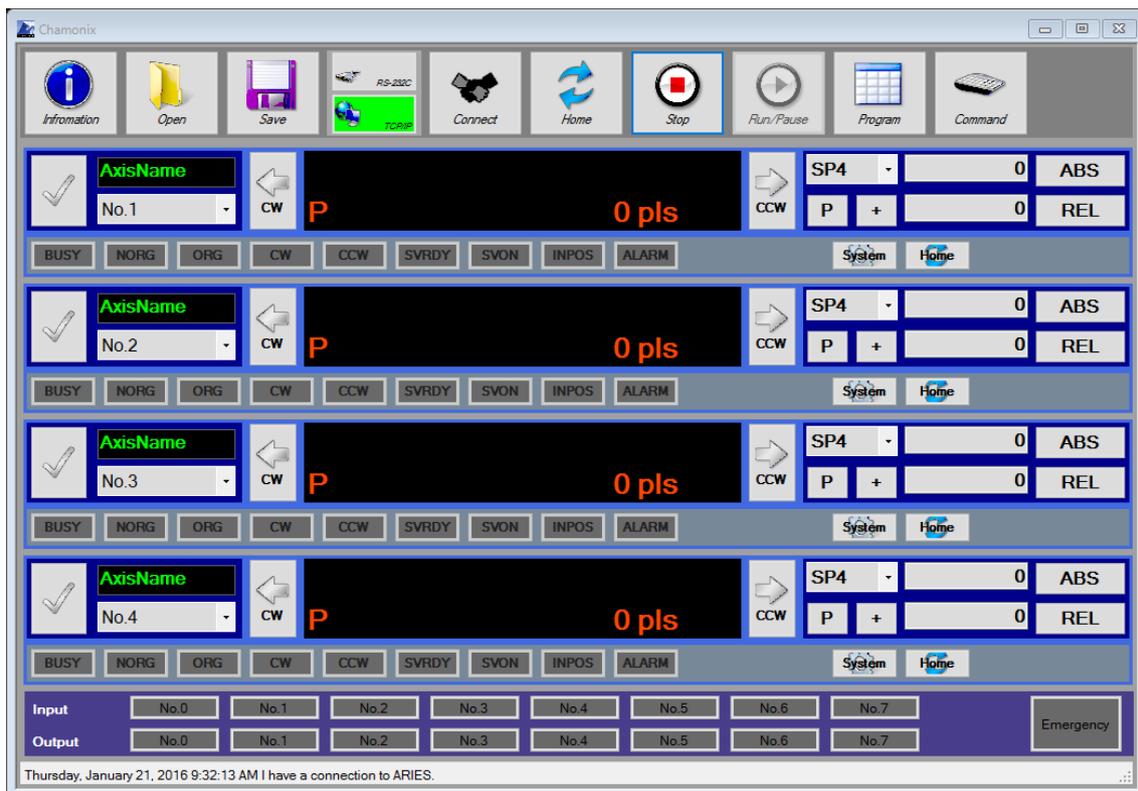




Chamonix

ユーザーズガイド

Rev. 1.2.3



神津精機株式会社

目次

目次	2
1. はじめに	6
1.1. Chamonix について	6
1.2. このユーザーズガイドについて.....	7
1.3. ご使用上の注意	8
1.4. 使用許諾契約の内容	9
1.5. 対応 OS と動作要件.....	10
1.6. 言語.....	11
2. インストール	12
2.1. インストールファイル	12
2.2. 「Setup」の起動	13
2.3. 旧バージョンの削除.....	14
2.4. 使用許諾契約の確認と同意	15
2.5. ユーザー情報の入力	16
2.6. インストールの実行	17
2.7. インストールの完了	18
2.8. インストールの確認	19
3. アンインストール	20
3.1. コントロールパネル	20
3.2. プログラムのアンインストール.....	21
3.3. 確認のダイアログ.....	22
4. コントローラとの配線	23
4.1. RS-232C	23
4.2. TCP/IP	24
4.3. CRUX との接続.....	26
4.3.1. CRUX の通信設定.....	26
CRUX の通信設定用ロータリースイッチの設定.....	26
CRUX の通信設定用ロータリースイッチの位置.....	27
4.3.2. CRUX の結線方法.....	28

4.4.	ARIES との接続	29
4.4.1.	ARIES の通信設定	29
	ARIES の通信設定用ロータリースイッチの位置	29
	ARIES の通信設定用ロータリースイッチの設定	30
4.4.2.	ARIES の結線方法	31
4.5.	SC-210/410 の場合	32
4.5.1.	SC-210/410 の通信設定	32
	SC-410 の通信設定用ロータリースイッチの位置	32
	SC-210/410 の通信設定用ロータリースイッチの設定	33
5.	起動してみましょう！	34
5.1.	デスクトップのアイコンから起動する	34
5.2.	保存したファイルから起動する	35
5.3.	起動画面	36
5.4.	はじめての起動	37
6.	コントローラに接続しましょう	38
6.1.	RS-232C でコントローラと接続する	38
6.1.1.	RS-232C を選択する	38
6.1.2.	COM ポートの選択	39
6.1.3.	COM ポートの番号の確認方法	40
6.2.	TCP/IP による接続	41
6.2.1.	TCP/IP を選択する (ARIES のみ)	41
6.2.2.	Chamonix の接続先を指定しましょう	42
6.2.3.	ARIES の IP アドレスの変更	43
6.3.	コントローラへの接続	44
7.	バージョン情報を確認しましょう	46
7.1.	このアプリのバージョン情報を確認する	46
8.	原点復帰をおこないましょう	48
8.1.	原点復帰とは	48
8.2.	原点復帰方式の変更	49
8.3.	全軸原点復帰	50
8.4.	原点復帰中のステータス	51
8.5.	単軸の原点復帰	52

9.	ステージを動かしてみよう	53
9.1.	制御する軸の選択	53
9.2.	軸の名前を設定する	54
9.3.	現在位置表示の切り替え	55
9.4.	速度テーブルの選択	56
9.5.	軸の分解能を設定する	57
9.6.	マニュアルでの移動	59
10.	位置決め	60
10.1.	絶対位置管理による位置決めの設定	60
10.2.	絶対位置管理による位置決めの実行	61
10.3.	相対位置管理による位置決めの設定	62
10.4.	相対位置管理による位置決めの実行	63
10.5.	多軸駆動を実行	64
11.	非常停止を受信!	66
11.1.	非常停止信号を受信 (ARIES)	66
11.2.	非常停止からの復帰 (ARIES)	67
11.3.	非常停止からの復帰 (CRUX)	68
12.	トリガー出力	69
12.1.	ワンショットトリガー信号の出力 (ARIES)	69
13.	システムパラメータを変更する	70
13.1.	パラメータ設定を起動する	70
13.2.	パラメータを設定する	72
13.3.	システムパラメータのファイルへの保存	74
13.4.	パラメータの印刷	76
13.5.	システムパラメータのプロパティ	78
13.6.	速度テーブルの変更	80
14.	プログラマブルグリッドを利用する	81
14.1.	プログラマブルグリッドの起動	81
14.2.	プログラマブルグリッドの基本機能	82
14.3.	プログラムを実行	83
14.4.	プログラムの停止	84
14.5.	プログラムの一時停止	85
14.6.	プログラムデータの入出力	86

14.7.	プログラマブルグリッドのコマンド	87
14.7.1.	ORG: 原点復帰	87
14.7.2.	ABS: 絶対位置管理による位置決め	88
14.7.3.	REL: 相対位置管理による位置決め	89
14.7.4.	WOT: 汎用出力の制御	90
14.7.5.	FOR: 繰り返しループ	91
14.7.6.	IF: 条件付きの実行	92
14.7.7.	GOTO: 行のジャンプ	93
14.7.8.	SBY: スタンバイ	94
14.7.9.	TFR: ワンショットトリガー	95
14.7.10.	TRG: 駆動に同期したトリガー出力	96
14.7.11.	プログラマブルグリッドのコマンド一覧	97
15.	直接リモートコマンドを入力する	98
15.1.	コマンドコントロールの起動	98
15.2.	コマンドコントロールからコマンドを送信	99
16.	APIを利用して外部から制御しましょう	101
16.1.	WindowsAPIによる外部制御	101
16.2.	外部のアプリケーションからの制御	102
16.3.	外部のアプリからAPSコマンドを送る	103
16.4.	外部のアプリからRPSコマンドを送る	105
16.5.	外部のアプリからSTPコマンドを送る	107
16.6.	外部のアプリからORGコマンドを送る	109
16.7.	外部のアプリからプログラマブルグリッドのスタンバイを解除する	111
16.8.	外部制御のサンプルプログラム	113
17.	Chamonixを終了する	114
17.1.	Chamonixを終了する	114
18.	その他	115
18.1.	通信ログの保存場所	115
18.2.	通信ログを確認	116
	改訂履歴	117

1. はじめに

1.1. Chamonix について

Chamonix (シャモニー) は神津精機製モーターコントローラを制御するアプリケーションです。神津精機製のモーターコントローラ、KOSMOS シリーズの ARIES 及び LYNX と旧製品 SC シリーズ¹の SC-20、SC-21、SC-200、SC-400、SC-210、SC-410 を制御できます。

Chamonix は 4 軸まで同時に制御できる GUI と簡易的なマクロのようなプログラミンググリッドを備えています。また WindowsAPI を通して外部のアプリケーションと連携をとることも可能です。

コントローラとの通信には、RS-232C のシリアル通信を利用しています。

**このアプリケーション予告なく変更される場合があります。
ご利用にあたっては使用許諾条件をよくお読みください。**

¹ SC シリーズをご利用の場合、ファームウェアのバージョンアップが必要な場合があります。

1.2. このユーザーズガイドについて

このガイドはモーターコントローラ制御アプリケーション「Chamonix（シャモニー）」を説明したものです。

「Chamonix」の性能ならびに機能を十分にご利用いただくために、よくお読みになり、十分理解した上でご使用ください。

また、いつでもご利用いただけますよう、は大切に保管してください。

本書は最終的にご使用いただく方のお手元に届くようご配慮ください

「Chamonix」は日本語、英語に対応しておりますが、このガイドの GUI は日本語で表示します。

1.3. ご使用上の注意



始業時または操作時には、このアプリケーションが正常に動作していることを確認してください。



このアプリケーションが万一故障した場合の各種損害を防止するために十分な安全対策を施してご使用ください。



取扱説明書や別途取り交わした仕様書などに記載された以外の不適当な条件下でのご使用に起因する場合、機能及び性能の保証はできかねますのでご注意ください。



本製品を他の機器と組み合わせて使用する場合、使用条件や環境などにより、機能および性能が満足できない場合がありますので十分検討の上ご使用ください。



人体の保護を目的とした使用はしないでください。



他のアプリケーションと併用しないでください。機能および性能が満足できない場合があります。

1.4. 使用許諾契約の内容

このアプリケーションをご使用になる前に、以下の契約条項を十分にお読みください。

対象

- この使用許諾契約の対象は、パッケージとして提供される制御アプリケーション「Chamonix」とそれに付属する各種のライブラリ、およびそれらの使用方法等を記入したマニュアル等の付属するすべてのドキュメントとします。以下ではChamonixと付属するライブラリおよび付属するすべてのドキュメントを総称して「このアプリケーション」と記すものとします。ドキュメントには紙、電子ファイル等のすべてのメディアが含まれます。

使用权

- 神津精機株式会社はお客様がこの契約条項のすべてに同意した場合に、ユーザーに対してこのアプリケーションの非独占的使用権のみを許諾します。この非独占的使用権では、神津精機株式会社から使用許諾を受けたユーザーは1台のコンピュータにこのアプリケーションをインストールし使用することができます。ユーザーは非独占的使用権以外の一切の権利を有しないものとします。この使用権には無償のユーザーサポートは含まれていないものとします。

著作権

- このアプリケーションと、このアプリケーションとライブラリに付属するすべてのドキュメントに関する著作権は、特に断りのない限り神津精機株式会社あるいは該当部分の著作権を保持する個人または法人に帰属します。これらは著作権に関する法律および国際条約によって保護されています。

禁止条項

- ユーザーは、このアプリケーションとライブラリあるいは付属するすべてのドキュメントを複製し、第三者への転売・配布をすることはできません。ユーザー自身のためにバックアップを保有する目的以外で、このアプリケーションとライブラリを複製することはできません。ユーザーがこのアプリケーションを改造、リバースエンジニアリング、逆アセンブルあるいは逆コンパイル等を行うことを固くお断りいたします。

免責

- 神津精機株式会社は、このアプリケーションを運用した結果により生じた、ユーザーもしくは第三者の直接的、間接的あるいは波及効果によるあらゆる損害に対して、いかなる責任も負わないものとします。また、ユーザーもしくは第三者のデータ、プログラムその他の無体財産に関する損害、使用利益および得べかり利益の喪失等に対しても神津精機株式会社は一切の責任を負わないものとします。

その他

- 神津精機株式会社は、いつでもこのアプリケーションを更新できるものとします。更新版の提供条件は更新時に神津精機が定めるものとします。更新された場合、神津精機は更新前のこのアプリケーションとの互換性を更新版で提供する義務を負わないものとします。神津精機株式会社は、いつでもこの使用許諾条件を変更できるものとし、変更がなされた場合には、従前の使用許諾条件は破棄され、最新の使用許諾条件が適用されるものとします。

契約の解除

- お客様が本許諾事項に違反したとき、あるいはこの使用許諾条項に同意されない場合は、直ちに契約を解除し、ソフトウェア及び複製物を神津精機株式会社に返却もしくは廃棄していただくものとします。

1.5. 対応 OS と動作要件

対応OS

- ChamonixはWindows7SP1^②あるいはWindows10^③を正式にサポートしています。
- x86とx64のいずれにも対応しています。X86の場合にはProgram Filesに、x64の場合はProgram Files(x86)にインストールされます。

前提条件

- Chamonixは、Microsoft .NET Frameworkを使用して構築されています。インストールにはMicrosoft .NET Framework Full 4.7.2以上が必要です。
- 起動時に600MB以上のメモリの空き領域が必要です。
- RS-232CあるいはTCP/IP等のコントローラと通信するインターフェースが必要です。

2 3 4

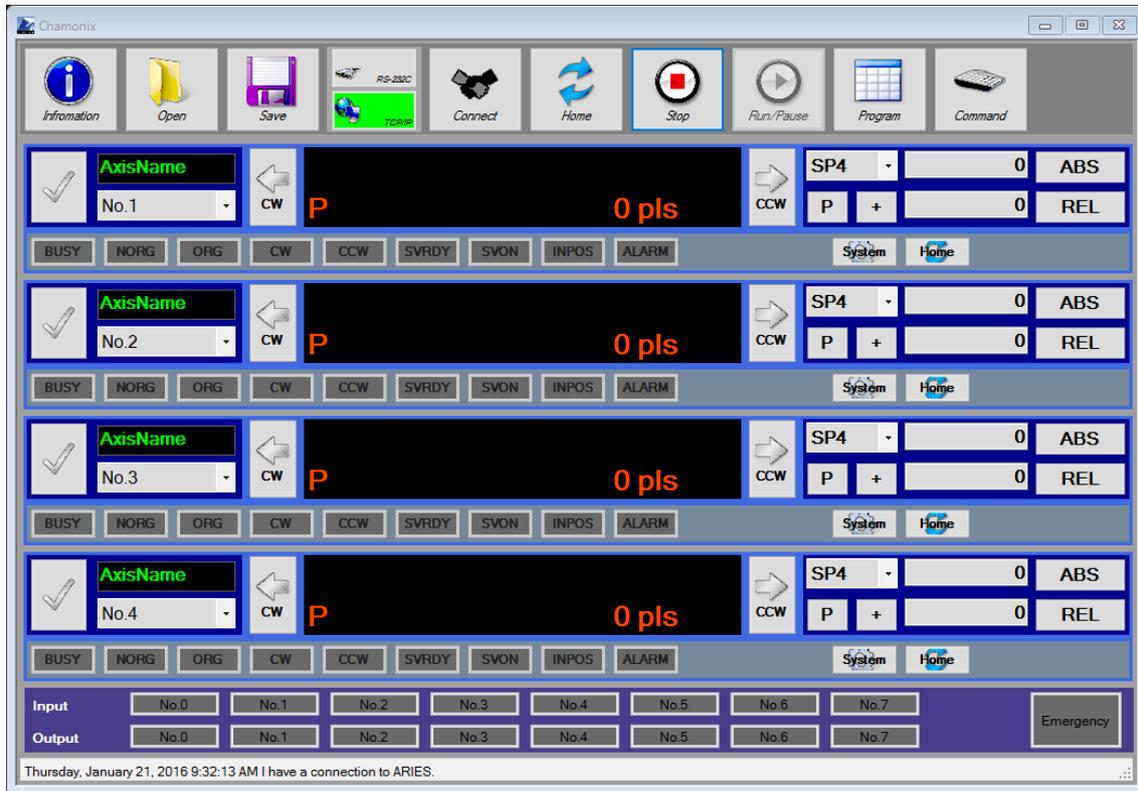
² Chamonix Ver.1.2.2.13 以上の場合です。

³ .NET Framework、Windows^④は米国 Microsoft Corporation の米国、およびその他の国における商標または登録商標です。

⁴ WindowsXP^④はサポート対象外です。

1.6. 言語

「Chamonix」は日本語と英語に対応しています。言語は Windows の入力の言語方式に従います。



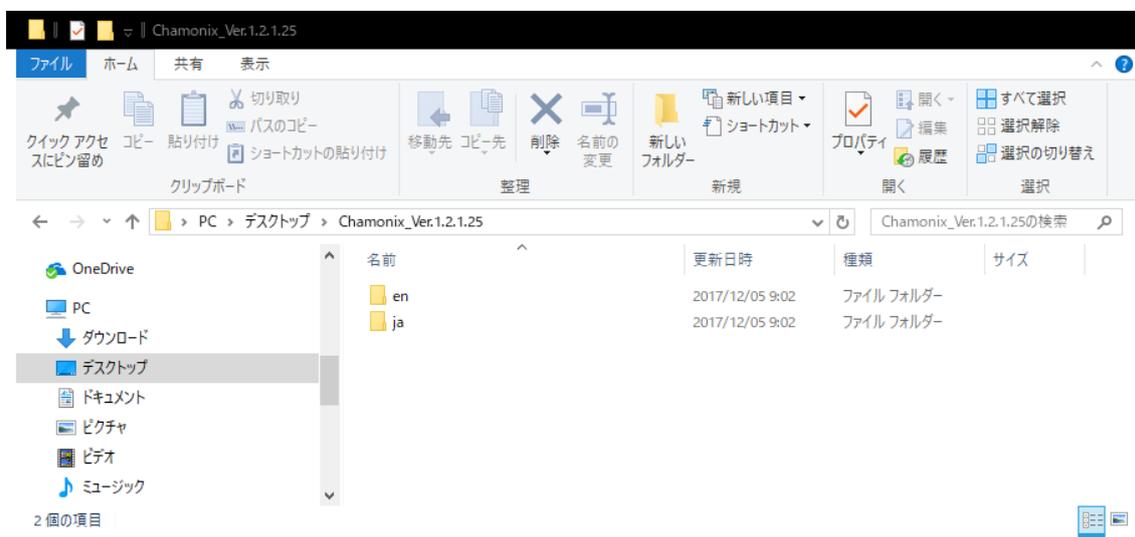
2. インストール

2.1. インストールファイル

インストールをするためには管理者権限が必要です。インストール DISK フォルダ以下のようなファイルで構成されています。インストールを実行するには「Setup」をクリックして実行してください。

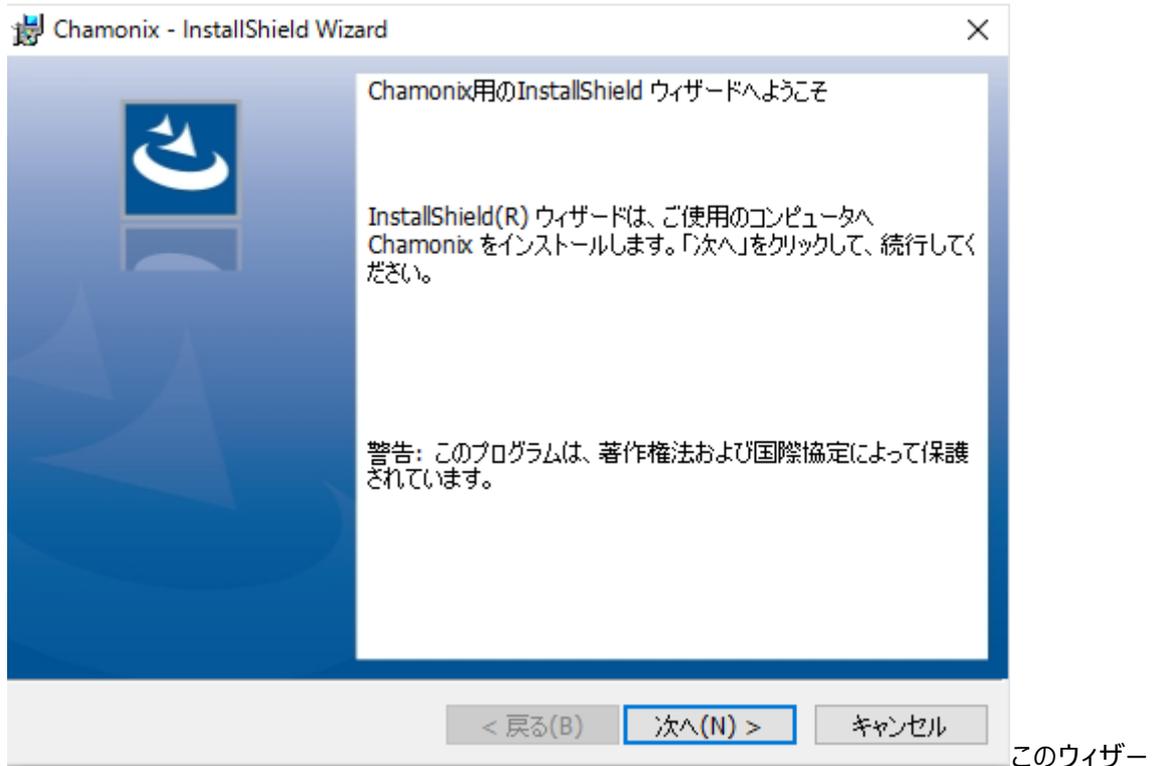
弊社 Web サイトからダウンロードいただいた場合、Zip ファイルで圧縮されています。解凍すると下記のようなフォルダが作成されます。

このフォルダのファイルは移動や変更をしないでください。



2.2. 「Setup」の起動

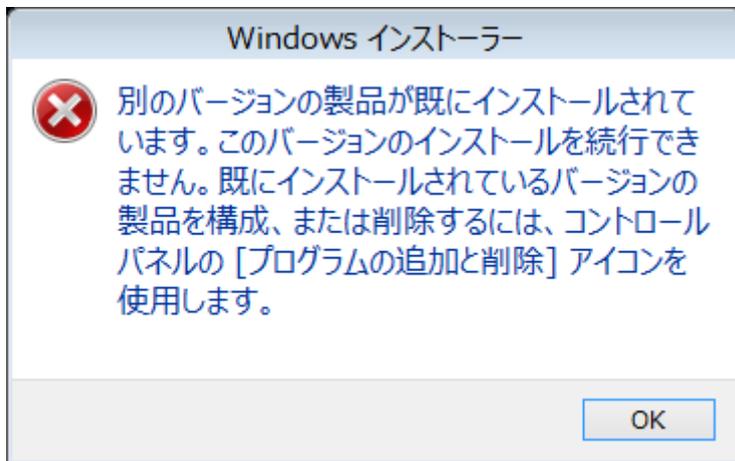
Setup.exe を起動すると Chamonix 用の InstallShield®⁵ウィザードが開きます。



ドを利用してインストールを続けるのなら「次へ」を押してください。「キャンセル」を押すとインストールを終了します。つぎに使用許諾契約の確認と同意の画面が開きます。

⁵ InstallShield は Flexera Software, Inc. および/ または InstallShield Co. Inc. の米国および/ または他の国における登録商標または商標です

2.3. 旧バージョンの削除



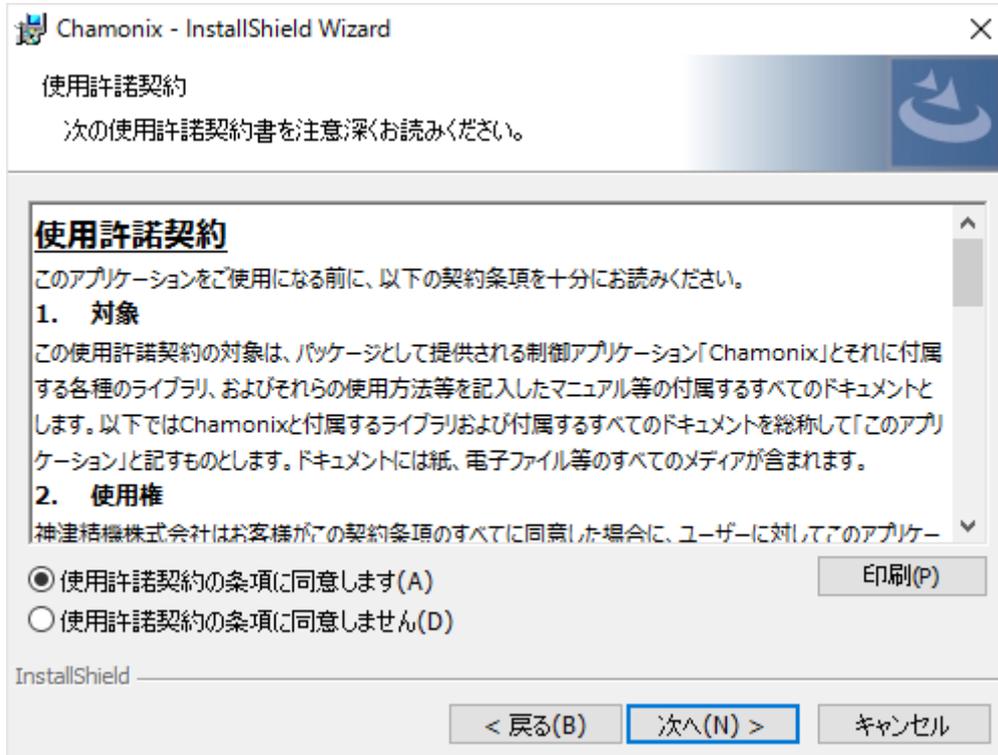
Chamonix あるいは Chamonix を構成するダイナミックリンクライブラリの一部が既にインストールされている場合図のようなメッセージ⁶が表示される場合があります。

この場合には、既にインストールされている Chamonix をコントロールパネルの [プログラムの追加と削除] アイコンを使用して、削除する必要がります。

削除の方法はアンインストールで説明します。

⁶ Microsoft Windows 8.1 の画面を示します。ご使用の OS によっては異なる場合があります。

2.4. 使用許諾契約の確認と同意



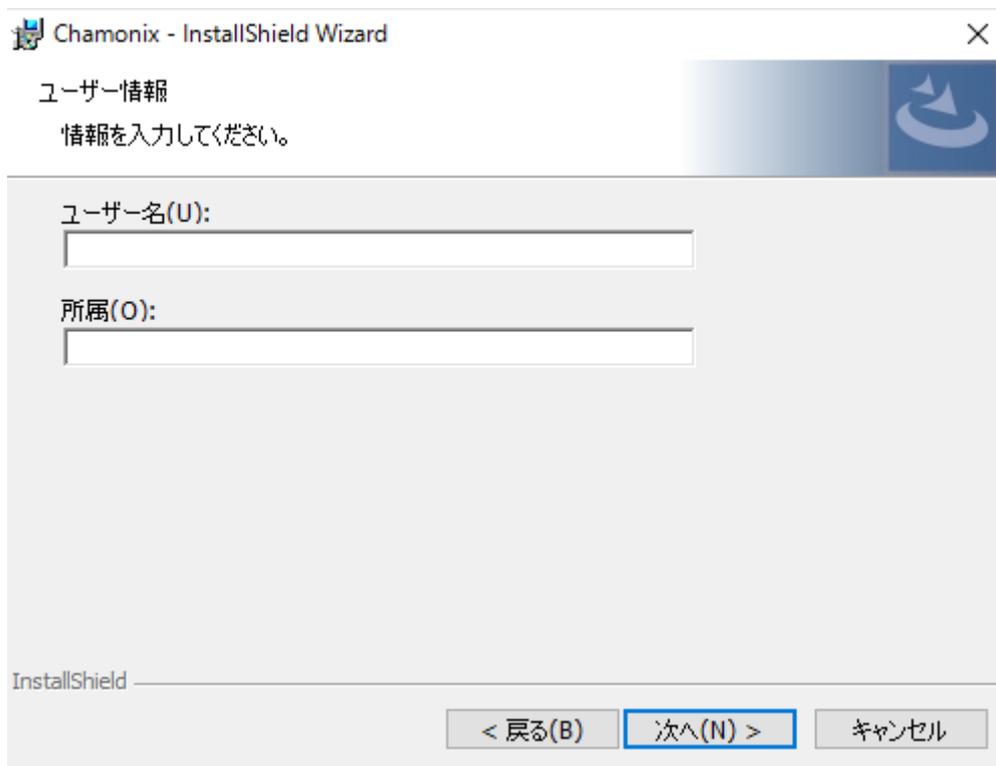
この使用許諾契約の内容に同意していただける場合は「使用許諾契約の条項に同意します」を選択してください。

同意していただけない場合は「使用許諾契約の条項に同意しません」を選択してください。この場合には、「次へ」は有効になりません。

「印刷」を押すと、「使用許諾の契約について」を印刷します。

[使用許諾契約の条項](#)に同意していただき、「次へ」を押すと「ユーザー情報の入力」へ移ります。

2.5. ユーザー情報の入力



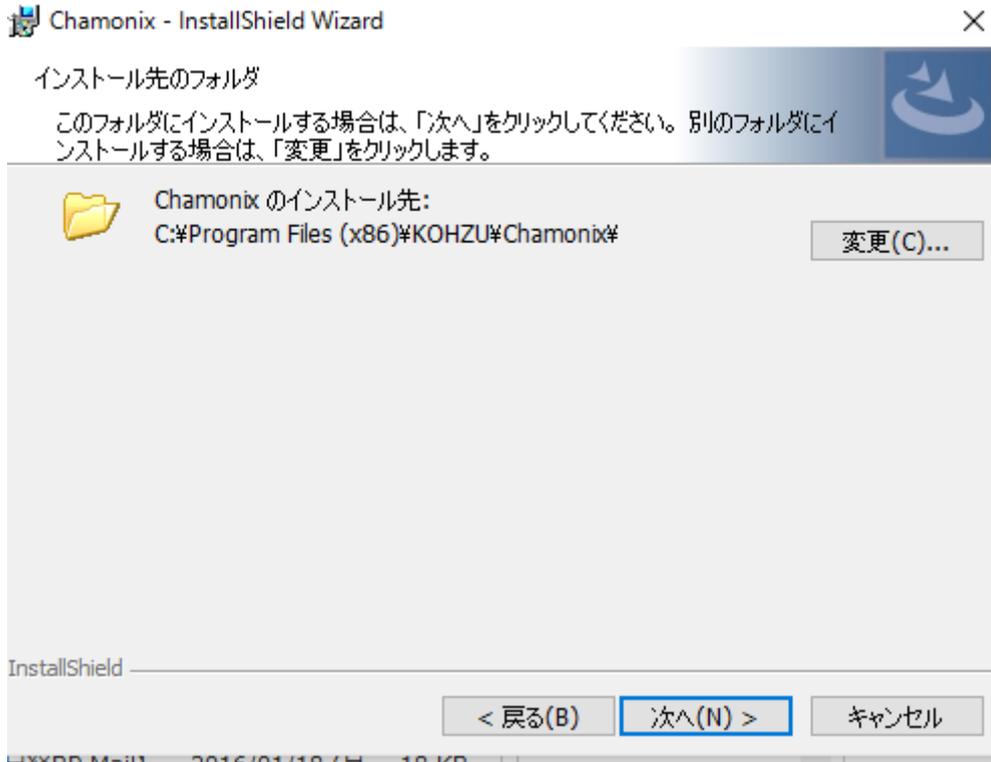
ユーザー名と所属を入力してください。

ユーザー情報の入力は任意です。

入力が終了したら「次へ」をクリックして下さい。もう一度使用許諾契約の内容を確認するためには「戻る」をクリックして下さい。

「キャンセル」をクリックするとインストールを中止します。

2.6. インストールの実行



インストールを実行するには「インストール」を押してください。

インストールを実行するためには、管理者権限が必要です。

インストールを実行中に「ユーザーアカウントの制御」ダイアログが開き、インストールを実行するかどうかたずねられる場合があります。この場合には、Windows インストーラの当社のコードサイン証明書をご確認いただき、「はい」を押してください。「いいえ」を押すとインストールは行われません。

インストール先フォルダは変更できません。

ユーザー情報の入力へ戻るには、「戻る」を押してください。

「キャンセル」を押すとインストールウィザードを終了します。

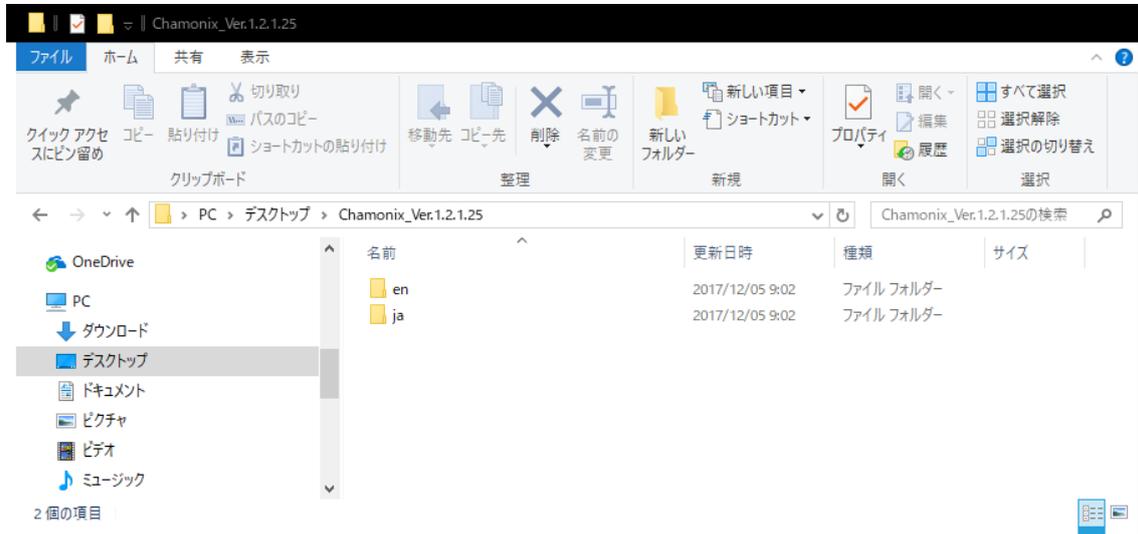
2.7. インストールの完了



前の画面で「インストール」を押すと、Chamonix のインストールが実行されます。上手の画面が表示をされたら正常にインストールされています。

「完了」をクリックするとウィザードは終了します。

2.8. インストールの確認



インストールが終了すると図のように C:\Program Files(x86)\KOHZU\Chamonix フォルダに上手の 6 ファイルが生成されます⁷。

⁷ Microsoft Windows10 の画面を示します。ご使用の OS によっては異なる場合があります。

3. アンインストール

3.1. コントロールパネル



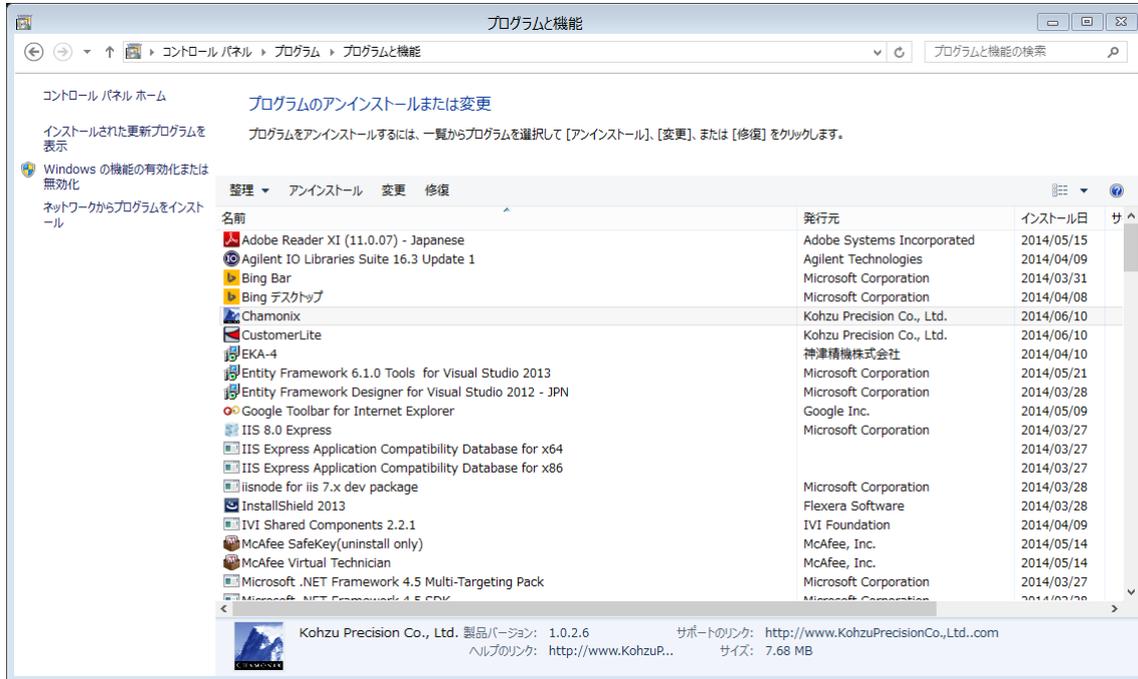
「コントロールパネル」⁸の「プログラム」を開きます。つぎに「プログラムの機能」をクリックします。

あるいは「コントロールパネル」から直接、「プログラムのアンインストール」をクリックします。

次ページのような「プログラムのアンインストールまたは変更」が開きます。

⁸ 右図は Microsoft Windows10 のコントロールパネルの例を示します。

3.2. プログラムのアンインストール



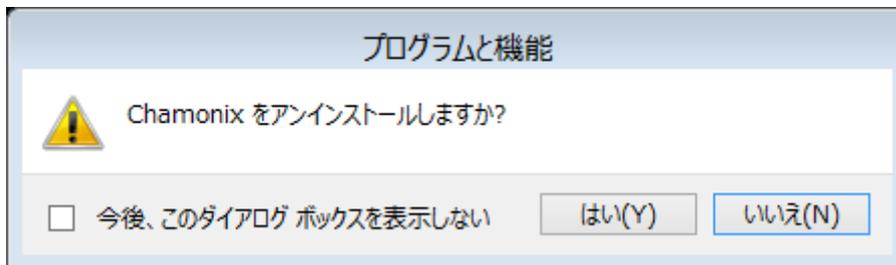
表示されたプログラムの一覧から Chamonix を選択します。

Chamonix をアンインストールする場合には、「アンインストール」をクリックします。

⇒確認のためのメッセージダイアログが表示されます。

Chamonix を修復する場合には、「修復」をクリックします。

3.3. 確認のダイアログ



メッセージダイアログの「はい」を押すとアンインストールされます。

アンインストールではインストールした Chamonix.exe、StageControl.dll、SerialControl.dll、SystemParam.dll や Chamonix.chm は消去されますが、インストール後にアプリケーションによって生成された SerialContl.log や config.xml は消去されません。

これらのファイルを消去するには、エクスプローラなどから直接ファイルを消去する必要があります。

該当するファイルは「C:¥Program Files(x86)⁹¥Common Files¥KOHZU」内に格納されています。

⁹ 32ビット版のOSの場合はC:¥Program Files ¥Common Files¥KOHZU¥Chamonixです。

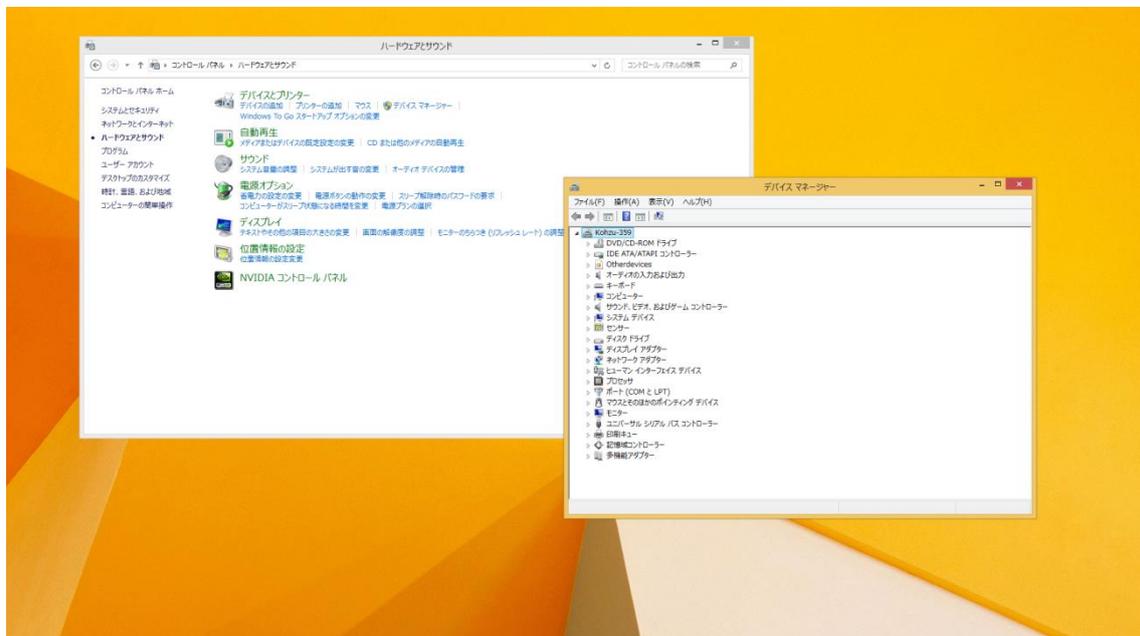
4. コントローラとの配線

4.1. RS-232C

ARIES、CRUX、旧 SC シリーズのコントローラは RS-232C によって接続できます。CRUX とは USB (仮想 COM ポート)¹⁰によっても接続できます。

あらかじめ Controller が接続されているポートをデバイスマネージャーで確認してください。

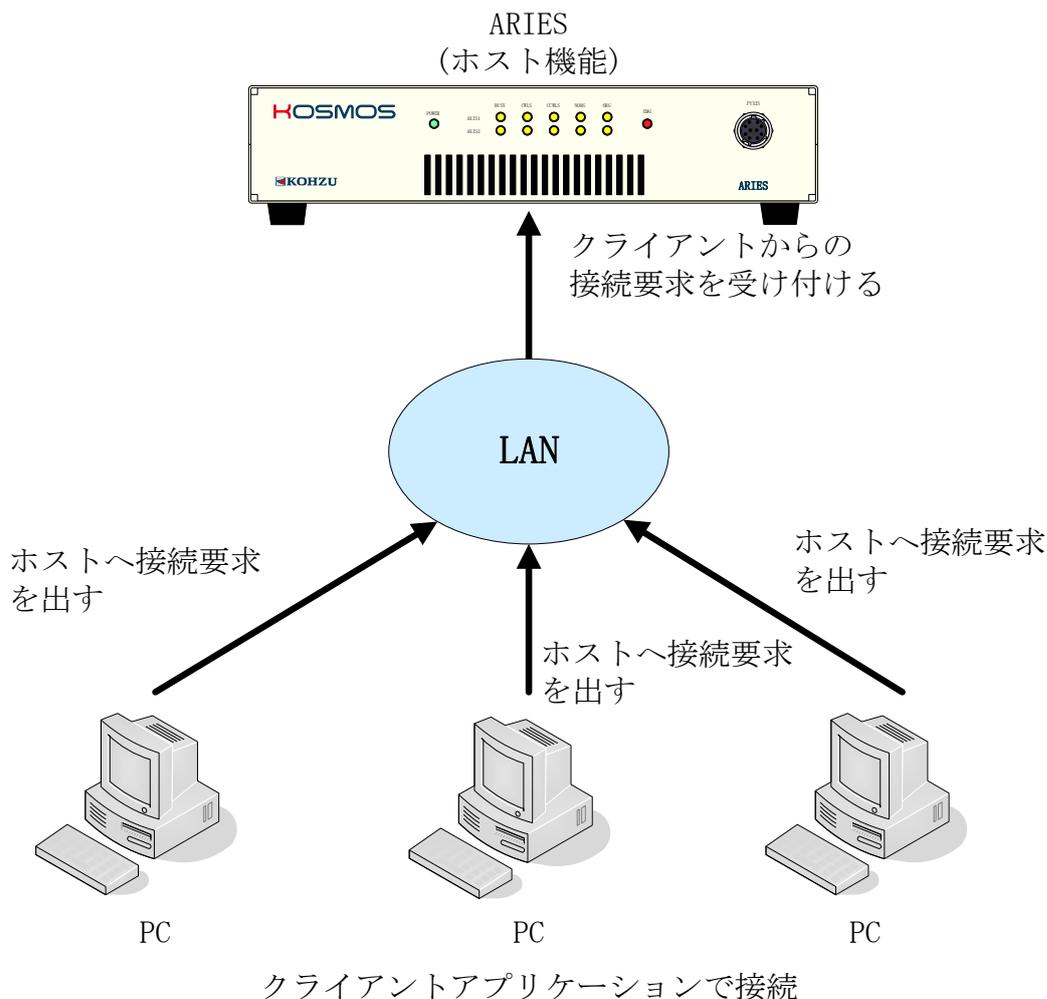
「コントロールパネル→ハードウェアとサウンド→デバイスマネージャー」で確認できます。デバイスマネージャーの起動には管理者権限が必要です。



¹⁰ USB 仮想 COM ポートは CRUX のみで使用できます。

4.2. TCP/IP

ARIESに対してはTCP/IPクライアントとして接続することが可能です。TCP/IPクライアントとしてARIESに接続するためには、あらかじめRS-232C通信などによってARIESのTP/IPアドレス、ポート番号等¹¹を設定する必要があります。この接続のイメージを示します。



¹¹ RS-232C通信で接続し、TCP/IPに「必要な各パラメータを設定していただく方法と、P2Pで次ページに示した設定値を用いて接続をして設定していただく方法があります。

出荷時にARIESに設定されている値は以下の通りです。詳しくは「ARIES/LYNXユーザーズマニュアル」をご覧ください。

機能	デフォルト設定	対応コマンド	
		書換	読出
ホスト/クライアント/Telnet設定	ホスト (設定値0)	WHC	RHC
ARIESのIPアドレス設定	192.168.1.120	WIP	RIP
サブネットマスク設定	255.0.0.0	WSN	RSN
ポート番号設定 ※	12321	WPT	RPT
クライアント制限設定	制限なし (クライアントNo.1の 設定値777.0.0.0)	WCL	RCL
ホストPCのIPアドレスの指定	192.168.1.102	PIP	RPI
デフォルトゲートウェイ設定	設定なし (設定値999.0.0.0)	WGW	RGW
パスワード設定 (書換のみ)	KOSMOS	WPS	-
アプリケーション接続数設定	32	WAP	RAP

4.3. CRUX との接続

4.3.1. CRUX の通信設定

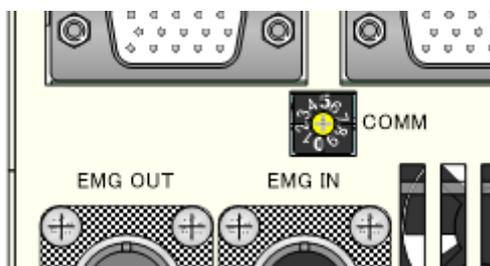
CRUX はリアパネルにあるロータリースイッチ（COMM）にて通信の条件を設定・変更することができます。初期設定は4（通常コマンド USB モード）です。

詳しくは **CRUX/CRUX-A ユーザーズマニュアル**をご覧ください。

※RS-232C 通信において速度(baud)以外の設定は以下のように設定してください。

パリティ	NON
語長	8bit
ストップビット	1

CRUX の通信設定用ロータリースイッチの設定



CRUXの通信設定用ロータリースイッチの位置

通信モード	通信設定		
	RS-232C速度	USB	
0	38400	*	通常 コマ ンド
1	57600	*	
2	19200	*	
3	9600	*	
4	*	USB	
5	38400	*	簡 単 コ マ ン ド
6	57600	*	
7	19200	*	
8	9600	*	
9	*	USB	

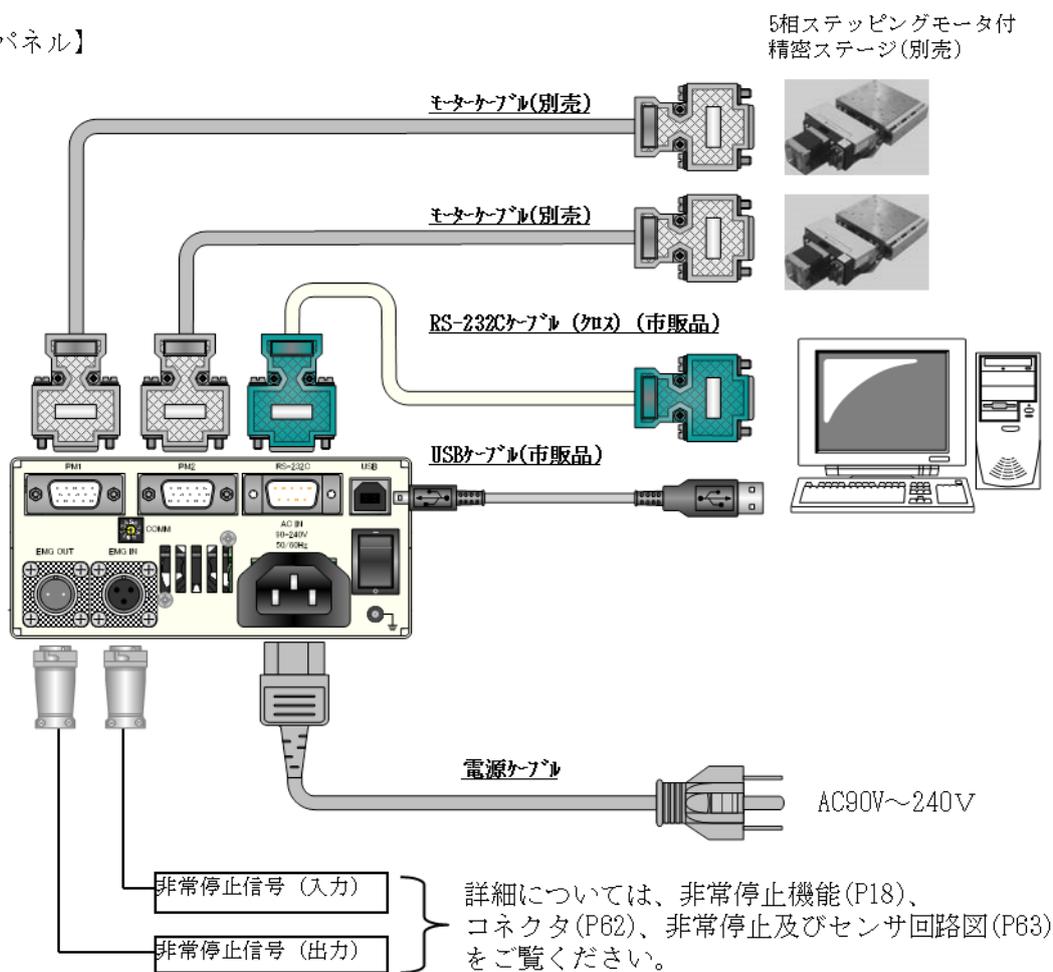
4.3.2. CRUX の結線方法



結線の抜き差しを行う際には、必ず本体の電源を切った状態で行ってください。

CRUX と、外部機器との接続/結線を説明します。

【リアパネル】



詳しくは CRUX/CRUX-A ユーザーズマニュアルをご覧ください。

4.4. ARIES との接続

4.4.1. ARIES の通信設定



通信設定を行う際には、必ず本体の電源を切った状態で行ってください。

ARIES はリアパネルにあるロータリースイッチ（COMM）にて通信の条件を設定・変更を行うことができます。初期設定はモード 4(RS-232C 115200baud)です。

詳しくは **ARIES/LYNX ユーザーズマニュアル**をご覧ください。

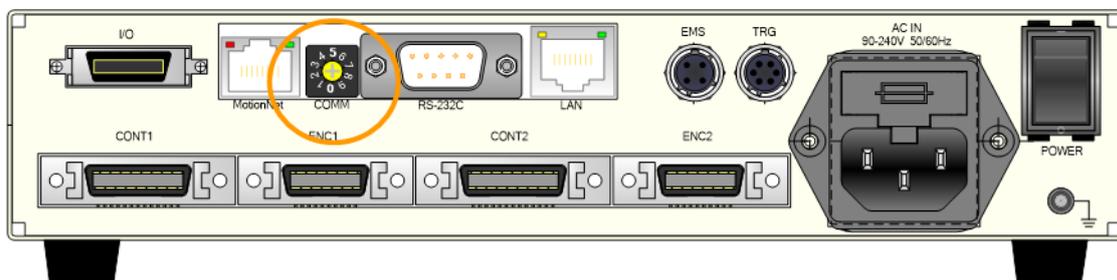
※RS-232C 通信において速度(baud)以外の設定は以下のように設定してください。

パリティ	NON
語長	8bit
ストップビット	1

ARIES の通信設定用ロータリースイッチの位置

通信設定用ロータリースイッチは以下の位置にあります。

【リアパネル】



ARIES の通信設定用ロータリースイッチの設定

設定は下表の通りです。(モード6~9 使用不可)

通信モード	通信設定	
	RS-232C速度 (baud)	LAN
0	9600	*
1	19200	*
2	38400	*
3	57600	*
4	115200	*
5	*	LAN
6	*	*
7	*	*
8	*	*
9	*	*

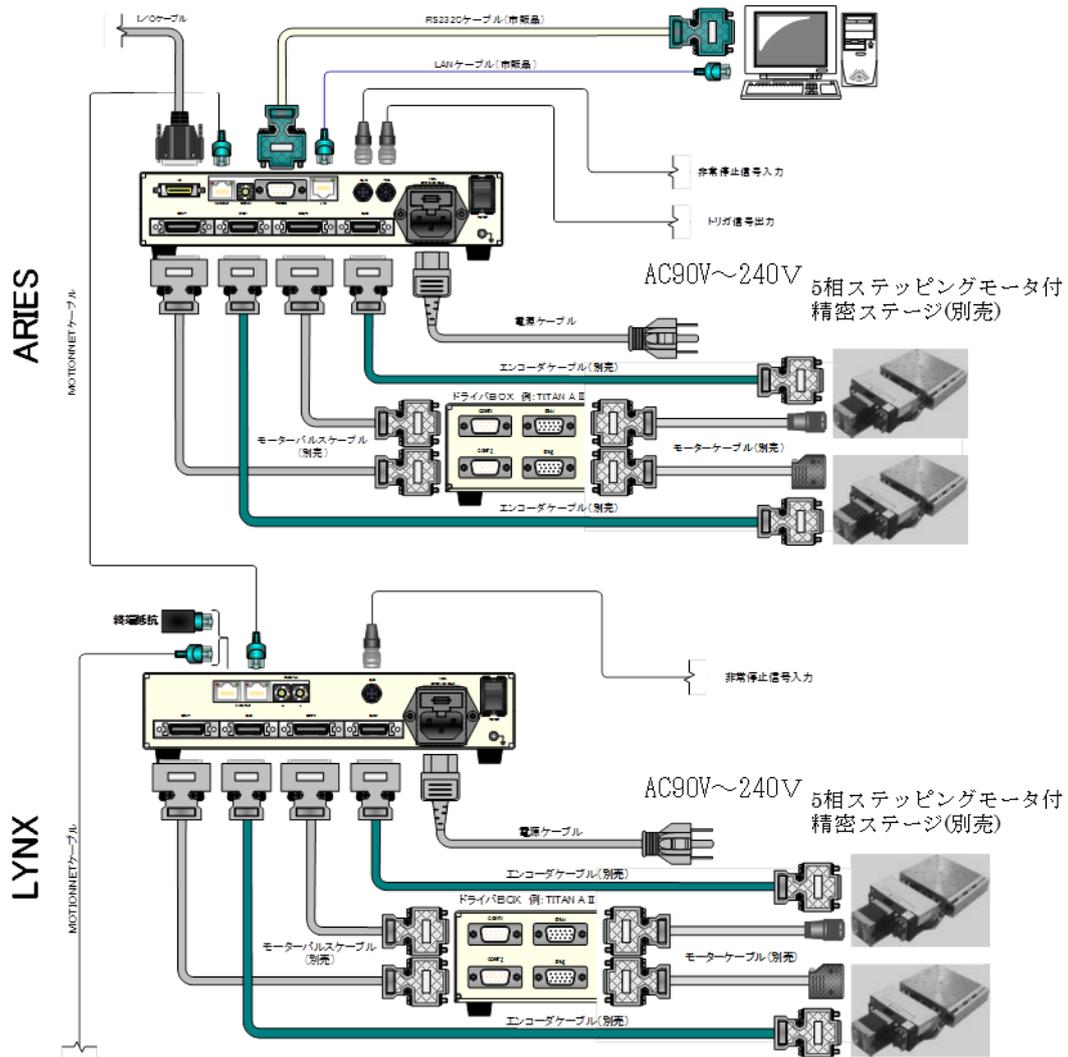
4.4.2. ARIES の結線方法



結線の抜き差しを行う際には、必ず本体の電源を切った状態で行ってください。

ARIES と、外部機器との接続/結線を説明します。

【リアパネル】



※Motionnet®ケーブル接続間にハブは使用しないでください。

詳しくは ARIES/LYNX ユーザーズマニュアルをご覧ください。

4.5. SC-210/410 の場合

4.5.1. SC-210/410 の通信設定



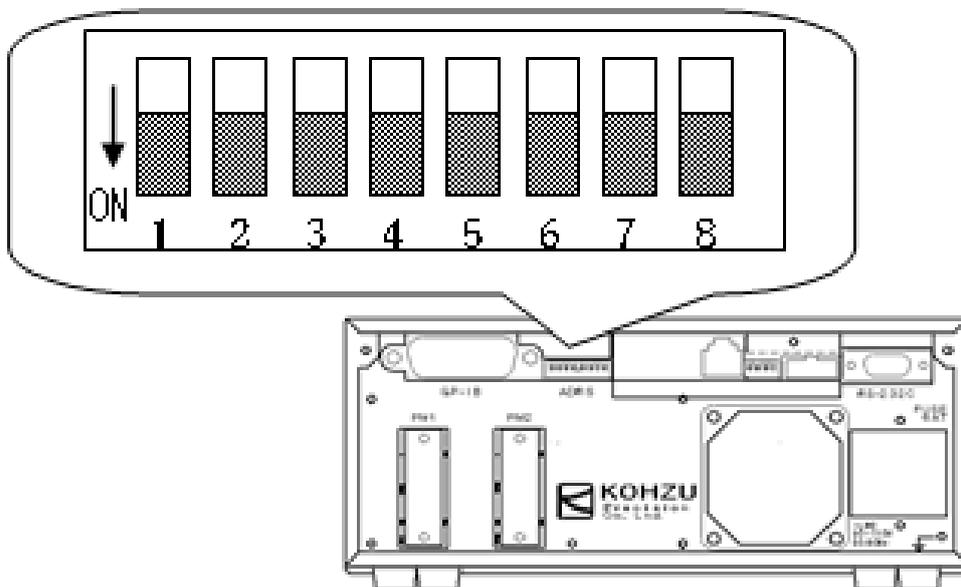
通信設定を行う際には、必ず本体の電源を切った状態で行ってください。

SC-210/410 はリアパネルにあるディップスイッチにて通信の条件を設定・変更を行うことができます。初期設定は RS-232C 38400baud、NON パリティ、語長は 8bit、ストップビットは 1 です。

詳しくは **SC-210/410 取扱説明書** をご覧ください。

SC-410 の通信設定用ロータリースイッチの位置

通信設定用ロータリースイッチは以下の位置にあります。



SC-020、SC-021、SC-200、SC-210、SC-400 については、それぞれの取扱説明書をご覧ください。

SC-210/410 の通信設定用ロータリースイッチの設定

設定は下表の通りです。

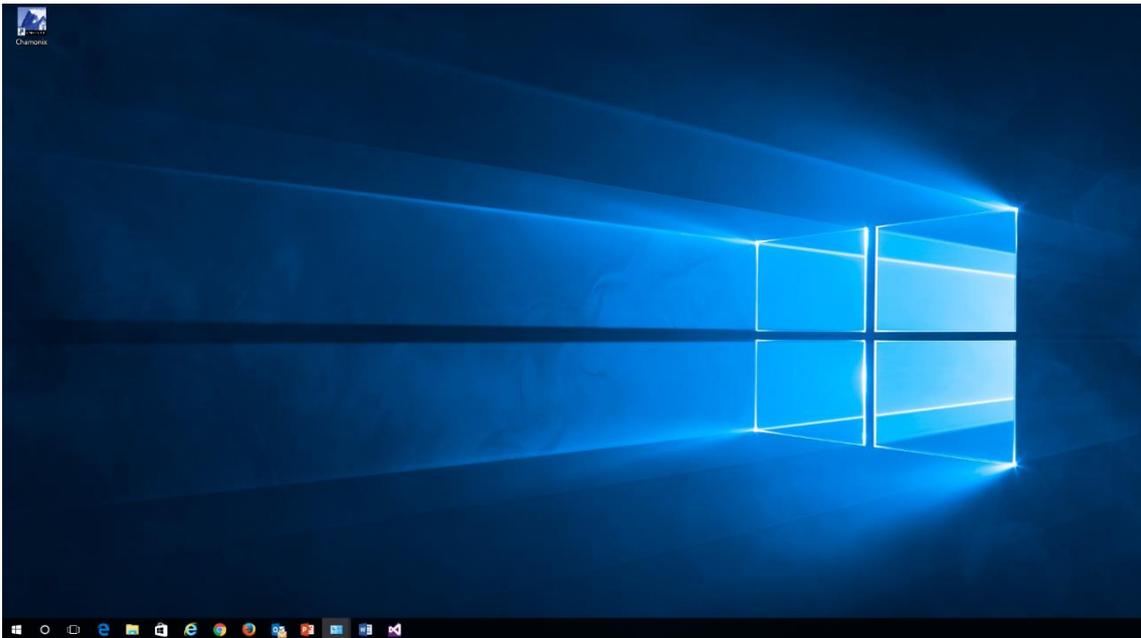
スイッチ設定								通信 モード	RS-232C 設定				GP-IB	
1	2	3	4	5	6	7	8		速度	パリティ	語長	ストップ	デリミタ	アドレス
OFF	OFF	*	*	*	*	*	OFF	RS	38400	*	*	*	*	*
ON	OFF	*	*	*	*	*	OFF	RS	28800	*	*	*	*	*
OFF	ON	*	*	*	*	*	OFF	RS	19200	*	*	*	*	*
ON	ON	*	*	*	*	*	OFF	RS	9600	*	*	*	*	*



Chamonix のインストール直後のボーレートは 115200baud で起動します。SC シリーズはこの速度に対応していないため、このままでは接続できません。38400baud に変更して動作させてください。ボーレートの変更については COM ポートの選択をご覧ください。

5. 起動してみましよう！

5.1. デスクトップのアイコンから起動する



インストールが成功すると、デスクトップには Chamonix のアイコンが生成されます。

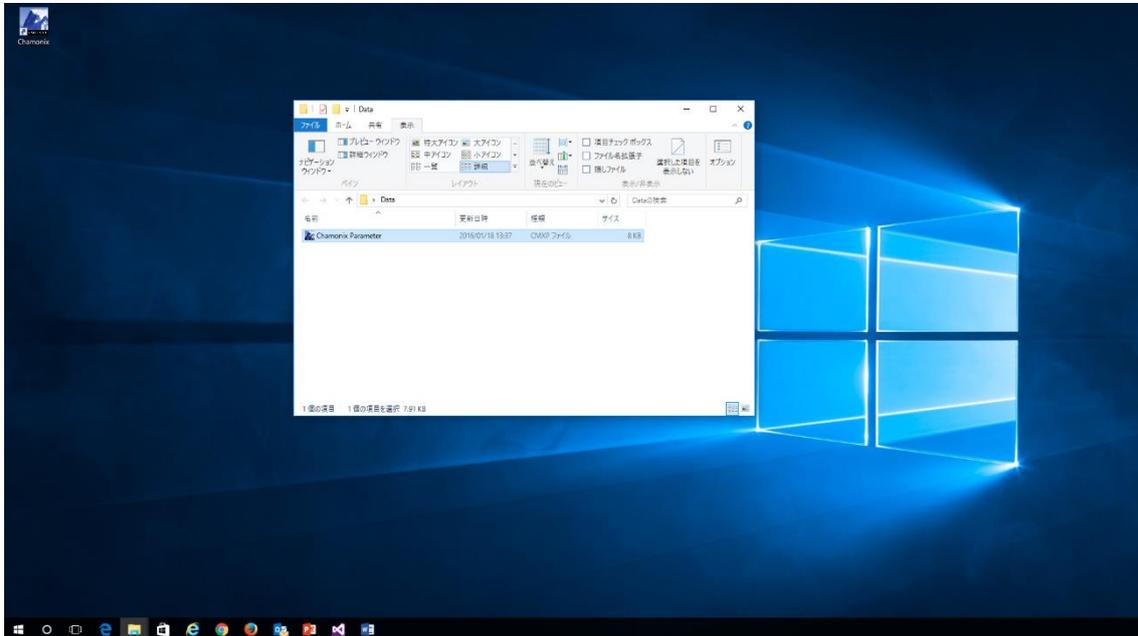
このアイコンをクリックすると前回終了時の時の通信設定で起動します。

初めて起動する場合は、デフォルトの通信設定で起動します。

デフォルトでは RS-232C がインターフェースとして選択されています。

COM ポートの番号は「COM1」、ボーレートは「115200baud」です。

5.2. 保存したファイルから起動する



保存した起動ファイルから起動できます。

起動ファイルにはポート番号、ボーレートなどの情報が保存されています。

保存した起動ファイルの拡張子は「.cmxp」です。

5.3. 起動画面



起動中、上記の画面が表示されます。

インストール時には、通信インターフェースは RS-232C、ポート番号は COM1、ボーレートは 115200baud がデフォルトとして設定されています。

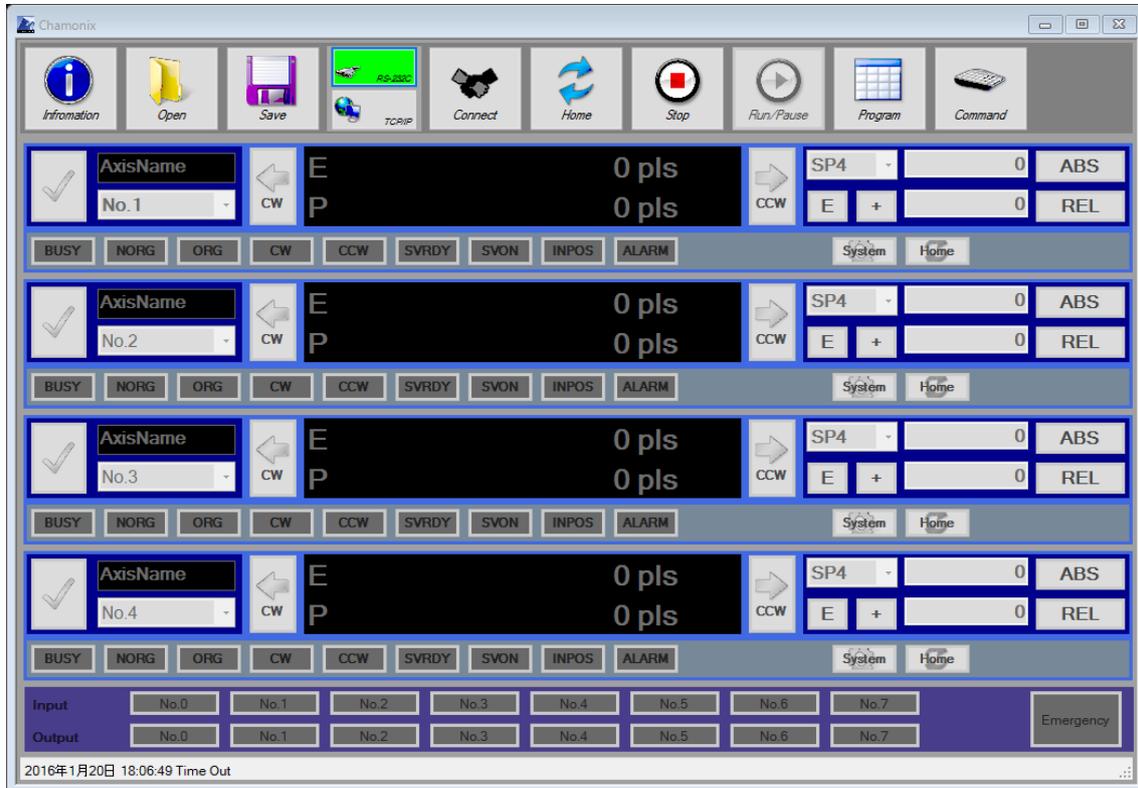
既に起動をしたことがある場合には、前回終了時の情報で起動します。

この値は「C:¥Program Files (x86)¹²¥Common Files¥KOHZU¥Chamonix」内の Config ファイル¹³に格納されています。

¹² 32ビット版の OS の場合は C:¥Program Files ¥Common Files¥KOHZU¥Chamonix です。

¹³ ファイル形式は XML 形式です。このファイルを編集すると Chamonix の起動が行えなくなる場合があります。

5.4. はじめての起動



初めて起動したときの Chamonix の画面の例を示します。デフォルトの通信インターフェースは「RS-232C」、RS-232C の設定はデフォルトでは COM ポートの番号は「COM1」、ボーレートは「115200baud」です。

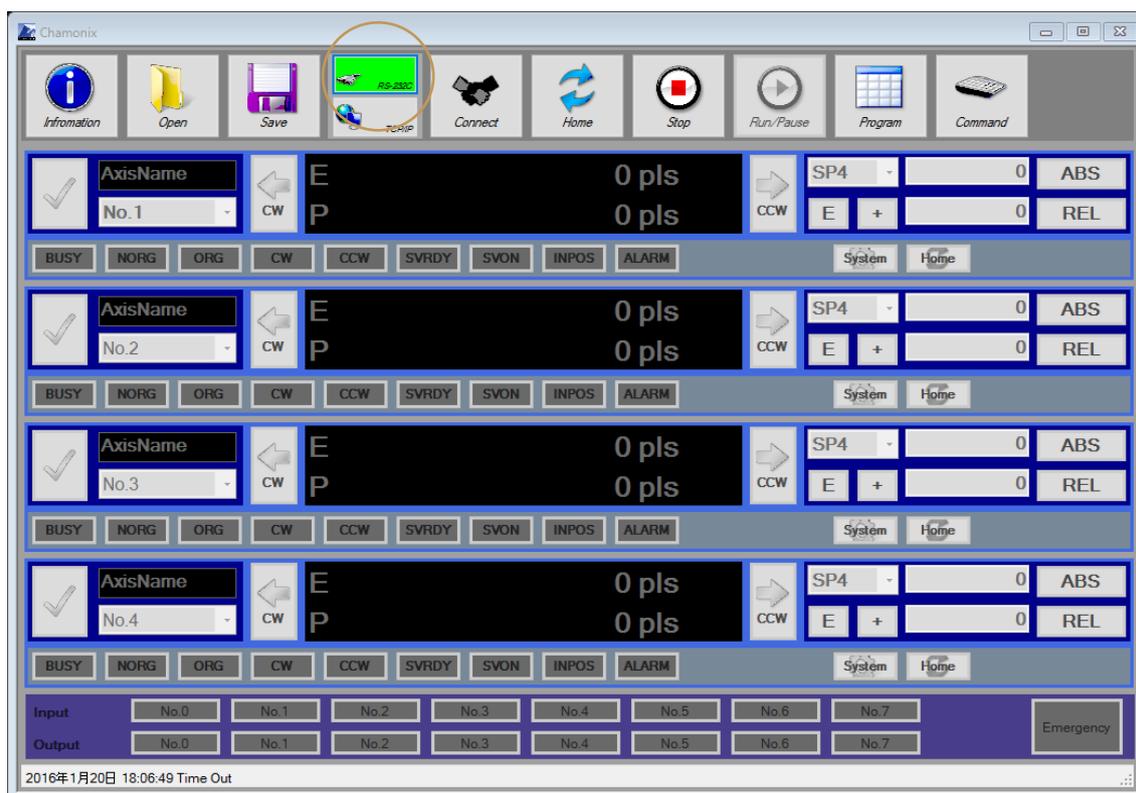
TCP/IP はクライアントとしてコントローラに接続し、接続するコントローラ IP アドレスは「192.168.1.120」、ポートは「12321」に設定されています。

コントローラとの接続設定がデフォルト値と異なる場合、Chamonix を起動しても、コントローラには接続されません。Chamonix がオンとローラに接続できない場合は上の例のように各々の文字がモノクロで表示されます。

6. コントローラに接続しましょう

6.1. RS-232Cでコントローラと接続する

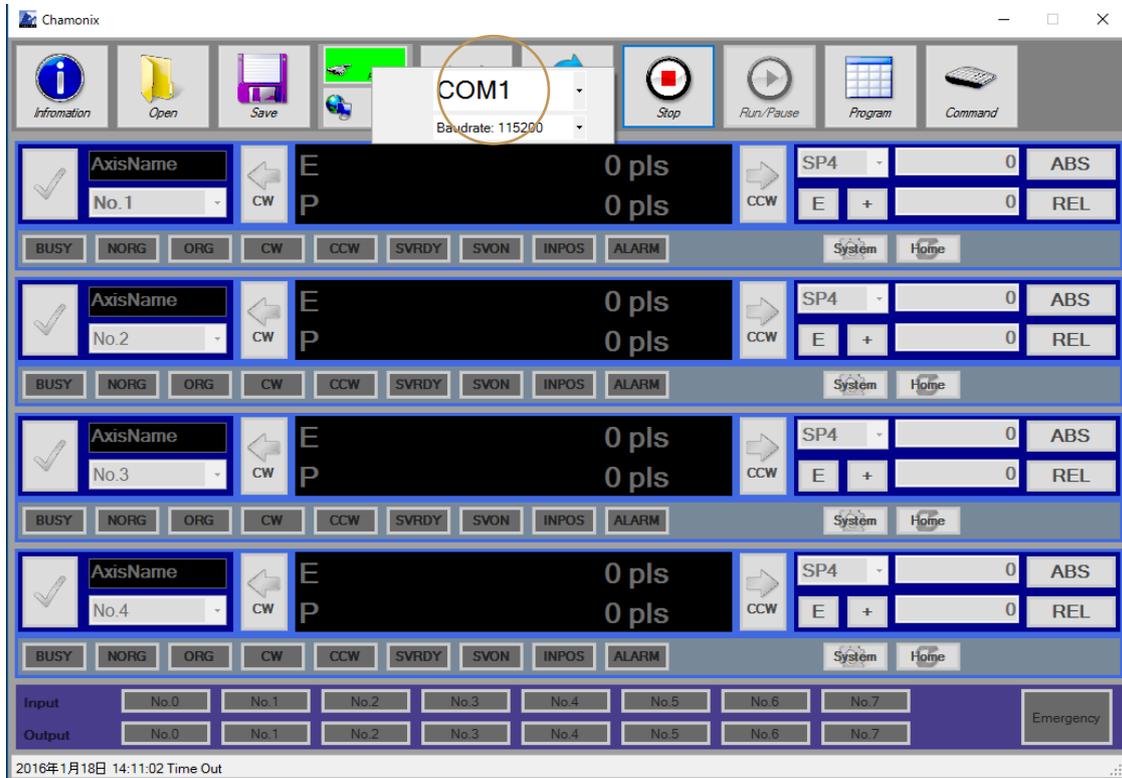
6.1.1. RS-232C を選択する



RS-232 ボタンを押すと、コントローラとの通信インターフェースとして RS-232C 準拠のシリアル通信が選択されます。選択されているインターフェースが図のように点灯します。

CRUX で USB による接続をしている場合も、この RS-232C を選択してください。仮想コネクタ（CDC クラス）として接続します。詳しくは CRUX の取扱説明書をご覧ください。

6.1.2. COMポートの選択



COMポートが選択されていないと図のように駆動軸の制御画面が表示されません。

「RS-232C」ボタンを右クリックしてメニューを表示してCOMポートを設定して下さい。

弊社のコントローラのRS-232C通信のボーレートは115200baudで出荷されています。詳しくは各コントローラの取扱説明書をご覧ください。

「Connect」ボタンを左クリックすると接続確認と初期化を行います。

COMポートが選択されて接続確認が正常に終了した後に、このアプリケーションを閉じて、新たに起動すると設定した値で起動します。

6.1.3. COMポートの番号の確認方法

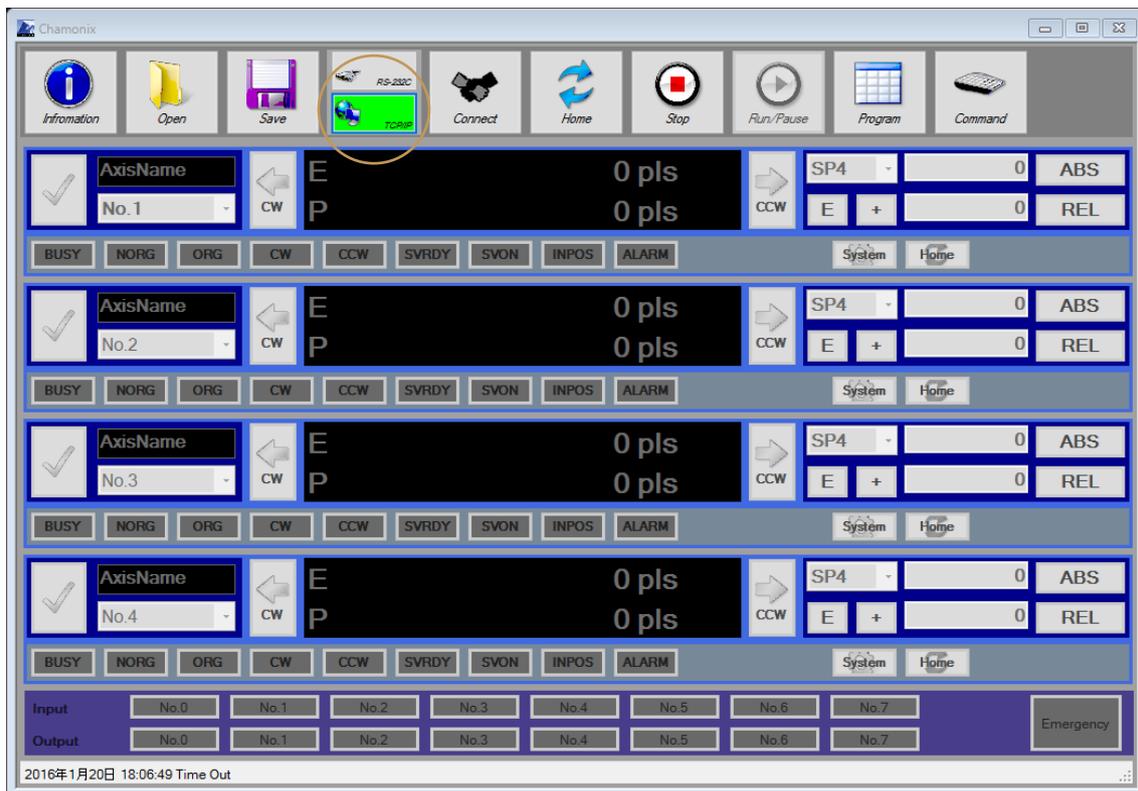


コントローラが接続されているポートをデバイスマネージャーで確認してください。

「コントロールパネル→ハードウェアとサウンド→デバイスマネージャー」で確認できます。デバイスマネージャーの起動には管理者権限が必要です。

6.2. TCP/IP による接続

6.2.1. TCP/IP を選択する (ARIES のみ)



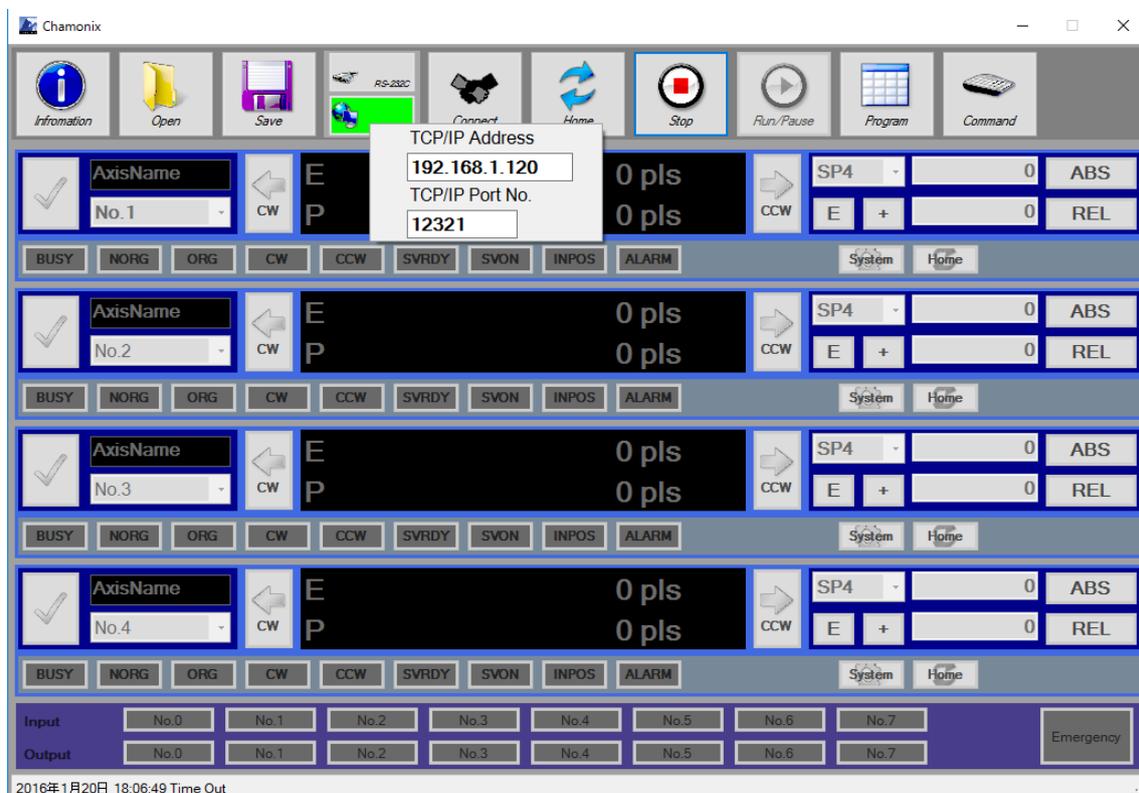
TCP/IP クライアントとしてコントローラに接続することを選択するには「TCP/IP」ボタンを押してください。選択されている通信インターフェースが図のように点灯します。

通信インターフェースとして TCP/IP が選択できるコントローラは ARIES だけです。
他のコントローラでは選択しないでください。

ARIES の「ホスト/クライアント/Telnet の設定」はホストの設定をしてください。ARIES の弊社出荷時にはホストが選択されています。この ARIES の設定には、コントローラに接続した状態で CommandControl (「15.直接リモートコマンドを入力する」を参照して下さい) 等から直接、送信コマンドを入力して送信する必要があります。

送信コマンドの詳細は ARIES の取扱説明書をご覧ください。

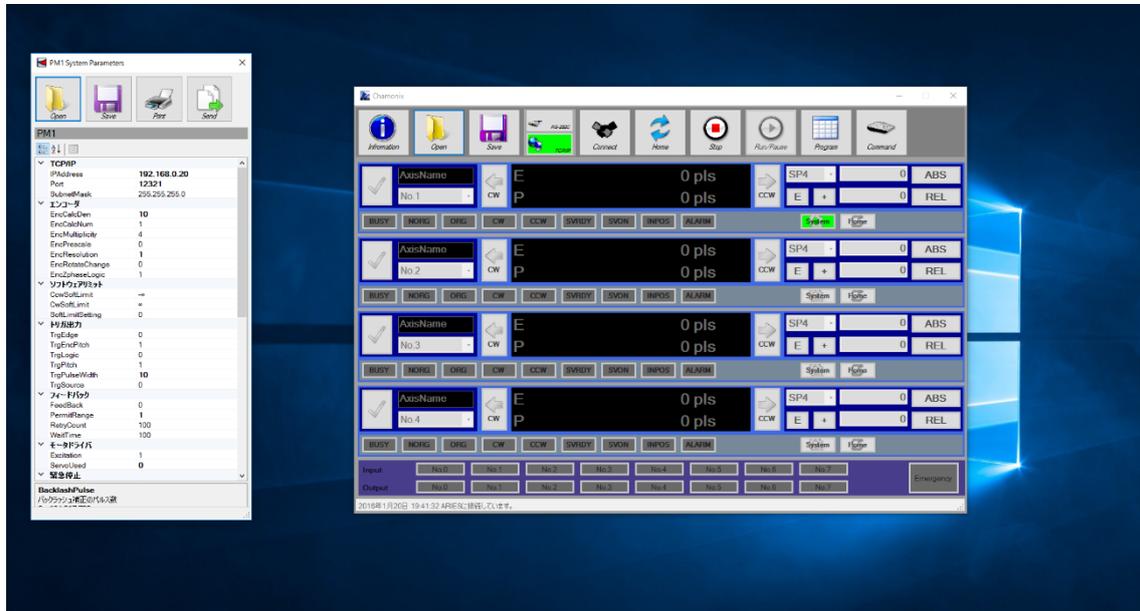
6.2.2. Chamonix の接続先を指定しましょう



「TCP/IP」ボタンを右クリックすると図のようなメニューが開きます。Chamonix が接続するコントローラの IP アドレスとポート番号を入力してください。

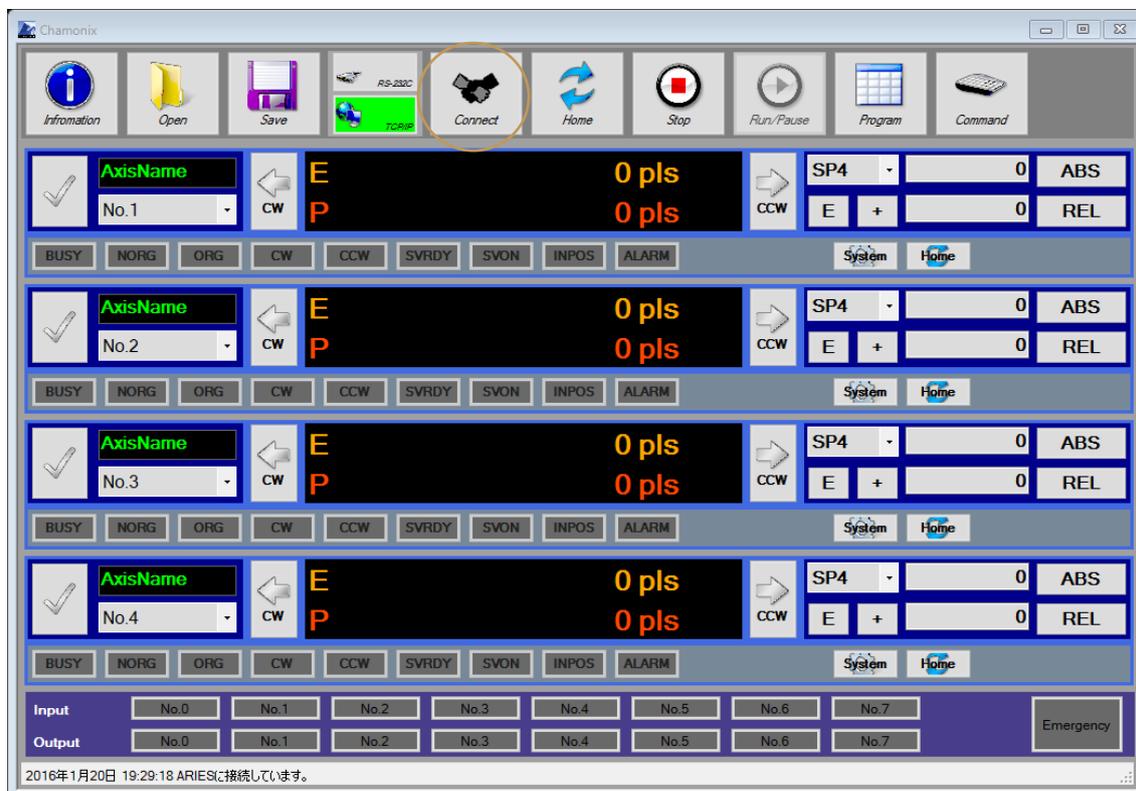
ARIES の工場出荷時のデフォルト値は IP アドレスが「192.168.1.120」、ポート番号は「12321」です。ご使用の環境で ARIES の設定値を変更する必要がある場合には、P2P あるいは RS-232C で接続して ARIES のパラメータを変更してから、Chamonix を再起動しこの設定を行ってください。

6.2.3. ARIES の IP アドレスの変更



ARIES に接続すると ARIES の IP アドレス、ポート番号、サブネットマスクを変更できます。表示されている軸いずれかの「System」ボタンを押すと「System Parameters」ウィンドウが開きます。TCP/IP の項目を設定し、コントローラに「Send」ボタンを押して送信してください。（「システムパラメータを変更する」も参照してください）

6.3. コントローラへの接続



Connect ボタンを押すと接続の確認と初期化を行います。

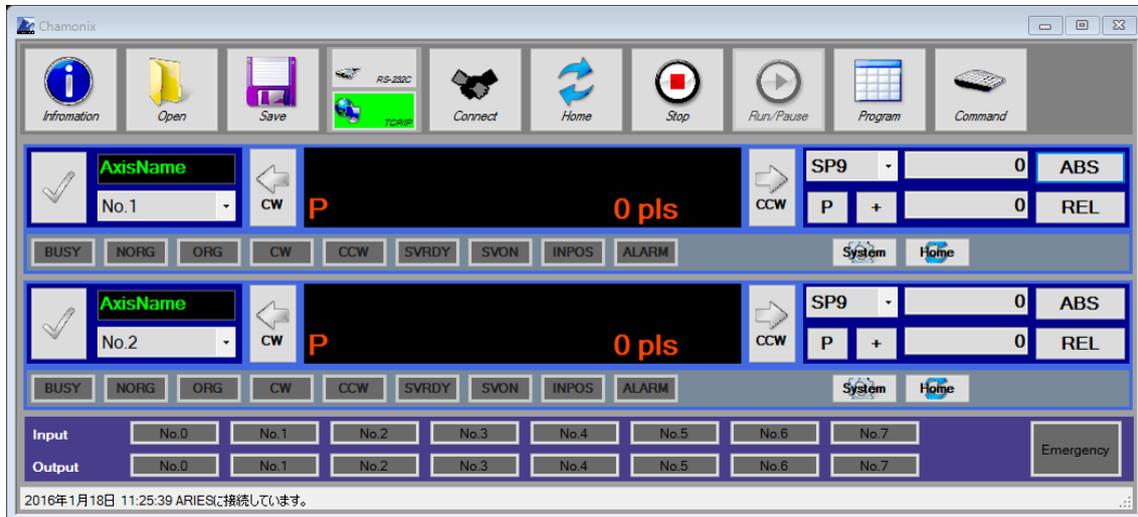
起動時に接続の設定が正常に読み込まれた場合には自動で接続の確認と初期化が行われます。

接続の確認と初期化に時間がかかる場合には、5 秒以上起動に時間がかかる場合があります。

接続が確認された場合、図のように駆動軸の制御画面（StageControl）が表示されます。駆動軸制御画面は最大で 4 軸まで表示されます¹⁴。

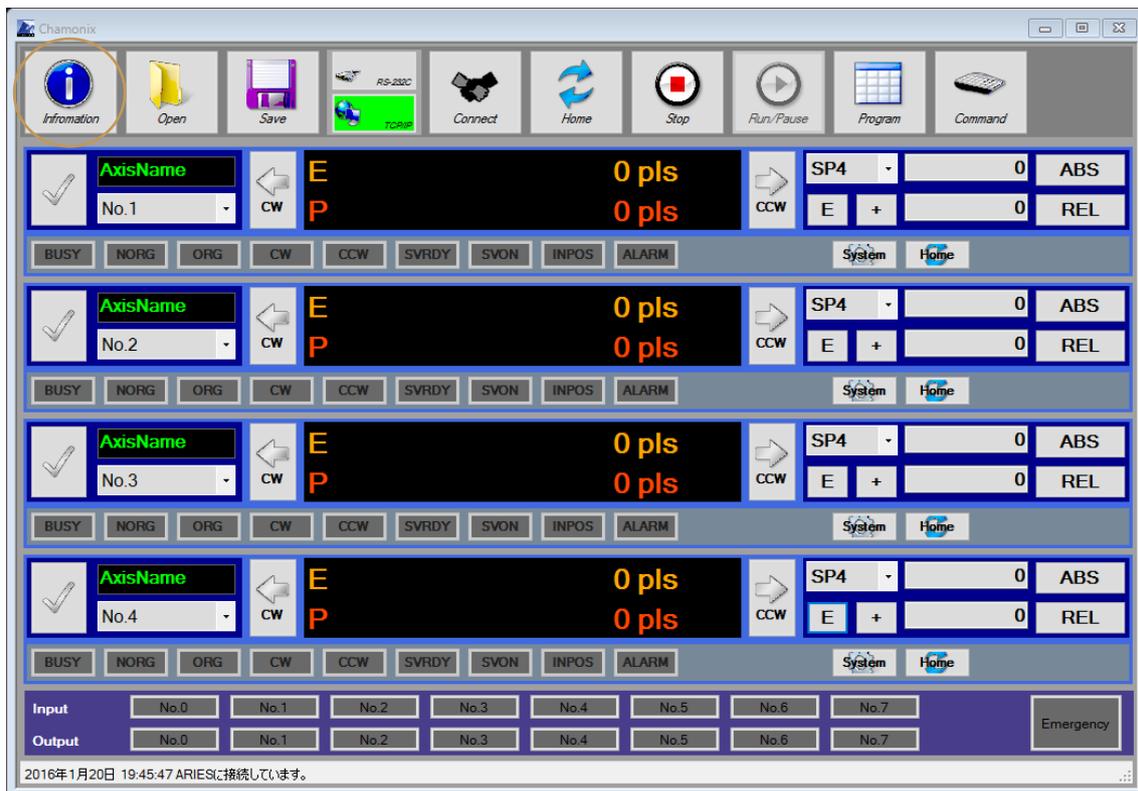
¹⁴ CRUX をご使用の場合、あるいは ARIES にスレーブコントローラ LYNX を接続していない場合など、総軸数が 2 軸の場合は 2 軸だけが表示されます。

CRUX をご使用の場合、あるいは ARIES にスレーブコントローラ LYNX を接続していない場合など、総軸数が 2 軸の場合は 2 軸だけが表示されます。



7. バージョン情報を確認しましょう

7.1. このアプリのバージョン情報を確認する



「Information」タンをクリックすると次頁のようにバージョン情報¹⁵を確認できます。

¹⁵ バージョン番号などの情報は実際と異なる場合があります。

Chamonix のバージョン情報



Chamonix

バージョン 1.1.0.4

Copyright © 2014 Kohzu Precision Co., Ltd.

神津精機株式会社 (Kohzu Precision Co., Ltd.)

ChamonixはWindowsベースのアプリケーションです。

このアプリケーションは著作権に関する法律および国際条約によって保護されています。

このアプリケーションの動作には.NET Framework 4.5.2以上が必要です。

OK

8. 原点復帰をおこないましょう

8.1. 原点復帰とは

「原点復帰」とは、コントローラに設定した原点復帰方式に従い、指定のセンサー付近までしてした速度テーブルの速度を用いて移動したのち、原点を検出するまで、スキャン速度で移動し停止する動作です。

各コントローラで利用できる原点復帰方式は、各コントローラで異なります。それぞれの取扱説明書の「原点復帰方式」をご参照ください。



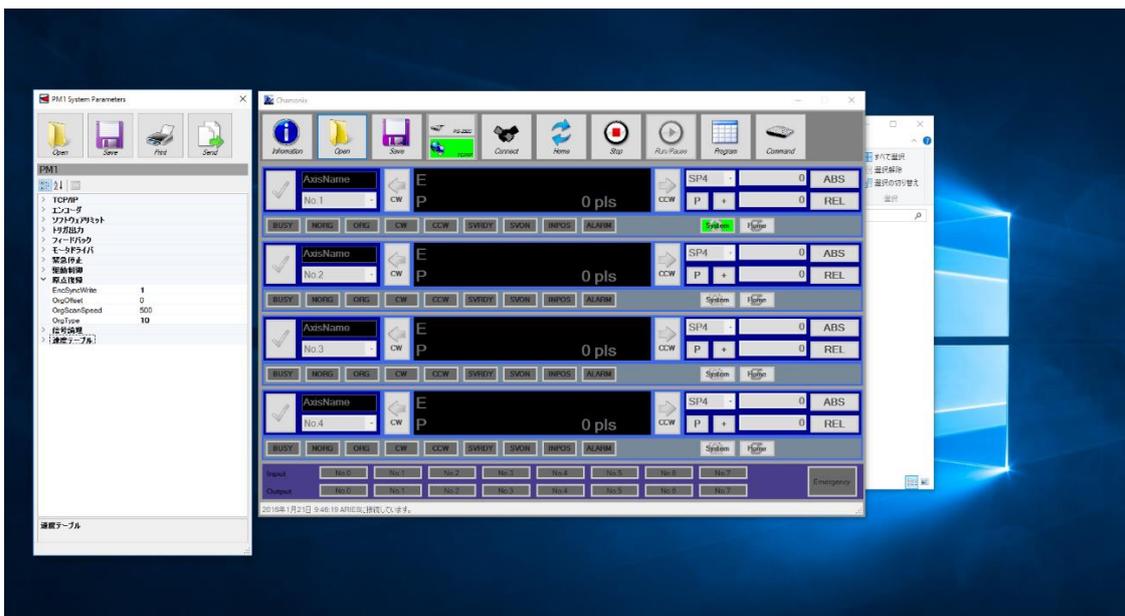
原点復帰を行うときは、全ストロークを駆動する場合があります。

安全を十分ご確認の上実行してください。

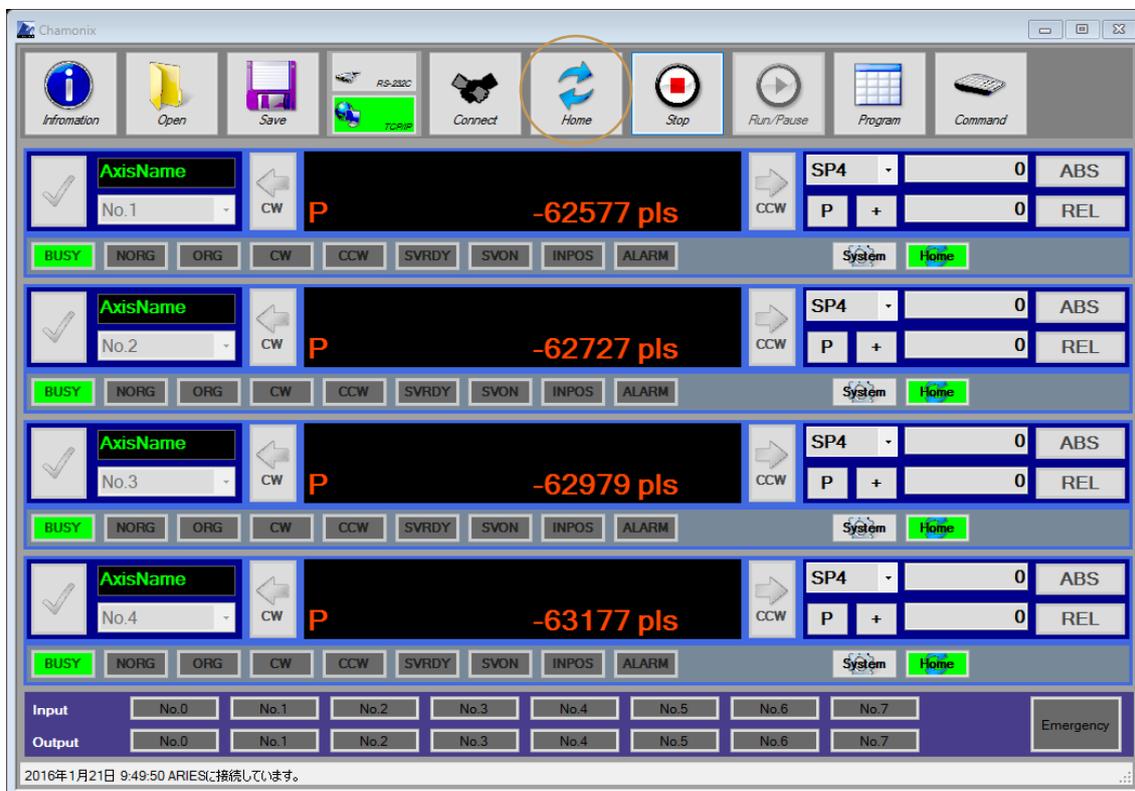
8.2. 原点復帰方式の変更

各軸の原点復帰方式は各軸の「System Parameters」ウィンドウ上の「ORG Type」プロパティを用いて設定できます（「13.2」も参照してください）。

原点復帰方式の詳細はコントローラの取扱説明書の「原点復帰方式」をお読みください。



8.3. 全軸原点復帰



「Home Positions」を押すと、コントローラが制御するすべての軸が原点復帰します。

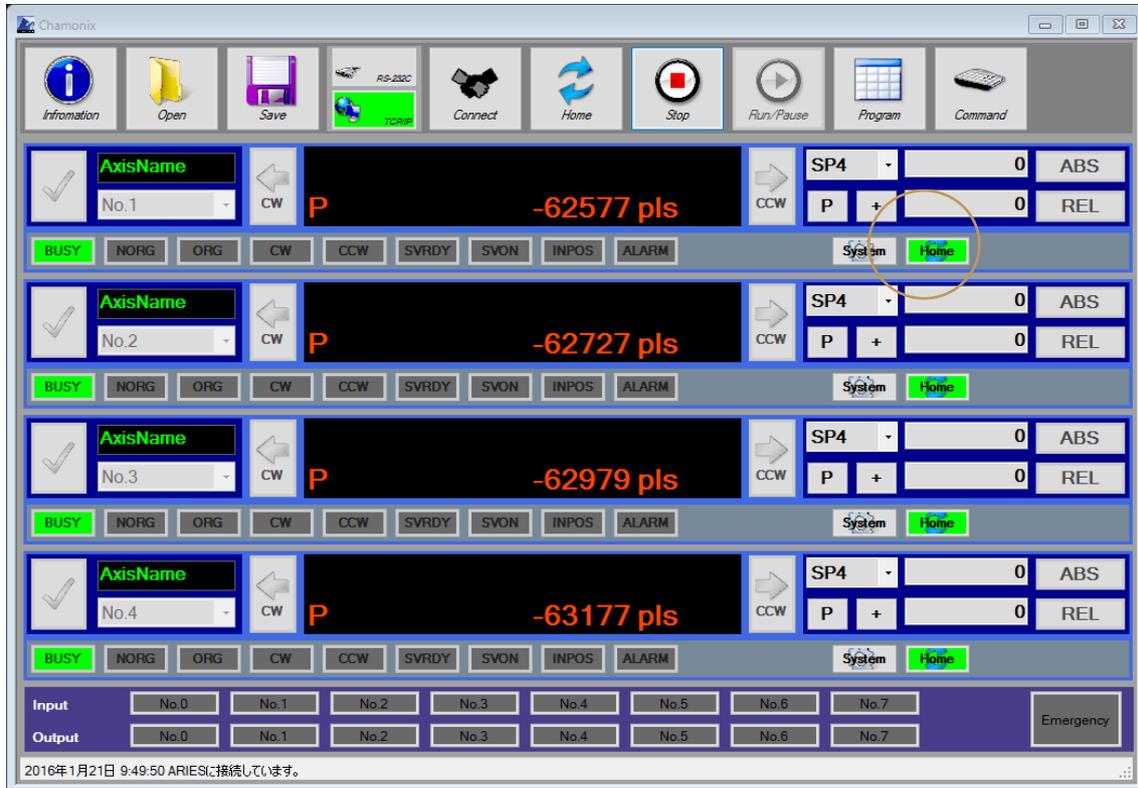
各軸は他の軸の動きを考慮せず独立して原点復帰動作を行います。それぞれの干渉に十分ご注意ください。

軸が干渉する恐れがある場合には、次項で説明する単軸の原点復帰をご利用ください。原点復帰のボタンはご使用のコントローラの説明書をご覧ください。原点復帰方式はコントローラのシステムパラメータの設定値に従います。システムパラメータの設定方法は後述します。



**全軸原点復帰を実行されたことにより、
もし軸同士が干渉し不具合が生じたとしても、当社として責任は負いかねます。**

8.4. 原点復帰中のステータス



原点復帰が実行されている間、各軸の「Home」ボタンが点灯します。

それぞれの軸の原点復帰が完了すると「Home」ボタンは消灯します。

点灯中の「Home」ボタンをクリックするとその軸だけ原点復帰を中止します。

駆動中は「BUSY」ランプも点灯します。

8.5. 単軸の原点復帰



原点復帰は軸の個別の「Recovery」ボタンをクリックすることで、特定の軸のみの原点復帰をすることも可能です。

軸同士が干渉する恐れがある場合には、軸毎に一つずつ原点復帰をされることをお勧めします。

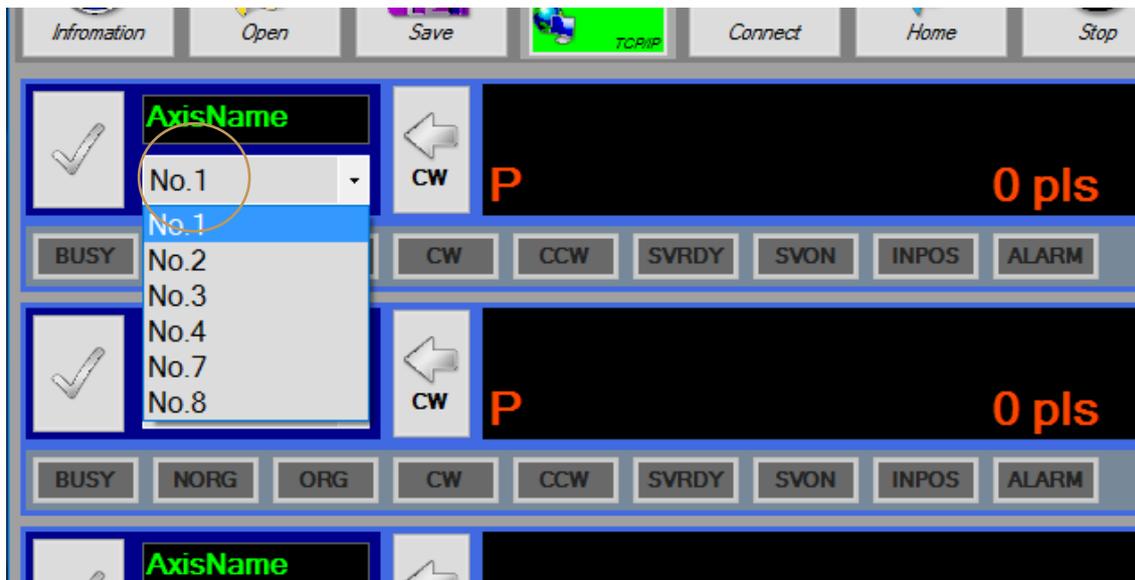
点灯中のボタンを押すと停止します。



全軸原点復帰を実行されたことにより、もし軸同士が干渉し不具合が生じたとしても、当社として責任は負いかねます。

9. ステージを動かしてみよう

9.1. 制御する軸の選択



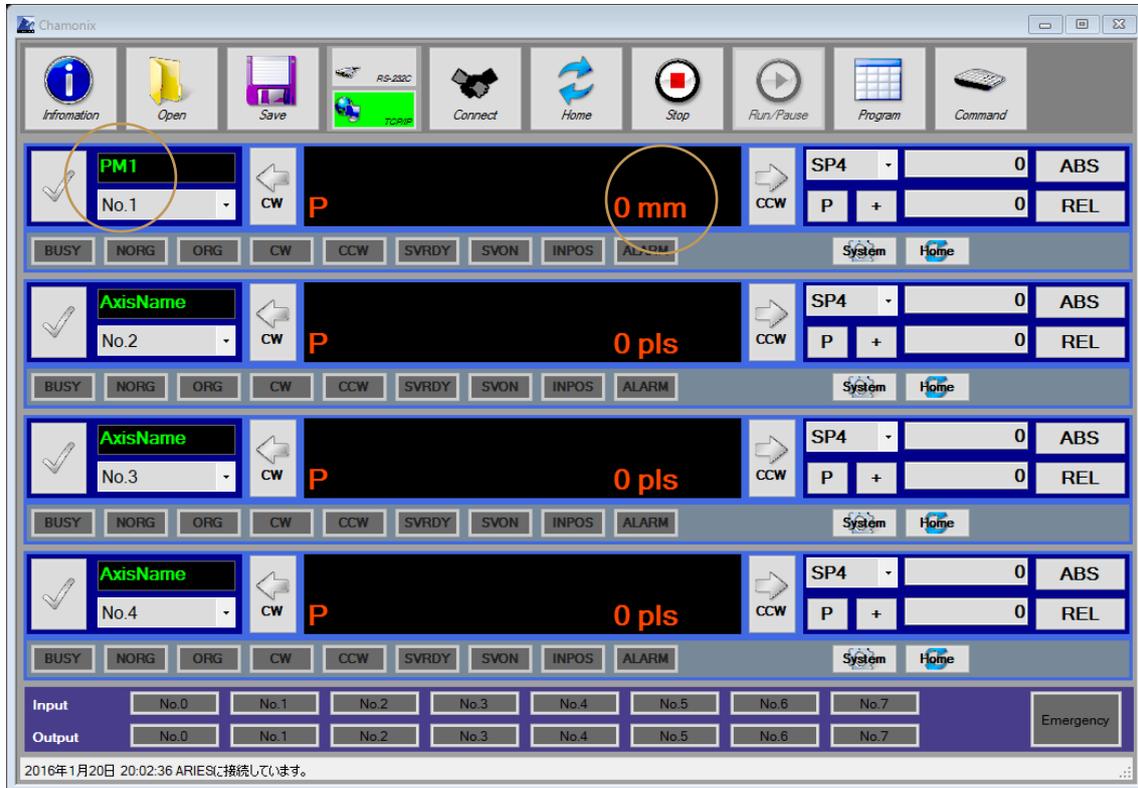
コンボボックスを使用して制御する軸を選択することができます。
駆動中に GUI に表示して制御する軸を変更することも可能です。

表示は最大 4 軸までですが、3 2 軸まで制御できます。

CRUX の場合は 2 軸までの制御になります。

ARIES の場合、LYNX の増設によって最大 32 軸まで制御できます

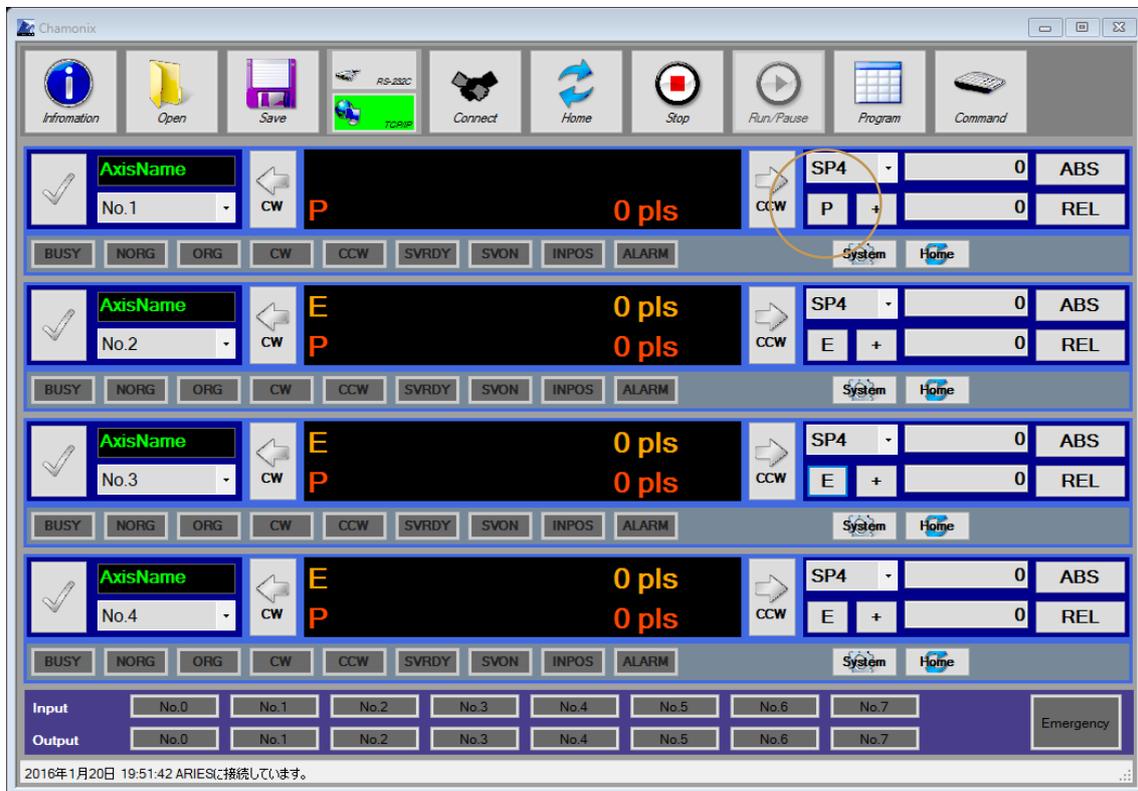
9.2. 軸の名前を設定する



軸の名称を入力できます。軸の単位も変更できます。これらの名前は Windows で入力ができる言語で書き込めます。文字コードは UTF-8 です。

これらの値は「C:\¥ProgramData¥KOHZU¥Chamonix」内の Config ファイルに保存されます。

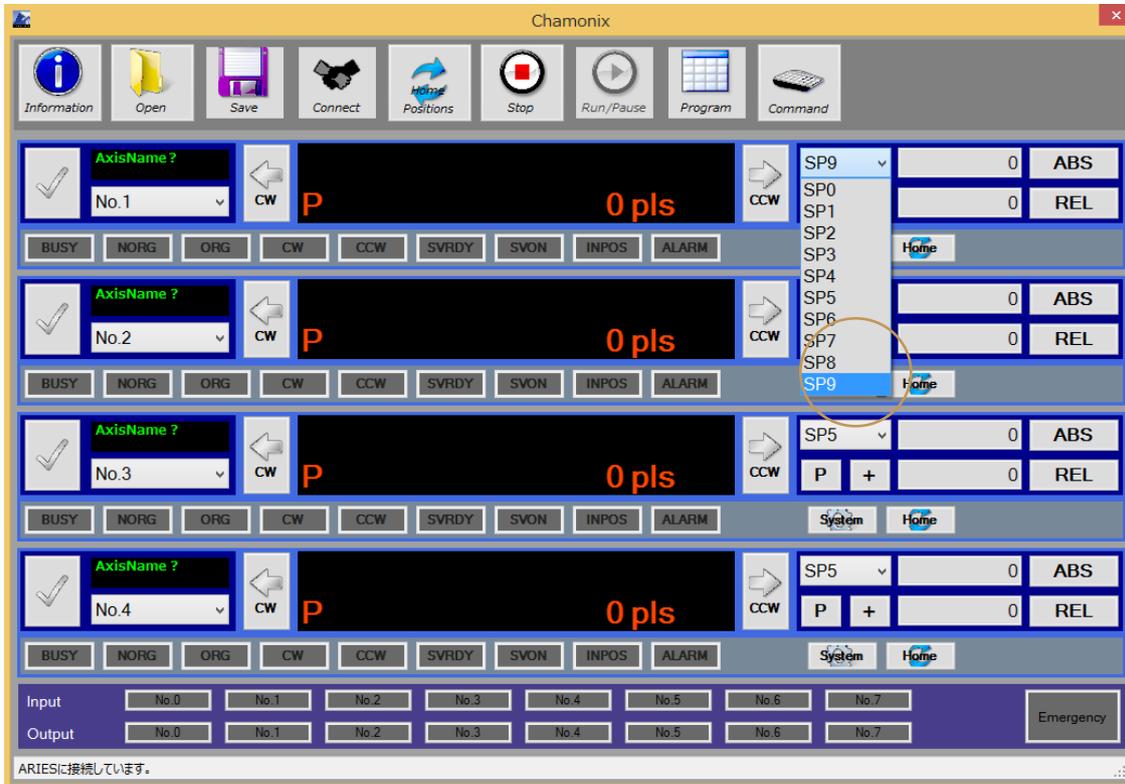
9.3. 現在位置表示の切り替え



「P」ボタンを押下すると表示が「E」へ変わり、エンコーダの現在位置が表示されます。

「E」ボタンを押下すると表示が「P」へ変わり、モーターパルスの現在位置が表示されます。

9.4. 速度テーブルの選択

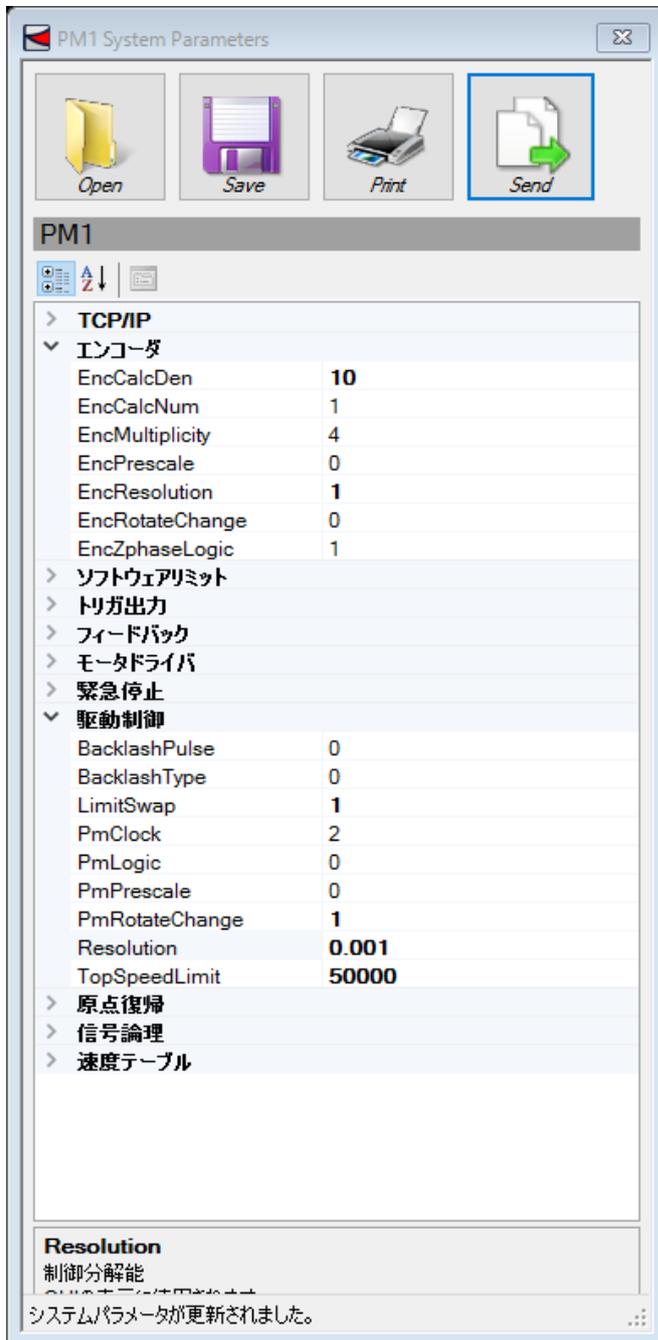


それぞれの軸の速度テーブルはコンボボックスから変更できます。

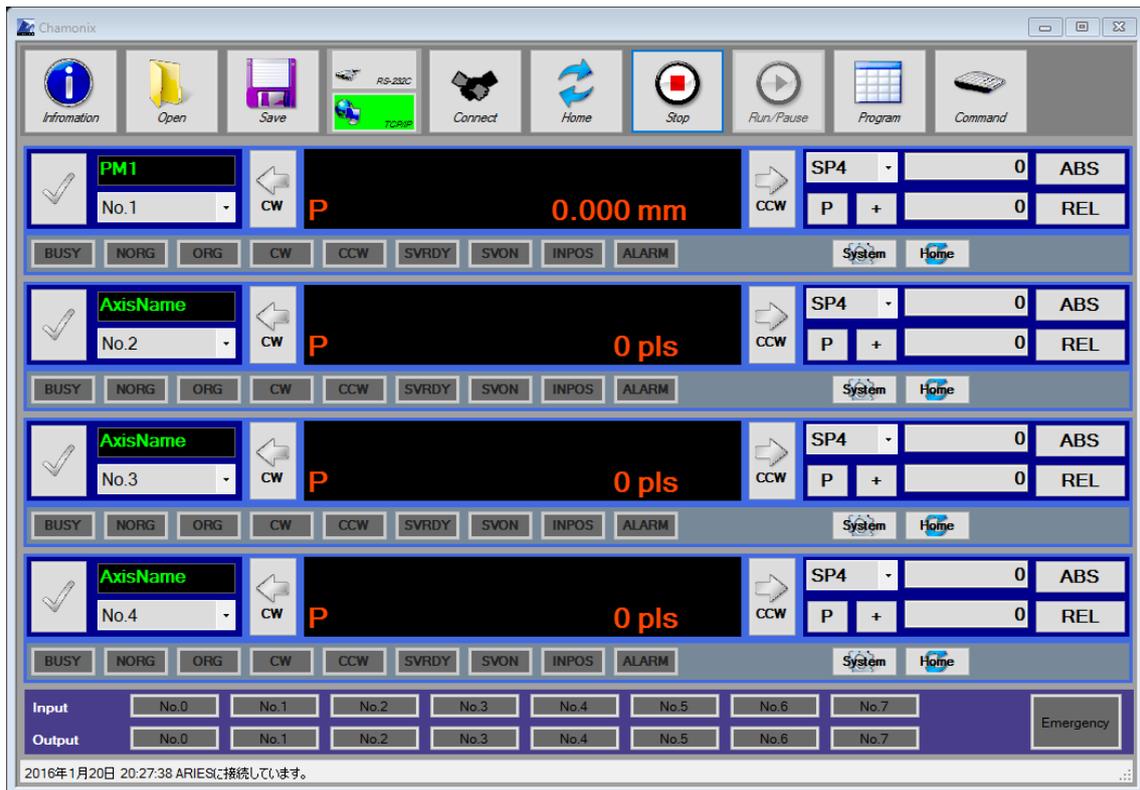
速度テーブルの設定値はシステムパラメータの変更を行うことで設定できます。システムパラメータの変更方法は「システムパラメータを変更する」で説明します。

速度テーブルの各パラメータはコントローラの型式によって異なります。詳しくは各コントローラの取扱説明書をご覧ください。

9.5. 軸の分解能を設定する



軸の分解能は「System Parameters」ウィンドウで設定できます。モーターパルスの分解能は「Resolution」、エンコーダパルスの分解能は「EncResolution」にそれぞれ 1 パルスあたりの値を入れてください。（「13 システムパラメータを変更する」も参照してください）
 メインウィンドウの表示に設定した分解能の値が反映されます。単位はキーボードから直接変更することができます。（EX） PLS→mm



9.6. マニュアルでの移動

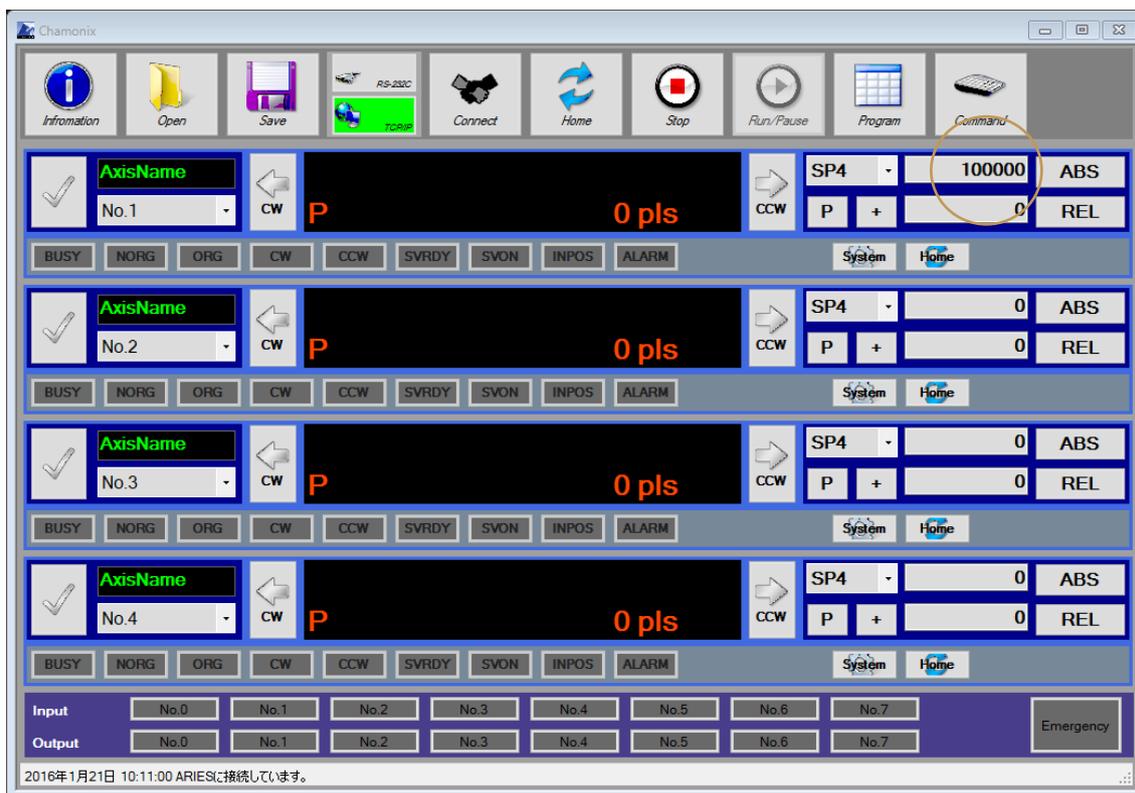


各軸の CW ボタンを押下すると、その間 CW 方向へ移動します。

同様に CCW ボタンを押下すると、押している間 CCW 方向へ移動します。

10. 位置決め

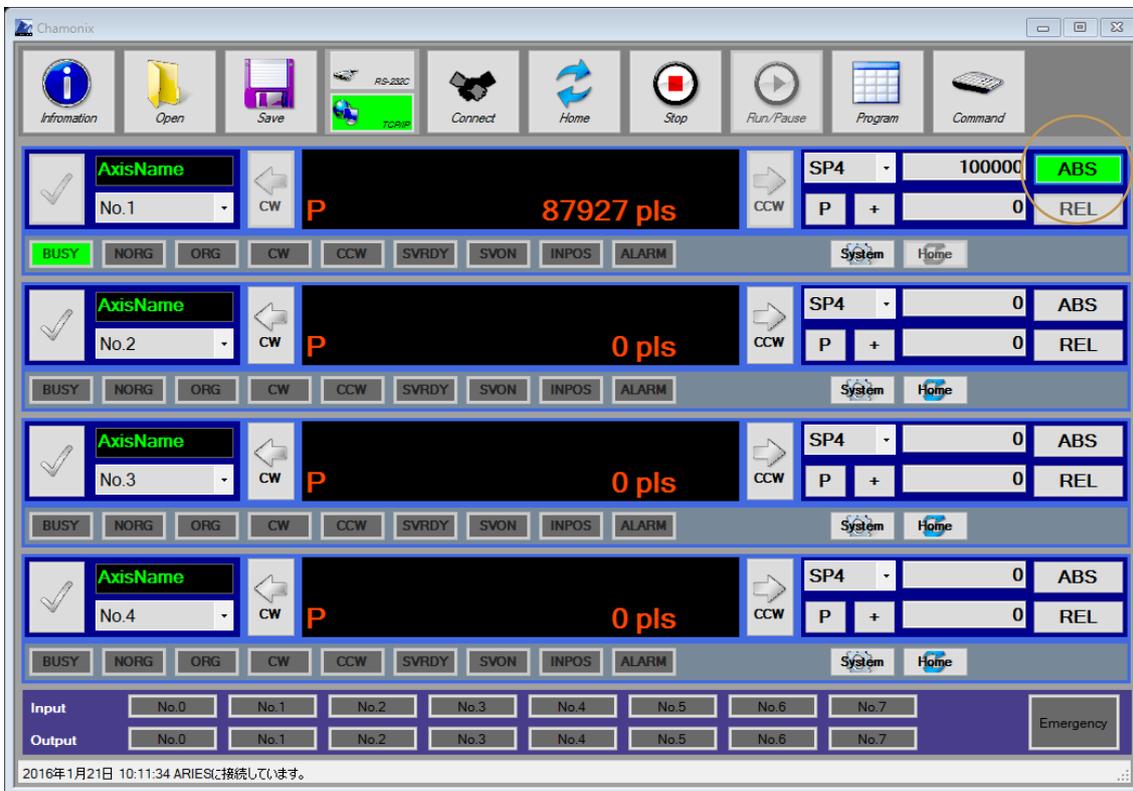
10.1. 絶対位置管理による位置決めの設定



このテキストボックスに移動先の絶対位置を入力してください。

システムパラメータに制御分解能の換算値が設定されている場合、換算値を入力してください。デフォルトではモーターパルスを入力する設定になっています。

10.2. 絶対位置管理による位置決めの実行

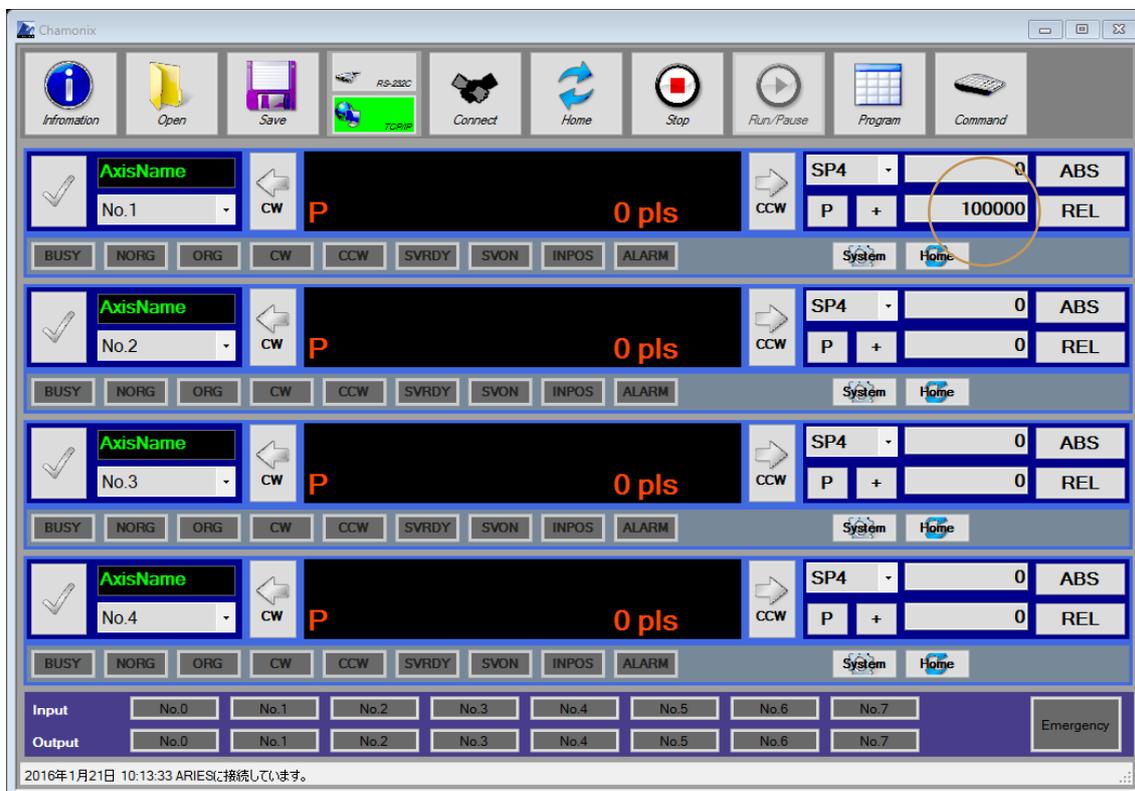


絶対位置を指定し、ABS ボタンをクリックすると、絶対位置管理で位置決めを実行します。

点灯中の「ABS」ボタンを押すとこの軸のみが停止します。

モータ駆動中は BUSY ランプが点灯します。

10.3. 相対位置管理による位置決めの設定

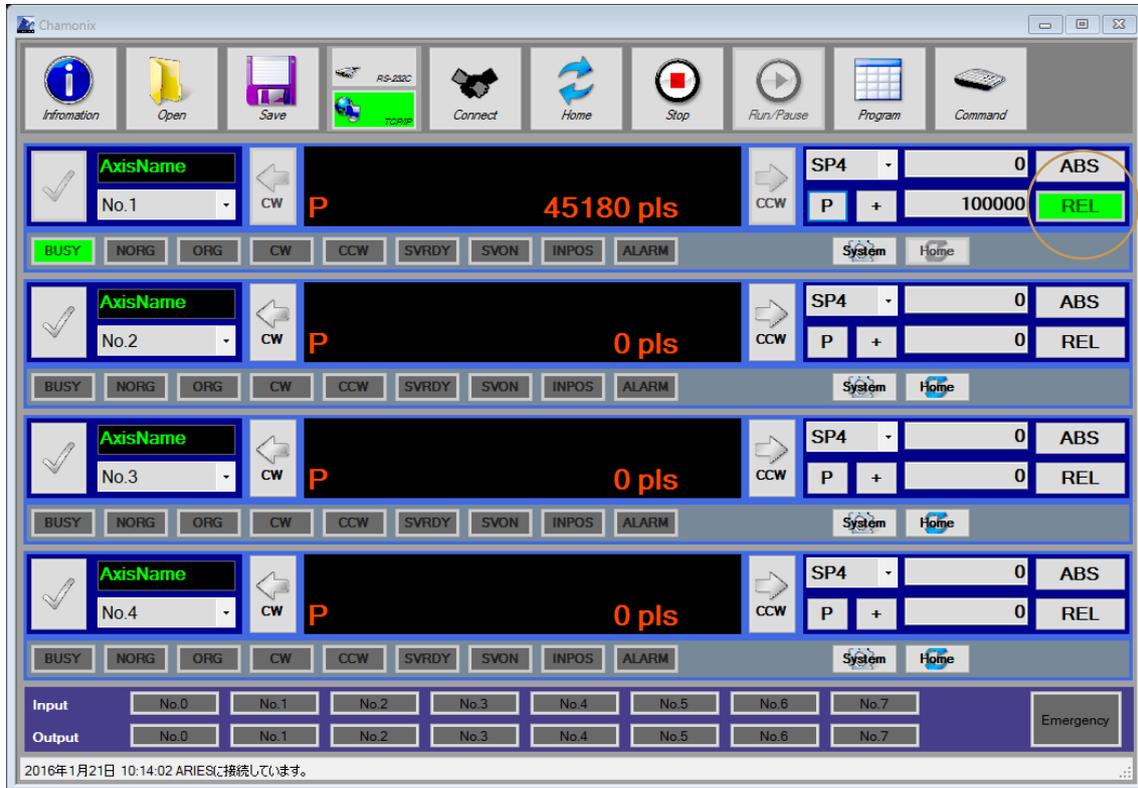


このテキストボックスに移動先の相対移動量を入力してください。

符号のボタンをクリックすると符号を変更できます。

システムパラメータに制御分解能の換算値が設定されている場合、換算値を入力してください。デフォルトではモーターパルスを入力する設定になっています。

10.4. 相対位置管理による位置決めの実行



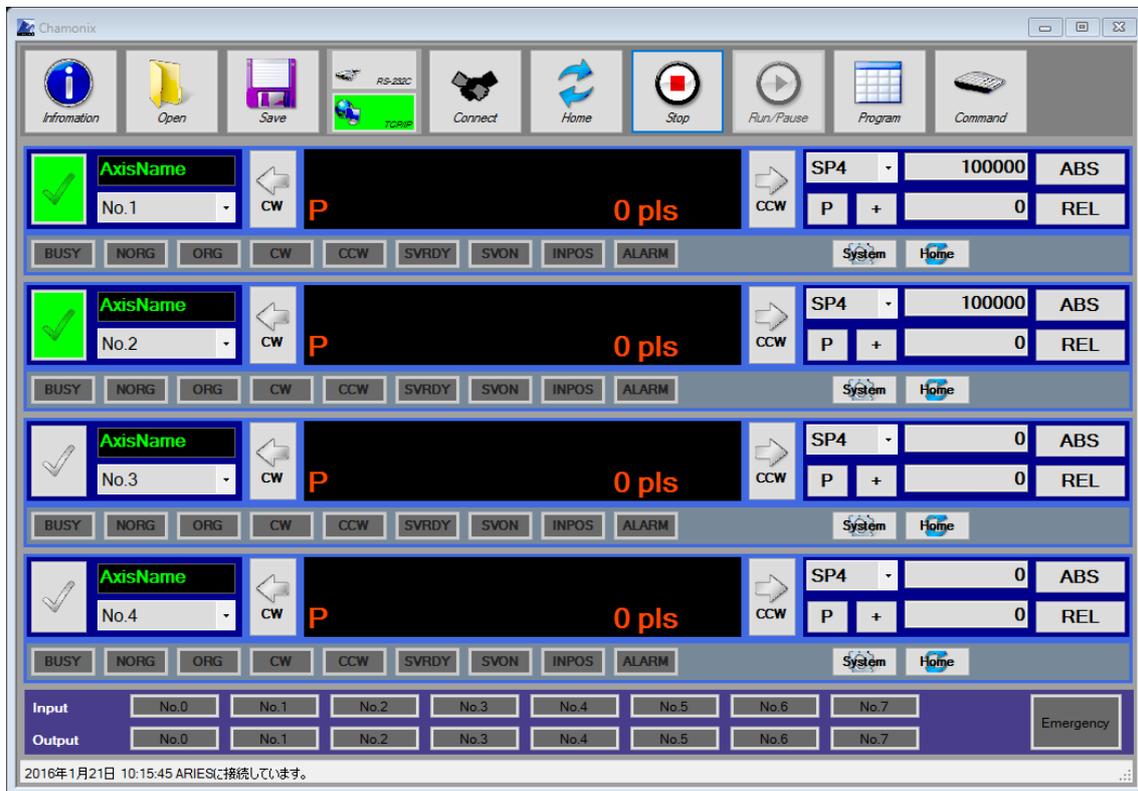
相対位置を指定し、REL ボタンをクリックすると、相対位置管理で移動します。

点灯中の「REL」ボタンを押すとこの軸のみが停止します。

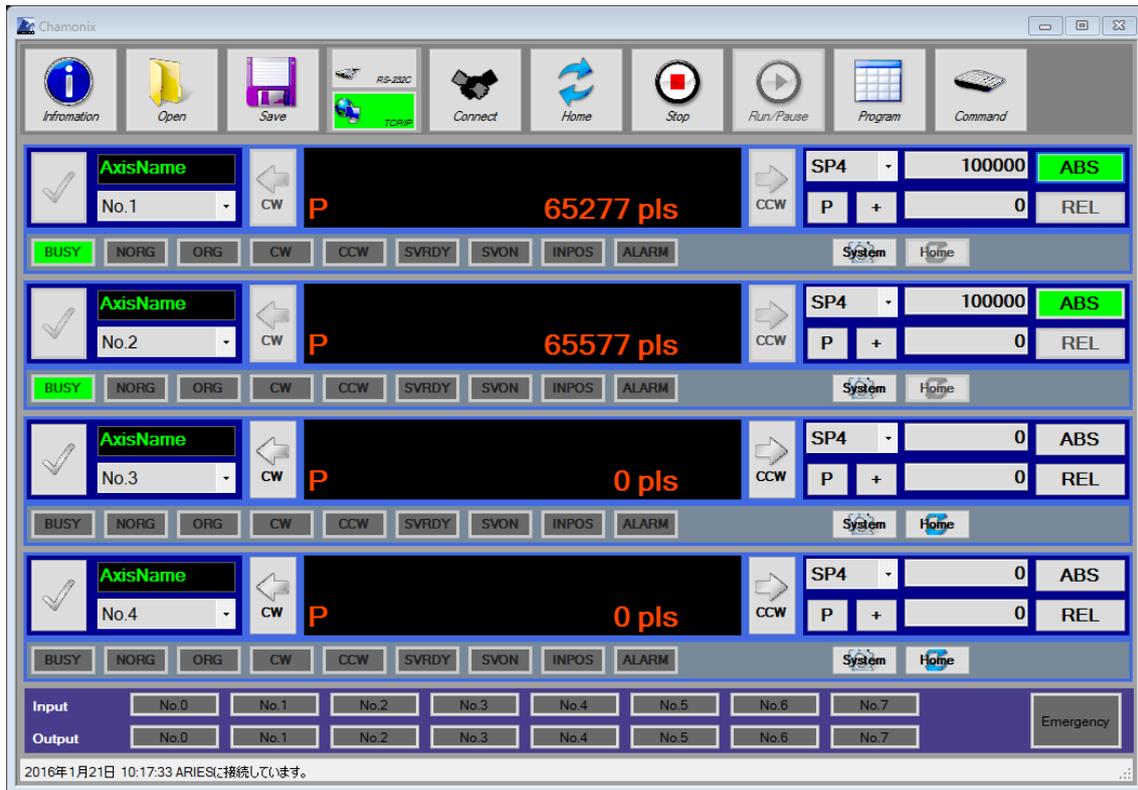
駆動中は BUSY ランプが点灯します。

10.5. 多軸駆動を実行

各軸のチェックボタンを選択してから



「ABS」あるいは「REL」ボタンを押すと選択された軸を続けて駆動できます。各軸は同時に駆動を開始します。

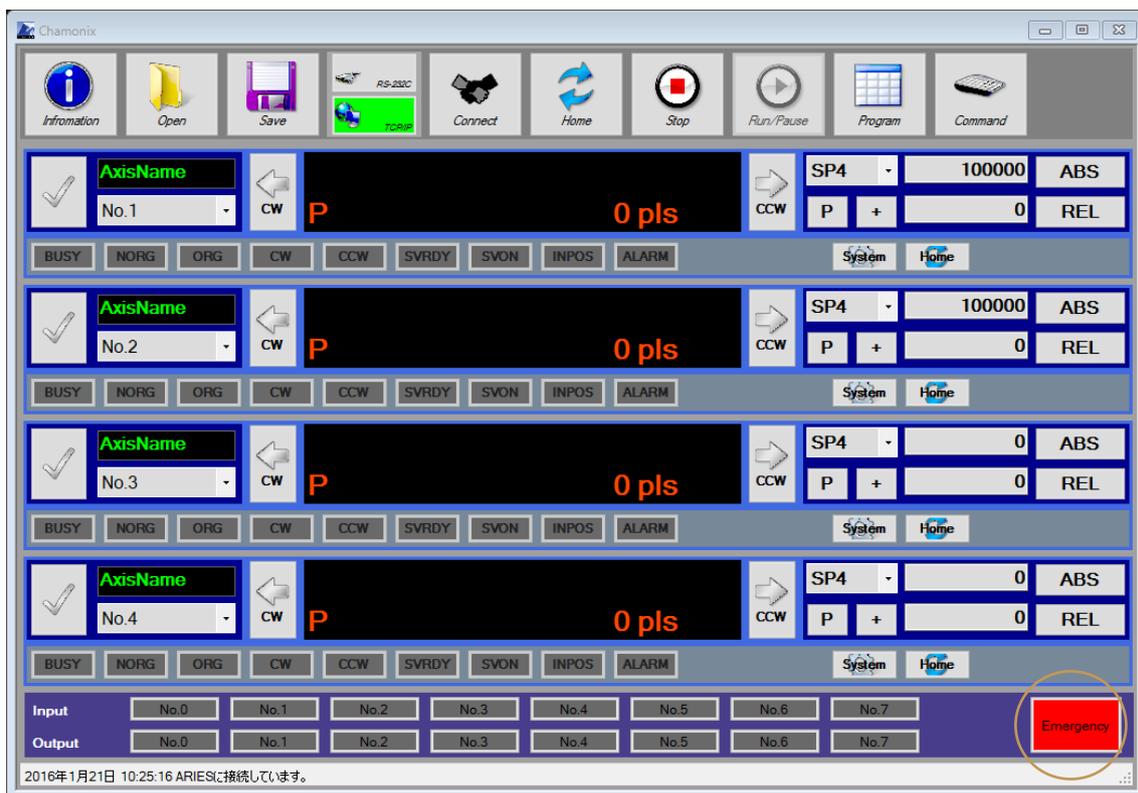


駆動中は「BUSY」ランプが点灯し、選択された駆動方式に応じて「ABS」(絶対位置管理)か「REL」(相対位置管理)のボタンが点灯します。

GUI の表示は一軸ごとに駆動ステータス、現在位置を読み込んでいる関係で、読み込みにタイムラグが生じているため同じタイミングで更新されません。

11. 非常停止を受信！

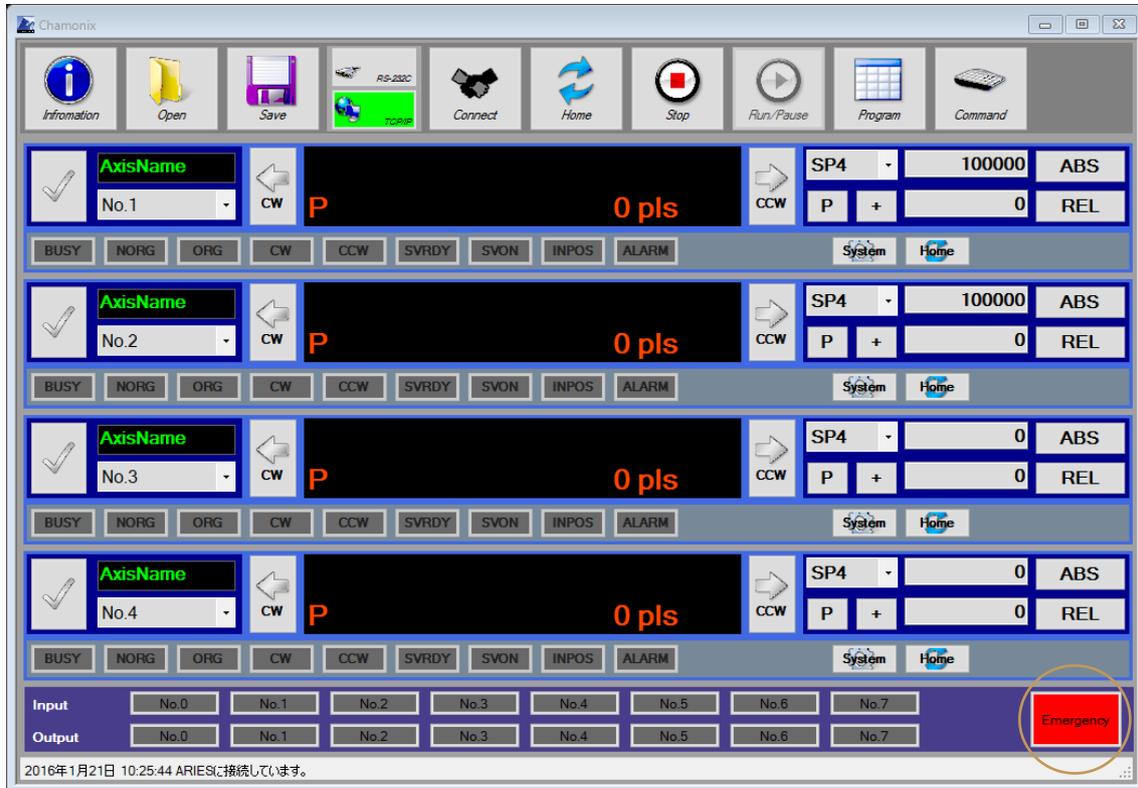
11.1. 非常停止信号を受信（ARIES）



コントローラに ARIES を利用している場合、非常停止信号を受信すると「Emergency」が赤く点灯します。

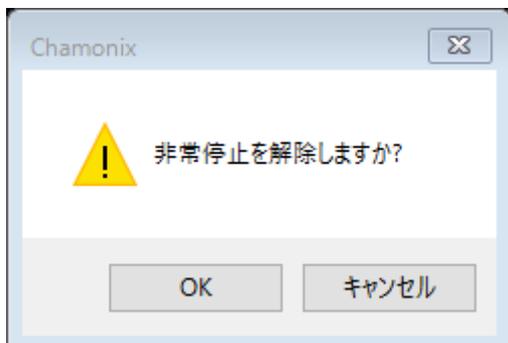
この状態ではコントローラはモーターパルスを一切発生しません。このため各モータは駆動できません。

11.2. 非常停止からの復帰 (ARIES)



ARIES をご利用の場合、非常停止信号の回路を復帰し、安全を確認した後、「Emergency」ボタンを押すと非常停止が解除できます。

「Emergency」を押下すると下図のメッセージが表示されます。「OK」を押すと非常停止が解除されます。「キャンセル」を押すと非常停止は継続します。



非常停止から復帰する際、非常停止の原因が取り除かれ、安全が確保されていることを十分に確認してください。

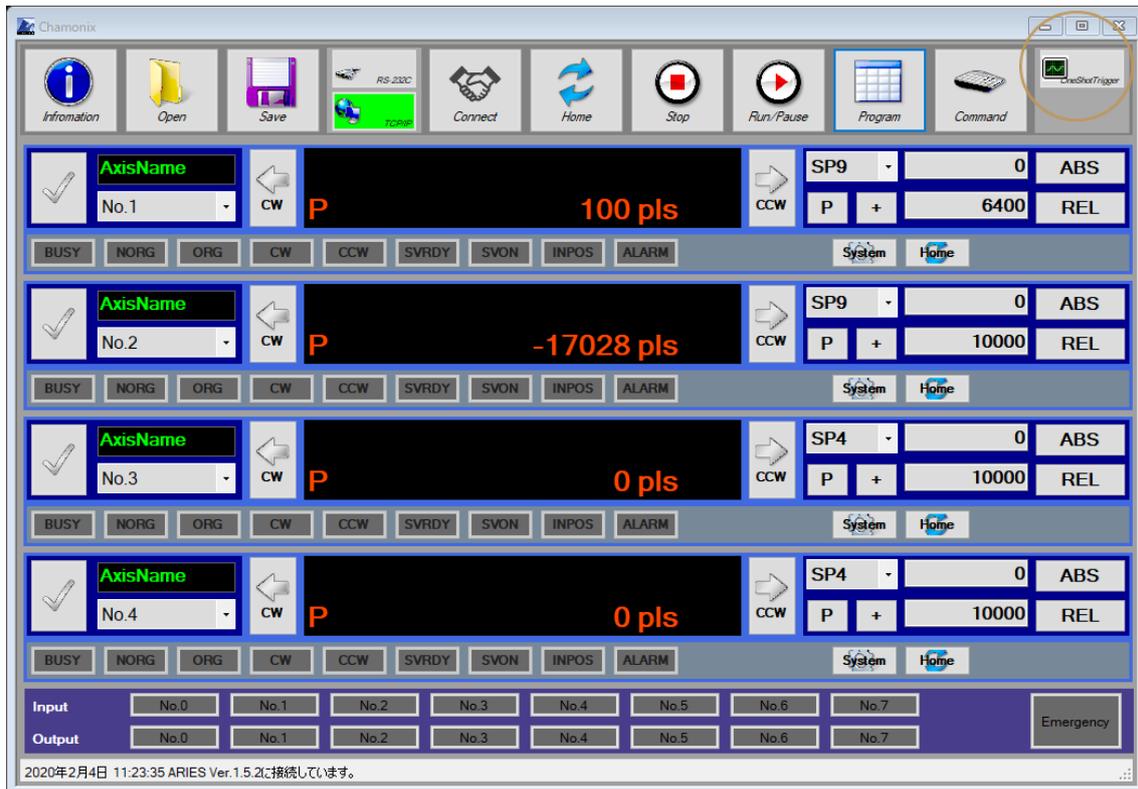
11.3. 非常停止からの復帰（CRUX）

CRUX の場合には、非常停止信号を Chamonix 側で表示することはありません。CRUX のフロントパネルで「EMG」が点灯している場合、非常停止が働いています。

非常停止信号の回路を復帰してください。

12. トリガー出力

12.1. ワンショットトリガー信号の出力 (ARIES) ¹⁶

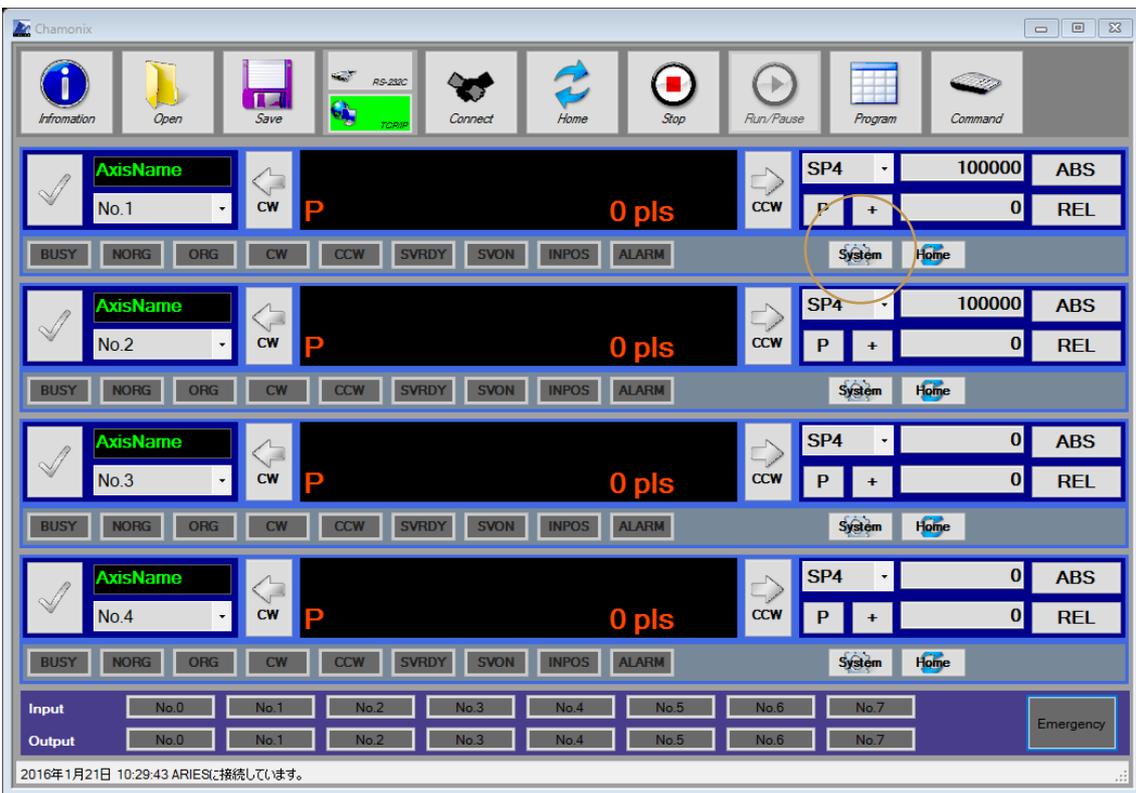


ARIES をご使用の場合は、外部のデバイスにトリガーを出力することが可能です。「OneShotTrigger」ボタンを押下すると、トリガー信号が出力されます。トリガー信号のパルス幅は「OneShotTrigger」ボタンを右クリックすると設定できます。

¹⁶ Chamonix Ver.1.2.3.*以降で使用可能です。

13. システムパラメータを変更する

13.1. パラメータ設定を起動する



各駆動軸の「System」ボタンをクリックするとシステムパラメータの設定ウィンドウが「System Parameters」起動します。起動時にコントローラからシステムパラメータを受信します。各軸のシステムパラメータは次頁に示したプロパティとして開きます。

設定できるパラメータはシステムパラメータと速度テーブルです。これらのパラメータはできる限りコントローラ間の互換性を維持するように設計されています¹⁷。

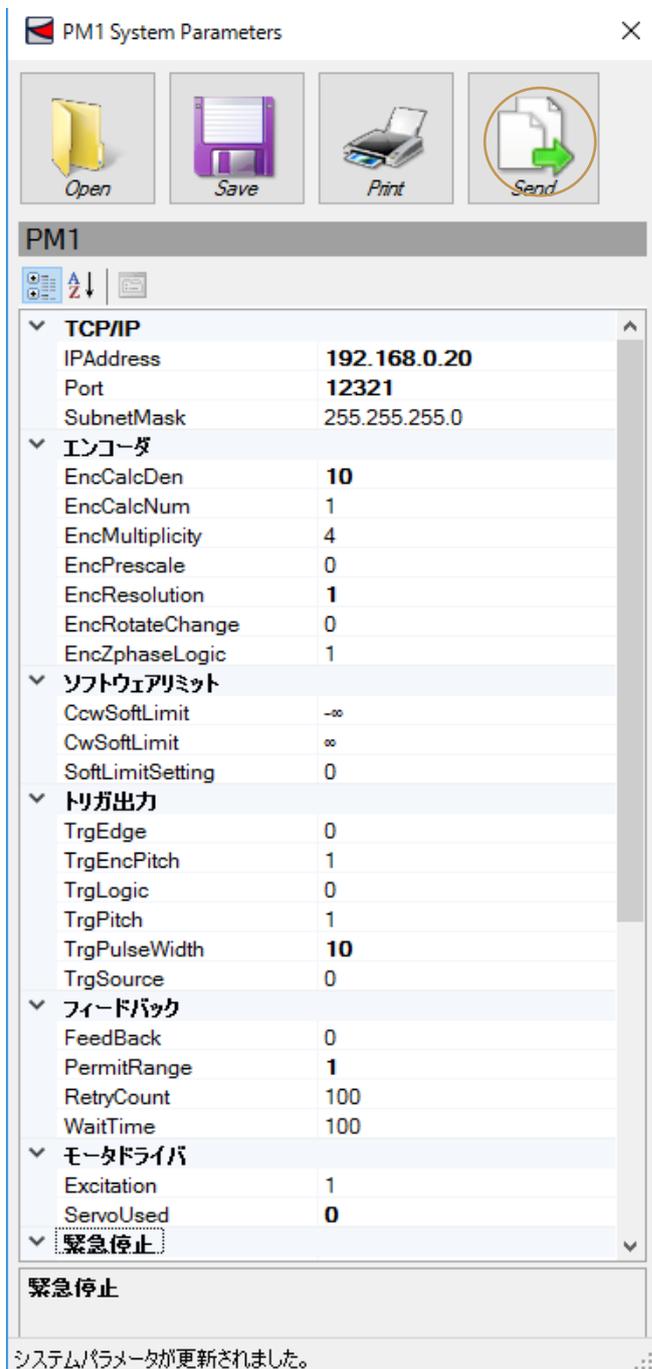
¹⁷異なる機種で完全に互換性を保証するという意味ではありません。



接続されているコントローラで欠けている機能は表には記載されません。上図は ARIES の例を示しています。保存されたファイルには、Chamonix のデフォルト値が保存されます¹⁸。

¹⁸ 上位機種において、下位機種で保存されたファイルを読み込んだ場合には、ファイルに保存されていないパラメータは Chamonix のデフォルト値で起動されます。

13.2. パラメータを設定する



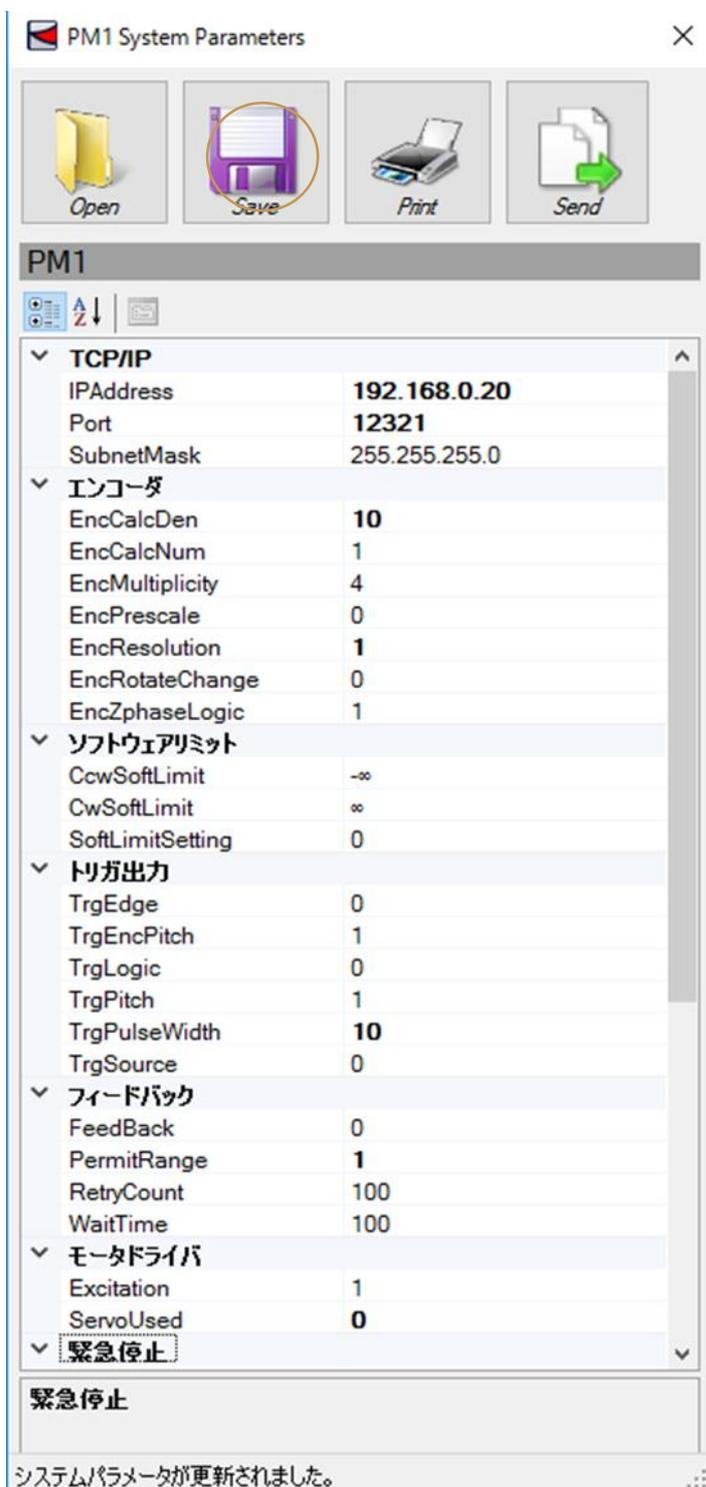
「System Parameters」には、コントローラのシステムパラメータの値を入力することにより変更できます。

システムパラメータの値の変更は「Send」ボタンを押すと一括してコントローラに反映されます。

アプリケーションのプロパティと一体化することでコントローラの機種ごとの番号の違いを吸収しています。

コントローラのデフォルト値が設定されているときプロパティの値はノーマルで、デフォルト値と異なる値が設定されているとき値はボールドで表示されます。

13.3. システムパラメータのファイルへの保存



「Save」ボタンを押下するとシステムパラメータは軸毎にファイルに保存できます。

保存したファイルはこのウィンドウに「Open」ボタンを押下することで読み込むことが可能です。

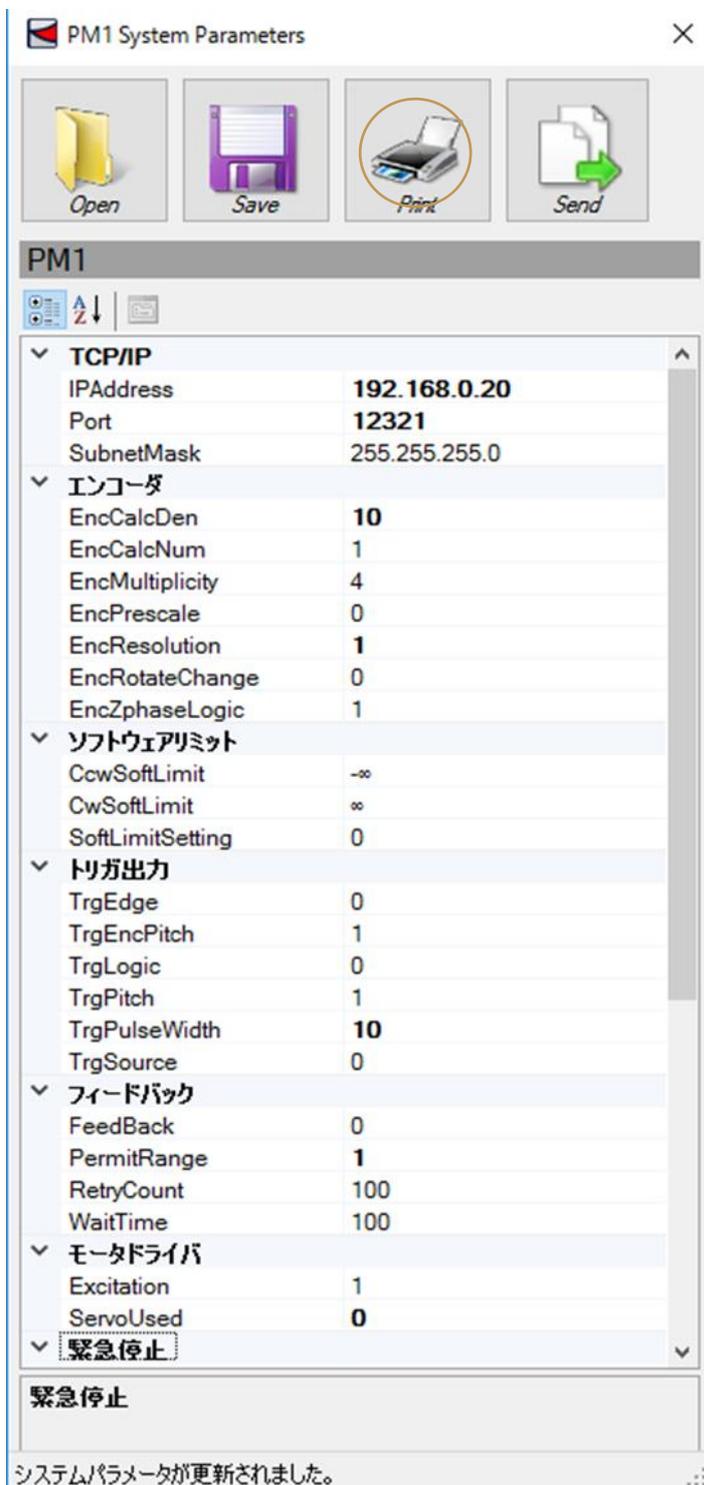
ファイル形式は、bin ファイル、XML ファイルです。これらのファイルは SC-210/410、CRUX、ARIES で共通に利用できるように設計されています¹⁹。

接続されているコントローラで欠けている機能は表には記載されません。保存されるときにはデフォルト値が保存されます。

上位機種において、下位機種で保存されたファイルを読み込んだ場合には、ファイルに保存されていないパラメータは Chamonix のデフォルト値で起動されます。

¹⁹異なる機種で完全に互換性を保証するという意味ではありません。

13.4. パラメータの印刷



「PRINT」ボタンをクリックするとパラメータを次ページのように印刷できます。

Date: 2016/01/21 17:29:30

Controller Type: ARIES

PM1

TCP/IP	
IPAddress	192.168.0.20
Port	12321
SubnetMask	255.255.255.0
エンコーダ	
EncCalcDen	10
EncCalcNum	1
EncMultiplicity	4
EncPrescale	0
EncResolution	1
EncRotateChange	0
EncZphaseLogic	1
ソフトウェアリミット	
CwSoftLimit	-∞
CwSoftLimit	∞
SoftLimitSetting	0
トリガ出力	
TrgEdge	0
TrgEncPitch	1
TrgLogic	0
TrgPitch	1
TrgPulseWidth	10
TrgSource	0
フィードバック	
FeedBack	0
PermitRange	1
RetryCount	100
WaitTime	100
モータドライバ	
Excitation	1
ServoUsed	0
緊急停止	
StopEmg	1
駆動制御	
BacklashPulse	0
BacklashType	0
LimitSwap	1
PmClock	2
PmLogic	0
PmPrescale	0
PmRotateChange	1
Resolution	1
TopSpeedLimit	50000
原点復帰	
EncSyncWrite	1
OrgOffset	0
OrgScanSpeed	500
OrgType	10
信号論理	
LimitLogic	1
NOrgSignalLogic	0
OrgSignalLogic	0
速度テーブル	
Speed_Hi	10,8000,50,15,3
Speed_Lo	5,4000,25,10,3
SpeedNo0	500,1000,16,16,2
SpeedNo1	500,2000,20,20,2
SpeedNo2	500,3000,24,24,2
SpeedNo3	500,4000,28,28,2
SpeedNo4	500,5000,32,32,2
SpeedNo5	500,6000,36,36,2
SpeedNo6	500,7000,40,40,2
SpeedNo7	500,8000,44,44,2
SpeedNo8	500,9000,48,48,2
SpeedNo9	500,10000,52,52,2

13.5. システムパラメータのプロパティ

システムパラメータのプロパティを簡単に説明します。

TCP/IP (ARIESのみ)

- **IPAddress** コントローラのIIPアドレス
- **Port** TCP/IPのポート
- **SubnetMask** コントローラのサブネットマスク

エンコーダ (CRUXおよびCRUX-Aを除く)

- **EncCalcDen** モーターパルス/エンコーダ分解能比率(分母)
- **EncCalcNum** モーターパルス/エンコーダ分解能比率(分子)
- **EncMultiplcty** エンコーダ進倍設定
- **EncPrescale** エンコーダ値のプリスケール
- **EncResolution** エンコーダ1パルス当たりの分解能
- **EncRotateChanige** エンコーダ加算方向の変更
- **EncZphaseLogic** エンコーダZ相の論理切り替え

ソフトウェアリミット

- **CwSoftLimit** CCW方向のソフトウェアリミット (パルスで入力してください)
- **CwSoftLimit** CW方向のソフトウェアリミット (パルスで入力してください)
- **CwSoftLimitSetting** ソフトウェアリミットの使用の選択

トリガ出力 (ARIESのみ)

- **TrgEdge** トリガ信号のエッジ選択
- **TrgEncPitch** エンコーダパルス同期の場合のトリガ信号分周比
- **TrgLogic** トリガ出力の論理切り替え
- **TrgPitch** モーターパルス同期の場合のトリガ信号の分周比
- **TrgPulseWidth** トリガ出力のパルス幅
- **TrgSource** トリガ信号源の選択

フィードバック

- **FeedBack** フィードバック制御方式
- **PermitRange** エンコーダパルス許容範囲
- **RetryCount** フィードバック時のリトライ回数
- **WaitTime** フィードバックの待機時間

モータドライバ

- **CurrentNo** 電池設定番号 (CRUXのみ)
- **Excitation** モーター励磁
- **MicroStepNo** マイクロステップ分割数番号 (CRUXのみ)
- **ServoUsed** サーボモータ使用 (ARIESのみ)

緊急停止

- **StopEmg** 緊急停止方式

駆動制御

- **BacklashPulse** バックラッシュ補正のパルス数
- **BacklashType** バックラッシュ補正方法
- **LimitSwap** CWとCCWのリミット信号うい入れ替え
- **PmClock** パルス出力方式の切り替え (ARIESのみ)
- **PmLogic** パルス出力論理方式の切り替え (ARIESのみ)
- **PmPrescale** パルス値プリスケール
- **PmRotateChange** モーター回転方向の変更
- **Resolution** 1パルスあたりのモーター分解能
- **TopSpeedLimit** 最高速度制限値 (ARIESのみ)
- **UsingEMGModeForJog** ジョグボタンの停止方式

原点復帰

- **EncSyncWrite** 原点復帰によるエンコーダ値のゼロリセット (CRUXおよびCRUX-Aを除く)
- **OrgOffset** 原点オフセット値
- **OrgScanSpeed** 原点サーチ時の速度 (SCシリーズを除く)
- **OrgType** 原点復帰方式

信号論理

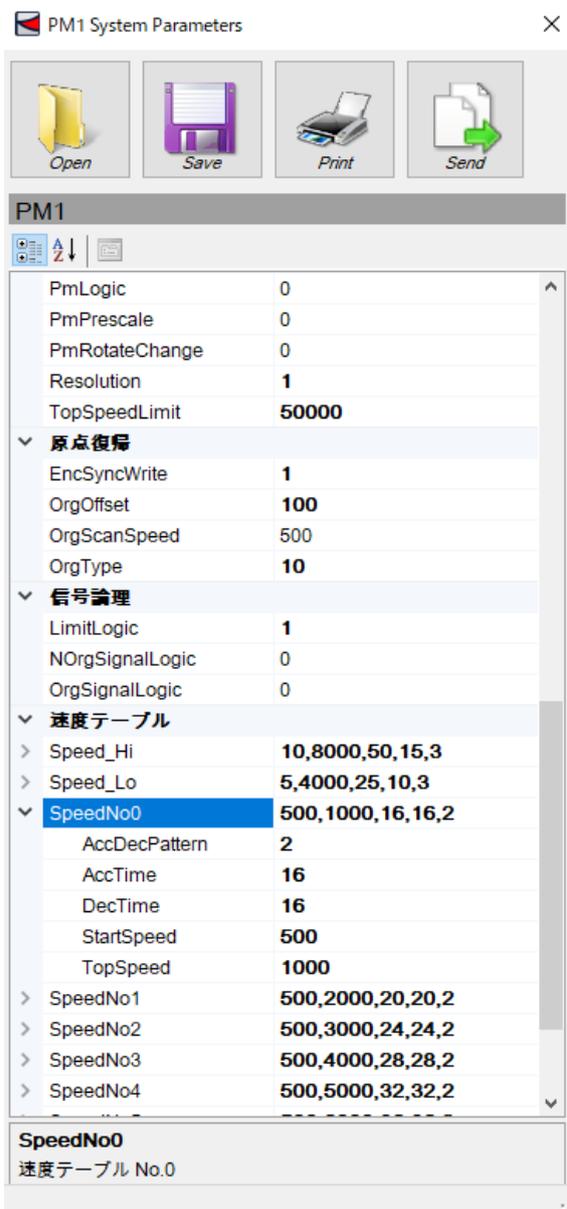
- **LimitLogic** CWとCCWのリミット信号を入れ替え
- **NOrgSignalLogic** NORGセンサ信号の論理切り替え
- **PrgSignalLogic** ORGセンサ信号の論理切り替え

速度テーブル

- **SpeedNo.0** 速度テーブルNo.0の設定
 - **AccDecPattern** スタート速度
 - **AccTime** 加速時間(×10msec)
 - **DecTime** 減速時間(×10msec、CRUXおよびCRUX-Aを除く)
 - **StartSpeed** スタート速度
 - **TopSpeed** 最高速度
- **SpeedNo.1** 速度テーブルNo.1の設定 (パラメータの定義はNo.0と同様です)
- **SpeedNo.2** 速度テーブルNo.2の設定 (パラメータはNo.0と同様です)
- **SpeedNo.3** 速度テーブルNo.3の設定 (パラメータの定義はNo.0と同様です)
- **SpeedNo.4** 速度テーブルNo.1の設定 (パラメータの定義はNo.0と同様です)
- **SpeedNo.5** 速度テーブルNo.2の設定 (パラメータの定義はNo.0と同様です)
- **SpeedNo.6** 速度テーブルNo.6の設定 (パラメータの定義はNo.0と同様です)
- **SpeedNo.7** 速度テーブルNo.7の設定 (パラメータの定義はNo.0と同様です)
- **SpeedNo.8** 速度テーブルNo.8の設定 (パラメータの定義はNo.0と同様です)
- **SpeedNo.9** 速度テーブルNo.9の設定 (パラメータの定義はNo.0と同様です)
- **Speed_Hi** PYXISによるジョグ操作の高速の速度テーブル
- **Speed_Lo** PYXISによるジョグ操作の低速の速度テーブル

13.6. 速度テーブルの変更

速度テーブルの変更の方法を説明します。スピードテーブルのテーブル名を展開すると、速度テーブルのパラメータが開きます。設定できるパラメータはコントローラの機種によって異なります。



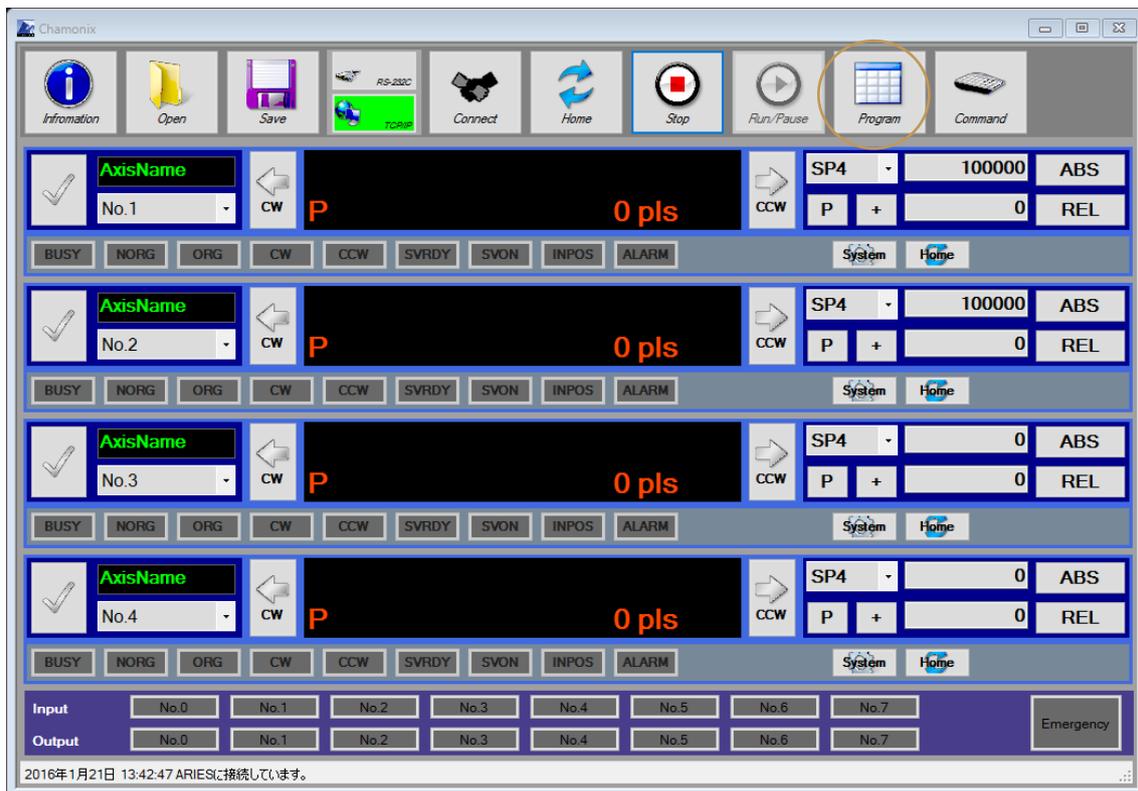
× ARIES/LYNXの場合、AccDecTimeは加減速パターン、AccTimeは加速時間(10msec単位)、DecTimeは減速時間(10msec単位)、StartSpeedはスタート速度(pls)、TopSpeedは最高速度(pps)を意味します。

CRUX/CRUX-Aの場合、AccDecTimeは加減速パターン、AccTimeは加速時間、(10msec単位)、StartSpeedはスタート速度(pls)、TopSpeedは最高速度(pps)を意味します。

なお、CRUX/CRUX-Aの場合、DecTimeはAccTimeと同じ値が設定されます。

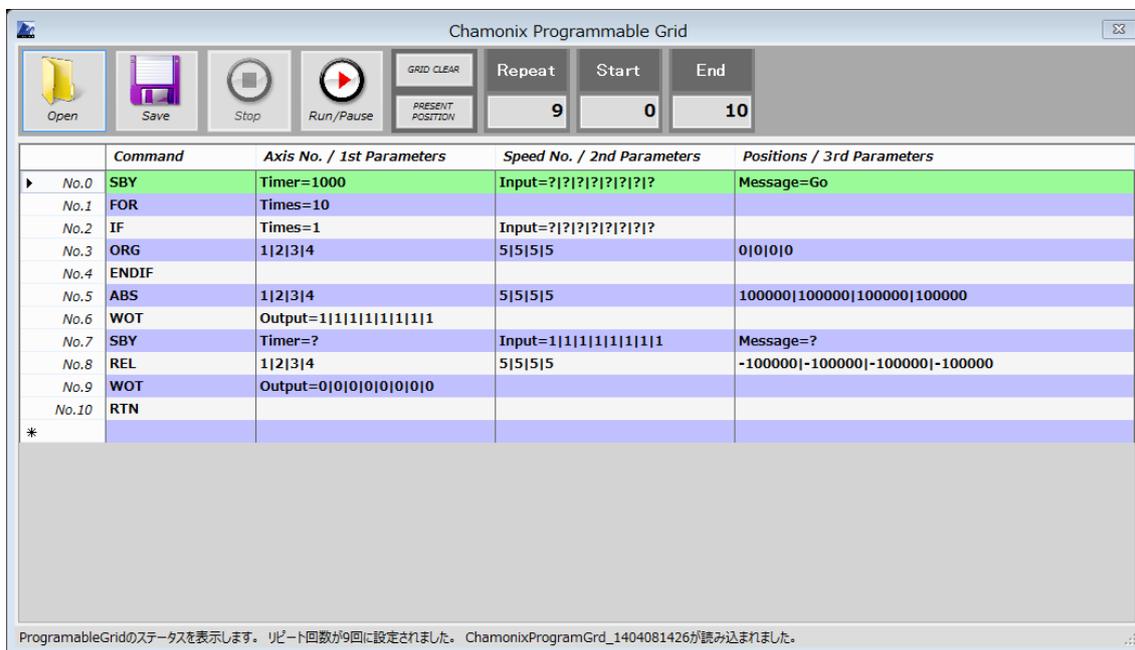
14. プログラマブルグリッドを利用する

14.1. プログラマブルグリッドの起動



「Program」ボタンをクリックすると、「Programmable Grid」が起動します。「Programmable Grid」は簡単なマクロのような機能まで付加したティーチング機能です。

14.2. プログラマブルグリッドの基本機能



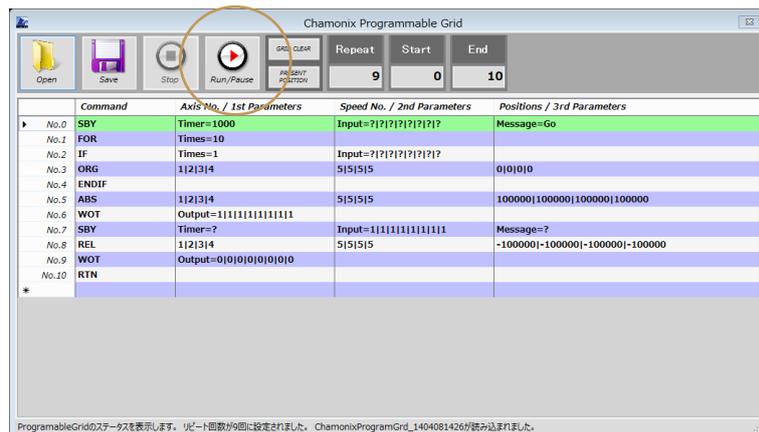
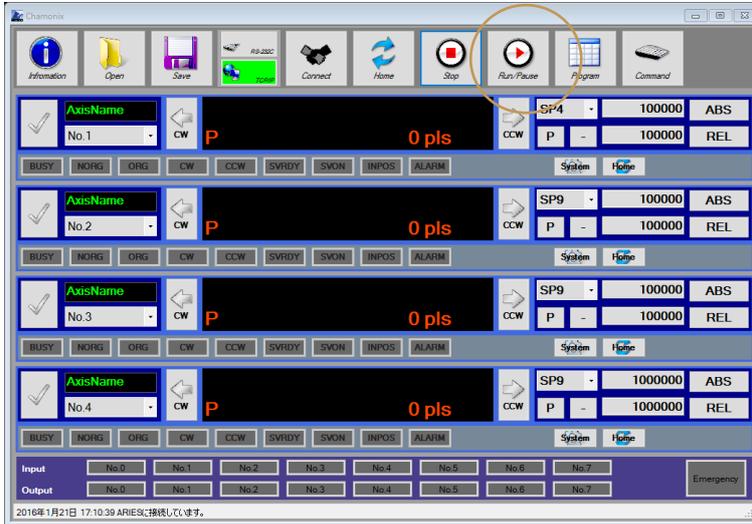
グリッド内に記述したプログラムに従ってコントローラに接続されたステージを自動駆動します。

グリッドの行に記述されたコマンドは原則として上から順番に実行されます。

Start で指定された行と End で指定された行を Repeat で指定された回数繰り返します。Repeat に「Endless」あるいは負の数字を入力すると無限に繰り返します。

上記の例では No.0 の行から No.10 の行まで 1 行ずつプログラムを実行し、それを 9 回繰り返します。

14.3. プログラムを実行

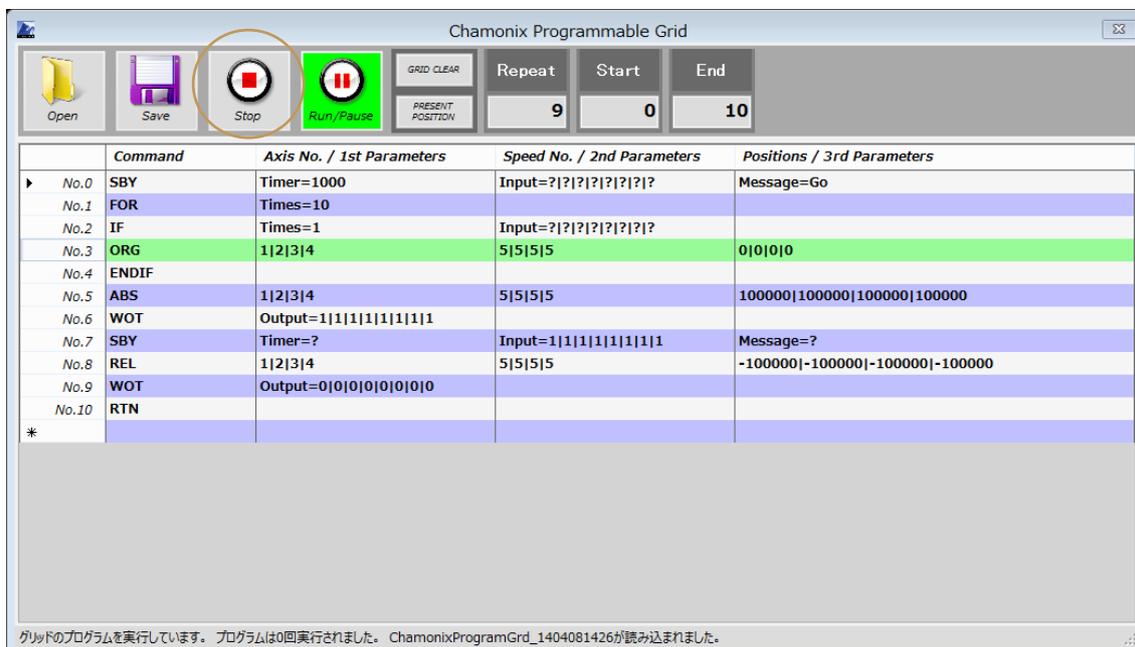


「Chamonix」あるいは「Chamonix ProgrammableGrid」の「RUN」ボタンをクリックすると実行します。

実行中は「RUN」ボタンは「PAUSE」ボタンになり、ランプが点灯します。実行中の行は緑色に点灯します。

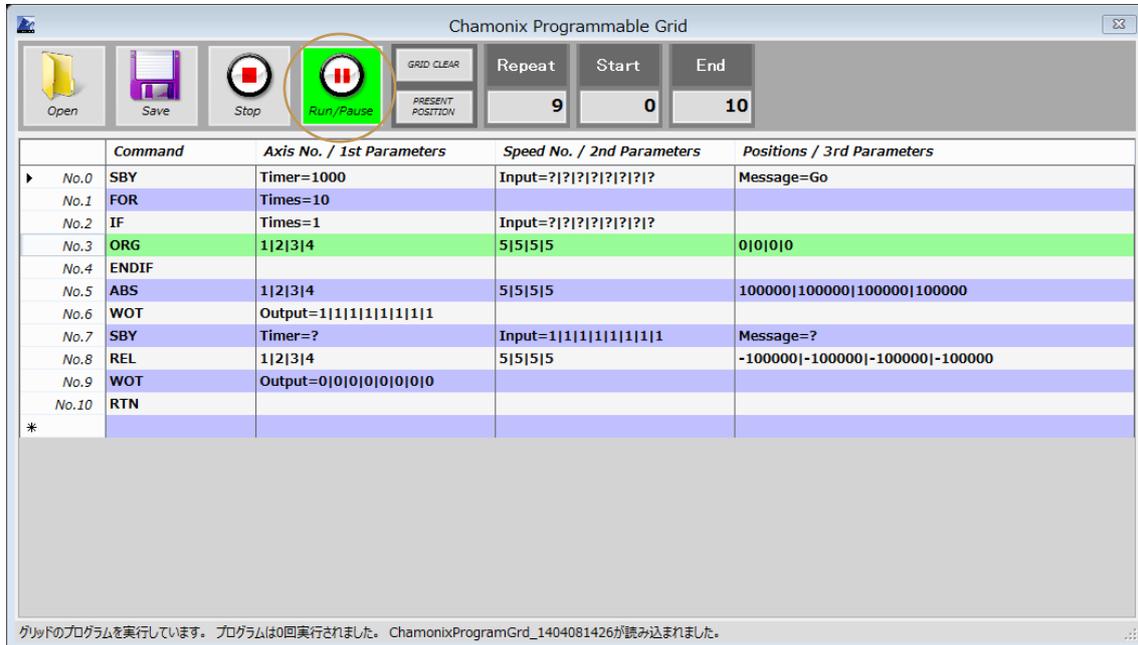
「PAUSE」ボタンをクリックすると実行中の行を実行したのちに一時停止します。一時停止からの再開は再び「RUN」ボタンを押して下さい

14.4. プログラムの停止

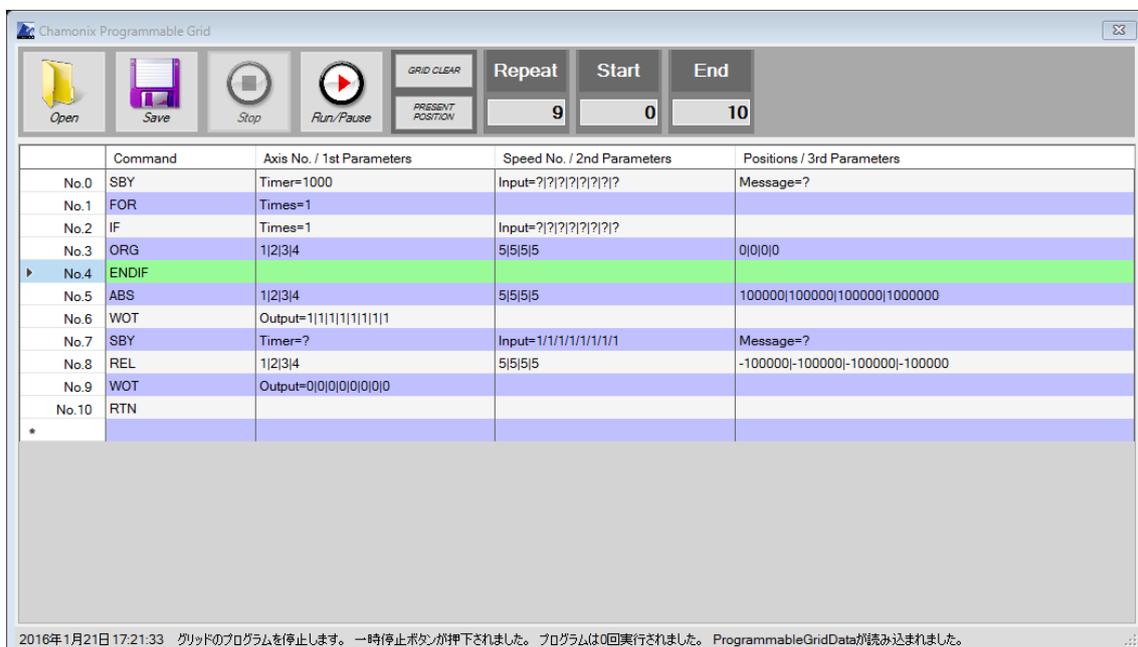


「STOP」ボタンをクリックすると停止します。「STOP」ボタンによる停止は緊急停止を意味し、直ちに停止します。また、停止位置からの再開はできません。再開する可能性がある場合は「Pause」ボタンをご利用ください。

14.5. プログラムの一時停止

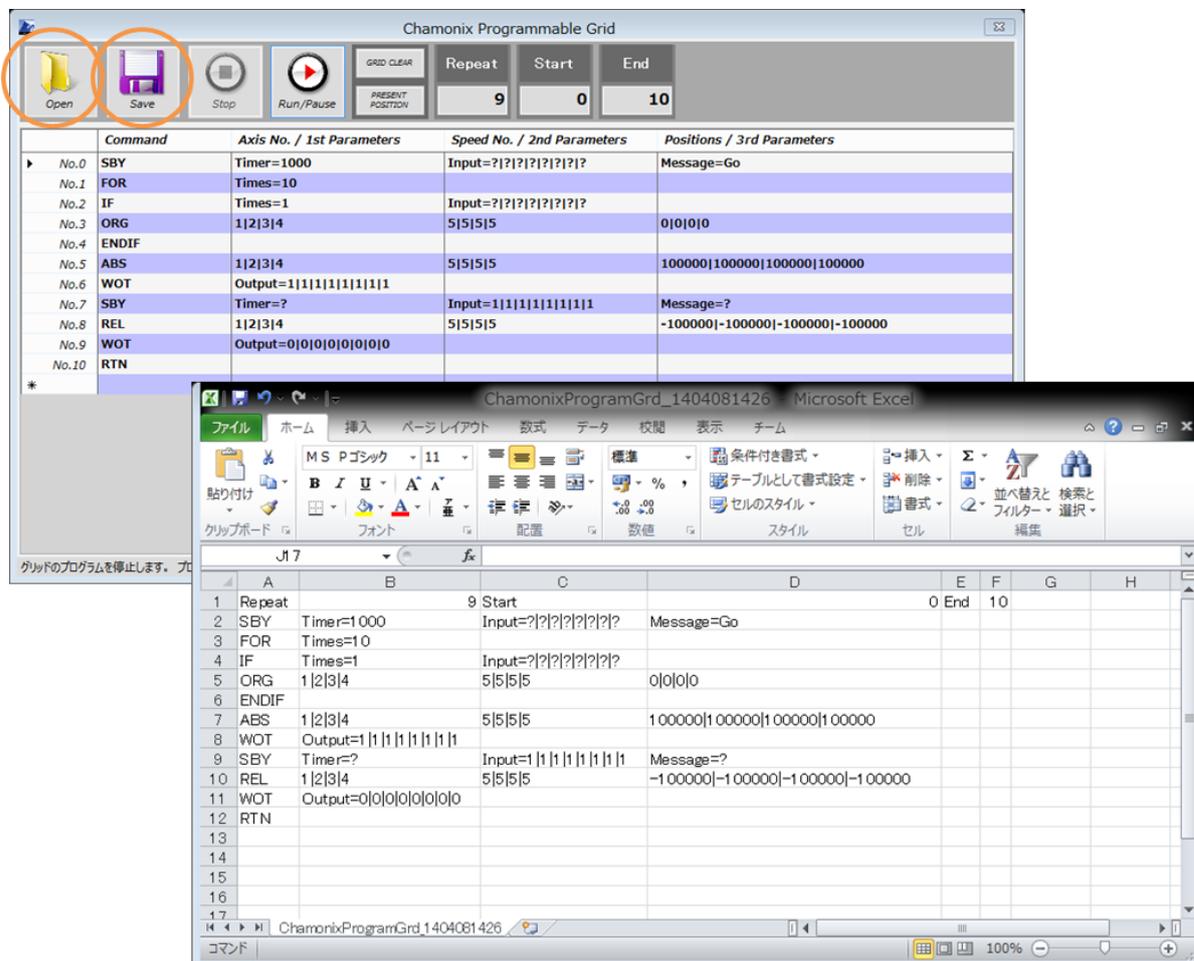


プログラム実行中は「Run」ボタンは点灯し、「Pause」ボタンになります。この状態で「Pause」ボタンを押すと、現在実行中の行を実行したのちにプログラムを一時停止し下図のようになります。



プログラムを再開するには再び「Run」ボタンを押します。

14.6. プログラムデータの入出力



プログラムデータはファイルに保存することができます。

データはファイルから読み込むことも可能です。

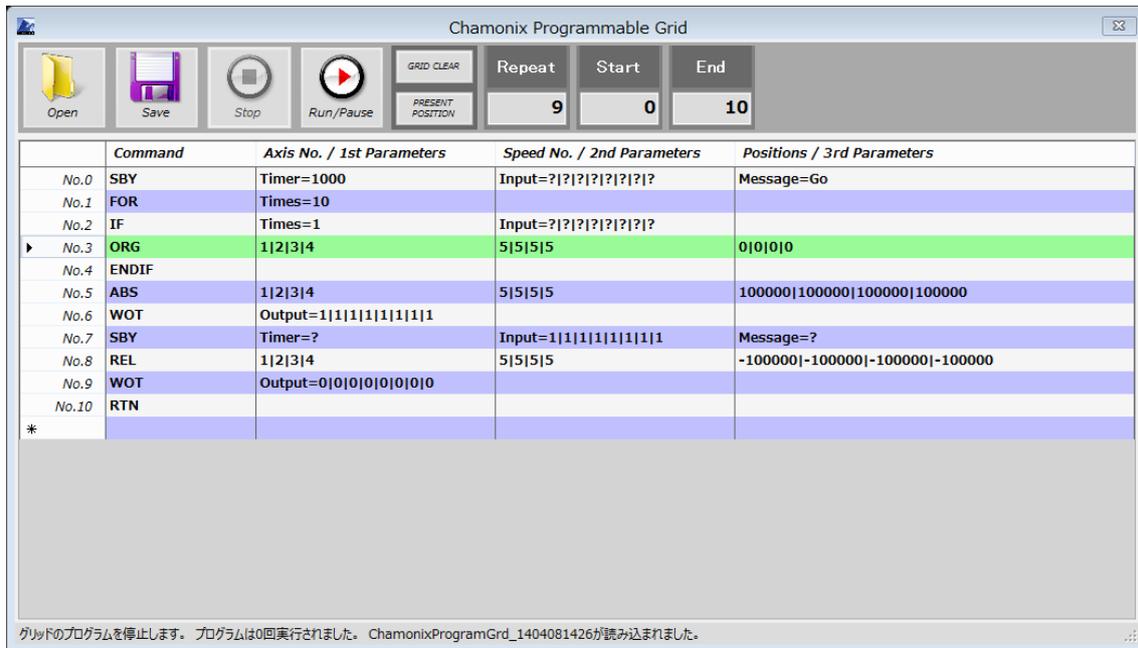
ファイルは DAT（標準）、XML、CSV（CSV1）の形式に対応しています。DAT の形式を推奨しています。

Parameters 内の各パラメータの区切り文字は「|」あるいは「/」を利用します。内部処理の際に「|」に読み替えられます。これは CSV を Microsoft Excel で処理する場合の不都合²⁰を考慮してのものです。

²⁰ Excel の場合、デフォルトの設定において、「1/2」は 1 月 2 日と自動的に日付関数に変換されてし

14.7. プログラマブルグリッドのコマンド

14.7.1. ORG: 原点復帰



ORG はシステムパラメータの設定に従って原点復帰動作を行います。

原点復帰動作完了後、

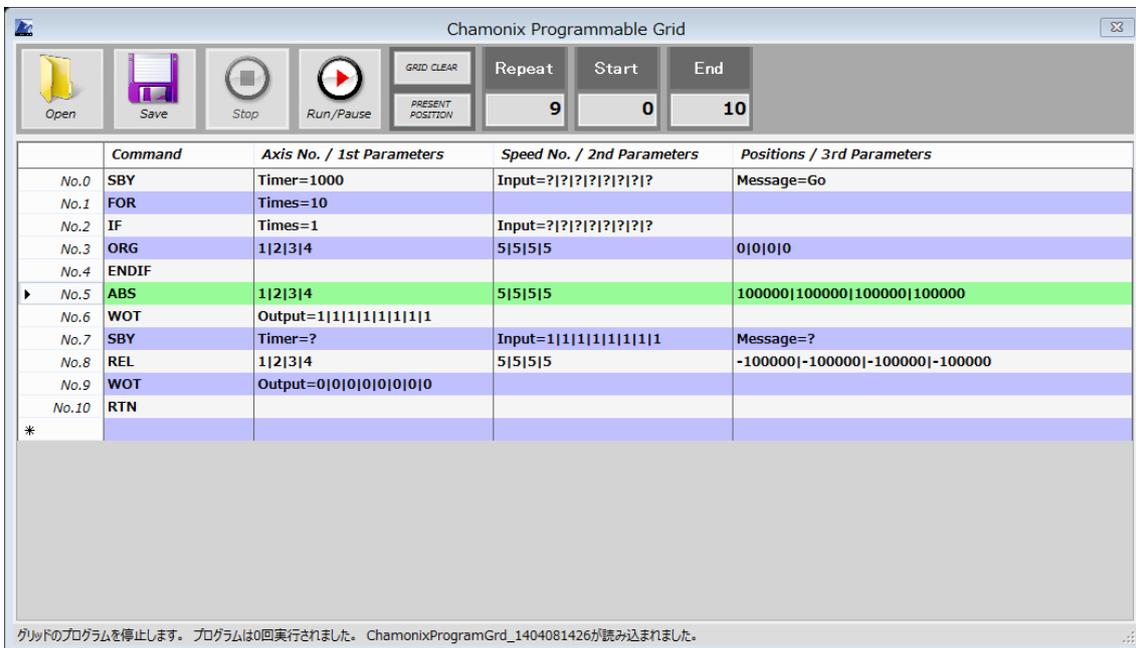
- 1st Parameter で指定された軸を、
- 2nd Parameters で指定した速度テーブルで、
- 3rd Parameters で指定された絶対位置へ移動します。

システムパラメータ設定で「Resolution」プロパティを設定している場合、換算値で指定して下さい²¹。

まうためです

²¹ デフォルトではパルスで入力するように設定されています。

14.7.2. ABS : 絶対位置管理による位置決め



ABSは絶対位置管理で

1st Parameters で指定した軸に

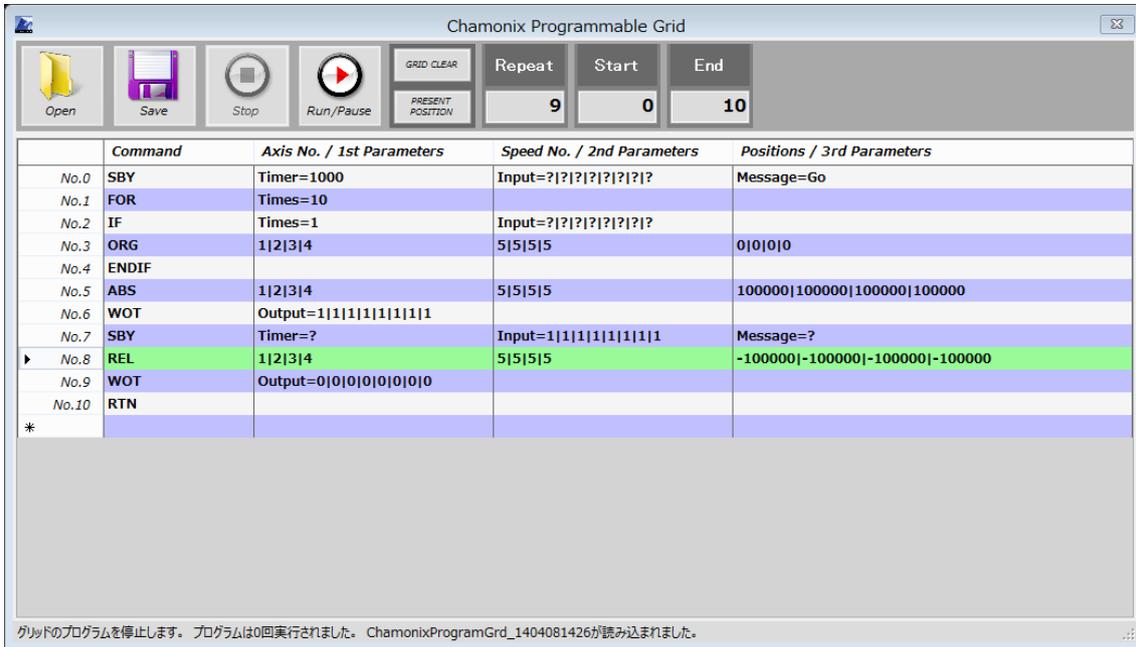
2nd Parameters で指定したスピードで、

3rd Parameters で指定した位置へ移動します。

システムパラメータ設定で「Resolution」プロパティを設定している場合、換算値で指定して下さい²²。

²² デフォルトではパルスで入力するように設定されています。

14.7.3. REL: 相対位置管理による位置決め



REL は相対位置管理で

1st Parameters で指定した軸に

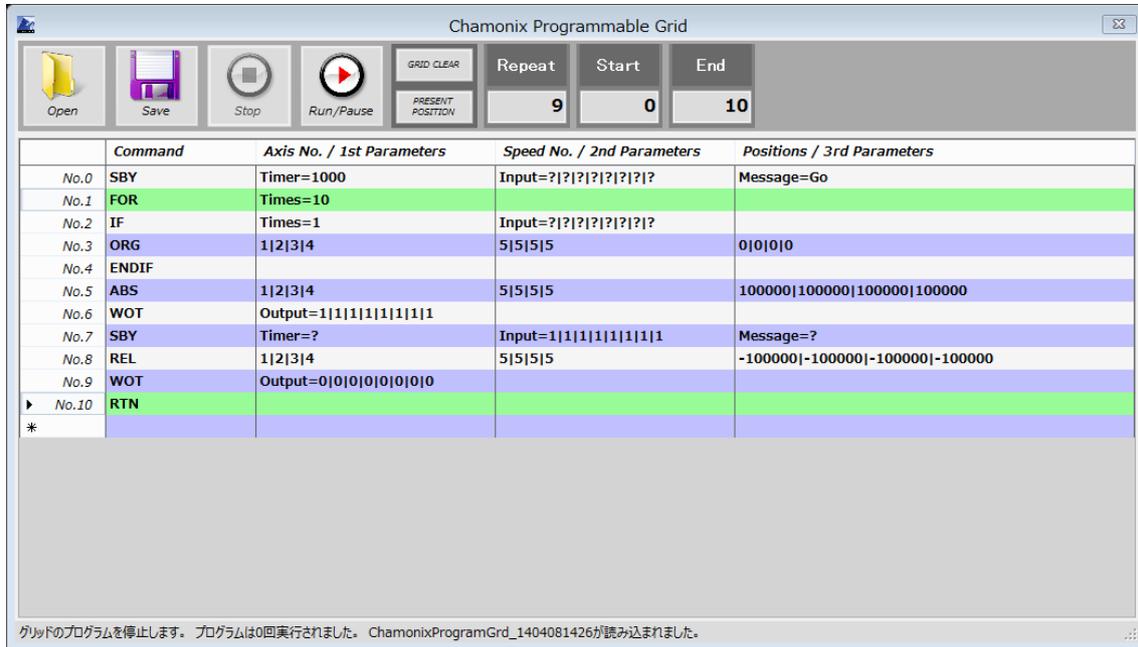
2nd Parameters で指定したスピードで、

3rd Parameters で指定した位置へ移動します。

システムパラメータ設定で「Resolution」プロパティを設定している場合、換算値で指定して下さい²³。

²³ デフォルトではパルスで入力するように設定されています。

14.7.5. FOR : 繰り返しループ



FOR コマンドは、これより下の行で最初に現れる RTN コマンドまでの間、

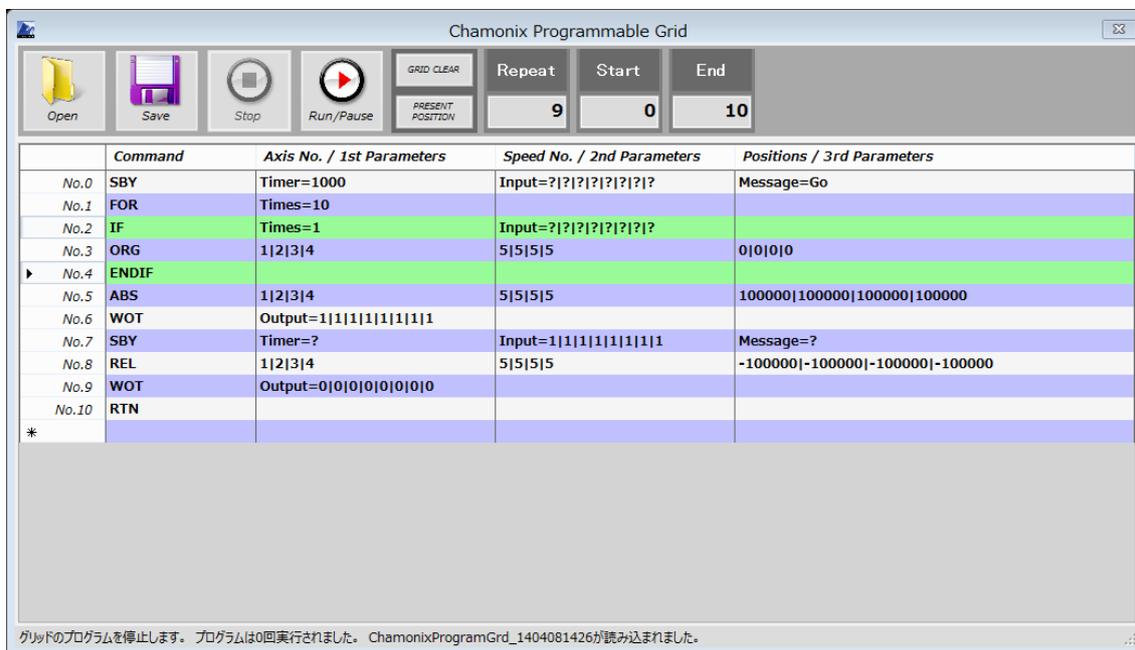
1st Parameters の「Times=」パラメータ指定子で指定された回数を繰り返します。

FOR コマンドは入れ子にすることができます。FOR コマンドが入れ子にされたとき、内側の FOR コマンドのループから実行されます。

IF コマンドの「Times=」パラメータ指定子も実行中のループの回数を対象とします。

FOR コマンドを指定した場合には、FOR コマンドひとつに対して、FOR コマンドより下側に必ず RTN コマンドを挿入する必要があります。

14.7.6. IF : 条件付きの実行



IF コマンドは 1st Parameters または 2nd Parameters に書かれた条件を満たした場合に、ENDIF コマンドまでの行を実行します。

1st Parameters には、Times= で指定したループの回数を指定できます。上の例ではループが 1 回目の場合のみ ORG コマンド（原点復帰）が実行されます。

2nd Parameters には、コントローラの I/O の入力ステータスを条件として指定できます。I/O を持たないコントローラでは無視されます。

「?」を入力するとその項目は無視されます。

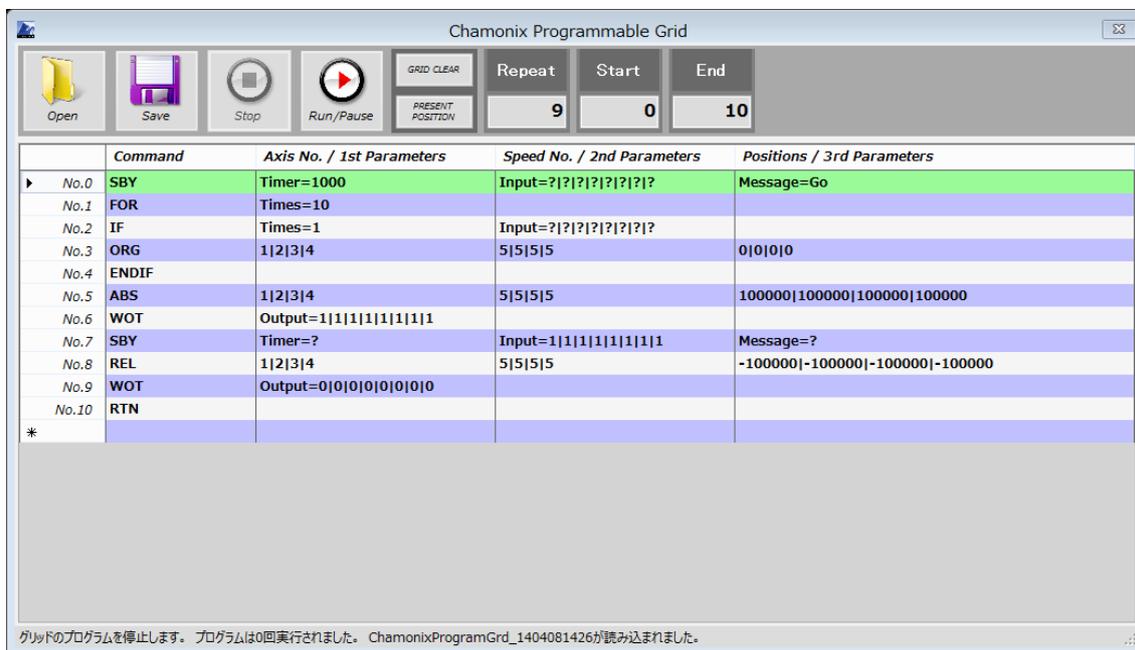
上の例では No.1 から No.10 まで 10 回繰り返すループで、最初の 1 回目のときだけ No.3 の行が実行されるように設定されています。

14.7.7. GOTO : 行のジャンプ

GOTO コマンドは 1st Parameters で指定された行にジャンプします。

GOTO コマンドは大変便利なコマンドですが、注意深く使用しないと、プログラムが複雑になり、間違ったプログラムを設定してしまう恐れがあります。

14.7.8. SBY : スタンバイ



SBY コマンドは 1st から 3rd までの Parameters のいずれかの条件を満たすまで待機します。

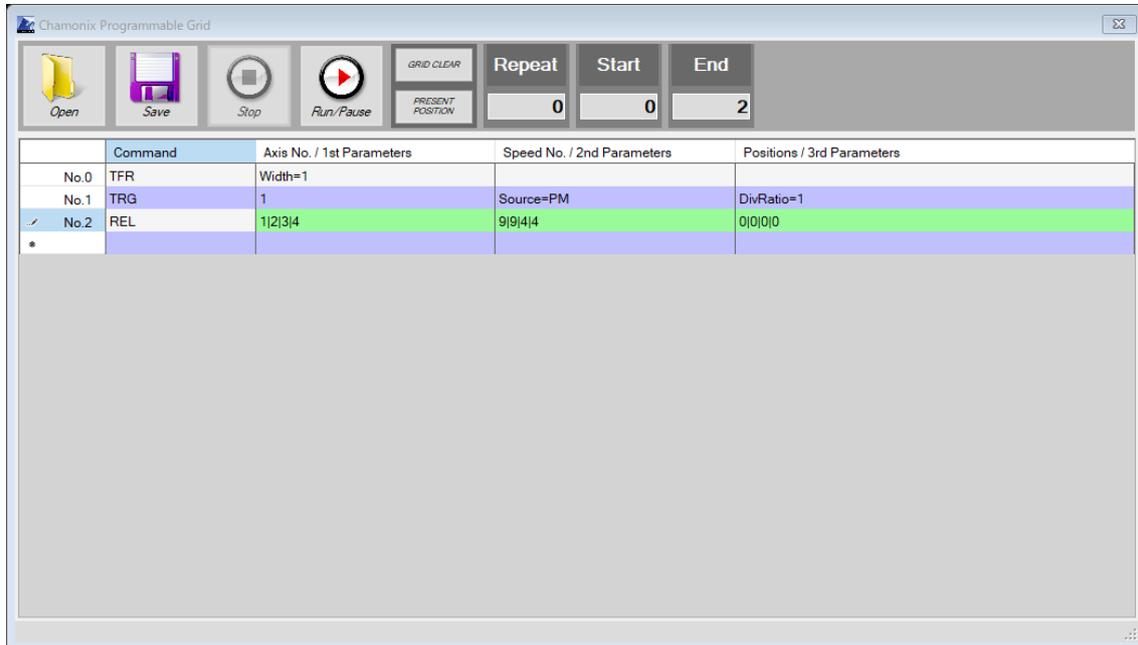
1st Parameters の「Timer=」のパラメータ指定子はミリ秒単位でタイマーを指定できます。

2nd Parameters の「Input=」の指定子は ARIES の I/O の入力を指定できます。

3rd Parameters の「Message=」のパラメータ指定子は指定された文字列が WindowsAPI の SendMessage を利用して Chamonix のメインウィンドウへ送信されたときに停止から抜けます。

「?」はそのパラメータを指定しないことを意味します。

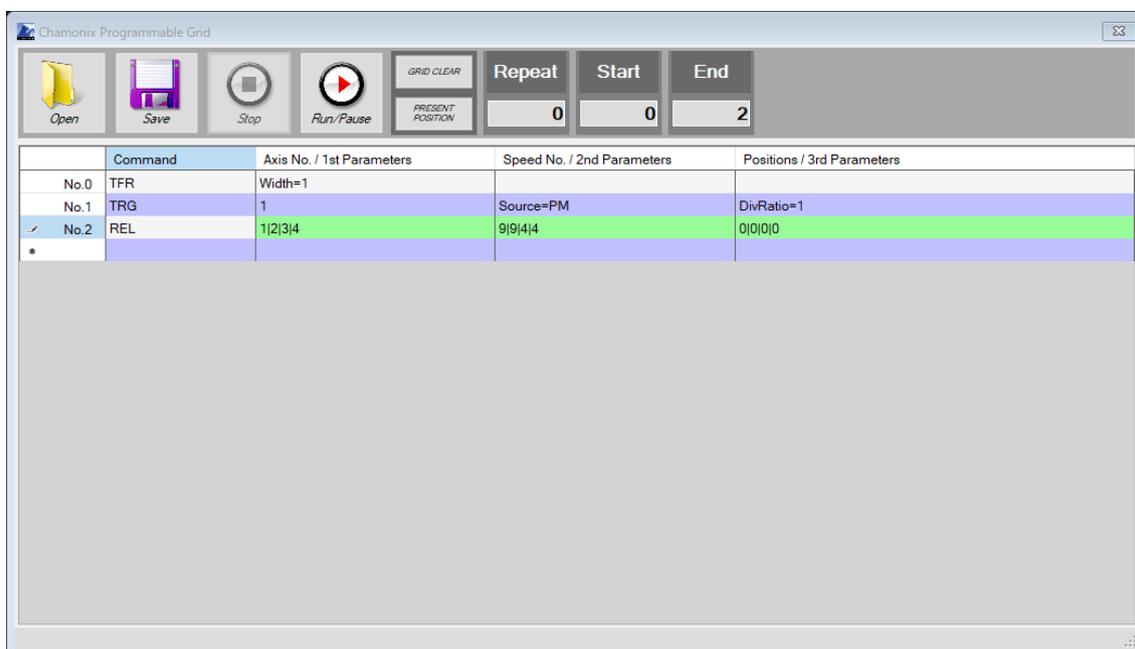
14.7.9. TFR：ワンショットトリガー



ARIES をご利用の場合、TFR コマンドが使用できます。TFR コマンドは、ワンショットのトリガー出力を実行します。

1st Parameters の「Width=」のパラメータ指定子はミリ秒単位でトリガーのパルス幅を指定します。

14.7.10. TRG : 駆動に同期したトリガー出力



ARIES をご利用の場合、TRG コマンドが使用できます。TRG コマンドは、ARIES の 1 軸目あるいは 2 軸目の駆動に同期したトリガー出力を設定します。TRG コマンドでトリガー出力の設定を行った後、次にコントローラから駆動をさせた際に、モーターパルスあるいはエンコーダパルスに同期したトリガーが出力されます。この設定は設定後の 1 回目の駆動にのみ有効です。

1st Parameters はトリガー出力のもととなる軸を指定します。1 か 2 です。

2nd Parameters の「Source=」の指定子は信号源を指定します。Source=PM : モーターパルス、Source=ENC : エンコーダパルスを信号源とします。

3rd Parameters の「DIVRatio=」のパラメータ指定子は分周比を指定します。例えば、DivRatio=1000 を指定した場合、1000 パルス毎にトリガーを出力します。

14.7.11. プログラマブルグリッドのコマンド一覧

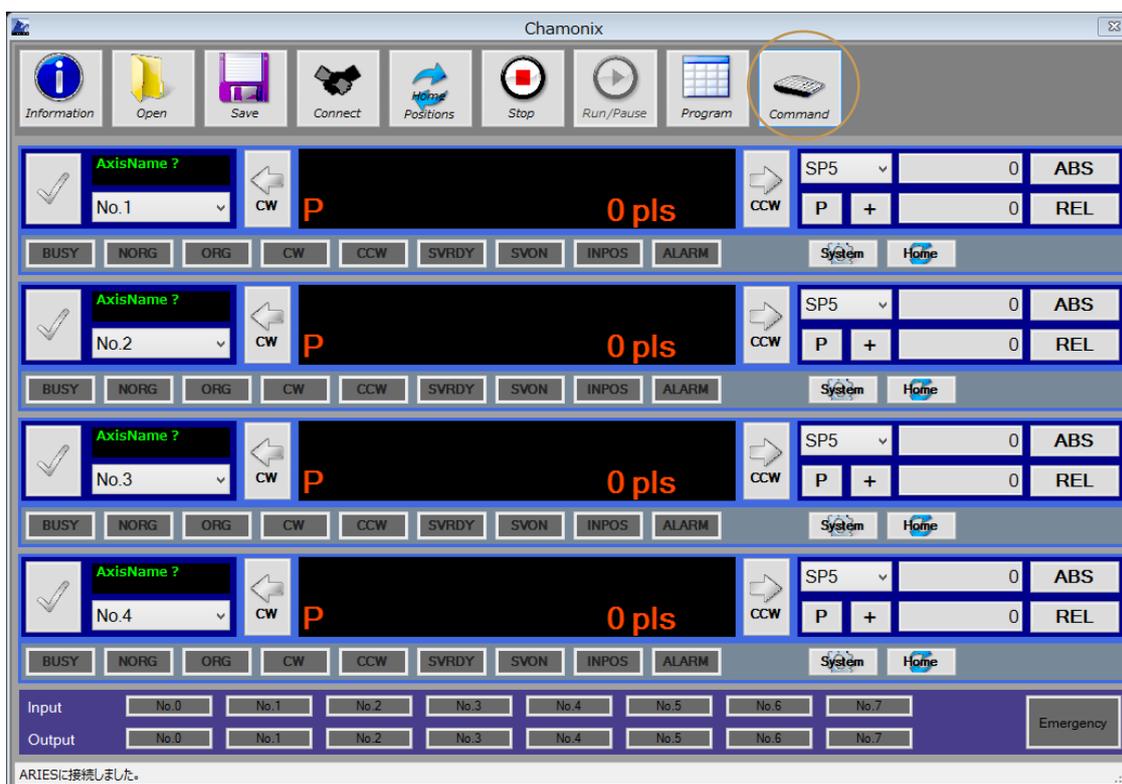
Programmable Grid のコマンド²⁴をまとめます。詳しくは各コマンドの説明を参照してください

ABS	• 絶対位置管理による位置決めを伴う駆動を行います
REL	• 相対位置管理による位置決めを伴う駆動を行います
ORG	• 原点復帰を行い、原点復帰後に絶対値管理で位置決めを行います
SBY	• 条件を満たすまで待機します
FOR	• RTNで指定された行まで、Times指定子で指定された回数を繰り返します
RTN	• FORループの折り返し行を指定します
IF	• ENDIFで指定された行までを条件を満たした場合に実行します
ENDIF	• IFコマンドで条件付き実行をする範囲の終端を意味します
WOT	• I/O出力を設定します
GOTO	• 指定した行にジャンプします
TFR (ARIESのみ)	• ワンショットトリガーを出力します。
TRG	• ARIESの1軸目あるいは2軸目に同期したトリガーを出力します。

²⁴ 特注対応のコマンドを除きます。

15. 直接リモートコマンドを入力する

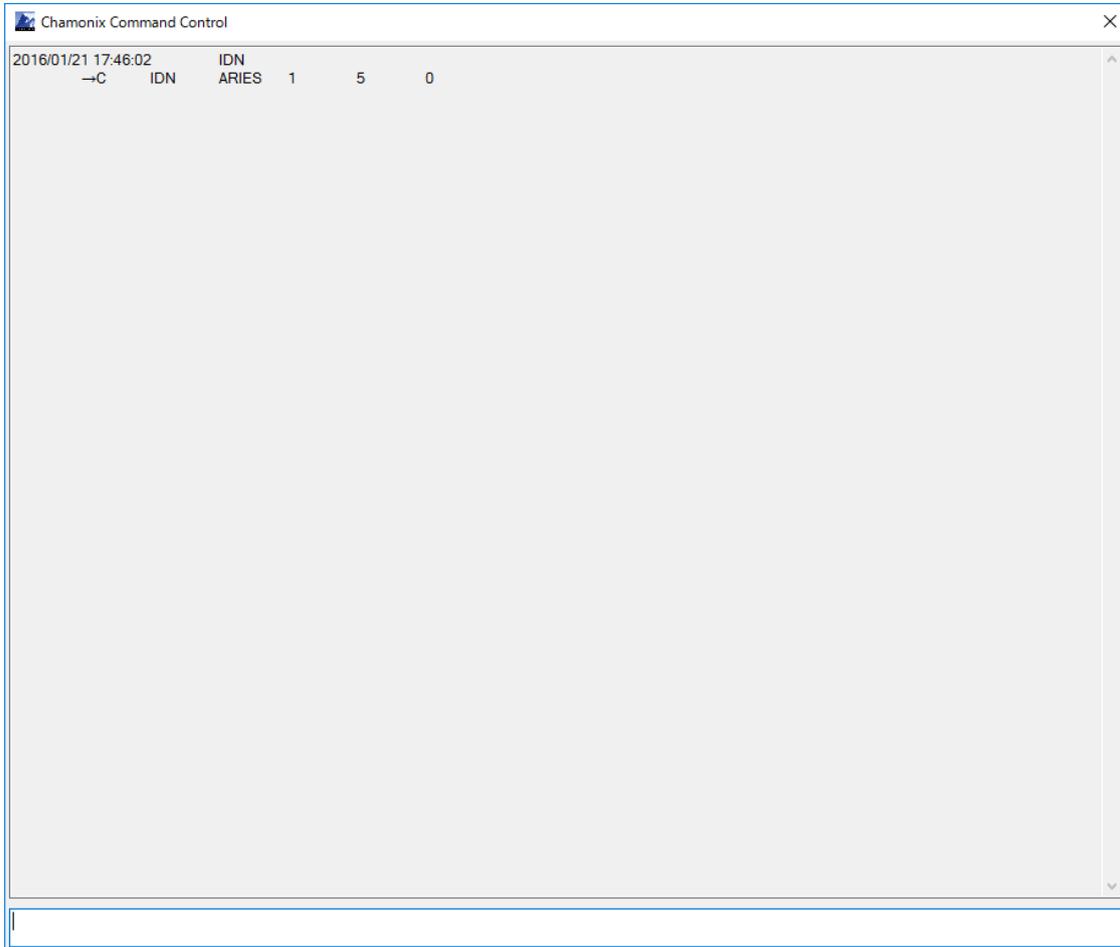
15.1. コマンドコントロールの起動



このボタンをクリックすると Chamonix Command Control が起動します。

Chamonix Command Control はコントローラへ送信するコマンドをキーボードから入力できます。

15.2. コマンドコントロールからコマンドを送信



送信ボックスにコマンドを入力し、「Enter」を押すとコントローラへコマンドが送信されます。RS-232C による通信に必要なヘッダーの STX 文字²⁵とデリミタの CRLF 文字²⁶は自動的に付加されます。リモートコントロールの書式については各々のコントローラの取扱説明書をご覧ください。入力は小文字で入力された場合も大文字に変換されます。

このウィンドウを使用してコマンドを送信した時間とコントローラからの返答がログウィンドウに表示されます。

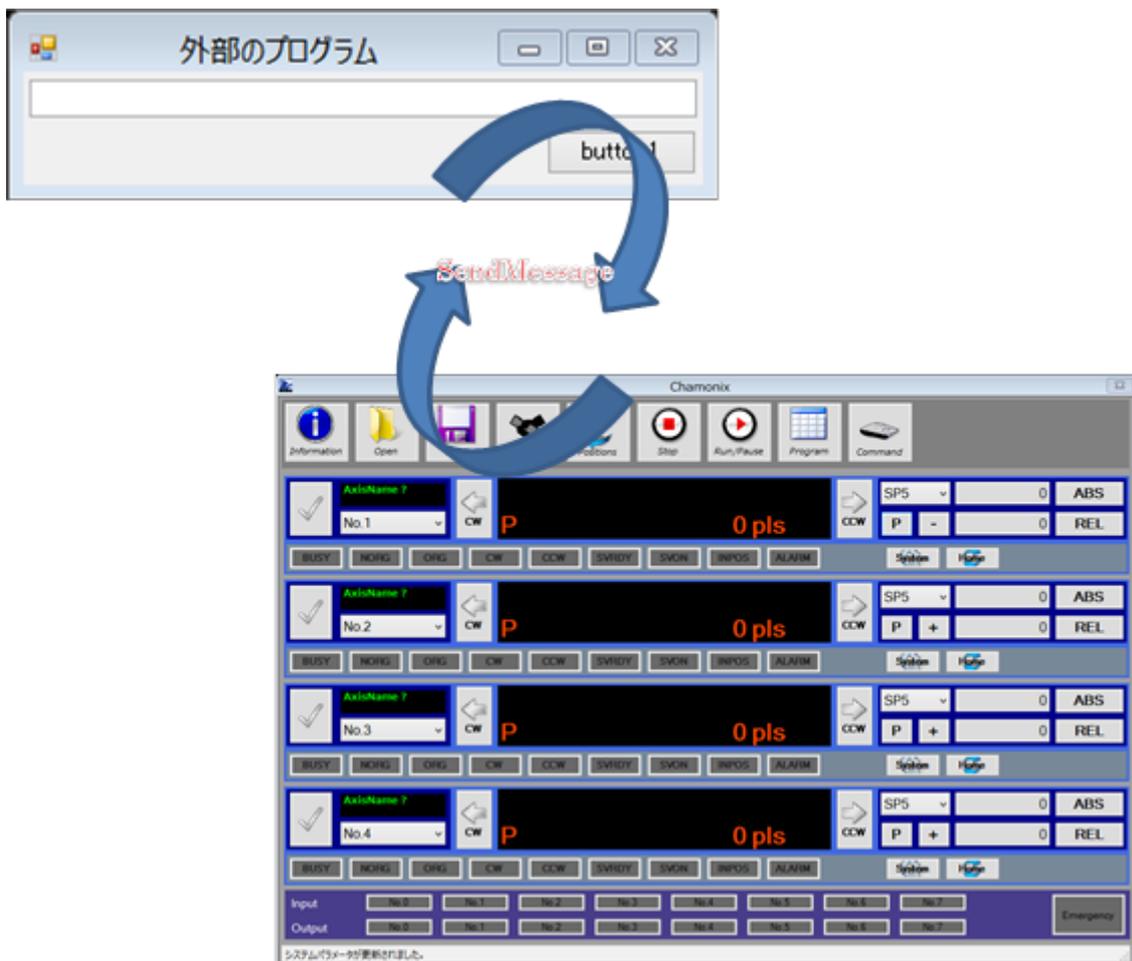
²⁵ 「STX」は「S」、「T」、「X」という 3 文字ではなく、「テキスト開始」を意味する 1 文字の制御文字です。

²⁶ 「CR」はローマ字二文字ではなく復帰を表す制御文字です。「LF」は同様に改行を表す制御文字です。

アプリ側では一切のエラーチェックを掛けていません。正しいコマンドを入力してください。誤った送信文字列がコントローラに送られた場合、コントローラの仕様に従ってエラーが返されます。詳しくはコントローラの取説をご覧ください。

16. API を利用して外部から制御しましょう

16.1. WindowsAPI による外部制御



Chamonix は Windows 上の他のアプリから Windows API の WM_COPYDATA を利用して駆動の制御を行うことが可能です。

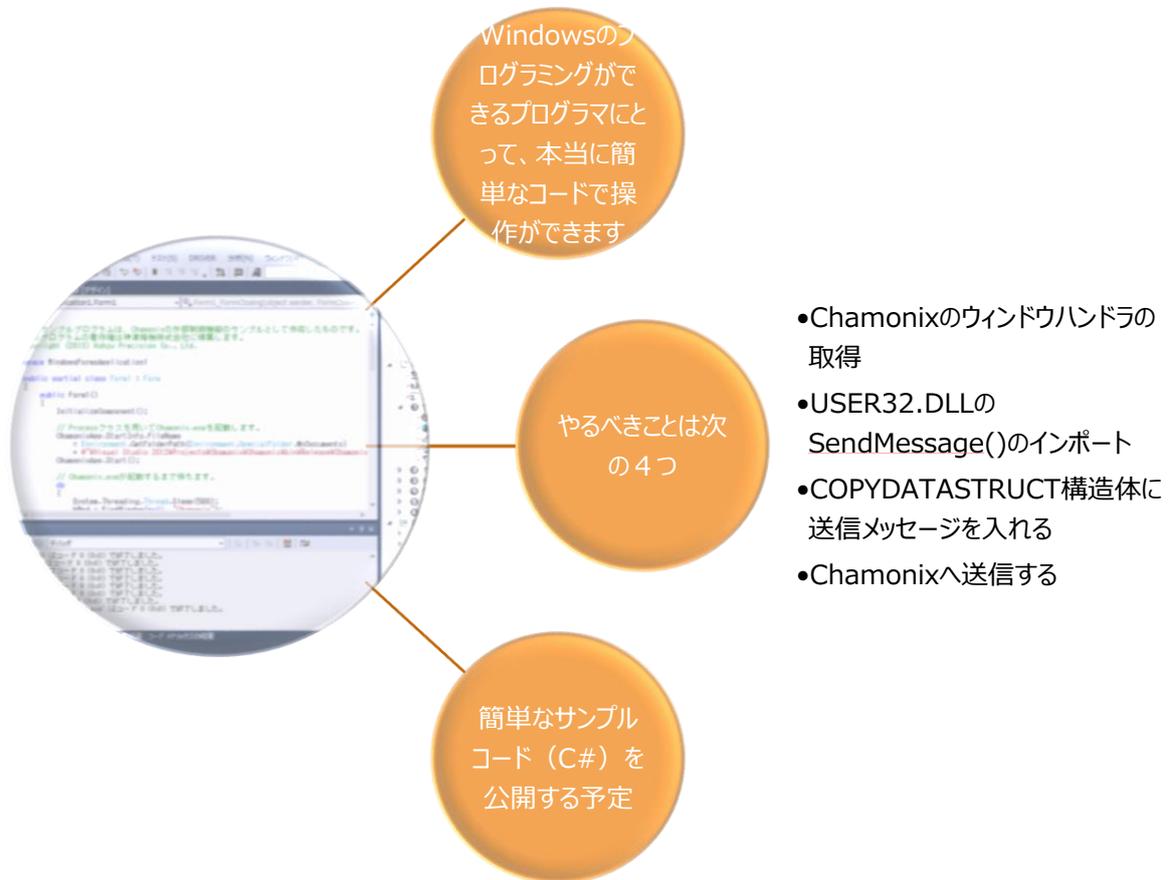
Windows API については Microsoft のウェブサイトなどを参照してください。

APS、RPS、ORG、STP のリモートコマンドに対応しています。

書式は ARIES のリモートコマンドの書式に準じます。

ただし返答方式のパラメータは省略されます。

16.2. 外部のアプリケーションからの制御



外部のアプリではSendMessage()のコードを1行追加するだけで簡単にChamonixを制御できます。

16.3. 外部のアプリから APS コマンドを送る

```
IntPtr hWnd;  
  
private int WM_COPYDATA = 0x004A;  
  
[USER32.dllのインポート]  
  
private void button1_Click(object sender, EventArgs e)  
{  
    // ChamonixのWindowハンドラを取得します。  
    hWnd = FindWindow(null, "Chamonix");  
  
    COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();  
  
    // 構造体stに値を入れます。  
    copyStruct.dwData = (IntPtr)0;  
    copyStruct.cbData = (uint)textBox1.Text.Length + 1;  
    copyStruct.lpData = this.textBox1.Text;  
  
    // SendMessageでChamonixにコマンドを送信します。  
    SendMessage((IntPtr)hWnd, WM_COPYDATA, this.Handle, ref copyStruct);  
  
    //Marshal.FreeCoTaskMem(lp);  
    textBox1.Text = null;  
}  
///  
/// <summary>  
/// SendMessageで送信する構造体です。  
/// </summary>  
private struct COPYDATASTRUCT  
{  
    public IntPtr dwData;  
    public uint cbData;  
    public string lpData;  
}
```

外部のアプリケーションから Windows API の SendMessage()関数²⁷を使って、直接 Chamonix の ウィンドウプロシージャにコマンドを渡します。

²⁷ 詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への SendMessage 関数についてのお問い合わせはご遠慮ください。

SendMessage()関数は次のような構文を持っています。

SendMessage(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)	
hWnd	Chamonix(メイン画面)のハンドルを入力
Msg	WM_COPYDATA (0x004A)
wParam	Chamonix からのメッセージを渡すウィンドウハンドル
lParam	送信するコマンド (COPYDATASTRUCT ²⁸) lParam.dwData : 0 lParam.cbData : 送信文字列のサイズ (例)13 lParam.lpData : 送信文字列 (例)APS1/9/10000

Chamonix に PM1 軸を、絶対位置管理で 10000pls の位置へ、速度テーブル No.9 で位置決めをさせたいとすると、

cbData に 12、

lParam に「APS1/9/10000」を入力します。

返答方式のパラメータは省略されます。

²⁸詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への COPYSTRUCT 構造体についてのお問い合わせはご遠慮ください。

16.4. 外部のアプリから RPS コマンドを送る

```
IntPtr hWnd;  
  
private int WM_COPYDATA = 0x004A;  
  
[USER32.dllのインポート]  
  
private void button1_Click(object sender, EventArgs e)  
{  
    // ChamonixのWindowハンドラを取得します。  
    hWnd = FindWindow(null, "Chamonix");  
  
    COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();  
  
    // 構造体stに値を入れます。  
    copyStruct.dwData = (IntPtr)0;  
    copyStruct.cbData = (uint)textBox1.Text.Length + 1;  
    copyStruct.lpData = this.textBox1.Text;  
  
    // SendMessageでChamonixにコマンドを送信します。  
    SendMessage((IntPtr)hWnd, WM_COPYDATA, this.Handle, ref copyStruct);  
  
    //Marshal.FreeCoTaskMem(lp);  
    textBox1.Text = null;  
}  
///  
/// <summary>  
/// SendMessageで送信する構造体です。  
/// </summary>  
private struct COPYDATASTRUCT  
{  
    public IntPtr dwData;  
    public uint cbData;  
    public string lpData;  
}
```

外部のアプリケーションから Windows API の SendMessage()関数²⁹を使って、直接 Chamonix の ウィンドウプロシージャにコマンドを渡します。

²⁹ 詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への SendMessage 関数についてのお問い合わせはご遠慮ください。

SendMessage()関数は次のような構文を持っています。

SendMessage(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)	
hWnd	Chamonix(メイン画面)のハンドルを入力
Msg	WM_COPYDATA (0x004A)
wParam	Chamonix からのメッセージを渡すウィンドウハンドル
lParam	送信するコマンド (COPYDATASTRUCT ³⁰) lParam.dwData : 0 lParam.cbData : 送信文字列のサイズ (例)13 lParam.lpData : 送信文字列 (例)RPS1/9/100000

Chamonix に PM1 軸を、相対位置管理で 10000pls の位置へ、速度テーブル No.9 で位置決めをさせたいとすると、

cbData に 12、

lpParam に「RPS1/9/10000」を入力します。

返答方式のパラメータは省略されます。

³⁰詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への COPYSTRUCT 構造体についてのお問い合わせはご遠慮ください。

16.5. 外部のアプリから STP コマンドを送る

```
IntPtr hWnd;  
  
private int WM_COPYDATA = 0x004A;  
  
[USER32.dllのインポート]  
  
private void button1_Click(object sender, EventArgs e)  
{  
    // ChamonixのWindowハンドラを取得します。  
    hWnd = FindWindow(null, "Chamonix");  
  
    COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();  
  
    // 構造体stに値を入れます。  
    copyStruct.dwData = (IntPtr)0;  
    copyStruct.cbData = (uint)textBox1.Text.Length + 1;  
    copyStruct.lpData = this.textBox1.Text;  
  
    // SendMessageでChamonixにコマンドを送信します。  
    SendMessage((IntPtr)hWnd, WM_COPYDATA, this.Handle, ref copyStruct);  
  
    //Marshal.FreeCoTaskMem(lp);  
    textBox1.Text = null;  
}  
///  
/// <summary>  
/// SendMessageで送信する構造体です。  
/// </summary>  
private struct COPYDATASTRUCT  
{  
    public IntPtr dwData;  
    public uint cbData;  
    public string lpData;  
}
```

外部のアプリケーションから Windows API の SendMessage()関数³¹を使って、直接 Chamonix の ウィンドウプロシージャにコマンドを渡します。

³¹ 詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への SendMessage 関数についてのお問い合わせはご遠慮ください。

SendMessage()関数は次のような構文を持っています。

SendMessage(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)	
hWnd	Chamonix(メイン画面)のハンドルを入力
Msg	WM_COPYDATA (0x004A)
wParam	Chamonix からのメッセージを渡すウィンドウハンドル
lParam	送信するコマンド (COPYDATASTRUCT ³²) lParam.dwData : 0 lParam.cbData : 送信文字列のサイズ (例)7 lParam.lpData : 送信文字列 (例)STP1/1

Chamonix に PM1 軸を緊急停止させたいとすると、
cbData に 7、
lpParam に「STP1/1」を入力します。

³²詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への COPYSTRUCT 構造体についてのお問い合わせはご遠慮ください。

16.6. 外部のアプリから ORG コマンドを送る

```
IntPtr hWnd;  
  
private int WM_COPYDATA = 0x004A;  
  
[USER32.dllのインポート]  
  
private void button1_Click(object sender, EventArgs e)  
{  
    // ChamonixのWindowハンドラを取得します。  
    hWnd = FindWindow(null, "Chamonix");  
  
    COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();  
  
    // 構造体stに値を入れます。  
    copyStruct.dwData = (IntPtr)0;  
    copyStruct.cbData = (uint)textBox1.Text.Length + 1;  
    copyStruct.lpData = this.textBox1.Text;  
  
    // SendMessageでChamonixにコマンドを送信します。  
    SendMessage((IntPtr)hWnd, WM_COPYDATA, this.Handle, ref copyStruct);  
  
    //Marshal.FreeCoTaskMem(lp);  
    textBox1.Text = null;  
}  
///  
/// <summary>  
/// SendMessageで送信する構造体です。  
/// </summary>  
private struct COPYDATASTRUCT  
{  
    public IntPtr dwData;  
    public uint cbData;  
    public string lpData;  
}
```

外部のアプリケーションから Windows API の SendMessage()関数³³を使って、直接 Chamonix の ウィンドウプロシージャにコマンドを渡します。

³³ 詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への SendMessage 関数についてのお問い合わせはご遠慮ください。

SendMessage()関数は次のような構文を持っています。

SendMessage(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)	
hWnd	Chamonix(メイン画面)のハンドルを入力
Msg	WM_COPYDATA (0x004A)
wParam	Chamonix からのメッセージを渡すウィンドウハンドル
lParam	送信するコマンド (COPYDATASTRUCT ³⁴) lParam.dwData : 0 lParam.cbData : 送信文字列のサイズ (例)7 lParam.lpData : 送信文字列 (例)ORG1/9

Chamonix に PM1 軸を速度テーブル No.9 で原点復帰させたいとすると
cbData に 7、
lpParam に「ORG1/9」を入力します。

³⁴詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への COPYSTRUCT 構造体についてのお問い合わせはご遠慮ください。

16.7. 外部のアプリからプログラマブルグリッドのスタンバイを解除する

```
IntPtr hWnd;  
  
private int WM_COPYDATA = 0x004A;  
  
USER32.dllのインポート  
  
private void button1_Click(object sender, EventArgs e)  
{  
    // ChamonixのWindowハンドルを取得します。  
    hWnd = FindWindow(null, "Chamonix");  
  
    COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();  
  
    // 構造体stに値を入れます。  
    copyStruct.dwData = (IntPtr)0;  
    copyStruct.cbData = (uint)textBox1.Text.Length + 1;  
    copyStruct.lpData = this.textBox1.Text;  
  
    // SendMessageでChamonixにコマンドを送信します。  
    SendMessage((IntPtr)hWnd, WM_COPYDATA, this.Handle, ref copyStruct);  
  
    //Marshal.FreeCoTaskMem(lp);  
    textBox1.Text = null;  
}  
/// <summary>  
/// SendMessageで送信する構造体です。  
/// </summary>  
private struct COPYDATASTRUCT  
{  
    public IntPtr dwData;  
    public uint cbData;  
    public string lpData;  
}
```

外部のアプリケーションから Windows API の SendMessage()関数³⁵を使って、直接 Chamonix の ウィンドウプロシージャにコマンドを渡します。

³⁵ 詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への SendMessage 関数についてのお問い合わせはご遠慮ください。

SendMessage()関数は次のような構文を持っています。

SendMessage(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)	
hWnd	Chamonix(メイン画面)のハンドルを入力
Msg	WM_COPYDATA (0x004A)
wParam	Chamonix からのメッセージを渡すウィンドウハンドル
lParam	送信するコマンド (COPYDATASTRUCT ³⁶) lParam.dwData : 0 lParam.cbData : 送信文字列のサイズ lParam.lpData : 送信文字列

Chamonix のプログラマブルグリッドを実行中に、SBY コマンドでスタンバイ状態になっているとき Message=指定子の文字列と lParam の文字列が一致したコマンドを受信すると、スタンバイ状態を解除します。

³⁶詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への COPYSTRUCT 構造体についてのお問い合わせはご遠慮ください。

16.8. 外部制御のサンプルプログラム

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();

        // Processクラスを用いてChamonix.exeを起動します。
        ChamonixApp.StartInfo.FileName
            = Environment.GetFolderPath(Environment.SpecialFolder.CommonProgramFilesX86)
            + @"\K0209\Chamonix\Chamonix.exe";
        ChamonixApp.Start();

        // Chamonix.exeが起動するまで待ちます。
        do
        {
            System.Threading.Thread.Sleep(500);
            hWnd = FindWindow(null, "Chamonix");
        }
        while (hWnd.ToInt32() == 0);
    }

    Process ChamonixApp = new Process();
    IntPtr hWnd;
    private int WM_COPYDATA = 0x004A;
    #region USER32.dllのインポート
    [DllImport("USER32.dll")]
    private static extern IntPtr SendMessage(IntPtr hWnd, int Msg, IntPtr wParam, ref COPYDATASTRUCT lParam);

    [DllImport("USER32.dll")]
    static extern IntPtr FindWindow(string lpClassName, string lpWindowName);
    #endregion

    private void button1_Click(object sender, EventArgs e)
    {
        // ChamonixのWindowハンドルを取得します。
        hWnd = FindWindow(null, "Chamonix");

        COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();

        // 構造体stに値を入れます。
        copyStruct.dwData = (IntPtr)0;
        copyStruct.cbData = (uint)textBox1.Text.Length + 1;
        copyStruct.lpData = this.textBox1.Text;

        // SendMessageでChamonixにコマンドを送信します。
        SendMessage((IntPtr)hWnd, WM_COPYDATA, this.Handle, ref copyStruct);

        //Marshal.FreeGCHandle(lp);
        textBox1.Text = null;
    }
    /// <summary>
    /// SendMessageで送信する構造体です。
    /// </summary>
    private struct COPYDATASTRUCT
    {
        public IntPtr dwData;
        public uint cbData;
        public string lpData;
    }

    protected override void WndProc(ref Message m)
    {
        if (m.Msg.Equals(WM_COPYDATA))
        {
            try
            {
                COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();
                copyStruct = (COPYDATASTRUCT)m.GetParam(typeof(COPYDATASTRUCT));
                string text = copyStruct.lpData;
            }
            catch
            {
                throw;
            }
            textBox1.Text = copyStruct.lpData;
        }
        base.WndProc(ref m);
    }

    private void Form1_FormClosing(object sender, FormClosingEventArgs e)
    {
        ChamonixApp.CloseMainWindow();
    }
}
```

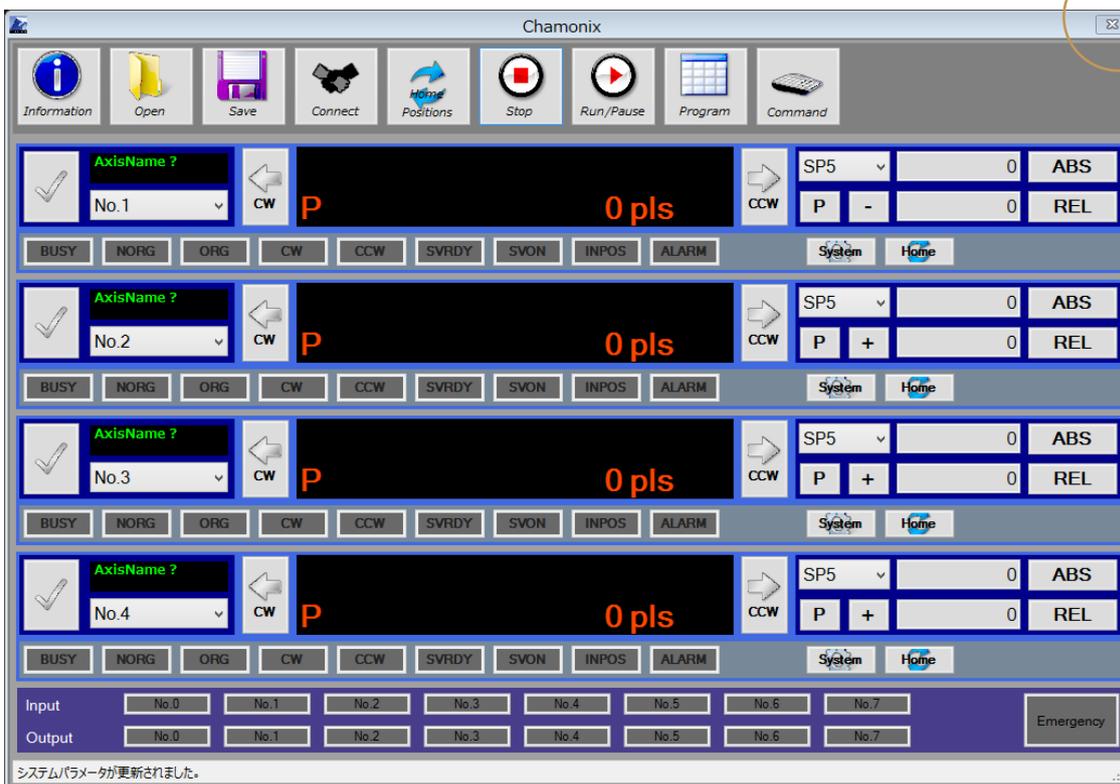
SendMessage を利用した外部のアプリケーションのサンプルコードは C# で Web から提供する予定です。サンプルコードはそれ自体を組み込んで実用に適するものとして設計しておりません。あくまでも SendMessage の利用法の一例として示しております。

サンプルコードをお客様が利用したことによるあらゆる責任を弊社としては負いかねます。

サンプルコードの著作権は神津精機株式会社に帰属します。

17. Chamonix を終了する

17.1. Chamonix を終了する



クローズボックスをクリックすると Chamonix は終了します。

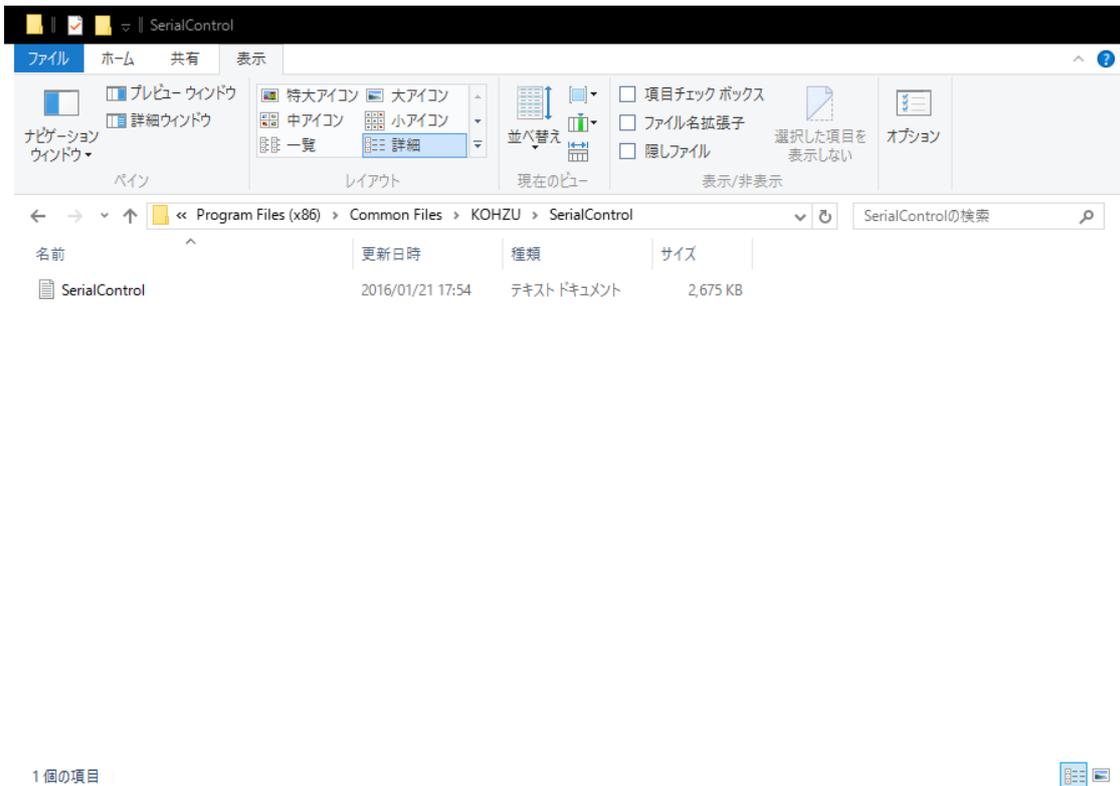
終了時に Program Data の Config.xml へ終了時のパラメータを書き込みます。COM ポート、ボーレート、各軸の換算係数、名称などが保存されます。コントローラのシステムパラメータは保存された値ではなくコントローラの設定値が優先されます。起動ファイルから起動されなかった場合、この Config.xml に保存された条件で起動します。

Chamonix が終了する際にコントローラ的全軸に対して STP コマンドを送信します。Chamonix で駆動をコントロールして、まだ駆動をしている最中に、Chamonix を落とした場合駆動はキャンセルされることを意味します。

CRUX が接続されている場合には、SAV コマンドが発行されます。CRUX の ROM に書き込みが終わるまで、すなわち Chamonix がクローズされるまで、CRUX の電源を切らないでください。

18. その他

18.1. 通信ログの保存場所



C:¥ProgramData¥KOHZU¥SerialControl³⁷フォルダ内にコントローラとの通信ログが残されています。

³⁷ Ver1.1.0.15 以前の場合 C:¥Program Files¥Common Files¥KOHZU¥SerialControl に格納されています。

18.2. 通信ログを確認

Timestamp	Direction	Device	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6	Value 7	Value 8
2014/06/09 19:14:03.081	←	STR1								
2014/06/09 19:14:03.089	→	C	STR1	0	1	0	0	0	0	0
2014/06/09 19:14:03.109	←	RIN								
2014/06/09 19:14:03.109	→	C	RIN	0	0	0	0	0	0	0
2014/06/09 19:14:03.145	←	RDP1								
2014/06/09 19:14:03.149	→	C	RDP1	0						
2014/06/09 19:14:03.165	←	RDE1								
2014/06/09 19:14:03.169	→	C	RDE1	0						
2014/06/09 19:14:03.241	←	STR2								
2014/06/09 19:14:03.249	→	C	STR2	0	1	0	0	0	0	0
2014/06/09 19:14:03.269	←	RIN								
2014/06/09 19:14:03.269	→	C	RIN	0	0	0	0	0	0	0
2014/06/09 19:14:03.289	←	RDP2								
2014/06/09 19:14:03.289	→	C	RDP2	0						
2014/06/09 19:14:03.369	←	STR3								
2014/06/09 19:14:03.369	→	C	STR3	0	1	0	0	0	0	0
2014/06/09 19:14:03.389	←	RIN								
2014/06/09 19:14:03.389	→	C	RIN	0	0	0	0	0	0	0
2014/06/09 19:14:03.409	←	RDP3								
2014/06/09 19:14:03.409	→	C	RDP3	0						
2014/06/09 19:14:03.481	←	STR4								
2014/06/09 19:14:03.489	→	C	STR4	0	1	0	0	0	0	0
2014/06/09 19:14:03.509	←	RIN								
2014/06/09 19:14:03.509	→	C	RIN	0	0	0	0	0	0	0
2014/06/09 19:14:03.529	←	RDP4								
2014/06/09 19:14:03.529	→	C	RDP4	0						
2014/06/09 19:14:03.605	←	STR1								
2014/06/09 19:14:03.609	→	C	STR1	0	1	0	0	0	0	0
2014/06/09 19:14:03.629	←	RIN								
2014/06/09 19:14:03.629	→	C	RIN	0	0	0	0	0	0	0
2014/06/09 19:14:03.649	←	RDP1								
2014/06/09 19:14:03.649	→	C	RDP1	0						
2014/06/09 19:14:03.665	←	RDE1								
2014/06/09 19:14:03.669	→	C	RDE1	0						
2014/06/09 19:14:03.741	←	STR2								
2014/06/09 19:14:03.749	→	C	STR2	0	1	0	0	0	0	0
2014/06/09 19:14:03.769	←	RIN								
2014/06/09 19:14:03.769	→	C	RIN	0	0	0	0	0	0	0
2014/06/09 19:14:03.789	←	RDP2								
2014/06/09 19:14:03.789	→	C	RDP2	0						
2014/06/09 19:14:03.861	←	STR3								
2014/06/09 19:14:03.869	→	C	STR3	0	1	0	0	0	0	0
2014/06/09 19:14:03.889	←	RIN								
2014/06/09 19:14:03.889	→	C	RIN	0	0	0	0	0	0	0
2014/06/09 19:14:03.909	←	RDP3								
2014/06/09 19:14:03.909	→	C	RDP3	0						
2014/06/09 19:14:03.981	←	STR4								
2014/06/09 19:14:03.989	→	C	STR4	0	1	0	0	0	0	0
2014/06/09 19:14:04.009	←	RIN								
2014/06/09 19:14:04.009	→	C	RIN	0	0	0	0	0	0	0
2014/06/09 19:14:04.029	←	RDP4								
2014/06/09 19:14:04.029	→	C	RDP4	0						
2014/06/09 19:14:04.101	←	STR1								

このログファイルにはコントローラとの最近の送受信が記録されています。

送受信の時間は、ミリ秒単位で記録されます。時間はシステムの時計を利用しているので正確ではありません。

問題が生じている場合、このファイルを解析することでより速いトラブルシューティングが期待されます。

改訂履歴

年月日	版数	その他
2014年6月30日	Rev.1.0.0	M.O.
2014年10月30日	Rev.1.0.1	M.O.
2014年12月25日	Rev.1.0.2	M.O.
2016年1月21日	Rev.1.1.0	M.O.
2017年12月25日	Rev.1.2.1	M.O.
2019年6月25日	Rev.1.2.2	M.O.
2020年2月4日	Rev.1.2.3	M.O.



本社

〒215-8521

神奈川県川崎市麻生区栗木 2 - 6 - 1 5

Tel : **0 4 4 - 9 8 1 - 2 1 3 1**

Fax : **0 4 4 - 9 8 1 - 2 1 8 1**

E- mail : sale@kohzu.co.jp

Web Site : <http://www.kohzu.co.jp/>

大阪支店

〒532-0004

大阪市淀川区西宮原 2 - 7 - 3 8

新大阪西浦ビル 2 0 2

Tel. : **0 6 - 6 3 9 8 - 6 6 1 0**

Fax : **0 6 - 6 3 9 8 - 6 6 2 0**