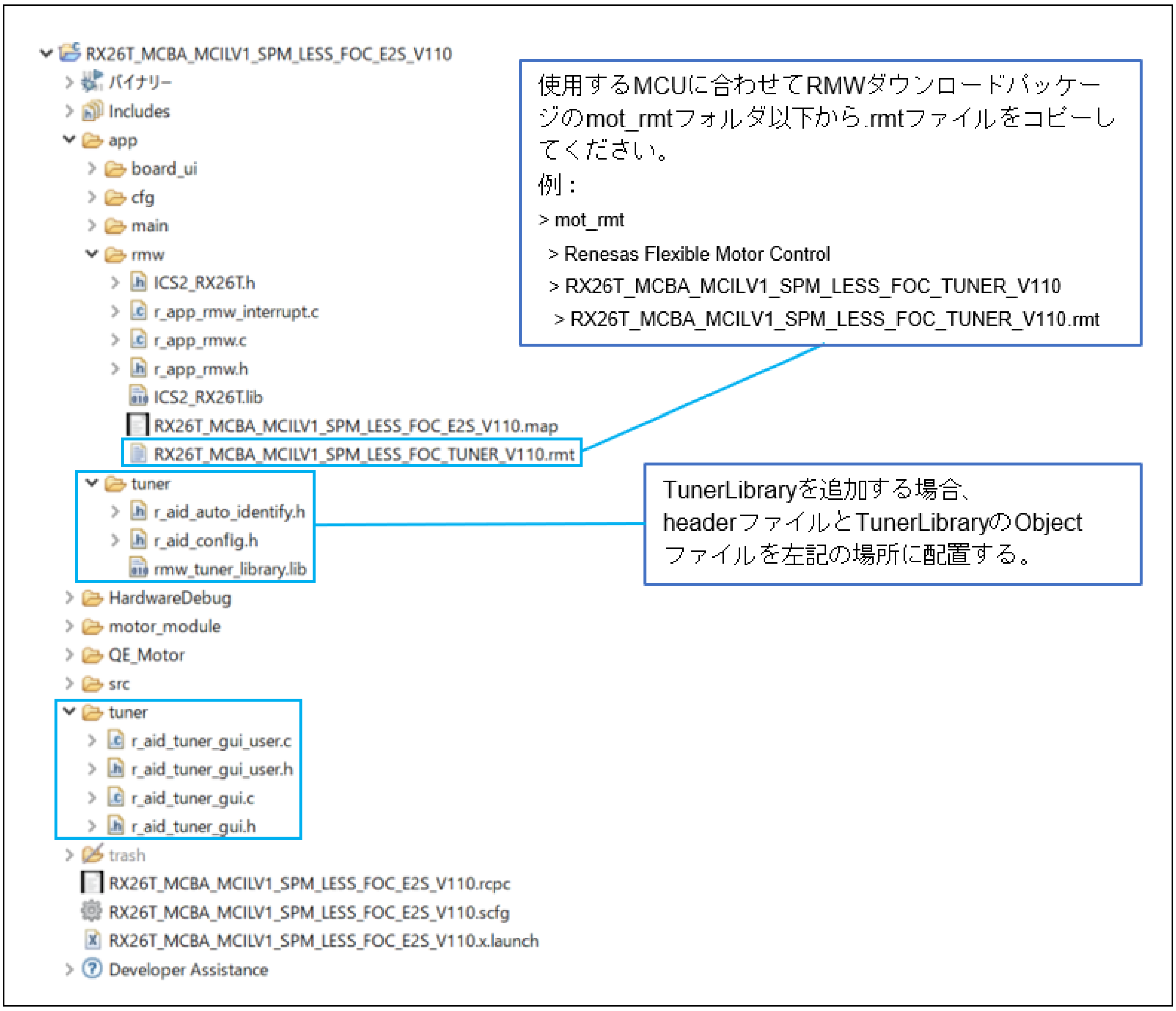


図 14-21 Tuner Library をサンプルプログラムに追加

インクルードパスを設定します。新規に追加した以下 2 つのパスを追加します。

RX26T_MCBA_MCILV1_SPM_LESS_FOC_TUNER_E2S_V110¥app¥tuner

RX26T_MCBA_MCILV1_SPM_LESS_FOC_TUNER_E2S_V110¥tuner

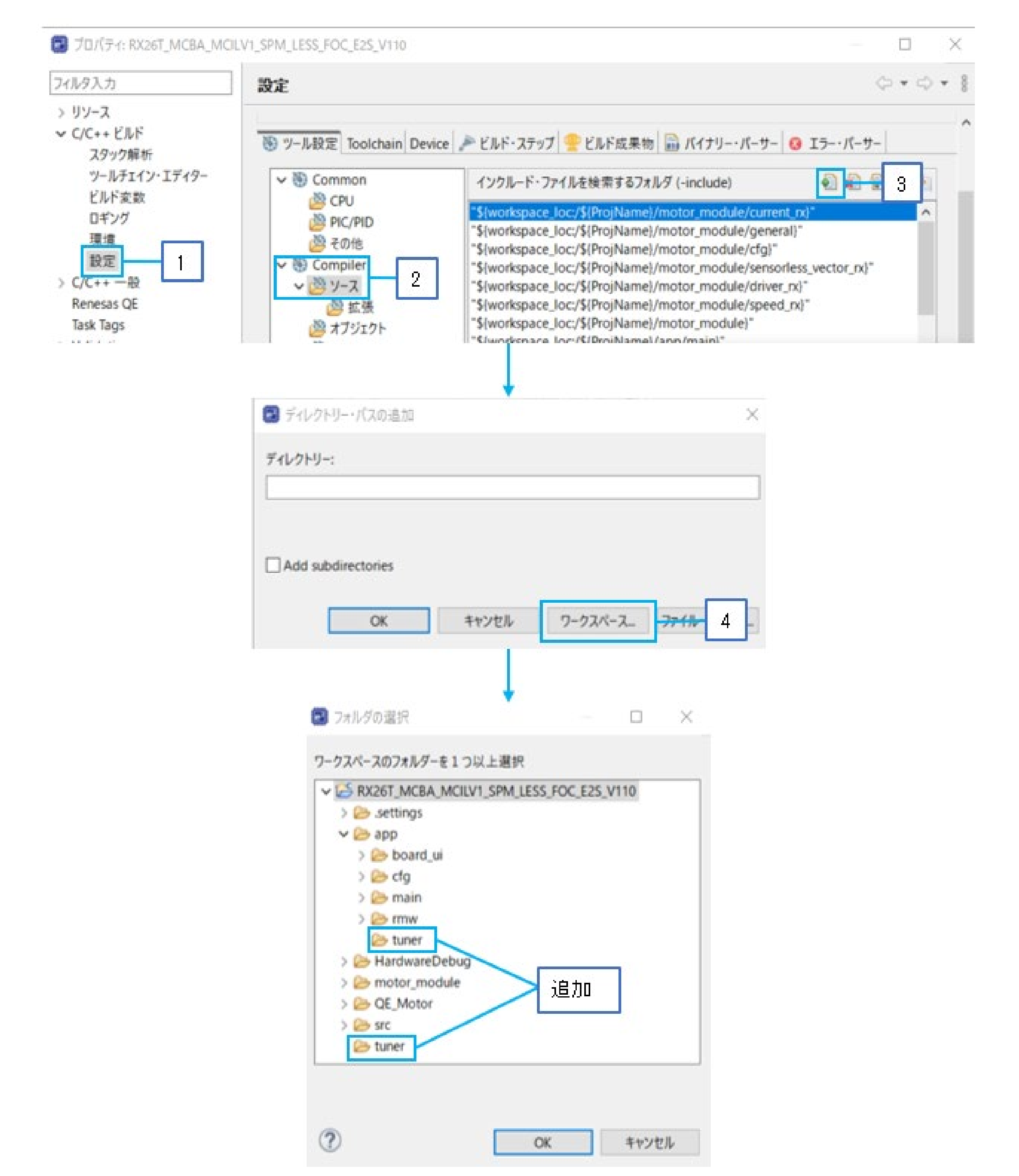


図 14-22　インクルードパスの設定

Link する Tuner Library を指定します。プロパティ画面を表示し、Tuner Library を以下の手順で追加します。

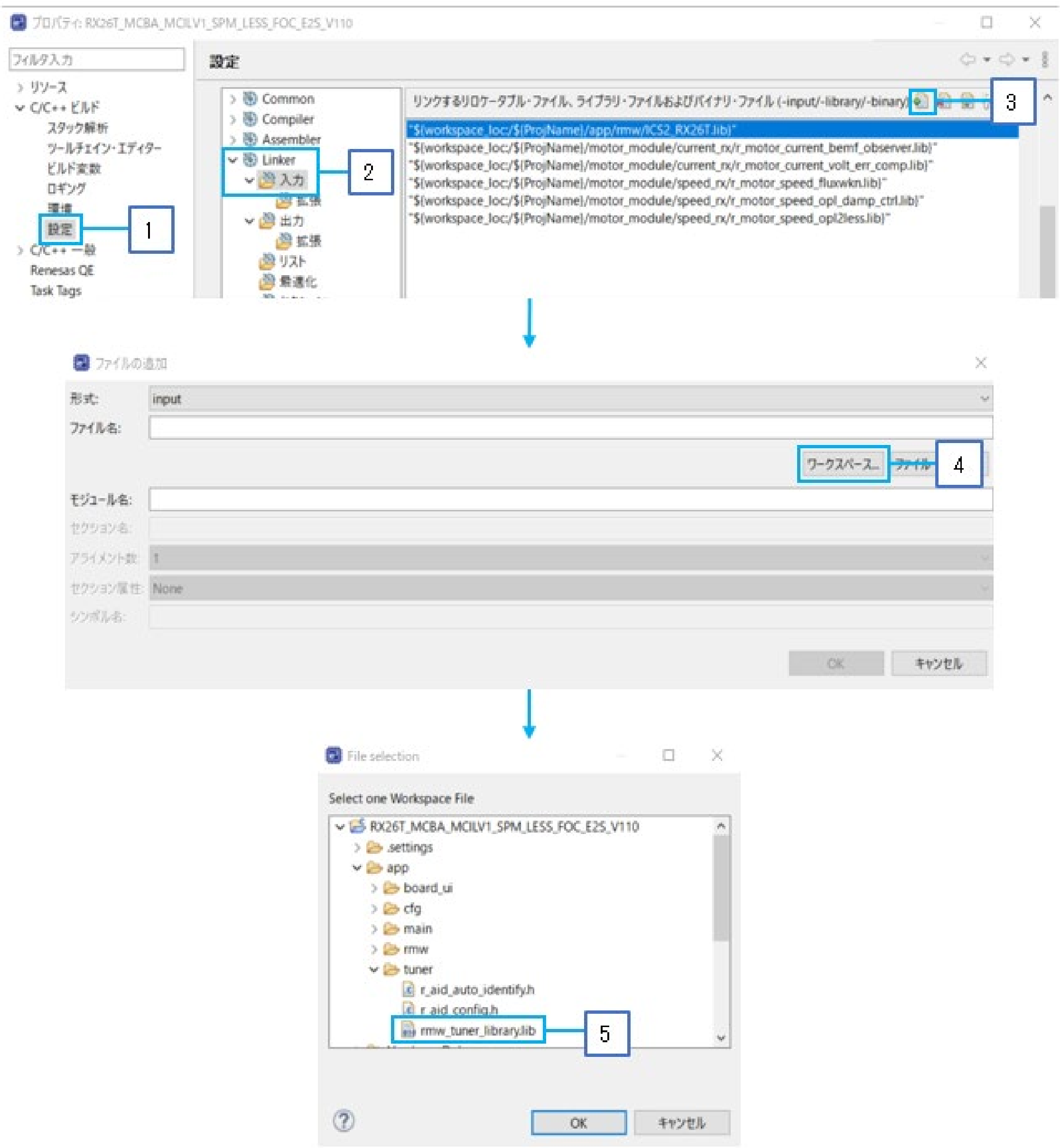


図 14-23　Tuner Library の追加

14.6.4.4 [RA] Tuner Library をサンプルコードに追加する

Tuner 機能が組み込まれているサンプルコードから Tuner Library を追加します。
ダウンロードしたサンプルコードの RA6T2_MCILV1_SPM_LESS_FOC_TUNER_E2S_V110（Tuner 機能入り）の src¥application¥identification フォルダと tuner フォルダからファイルをコピーします。

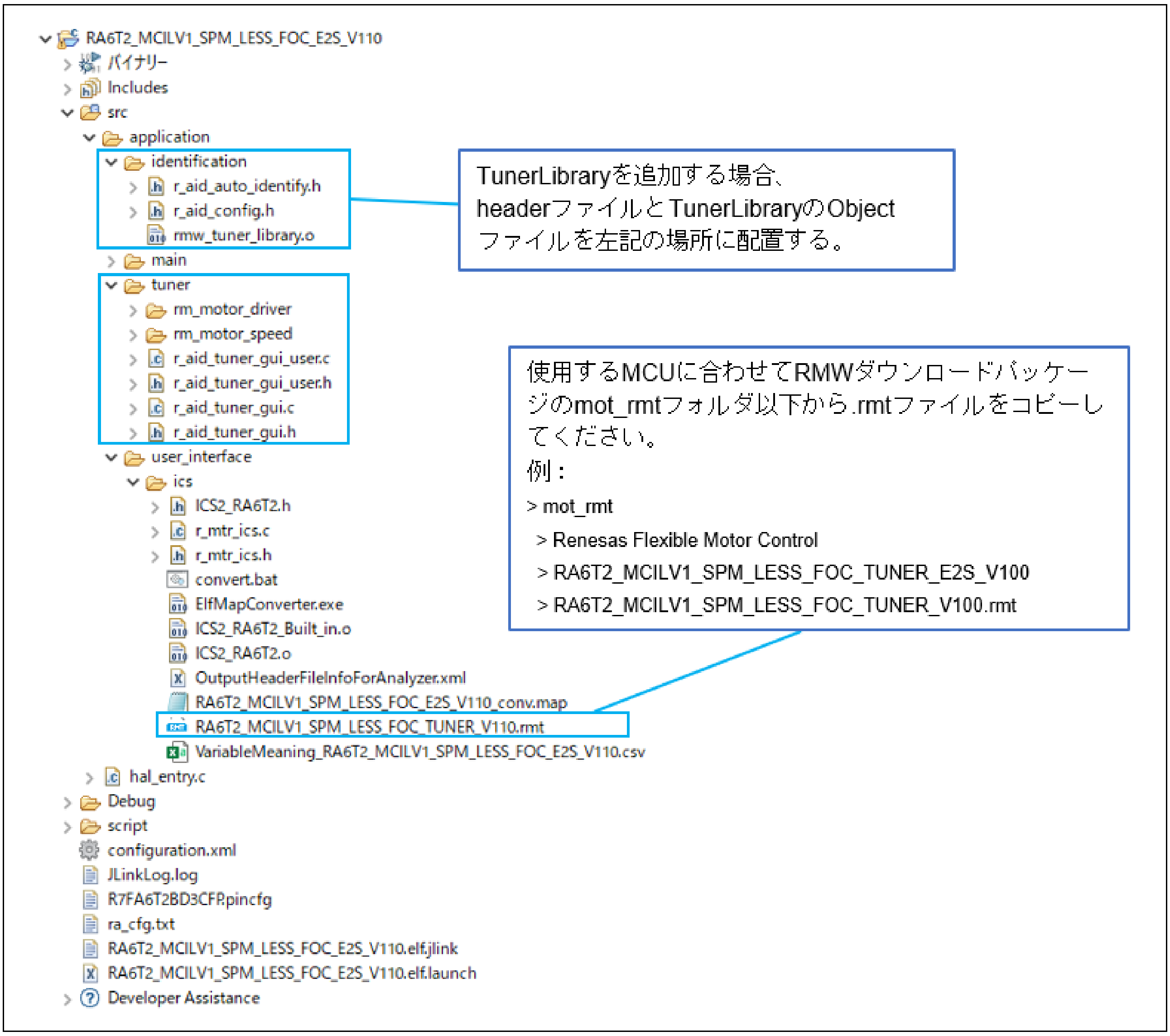


図 14-24 Tuner Library をサンプルプログラムに追加

インクルードパスを設定します。新規に追加した以下 2 つのパスを追加します。

RA6T2_MCILV1_SPM_LESS_FOC_E2S_V110¥src¥application¥identification

RA6T2_MCILV1_SPM_LESS_FOC_E2S_V110¥src¥application¥tuner

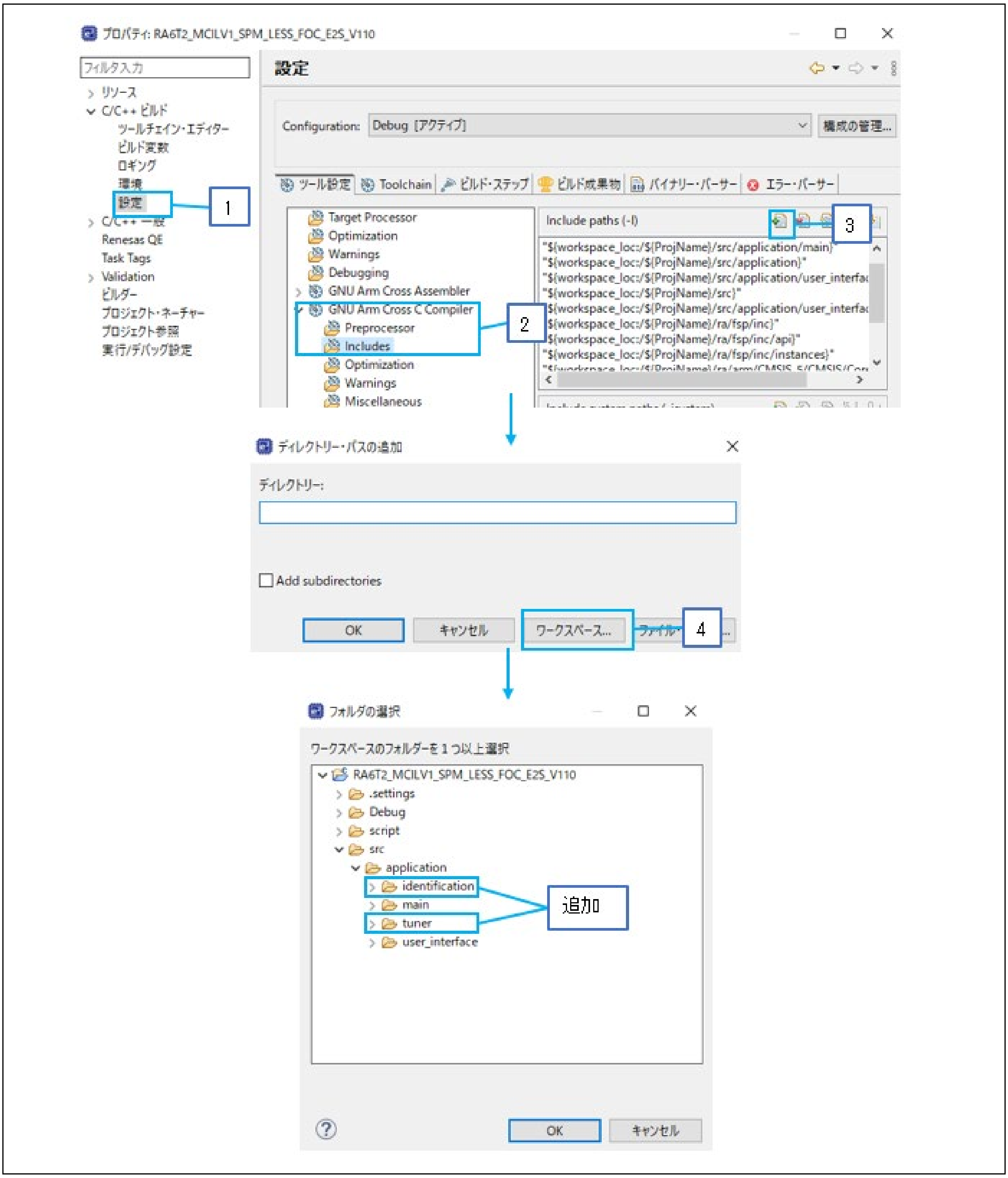


図 14-25 インクルードパスの設定

Link する Tuner Library を指定します。プロパティ画面を表示し、Tuner Library を以下の手順で追加します。

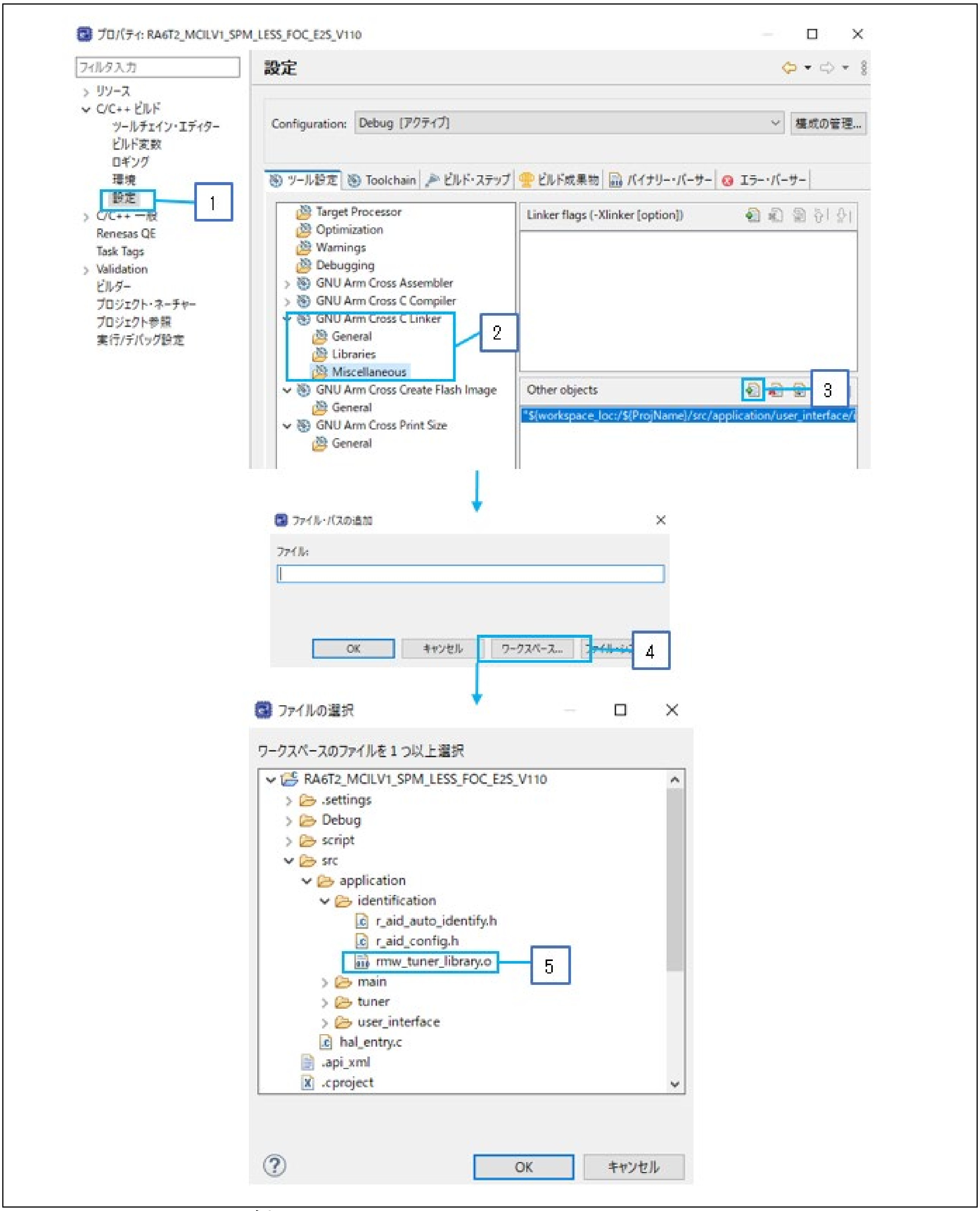


図 14-26 Tuner Library の追加

14.6.4.5 マクロ定義を実施

マクロ定義を[r_aid_tuner_gui_user.h]に行います。
表 14.7 は、サンプルコードで定義されたマクロ一覧を示しています。必要に応じて変更してください。

表 14-10　マクロ一覧

マクロ定義	概要
AIDU_CURRENT_OMEGA	電流制御系の固有周波数
AIDU_CURRENT_ZETA	電流制御系の減衰係数
AIDU_SPEED_OMEGA	速度制御系の固有周波数
AIDU_SPEED_ZETA	速度制御系の減衰係数
AIDU_E_OBS_OMEGA	誘起電圧オブザーバの固有周波数
AIDU_E_OBS_ZETA	誘起電圧オブザーバの減衰係数
AIDU_PLL_EST_OMEGA	PLL 速度推定系の固有周波数
AIDU_PLL_EST_ZETA	PLL 速度推定系の減衰係数
AIDU_INPUT_V	入力電圧
AIDU_SPEED_CTRL_PERIOD	速度制御系の周期
AIDU_INT_DECIMATION	割り込み間引き回数初期値
AIDU_INV_INFO_OVERVOLTAGE_TH	過電圧エラー閾値[V]
AIDU_INV_INFO_UNDERVOLTAGE_TH	低電圧エラー閾値[V]
AIDU_INV_INFO_PWM_CYCLE_S	PWM 周期[s]
AIDU_INV_INFO_PWM_DEADTIME_S	デッドタイム[s]
AIDU_CARRIER_SET_BASE	PWM キャリア周期設定レジスタへの格納値 （PWM タイマ周期[Hz]/キャリ周期[Hz] / 2）
AIDU_DEADTIME_SET	デッドタイム設定レジスタへの格納値
AIDU_INV_INFO_OVERCURRENT_TH	過電流エラー閾値[A]
AIDU_INV_INFO_CURRENT_RANGE	検出電流のレンジ[A] AD 変換の最小値から最大値のレンジを設定
AIDU_PWM_TIMER_W_REG	W 相 PWM タイマカウントレジスタへのポインタ
AIDU_PWM_TIMER_V_REG	V 相 PWM タイマカウントレジスタへのポインタ
AIDU_PWM_TIMER_U_REG	U 相 PWM タイマカウントレジスタへのポインタ

14.6.4.6 [RX] Tuner Library 機能の呼び出し

・コンパイルオプション定義

以下のファイルにコンパイルオプションとして使用するマクロ以下のマクロ定義を追加します。

app¥cfg¥r_app_control_cfg.h

```
#define APP_CFG_SCI_CH_SELECT     (0x60)

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

#endif /* R_APP_CONTROL_CFG_H */
```

図 14-27 [r_app_control_cfg.h]ファイルへの追加

・初期化処理の実装

以下のファイルに初期化処理を追加します。

app¥main¥r_app_main.c

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

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

    /* LED off */
    r_app_board_ui_led_control(STATEMACHINE_STATE_STOP);

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

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

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

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

図 14-28 [r_app_main.c]ファイルへの追加

・Tuner 機能動作制御処理の実装

以下のファイルに Tuner 機能動作制御処理を追加します。

app¥main¥r_app_main.c

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

図 14-29 [r_app_main.c]ファイルへの追加

・Tuner 処理結果設定処理の実装

以下のファイルに Tuner 処理結果設定を追加します。

app¥rmw¥r_app_rmw.c

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

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

        /* Call ICS */
        ics2_watchpoint();
    }

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

* Function Name : r_app_rmw_system_mode
```

図 14-30 [r_app_rmw.c]ファイルへの追加

・割り込み処理を追加

以下の３つのファイルに割り込み処理を追加します。

src¥Config_CMT0¥Config_CMT0_user.c

src¥Config_MOTOR¥Config_MOTOR_user.c

src¥Config_POE¥Config_POE_user.c

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

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

図 14-31 [Config_CMT0_user.c]ファイルへの追加

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

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

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

図 14-32 [Config_MOTOR_user.c]ファイルへの追加

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

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

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

図 14-33 [Config_POE_user.c]ファイルへの追加

・変数初期値を変更

以下のファイルの変数初期値を変更します。

src¥app¥r_app_rmw.c

「g_u2_conf_tool」の設定値は、2.3.5 章を参照してください。

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

図 14-34　[r_app_rmw.c]ファイルへの追加

・Tuner Library へのマクロ定義設定

以下のファイルに、お使いのボードでのピンアサインに合わせて設定してください。

tuner¥r_aid_tuner_gui_user.h

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

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

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

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

図 14-35　[r_aid_tuner_gui_user.h]への追加

14.6.4.7 [RA] Tuner Library 機能の呼び出し

・初期化処理の実装

以下のファイルに初期化処理を追加します。

src¥application¥main¥mtr_main.c

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

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

図 14-36 [mtr_main.c]ファイルへの追加

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

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

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

図 14-37 [mtr_main.c]ファイルへの追加

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

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

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

図 14-38 [mtr_main.c]ファイルへの追加

・Tuner 機能動作制御処理の実装

以下のファイルに Tuner 機能動作制御処理を追加します。

src¥application¥main¥mtr_main.c

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

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

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

図 14-39 [mtr_main.c]ファイルへの追加

・過電流処理の実装

以下のファイルに過電流処理を追加します。

src¥application¥main¥mtr_main.c

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

図 14-40 [mtr_main.c]ファイルへの追加

・変数初期値を変更

以下のファイルの変数初期値を変更します。

src¥application¥main¥mtr_main.c

「g_u2_conf_tool」の設定値は、2.3.5 章を参照してください。

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

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

図 14-41　[mtr_main.c]ファイルへの追加

・Tuner Library へのマクロ定義設定

以下のファイルに、お使いのボードでのピンアサインに合わせて設定してください。

src¥application¥tuner¥r_aid_tuner_gui_user.h

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

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

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

#define AIDU_TUNE_VOLTERR_ENABLE      (true)
```

図 14-42　[r_aid_tuner_gui_user.h]への追加

14.6.4.8 ユーザ実装関数のコーディング

Tuner Library が周辺機能にアクセスするために、ユーザ実装関数を設定する必要があります。Tuner Library は設定されたユーザ実装関数の関数ポインタから周辺機能にアクセスします。
[r_aid_tuner_gui_use.c]ファイルにユーザ実装関数の処理をコーディングします。サンプルコードを変更した場合は、変更内容に応じ一覧表内の青字部分を変更して下さい。

表 14-11 ユーザ実装関数一覧 (1/6)

関数名	引数	型	概要	関数ポインタ変数名
aid_mtr_inv_set_uvw	float f4_duty_u : U 相変調率 float f4_duty_v : V 相変調率 float f4_duty_w : W 相変調率	void	PWM デューティを設定します。	g_fp_aid_internal_inv_set_uvw

	void aid_mtr_inv_set_uvw(float f4_duty_u, float f4_duty_v, float f4_duty_w)
	{
	R_MOTOR_DRIVER_BldcDutySet(g_st_sensorless_vector.p_st_driver, f4_duty_u, f4_duty_v, f4_duty_w);
	}			

aid_mtr_inv_get_uvw	float *f4_duty_u : U 相変調率 float *f4_duty_v : V 相変調率 float *f4_duty_w : W 相変調率	void	PWM 出力の実際のデューティ周期を得ます。	ー

	void aid_mtr_inv_get_uvw(float *f4_duty_u, float *f4_duty_v, float *f4_duty_w)
	{
	*f4_duty_u = 1.0f - (((float)AIDU_PWM_TIMER_U_REG - ((float)AIDU_DEADTIME_SET * 0.5f) - 1.0f) / (float)AIDU_CARRIER_SET_BASE);
	*f4_duty_v = 1.0f - (((float)AIDU_PWM_TIMER_V_REG - ((float)AIDU_DEADTIME_SET * 0.5f) - 1.0f) / (float)AIDU_CARRIER_SET_BASE);
	*f4_duty_w = 1.0f - (((float)AIDU_PWM_TIMER_W_REG - ((float)AIDU_DEADTIME_SET * 0.5f) - 1.0f) /			

	(float)AIDU_CARRIER_SET_BASE); }	

表 14-12　ユーザ実装関数一覧 (2/6)

<table>
<tr><th>関数名</th><th>引数</th><th>型</th><th>概要</th><th>関数ポインタ変数名</th></tr>
<tr><td rowspan="2">aid_mtr_get_current_iuiw</td><td>float *f4_iu_ad : U-phase current
float *f4_iw_ad : W-phase current</td><td>void</td><td>U 相, W 相の電流値を得ます。identification の期間、最新値を得るために常にコールされます。</td><td rowspan="2">g_fp_aid_internal_get_current_iuiw</td></tr>
<tr><td colspan="3">void aid_mtr_get_current_iuiw(float *f4_iu_ad, float *f4_iw_ad)
{
    r_mtr_adc_tb st_ad_data;

    g_st_sensorless_vector.p_st_driver->ADCDataGet(&st_ad_data);
    *f4_iu_ad = (MOTOR_MCU_CFG_ADC_OFFSET - st_ad_data.u2_iu_ad) * g_st_sensorless_vector.p_st_driver->f4_ad_crnt_per_digit;
    *f4_iw_ad = (MOTOR_MCU_CFG_ADC_OFFSET - st_ad_data.u2_iw_ad) * g_st_sensorless_vector.p_st_driver->f4_ad_crnt_per_digit;
}</td></tr>
<tr><td rowspan="2">aid_mtr_get_vdc</td><td>－</td><td>float : Supply voltage</td><td>供給電圧値を得ます。identification の期間、最新値を得るために常にコールされます。</td><td rowspan="2">g_fp_aid_internal_get_vdc</td></tr>
<tr><td colspan="3">float aid_mtr_get_vdc(void)
{
    r_mtr_adc_tb st_ad_data;
    float temp_vdc;

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

表 14-13　ユーザ実装関数一覧 (3/6)

<table>
<tr><th>関数名</th><th>引数</th><th>型</th><th>概要</th><th>関数ポインタ変数名</th></tr>
<tr><td rowspan="2">aid_mtr_ctrl_start</td><td>ー</td><td>void</td><td>aid_mtr_inv_set_uvw 関数で制御可能な PWM 出力をドライブモードに遷移します。START ID 時にコールされます。</td><td rowspan="2">g_fp_aid_internal_ctrl_start</td></tr>
<tr><td colspan="3">void aid_mtr_ctrl_start(void)
{

R_MOTOR_DRIVER_BldcDutySet(g_st_sensorless_vector.p_st_driver, 0.5f, 0.5f, 0.5f);

R_MOTOR_DRIVER_PWMControlStart(g_st_sensorless_vector.p_st_driver);
}</td></tr>
<tr><td rowspan="2">aid_mtr_ctrl_stop</td><td>ー</td><td>void</td><td>ドライブモードから STOP モードに遷移します。</td><td rowspan="2">g_fp_aid_internal_ctrl_stop</td></tr>
<tr><td colspan="3">void aid_mtr_ctrl_stop(void)
{

R_MOTOR_DRIVER_PWMControlStop(g_st_sensorless_vector.p_st_driver);

R_MOTOR_DRIVER_BldcDutySet(g_st_sensorless_vector.p_st_driver, 0.5f, 0.5f, 0.5f);
}</td></tr>
<tr><td rowspan="2">aid_mtr_clear_oc_flag</td><td>ー</td><td>void</td><td>過電流検出時の強制遮断フラグをクリアします。</td><td rowspan="2">g_fp_aid_internal_clear_oc_flag</td></tr>
<tr><td colspan="3">void aid_mtr_clear_oc_flag(void)
{
    R_Config_POE_Stop();
    R_Config_POE_Start();
}</td></tr>
</table>

表 14-14 ユーザ実装関数一覧 (4/6)

<table>
<tr><th>関数名</th><th>引数</th><th>型</th><th>概要</th><th>関数ポインタ変数名</th></tr>
<tr><td rowspan="2">rmw_apply_identified_params</td><td>−</td><td>void</td><td>RMW のパラメータを設定します。</td><td rowspan="2">−</td></tr>
<tr><td colspan="3">void rmw_apply_identified_params(void)
{
st_aid_id_setting_t id_setting;
float ia_max;
float va_max;

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

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

com_u1_enable_write ^= 1;
}</td></tr>
<tr><td rowspan="2">rmw_apply_reset</td><td>−</td><td>void</td><td>RMW のパラメータをリセットします。</td><td rowspan="2">−</td></tr>
<tr><td colspan="3">void rmw_apply_reset(void)
{
com_u1_system_mode = STATEMACHINE_EVENT_RESET;
}</td></tr>
</table>

表 14-15 ユーザ実装関数一覧 (5/6)

<table>
<tr><th>関数名</th><th>引数</th><th>型</th><th>概要</th><th>関数ポインタ変数名</th></tr>
<tr><td rowspan="2">mtr_ics_1_parameter</td><td>ー</td><td>void</td><td>Tune ウィンドウの Easy タブ設定内容を反映します。</td><td rowspan="2">ー</td></tr>
<tr><td colspan="3"><pre>void mtr_ics_1_parameter(void)
{
    uint8_t u1_temp;

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

表 14-16　ユーザ実装関数一覧 (6/6)

関数名	引数	型	概要	関数ポインタ変数名
aid_mtr_ics_interrupt	—	void	RMW 割り込み処理	ー

	void aid_mtr_ics_interrupt(void)
	{
	if (R_TUNER_GetFlugReset() == AIDU_FLUG_RESET_MDOE1)
	{
	com_u1_system_mode = STATEMACHINE_EVENT_RESET;
	R_TUNER_SetFlugReset(AIDU_FLUG_RESET_MDOE2);
	}
	}			

aid_mtr_get_inv_info	st_aid_inv_info_t* st_inv_info : inv info structure pointer	void	インバータの情報を得ます。	ー

	void aid_mtr_get_inv_info(st_aid_inv_info_t* st_inv_info)
	{
	st_inv_info->duty_min = AIDU_INV_INFO_DUTY_MIN;
	st_inv_info->duty_max = AIDU_INV_INFO_DUTY_MAX;
	st_inv_info->overcurrent_th = AIDU_INV_INFO_OVERCURRENT_TH;
	st_inv_info->overvoltage_th = AIDU_INV_INFO_OVERVOLTAGE_TH;
	st_inv_info->undervoltage_th = AIDU_INV_INFO_UNDERVOLTAGE_TH;
	st_inv_info->pwm_cycle_s = AIDU_INV_INFO_PWM_CYCLE_S;
	st_inv_info->pwm_deadtime_s = AIDU_INV_INFO_PWM_DEADTIME_S;
	st_inv_info->pwm_lsb = 1.0f / (float)AIDU_CARRIER_SET_BASE;
	st_inv_info->current_lsb = AIDU_INV_INFO_CURRENT_RANGE / 4096; /* Full current range / ADC max digits */
	}			

14.6.4.9 サンプルコードのビルド

変更が完了したら以下の手順でビルドしエラーが無いことを確認して下さい。

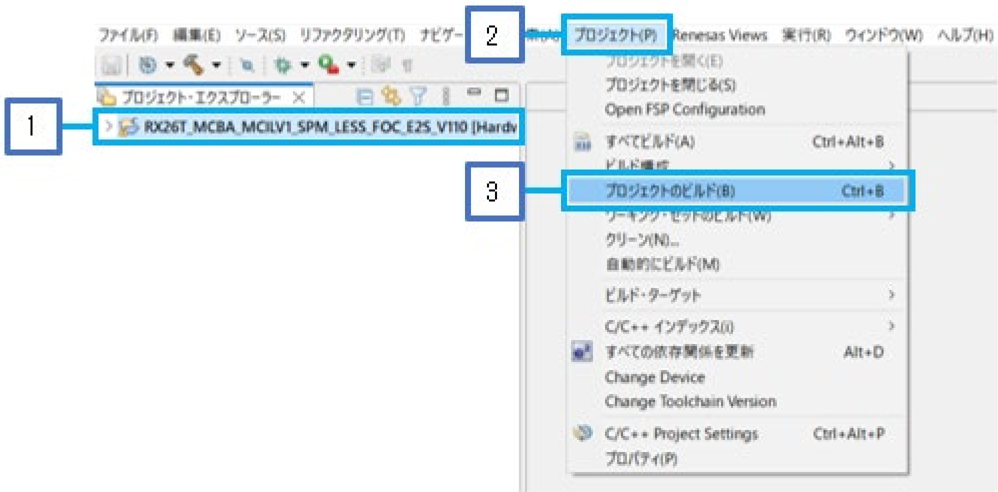


図 14-43 サンプルコードのビルド

```
'Build complete.'

08:32:57 Build Finished. 0 errors, 1 warnings. (took 1s.306ms)
```

図 14-44 サンプルコードのビルド結果

エラーが発生した場合は、14.6.4.3 章以降の手順で行ったファイルの追加、またはコードの追加・変更箇所を確認してください。

15. Tuner ツール（RL78 向け）

15.1 概要

本章では RL78 マイコン向けの Tuner 機能の使用方法を説明します。モータ (永久磁石同期モータ)固有のパラメータ(抵抗、インダクタンス等)を自動測定し、センサレスベクトル制御をする際に必要な各種パラメータ(制御パラメータ、スケーリングパラメータ等)を自動調整する機能です。

対応した実行ファイルを書き込んだ CPU ボードと、Renesas Motor Workbench を接続した際に、RL78 向け Tuner 機能を利用することが可能となります。その場合、Main Window 画面の「Select Tool」に「Tuner」が選択可能な状態で表示されます。

注意　本ツールは、すべてのモータに対して有効に機能することを保証するものではありません。

15.2 特徴

- モータの固有パラメータを自動で測定し、モータの駆動に必要なパラメータを自動で設計します。
- パラメータ設計結果を PDF 形式のレポートや、ルネサス製制御プログラムのヘッダファイル形式で出力可能です。
- 設計したパラメータを使用して、基本的な駆動のテストを簡単に行えます。

15.3　画面構成

Tuner ツールの構成を以下に示します。

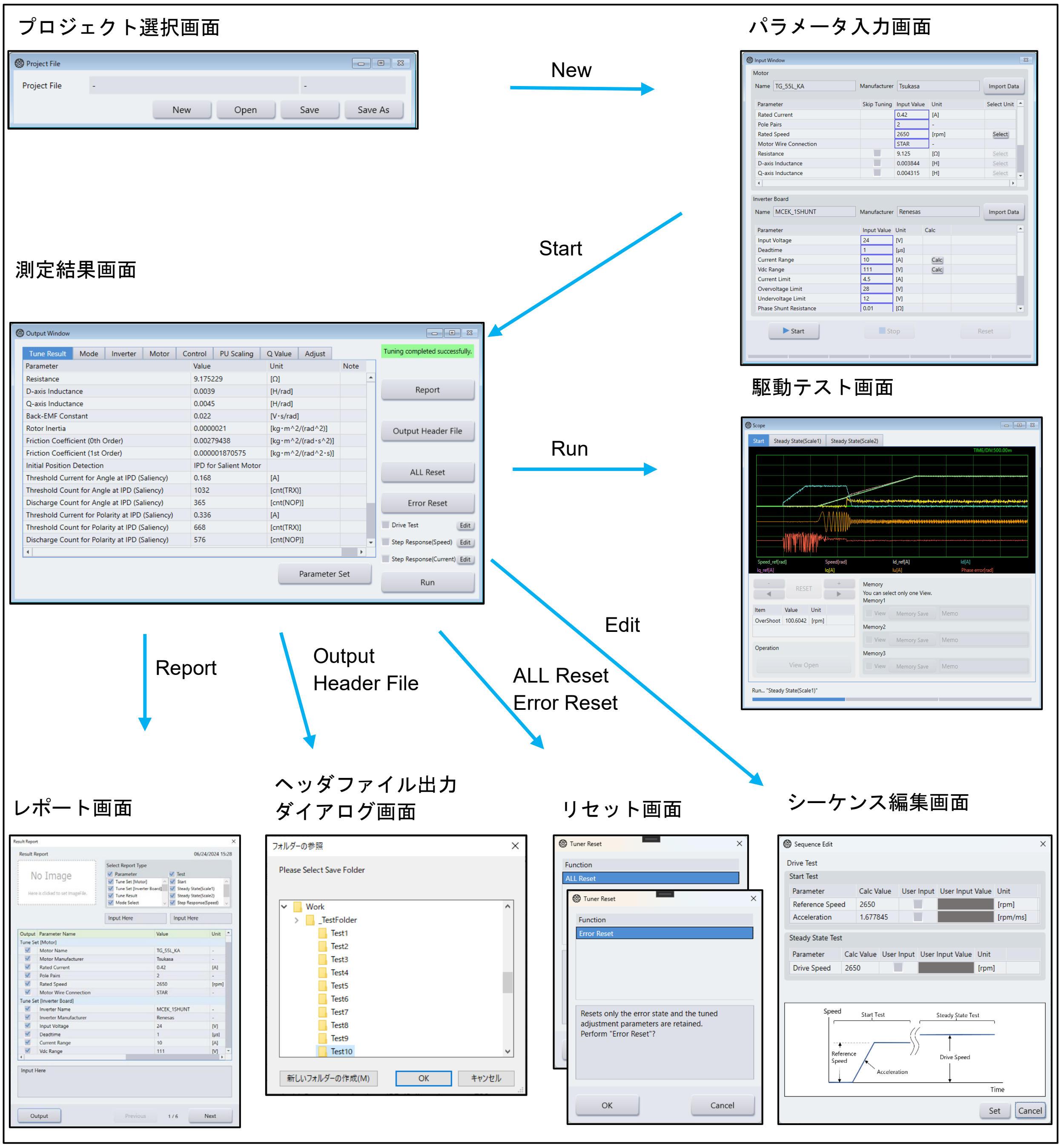


図 15-1Tuner ツール画面

15.4 操作説明

15.4.1 事前準備

①実行ファイルの書き込み

Tuner 用 MOT ファイルを対象の CPU ボードに書き込みます。

・***_MCEK_LESS_FOC_APM_V***.mot

②Renesas Motor Workbench の起動

デスクトップにあるショートカットアイコン「Renesas Motor Workbench」より起動します。

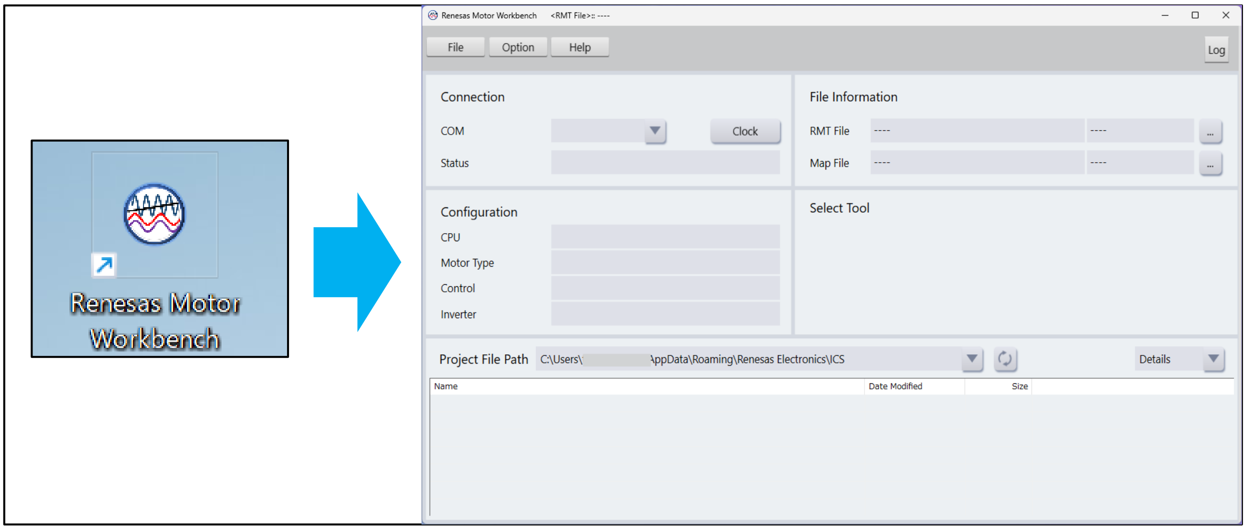


図 15-2 Renesas Motor Workbench の起動

③RMT ファイルの読み込み
Tuner 用 RMT ファイルを読み込みます。
・***_MCEK_LESS_FOC_APM_V***.rmt

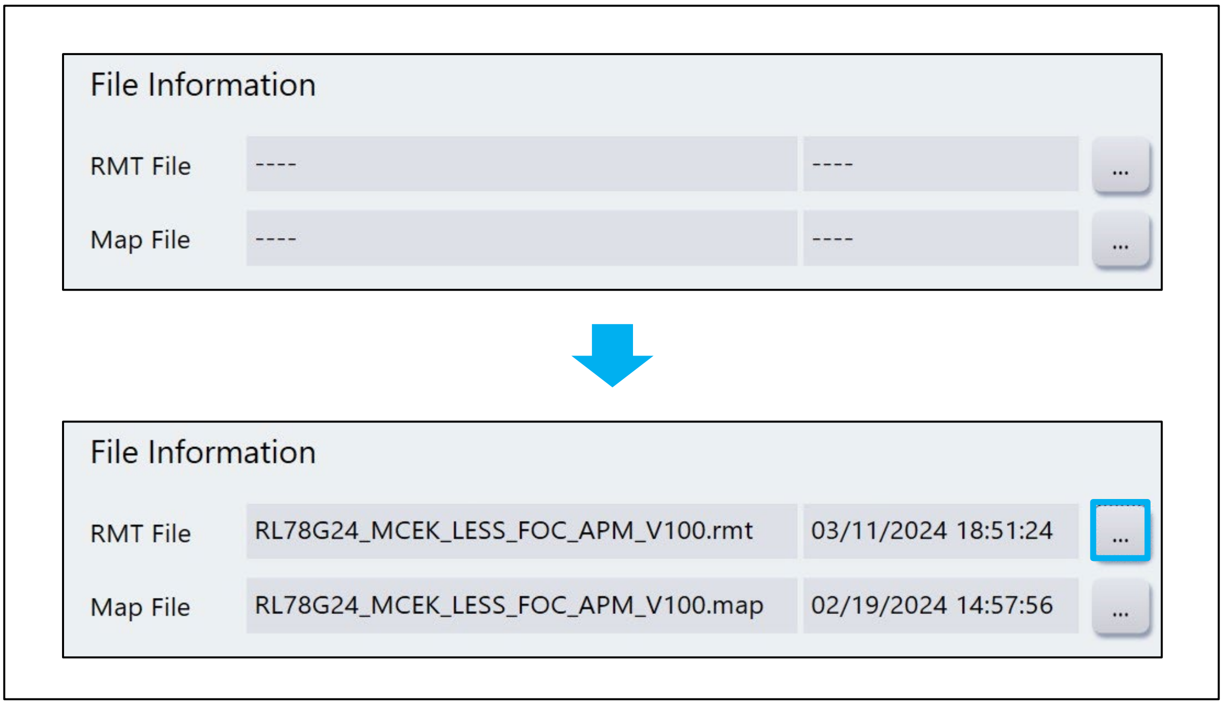


図 15-3 RMT ファイル読み込み

④通信の確立
プルダウンメニューから有効な接続(COMx)を選択します。通信が確立された場合、設定内容が「Configuration」に表示されます。

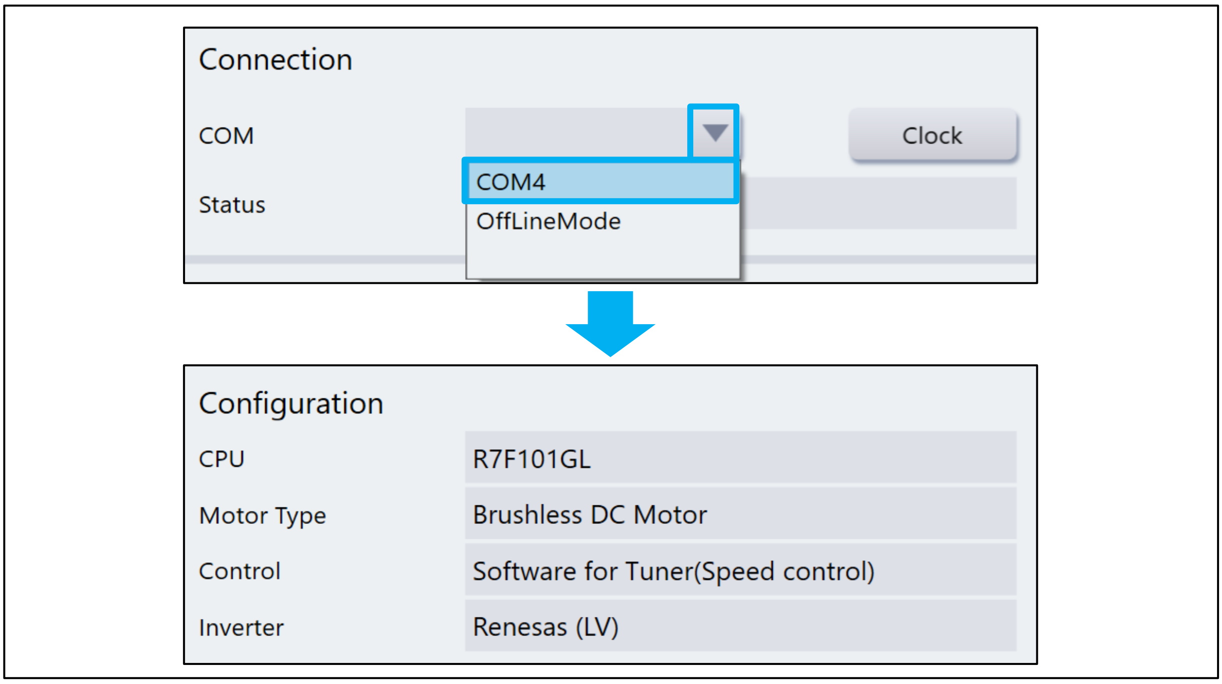


図 15-4 Renesas Motor Workbench との通信を確立

15.4.2 自動測定実行

15.4.2.1 Tuner の起動

「Select Tool」から「Tuner」を選択して起動します。

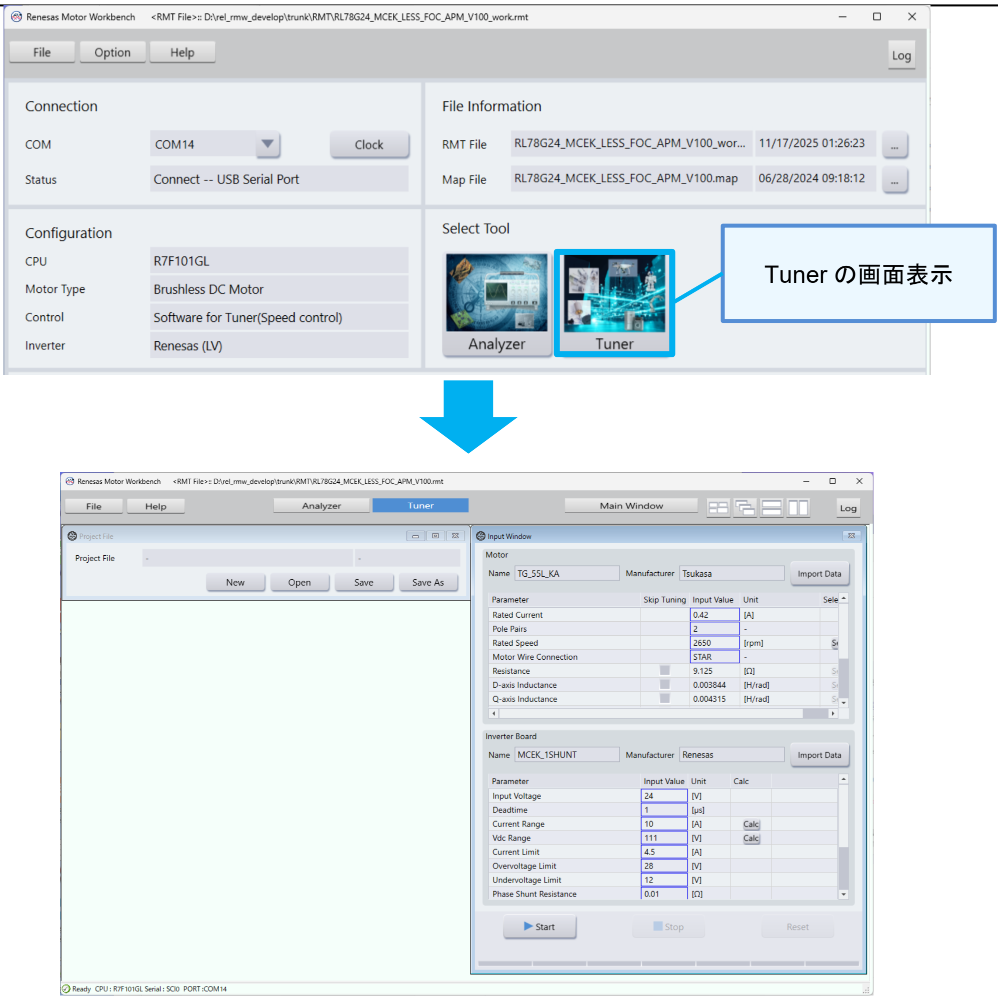


図 15-5 Tuner の起動

起動すると Input Window 画面と Project File 画面が表示されます。Project File 画面は、常時表示されます。機能については、「15.5.1 プロジェクトファイル管理機能」をご確認ください。

図 15-6 Project File 画面

15.4.2.2　　自動測定用パラメータの入力

Tuner を起動すると、Input Window 画面が表示されます。

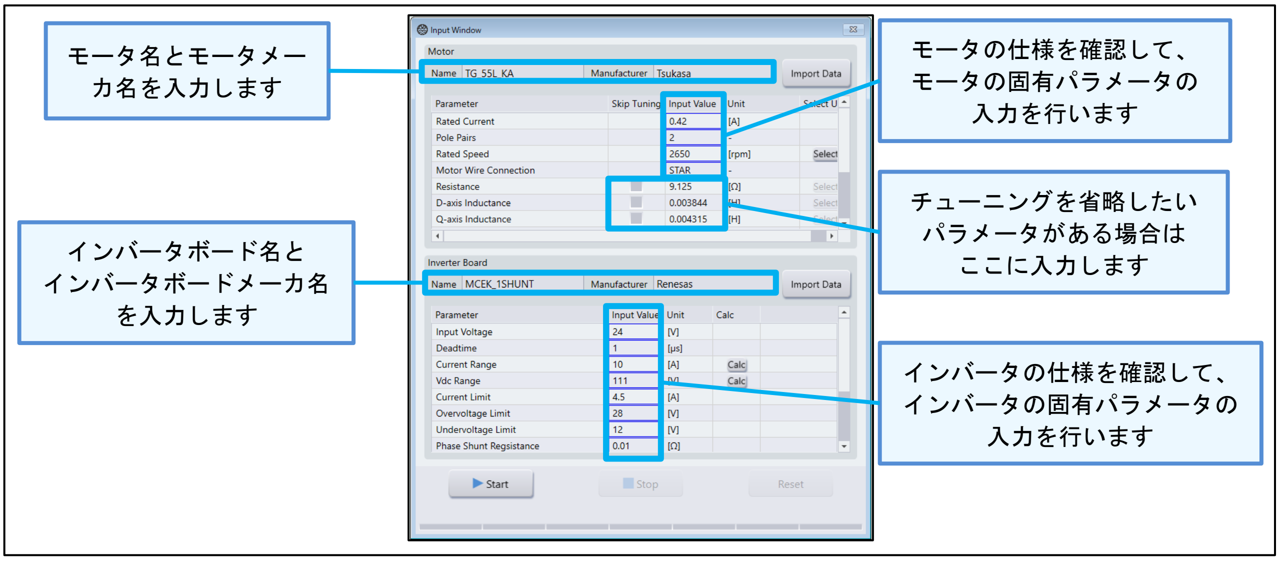


図 15-7 Input Window 画面

自動測定に必要なパラメータを入力します。入力パラメータについては、「表 15-1　モータの固有パラメータの入力必須項目」「表 15-2　インバータの固有パラメータの入力必須項目一覧」をご確認ください。

「Skip Tuning」がチェックされていないパラメータのみ自動測定が行われ、自動測定を行わないパラメータは、「Input Value」で入力した値を使用します。

また、「Import Data」ボタンからプロジェクトファイルを読み込むことによって、過去に保存したパラメータを使用することができます。

※現行のバージョンでは、インバータの固有パラメータはデフォルト値のみ使用可能であり、値の変更には対応しておりません。

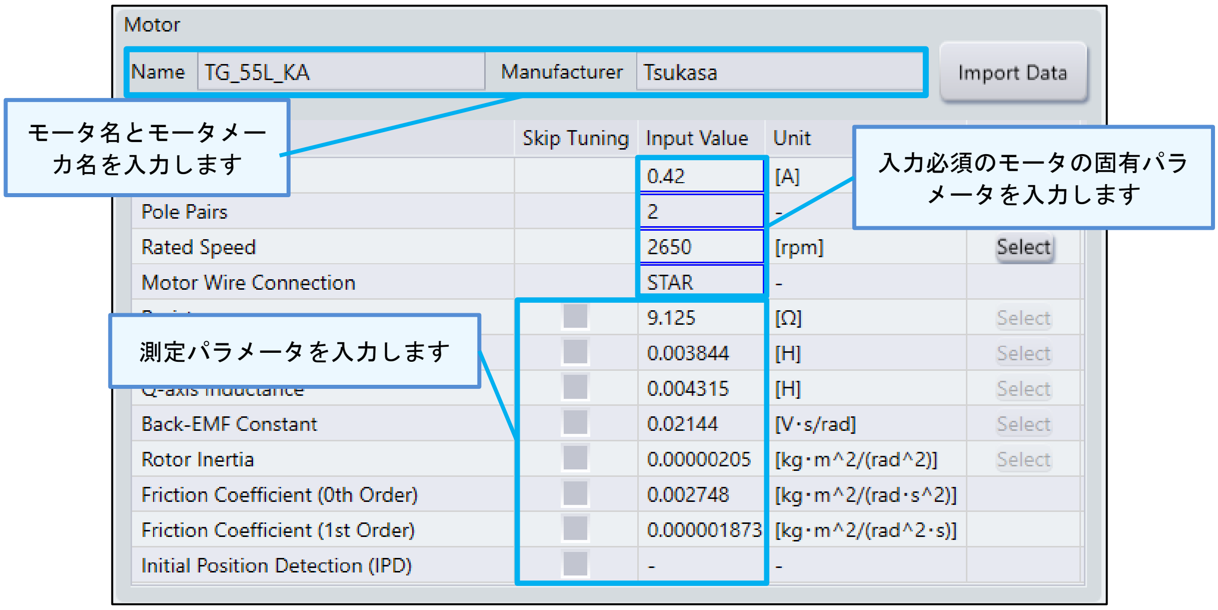


図 15-8 モータの固有パラメータの入力必須項目

表 15-1　モータの固有パラメータの入力必須項目

項目名	概要	入力必須 / 測定
Motor Name	モータ名	入力必須項目
Motor Manufacturer	モータメーカ名	入力必須項目
Rated Current	定格電圧 [A]	入力必須項目
Pole Pairs	極対数	入力必須項目
Rated Speed	定格速度 [rpm]	入力必須項目
Motor Wire Connection	結線方式	入力必須項目
Resistance	抵抗 [Ω]	測定項目
d-axis inductance	d 軸インダクタンス [H/rad]	測定項目
q-axis inductance	q 軸インダクタンス [H/rad]	測定項目
Back-EMF Constant	誘起電圧定数 [V･s/rad]	測定項目
Rotor Inertia	イナーシャ(慣性モーメント) [kg･m^2/(rad^2)]	測定項目
Friction Coefficient (0th Order)	摩擦トルクの切片 [kg･m^2/(rad･s^2)]	測定項目
Friction Coefficient (1st Order)	摩擦トルクの傾き [kg･m^2/(rad^2･s)]	測定項目
Initial Position Detection (IPD)	初期位置検出	測定項目

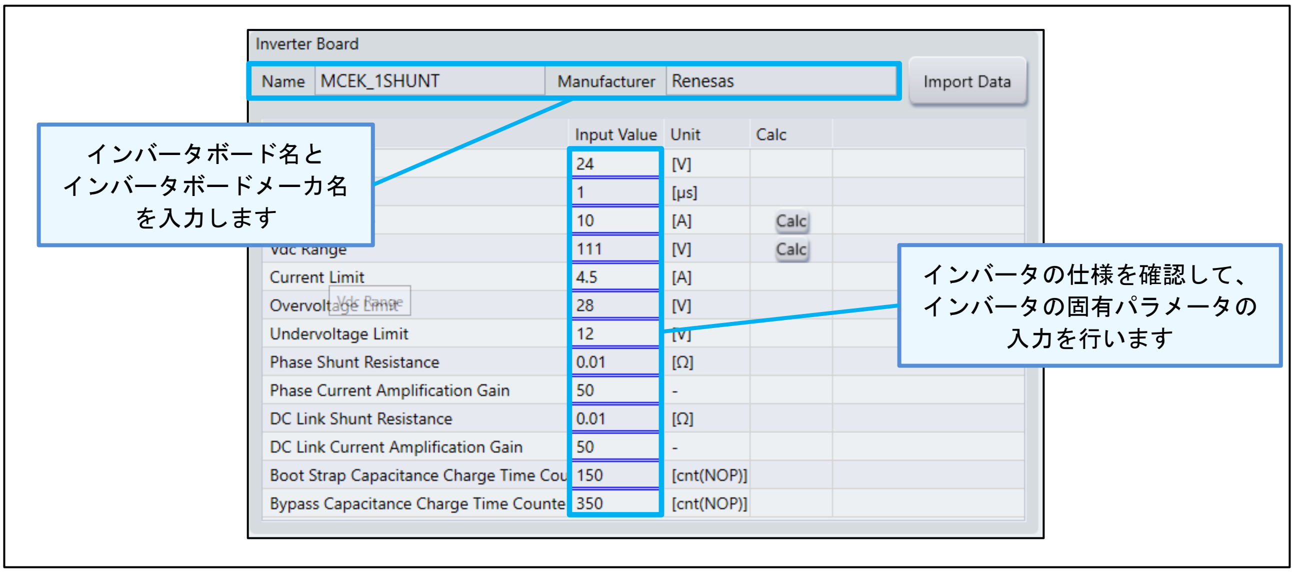


図 15-9 インバータの固有パラメータの入力必須項目

表 15-2　インバータの固有パラメータの入力必須項目一覧

項目名	概要
Inverter Name	インバータボード名
Inverter Manufacturer	インバータボードメーカ名
Input Voltage	入力電圧 [V]
Deadtime	デッドタイム [μsec]
Current Range	インバータボード回路の電流レンジ [A]
Vdc Range	インバータボード回路の電圧レンジ [V]
Current Limit	電流リミット [A]
Overvoltage Limit	過電圧リミット [V]
Undervoltage Limit	低電圧リミット [V]
Phase Shunt Resistance	3 相のシャント抵抗値 [Ω]
Phase Current Amplification Gain	相電流の増幅率
DC Link Shunt Resistance	DC リンクシャント抵抗値 [Ω]
DC Link Current Amplification Gain	DC リンク電流の増幅率
Boot Strap Capacitance Charge Time Counter	ブートストラップコンデンサチャージ時間カウント値
Bypass capacitance charge time Counter	平滑コンデンサチャージ時間カウント値

15.4.2.3 自動測定実行

「Start」ボタンをクリックすると、自動測定を開始します。プログレスバーに測定の進捗度合いが表示され、自動測定が完了すると Output Window 画面を表示します。

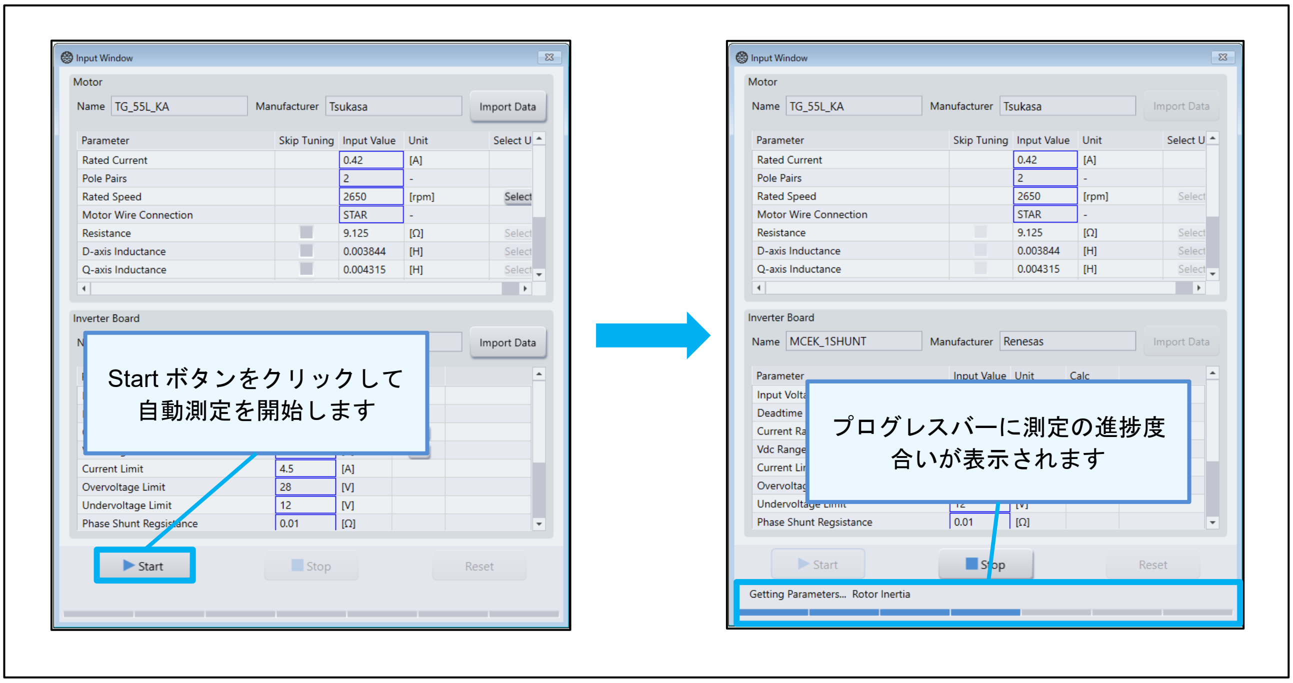


図 15-10 自動測定実行中画面

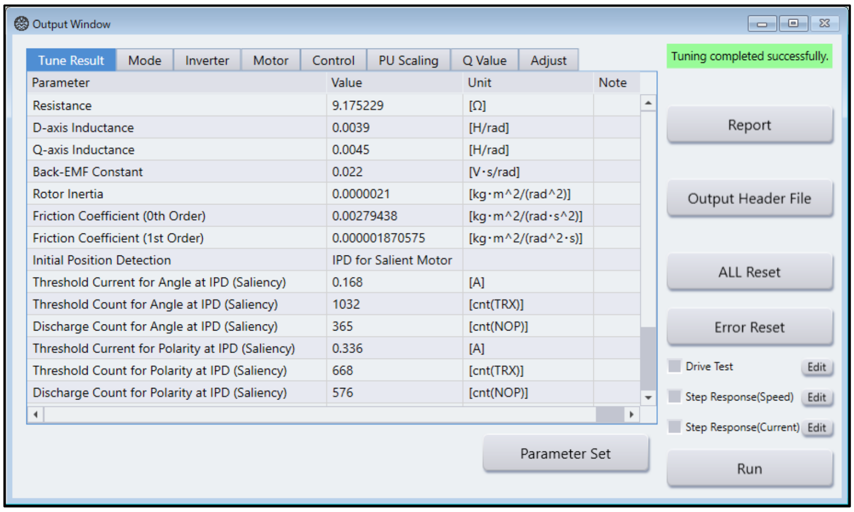


図 15-11 Output Window 画面

自動測定時にエラーが発生した場合、エラーメッセージを確認し、「Reset」ボタンをクリックしてください。また、自動測定中に停止させたい場合は、「Stop」ボタンをクリックしてください。

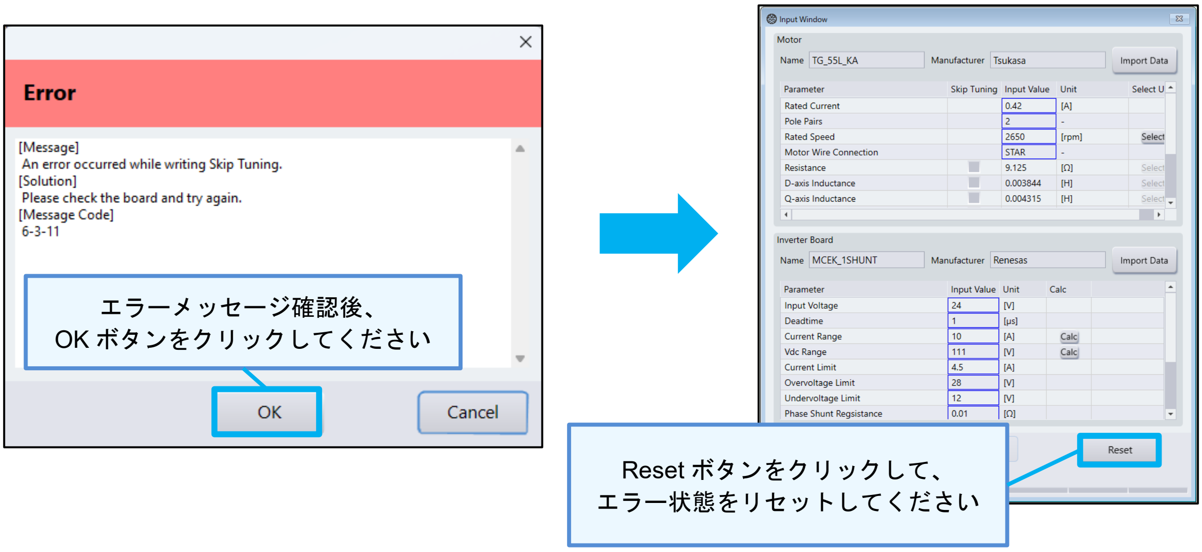


図 15-12 自動測定中のエラーメッセージ例

15.4.3 パラメータ設計

15.4.3.1　　自動測定結果の確認

自動測定結果は、Output Window 画面の Tune Result タブに表示されます。

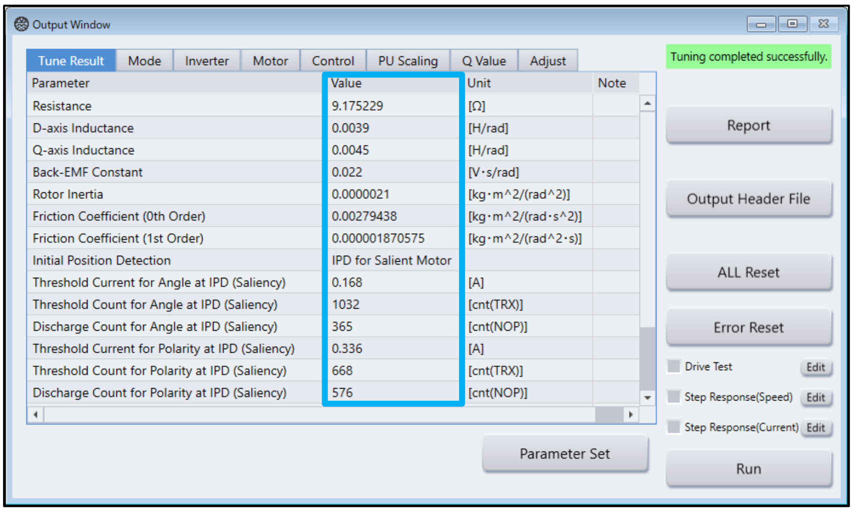


図 15-13 Tune Result タブ

15.4.3.2 パラメータ反映

「Inverter」、「Motor」、「Control」、「PU Scaling」、「Q Value」、「Adjust」各タブの Value 列には自動測定したパラメータから算出された値が格納されます。青枠で囲われたパラメータの値は編集可能です。「Parameter Set」ボタンをクリックすることでマイコンに書き込まれているプログラムの対応する変数へ反映することができます。一部、プログラムへの反映に対応していないパラメータもありますが、その場合もヘッダファイルへの出力には反映されます。

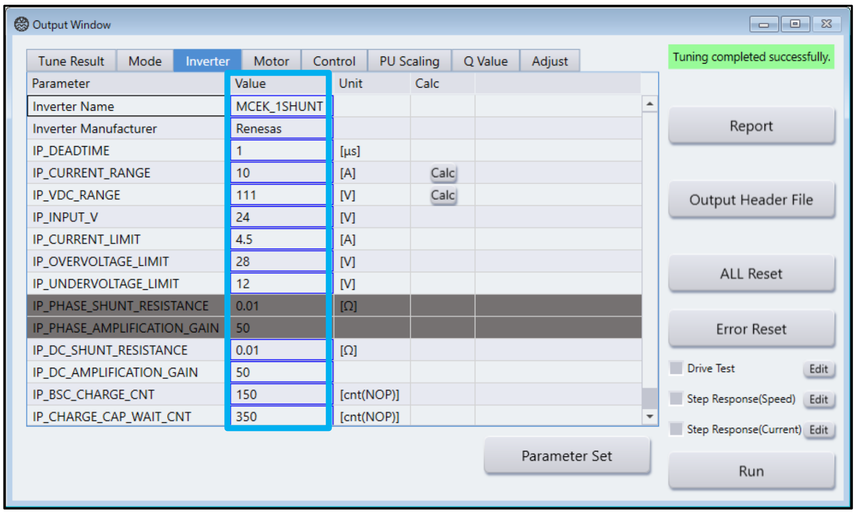


図 15-14 Inverter タブ

15.4.3.3　制御モードの編集

制御モードは、Mode タブで編集します。制御モードの設定は「Parameter Set」ボタンをクリックしてもマイコンに書き込まれているプログラムへ反映されません。設定した値は、ヘッダファイルに出力されます。

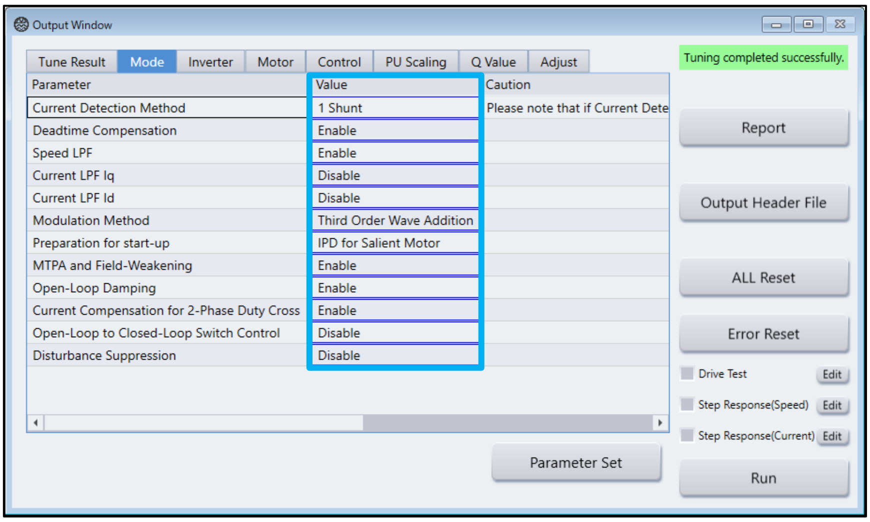


図 15-15 Mode タブ

15.4.4 駆動テスト

駆動テストは「始動と定常駆動」、「ステップ応答(速度)」、「ステップ応答(電流)」の 3 種類のテストから実行するテストを選択することが可能で、複数のテストを連続で実行することも可能です。

15.4.4.1、15.4.4.2、15.4.4.3 において波形が自動で表示されますが、接続されているモータによっては波形が完全に表示されない場合があります。その場合、「View Open」ボタンをクリックして Scope Channel Info 画面を表示して、波形の表示設定を編集することができます。

※駆動テスト時にモータが正常に駆動しない場合があります。モータの駆動に異常を感じたら、すぐに Output Window 画面の「Stop」ボタンを押下してモータを停止してください。

※モータの始動時間が長い場合、始動テストの波形が時間スケールで収まらず、後続の駆動テストも適切に実行されない場合があります。また、駆動テストを 2 回以上実行する場合、初回実行時と同じ縦軸のスケールとオフセットで波形が表示されます。

15.4.4.1 始動と定常駆動

始動と定常駆動を実行します。Output Window 画面の①「Drive Test」のチェックボックスにチェックを入力します。その後、②「Run」ボタンをクリックすると、Scope 画面が表示されます。

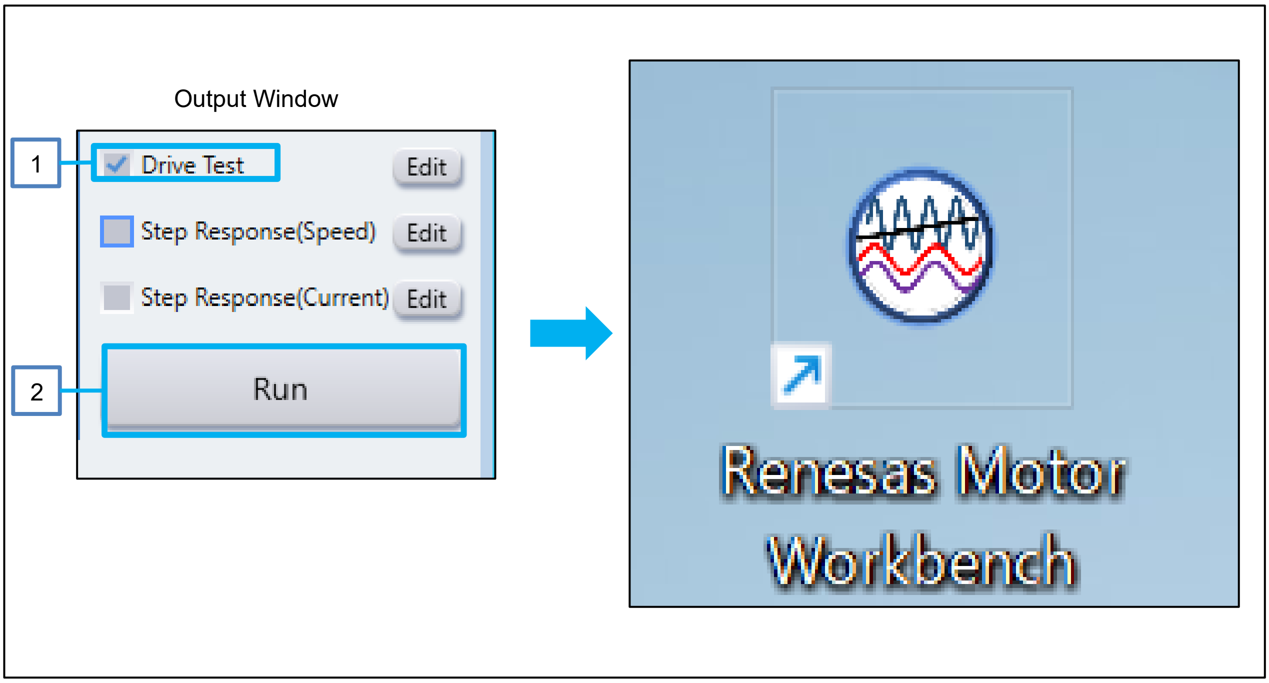


図 15-16 駆動テスト(始動と定常駆動)

テスト条件を変更する際には、「Edit」ボタンから使用するシーケンスの値を調整することができます。

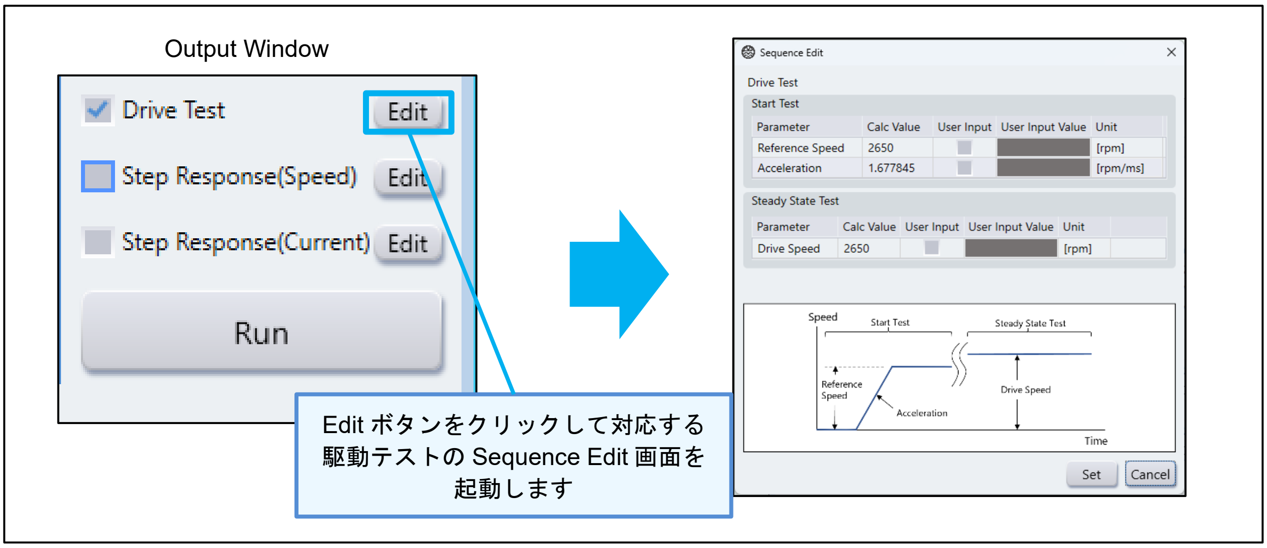


図 15-17 Sequence Edit 画面

15.4.4.2　　ステップ応答(速度)

速度のステップ応答を実行します。Output Window 画面の①「Step Response(Speed)」のチェックボックスにチェックを入れます。そのあとに②「Run」ボタンをクリックすると、Scope 画面が表示されます。

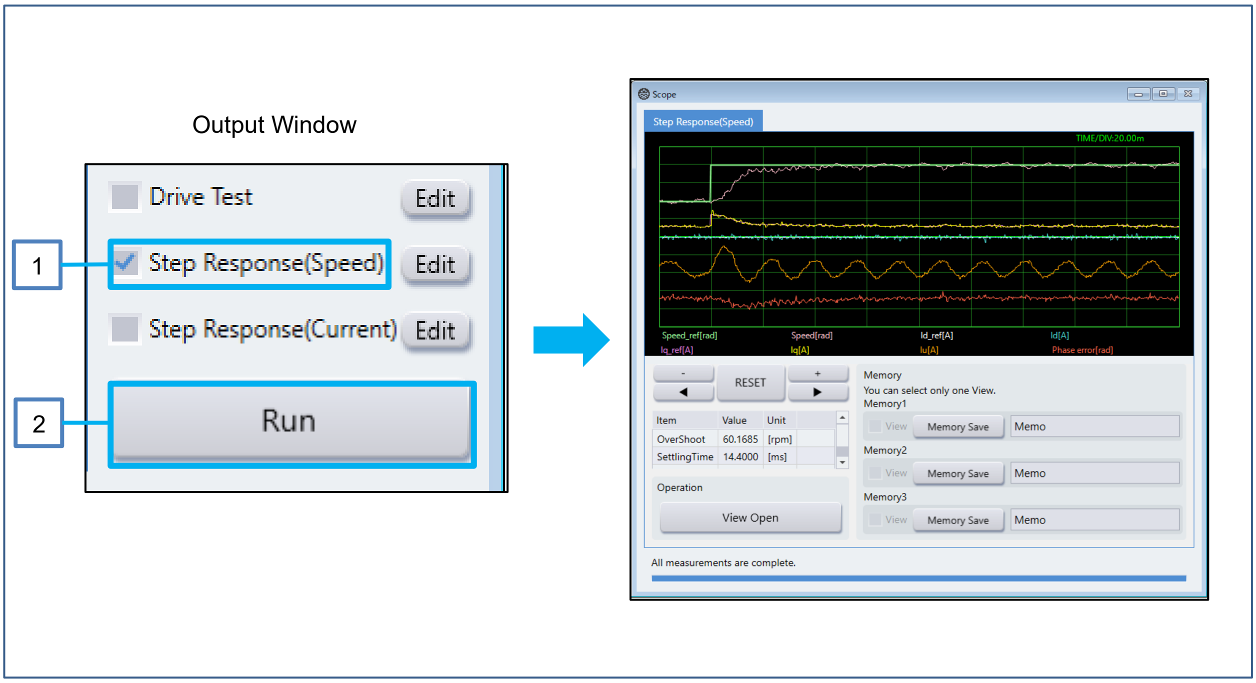


図 15-18 駆動テスト(ステップ応答(速度))

15.4.4.3　　ステップ応答(電流)

電流のステップ応答を実行します。Output Window 画面の①「Step Response(Current)」のチェックボックスにチェックを入れます。そのあとに②「Run」ボタンをクリックすると、Scope 画面が表示されます。

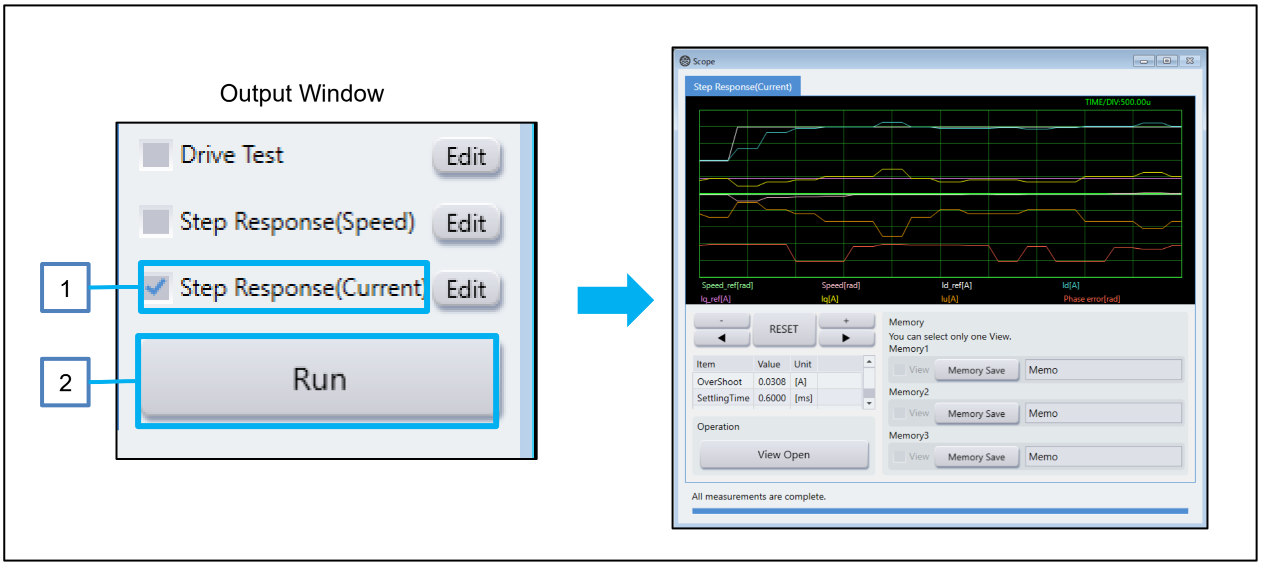


図 15-19 駆動テスト(ステップ応答(電流))

15.4.5 結果出力

15.4.5.1 レポート出力

設計したパラメータ、および駆動テスト結果をレポートファイルとして出力します。「Report」ボタンをクリックすると Result Report 画面が表示され、「Output」ボタンでレポートを出力することができます。

※レポートファイルにパラメータの値が適切に反映されない場合があります。正しいパラメータの値は Output Window 画面をご確認ください。

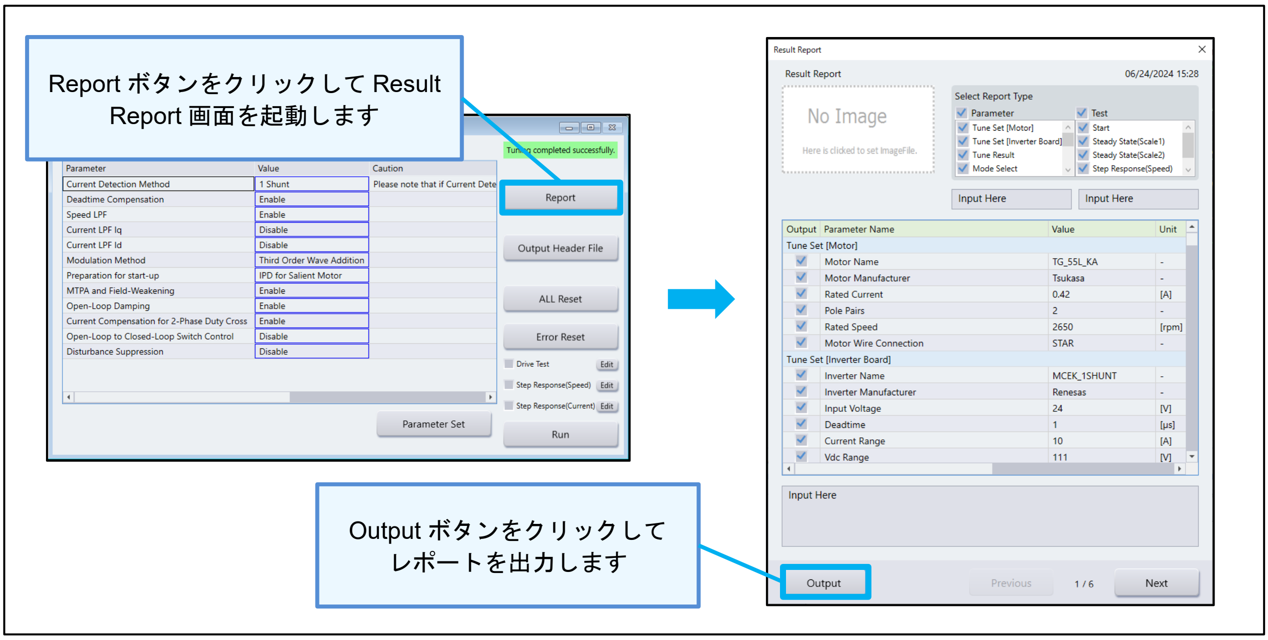


図 15-20 Result Report 画面(パラメータ表示)

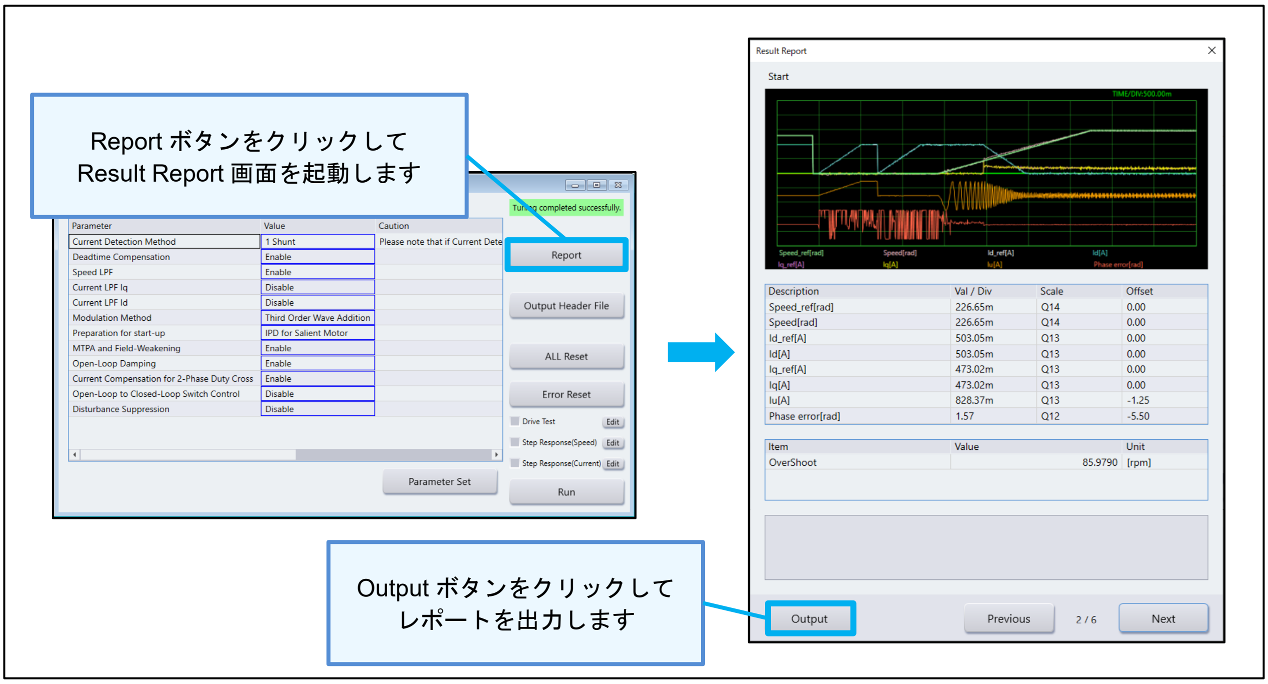


図 15-21 Result Report 画面(駆動テスト結果)

15.4.5.2 ヘッダファイル出力

設計したパラメータをヘッダファイルとして出力します。

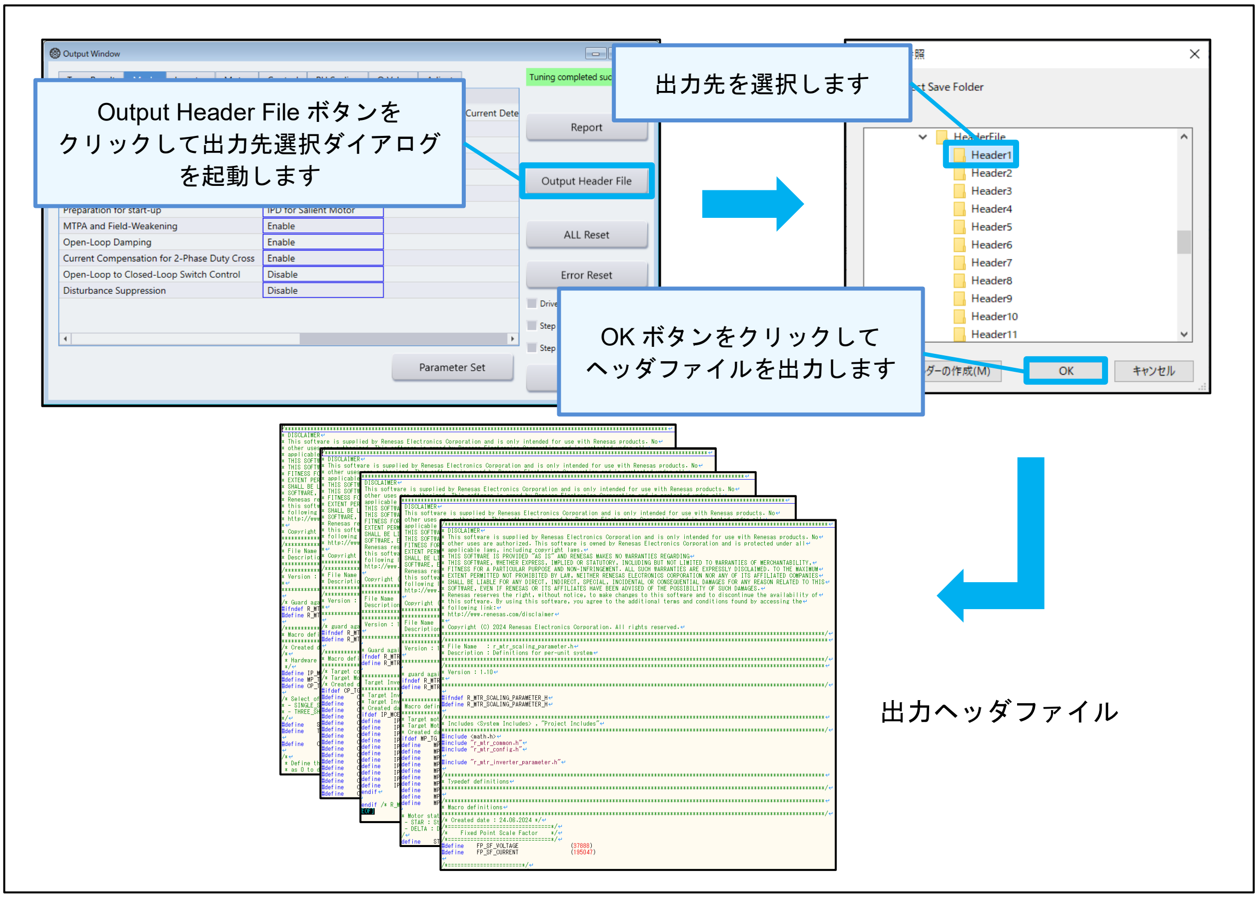


図 15-22 ヘッダファイル出力

15.4.6 Tuner ツールの終了

モータをテスト駆動している場合は、「STOP」ボタンでテストを終了します。
その後「Main Window」ボタンで Main Window に戻るか、またはツール切り替えボタンの操作を行ってください。

15.5　機能説明

15.5.1 プロジェクトファイル管理機能（Project File 画面）

Project File 画面はプロジェクトファイル(.apmpj)の保存と読み取りを行います。プロジェクトファイルは Input Window と Output Window に表示されているパラメータ、駆動テスト結果を保持します。Project File 画面は、Main Window 画面の「Select Tool」から「Tuner」を選択すると表示され、機能実行中は常時表示されます。

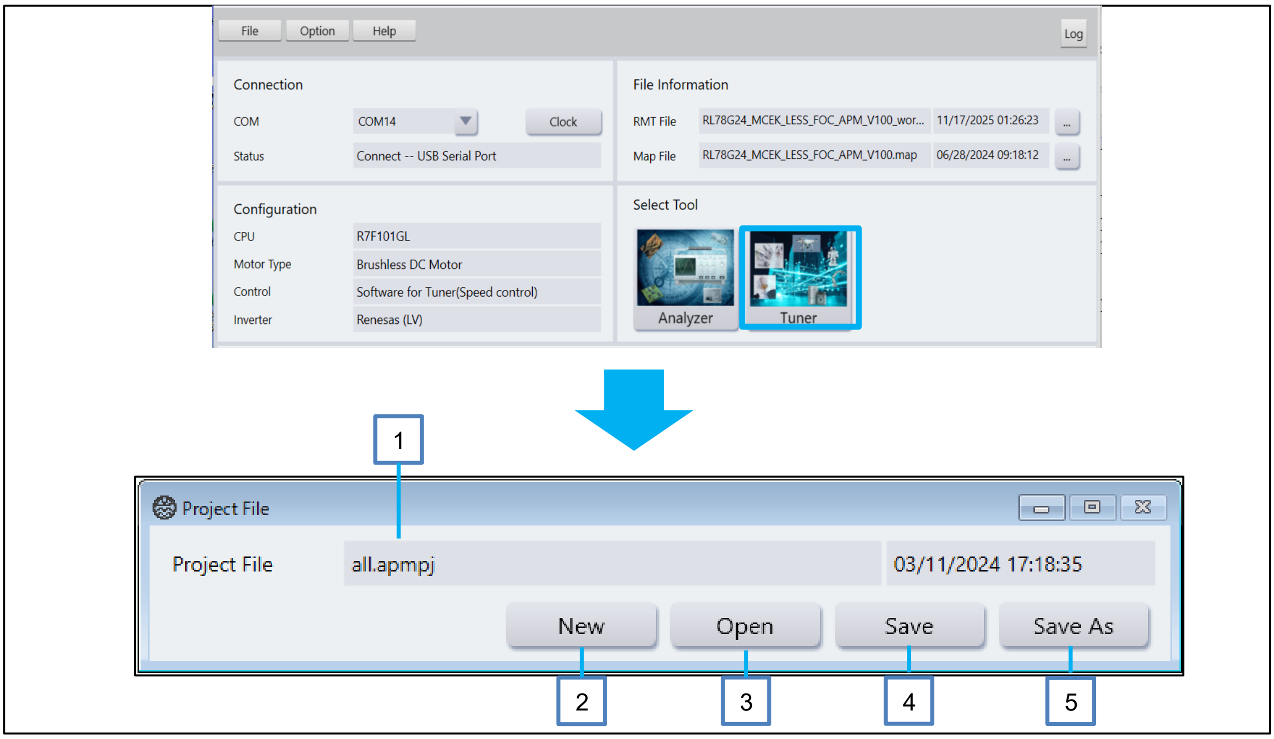


図 15-23 Project File 画面

表 15-3　Project File 画面の項目一覧

No	名称	説明
1	Project File	選択中のプロジェクトファイル名、更新日時を表示します。
2	New	プロジェクトファイルを新規作成します。
3	Open	プロジェクトファイルを読み込みます。
4	Save	選択中のプロジェクトファイルを上書き保存します。
5	Save As	プロジェクトファイルの名前を付けて保存をします。

15.5.2 パラメータ入力機能（Input Window 画面）

Input Window 画面は、Main Window 画面の「Select Tool」から「Tuner」を選択すると表示されます。パラメータ入力機能では、モータ・インバータの固有パラメータを設定し、「Start」ボタンをクリックして自動測定を開始します。青枠で囲われたパラメータの値は編集可能です。

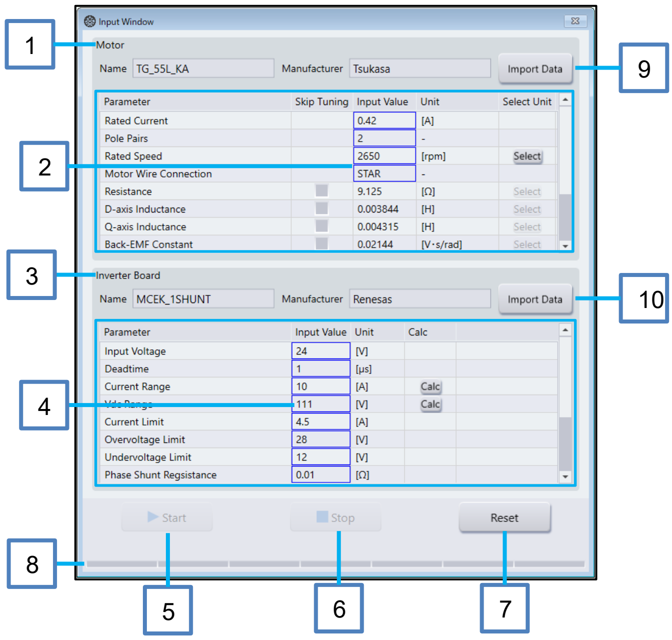


図 15-24 Input Window 画面

※現行のバージョンでは、インバータの固有パラメータはデフォルト値のみ使用可能であり、値の変更には対応しておりません。

表 15-4　Input Window 画面の項目一覧

<table>
<tr><th>No</th><th>名称</th><th>説明</th></tr>
<tr><td>1</td><td>モータ情報</td><td>モータ情報の情報を入力、表示します。
<table><tr><th>項目名</th><th>説明</th><th>備考</th></tr><tr><td>Motor Name</td><td>モータ名</td><td>-</td></tr><tr><td>Motor Manufacturer</td><td>モータメーカ名</td><td>-</td></tr></table></td></tr>
<tr><td>2</td><td>モータ入力パラメータ</td><td>自動測定に必要なモータのパラメータの設定を行います。
<table><tr><th>項目名</th><th>説明</th><th>備考</th></tr><tr><td>Parameter</td><td>パラメータ名称</td><td>-</td></tr><tr><td>Skip Tuning</td><td>対象パラメータを自動測定しない場合にチェックします。</td><td>-</td></tr><tr><td>Input Value</td><td>自動測定しない場合の設定値を入力します。</td><td>設定値に設けている上下限値をオーバーした場合、不正な値として赤枠を表示します。</td></tr><tr><td>Unit</td><td>単位名</td><td>-</td></tr><tr><td>Select Unit ボタン</td><td>Select ボタンクリックで Unit Conversion 画面を表示します。</td><td>-</td></tr></table></td></tr>
<tr><td>3</td><td>インバータボード情報</td><td>インバータボード情報の情報を入力、表示します。
<table><tr><th>項目名</th><th>説明</th><th>備考</th></tr><tr><td>Inverter Name</td><td>インバータボード名</td><td>-</td></tr><tr><td>Inverter Manufacturer</td><td>インバータボードメーカ名</td><td>-</td></tr></table></td></tr>
<tr><td>4</td><td>インバータボード入力パラメータ</td><td>自動測定に必要なインバータボードのパラメータの設定を行います。
<table><tr><th>項目名</th><th>説明</th><th>備考</th></tr><tr><td>Parameter</td><td>パラメータ名称</td><td>-</td></tr><tr><td>Input Value</td><td>設定値</td><td>設定値に設けている上下限値をオーバーした場合、不正な値として赤枠を表示します。</td></tr><tr><td>Unit</td><td>単位名</td><td>-</td></tr><tr><td>Calc ボタン</td><td>Calc ボタンクリックで計算画面を表示します。</td><td>-</td></tr></table></td></tr>
<tr><td>5</td><td>Start ボタン</td><td>自動測定を開始します。</td></tr>
<tr><td>6</td><td>Stop ボタン</td><td>自動測定を停止します。</td></tr>
<tr><td>7</td><td>Reset ボタン</td><td>ボタンクリックで CPU ボードに「Reset」コマンドを送信します。
CPU ボードから状態を取得し、エラーの場合に表示されます。</td></tr>
<tr><td>8</td><td>プログレスバー</td><td>CPU ボードでの自動測定中に進捗表示を行います。</td></tr>
<tr><td>9</td><td>Import Data(Motor)</td><td>保存されているプロジェクトファイルのモータ情報を読み込み、モータ入力パラメータの設定値に反映します。</td></tr>
<tr><td>10</td><td>Import Data(Inverter)</td><td>保存されているプロジェクトファイルのインバータボード情報を読み込み、インバータボード入力パラメータの設定値に反映します。</td></tr>
</table>

15.5.2.1 保存パラメータ読み込み機能

「Import Data」ボタンをクリックし、保存されているプロジェクトファイルを選択することで、保存パラメータを読み込みます。

「Motor」の「Import Data」でファイルを選択した場合、「Motor」のパラメータのみプロジェクトファイルの設定値に書き換わります。「Inverter Board」の「Import Data」でファイルを選択した場合、「Inverter Board」のパラメータのみプロジェクトファイルの設定値に書き換わります。

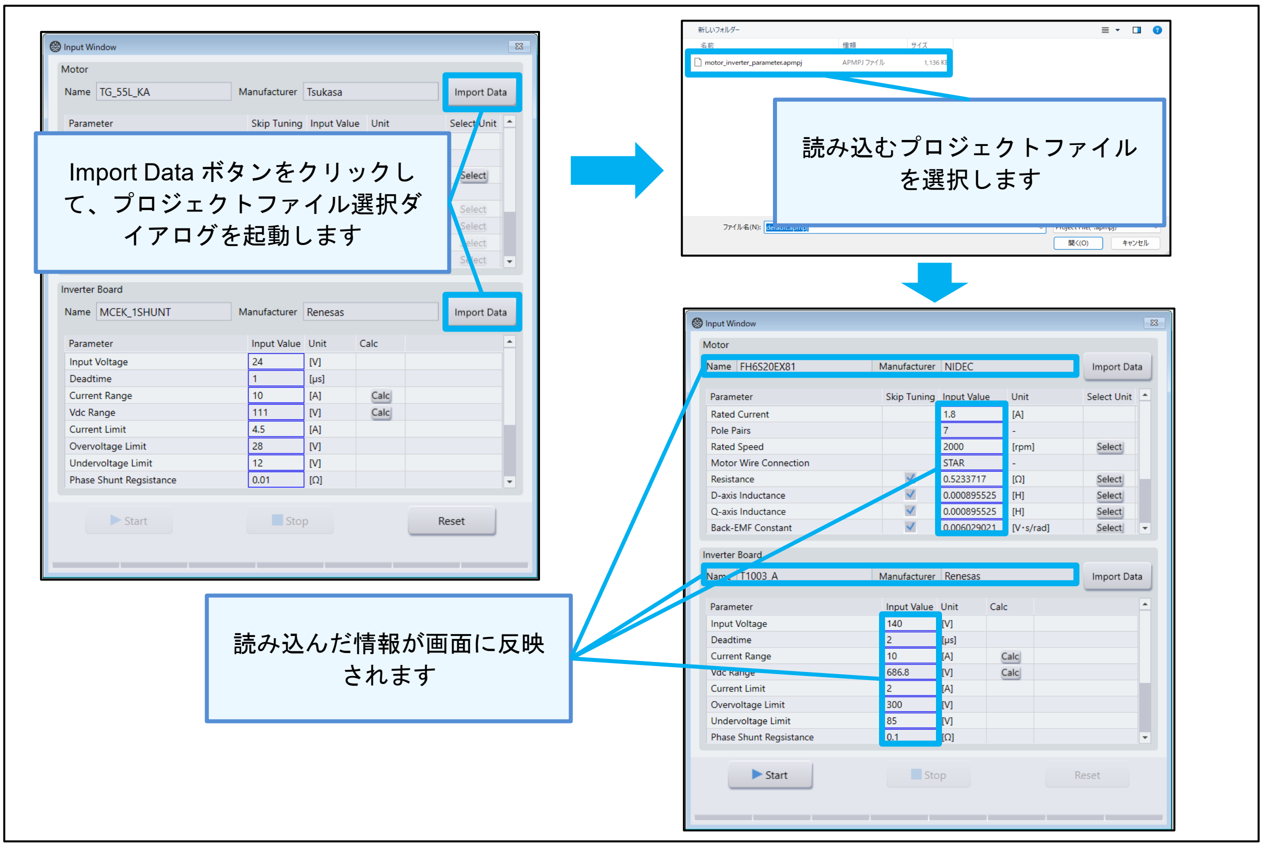


図 15-25 保存パラメータ読み込み

15.5.2.2 単位変換機能

「モータの固有パラメータ」は、別の単位系からルネサス製制御プログラムで使用している単位へ変換することができます。

Unit Conversion 画面では、変換を行った値を「モータ入力パラメータ」のパラメータ値として入力します。青枠表示された項目は、編集可能となります。「Set」ボタンをクリックすると、各画面に値を反映します。

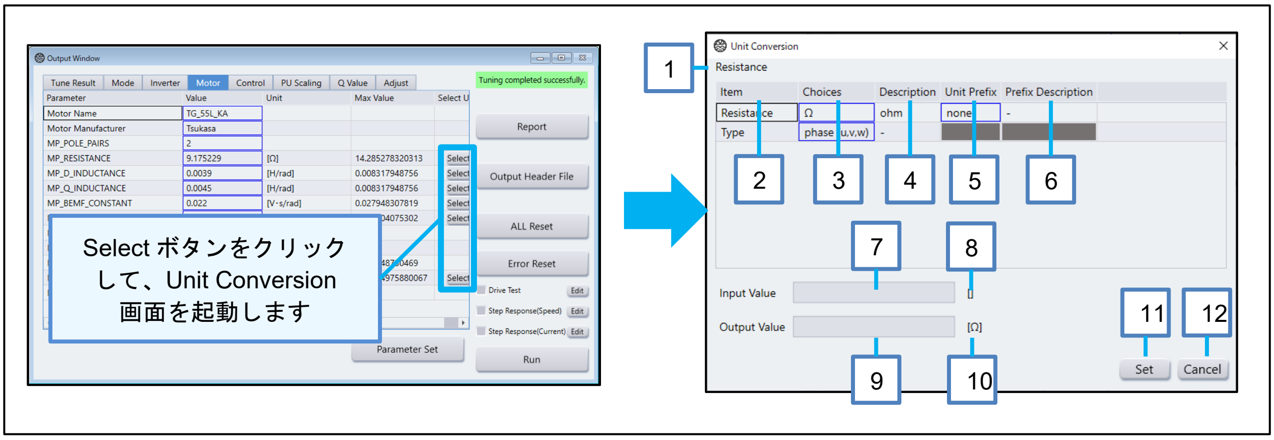


図 15-26 Unit Conversion 画面

表 15-5 Unit Conversion 画面の項目一覧

No	名称	説明	備考
1	入力パラメータ名	入力パラメータ名を表示します。	-
2	Item	項目を表示します。	-
3	Choices	単位を選択します。	-
4	Description	選択した項目の説明を表示します。	-
5	Unit Prefix	補助単位を選択します。	-
6	Prefix Description	選択した補助単位の説明を表示します。	-
7	Input Value	単位変換前の数値を入力します。	0 未満の値の場合、不正な値として赤枠を表示します。
8	入力単位	入力単位を表示します。	-
9	Output Value	単位変換後の数値を表示します。	-
10	出力単位	出力単位を表示します。	-
11	Set ボタン	単位変換結果を各画面へ反映し、Unit Conversion 画面を閉じます。	-
12	Cancel ボタン	Unit Conversion 画面を閉じます。	-

15.5.2.3 スケーリング値計算機能

インバータ上の検出回路情報から、電流へのスケーリング値「IP_CURRENT_RANGE」、電圧へのスケーリング値「IP_VDC_RANGE」を設計することができます。

(1) 電流スケーリング値計算機能

Current Range Calculation 画面では、シャント抵抗値 Rs、増幅率 G、電圧基準値の 3 つの項目から以下の式(1)でスケーリング値を算出します。「Set」ボタンをクリックすると、各画面に値を反映します。

電流スケーリング値[A] ＝ 電圧基準値[V] / (シャント抵抗値 Rs * 増幅率 G) …(1)

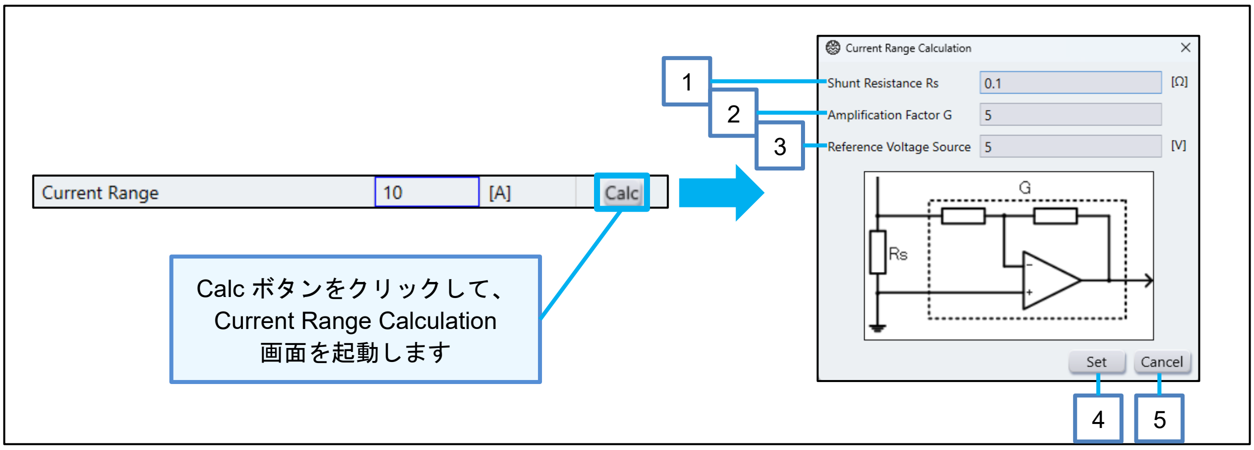


図 15-27 Current Range Calculation 画面

表 15-6 Current Range Calculation 画面の項目一覧

No	名称	説明	備考
1	Shunt Resistance Rs	対象となるインバータのシャント抵抗値を入力します。	0 以下の値の場合、不正な値として赤枠を表示します。
2	Amplification Factor G	電流検出アンプ増幅率を入力します。	0 以下の値の場合、不正な値として赤枠を表示します。
3	Reference Voltage Source	MCU 側の AD 検出基準電圧値を入力します。	0 以下の値の場合、不正な値として赤枠を表示します。
4	Set ボタン	計算結果を各画面へ反映し、Current Range Calculation 画面を閉じます。	-
5	Cancel ボタン	Current Range Calculation 画面を閉じます。	-

(2) 電圧スケーリング値計算機能

Voltage Range Calculation 画面では、分圧抵抗 R1、R2、電圧基準値の３つの項目から以下の式(2)でスケーリング値を算出します。「Set」ボタンをクリックすると、各画面に値を反映します。

電圧スケーリング値[V] ＝ 電圧基準値[V] * (R1＋R2) / R2 …(2)

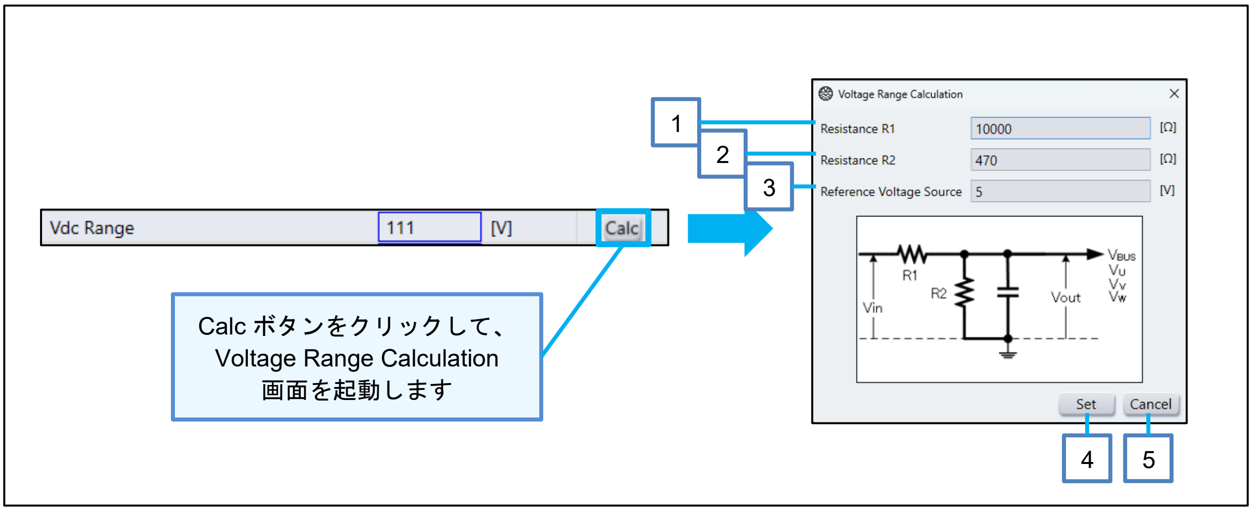


図 15-28 Voltage Range Calculation 画面

表 15-7 Voltage Range Calculation 画面の項目一覧

No	名称	説明	備考
1	Resistance R1	対象となるインバータの分圧抵抗値を入力します。	0 以下の値の場合、不正な値として赤枠を表示します。
2	Resistance R2	対象となるインバータの分圧抵抗値を入力します。	0 以下の値の場合、不正な値として赤枠を表示します。
3	Reference Voltage Source	MCU 側の AD 検出基準電圧値を入力します。	0 以下の値の場合、不正な値として赤枠を表示します。
4	Set ボタン	計算結果を各画面へ反映し、Voltage Range Calculation 画面を閉じます。	-
5	Cancel ボタン	Voltage Range Calculation 画面を閉じます。	-

※インバータボードの電圧・電流検出回路が上記と異なる場合は、対象のインバータボード回路に合わせて適切にスケーリング値を設定してください。

15.5.3 測定結果表示機能（Output Window 画面）

パラメータの自動測定結果を表示し、各パラメータの編集と駆動テストを行います。

Output Window 画面起動時には、「Tune Result」(測定結果)、「Mode」(制御モード選択)、「Inverter」(インバータの固有パラメータ)、「Motor」(モータの固有パラメータ)、「Control」(制御パラメータ)、「PU Scaling」(スケーリング値)、「Q Value」(Q 値)、「Adjust」(制御パラメータの調整値)の各種パラメータが表示され、パラメータの編集が可能です。編集可能なパラメータは青枠表示されます。

また、「Mode」パラメータの選択内容によって各タブのパラメータの有効/無効が切り変わります。無効化されたパラメータはグレーアウトされます。

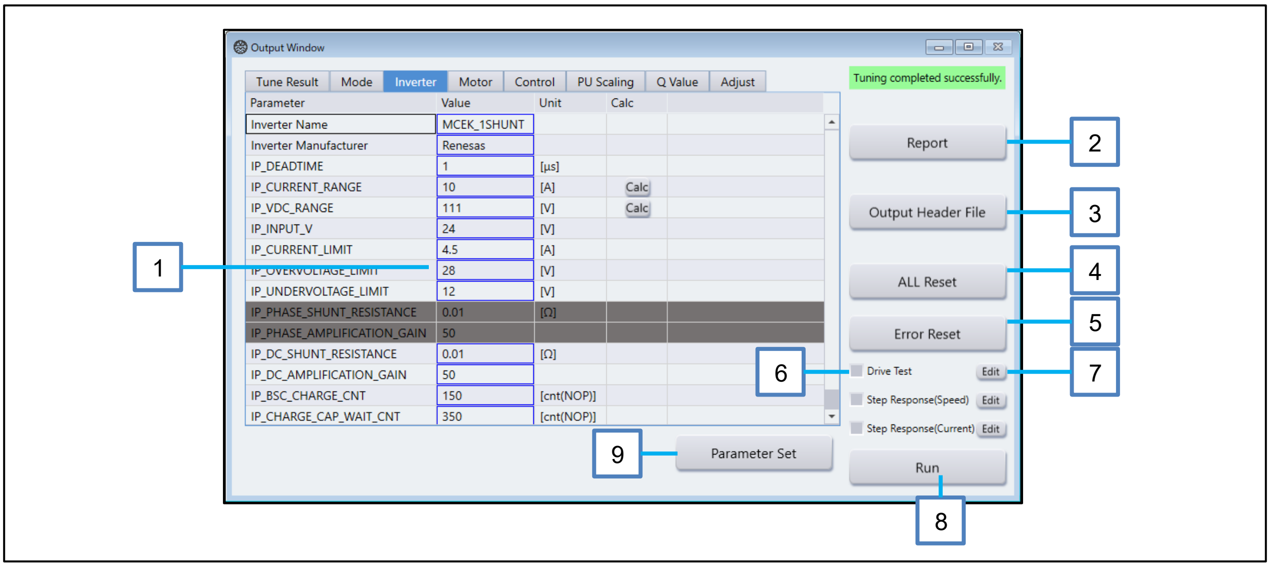


図 15-29 Output Window 画面

表 15-8　Output Window 画面の項目一覧

<table>
<tr><th>No</th><th>名称</th><th>説明</th></tr>
<tr><td>1</td><td>パラメータ</td><td>「Tune Result」、「Mode」、「Inverter」、「Motor」、「Control」、「PU Scaling」、「Q Value」、「Adjust」の各タブに各種パラメータが表示され、パラメータの編集が可能です。
※編集可能な欄は青枠表示されます。</td></tr>
<tr><td>2</td><td>Report ボタン</td><td>Result Report 画面を表示し、設定したパラメータ、駆動テスト結果を出力します。</td></tr>
<tr><td>3</td><td>Output Header File ボタン</td><td>設定したパラメータをヘッダファイルとして出力します。フォルダ選択ダイアログ表示し、拡張子「.h」のヘッダファイルを保存します。</td></tr>
<tr><td>4</td><td>ALL Reset ボタン</td><td>Tuner 機能の計測データをリセットします。</td></tr>
<tr><td>5</td><td>Error Reset ボタン</td><td>Tuner 機能のエラーをリセットします。</td></tr>
<tr><td>6</td><td>駆動モード</td><td>駆動テストを実施する駆動モードを選択します。
駆動モードは以下です。
<table>
<tr><th>項目名</th><th>説明</th></tr>
<tr><td>Drive Test</td><td>始動＋定常駆動</td></tr>
<tr><td>Step Response(Speed)</td><td>ステップ応答(速度)</td></tr>
<tr><td>Step Response(Current)</td><td>ステップ応答(電流)</td></tr>
</table></td></tr>
<tr><td>7</td><td>Edit ボタン</td><td>シーケンス編集画面を表示し、駆動テストで使用するシーケンスを編集します。
編集内容は以下です。
<table>
<tr><th>項目名</th><th>説明</th></tr>
<tr><td>Drive Test</td><td>目標速度、加速度、駆動速度</td></tr>
<tr><td>Step Response(Speed)</td><td>変化前速度、変化後速度</td></tr>
<tr><td>Step Response(Current)</td><td>変化前電流、変化後電流</td></tr>
</table>
駆動テストシーケンス編集機能の詳細は、「15.5.4 駆動テストシーケンス編集機能（Sequence Edit 画面）」を参照してください。</td></tr>
<tr><td>8</td><td>駆動テスト実行・停止ボタン</td><td>「Run」表示中は、Scope 画面を表示し、駆動テストを実施します。
駆動テスト実行中は「Stop」表示とし、駆動テストの停止が可能です。</td></tr>
<tr><td>9</td><td>Parameter Set ボタン</td><td>各タブで調整したパラメータを CPU ボードに送信します。</td></tr>
</table>

15.5.3.1　Tune Result タブ

Tune Result タブでは、Input Window 画面で自動測定されたパラメータの測定結果を表示します。このタブでは測定結果を表示するため、パラメータは編集不可能となっています。

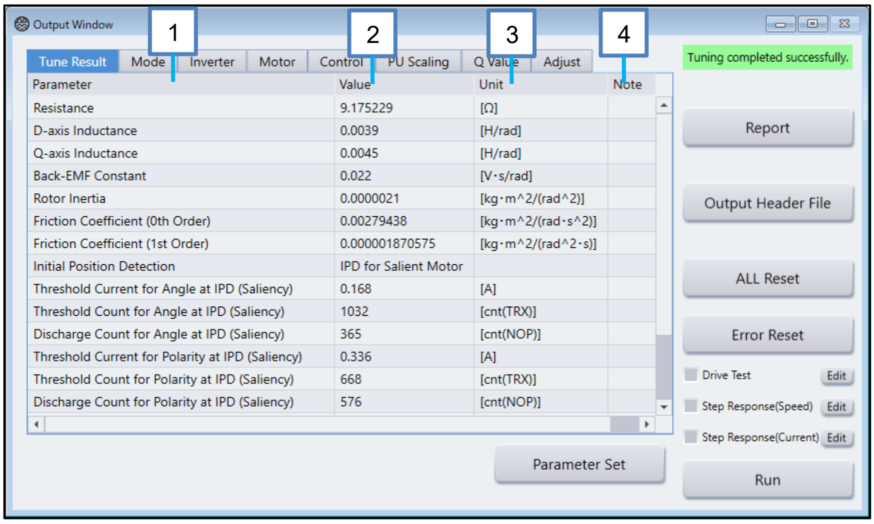


図 15-30 Tune Result タブ

表 15-9　Tune Result タブの項目一覧

No	名称	説明
1	Parameter	パラメータ名を表示します。
2	Value	パラメータ値を表示します。
3	Unit	パラメータの単位を表示します。
4	Note	備考を表示します。 Manual Input : 自動測定をスキップしたことを示します。 Calculated : 計算でパラメータを設計したことを示します。

15.5.3.2 Mode タブ

Mode タブでは各種制御モードを選択します。この設定は、「Output Header File」ボタンで出力する「r_mtr_config.h」に反映されます。

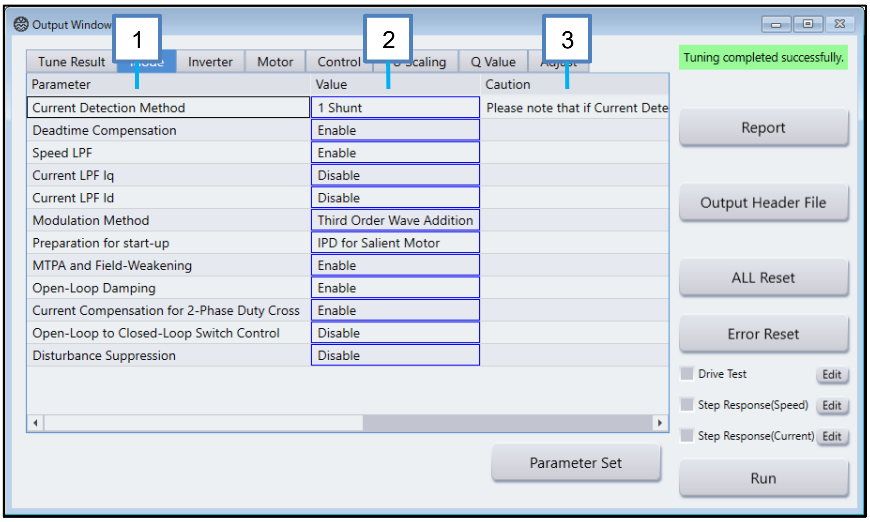


図 15-31 Mode タブ

表 15-10 Mode タブの項目一覧

No	名称	説明
1	Parameter	パラメータ名を表示します。
2	Value	パラメータ値を選択します。
3	Caution	注意事項を表示します。

15.5.3.3 Inverter タブ

Inverter タブでは、インバータボードの固有パラメータを設定します。この設定は、「Output Header File」ボタンで出力する「r_mtr_inverter_parameter.h」に反映されます。

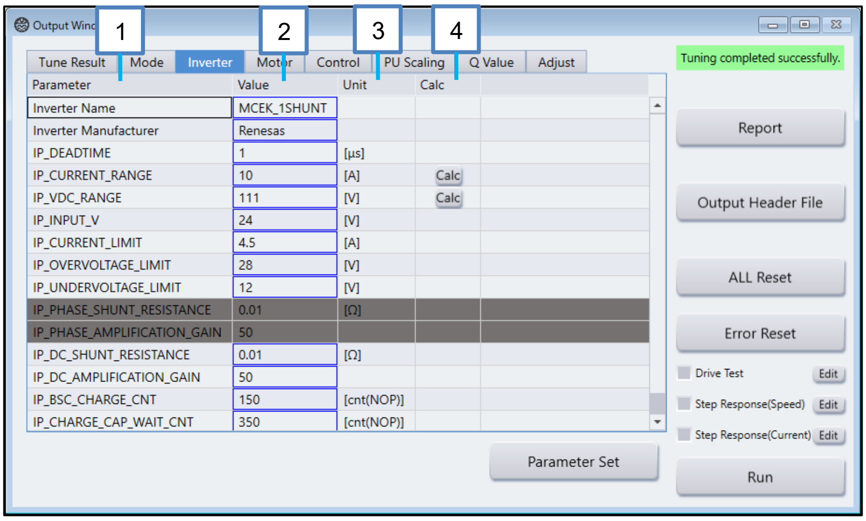


図 15-32 Inverter タブ

表 15-11 Inverter タブの項目一覧

No	名称	説明	備考
1	Parameter	編集するパラメータ名を表示します。	-
2	Value	パラメータ値を入力します。	設定値に設けている上下限値をオーバーした場合、不正な値として赤枠を表示します。 また、設定値が変数の型に一致しない場合も同様に赤枠を表示します。
3	Unit	パラメータの単位を表示します。	-
4	Calc	Calc ボタンクリックで計算画面を表示します。	-

15.5.3.4 Motor タブ

Motor タブでは、モータの固有パラメータを設定します。この設定は、「Output Header File」ボタンで出力する「r_mtr_motor_parameter.h」に反映されます。

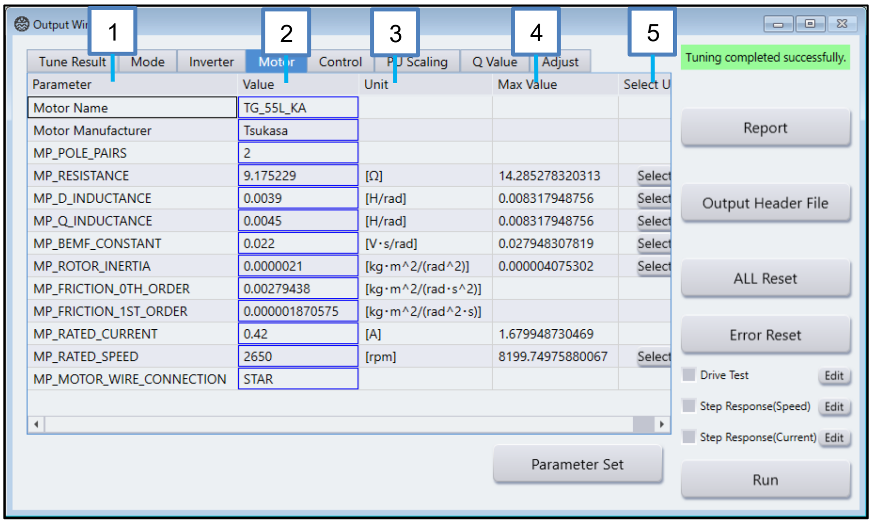


図 15-33 Motor タブ

表 15-12 Motor タブの項目一覧

No	名称	説明	備考
1	Parameter	編集するパラメータ名を表示します。	-
2	Value	パラメータ値を入力します。	設定値に設けている上下限値をオーバーした場合、不正な値として赤枠を表示します。 また、設定値が変数の型に一致しない場合も同様に赤枠を表示します。
3	Unit	パラメータの単位を表示します。	-
4	Max Value	パラメータ最大値を表示します。	-
5	Select Unit	Select ボタンクリックで Unit Conversion 画面を表示します。	-

15.5.3.5 Control タブ

Control タブでは、制御パラメータを設定します。この設定は、「Output Header File」ボタンで出力する「r_mtr_control_parameter.h」に反映されます。

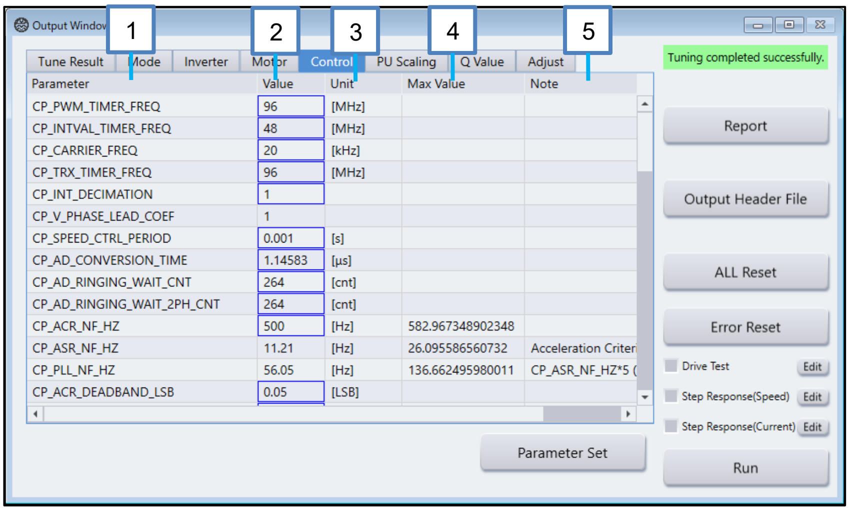


図 15-34 Control タブ

表 15-13　Control タブの項目一覧

No	名称	説明	備考
1	Parameter	編集するパラメータ名を表示します。	-
2	Value	パラメータ値を入力します。	設定値に設けている上下限値をオーバーした場合、不正な値として赤枠を表示します。 また、設定値が変数の型に一致しない場合も同様に赤枠を表示します。
3	Unit	パラメータの単位を表示します。	-
4	Max Value	パラメータ最大値を表示します。	設定値に設けている上下限値をオーバーした場合、不正な値として赤枠を表示します。 また、設定値が変数の型に一致しない場合も同様に赤枠を表示します。
5	Note	パラメータ値の計算式および関連する Adjust タブの項目番号を表示します。	-

15.5.3.6 PU Scaling タブ

PU Scaling タブでは、スケーリングパラメータを表示します。スケーリングパラメータはこれまでに設定されたインバータボードとモータの固有パラメータ及び制御パラメータから自動で計算されます。この設定は、「Output Header File」ボタンで出力する「r_mtr_scaling_parameter.h」に反映されます。

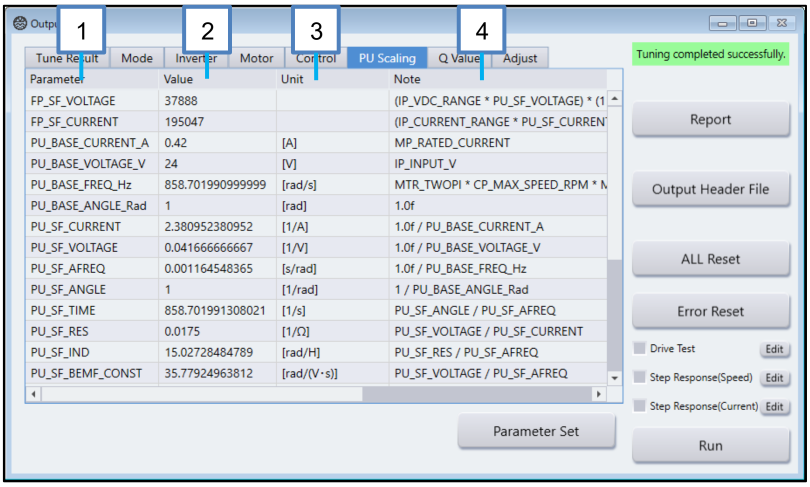


図 15-35 PU Scaling タブ

表 15-14 PU Scaling タブの項目一覧

No	名称	説明	備考
1	Parameter	編集するパラメータ名を表示します。	-
2	Value	パラメータ値を表示します。	設定値に設けている上下限値をオーバーした場合、不正な値として赤枠を表示します。 また、設定値が変数の型に一致しない場合も同様に赤枠を表示します。
3	Unit	パラメータの単位を表示します。	-
4	Note	パラメータ値の計算式を表示します。	-

15.5.3.7 Q Value タブ

Q Value タブでは、Q 値を設定します。16bit 変数に値が収まるように Q 値を自動で設計しています。値を変更したい場合は、「User Input」にチェックを入れることで変更可能になります。この設定は、「Output Header File」ボタンで出力する「r_mtr_scaling_parameter.h」に反映されます。

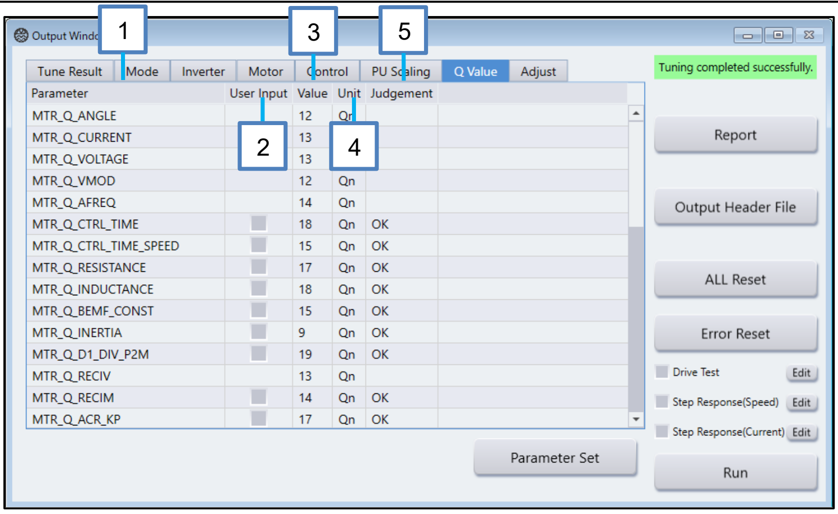


図 15-36 Q Value タブ

表 15-15 Q Value タブの項目一覧

No	名称	説明	備考
1	Parameter	編集するパラメータ名を表示します。	-
2	User Input	パラメータ入力を行う場合は、チェックします。	-
3	Value	パラメータ値を入力します。	設定値に設けている上下限値をオーバーした場合、不正な値として赤枠を表示します。 また、設定値が変数の型に一致しない場合も同様に赤枠を表示します。
4	Unit	パラメータの単位を表示します。	-
5	Judgement	入力パラメータの入力値判定結果を表示します。	-

15.5.3.8　Adjust タブ

Adjust タブでは、調整パラメータの入力を行います。Control タブの対応するパラメータを調整することができます。

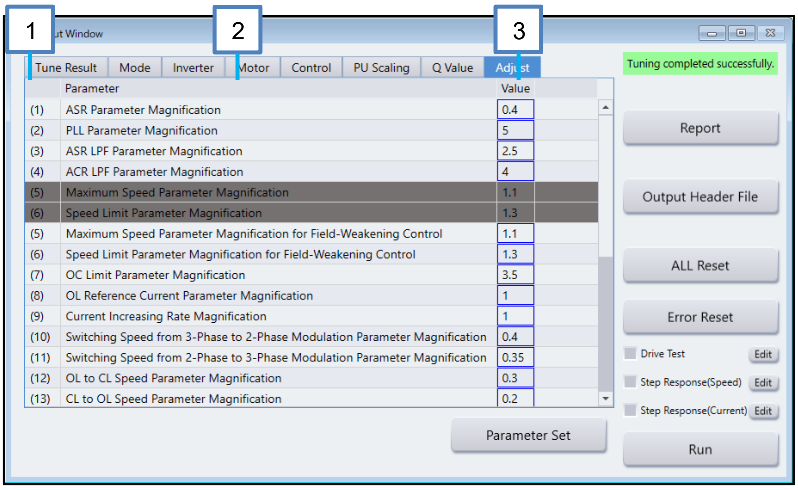


図 15-37 Adjust タブ

表 15-16　Adjust タブの項目一覧

No	名称	説明	備考
1	-	項目番号を表示します。	Control タブの Note に、対応する項目番号が記載されます。
2	Parameter	編集するパラメータ名を表示します。	-
3	Value	パラメータ値を入力します。	設定値に設けている上下限値をオーバーした場合、不正な値として赤枠を表示します。 また、設定値が変数の型に一致しない場合も同様に赤枠を表示します。

15.5.4 駆動テストシーケンス編集機能（Sequence Edit 画面）

駆動テストシーケンス編集機能では、駆動テストに使用するパラメータのシーケンスの編集を行います。デフォルト設定では、自動で計算された値(Calc Value)が使用されます。User Input にチェックを入れて、User Input Value に値を入力することで、任意の値をシーケンス値として設定することが可能です。

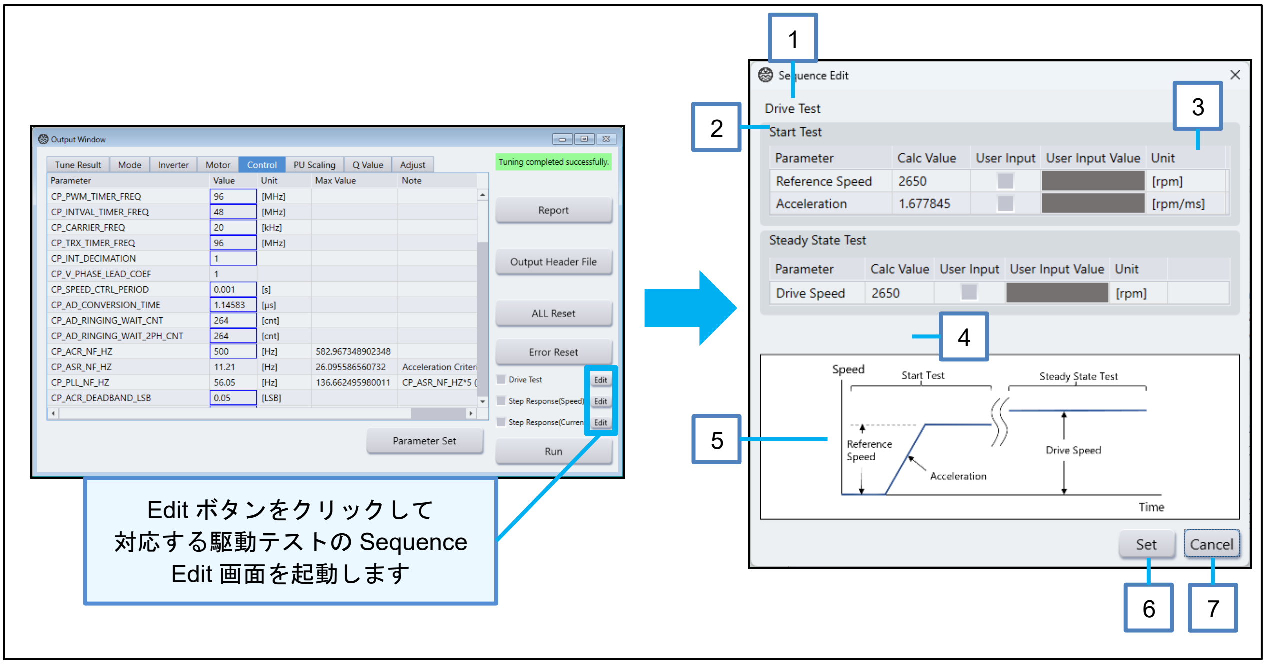


図 15-38 Sequence Edit 画面

表 15-17　Sequence Edit 画面の項目一覧

<table>
<tr><th>No</th><th>名称</th><th>説明</th></tr>
<tr><td>1</td><td>駆動テスト名</td><td>シーケンスを編集する対象の駆動テスト名を表示します。</td></tr>
<tr><td>2</td><td>駆動モード</td><td>対象の駆動テスト名を表示します。</td></tr>
<tr><td>3</td><td>シーケンスパラメータ</td><td>シーケンスパラメータの表示、編集を行います。
<table>
<tr><th>項目名</th><th>説明</th><th>備考</th></tr>
<tr><td>Parameter</td><td>編集するパラメータ名を表示します。</td><td>-</td></tr>
<tr><td>Calc Value</td><td>パラメータの計算値を表示します。パラメータの入力を行わない場合、計算値がパラメータ値として使用されます。</td><td>不正な値の場合、赤枠を表示します。</td></tr>
<tr><td>User Input</td><td>パラメータ入力を行う場合、チェックします。</td><td>-</td></tr>
<tr><td>User Input Value</td><td>パラメータ値を入力します。</td><td>不正な値の場合、赤枠を表示します。</td></tr>
<tr><td>Unit</td><td>パラメータの単位を表示します。</td><td></td></tr>
</table>
</td></tr>
<tr><td>4</td><td>エラーメッセージ</td><td>パラメータ値が不正な値の場合、エラーメッセージを表示します。</td></tr>
<tr><td>5</td><td>イメージ画像</td><td>駆動テスト波形におけるシーケンスパラメータの対応図を表示します。</td></tr>
<tr><td>6</td><td>Set ボタン</td><td>編集したシーケンスパラメータを反映し、Sequence Edit 画面を閉じます。</td></tr>
<tr><td>7</td><td>Cancel ボタン</td><td>Sequence Edit 画面を閉じます。</td></tr>
</table>

15.5.5 駆動テスト機能（Scope 画面）

設定したパラメータを使用して、駆動テストを実施します。実施できる駆動テストの種類は、始動、定常駆動、速度ステップ応答、電流ステップ応答です。

Scope 画面起動時には、Output Window 画面で選択された駆動モードのタブが表示され、駆動テストが実施されます。

※駆動テスト時にモータが正常に駆動しない場合があります。モータの駆動に異常を感じたら、すぐに Output Window 画面の「Stop」ボタンを押下してモータを停止してください。

※モータの始動時間が長い場合、始動テストの波形が時間スケールで収まらず、後続の駆動テストも適切に実行されない場合があります。また、駆動テストを 2 回以上実行する場合、初回実行時と同じ縦軸のスケールとオフセットで波形が表示されます。

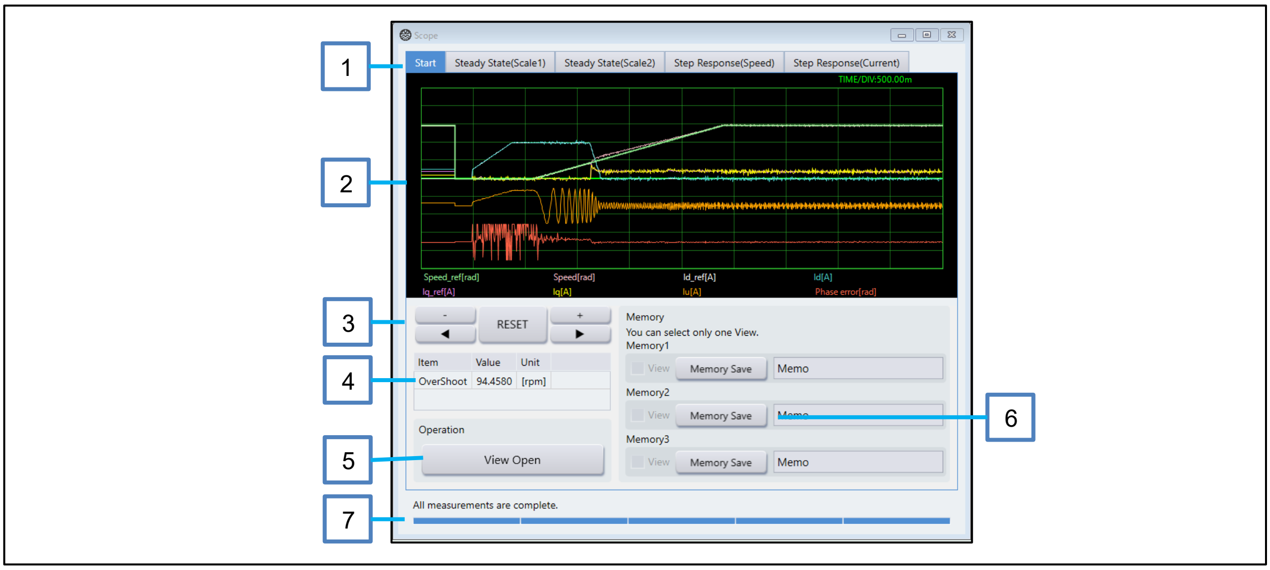


図 15-39 Scope 画面

表 15-18　Scope 画面の項目一覧

No	名称	説明
1	駆動モードタブ	Output Window 画面で選択された駆動モードのタブを表示します。 タブ選択により該当駆動モードの駆動テスト結果を表示します。 駆動モードのタブは以下です。 Start: 　モータ始動時の波形を表示します。 Steady State(Scale1): 　モータ定常状態の波形を速度制御の固有周波数に合わせたスケールで表示します。 Steady State(Scale2): 　モータ定常状態の波形を電流制御の固有周波数に合わせたスケールで表示します。 Step Response(Speed): 　モータ速度をステップ変化させた波形を表示します。 Step Response(Current): 　モータ電流をステップ変化させた波形を表示します。
2	波形データ	駆動テスト結果の波形表示を行います。制御パラメータの値に応じて、適切な時間スケーリングで表示します。 グラフの縦軸のスケーリングはズーム機能で変更可能です。 各駆動テストの測定が完了したタイミングで、駆動波形が表示されます。
3	ズーム機能	駆動テスト結果の波形の拡大/収縮とレンジの移動を行います。
4	算出値表示	駆動テストの結果を基に指標となる値を表示します。表示項目は以下です。 Start： 　オーバーシュート Steady State(Scale1)： 　リプル、定常偏差 Steady State(Scale2)： 　リプル、定常偏差 Step Response(Speed): 　オーバーシュート、整定時間、立ち上がり時間 Step Response(Current): 　オーバーシュート、整定時間、立ち上がり時間 scale1:速度制御の固有周波数に合わせたスケール scale2:電流制御の固有周波数に合わせたスケール
5	View Open ボタン	Scope Channel Info 画面を表示します。
6	波形保存機能	最大 3 つまで表示中の波形を保存します。
7	プログレスバー	駆動テスト中の進捗を表示します。

15.5.5.1 ズーム機能

ズーム機能では、グラフ横軸の 1 div の時間と表示領域の変更ができます。

図 15-40 ズーム機能

表 15-19 ズーム機能の項目一覧

No	名称	説明
1	-ボタン	ズームアウトします。
2	+ボタン	ズームインします。
3	◂ボタン	左に移動します。
4	▸ボタン	右に移動します。
5	RESET ボタン	波形表示範囲を初期状態に戻します。

15.5.5.2　チャネル情報編集機能

「View Open」ボタンをクリックすると Scope Channel Info 画面を表示し、波形の表示情報を確認できます。また、設定値の編集により、グラフの縦軸スケーリングの変更が可能です。

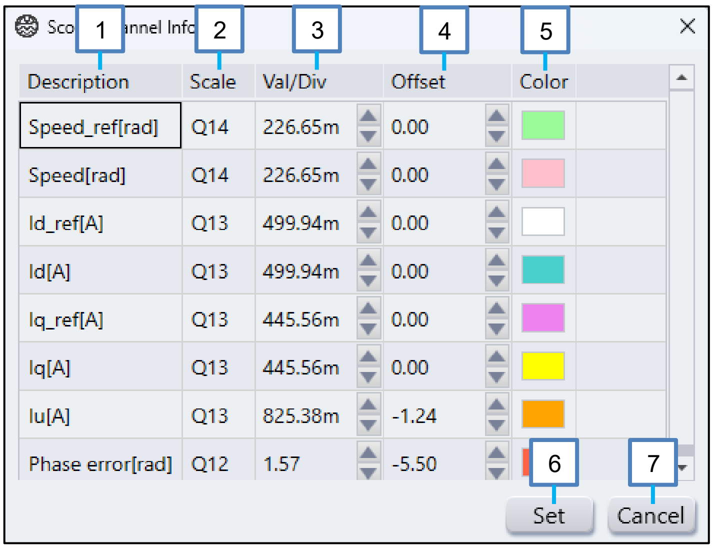


図 15-41 Scope Channel Info 画面

表 15-20　Scope Channel Info 画面の項目一覧

No	名称	説明	備考
1	Description	波形の説明を表示します。	-
2	Scale	縦軸スケールを設定します。 Q 値、およびスケール値での入力が可能です。	Q 値もしくはスケール値以外の場合、不正な値として赤枠を表示します。
3	Val/Div	1 メモリあたりの数値を設定します。	-
4	Offset	表示オフセット値を設定します。	-
5	Color	波形の色を設定します。	-
6	Set ボタン	編集したチャネル情報を反映し、Scope Channel Info 画面を閉じます。	-
7	Cancel ボタン	Scope Channel Info 画面を閉じます。	-

15.5.5.3 波形保存機能

波形保存機能では、表示中波形を最大 3 つまで保存でき、各波形を切り替えて表示することができます。

「Memory Save」ボタンをクリックすると波形を記憶し、表示が「Memory Clear」に変わります。再度クリックするとメモリをクリアします。「View」チェックボックスにチェックをすると、保存した波形が表示されます。

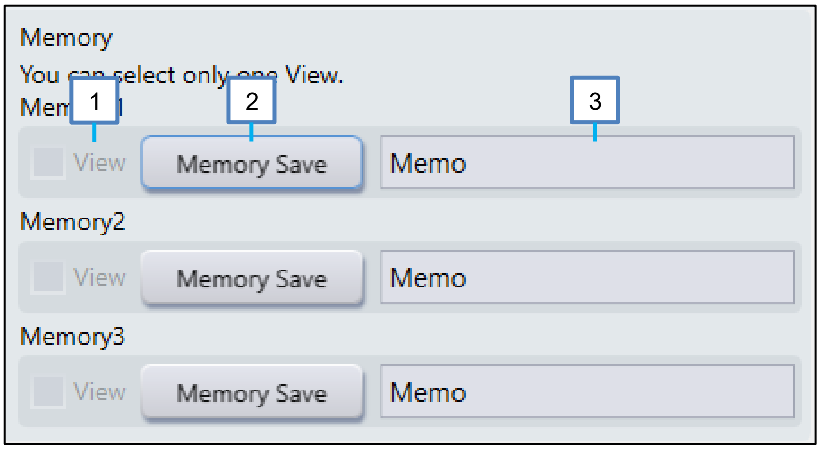


図 15-42 波形保存機能

表 15-21 波形保存項目一覧

No	名称	説明
1	View チェックボックス	保存した波形データの表示有無を選択します。
2	Memory Save ボタン	測定した波形データを保存します。「Memory Clear」ボタンクリックで、メモリクリアします。
3	Memo	メモ欄です。

15.5.6 レポート機能（Output Window 画面）

設定したパラメータ、および駆動テスト結果をレポートファイルとして出力します。

※レポートファイルにパラメータの値が適切に反映されない場合があります。正しいパラメータの値はOutput Window 画面をご確認ください。

15.5.6.1 出力内容の表示、出力パラメータの選択

Result Report 画面の 1 ページ目では出力内容の選択、出力パラメータの選択を行います。

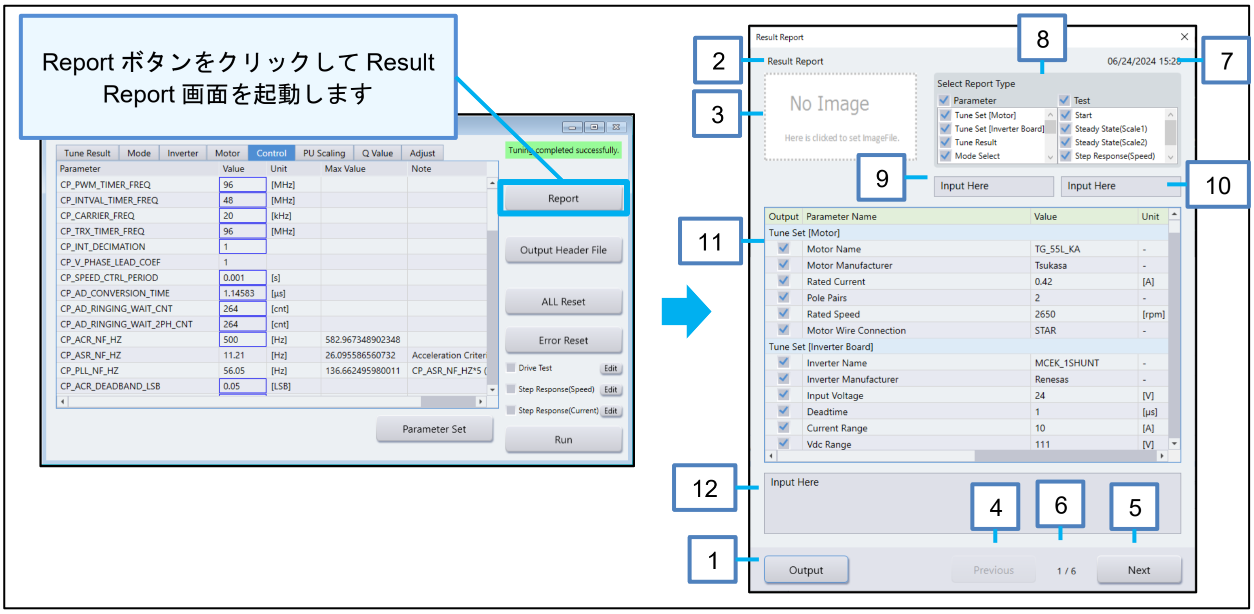


図 15-43 Result Report 画面(1 ページ目)

表 15-22 Result Report 画面(1 ページ目)の項目一覧

No	名称	説明
1	Output ボタン	PDF 形式のレポートを作成します。
2	ページタイトル表示	ページタイトルを表示します。
3	画像イメージ選択	レポートに表示する画像を選択することができます。
4	Previous ボタン	Result Report 画面の 1 ページ前へ遷移します。
5	Next ボタン	Result Report 画面の 1 ページ次へ遷移します。
6	ページ表示	現在のページ数／総ページ数を表示します。
7	レポート日付表示	レポート出力した(Report ボタンクリック時)日時を表示します。
8	出力内容選択	パラメータ、駆動テストのどの情報を出力するかチェックボックスで選択します。 パラメータは、下記分類から選択します。 - Tune Set [Motor] - Tune Set [Inverter Board] - Tune Result - Mode Select - Inverter Parameter - Motor Parameter - Control Parameter - Scaling Parameter - Adjust Parameter 駆動テストは、下記モードの駆動結果と波形保存機能で保存されている結果全てから選択します。 - Drive Test - Step Response(Speed) - Step Response(Current)
9	フリーテキスト領域 1	ユーザが自由に入力できるテキスト欄です。
10	フリーテキスト領域 2	ユーザが自由に入力できるテキスト欄です。
11	結果表示グリッド	パラメータを表示します。 各パラメータの出力可否をチェックボックスで選択可能です。
12	フリーテキスト領域 3	ユーザが自由に入力できるテキスト欄です。複数行の入力が可能です。

チューニング結果として出力されるパラメーター覧を以下に示します。

表 15-23　出力パラメーター覧

マクロ名	概要	単位
MP_RESISTANCE	抵抗	[Ω]
MP_D_INDUCTANCE	d 軸インダクタンス	[H/rad]
MP_Q_INDUCTANCE	q 軸インダクタンス	[H/rad]
MP_BEMF_CONSTANT	誘起電圧定数	[V･s/rad]
MP_ROTOR_INERTIA	イナーシャ	[kg･m^2/(rad^2)]
MP_FRICTION_0TH_ORDER	静止摩擦係数	[kg･m^2/(rad･s^2)]
MP_FRICTION_1ST_ORDER	動摩擦係数	[kg･m^2/(rad^2･s)]
CP_SAL_ANGLE_CURRENT	突極性モータの角度検出における閾値電流	[A]
CP_SAL_ANGLE_TRX_THRESHOLD	突極性モータの角度検出の TRX カウント値差分閾値	[cnt(TRX)]
CP_SAL_ANGLE_DISCHARGE	突極性モータの角度検出の放電時間	[cnt(NOP)]
CP_SAL_POLARITY_CURRENT	突極性モータの極性検出における閾値電流	[A]
CP_SAL_POLARITY_TRX_THRESHOLD	突極性モータの極性検出の TRX カウント値差分閾値	[cnt(TRX)]
CP_SAL_POLARITY_DISCHARGE	突極性モータの極性検出の放電時間	[cnt(NOP)]
CP_NON_SAL_CURRENT	非突極性モータの角度検出における閾値電流	[A]
CP_NON_SAL_TRX_THRESHOLD	非突極性モータの角度検出の TRX カウント値差分閾値	[cnt(TRX)]
CP_NON_SAL_DISCHARGE	非突極性モータの角度検出の放電時間	[cnt(NOP)]
CP_IPD_NOISE_AVOID_CNT	ノイズ回避回数	[cnt(TRX)]

15.5.6.2　駆動テスト結果表示

2 ページ目以降では、駆動テスト結果の表示を行います。

駆動テスト結果は、Scope 画面の駆動モード別結果および波形保存機能で保存された内容を表示します。

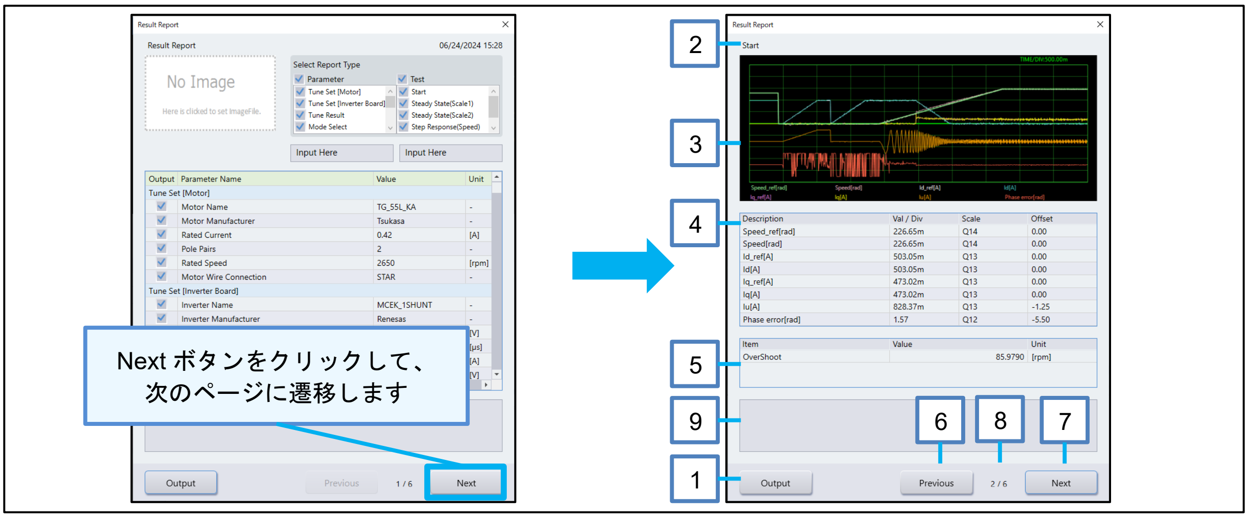


図 15-44 Result Report 画面(2 ページ目)

表 15-24　Result Report 画面(2 ページ目)の項目一覧

No	名称	説明
1	Output ボタン	PDF 形式のレポートを作成します。
2	ページタイトル表示	ページタイトルを表示します。 駆動モード、波形保存機能の「Memo」を表示します。
3	波形データ	駆動テスト結果の波形を表示します。
4	チャネル情報	駆動テスト時のチャネル情報を表示します。
5	算出値表示	駆動テスト結果で得られた算出値を表示します。
6	Previous ボタン	Result Report 画面の 1 ページ前へ遷移します。
7	Next ボタン	Result Report 画面の 1 ページ次へ遷移します。
8	ページ表示	現在のページ数／総ページ数を表示します。
9	フリーテキスト領域	ユーザが自由に入力できるテキスト欄です。複数行の入力が可能です。

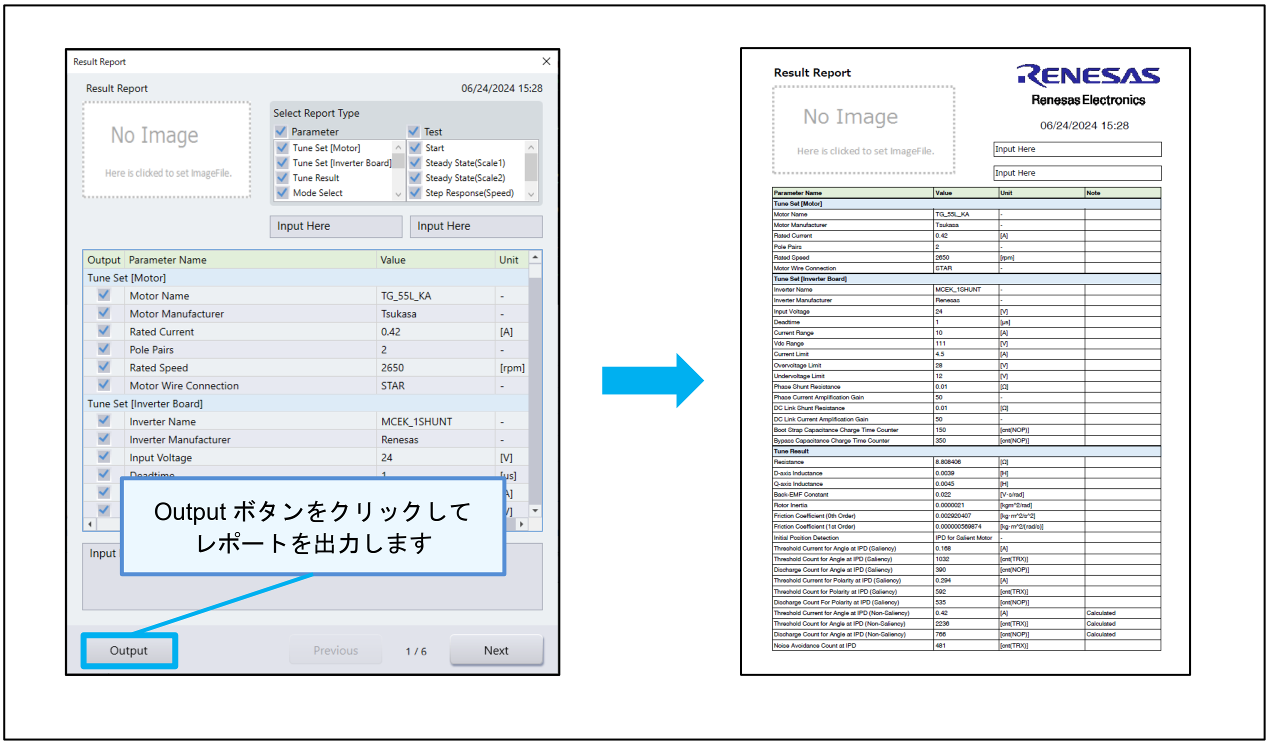


図 15-45 PDF 出力結果(設計したパラメータの出力結果)

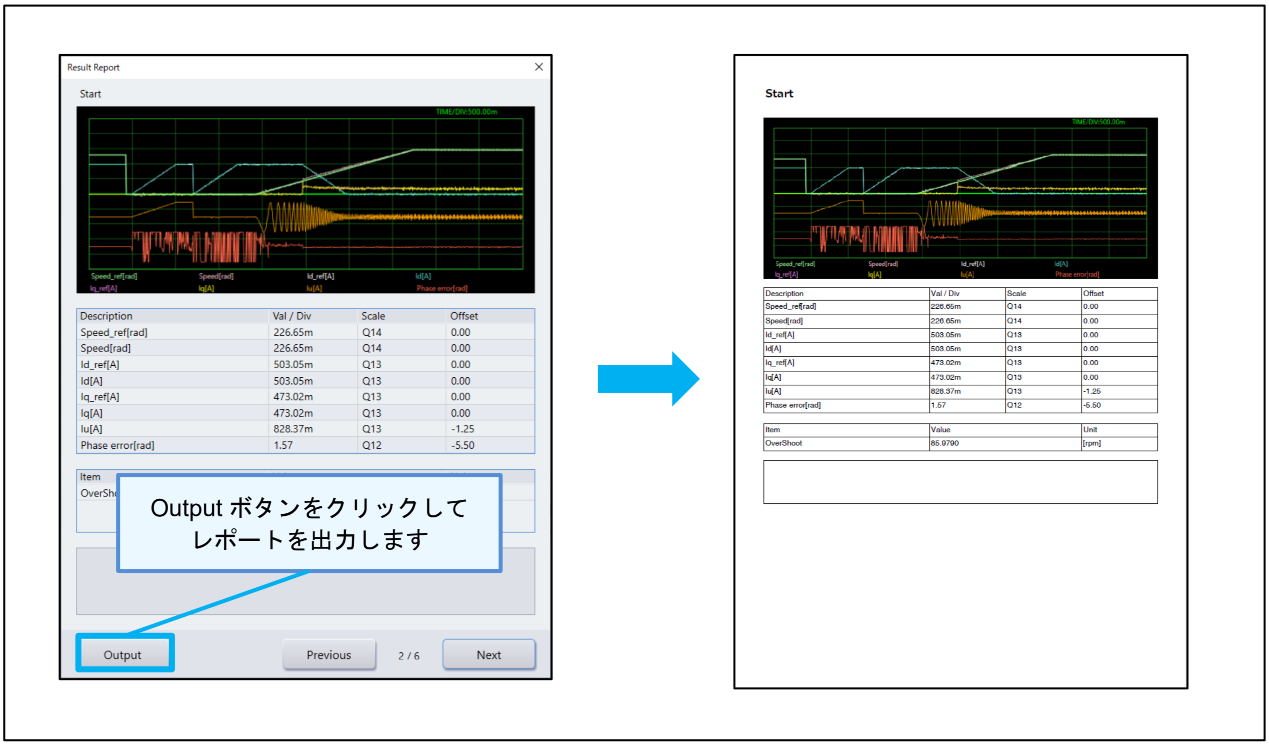


図 15-46 PDF 出力結果(駆動テスト出力結果)

15.5.7 ヘッダファイル出力機能（Output Window 画面）

設定したパラメータを対象のルネサス製制御プログラムに対応したヘッダファイルとして出力します。

Output Window 画面の「Output Header File」ボタンをクリックすることで、フォルダ選択ダイアログを表示し、選択フォルダにヘッダファイルを保存します。

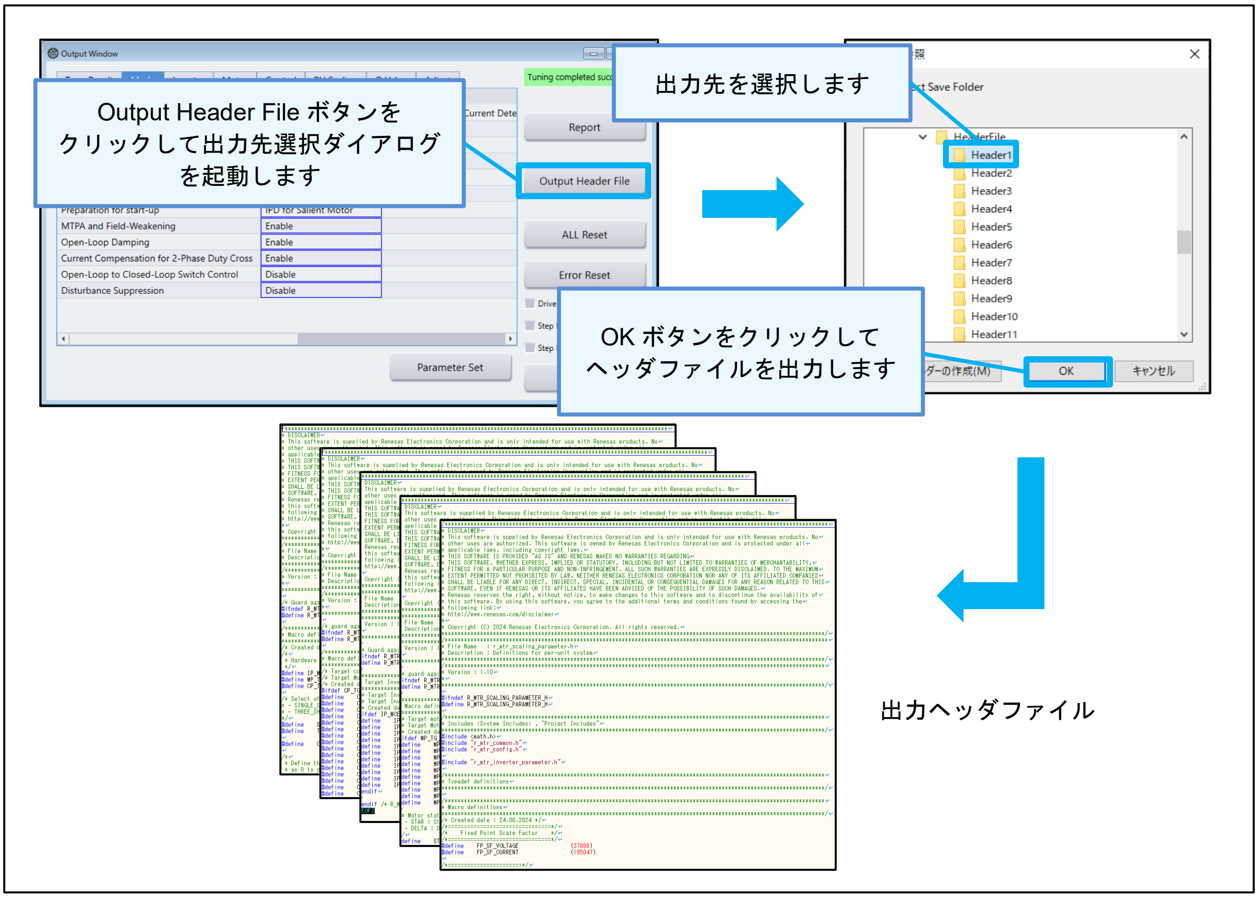


図 15-47 ヘッダファイル出力機能

表 15-25 出力ヘッダファイル一覧

No	名称	説明
1	r_mtr_config.h	コンフィギュレーション定義ファイル
2	r_mtr_inverter_parameter.h	インバータパラメータ定義ファイル
3	r_mtr_motor_parameter.h	モータパラメータ定義ファイル
4	r_mtr_control_parameter.h	制御パラメータ定義ファイル
5	r_mtr_scaling_parameter.h	スケーリングパラメータ定義ファイル

15.5.8 リセット機能（Output Window 画面）

パラメータ設定後に駆動プログラムにエラーが発生した場合は、リセットの種類を選択して実行します。Output Window 画面で「ALL Reset」「Error Reset」ボタンをクリックすると Tuner Reset 画面を表示できます。

- ALL Reset

  設計したパラメータを全てリセットします。Input Window 画面に遷移しますので、自動測定を再度実行してください。

- Error Reset

  エラー状態のみをリセットし、設計したパラメータは保持されます。

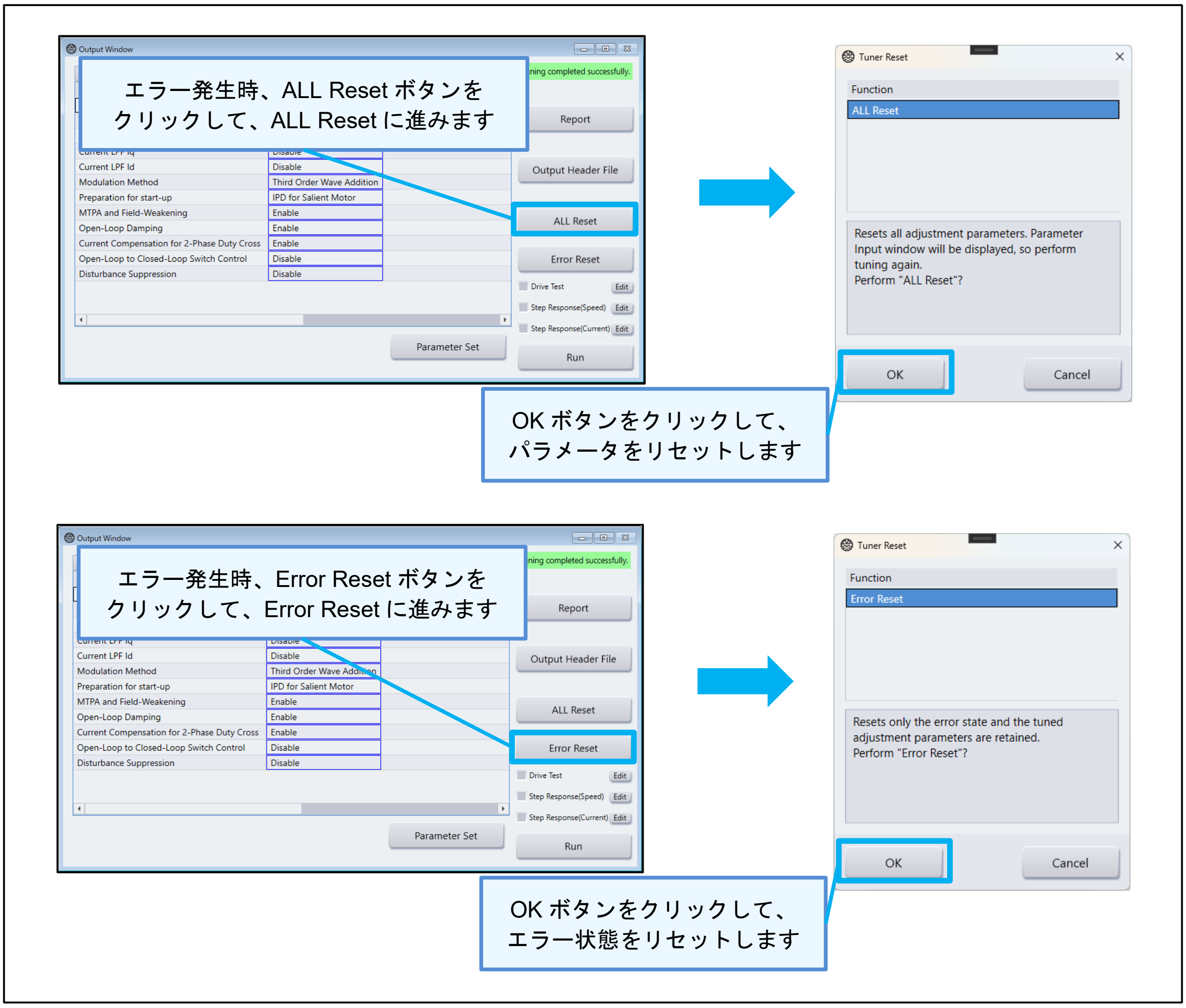


図 15-48 リセット機能

16.　Tuner ツール（MCI-HV-1 向け）

本章では、Tuner ツールの使い方について説明します。Select Tool で Tuner アイコンをクリックすると、Tuner ツールが起動します。

本章は、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)の 14 章に記載された内容を元に、MCI-HV-1 向けに再編集した内容となります。

16.1　実行環境準備

本章では、MCI-HV-1 を使って Tuner ツール対応のサンプルプログラムを起動させるまでの準備について説明します。

本資料で使用するサンプルプログラムは、CPU ボードとして MCB-RA6T2 を使用します。

MCI-HV-1 への入力電圧として、AC 入力と DC 入力の 2 系統あります。本サンプルプログラムは、AC 入力(AC 100V or 200V)、DC 入力(DC390V)に対応します。母線に引加する電圧は、DC390V から下げて使用することもできます。使用するモータの耐圧と、低電圧保護レベルの範囲で選定ください。

DC 入力で使用する場合には、外部から補助電源(+15V)の入力も同時に必要となります。

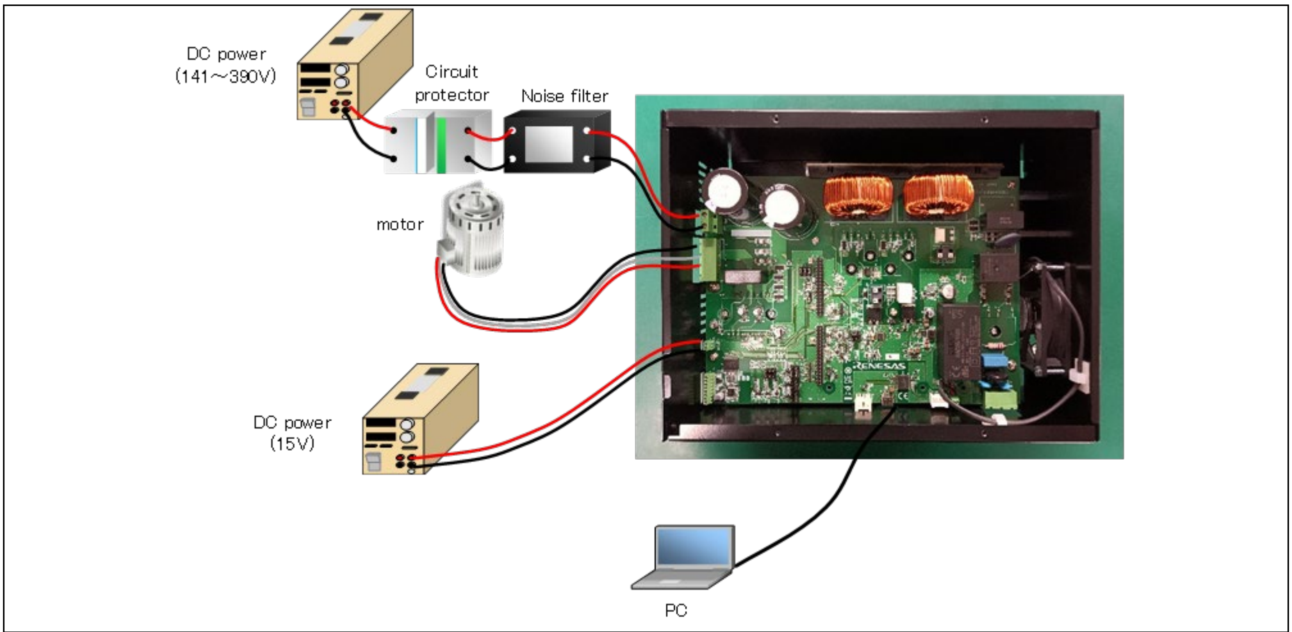


図 16-1　DC 接続イメージ (MCI-HV-1 ユーザーズマニュアルより抜粋)

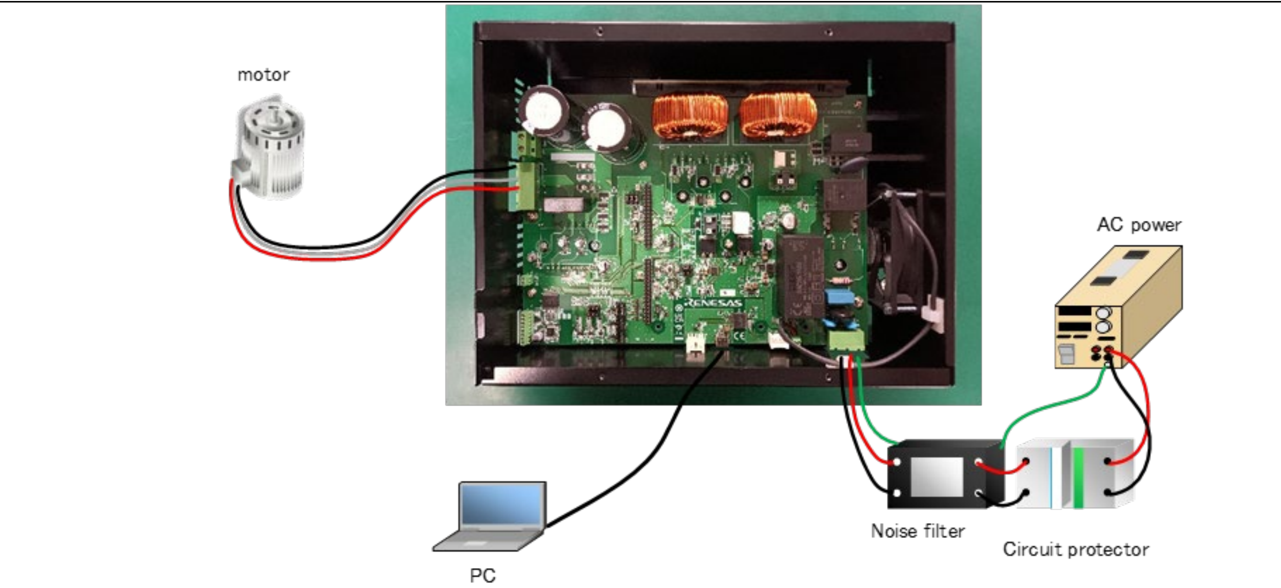


図 16-2　AC 接続イメージ(MCI-HV-1 ユーザーズマニュアルより抜粋)

16.1.1　　事前確認

MCI-HV-1 ユーザーズマニュアル(r12uz0138)をお読み頂き、MCI-HV-1 使用時の注意点をご確認ください。

16.1.2　　MCI-HV-1 の設定変更

MCI-HV-1 ユーザーズマニュアルの「3.2 電源供給」、「4.1 電源」をお読みの上、必要に応じて対応してください。

16.1.3　サンプルプログラムの書き込み

(a)　Renesas Motor Workbench に同梱されている実行ファイルの場合
Tuner 用の実行ファイルを CPU ボードに書き込みます。
・　R***_MCIHV1_PM_LESS_FOC_TUNER_***.HEX

(b)　ルネサス提供のサンプルプログラム（プロジェクト形式）の場合
ビルドした Tuner 用の実行ファイルを CPU ボードに書き込みます。
Tuner 機能をサンプルプログラムに組み込む手順は、14.6 章を参照ください。

PC（Windows10 または 11）と CPU ボードの接続方法は、MCI-HV-1 ユーザーズマニュアルの「3. セットアップガイド」をお読みください。CPU ボード側の USB コネクタは、USB-TypeC になります。

フォルダパスに日本語が含まれているとビルドできません。日本語が含まれないフォルダパスにプロジェクトを展開してください。

16.1.4　各種ボード、モータの接続

MCI-HV-1、CPU ボード、RMW 用通信ボードの 3 ボードと、ご使用予定のモータの接続方法は、MCI-HV-1 ユーザーズマニュアルの「3.1 ハードウェアセットアップ」をお読みください。

16.1.5　電源の接続

MCI-HV-1 ユーザーズマニュアルの「3. セットアップガイド」をお読みの上、接続してください。
DC 電源は、DC390V を出力でき、モータを無負荷で運転可能な容量を持った装置を選定してご利用ください。印加する電圧は、DC390V 以下でも Tuner 機能は使用できます。なお、印加される電圧が高いため、配線や端子の絶縁には十分に気を付けて行ってください。

DC 電源を用いて電源供給を行う時は、下記の順番で必ず供給してください。補助電源より先に主回路用 DC 電源を供給すると、インバータの IGBT 部分が故障する場合があります。

(1)　CN6 補助電源(+15V)

(2)　CN3 主回路用 DC 電源(DC100V 超、DC450V 未満の範囲で使用可能)

AC 電源入力の場合

インバータボードの電源端子(CN2)に AC 電源をサーキットプロテクタとノイズフィルタを介して接続します。

16.2　概要

Tuner ツールは、永久磁石同期モータ固有のパラメータ（抵抗、インダクタンス等）を自動測定し、エンコーダ位置制御やセンサレス速度制御をする際に必要な各種制御パラメータ（PI 制御のゲイン等）を自動調整するツールです。

Tuner ツールは、制御プログラムで Tuning 機能がサポートされている場合に限り利用可能となります。その場合、Main Window の「Select Tool」に「Tuner」が選択可能な状態で表示されます。

注意　本ツールは、すべてのモータに対して有効に調整可能であることを保証するものではありません。

16.3　特徴

- Manual と Easy の 2 つの調整モードを提供します。
- Tuning 結果（調整済みのパラメータ）を PDF 形式のレポートや、ルネサス製制御プログラムのヘッダファイル形式で出力可能です。
  出力された形式のままでは、MCI-HV-1 のサンプルプログラムではご利用いただけません。

16.4　使用上の注意

AC200V のモータに対する Tuner ツールとなっているため、以下の点に注意してご使用ください。

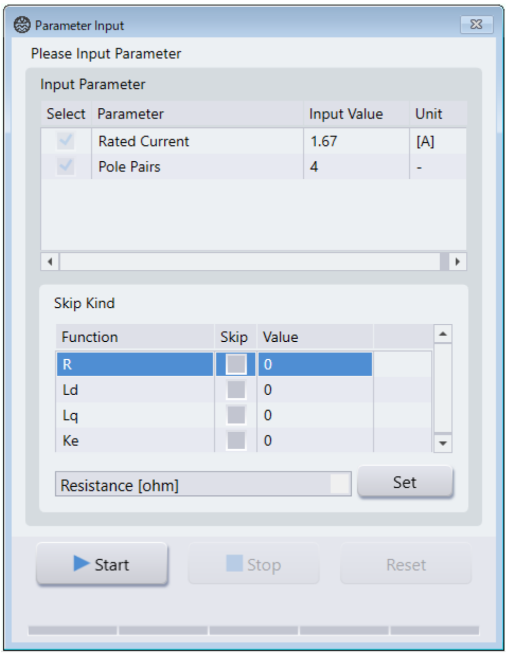
表 16-1　Tuner ツール・使用上の注意

注意事項	注意点
インバータの電圧保護レベル	過電圧レベル 420V 低電圧レベル 8V
インバータの電流保護レベル	過電流レベル 21.2A
適用可能なモータ	・AC200V 系の PM モータ(SPM 及び IPM モータ) ・容量は MCI-HV-1 及び電源容量以下であること ・モータは無負荷としてください。負荷は繋げないでください。
その他	・モータの回転部は、金属カバーなどで覆ってください ・モータが回転します。安全な環境下で実行してください。 ・モータに電力を供給します。導電部に接触するなどの感電に注意してください。

16.5　画面構成

Tuner ツールの構成を以下に示します。

(a) Parameter Input 画面



(b) Tune Window（Manual タブ／Easy タブ）

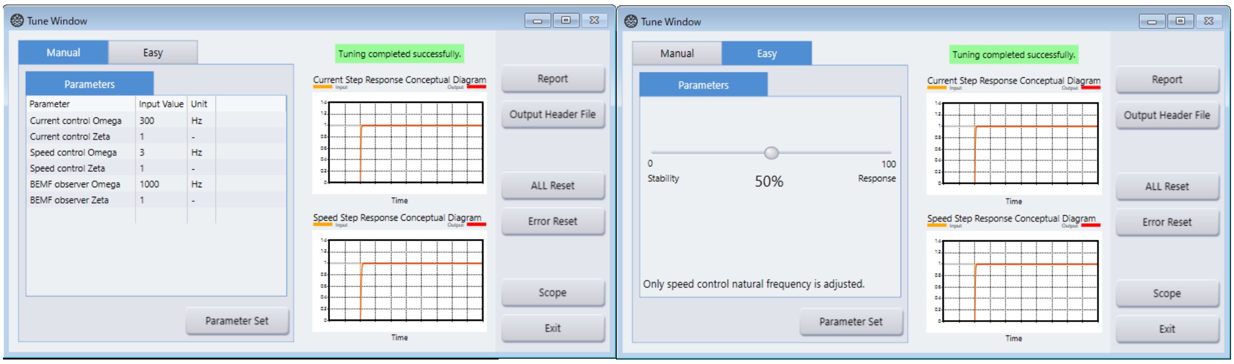


図 16-3　Tuner ツール画面

16.6 操作説明

この章では、チューニング作業の手順について説明します。各画面にある機能の詳細については、「2.4 Main Window の操作」を参照してください。

16.6.1 事前準備

16.6.1.1 実行ファイルの書き込み

「16.1.3 サンプルプログラムの書き込み」をお読みください。

16.6.1.2 Renesas Motor Workbench の起動

デスクトップにあるショートカットアイコンで Renesas Motor Workbench を起動します。

16.6.1.3 RMT ファイルの読み込み

Renesas Motor Workbench に Tuner 用 RMT ファイルを読み込みます。

- RMT ファイル
  RA6T2_MCIHV1_PM_LESS_FOC_TUNER_E2S_V***.rmt

16.6.1.4 Map ファイルの読み込み

Map ファイルを読み込みます。

- Map ファイル
  RA6T2_MCIHV1_PM_LESS_FOC_TUNER_E2S_V***.map

16.6.1.5 通信の確立

Renesas Motor Workbench と評価ボードの通信を確立します。

16.6.1.6 Tuner ツールの起動

Renesas Motor Workbench の Select Tool から Tuner ツールを選択して Tuner ツールを起動します。

16.6.2 チューニングの実行

16.6.2.1 チューニング用パラメータの入力

Tuner ツールを起動すると、Parameter Input 画面が表示されます。上の Input Parameter 部に、モータの仕様書や銘板を参照して、Rated Current（定格電流）と Pole Pairs（極対数）を入力します。

あらかじめモータの特性が分かっていて、チューニングをスキップしたい項目がある場合は、下の Skip Kind 部にスキップするパラメータを設定します（詳しくは、「16.7.1 Skip 機能（Parameter Input 画面）」を参照してください）。

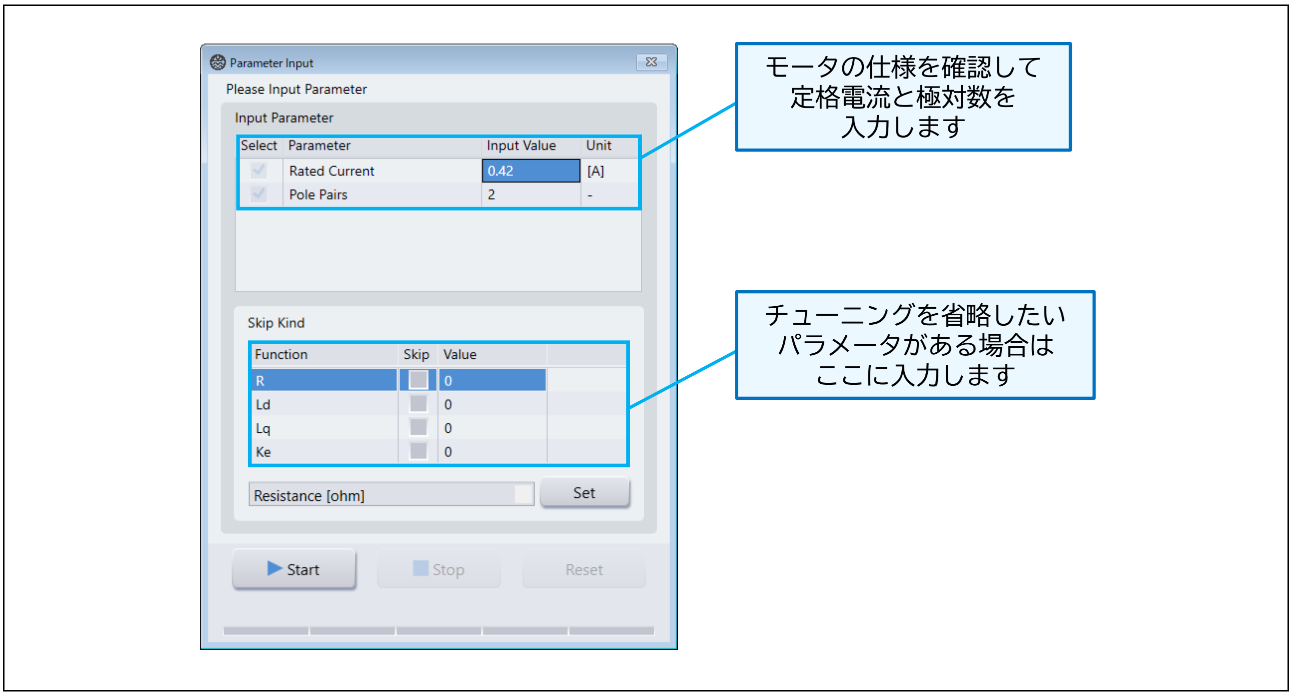


図 16-4 Parameter Input 画面

16.6.2.2　チューニングの実行

「Start」ボタンをクリックするとチューニングを開始し、チューニング中はプログレスバーが表示されます。チューニングを途中で停止したい場合は、「Stop」ボタンをクリックしてください。チューニングが完了すると Tune Window が表示されます。

注意　チューニングの実行中にモータが回転します。モータの回転部に取り付けられたカップリングが飛散しないよう、金属カバーするなどして、モータが回転しても良い環境で実行してください。実行前に再度 MCI-HV-1 ユーザーズマニュアル(r12uz0138)の「6. 注意事項」をお読みください。

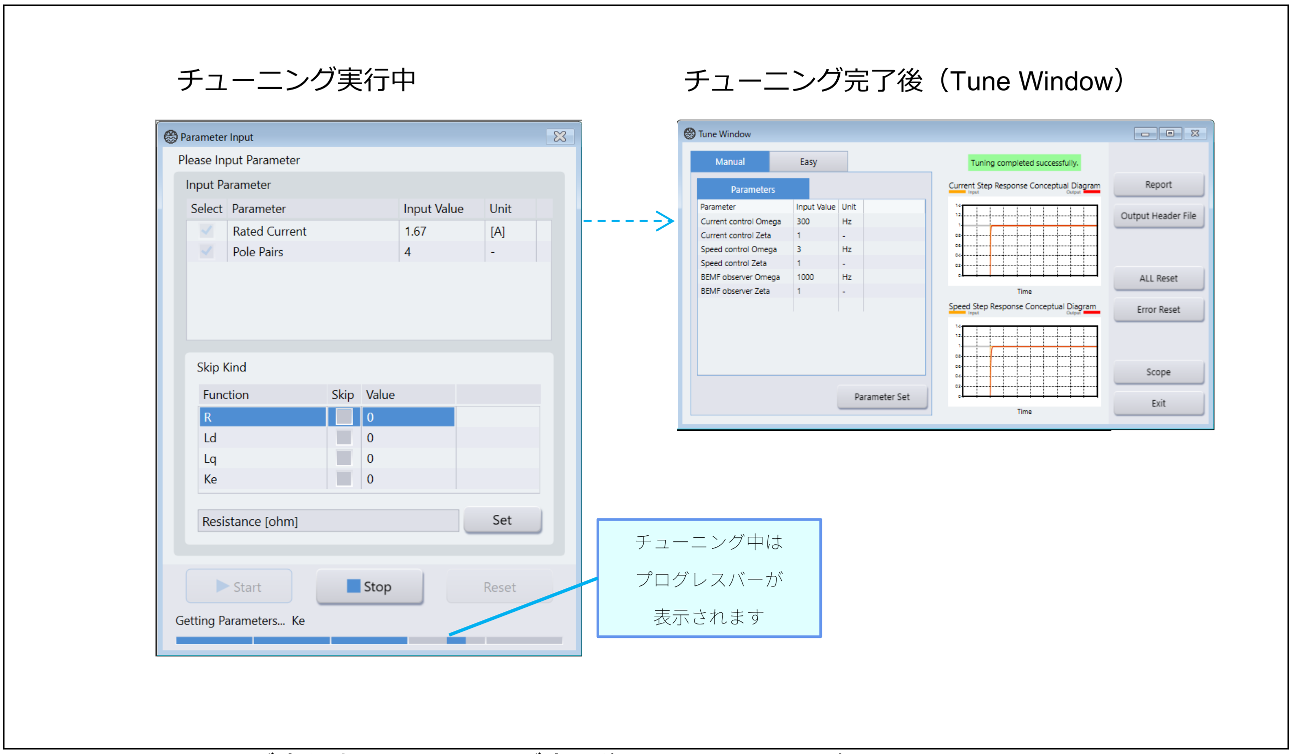


図 16-5　チューニング実行中とチューニング完了後の Tune Window 表示

Tuning 時にエラーが発生した場合は、エラーメッセージを確認して「Reset」ボタンをクリックしてください。

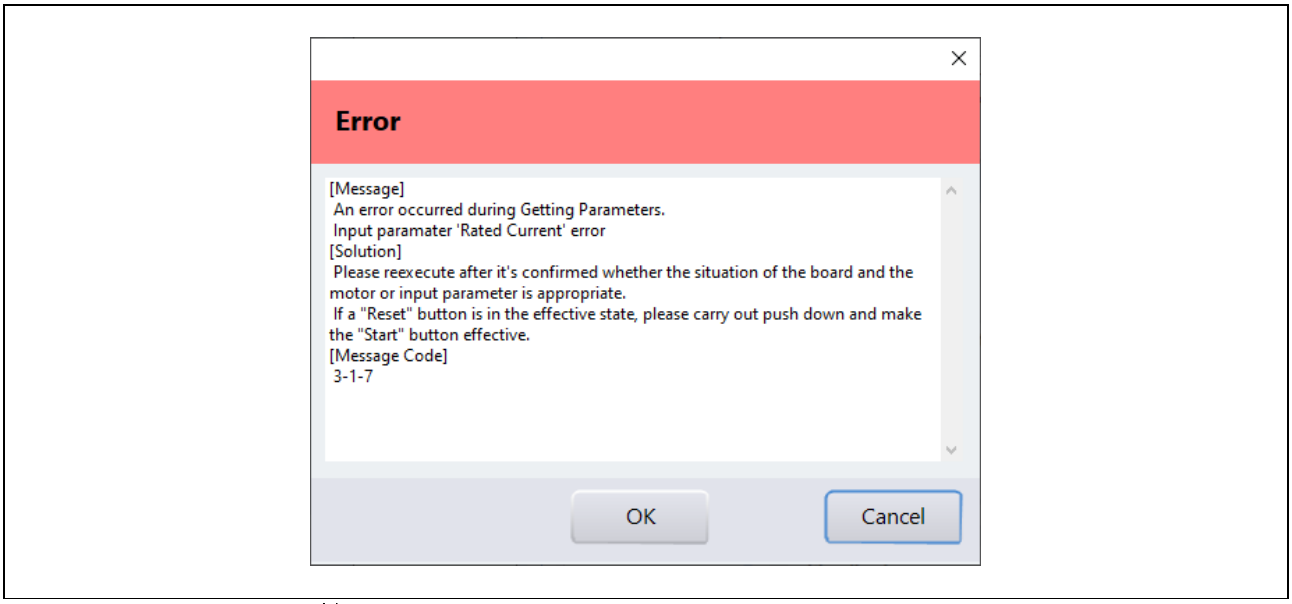


図 16-6　エラーメッセージ例

16.6.2.3　チューニング結果の確認

Tuning Window の「Report」ボタンをクリックして、Result Report 画面でチューニングの詳細を確認します。

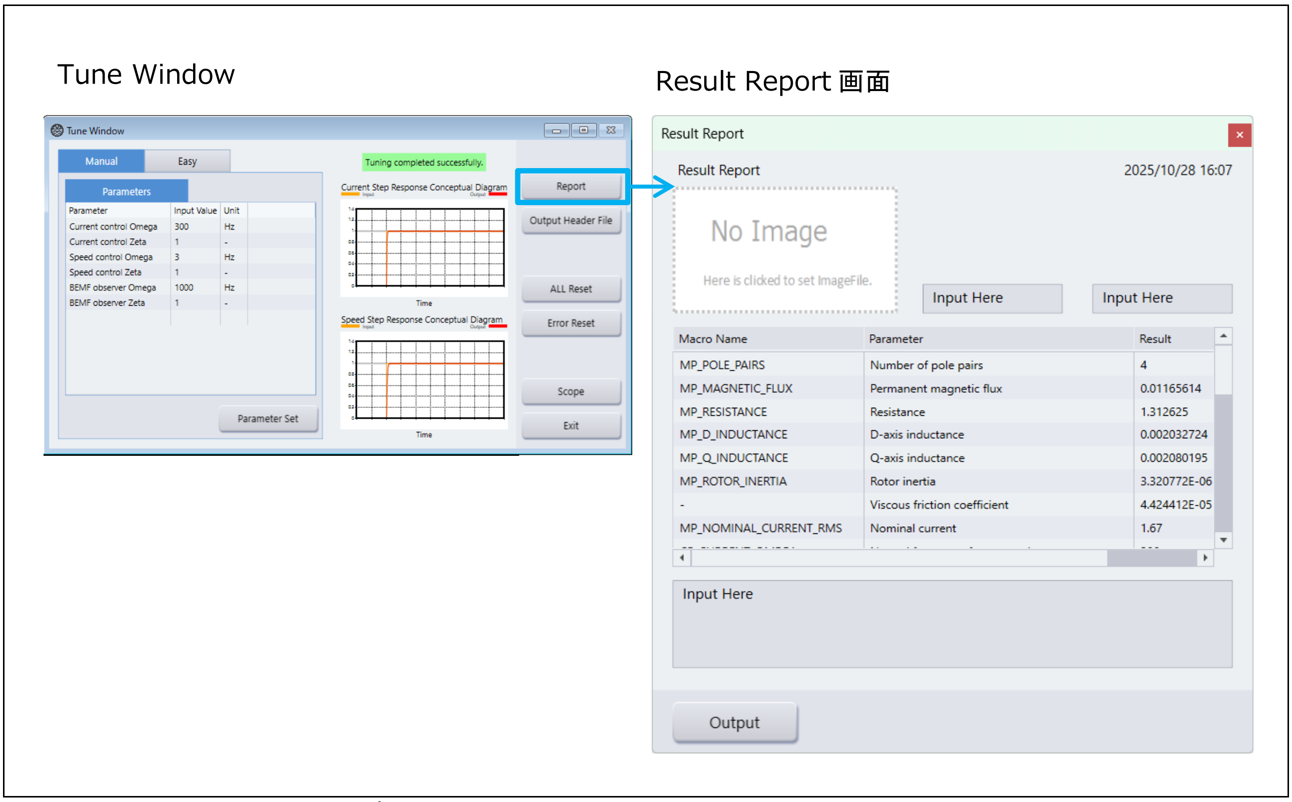


図 16-7　Result Report 画面の表示

16.6.2.4　Tuning の終了

Tuning Window の「Exit」ボタンをクリックすると、Parameter Input 画面に戻ります。

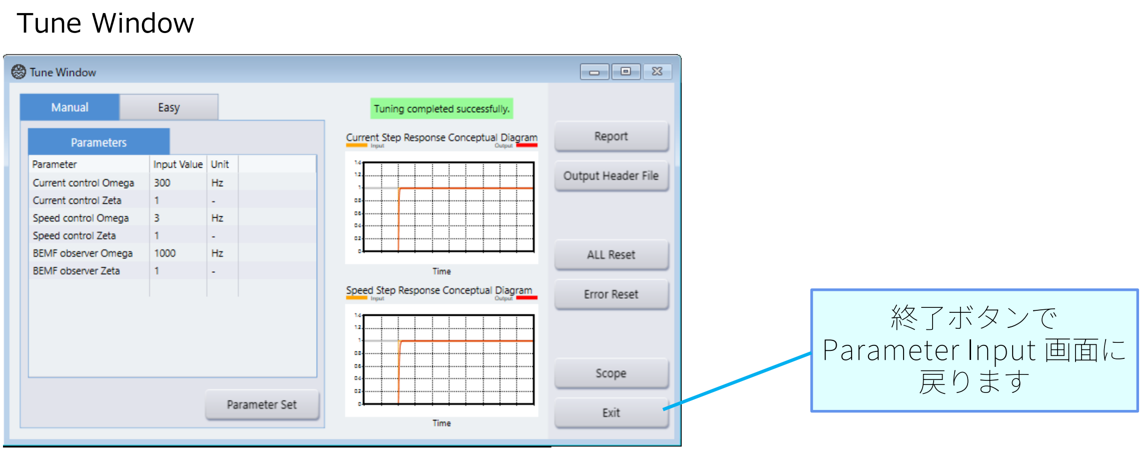


図 16-8　Exit ボタン

16.6.3　動作の確認

MCI-HV-1 用インバータソフトにチューニングの結果を反映できないため、モータの駆動テストが行えません。チューニング結果は、ご自身でサンプルプログラムのモータパラメータヘッダファイルに反映してください。詳細は 16.7.4(2)を参照ください。

16.6.4　Tuner ツールの終了

モータをテスト駆動している場合は、「STOP」ボタンでテストを終了します。Tune Window の「Exit」ボタンをクリックすると、Parameter Input 画面に戻ります。

その後「Main Window」ボタンで Main Window に戻るか、またはツール切り替えボタンの操作を行ってください。

16.7　機能説明

16.7.1　Skip 機能（Parameter Input 画面）

チューニングパラメータのうち、抵抗値、d 軸/q 軸インダクタンス、磁石磁束をチューニング前に入力することで、入力したパラメータの測定を省略できます。

Parameter Input 画面の Skip Kind 部で、省略するパラメータの Skip チェックボックスにチェックを入れてパラメータ値を入力してください。

入力が終わったら、「SET」ボタンをクリックします。

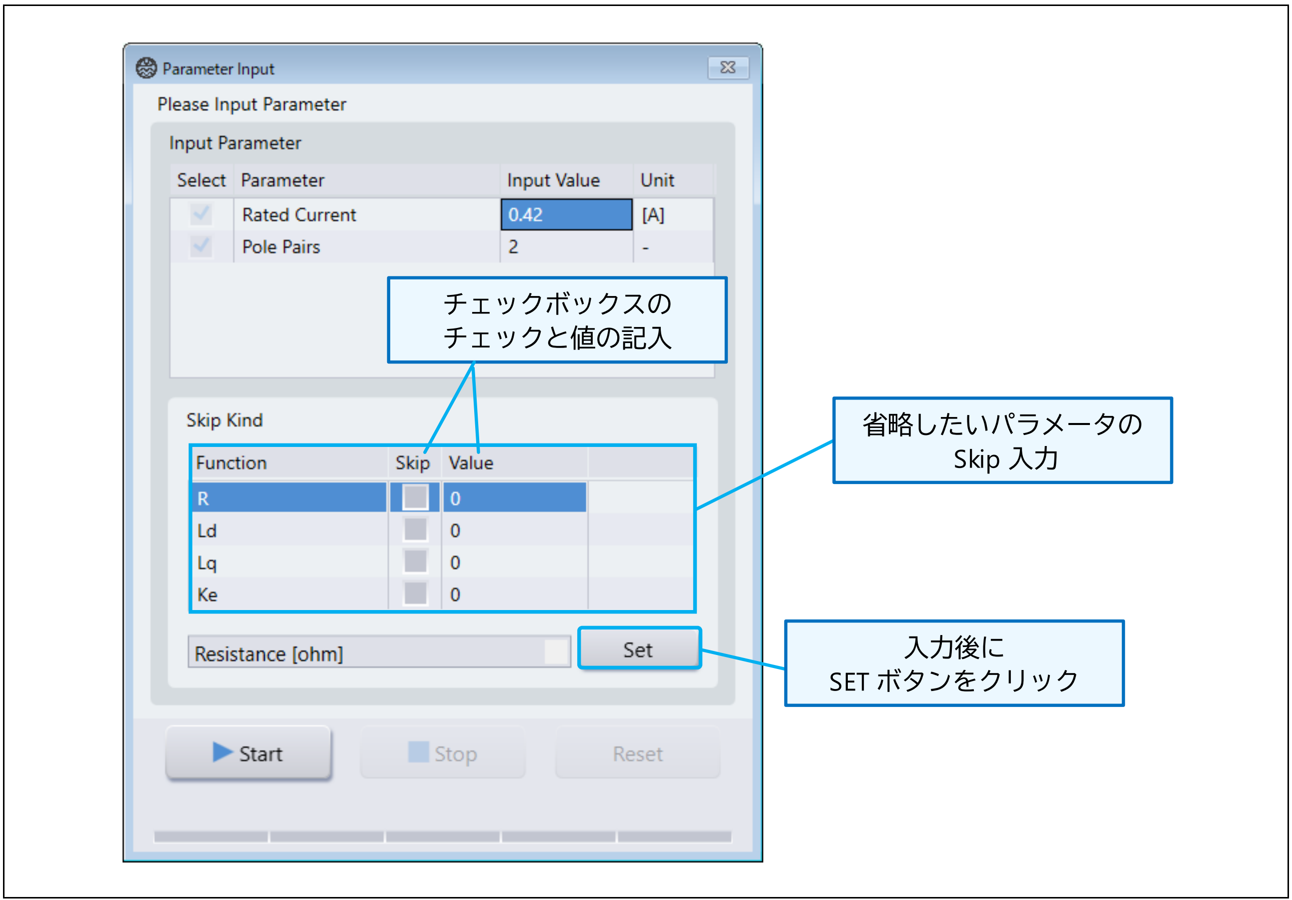


図 16-9　Skip 入力画面

16.7.2　　制御パラメータの調整（Tune Window）

Tune Window は、制御パラメータの設定用 Window で、設定方法は 2 種類の方法があり、タブによって切り替えます。タブには、Manual タブと Easy タブがあります。

Manual タブでは、制御の中で使用する各フィードバックループの固有周波数と減衰係数を調整し制御に反映することができます。

Easy タブでは、初期値を 50%として、スライダを左右へ操作する事で調整し制御へ反映する事ができます。

制御へ反映するには、「Parameter SET」ボタンをクリックしてください。

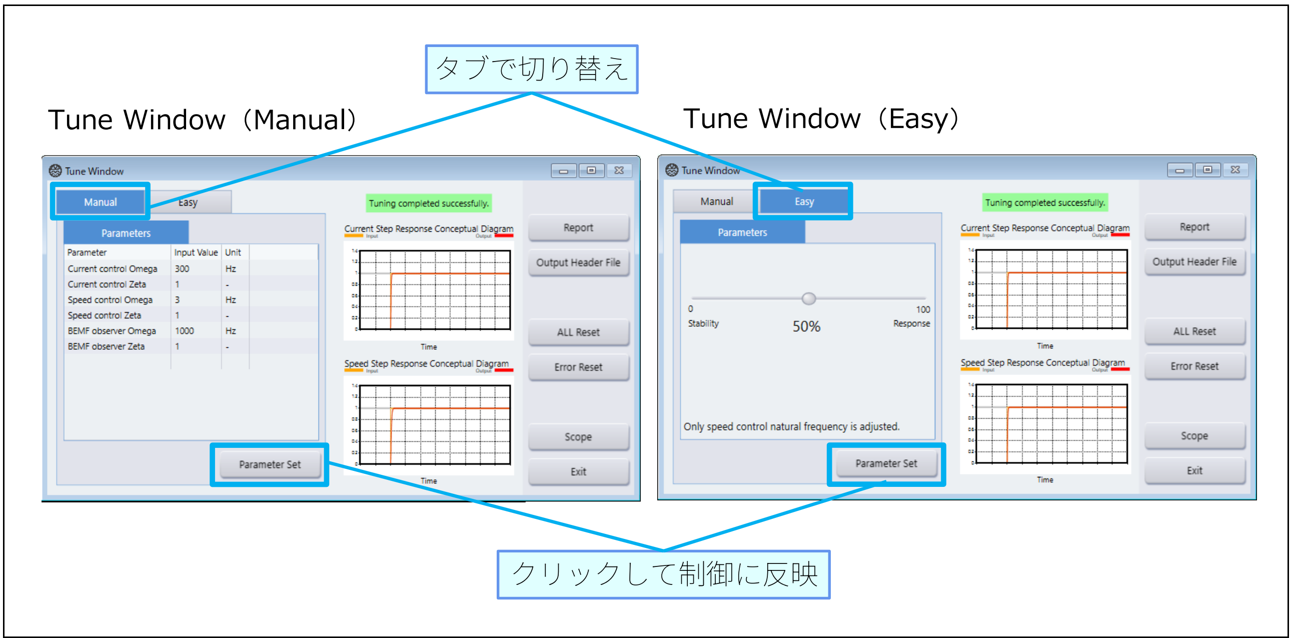


図 16-10　Manual タブと Easy タブ

表 16-2　調整パラメータ一覧（Manual タブ）

画面表示	調整パラメータ	備考
Current control Omega	電流制御系固有周波数 [Hz]	
Current control Zeta	電流制御系減衰係数	
Speed control Omega	速度制御系固有周波数 [Hz]	
Speed control Zeta	速度制御系減衰係数	
BEMF observer Omega	誘起電圧推定系固有周波数 [Hz]	センサレス速度制御のみ
BEMF observer Zeta	誘起電圧推定系減衰係数	センサレス速度制御のみ
Position control Omega	位置制御系固有周波数 [Hz]	エンコーダ位置制御のみ
Encoder counts per revolution	1 回転あたりエンコーダパルス数	エンコーダ位置制御のみ

16.7.3　リセット機能（Tune Window）

チューニング後に駆動プログラムにエラーが発生した場合は、リセットの種類を選択して実行します。Tune Window で「ALL Reset」「Error Reset」ボタンをクリックするとリセット画面を表示できます。

- ALL Reset
  調整パラメータすべてをリセットします。Parameter Input 画面に遷移しますので、チューニングを再度実行してください。

- Error Reset
  エラー状態のみをリセットし、チューニングした調整パラメータは保持されます。

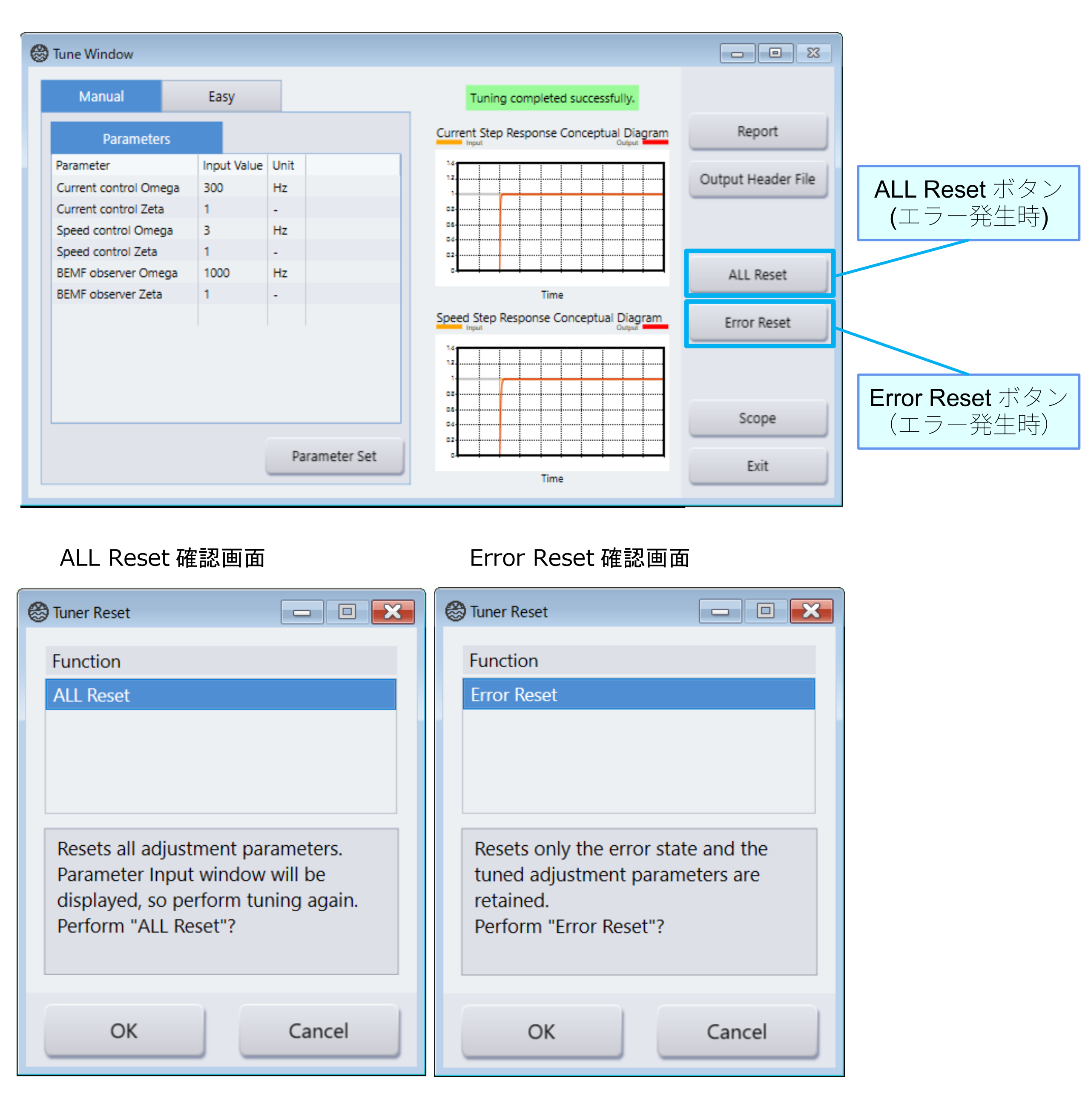


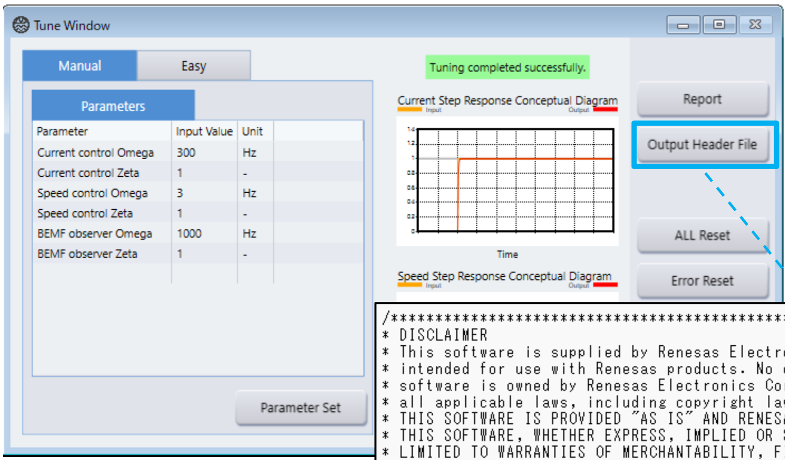
図 16-11　リセット機能

16.7.4 Header 出力（Tune Window）

(1) ヘッダファイル出力

Tuning 結果をルネサス製モータ制御プログラムのヘッダファイル形式で出力することができます。
Tune Window の「Output Header File」ボタンをクリックすると、「r_mtr_control_parameter.h」と「r_mtr_motor_parameter.h」の保存画面が表示されます。

Tune Window



ヘッダファイル出力例
（r_mtr_motor_parameter.h）

```
/*******************************************************************************
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only
* intended for use with Renesas products. No other uses are authorized. This
* software is owned by Renesas Electronics Corporation and is protected under
* all applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
* THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
* LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
* ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
* BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software
* and to discontinue the availability of this software. By using this software,
* you agree to the additional terms and conditions found by accessing the
* following link:
* http://www.renesas.com/disclaimer
*
* Copyright (C) 2020 Renesas Electronics Corporation. All rights reserved.
*******************************************************************************/

/******************************************************************************
* File Name   : r_mtr_motor_parameter.h
* Description : Definition of the target motor parameters
*                 (Renesas Motor Workbench Output file)
******************************************************************************/
/******************************************************************************
* Date : 2025.10.29 09:59
******************************************************************************/

/* Guard against multiple inclusion */
#ifndef R_MTR_MOTOR_PARAMETER_H
#define R_MTR_MOTOR_PARAMETER_H

/******************************************************************************
* Macro definitions
******************************************************************************/
#define    MTR_MOTOR_PARAMETER     (1)

/* Target motor definitions */
#define    MP_POLE_PAIRS           (4)                /* Number of pole pairs */
#define    MP_RESISTANCE           (1.278677f)        /* Resistance [ohm] */
#define    MP_D_INDUCTANCE         (0.002035681f)     /* D-axis inductance [H] */
#define    MP_Q_INDUCTANCE         (0.002070462f)     /* Q-axis inductance [H] */
#define    MP_MAGNETIC_FLUX        (0.01169281f)      /* Permanent magnetic flux [Wb] */
#define    MP_ROTOR_INERTIA        (3.310325E-06f)    /* Rotor inertia [kgm^2] */
#define    MP_NOMINAL_CURRENT_RMS  (1.67f)            /* Nominal current [Arms] */

#endif /* R_MTR_MOTOR_PARAMETER_H */
```

図 16-12 Output Header File ボタン

(2)　サンプルプログラムへの反映

出力されたヘッダファイルが既にサンプルプログラムに組み込まれている場合は、ヘッダファイルを上書きすることで、出力されたパラメータの反映が可能です。ただし、MCI-HV-1 用として提供されているサンプルプログラム(例：RA6T2_MCIHV1_PM_LESS_FOC_PFC_E2S_V***)では、手動で反映させる必要があります。

各パラメータを管理しているマクロ名の対比表を下記に示します。下記を参考にサンプルプログラム側の値を変更してください。

表 16-3　モータ用パラメータマクロ対比表

ファイル/マクロ説明	Tuner ツール出力ファイル	サンプルプログラム 例：RA6T2_MCIHV1_PM_LESS_FOC_PFC_E2S_V***
ヘッダファイル	r_mtr_motor_parameter.h	r_motor_targetmotor_cfg.h
極対数	MP_POLE_PAIRS	MOTOR_CFG_POLE_PAIRS
抵抗 [ohm]	MP_RESISTANCE	MOTOR_CFG_RESISTANCE
d 軸のインダクタンス [H]	MP_D_INDUCTANCE	MOTOR_CFG_D_INDUCTANCE
q 軸のインダクタンス [H]	MP_Q_INDUCTANCE	MOTOR_CFG_Q_INDUCTANCE
磁束 [wb]	MP_MAGNETIC_FLUX	MOTOR_CFG_MAGNETIC_FLUX
ロータのイナーシャ [kg m2]	MP_ROTOR_INERTIA	MOTOR_CFG_ROTOR_INERTIA
定格電流 [A]	MP_NOMINAL_CURRENT_RMS	MOTOR_CFG_NOMINAL_CURRENT_RMS

表 16-4　モータ制御用パラメータマクロ対比表

ファイル/マクロ説明		Tuner ツール出力ファイル	サンプルプログラム RA6T2_MCIHV1_PM_LESS _FOC_PFC_E2S_V***
ヘッダファイル		r_mtr_control_parameter.h	r_motor_module_cfg.h
	電流制御系固有周波数 [Hz]	CP_CURRENT_OMEGA	CURRENT_CFG_OMEGA
	電流制御系減衰係数	CP_CURRENT_ZETA	CURRENT_CFG_ZETA
	速度制御系固有周波数 [Hz]	CP_SPEED_OMEGA	SPEED_CFG_OMEGA
	速度制御系減衰係数	CP_SPEED_ZETA	SPEED_CFG_ZETA
	誘起電圧推定系固有周波数 [Hz]	CP_E_OBS_OMEGA	CURRENT_CFG_E_OBS_OMEGA
	誘起電圧推定系減衰係数	CP_E_OBS_ZETA	CURRENT_CFG_E_OBS_ZETA
	位置推定系固有周波数 [Hz]	CP_PLL_EST_OMEGA	CURRENT_CFG_PLL_EST_OMEGA
	位置推定系減衰係数	CP_PLL_EST_ZETA	CURRENT_CFG_PLL_EST_ZETA
	d 軸電流指令値減算開始速度（機械角）[rpm]	CP_ID_DOWN_SPEED_RPM	SENSORLESS_VECTOR_ID_DOWN_SPEED_RPM
	d 軸電流指令値加算開始速度（機械角）[rpm]	CP_ID_UP_SPEED_RPM	SENSORLESS_VECTOR_ID_UP_SPEED_RPM
	速度制限値 [rpm]	CP_OVERSPEED_LIMIT_RPM	SPEED_CFG_SPEED_LIMIT_RPM
	オープンループ時の d 軸電流指令値[A]	CP_OL_ID_REF	CURRENT_CFG_REF_ID_OPENLOOP
ヘッダファイル		上記と同じ	r_motor_targetmotor_cfg.h
	最大速度 [rpm]	CP_MAX_SPEED_RPM	MOTOR_CFG_MAX_SPEED_RPM

16.7.5　PDF 出力（Tune Window）

Tune Window の「Report」ボタンをクリックすると、Result Report 画面が表示されます。この画面の「Output」ボタンで、チューニング結果を PDF ファイルとして出力できます。

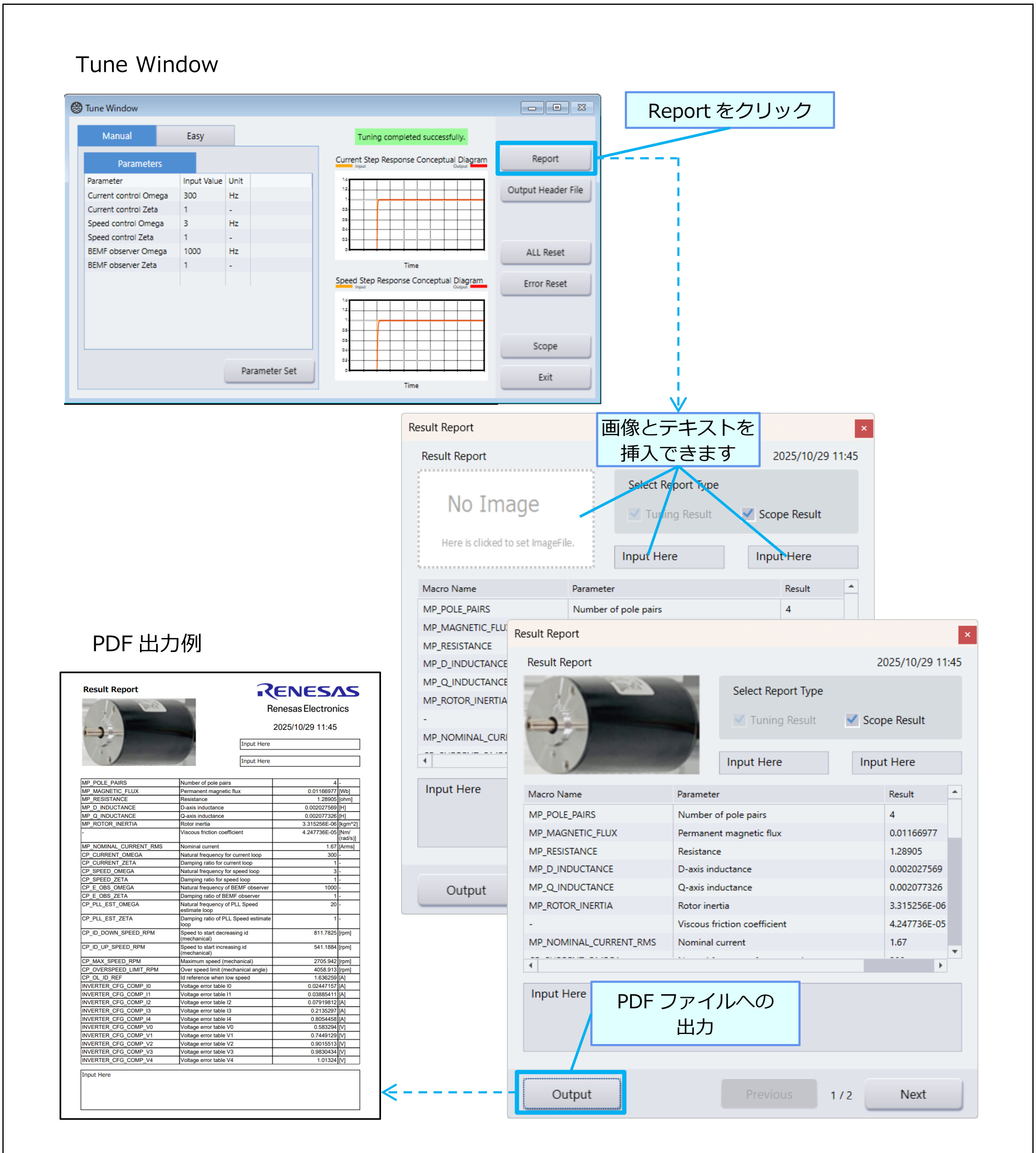


図 16-13　Output ボタン（PDF 出力）

チューニング結果として出力されるパラメーター覧を以下に示します。

表 16-5　出力パラメーター覧

マクロ名	概要	単位
MP_POLE_PAIRS	極対数	-
MP_MAGNETIC_FLUX	磁束	[wb]
MP_RESISTANCE	抵抗	[ohm]
MP_D_INDUCTANCE	d 軸のインダクタンス	[H]
MP_Q_INDUCTANCE	q 軸のインダクタンス	[H]
MP_ROTOR_INERTIA	ロータのイナーシャ	[kg m2]
MP_NOMINAL_CURRENT_RMS	定格電流	[A]
-	粘性摩擦係数	[Nm/(rad/s)]
CP_CURRENT_OMEGA	電流制御系固有周波数	[Hz]
CP_CURRENT_ZETA	電流制御系減衰係数	-
CP_SPEED_OMEGA	速度制御系固有周波数	[Hz]
CP_SPEED_ZETA	速度制御系減衰係数	-
CP_E_OBS_OMEGA	誘起電圧推定系固有周波数	[Hz]
CP_E_OBS_ZETA	誘起電圧推定系減衰係数	-
CP_PLL_EST_OMEGA	位置推定系固有周波数	[Hz]
CP_PLL_EST_ZETA	位置推定系減衰係数	-
CP_ID_DOWN_SPEED_RPM	d 軸電流指令値減算開始速度（機械角）	[rpm]
CP_ID_UP_SPEED_RPM	d 軸電流指令値加算開始速度（機械角）	[rpm]
CP_MAX_SPEED_RPM	最大速度	[rpm]
CP_OVERSPEED_LIMIT_RPM	速度制限値	[rpm]
CP_OL_ID_REF	オープンループ時の d 軸電流指令値	[A]
INVERTER_CFG_COMP_I0	電圧誤差補償用係数	[A]
INVERTER_CFG_COMP_I1	電圧誤差補償用係数	[A]
INVERTER_CFG_COMP_I2	電圧誤差補償用係数	[A]
INVERTER_CFG_COMP_I3	電圧誤差補償用係数	[A]
INVERTER_CFG_COMP_I4	電圧誤差補償用係数	[A]
INVERTER_CFG_COMP_V0	電圧誤差補償用係数	[V]
INVERTER_CFG_COMP_V1	電圧誤差補償用係数	[V]
INVERTER_CFG_COMP_V2	電圧誤差補償用係数	[V]
INVERTER_CFG_COMP_V3	電圧誤差補償用係数	[V]
INVERTER_CFG_COMP_V4	電圧誤差補償用係数	[V]

16.7.6 波形情報の表示（Scope 画面）

Scope 画面の「View Open」ボタンをクリックすると、波形表示に関する情報を確認できます。

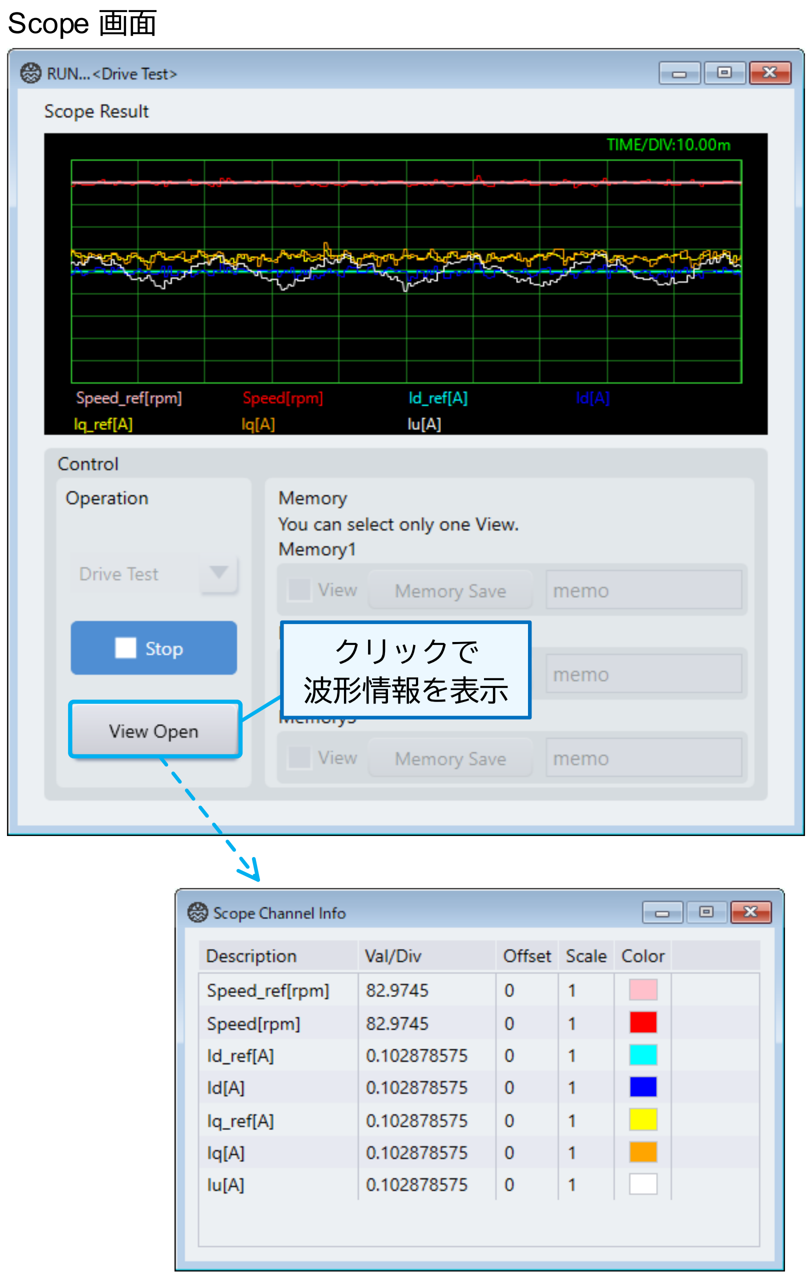


図 16-14 View Open ボタン

16.7.7　Memory 機能（Scope 画面）

Scope 画面では、モータ駆動時の波形を最大 3 つまでメモリし、各波形を切り替えて表示することができます。

「Memory Save」ボタンをクリックすると波形をメモリし、ボタンの表示は Memory Clear に変わります。再度クリックするとメモリをクリアします。

View チェックボックスにチェックをすると、メモリした波形が表示されます。

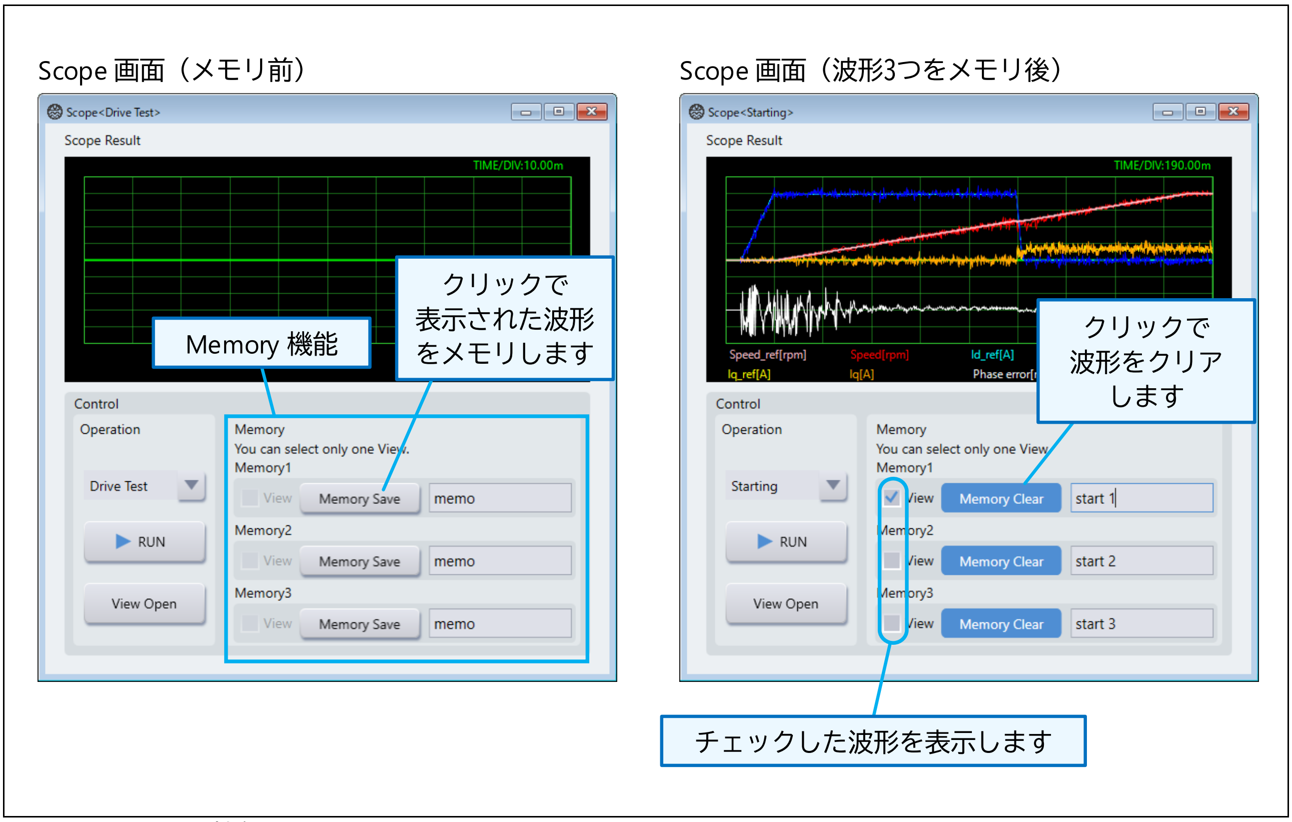


図 16-15　メモリ機能

17. 通信ライブラリ

17.1 概要

モータ制御開発支援ツールである Renesas Motor Workbench（以下、RMW と記述）は、MCU と変数情報を通信しています。RMW と MCU は、MCU のプログラムに通信ライブラリまたはビルトイン型通信ライブラリを組み込むことで、容易に通信することができます。

ビルトイン型通信ライブラリは、通信ライブラリと同等の機能に加え、RMW コマンド処理やスコープ処理を行うことができます。

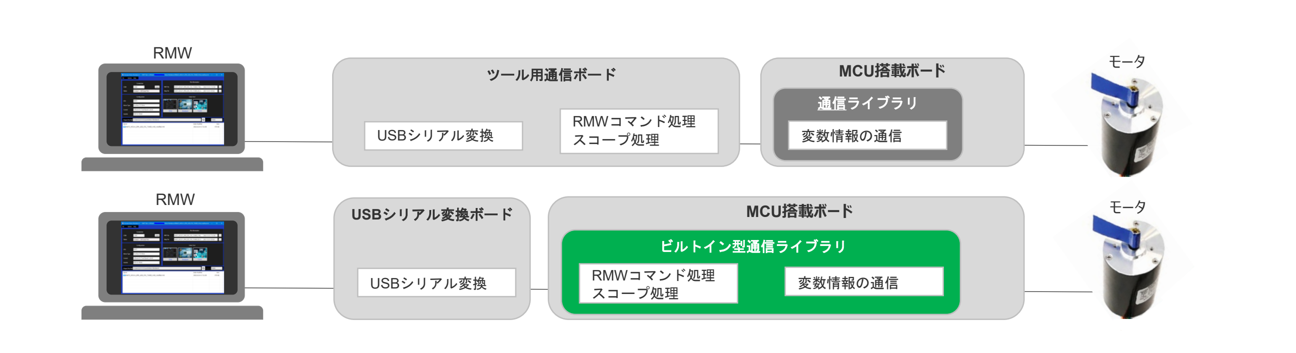


図 17-1 ライブラリの概要

17.2 特徴

通信ライブラリを使用することで、RMW と MCU の通信が可能になります。また、ビルトイン型通信ライブラリを使用することで、ツール用通信ボードの代わりに市販の USB シリアル変換ボードなどで RMW と MCU の通信が可能になります。

ビルトイン型通信ライブラリは、モータ制御 MCU の RAM 領域を使用するため、スコープウィンドウに表示できるデータ数に制限がありますが、簡易的な表示やデバッグ用途に最適です。より高速なサンプリングデータを表示する場合は、ツール用通信ボードと通信ライブラリを使用することを推奨します。

表 17-1　通信ライブラリとビルトイン型通信ライブラリの違い

<table>
<tr><th colspan="2">項目</th><th>通信ライブラリ</th><th>ビルトイン型通信ライブラリ</th></tr>
<tr><td colspan="2">推奨用途</td><td>高速なサンプリングデータの表示</td><td>簡易的な表示やデバッグ</td></tr>
<tr><td rowspan="3">対応 MCU</td><td>RX ファミリ</td><td>RX13T, RX23T, RX24T, RX24U
RX26T, RX66T, RX72T, RX72M</td><td>—</td></tr>
<tr><td>RL78 ファミリ</td><td>RL78/G14, RL78/G1F</td><td>—</td></tr>
<tr><td>RA ファミリ</td><td>RA6T1, RA6T2, RA6T3, RA4T1, RA8T1, RA8T2, RA2T1</td><td>RA6T2, RA6T3, RA4T1, RA8T1, RA8T2</td></tr>
<tr><td colspan="2">必要な通信ボード</td><td>MC-COM</td><td>市販の USB シリアル変換ボード</td></tr>
<tr><td colspan="2">スコープウィンドウ表示データ数</td><td>100,000</td><td>• 1,024 (RA6T2, RA8T1, RA8T2)
• 512 (RA6T3, RA4T1)</td></tr>
<tr><td colspan="2">サンプリング周期</td><td>Max. 20us/4ch
※変数情報を通信ボードと送受信</td><td>任意の周期
※変数情報をモータ制御 MCU の RAM 領域に保存</td></tr>
<tr><td colspan="2">使用可能機能</td><td>全機能</td><td>全機能</td></tr>
<tr><td colspan="2">スコープウィンドウ用 RAM 容量</td><td>—</td><td>• 32KB (RA6T2, RA8T1, RA8T2)
• 8KB (RA6T3, RA4T1)</td></tr>
<tr><td colspan="2">ライブラリの役割</td><td>変数情報を PC - MCU 間で通信</td><td>変数情報を PC - MCU 間で通信
RMW コマンド処理、スコープ処理</td></tr>
<tr><td colspan="2">ライブラリの仕様</td><td colspan="2">通信ライブラリ、ビルトイン型通信ライブラリともに同じ仕様
• 対象マイコンによって初期化関数の引数が異なる
• 対象マイコンによって使用可能な端子が異なる</td></tr>
</table>

17.3 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 17-2 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 17-3　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 17-4　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 17-5　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 17-6　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 17-7　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 17-8　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 17-9　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

17.4 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 17-10 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		

17.5　ユーザプログラムの設定方法

17.5.1　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、「17.5.4 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。

17.5.2　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 17−2　通信用割り込み関数記述例

17.5.3 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 17-11　ライブラリ関数（ics2_init）　（1/2）

<table>
<tr><td>関数名</td><td colspan="2">void　ics2_init(void* addr, char port, char level, char speed, char mode)</td></tr>
<tr><td>戻り値</td><td>void</td><td>なし</td></tr>
<tr><td rowspan="4">引数</td><td>void* addr</td><td>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</td></tr>
<tr><td>char port</td><td>SCI のポート番号や、SCI の使用するピンを設定します。</td></tr>
<tr><td>char level</td><td>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</td></tr>
<tr><td>char speed</td><td>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。

• 通信レート = CLOCK / (x × (speed + 1))　[Mbps]
通信レート(RA2T1) = CLOCK / (x × (speed + 1) × (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：SCISPICLK 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</td></tr>
</table>

表 17-12　ライブラリ関数（ics2_init）　（2/2）

引数	char mode	• 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	

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。	

表 17-13　ライブラリ関数（ics2_watchpoint）

17.5.4 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 17-2 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 17−3　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 17−4　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 17-5 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 17－5　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 17－6　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 17-7 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 17－7 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 17－8 データ転送関数 ics2_watchpoint() の呼び出し（RA6T1 での記述例）

17.6 HW 構成

ライブラリを取り込んで、RMW を使用する場合の HW 構成を説明します。

17.6.1 MCK モータ制御評価キットを使用する場合

MCK モータ制御評価キットで RMW を使用する場合の HW 構成例を図 17-9 に示します。

- MCK モータ制御評価キットの例：
  MCK-RA6T2

- 通信ライブラリを使用する場合の HW 構成例：
  CPU ボード上の MCU に、通信ライブラリを取り込んだプログラムをダウンロード
  RMW をインストールした PC と CPU ボードの間に、ツール用通信ボードを接続
- ビルトイン型通信ライブラリを使用する場合の HW 構成例：
  CPU ボード上の MCU に、ビルトイン型通信ライブラリを取り込んだプログラムをダウンロード
  RMW をインストールした PC と CPU ボードの間に、市販の USB シリアル変換ボードを接続

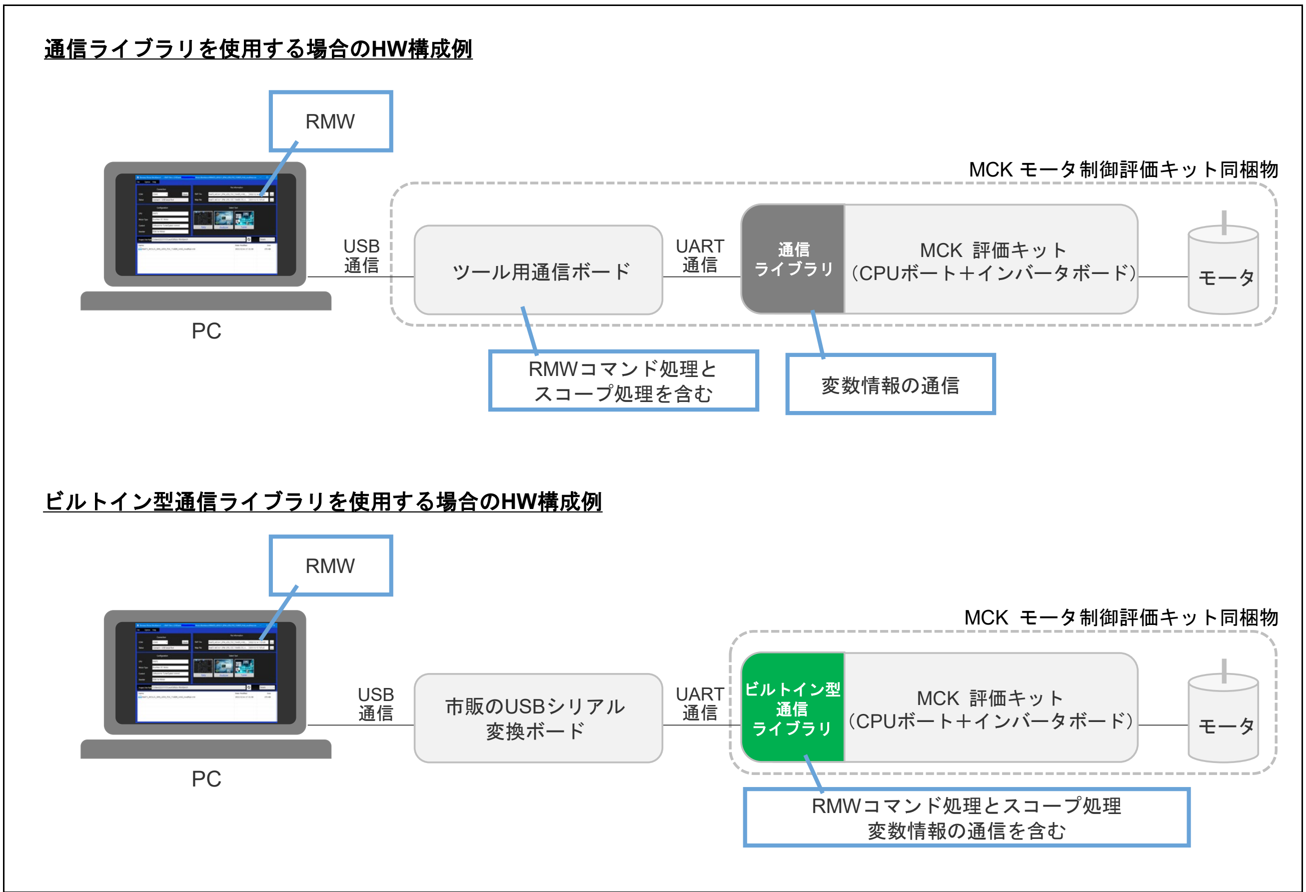


図 17-9 MCK モータ制御評価キットを使用する場合の HW 構成例

17.6.2 Evaluation System を使用する場合

Evaluation System で RMW を使用する場合の HW 構成例を図 17-10 に示します。

- Evaluation System の例：
  - Evaluation System for Stepping Motor with Resolver
  - Evaluation System for BLDC Motor
  - Motor Control Evaluation System for RA Family - RA6T1 Group

- 通信ライブラリを使用する場合の HW 構成例：
  CPU カード上の MCU に、通信ライブラリを取り込んだプログラムをダウンロード
  RMW をインストールした PC と Evaluation System を接続

- ビルトイン型通信ライブラリを使用する場合の HW 構成例：
  CPU カード上の MCU に、ビルトイン型通信ライブラリを取り込んだプログラムをダウンロード
  RMW をインストールした PC と CPU カードの間に、市販の USB シリアル変換ボードを接続

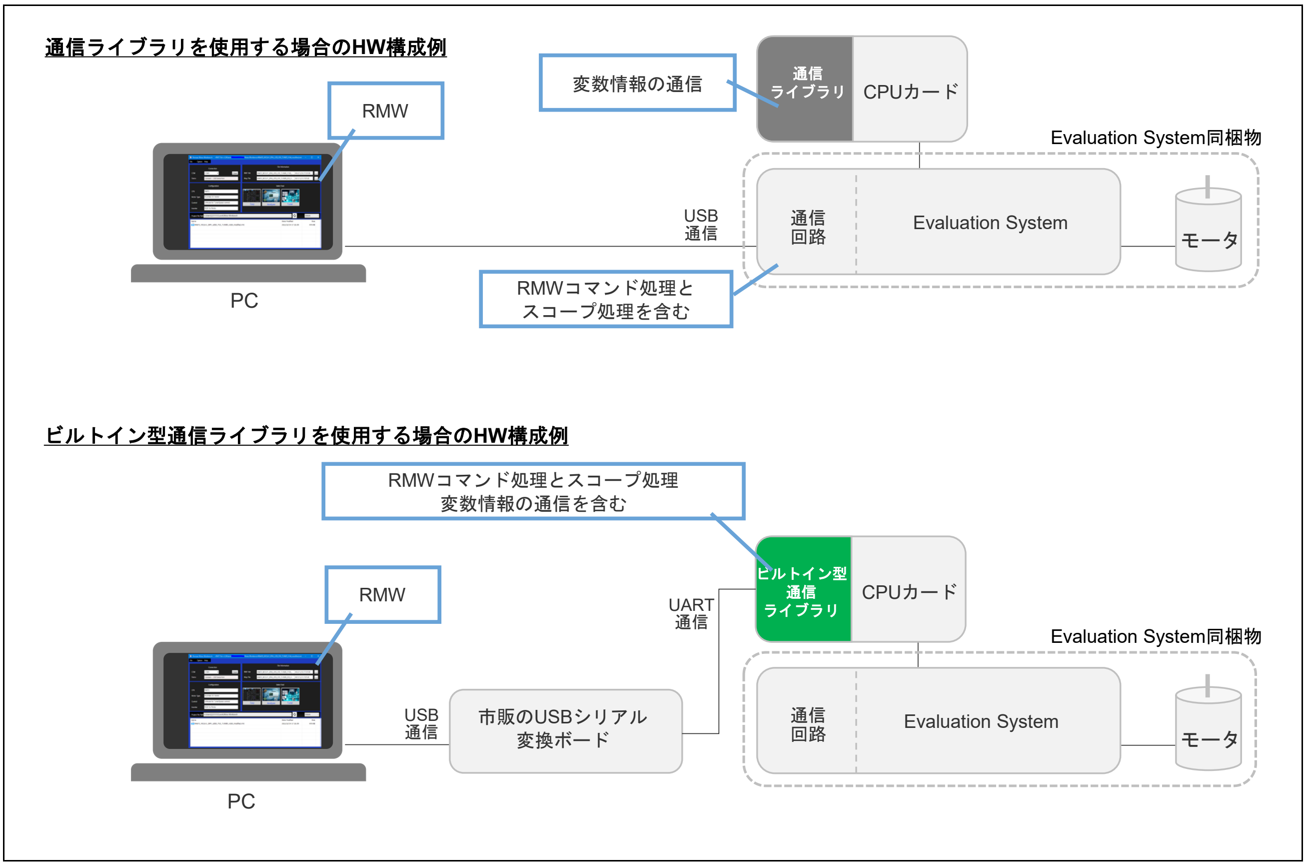


図 17-10 Evaluation System を使用する場合の HW 構成例

17.6.3 ユーザボードを使用する場合

ユーザボードで RMW を使用する場合の HW 構成例を図 17-11 に示します。

- ユーザボードの例：
  ユーザが開発したボード

- 通信ライブラリを使用する場合の HW 構成例：
  ユーザボード上の MCU に、通信ライブラリを取り込んだプログラムをダウンロード

  RMW をインストールした PC とユーザボードの間に、ツール用通信ボードを接続

- ビルトイン型通信ライブラリを使用する場合の HW 構成例：
  ユーザボード上の MCU に、ビルトイン型通信ライブラリを取り込んだプログラムをダウンロード

  RMW をインストールした PC とユーザボードの間に、市販の USB シリアル変換ボードを接続

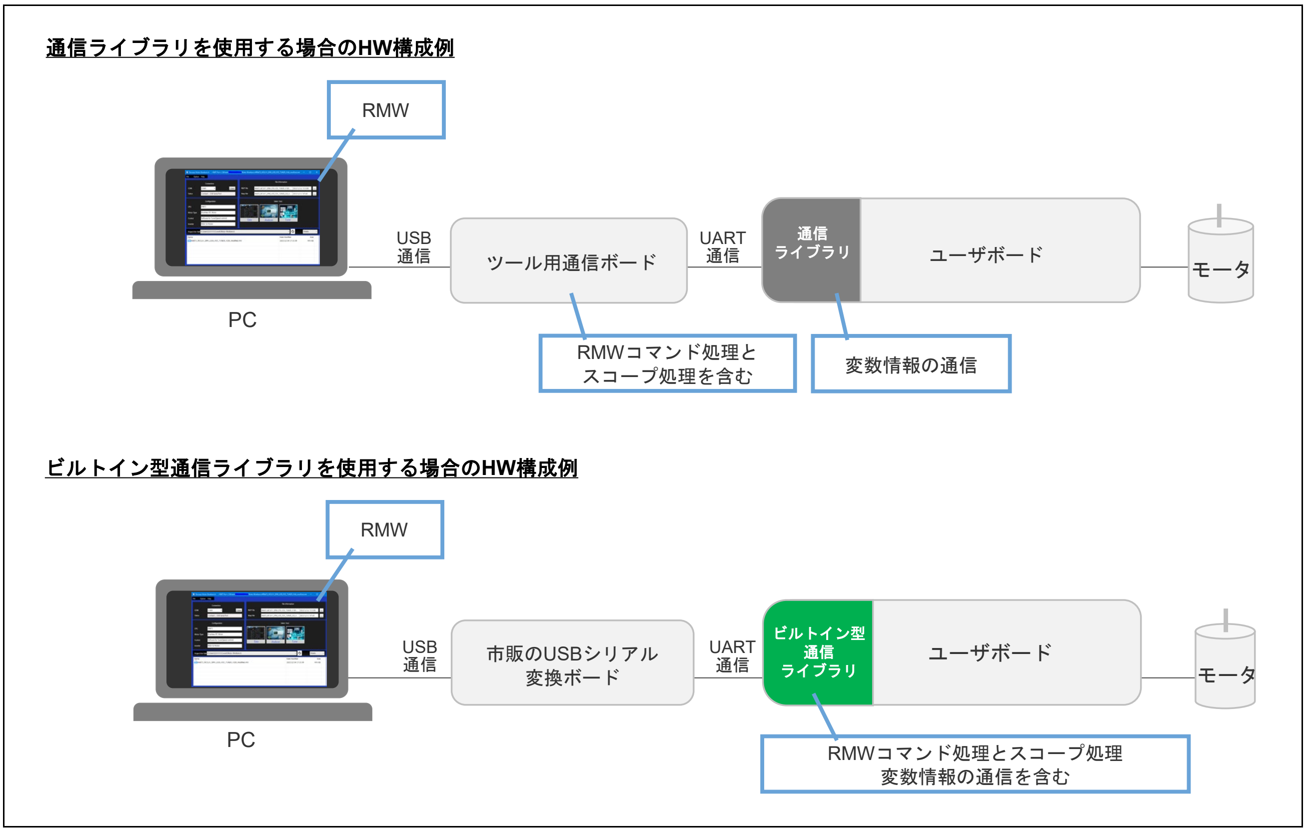


図 17-11 ユーザボードを使用する場合の HW 構成例

17.7 通信ライブラリの使用例

モータ制御プログラムに通信ライブラリを取り込んで RMW を使用する場合の手順は、以下のとおりです。

- 使用端子の決定：通信ボードと MCU を搭載したボードの接続に使用する端子を決定
- 通信速度の決定：RMW とボード間の通信速度を決定
- モータ制御プログラムの作成
  - 通信ライブラリファイルをコピー
  - 通信ライブラリファイルを追加
  - 通信ライブラリ関数を呼び出し
- 通信ライブラリを取り込んだモータ制御プログラムを MCU にダウンロード
- RMW と接続

本章では、MCK-RA6T2 の MCK モータ制御評価キットで RMW を使用する場合の使用例を説明します。
ユーザボードで RMW を使用する場合の手順は、MCK モータ制御評価キットを使用する場合と同等です。

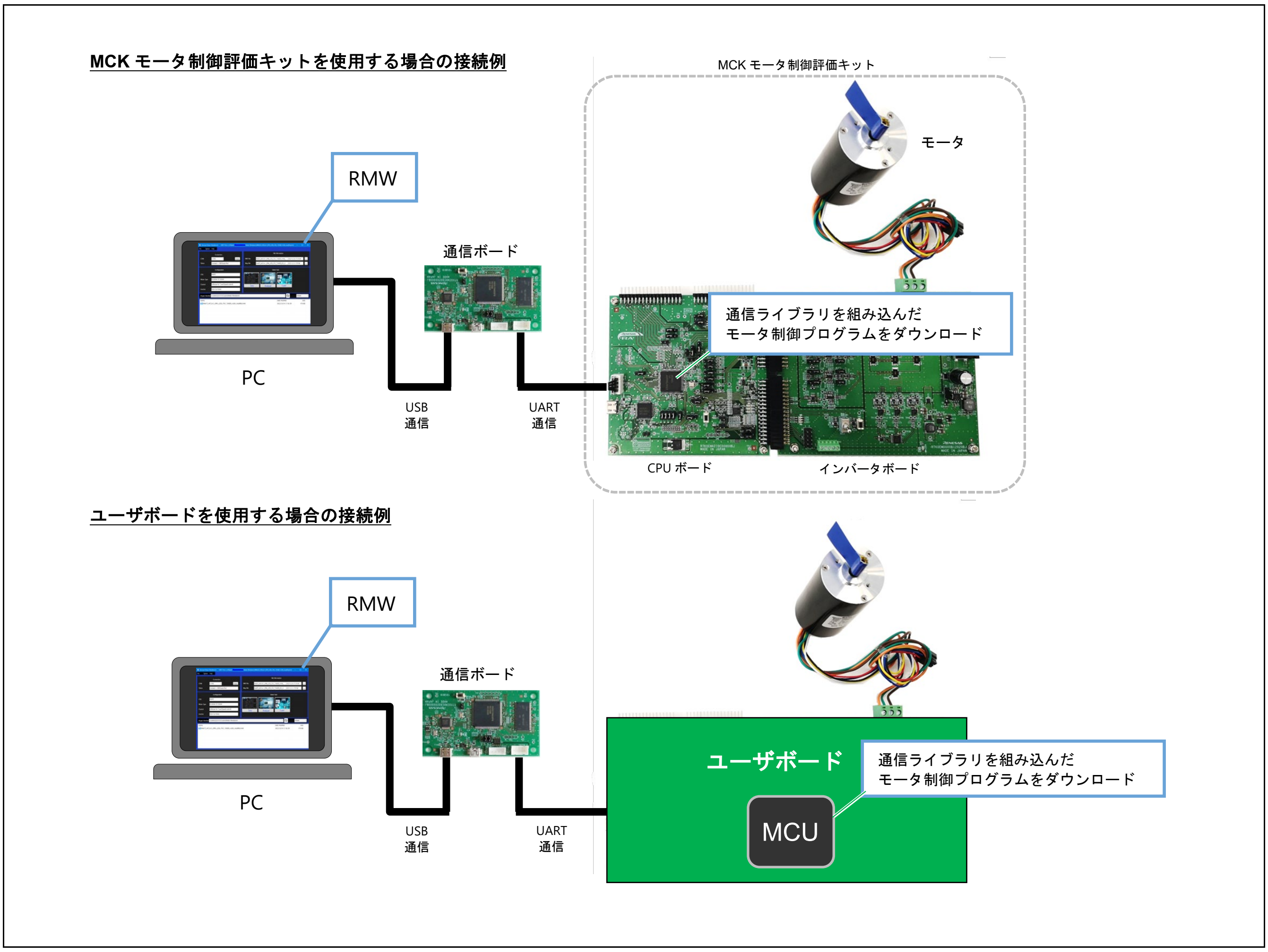


図 17-12 通信ライブラリを取り込んで RMW を使用する場合の接続例

17.7.1 HW 準備

MCK-RA6T2 及び、同梱の通信ボード(MC-COM)を使用します。

MCK-RA6T2 についての詳細は、下記ドキュメントを参照してください。

- R12QS0047 MCK-RA6T2 クイックスタートガイド
- R12UZ0091 MCK-RA6T2 ユーザーズマニュアル

通信ボード(MC-COM)についての詳細は、下記ドキュメントを参照してください。

- R12UZ0093 MC-COM ユーザーズマニュアル

MCK-RA6T2 同梱物

図 17-13 使用機材

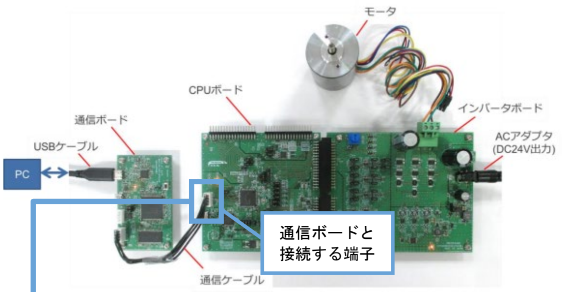
17.7.2 使用端子の決定

CPU ボードと通信ボードの SCI 端子間を通信ケーブルで接続します。使用する SCI 端子についての詳細は MCK-RA6T2 のユーザーズマニュアルを参照してください。

MCK-RA6T2 の場合、SCI9 と PD05/TXD9_A と PD06/RXD9_A を使用します。

本章で決定した使用端子は、以下の章で使用します。

- 17.7.4.3 章 ライブラリ関数の呼び出し：ics2_init()の引数 port に設定
- 17.7.6 章 RMW と接続 ：USB シリアル変換ボードとの接続



対応 CPU	RA6T1	RA6T2
ファイル	ICS2_RA6T1.o ICS2_RA6T1.h	ICS2_RA6T2.o ICS2_RA6T2.h
通信レート	0.5 Mbps ～ 15.0 Mbps	0.5 Mbps ～ 20.0 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
サポート変数タイプ	8bit 符号なし 整数型（数値表示・設定 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

通信ライブラリで
使用できる端子

表 6-5 SCI コネクタ（CN10）ピンアサイン

端子 No.	端子機能	RA6T2 接続端子
1	GND	-
2	マイコン受信側	PD06/RXD9_A
3	マイコン送信側	PD05/TXD9_A
4	VCC	-

SCI9-A

RMW で使用できる端子

MCK-RA6T2 ユーザーズマニュアルから抜粋

図 17-14 使用端子の決定

17.7.3 通信速度の決定

PC と通信ボード、および CPU ボードの通信速度を決定します。通信速度は、以下の条件を満たす必要があります。

- ライブラリで使用できる通信速度であること
- MC-COM ボードが使用できる通信速度であること

本章で決定した通信速度は、下記の章で使用します。

- 17.7.4.3 章 ライブラリ関数の呼び出し : ics2_init()の引数 speed に設定
- 17.7.6 章 RMW と接続 : RMW に通信速度を設定

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 17-14 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（66666666 Hz）、12.5 Mbps（10 MHz）、 16.66 Mbps（133333333 Hz）

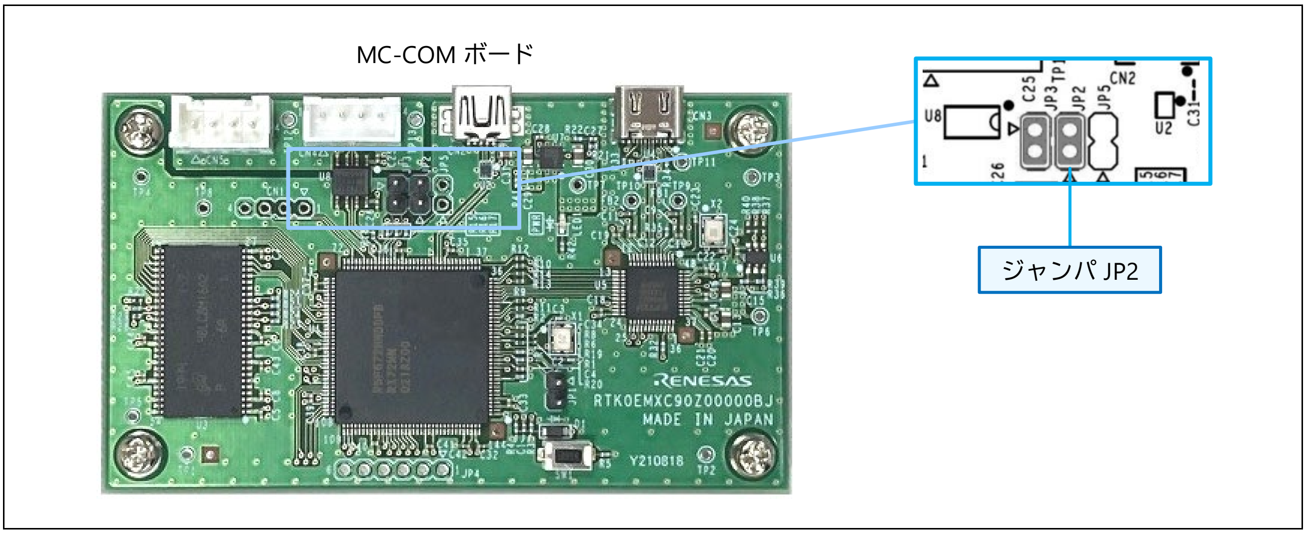


図 17-15 ツール用通信ボード MC-COM のジャンパ JP2

17.7.4　　プログラムの作成

MCK-RA6T2 の CPU ボードに搭載された RA6T2 にダウンロードする、モータ制御プログラムを作成します。

本章では、すでに e² studio で作成されたモータ制御プログラムに、通信ライブラリを取り込む方法を説明します。

17.7.4.1　　ライブラリファイルのコピー

通信ライブラリのヘッダファイルとオブジェクトファイルは、RMW インストーラをダウンロードしたファイルに同梱されています。通信ライブラリのヘッダファイルとオブジェクトファイルを、モータ制御プログラムの任意の場所にコピーします。

ライブラリファイルをコピーしたフォルダ構成を、サンプルコードを例に図 17-16 に示します。

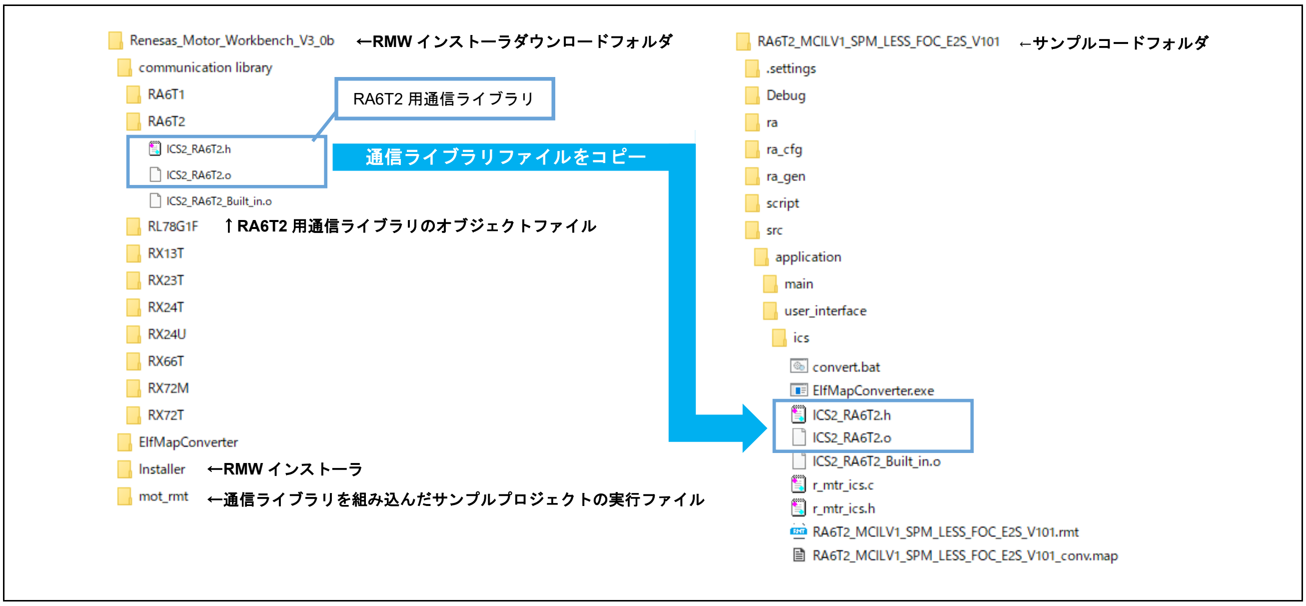


図 17-16　ライブラリファイルをコピーしたフォルダ構成例

17.7.4.2　ライブラリファイルの追加

コピーしたライブラリファイルを e² studio で追加します。e² studio でプロジェクトのプロパティを開き、オブジェクトファイルを図 17-17 のとおり、ヘッダファイルパスを図 17-18 のとおり追加します。

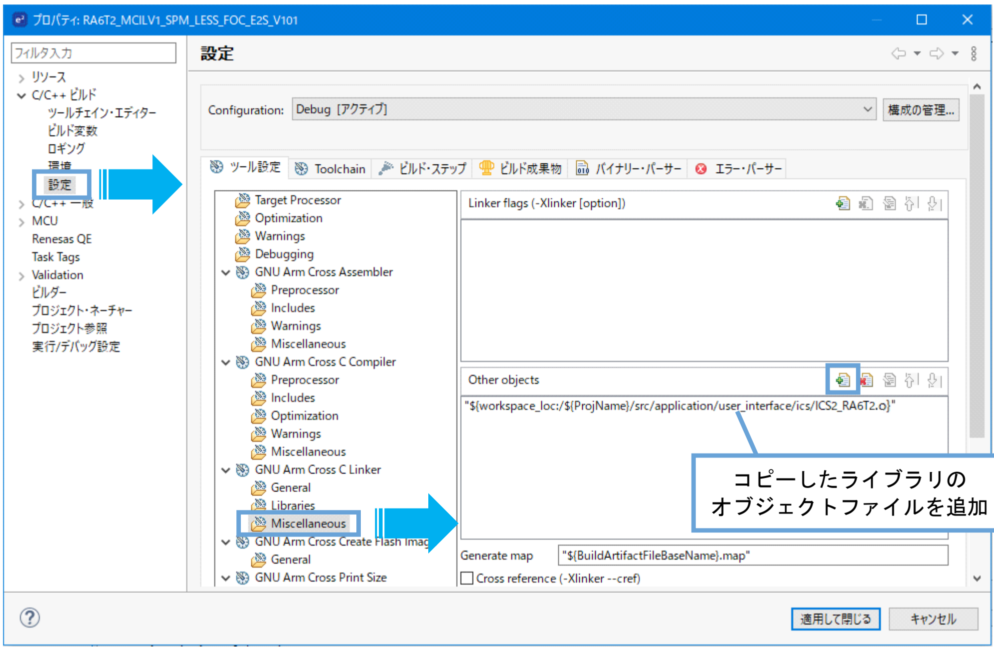


図 17-17　e² studio でライブラリのオブジェクトファイルを追加

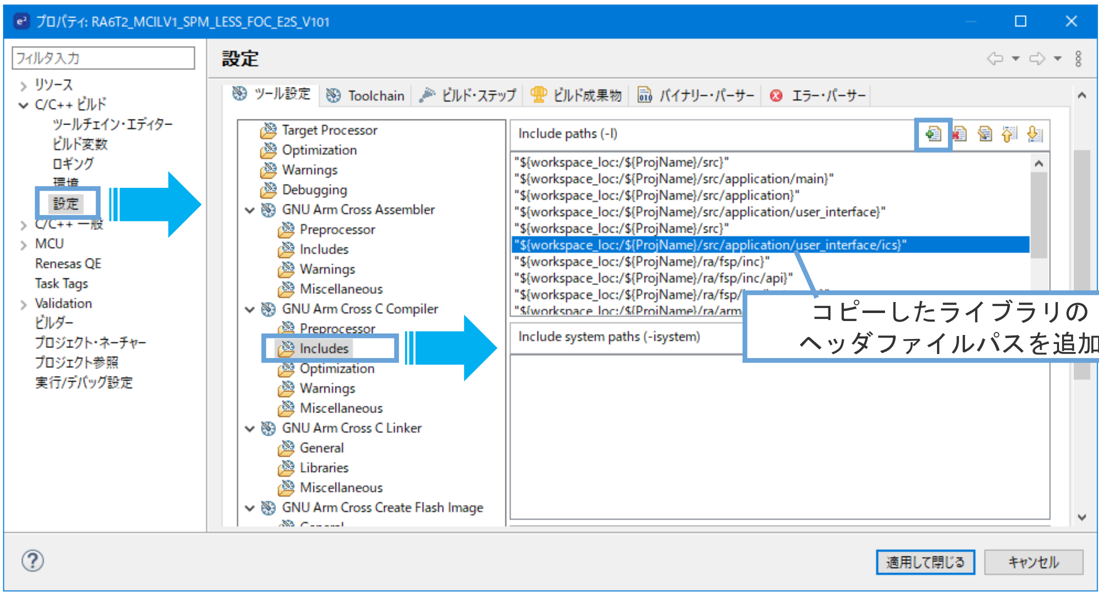


図 17-18　e² studio でライブラリのヘッダファイルパスを追加

17.7.4.3 ライブラリ関数の呼び出し

ライブラリの初期化関数では、17.7.2 章で決定した使用端子と、17.7.3 章で決定した通信速度を使用します。

RA6T2 用の通信ライブラリには、以下の 2 種類の関数が用意されています。

- void ics2_init(uint8_t port, uint8_t speed, uint8_t mode） ：初期化関数
- void ics2_watchpoint(void） ：周期的に呼び出すデータ転送関数

対象マイコンによって ics2_init()関数の引数が異なります。ics2_init()関数の引数 port に設定する端子は、ヘッダファイル ICS2_RA6T2.h に定義されています。
RA6T2 のモータ制御プログラムから、通信ライブラリ関数を呼び出す場合の例は以下のとおりです。

(1) 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
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_RA6T2.h"

void main(void)
{
      /* port  = SCI9_A (PD05/TXD9 and PD06/RXD9): defined in ICS2_RA6T2.h    */
      /* Rate  = (SCI Clock / (6 x (speed + 1)))[Mbps] :                      */
      /*           Rate = 1Mbps, SCISPICLK = 120MHz -> speed = 19             */
      /* mode  = 32bit x 8ch                                                  */
      ics2_init(ICS_SCI9_PD05_PD06, 19, 1);
}
```

図 17-19 初期化関数 ics2_init() の呼び出し（RA6T2 での記述例）

(2) 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。

```
/* Call the data transfer function cyclically                              */
/* If the transfer rate is Rate[Mbps], the minimum sampling cycle is       */
/* 70+(180/Rate) [μsec], and it must be called at a cycle greater than this. */

void int_TM50u(void)                    /* 50 μsec 間隔 */
{
      g_u1_cnt_decimation++;
      if (5 <= g_u1_cnt_decimation)     /* decimation of ICS call */
      {
            g_u1_cnt_decimation = 0;
```

```
            ics2_watchpoint();          /* data transfer */
        }
    }
```

図 17-20 データ転送関数 ics2_watchpoint() の呼び出し（RA6T2 での記述例）

17.7.5　プログラムのダウンロード

PC と CPU ボードを接続し、CPU ボード上の RA6T2 にモータ制御プログラムをダウンロードします。モータ制御プログラムを CPU ボードにダウンロードする際は、インバータボード、モータへの接続は不要です。

プログラムのダウンロード方法についての詳細は、使用する評価キットのクイックスタートガイドや、書き込みツールのマニュアルを参照してください。

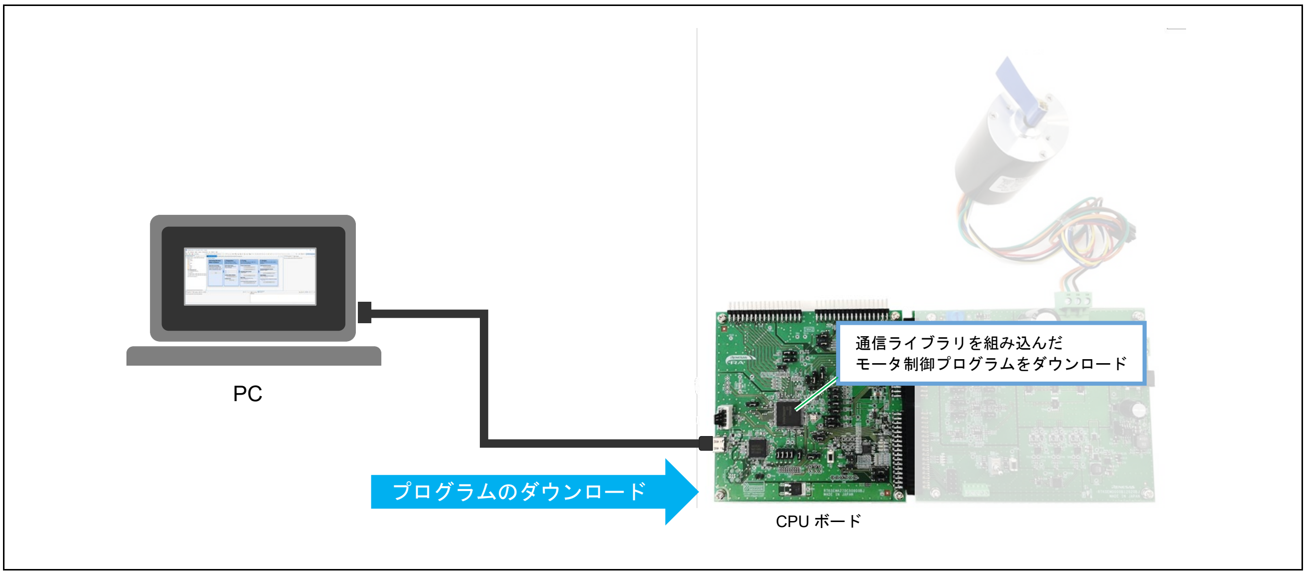


図 17-21　プログラムダウンロード時の接続

17.7.6 RMW と接続

RMW をインストールした PC と各ボードを図 17-22 のとおり接続します。CPU ボードと通信ボードの接続には、17.7.2 章で決定した端子を使用します。

RMW の通信速度を図 17-23 のとおり設定します。RMW の Main Window 画面の「Connection」で「Clock」ボタンをクリックし、17.7.3 章で決定した通信速度を設定します。RMW に設定する通信速度は、ライブラリの初期化関数 ics2_init()に設定した通信速度と同一である必要があります。

RMW に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

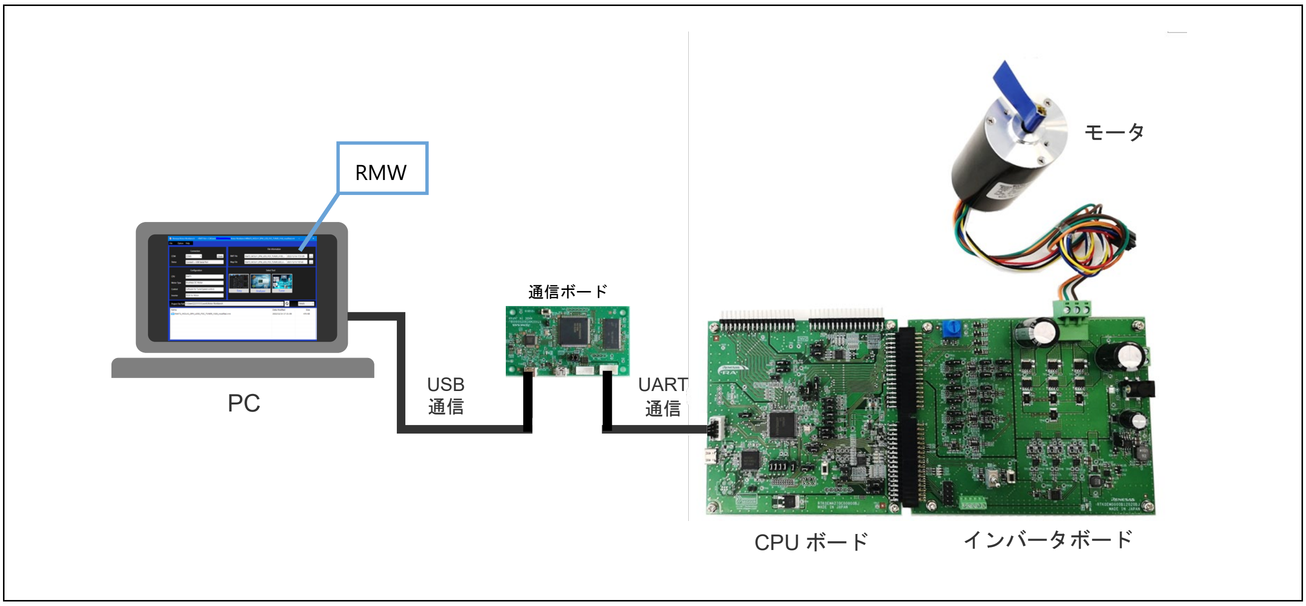


図 17-22 RMW 使用時の接続

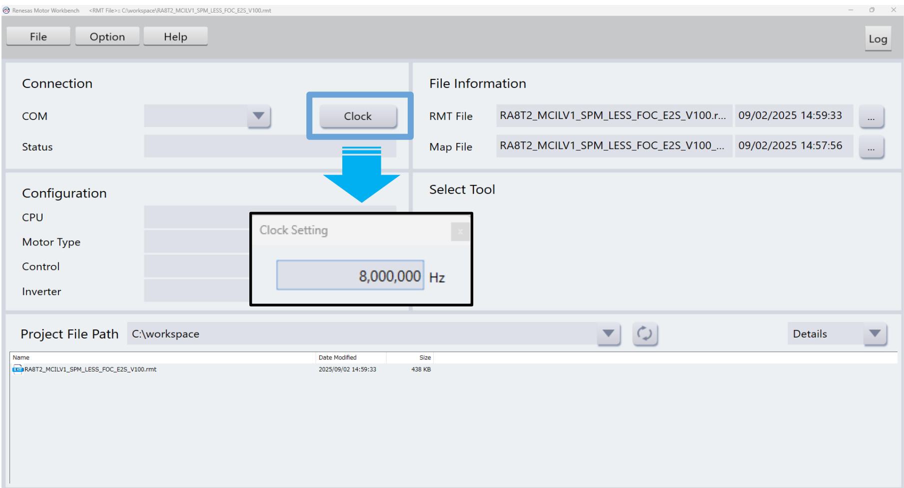


図 17-23 RMW の通信速度の設定

17.8 ビルトイン型通信ライブラリの使用例

モータ制御プログラムにビルトイン型通信ライブラリを取り込んで RMW を使用する場合の手順は、以下のとおりです。

- 使用端子の決定：USB シリアル変換ボードと MCU を搭載したボードの接続に使用する端子を決定
- 通信速度の決定：RMW とボード間の通信速度を決定
- モータ制御プログラムの作成
  - ➢ ビルトイン型通信ライブラリファイルをコピー
  - ➢ ビルトイン型通信ライブラリファイルを追加
  - ➢ ビルトイン型通信ライブラリ関数を呼び出し
- ビルトイン型通信ライブラリを取り込んだモータ制御プログラムを MCU にダウンロード
- RMW と接続

本章では、MCK-RA6T2 の MCK モータ制御評価キットで RMW を使用する場合の使用例を説明します。

ユーザボードで RMW を使用する場合の手順は、MCK モータ制御評価キットを使用する場合と同等です。

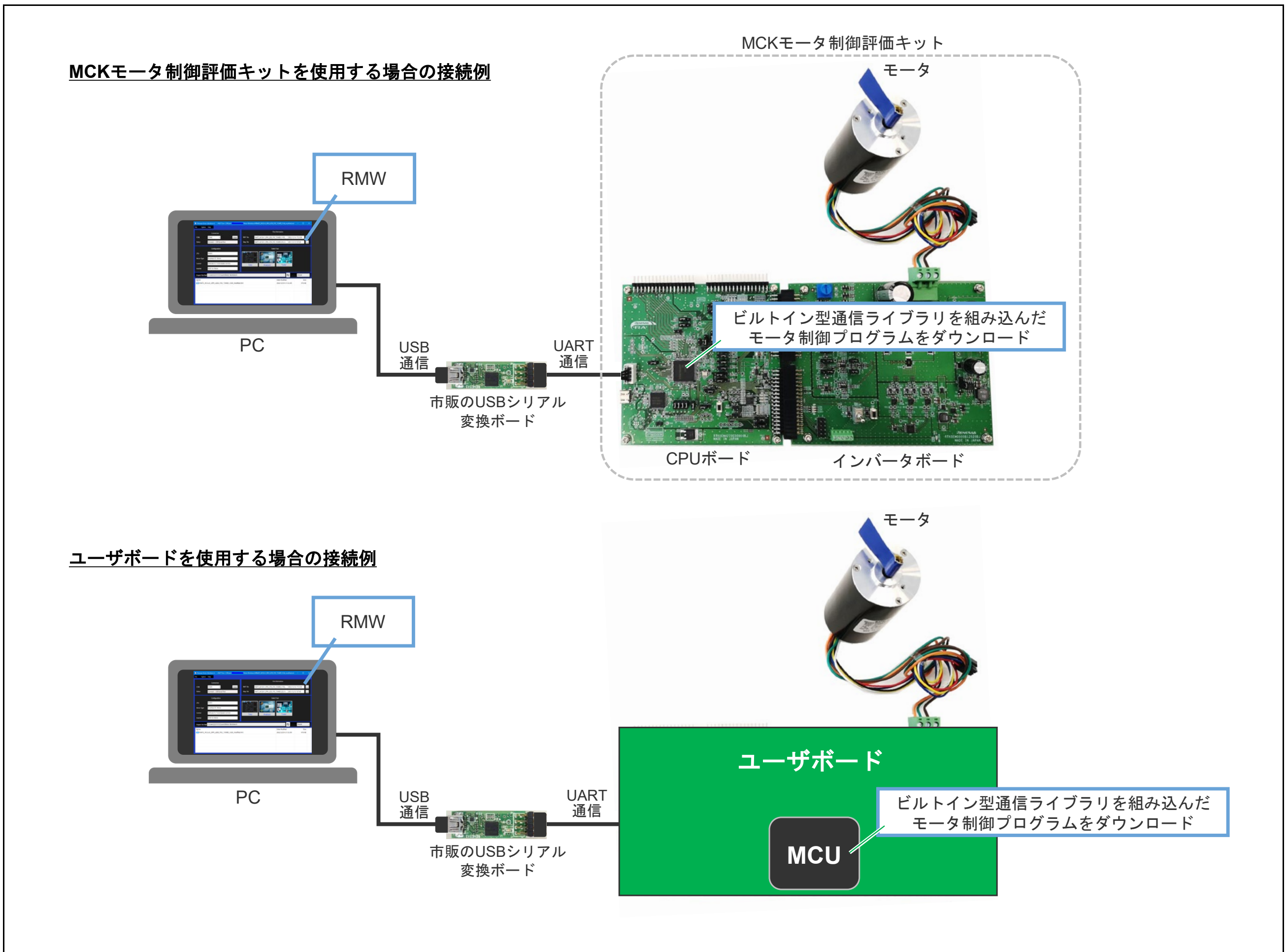


図 17-24 ビルトイン型通信ライブラリを取り込んで RMW を使用する場合の接続例

17.8.1 HW 準備

MCK-RA6T2 と市販の USB シリアル変換ボードを使用します。

ビルトイン型通信ライブラリを使用するため、MCK-RA6T2 に同梱されている通信ボードは使用しません。

MCK-RA6T2 についての詳細は、下記ドキュメントを参照してください。

- R12QS0047 MCK-RA6T2 クイックスタートガイド
- R12UZ0091 MCK-RA6T2 ユーザーズマニュアル

MCK-RA6T2同梱物

市販のUSBシリアル変換ボード

図 17-25 使用機材

17.8.2　　使用端子の決定

CPU ボードと USB シリアル変換ボードを接続する端子を決定します。使用端子は、下記の条件を満たす必要があります。

- ライブラリで使用できる端子であること
- CPU ボードの該当端子が USB シリアル変換ボードと接続できること

MCK-RA6T2 の場合、図 17-26 のとおり SCI9 と PD05/TXD9_A と PD06/RXD9_A を使用します。

本章で決定した使用端子は、以下の章で使用します。

- 17.8.4.3 章 ライブラリ関数の呼び出し：ics2_init()の引数 port に設定
- 17.8.6 章 RMW と接続　　　　　　　　：USB シリアル変換ボードとの接続

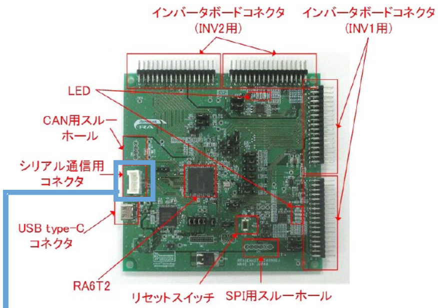


表 6-5 SCI コネクタ（CN10）ピンアサイン

端子 No.	端子機能	RA6T2 接続端子
1	GND	-
2	マイコン受信側	PD06/RXD9_A
3	マイコン送信側	PD05/TXD9_A
4	VCC	-

SCI9-A

USB シリアル変換ボードと
接続できる端子

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 XD0:P411 XD0:P410 XD9: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		

ビルトイン型通信ライブラリで
使用できる端子

図 17-26　使用端子の決定

17.8.3 通信速度の決定

PC と USB シリアル変換ボード、および CPU ボードの通信速度を決定します。通信速度は、以下の条件を満たす必要があります。

- ライブラリで使用できる通信速度であること
- USB シリアル変換ボードに設定可能な通信速度であること

USB シリアル変換ボードに設定可能な通信速度は、図 17-27 のとおり PC のデバイスマネージャのプロパティから確認することができます。

本章で決定した通信速度は、下記の章で使用します。

- 17.8.4.3 章 ライブラリ関数の呼び出し：ics2_init()の引数 speed に設定
- 17.8.6 章 RMW と接続 ：RMW に通信速度を設定

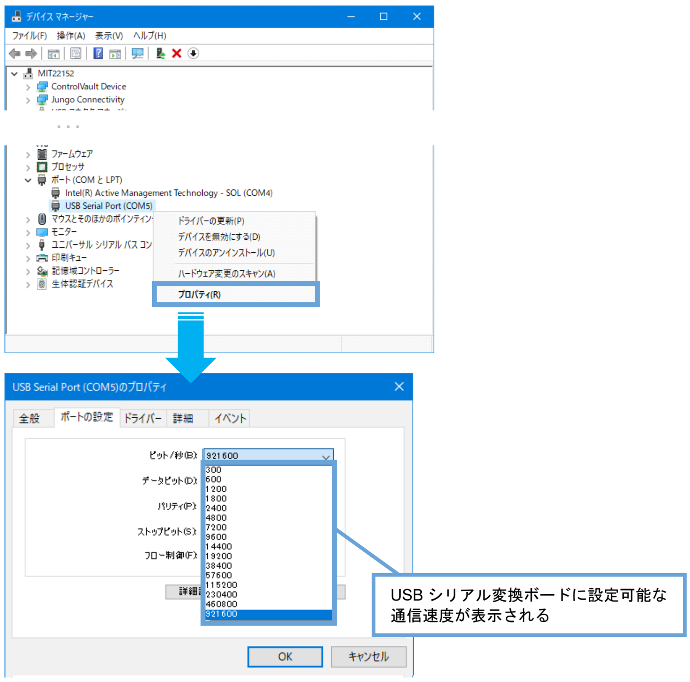


図 17-27 RMW で設定可能な通信速度の確認

17.8.4 プログラムの作成

MCK-RA6T2 の CPU ボードに搭載された RA6T2 にダウンロードする、モータ制御プログラムを作成します。

本章では、すでに e[2] studio で作成されたモータ制御プログラムに、ビルトイン型通信ライブラリを取り込む方法を説明します。

17.8.4.1 ライブラリファイルのコピー

ビルトイン型通信ライブラリのヘッダファイルとオブジェクトファイルは、RMW インストーラをダウンロードしたファイルに同梱されています。ビルトイン型通信ライブラリのヘッダファイルとオブジェクトファイルを、モータ制御プログラムの任意の場所にコピーします。

ライブラリファイルをコピーしたフォルダ構成を、サンプルコードを例に図 17-28 に示します。

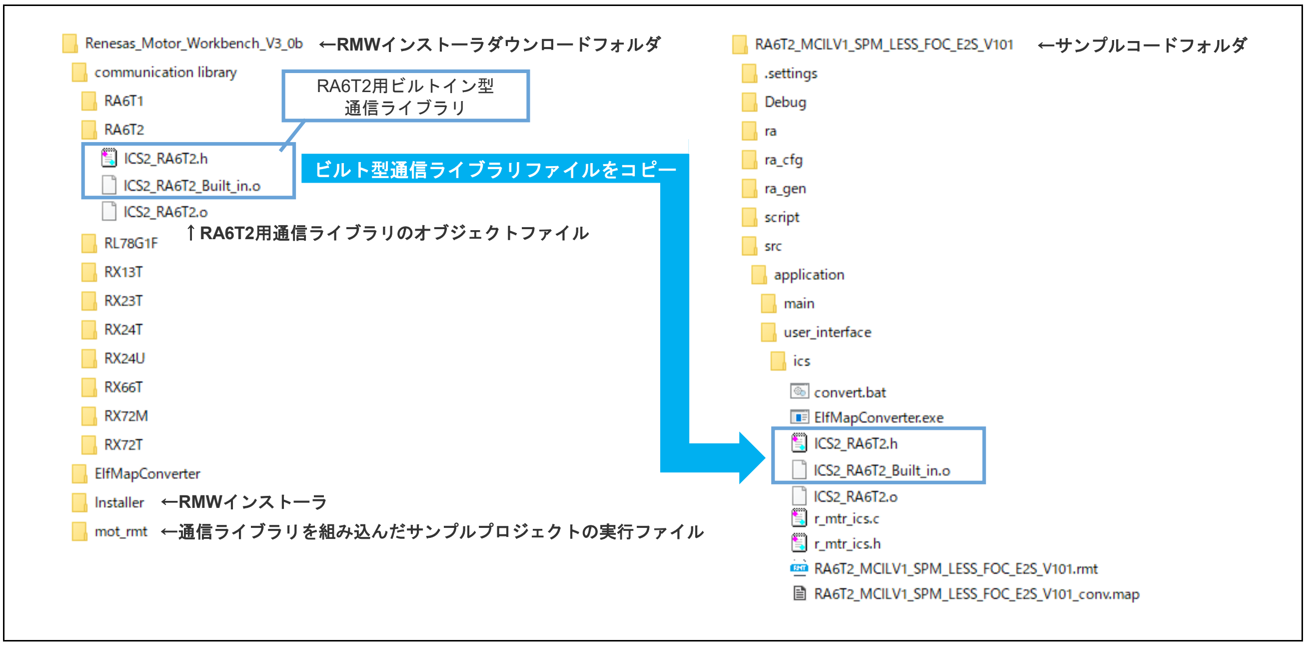


図 17-28 ライブラリファイルをコピーしたフォルダ構成例

17.8.4.2　ライブラリファイルの追加

コピーしたライブラリファイルを e2 studio で追加します。e2 studio でプロジェクトのプロパティを開き、オブジェクトファイルを図 17-29 のとおり、ヘッダファイルパスを図 17-30 のとおり追加します。

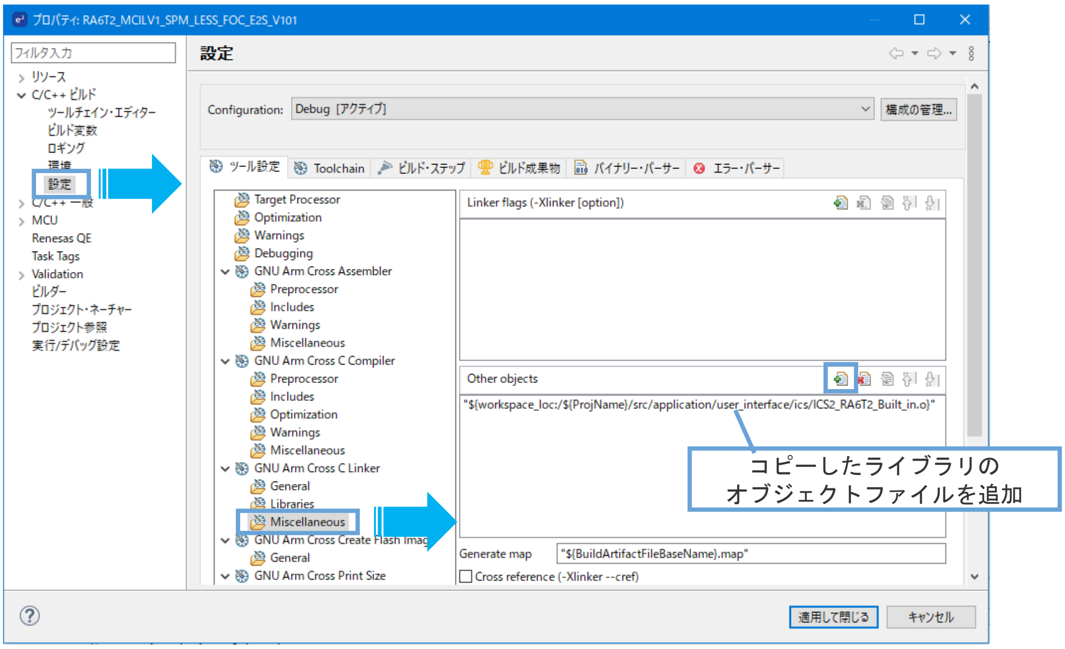


図 17-29　e2 studio でライブラリのオブジェクトファイルを追加

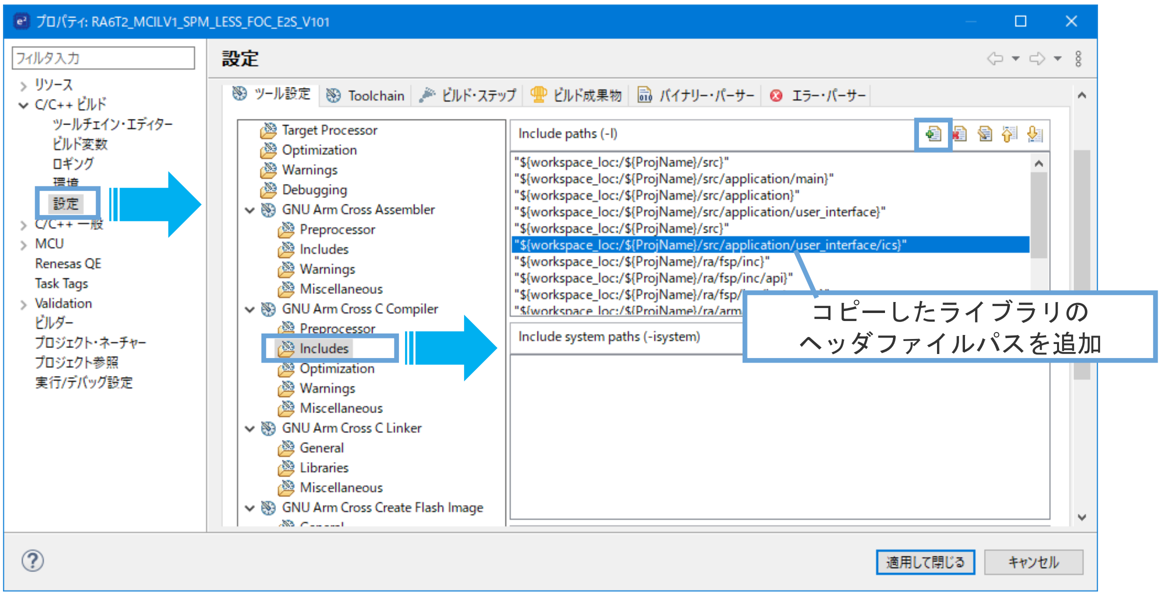


図 17-30　e2 studio でライブラリのヘッダファイルパスを追加

17.8.4.3 ライブラリ関数の呼び出し

ライブラリの初期化関数では、17.8.2 章で決定した使用端子と、17.8.3 章で決定した通信速度を使用します。

RA6T2 用のビルトイン型通信ライブラリには、以下の 2 種類の関数が用意されています。

- void ics2_init(uint8_t port, uint8_t speed, uint8_t mode） ：初期化関数
- void ics2_watchpoint(void） ：周期的に呼び出すデータ転送関数

対象マイコンによって ics2_init()関数の引数が異なります。ics2_init()関数の引数 port に設定する端子は、ヘッダファイル ICS2_RA6T2.h に定義されています。

RA6T2 のモータ制御プログラムから、ビルトイン型通信ライブラリ関数を呼び出す場合の例は以下のとおりです。

```
#include "ICS2_RA6T2.h"
void main(void)
{
    /* port  = SCI9_A (PD05/TXD9 and PD06/RXD9): defined in ICS2_RA6T2.h     */
    /* Rate  = (SCI Clock / (6 x (speed + 1)))[Mbps] :                       */
    /*          Rate = 921600bps, SCISPICLK = 120MHz -> speed = 21            */
    /* mode  = 32bit x 8ch                                                   */
    ics2_init(ICS_SCI9_PD05_PD06, 21, 1);
}

/* Call the data transfer function cyclically                                */
void timer_isr(void)
{
    ics2_watchpoint();
}
```

17.8.5　　プログラムのダウンロード

PC と CPU ボードを接続し、CPU ボード上の RA6T2 にモータ制御プログラムをダウンロードします。モータ制御プログラムを CPU ボードにダウンロードする際は、インバータボード、モータへの接続は不要です。

プログラムのダウンロード方法についての詳細は、使用する評価キットのクイックスタートガイドや、書き込みツールのマニュアルを参照してください。

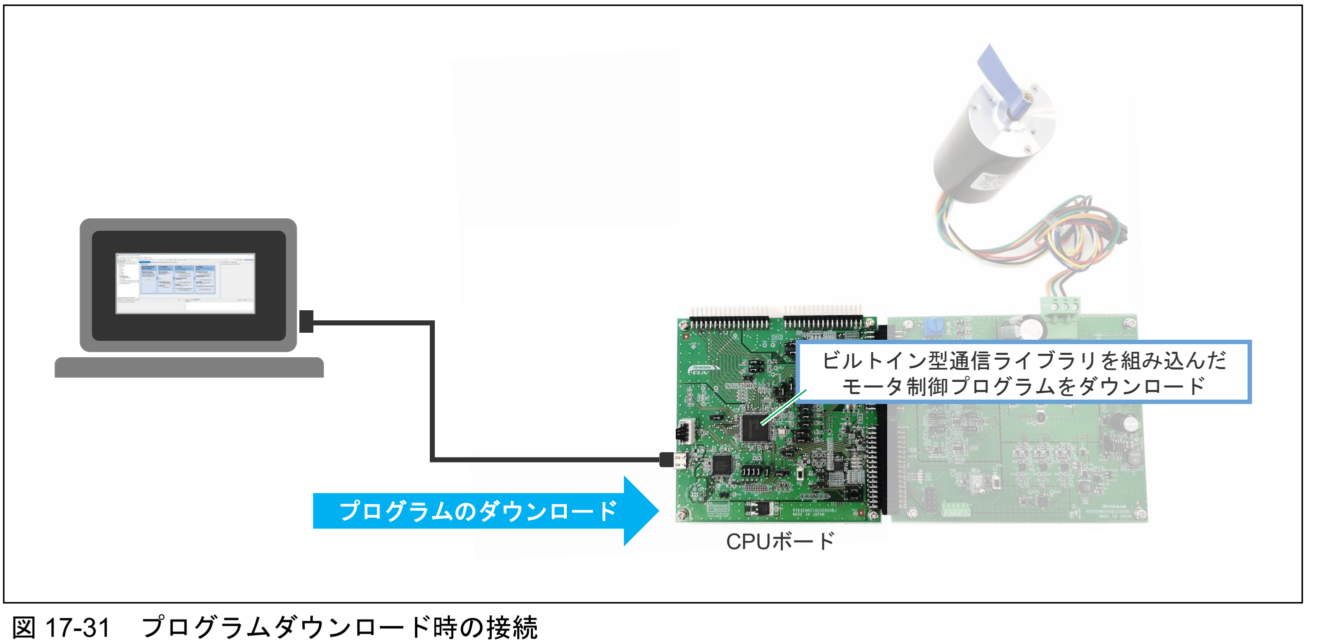


図 17-31　プログラムダウンロード時の接続

17.8.6　　RMW と接続

RMW をインストールした PC と各ボードを図 17-32 のとおり接続します。CPU ボードと USB シリアル変換ボードの接続には、17.8.2 章で決定した端子を使用します。

RMW の通信速度を図 17-33 のとおり設定します。RMW の Main Window 画面の「Option」メニューから「Baudrate Dialog」を選択し 17.8.3 章で決定した通信速度を設定します。RMW に設定する通信速度は、ライブラリの初期化関数 ics2_init()に設定した通信速度と同一である必要があります。

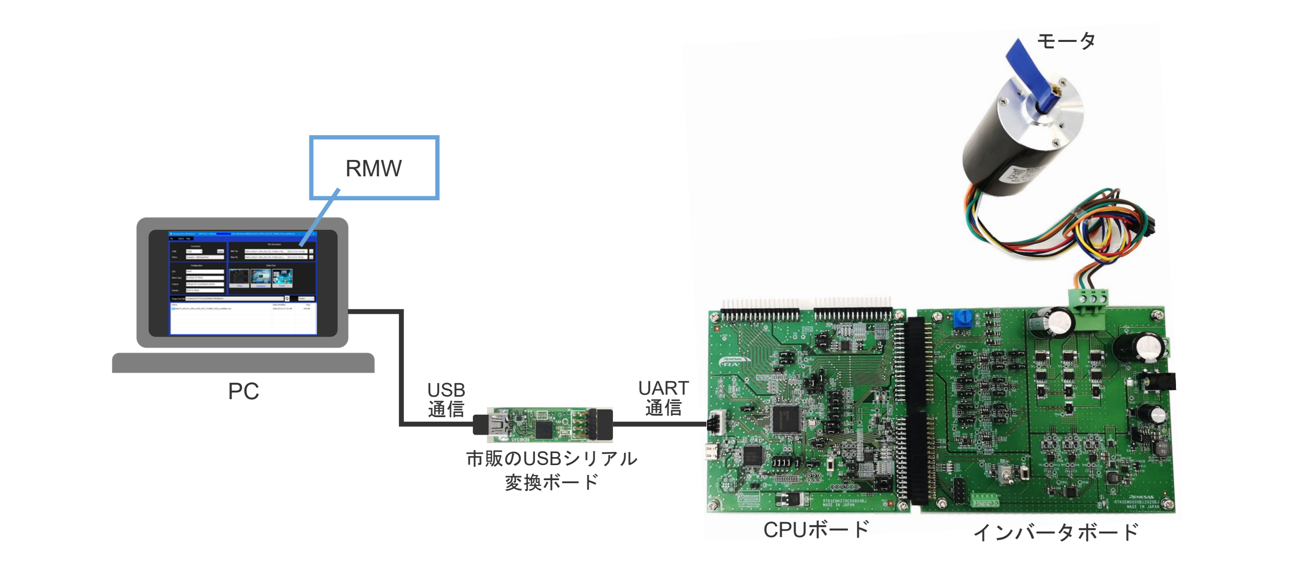


図 17-32　RMW 使用時の接続

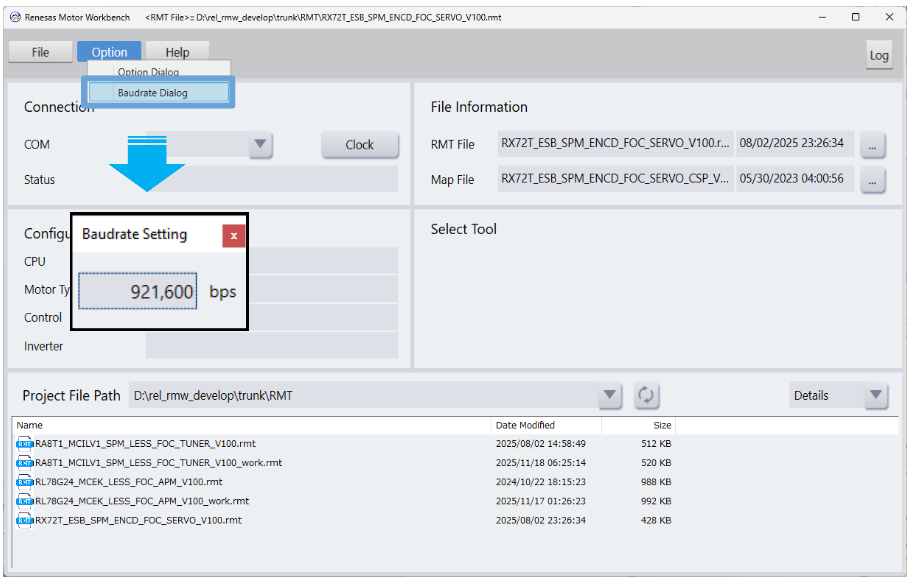


図 17-33　RMW の通信速度の設定

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

https://www.renesas.com/

お問合せ先

https://www.renesas.com/contact

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
4.00	May.30.23	―	• Rev3.00 記載内容をクイックスタートガイドへ移行（R21QS0011） • 改訂新版発行 • 3.1 版新機能(10 章、11 章、13 章)を追加 • 3.1 版新機能追加に伴い 1 章、2 章、3 章、4 章の記載内容見直し
4.01	Oct.27.23	―	• 3.1.1 版「1.2 小数点の表示と入力」追加 • 「2.4.1.1 新しい認証ファイルの入手」認証ファイルの入手先 URL を Renesas Motor Workbench の web サイトに変更
4.02	Jan.30.24	―	• RA8T1 を追加 • 「13. Servo」の内容を改訂 • 「14.6 Tuner Library」を追加
4.10	Feb.07.25	―	• 「2.4.1 認証ファイルの読み込み」を削除 • 「15 Tuner ツール（RL78 向け）」を追加 • 「16 Tuner ツール（MCI-HV-1 向け）」を追加
4.11	Nov.10.25	―	RA8T2、RA2T1 を追加
4.20	Dec.15.25	―	• 「2.3.7 ログ表示」を追加 • 「8.3.3 Status Indicator Setting」の内容を改定 • 「14. Tuner ツール」の内容を改定 • 「15. Tuner ツール（RL78 向け）」の内容を改定 • 「16. Tuner ツール（MCI-HV-1 向け）」の内容を改定 • 「17.通信ライブラリの使用例」の内容を改定

備考

- 本書は、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としたものです。

参考：旧 Renesas Motor Workbench 機能説明書（R20AN0527JJ）改訂履歴

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Nov.27.18	―	新規発行
2.00	Dec.09.21	―	Renesas Motor Workbench3.0 リリースにより改訂

参考：旧 Renesas Motor Workbench Tuner 機能説明書（R20AN0528）改訂履歴

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Nov.27.18	―	新規発行
2.00	Dec.09.21	―	Renesas Motor Workbench3.0 リリースにより改訂

- 従来の Renesas Motor Workbench ユーザーズマニュアル（R21UZ0004）は、
  - Renesas Motor Workbench クイックスタートガイド（R21QS0011）

  になりました。本書と合わせてご利用ください。

モータ制御開発支援ツール Renesas Motor Workbench 3.3.0
ユーザーズマニュアル

発行年月日　　2025 年 12 月 15 日 Rev.4.20

発行　　ルネサス エレクトロニクス株式会社
〒135-0061　東京都江東区豊洲 3-2-24（豊洲フォレシア）

モータ制御開発支援ツール
Renesas Motor Workbench 3.3.0

ルネサス エレクトロニクス株式会社

R21UZ0004JJ0420