

改訂新版

モータ制御開発支援ツール Renesas Motor Workbench 3.1.2 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。

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しないでください。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、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っていると、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

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

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

1. Renesas Motor Workbench の概要	1
1.1 RMW の機能	1
1.2 小数点の表示と入力	2
1.3 関連資料	2
2. Main Window	3
2.1 概要	3
2.2 ウィンドウ構成	3
2.3 Main Window 各部の説明	4
2.3.1 メニューバー	4
2.3.2 Connection 画面	6
2.3.3 Configuration	7
2.3.4 File Information	7
2.3.5 Select Tool	8
2.3.6 Project File Path と File List	9
2.4 Main Window の操作	10
2.4.1 認証ファイルの読み込み	10
2.4.2 RMT ファイルの読み込み	13
2.4.3 RMT ファイルの保存	15
2.4.4 Map ファイル（変数リスト）の読み込み	16
2.4.5 Option メニューの「Option Dialog」	17
2.4.6 USB の接続	20
2.4.7 ウィンドウの基本操作	23
2.4.8 バージョンの確認方法	25
2.5 Select Navigation 機能	26
3. Analyzer ツール	27
3.1 概要	27
3.2 Analyzer ツールの構成	28
3.3 Analyzer ツールの外観	29
4. [Analyzer] Control Window	30
4.1 概要	30
4.2 特徴	30
4.3 画面構成	30
4.3.1 操作ボタン	31
4.3.2 Variable Data タブ	32
4.3.3 Variable List タブ	33
4.3.4 Alias Name タブ	34
4.4 変数の設定（Variable Data タブ）	35
4.4.1 Variable Name の設定	35
4.4.2 Data Type の設定	37
4.4.3 Scale の設定	37
4.4.4 Base の設定	38
4.4.5 変数情報の Note 欄	38

4.4.6	変数値の読み込み.....	39
4.4.7	変数値の書き込み.....	40
4.4.8	変数表示の並び替え.....	41
4.4.9	変数表示の背景色変更.....	41
4.4.10	変数の削除.....	42
4.4.11	変数情報の保存／読み込み.....	43
4.5	変数の一覧表示 (Variable List タブ).....	44
4.6	別名の設定 (Alias Name タブ).....	45
5.	[Analyzer] Scope Window.....	47
5.1	概要.....	47
5.2	特徴.....	47
5.3	画面構成.....	48
5.3.1	波形表示部.....	49
5.3.2	チャンネル情報表示部.....	51
5.3.3	計測設定部.....	52
5.3.4	Window-Channel Settings 画面.....	53
5.3.5	Image Editor 画面.....	55
5.4	測定チャンネルの設定.....	56
5.4.1	Window-Channel Settings の表示.....	56
5.4.2	Channel Settings 部の設定.....	56
5.5	Math 機能.....	58
5.5.1	Math Channel Setting 部の設定.....	58
5.5.2	波形データの扱い.....	62
5.5.3	Math チャンネル波形の描画.....	62
5.6	波形の計測.....	63
5.6.1	トリガ設定部.....	63
5.6.2	計測設定部の Acquisition 設定.....	65
5.6.3	波形計測の開始と停止.....	66
5.6.4	波形データの保存と読み込み.....	67
5.7	波形の表示.....	68
5.7.1	Scope Window の表示.....	68
5.7.2	波形グラフの表示切り替え.....	68
5.7.3	Main1 と Main2 波形グラフの切り替え.....	68
5.7.4	Zoom 波形グラフの範囲指定.....	69
5.7.5	チャンネル情報表示部の分離.....	70
5.7.6	Zoom 画面からの Main 波形グラフの分離.....	71
5.7.7	Active Channel 表示.....	72
5.7.8	カーソル表示.....	72
5.7.9	波形表示ポイント数の間引き.....	73
5.7.10	波形のスモーキング.....	73
5.7.11	画面背景色の変更.....	74
5.7.12	ショートカットキー.....	75
5.8	Image Editor 機能 (Scope Capture ボタン).....	76
5.8.1	Image Editor の起動／終了.....	76
5.8.2	表示位置とサイズの変更.....	76
5.8.3	Image Editor の操作ボタン.....	77
5.8.4	PDF 出力領域の調整.....	78

6.	[Analyzer] User Button Window.....	79
6.1	概要	79
6.2	特徴	79
6.3	画面構成.....	79
6.3.1	実行ボタン部.....	80
6.3.2	シーケンス編集部.....	81
6.3.3	ボタン統合部.....	82
6.4	操作説明.....	83
6.4.1	User Button の新規作成	83
6.4.2	User Button の削除.....	83
6.4.3	User Button の表示／非表示	83
6.4.4	シーケンスの編集.....	83
6.4.5	内部変数の利用	87
6.4.6	画面の表示設定	87
7.	[Analyzer] Commander 画面	89
7.1	概要	89
7.2	特徴	89
7.3	画面構成.....	89
7.3.1	操作ボタン.....	90
7.3.2	Write Data タブ	91
7.3.3	Result List タブ.....	92
7.3.4	ステータスバー	92
7.4	操作説明.....	93
7.4.1	Commander の起動／終了	93
7.4.2	Send Checker の実行.....	93
7.4.3	CSV ファイルの指定.....	94
7.4.4	CSV ファイルの編集（「CSV Edit」ボタン）	95
7.4.5	シーケンス実行の準備	99
7.4.6	シーケンスの実行.....	100
7.4.7	シーケンス実行の結果	101
8.	[Analyzer] Status Indicator 画面	102
8.1	概要	102
8.2	特徴	102
8.3	画面構成.....	102
8.3.1	Status Indicator Menu	103
8.3.2	Status Indicator Detail	104
8.3.3	Status Indicator Setting	105
8.4	操作説明.....	106
8.4.1	Menu 画面の表示／非表示	106
8.4.2	Setting 画面（監視条件の設定）	106
8.4.3	監視の開始.....	107
8.4.4	監視の停止.....	107
8.4.5	監視中の表示	107
8.4.6	監視結果のクリア.....	108
8.4.7	監視中にできない操作	108

8.4.8	RMT ファイルへの保存	108
9.	[Analyzer] One Shot 画面	109
9.1	概要	109
9.2	特徴	109
9.3	画面構成	109
9.3.1	チャンネル情報部	110
9.3.2	取得設定部	110
9.4	操作説明	111
9.4.1	One Shot 画面の表示	111
9.4.2	チャンネル情報部の設定	112
9.4.3	データの取得	113
9.4.4	カーソルの設定	113
9.4.5	取得データの保存	113
10.	[Analyzer] Parameter Output	114
10.1	概要	114
10.2	特徴	114
10.3	画面構成	114
10.3.1	出力変数選択画面	115
10.3.2	出力内容確認画面	116
10.4	操作説明	117
10.4.1	Parameter Output 画面の表示	117
10.4.2	出力する変数の選択	118
10.4.3	マクロ名の入力	118
10.4.4	出力内容の確認	119
10.4.5	ヘッダファイルの出力	119
10.4.6	ヘッダテンプレートファイルの設定	120
11.	Navigation	123
11.1	概要	123
11.2	特徴	123
11.3	画面構成	123
11.3.1	Navigation 画面	123
11.4	操作説明	124
11.4.1	表示機能の切り替え	124
11.4.2	操作手順の選択	124
11.4.3	画像拡大率の変更	125
11.4.4	自動起動の変更	125
11.4.5	ページの移動	127
12.	Easy GUI	128
12.1	概要	128
12.2	特徴	128
12.3	画面説明	129
12.4	事前準備	130

12.4.1	Easy GUI 画面表示.....	130
12.4.2	GUI 表示変数の設定.....	130
12.5	操作説明.....	131
12.5.1	基本機能.....	131
12.5.2	スライダ機能.....	132
12.5.3	シーケンス機能.....	133
12.5.4	計測機能.....	136
12.6	変数の設定.....	137
12.6.1	RUN/STOP ボタン.....	138
12.6.2	ステータス表示.....	139
12.6.3	パラメータテーブル.....	140
12.6.4	スライダ.....	141
12.6.5	シーケンス.....	142
12.6.6	シーケンスグラフィカル入力.....	143
12.6.7	オプションスイッチ.....	144
12.6.8	メータ.....	145
12.6.9	グラフ.....	146
12.7	テーマ色切替え.....	147
12.8	チュートリアル.....	148
13.	Servo ツール.....	149
13.1	概要.....	149
13.1.1	サーボ制御とは.....	149
13.1.2	Servo ツールとは.....	151
13.2	特徴.....	152
13.3	イナーシャ推定.....	152
13.3.1	機能説明.....	152
13.3.2	画面構成・パラメーター一覧.....	155
13.3.3	操作説明.....	157
13.4	サーボ調整.....	159
13.4.1	機能説明.....	159
13.4.2	画面構成・パラメーター一覧.....	161
13.4.3	操作説明.....	164
13.5	原点復帰.....	166
13.5.1	機能説明.....	166
13.5.2	画面構成・パラメーター一覧.....	168
13.5.3	操作説明.....	170
13.6	Point to Point 動作.....	172
13.6.1	機能説明.....	172
13.6.2	画面構成・パラメーター一覧.....	173
13.6.3	操作説明.....	176
13.7	Waveform 画面.....	178
14.	Tuner ツール.....	179
14.1	概要.....	179
14.2	特徴.....	179

14.3	画面構成	180
14.4	操作説明	181
14.4.1	事前準備	181
14.4.2	チューニングの実行	182
14.4.3	動作の確認	185
14.4.4	Tuner ツールの終了	187
14.5	機能説明	188
14.5.1	Skip 機能 (Parameter Input 画面)	188
14.5.2	制御パラメータの調整 (Tune Window)	189
14.5.3	リセット機能 (Tune Window)	190
14.5.4	Header 出力 (Tune Window)	191
14.5.5	PDF 出力 (Tune Window)	192
14.5.6	波形情報の表示 (Scope 画面)	193
14.5.7	Memory 機能 (Scope 画面)	194
14.6	Tuner Library 組み込み手順	195
14.6.1	Tuner Library 全体構成	195
14.6.2	Tuner Library API 一覧	196
14.6.3	Tuner Library マクロ一覧	200
14.6.4	Tuner Library 組み込み手順	202
15.	ビルトイン型通信ライブラリ	227
15.1	概要	227
15.2	特徴	228
15.3	HW 構成	228
15.3.1	MCK モータ制御評価キットを使用する場合	229
15.3.2	Evaluation System を使用する場合	230
15.3.3	ユーザボードを使用する場合	231
15.4	使用例	232
15.4.1	HW 準備	233
15.4.2	使用端子の決定	234
15.4.3	通信速度の決定	235
15.4.4	プログラムの作成	236
15.4.5	プログラムのダウンロード	239
15.4.6	RMW と接続	240
	ホームページとサポート窓口	241
	改訂記録	242

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。

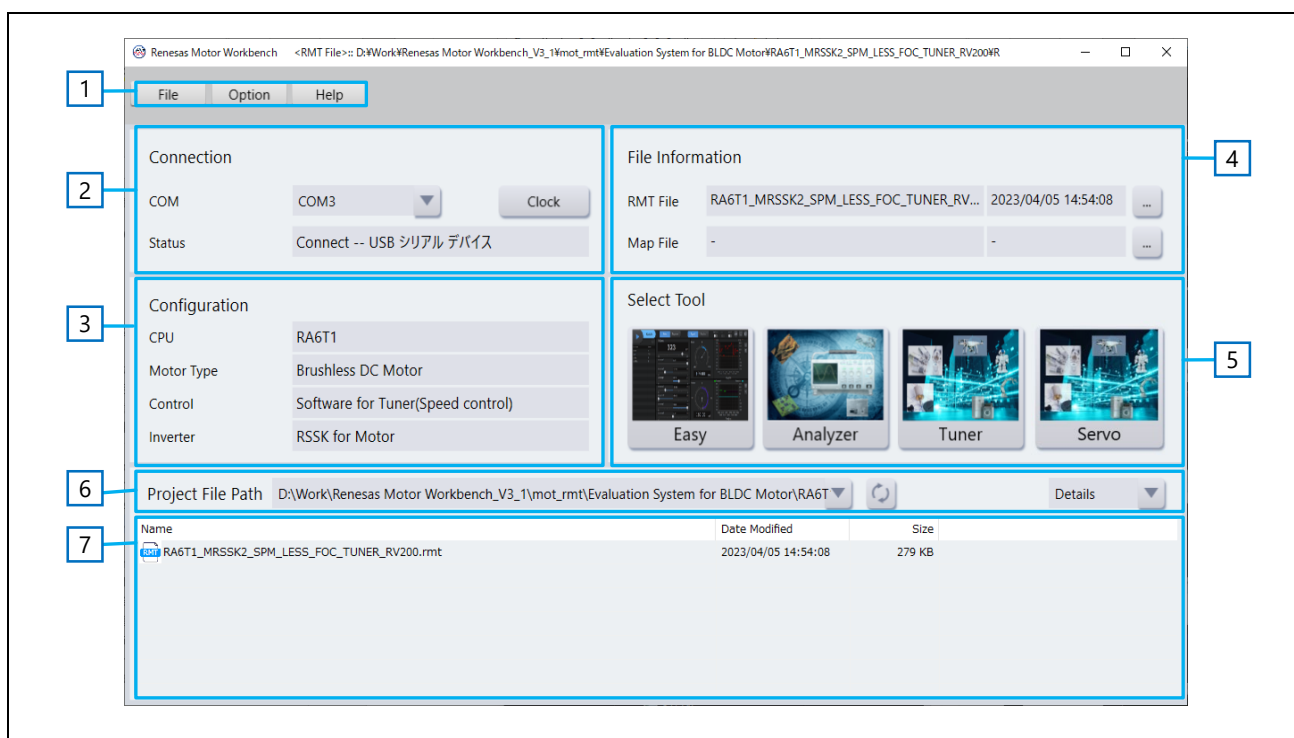


図 2.1 Main Window の画面構成

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	

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。

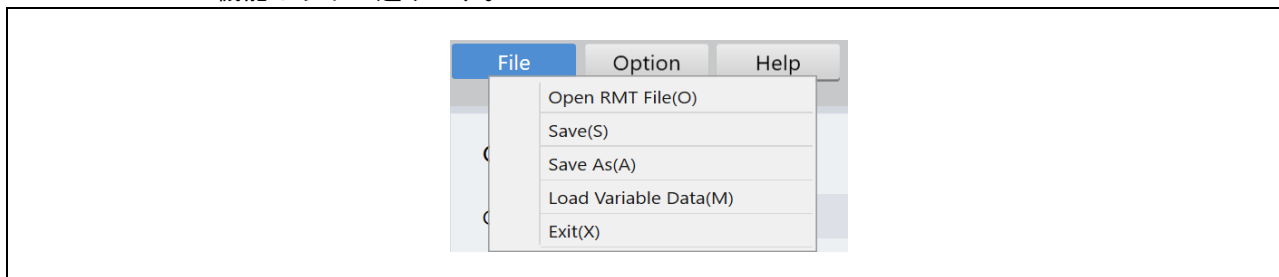


図 2.2 File メニュー

表 2.2 File メニューの機能

名称	説明
Open RMT File	RMT ファイル [※] 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。

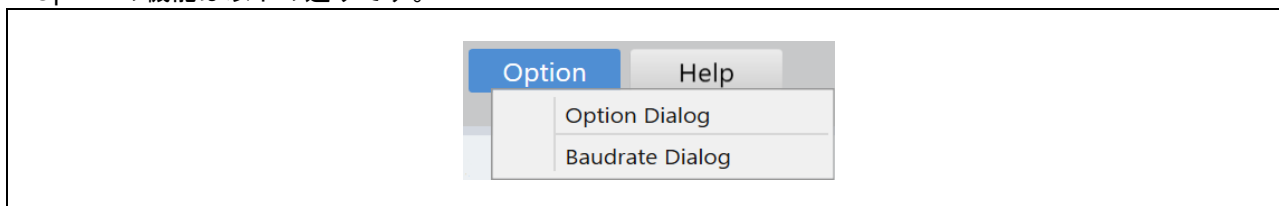


図 2.3 Option メニュー

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。

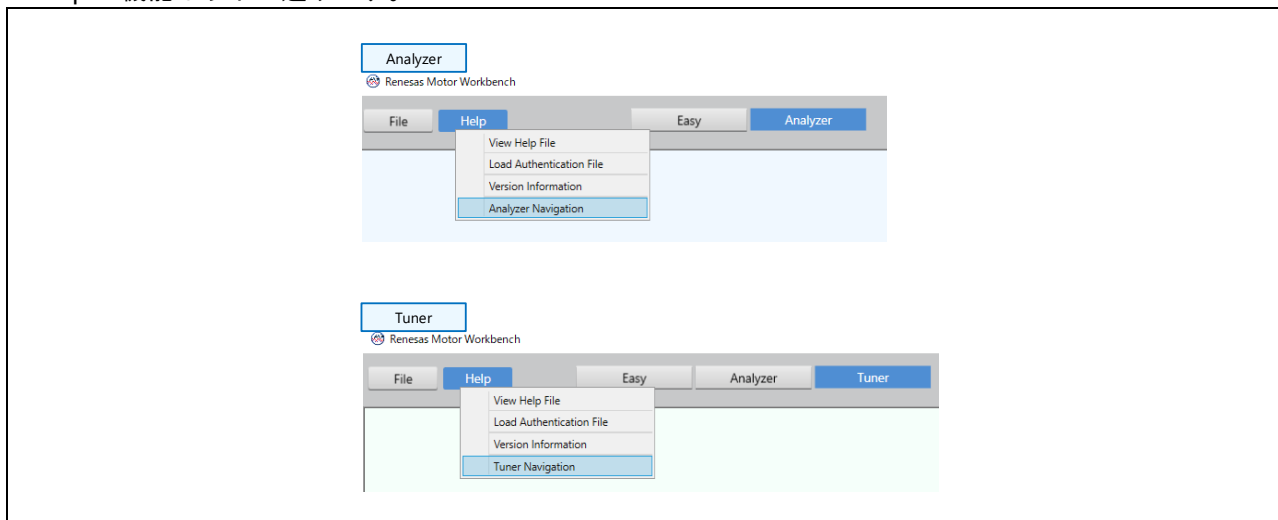


図 2.4 Help メニュー

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。
Load Authentication file	RMW を利用するには、認証ファイルが必要です。このメニューから認証ファイルを読み込むことで、本ツールが利用可能な状態となります。
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。



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 [※] 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。

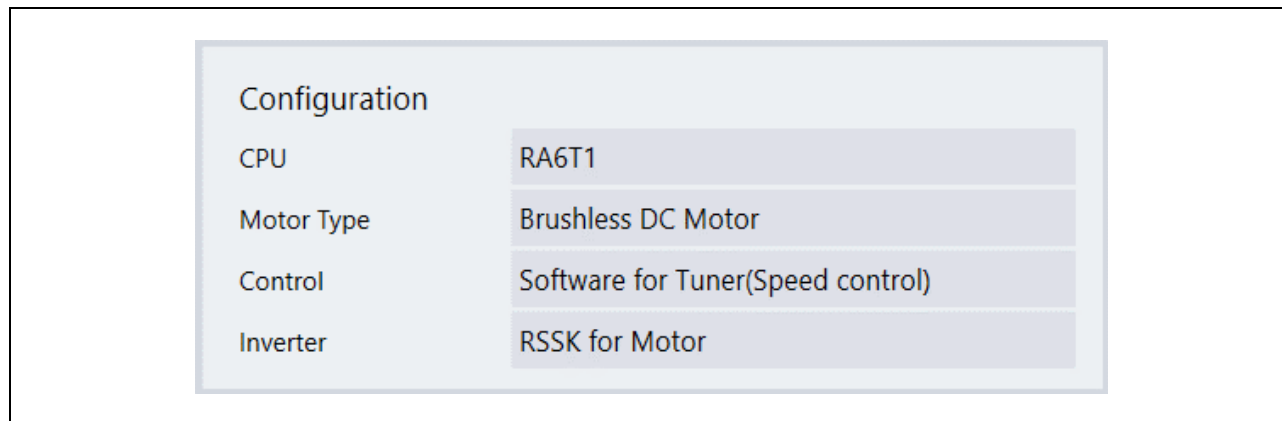


図 2.6 Configuration 画面

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。

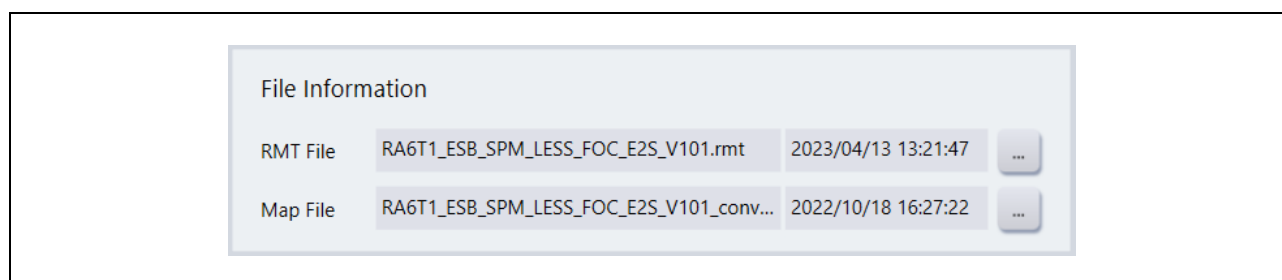


図 2.7 File Information 画面

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。

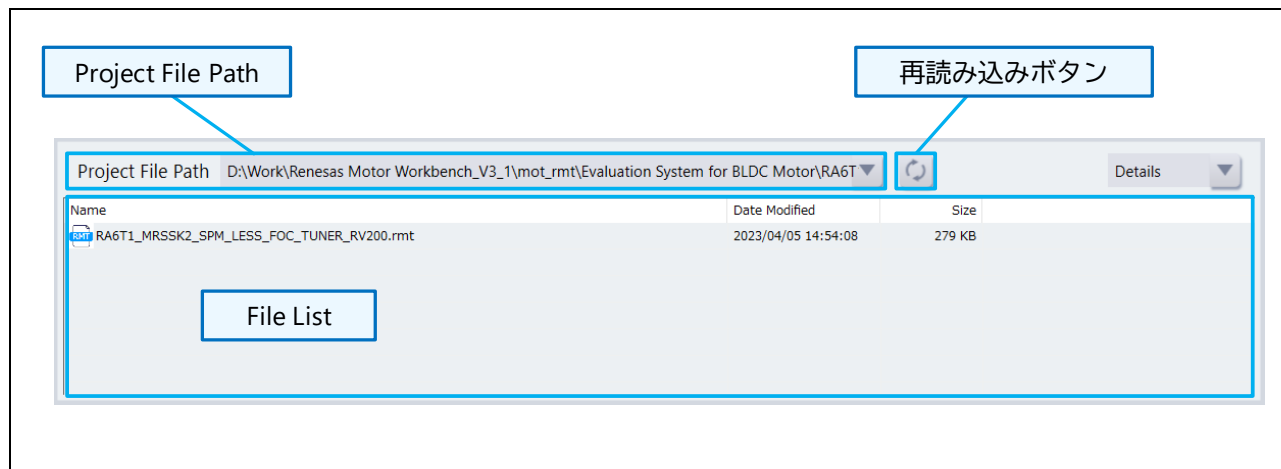


図 2.9 Project File Path、File List 画面とプロジェクトフォルダの関係

表 2.10 Project File Path と File List の機能

名称	説明
Project File Path	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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： <ul style="list-style-type: none"> 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。

2.4 Main Window の操作

2.4.1 認証ファイルの読み込み

RMW を利用するには、認証ファイル (*.atf) の読み込みが必要です。認証状況や Help メニュー操作から認証メッセージが画面に表示されます。

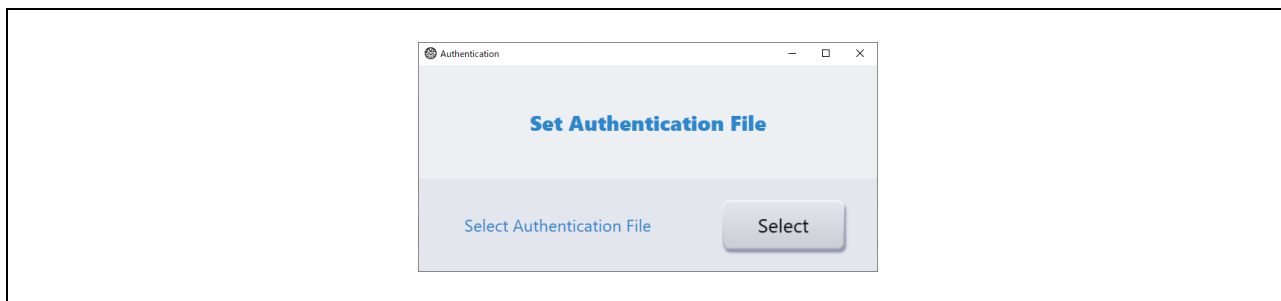


図 2.10 認証画面のメッセージ表示例

表 2.11 認証関連のメッセージとその対処方法

メッセージ	状況	RMW の利用	対処方法
Termfile not exist	読み込み済みの認証情報が見つからない場合に表示されます。	不可	認証ファイルを読み込んでください。
Set Authentication file	Help メニューの「Load Authentication File」選択時に表示されます。	可	
License expires soon	認証期限が残り 30 日以下となった場合に表示されます。	可	利用期限後も継続利用する場合は、新しい認証ファイルを取得して読み込んでください。
Expiration data out	認証の利用期限が切れた場合に表示されます	不可	
Expired File	読み込んだ認証ファイルが無効な場合に表示されます。	不可	読み込んででも状況が変わらない場合は、最新の認証ファイルを取得し、再度読み込んでください。

RMW 初回起動時に、認証画面で「Termfile not exist」が表示されることがあります（このメッセージが表示されている間は Renesas Motor Workbench は利用できません）。その場合は、パッケージ内の「Installer」フォルダにある、認証ファイル (*.atf) を「Select」ボタンから指定して読み込んでください。

RMW の認証には、1 年間の利用期限を設けており、期限切れが近づくと認証画面にメッセージ「License expires soon」が、期限が過ぎると「Expiration data out」が表示されます。期限を過ぎた場合は、新しい認証ファイルの読み込みが必要になります。

2.4.1.1 新しい認証ファイルの入手

利用期限を更新するには、新しい認証ファイル (*.atf) をルネサス WEB の下記ページから入手して読み込んでください。

- Renesas Motor Workbench

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



図 2.11 新しい認証ファイルの入手先

2.4.1.2 認証ファイルの読み込み

各 Authentication メッセージの「Select」ボタンをクリックすると認証ファイルの読み込みができます。認証ファイルの読み込みが正常に終了すると、RMW が利用可能になります。

操作手順は以下の通りです。

1. Help メニューから「Load Authentication File」を選択すると、「Set Authentication file」のメッセージが表示されます。
2. 表示されたメッセージ内の「Select」ボタンをクリックすると、認証ファイルの置かれた場所を聞いて来るので、対象のファイルを指定します。
3. ファイルオープン後、Authentication 画面が消えれば正常終了です。なお、すでに有効な認証ファイルを読み込み済みの場合は「Already Read File」と表示されます。

2.4.2 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。

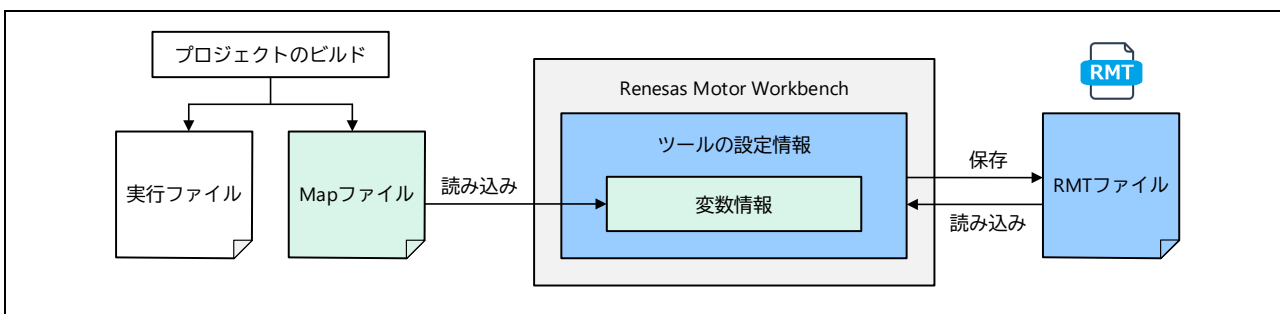


図 2.12 RMT ファイル

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。

2.4.2.1 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。

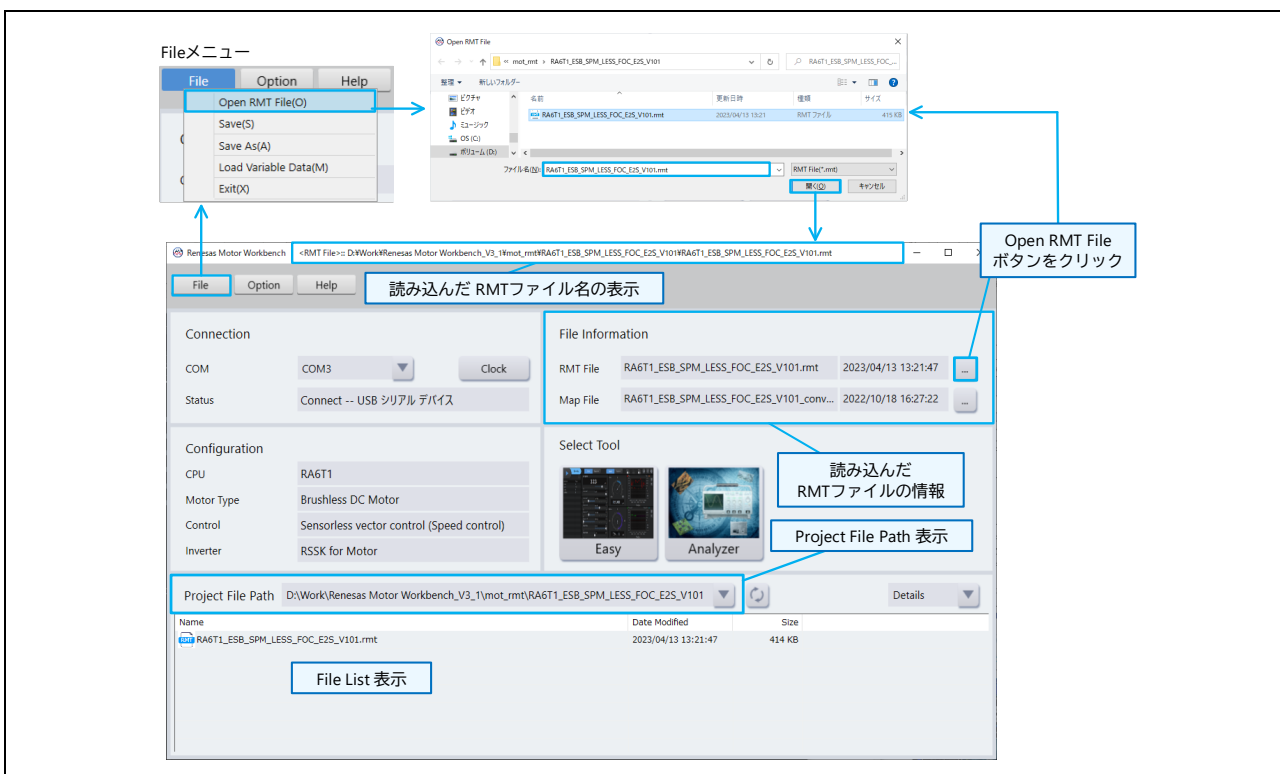


図 2.13 File メニューからの RMT ファイル読み込み

2.4.2.2 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。

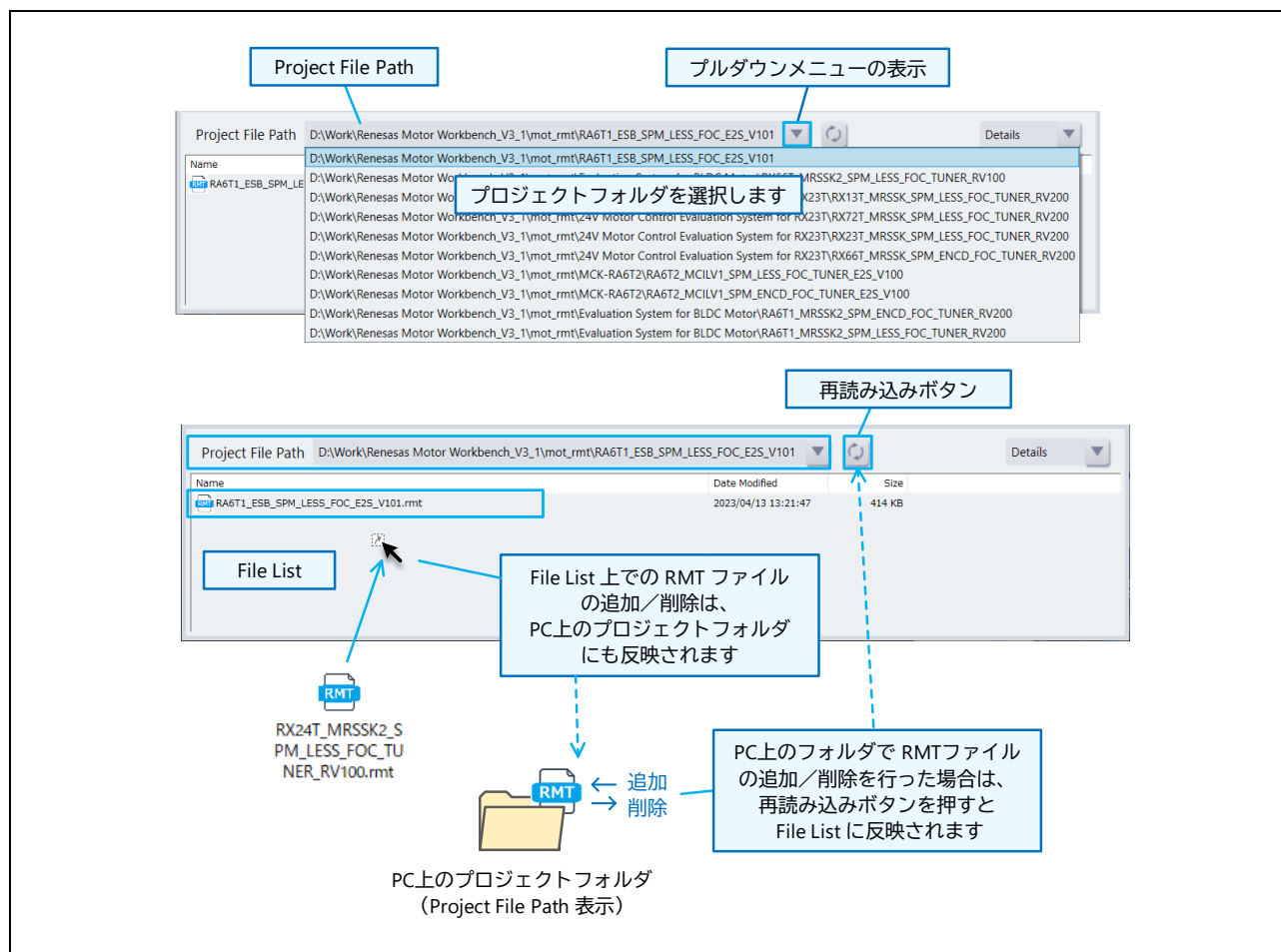


図 2.14 File List からの RMT ファイル読み込み

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)。

2.4.2.3 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。

2.4.3 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。

2.4.3.1 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。

2.4.3.2 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。

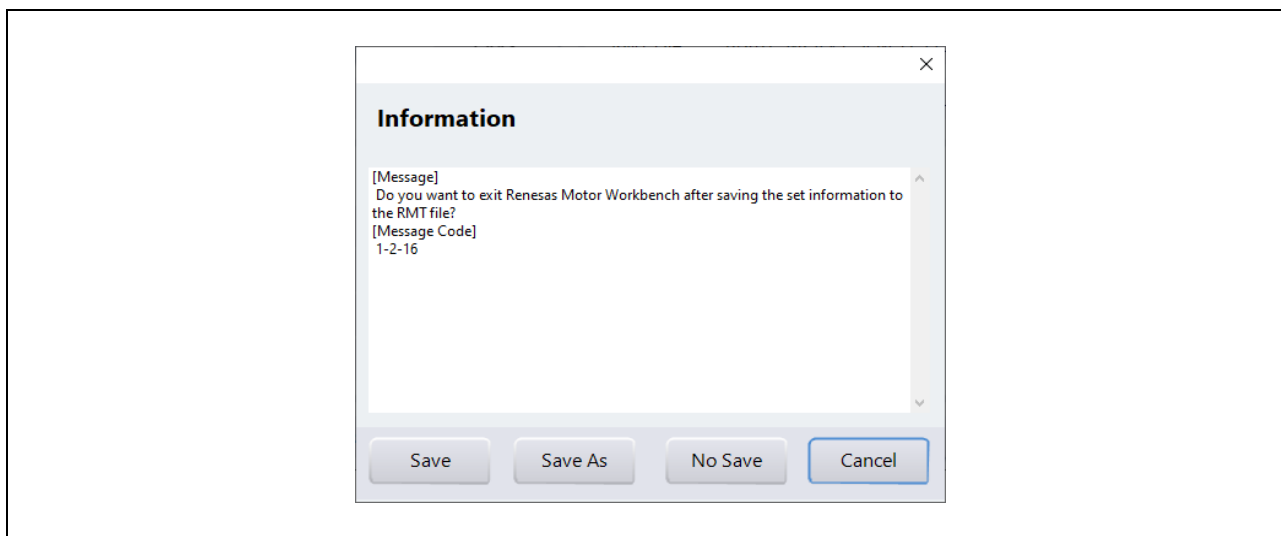


図 2.15 Renesas Motor Workbench の終了画面

表 2.12 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。

2.4.4 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。

2.4.4.1 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。

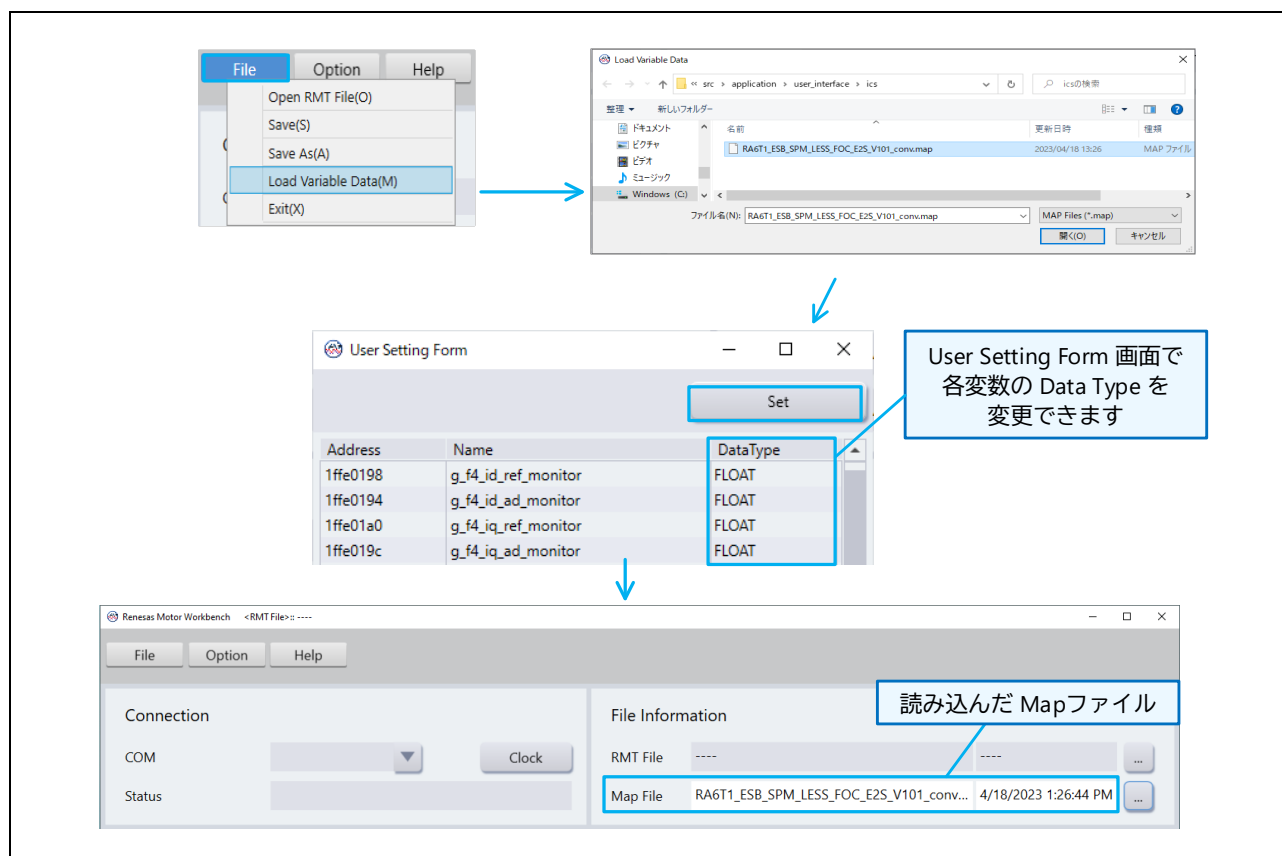


図 2.16 変数リスト（Map ファイル）の読み込み

2.4.4.2 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。

2.4.5 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 2.13 Option Dialog メニューの一覧

タブ	機能	説明
Main Window	Load Variable Data	<ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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。 <ul style="list-style-type: none"> 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)

2.4.5.1 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。

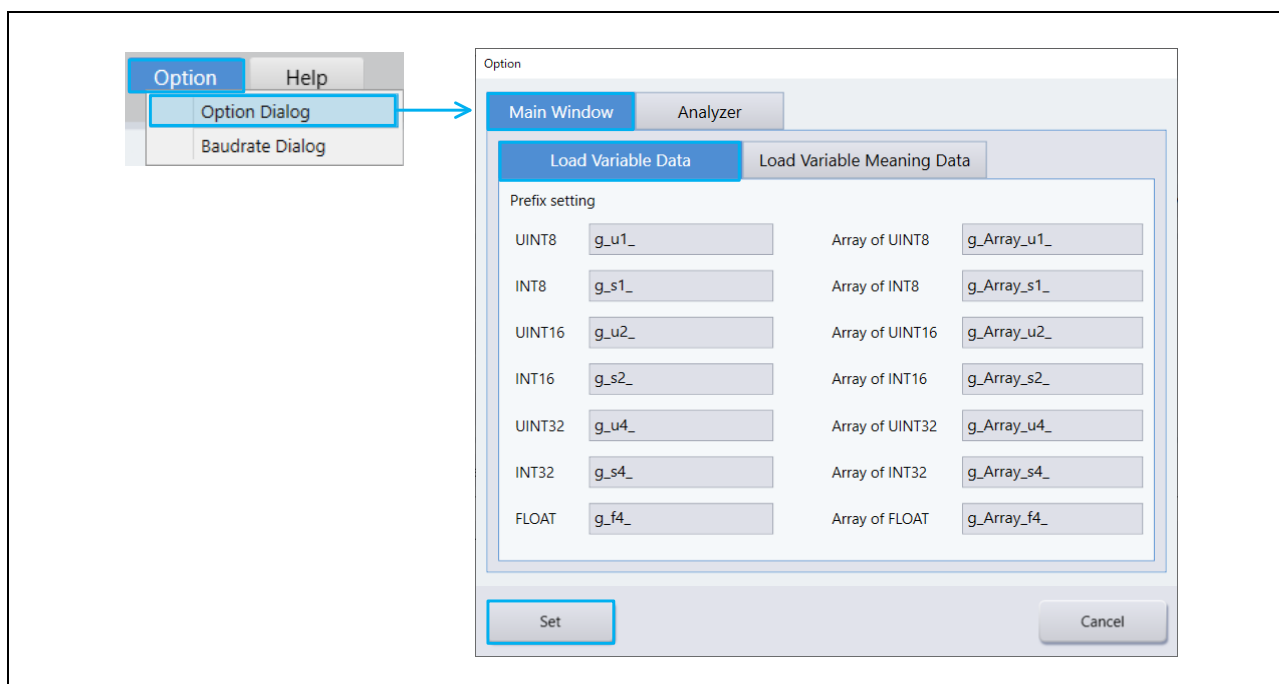


図 2.17 Main Window タブの Load Variable Data 設定

2.4.5.2 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、2.4.3RMT 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。

	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 2.18 変数の用途情報ファイル (csv 形式) の作成例

2.4.5.3 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）。

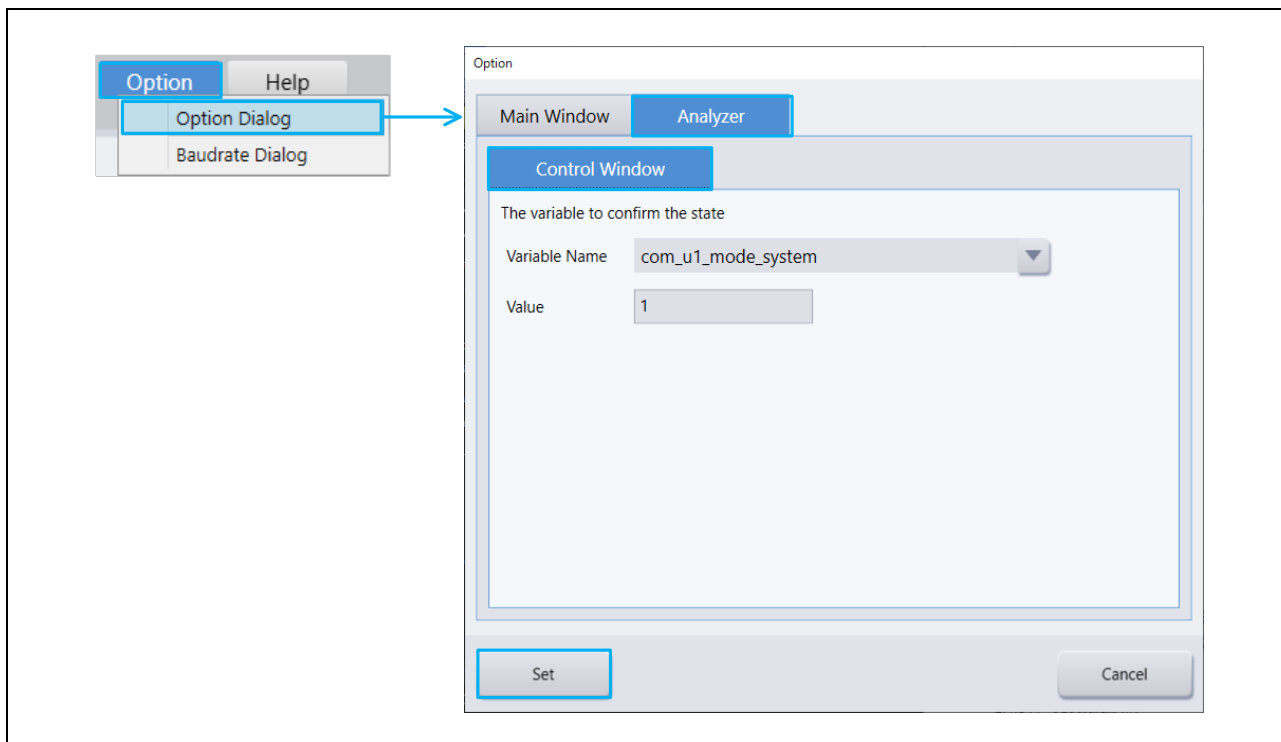


図 2.19 Analyzer タブの Control Window 設定（「モータ駆動中」を判定条件とした例）

2.4.6 USB の接続

2.4.6.1 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 2.14 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	<ul style="list-style-type: none"> 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。

2.4.6.2 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。

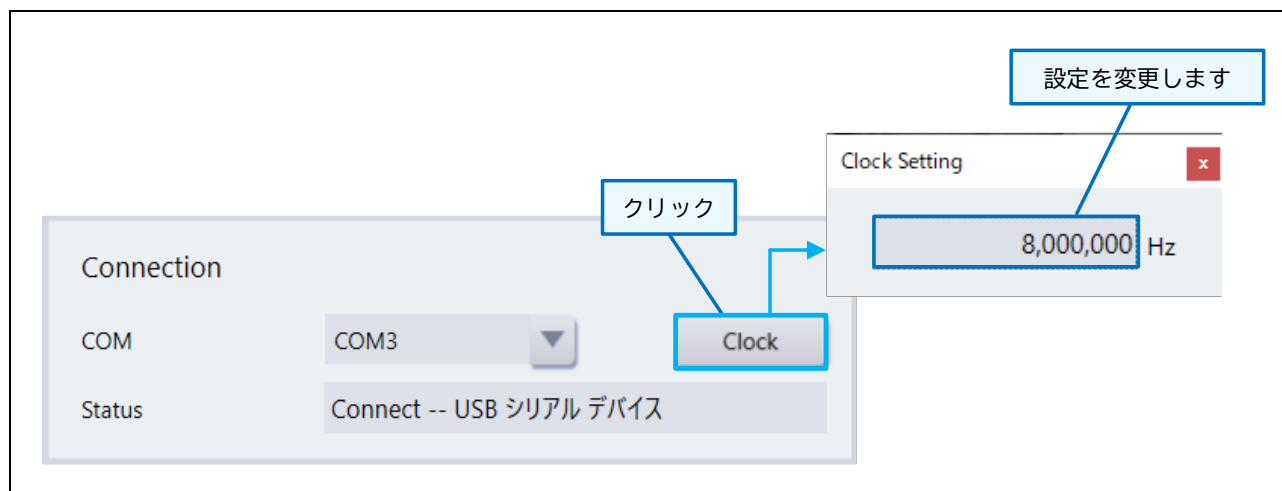


図 2.21 Clock Setting ダイアログ

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 2.15 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）

2.4.6.3 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、15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。

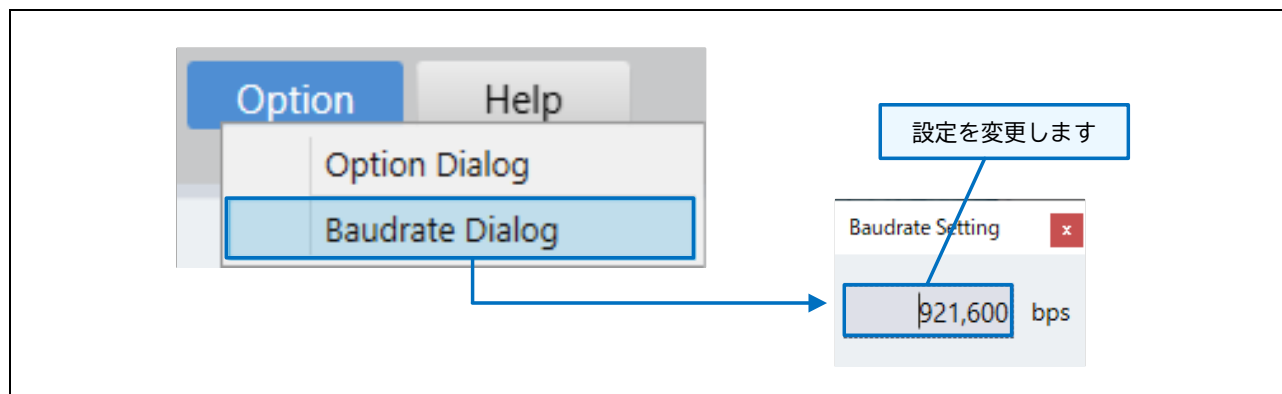


図 2.22 ボーレートの設定

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。

2.4.7 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、Easy) 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、2.4.5 章を参照してください)。

2.4.7.1 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、2.4.5.2 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。

2.4.7.2 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。

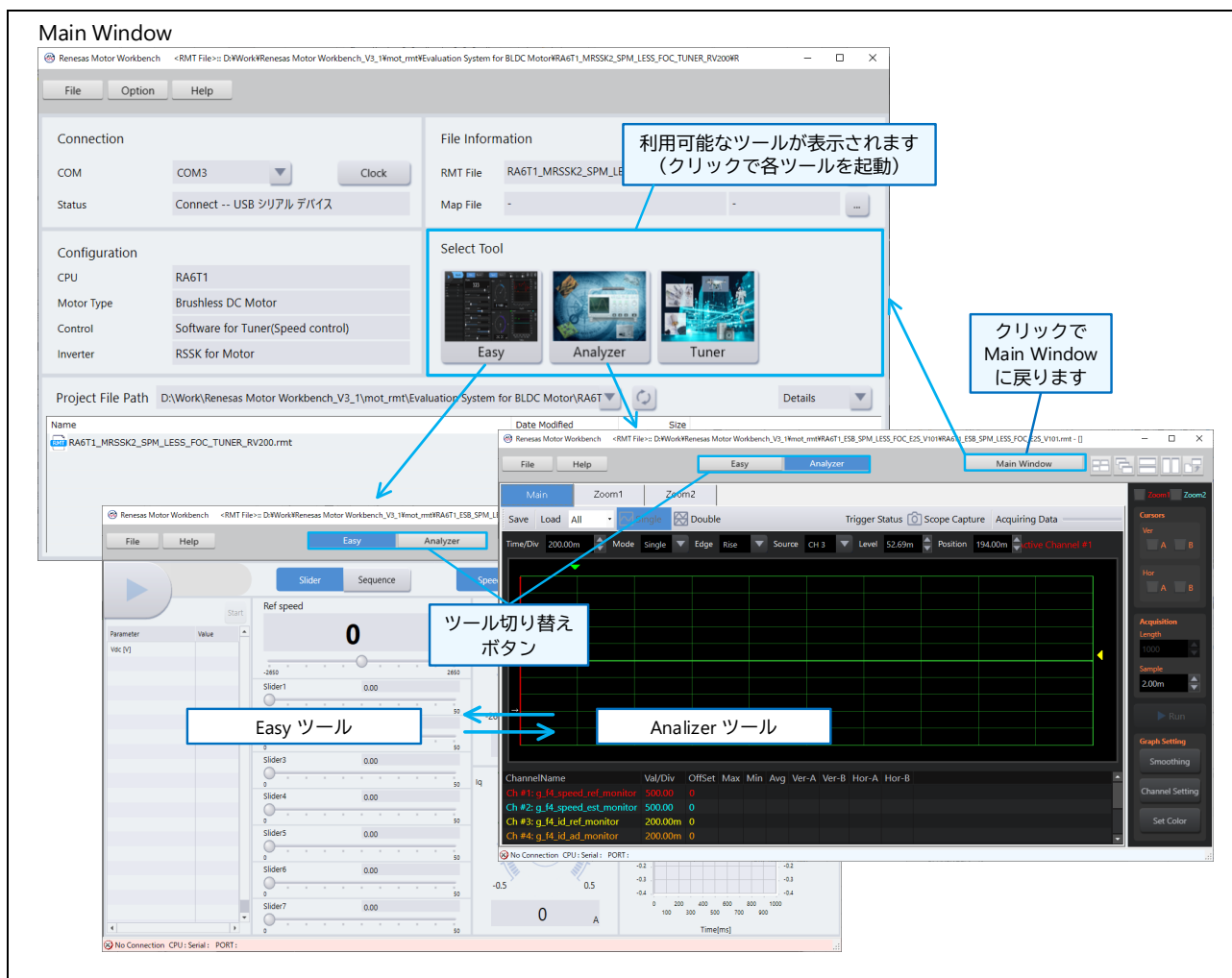


図 2.23 ツール切り替えボタンと Main Window ボタン

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。

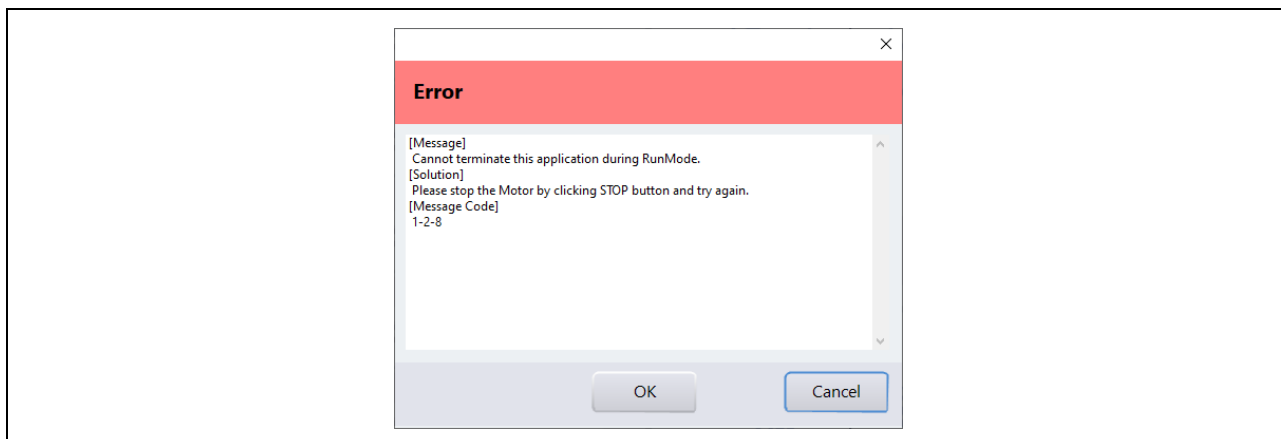


図 2.24 処理実行中のツール切り替えエラーメッセージ

2.4.7.3 ウィンドウ表示切り替えボタン

各ツール（Easy、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）。

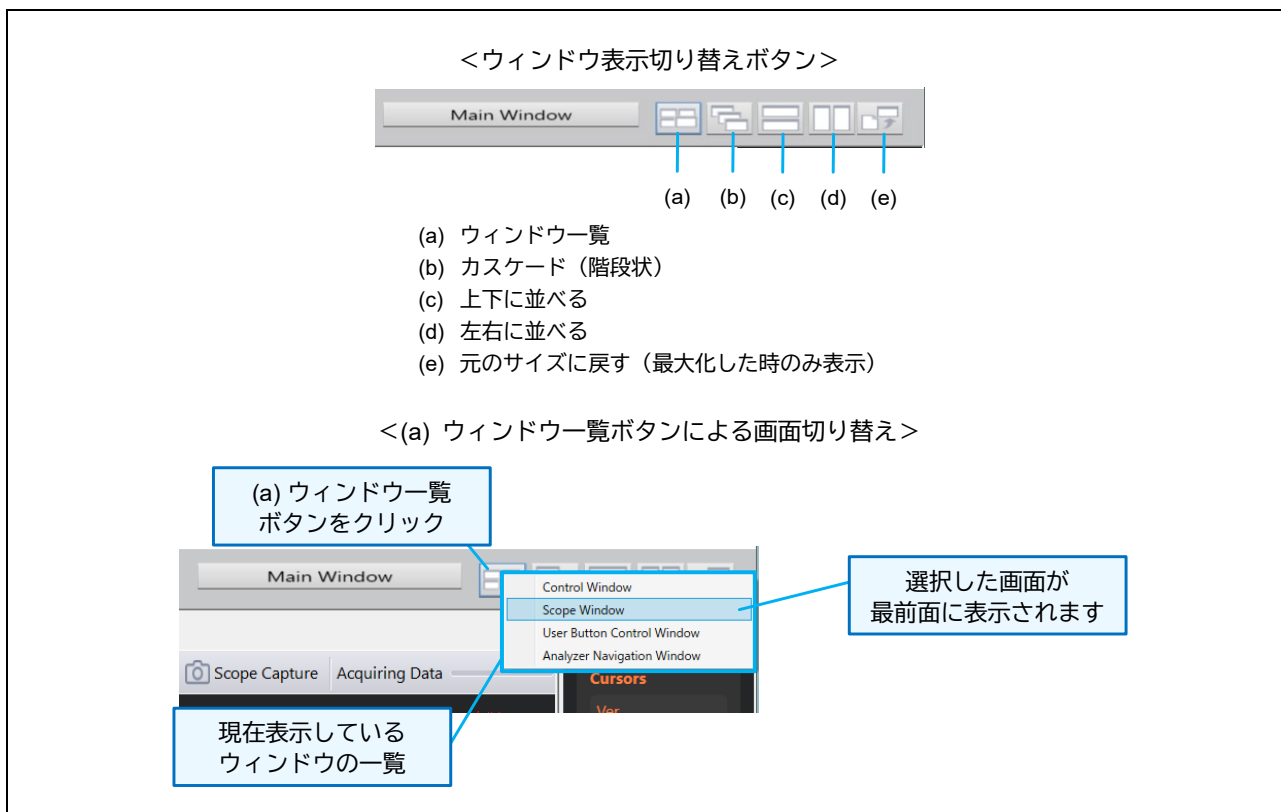


図 2.25 ウィンドウ切り替えボタン

2.4.7.4 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。

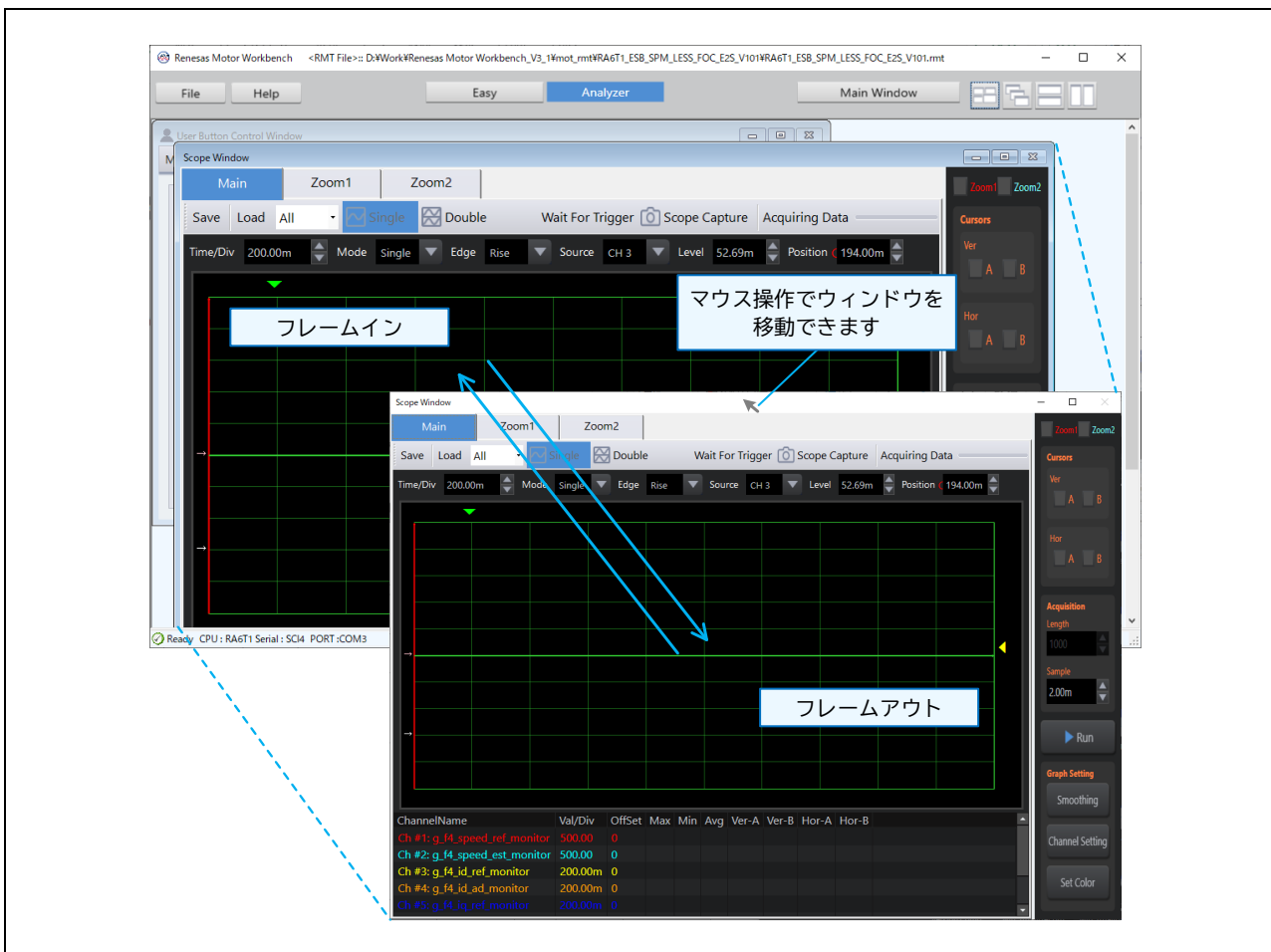


図 2.26 ウィンドウのフレームアウトとフレームイン

2.4.8 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。

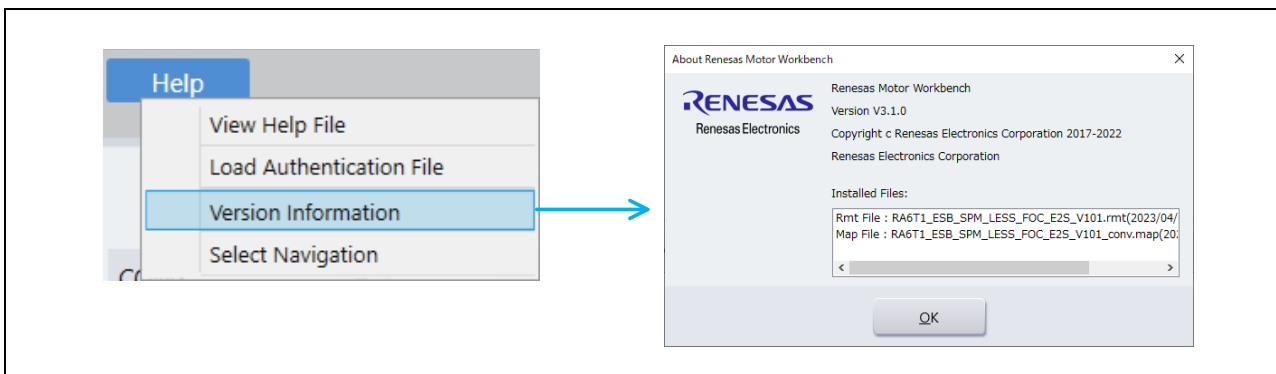


図 2.27 バージョン情報の表示

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 features 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。

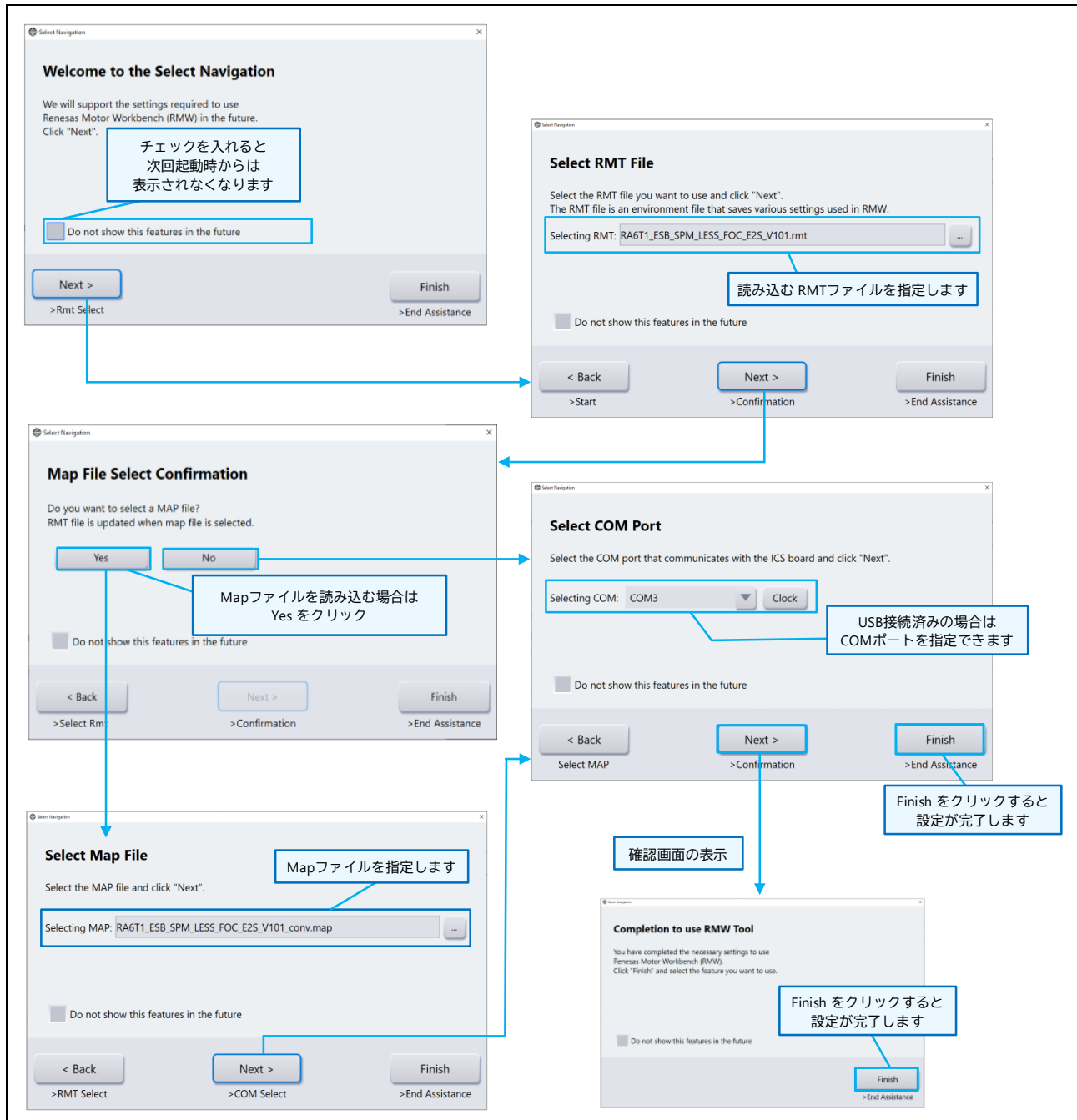


図 2.28 Select Navigation 機能

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、「2.4.7 ウィンドウの基本操作」を参照してください）。

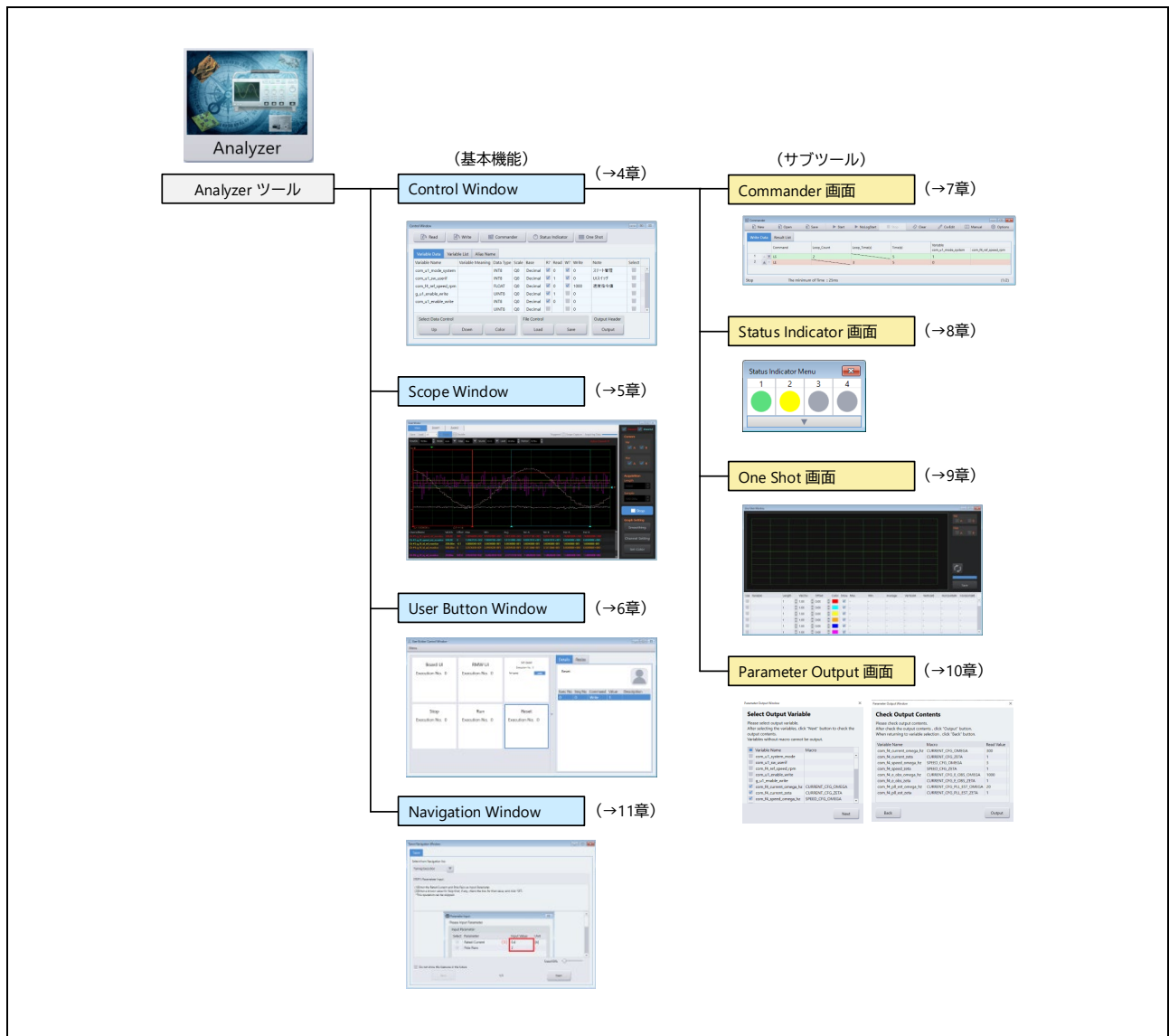


図 3.1 Analyzer ツールの構成

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）。

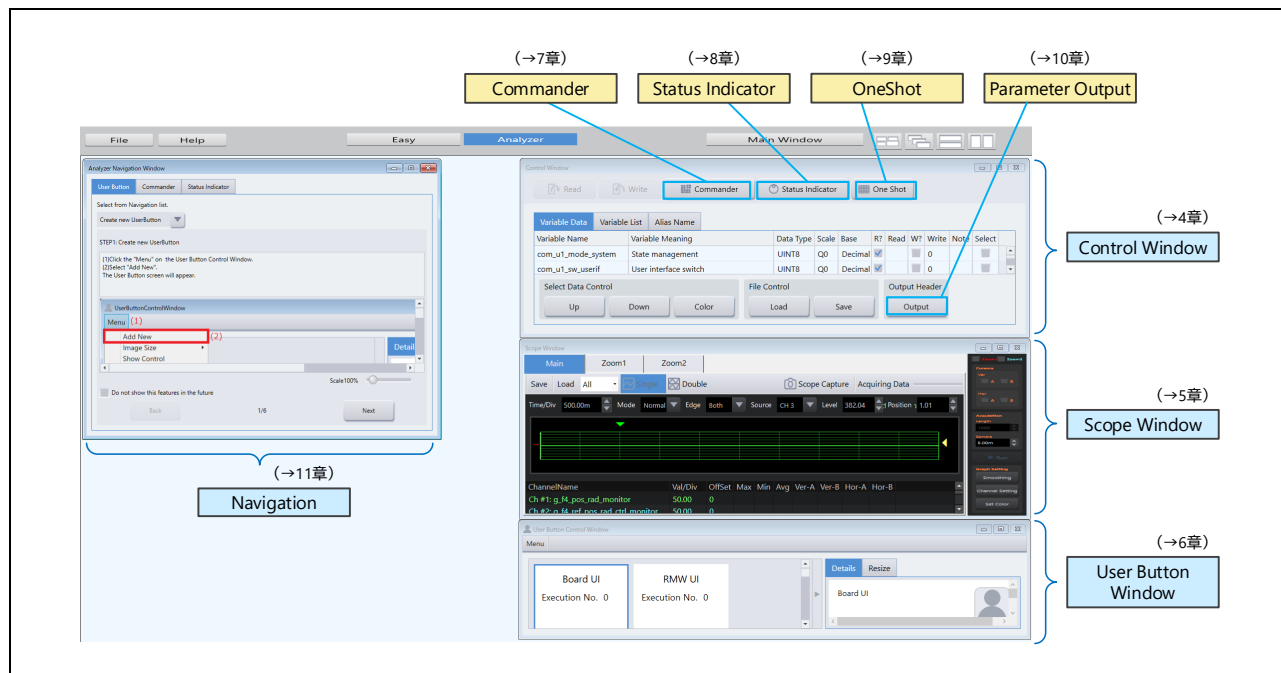


図 3.2 Analyzer ツールの外観

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。

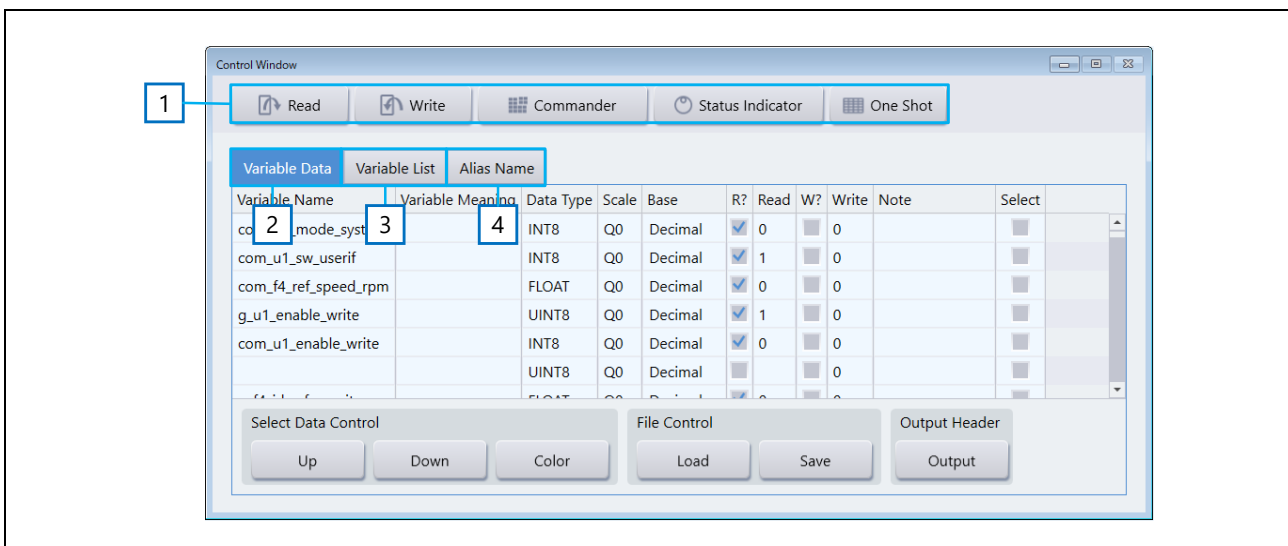


図 4.1 Control Window の画面構成

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。

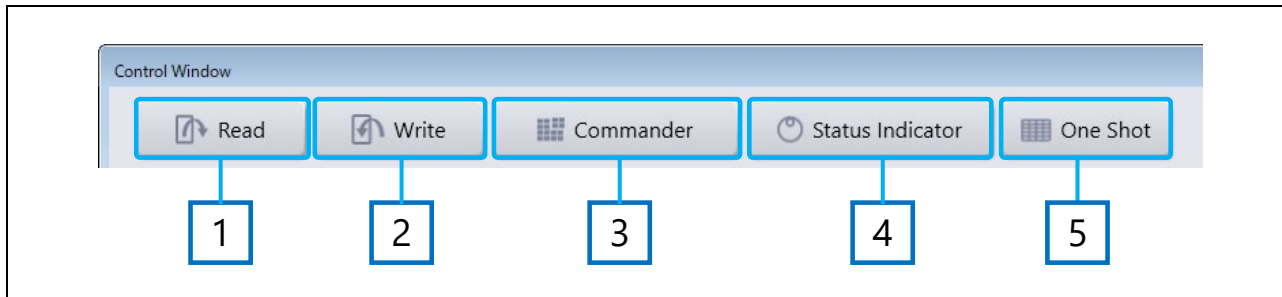


図 4.2 操作ボタン

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. [Analyzer] Commander」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. [Analyzer] Status Indicator」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「0. [Analyzer] One Shot」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。

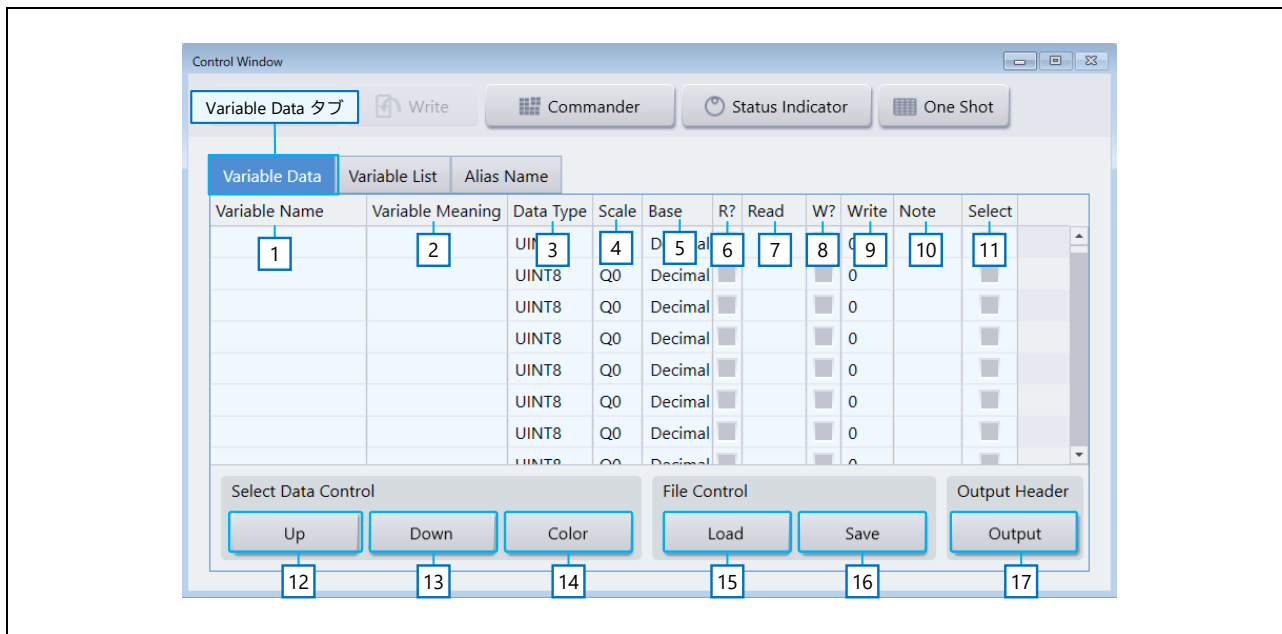


図 4.3 Variable Date タブ画面

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。

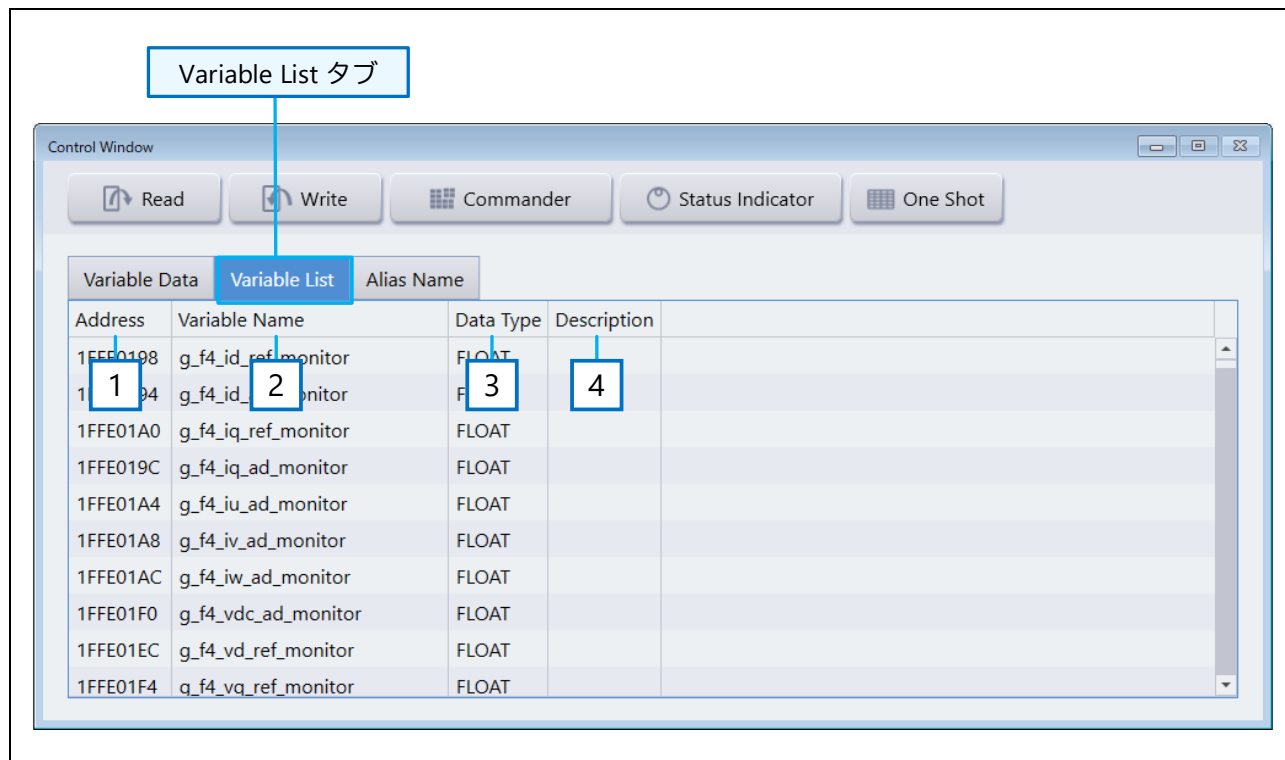


図 4.4 Variable List タブ画面

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。

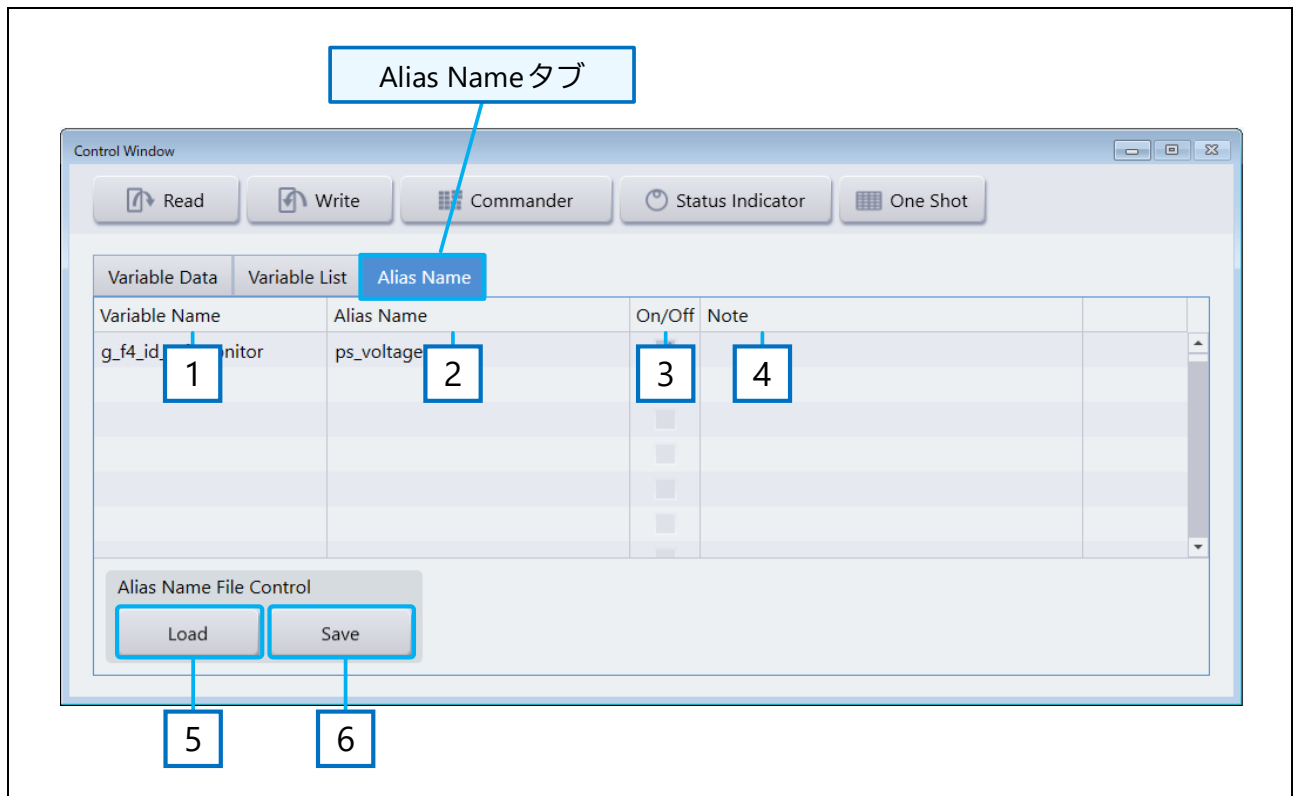


図 4.5 Alias Name タブ画面

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。

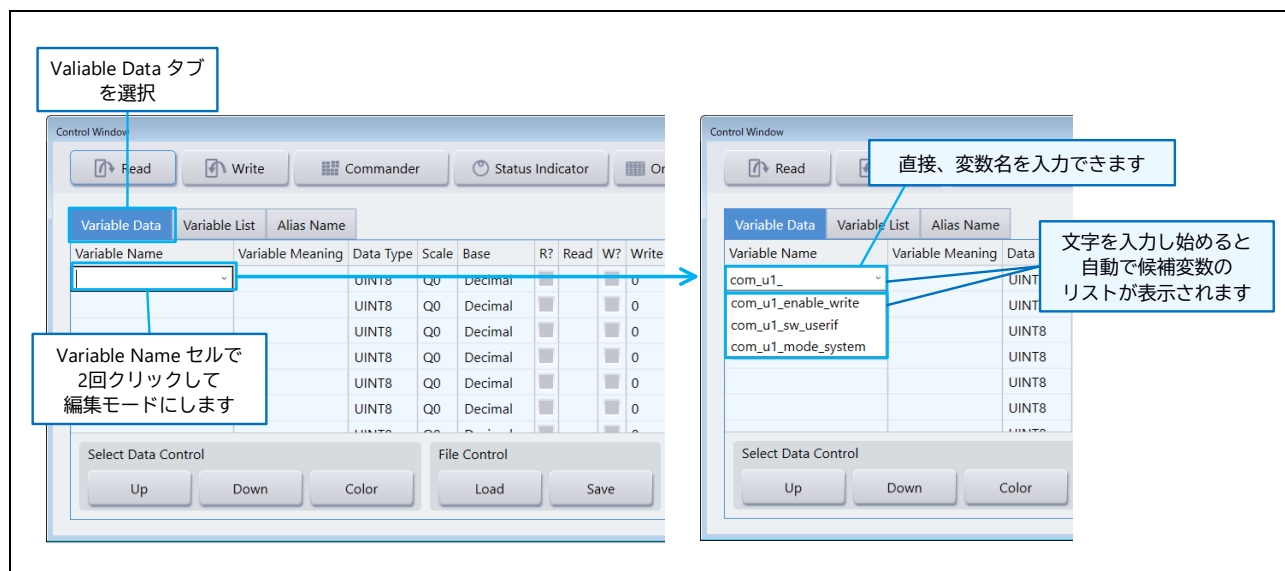


図 4.6 セルへの直接入力

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。

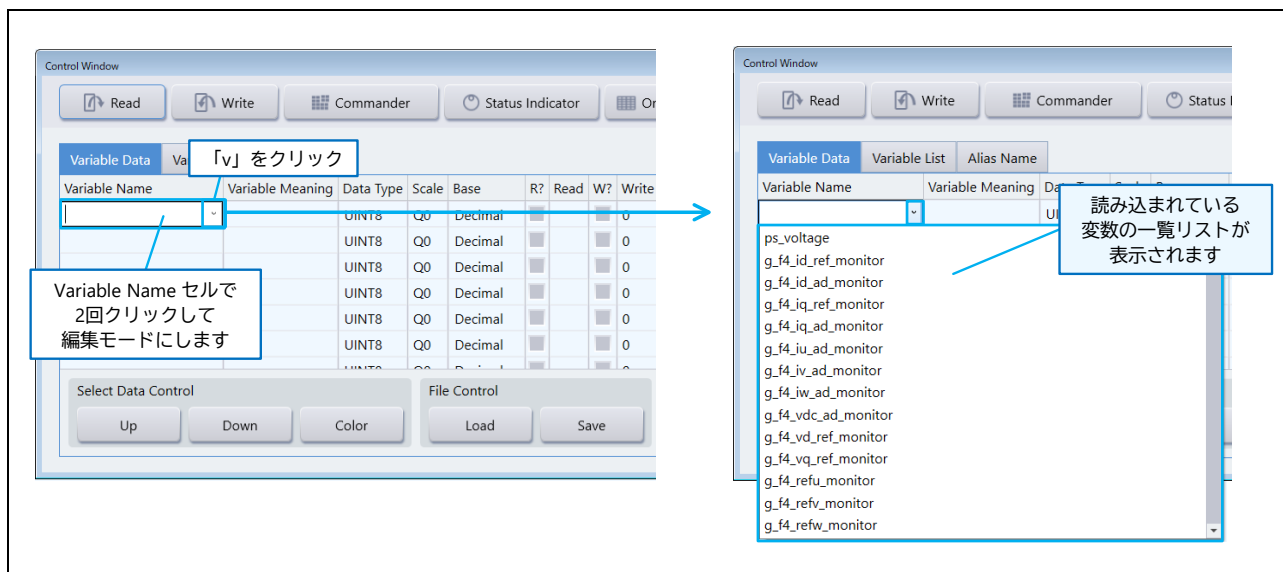


図 4.7 一覧リストからの変数選択

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。

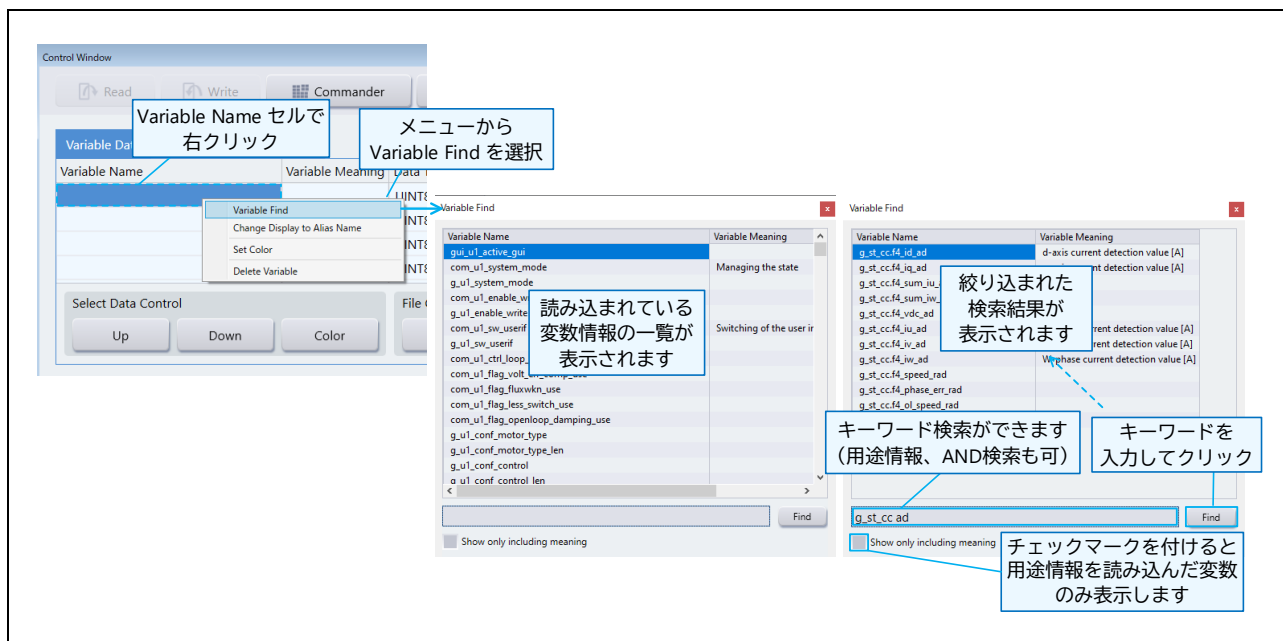


図 4.8 Variable Find 機能を使った変数選択

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。

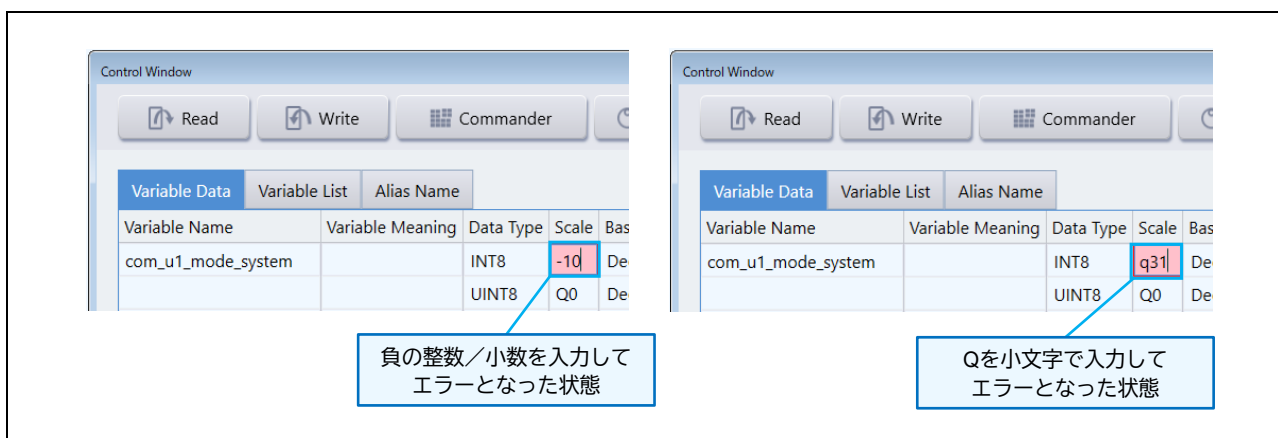


図 4.9 Scale 入力時のエラー表示

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）。

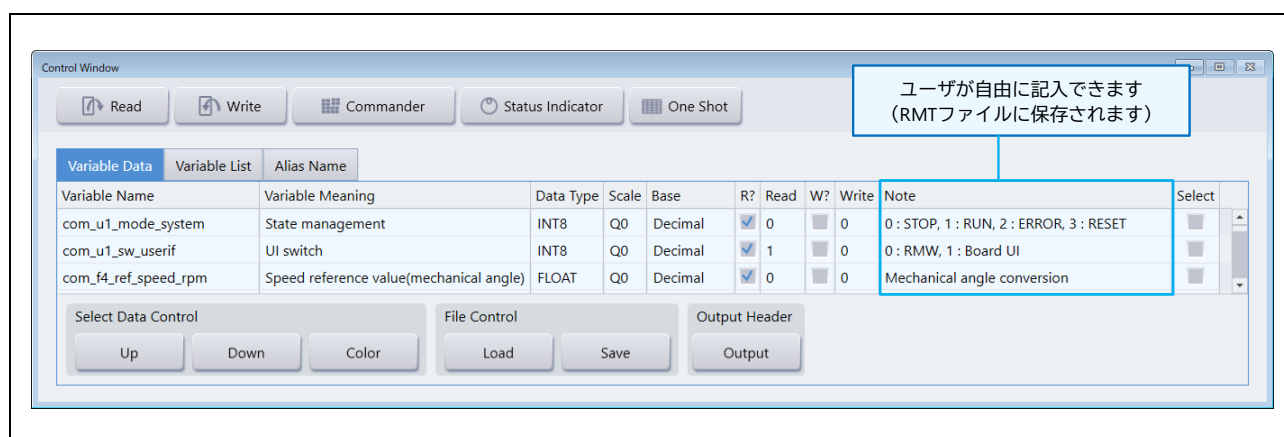


図 4.10 Variable Data タブの Note 欄

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）。

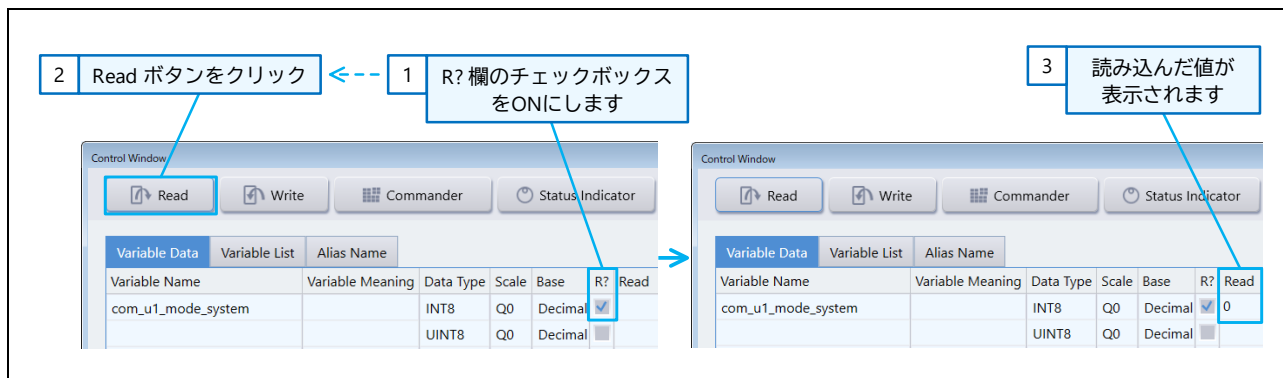


図 4.11 変数値の読み込み

「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。

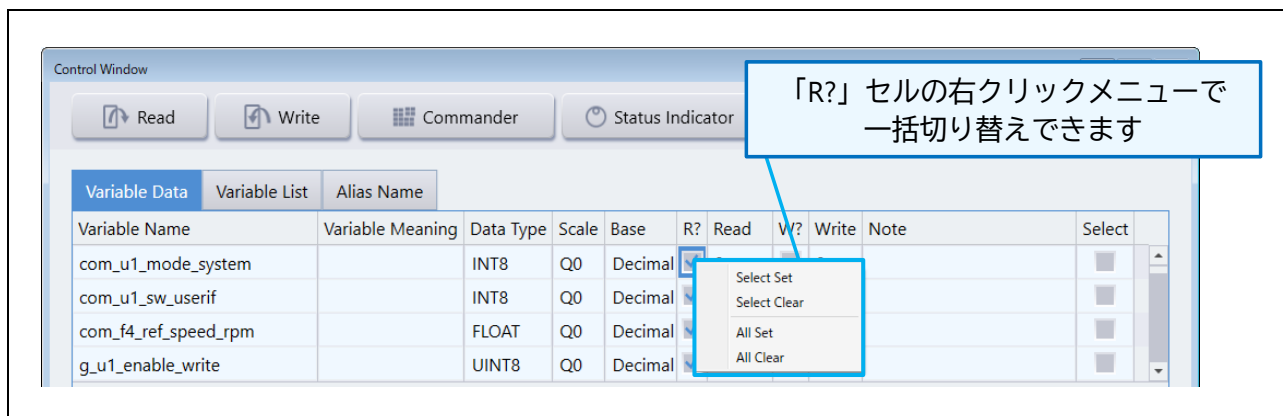


図 4.12 「R?」チェックボックスの一括切り替え

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）。

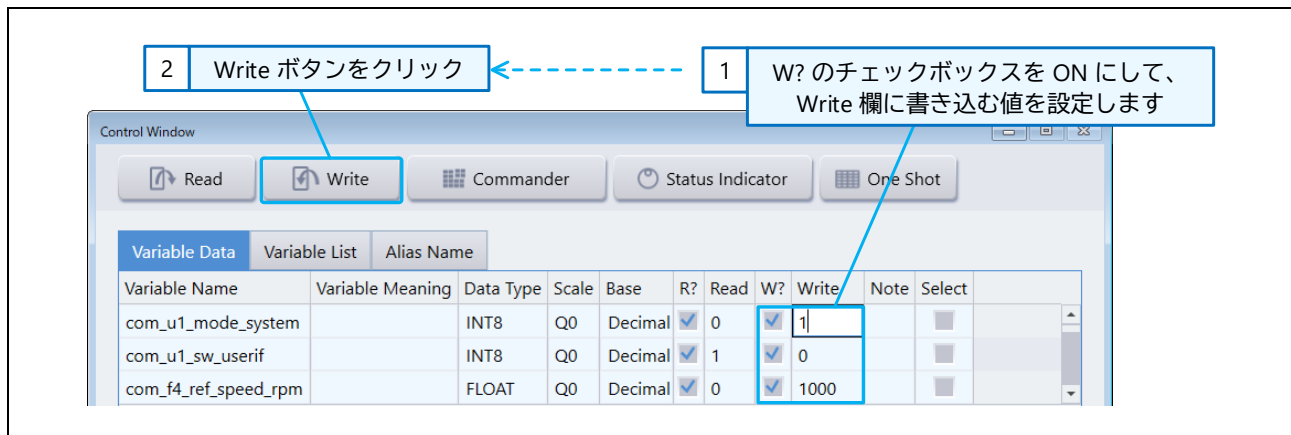


図 4.13 変数値の書き込み

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。

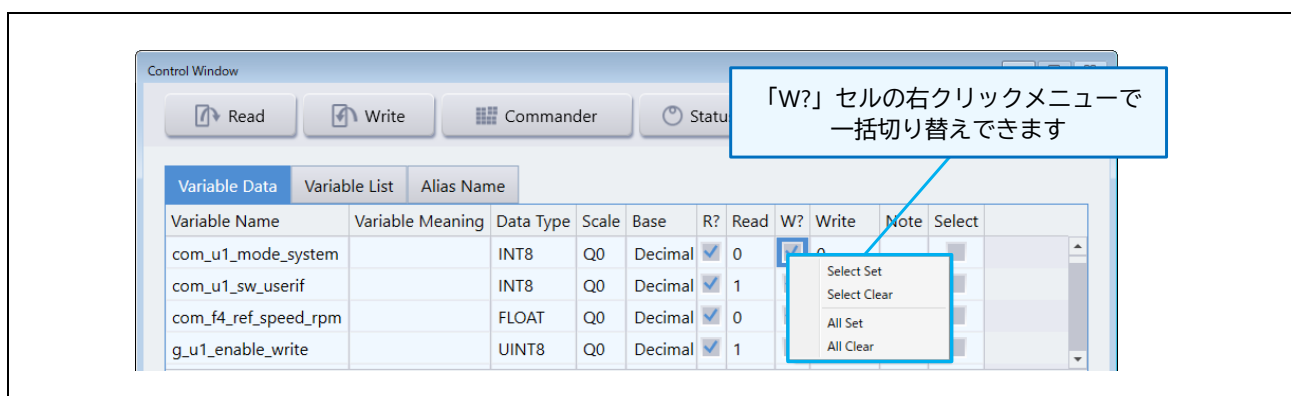


図 4.14 「W?」チェックボックスの一括切り替え

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。

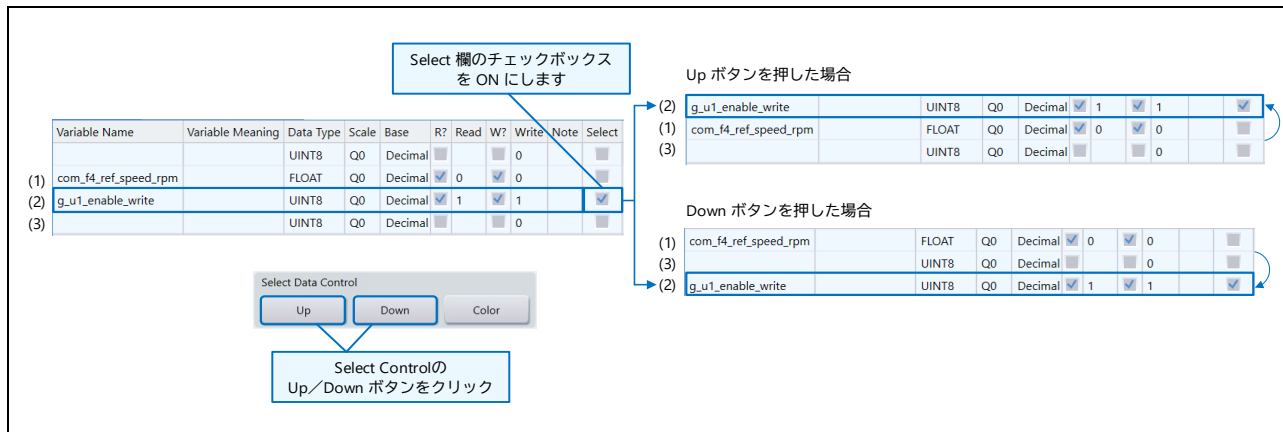


図 4.15 変数表示の並び替え

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。

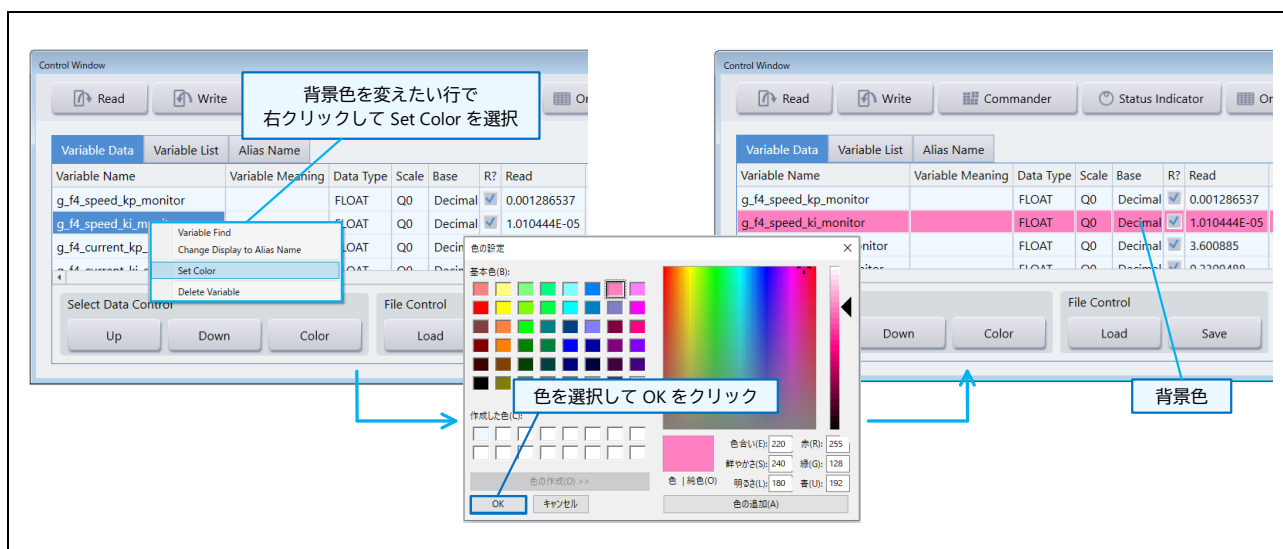


図 4.16 変数表示の背景色変更（右クリックメニュー）

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。

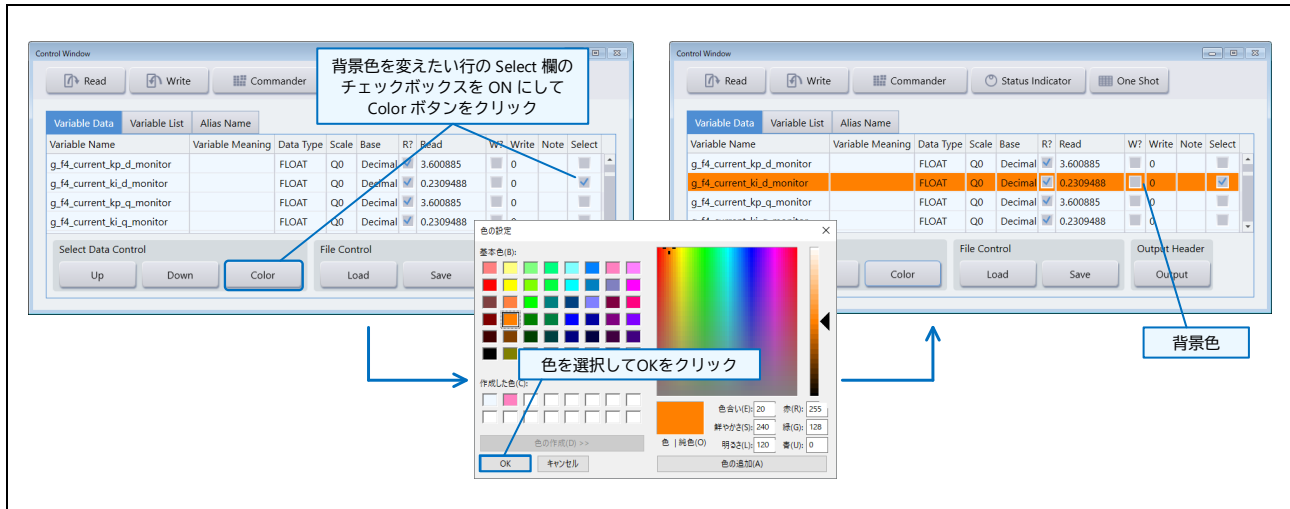


図 4.17 変数表示の背景色変更 (Select 欄使用による行選択)

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。

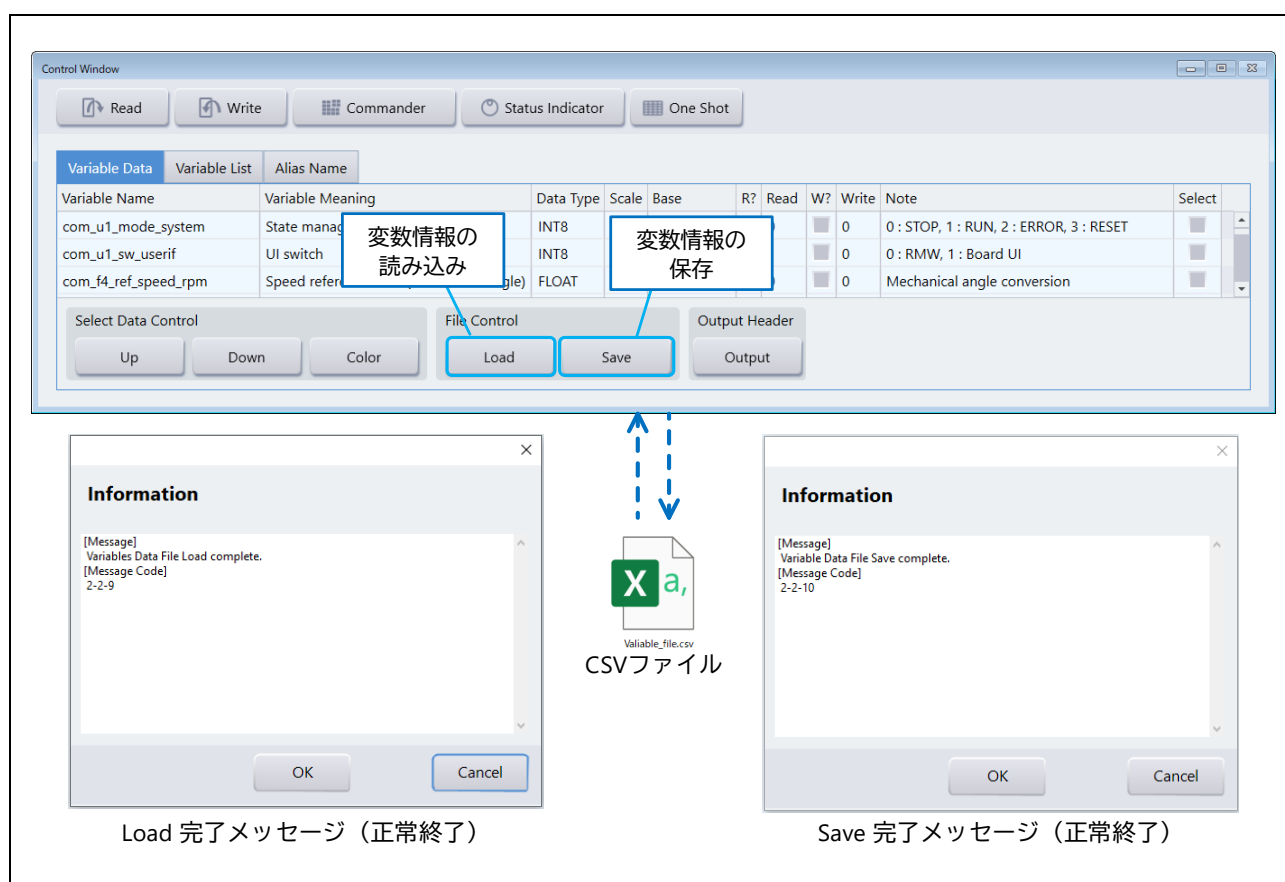


図 4.18 変数情報の保存／読み込み (CSV ファイル形式)

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。

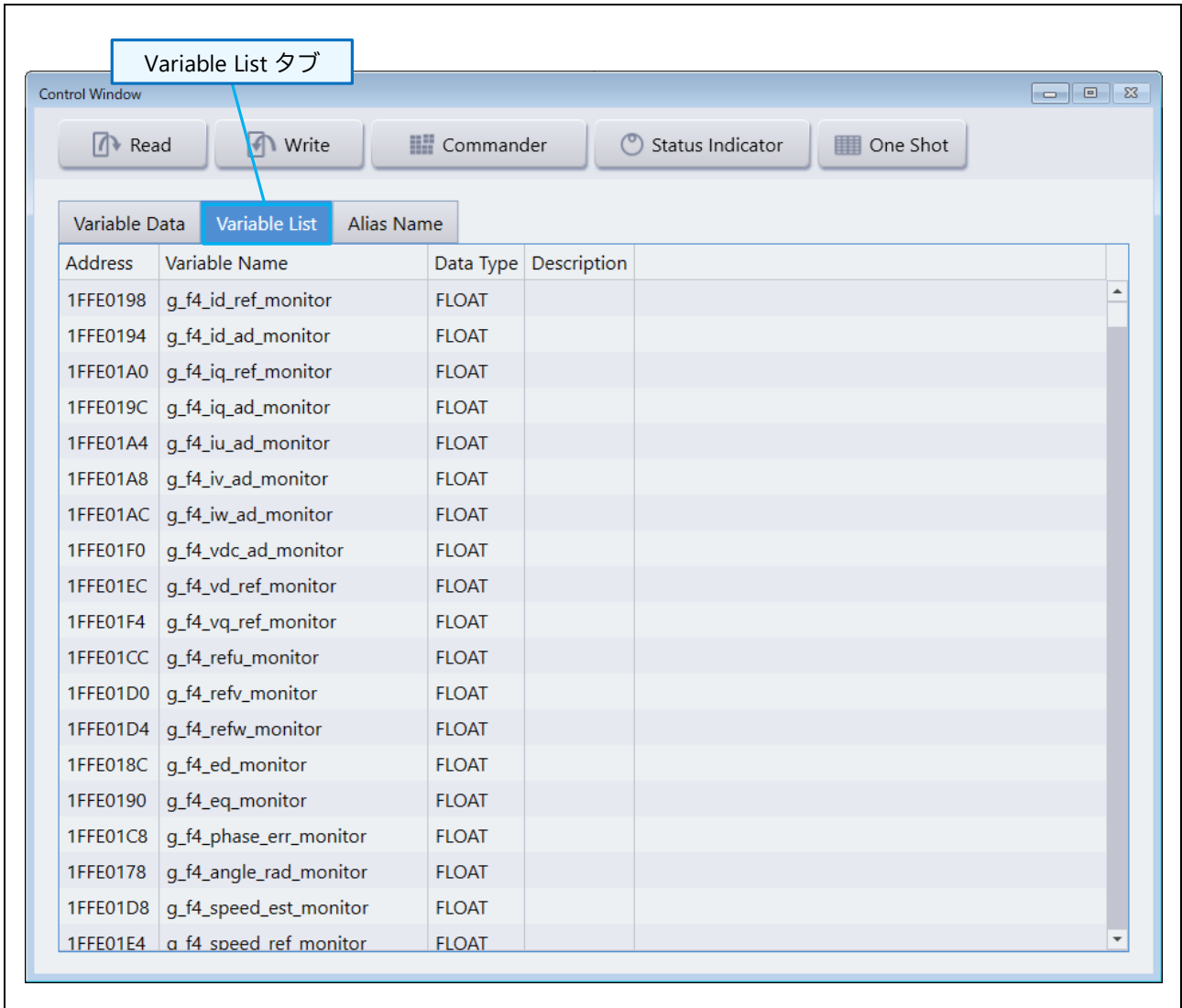


図 4.19 Variable List タブの表示

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 Variable 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。

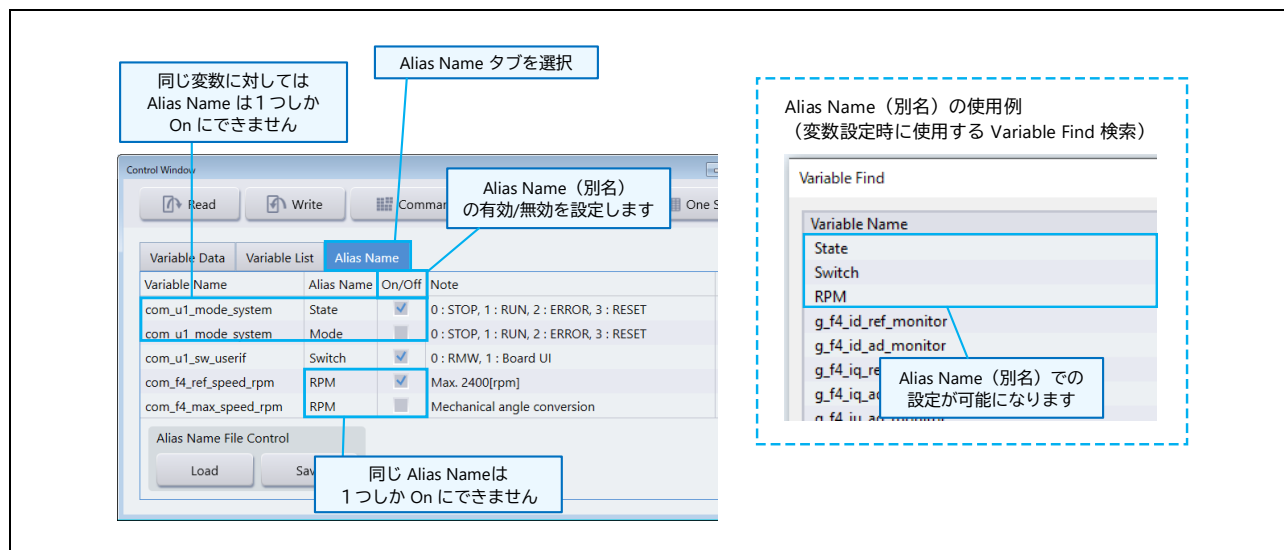


図 4.20 Alias Name (別名) の有効/無効の設定

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。

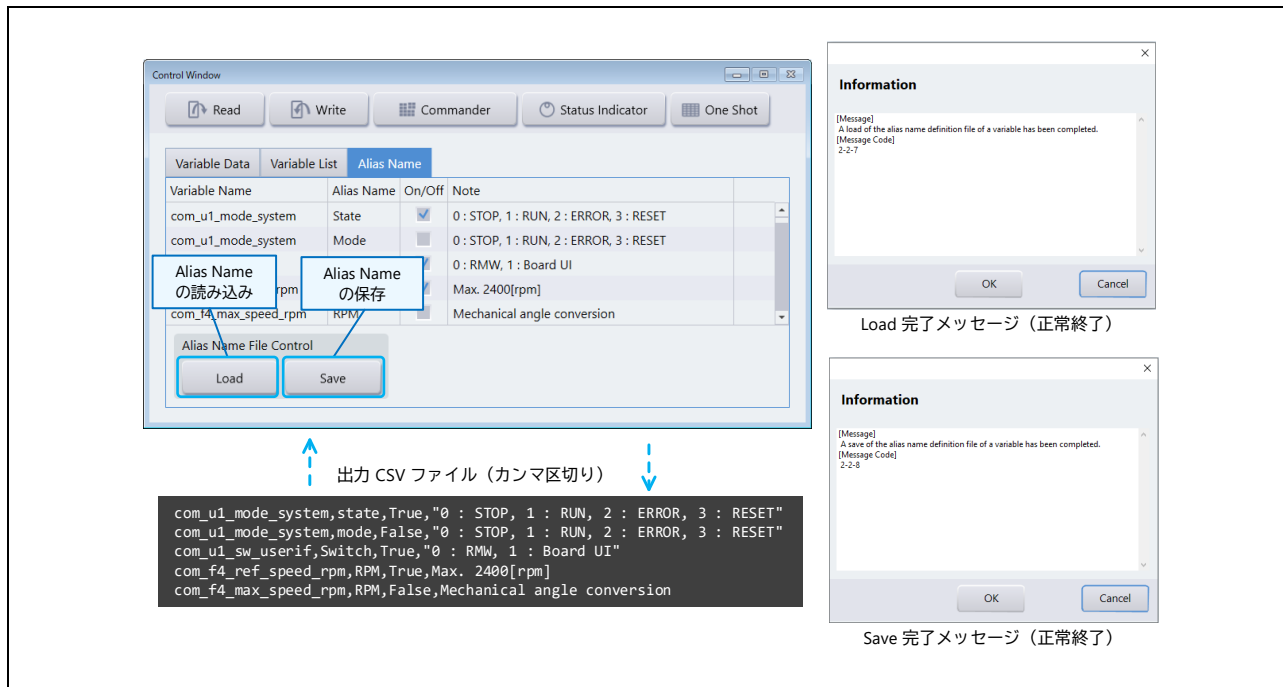


図 4.21 別名リストの保存/読み込み (CSV ファイル形式)

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。

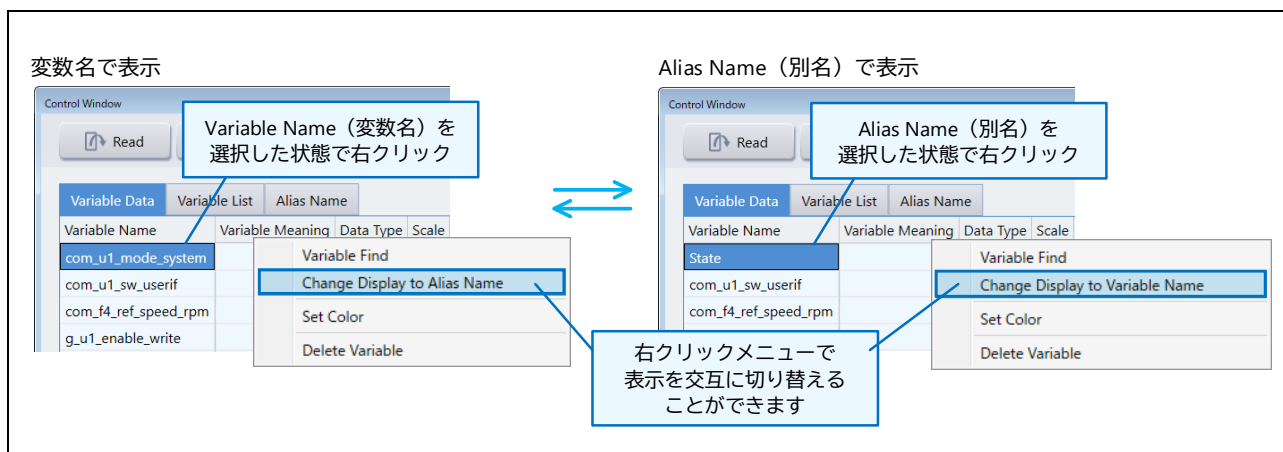


図 4.22 Alias Name (別名) と Variable Name (変数名) の表示切り替え

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）。
- 各チャン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。

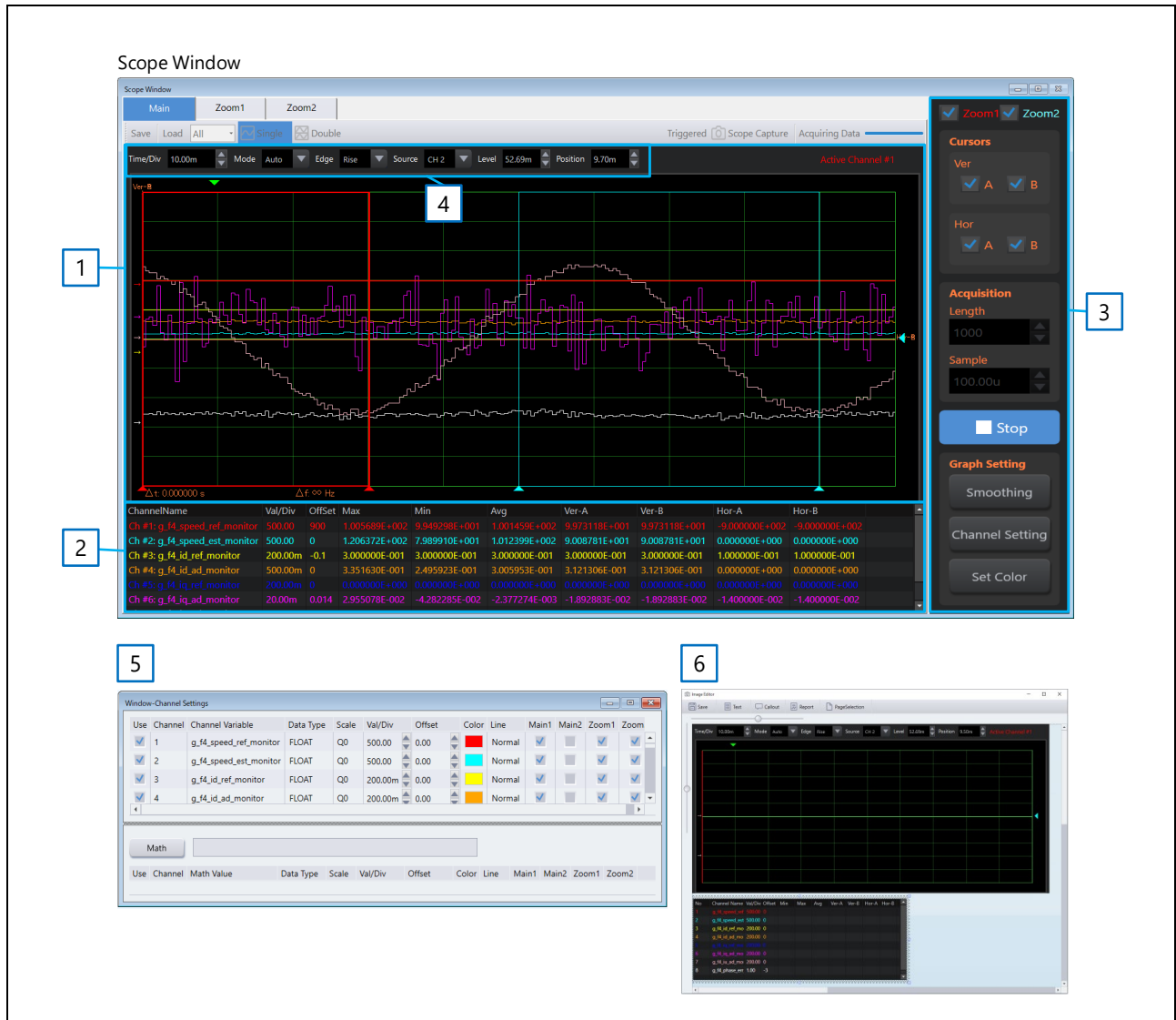


図 5.1 Scope Window の構成

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	チャンネル情報表示部	チャン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、チャン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

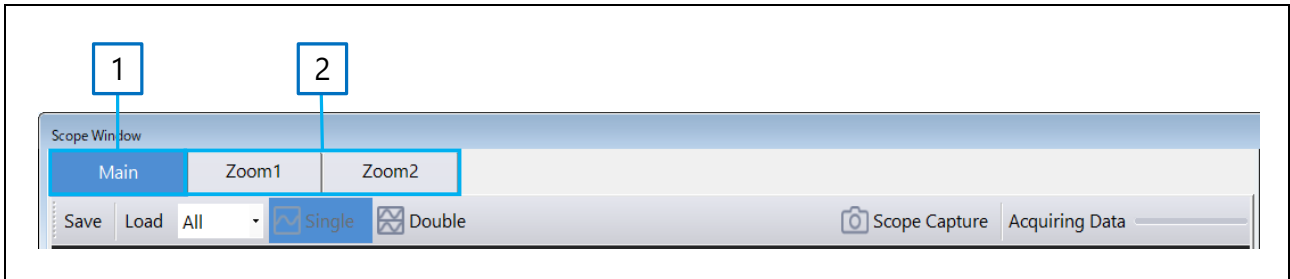


図 5.2 画面表示切り替えタブ

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

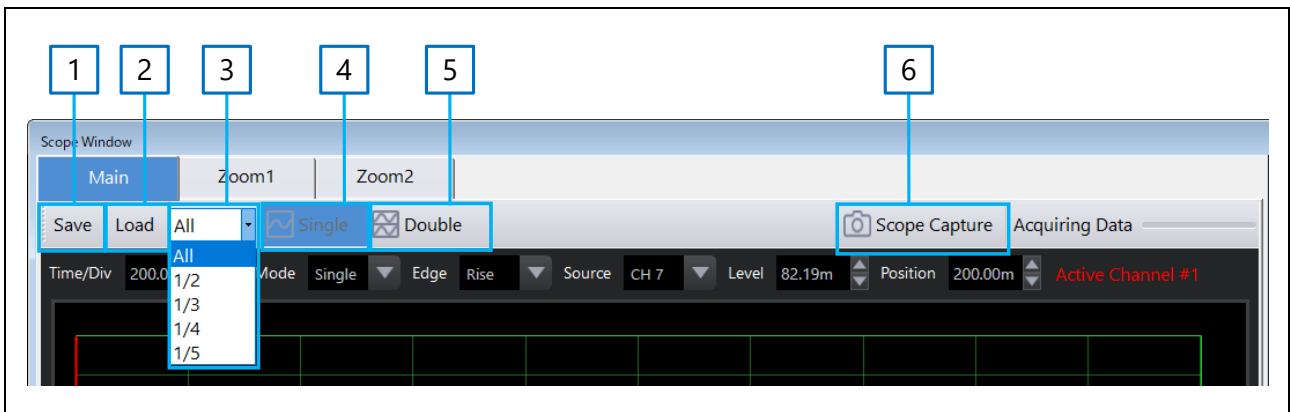


図 5.3 ツールバー

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ンネル情報が 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

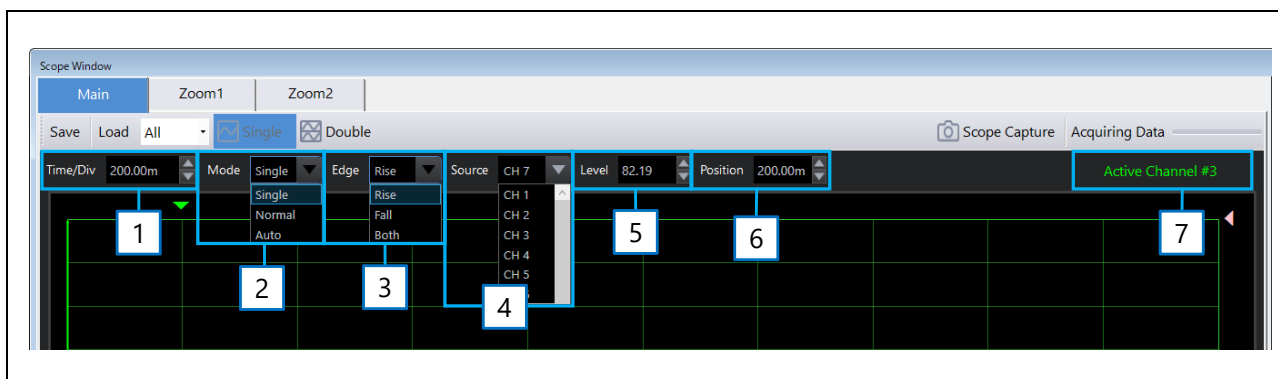


図 5.4 トリガ設定部

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ン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 に設定中のチャン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。

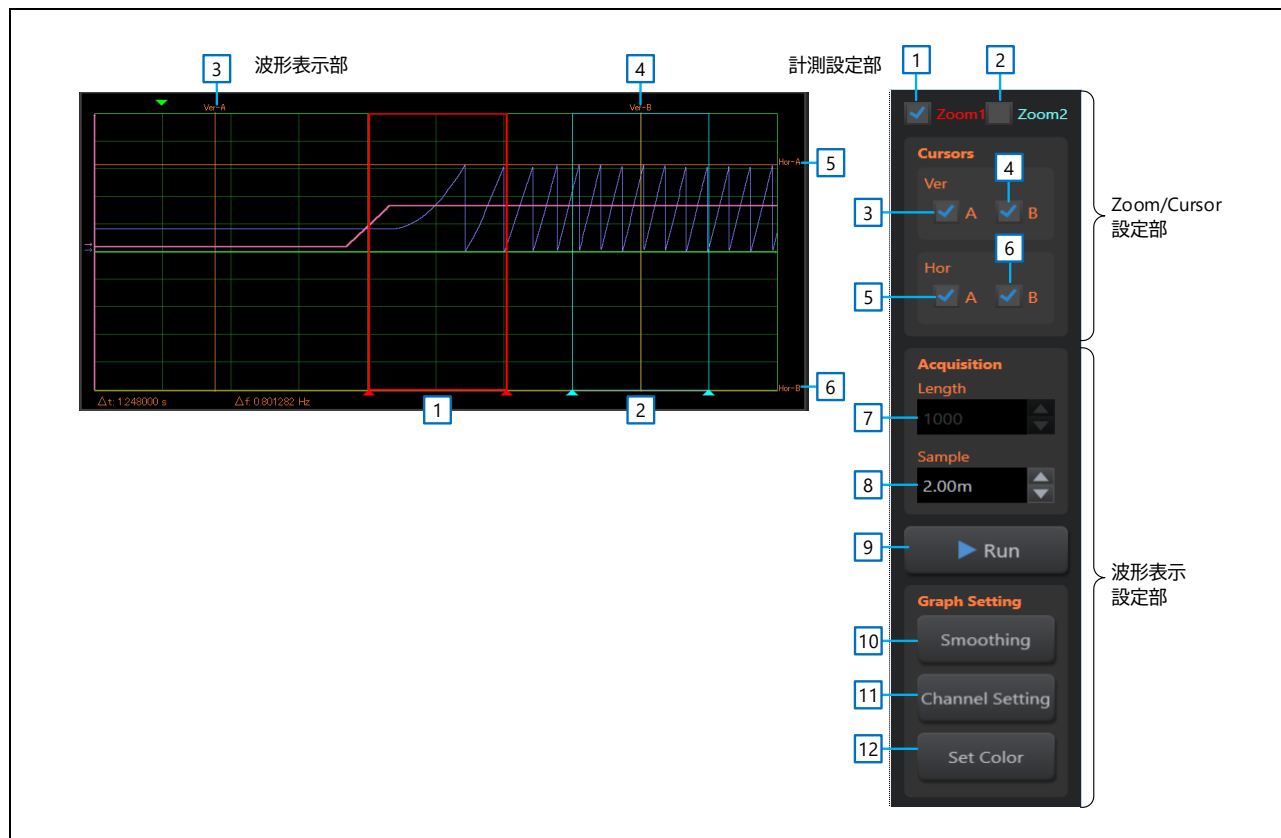


図 5.6 計測設定部

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。

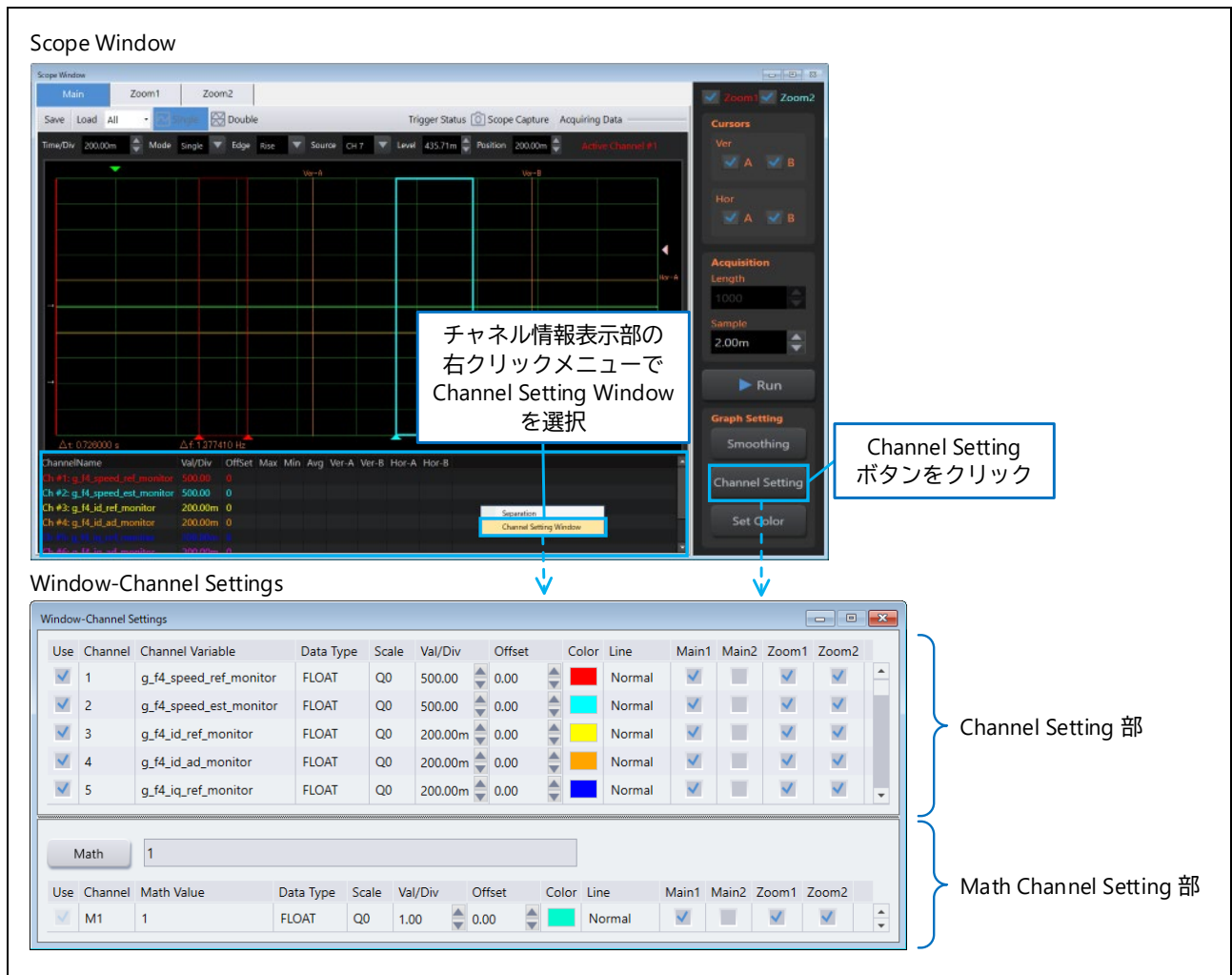


図 5.7 Window-Channel Settings 画面

5.3.4.1 Channel Setting 部

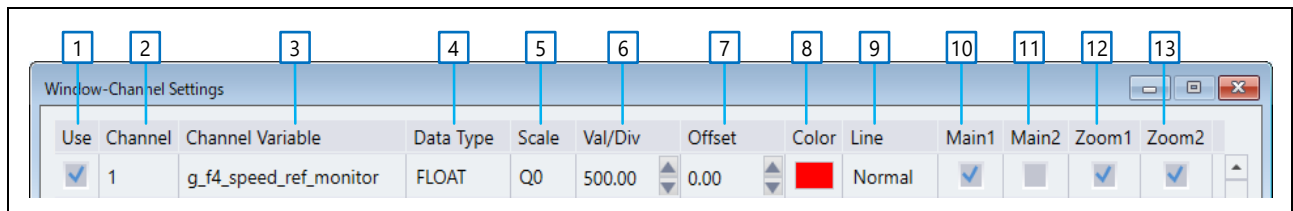


図 5.8 Channel Setting 部

表 5.7 Channel Setting 部の説明

No.	名称	説明
1	Use	波形表示するチャンネルを選択します。ON/OFF 操作ができます。
2	Channel	チャン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

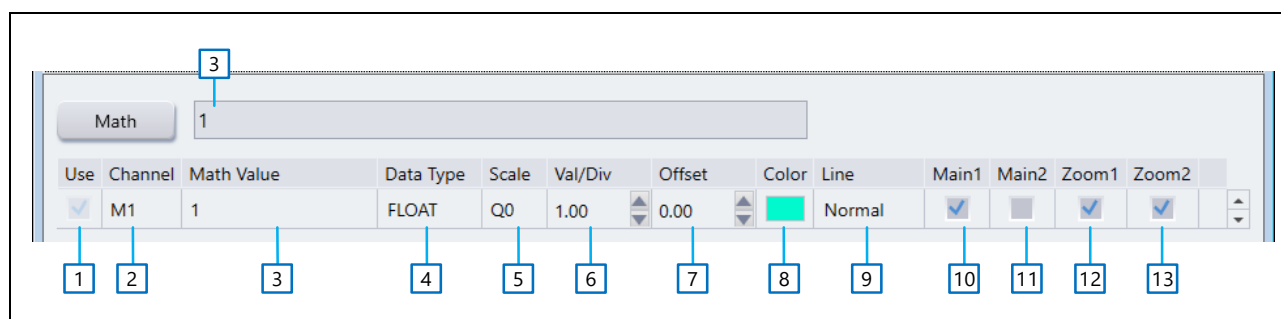


図 5.9 Math Channel Setting 部

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 用のチャンネル番号 (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。

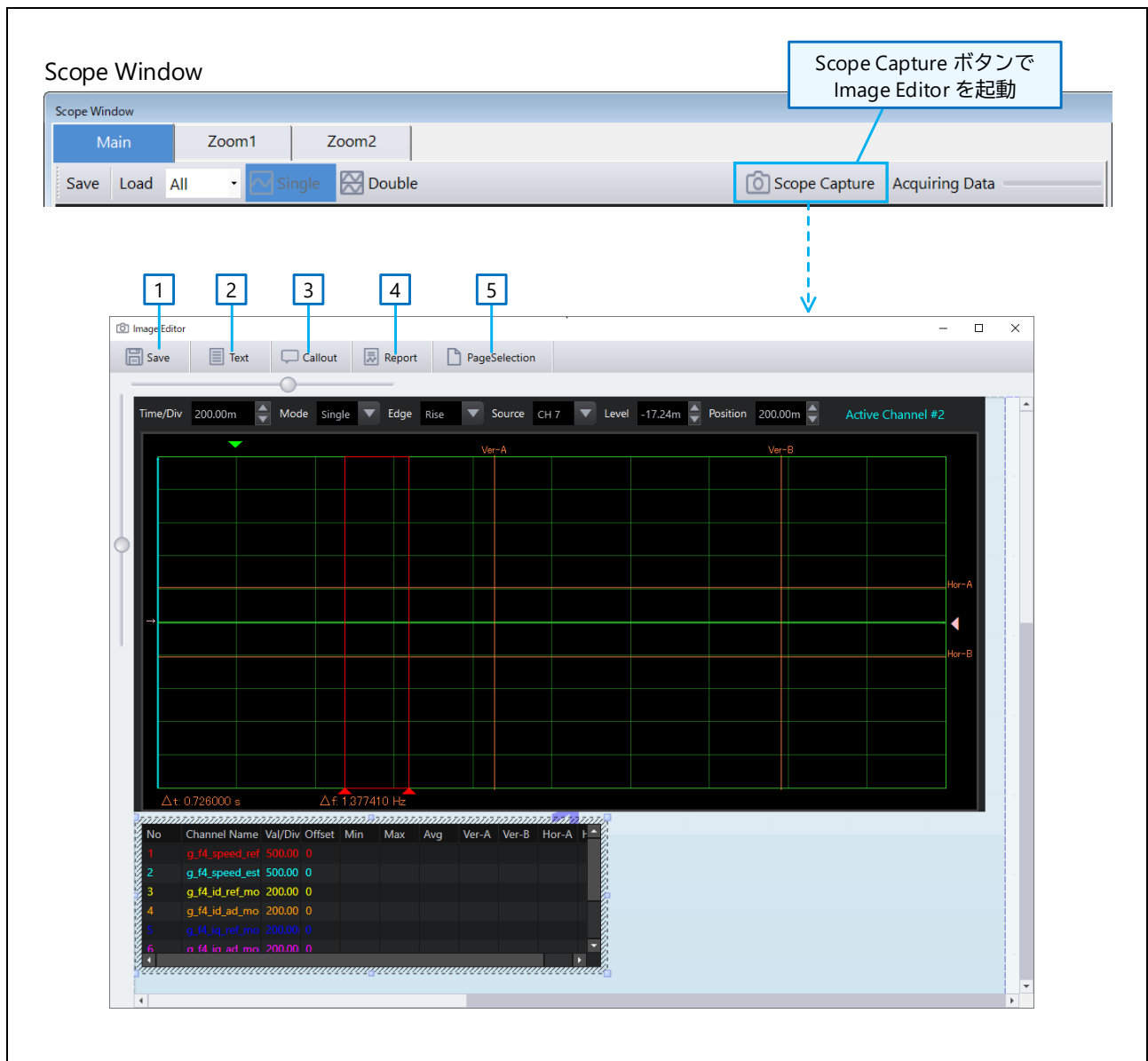


図 5.10 Image Editor 画面

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。

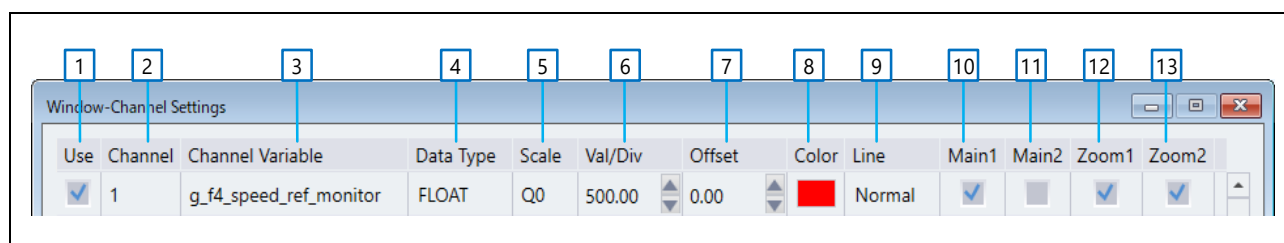


図 5.11 Channel Setting 部

(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、チャン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、チャン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、チャン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、チャン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、チャンネル設定した変数の波形表示の線の太さを選択します。

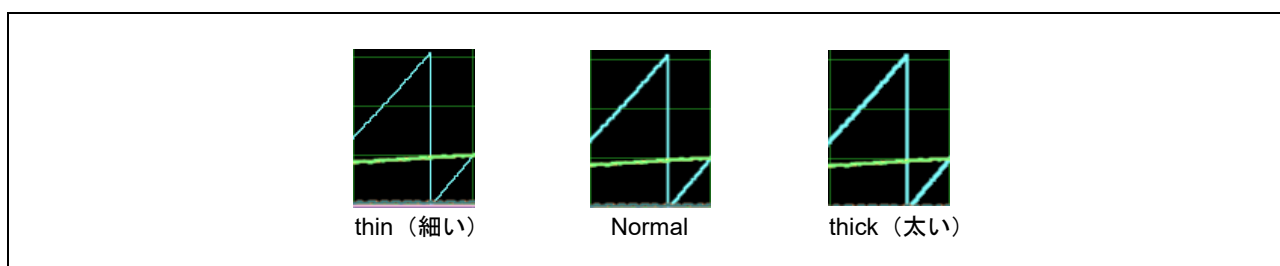


図 5.12 Line で指定できる波形表示の線の太さ

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

(10)~(13) の欄は、チャン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。

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

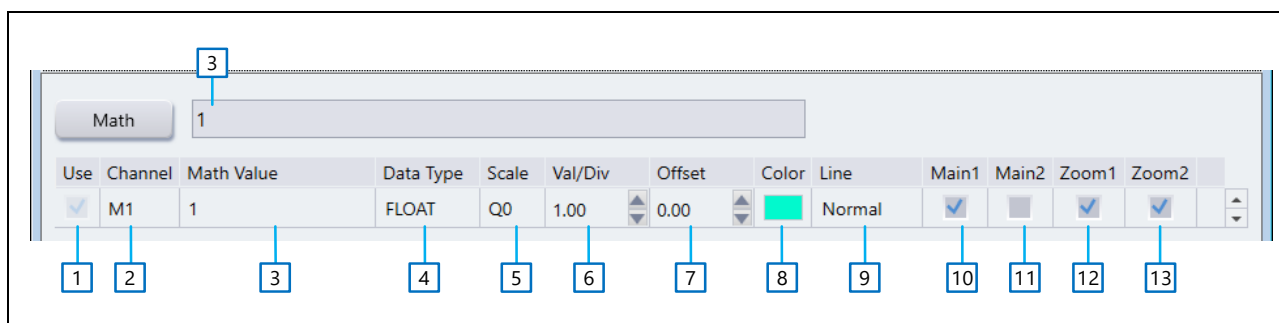


図 5.14 Math Channel Setting 部

(1) Use

Use は、Math 用チャン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 から始まるチャンネルが割り当てられます。チャンネル番号を変更することはできません。

(3) Math Value

Math Value には、チャンネル番号、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、以下の場合には、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
- 指定チャン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
チャンネル	チャンネル番号	ch(n)	ch1, ch2...	チャンネルの値は、Scale 値を適用した値（表示している値）を使用します。
	Math チャンネル番号	M(n)	M1, M2...	チャンネルの値は、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	<ul style="list-style-type: none"> $z^1(ch1)+ch2$ $z^{-1}(M1)$ 	優先度 1 (高)
	逆正接 ^{注2}	Atan2(ch(n),ch(n))	<ul style="list-style-type: none"> Atan2(ch1,ch2) Atan2(2,3) (= 0.9827937) 	優先度 2
	正弦 ^{注3}	sin()	<ul style="list-style-type: none"> sin(ch1) sin(60) (= -0.3048106) 	
	余弦 ^{注3}	cos()	<ul style="list-style-type: none"> cos(ch1) cos(60) (= -0.952413) 	
	正接 ^{注3}	tan()	<ul style="list-style-type: none"> tan(ch1) tan(60) (= 0.3200404) 	
	逆正接 ^{注3}	Atan()	<ul style="list-style-type: none"> Atan(ch1) Atan(60) (= 1.554131) 	
	平方根	sqrt()	<ul style="list-style-type: none"> sqrt(2) (= 1.414214) 	優先度 3
演算子	べき乗	\wedge	<ul style="list-style-type: none"> 2^3 (= 8) 	優先度 4
	乗算 ^{注4}	*	<ul style="list-style-type: none"> ch1*7 	優先度 5
	除算	/	<ul style="list-style-type: none"> ch1/7 	
	加算	+	<ul style="list-style-type: none"> ch1+5 	優先度 6 (低)
	減算	-	<ul style="list-style-type: none"> 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 の引数はチャンネル情報 (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: $2(3+4)$, $(1+3)(2+4)$, $5\sin(60)$ 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、チャン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、チャン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、チャン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、チャンネル設定した変数の波形表示の線の太さを選択します。

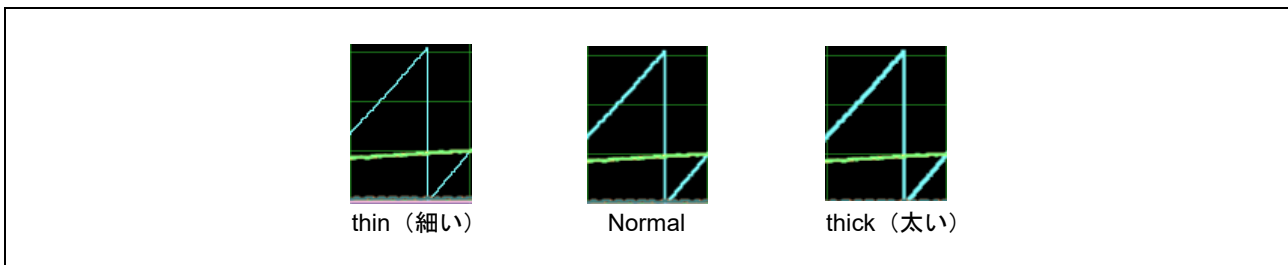


図 5.15 Line で指定できる波形表示の線の太さ

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

(10)~(13) の欄は、チャン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 用チャンネルの Math Value 欄でチャンネル番号 (ch7 など) または Math チャンネル番号 (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。

Window-Channel Settings

Use	Channel	Channel Variable	Data Type	Scale
<input checked="" type="checkbox"/>	7	g_motor_current0_ctrl.f_id_ref	FLOAT	Q2

Math: ch7

Use	Channel	Math Value	Data Type	Scale
<input checked="" type="checkbox"/>	M1	ch7	FLOAT	2
<input checked="" type="checkbox"/>	M2	M1	FLOAT	1

(Scale)

→ $1/2^2 = 1/4$

→ $1/2$

→ $1/1 = 1$

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 用チャンネル情報を編集し、「Math」ボタンをクリックすると Math 用チャン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。

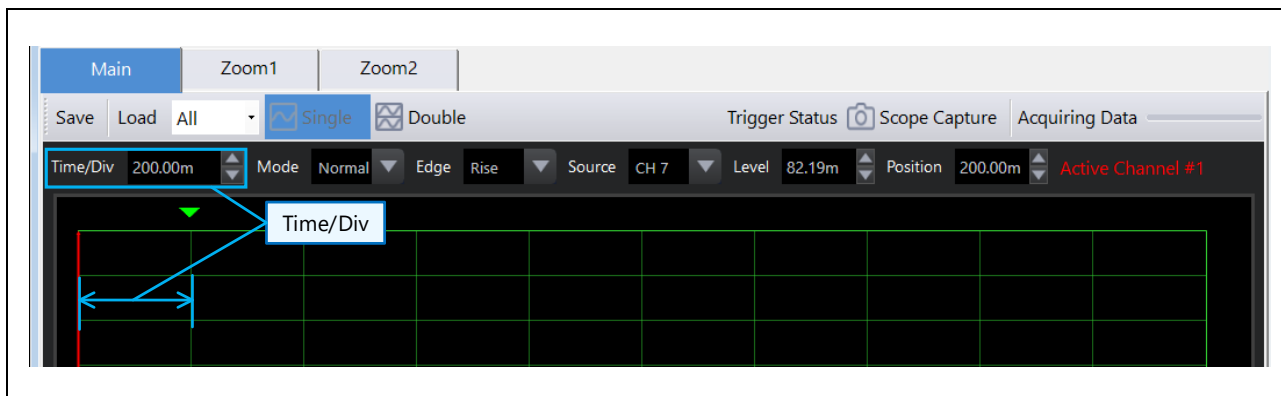


図 5.17 Time/Div 選択

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。

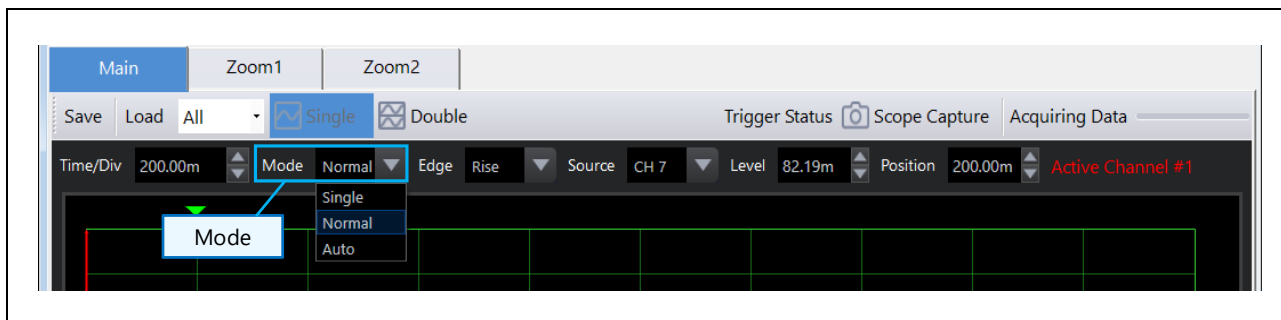


図 5.18 Mode 選択

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。

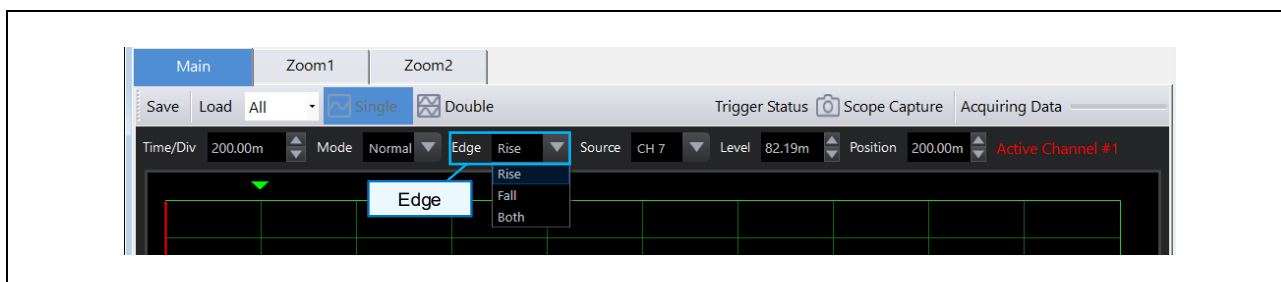


図 5.19 Edge 選択

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。

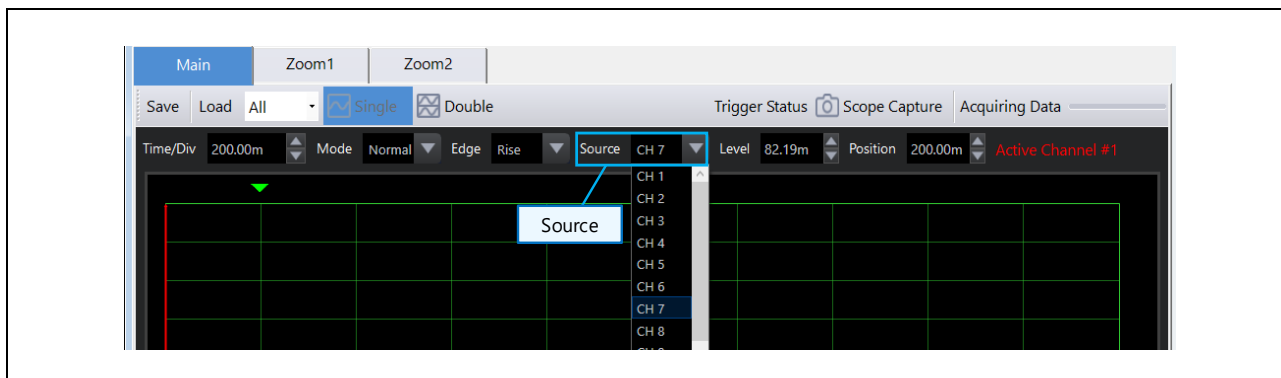


図 5.20 Source 選択

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。

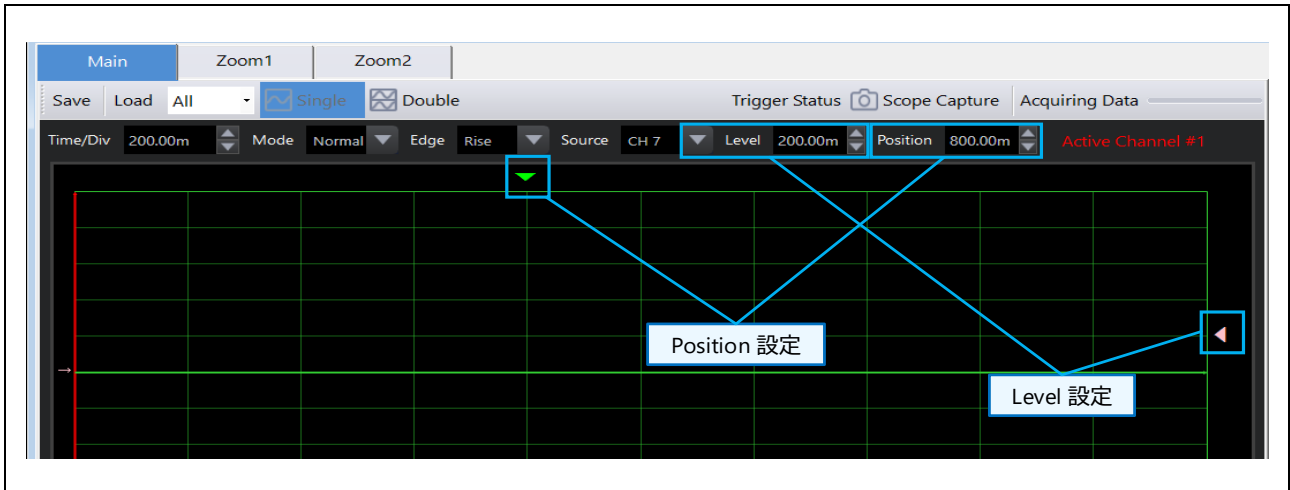


図 5.21 Level と Position 設定

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）。

$$\text{Length (表示画面の横幅)} = \text{Time/Div} \times 10 \div \text{Sample}$$

(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。

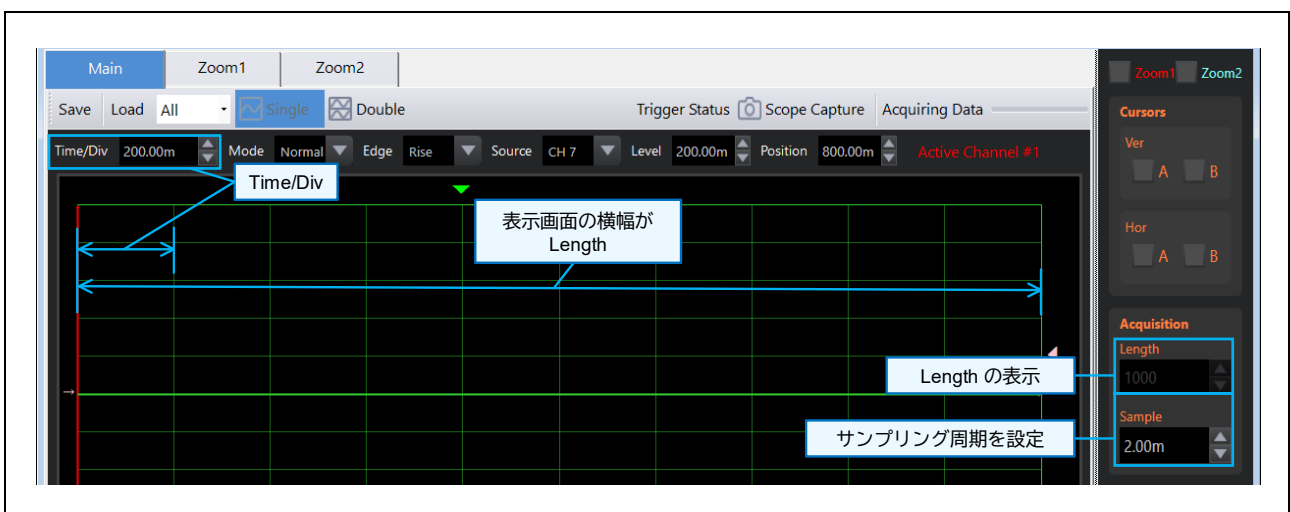


図 5.22 計測設定部の Acquisition (Length と Sample)

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。

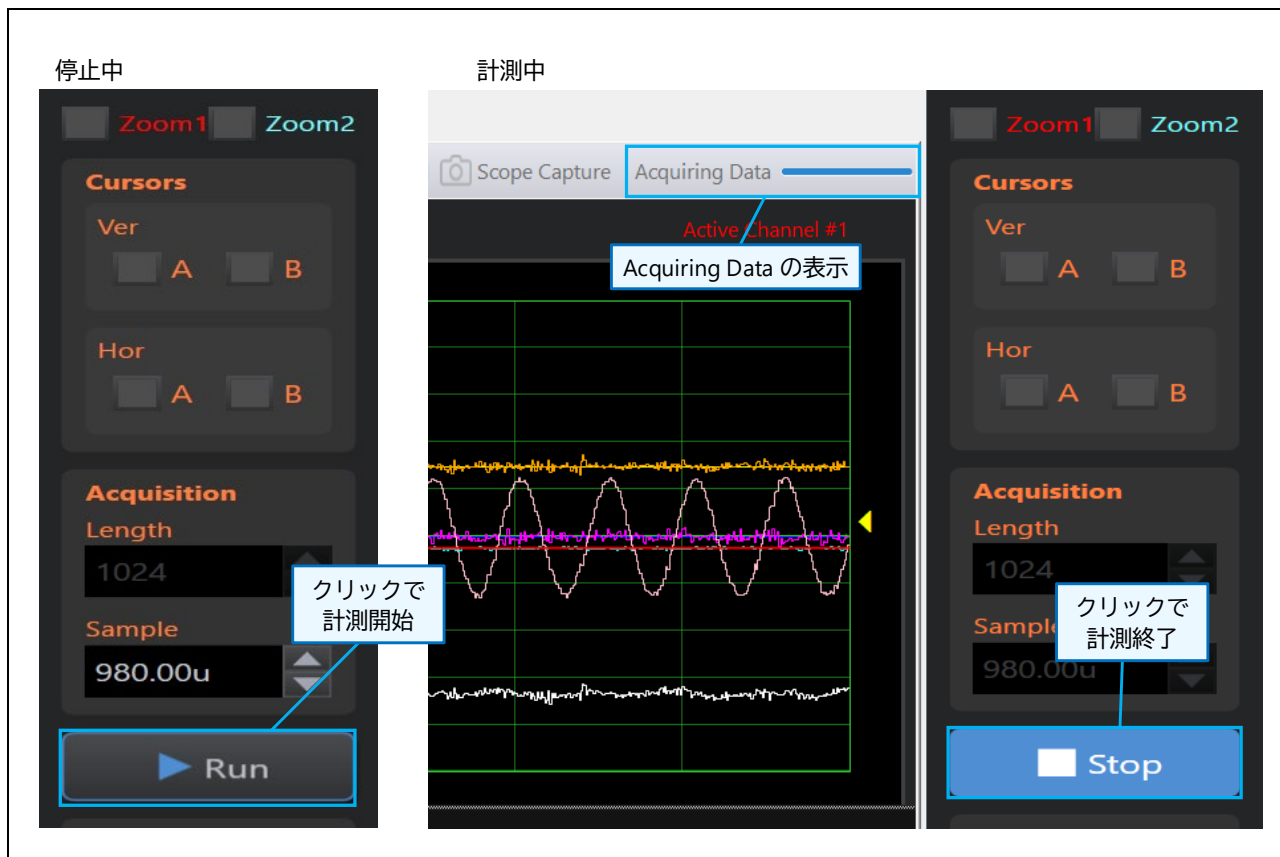


図 5.23 波形計測の開始と停止

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)	<ul style="list-style-type: none"> 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)	<ul style="list-style-type: none"> 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。

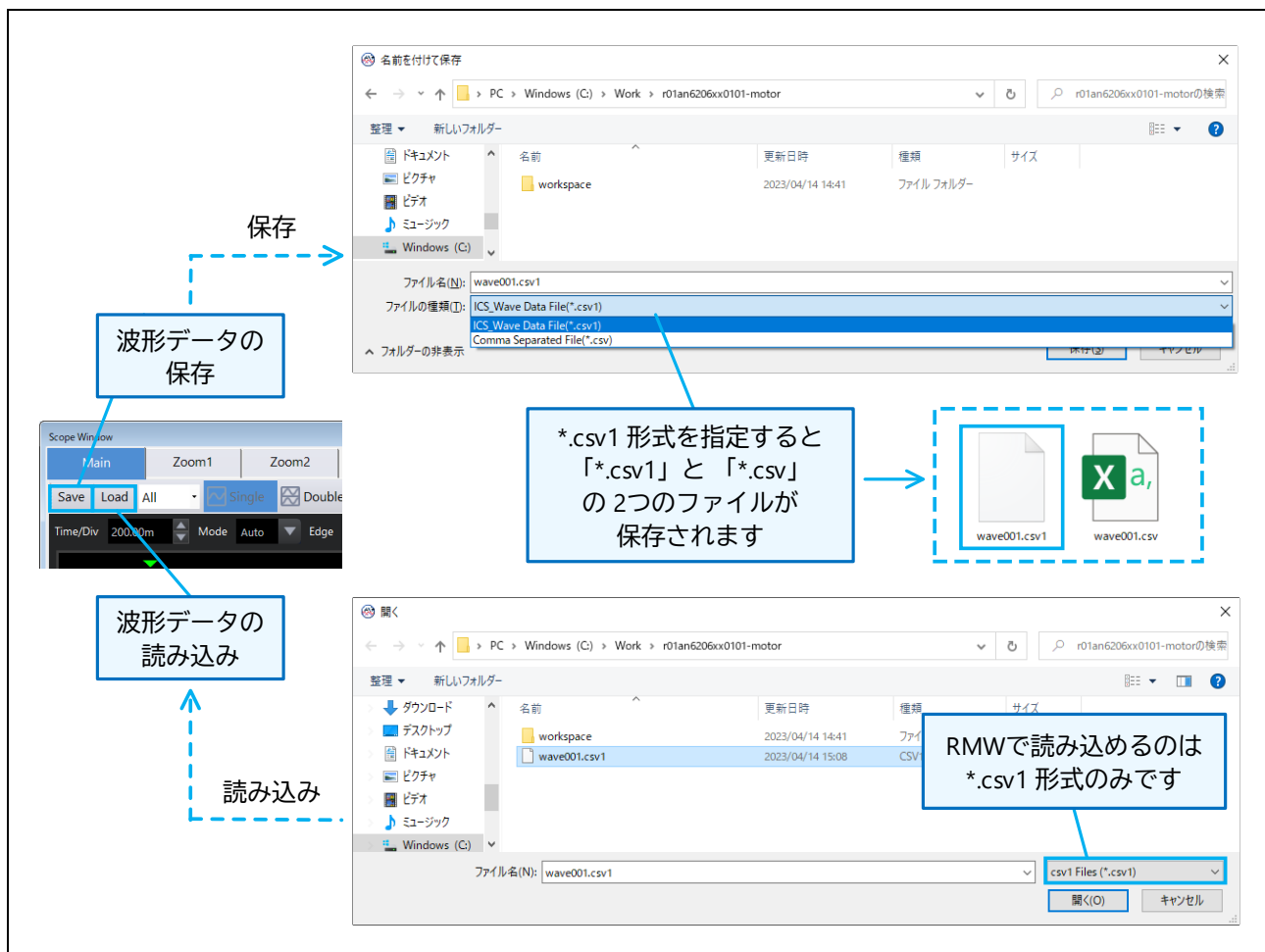


図 5.24 波形データの保存と読み込み

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。

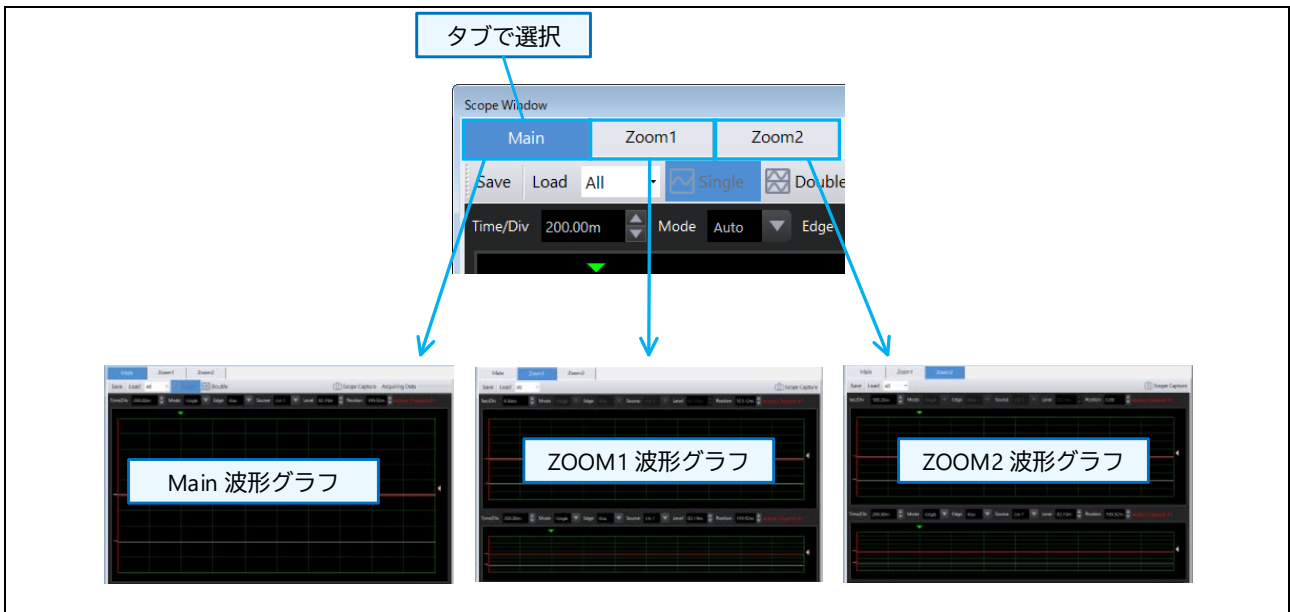


図 5.25 波形グラフの表示切り替え

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。

各チャン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。

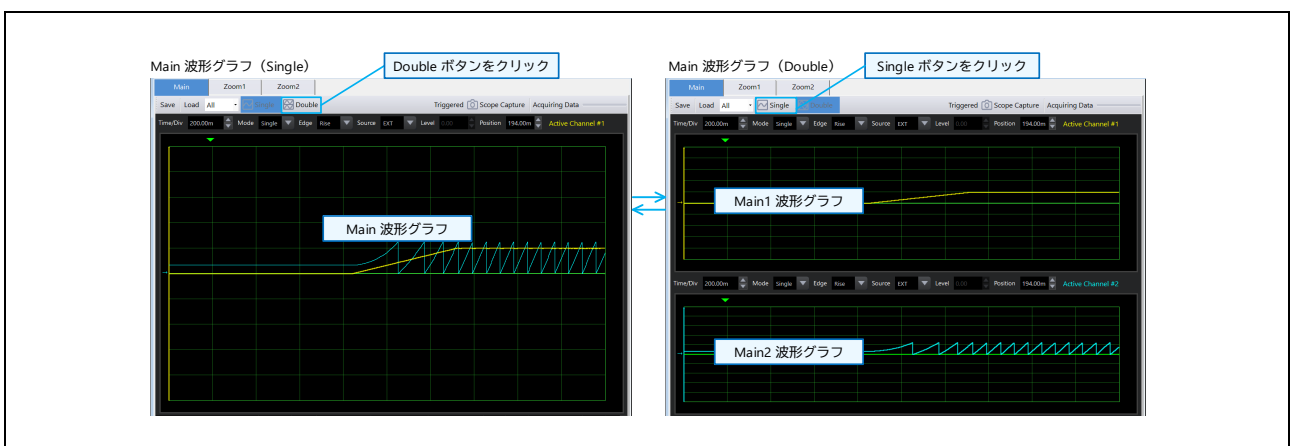


図 5.26 波形 2 画面表示

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際は、左下の▲は動きません。

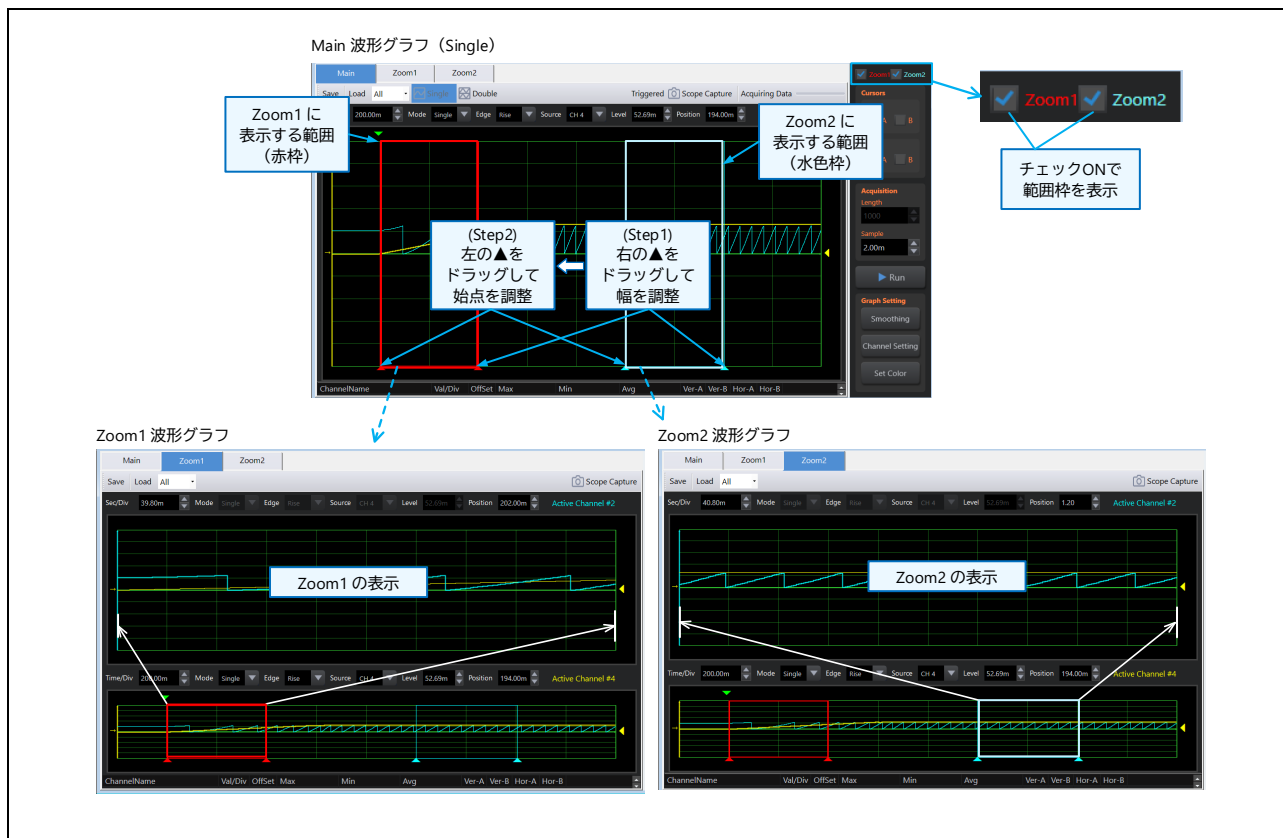


図 5.27 Zoom 表示の範囲指定 (Single 表示時)

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。

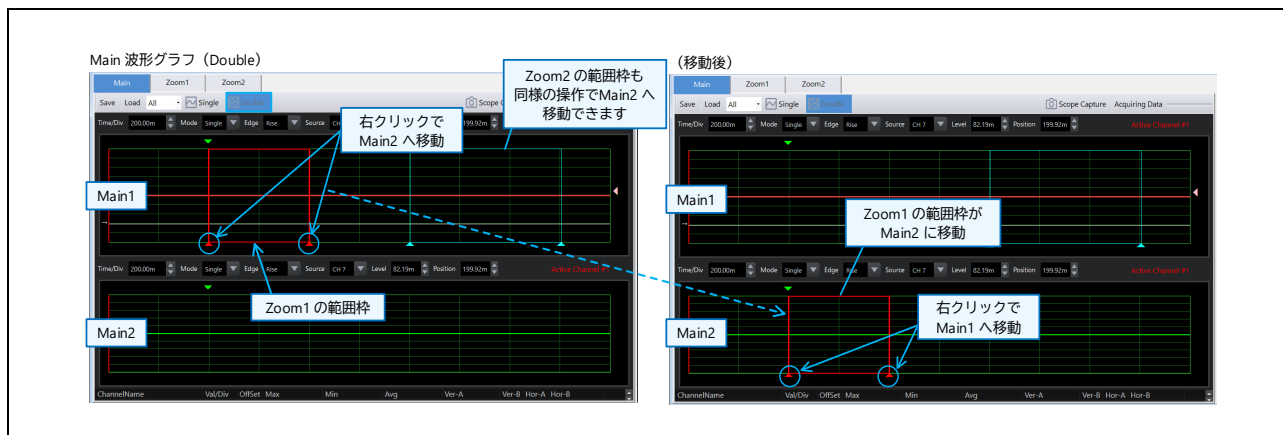


図 5.28 Zoom 表示の範囲指定(Double 表示時)

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

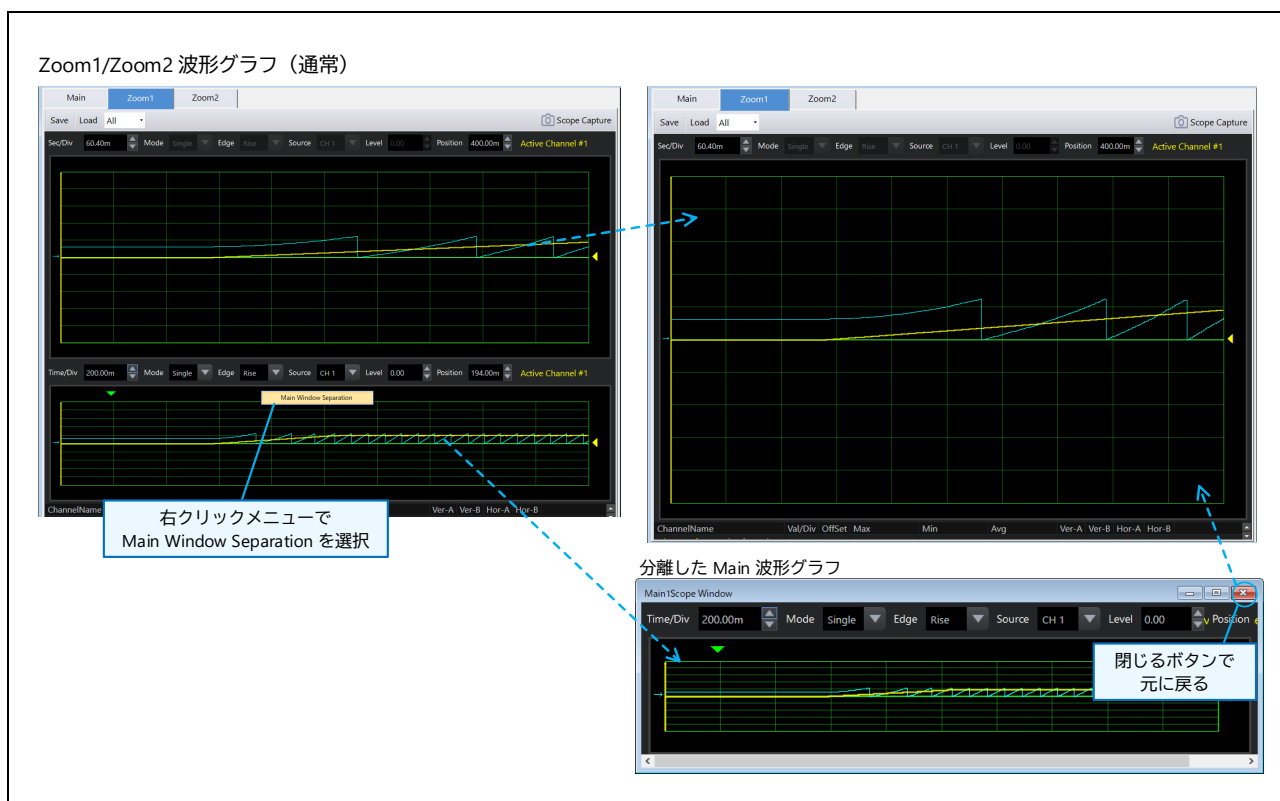


図 5.30 Main 波形画面の分離

5.7.7 Active Channel 表示

Main や Zoom のチャンネル情報表示部で、チャンネルを選択すると 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。

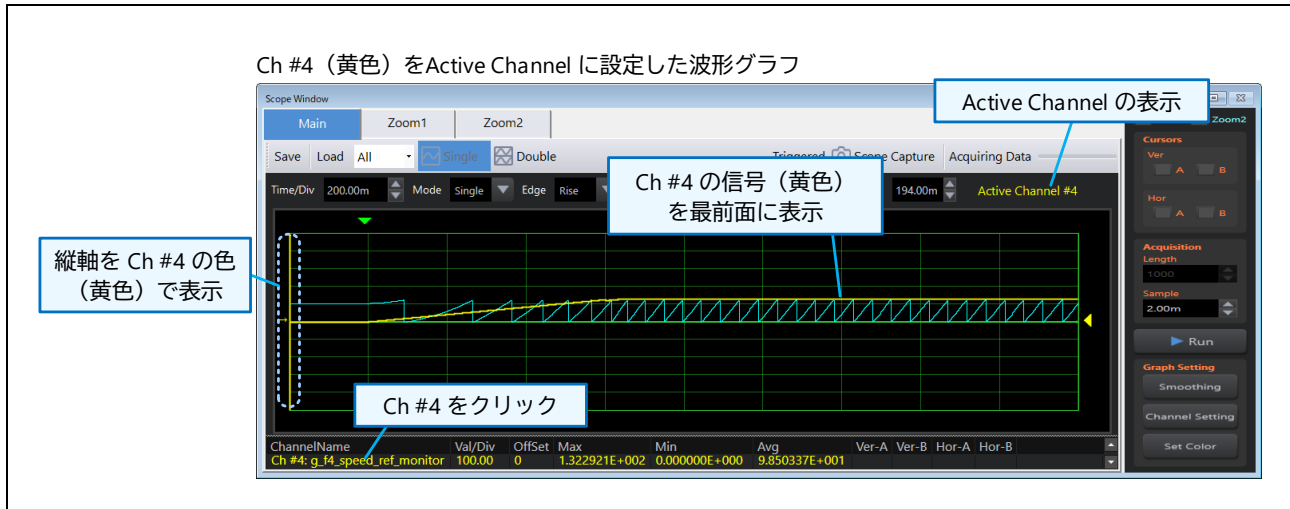


図 5.31 Active Channel と波形グラフの表示

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、画面下部のチャン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。

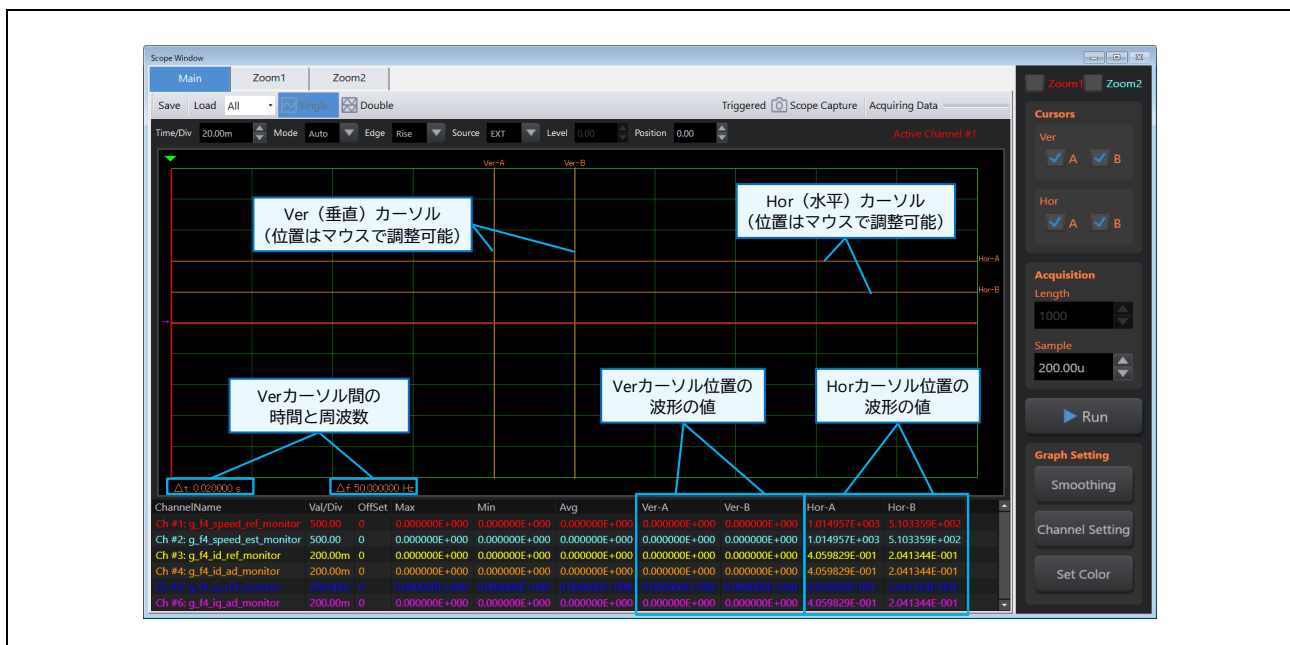


図 5.32 カーソル表示

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。

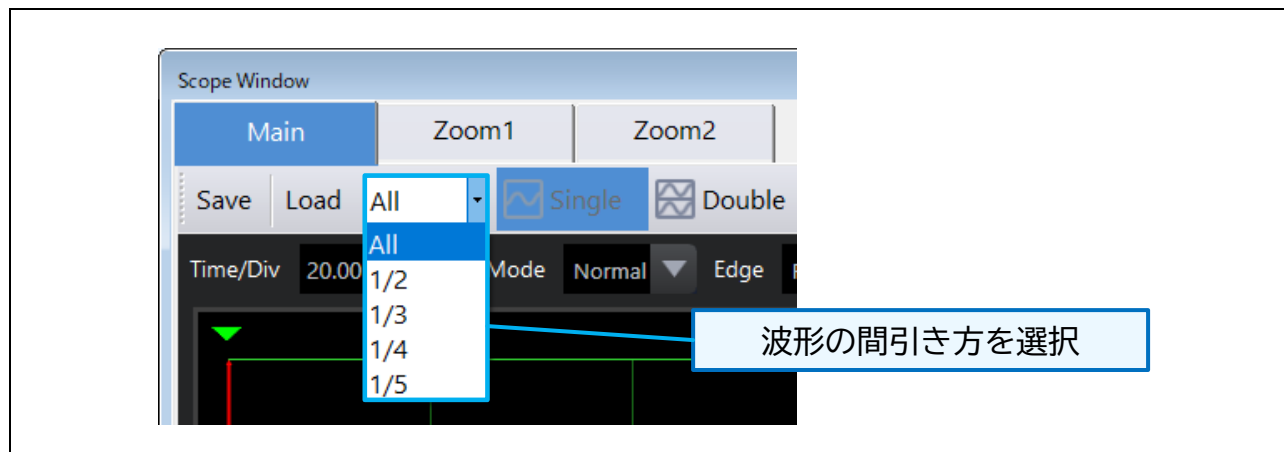


図 5.33 波形表示ポイントの設定

5.7.10 波形のスモーキング

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。

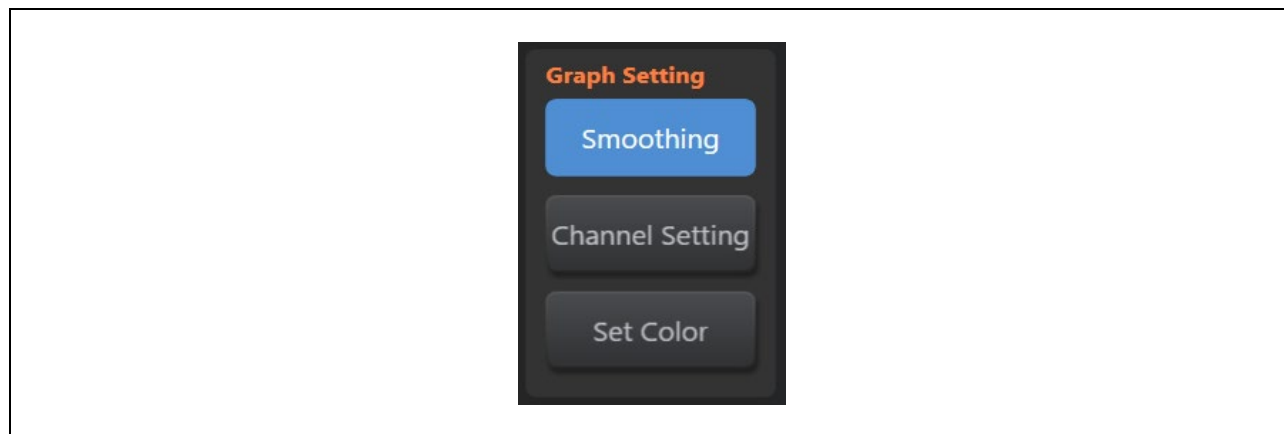


図 5.34 Smoothing ボタン (ON の状態)

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。

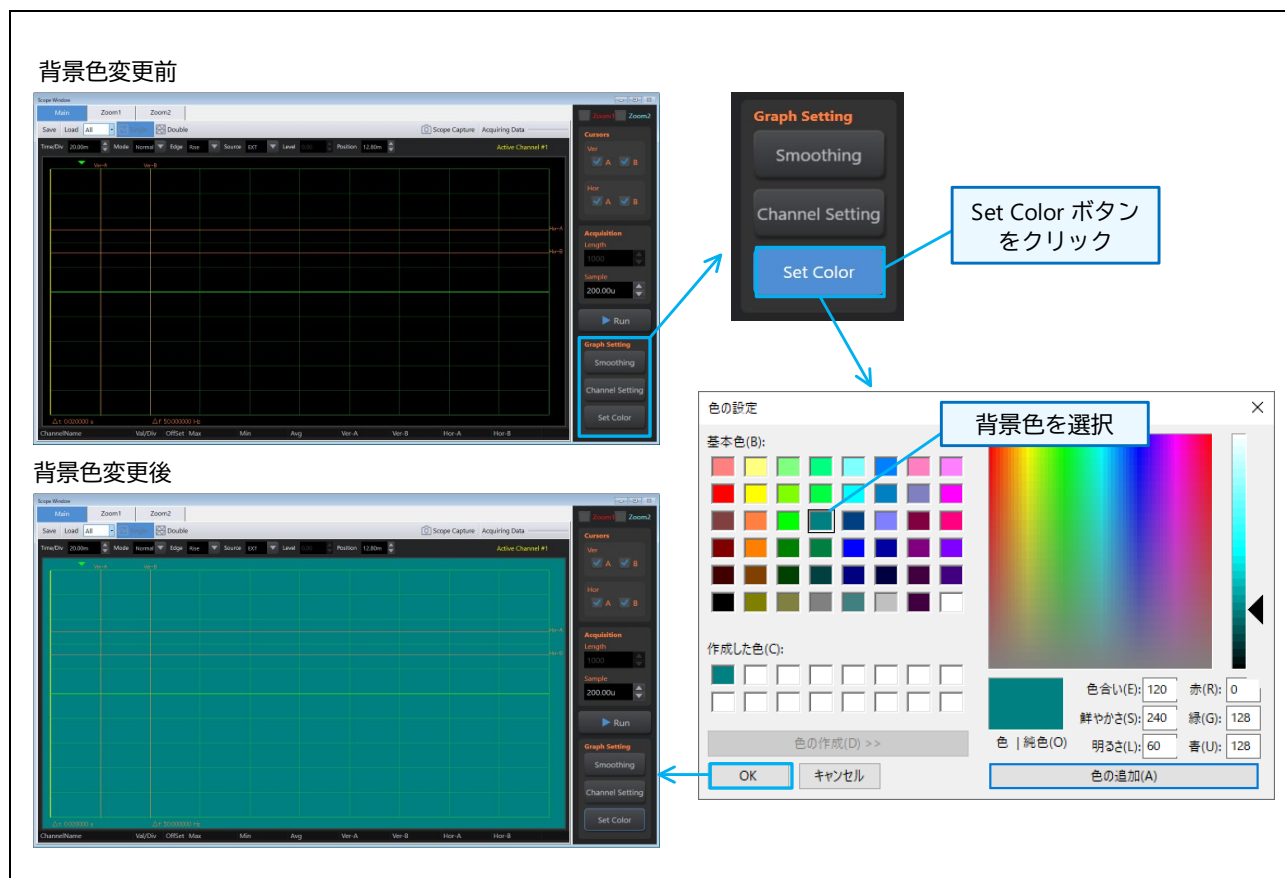


図 5.35 Set Color による背景色の変更

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ン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ンネル情報が 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ン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。

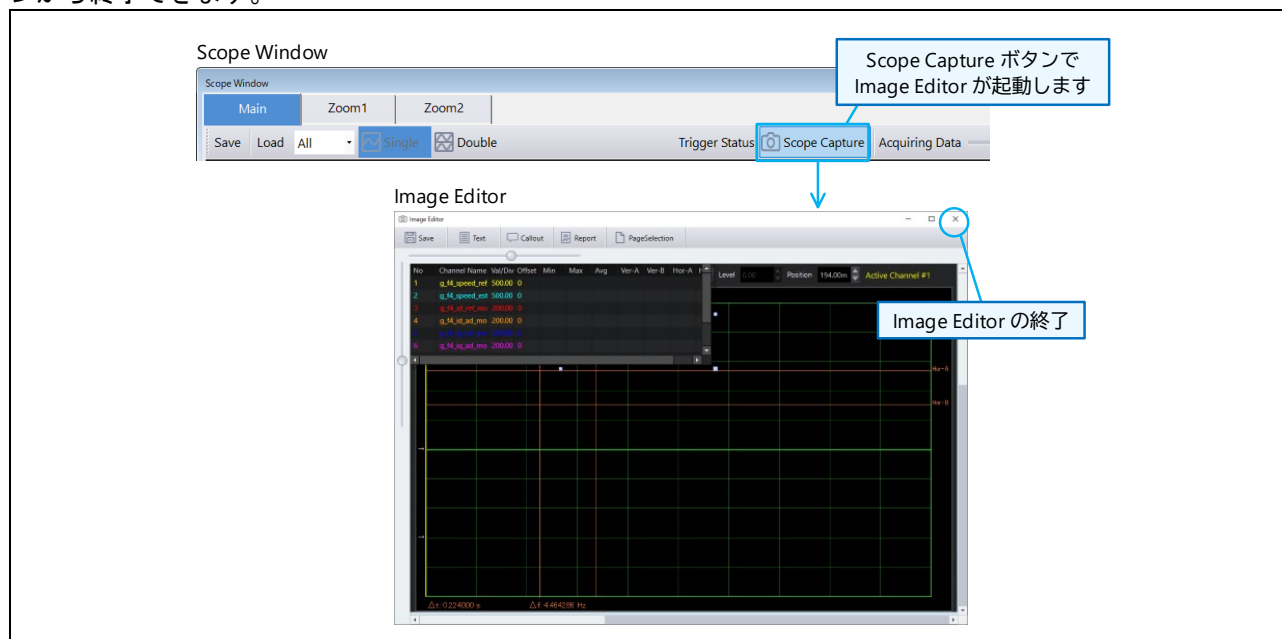


図 5.36 Image Editor の起動と終了

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。

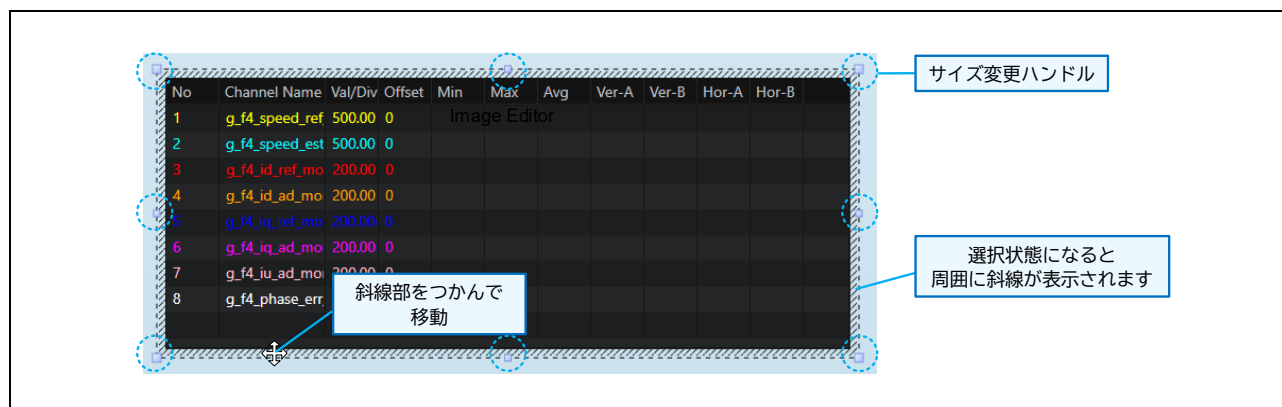


図 5.37 表示位置／サイズの変更

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。

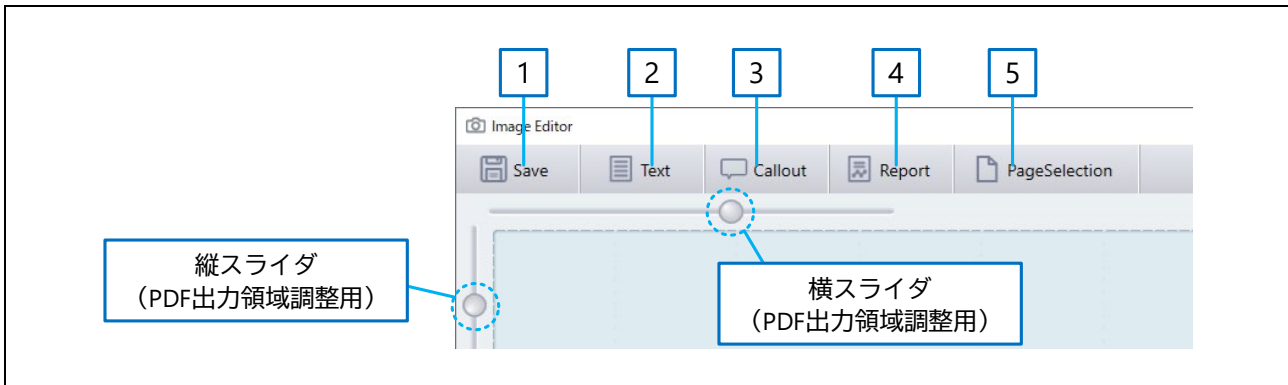


図 5.38 Image Editor の操作ボタン

(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。

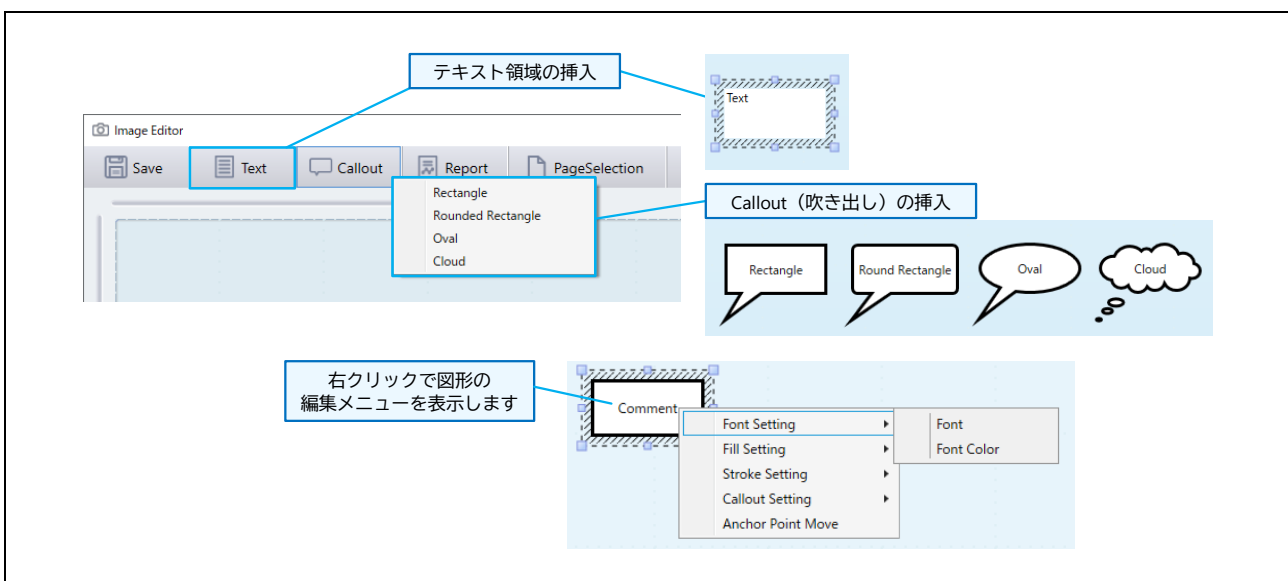


図 5.39 Callout (吹き出し) の種類

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。

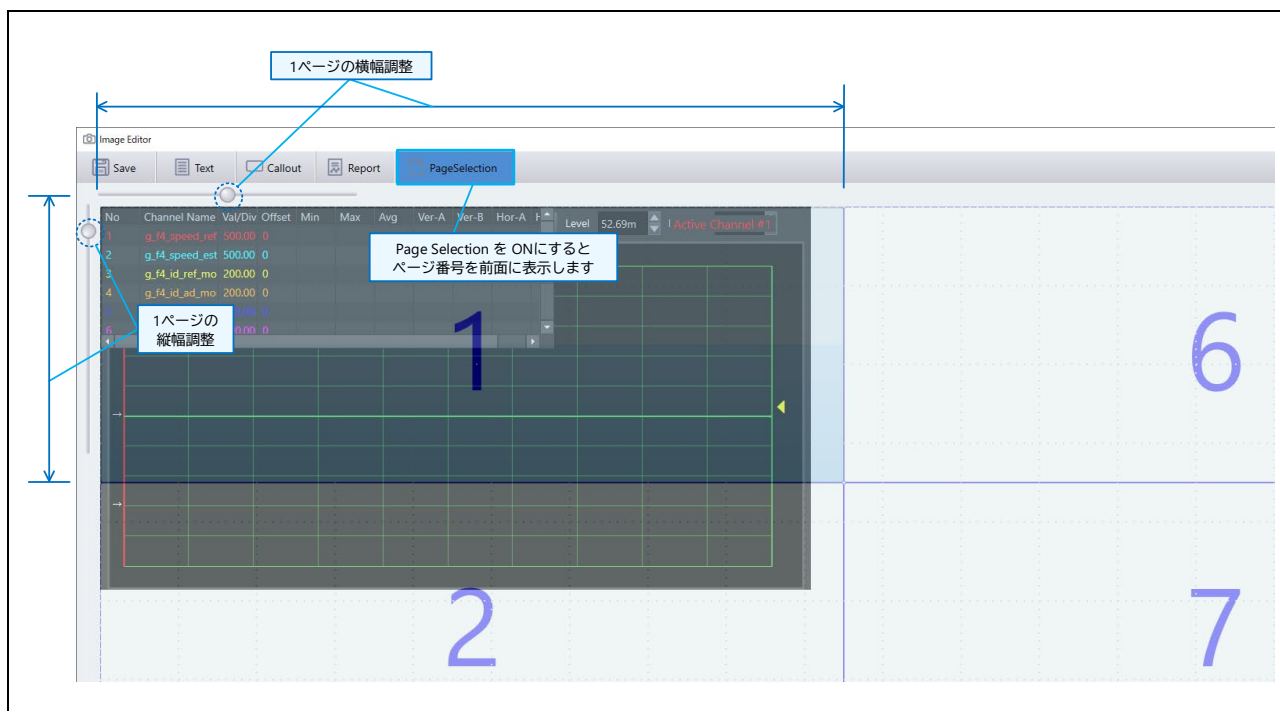


図 5.40 PDF 出力領域の調整

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、Analyzer 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。

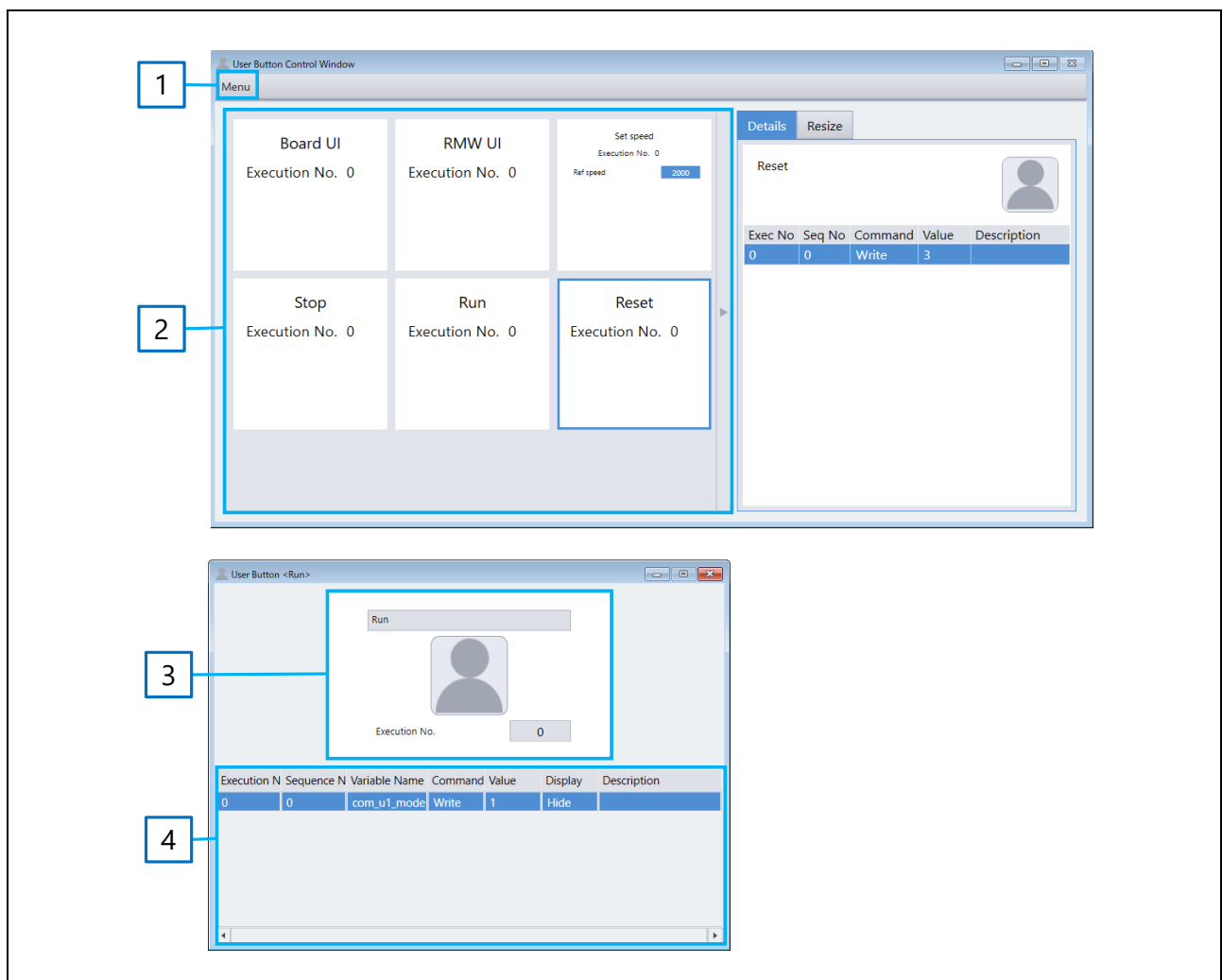


図 6.1 User Button 編集画面

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。

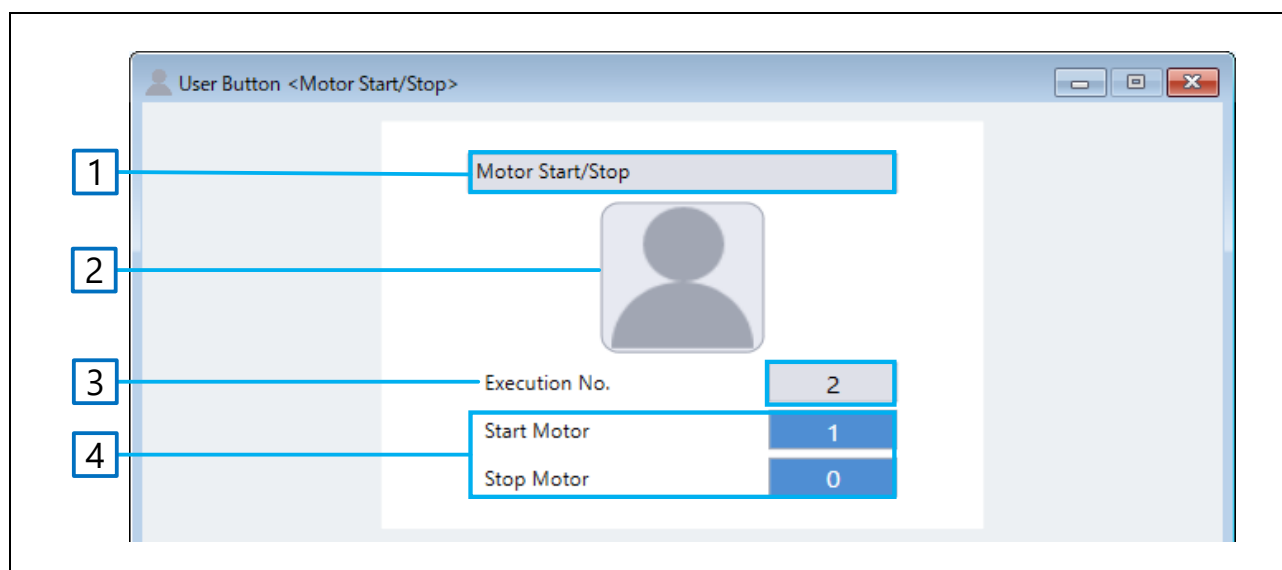


図 6.2 ボタン部画面

表 6.2 ボタン部の機能

No.	名称	説明
1	ボタン名	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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。

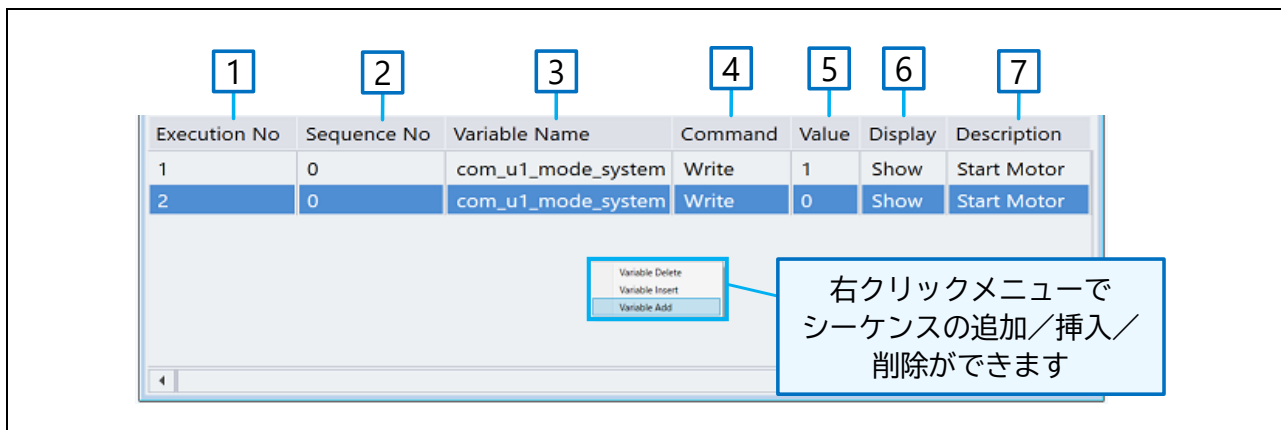


図 6.3 シーケンス編集部画面

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。

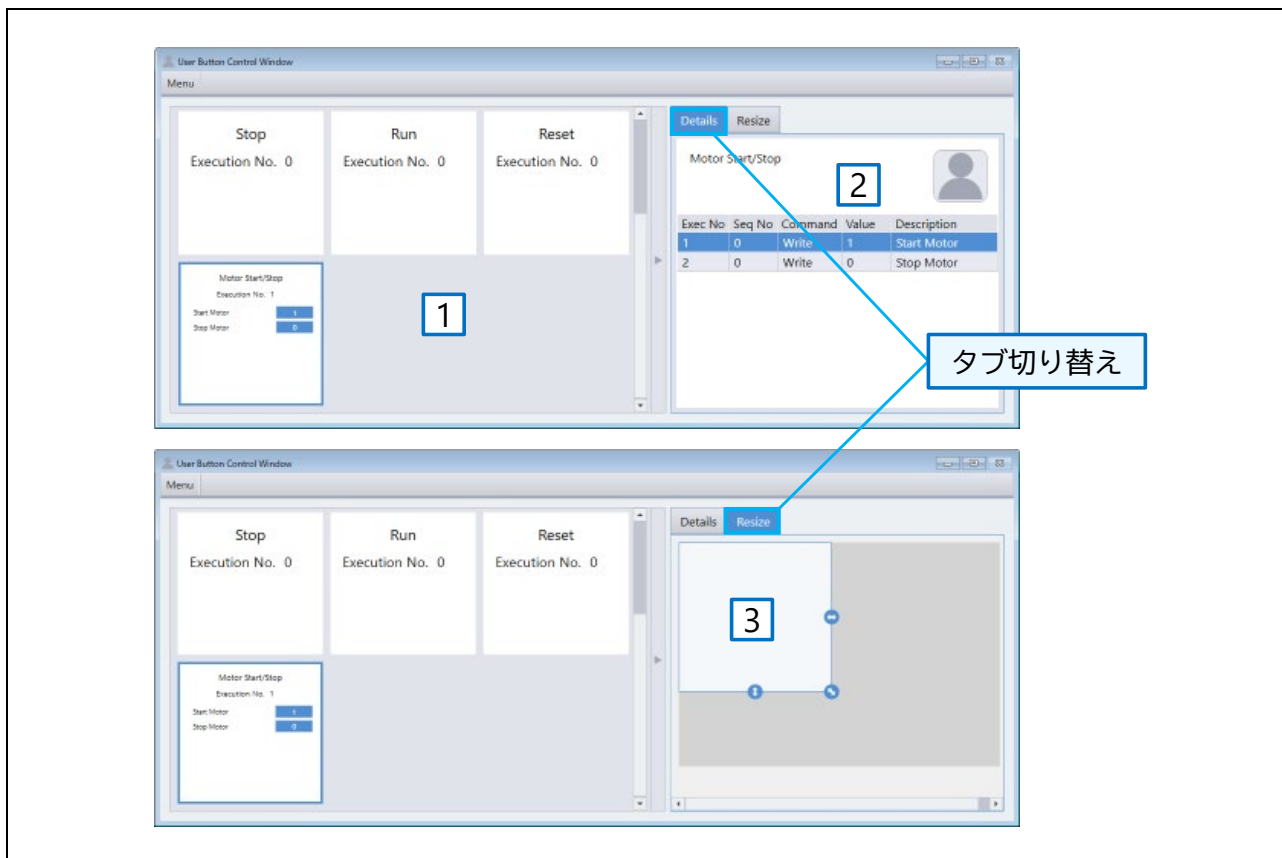


図 6.4 ボタン部画面

表 6.4 ボタン部の機能

No.	名称	説明
1	ボタン一覧	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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。

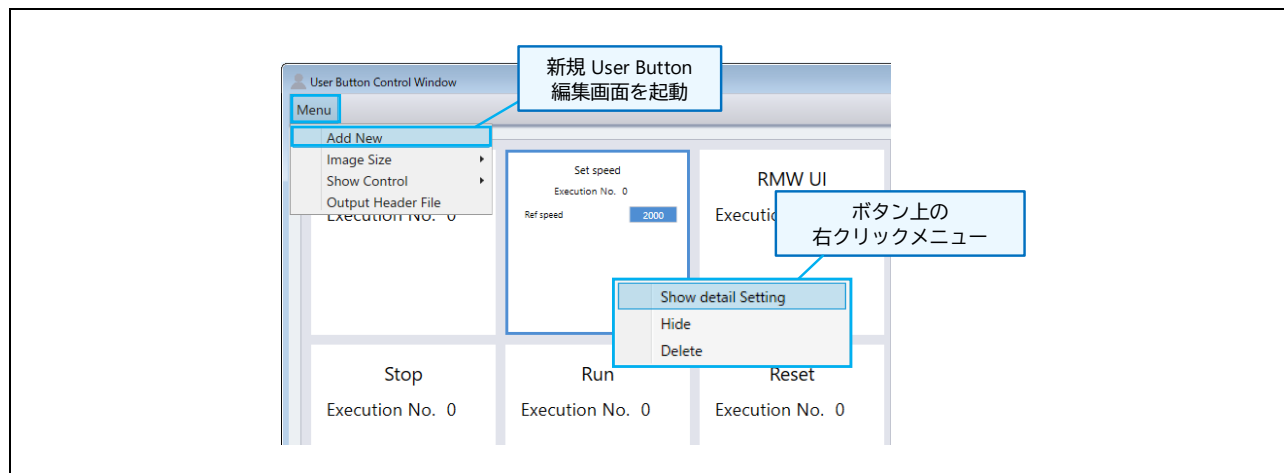


図 6.5 User Button の新規作成

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せん。

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ず、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。

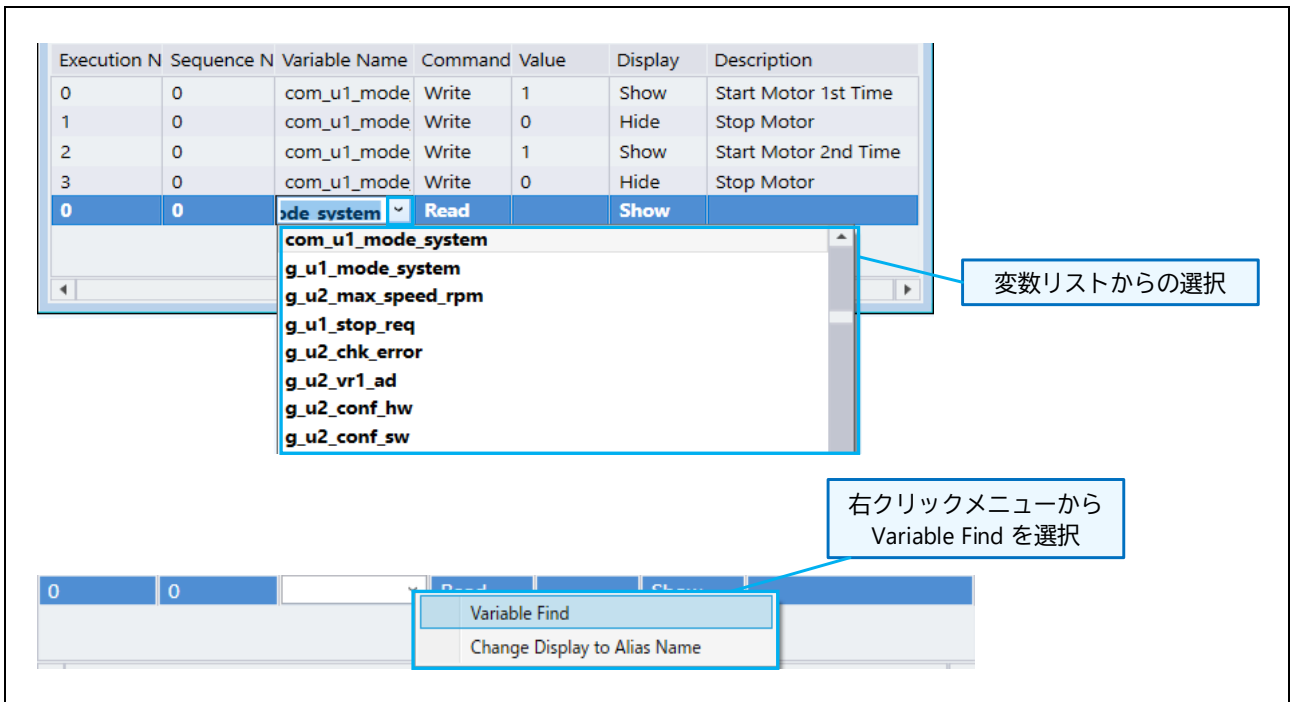


図 6.8 変数名の入力方法

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

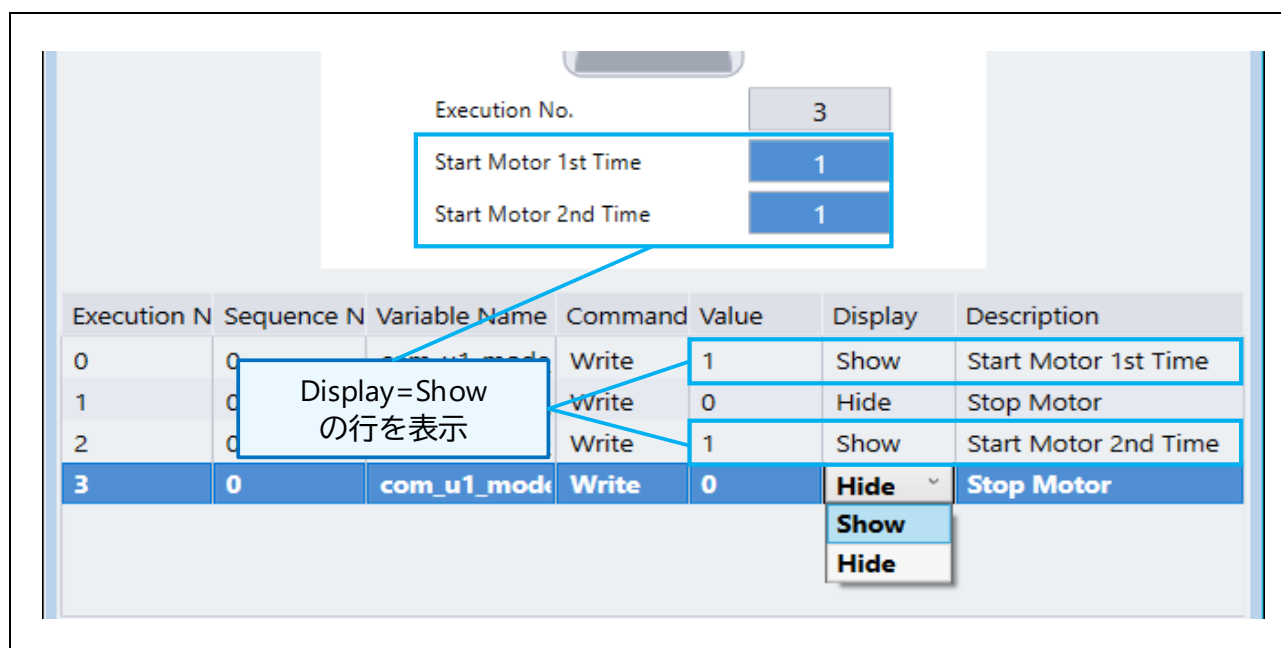


図 6.9 Description の設定

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。

Execution N	Sequence N	Variable Name	Command	Value	Display	Description
0	0	com_f4_ref_spe	Write	2000	Show	Ref speed
0	1	g_u1_enable_w	Read	A1	Hide	
0	2	com_u1_enabl	Write	A1	Hide	

3行目
4行目

内部変数 (A + 数字)

内部変数の使用例
 3 行目：変数の値を読み込んで内部変数「A1」に格納する。
 4 行目：内部変数「A1」の値を変数に書き込む。

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。

実行ボタン周辺で右クリックして Image を選択、画像を指定します

実行ボタンに画像が表示されます

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

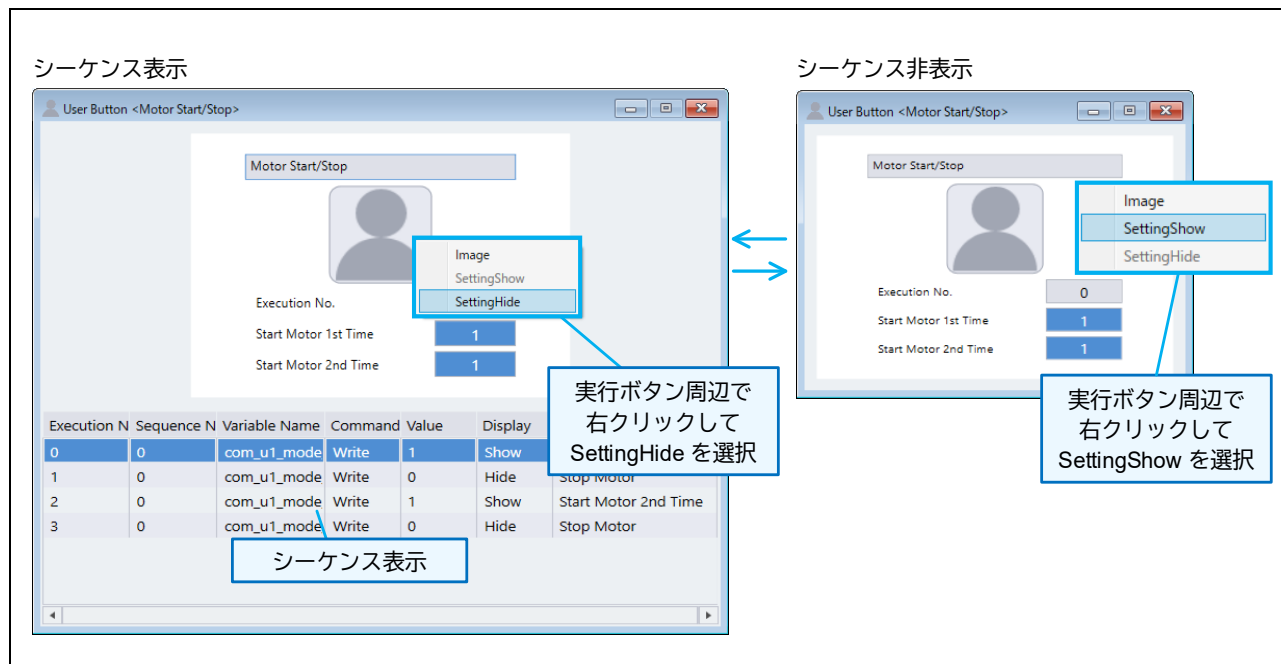


図 6.12 シーケンス編集部の表示／非表示

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。

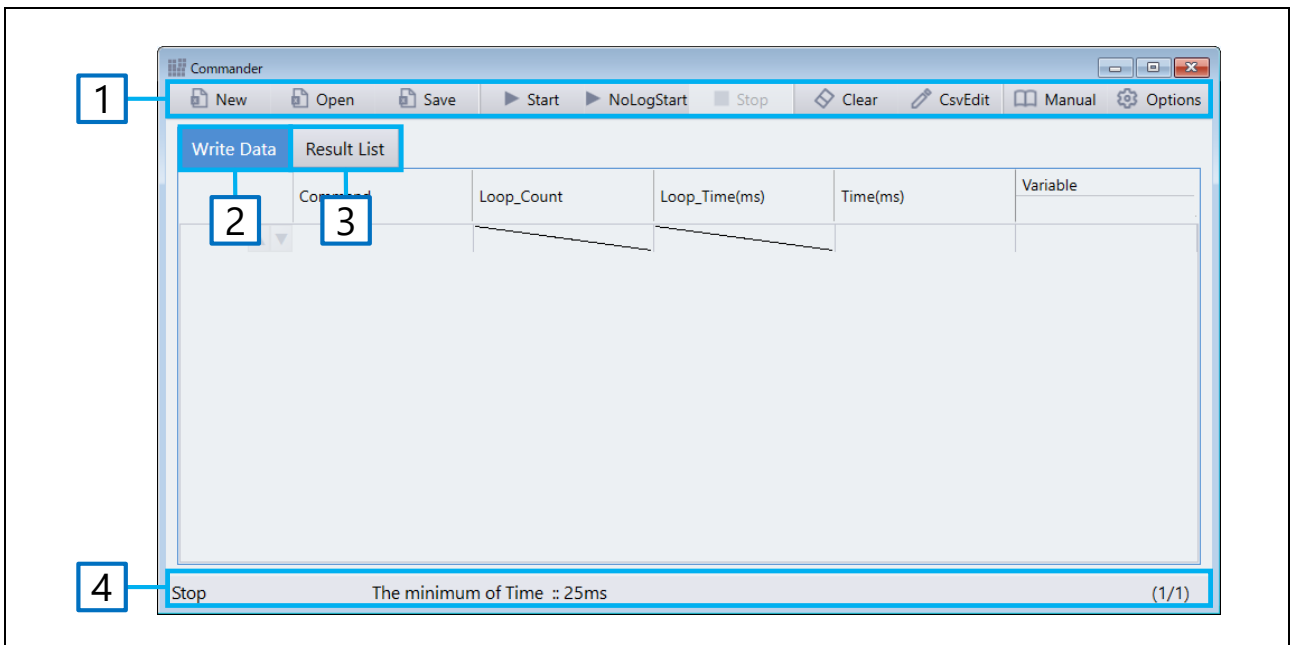


図 7.1 Commander 画面

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。

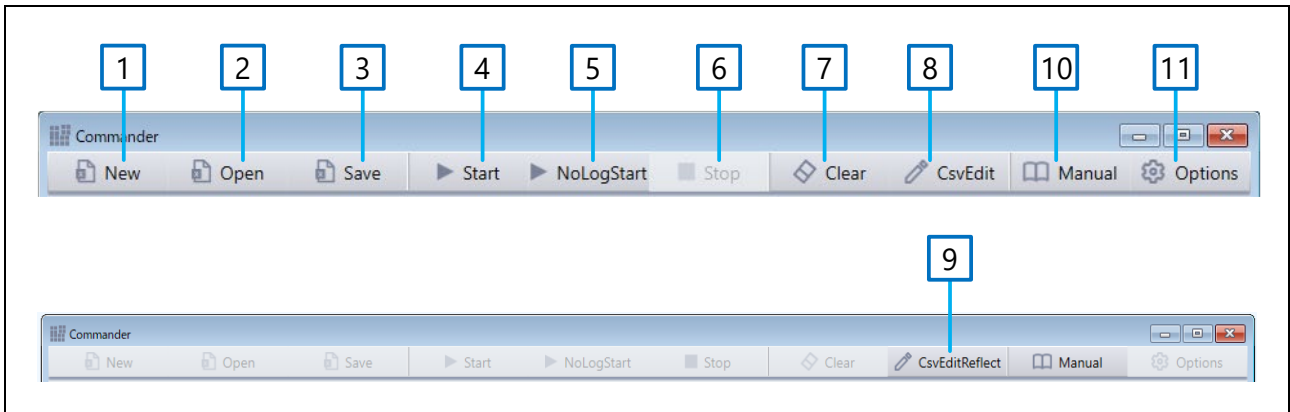


図 7.2 ツールバー

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。

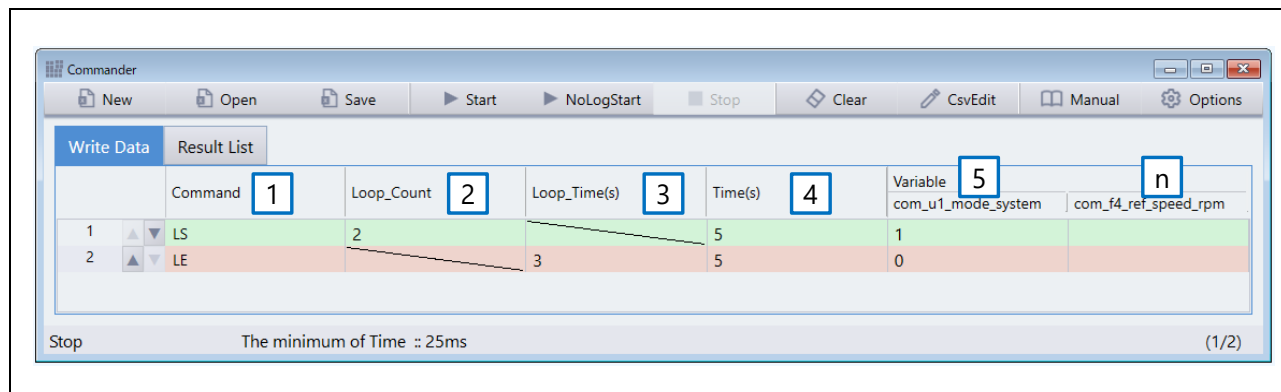


図 7.3 Write Data タブ画面

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。

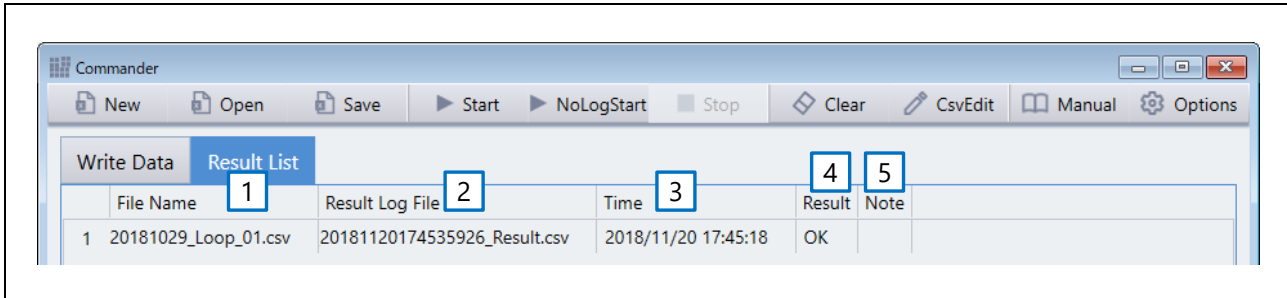


図 7.4 Result タブ画面

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。

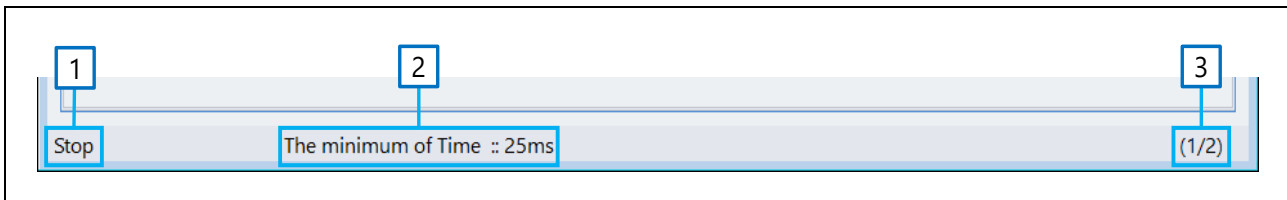


図 7.5 ステータスバー

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。

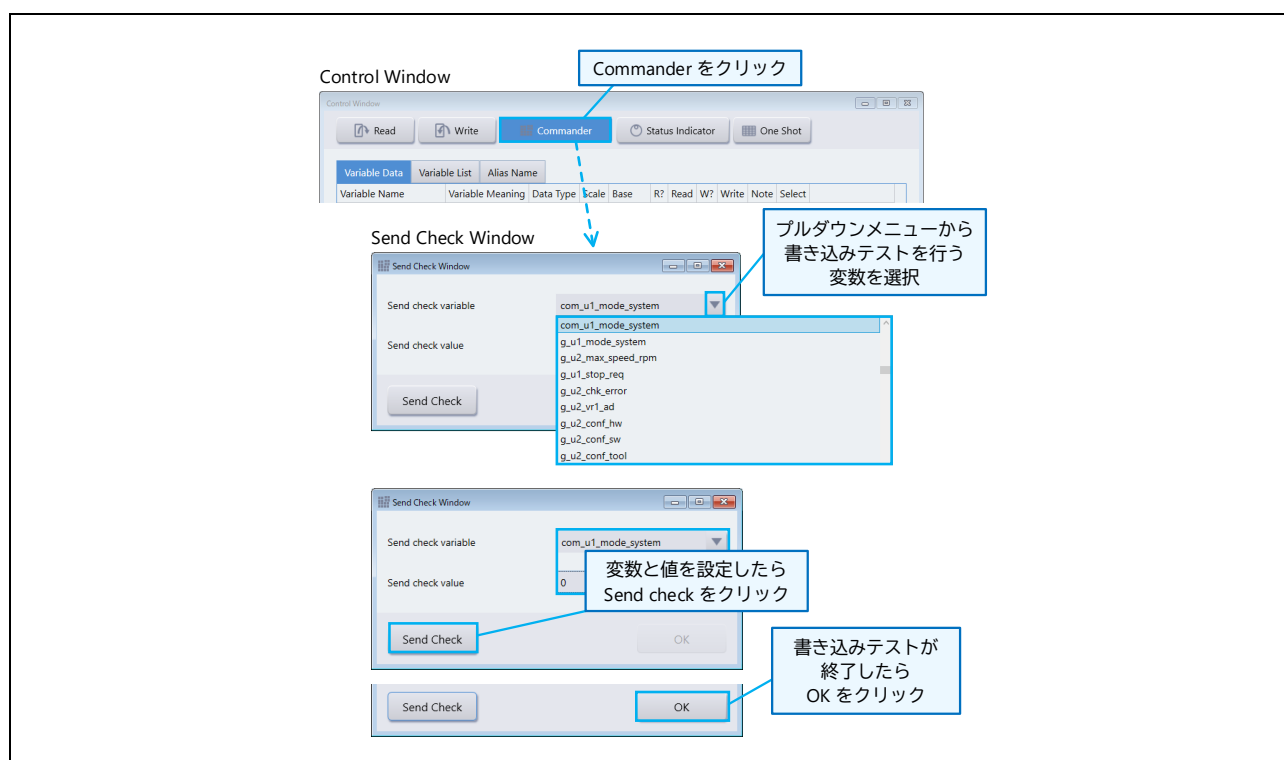


図 7.6 Commander の起動と Send Check Window

[操作]

- 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。

- 「Send check」ボタンを押して計測します。
- 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[※]。
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。

編集前						
	A	B	C	D	E	F
1	Command	Loop_Count	Loop_Time(ms)	Time(ms)	Please register available variables	
2						
3						

A~D 列の見出しは必須です
単位は変更できません (s/ms/ds)
表示を消して、E1セルから変数名を指定します

編集後 (例)						
	A	B	C	D	E	F
1	Command	Loop_Count	Loop_Time(ms)	Time(ms)	com_u1_mode_system	
2				5000		1
3				5000		0

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。繰り返しは複数（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。

ループ処理例

	A	B	C	D	E	F
1	Command	Loop_Count	Loop_Time(ms)	Time(ms)	com_u1_mode_system	
2	LS	2		5000		1
3	LE		3000	5000		0

繰り返し回数を指定します

繰り返しの開始行「LS」
繰り返しの終了行「LE」

繰り返しのインターバル時間を終了行で指定します

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

	A	B	C	D	E	F
1	Command	Loop_Count	Loop_Time(ms)	Time(ms)	com_u1_mode_system	
2	LS	2		5000		1
3	LE		3000	5000		0
4	LS	3		5000		1
5	LE		2000	5000		0

A

B

A ループ1回目

LS行	LE行	インターバル (3秒)	LS行	LE行
変数値 = 1 (5秒)	変数値 = 0 (5秒)		変数値 = 1 (5秒)	変数値 = 0 (5秒)

ループ2回目

B ループ1回目

LS行	LE行	インターバル (2秒)	LS行	LE行	インターバル (2秒)	LS行	LE行
変数値 = 1 (5秒)	変数値 = 0 (5秒)		変数値 = 1 (5秒)	変数値 = 0 (5秒)		変数値 = 1 (5秒)	変数値 = 0 (5秒)

ループ2回目

ループ3回目

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。

時間の単位を ms (ミリ秒) から s (秒) に変更します

変更前

	A	B	C	D	E	F
1	Command	Loop_Count	Loop_Time(ms)	Time(ms)	com_u1_mode_system	
2	LS	2		5000		1
3	LE		3000	5000		0

時間の単位変更後 (時間は上と同じ設定)

	A	B	C	D	E	F
1	Command	Loop_Count	Loop_Time(s)	Time(s)	com_u1_mode_system	
2	LS	2		5		1
3	LE		3	5		0

単位に合わせて時間を記述します

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。

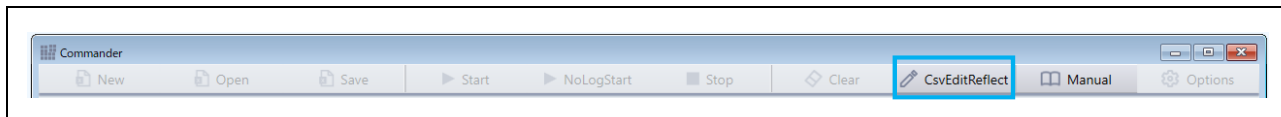


図 7.12 CSV Edit Reflect ボタン

(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^{※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^{※2}、ボタン表示が「CSV Edit」に切り替わります。

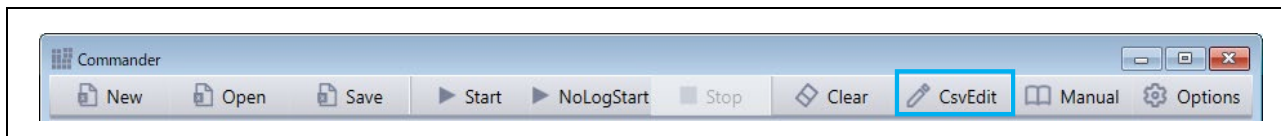


図 7.13 CSV Edit ボタン

- 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。

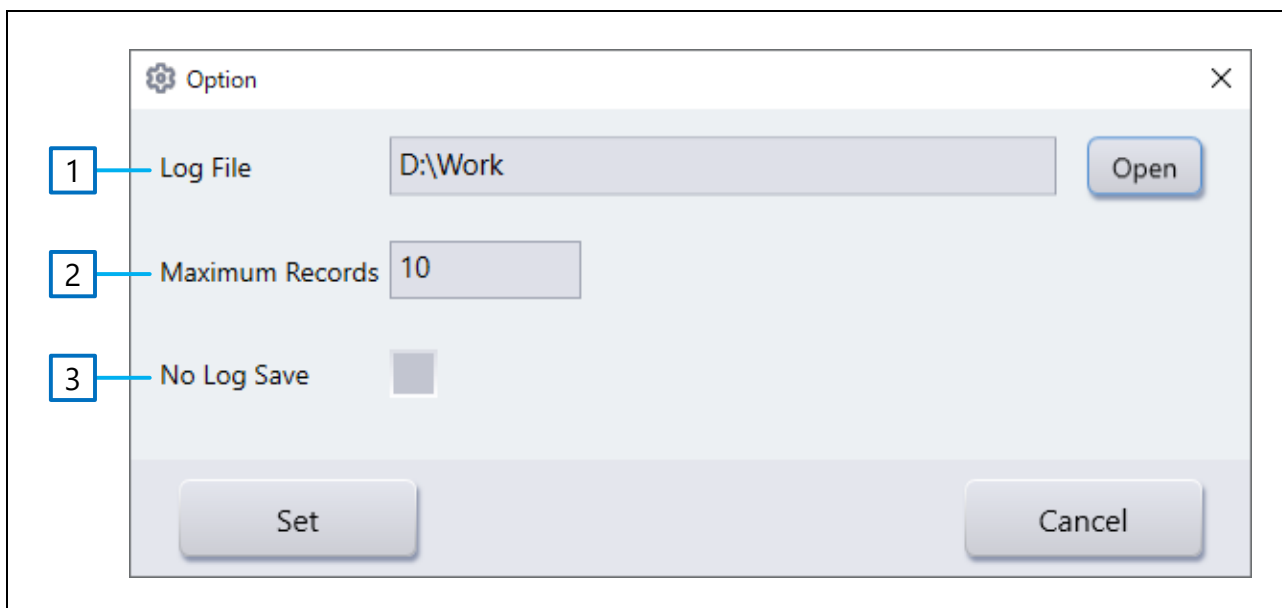


図 7.14 Option 設定画面

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。

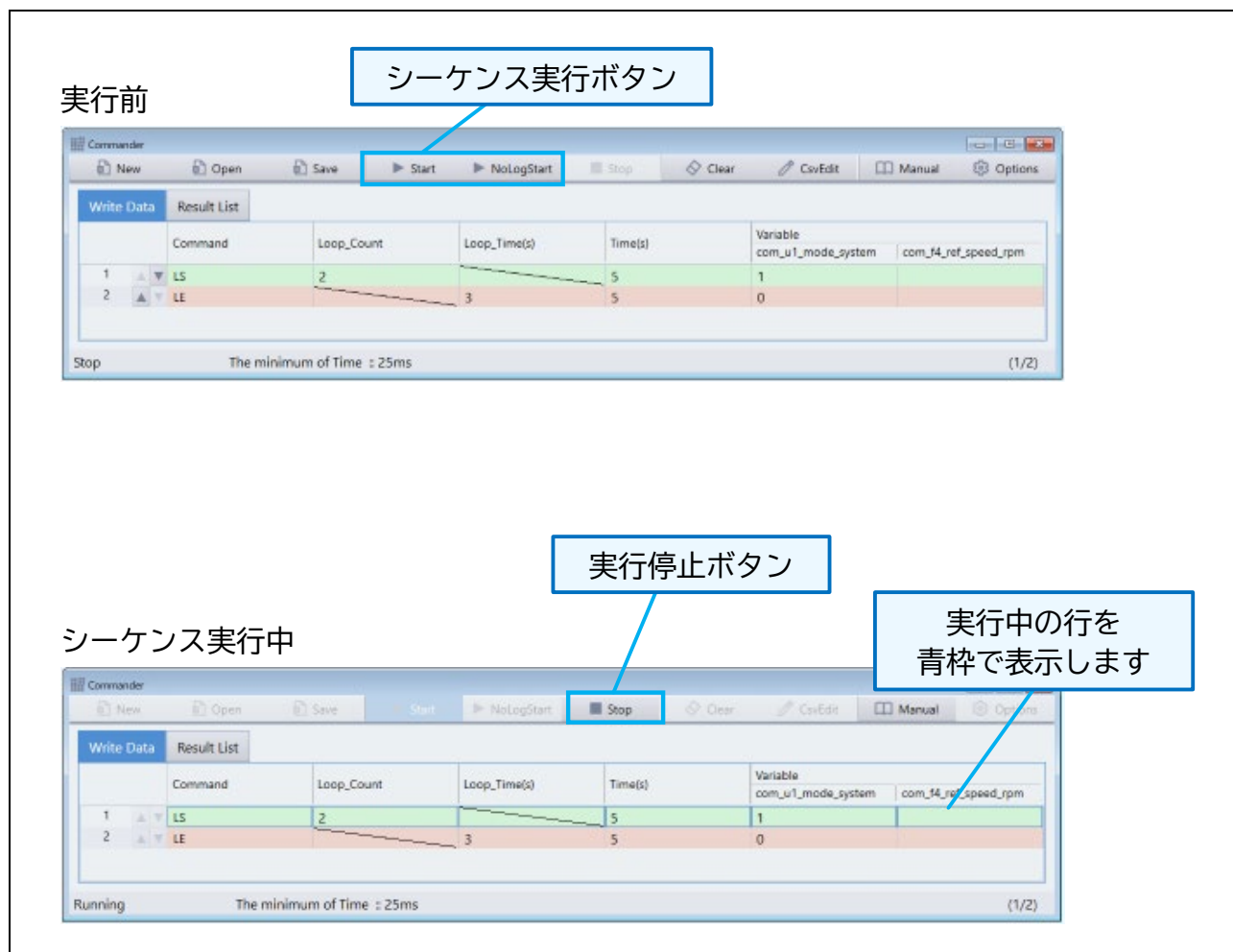


図 7.15 シーケンスの実行

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）。

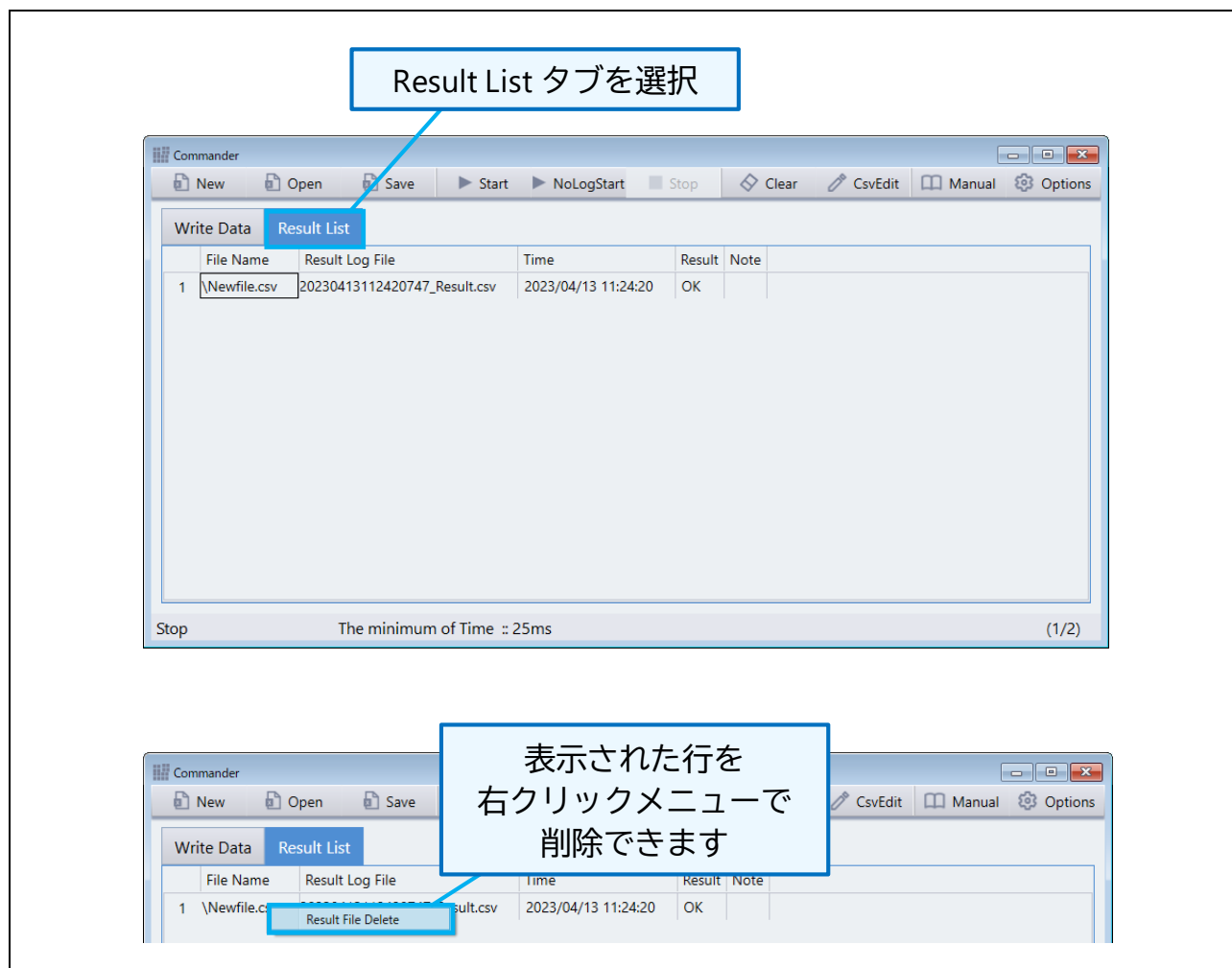


図 7.16 Result List 画面

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。

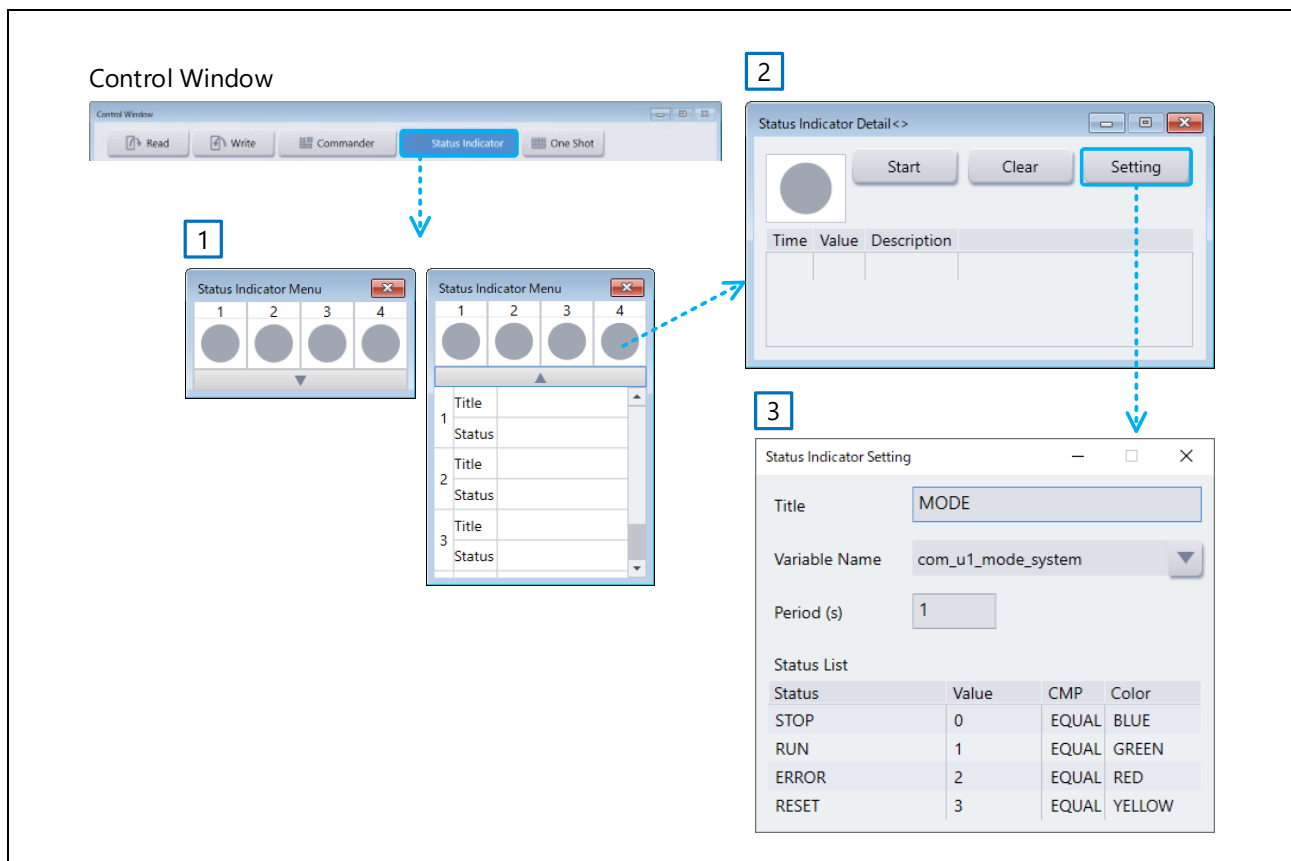


図 8.1 Status Indicator の画面構成

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。

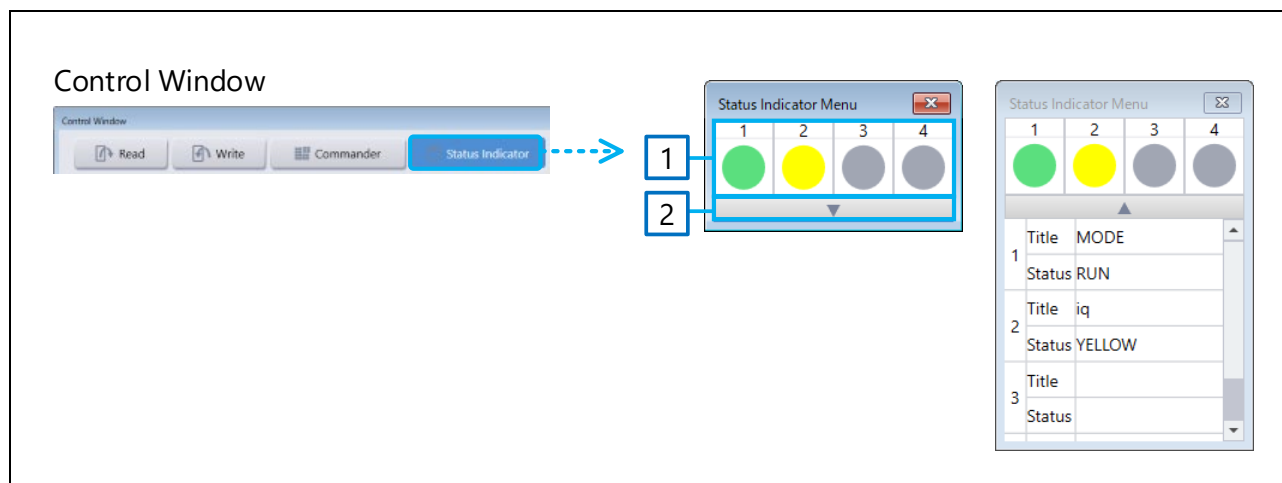


図 8.2 Status Indicator Menu 画面

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。

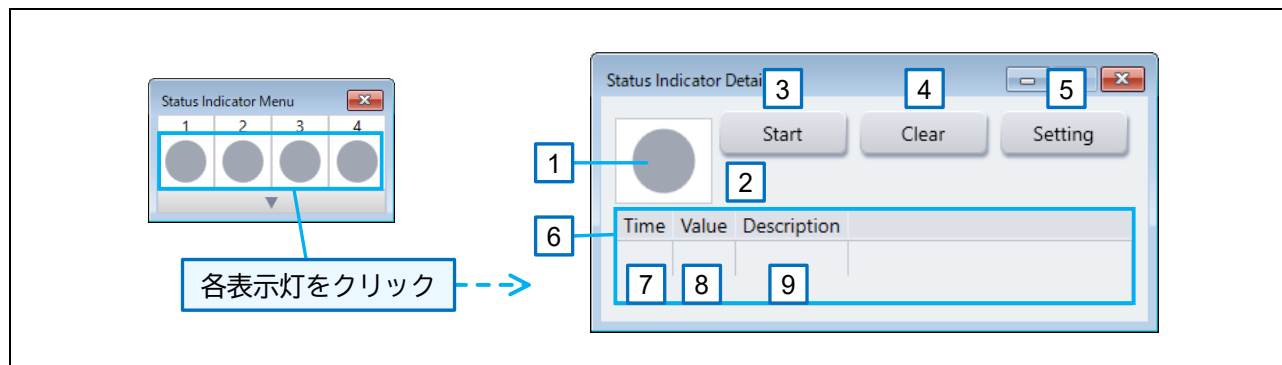


図 8.3 Status Indicator Detail 画面

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」ボタンをクリックすると、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。

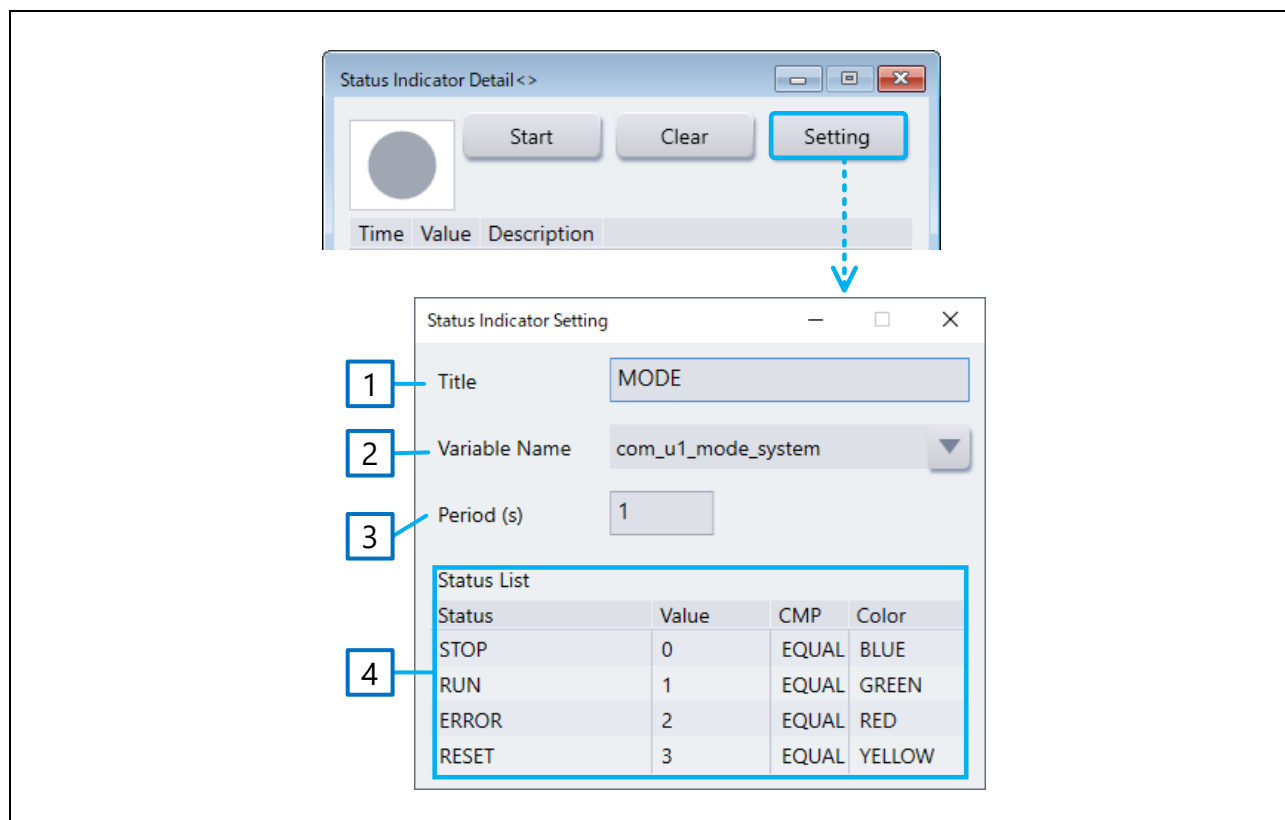


図 8.4 Status Indicator Setting 画面

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。

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。

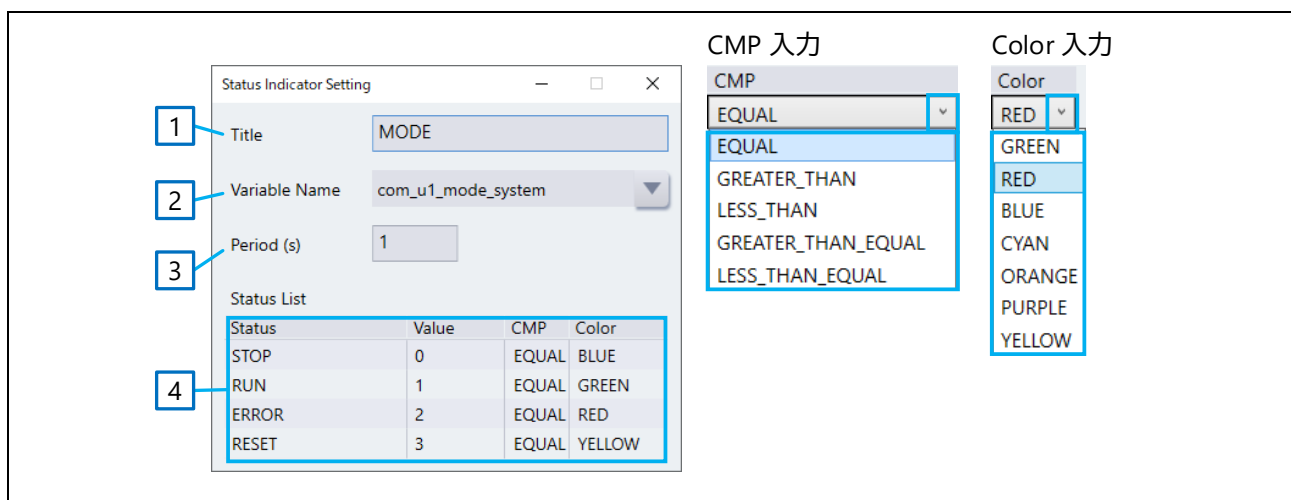


図 8.5 監視設定画面

(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。

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、監視中の状態は保持（継続）されます。

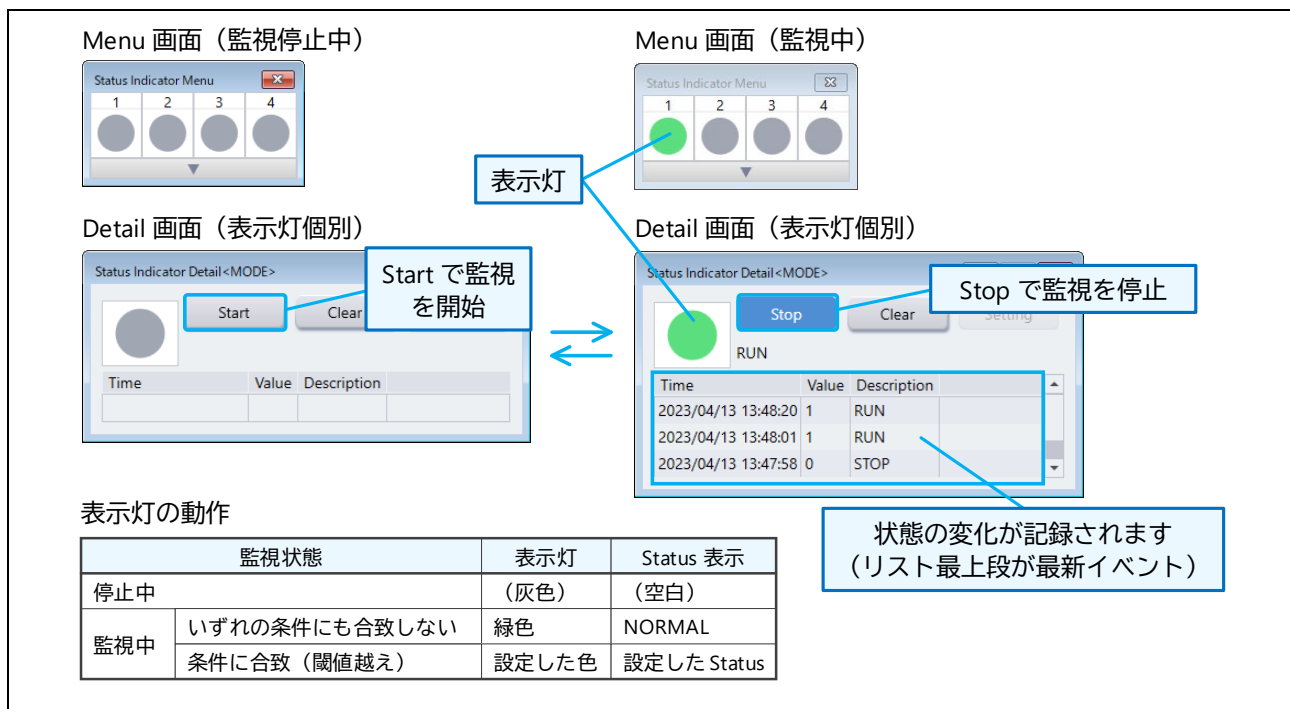


図 8.6 監視の開始／停止

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。

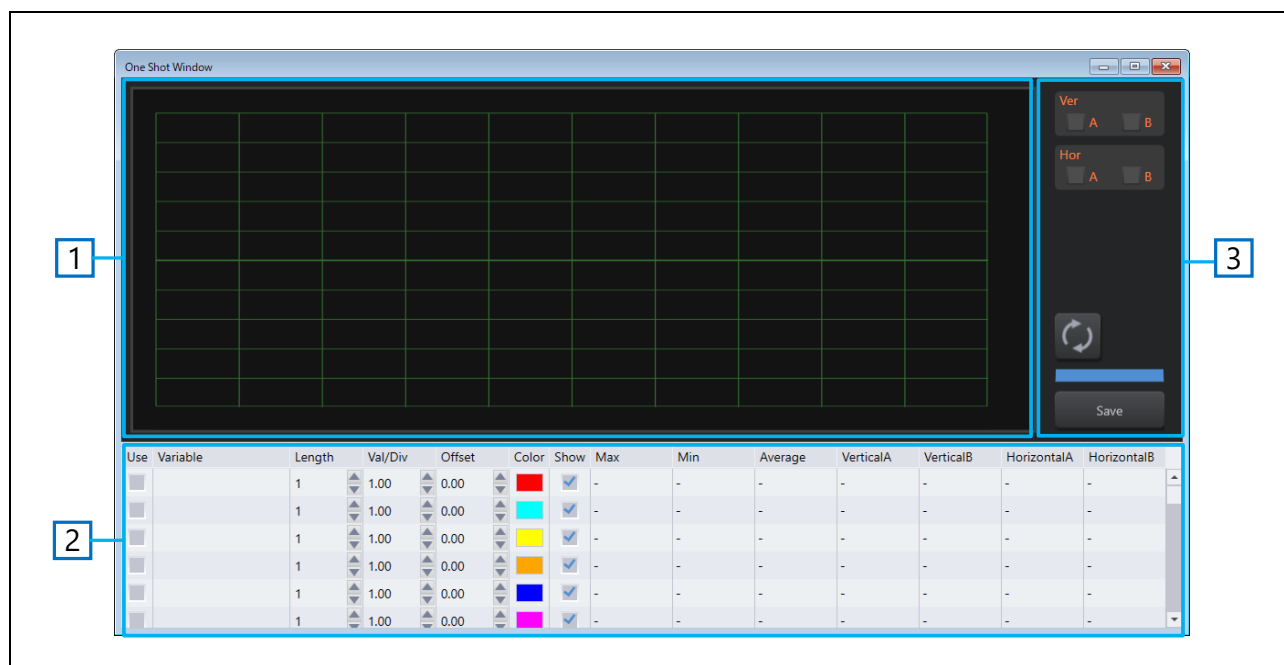


図 9.1 One Shot 画面

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	チャンネル情報部	チャン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。

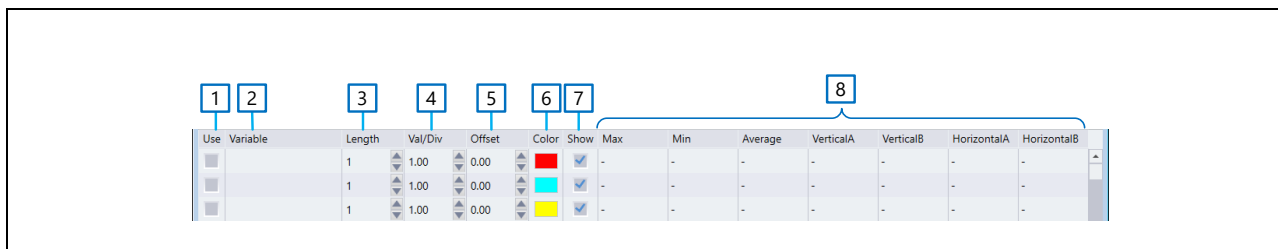


図 9.2 チャネル情報部

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。

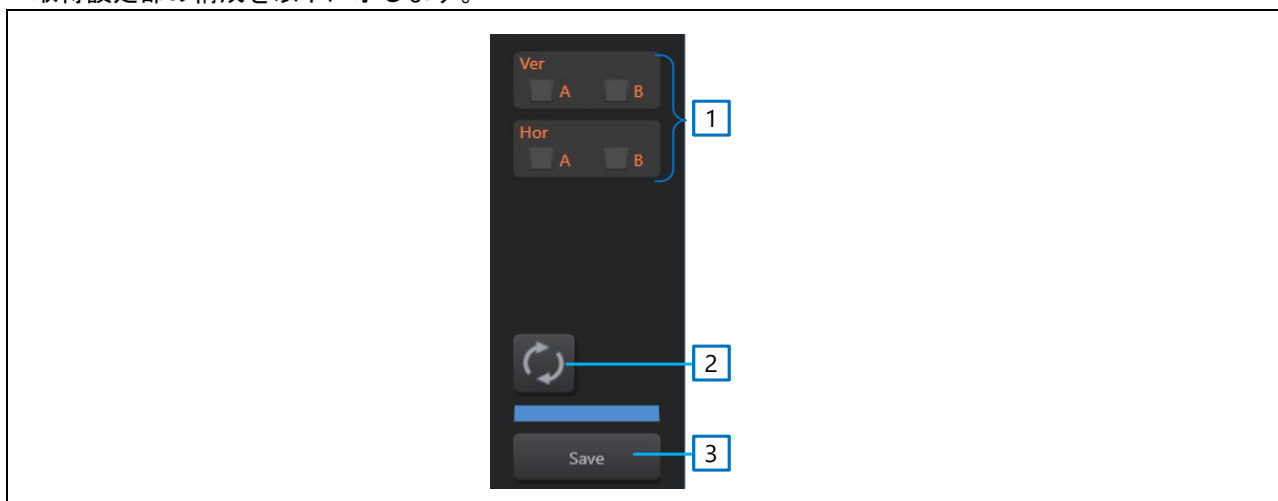


図 9.3 取得設定部

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。

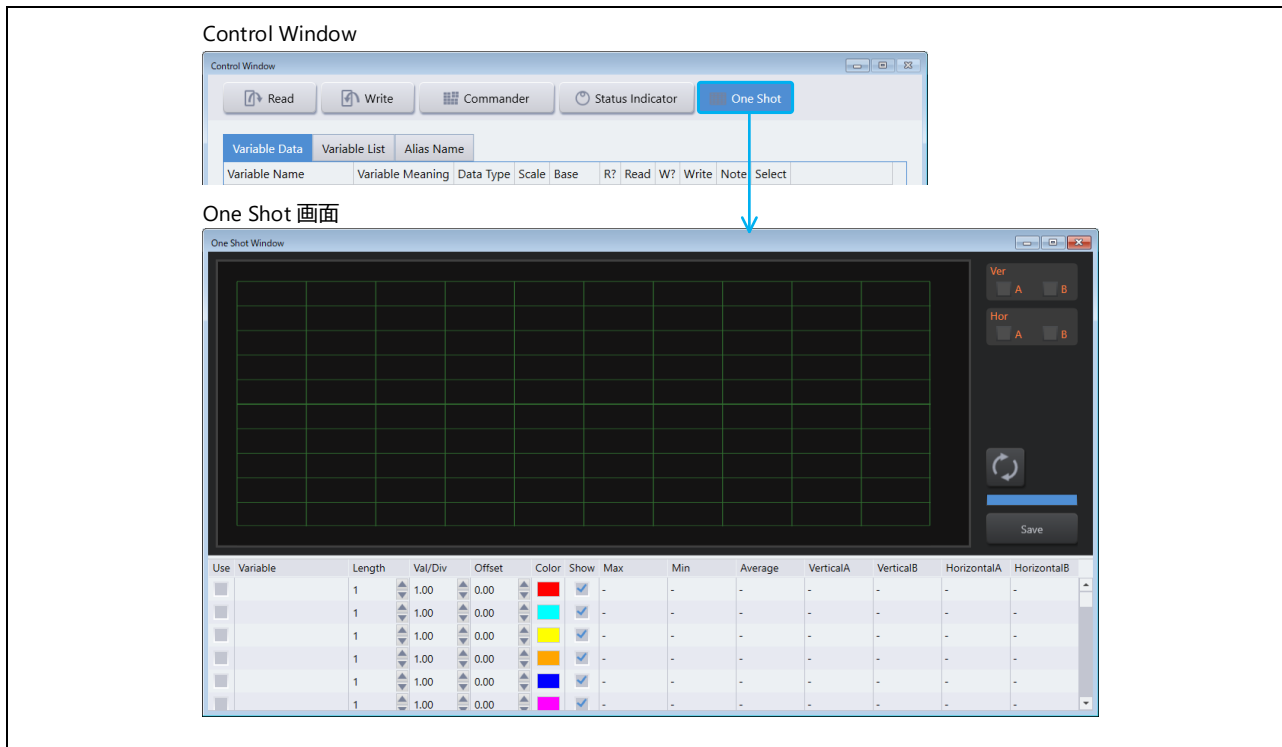


図 9.4 One Shot 画面の起動

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。

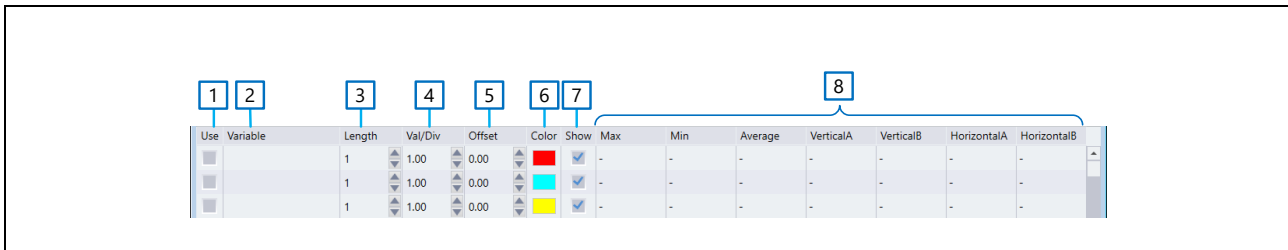


図 9.5 チャネル情報部

(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、チャン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。

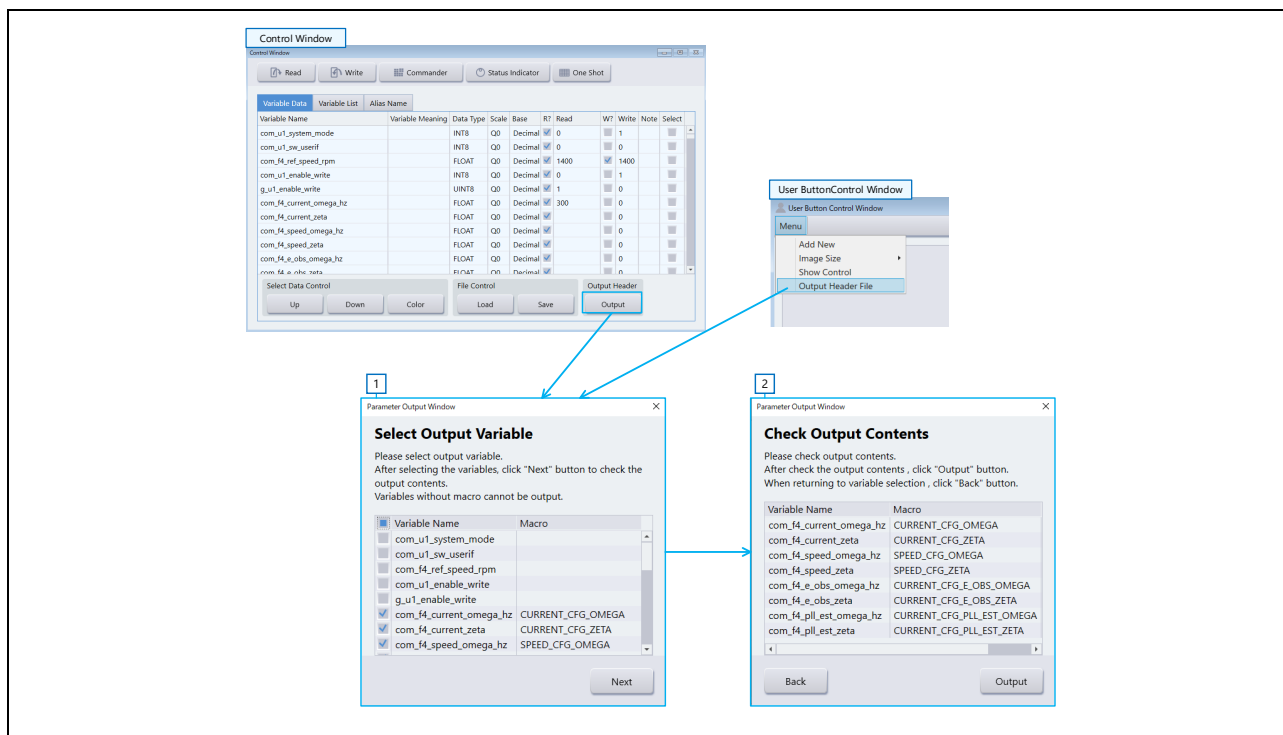


図 10.1 Parameter Output の画面構成

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。

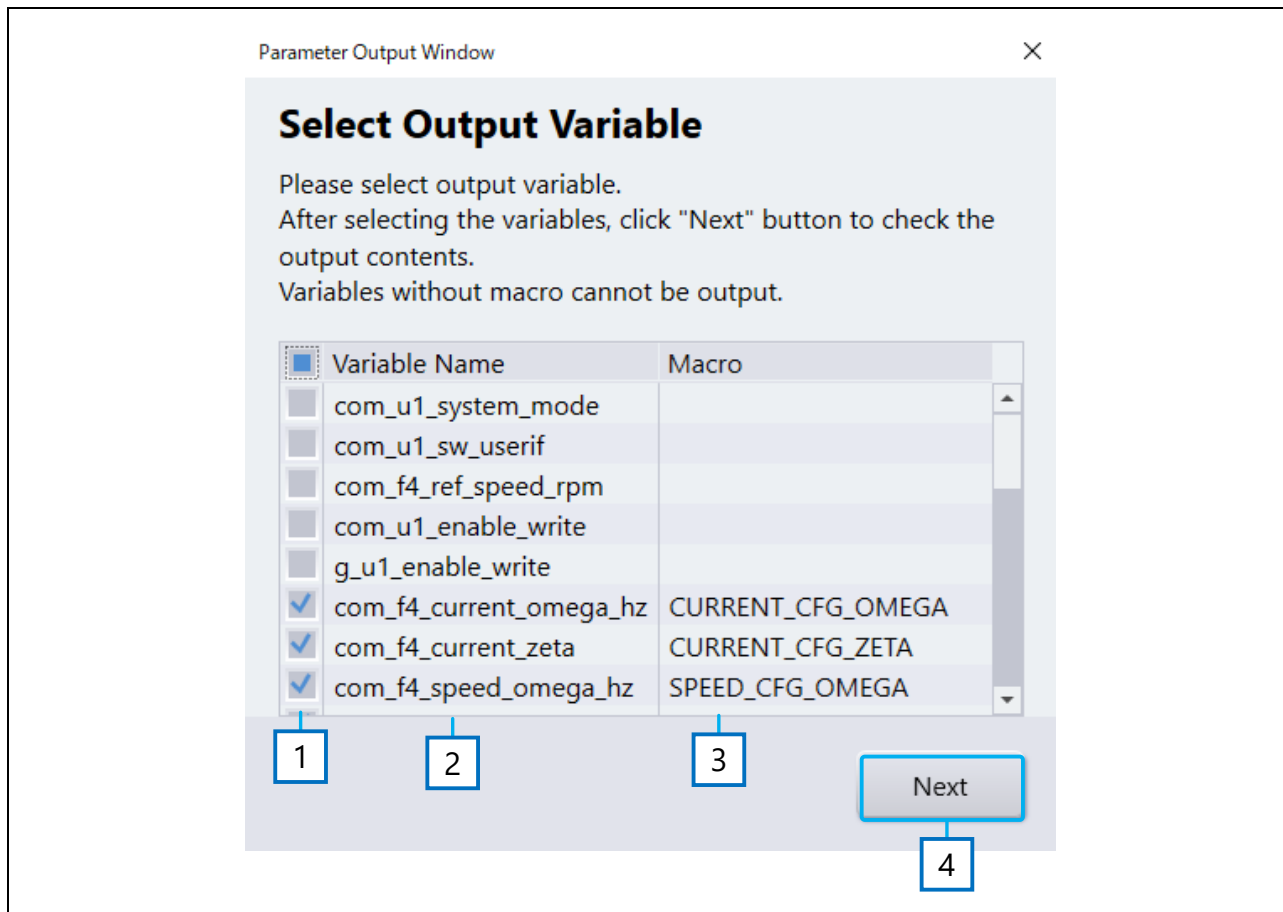


図 10.2 出力変数選択画面

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。

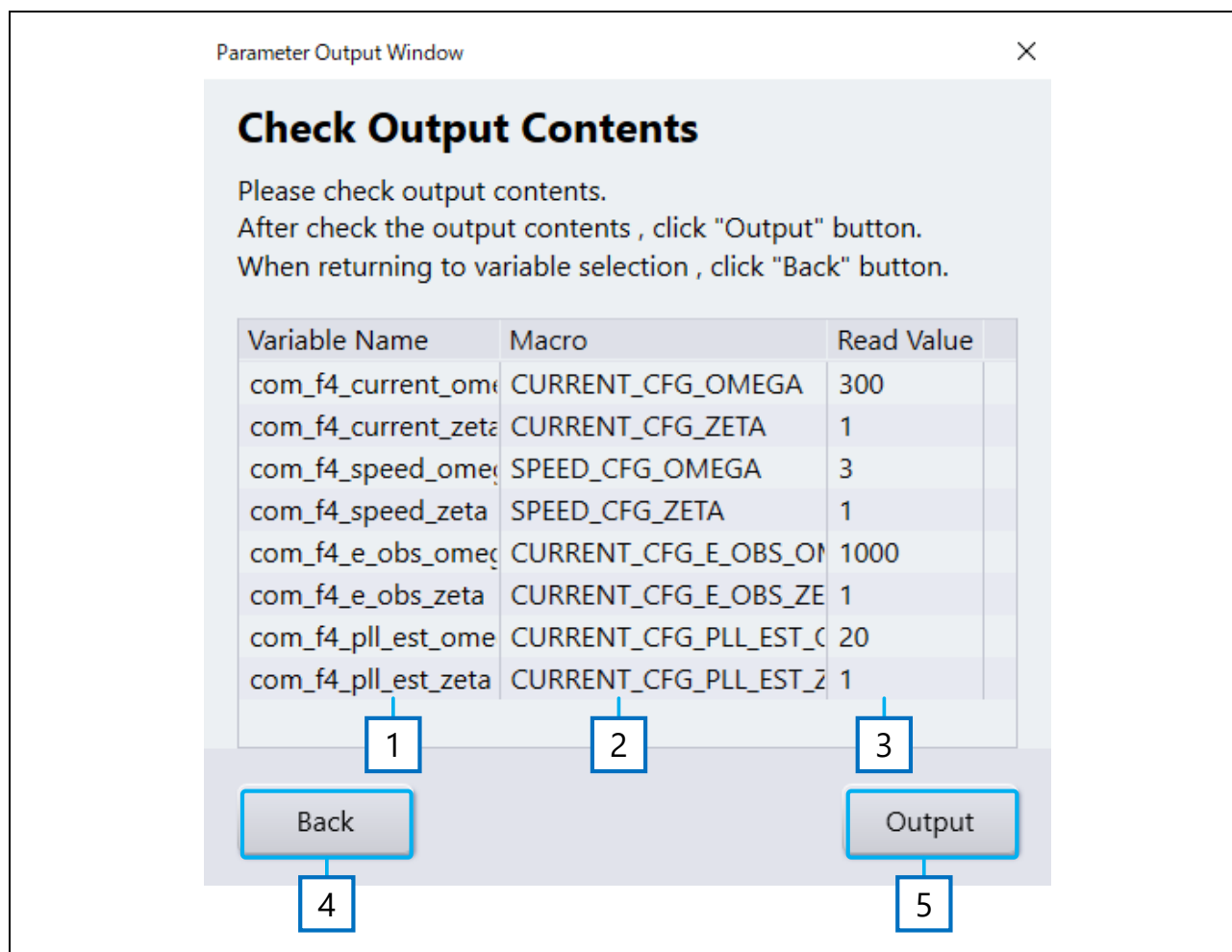


図 10.3 出力内容確認画面

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。

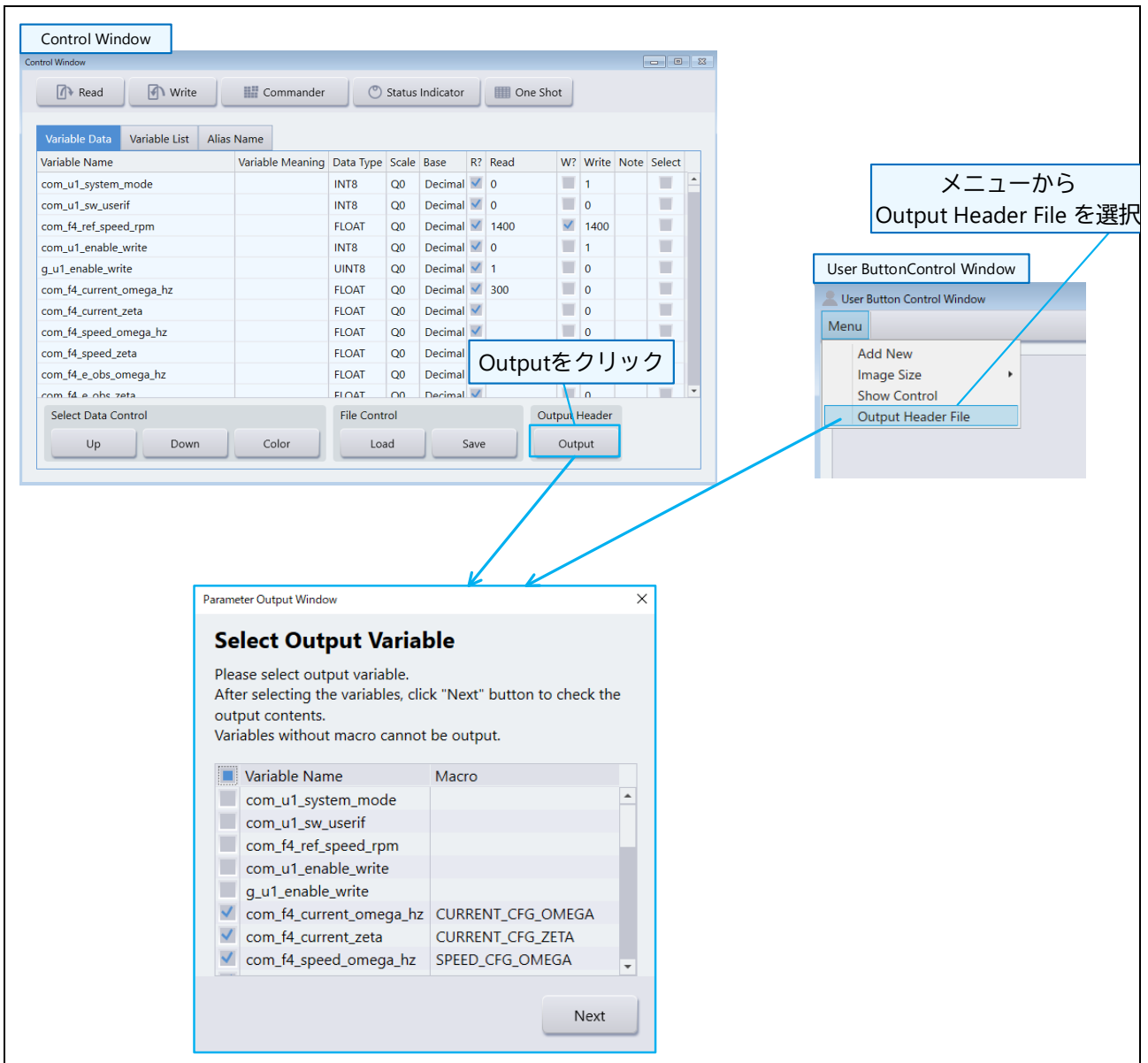


図 10.4 Parameter Output 画面の表示

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を選択します。

また、ヘッダ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。

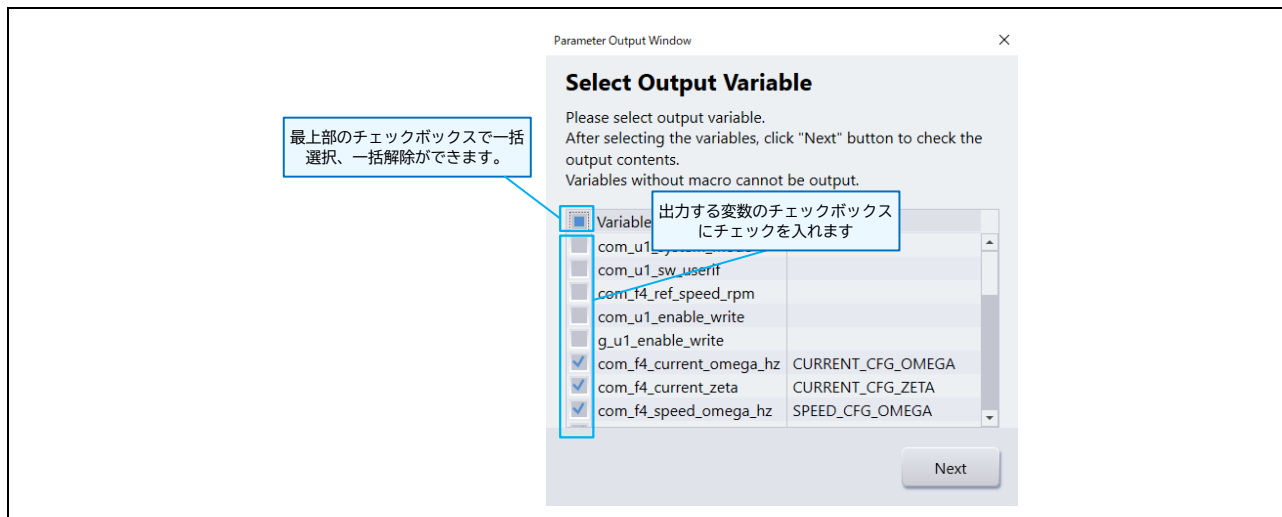


図 10.5 出力する変数の選択

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。

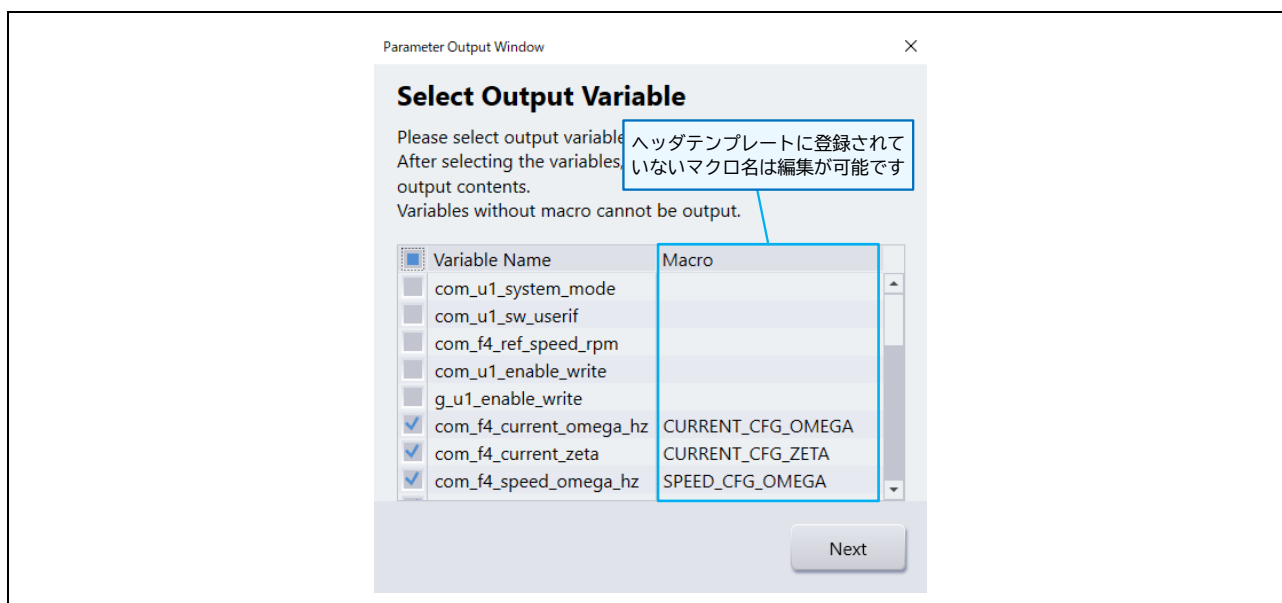


図 10.6 マクロ名の入力

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。

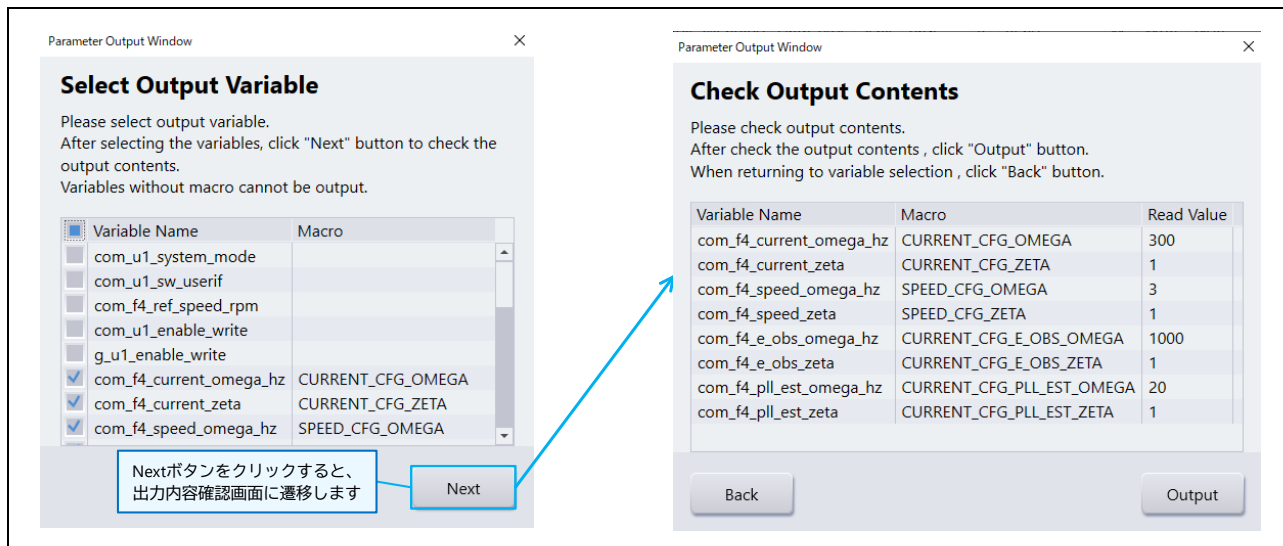


図 10.7 出力内容の確認

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。

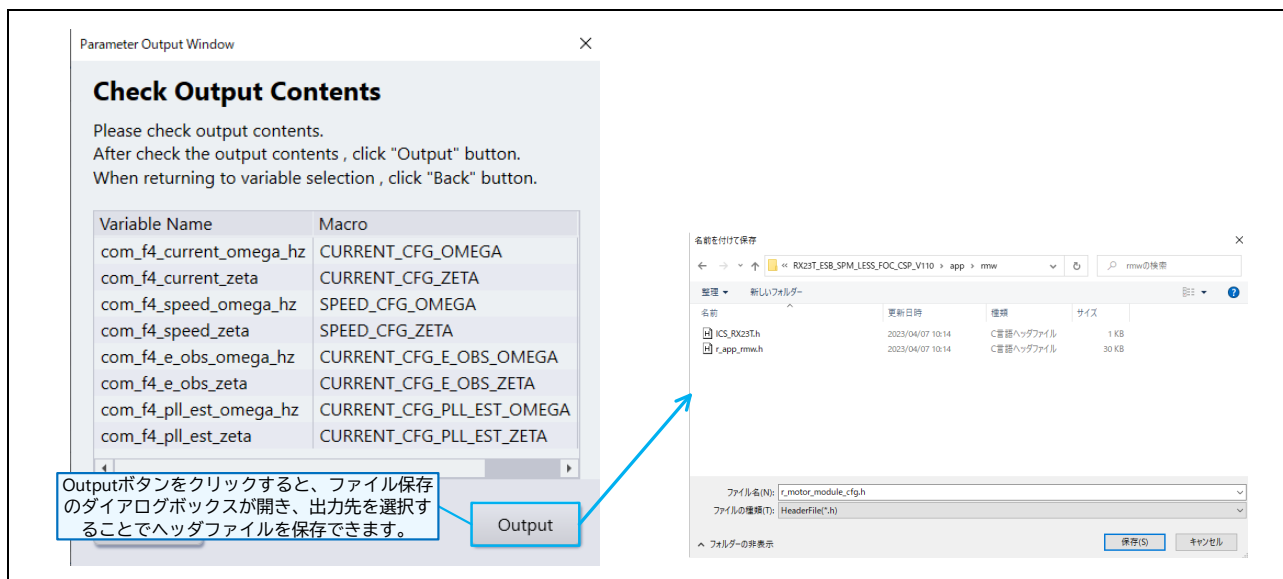


図 10.8 ヘッダファイルの出力

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。

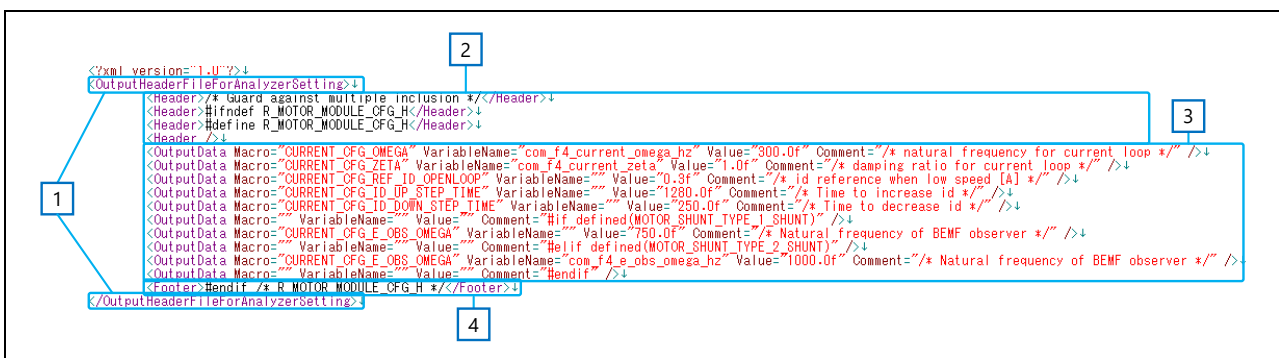


図 10.9 ヘッダテンプレートファイルの構成要素

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。

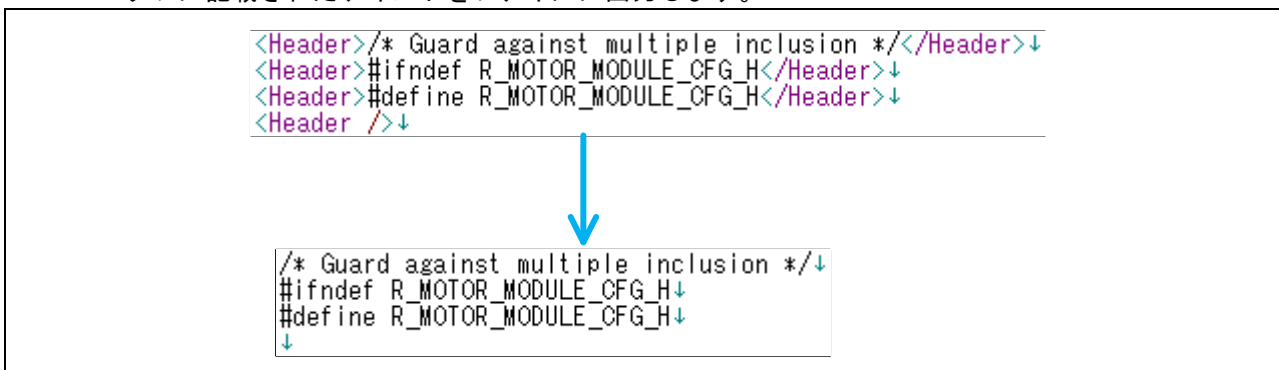


図 10.10 ヘッダ部

10.4.6.3 データ部

OutputData タブは以下の属性で構成します。

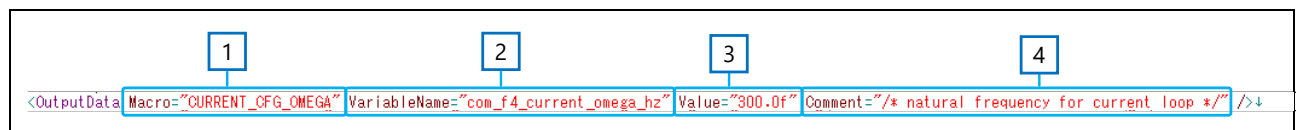


図 10.11 Output タブの属性

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。

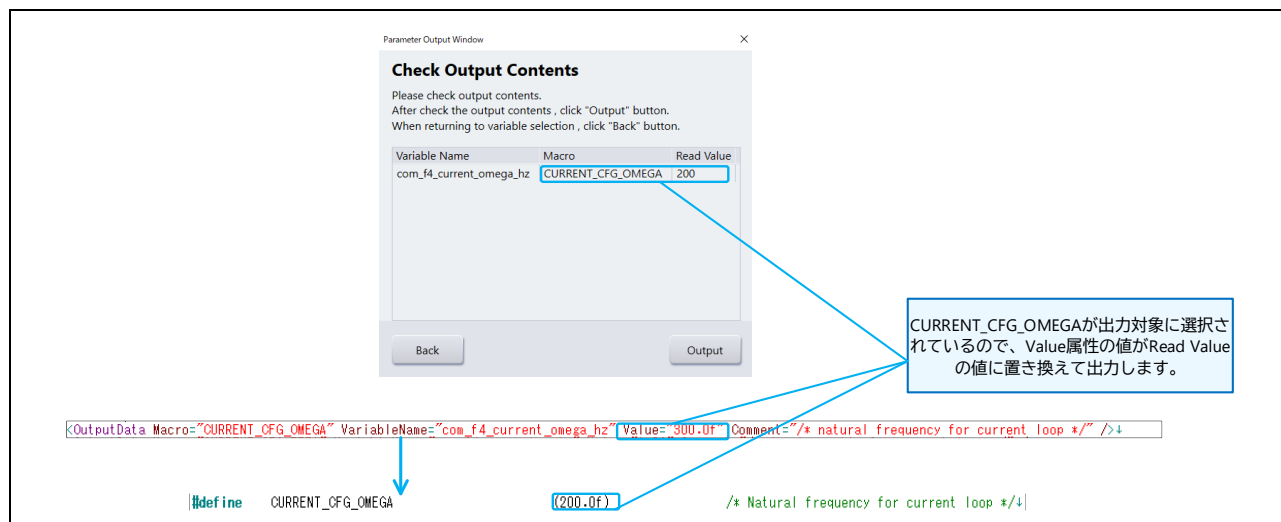


図 10.12 データ部の出力例

- 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。

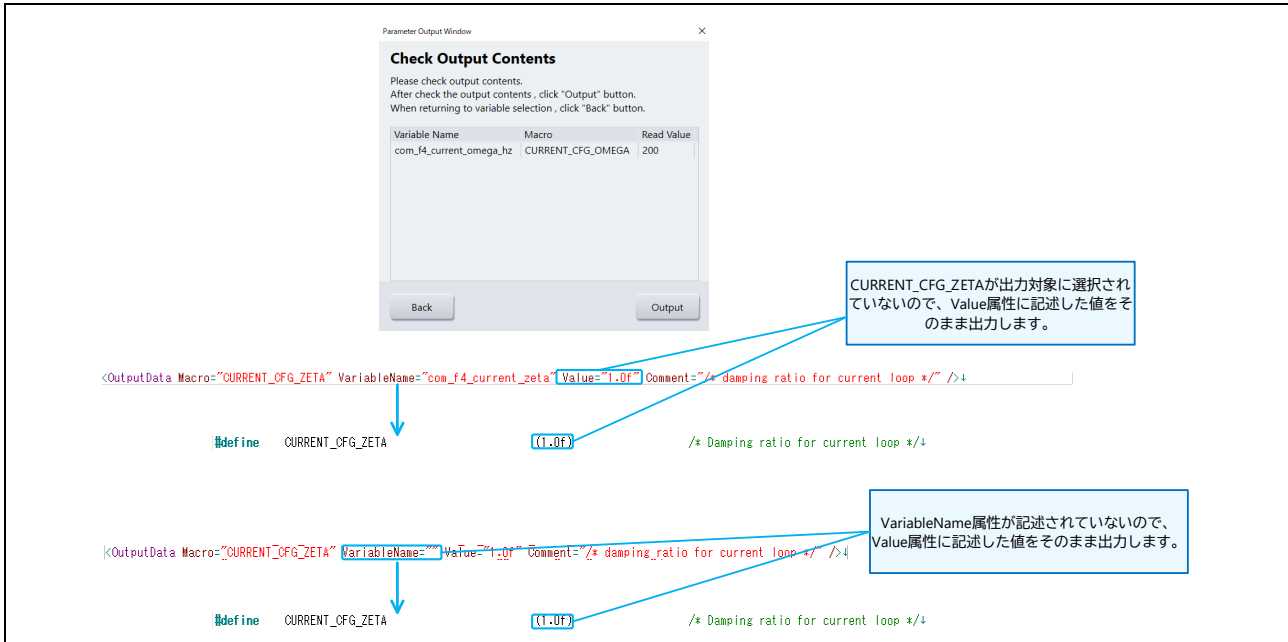


図 10.13 データ部の出力例

- 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。

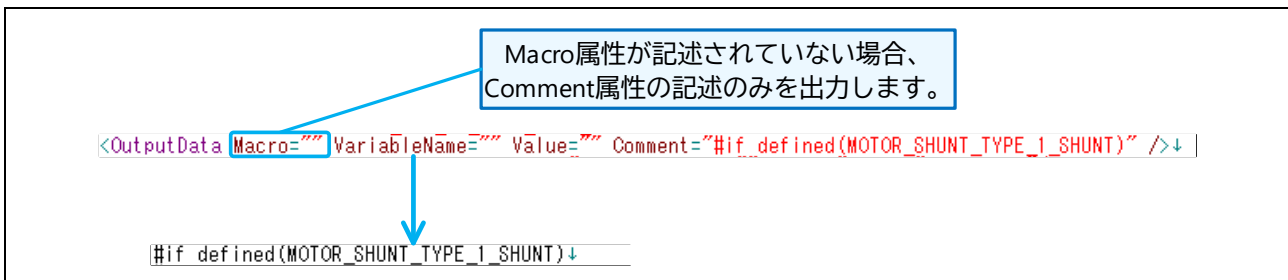


図 10.14 データ部の出力例

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。

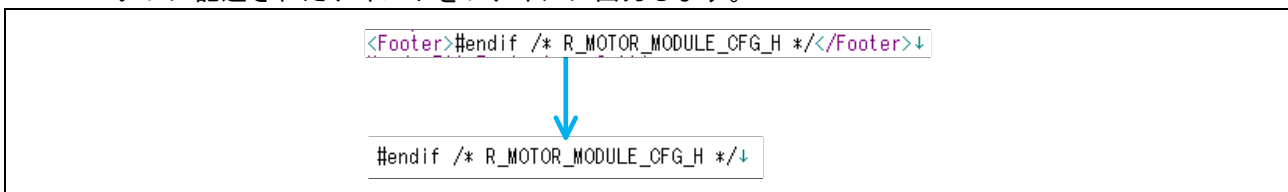


図 10.15 フッタ部

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。

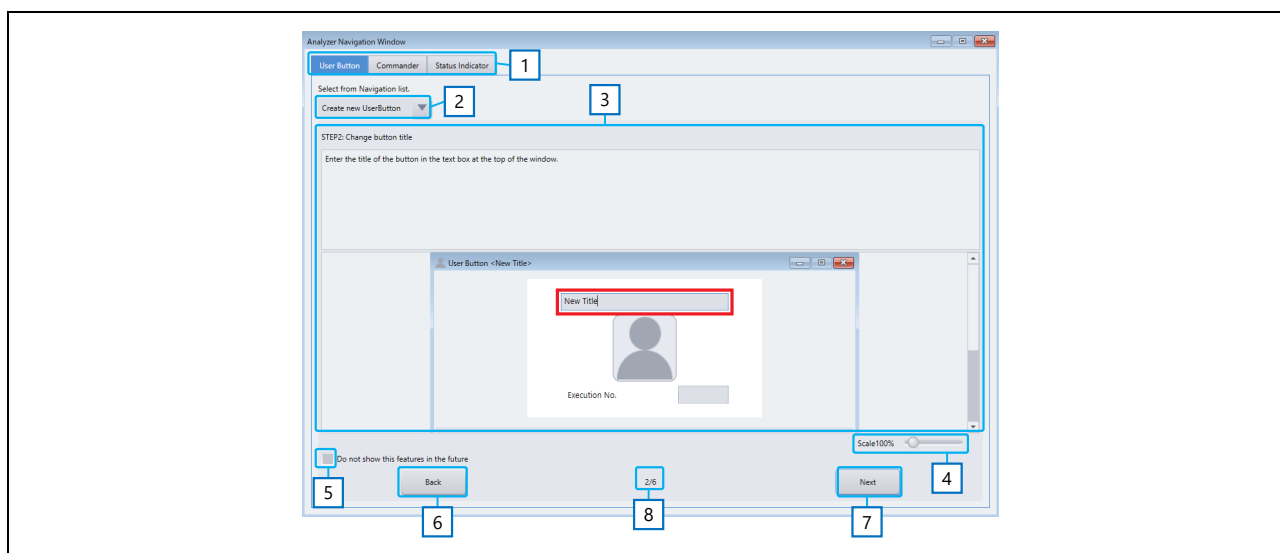


図 11.1 Navigation 画面

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。

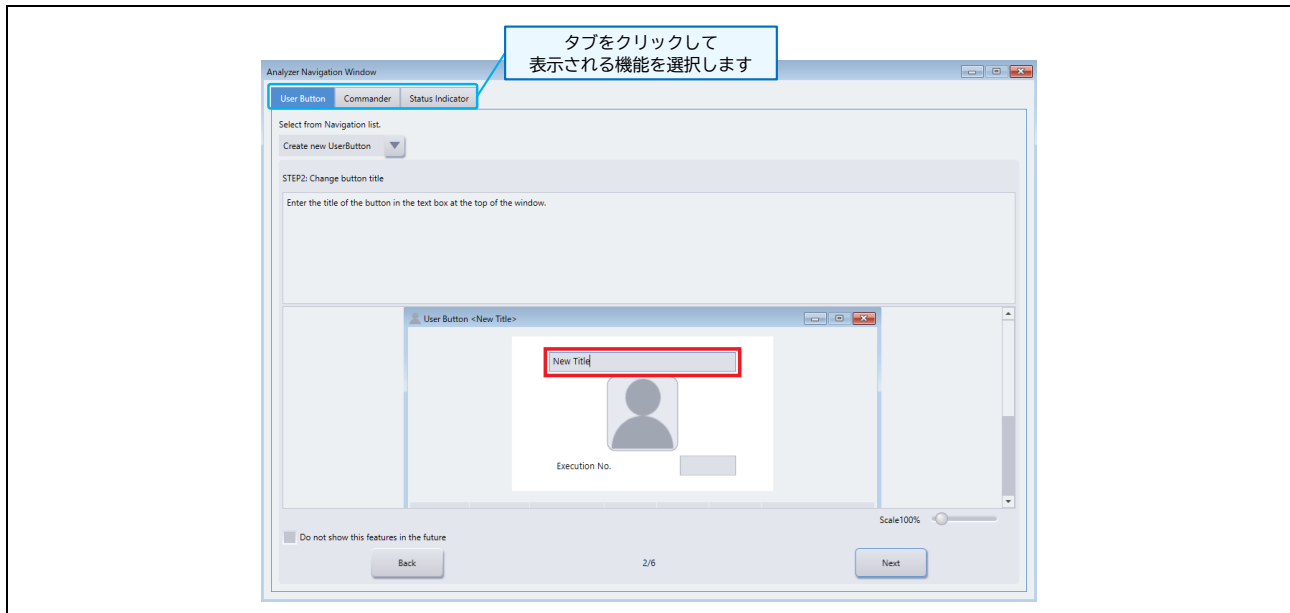


図 11.2 表示機能の切り替え

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。

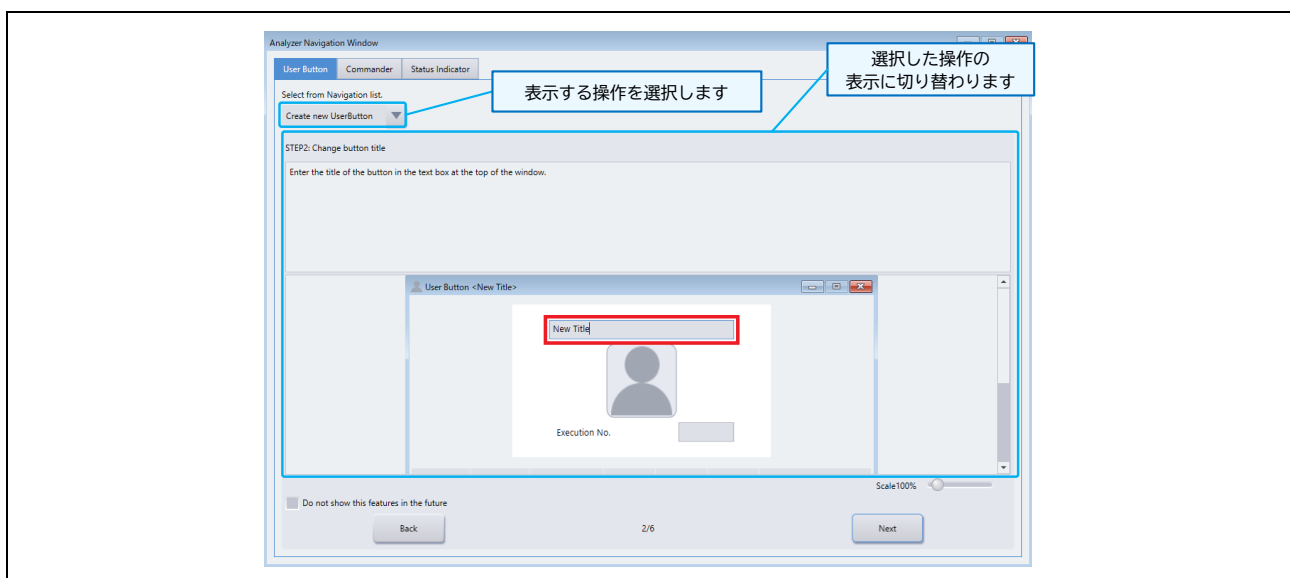


図 11.3 操作手順の選択

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。

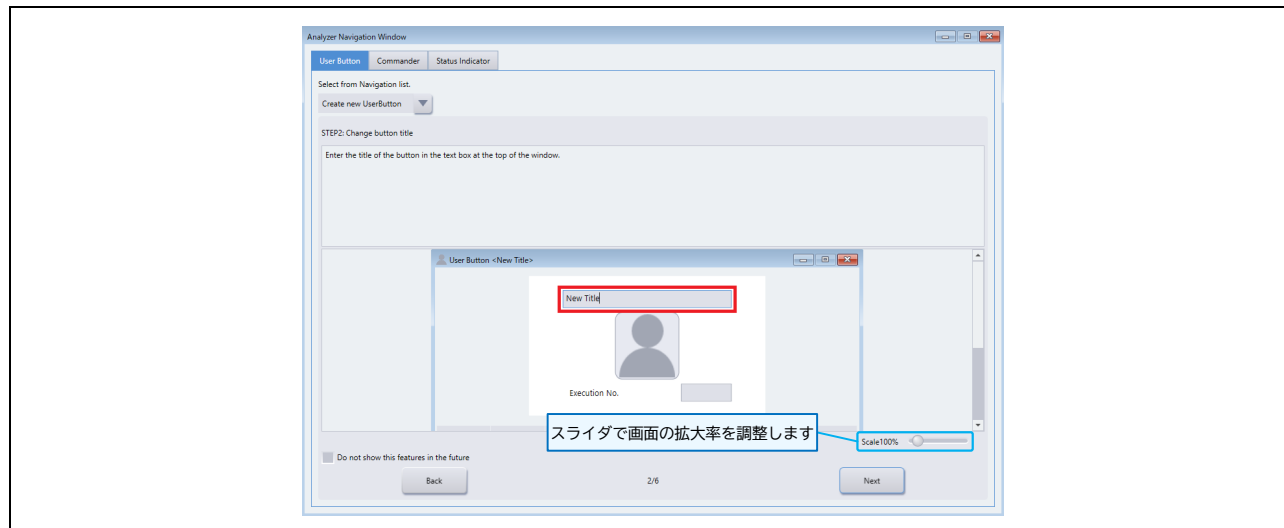


図 11.4 画像拡大率の変更

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。

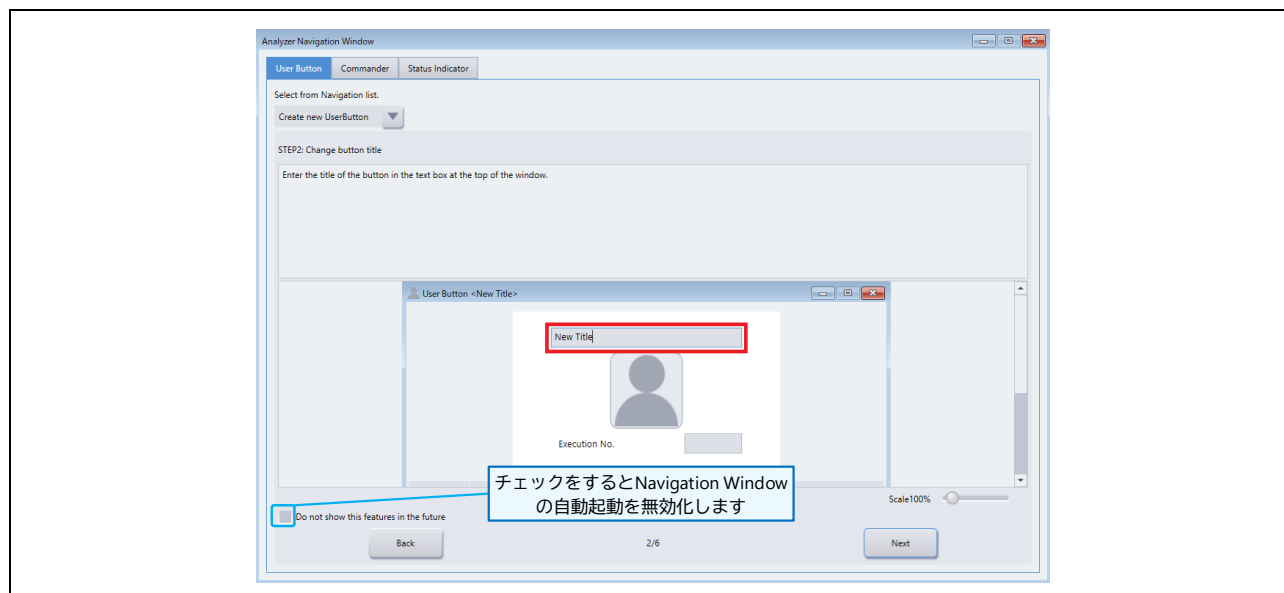


図 11.5 Navigation 自動起動の変更

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。

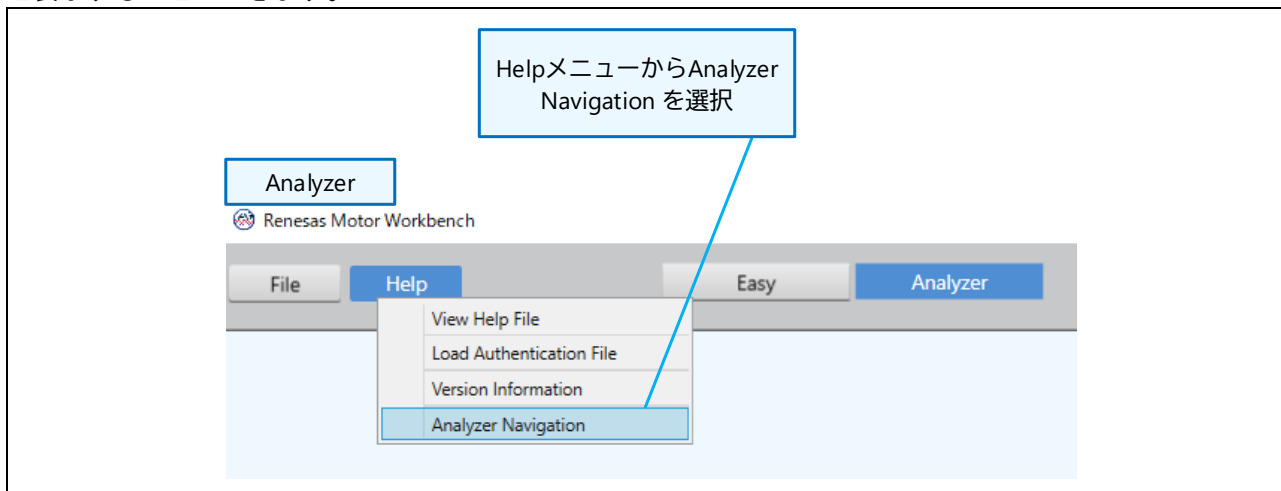


図 11.6 Analyzer Navigation の起動

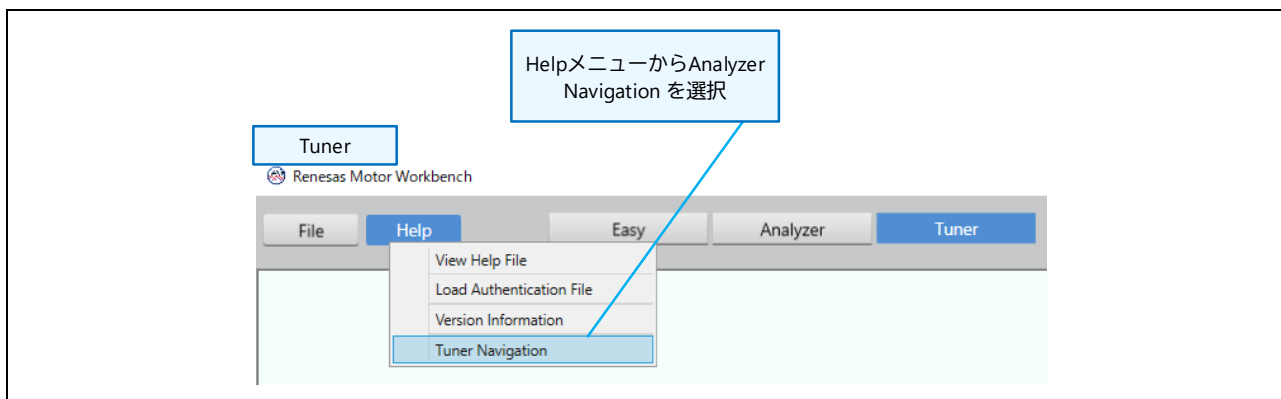


図 11.7 Tuner Navigation の起動

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。

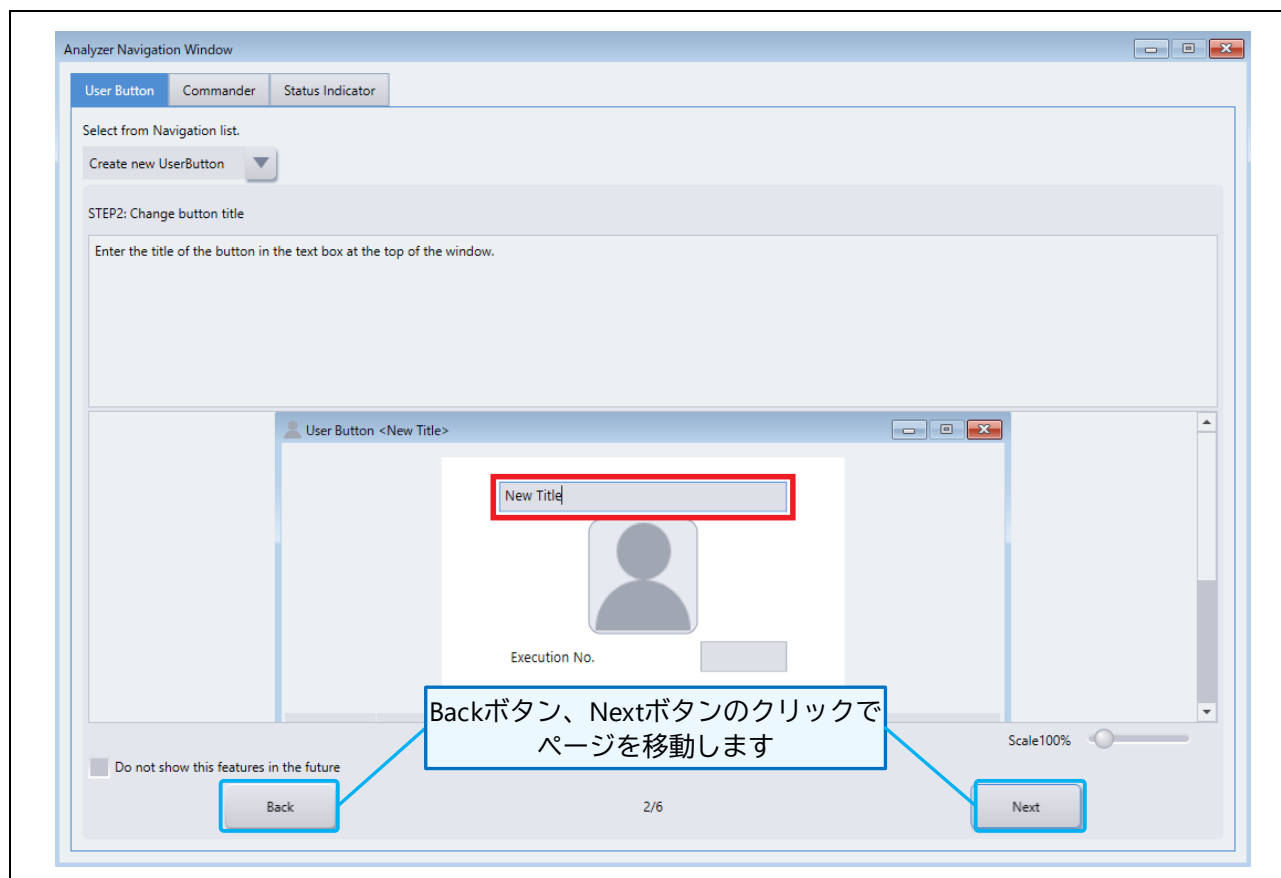


図 11.8 ページの移動

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、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。

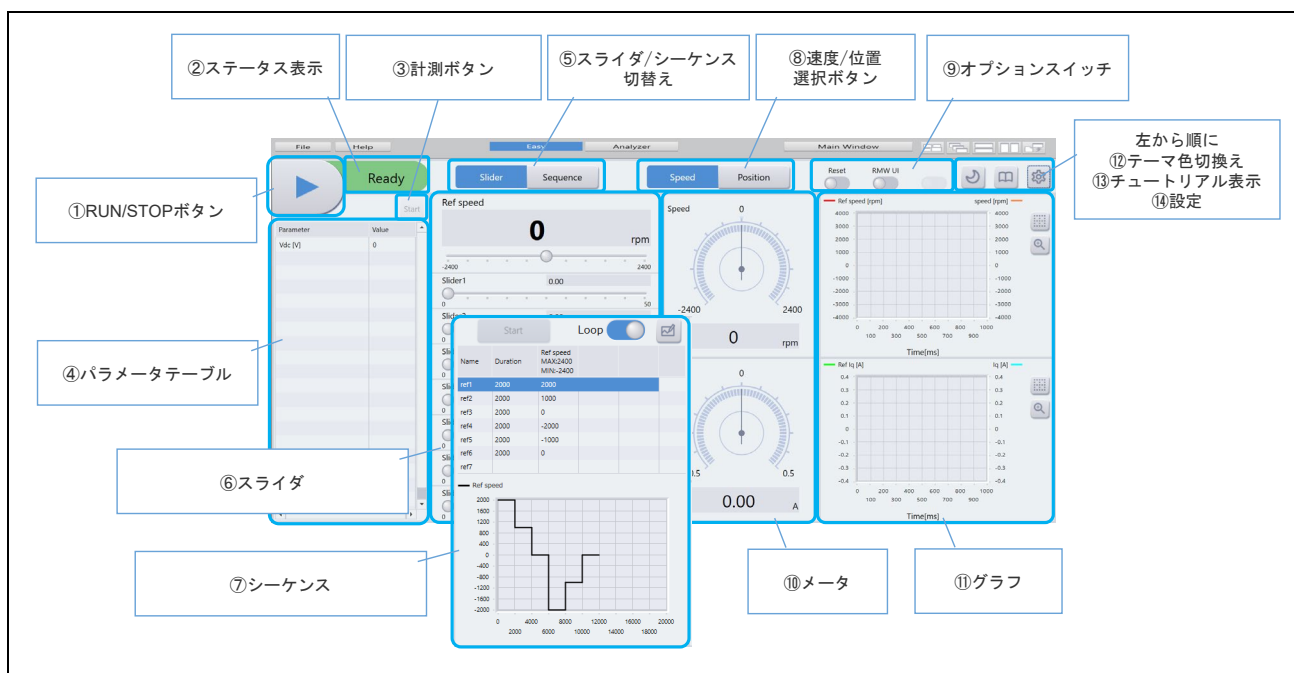


図 12.2 Easy GUI 画面の機能

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	スライダー/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	スライダー	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。

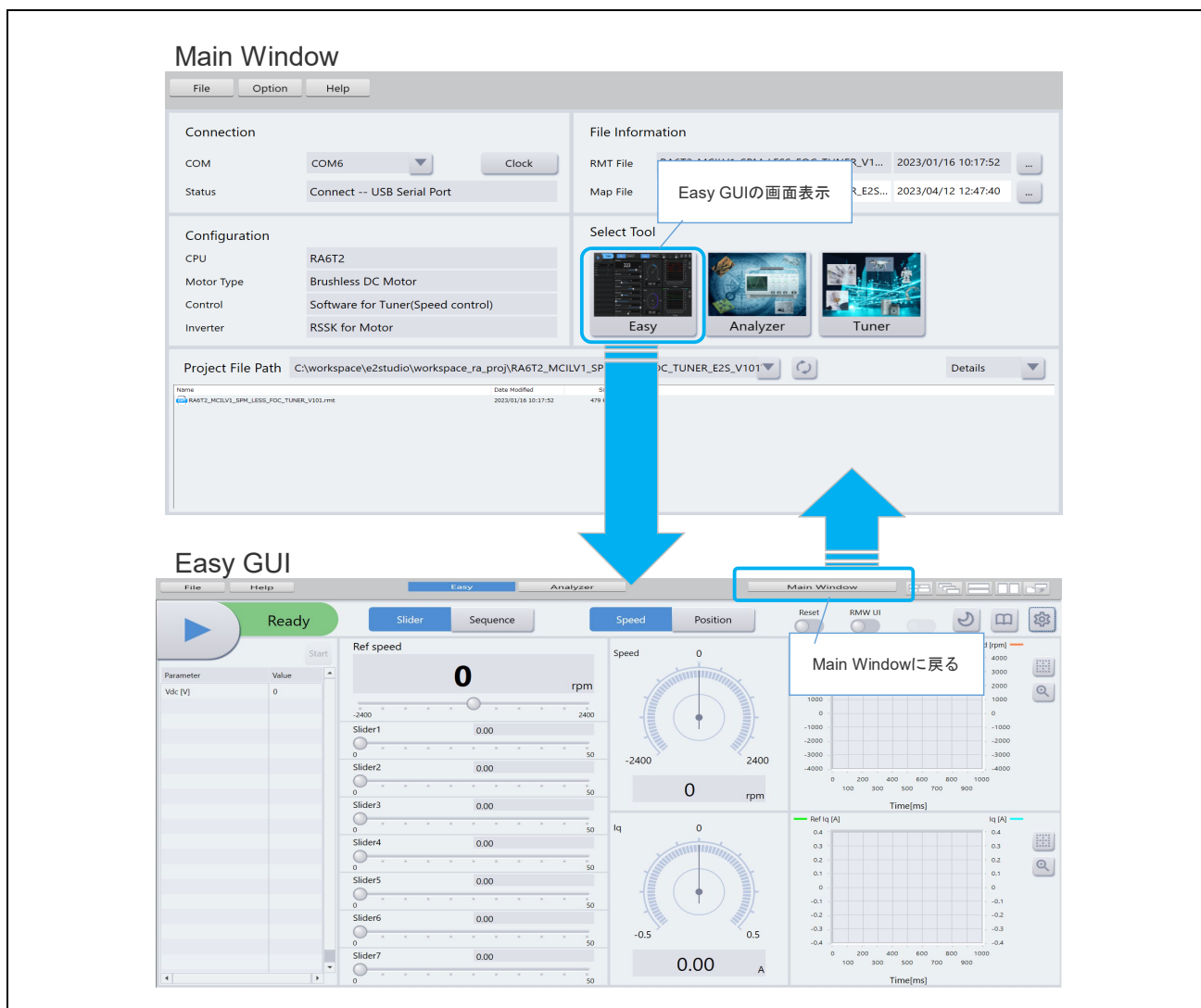


図 12.3 Easy GUI 画面の表示

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

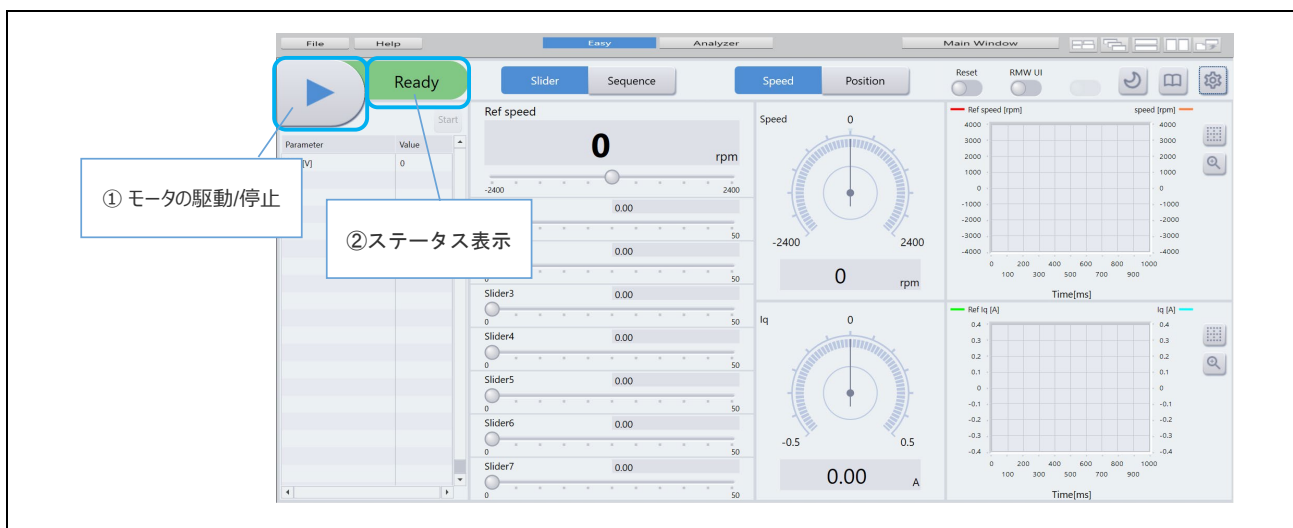


図 12.4 基本機能

12.5.2 スライダー機能

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

- ① スライダー画面表示 : スライダー/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 スライダーを操作 : スライダーを動かし変数の値を変更します
スライダー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

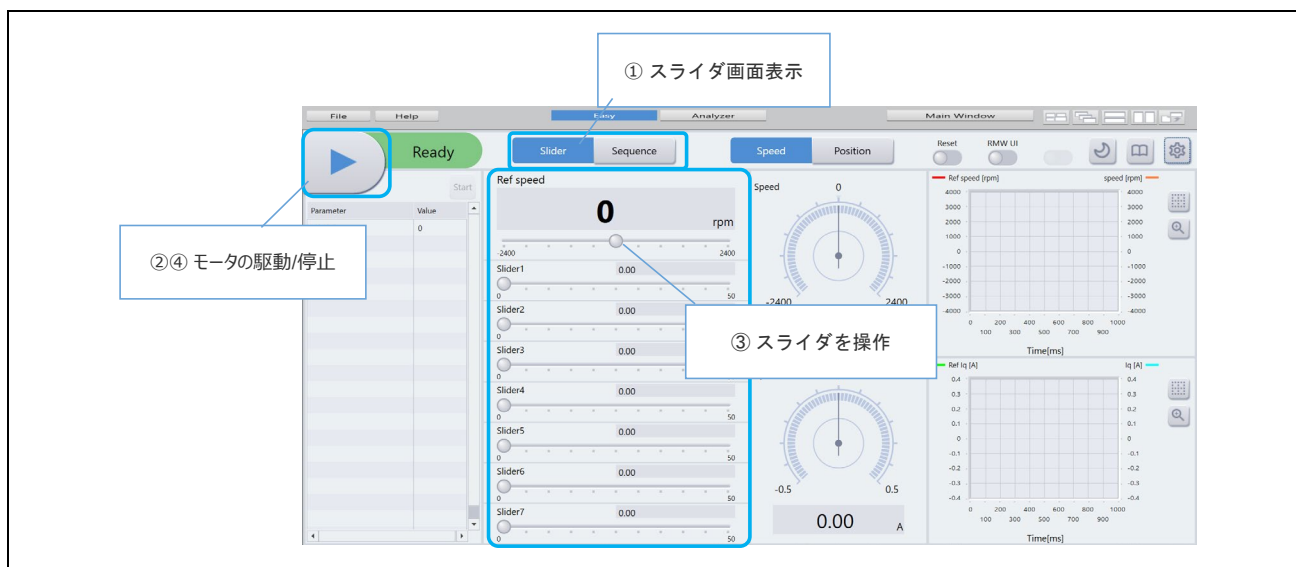


図 12.5 スライダー機能

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 : スライダー/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 : スライダー/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

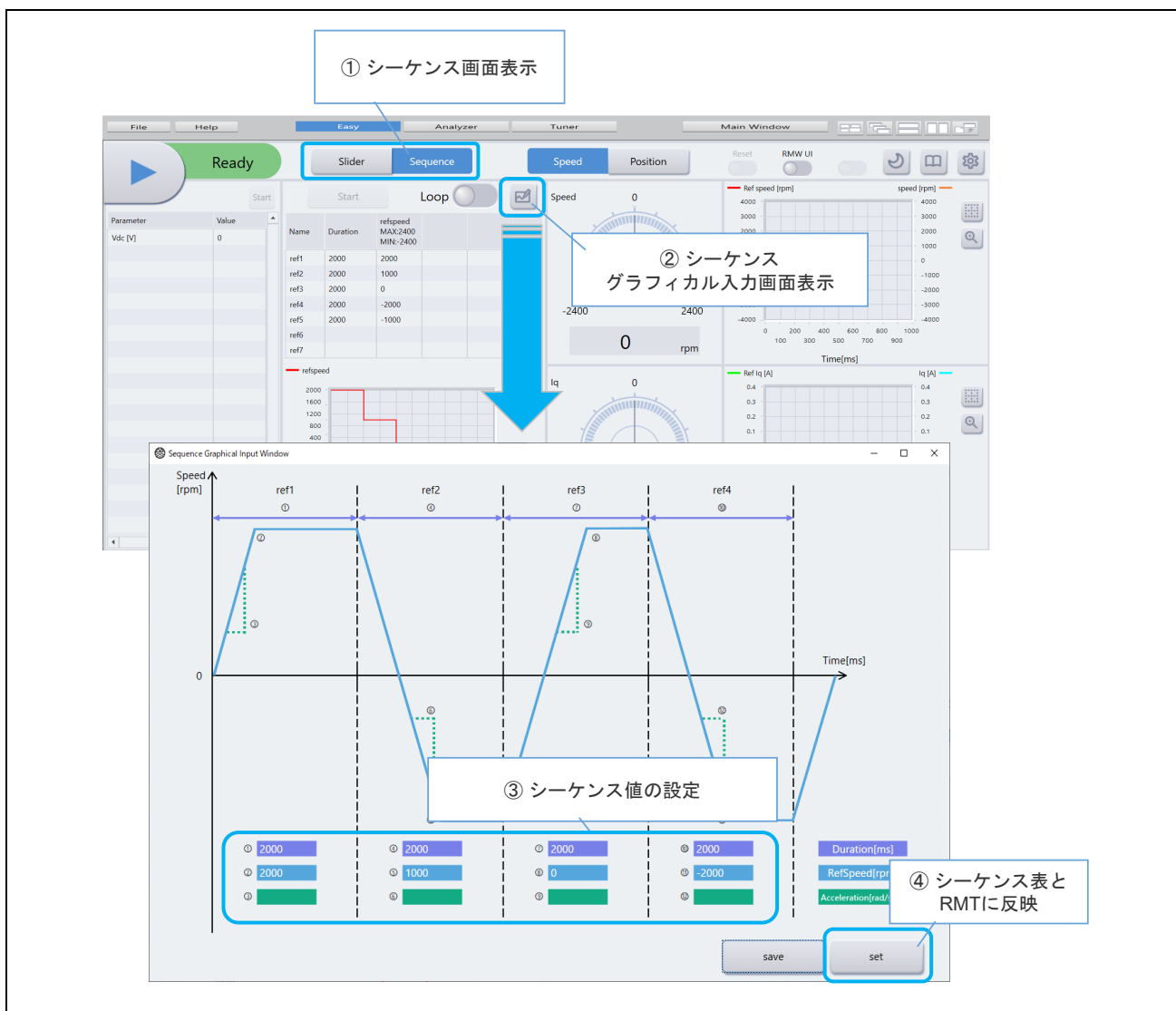


図 12.7 シーケンスグラフィカル機能

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 : スライダー/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

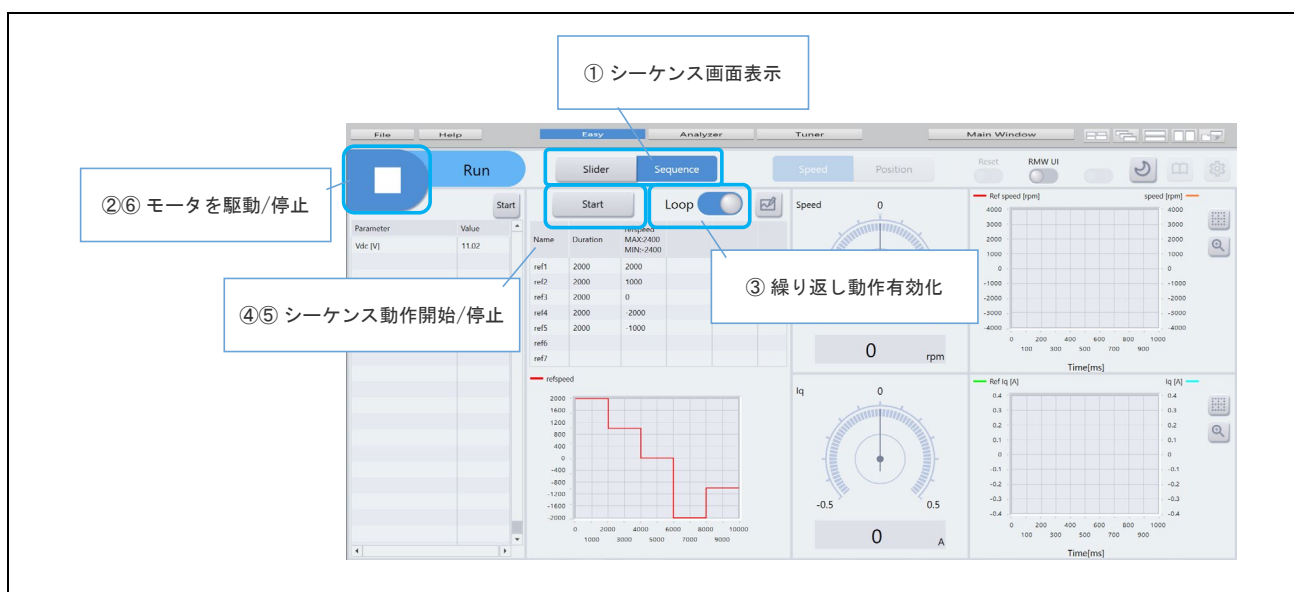


図 12.8 シーケンスの実行

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。

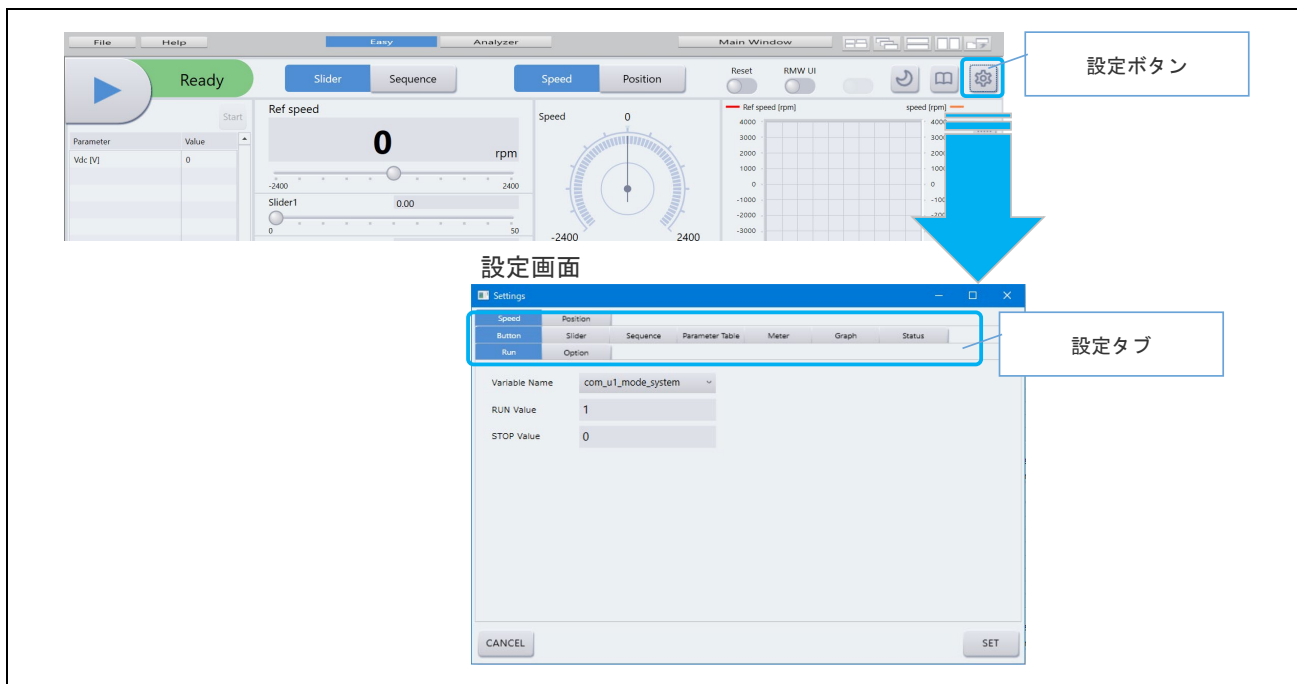


図 12.10 設定画面の表示

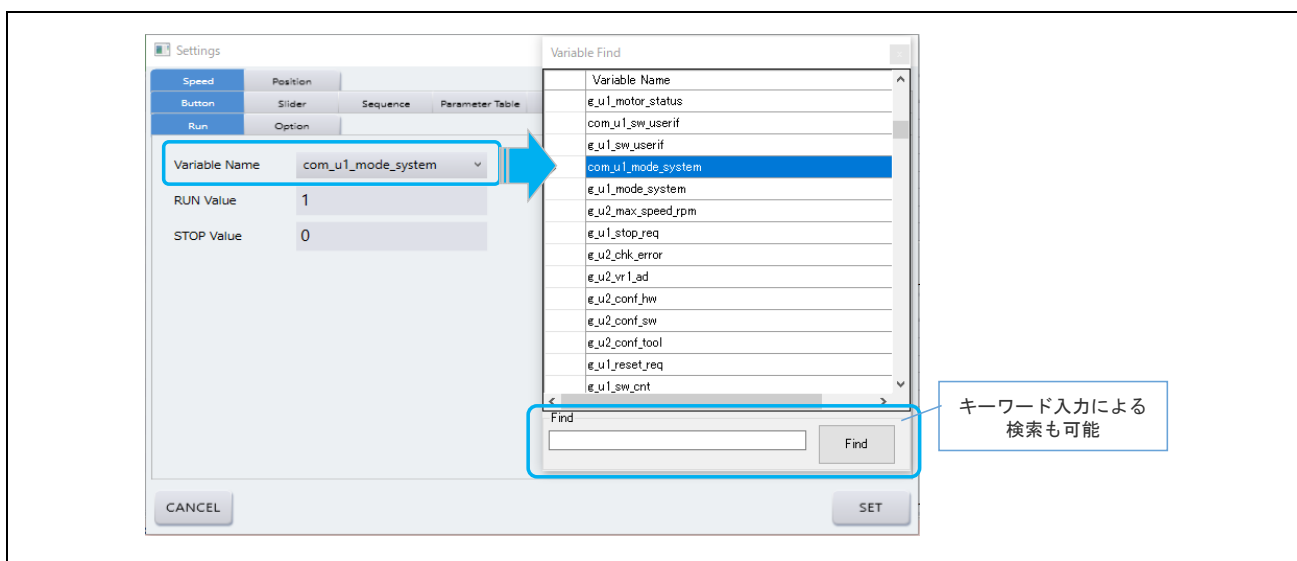




図 12.11 変数名の設定例

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		<ul style="list-style-type: none"> 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

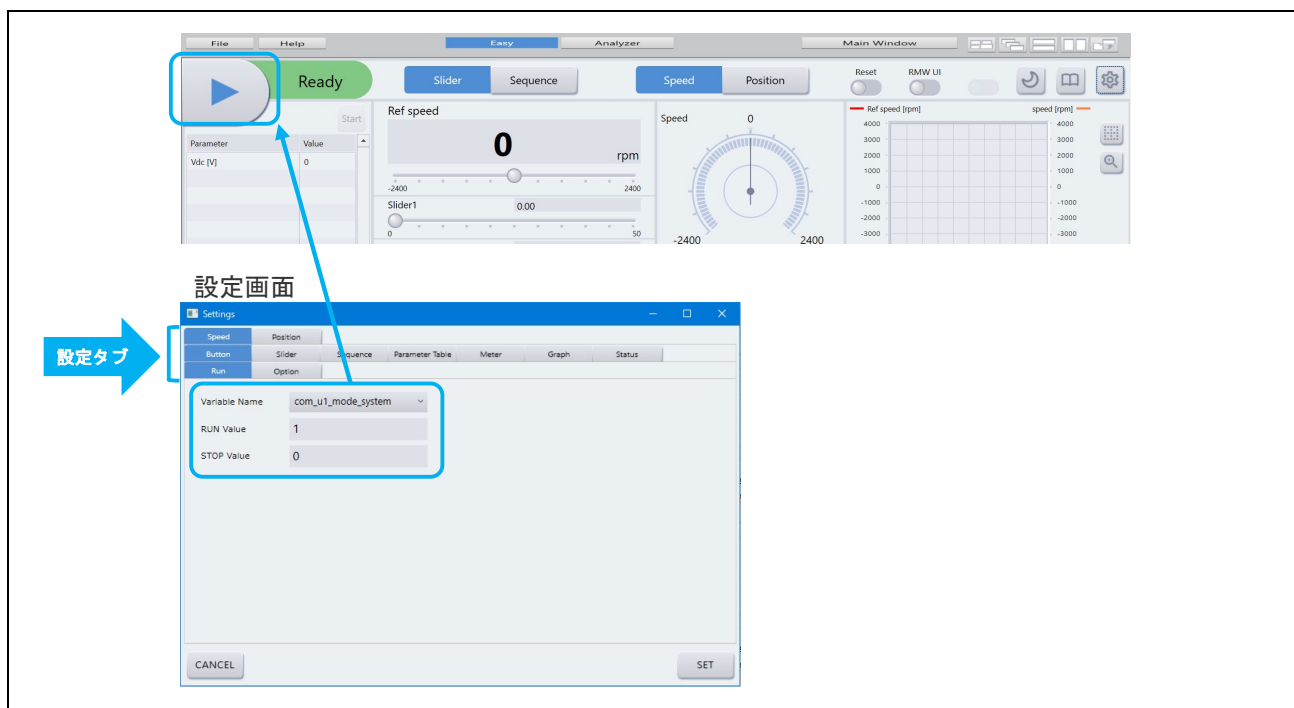


図 12.12 RUN/STOP ボタン設定例

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		<ul style="list-style-type: none"> 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

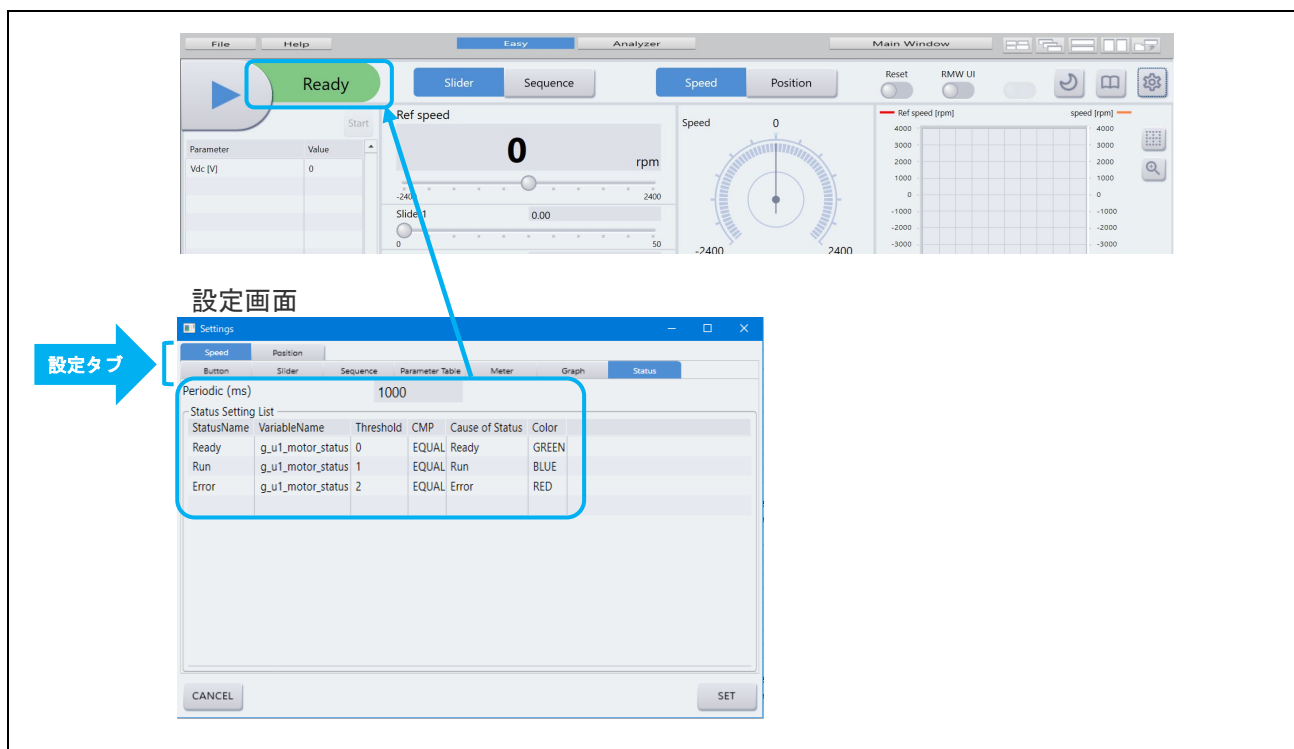


図 12.13 ステータス表示設定例

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		<ul style="list-style-type: none"> 「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

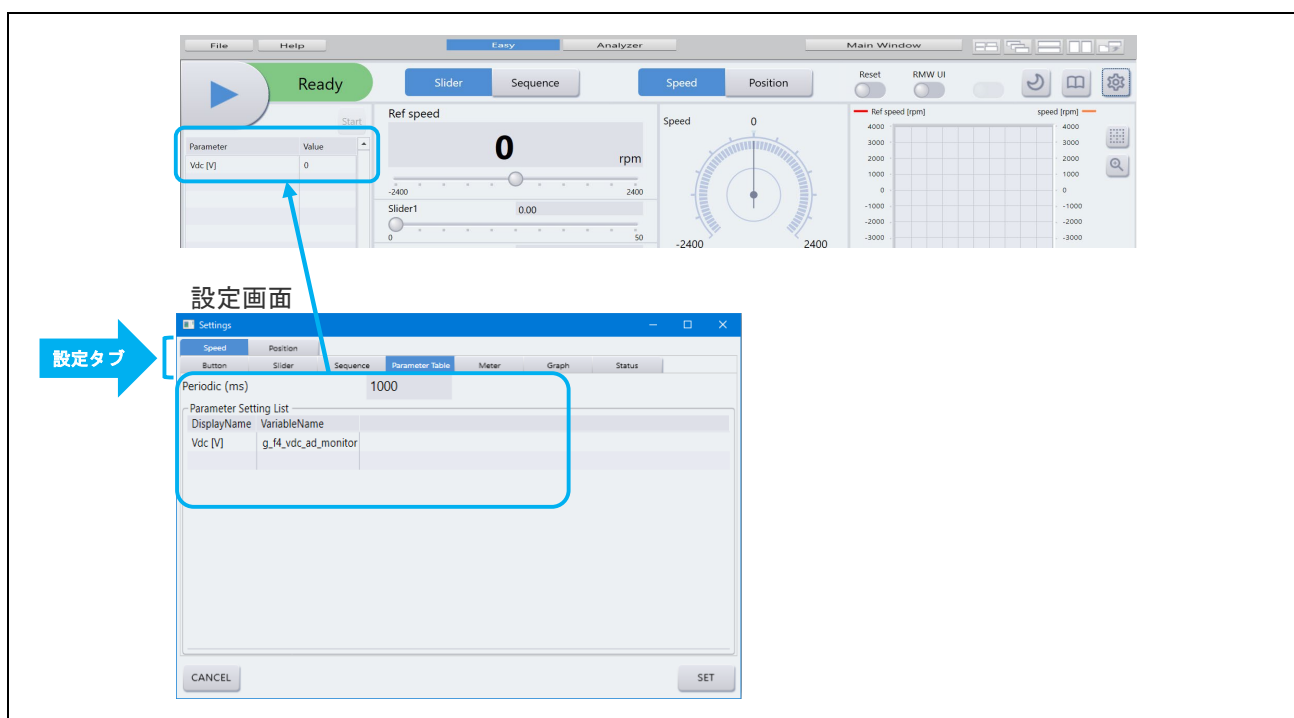


図 12.14 パラメータテーブル設定例

12.6.4 スライダー

スライダー操作した値を変数に書き込みます。最大 8 個のスライダー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、スライダー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。

スライダーに関連付ける変数の設定例は以下のとおりです。

表 12.5 スライダー設定例

項目		内容
設定タブ	Speed 選択時	Speed → Slider → Slider1~8
	Position 選択時	Position → Slider → Slider1~8
スライダー		<ul style="list-style-type: none"> 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 100ms 周期でスライダーの値を変数に反映

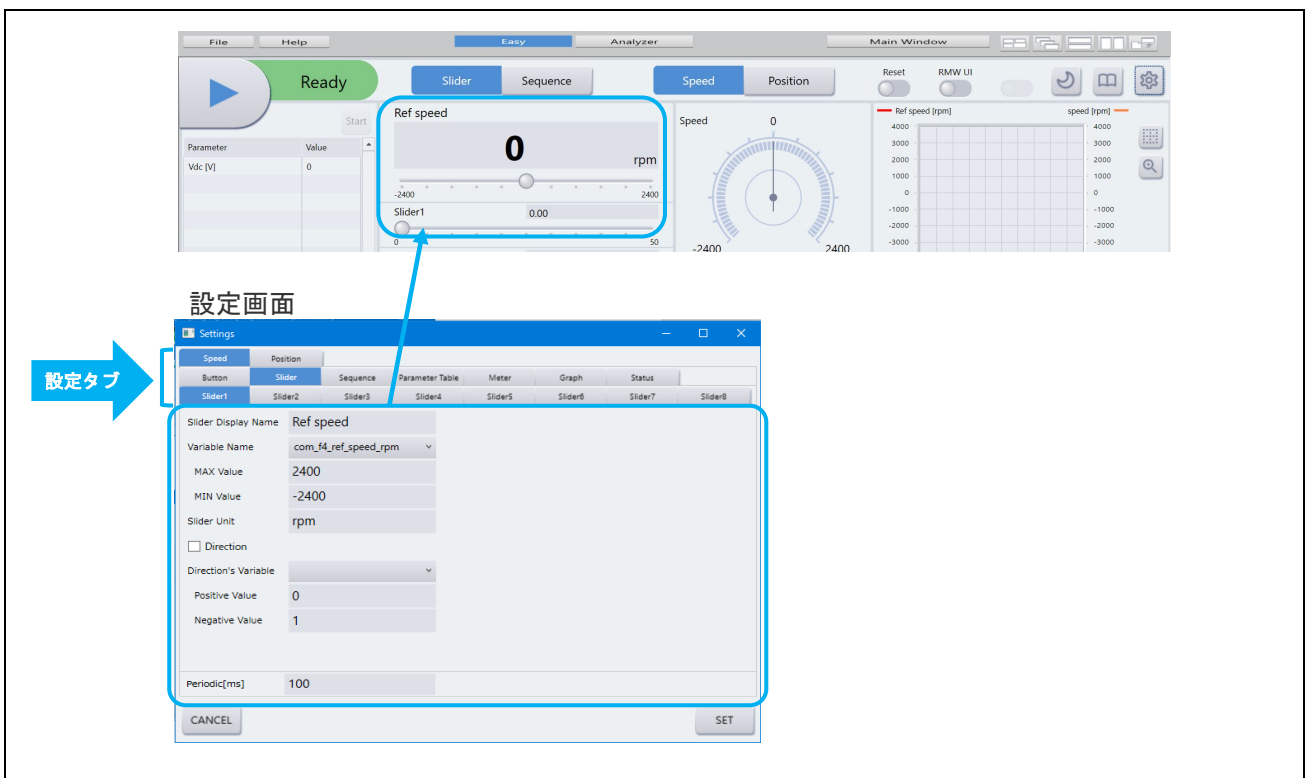


図 12.15 スライダー設定例

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		<ul style="list-style-type: none"> 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

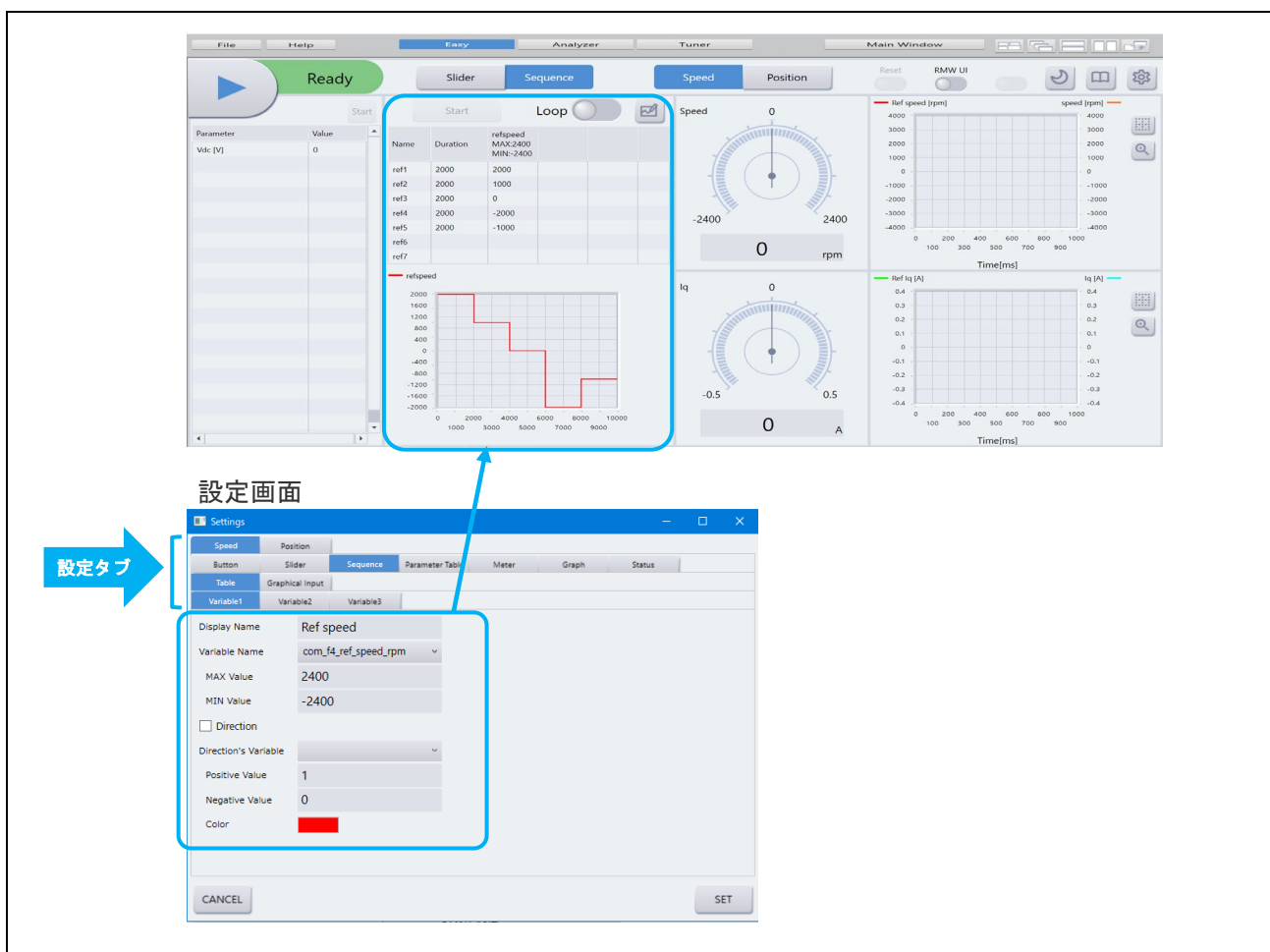


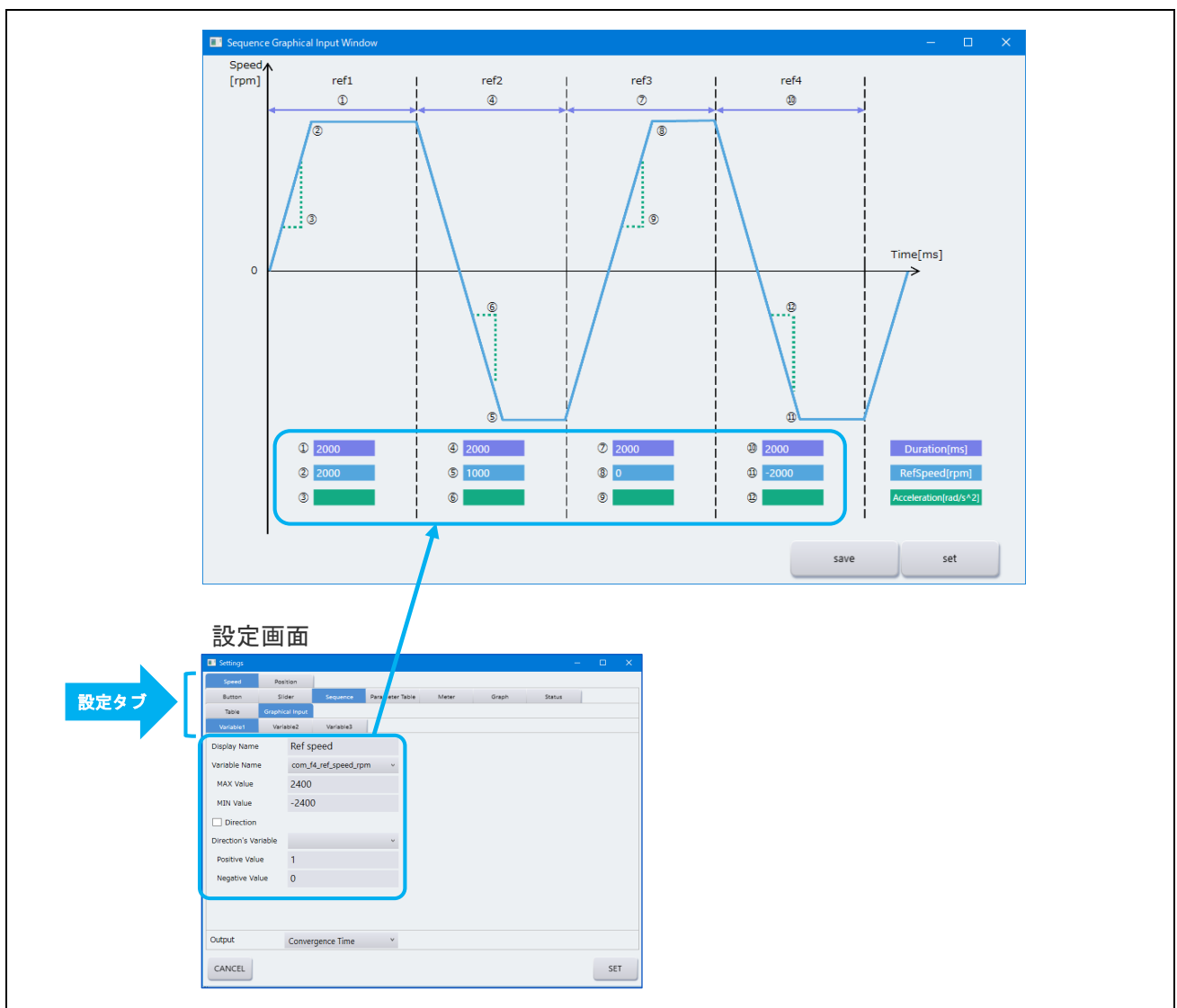
図 12.16 シーケンス設定例

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		<ul style="list-style-type: none"> 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





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		<ul style="list-style-type: none"> 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

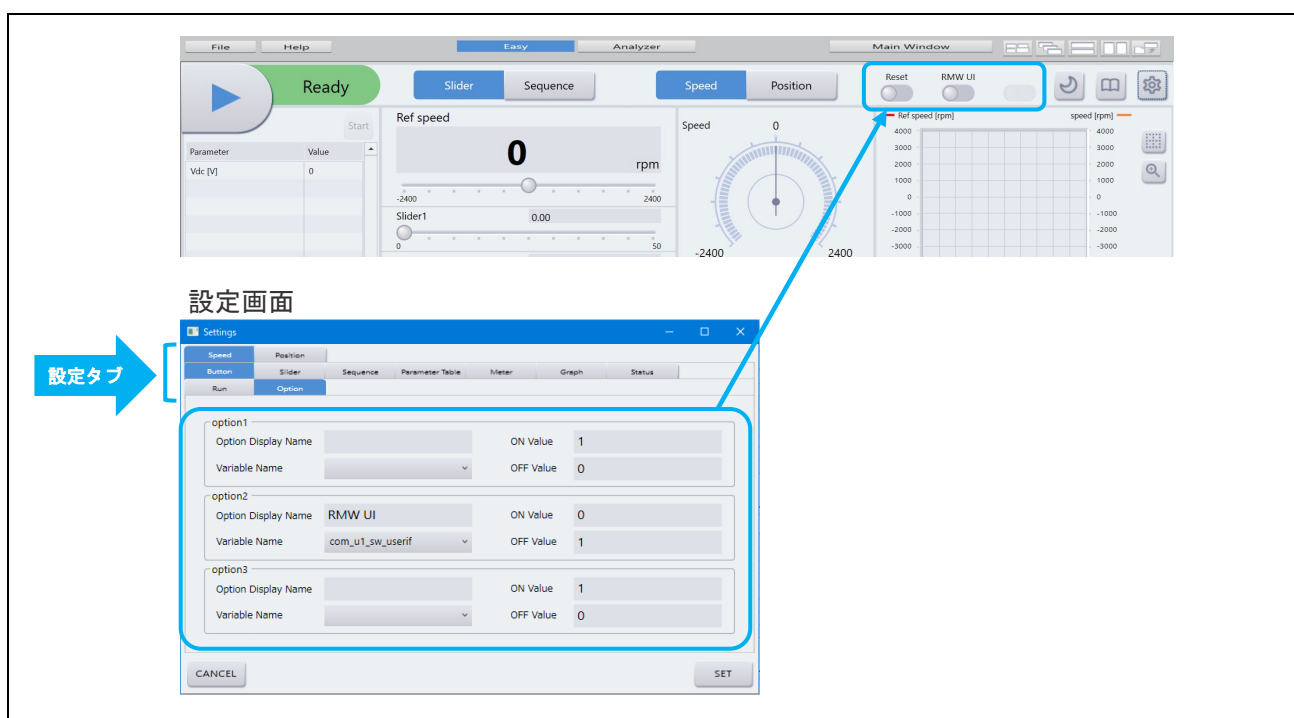


図 12.18 オプションスイッチ設定例

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 <ul style="list-style-type: none"> 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 の色で表示し、変化後を の色で表示 100ms 周期で変数の値をメータに表示 	

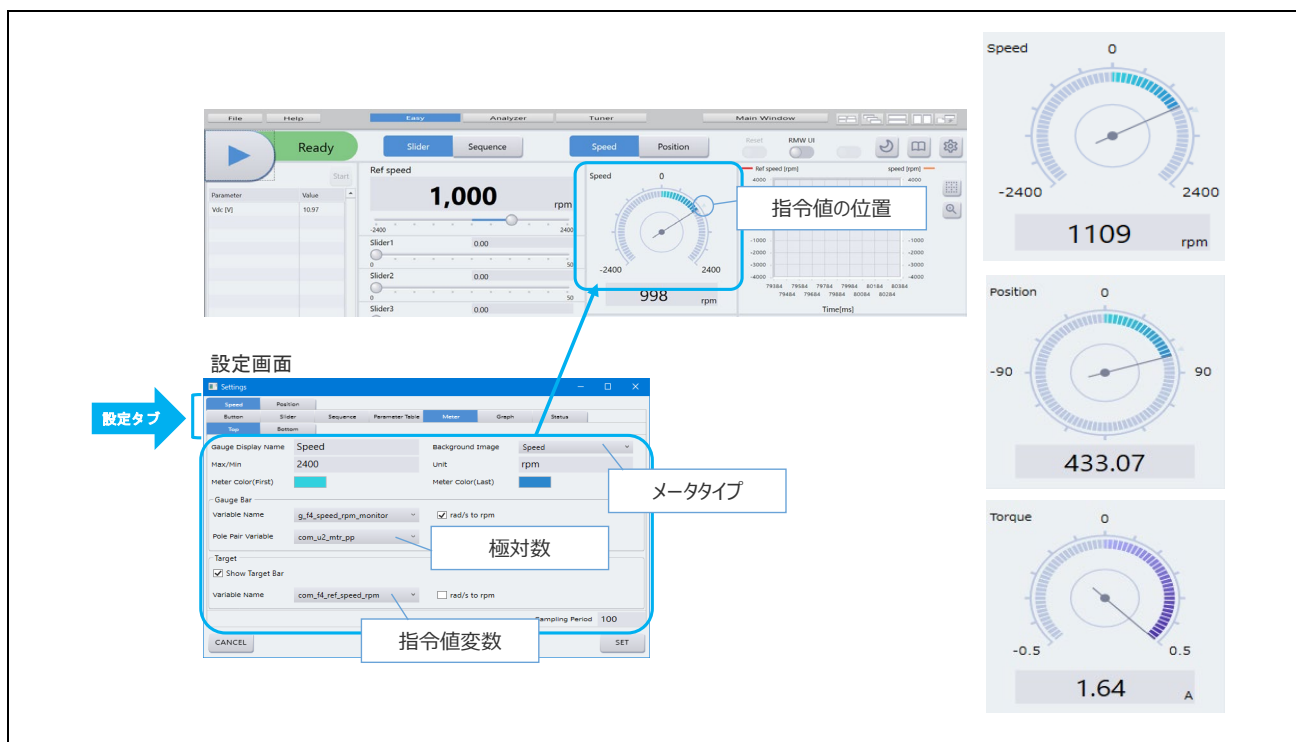


図 12.19 メータ設定例

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 <ul style="list-style-type: none"> 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

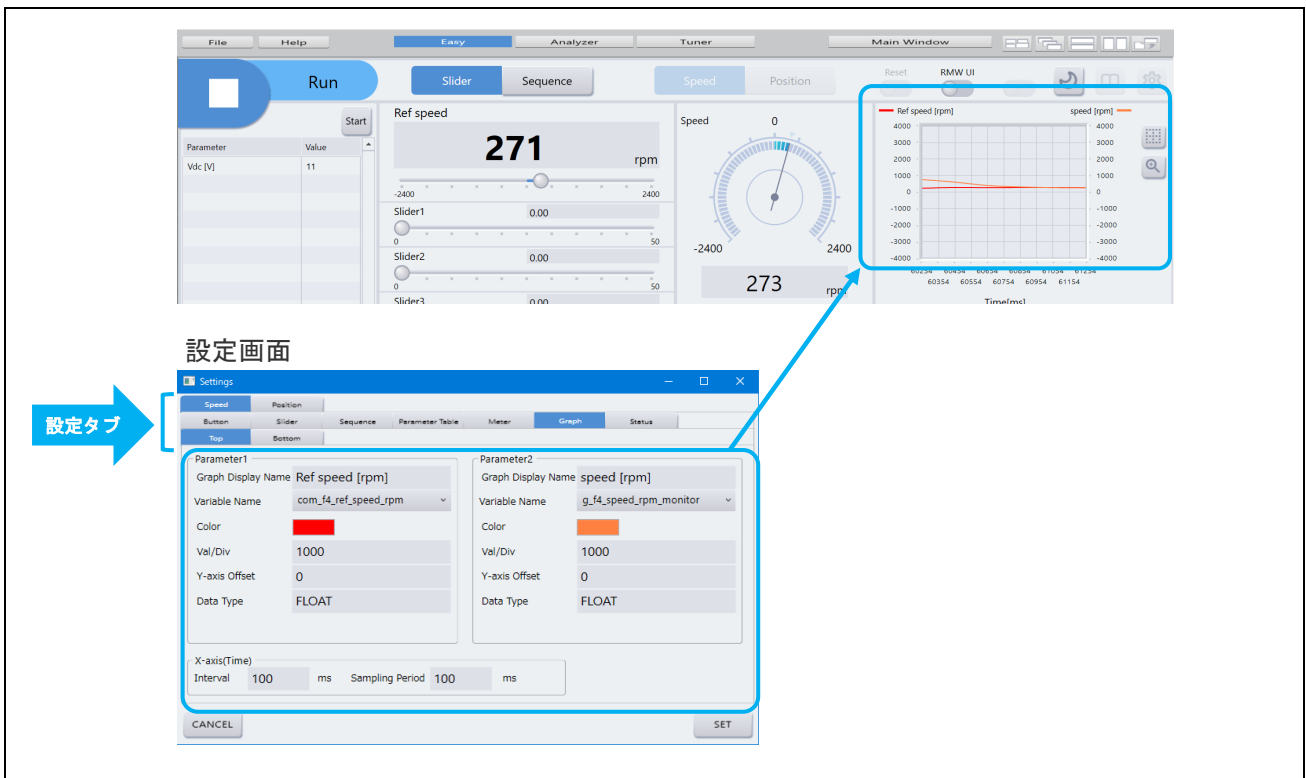


図 12.20 グラフ設定例

12.7 テーマ色切替え


GUI 画面右上の  を押しと、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、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。

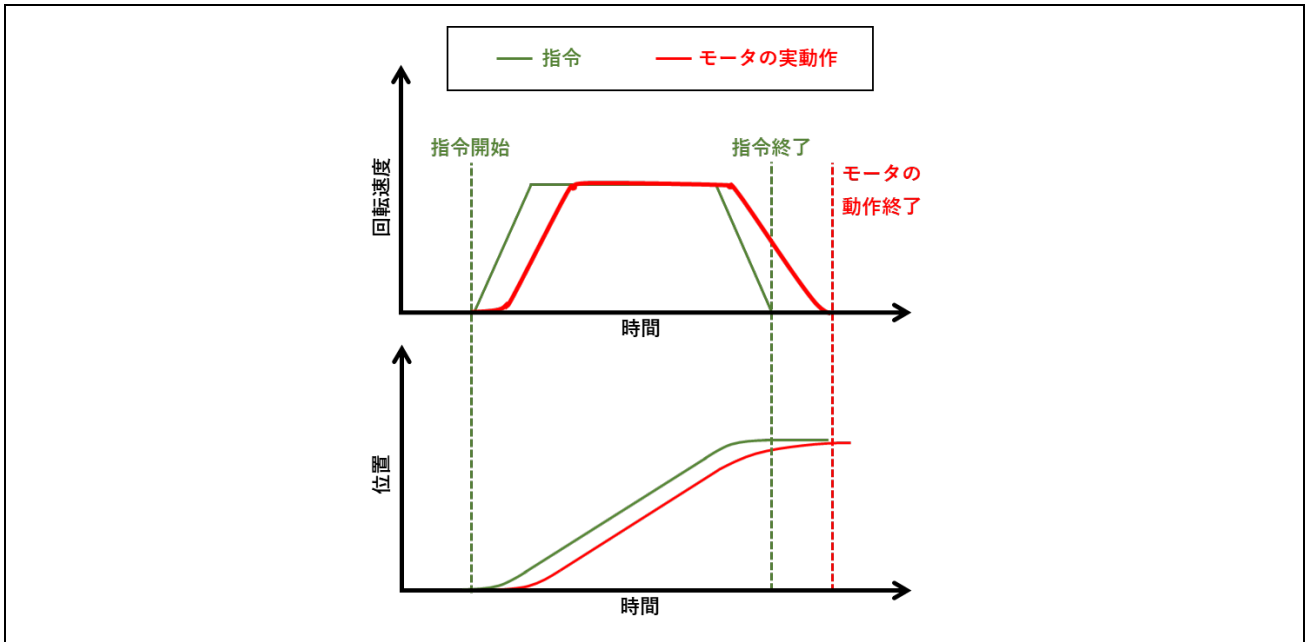


図 13.1 指令と実際の動作

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）を組み合わせる制御を行います。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。

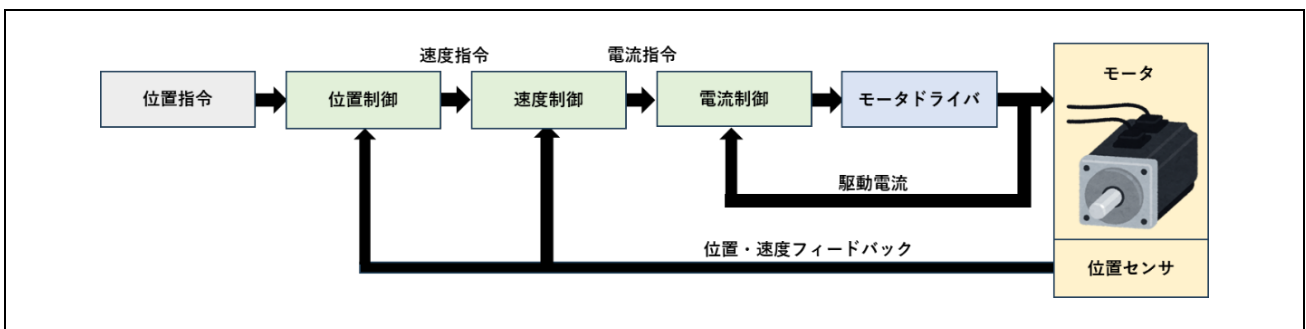


図 13.2 サーボ制御のブロック線図

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。

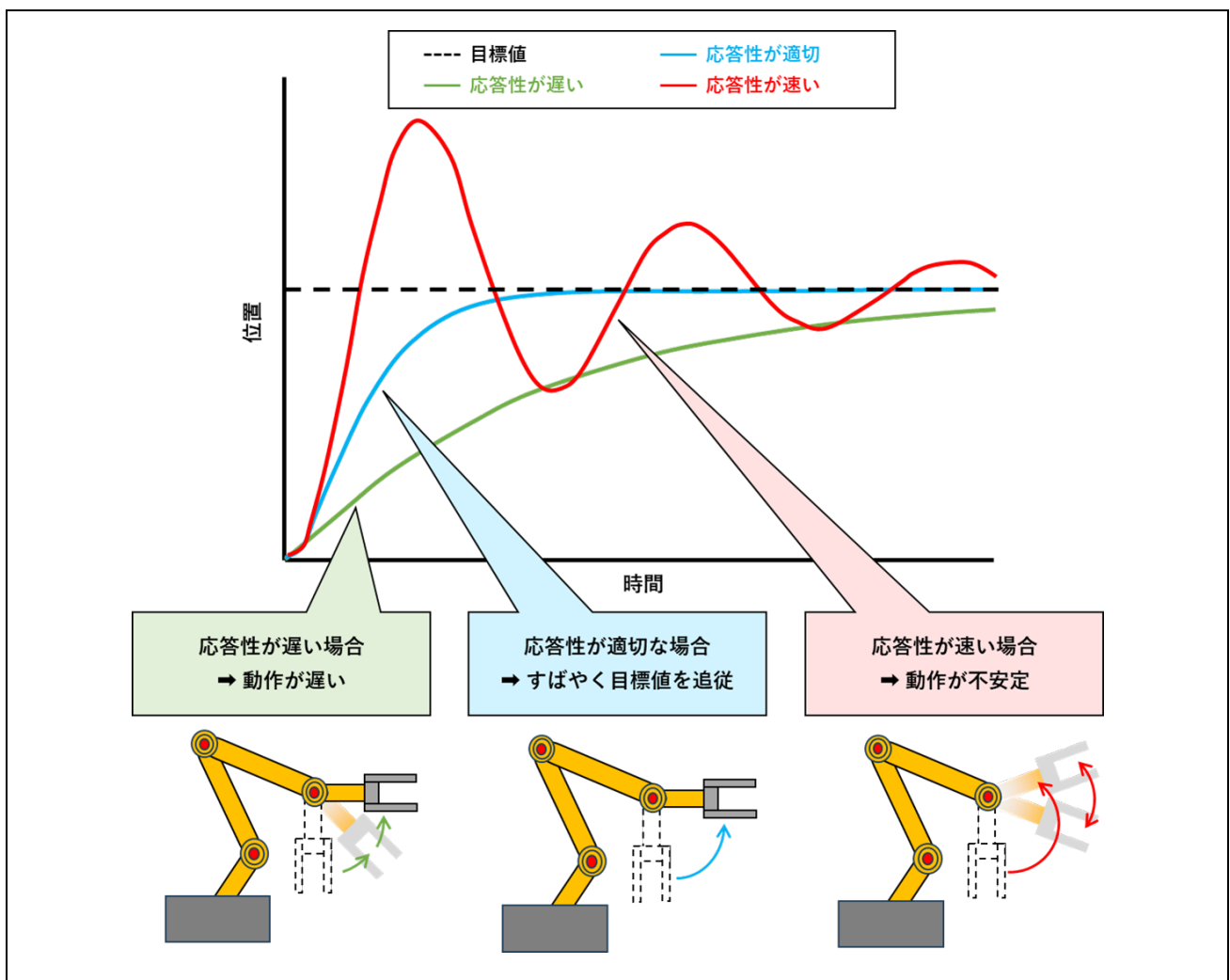


図 13.3 サーボ制御の応答性

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。

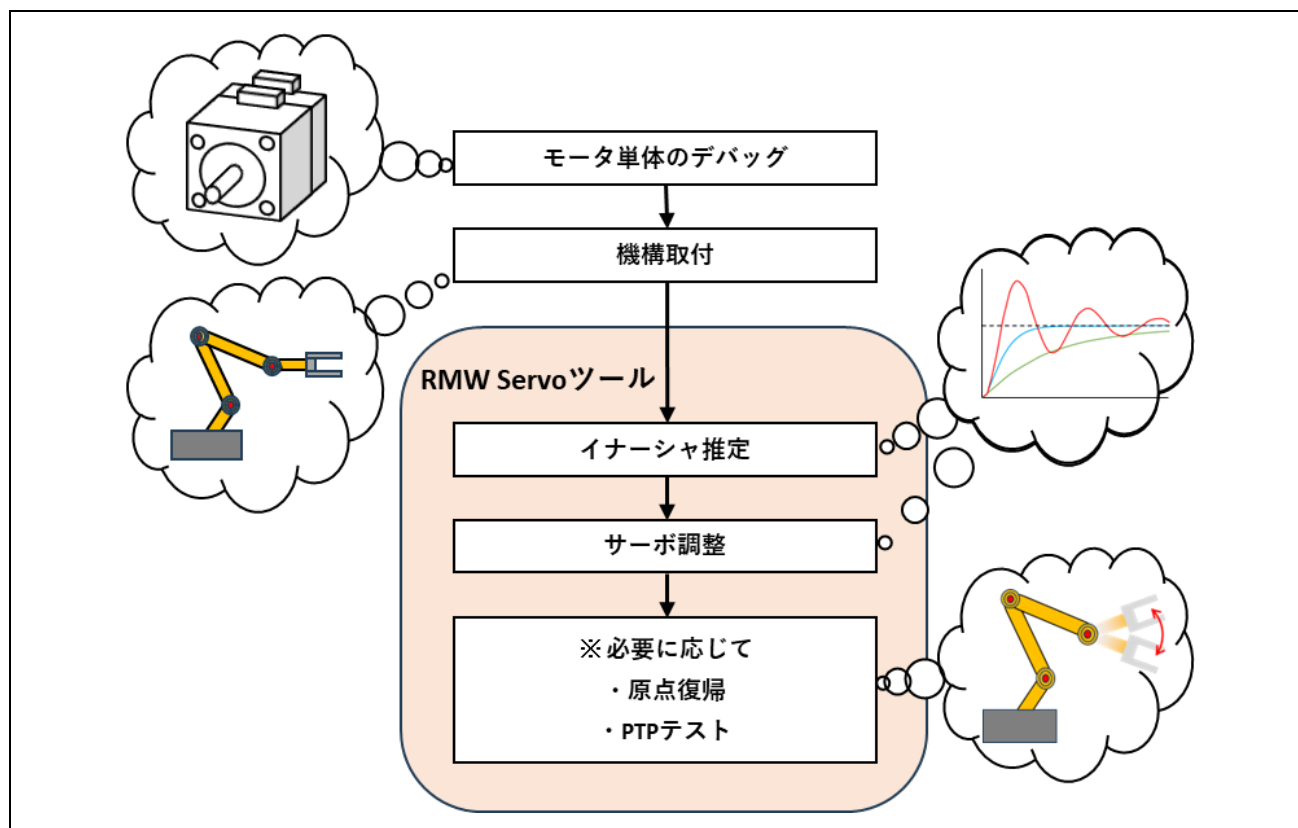


図 13.4 Servo ツール使用の流れ

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、物体は回りが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。

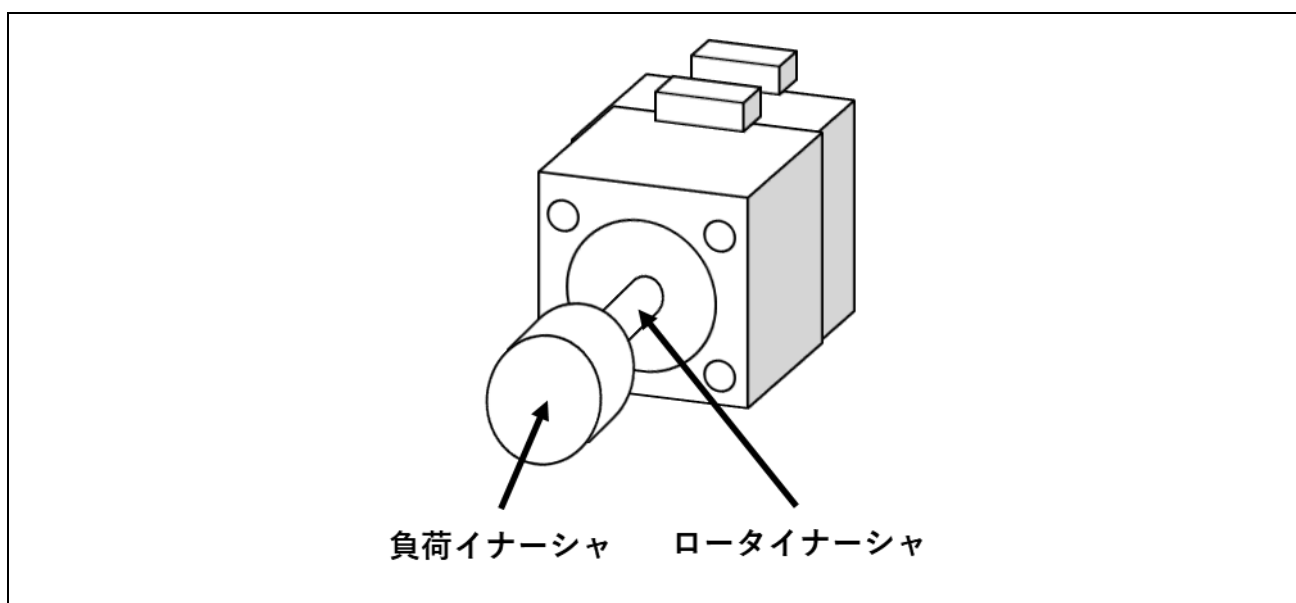


図 13.5 モータと機構のイナーシャ

13.3.1.2 インナーシャ推定とは

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。

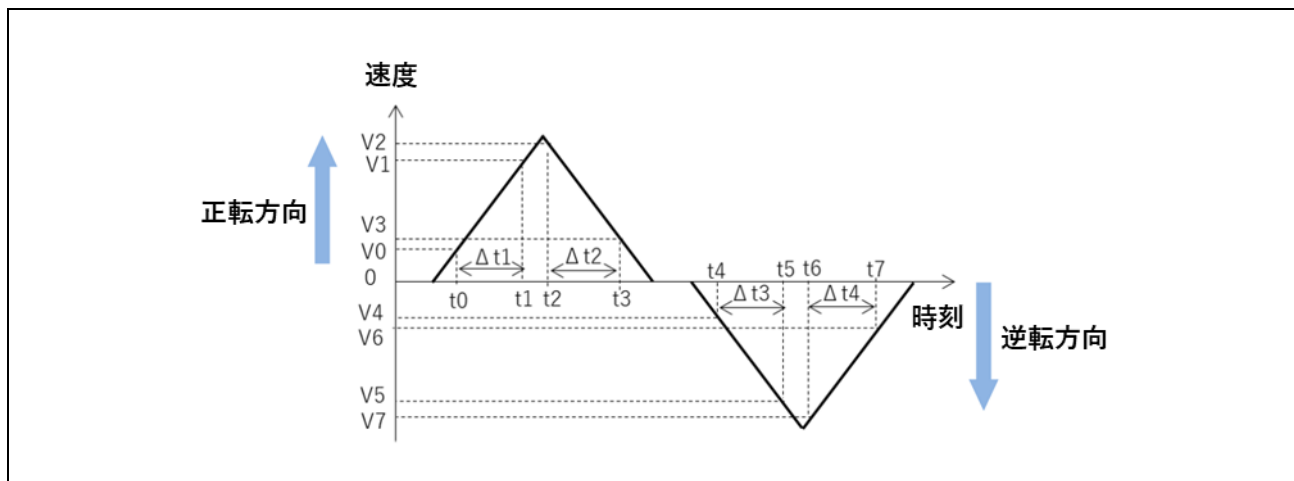


図 13.6 インナーシャ推定時動作

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。

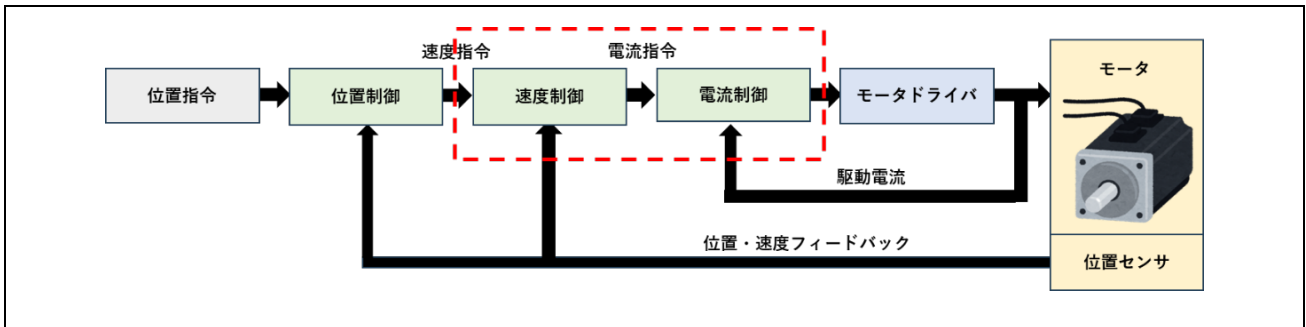


図 13.7 サーボ制御におけるイナーシャ推定の影響範囲

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

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。

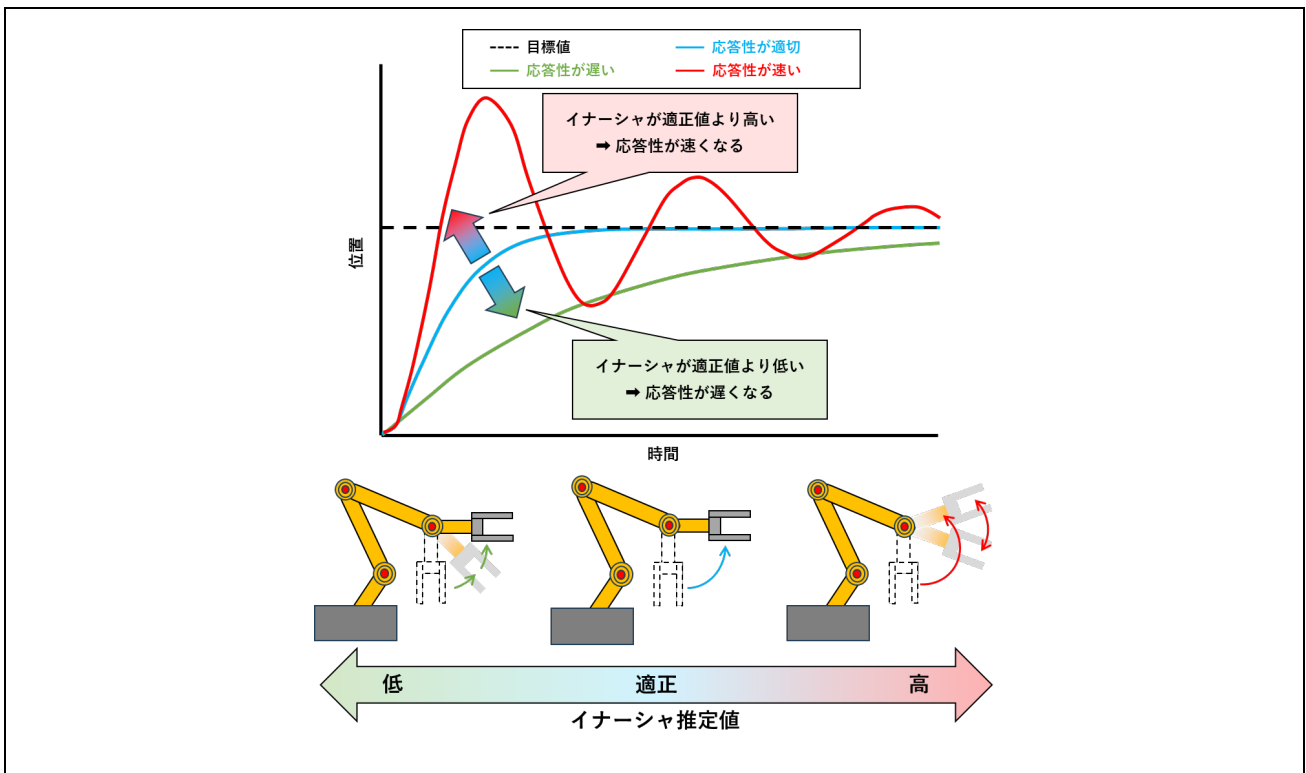


図 13.8 サーボ制御におけるイナーシャの影響

13.3.2 画面構成・パラメーター一覧

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。

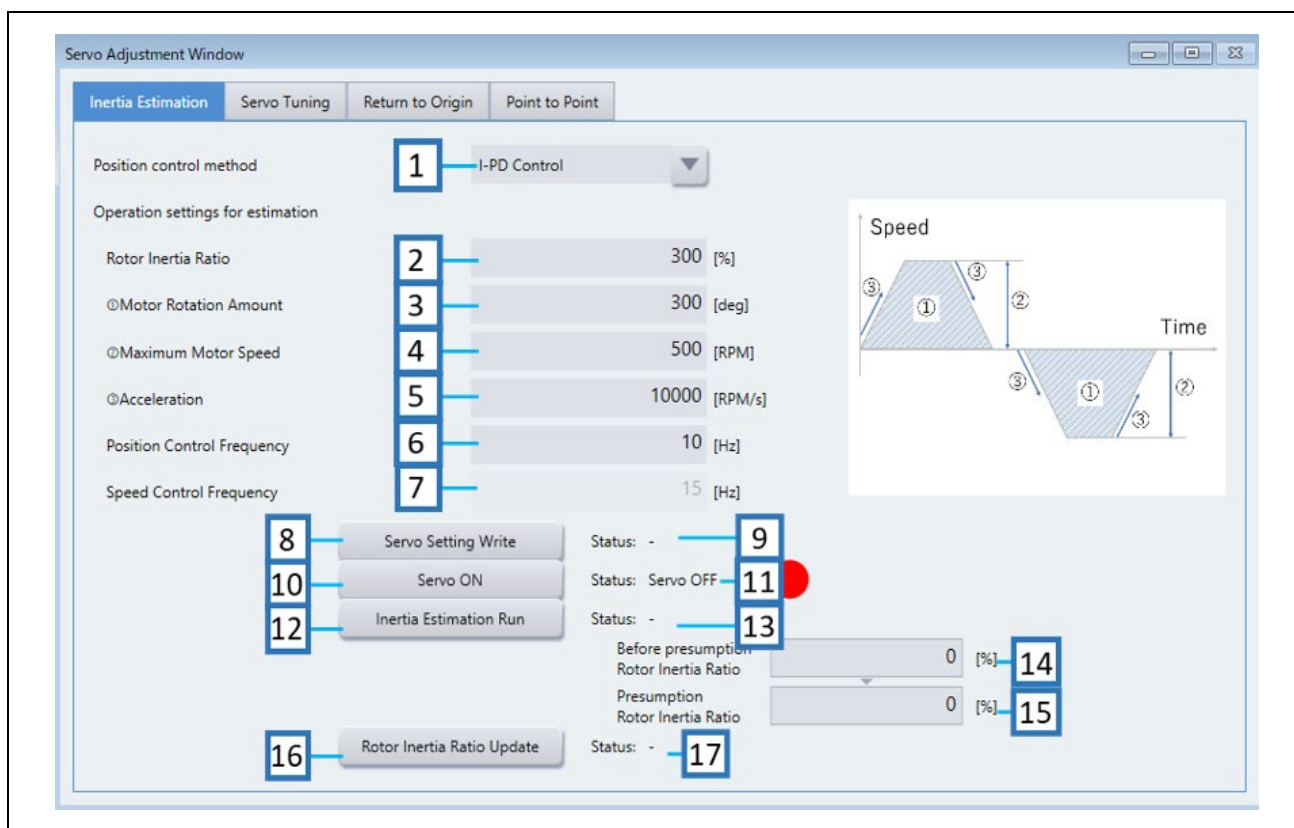


図 13.9 Inertia Estimation タブ画面

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。

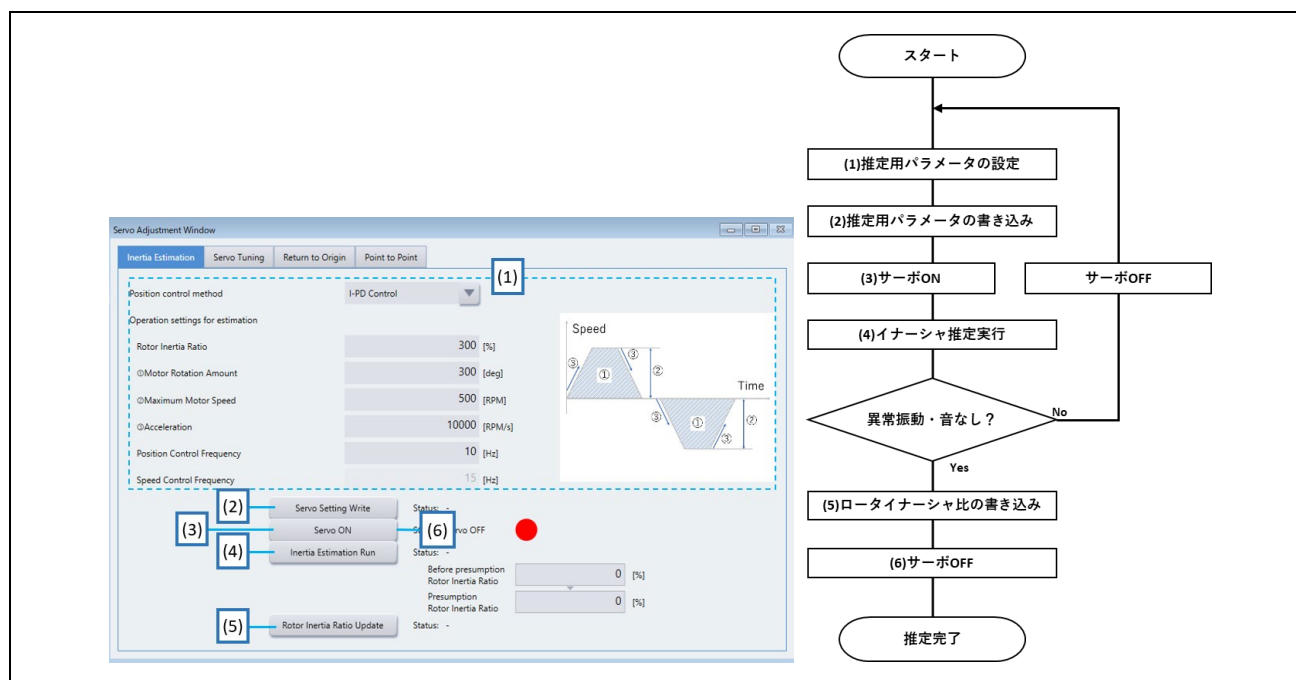


図 13.10 イナーシャ推定操作手順

(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・パラメーター一覧」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ervo調整

13.4.1 機能説明

13.4.1.1 サervo調整とは

使用する環境（装置・機構）に合わせてサervo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る制御し、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。

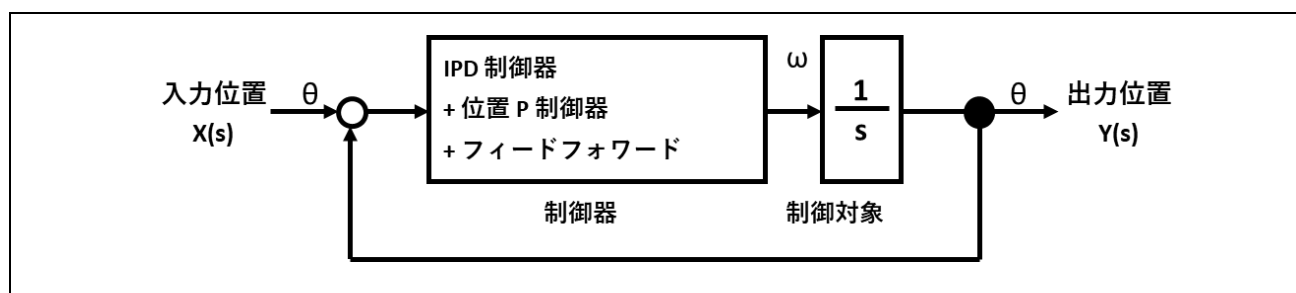


図 13.11 IPD 制御のモデル(位置)

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。

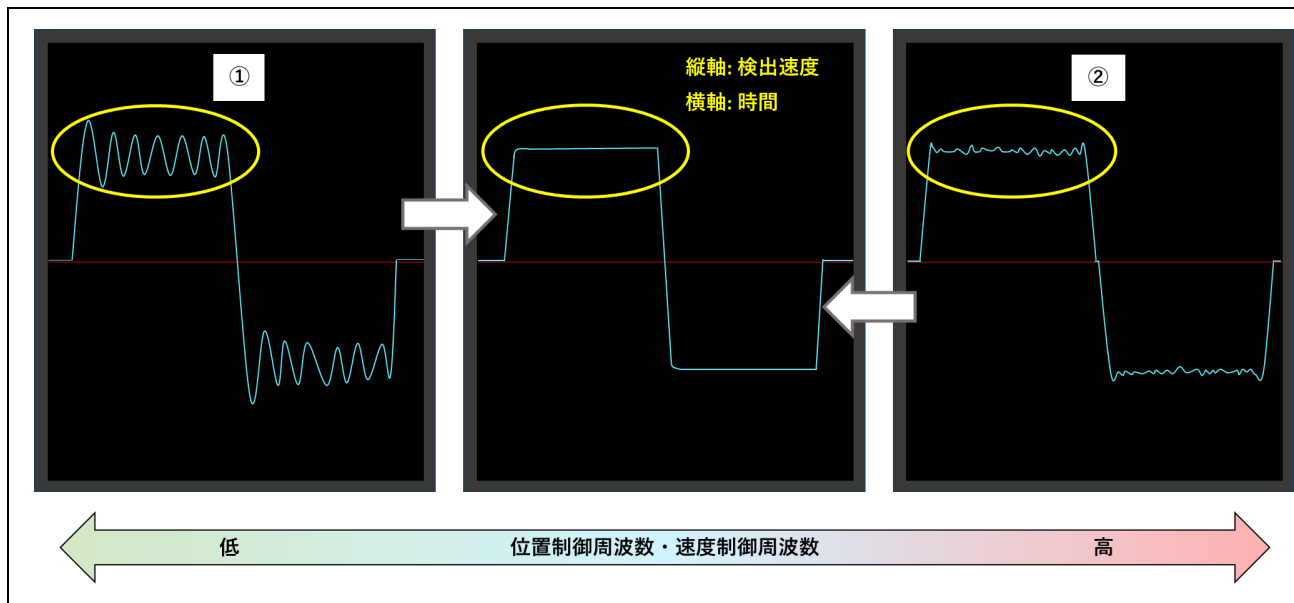


図 13.12 制御周波数と検出速度の波形

13.4.2 画面構成・パラメーター一覧

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。

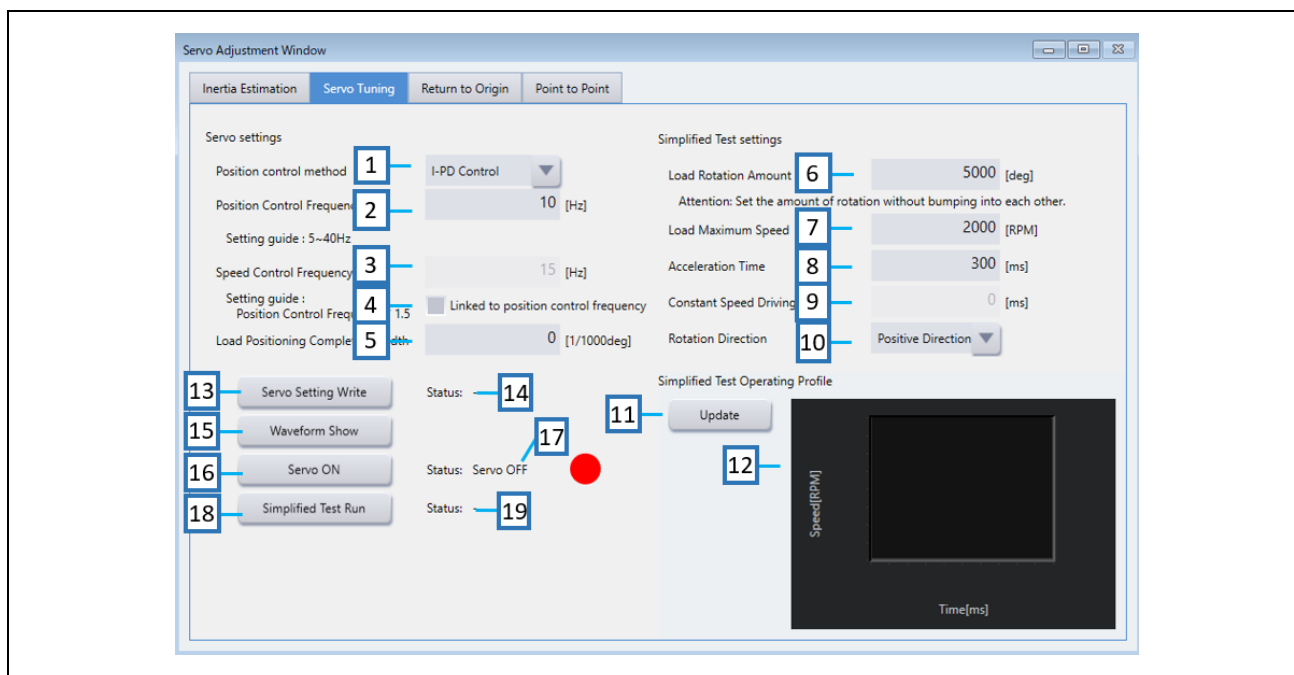


図 13.13 Servo Tuning タブ画面

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。

- (ア)No.6(Load Rotation Amount)
- (イ)No.7(Load Maximum Speed)
- (ウ)No.8(Acceleration Time)
- (エ)No.9(Constant Speed Driving Time)

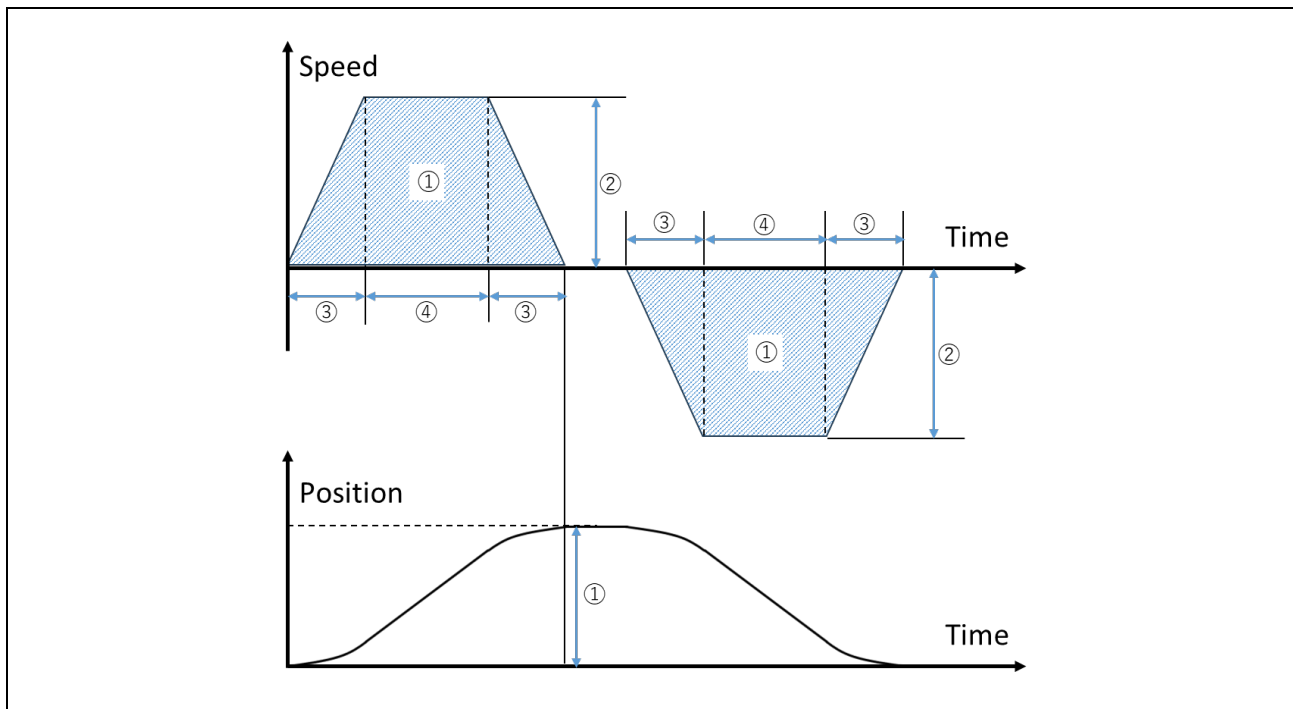


図 13.14 パラメータとプロファイルの関係

13.4.3 操作説明

サーボ調整の手順を以下に示します。

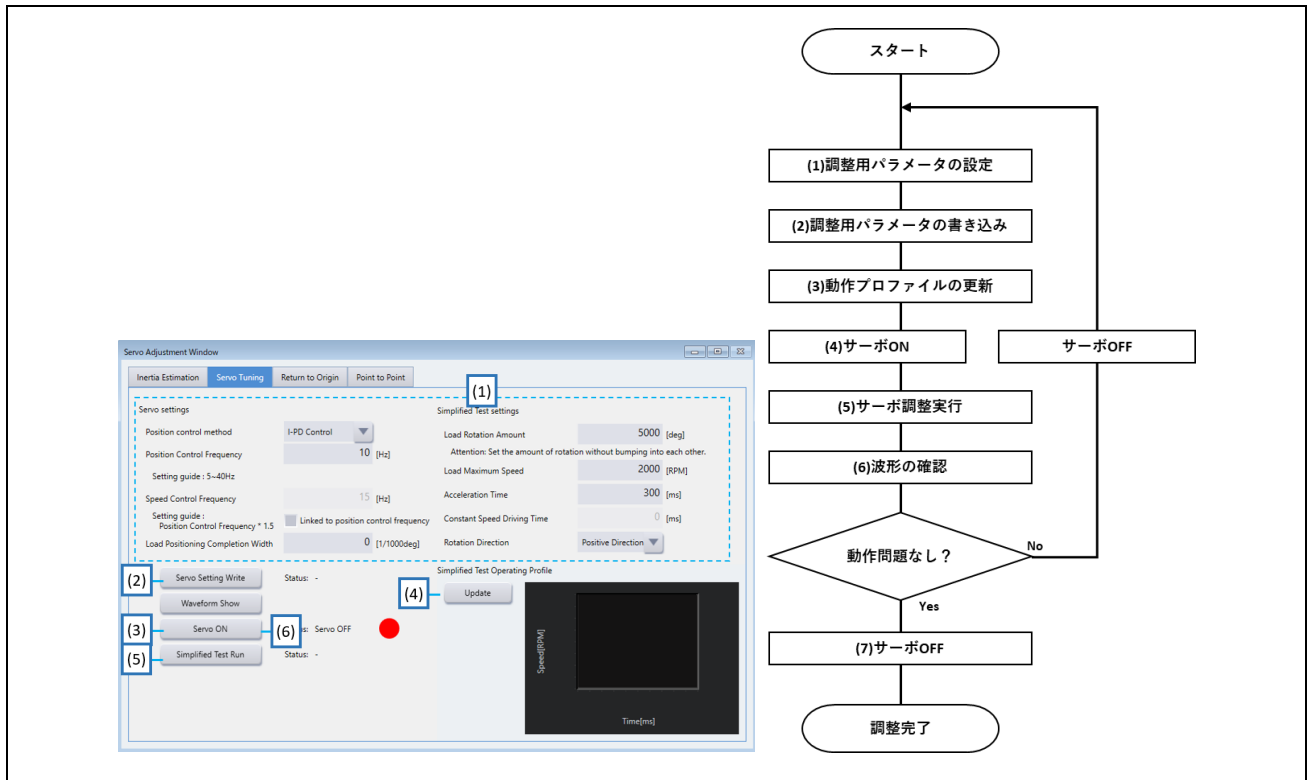


図 13.15 サーボ調整操作手順

(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・パラメーター一覧」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。

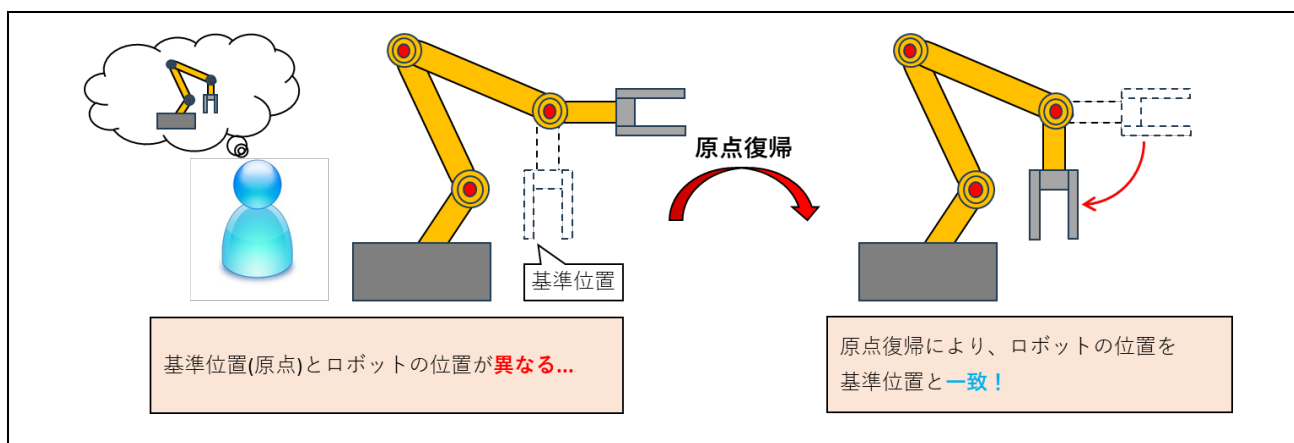


図 13.16 原点復帰とは

13.5.1.2 押し当て方式とは

押し当て方式は、メカをストップなどに接触させ、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。

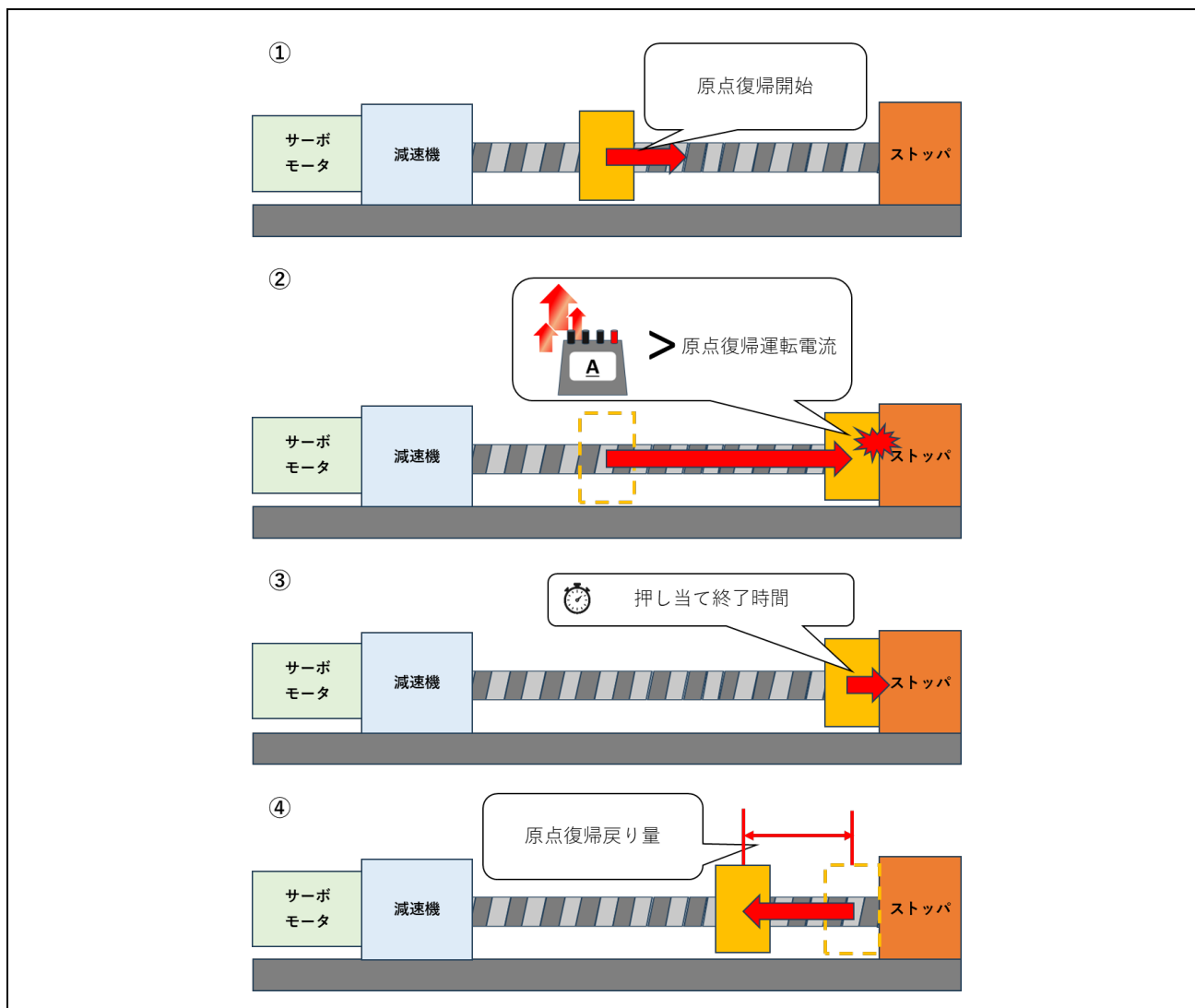


図 13.17 原点復帰動作押し当て方式

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 機構がストップ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・パラメーター一覧

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。

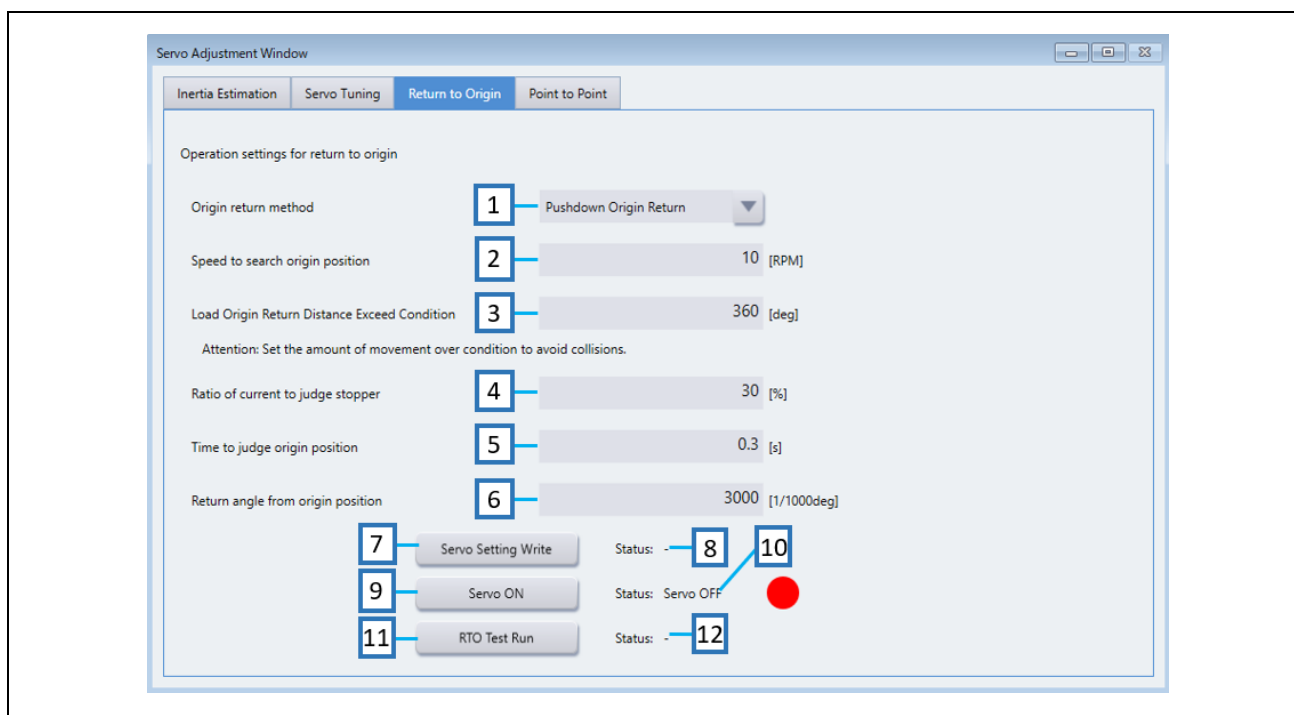


図 13.18 Return to Origin タブ画面

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。

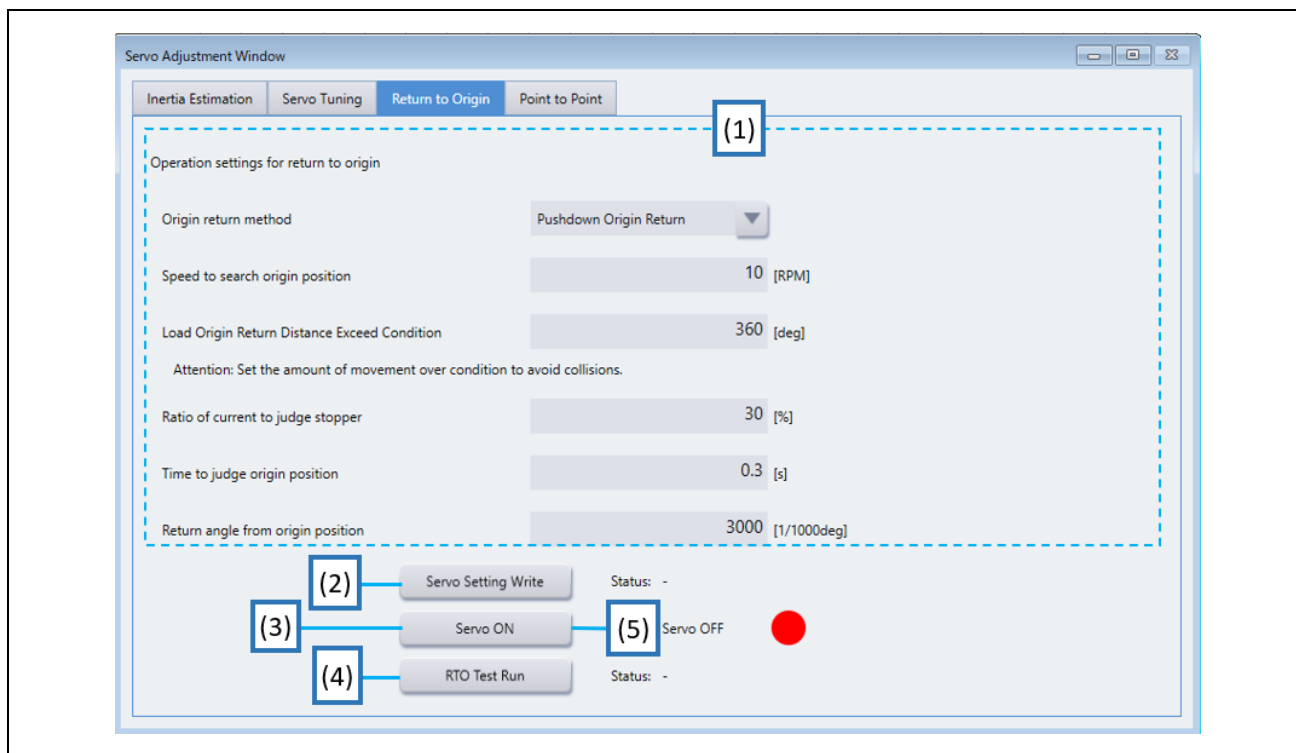


図 13.19 原点復帰操作手順

(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・パラメーター一覧」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。

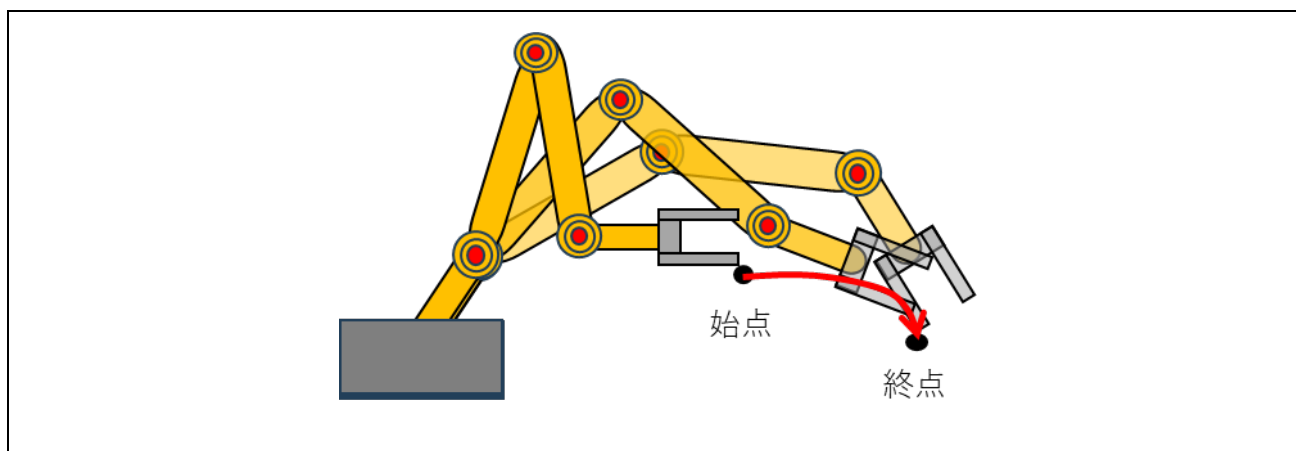


図 13.20 PTP 動作例

13.6.2 画面構成・パラメーター一覧

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。

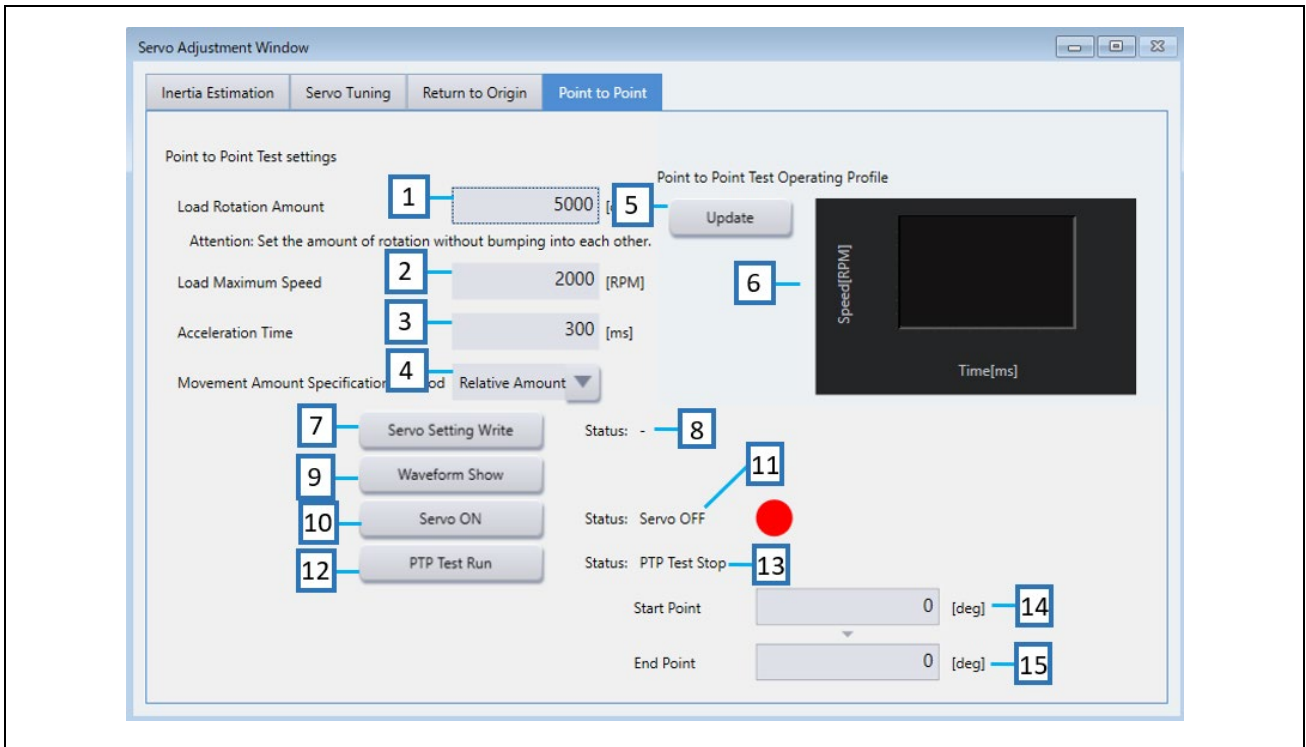


図 13.21 Point to Point タブ画面

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)

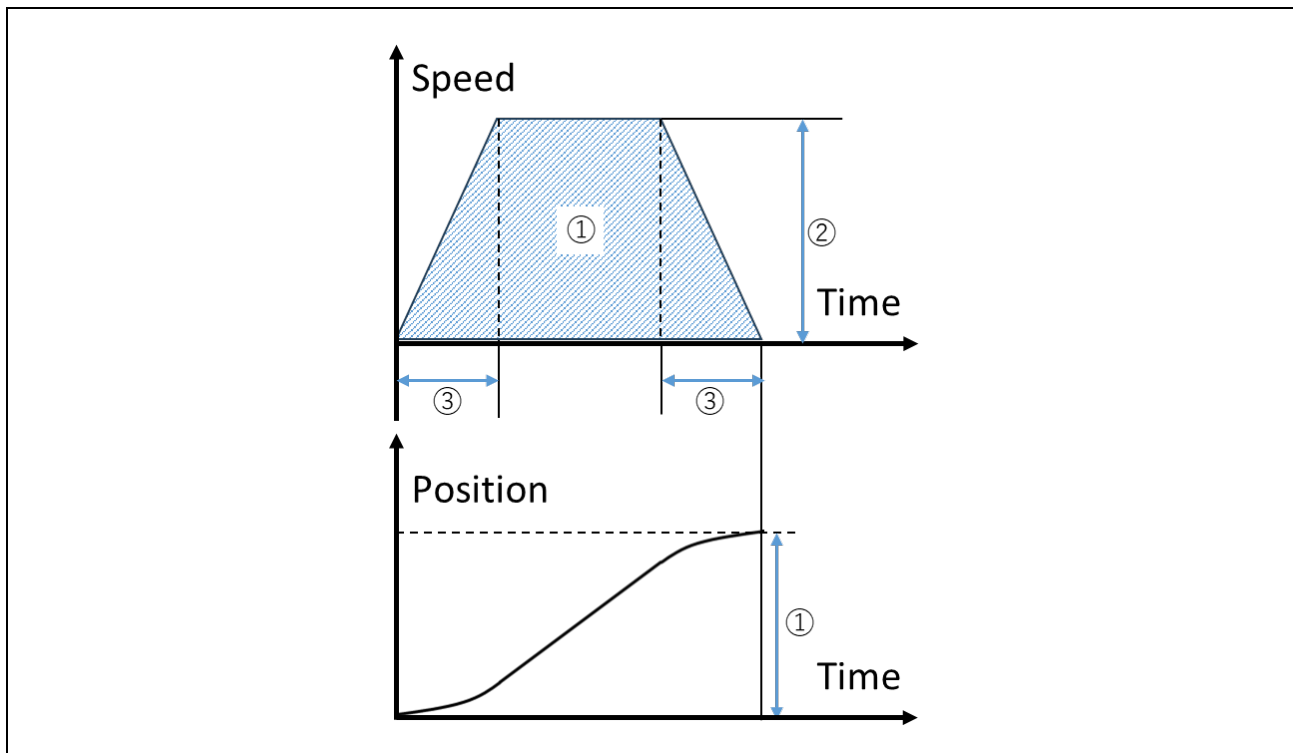


図 13.22 パラメータとプロファイルの関係

13.6.3 操作説明

Point to Point 動作の手順を以下に示します。

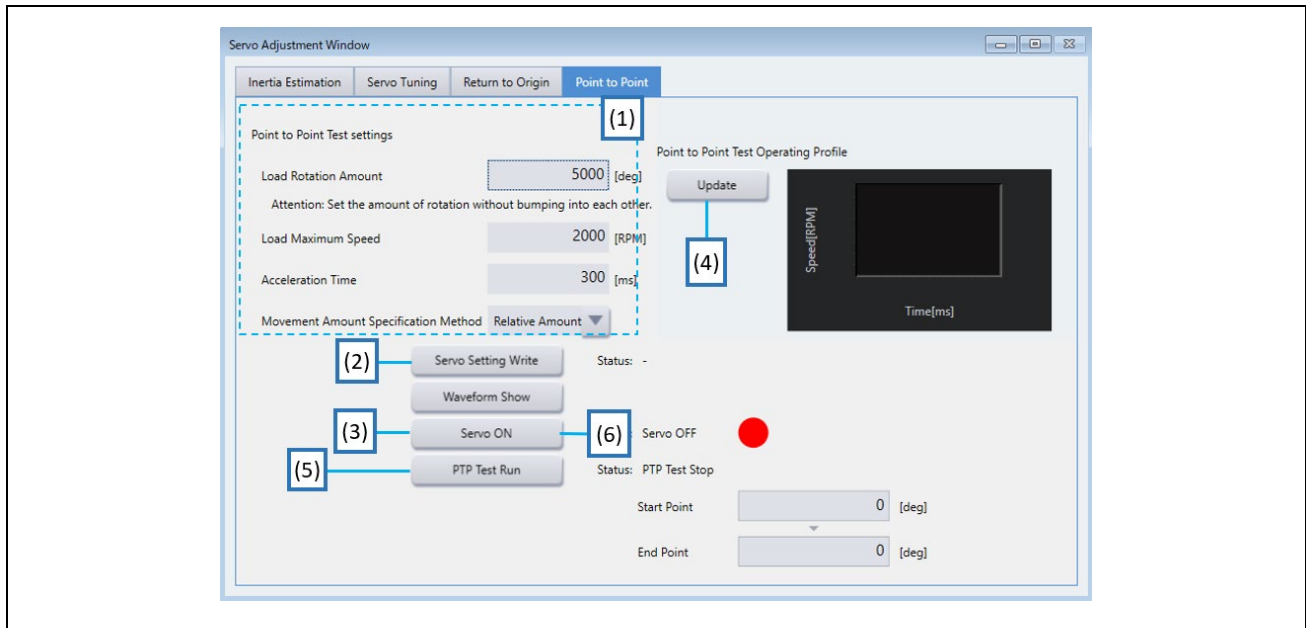


図 13.23 Point to Point 動作操作手順

(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・パラメーター一覧」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。

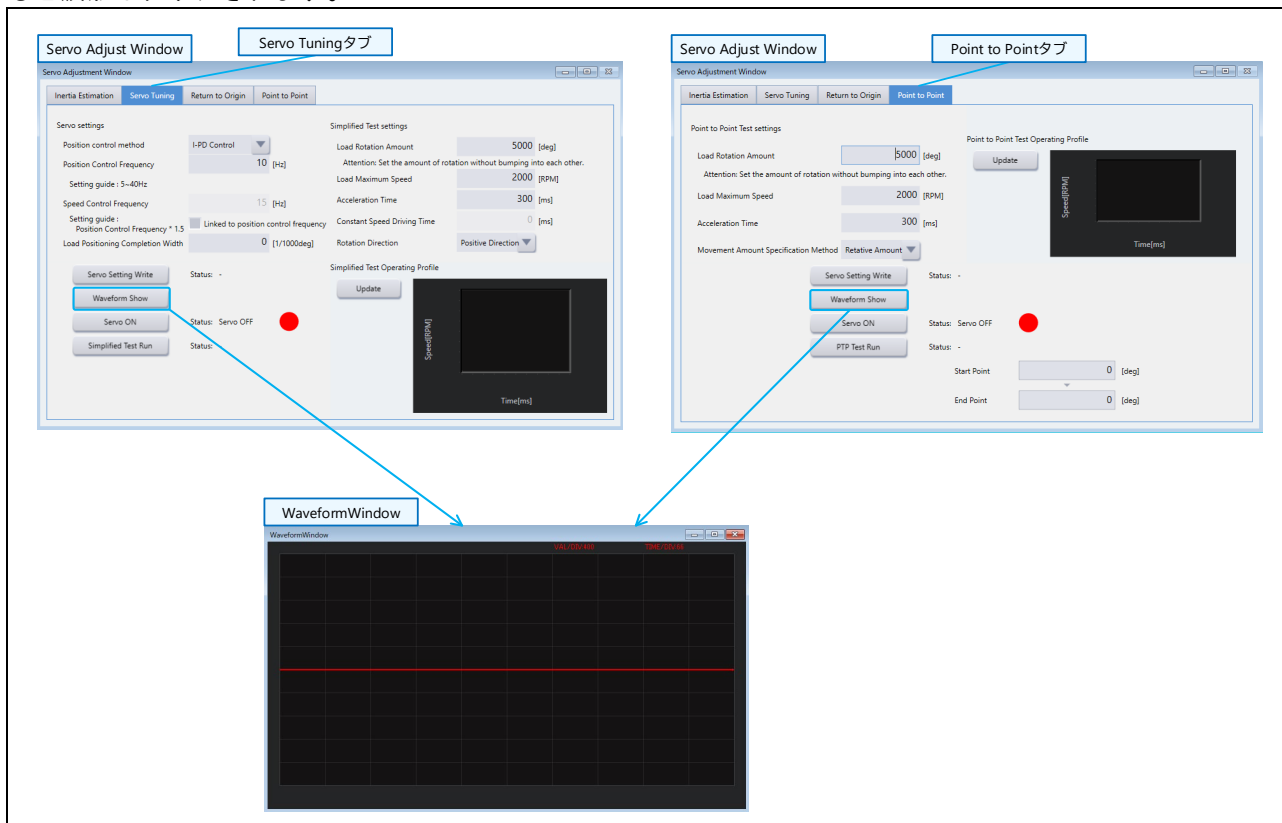


図 13.24 Waveform 画面

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。

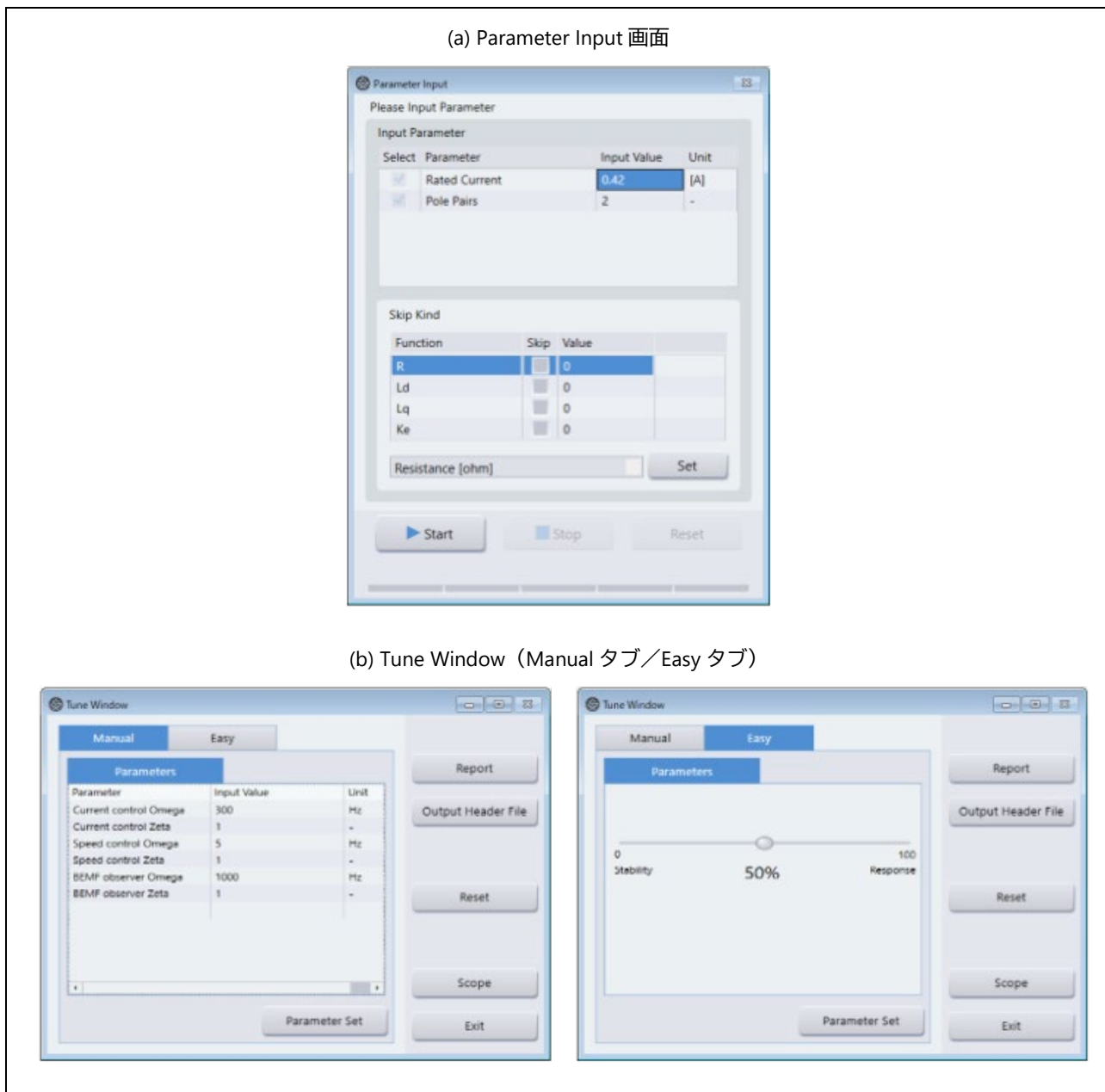


図 14.1 Tuner ツール画面

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る、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）。

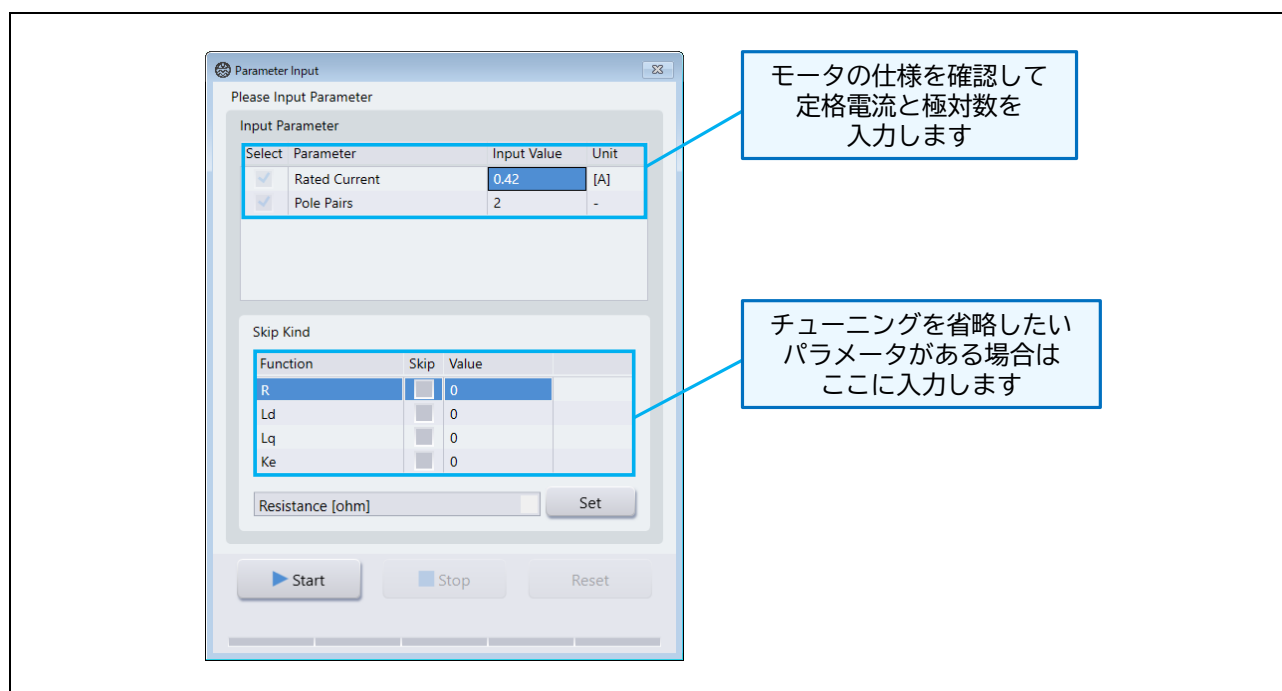


図 14.2 Parameter Input 画面

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。

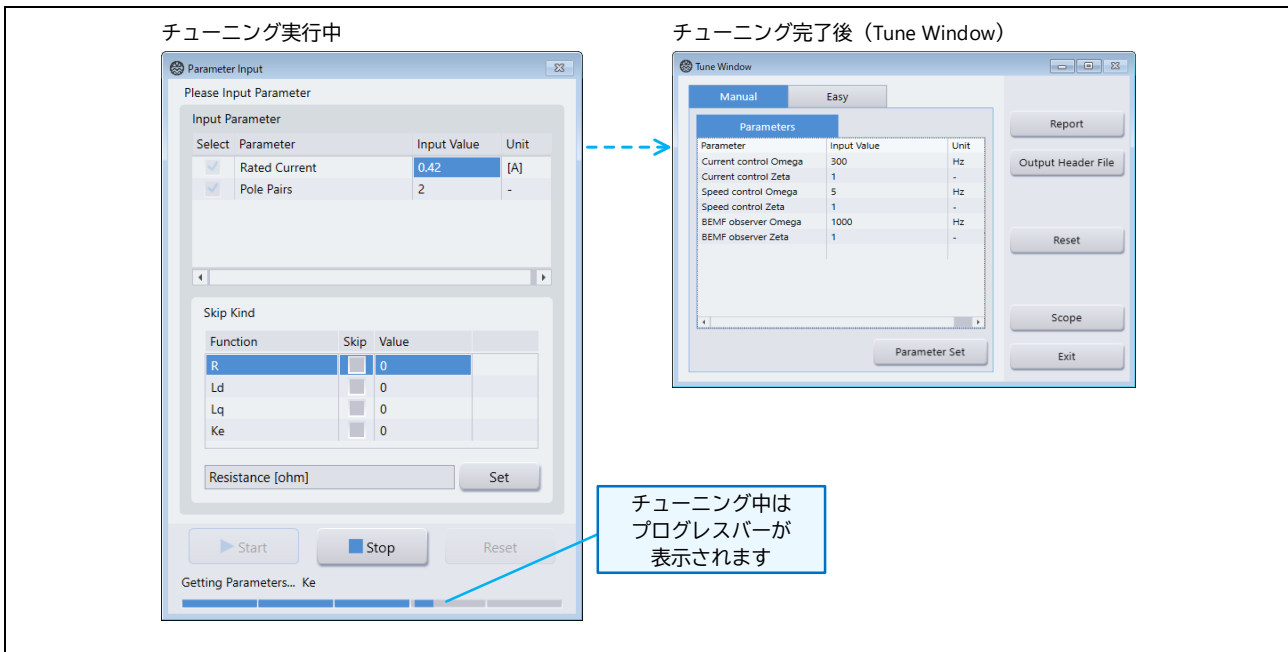


図 14.3 チューニング実行中とチューニング完了後の Tune Window 表示

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。

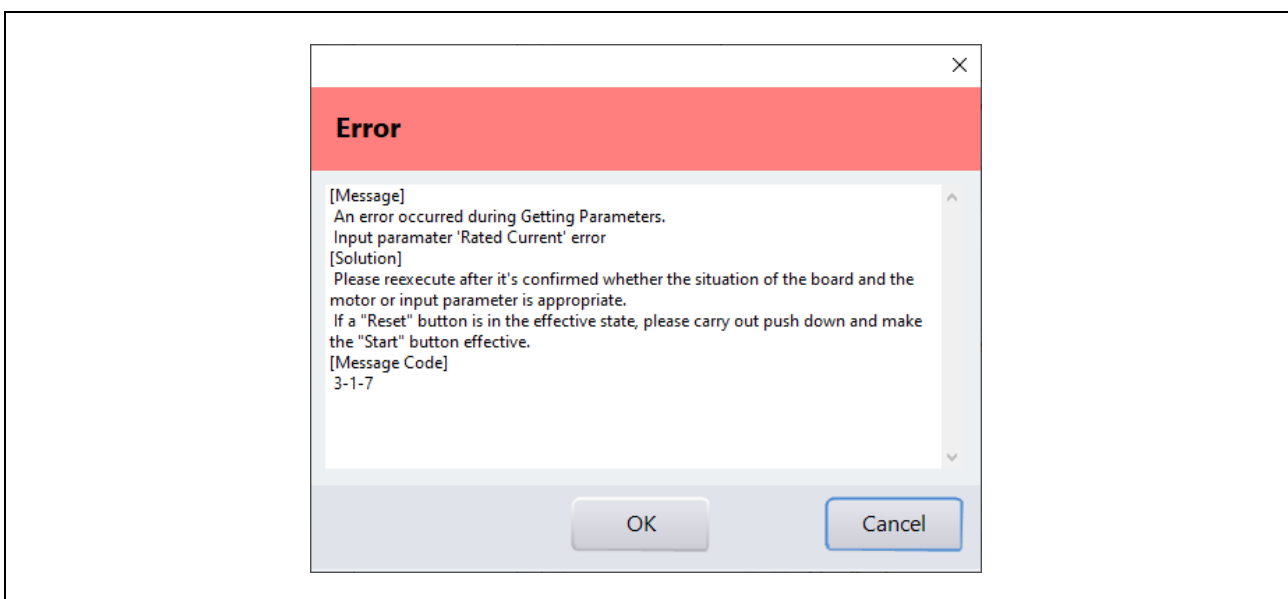


図 14.4 エラーメッセージ例

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。

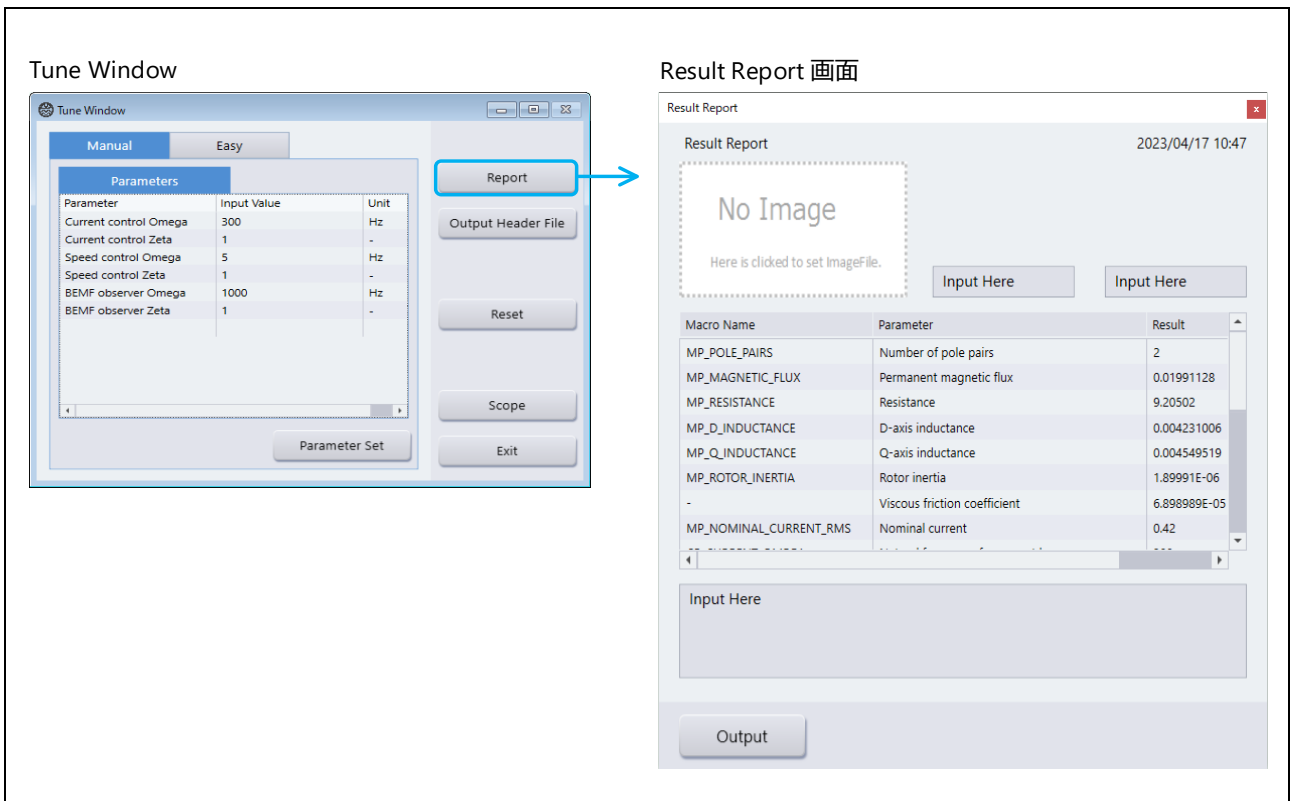


図 14.5 Result Report 画面の表示

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。

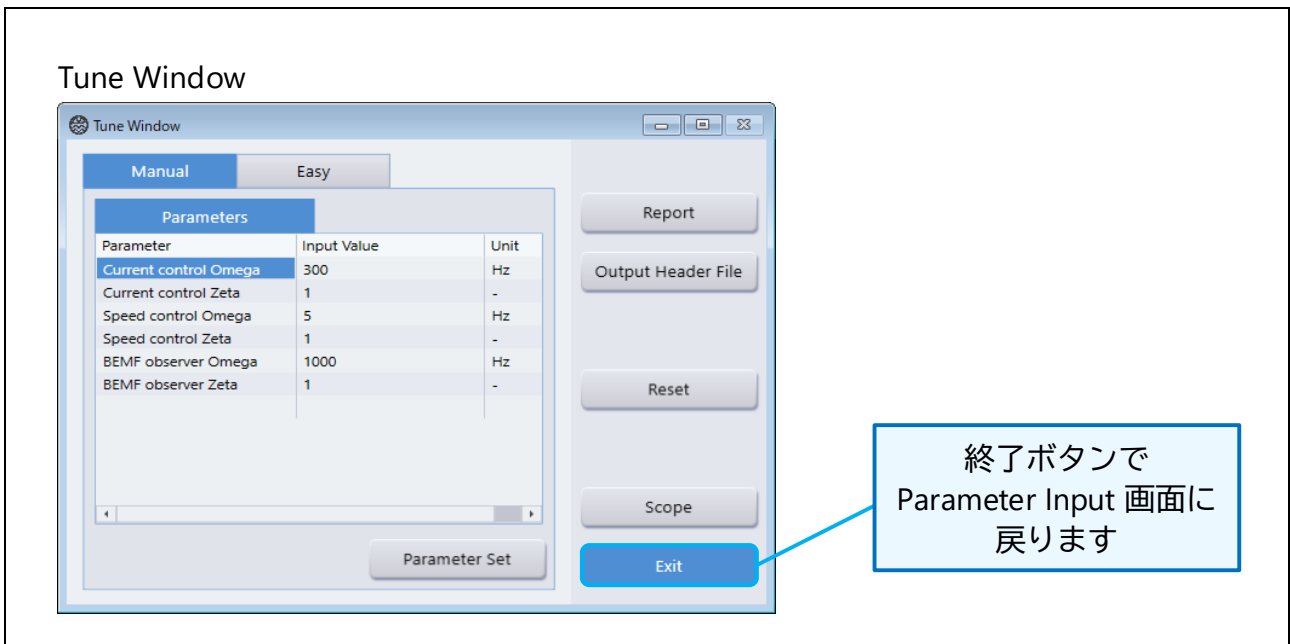


図 14.6 Exit ボタン

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。

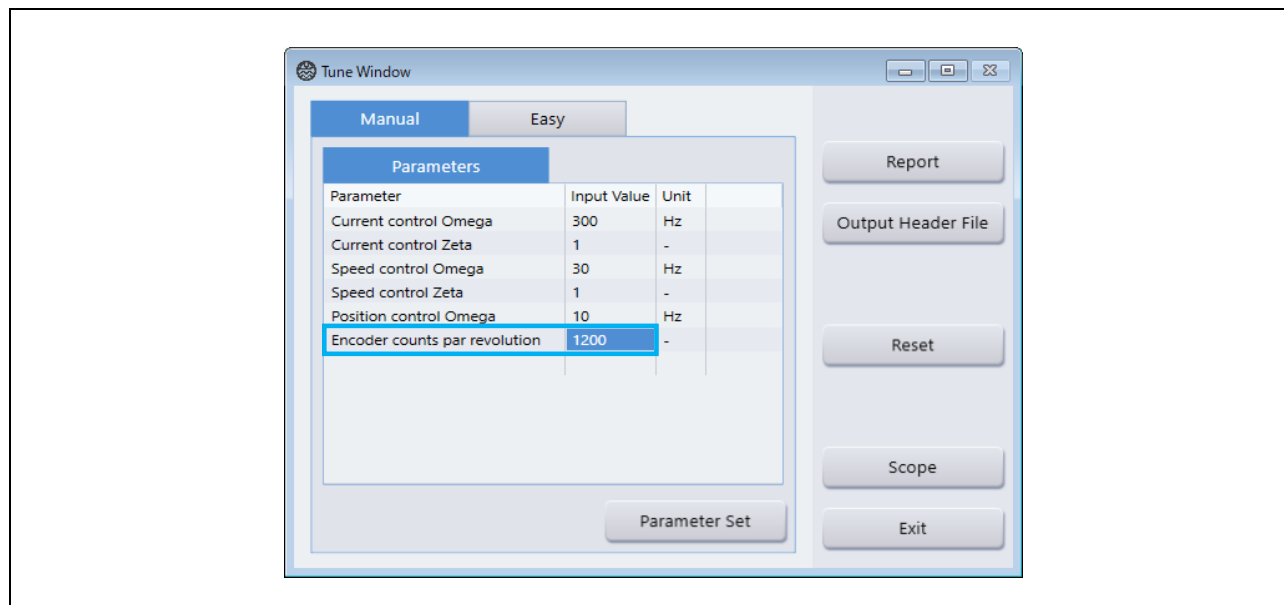


図 14.7 エンコーダパルス数の入力（エンコーダ用プログラムのみ）

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。

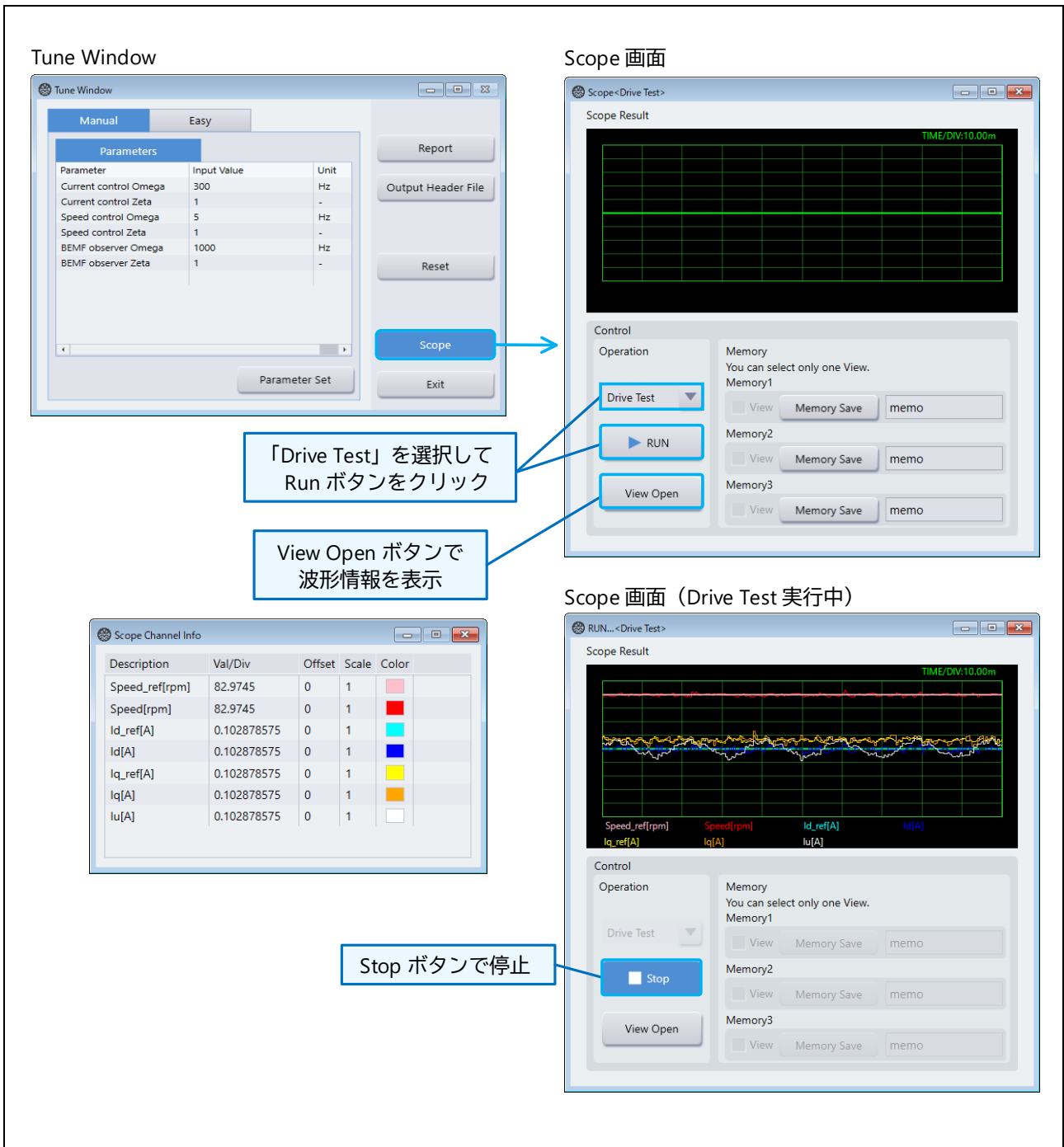


図 14.8 Drive Test 画面

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。

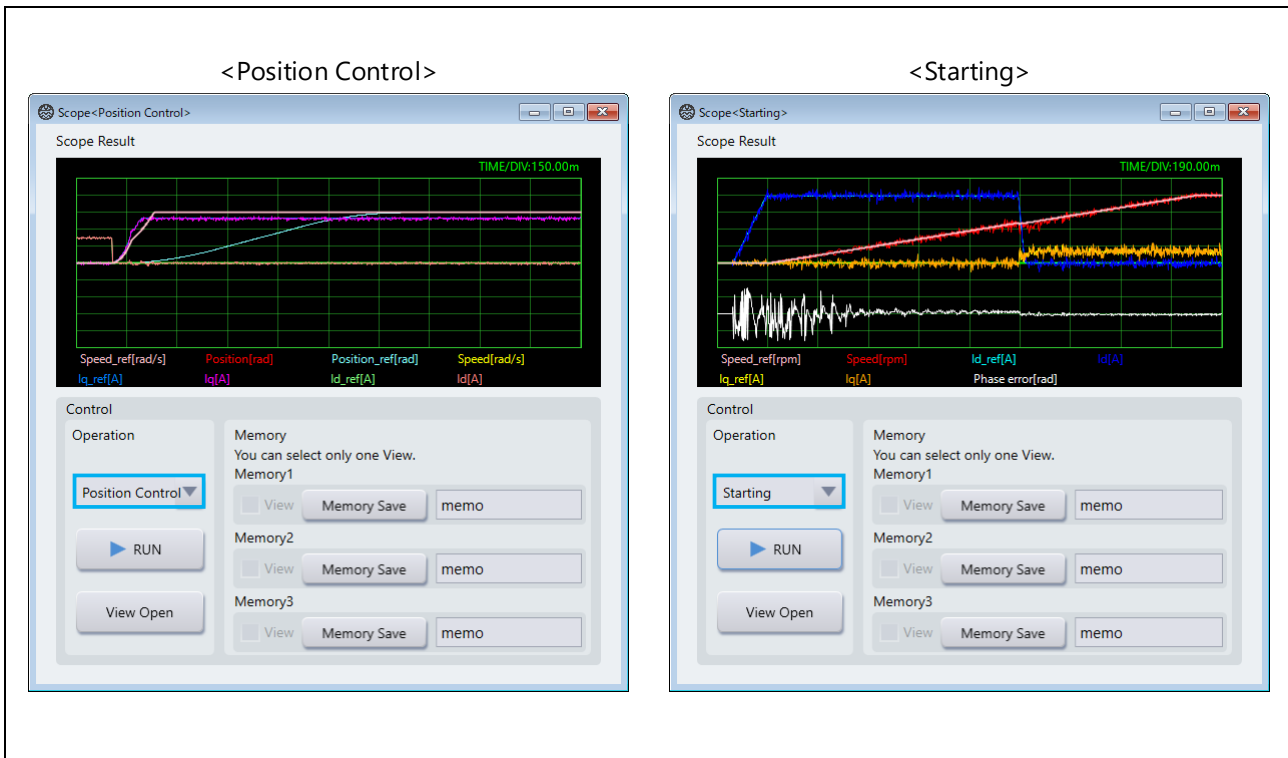


図 14.9 Position Control / Starting

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。

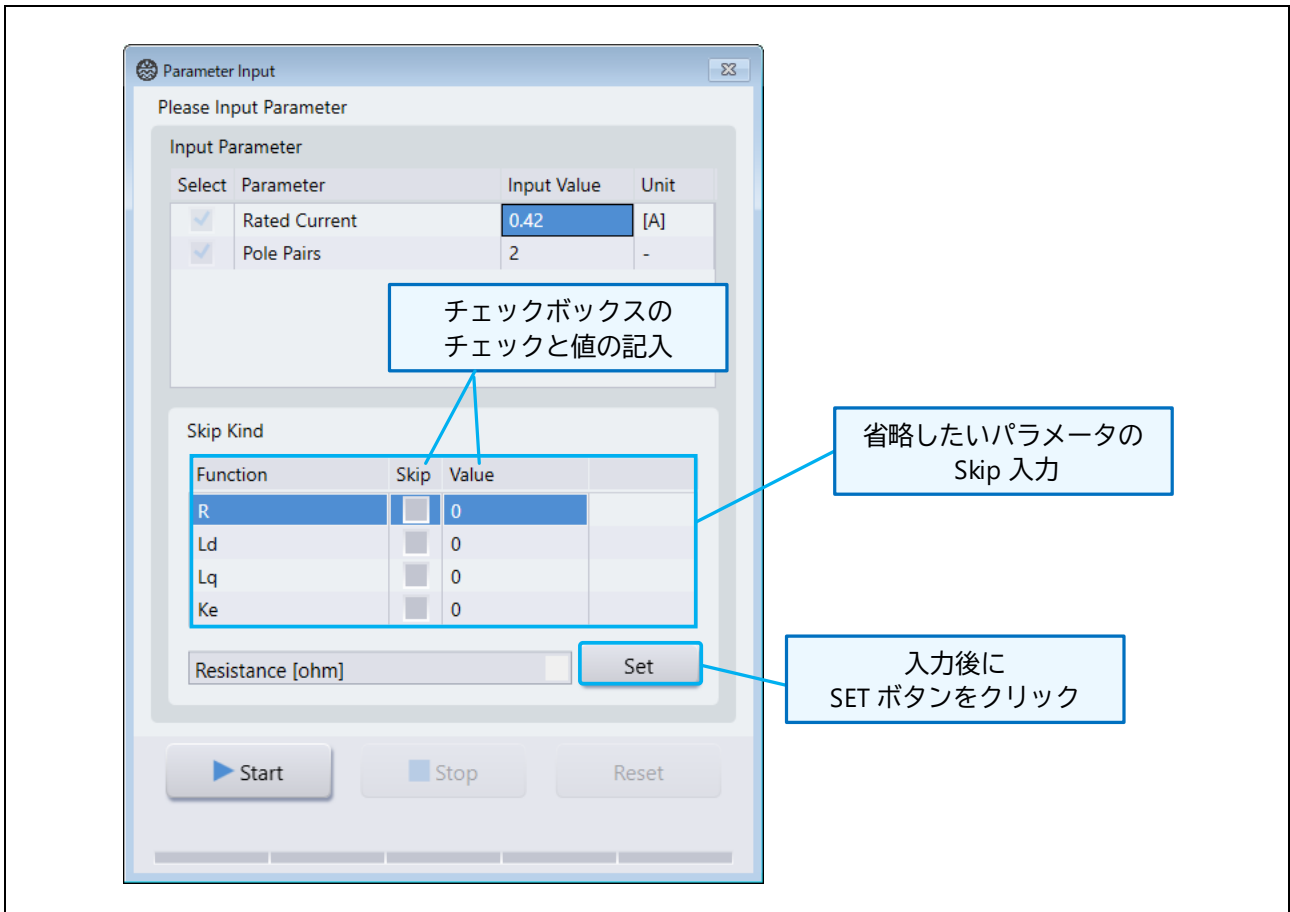


図 14.10 Skip 入力画面

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。

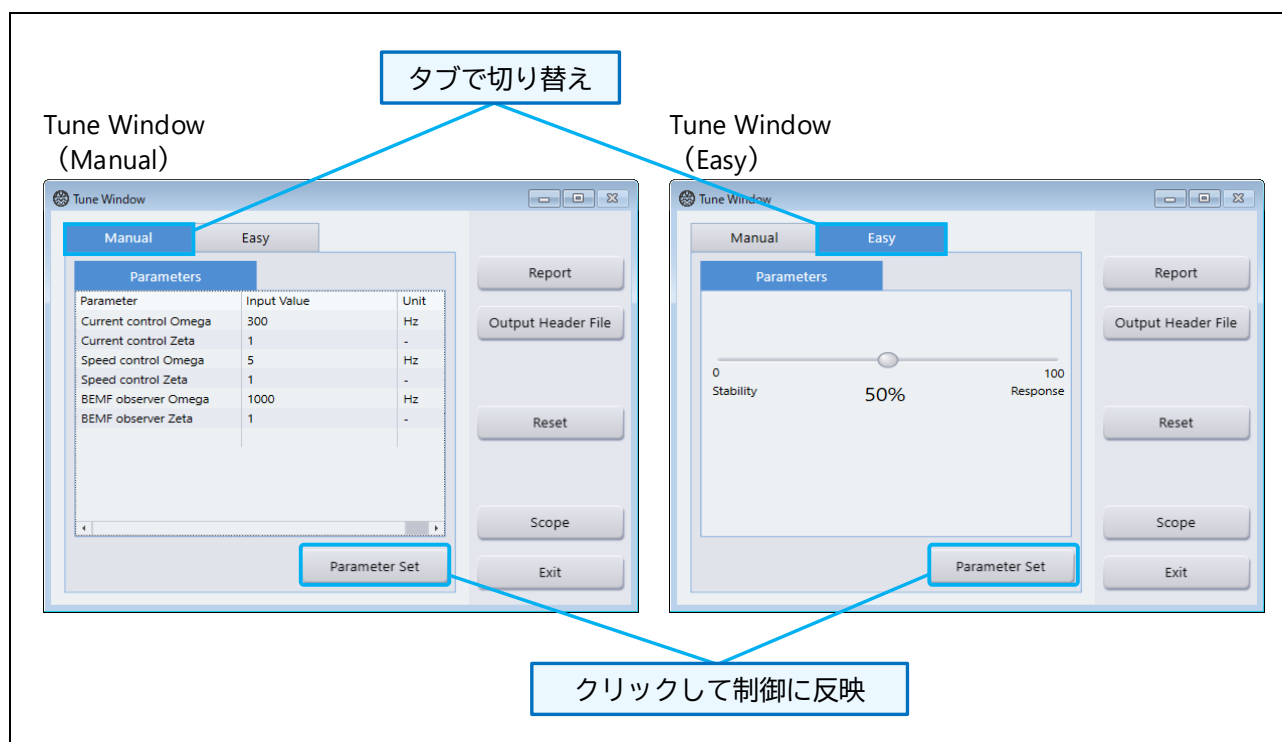


図 14.11 Manual タブと Easy タブ

表 14.1 調整パラメーター一覧 (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

14.5.3 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「Reset」 ボタンをクリックすると選択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。

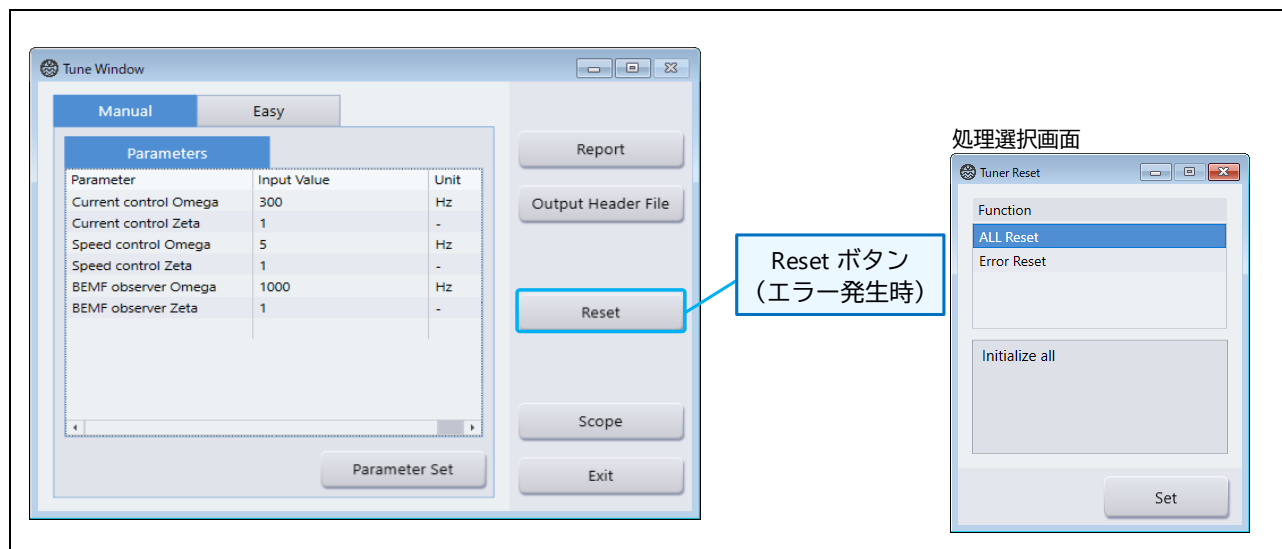


図 14.12 リセット機能

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。

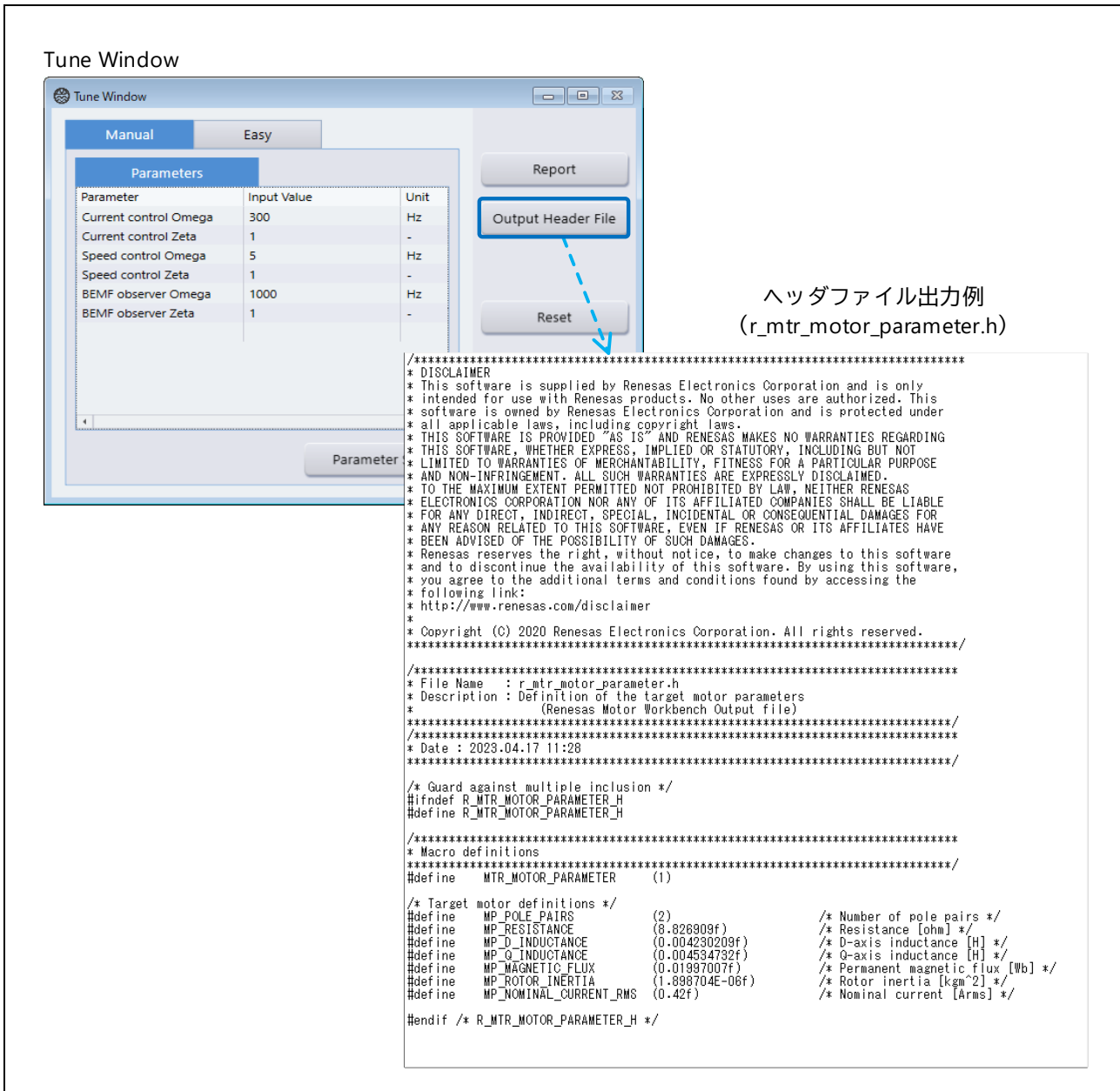


図 14.13 Output Header File ボタン

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。

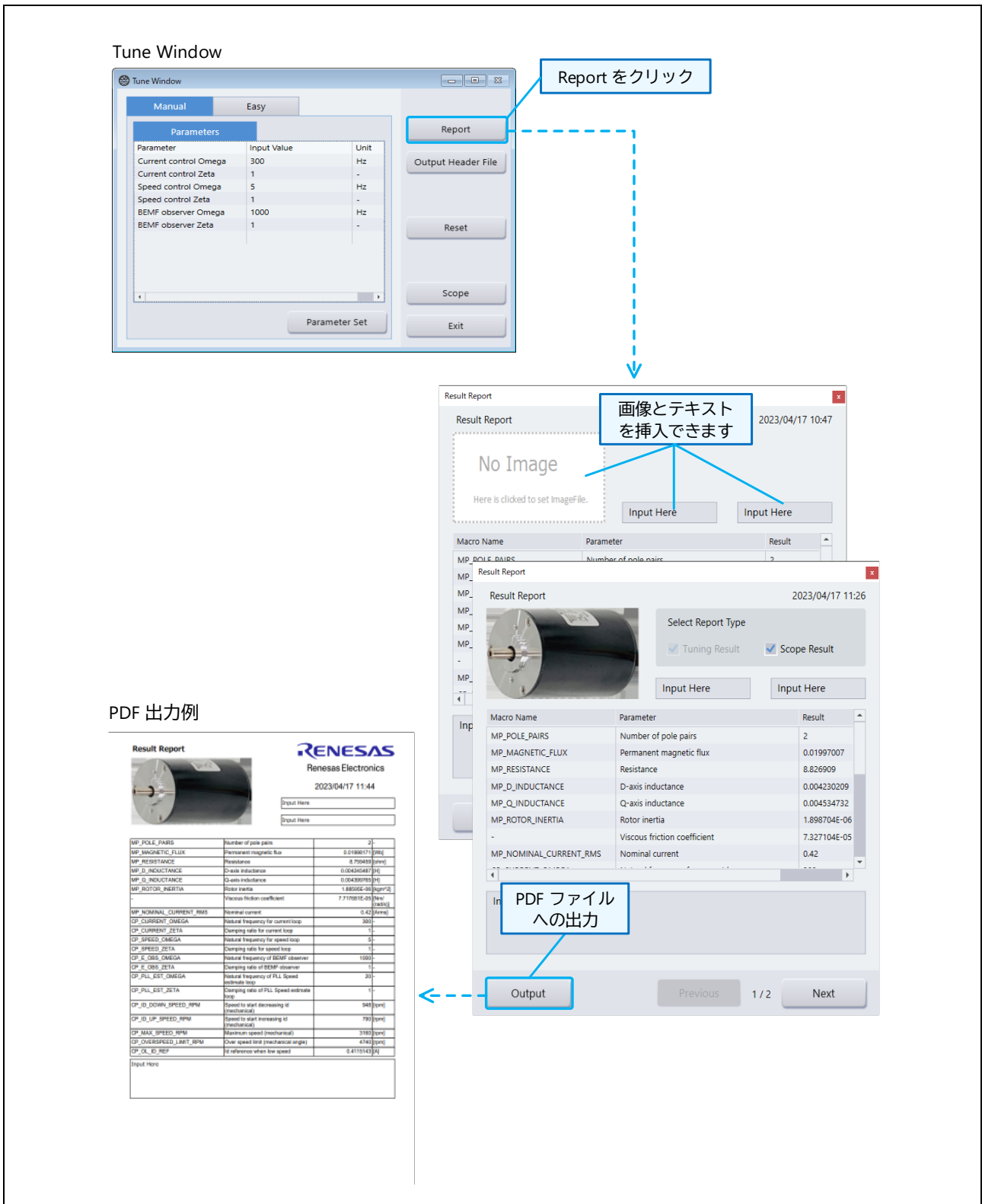


図 14.14 Output ボタン (PDF 出力)

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。

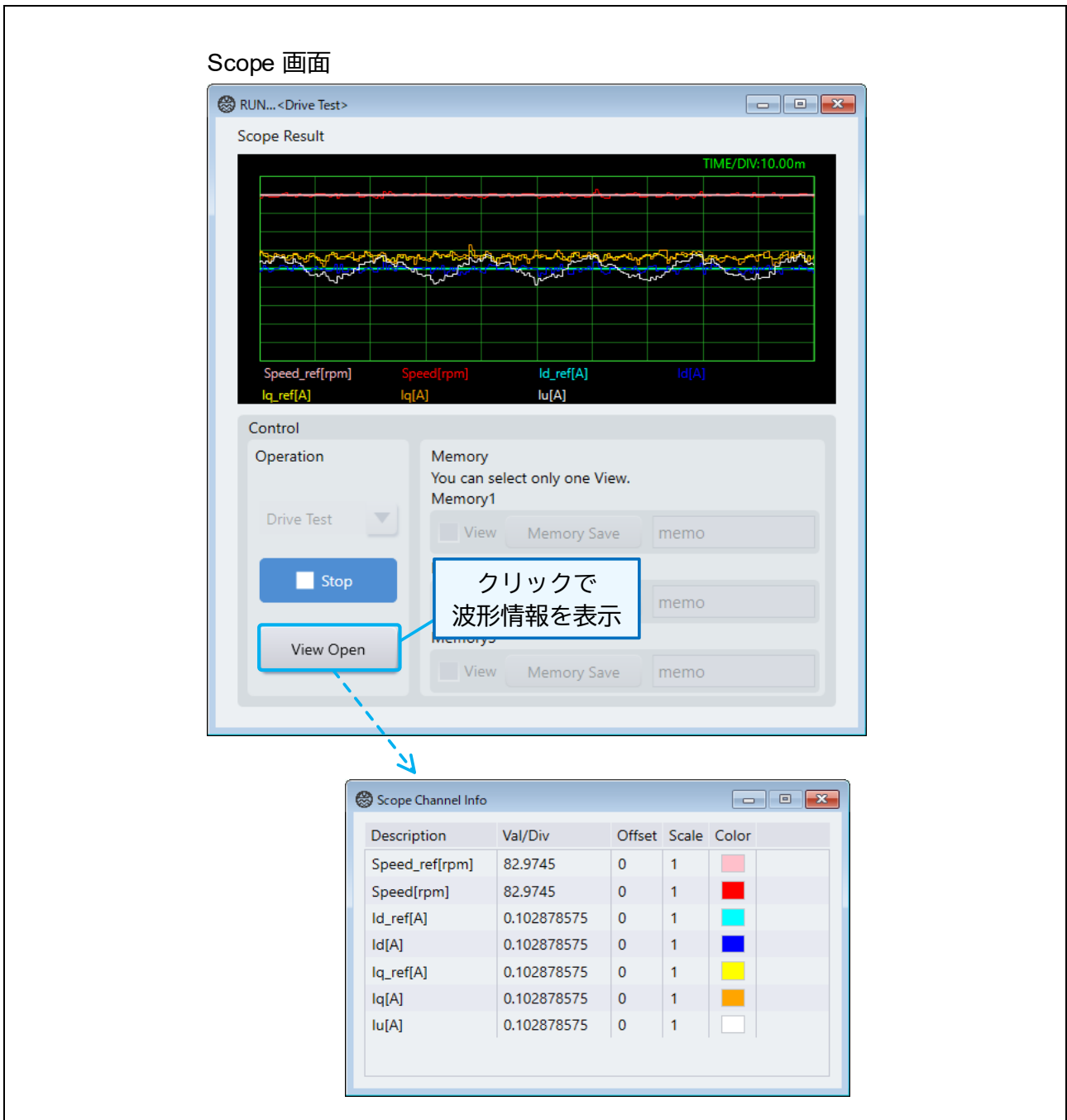


図 14.15 View Open ボタン

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。

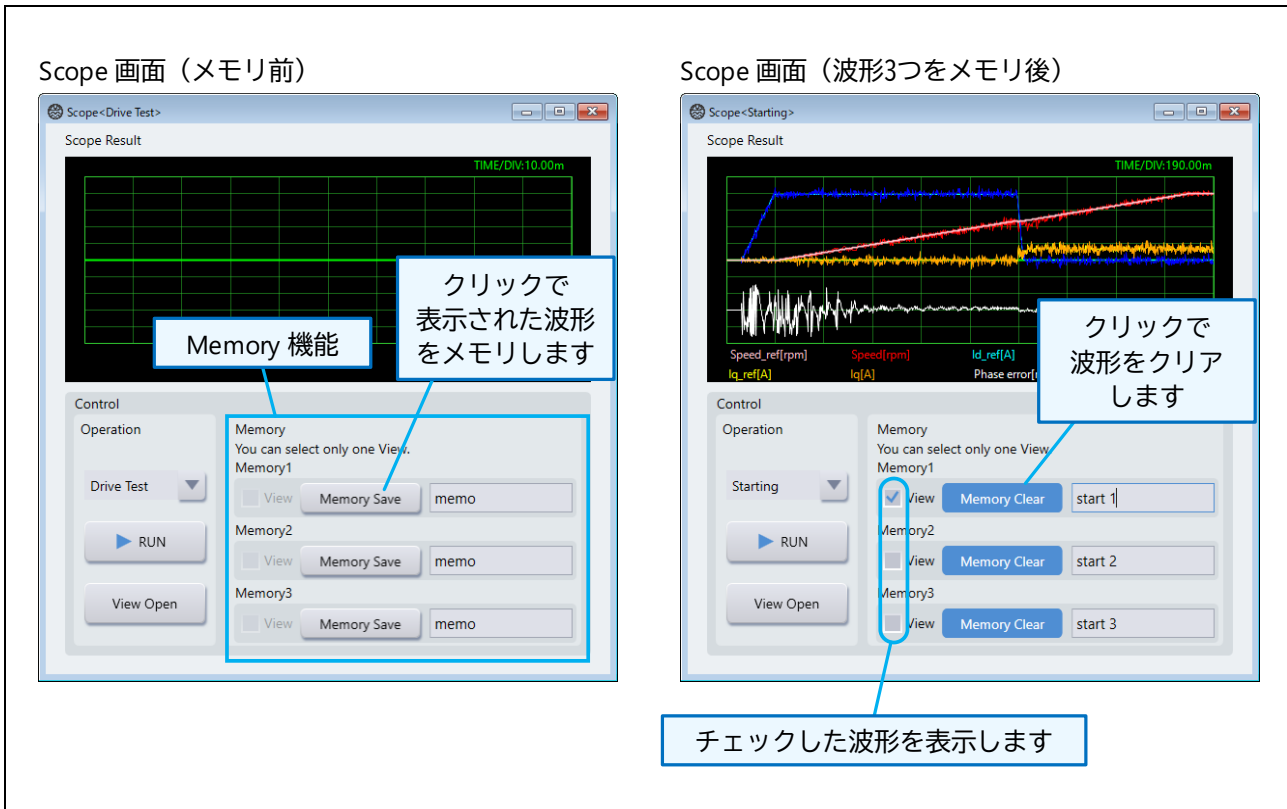


図 14.16 メモリ機能

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。

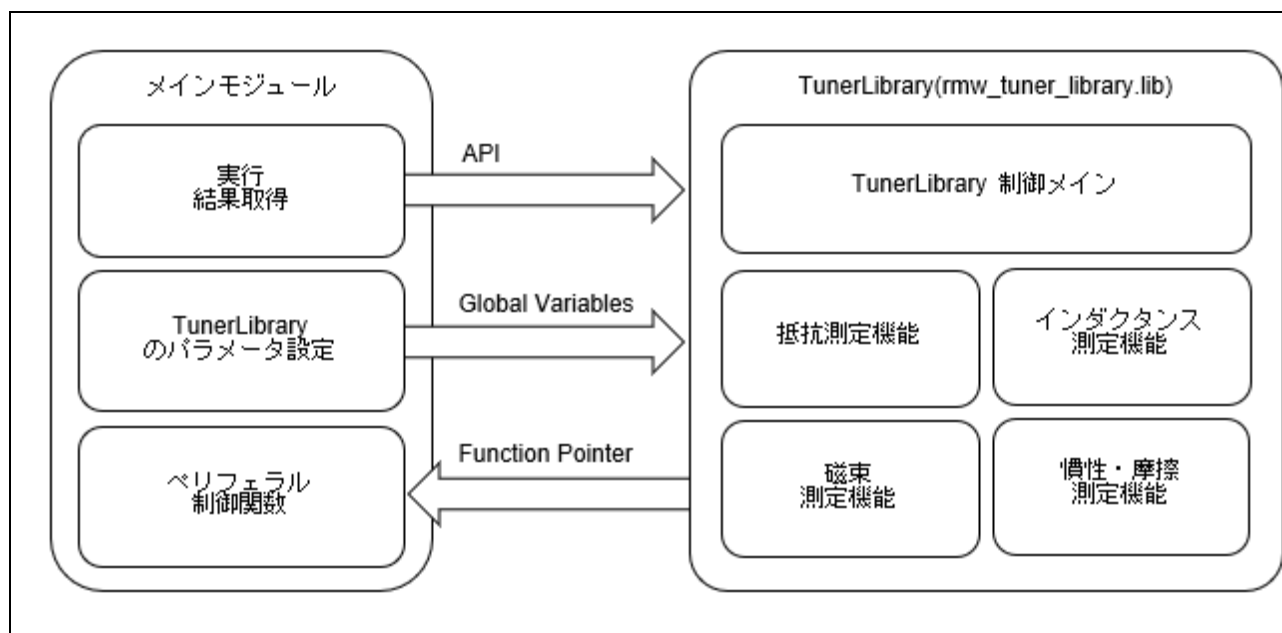


図 14.17 Tuner Library 全体構成

14.6.2 Tuner Library API 一覧

表 14.2 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 14.3 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 処理内部ステータス取得	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 14.4 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 14.5 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 14.6 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 14.7 構造体

型	定義名	ファイル名	概要
struct	<pre>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;</pre>	r_aid_auto_id_notify.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 of 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 14.8 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

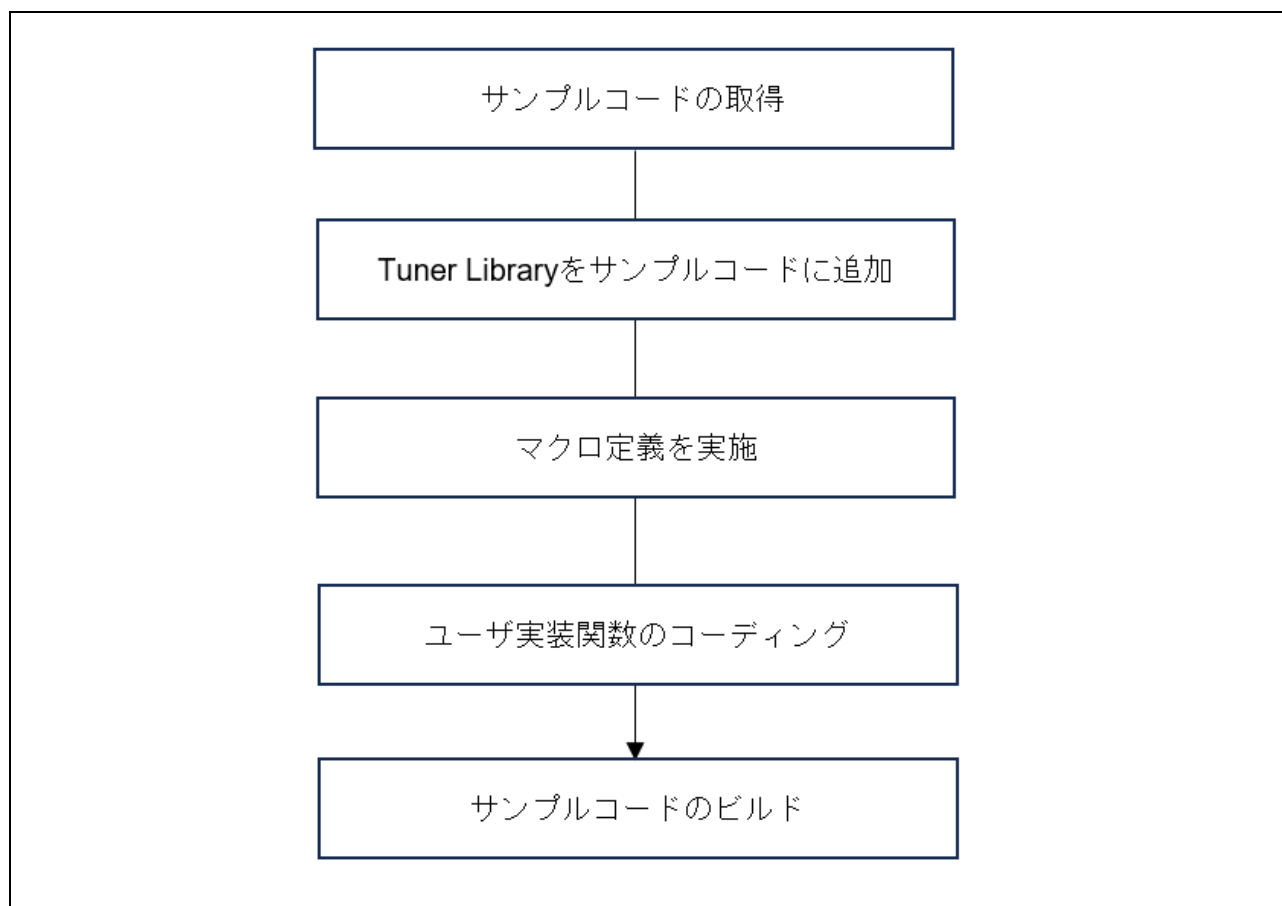


図 14.18 Tuner Library 組み込み手順

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>

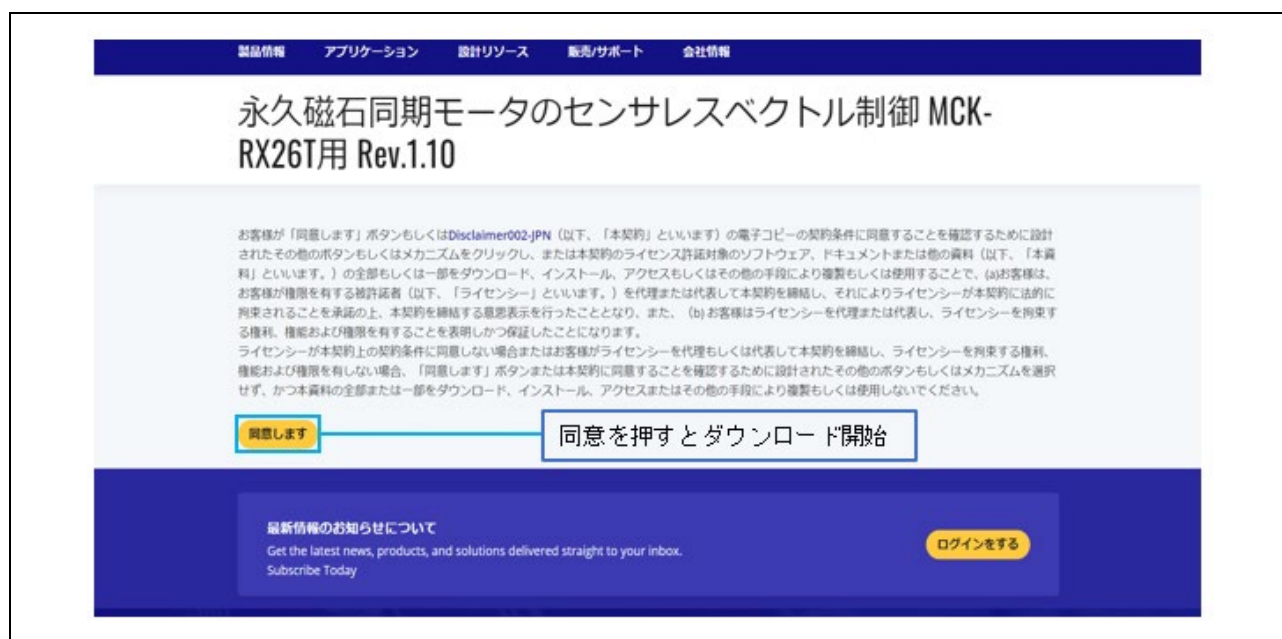


図 14.19 サンプルコード取得

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。

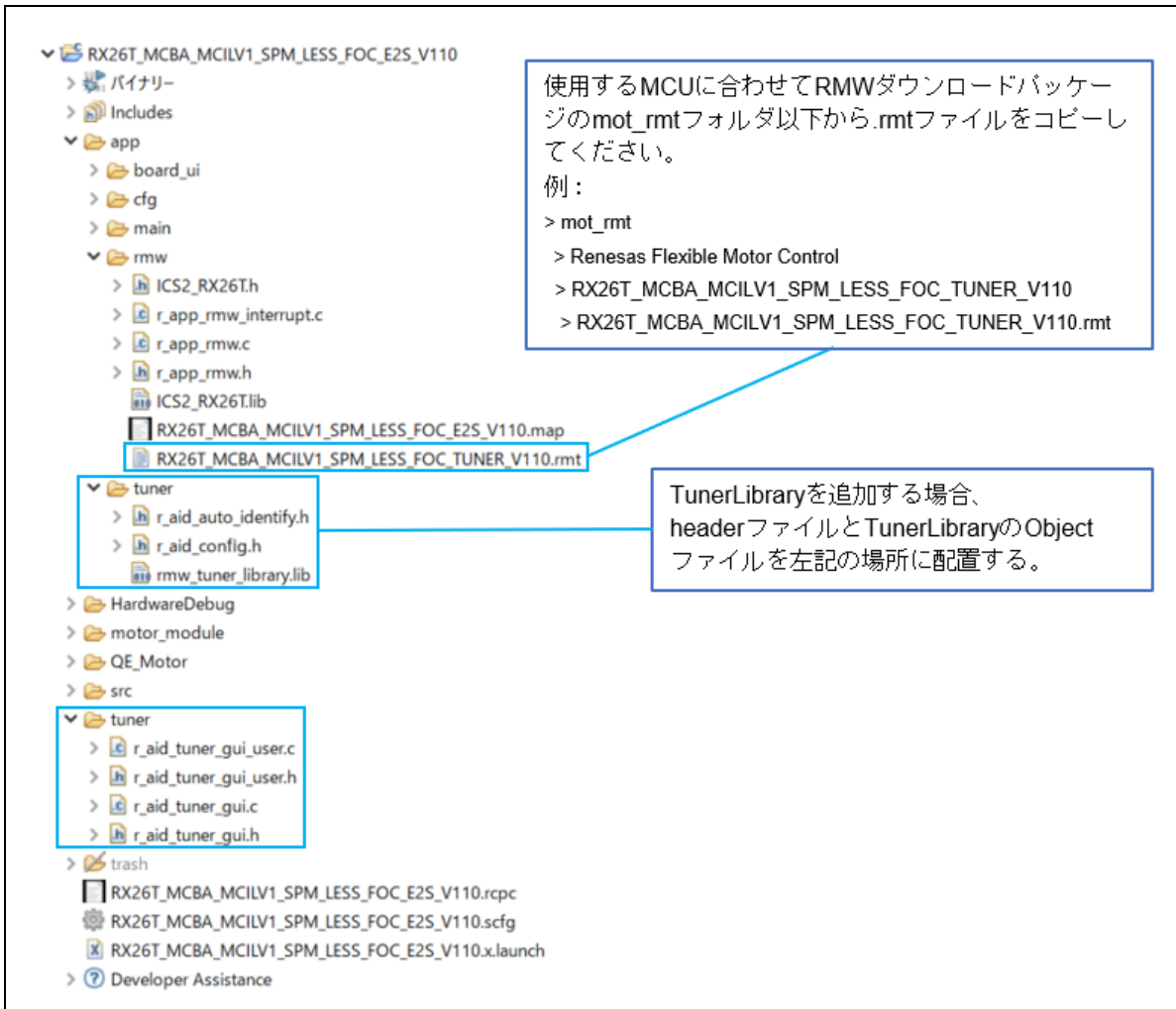


図 14.20 Tuner Library をサンプルプログラムに追加

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

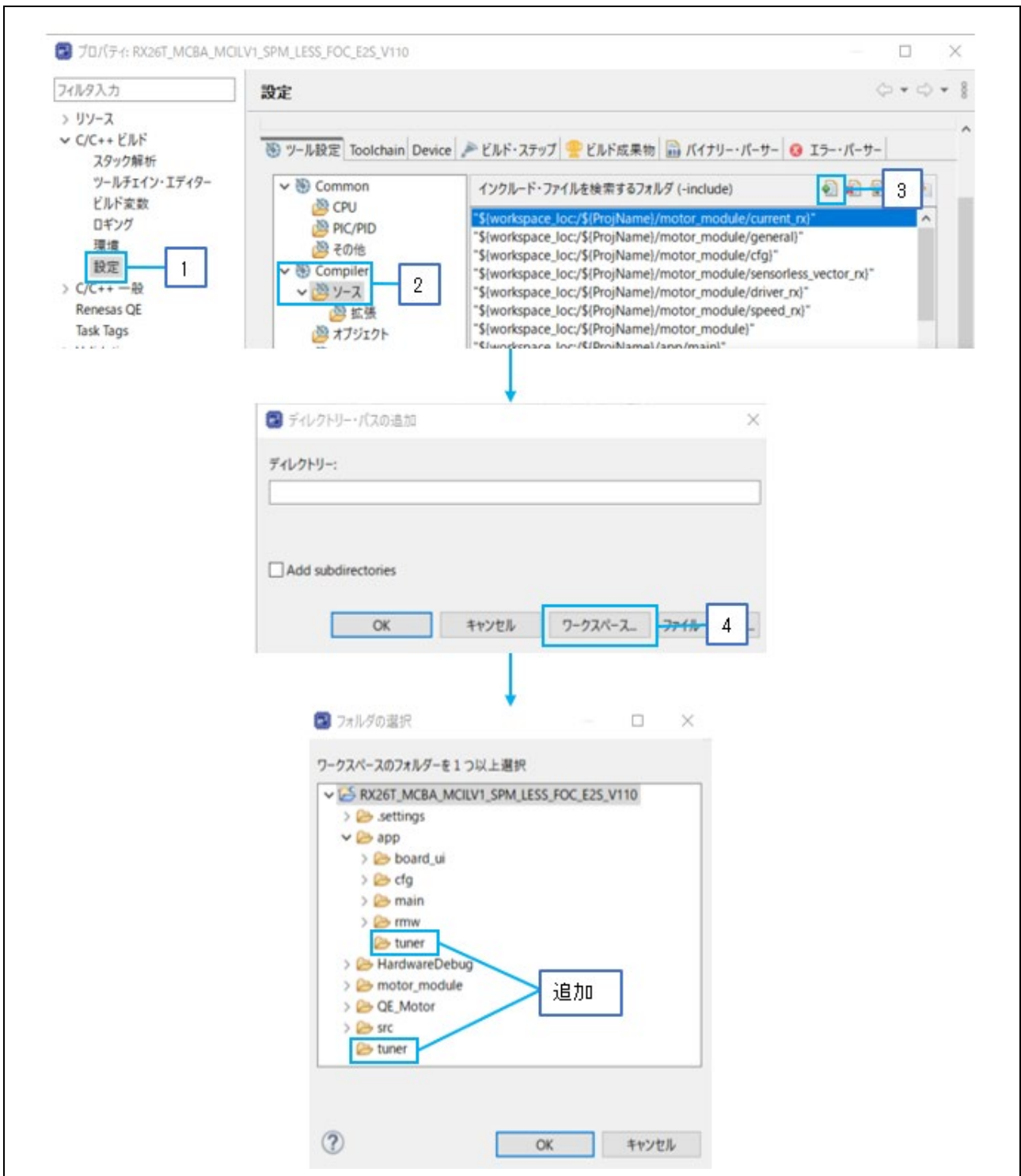


図 14.21 インクルードパスの設定

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。

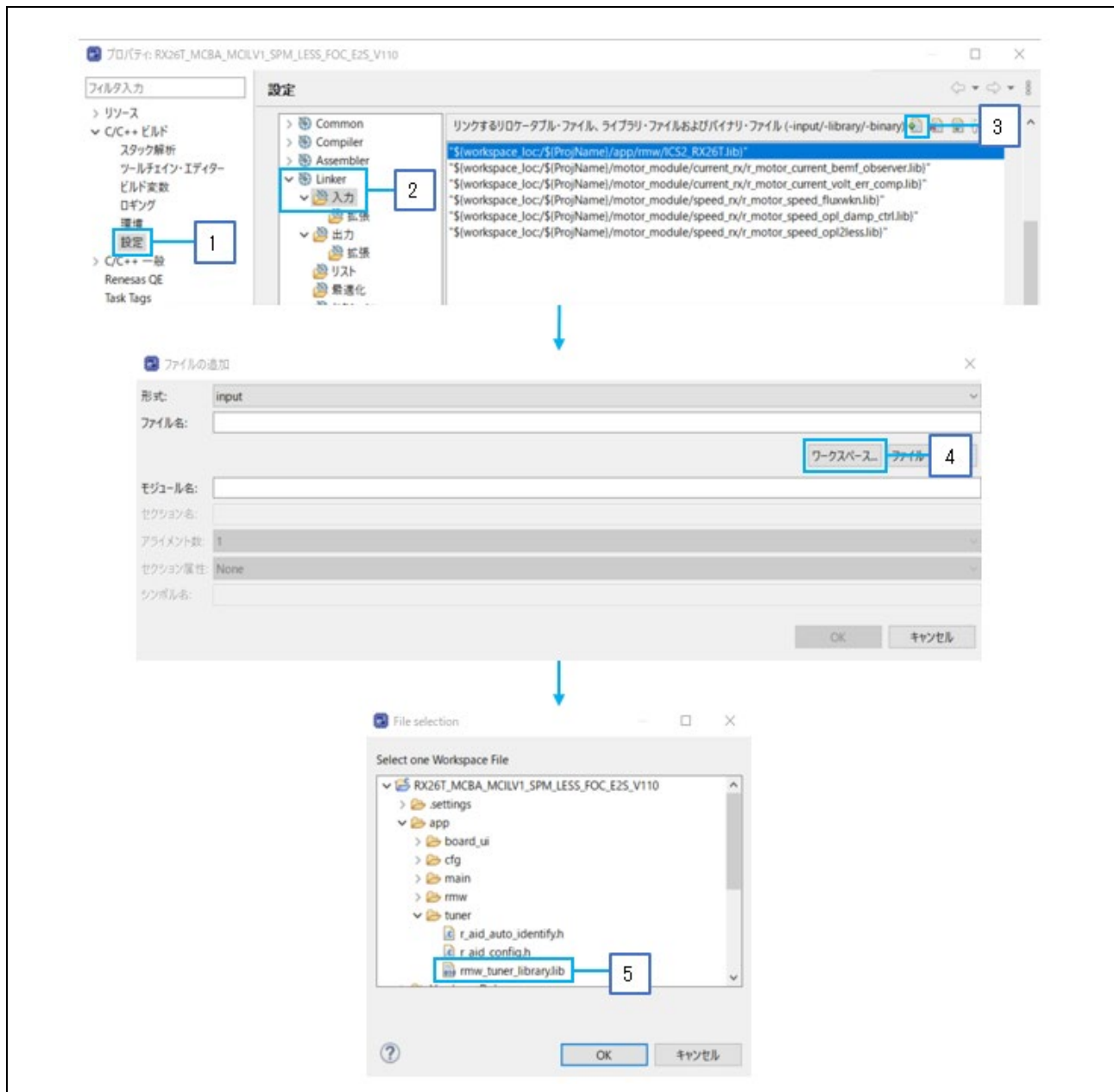


図 14.22 Tuner Library の追加

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。

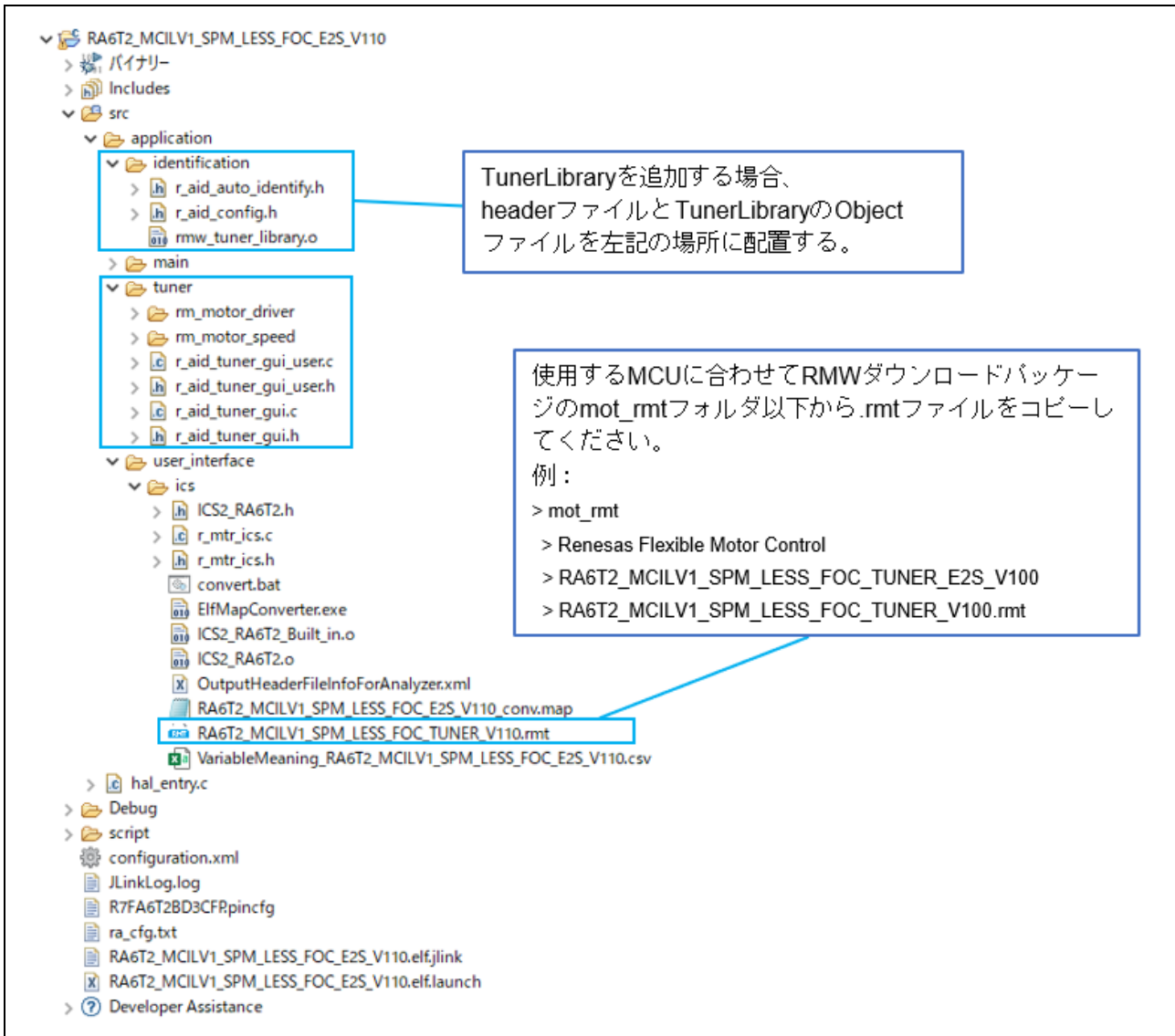


図 14.23 Tuner Library をサンプルプログラムに追加

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

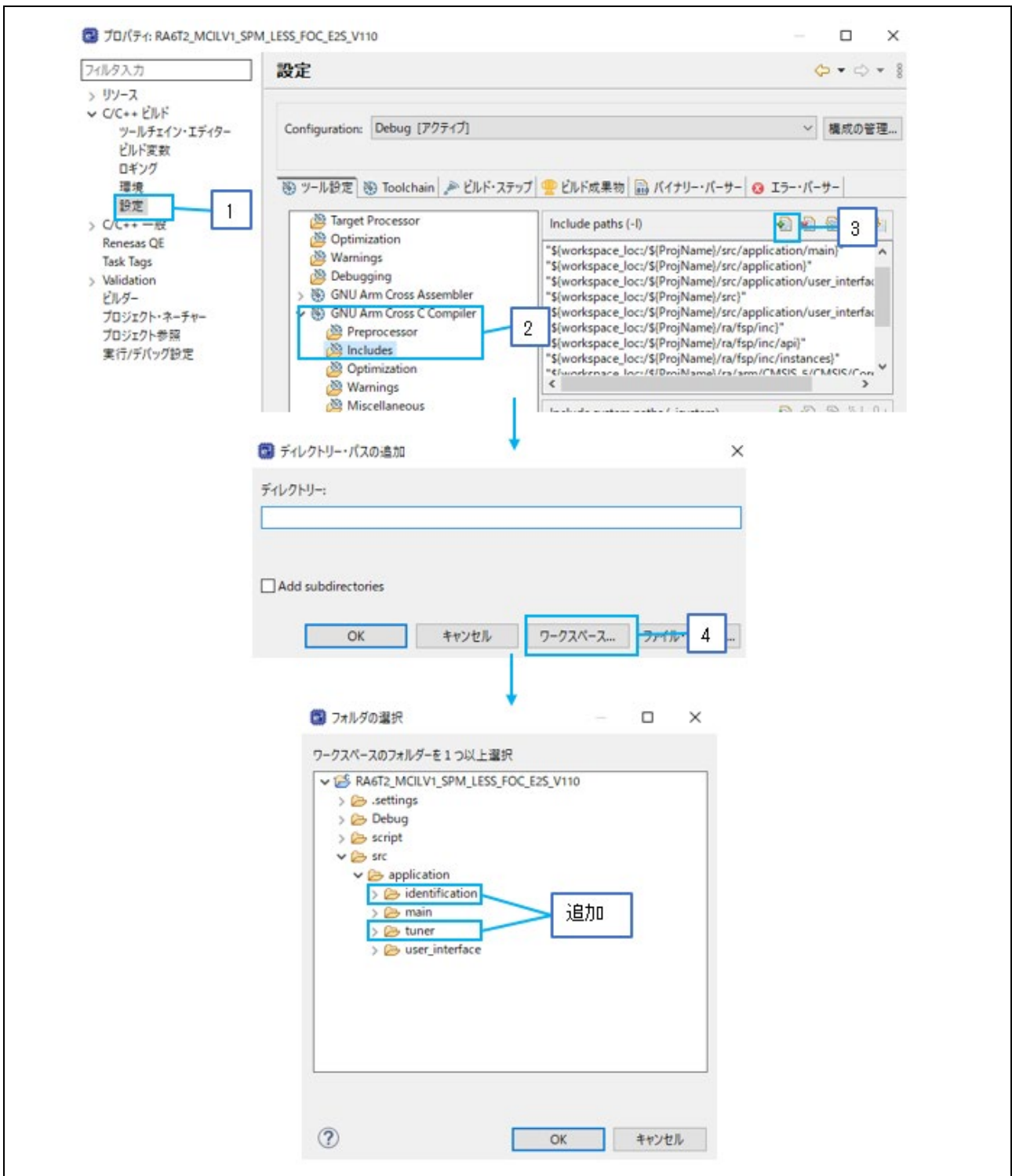


図 14.24 インクルードパスの設定

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。

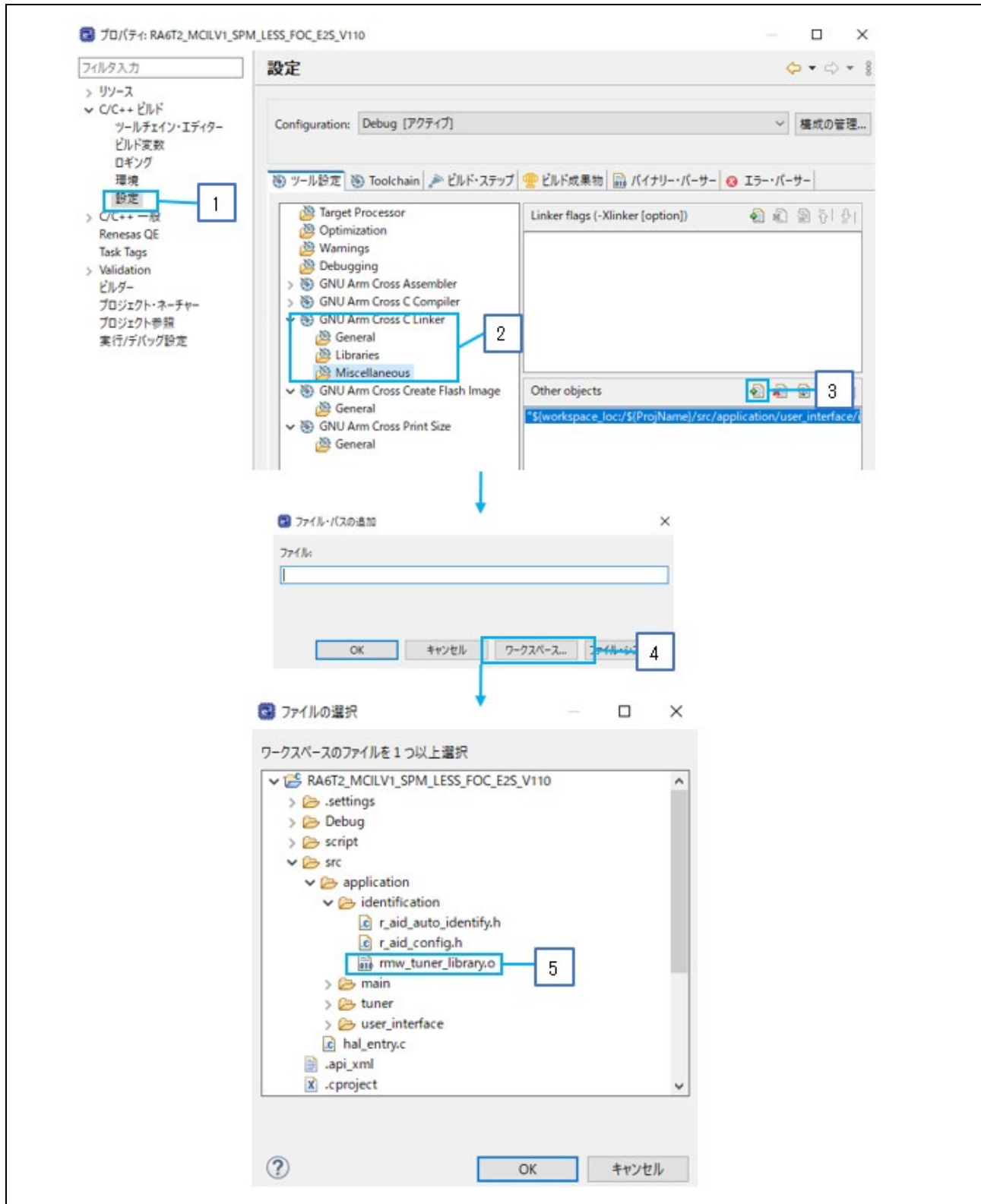


図 14.25 Tuner Library の追加

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 14.9 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]/キャリア周期[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 14.26 [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_RMW_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 14.27 [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_ul_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_ul_sw_userif)
{
    /* Main process for board UI */
    r_app_board_ui_mainloop();
}
else
{
    /* Do Nothing */
}

```

図 14.28 [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_ul_cnt_ics++;

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

        /* Call ICS */
        ics2_watchpoint();
    }

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

* Function Name : r_app_rmw_system mode[]

```

図 14.29 [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 14.30 [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 14.31 [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_oeil_interrupt(void)
{
    /* Start user code for r Config POE oeil 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 14.32 [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。

```

* 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_ul_active_gui;
uint16_t  g_u2_conf_sw_ver;

```

図 14.33 [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

```

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 14.34 [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 14.35 [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 14.36 [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)
        r

```

図 14.37 [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_RM_W_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_RM_W_TUNER */
}
else
{
    .. . . . .
}

```

図 14.38 [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 14.39 [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; /* 000000000001000b */
    /* USE_RPM_TUNER */
    g_u2_conf_sw = 0x4000; /* 0100000000000000b */
    g_u2_conf_tool = 0x0600; /* 0000011000000000b */
    /* USE_RPM_TUNER */

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

図 14.40 [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 14.41 [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 14.10 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求めます。	—
	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 14.11 ユーザ実装関数一覧 (2/6)

関数名	引数	型	概要	関数ポインタ変数名
aid_mtr_get_current_iuiw	float *f4_iu_ad : U-phase current float *f4_iw_ad : W-phase current	void	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。	g_fp_aid_internal_get_current_iuiw
	<pre>void aid_mtr_get_current_iuiw(float *f4_iu_ad, float *f4_iw_ad) { r_mtr_adc_tbst_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; }</pre>			
aid_mtr_get_vdc	—	float : Supply voltage	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。	g_fp_aid_internal_get_vdc
	<pre>float aid_mtr_get_vdc(void) { r_mtr_adc_tbst_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); }</pre>			

表 14.12 ユーザ実装関数一覧 (3/6)

関数名	引数	型	概要	関数ポインタ変数名
aid_mtr_ctrl_start	-	void	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。	g_fp_aid_internal_ctrl_start
			<pre>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); }</pre>	
aid_mtr_ctrl_stop	-	void	ドライブモードから STOP モードに遷移します。	g_fp_aid_internal_ctrl_stop
			<pre>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); }</pre>	
aid_mtr_clear_oc_flag	-	void	過電流検出時の強制遮断フラグをクリアします。	g_fp_aid_internal_clear_oc_flag
			<pre>void aid_mtr_clear_oc_flag(void) { R_Config_POE_Stop(); R_Config_POE_Start(); }</pre>	

表 14.13 ユーザ実装関数一覧 (4/6)

関数名	引数	型	概要	関数ポインタ変数名
rmw_apply_identified_params	—	void	RMW のパラメータを設定します。 <pre>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.f4RatedCurrent; 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; }</pre>	—
rmw_apply_reset	—	void	RMW のパラメータをリセットします。 <pre>void rmw_apply_reset(void) { com_u1_system_mode = STATEMACHINE_EVENT_RESET; }</pre>	—

表 14.14 ユーザ実装関数一覧 (5/6)

関数名	引数	型	概要	関数ポインタ変数名
mtr_ics_1_parameter	-	void	<p>Tune ウィンドウの Easy タブ設定内容を反映します。</p> <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>	-

表 14.15 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	—
	<pre>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); } }</pre>			
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。	—
	<pre>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 */ }</pre>			

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。

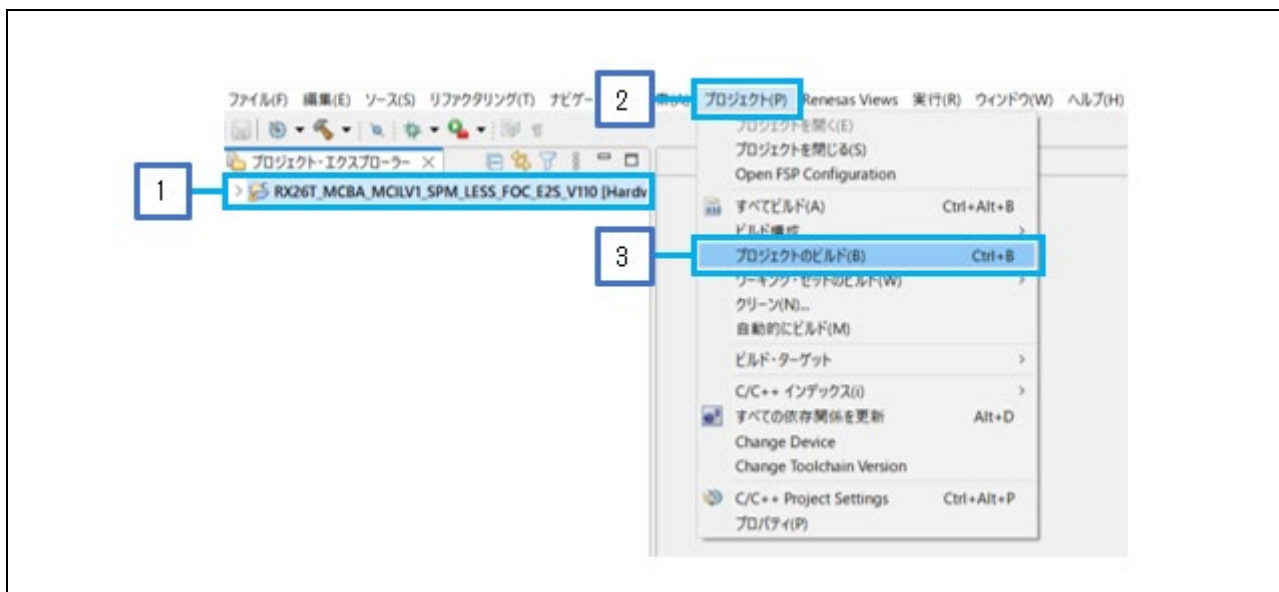


図 14.42 サンプルコードのビルド

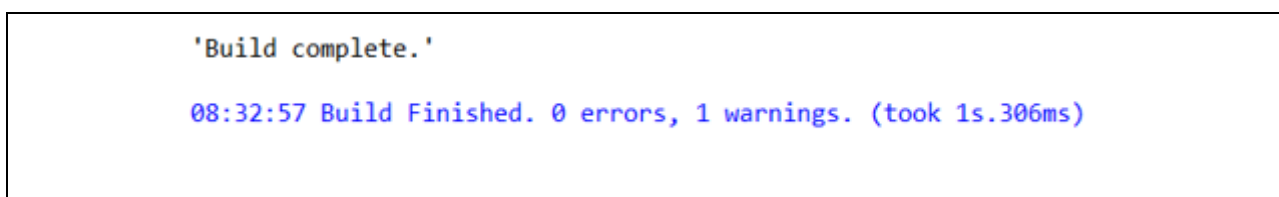


図 14.43 サンプルコードのビルド結果

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. ビルトイン型通信ライブラリ

15.1 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。

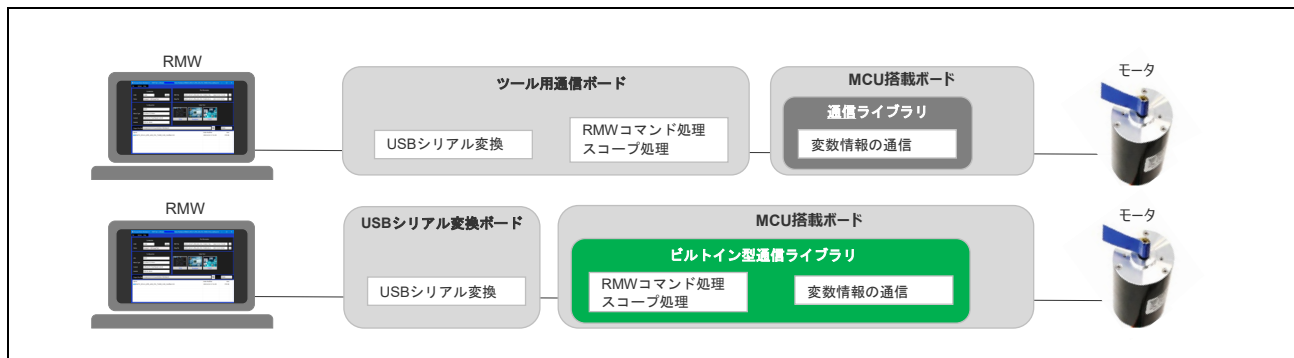


図 15.1 ライブラリの概要

15.2 特徴

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 15.1 通信ライブラリとビルトイン型通信ライブラリの違い

項目		通信ライブラリ	ビルトイン型通信ライブラリ
推奨用途		高速なサンプリングデータの表示	簡易的な表示やデバッグ
対応 MCU	RX ファミリ	RX13T, RX23T, RX24T, RX24U RX26T, RX66T, RX72T, RX72M	—
	RL78 ファミリ	RL78/G14, RL78/G1F	—
	RA ファミリ	RA6T1, RA6T2, RA6T3, RA4T1, RA8T1	RA6T2, RA6T3, RA4T1, RA8T1
必要な通信ボード		MC-COM	市販の USB シリアル変換ボード
スコープウィンドウ表示データ数		100,000	<ul style="list-style-type: none"> • 1,024 (RA6T2, RA8T1) • 512 (RA6T3, RA4T1)
サンプリング周期		Max. 20us/4ch ※変数情報を通信ボードと送受信	任意の周期 ※変数情報をモータ制御 MCU の RAM 領域に保存
使用可能機能		全機能	全機能
スコープウィンドウ用 RAM 容量		—	<ul style="list-style-type: none"> • 32KB (RA6T2, RA8T1) • 8KB (RA6T3, RA4T1)
ライブラリの役割		変数情報を PC - MCU 間で通信	変数情報を PC - MCU 間で通信 RMW コマンド処理、スコープ処理
ライブラリの仕様		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 <ul style="list-style-type: none"> • 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 	

15.3 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。

15.3.1 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 12-2 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

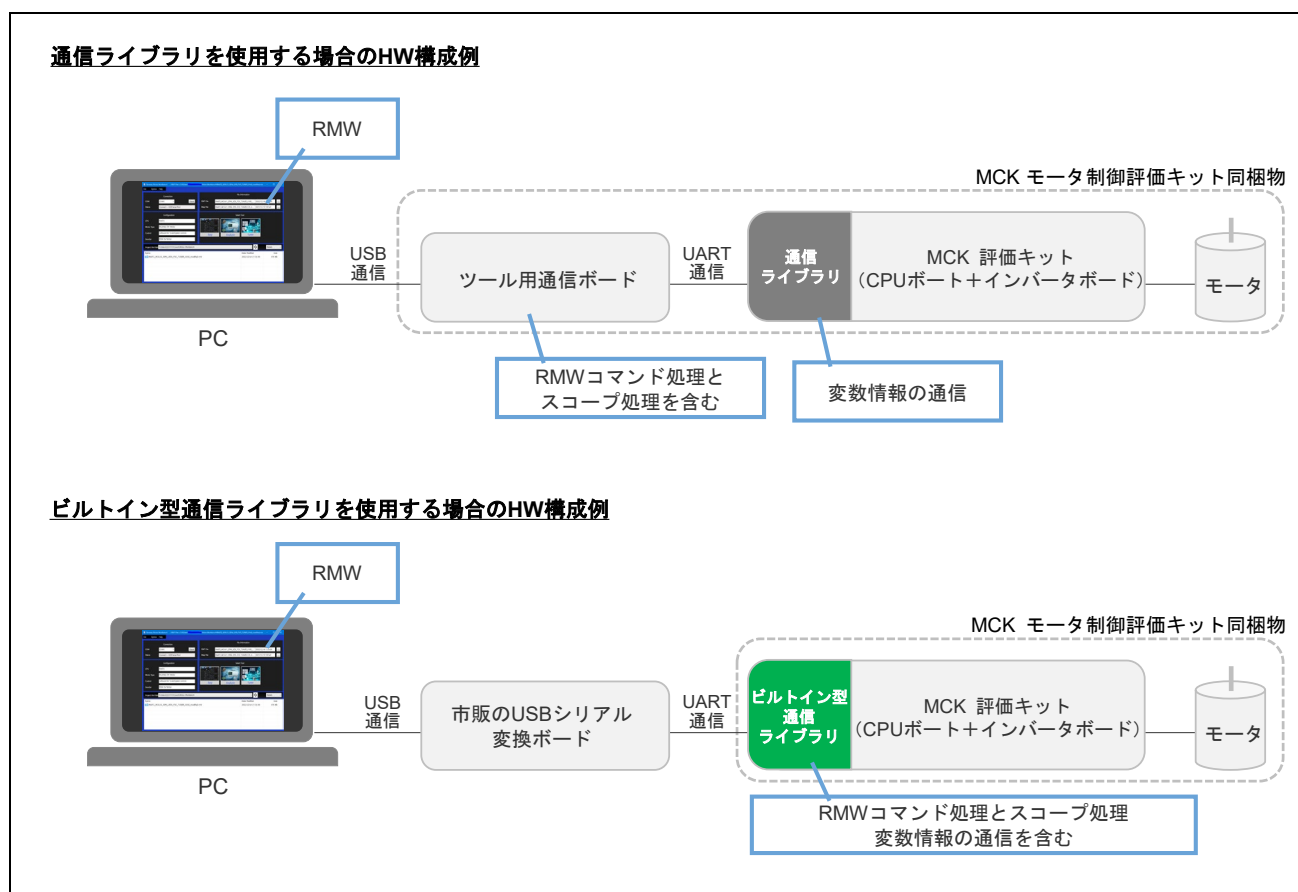


図 15.2 MCK モータ制御評価キットを使用する場合の HW 構成例

15.3.2 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 15-3 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

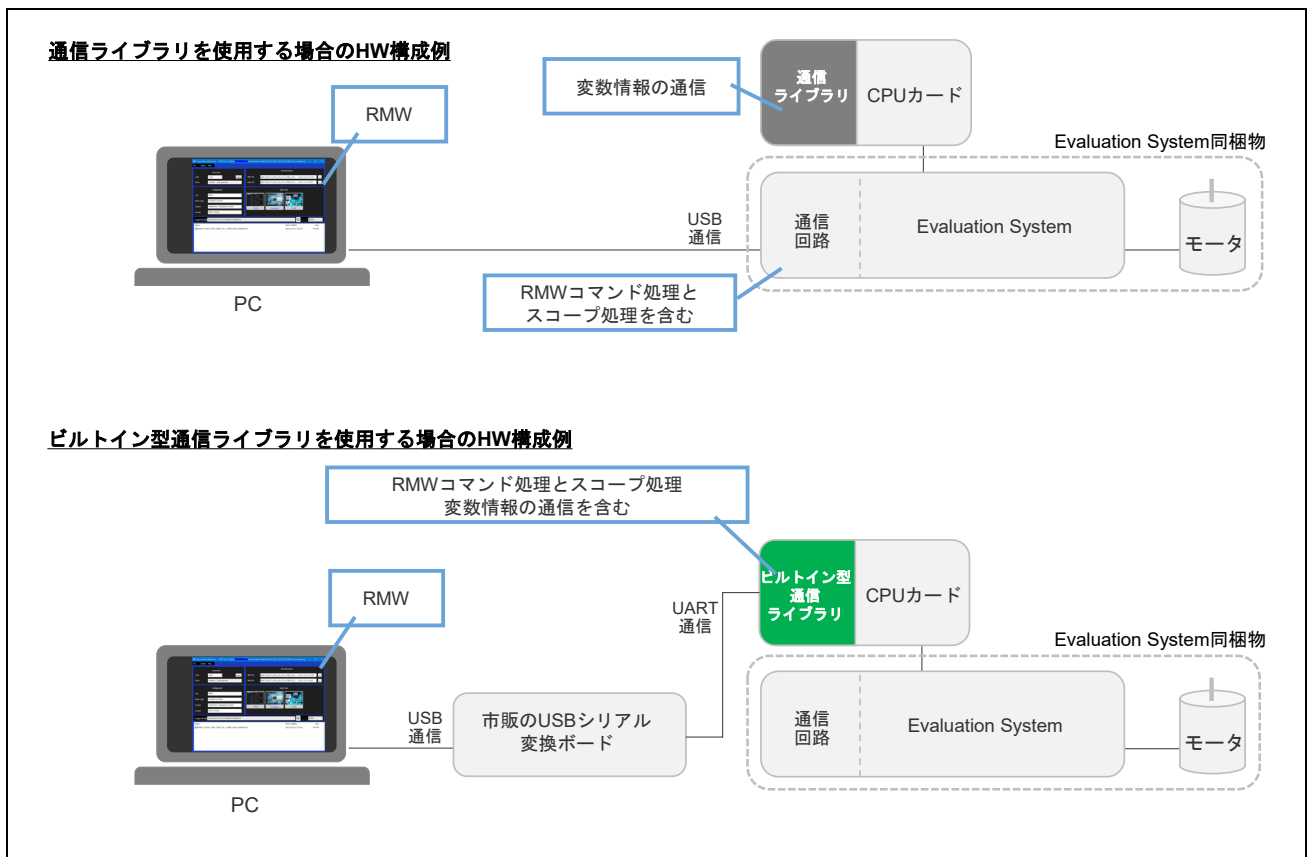


図 15.3 Evaluation System を使用する場合の HW 構成例

15.3.3 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 15-4 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

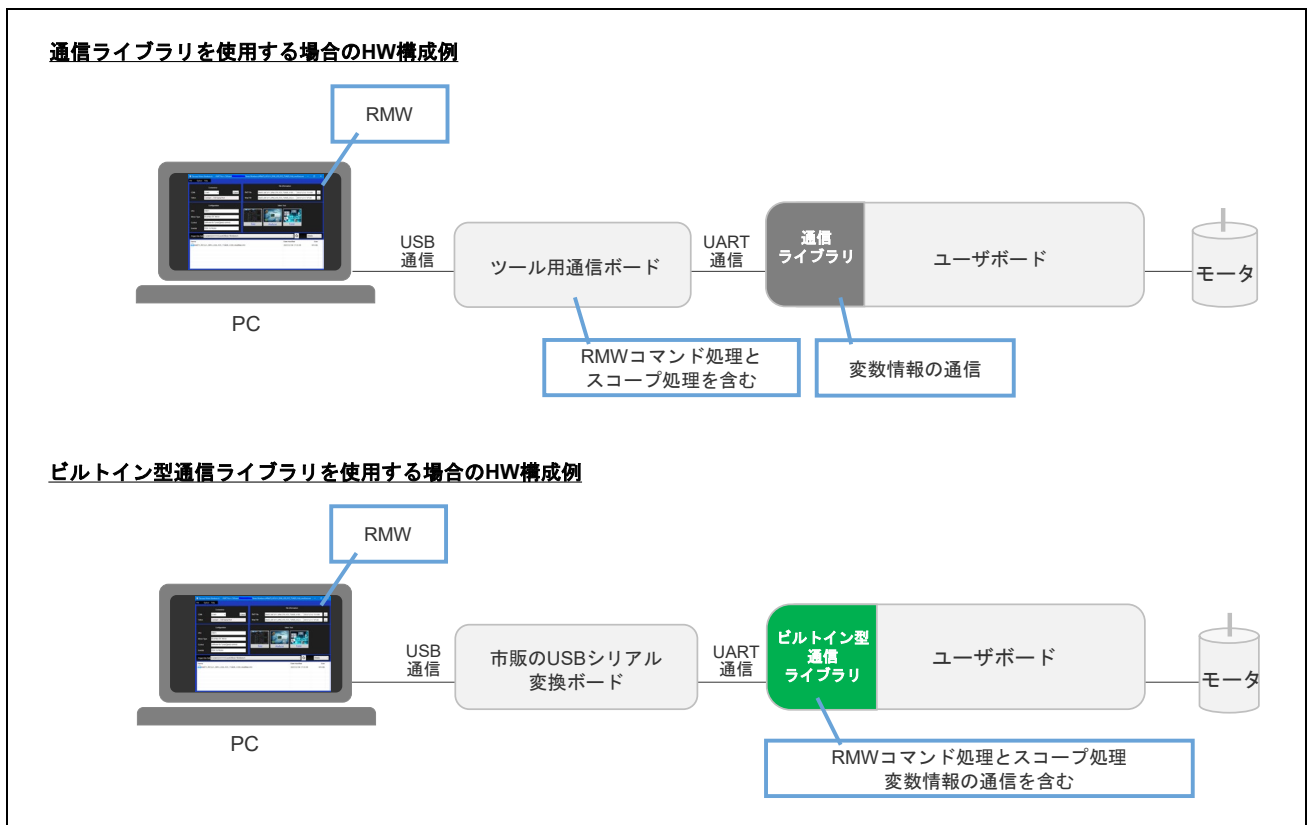


図 15.4 ユーザボードを使用する場合の HW 構成例

15.4 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。

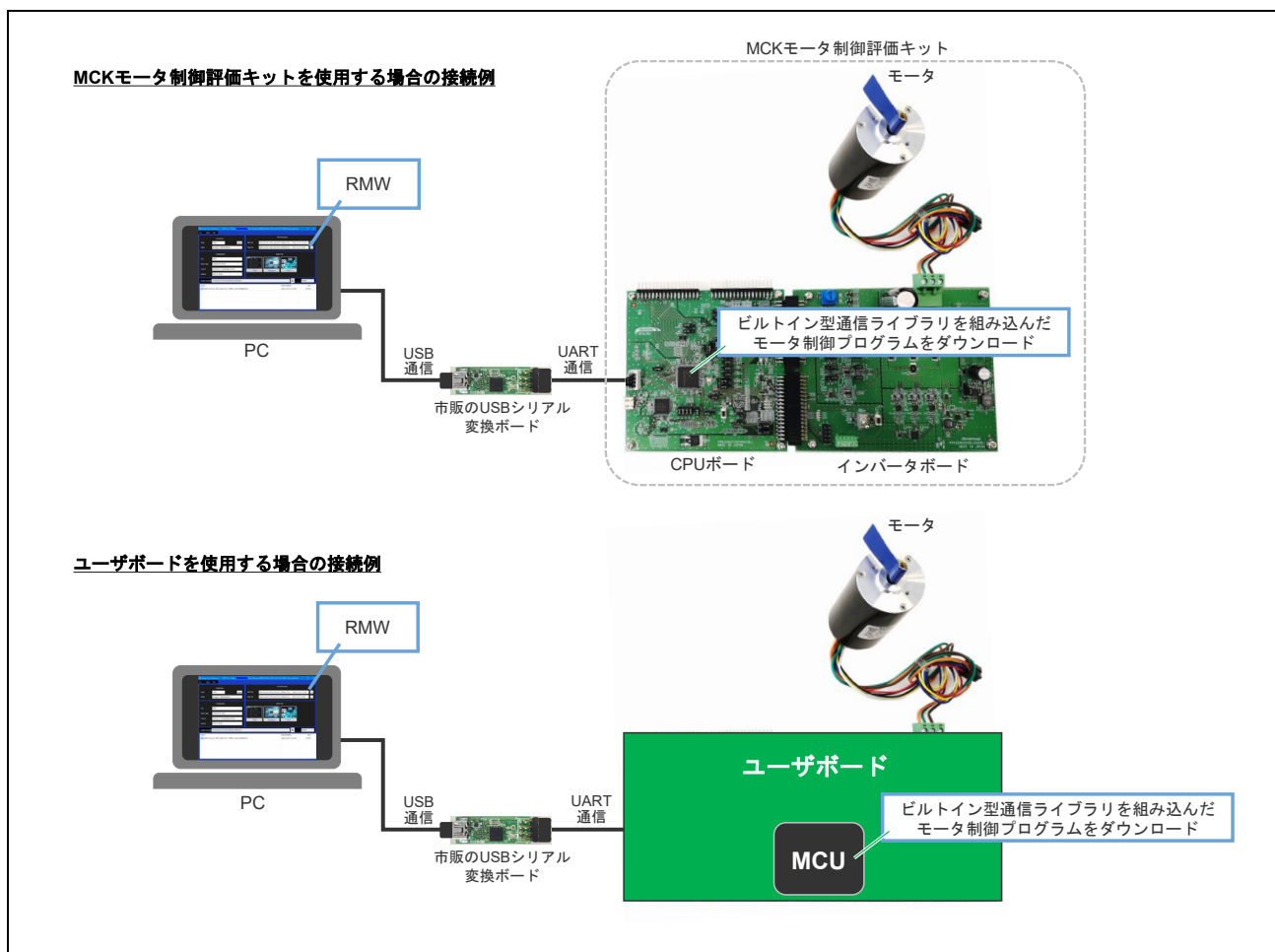


図 15.5 ビルトイン型通信ライブラリを取り込んで RMW を使用する場合は、MCK モータ制御評価キットを使用する場合と同等です。

15.4.1 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



図 15.6 使用機材

15.4.2 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 15-7 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。

- 15.4.4.3 章 ライブラリ関数の呼び出し : ics2_init()の引数 port に設定
- 15.4.6 章 RMW と接続 : USB シリアル変換ボードとの接続

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	-

対応 CPU	RA6T1	RA6T2
ファイル	ICS2_RA6T1.o ICS2_RA6T1.h	ICS2_RA6T2.o ICS2_RA6T2_Built_in.o ICS2_RA6T2.h
通信レート	0.5 [Mbps] ~ 7.5 [Mbps]	0.5 [Mbps] ~ 10.0 [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・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

Renesas Motor Workbenchユーザーズマニュアルから抜粋

ビルトイン型通信ライブラリで使用できる端子

USBシリアル変換ボードと接続できる端子

図 15.7 使用端子の決定

15.4.3 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 15-8 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。

- 15.4.4.3 章 ライブラリ関数の呼び出し : ics2_init()の引数 speed に設定
- 15.4.6 章 RMW と接続 : RMW に通信速度を設定

対応 CPU	RA6T1	RA6T2
ファイル	ICS2_RA6T1.o ICS2_RA6T1.h	ICS2_RA6T2.o ICS2_RA6T2_Built_in.o ICS2_RA6T2.h
通信レート	0.5 [Mbps] ~ 7.5 [Mbps]	0.5 [Mbps] ~ 10.0 [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
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	SClX RX、SClX TX	SClX RX、SClX TX

Renesas Motor Workbenchユーザーズマニュアルから抜粋

図 15.8 RMW で設定可能な通信速度の確認

15.4.4 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、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。

ビルトイン型通信ライブラリについての詳細は、下記ドキュメントを参照してください。

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

15.4.4.1 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 15-9 に示します。

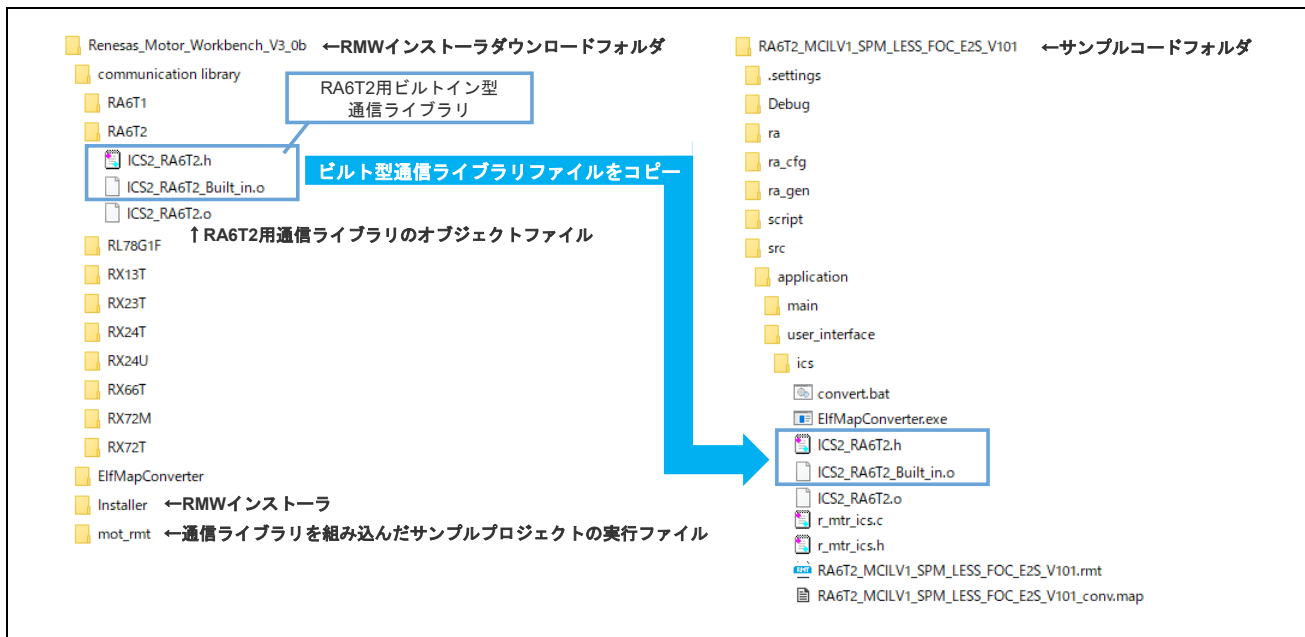


図 15.9 ライブラリファイルをコピーしたフォルダ構成例

15.4.4.2 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 15-10 のとおり、ヘッダファイルパスを 図 15-11 のとおり追加します。

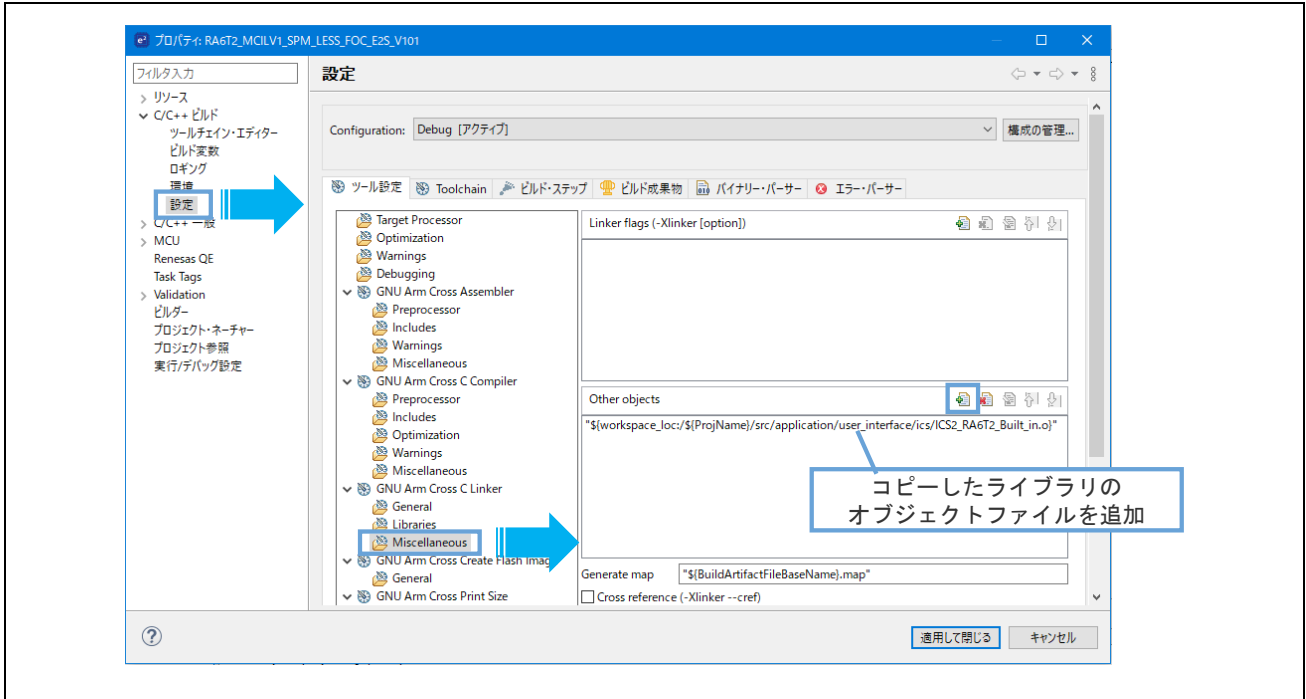


図 15.10 e² studio でライブラリのオブジェクトファイルを追加

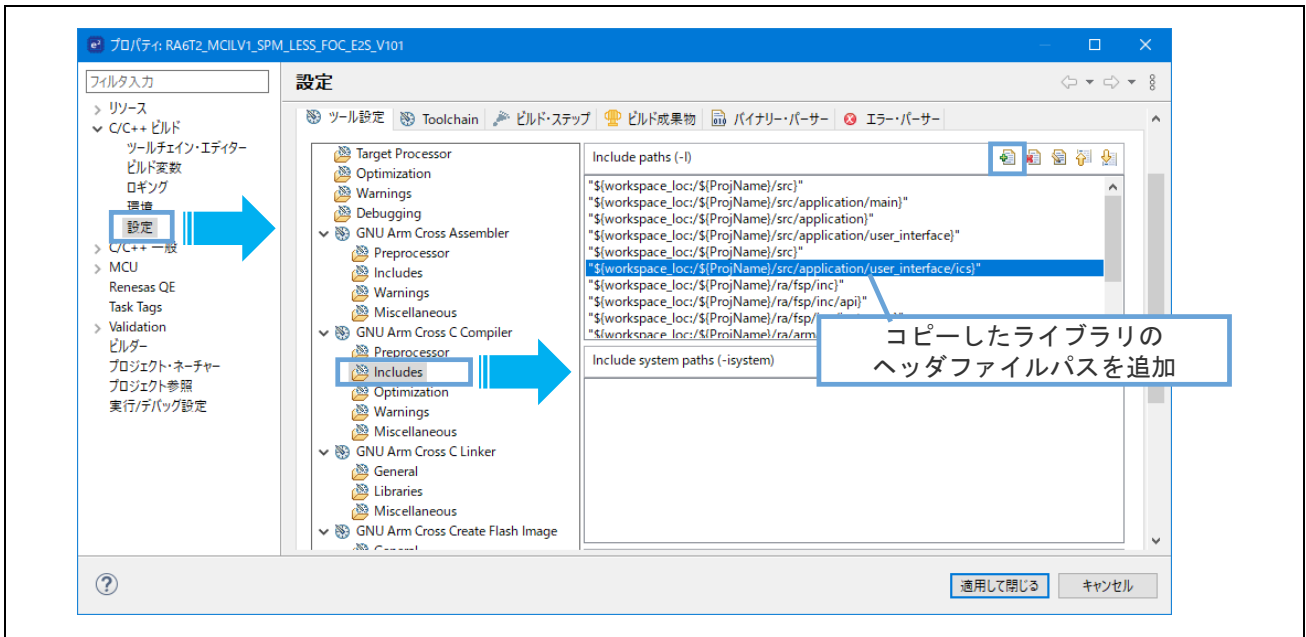


図 15.11 e² studio でライブラリのヘッダファイルパスを追加

15.4.4.3 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、15.4.2 章で決定した使用端子と、15.4.3 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, SCISPICKL = 120MHz -> speed = 21 */
    /* mode = 32bit x 8ch */
    ics2_init(ICS_SCI9_PD05_PD06, 21, 1);

    test();
}

/* 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 timer_isr(void)
{
    ics2_watchpoint();
}
```

15.4.5 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。

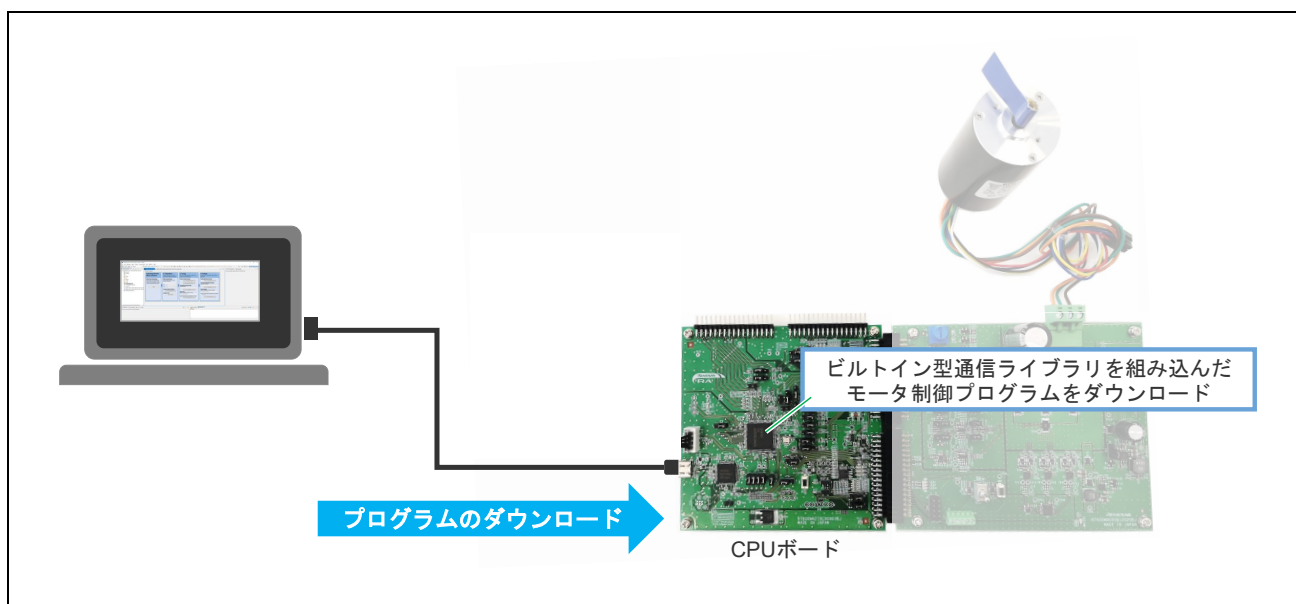


図 15.12 プログラムダウンロード時の接続

15.4.6 RMW と接続

RMW をインストールした PC と各ボードを図 15-13 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、15.4.2 章で決定した端子を使用します。

RMW の通信速度を図 15-14 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 15.4.3 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 の使用方法については、下記ドキュメントを参照してください。

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

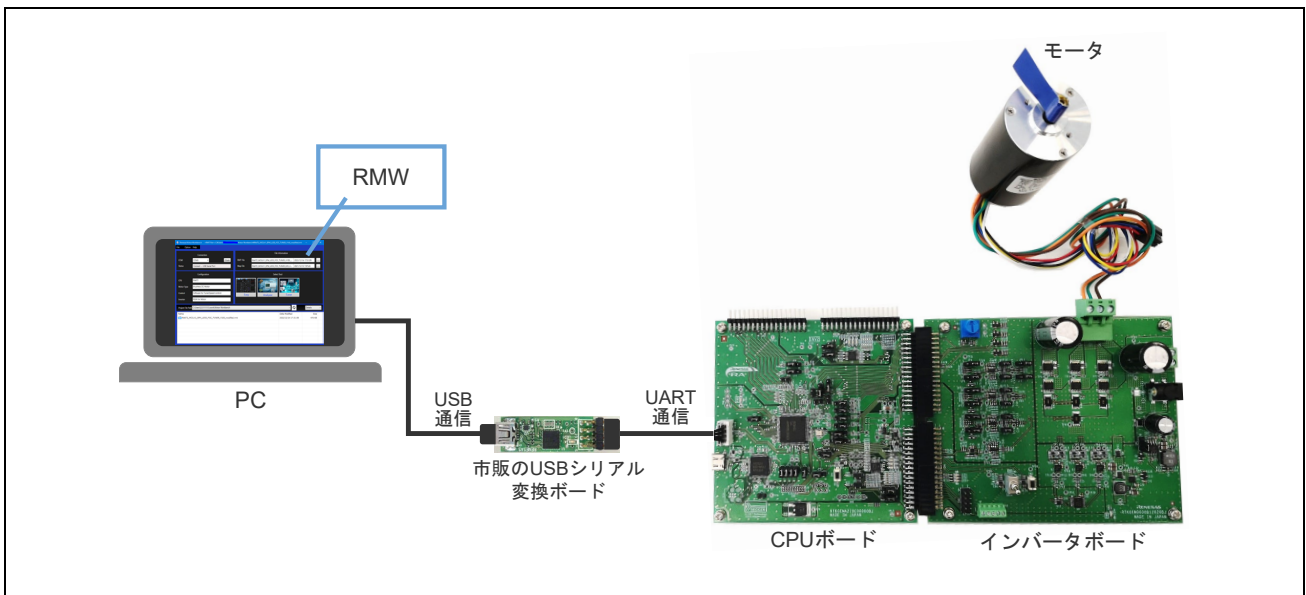


図 15.13 RMW 使用時の接続

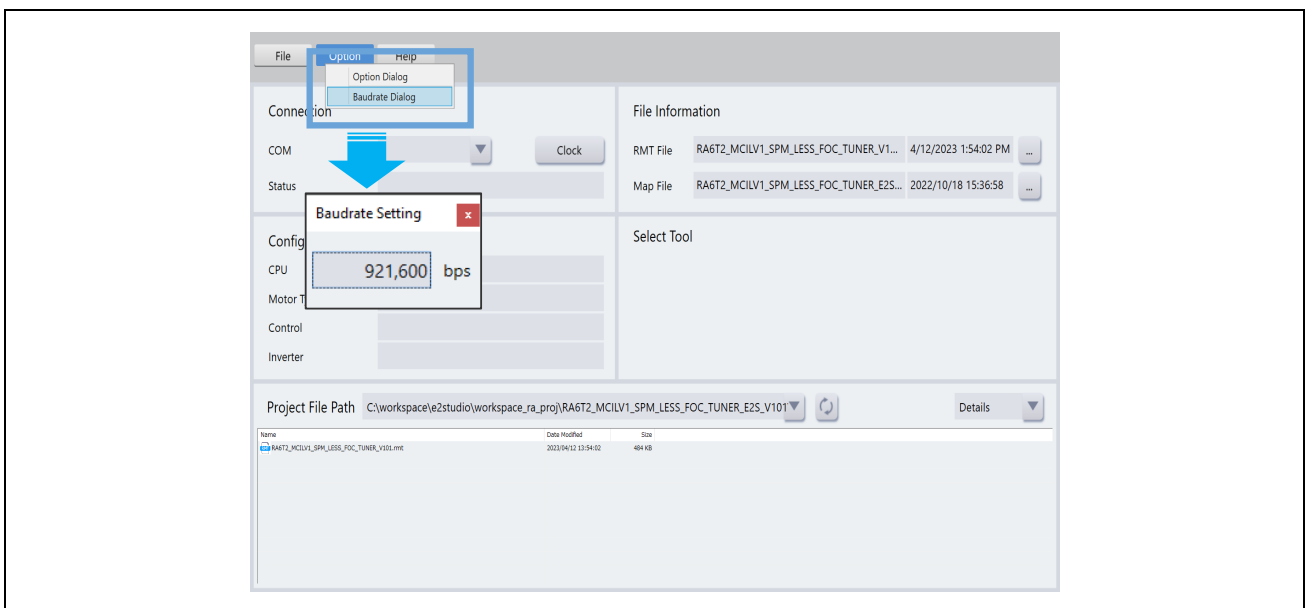


図 15.14 RMW の通信速度の設定

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/>

お問い合わせ先

<https://www.renesas.com/contact>

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。

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	—	<ul style="list-style-type: none"> • 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	—	<ul style="list-style-type: none"> • 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	—	<ul style="list-style-type: none"> • RA8T1 を追加 • 「13. Servo」の内容を改訂 • 「14.6 Tuner Library」を追加

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 3.1.2
ユーザーズマニュアル

発行年月日 2024 年 1 月 30 日 Rev.4.02

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 3.1.2