

ROKAE

xCore

制御システム操作マニュアル



xCore

制御システム操作マニュアル

制御システムバージョン： xCore V3.0

文書バージョン： A_J2

本マニュアルに記載されている内容は、予告なく変更される場合があります。本書の記載内容の誤りについて、当社は一切の責任を負いません。

本書および本書に記載された製品の使用に起因する偶発的または間接的な損害については、当社は一切の責任を負いませんので予めご了承ください。

すべての危険や結果を予測することはできないため、本書は考えられるすべての危険に対して警告するものではありません。

当社からの許可無く、本書の内容の一部または全部を複写・複製することは禁止されています。

本書に誤りを発見した場合、または本書に対する改善・追加意見がある場合は、弊社までご連絡ください。

本書の原語は中国語です。中国語以外の版は、すべて中国語版から翻訳されたものです。

© Copyright 2015-2024 ROKAE All rights reserved
ROKAE（山東）机器人集团有限公司
中国・山東省

目次

1 概要	11
1.1 本書について	11
1.2 本書の対象者	11
1.3 本書の読み方	11
1.4 本書のイラスト	11
1.5 お問い合わせ	11
1.6 本書の読み方	12
1.7 改訂履歴	12
1.8 関連マニュアル	13
2 安全	14
2.1 概要	14
2.2 安全上の責任について	14
2.3 表記	14
2.3.1 危険レベル	14
2.3.2 リスク説明	14
2.4 安全停止	15
2.5 安全装置	16
2.5.1 非常停止ボタン	16
2.5.2 イネーブル装置	17
2.6 様々な状況における安全上の注意事項	18
2.6.1 手動モードでの安全上の注意事項	18
2.6.2 自動モードでの安全上の注意事項	18
2.6.3 設置・操作時の安全要求事項	18
2.6.4 デバッグ時の安全要求事項	18
2.6.5 保守時の安全要求事項	19
2.6.6 生産ラインでの安全対策	19
2.6.7 火災時の安全	19
2.6.8 感電時の安全	20
3 用語	21
4 ロボットの基礎知識	22
4.1 概要	22
4.2 座標系	22
4.3 特異点	23
4.3.1 ロボットの典型的な特異点	23
4.3.2 特異点回避	25
4.4 滑らか軌跡エリア	28
4.5 先読み機能	28
4.6 力制御	28
4.6.1 力制御の概要	28

4.6.2 インピーダンス制御.....	29
4.6.3 力制御探索	30
4.6.4 力制御の応用.....	31
5 ロボットシステムの構成と接続	32
5.1 本章の概要	32
5.2 制御システムの構成.....	32
5.2.1 xPad2 教示操作盤の概要	32
5.3 産業用ロボットのシステム構成.....	33
5.3.1 XBC5 シリーズ制御キャビネットの概要	33
5.3.2 XBC5-M 制御キャビネットの配線、電源投入、起動	34
5.3.3 XBC5 制御キャビネットの配線、電源投入、起動	35
5.3.4 XBC5-E 制御キャビネットの配線、電源投入、起動.....	36
5.4 協働ロボットのシステム構成	37
5.4.1 ER 及び ER PRO	37
5.4.2 CR 及び SR.....	38
5.4.3 CR-C 及び SR-C	40
5.5 HMI とロボットの接続	42
5.5.1 xPad2 とロボットの接続.....	43
5.5.2 PC とロボットの接続	44
5.5.3 ロボットの検出と接続.....	50
6 HMI の概要.....	52
6.1 本章の概要	52
6.2 ROBOTASSIST ソフトウェアの概要	52
6.3 HMI 全体のレイアウト	53
6.3.1 上部ステータスバー	53
6.3.2 左サイドバー	54
6.3.3 右操作パネル.....	55
6.3.4 下部ステータスバー	57
6.4 状態監視.....	58
6.4.1 3D モデル	59
6.4.2 タスク	59
6.4.3 IO 信号	60
6.4.4 ネットワーク接続.....	61
6.4.5 レジスタ.....	62
6.4.6 コンペア.....	63
6.4.7 変数モニター	63
6.5 プログラミングメニューの一覧	64
6.6 設定メニューの一覧.....	65
6.7 通信メニューの一覧.....	65
6.8 安全機能メニューの一覧	66
6.9 パッケージメニューの一覧.....	66
6.10 ログメニューの一覧	66

6.11 オプションの一覧	66
7 制御システムの基本操作	67
7.1 本章の概要	67
7.2 動作モード	67
7.2.1 手動モード	67
7.2.2 自動モード	67
7.2.3 モードの確認と切り替え	68
7.3 モーター電源のオン/オフ	69
7.3.1 モーター電源オン	69
7.3.2 モーター電源オフ	70
7.4 動作制御	70
7.4.1 ジョグ動作	70
7.4.2 クイック移動	71
7.4.3 ドラッグ	72
7.5 パスの再生	73
7.6 操作例 1: 産業用ロボットのジョグ動作	73
7.7 操作例 2: CR 協働ロボットのドラッグ	74
8 プログラミング	76
8.1 本章の概要	76
8.2 プロジェクト	76
8.3 RL エディタ	78
8.3.1 概要	78
8.3.2 ツールの紹介	79
8.3.3 プログラム補助	81
8.3.4 位置オフセット	88
8.3.5 Move to 機能	91
8.3.6 位置表示機能	92
8.4 プロジェクト設定	93
8.4.1 ロボットを選択	93
8.4.2 プロジェクト	93
8.4.3 同期	94
8.4.4 プロジェクトの復元	94
8.4.5 事前定義パラメーター	94
8.5 ユーザー定義生産画面	95
8.5.1 概要	95
8.5.2 基本操作	96
8.5.3 ウィジェット紹介	97
8.6 タスク一覧	105
8.6.1 タスクプロパティ	105
8.6.2 一般タスク、モーションタスク	106
8.6.3 自動実行タスク	108
8.6.4 タスク監視	110

8.7 変数一覧	111
8.7.1 変数の命名規則	111
8.7.2 変数のスコープ	111
8.7.3 変数と定数	111
8.7.4 システム予約語	112
8.7.5 進数変換	113
8.7.6 変数の宣言	113
8.7.7 持続変数 (PERS 変数)	114
8.7.8 変数一覧の操作	116
8.8 位置一覧	117
8.8.1 概要	117
8.8.2 操作例	120
8.9 パス一覧	123
8.9.1 概要	123
8.9.2 操作例	124
8.10 IO 信号一覧	125
8.10.1 概要	125
8.10.2 操作例	126
8.11 ユーザー座標系一覧	127
8.11.1 概要	127
8.11.2 ユーザー座標系のキャリブレーション	127
8.12 ツール一覧	128
8.12.1 概要	128
8.12.2 基本概念	128
8.12.3 操作例	133
8.13 ワーク一覧	138
8.13.1 概要	138
8.13.2 ワーク座標系の使用	139
8.13.3 操作例	140
8.14 変数モニター	143
8.14.1 概要	143
8.14.2 操作例	143
8.15 ビジョン	145
8.16 RL プログラムについて	145
8.16.1 RL プログラムの書式と構文	145
8.16.2 プログラムのデバッグ	146
8.16.3 デバッグの例	153
9 設定	155
9.1 本章の概要	155
9.2 コントローラー設定	155
9.2.1 基本設定	155
9.2.2 詳細設定	156

9.2.3 認証設定	158
9.3 HMI 設定	159
9.3.1 基本設定	159
9.3.2 教示操作盤モード	162
9.4 ユーザーグループ	163
9.5 キャリブレーション	164
9.5.1 ゼロキャリブレーション	164
9.5.2 ソフトウェアキャリブレーション	165
9.5.3 カセンサーキャリブレーション	166
9.5.4 ベース座標系設定	168
9.6 座標系設定	169
9.6.1 グローバルツール一覧	169
9.6.2 グローバルワーク一覧	173
9.6.3 グローバルユーザー座標系一覧	176
9.7 動力学設定	177
9.7.1 動的フィードフォワード	177
9.7.2 動力学制約	177
9.7.3 振動抑制	178
9.8 本体パラメーター設定	178
9.8.1 RD パラメーター	178
9.8.2 DH パラメーター	180
9.8.3 減速比	181
9.8.4 過負荷係数	181
9.8.5 結合係数	182
9.9 運動パラメーター設定	182
9.9.1 モーションパラメーター	182
9.9.2 詳細設定	184
9.10 力制御パラメーター設定	185
9.10.1 力制御パラメーター	185
9.10.2 力制御モデル	186
9.10.3 ドラッグの最適化	186
9.10.4 ドラッグの自動有効化	186
9.10.5 力制御モデル誤差閾値	186
9.10.6 センサーデュアルチャンネル電圧差閾値	186
9.11 クイック移動	187
9.12 電子銘板	188
9.13 エラーコードフィルター	191
9.14 カスタムボタン	192
10 通信	193
10.1 本章の概要	193
10.2 システム IO	193
10.2.1 システム入力	193

10.2.2 システム出力	195
10.3 外部通信	196
10.3.1 概要	196
10.3.2 設定	197
10.3.3 対話型指令	198
10.4 バスデバイス	203
10.4.1 バスデバイスの概要	203
10.4.2 バスデバイスのパラメーター設定	203
10.5 レジスタ	210
10.5.1 レジスタの概要	210
10.5.2 レジスタのパラメーター設定	211
10.5.3 レジスタの種類	213
10.5.4 レジスタの機能コード	214
10.5.5 レジスタの例	221
10.5.6 レジスタのリモートコントロール	222
10.5.7 レジスタのインポートとエクスポート	229
10.6 IO デバイス	231
10.6.1 概要	231
10.6.2 パラメーター設定	231
10.6.3 Modbus 拡張 IO の例	235
10.7 先端ツール	237
10.8 RCI 設定	238
10.9 xPANEL 設定	238
10.10 電動グリッパー/吸盤	239
10.10.1 概要	239
10.10.2 設定	239
10.11 シリアルポート設定	244
10.12 エンコーダー	245
10.13 OPC-UA	245
10.13.1 概要	245
10.13.2 オン/オフ	246
10.13.3 セキュリティ	246
10.13.4 証明書	247
10.13.5 カスタム変数設定	247
10.13.6 イベント(event)	248
11 安全機能	249
11.1 本章の概要	249
11.2 安全機能パスワード	249
11.3 軸制限	249
11.3.1 機能の概要	249
11.3.2 軸位置制限	250
11.3.3 軸速度制限	251

11.3.4 軸トルク制限	251
11.3.5 軸出力制限	251
11.4 ロボット制限	252
11.5 バーチャルフェンス	253
11.5.1 機能紹介	253
11.6 衝突検知	253
11.6.1 機能紹介	253
11.6.2 注意事項	256
11.7 安全エリア	257
11.7.1 機能紹介	257
11.7.2 安全エリアの設定方法	259
11.7.3 安全エリアの判定	259
11.7.4 安全エリア脱出機能	259
11.7.5 安全エリアとレジスタの関係	262
11.8 ツール設定	262
11.8.1 ツール位置制限	262
11.8.2 ツール姿勢制限	263
11.9 安全位置	263
11.9.1 機能紹介	263
11.9.2 安全位置とレジスタの関連付け	265
11.10 安全チェックサム	265
11.11 安全コントローラー	266
11.11.1 安全コントローラーを追加した後の変更点	266
12 パッケージ	268
12.1 コンベアトラッキング	268
12.2 走行軸	268
12.3 パレタイジング	268
12.4 トレイ	269
12.5 ソーラーバッテリー組立	269
12.6 ソーラーバッテリー挿入	269
13 ログ	270
13.1 HMI ログ	270
13.2 コントローラーログ	271
13.3 実行ログ	271
13.4 ログタイムライン	272
13.5 内部ログ	273
13.6 ハードウェアの状態	273
13.7 診断設定	274
14 オプション	275
14.1 接続	275
14.2 ROKAE について	276

14.3 ソフトウェア更新	276
14.3.1 コントローラーのアップグレード	276
14.3.2 コントローラーのバックアップ	276
14.3.3 HMI アップグレード	277
14.3.4 ロボット再起動	277
14.3.5 構成を消去	277
14.3.6 出荷時の設定に戻す	277
14.3.7 制御システムのアップグレード操作例	278
14.4 エクスポート	279
14.5 インポート	280
14.6 ファイルマネージャー	280
14.7 デモンストレーション	281
14.7.1 7 軸冗長動作	281
14.7.2 障害物回避動作	283
14.7.3 衝突検知	285
14.7.4 インピーダンス制御(柔軟性デモ)	287
15 RL 命令	289
15.1 変数の型	289
15.1.1 int	289
15.1.2 double	289
15.1.3 bool	289
15.1.4 string	290
15.1.5 array	290
15.1.6 byte	290
15.1.7 clock	291
15.1.8 暗黙の型変換	291
15.1.9 Confdata	291
15.1.10 jointtarget	293
15.1.11 load	293
15.1.12 orient	295
15.1.13 pos	296
15.1.14 pose	296
15.1.15 robtargt	296
15.1.16 signalxx	297
15.1.17 speed	298
15.1.18 tool	300
15.1.19 trigdata	302
15.1.20 wobj	302
15.1.21 zone	304
15.1.22 torqueinfo	306
15.1.23 SocketServer	306
15.1.24 SocketConn	307
15.1.25 FCBoxVol	309

15.1.26 FCSphereVol	309
15.1.27 FCXYZNum	310
15.1.28 FCCartNum.....	310
15.2 基礎変数と構造体	311
15.2.1 構造体の構成	311
15.2.2 構造体の使い方	311
15.3 関数.....	312
15.3.1 関数の定義	312
15.3.2 関数の呼び出し.....	313
15.4 命令.....	313
15.4.1 変数の型変換	313
15.4.2 動作命令	316
15.4.3 トリガー命令	327
15.4.4 力制御命令	332
15.4.5 ドラッグ再生命令	347
15.4.6 IO 命令	347
15.4.7 通信命令	350
15.4.8 ネットワークコマンド	360
15.4.9 論理命令	364
15.4.10 ホーム位置命令	372
15.4.11 数学命令	373
15.4.12 ビット操作.....	376
15.4.13 文字列操作	380
15.4.14 演算子	385
15.4.15 クロック命令	389
15.4.16 高級命令	390
15.4.17 機能命令	405
15.4.18 レジスタ命令.....	411
15.4.19 先端ツール命令	412
16 付録.....	422
16.1 ユーザー権限の詳細	422
16.2 協働ロボットのハンドルの紹介	424
16.2.1 ER シリーズ	424
16.2.2 CR シリーズ	424
16.3 位置、パスの教示(協働ロボットハンドルを使用する場合)	425
16.3.1 位置の教示	425
16.3.2 パスの教示	425
16.4 OPC-UA ROBOTICS モデル	426
16.4.1 MotionDevices モデル	427
16.4.2 Controllers モデル	429
16.4.3 SafetyStates	430
16.4.4 CustomVariables.....	431

17 トラブルシューティング	432
17.1 制御システムのエラーコード	432
17.1.1 1XXXX	432
17.1.2 3XXXX	452
17.1.3 4XXXX	506
17.1.4 5XXXX	509
17.1.5 6XXXX	517

1 概要

1.1 本書について

この度は、当社のロボットシステムをご購入いただき、誠にありがとうございます。

本書には xCore 制御システムに関する次の内容が記載されています。

- 制御システムの構成と基本操作
- 制御システムのプログラミングと高度なパラメーター設定
- 制御システムのオプション機能の紹介
- RL 命令セット
- 制御システムのエラーコード一覧

ロボットシステムを設置・使用する前に、本書およびその他の関連マニュアルをよくお読みください。本書をお読みにになった後は、いつでも参照できるように大切に保管してください。

1.2 本書の対象者

本マニュアルは、以下の読者を対象としています。

- オペレーター
- 製品技術者
- 技術サービススタッフ
- ロボットプログラマー

上記担当者は、機械・電気設備の設置や保守に必要な知識を有し、弊社から関連するトレーニングを受けていることを確認してください。

1.3 本書の読み方

本書には安全に関する章が含まれています。設置又は保守作業を行う前に必ずお読みください。

1.4 本書のイラスト

製品のバージョンアップなどにより、本書に掲載されている画像が実際の製品と異なる場合がありますが、操作方法・手順の説明は正しいです。

また、一般的な情報については、他の型番のロボットの画像を使用して説明することがあります。

1.5 お問い合わせ

ロボットの保守、修理、その他関連事項につきましては、弊社又は販売店にお問い合わせください。

お問い合わせの際は、以下の情報をご用意ください。

- コントローラーの型番/シリアル番号
- ロボットの型番/シリアル番号
- ソフトウェア名/バージョン
- システムの問題点

1.6 本書の読み方

本書は、以下の章に分かれています

章	タイトル	概要
1	概要	マニュアル全体の概要
2	安全	安全に関する事項
3	用語	本マニュアル内で使用される用語の説明
4	ロボットの基礎知識	ロボットに関する重要な基礎知識
5	ロボットシステムの構成と接続	ロボットシステムの構成及び物理的接続
6	HMI の概要	HMI の全体レイアウトと機能
7	制御システムの基本操作	制御システムで一般的に使用される基本操作； 例を用いて産業用・協働ロボットの基礎的な操作を説明
8	プログラミング	プログラミングモジュールの使用方法
9	設定	制御システムの各種設定
10	通信	通信機能の詳細と使用方法
11	安全機能	安全に関する機能
12	パッケージ	パッケージの概要
13	ログ	ログ機能に関する説明および使用方法
14	オプション	オプションモジュールの機能と使用方法
15	RL 命令	全ての RL 命令について詳しく説明
16	付録	ユーザー権限の詳細、協働ロボットの先端ハンドルの機能と使い方
17	トラブルシューティング	エラーコード及び対応手順一覧

1.7 改訂履歴

バージョン	日付	主な改定内容
V2.2	2024.05	xCore V2.2 バージョンの日本語マニュアル作成
V2.2-A-J1	2024.09	画像を日本語版に変更、誤記修正
V3.0-A-J1	2025.02	xCore V3.0 新機能の追加
V3.0-A-J2	2025.10	レジスタ機能コード：レジスタ長の修正 8→7

1.8 関連マニュアル

xCore 制御システムは、充実した基本機能に加えて、豊富な拡張機能を備えています。拡張機能に関して、以下のマニュアルを提供しています。

名前	概要
xCore 制御システム操作マニュアル	xCore 制御システムの基本機能の使用方法
xVision ユーザーマニュアル	xVision ロボットビジョンの基本的な使用方法
xCore_SDK_Android ユーザーマニュアル	xCore-SDK (Android 版)の使用方法
xCore_SDK_Python ユーザーマニュアル	xCore-SDK (Python 版)の使用方法
xCore_SDK_C++ユーザーマニュアル	xCore-SDK (C++版)の使用方法
xCore_RCI ユーザーマニュアル	RCI の使用方法
コンベアトラッキング機能 ユーザーマニュアル	コンベアトラッキング機能の使用方法
トレイパッケージ ユーザーマニュアル	トレイパッケージの使用方法
パレタイジングパッケージ ユーザーマニュアル	パレタイジングパッケージの使用方法
RokaeStudio ユーザーマニュアル	オフラインプログラミングソフトウェアの使用方法
ソーラーパネルレイアウトパッケージユーザーマニュアル	ソーラーパネルレイアウトパッケージの使用方法
ソーラーバッテリー挿入パッケージユーザーマニュアル	ソーラーバッテリー挿入パッケージの使用方法
Profinet ユーザーマニュアル	Profinet バスの使用方法
EthernetIP ユーザーマニュアル	EthernetIP バスの使用方法
外部走行軸 ユーザーマニュアル	外部走行軸の使用方法

2 安全

2.1 概要

本章では、ロボットを安全に使用するために注意すべき安全原則と手順について説明します。ロボットの外部安全装置の設計・設置に関する内容については、本書では説明しません。システムインテグレーターにお問い合わせください。

2.2 安全上の責任について

弊社は提供する情報の信頼性を確保するよう努めていますが、これに対して責任を負うものではありません。すべての作業が安全な操作手順に従って行われたとしても、ロボットが人身および財産の損失を引き起こさないことを保証するものではありません。

2.3 表記

ロボットを操作する場合は、様々な危険な状況に遭遇する可能性があります。本書には、危険を引き起こす恐れのある操作説明の近くにユーザーの注意を喚起するための安全注意記号などが配置されています。

その内容は以下の通りです。

- 危険レベルを示す図記号とそれに対応する名前（例えば：警告、危険、注意など）；
- 危険を除去しなかった場合に起こり得る結果についての簡単な説明；
- 危険の除去方法及び操作説明；

2.3.1 危険レベル

アイコン	名前	説明
	危険	規定に従って操作しないと、作業者の重傷または死亡事故につながり、ロボットに重大な損傷を与える可能性があることを示します。
	警告	規定に従って操作しないと、作業者の重傷または死亡事故につながる可能性があり、ロボットにも重大な損傷を与える可能性があることを示します。
	感電注意	感電の危険性があり、重傷又は致命傷を引き起こす可能性があることを示します。
	注意	規定に従って操作しないと、人身事故につながる可能性があり、またロボットにも損傷を与える可能性があることを示します。
	静電気防止 (ESD)	静電気に弱い部品に関して、仕様に従って操作しないと、部品に損傷を与える可能性があることを示します。
	ヒント	重要な情報や前提条件を示します。

2.3.2 リスク説明

アイコン	名前	説明
	挟まれ注意	デバッグ、整備、点検、設置の際に、作業者がロボットの動作範囲に立ち入ると、怪我をするおそれがあります。

	指づめ注意	作業中に作業者が駆動部品に近づくと、手が挟まれる危険があります。
	衝突注意	デバッグ、整備、点検、設置の際に、作業者がロボットの動作範囲に立ち入ると、重大な怪我をするおそれがあります。
	摩擦注意	デバッグ、整備、点検、設置の際に、作業者がロボットの動作範囲に立ち入ると、怪我をするおそれがあります。
	部品飛出し 注意	デバッグ、整備、点検、設置の際に、作業者がロボットの動作範囲に立ち入ると、ツールやワークが飛び出すなど、重大な怪我をするおそれがあります。
	火災注意	電気回路がショートしたり電線や部品が発火したりすると、火災が発生し重大な損害を引き起こすおそれがあります。
	高温注意	機器の点検や整備を行う際に、高温部表面に触れると、やけどを負うおそれがあります。

2.4 安全停止

ロボットの停止には、STOP 0、STOP 1、STOP 2 の 3 種類があります。

安全停止は、安全コントローラーによってトリガーされる停止です。安全コントローラーは STOP 0 と STOP 1 をトリガーします。STOP 2 は制御システムによってトリガーされます。

STOP 0	STOP 0 がトリガーされると、直ちにモーターへの電源供給が遮断され、各関節のブレーキが作動します。これは最も速い停止方法ですが、停止中はロボットが制御不能な状態になり、停止後にプログラムされた経路から逸脱する可能性があります。 手動モードの安全停止は STOP 0 です。 STOP 0 は、最大能力減速停止を行います。
STOP 1	STOP 1 がトリガーされると、制御システムはプログラムされた経路に沿って直ちに減速を実行します。その後、ロボットが完全に停止したかどうかにかかわらず、安全コントローラーはモーターへの電源供給を遮断し、各関節のブレーキが作動します。この停止方法は制御された停止であり、ロボットがプログラムされた経路上で停止するため、周辺機器に最適な保護を提供します。 自動モードでのセーフティゲート/セーフティライトカーテンの作動及び自動モードで非常停止ボタンを押すことによる停止は、すべて STOP 1 になります。 STOP 1 は、最大能力減速停止を行います。
STOP 2	STOP 2 がトリガーされると、制御システムはプログラムされた経路に沿って直ちに減速を実行します。ロボットが完全に停止するまでモーターへの電力供給は維持されます。 STOP 2 には、一定時間停止、一定距離停止、最大能力減速停止、通常計画停止の 4 種類があります。



ヒント

1. 非常停止は、危険な状況でロボットを直ちに停止させるためにのみ使用します。

ロボットのブレーキシステムやドライブトレインに異常な摩耗を引き起こし、ロボットの寿命を縮めるおそれがあるため、非常停止を通常のプログラム停止として使用しないでください。

2. 安全停止トリガー時、停止パラメーターが正しく設定されていない場合は、ポップアップエラーメッセージが表示されます。エラーメッセージは2種類あります。

- (1) 設定ファイルのバージョンが異常な場合はエラーとなります。設定ファイルに対応する安全停止パラメーターが不足している場合は、エラーメッセージが表示され、デフォルトのパラメーターを使用して安全停止を行います。

- コントローラーログ

詳細
内容: 構成ファイルのバージョンを確認してください。:SAFETY_CONTROL.STOP0_TIME、デフォルト構成に戻してください:50.000000;
原因: 現在の構成ファイルには、安全停止パラメーターフィールドがありません。
修復方法: 最新の設定ファイルアップグレードしてください。

- (2) 安全停止パラメーターの設定が異常な場合はエラーとなります。安全停止パラメーターが正しく設定されていない場合は、エラーメッセージが表示され、デフォルトのパラメーターを使用して安全停止を行います。例えば、安全停止パラメーターが0に設定されている場合などです。

← コントローラーログ

詳細
内容: 安全停止パラメーターの設定が異常です。
原因: 安全停止パラメーターの値が異常です。値が0です。
修復方法: 安全停止パラメーターを適切な値に変更してください。

2.5 安全装置

2.5.1 非常停止ボタン

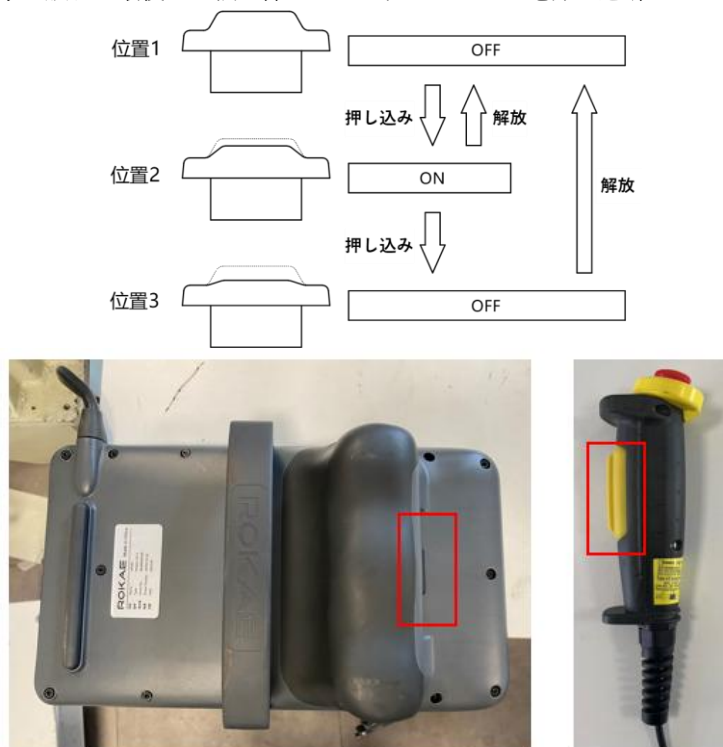
非常停止ボタンは、手動で非常停止を行うボタンです。多くは赤色のキノコ型であり、通常は黄色の板、保護カバー又は警告サインが装備されています。

非常停止ボタンを押すと、ボタンは機械的にロックされます。これは非常停止ボタンの安全ロック機構であり、手動で解除する必要があります。ほとんどの非常停止ボタンは回転解除を採用しており、回転方向はボタンの表面に表示されています。また、直接引き上げる解除方法のものもあります。



2.5.2 イネーブル装置

イネーブル装置は、2段階の押し込み・3つのポジションを持った特別なスイッチで、3ポジションイネーブルスイッチ（以下、イネーブルスイッチ）とも呼ばれます。手動モードでロボットのモーター電源供給を制御し、ロボットの動作を有効にします。イネーブルスイッチを中間位置に保持している状態でのみモーターに電源が入り、ロボットは動作可能な状態になります。手を放すか最後まで強く押し込むと、モーターの電源が遮断されます。



ヒント

携帯型安全装置では、黄色いボタンがイネーブルスイッチです。中間位置まで押し込み、モーターの電源が入り、ジョグやプログラムの実行が可能になります。手を放すか最後まで強く押し込むと、モーターの電源が遮断されます。

安全に使用するために、次の要件に従う必要があります。

1. いかなる場合でも、イネーブルスイッチが正しく機能することを確認してください。
2. プログラミングやデバッグ中にロボットの動作が必要ない場合は、できる限り早くイネーブルスイッチを解放してください。
3. ロボットの動作範囲に立ち入る者は、その他の者が知らないうちにロボットを動作させるのを防ぐために、携帯型のイネーブルスイッチを携帯する必要があります。



警告

外部機器を使用してイネーブルスイッチを中間位置に保持することは禁止されています！

2.6 様々な状況における安全上の注意事項

2.6.1 手動モードでの安全上の注意事項

手動モードでは、ロボットの動作は手動で制御されます。イネーブルスイッチが中間位置にあるときのみ、ロボットのジョグ、プログラム実行を行うことができます。手動モードは、ロボットプログラムの作成とデバッグ、ワークステーションの試運転調整に使用されます。

2.6.1.1 手動モードでの速度制限

手動モードでは、プログラムで設定された速度に関係なくロボット先端の最大動作速度は250mm/sに制限されています。

2.6.1.2 外部安全信号のバイパス

手動モードでは、セーフティーゲートやセーフティーライトカーテンなどの外部安全装置からの信号はバイパスされます。セーフティーゲートが開いていても、システムはセーフティーゲートが開いているというメッセージを出しません。ロボットは動作を実行できるため、デバッグが容易になります。この信号は、CRシリーズ及びSRシリーズでは停止エリア入力信号に該当します。

2.6.2 自動モードでの安全上の注意事項

自動モードは、正式な生産プロセスでロボットプログラムを実行するために使用されます。自動モードではイネーブルスイッチがバイパスされるため、ロボットは人間の介入なしに自動的に動作できます。

2.6.2.1 外部安全信号の有効化

セーフティーゲートやセーフティーライトカーテンなどの外部安全装置は、自動モードで有効になります。セーフティーゲートを開けるとモーターの電源が遮断され、ブレーキが作動します。この信号は、CRシリーズ及びSRシリーズでは停止エリア入力信号に該当します。

2.6.3 設置・操作時の安全要求事項

- 誤った操作により、ロボットが転倒し作業者の人身事故や設備の破損が発生する可能性があるため、当社のマニュアルに記載された方法に従ってロボットを運搬、設置してください。
- ロボットを設置後初めて使用するとき、最初は必ず低速で動作させ、徐々に速度を上げてください。
- プログラムやシステム変数などの情報は、デフォルトでコントローラーの記憶装置に保存されています。事故によるデータ損失を防ぐため、定期的にデータのバックアップをしてください。

2.6.4 デバッグ時の安全要求事項

デバッグはできる限りロボット動作エリア外で行ってください。やむを得ずロボット動作エリア内で行う場合は、次の要求事項に特にご注意ください。

- ロボット動作エリアに入る前に、エリア内の状況を注意深く観察し、危険がないことを確認してください。
- ロボット動作エリア内のすべての作業員の位置を確認してください。
- システム全体の状態を確認した上で作業してください。

- 非常停止ボタンはいつでも押せるようにしてください。
- ロボットは低速で動作させてください。
- デバッグ後は、作業員はロボット動作エリア外で作業してください。

2.6.5 保守時の安全要求事項

- ロボット動作エリアに入る前に、エリア内の状況を注意深く観察し、危険がないことを確認してください。
- ロボット動作エリア内のすべての作業員の位置を確認してください。
- 感電の危険があるため、できる限りロボット設備とシステムの電源を切ってから保守作業を行ってください。
- 保守中は、他の人が不用意に電源を入れないように対策してください。
- 不要な人身傷害や装置への悪影響を避けるために、体のいかなる部位もロボット設備のいかなる部分に当てないでください。
- 保守中は、適切な照明設備を使用してください。
- 指定以外の部品を使用するとロボット設備が破損する可能性があるため、部品交換が必要な場合は必ず当社指定の部品を使用してください。
- 部品交換の際に外した部品（ねじなど）は、元の位置に正しく取り付けてください。部品が不足している又は残っている場合は、再度確認し正しく取り付けてください。

2.6.6 生産ラインでの安全対策

ほとんどの場合、ロボットは生産ラインの一部であるため、ロボットの故障はロボットだけではなく生産ライン全体に影響します。同様に生産ラインの他の部分で問題が発生した場合にもロボットに影響する可能性があります。したがって、システム全体の安全性を向上させるために、生産ライン全体を熟知している担当者が故障復帰計画を作成する必要があります。

- ロボットと相互作用する他の装置に注意してください。

例えば、ロボットの保守が必要な場合は、まずそのロボットを生産ラインから切り離します。また、ワークを供給するロボットなど、相互作用する他の装置からも切り離す必要があります。

- ロボットの周りで動作している他の装置に注意してください。

例えば、生産ライン上のロボットがコンベアからワークを掴む場合、ロボットが故障した際に、生産プロセスが中断されないようにロボットの修理中でもコンベアが動作していることがあります。このときは、ロボットの保守担当者は特に安全に注意を払う必要があります。走行中のコンベアによって発生する恐れのあるリスクを事前に考慮し、そのような環境で作業するための詳しい安全対策を講じる必要があります。

2.6.7 火災時の安全

火災の危険が迫っている場合、または火災が発生したがまだ燃え広がっていない場合は、あわてず落ち着いて、現場に備え付けの消火器を使用して消火してください。ショートによる火災に対しては、水を使用しないでください。



警告

ロボットの作業現場で使用する消火装置はユーザーにてご用意ください。現場の実情に応じて適切な消火装置を選択してください。

火災が拡大し制御不能な状態になった場合は、現場にいる作業者は消火活動を行わず、直ちに他の作業者に連絡し、何も持たず非常口から速やかに避難してください。避難時はエレベーターを使用しないでください。また、迅速に消防署に連絡してください。

衣服に火がついた人がいる場合は、すばやく地面に寝かせ、衣服などの適切な道具や方法で消火してください。

2.6.8 感電時の安全

感電している人が発見された場合は、慌てず、まずは速やかに電源を遮断してください。現場の状況に応じて、適切な方法・措置を講じてください。一般的には次のような方法と対策があります。

1. 電源スイッチまたはボタンが感電が発生した場所に非常に近い場合は、素早くスイッチを使用して電源を遮断してください。
2. 電源スイッチまたはボタンが感電が発生した場所から遠く離れている場合は、絶縁ベンチ又は乾いた木製の柄が付いた斧、ナイフ、シャベルなどを使用して、電源側（電源供給側）のケーブルを切断してください。切断されたケーブルが人体に触れないようにしてください。
3. ケーブルが感電している人の体に接触している、または体の下に入り込んでいる場合は、乾燥した木の棒、板、竹の棒又は絶縁された柄のついた道具で素早くケーブルを取り除いてください。救助する者が感電しないように、金属棒や濡れたものでケーブルに触れてはいけません。

感電の負傷者を電源から離れた後の措置：

1. 負傷者の意識がある場合は、その場で仰向けに寝かせ、よく観察してください。当分の間立たせたり動かしたりしないでください。
2. 負傷者が意識不明の場合は、その場で仰向けに寝かせてください。気道が開いていることを確認し、5 秒間隔で負傷者に声をかけたり肩を軽く叩いたりし、意識の有無を判断してください。負傷者の頭を振って声をかけないでください。その場で救護を実行しながら、速かに救急に連絡してください。
3. 負傷者が意識を失っており呼吸や心拍が停止している場合は、10 秒以内に呼吸数と心拍数を測定してください。呼吸も脈拍もない場合は、直ちに心肺蘇生を行ってください。

3 用語

この章では、マニュアル内に登場する一部の用語について簡単に説明します。

用語	説明
RobotAssist	ROKAE ロボット・xCore 制御システムのホストコンピュータソフトウェアです。ロボットの動作制御、プログラム作成、パラメーター設定、状態監視などの機能を備えています。xPad2 教示操作盤や PC などで行えます。
HMI	Human Machine Interface の略です。操作をする際のユーザーインターフェースです。
HMID	HMI Device の略です。操作をする際のデバイスです。
RC	Robot Controller の略です。ロボットのコントローラーです。
RCI	Rokae Control Interface の略です。ROKAE ロボットの外部制御インターフェースで、リアルタイム制御をサポートします。
SDK	Software Development Kit の略です。C++などのプログラミング言語によるロボットの制御をサポートするソフトウェア開発ツールキットです。将来的に RCI を廃止し、段階的に SDK に置き換えていく予定です。
Project	プロジェクトは、ロボット動作を制御するプログラムやタスクなどのオブジェクトの集合体です。プロジェクトのデータはエクスポートでき、他のプロジェクトやロボットで再利用できます。
Task	xCore 内でタスクと呼ばれるものです。
Module	xCore 内でモジュールと呼ばれるものです
Elbow	アーム平面と基準平面の間の角度を指します。アーム平面とはロボットの上腕と前腕で形成される平面で、基準平面は 3 軸がゼロのときにエンドエフェクタが同じ姿勢になる位置で形成される平面を指します。
RL	Rokae Robot Language の略です。豊富な命令を含む ROKAE ロボットのプログラミング言語は、RL に基づいてロボットのタスクを構築します。
xPad2	教示操作盤の名称です。
RSC	Robot Safety Controller の略です。安全コントローラーです。
JOG	ジョグ動作のことです。
零空間動作	冗長自由度を持つロボットでは、零空間動作を通じて、ロボットの関節は動くがエンドエフェクタは静止したままの動きを実現できます。
PERS 変数	プログラムの実行中、PERS 変数の値が変化すると、初期値が自動的に現在の値に更新され、「永続的」に保存される効果を持ちます。

4 ロボットの基礎知識

4.1 概要

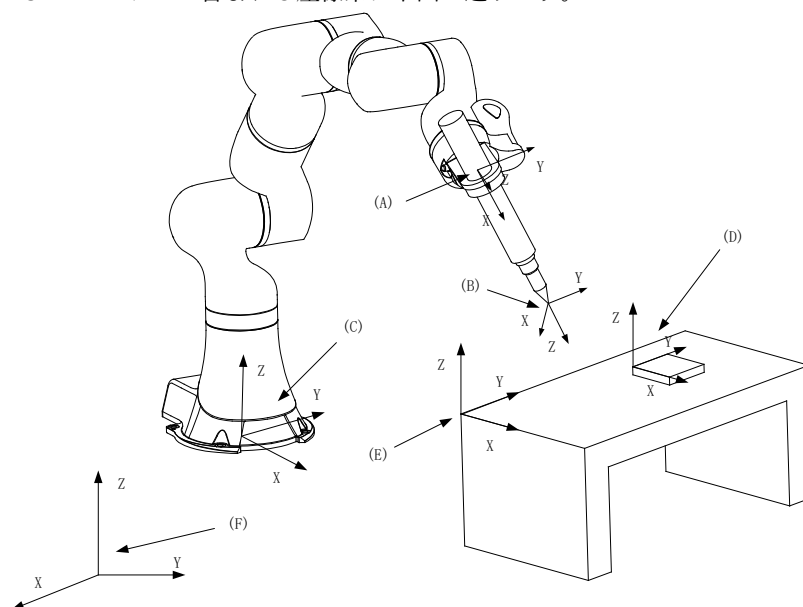
本章では、ロボット工学に関する基礎知識を紹介します。本章の内容を理解することで、制御システムやロボットの使い方をより深く理解し、効果的に活用できるようになります。

4.2 座標系

空間内の物体（ツール、ワークなど）は、3つの並進自由度と3つの回転自由度、合計6つの自由度を持っています。3つの並進自由度は位置を構成し、3つの回転自由度は姿勢を構成します。物体の位置と姿勢はその物体に付属する座標系によって記述することができ、一般的には直交座標系（以下、座標系という）が用いられます。

ロボットは多自由度機構であり、一般的にはフランジに固定されたツールを用いて外部のワークに対して相対運動を実行します。この動作は、座標系およびその相対運動で記述することができます。

xCore システムに含まれる座標系は下図の通りです。



記号	座標系	意味
A	フランジ座標系	ロボットフランジの中心に定義されます。 フランジ座標系はベース座標系に対して定義されます。
B	ツール座標系	ツール先端に定義されます。 ツール座標系は、ツールがフランジに取り付けられている場合は、フランジ座標系を基準にして定義されます。 ツールが外部に設置されている場合は、ユーザー座標系を基準にして定義されます。
C	ベース座標系	ロボットベースの中心に定義されます。 ベース座標系はワールド座標系に対して定義されます。 注意：ロボットがデフォルトの設置方法で設置されていない場合（天吊り設置や傾斜設置など）は、まずはベース座標系を設定する必要があります。

D	ワーク座標系	ワークに定義される座標系です。 ワーク座標系は、ワークが外部（一般ワーク）の場合は、ユーザー座標系に対して定義されます。 ワークがフランジに取り付けられている場合は、フランジ座標系に対して定義されます。
E	ユーザー座標系	単独で使用されることなく、ワーク座標系を定義する際の基準系として使用されます。 ユーザー座標系はベース座標系に対して定義されます。
F	ワールド座標系	ワールド座標系は、一般に基準となる座標系として使用され、特定の位置を持ちません。単体のロボットが通常設置されている場合は、デフォルトでロボットのベース座標系と一致します。複数のロボットや周辺機器が連携する場合は、それぞれのワールド座標系を同じ座標系に統一することで、動作の基準の統一を実現できます。

4.3 特異点

ロボットの動作範囲には、無数の異なる関節構成で到達できる特別な位置がいくつかあり、これらの位置は特異点と呼ばれます。制御システムが直交空間の位置に基づいて関節角度を計算する際に、特異点による問題が発生する可能性があります。

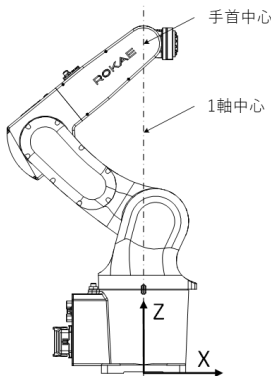
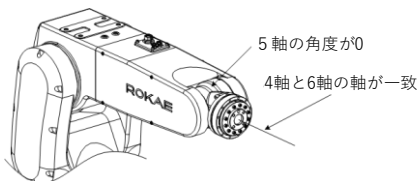
ロボットが軸動作を実行する際には、特異点の問題は存在しません。

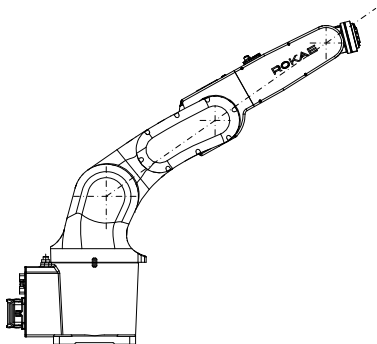
ロボットが特異点付近を直線移動する際に、ある関節の速度が非常に速くなり、エラーが発生してロボットの動作が中止されることがあります。

4.3.1 ロボットの典型的な特異点

ロボットの構成によって特異点は異なります。以下では、いくつかのロボット構造における典型的な特異点について説明します。

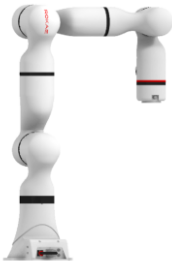
4.3.1.1 6 軸ロボットの特異点

特異点	構成	説明
肩部特異点		ロボットの手首の中心が 1 軸の軸上にある場合。
手首特異点		4 軸と 6 軸の軸が一致している場合（5 軸の角度は 0）。

肘部特異点		手首の中心、2 軸の回転軸、3 軸の回転軸が一直線上にある場合。
-------	---	----------------------------------

4.3.1.2 ER PRO 協働ロボットの特異点

ER PRO 協働ロボットの特異点は次のように分類されます。

特異点	構成	説明
2 軸特異点		2 軸の角度が 0° の場合は、逆運動学を解く際に 1 軸の角度と 3 軸の角度を区別できません。
4 軸特異点		4 軸の角度が 0° の場合は、3 軸または 5 軸に平行な方向へのロボットの動作が制限されます。この特異点により、ロボットは手首の付け根の運動の自由度が 1 つ失われます（手首の付け根がアーム軸に沿って動作できなくなります）。このときに逆運動学を解く場合は、3 軸と 5 軸の位置は解けません。
6 軸特異点		ロボットの 6 軸の角度が 0° の場合は、逆運動学を解くときに 5 軸の角度と 7 軸の角度を区別できません。

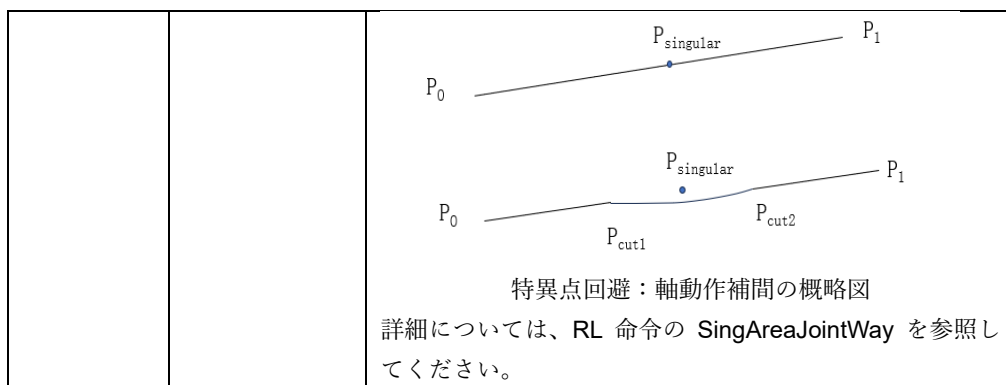
手首中心特異点		手首の中心が 1 軸の真上にある場合は、逆運動学を解くときに 1 軸の角度を正しく解けません。
---------	---	---

4.3.2 特異点回避

特異点の問題は、ロボットの構成に起因するものであり、完全に回避することはできません。実際のプログラミングにおいて、ロボットが特異点付近を通過する必要がある場合は、スムーズに特異点を通過できるように、いくつかの制約（姿勢や軌跡精度など）を設けて特異点の回避を検討してください。

xCore 制御システムは、様々な特異点回避方法を提供します。

方法 1	4 軸ロック	この特異点回避方法を有効にする前に、ロボットの 4 軸を 0 度または ± 180 度にしてください。この方法を開始すると、ロボットの 4 軸を動かさず、TCP 位置の精度を確保しながら、特殊な形式で姿勢補間を行います。RL 命令の <code>SingAreaLockAxis4 on/off</code> とジョグモードの章を参照してください。
方法 2	形態変更補間	この特異点回避方法を有効にすると、ロボットの形態を変更することで手首の特異点を通過させることができます。 注意：この機能を使用する場合は、ロボット動作時の手首の形態（特異点を通過する教示点だけでなく、後続の教示点の形態も含む）が教示内容と異なる場合があります。
方法 3	軸動作補間	この特異点回避方法を有効にすると、制御システムはこの機能を終了するまで後続の直線動作で特異点の検出を実行します。 特異点を含まない直線動作は、通常の軌跡に従って動作します。特異点を含む直線動作の場合は、制御システムは特異点を検出した後、特異点を中心にして軌跡 P_0 - P_1 を P_0 - P_{cut1} 、 P_{cut1} - P_{cut2} 、 P_{cut2} - P_1 の 3 つに切り分けます。 P_0 - P_{cut1} と P_{cut2} - P_1 は通常の直線動作を行い、 P_{cut1} - P_{cut2} は軸動作を行います。下図に示すように、3 つのセグメント間をスムーズに移行するために、滑らか軌跡を利用します。



注意：

- ロボットが特異点付近にいる場合は、関節の動作範囲が非常に大きくなる場合があります。特異点回避命令を使用する前に、まずは教示点を変更することで、特異点を回避することをお勧めします。
- 特異点回避命令を使用する場合は、特異点回避命令追加後のロボットの軌道が動作要件を満たしていることを確認してください。
- ロボットが特異点付近にいる場合、関節の動作範囲が非常に大きくなる場合があるため、使用前に周囲の環境を確認してください。
- 上記の理由により、ロボットの動作やプログラムロジックが外部信号の影響を受ける場合は、プログラムロジックと軌跡をよくご確認の上ご使用ください。

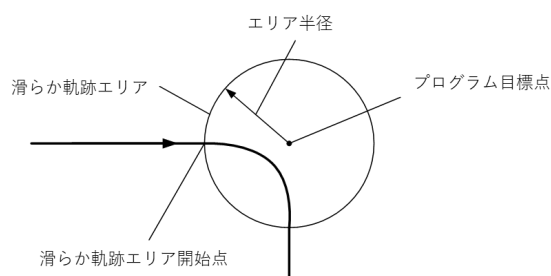
上記 3 つの特異点回避の具体的な特徴は次の通りです。

方式と特徴	4 軸ロック	形態変更補間	軸動作補間
動作の特徴	- この命令をオンにする前に、ロボットの 4 軸を 0 度または ± 180 度にする必要があります。 - この命令をオンにした後は、ロボットは次の動作命令で 4 軸を動かさないようにします。	- この命令をオンにすると、ロボットは手首の形態とツールの姿勢を変更し、手首の特異点を含む軌跡を通過できるようにします。 - 動作時の手首の形態（特異点を通過する教示点だけでなく、後続の教示点の形態も含む）が教示内容と異なる場合があります。	軌道上の特異点の前後の一部は軸動作（MoveAbsJ）で動作し、残りの部分は元の直線動作で動作します。この軸動作と直線動作の切り替わりは、滑らか軌跡を利用して遷移します。
軌跡の変化	- 位置軌跡は変わりません - 特殊な姿勢補間方法をとります	- 位置軌跡は変わりません - 姿勢補間方法を変更します	- 位置軌跡は変わります - 姿勢軌跡は変わります
目標点への到達可能性	ロボットの動作範囲の一部に到達可能です。具体的には、4 軸は 0 度、または ± 180 度の目標点に到達可能です。	ロボットの動作範囲の一部に到達可能です。	ロボットの動作範囲全体に到達可能です。
滑らか軌跡	- SingAreaLockAxis4 on と SingAreaLockAxis4 off の間の動作は、滑らか軌跡の生成をサポートします。 - SingAreaLockAxis4 on/off 命令前後の動作は、滑らか軌	- SingAreaWrist on と SingAreaWrist4 off との間の動作は、滑らか軌跡の生成をサポートします。 - 軸動作と特異点回避軌道との間では滑らか軌跡を生成	特異点回避軌跡間、特異点回避軌跡と通常軌跡との間で、滑らか軌跡を生成できます。

	跡の生成をサポートしません。	できます。 直線動作と特異点回避軌道との間では、滑らか軌跡を生成できません。	
先読み	先読みを中断します。すなわち SingAreaLockAxis4 on/off はブロッキング命令です。ロボットが SingAreaLockAxis4 on の前の軌跡に到達して初めて、制御システムは特異点回避命令の解析を続けます。同様に、ロボットが SingAreaLockAxis4 off の前の動作指令に到達して初めて、制御システムは後続の命令の解析を続けます。	先読みを中断しません。	先読みを中断しません。
特異点回避の影響範囲	4 軸ロックをオンにした後、当該機能をオフにするまで、すべての直線動作命令が 4 軸ロックに対応する特別な補間方法で補間されます。	形態変更補間をオンにした後、当該機能をオフにするまで、すべての直線動作命令は特別な補間方法で補間されます。	本特異点回避をオンにすると、制御システムは直線動作軌跡ごとに特異点を含むかどうか自動的に検出します。特異点を含む軌跡のみが特別な方法で補間され、特異点を含まない軌跡は通常の動作形式で動作します。 注意：オンにすると制御システムの計算量が増えるので、必要な場合以外はオンにしないでください。
活用シーン	1. フランジがベースと平行、またはベースの Z 軸に沿って運動する軌道の場合。 2. パレタイジング動作など、4 軸を動かさない動作の場合。	軌跡の姿勢精度やロボットが目標点に到達する形態を考慮せず、目標点の直交位置に到達できることのみを求める場合。	軌跡の位置精度・姿勢精度・ロボットが目標点に到達する形態を考慮せず、目標点に到達できることのみを求める場合。
対応機種	標準 6 軸産業用ロボットシリーズ (XB、NB モデル) 協働ロボット xMate CR/SR	標準 6 軸産業用ロボットシリーズ (XB、NB モデル) 協働ロボット xMate CR/SR	標準 6 軸産業用ロボットシリーズ (XB、NB モデル)
Jog への対応	対応	非対応	非対応

4.4 滑らか軌跡エリア

ロボットの動作は、プログラムによって設定された複数の軌跡を順次実行します。通常、これらの軌跡は滑らかにつながっているわけではなく、様々な“停止点”があります。これらの“停止点”のため、ロボットが次の軌跡を開始する前に 1 つの軌跡の終点で停止する必要があります。ロボットが異なる軌跡間を連続的に動作するために、前述の“停止点”をなくし、異なる軌跡を「滑らか軌跡」を介して滑らかにつなげる必要があります。下図を参照してください。



滑らか軌跡には、直線動作滑らか軌跡と軸動作滑らか軌跡があります。滑らか軌跡の詳細な定義とパラメーターについては、「RL 命令」 - 「zone」の章を参照してください。

4.5 先読み機能

先読み（Lookahead）とは、ロボットの動作中に制御システムが現在実行されている命令の後のプログラム命令を事前に処理することです。先読み機能には次のような利点があります。

- 先の軌跡の速度と加速度の要件およびロボット自体の制限条件の情報を取得することで、最適な制御戦略を計画できます。
- プログラムされた滑らか軌跡の設定に従って、滑らか軌跡を計画します。
- ソフトウェアリミット/境界に近い場合や特異点に近いなどの異常状態を取得し、事前に処理できます。

先読み機能の詳細については、後述の「プログラミング」 - 「RL プログラムについて」 - 「RL プログラムのデバッグ」を参照してください。

4.6 力制御

4.6.1 力制御の概要

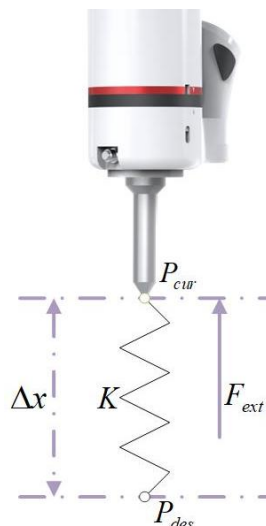
ロボットの力制御は、ロボット先端と外部環境との間に力の相互作用のある制御プロセスです。

相互作用のないロボットの制御プロセスでは、位置制御（速度と精度）のみが考慮されます。相互作用がある場合は、単純な位置制御ではロボットと環境に非常に高い精度が要求され、位置のずれによって生じる接触力からの損傷を避ける必要があります。

単純な位置制御とは異なり、ロボット力制御は環境との相互作用中に力/トルクのフィードバックループを導入します。力/トルクのフィードバックループを通じてロボットの動作特性を変更することで、外部環境との動的相互作用を実現します。ロボットと外部環境の間に偏差や不確実性がある場合は、力制御は位置軌道を賢く調整し、位置偏差によって引き起こされる内力を排除し、スムーズで安全な相互作用プロセスを保証します。

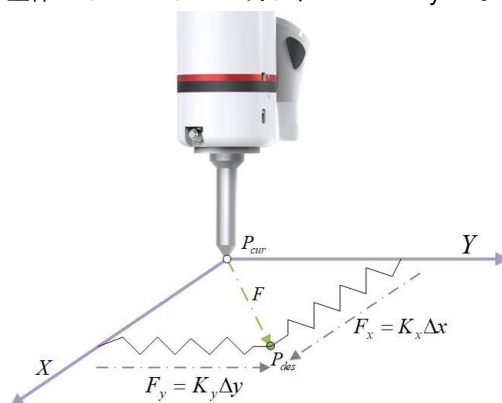
4.6.2 インピーダンス制御

従来の産業用ロボットと比較し、**xMate** 協働ロボットは関節にトルクセンサーを備えており、関節トルクを正確に感知できます。関節のトルク情報により、**xMate** 協働ロボットはインピーダンスを介して力制御を実現でき、インピーダンス制御によってロボットは相互作用のある動作を実現しています。これにより、ロボットと環境の相互作用が、仮想的なバネの剛性および減衰システムとして表せます。このときロボットは外力に敏感であり、外力の作用によりロボットが設定された軌道からずれる可能性があります。外力がなくなると、バネの反発効果で元の軌跡に戻ります。



インピーダンス動作中は、外力の作用によりロボットの実際の位置は目標位置から一定の距離だけずれます。ずれの距離はインピーダンス剛性と外力の大きさに依存し、偏差は外力とインピーダンス剛性の比によって求めることができます。

上図に示すように、インピーダンス制御モードでは、インピーダンス剛性を「 K 」に設定した時、外力 F_{ext} の作用により、ロボットの現在位置 P_{cur} が目標位置 P_{des} から位置偏差 Δx だけずれます。この偏差によって生じるインピーダンス力と外力は最終的にバランスがとれます。各方向のインピーダンス剛性は個別に設定でき、各方向のインピーダンス力はその方向のインピーダンス剛性と位置偏差の積となります。最終的に各方向のインピーダンス力が合成されて全体のインピーダンス力となります。下図に示すように、インピーダンスモードでは、外力の作用により、ロボットの現在位置 P_{cur} が目標位置 P_{des} からずれます。X、Y 方向の偏差をそれぞれ Δx と Δy 、インピーダンス剛性をそれぞれ K_x と K_y 、インピーダンス力をそれぞれ F_x と F_y とします。全体のインピーダンス力は、 $F = F_x + F_y$ となります。

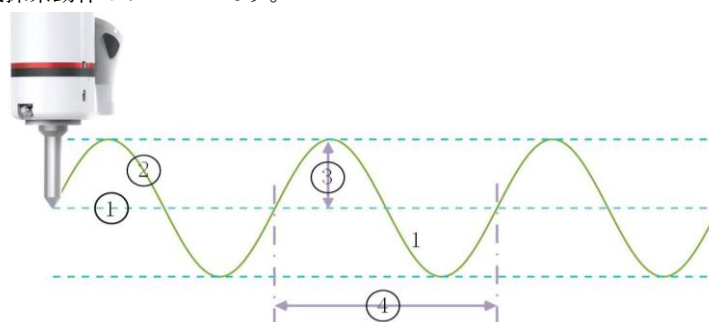


4.6.3 力制御探索

力制御探索は、人間の手が部品組立中に力の変化を感知でき、障害物（ワーク詰まり）を感知した場合はワークがスムーズに組み立てられるように振動を加える手法を模倣したものです。

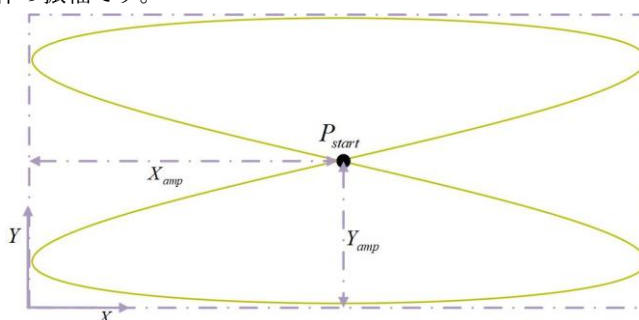
ロボットは、力制御探索として軸周りに回転する正弦波探索動作や平面内のリサージュ曲線探索動作をサポートします。探索動作はロボットの所定の動作に重ねて追加される動作で、組立中に障害物をよりよく克服できるように、ロボットに一定量の振動を生じさせます。

下図は正弦波探索動作のプロセスです。



①	目標軌跡
②	実際の軌跡（目標軌跡+探索動作）
③	探索動作振幅
④	探索動作周期

リサージュ曲線探索動作とは、平面内で互いに直交する 2 方向にそれぞれ正弦波探索動作を加えることです。2 方向の正弦波探索動作の周波数は、通常一定の比率を持っています。例えば、下図は XY 平面内のリサージュ曲線探索動作を示しています。X と Y 方向の探索動作の周波数の比率は 2:1 です。中心点 P_{start} は目標点、 X_{amp} は X 方向の探索動作の振幅、 Y_{amp} は Y 方向の探索動作の振幅です。

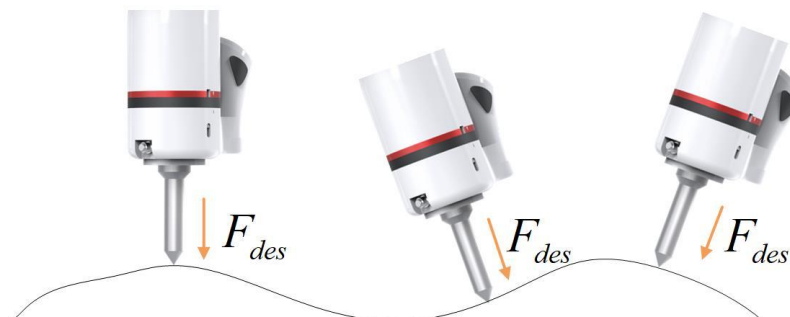


4.6.4 力制御の応用

産業用ロボットの力制御の適用シーンは、定力追跡と力制御組立の2つに大別されます。

4.6.4.1 定力追跡

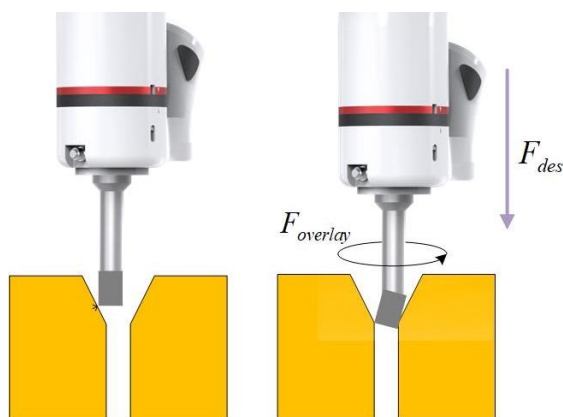
下図に示すように、定力追跡ではロボットは曲面との接触力を一定値 F_{des} に維持し、曲面の起伏の変化に適応します。定力追跡は主に研削やバリ取りなどで使用されます。



4.6.4.2 力制御組立

単純な位置制御では、組立工程中に位置やモデリング誤差によりロボットがワークに衝突しやすく、ワークやロボットに損傷を与える可能性があります。

力制御による組立では、ロボットは一定範囲を超えた外力（ワーク詰まり）を感知した場合は、障害物を克服してワークがスムーズに組み立てられるように探索動作（振動）を加えます。下図に示すように、左側の単純な位置制御は衝突を引き起こしますが、右側の力制御は目標力 F_{des} によってロボットを組立穴に押し込み、探索動作 $F_{overlay}$ によってワーク詰まりを克服します。



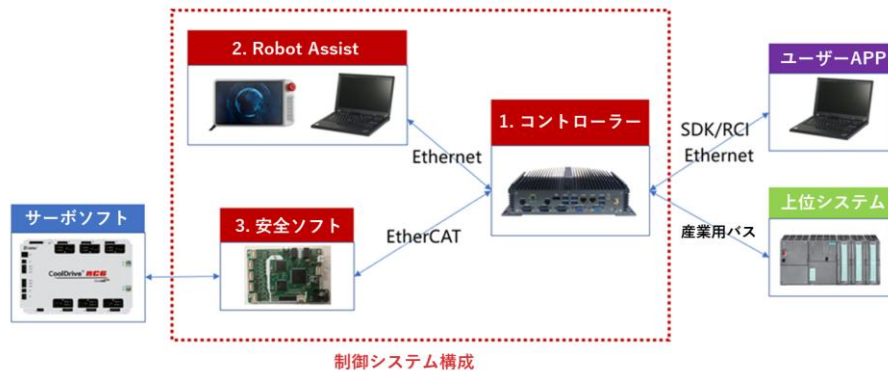
5 ロボットシステムの構成と接続

5.1 本章の概要

ROKAE ロボットは複数のシリーズに分かれています。本章では各シリーズのロボットのシステム構成と接続方法を紹介し、ロボットシステムへの理解を深めます。
ユーザーはご使用の機種に応じて、本章の内容を選択して読んでください。

5.2 制御システムの構成

xCore 制御システムは CS アーキテクチャを採用しており、主に HMI ソフトウェア (RobotAssist) とコントローラソフトウェア RC で構成されています。



5.2.1 xPad2 教示操作盤の概要

xPad2 教示操作盤のボタンと機能を紹介します。



①	非常停止ボタン
②	タッチディスプレイ
③	物理ボタン
④	USB ポート
⑤	接続ケーブル（コントローラーまたはロボットとの接続に使用）
⑥	3 ポジションイネーブルスイッチ

5.3 産業用ロボットのシステム構成

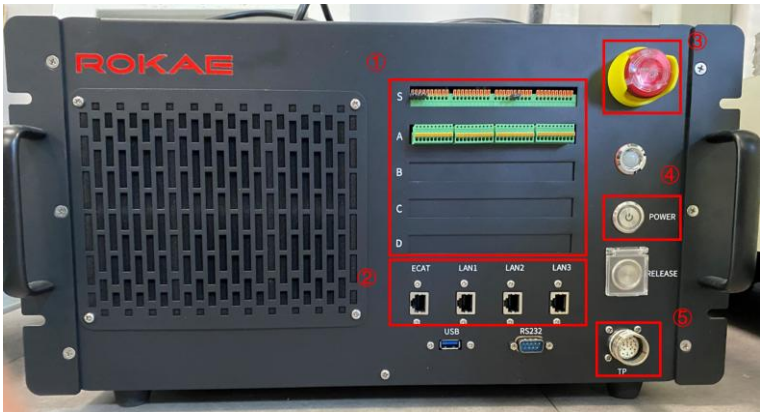
産業用ロボットの構成、配線、電源投入方法などを紹介します。ロボット本体および制御キャビネットはロボットの型番により異なります。詳細については、ロボット本体または制御キャビネットのマニュアルを参照してください。

産業用ロボットシステムの主な構成と配線関係を下図に示します。主にロボット本体、教示操作盤、制御キャビネット、ロボット接続ケーブル、電源ケーブルなどで構成されています。



5.3.1 XBC5 シリーズ制御キャビネットの概要

XBC5 シリーズ制御キャビネットには、XBC5、XBC5-E、XBC5-M の 3 つのモデルがあります。XBC5-M を例に、主な部品と機能を簡単に紹介します。他の型番も同様です。



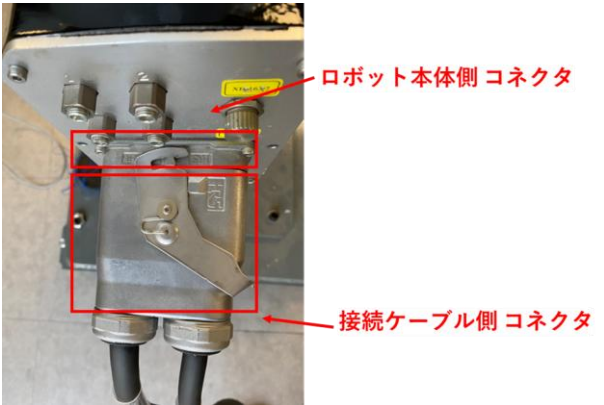
①	安全/汎用 IO 接続端子
②	ネットワークポート：デバッグ用、EtherCAT 拡張用、ビジョン用ポートなどがあります。
③	非常停止スイッチ：緊急時にモーターのブレーキを制御します。

④	電源スイッチ：ロボットの電源オン/オフを制御します。
⑤	教示操作盤接続口：xPad を接続します。

5.3.2 XBC5-M 制御キャビネットの配線、電源投入、起動

手順	図	説明
1. ロボット本体と制御キャビネットをロボット接続ケーブルで接続します。		ロボット接続ケーブル両端の形状は異なりますので、接続前に確認してください。
2. 図に示すように、教示操作盤と制御キャビネットを接続します。		
3. 電源ケーブルを制御キャビネットに接続します。		制御キャビネット側の電源接続口は、ロック式です。
4. 電源を投入します。		電源に接続し起動ボタンを押すと、xPad2 教示操作盤が自動的に起動します。

5.3.3 XBC5 制御キャビネットの配線、電源投入、起動

手順	図	説明
1. ロボット本体と制御キャビネットをロボット接続ケーブルで接続します。		制御キャビネット側の電源ケーブルとロボット接続ケーブルは、一体型ユニットです。
2. 図に示すように、教示操作盤と制御キャビネットを接続します。		
3. 電源を投入します。		電源に接続し起動ボタンを押すと、xPad2 教示操作盤が自動的に起動します。

5.3.4 XBC5-E 制御キャビネットの配線、電源投入、起動

手順	図	説明
1. ロボット本体と制御キャビネットをロボット接続ケーブルで接続します。		制御キャビネット側の電源ケーブルとロボット接続ケーブルは、一体型ユニットです。
2. 図に示すように、教示操作盤と制御キャビネットを接続します。		
3. 電源を投入します。		電源に接続し起動ボタンを押すと、xPad2 教示操作盤が自動的に起動します。

5.4 協働ロボットのシステム構成

5.4.1 ER 及び ER PRO

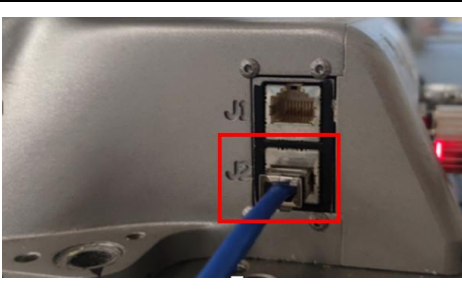
ER と ER PRO シリーズは制御キャビネットなしの設計を採用しています。システム構成は下図の通りです。

注意：ER シリーズロボットは xPad2 教示操作盤をサポートしていません。



ER シリーズロボットの場合は、次の手順に従って接続し電源を投入してください。

手順	図	説明
1. ロボット接続ケーブルでロボット本体と電源アダプターを接続します。		
2. 携帯型安全装置を接続します。		

3. 電源ケーブルを接続します		
4. HMI に接続します。		ER シリーズロボットは 教示操作盤の接続に対 応していません。PC 経 由で接続してくださ い。詳細については、 下記を参照してくださ い。
5. 電源に接続し、電源アダプ ターの電源を『オン』にし た後、ロボット本体の電源 を押します。		

5.4.2 CR 及び SR

CR と SR シリーズ協働ロボットは制御キャビネットなしの設計を採用しています。システム構成は下図の通りです。



CR シリーズロボットの場合は、次の手順に従って接続し電源を投入してください。SR シリーズも同様です。

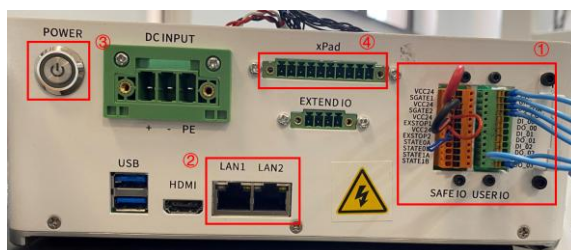
手順	図	説明
1. ロボット接続ケーブルでロボット本体と電源アダプターを接続します。		
2. 教示操作盤を接続します。		
3. 電源コードを接続します。		
4. 電源に接続し、電源アダプターの電源を『オン』にした後、ロボット本体の電源を押します。		ロボットが起動すると、教示操作盤が自動的に起動します。

5.4.3 CR-C 及び SR-C

CR-C と SR-C シリーズ協働ロボットは、制御キャビネット付きの設計を採用しています。システム構成は下図の通りです。

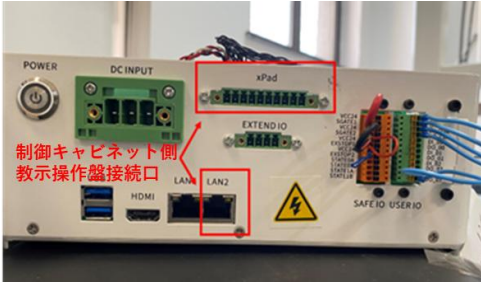
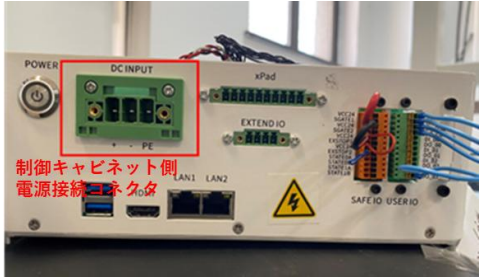
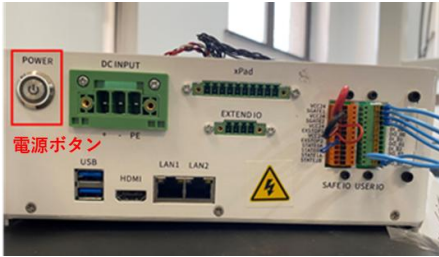


5.4.3.1 SR-C 制御キャビネットの配線、電源投入、起動



①	安全/汎用 IO 接続端子
②	ネットワークポート：デバッグ用、ビジョン用ポートなどがあります。
③	電源スイッチ：ロボットの電源オン/オフを制御します。
④	教示操作盤接続口：xPad を接続します。

手順	図	説明
1. ロボット本体と制御キャビネットをロボット接続ケーブルで接続します。	制御キャビネット側 接続ケーブル用コネクタ ロボット本体側 接続ケーブル	

2. 図に示すように、教示操作盤と制御キャビネットを接続します。		
3. 電源ケーブルを制御キャビネットに接続します。		
4. 電源を投入します。		電源に接続し起動ボタンを押すと、xPad2 教示操作盤が自動的に起動します。

5.4.3.2 CR-C 制御キャビネットの配線、電源投入、起動



①	安全/汎用 IO 接続端子
②	中継ケーブル口：ロボットと制御キャビネットを接続します。
③	教示操作盤接続口：xPad を接続します。

手順	図	説明
1. ロボット本体と制御キャビネットをロボット接続ケーブルで接続します。		注意： CR-C シリーズ制御キャビネットには、電源接続ケーブルと信号接続ケーブルがあります。
2. 図に示すように、教示操作盤と制御キャビネットを接続します。		
3. 電源ケーブルを制御キャビネットに接続します。		制御キャビネット側の電源接続口は、ロック式です。
4. 電源を投入します。		電源に接続し起動ボタンを押すと、xPad2 教示操作盤が自動的に起動します。

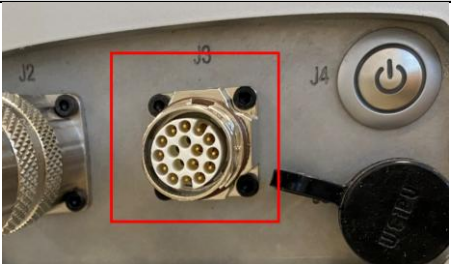
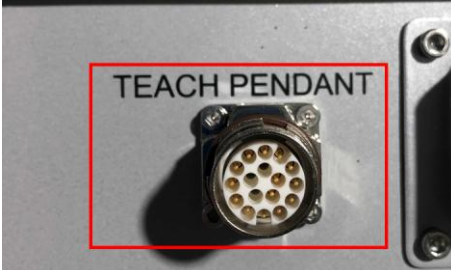
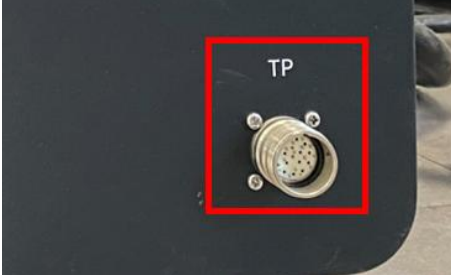
5.5 HMI とロボットの接続

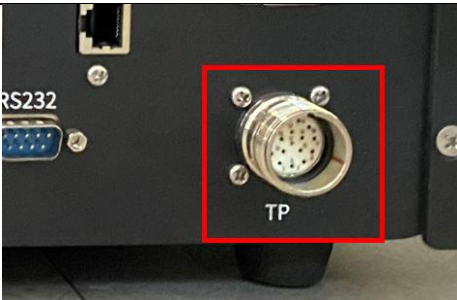
Robot Assist はロボットのホストコンピューターソフトウェアで、PC や xPad2 などで行えます。Robot Assist ソフトウェアが搭載されている機器をロボットと同じ LAN に接続し、ロボット検出やアドレスの手動入力などの方法により、ロボットとの接続を確立できます。

5.5.1 xPad2 とロボットの接続

xPad2 教示操作盤を使用する場合、教示操作盤のデフォルトのネットワークセグメントは 192.168.1.X です。教示操作盤の IP アドレスをロボット本体と同じネットワークセグメントに変更してから、xPad2 をロボットの対応するポートに接続する必要があります。

5.5.1.1 ハードウェア接続

機種	概要	図
CR	xMate CR シリーズロボットの xPad2 接続口は、ベースにあります。	
CR-C	xMate CR-C シリーズロボットの xPad2 接続口は、制御キャビネットの上部にあります。	
SR	xMate CR シリーズロボットの xPad2 接続口は、ベースにあります。	
SR-C	xMate SR-C シリーズロボットの xPad2 接続口は、SR 専用制御キャビネットにあります (別途アダプターが必要)。	
XBC5	XBC5 シリーズ制御キャビネットの xPad2 接続口は、制御キャビネットの下部にあります。	

XBC5-M	XBC5-M シリーズ制御キャビネットの xPad2 接続口は、制御キャビネットの下部にあります。	
XBC5-E	XBC5-E シリーズ制御キャビネットの xPad2 接続口は、制御キャビネットの下部にあります。	

5.5.1.2 接続設定

ハードウェア接続を完了しロボットが起動すると、xPad2 は自動的に起動して内蔵の Robot Assist ソフトウェアを実行します。

5.5.2 PC とロボットの接続

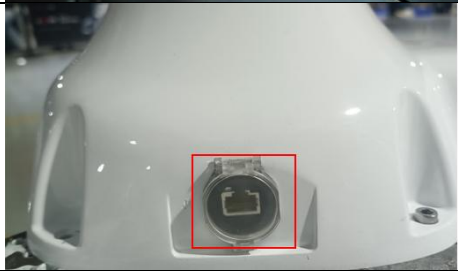
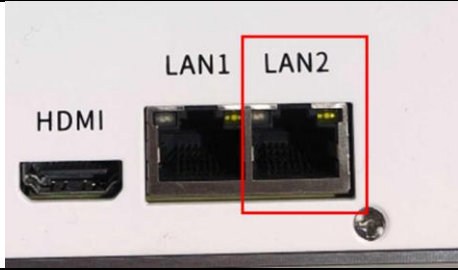
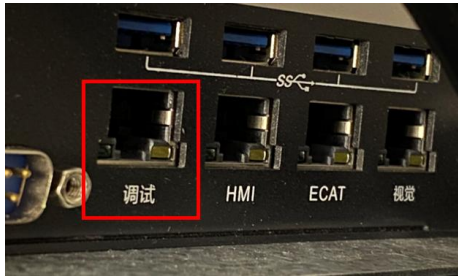
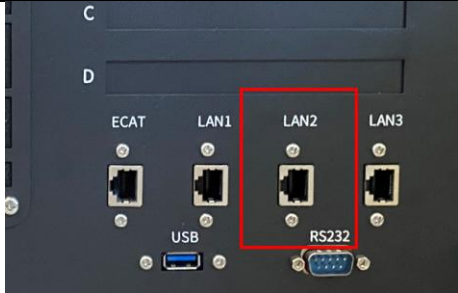
Robot Assist ソフトウェアは PC で実行でき、PC とロボットまたは制御キャビネットとの接続を確立します。

5.5.2.1 ハードウェア接続

5.5.2.1.1 HMI とロボットの 1 対 1 接続

RobotAssist を実行している PC を使用してロボットをデバッグする場合は、ネットワークケーブルでロボットシステムに直接接続できます。

機種	概要	図
ER/ER PRO	xMate ER シリーズ協働ロボットのベースには、2 つの Ethernet ポートがあります。J2 ポートの工場出荷時のデフォルトの固定 IP アドレスは 192.168.0.160 です。	
CR	xMate CR シリーズ協働ロボットのベースには、1 つの Ethernet ポート (J1) があります。工場出荷時のデフォルトの固定 IP アドレスは 192.168.2.160 です。	

CR-C	xMate CR-C には、制御キャビネットがあり、Ethernet ポートが装備されています。工場出荷時のデフォルトの固定 IP アドレスは 192.168.2.160 です。	
SR	xMate SR は、ベースに J2 ポートがあります。工場出荷時のデフォルトの固定 IP アドレスは 192.168.2.160 です。	
SR-C	xMate SR-C のデバッグポートは LAN2 で、デフォルトの IP アドレスは 192.168.0.160 です。	
XBC5/XBC5-E	制御キャビネットには、左から右に次の4つの Ethernet ポートがあります。 ● デフォルトで 192.168.0.160 の固定 IP アドレスに設定されている、デバッグポート。 ● オプションのバス拡張ポート ● スレーブ拡張用の EtherCAT デバイス拡張ポート。 ● デフォルトで 192.168.2.160 の固定 IP アドレスに設定されている、産業用カメラ接続用ビジョンポート。	
XBC5-M	XBC5-M 制御キャビネットのデバッグポートは LAN2 で、デフォルトの IP アドレスは 192.168.0.160 です。	

5.5.2.1.2 HMI とロボットの 1 対多接続

複数のロボットとの接続を切り替える必要がある場合は、複数のロボットを同じ LAN に接続し、RobotAssist を実行している PC が同じネットワークを介して接続可能なロボットを検出することができます。

5.5.2.1.3 無線接続

AGV 搭載など有線接続が不便な場合は、制御キャビネットの予備ポート（xMate 協働ロボットのベースのポート、産業用ロボット制御キャビネットのビジョン用/デバッグポート）を使用し、無線ルーターを接続した後、HMID に無線で接続することができます。

5.5.2.2 接続設定

5.5.2.2.1 ネットワークケーブルでの直接接続

ロボットのベースや制御キャビネットには、デバッグポートとして固定 IP アドレス 192.168.0.160 に設定されているポートがあります。このアドレスはすべてのロボットで同じであるため、変更しないことをお勧めします。RobotAssist を実行している PC をネットワークケーブルでこのポートに接続し、ロボットを制御することができます。

5.5.2.2.2 外部ネットワークポート接続

ロボットの外部ポート接続は、IP アドレスの自動取得と静的 IP アドレスの設定の 2 つの方法をサポートしています。

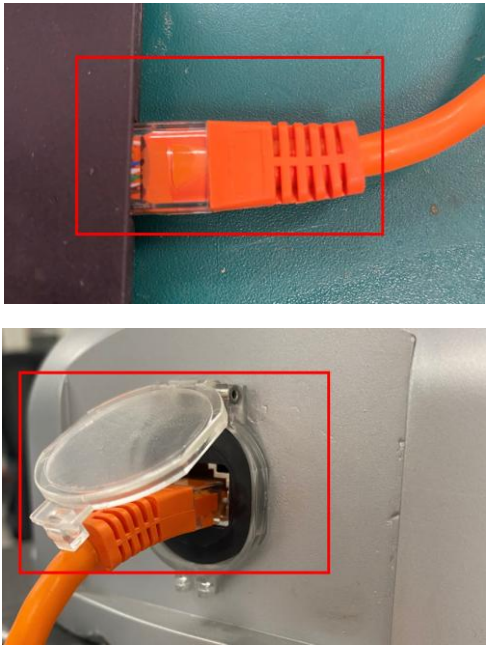
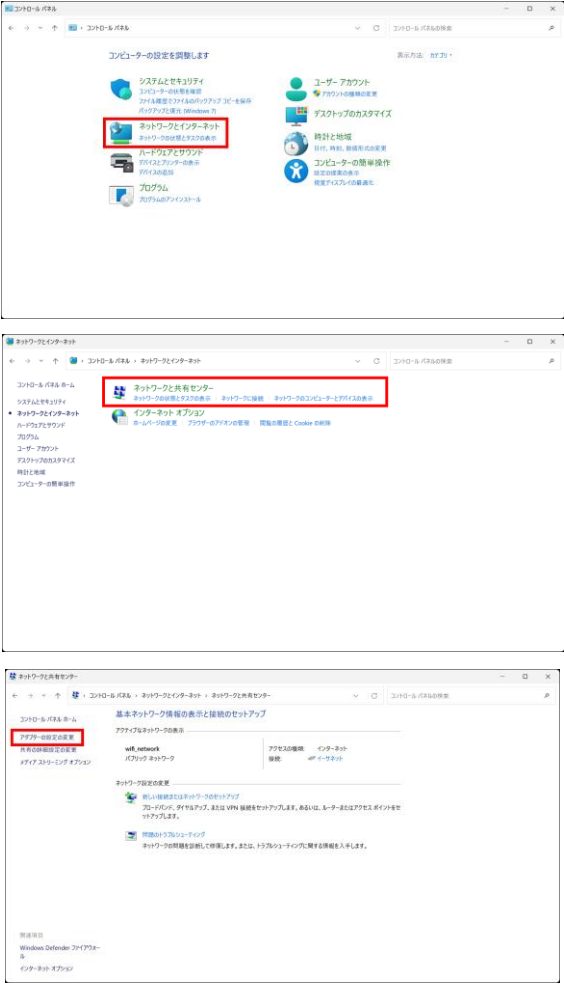
IP アドレスを自動的に取得する場合は、協働ロボットは J1 ポートを DHCP モードに設定し、産業用ロボットはビジョン用ポートを DHCP モードに設定します。ロボットはこのポートを介して DHCP 機能のルーターに接続され、ロボットに IP アドレスが自動的に割り当てられます。ロボット検出機能により、ロボットを検出して接続することができます。

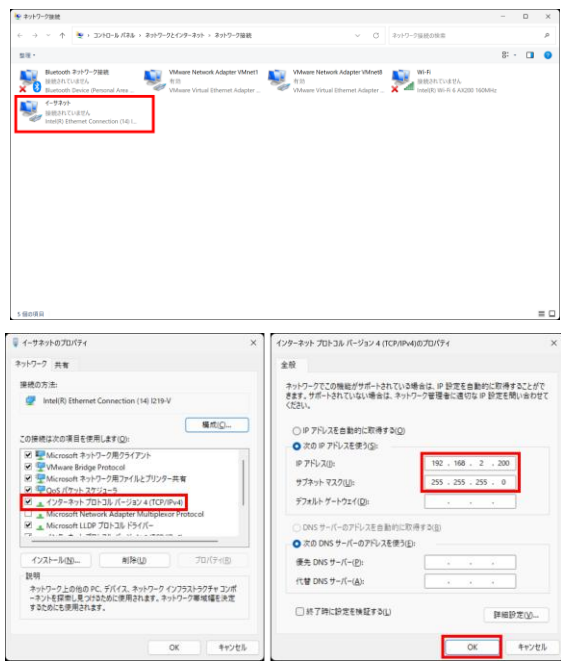
静的 IP アドレスを使用する場合は、協働ロボットは J1 ポートを任意の IP アドレスに設定し、産業用ロボットはビジョン用ポートを任意の IP アドレスに設定します。ロボットはこのポートを通してルーターに接続され、ロボットの IP アドレスを使用してロボットにアクセスおよび制御することができます。

5.5.2.2.3 PC 等の機器との有線接続

ロボットのベースや制御キャビネットには、デバッグポートとしてあらかじめ固定 IP アドレス 192.168.0.160 に設定されているポートがあります。このアドレスはすべてのロボットで同じであるため、変更しないことをお勧めします。RobotAssist を実行している PC をネットワークケーブルを使用してこのポートに接続し、ロボットを制御することができます。

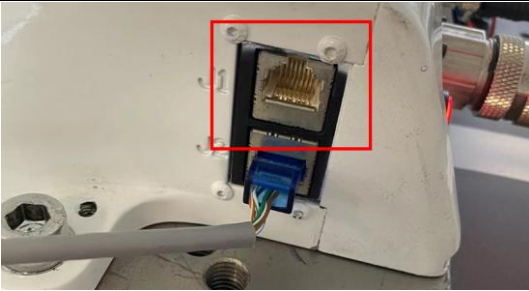
PC 等をロボットと接続する場合は、PC 等の LAN ポートのアドレスがロボットと同じネットワークにあることを確認する必要があります。PC（windows11）の静的 IP アドレスの変更方法とロボット（CR シリーズ）への接続については、以下の手順を参照してください。

手順	図	説明
1. ロボットとネットワークケーブルで接続します。ネットワークケーブルの一端を PC のポートに接続し、もう一端をロボットのポートに接続します。		CR シリーズロボットのベース側面にあるポートのデフォルトのネットワークセグメントは「192.168.2.XX」です。
2. ローカルの静的 IP アドレスを変更します。 PC の [コントロールパネル] → [ネットワークとインターネット] → [ネットワークと共有センター] → [アダプターの設定の変更] → 右クリックして対応するポートの[プロパティ]を開く → [インターネットプロトコルバージョン 4 (TCP/IPv4)]をダブルクリックする → 端末機器 (PC) の IP アドレス、サブネットマスク、デフォルトゲートウェイを変更して[OK]をクリックする。		PC の IP アドレスは、ロボットと同じネットワークセグメント内の空いている IP アドレスにします。サブネットマスクとデフォルトゲートウェイはロボットと同じにします。

		
3. HMI とロボットを接続します。		

5.5.2.2.4 PC 等の機器との無線接続

協働ロボットは J1 ポートを DHCP モードに設定し、産業用ロボットはビジョン用ポートを DHCP モードに設定することができます。ロボットはこのポートを介して DHCP 機能のルーターにアクセスし、ロボットに IP アドレスを自動的に割り当てます。ロボット検出機能によりロボットを検出して接続することができます。

手順	図	説明
1. ロボットシステムの IP プロパティを自動モードに変更します。		詳細は第 6 章を参照してください。
2. ロボットとルーターを接続します。		協働ロボットは J1 ポートを DHCP モードに設定し、産業用ロボットはビジョン用ポートを DHCP モードに設定することができます。ロボットはこのポートを介して DHCP 機能のルーターにアクセスし、ロボットに IP アドレス

		を自動的に割り当てます。
3. PC を同じネットワークセグメント内のルーターネットワークにアクセスさせ、IP 取得モードを DHCP に設定します。		[インターネットプロトコルバージョン 4 (TCP/IPv4)]ページに入る方法は、「IP アドレスの手動変更」の手順 2 を参照してください。
4. HMI とロボットを接続します。		

5.5.2.2.5 IP アドレスの変更について

Windows10 システムを例として、IP アドレスの変更方法を説明します。

ネットワークケーブルの一端をロボットの J2 ポートに接続し、もう一端を端末機器 (PC) に接続します。

端末機器 (PC) で「スタート > コントロールパネル」をクリックして「ネットワークと共有センター」を選択します。「ネットワークと共有センター」ウィンドウが表示されます。

「ネットワークと共有センター」で「ローカルエリア接続」をクリックして、「ローカルエリア接続の状態」を表示します。

「ローカルエリア接続の状態」で「プロパティ」をクリックして、「ローカルエリア接続のプロパティ」を表示します。

「ローカルエリア接続のプロパティ」で「インターネットプロトコルバージョン 4 (TCP/IPv4)」をダブルクリックして、「インターネットプロトコルバージョン 4 (TCP/IPv4) のプロパティ」を表示します。

「インターネットプロトコルバージョン 4 (TCP/IPv4) のプロパティ」で「次の IP アドレスを使う」を選択して、端末機器 (PC) の IP アドレス、サブネットマスク、デフォルトゲートウェイを変更して「OK」をクリックします。(端末機器 (PC) の IP アドレスは、ロボットと同じネットワークセグメント内の空いている IP アドレスに変更し、サブネットマスクとデフォルトゲートウェイはロボットの IP アドレスと同じにします。)



警告

ロボットのポートや IP アドレスを手動で変更する場合は、異なるポートを同じネットワークセグメント内の同じ静的 IP アドレスに設定しないでください。

デバッグポートのネットワークモードと IP アドレス（192.168.0.160）を変更しないでください。

教示操作盤 xPad のネットワークカードのネットワークモードと IP アドレス（192.168.1.160）も変更しないでください。

5.5.3 ロボットの検出と接続

HMI は、同じネットワークセグメント内の接続可能なすべてのロボットを検出して表示することができます。以下の手順でロボットを検出して表示することができます。

手順	図	説明
1. 使用可能なロボットを検索します。 下部のステータスバーのネットワークアイコンをタップすると、ロボット検索インターフェースに入ります。 [使用可能なロボットを検索]をタップして、利用可能なロボットを検出します。		ロボットを検索する際は、RobotAssist を実行している機器とロボットが同じネットワークにあり、ネットワークが接続されていることを確認してください。

2. ロボットに接続します。ロボットを検索で検出されたロボットの[接続する]ボタンをタップするか、ロボットの IP アドレスを入力し右側の[接続する]をタップします。	ロボットを検索 システムは、ロボット検索用IPアドレスの監視によりロボットを自動的に検出します。ネットワークカードのIPアドレスを変更する場合は、ここをクリックしてIPアドレスを変更できます。本操作は再起動後に有効になります。 ロボット検索用IPアドレス: 192.168.21.2  xMateCR12・2.2.1・192.168.21.9・シミュレーション 接続する ロボット接続 コントローラーサービス: 切断 アップグレードサービス: 切断 IPアドレス: 192.168.21.9 接続する ロボットを検索 システムは、ロボット検索用IPアドレスの監視によりロボットを自動的に検出します。ネットワークカードのIPアドレスを変更する場合は、ここをクリックしてIPアドレスを変更できます。本操作は再起動後に有効になります。 ロボット検索用IPアドレス: 192.168.21.2  xMateCR12・2.2.1・192.168.21.9・シミュレーション 切断する ロボット接続 コントローラーサービス: [192.168.21.9:5050]に接続 アップグレードサービス: [192.168.21.9:4567]に接続 IPアドレス: 192.168.21.9 切断する	ロボットが正常に接続されると、[コントローラーサービス]と[アップグレードサービス]に「XXXX に接続」と表示されます。 下部のステータスバーのアイコンは  になります。 複数の RobotAssist ソフトウェアの同時接続はサポートしていません。 現在の RobotAssist ソフトウェアの接続を切断するか、またはロボットを再起動すると、他の RobotAssist ソフトウェアを接続できます。
3. ロボットとの接続を切断します。 接続インターフェースで「切断」のボタンをタップして、RobotAssist ソフトウェアとコントローラーとの接続を切断できます。	ロボットを検索 システムは、ロボット検索用IPアドレスの監視によりロボットを自動的に検出します。ネットワークカードのIPアドレスを変更する場合は、ここをクリックしてIPアドレスを変更できます。本操作は再起動後に有効になります。 ロボット検索用IPアドレス: 192.168.21.2  xMateCR12・2.2.1・192.168.21.9・シミュレーション 切断する ロボット接続 コントローラーサービス: [192.168.21.9:5050]に接続 アップグレードサービス: [192.168.21.9:4567]に接続 IPアドレス: 192.168.21.9 切断する	RobotAssist ソフトウェアを再接続する手順は、接続プロセスと同じです。

6 HMI の概要

6.1 本章の概要

本章では、HMI ソフトウェアの基本的なレイアウト、主な機能と役割の概要を紹介します。実際にロボットを操作・使用する前に、本章をお読みください。

6.2 RobotAssist ソフトウェアの概要

RobotAssist は、xCore 制御システムのホストコンピューターソフトウェアで、ロボットの動作制御、タスク編集、パラメーター設定、状態監視などの機能を備えています。このソフトウェアは PC、タブレット、xPad2 教示操作盤に搭載可能です。ロボットと同じネットワークセグメント内でロボットに接続すると、ロボットを制御することができます。

xPad2 を使用しない場合は、タブレットまたは PC を操作端末として使用することを推奨します。推奨構成は以下の通りです。

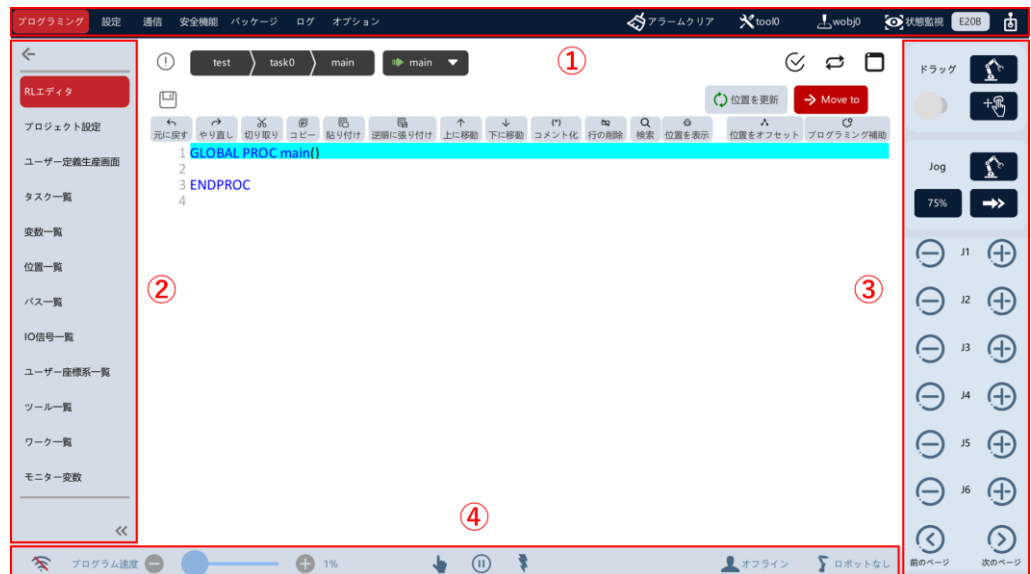
端末タイプ	タブレット	PC
ストレージ容量	32GB	
システムメモリ	4GB	
グラフィックボード	—	Intel HD Graphics 4000 以上
画面サイズ	8.0 インチ以上	—
ネットワーク通信	Wi-Fi	Wi-Fi または有線ネットワークカード
オペレーティングシステム	Windows7 64bit , Windows10 64bit, Ubuntu20.04	
プロセッサー	Intel Core i3 以上	



ヒント

RobotAssist ソフトウェアはリアルタイムでコントローラーと連動しています。PC 上でウィンドウサイズを頻繁に変更すると、監視データが更新されなくなることがあります。その場合は **Alt+Tab** を押してウィンドウを切り替えてください。

6.3 HMI 全体のレイアウト



①	上部ステータスバー
②	左サイドバー
③	右操作パネル
④	下部ステータスバー

6.3.1 上部ステータスバー

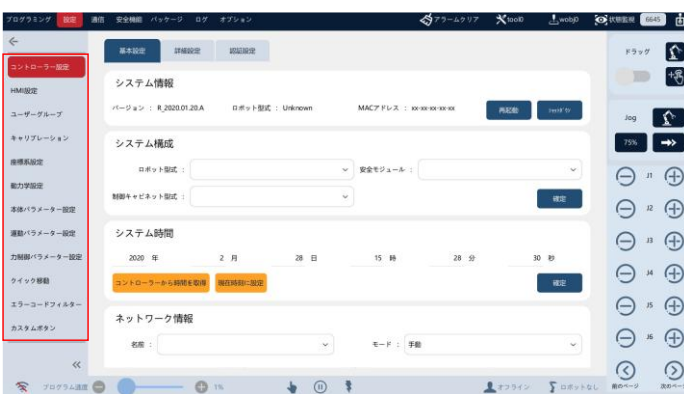
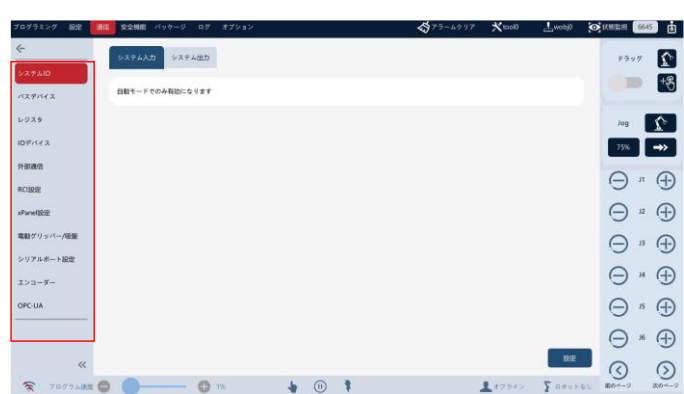
上部ステータスバーには、メインメニュー、インスタントログ、アラームクリアボタン、ツール・ワーク選択ボタン、状態監視ボタン、RSC リセットボタン、安全チェックサムボタン、操作パネルボタンがあります。

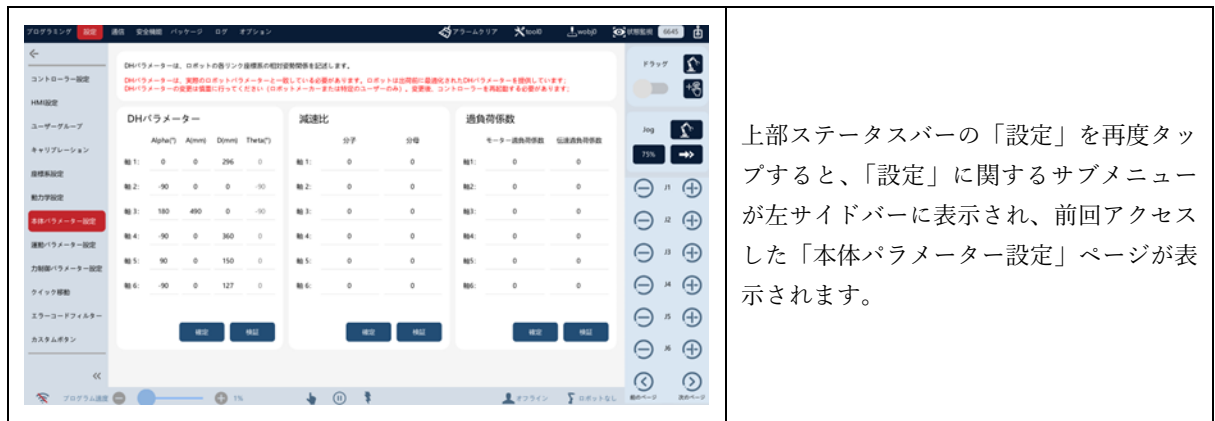
	①	②	③	④	⑤	⑥	⑦	⑧	⑨
①	メインメニュー： プログラミング、設定、通信、安全機能、パッケージ、ログ、オプションなどがあります。タップすると、各メニューのインターフェースに移動します。								
②	システムの最新のコントローラーログ情報を表示します。タップすると、コントローラーログのインターフェースに移動します。								
③	アラームクリアボタン： タップすると、インスタントログとコントローラーのアラームメッセージがクリアされます。								
④	ツール選択ボタン： 設定されているツールの情報を表示します。タップするとツールを選択できます。								
⑤	ワーク選択ボタン： 設定されているワークの情報を表示します。タップするとワークを選択できます。								
⑥	状態監視ボタン： タップすると、状態監視パネルを開閉します。								
⑦	非常停止リセットボタン（安全コントローラー搭載のロボットでのみ利用可能）：								

	ロボットが非常停止、セーフティゲート、安全停止状態にある場合は、上記状態になった要因を排除した後、このボタンをタップするとロボットの状態をリセットできます。
⑧	安全チェックサムボタン： 軸制限、ロボット制限、バーチャルフェンス、衝突検知、安全エリア、ツール設定、安全位置の内容を確認できます。また、これらの設定を変更した際は、確認のためこのページが表示されます。
⑨	操作パネルボタン： タップすると右操作パネルを開閉します。

6.3.2 左サイドバー

上部ステータスバーで機能（プログラミング、設定、通信など）を切り替えると、該当機能のサブメニューが左サイドバーに表示されます。

	上部ステータスバーの「設定」をタップすると、「設定」に関するサブメニューが左サイドバーに表示されます。
	「本体パラメーター設定」をタップすると、本体パラメーターを設定するページに入ります。
	上部ステータスバーの「通信」をタップすると、「通信」に関するサブメニューが左サイドバーに表示されます。

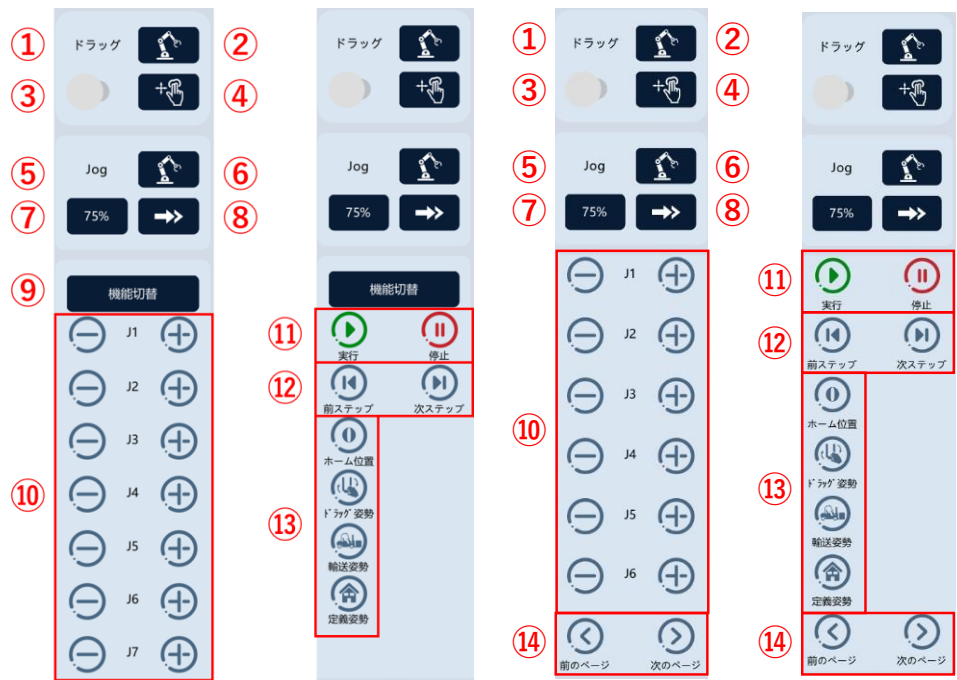


上部ステータスバーの「設定」を再度タップすると、「設定」に関するサブメニューが左サイドバーに表示され、前回アクセスした「本体パラメーター設定」ページが表示されます。

6.3.3 右操作パネル

上部ステータスバーの または ボタンをタップすると、右操作パネルが開きます。制御モードの切替、ロボットの動作制御、姿勢教示などを行う事ができます。

ロボットは、ジョグモードとドラッグモード（協働ロボットのみ）の 2 つのティーチング方法をサポートしています。ジョグモードでは、ジョグボタンを使用してロボットの動作を制御します。ドラッグモードでは、ロボット先端にあるハンドルをドラッグすることで、ロボットを手動で直接ガイドします。



7 軸ロボットの操作パネル

6 軸以下のロボットの操作パネル

①	ドラッグ設定エリア： ②、③、④がドラッグに関連することを示しています。
②	ドラッグモード設定： 軸動作ドラッグ、直交動作ドラッグを切り替えます。
③	ドラッグオン/オフスイッチ： ドラッグモードのオンとオフを切り替えます。

④	ドラッグ動作の設定： 軸動作ドラッグは自由ドラッグのみをサポートしています。 直交動作ドラッグは直線移動のみ、回転のみ、自由ドラッグの 3 種類をサポートしています。
⑤	ジョグ設定エリア： ⑥、⑦、⑧がジョグに関連することを示しています。
⑥	ジョグ時の座標系の設定： ジョグ時の軸動作及び直交動作での基準座標系（ワールド座標系、ベース座標系、フランジ座標系、ツール座標系、ワーク座標系）を選択します。
⑦	ジョグ速度の設定： ジョグ時の動作速度を調整するために使用します。調整範囲は 1～100% で、ジョグの最大速度制限（250mm/s）との割合です。
⑧	ジョグステッピングモードの設定： 連続ジョグとステップジョグを選択します。ステップの大きさも選択できます。
⑨、⑭	機能の切り替え： ジョグボタンと⑪、⑫、⑬ボタンを切り替えます。
⑩	ジョグボタン： 7 軸ロボットの場合は、軸動作時は J1～J7、直交動作時は X/Y/Z/A/B/C および Elbow が表示されます。 6 軸ロボットの場合は、軸動作時は J1～J6、直交動作時は X/Y/Z/A/B/C が表示されます。
⑪	プログラム実行・停止ボタン： プログラムの実行と停止を行います。
⑫	プログラムをステップ実行する際に使用します。「前ステップ」と「次ステップ」が実行できます。
⑬	カスタムボタンエリア： 「設定」 - 「カスタムボタン」でこのエリアにショートカット機能を設定できます。デフォルトでは、ホーム位置、ドラッグ姿勢、輸送姿勢、定義姿勢の 4 つのボタンがあります。 姿勢の詳細については、「設定」 - 「クイック移動」の章を参照してください。



ヒント

ジョグ及びドラッグをオンにする前に、現在のロボットの動作モードが手動モードかつモーターの電源がオフの状態であることを確認してください。

一時停止機能に割り当てられているレジスタまたはシステム IO がリセットされていない場合は、プログラムの起動ができません。

6.3.4 下部ステータスバー

下部ステータスバーには、RobotAssist ソフトウェアとロボットとの接続状態、プログラムの実行速度、ロボットの動作モード、ロボットの状態、ロボットのモーターの状態、ユーザーログイン情報、ロボットの型番情報が表示されます。



①	RobotAssist ソフトウェアとロボットとの接続状態： このボタンをタップすると、ロボットとの接続設定ページが表示されます。アイコン は未接続、 は接続済み、 はロボットへの接続試行中、 はアップグレードサービスが接続済みだがコントローラーが未接続の状態を表します。 の状態では制御システムのアップグレード操作は実行できますが、ロボットの操作やロボットパラメーターの設定はできません。												
②	プログラムの実行速度： RL プログラムを実行する際の動作速度を調整します（調整範囲は 1%~100%）。このパラメーターは手動モードと自動モードのプログラム実行速度で独立しています。スライダーまたは「-/+' ボタンをタップして、プログラム速度を微調整（-/+'1%）できます。 注意：プログラム速度の上限は、「コントローラー設定」－「詳細設定」の「手動モード時のプログラム速度上限」、「自動モード時のプログラム初速度の上限」の影響を受ける場合があります。詳しくは関連内容の章を参照してください。												
③	動作モード： ロボットの動作モードは、手動モードと自動モードに分けられます。詳しくは、「制御システムの基本操作」の章を参照してください。 は手動モードを表します。通常このモードでプログラミングとデバッグを行います。 は自動モードを表します。通常このモードで自動連続生産を行います。												
④	ロボットの状態： ロボットの状態には、以下の状態があります。 <table border="1"> <tr> <td></td> <td>：アイドル状態 プログラムが停止しており、システムが動作していない状態です。</td> </tr> <tr> <td></td> <td>：プログラム実行中 プログラムが実行中で、システムが動作している状態です。ロボットが動作すると赤色になります。</td> </tr> <tr> <td></td> <td>：ドラッグモード ロボットをドラッグで操作できます。ロボットが動作すると赤色になります。</td> </tr> <tr> <td></td> <td>：Demo モード デモ動作を開始できます。ロボットが動作すると赤色になります。</td> </tr> <tr> <td></td> <td>：推定モード 推定モードであることを表しています。ロボットが動作すると赤色になります。</td> </tr> <tr> <td></td> <td>：ジョグモード ジョグモードであることを表しています。ロボットが動作すると赤色になります。</td> </tr> </table>		：アイドル状態 プログラムが停止しており、システムが動作していない状態です。		：プログラム実行中 プログラムが実行中で、システムが動作している状態です。ロボットが動作すると赤色になります。		：ドラッグモード ロボットをドラッグで操作できます。ロボットが動作すると赤色になります。		：Demo モード デモ動作を開始できます。ロボットが動作すると赤色になります。		：推定モード 推定モードであることを表しています。ロボットが動作すると赤色になります。		：ジョグモード ジョグモードであることを表しています。ロボットが動作すると赤色になります。
	：アイドル状態 プログラムが停止しており、システムが動作していない状態です。												
	：プログラム実行中 プログラムが実行中で、システムが動作している状態です。ロボットが動作すると赤色になります。												
	：ドラッグモード ロボットをドラッグで操作できます。ロボットが動作すると赤色になります。												
	：Demo モード デモ動作を開始できます。ロボットが動作すると赤色になります。												
	：推定モード 推定モードであることを表しています。ロボットが動作すると赤色になります。												
	：ジョグモード ジョグモードであることを表しています。ロボットが動作すると赤色になります。												

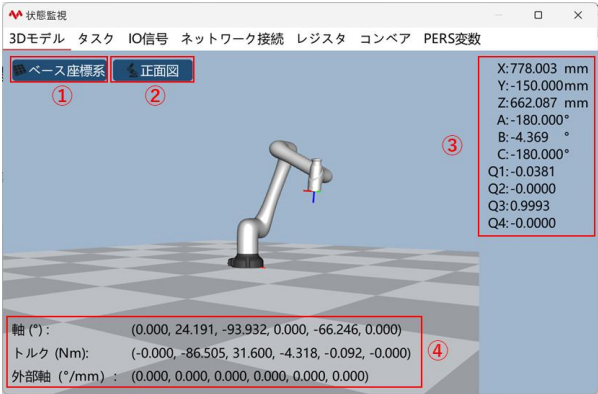
	 RCI : RCI モード RCI モードであることを表しています。ロボットが動作すると赤色になります。
	 ! : 協働モード 協働モードであることを表しています。他の状態と組み合わせてアイコンの右上隅に表示されます。
	 — : エラー状態 ロボットシステムにエラーがあることを表しています。
	 ! : デバッグモード デバッグモードであることを表しています。ロボットが動作すると赤色になります。
⑤	 \$: 自動実行タスク実行中 自動実行タスクが実行状態の場合は、このボタンが表示されます。それ以外の場合は表示されません。このボタンをタップすると、自動実行タスクを停止できます。
⑥	 ⚡ : モーター電源オン状態 ロボットのモーターが電源オンの状態です。タップするとモーターの電源がオフになります。
	 ⚡ : モーター電源オフ状態 ロボットのモーターが電源オフの状態です。タップするとモーターの電源がオンになります。
	 E : 非常停止状態 ロボットが非常停止の状態です。この状態の時は、モーターの電源をオンにできません。
	 G : セーフティーゲート状態 ロボットがセーフティーゲートの状態です。この状態の時は、モーターの電源をオンにできません。CR シリーズ及び SR シリーズの場合は、停止エリア入力状態に該当します。
	 S : 安全停止状態（安全コントローラー搭載の型番のみ）： ロボットが安全停止の状態です。安全コントローラーが動作や通信の異常を検出したか、パラメーターが安全コントローラーに設定された安全しきい値を超えたことを示しています。この状態の時は、モーターの電源をオンにできません。
⑦	ユーザーログイン情報： オペレーター、教示者、プログラマー、管理者、スーパー管理者を表示します。ボタンをタップすると、ユーザーログイン画面に移動します。
⑧	ロボット型番： ロボットの型番が表示されます。

6.4 状態監視

上部ステータスバーの「状態監視」ボタンをタップすると、状態監視ウィンドウが開きます。状態監視ウィンドウでは、ロボットの 3D モデル、タスクの実行状態、IO 信号、ネットワーク接続状態、レジスタ、コンベア状態、変数モニターなどを監視でき、ロボットの状態を素早く把握することができます。

6.4.1 3D モデル

このインターフェースは、ロボットの現在の状態を 3D モデルで直感的に表示します。タップとドラッグで視点を切り替えることができ、「正面を向ける」ボタンをタップするとデフォルトの視点に戻すことができます。



①	座標系切り替え： ベース座標系、ワールド座標系、ワーク座標系が選択可能です。ロボットが床置き設置ではない、またはロボットグループを使用する場合は、ベース座標系とワールド座標系が一致しないことがあります。
②	正面を向ける： ボタンをタップすると、モデルをデフォルトの視点に戻すことができます。
③	先端の位置/姿勢： 座標系（ワーク座標系、ベース座標系、ワールド座標系）に対するロボット先端の位置と姿勢（オイラー角またはクォータニオン）を表示します。
④	軸・トルク・外部軸： 関節角度、関節トルク、外部軸情報を表示します。



ヒント

3D モデル表示をサポートしているのは PC 上の RobotAssist ソフトウェアのみです。
xPad2 では、状態データのみを表示できます。

6.4.2 タスク

このインターフェースは、現在のプロジェクト内のタスク名、タスクタイプ、実行状況を表示します。



6.4.3 IO 信号

汎用 IO 及び安全 IO の状態を表示します。



①	フィルタ表示設定： IO をフィルタ表示します。種類、IO ボード、信号タイプ、名前などのフィルタ条件を選択できます。「リセット」ボタンをタップすると、フィルタ条件をリセットできます。
②	IO 信号一覧： IO 信号の一覧が表示されます。
③	IO シミュレーションモード： オンにすると、DI 信号値をシミュレーションできます。

手順	図	説明
1. IO 信号ページに入り、[IO シミュレーションモード]のボタンをタップして、シミュレーションモードをオンにします。		
2. 対応する DI または DO の「値の変更」ボタンをタップすると、シミュレーションできます。		シミュレーションモードでない場合も、DO は強制的に出力することができます。

3.
[IO シミュレーションモード]
のボタンをタップして、シ
ミュレーションモードをオ
フにします。



実際の値と値の変更は
関係がありません。シ
ミュレーションモード
を終了すると、値の変
更はすべて **False** にな
ります。

安全 IO の信号名と対応する機能は以下の表を参照してください。

安全 IO	対応機能
S-DI0	非常停止状態
S-DI1	モーター電源オン状態
S-DI2	停止エリア入力
S-DI3	外部安全装置入力
S-DI4	外部非常停止入力
S-DI5	協働エリア入力
S-DI6	リセット
S-DI7	無
S-DO0	非常停止出力
S-DO1	停止状態出力
S-DO2	安全ホーム位置出力
S-DO3	無
S-DO4	STO 状態

6.4.4 ネットワーク接続

このインターフェースは、名前、種類、IP アドレス、ポート番号、状態など、コントローラ
ーに接続されているネットワークの情報を表示します。

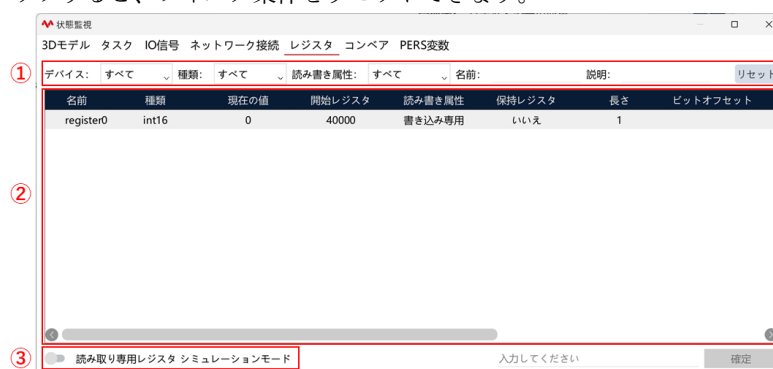


①	名前： MODBUS、RCI、SYS_SOCKET はシステムのデフォルト固有名です。ユーザーが 新規作成した名前は、その定義した名前で表示されます。
②	種類： MODBUS、RCI、SOCKET などのタイプの接続を表示します。対応する接続は関 連するインターフェースで追加または設定できます。SYS_SOCKET は外部通信に

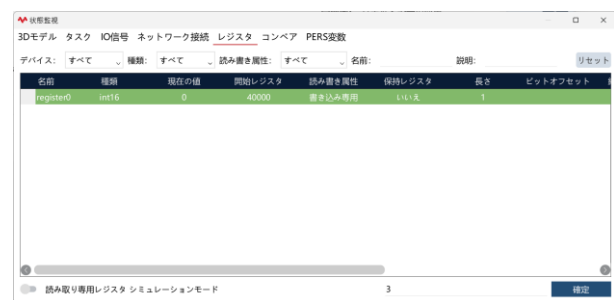
	対応する接続を指します。
③	IP アドレス： クライアント側の接続の場合はターゲットサーバーの IP アドレスが表示され、サーバー側の場合は自身の IP アドレスが表示されます。
④	ポート番号： クライアント側の接続の場合はターゲットサーバーのポート番号が表示され、サーバー側の場合は自身のポート番号が表示されます。
⑤	状態： 「接続済み」、「未接続」、「接続中」の 3 種類があります。サーバー側の接続の場合は、未接続時は「接続待機中」と表示されます。

6.4.5 レジスタ

このインターフェースは、各レジスタの情報を表示します。フィルタ表示ができ、選択可能なフィルタ条件には、デバイス、種類、読み書き属性、名前、説明があります。「リセット」ボタンをタップすると、フィルタ条件をリセットできます。



①	フィルタ表示設定： レジスタをフィルタ表示します。デバイス、種類、読み書き属性、名前、説明などのフィルタ条件を選択できます。「リセット」ボタンをタップすると、フィルタ条件をリセットできます。
②	レジスタ一覧： レジスタの一覧が表示されます
③	読み取り専用レジスタシミュレーションモード： レジスタシミュレーションモードをオンにすると、レジスタ値をシミュレーションできます。

手順	図	説明
書き込み専用レジスタの割り当て： 書き込み専用レジスタを選択し、「確定」ボタンの左側の入力ボックスに変更後の値を入力し「確定」ボタンをタップすると、このレジスタの現在値を変更できます。		- オペレーター権限では書き込むことができません； - 書き込み専用レジスタで機能コードに割り当てられていないレジスタに対してのみ、値を割り当てることができます。

読み取り専用レジスタの割り当て：

「読み取り専用レジスタシミュレーションモード」をオンにします。「確定」ボタンの左側の入力ボックスに変更後の値を入力し「確定」ボタンをタップすると、このレジスタの現在値を変更できます。

6.4.6 コンペア

コンペアパッケージと組み合わせ、コンペアの状態を監視します。詳細については、コンペアアトラッキング・パッケージを参照してください。



6.4.7 変数モニター

この画面には、変数モニターに設定した変数のリアルタイム情報が表示されます。

① 型: 変数名: リセット

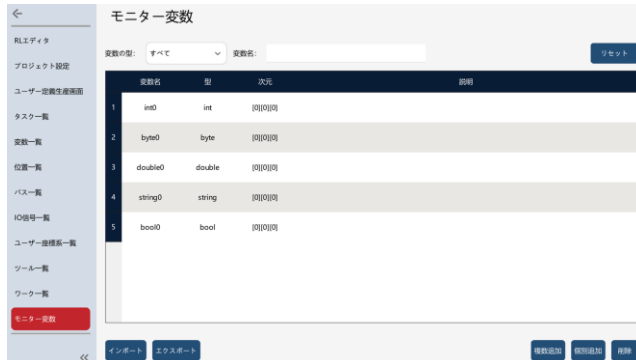
変数名	型	現在の値	説明
int0	int	120	
byte0	byte	0	
double0	double	0	
string0	string		
bool0	bool	FALSE	

②

③ 前のページ 1 / 1 次のページ

④ 書き込み

①	フィルタ表示設定： 型と変数名を使用してフィルタリングできます。 注意：対応する型は、bool、int、double、byte、string の 5 種類で、変数と配列を確認できます。
②	モニターする変数一覧
③	1 ページ当たり最大 10 個の変数を表示できます
④	入力ボックスに変更後の値を入力し、「書き込み」ボタンをタップすると、値を変更できます。

手順	図	説明
1. 変数モニター画面で、「複数追加」ボタンまたは「個別追加」ボタンをタップして、モニターする変数を追加します。		変数モニターで追加された変数のみを監視します。
2. 書き込み：「値を変更」列で変更したい変数の値を入力し、「書き込み」ボタンをクリックすると、その値がリアルタイムで変数に書き込まれます。		オペレーター権限では書き込みできません。変更する変数が PERS 変数ではない場合、 pptomain を実行すると、変更された値が初期値に戻ります。

注意：新しく変数一を追加した後、変数データの読み取りおよび変更を行うためには、まず **pptomain** 操作を実行する必要があります。

6.5 プログラミングメニューの一覧

RL エディタ	主に、RL プログラムの作成とデバッグに使用します。
プロジェクト設定	プロジェクトの新規作成、読み込み、インポート/エクスポート、プッシュなどの操作に使用します。
ユーザー定義生産画面	ウィジェットをカスタマイズして、生産画面を作成します。レジスタ、DI/DO 信号、プロジェクト変数を監視および編集できます。
タスク一覧	タスクの表示、新規作成、編集、インポート/エクスポートなどに使用します。
変数一覧	変数の表示、新規作成、編集、インポート/エクスポートなどに使用します。
位置一覧	位置の表示、新規作成、編集、インポート/エクスポートなどに使用します。
パス一覧	パスの表示、新規作成、編集、インポート/エクスポートなどに使用します。
IO 信号一覧	IO 信号の表示、新規作成、編集、インポート/エクスポートなどに使用します。
ユーザー座標系一覧	ユーザー座標系の表示、新規作成、編集、インポート/エクスポートなどに使用します。
ツール一覧	ツールの表示、新規作成、編集、インポート/エクスポートなどに使用します。
ワーク一覧	ワークの表示、新規作成、編集、インポート/エクスポートなど

	に使用します。
変数モニター	状態監視でモニターする変数のインポート/エクスポート、複数追加、個別追加、削除などに使用します。

6.6 設定メニューの一覧

コントローラー設定	ロボットタイプ、システム時間、IP アドレスなどのコントローラーシステム関連の設定を行います。
HMI 設定	HMI に関する設定、言語切り替え、教示操作盤の IP アドレスの設定などを行います。
ユーザーグループ	ログインやパスワードの変更など、ユーザー管理の設定を行います。
キャリブレーション	ロボットのゼロキャリブレーション、力センサーキャリブレーション、ソフトウェアキャリブレーション、ベース座標系キャリブレーションを実行します。
座標系設定	グローバルツール、グローバルワーク、グローバルユーザー座標系の設定を行います。
動力学設定	ロボットの動力学関連の設定を行います。
本体パラメーター設定	DH パラメーター、減速比、過負荷係数などの設定を行います。
運動パラメーター設定	ロボットの加減速性能、安全制御などの設定を行います。
力制御パラメーター設定	ロボットの力制御関連の設定を行います。
クイック移動	クイック移動の設定を行います。
電子銘版	電子銘板関連の設定を行います。(一部機種のみ対応)
エラーコードフィルター	エラーコードフィルタリング関連の設定を行います。
カスタムボタン	右側操作パネルのショートカットエリアの設定を行います。

6.7 通信メニューの一覧

システム IO	システム入力/出力信号の設定を行います。
バスデバイス	各種バス拡張モジュールの設定を行います。
レジスタ	レジスタ関連の設定を行います。
IO デバイス	IO デバイスの設定を行います。
外部通信	Socket ベースの外部通信インターフェースの設定を行います。
先端ツール	DH グリッパーの設定を行います。
RCI 設定	RCI 通信の設定を行います。
xPanel 設定	CR ロボットの xPanel 設定を行います。
電動グリッパー/吸盤	各種電動グリッパーや吸盤の設定とテストを行います。
シリアルポート設定	シリアルポート関連の設定を行います。
エンコーダー	コンベアトラッキング機能に必要なエンコーダーの設定を行います。
OPC-UA	OPC-UA 関連の設定を行います。

6.8 安全機能メニューの一覧

軸制限	ロボットの軸の制限を設定します。
ロボット制限	ロボットの速度、協働モード速度、出力、運動量などの制限を設定します。
バーチャルフェンス	直交ドラッグ時に作業範囲を制限するためのバーチャルフェンス機能の設定を行います。
衝突検知	衝突検知関連のパラメーターを設定します。
安全エリア	安全エリア関連の設定を行います。
安全位置	安全位置関連の設定を行います。

6.9 パッケージメニューの一覧

パレタイジング、トレイ、コンベアトラッキング、ソーラーバッテリー部品組み立てなどのパッケージをサポートしています。詳細については、該当する章を参照してください。

6.10 ログメニューの一覧

HMI ログ	HMI のログ情報を表示します。
コントローラーログ	現在接続されているロボットコントローラーのログを表示します。
実行ログ	Jog 及びプログラム動作に関するログを表示します。
ログタイムライン	タイムラインによりログ履歴を直感的に表示します。
内部ログ	教示操作盤や RobotAssist の内部ログ情報を表示します。
診断設定	問題の診断を支援します。
ハードウェアの状態	産業用コンピューターおよび教示操作盤の状態を表示します。

6.11 オプションの一覧

接続	RobotAssist ソフトウェアとコントローラーとの接続に関連する操作と設定を行います。
ROKAE について	バージョン情報、会社紹介などを表示します。
ソフトウェア更新	制御システムソフトウェアのアップグレード、バックアップ関連操作を行います。
エクスポート	制御システム構成のエクスポートを行います。
インポート	制御システム構成のインポートを行います。
ファイルマネージャー	RobotAssist ソフトウェアのフォルダーに関連する情報を表示します。
デモンストレーション	一部の機能のデモンストレーションを実行できます。

7 制御システムの基本操作

7.1 本章の概要

本章では、ロボットの一般的で基本的な必須操作を説明します。
実際にロボットを操作・使用する前に、本章をお読みください。

7.2 動作モード

ロボットの動作モードには、手動モードと自動モードがあります。

7.2.1 手動モード

手動モードは、主にロボットのプログラミングやデバッグに使用します。手動モードでは、ロボットのすべての動作はユーザーによって手動で制御されます。ロボットは、動作が有効化された時（3 ポジションイネーブルスイッチが中間位置にあるとき）にのみモーターに電源を投入し、動作命令に応答します。

手動モードは通常、以下のタスクに使用されます。

- 非常停止後、プログラムを続行するためにロボットをパス付近にジョグする；
- RL プログラムの作成；
- RL プログラムのデバッグ（起動、停止、シングルステップ動作、プログラム位置更新などを含みますが、それらに限りません）；
- 制御システムのパラメーター設定、座標系の設定など；
- 変数の表示と変更；

7.2.2 自動モード

自動モードは連続的な自動化生産に使用されます。3 ポジションイネーブルスイッチはバイパスされ、ロボットは人間の介入なしに正常に動作できます。自動モードでは、システム IO 信号などを通じてロボットを制御したり、その動作状態を取得できます。

自動モードは通常、以下のタスクに使用されます。

- RL プログラムのロード、起動及び停止；
- 非常停止後、プログラムされたパスへの復帰；
- システムのバックアップ；
- ツールの洗浄（プロセスに応じる）；
- ワークの加工と処理；

自動モードでの使用には以下の制限がありますが、これらに限定されるものではありません。

- ジョグ操作はできません
- 設定ファイル・IO ボードの枚数・ロボットの設置方法の変更はできません。
- バックアップの復元はできません。
- ライセンス認証は許可されません。
- ソフトウェアリミットの設定はできません。
- IO の新規作成、変更、削除はできません。
- パラメーター推定はできません。
- 協働モードのオン/オフはできません。
- ドラッグ教示のオン/オフはできません。

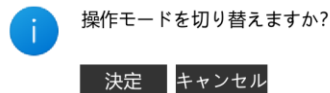
- キャリブレーションはできません。
- 変数の新規作成はできません。
- アップグレード/工場出荷時の設定への復元はできません。

7.2.3 モードの確認と切り替え

HMI ソフトウェアの下部ステータスバーの動作モードのアイコンで、制御システムの現在の動作モードを知ることができます。

	手動モード
	自動モード

HMI インターフェース上の動作モードのアイコンをタップすることで、動作モードの切り替え操作を行うことができます。切り替える際には、ポップアップウィンドウが出ます。「決定」をタップすると、動作モードが切り替わります。「キャンセル」をタップすると、動作モードは切り替わりません。

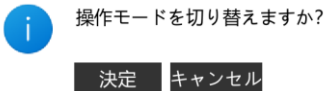


注意：安全確保のため、モードが切り替わる際にモーター電源が遮断されます。システムが RL プログラムを実行中でロボットが動作している場合は、システムは **STOP 1** をトリガーして停止します。

7.2.3.1 手動モードから自動モードへの切替

プログラムの全状態の検証を行う場合、またはプログラムが正式な生産準備が整った場合に、自動モードに切り替えます。

手動モードから自動モードへの切替は、以下の方法で行うことができます。

方法 1	HMI 上のボタン： HMI インターフェース上の動作モードのアイコン  をタップすることで、動作モードの切り替え操作を行うことができます。切り替える際には、ポップアップウィンドウが出ます。「決定」をタップすると、動作モードが切り替わります。「キャンセル」をタップすると、動作モードは切り替わりません。 
	方法 2 システム IO：「Switch Auto」
	方法 3 外部通信：「switch_mode:auto」+「\r」
	方法 4 レジスタ機能コード：ctrl_switch_operation_auto_manual
	方法 5 SDK

7.2.3.2 自動モードから手動モードへの切替

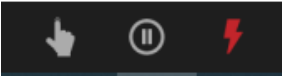
方法 1	HMI 上のボタン： HMI インターフェース上の動作モードのアイコンをタップすることで、動作モードの切り替え操作を行うことができます。切り替える際には、ポップアップウィンドウが出ます。「決定」をクリックすると、動作モードが切り替わります。「キャンセル」をクリックすると、動作モードは切り替わりません。  操作モードを切り替えますか？ 
方法 2	システム IO：「手動モード」
方法 3	外部通信：「switch_mode>manual”+“\r」
方法 4	レジスタ機能コード：ctrl_switch_operation_auto_manual
方法 5	SDK

7.3 モーター電源のオン/オフ

まず第 2 章-安全装置にて、イネーブル装置の紹介をお読みください。

7.3.1 モーター電源オン

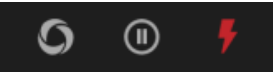
手動モードでは、ユーザーが 3 ポジションイネーブルスイッチを中間位置に保持することでモーターの電源がオンになります。モーターの電源オンが成功したかどうかは、モーター電源オンの音を聞くか、HMI ソフトウェアインターフェースの下部ステータスバーのモーター電源状態表示が赤くなることで判断できます。



 ヒント

モーターの電源オンに失敗した場合は、リアルタイムログを確認してロボットの状態を判断し、モーター電源オン可能な状態にしてから再びモーターの電源を入れます。

自動モードでは、HMI ソフトウェアの下部ステータスバーのモーター電源状態をタップしてモーターの電源を入れます。手動モードと同じ方法で、モーターの電源オンが成功したかどうかを判断できます。



7.3.2 モーター電源オフ

手動モードでは、ユーザーが 3 ポジションイネーブルスイッチを解放するか、強く押し込むことで、モーターの電源をオフにします。

自動モードでは、HMI ソフトウェアの下部ステータスバーのモーター電源状態をタップして、モーターの電源をオフにします。



警告

緊急時には、非常停止ボタンを押してモーターの電源を緊急遮断します。再び電源を入れる時は、まず手動で非常停止スイッチをリセットしてください。

7.4 動作制御

7.4.1 ジョグ動作

ジョグ動作は様々な座標系/モードをサポートします。詳細は下表を参照してください。

	説明	備考
ワールド座標系	直行ジョグ動作は、指定された座標系方向に沿って動作します。例えば、「ワールド座標系」を選択した場合、X をジョグすると、ワールド座標系の X 方向に沿って平行移動します。「ワーク座標系」を選択した場合、Ry をジョグすると、ワーク座標系の Y 方向を中心に回転します。	
フランジ座標系		
ベース座標系		
ツール座標系		
ワーク座標系		
特異点回避	主に直交ジョグ動作の時に手首の特異点を回避するために使用され、XYZ 動作はすべてベース座標系を基準とします。 直交ジョグをする前に、まず 4 軸を 0 または±180 にする必要があります。このモードでは、軸 4 を上記の角度に素早くジョグすることができます。その上で、XYZ をジョグすると、軸 4 の角度はロックされて変化しなくなり、ロボットの姿勢はアーム平面に沿って回転します。Ry をジョグすると、ロボットのフランジは、ロボットの上下腕で構成する平面の法線を中心に回転します（軸 4 角度が 0 または±180 のときのみ Ry をジョグできます）。J6 のジョグは、軸 6 の軸動作ジョグと同じであり、軸 6 のみを調整します。 CR シリーズの 5 軸モデルのベース座標系ジョグは、6 軸の協働モデルの特異点回避モードに対応します。すなわち、X/Y/Z をジョグする時、ロボットの姿勢はアーム平面に沿って回転します。Ry をジョグすると、ロボットフランジはアーム平面に沿って回転します。J5 のジョグは、軸 5 の軸動作ジョグと同じです。	まず「設定」-「運動パラメーター設定」-「詳細設定」ページでパレットデバッグモードを有効にする必要があります。 現行バージョンの適用機種： 産業用 6 軸ロボットシリーズ（XB、NB モデル） 協働ロボット xMate CR/SR シリーズ
ベースと平行	主に直交ジョグ動作の時に手首の特異点を回避するために使用され、XYZ の動作はすべてベース座標系に対して行われます。	

	直交 XYZ をジョグする前に、まずフランジをベースと平行な状態にジョグする必要があります。このモードで J4 と Ry を使用すると、フランジを素早くベースと平行な状態にすることができます。XYZ をジョグすると、ロボットの姿勢は変化しませんが、手首の特異点を自動的に回避できます。J4、Ry、J6 の使用法は特異点回避モードと同じです。 CR シリーズの 5 軸モデルは、このモードで Ry を使用すると、フランジを素早くベースと平行な状態にすることができます。	
軸動作	各軸の動きを個別に制御します。	



ヒント

xMate シリーズの 5 軸モデルは、構造上の制約により通常の座標系で XYZ をジョグすると、ロボットの姿勢がアーム平面に沿って回転します。A をジョグすると、ロボットフランジがアーム平面の法線を中心に回転します。B のジョグは 5 軸の軸動作ジョグと同じです。C ボタンは無効です。

ジョグ動作は連続モードとステップモードの 2 つのモードをサポートします。

ジョグ動作は連続とステッピングの 2 つのモードをサポートします。

- 連続モードでは、モーターに電源を入れた後ジョグボタンを押し続けると、ロボットはイネーブルまたはジョグボタンを放すまで、設定されたジョグ速度で連続的に動作します。
- ステッピングモードでは、モーターに電源を入れた後、ジョグボタンが押されるたびに指定のステップ長を動作します。ユーザーは必要に応じて適切なステップサイズを選択できます。主にロボットの位置/姿勢を正確に調整するために使用されます。



ヒント

ステップモードでは、ジョグボタンを長押ししてロボットが指定されたステップ長を移動した後、一度ジョグボタンを放す必要があります。短く押すと、ロボットはステップ長の途中で停止する可能性があります。

ジョグ速度設定：

ジョグ中のロボットの動作速度を制御するために使用されます。選択可能な範囲は 1%～100% です。100% の時はロボットの最大 TCP 速度 250mm/s に対応します。安全規制により、TCP 速度は 250mm/s を超えません。

7.4.2 クイック移動

HMI 右側の操作パネルは、クイック移動機能を提供します。サポートされるクイック移動は、ドラッグ姿勢、輸送姿勢、ホーム位置及びユーザー定義位置です。

クイック移動の手動モードでの使用法は、ジョグ操作に似ています。手動モードでは、イネーブルスイッチによってモーター電源をオンにし、対応するボタンを押して、ロボットを軸動作で目標位置まで動作させます。動作中の速度はジョグ速度で調整できます。

ドラッグ姿勢、輸送姿勢、ホーム位置はユーザー定義をサポートします。「設定」-「クイック移動」ページで設定できます。設定されていない場合は、デフォルト設定を使用します。

7.4.3 ドラッグ

教示の際に、ドラッグ機能を使用することでプログラミング時間を大幅に短縮できます。また、ロボットのドラッグは、パスの記録及びパスの再現を組み合わせることで、連続した軌跡の教示を簡素化できます。

協働ロボット各モデルの先端ハンドルの紹介と使用例については、付録を参照してください。

注意：ドラッグとその拡張機能（先端ハンドル、教示、パスの記録、パスの再現）は **xMate** シリーズ協働ロボットでのみ利用可能です。

ドラッグ方法	モード	説明
軸動作		各軸の動作は独立し、直接各軸の位置を希望する姿勢に調整することができます；
直交動作	直交移動のみ	ロボットを直交空間の各方向に沿って平行移動させることができます；
	回転のみ	ロボットの TCP を中心に姿勢を回転させることができます；
	自由ドラッグ	自由にドラッグすることができます；

手動モードかつモーター電源オフの状態、操作パネルでドラッグイネーブルスイッチをオンにすると、ロボットは自動的にモーター電源オンになりドラッグモードが有効になります。その状態で手で先端ハンドルのドラッグイネーブルボタンを押すと、ロボットをドラッグして教示とパスの記録を実行できます。



ヒント

ドラッグモードを有効にする前に、適切なドラッグ方法、モードを設定してください。ドラッグモードを有効にすると、ロボットは自動的にモーター電源オンになります。ドラッグモードが有効な状態ではドラッグ方法とモードを設定できないため、ドラッグを無効にしてから設定してください。



警告

ドラッグモードを有効にする前に、ロボットの動力学パラメーターと負荷パラメーターを正確に設定する必要があります。さもなくば、ドラッグモードが正確にオンにならず、ドラッグ中に不安定になる可能性があります。パラメーターの設定は、システムが提供する動力学パラメーターと負荷推定機能を使って実現します。



危険

ドラッグ教示を使用する前に、以下のパラメーターが正しく設定されていることを確認してください。さもなくば、コントローラーが正しい出力トルクを計算できず、ロボットのドラッグ機能も正常に使用できなくなります。

- ロボットの型番
- ロボットの設置方法（床置き設置または天吊り設置）
- 本体と負荷の動力学パラメーター
- 機械ゼロキャリブレーション

7.5 パスの再生

パスの記録が成功した後、確認のためパスを再生できます。確認後、手動で記録済みのパスを保存します。

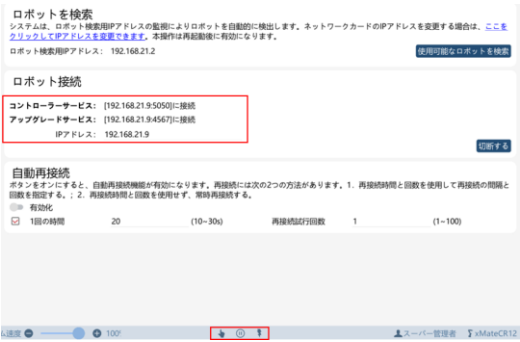
再生モードで「繰り返し動作」にチェックを入れると、ループ再生することができます。

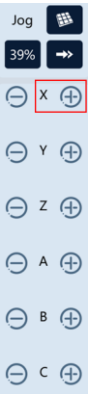
再生速度は 1%から 300%まで設定できます。推奨範囲は 1%から 100%です。再生速度が100%を超える場合、ドライバが追従エラーが発生する可能性があります。

操作例については、「プログラミング」-「パス一覧」の項を参照してください。

7.6 操作例 1：産業用ロボットのジョグ動作

操作を行う前に、HMI がロボットに正しく接続され、モーター電源が正しく投入できることを確認する必要があります（詳細は第 5 章を参照）。

手順	図	説明
1. ロボットの状態を確認します。ロボットが正常に接続されており、手動モードになっていることを確認してください。		
2. ユーザーを切り替えます		
3. 【ベース座標系】を選択します（詳細は第 4 章を参照してください）。 【連続動作】を選択します。		ロボットアーム末端の動作方向は、ベース座標系の方向と同じです。

4. 教示操作盤のイネーブルスイッチを握ります。		モーターの電源を入れると、明確なモーター電源オンの音が鳴り、下部ステータスバーには手動モードでモーター電源オン状態であることが表示されます。ジョグ中はイネーブルスイッチを押したままの状態に保持する必要があります。
5. ロボットのジョグを行います。移動する方向をタップします。この例では、ベース座標系のX正方向に沿って動作することを示しています。		ボタンは押し続ける必要があります。ボタンを放すとロボットの動作が止まります。
6. ジョグを終了するには。ロボットの動作ボタンを放し、イネーブルボタンを放します。		下部ステータスバーは手動モードでモーター電源オフ状態を示します。

7.7 操作例 2：CR 協働ロボットのドラッグ

操作を行う前に、HMI がロボットに正しく接続され、モーター電源が正しく投入できることを確認する必要があります（詳細は第 5 章を参照）。

xMate CR7 協働ロボットを例にドラッグ機能を示します。

手順	図	説明
1. ロボットの状態を確認します。ロボットが正常に接続されており、手動モードになっていることを確認してください。		ドラッグモードを開始する前に、ロボットをドラッグ開始の姿勢にしてください。ロボットが自動モードの場合は、ドラッグボタンは選択できません。

2. ユーザーを切り替えます		
3. 【ドラッグ】ボタンをタップします。ボタンはドラッグがオンの状態を表示し、ロボットのステータスバーはドラッグオン状態に変わります。ロボットから明確なモーター電源オンの音が鳴ります。		
4. ロボットのドラッグイネーブルスイッチを押しながら、ロボットを動作範囲内の任意の位置にドラッグします。		
5. ロボットのドラッグイネーブルスイッチを放し、【ドラッグ】ボタンをタップします。ドラッグボタンはドラッグオフ状態を表示し、ロボットのステータスバーは手動モード/モーター電源オフ状態に変わります。ロボットから明確なモーター電源オフの音が鳴ります。		ドラッグ状態を終了した後、その他の操作を行います。

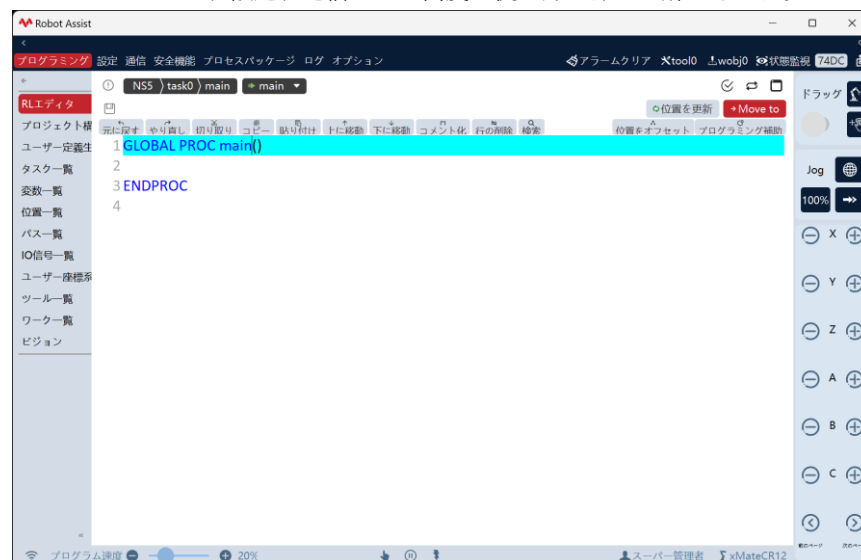
8 プログラミング

8.1 本章の概要

産業用/協働ロボットは非常に柔軟な生産ツールであり、プログラミングによってさまざまなニーズに対応できます。

本章は、xCore 制御システムのプログラミングについて紹介します。

xCore のプログラミング、設定、通信などの高度な使い方を深く理解できます。



8.2 プロジェクト

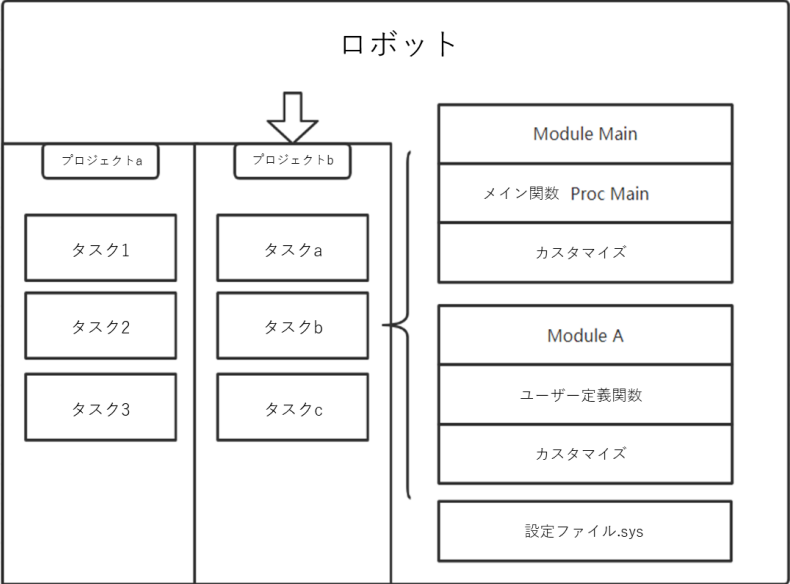
xCore は、プロジェクト単位でプログラムを管理します。プロジェクトには、RL プログラム、ユーザー定義生産画面、タスク、変数、位置、パス、IO、ユーザー座標系、ツール、ワークなどの情報が含まれます。

プログラムは、範囲に応じて 4 つのレベルに分かれています：

プロジェクト	Project	プログラムの最上位レベルで、様々な情報が含まれます。 ロボットは一度に 1 つのプロジェクトしか選択できません。
タスク	Task	1 つのプロジェクトに対して複数のタスクを作成できます。 それぞれのタスクは原則的に互いに独立しており、提供されるインターフェースを介してのみ相互に連携します。
モジュール	Module	モジュールにはプログラムモジュール (.mod) とシステムモジュール (.sys) の 2 種類があります。1 つのモジュールは 1 つのプログラムファイルです。 各モジュールには、特定のロボット機能を実現するためのいくつかのデータ変数と関数が含まれています。 1 つのタスクに複数のプログラムモジュールを作成できますが、main.mod は 1 つしか作成できません。 main.mod には GLOBAL PROC main という関数を含める必要があります。この GLOBAL PROC main はプロジェクト全体の最初に実行される関数として機能します。 各モジュールは、コピーや削除などの通常のファイル操作を行うことができます。

関数	Routine	ユーザーは必要に応じて、モジュールの中でロボット機能またはその他の関数を呼び出すことができます。 ユーザー定義関数をサポートし、異なるカスタム関数を同じプログラムファイルまたは異なるプログラムファイルに保存できます。 関数の定義方法： <pre>PROC main() ... ENDPROC PROC test1() ... ENDPROC PROC test2() ... ENDPROC</pre>
----	---------	---

プロジェクト、タスク、モジュール及び関数の相互関係は下図の通りです。





ヒント

各モジュールの最初の関数の前にあるコードエリアは、宣言セクションと呼ばれ、GLOBAL と LOCAL スcopeの変数を宣言するために使用されます。このセクションで定義された変数は、pptomain が実行されるごとに初期化されます。

例：int0 は変数一覧で定義されており、初期値は 0 です。int1 は宣言セクションで定義されており、初期値は 0 です。

ループ実行の際、int0 は実行されるたびに 1 ずつ増加し、「1, 2, 3……」と表示されます。int1 は実行されるたびに 0 に初期化され、「1, 1, 1……」と表示されます。

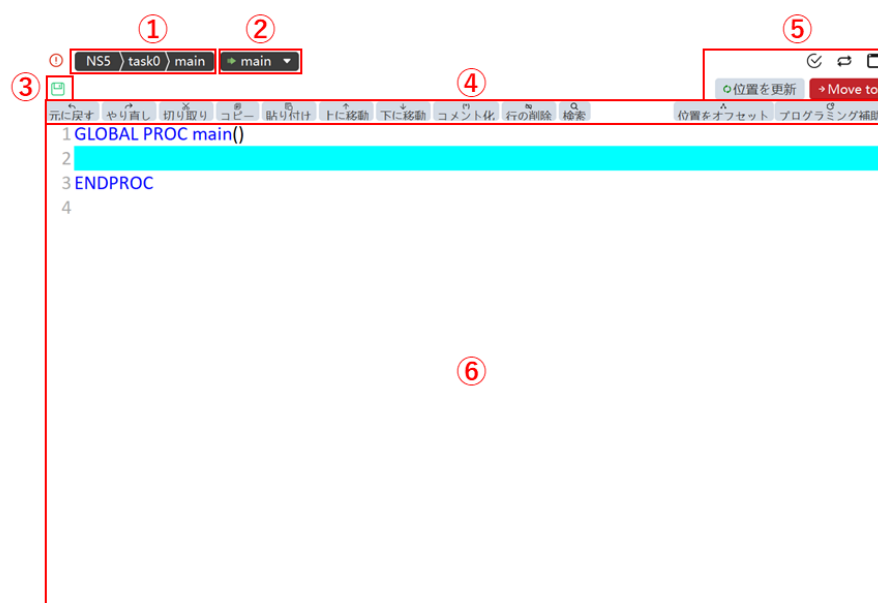
```

1// 宣言セクション
2int int1 = 0
3//
4GLOBAL PROC main()
5  int0++ //変数一覧で定義
6  int1++ //宣言セクションで定義
7  print("int0=",int0)
8  print("int1=",int1)
9  pause
10ENDPROC

```

8.3 RL エディタ

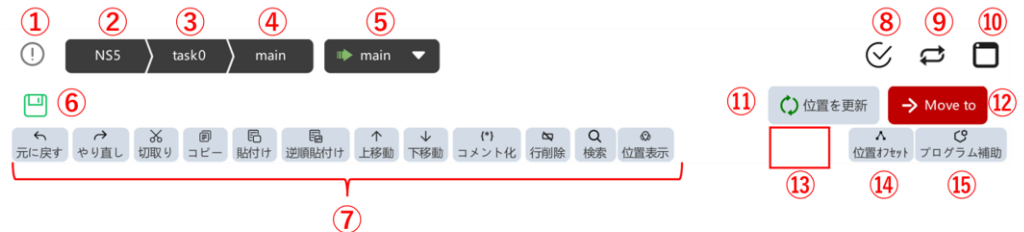
8.3.1 概要



①	プロジェクト、タスク、モジュールの選択、切替、編集を行います。
②	特定の場所へ素早く移動することができます。 (1) main 関数へ移動 (2) カーソル位置へ移動 (3) 現在のモジュールの特定の関数へ移動をサポートしています。
③	現在のタスクを保存します。
④	元に戻す、やり直し、切り取り、コピー、貼付け、逆順貼付け、上移動、下移動、コメント化、行削除、検索、位置表示、位置オフセット、プログラム補助などのボタンがあります。

⑤	位置を更新、Move to、プログラムをチェック、ループモード、出力ウィンドウなどがあります。
⑥	プログラム編集セッションです。プログラミング補助や直接入力で、RL 命令を編集します。

8.3.2 ツールの紹介



①	プロジェクトの同期： 赤：コントローラーにロードされたプログラムが、HMI の現在のプログラムと同期していないことを示しています。 グレー：コントローラーにロードされたプログラムが、HMI の現在のプログラムと同期していることを示しています。
②	現在のプロジェクト： タップすると、プロジェクト設定のページに移動します。
③	現在のタスク： タップすると、タスクを切り替えることができます。
④	現在のモジュール： タップすると、モジュールを切り替えることができます。
⑤	ドロップダウンリストでポインタを移動するターゲットを選択できます。Main 関数に移動したり（プログラムリセットに相当）、カーソルのある行、または特定の関数に移動したりすることができます。
⑥	保存ステータスの表示： ボタンをタップすると、直ちにローカルに保存され、コントローラーにプッシュされます。 緑：プログラムが編集されたが、コントローラーに同期されていない状態です。 グレー：プログラムがコントローラーに同期された状態です。
⑦	前の操作を元に戻します。 元に戻す操作を復元させます。 選択したコード行を切り取ります。同時に複数行の切り取りが可能です。 選択したコード行をコピーします。同時に複数行のコピーが可能です。 切り取りまたはコピーしたコードを、カーソルのある行に挿入します。 コピーしたコード行を逆順で貼り付けます。例えば、1、2、3 行をコピーした場合、貼り付け後は 3、2、1 行の順に表示されます。 選択したコード行を 1 行上に移動します。同時に複数行を移動することができます。 選択したコード行を 1 行下に移動します。同時に複数行を移動することができます。 選択したコードをコメント化またはコメント解除します。

	選択行またはカーソルのある行を削除します。 キーワードを検索します。 位置を表示します。詳細は「位置表示機能」の項を参照ください。
⑧	プログラムをチェック： 現在のプログラムに明らかなエラー（関数名の重複や重要な識別子の欠落など）がないかをチェックします。すべての構文エラーをチェックすることはできません。
⑨	ループモード切り替え： ：プログラムがループで実行されます。 ：プログラムは一度だけ実行されます。
⑩	出力ウィンドウ： Print 出力情報と構文情報を表示します。
⑪	位置を更新： カーソルのある行の位置/姿勢を、現在の位置/姿勢に更新します。 注意：カーソルがポイントがある行になければなりません。
⑫	Move to： 詳細については、「Move to」の項を参照ください。
⑬	教示操作盤キーボード： 教示操作盤のキーボードの表示/非表示を切り替えます
⑭	位置をオフセット： 詳細については、「位置をオフセット」の項を参照ください。
⑮	プログラム補助： 詳細については、「プログラム補助」の項を参照ください。

8.3.3 プログラム補助

プログラム補助パネルは、プログラマーがプログラムを素早く構築できるように、プログラム命令の挿入や命令プロパティ設定の変更を支援します。プログラム補助パネルは、命令を挿入とプロパティ設定の2つの内容に分かれています。

8.3.3.1 命令を挿入

命令を挿入は、プログラムに命令を挿入します。

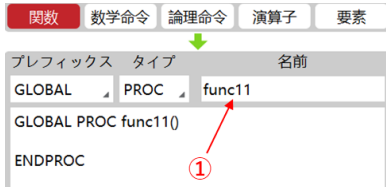


番号	説明
①	プログラム命令セットの選択セクションです。
②	プログラムの共通要素の選択セクションです。
③	プログラム命令/要素の選択セクションです。詳細な命令を選択します
④	命令/選択挿入確認セクションです。選択した命令のデフォルトパラメーターや要素をプログラムに挿入します。

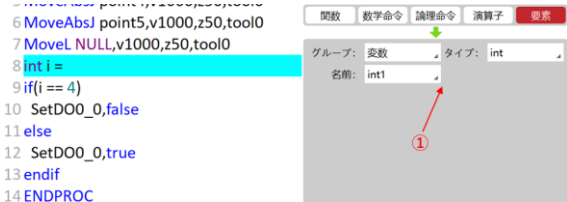
8.3.3.1.1 例 1 : MoveL 命令の挿入

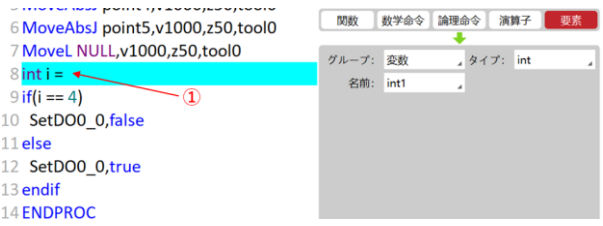
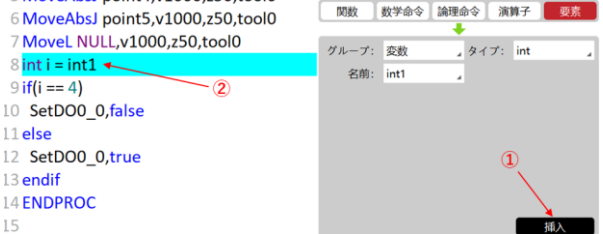
手順	図	説明
1. 「プログラム補助」 → 「命令の挿入」 ボタンをタップし、命令の 挿入パネルを開きます。		
2. 「動作命令」 → 「MoveL」 を選択します。		「動作命令」を選択すると、他の命令は非表示になります。 「MoveL」命令を選択します。
3. カーソルで、挿入したい行の 1つ前の行を選択します。		
4. 「下の行に挿入」ボタンを タップします；		命令を、選択した行の 次の行に挿入します。

8.3.3.1.2 例2：関数の挿入

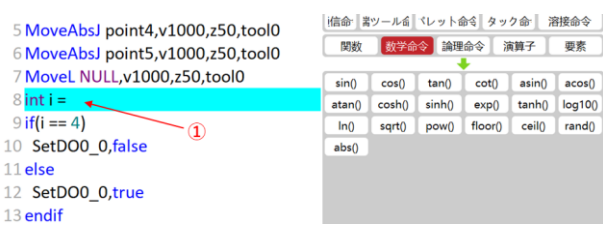
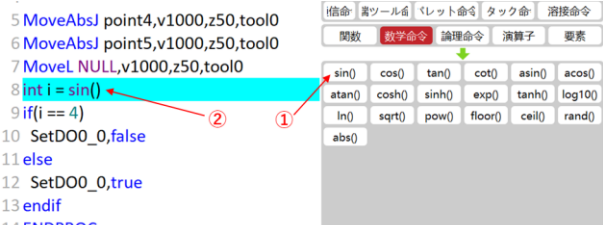
手順	図	説明
1. 「プログラム補助」 → 「命令の挿入」 ボタンをタップし、命令の 挿入パネルを開きます。		
2. 「関数」を選択します。		
3. 「名前」の下にある記入欄 に、関数の名前を入力しま す。		
4. 「挿入」ボタンをタップす します。		「挿入」ボタンをタッ プすると、関数が末尾 に作成されます。

8.3.3.1.3 例3：要素の挿入

手順	図	説明
1. 「プログラム補助」 → 「命令の挿入」 ボタンをタップし、命令の 挿入パネルを開きます。		
2. 「要素」を選択します。		
3. 「要素」パネルで変数など の情報を選択します。		

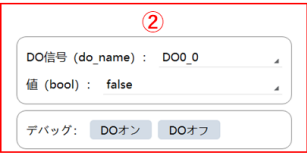
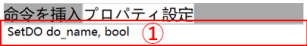
4. 挿入する位置にカーソルを移動します。		
5. 「挿入」ボタンをタップします。		「挿入」ボタンをタップすると、カーソルの位置に文が生成されます。

8.3.3.1.4 例4：数学/論理/演算子の挿入

手順	図	説明
1. 「プログラム補助」 → 「命令の挿入」 ボタンをタップし、命令の挿入パネルを開きます。		
2. 「数学」を選択します。		
3. 挿入する位置にカーソルを移動します。		
4. 「sin()」ボタンをタップします。		ボタンをタップすると、カーソルの位置に sin() が生成されます。

8.3.3.2 プロパティ設定

プロパティ設定は、選択行の命令のパラメーター情報を更新します。



番号	説明
①	定義セクション： 命令のパラメーター名を表示します
②	設定セクション： 命令の各パラメーターを設定します
③	確認セクション： 「更新」ボタンをタップし、パラメーターの変更を実行します

8.3.3.2.1 例 1：SetDO 命令プロパティの変更

手順	図	説明
1. プロパティを変更したい命令のある行を選択します。		プロパティ設定パネルが開き、命令情報がパネルに表示されます。
2. 「プロパティ設定」パネルでパラメーターを設定します。		
3. 「更新」ボタンをタップします		「更新」ボタンをタップすると、選択行の命令パラメーターは設定したパラメーターに更新されます。

8.3.3.2.2 例 2 : MoveAbsJ 命令プロパティの変更

速度 (V) : v1000 滑らか軌跡エリア (Z) : 1
 ツール (T) : tool0 ワーク (W) : wobj0

目標位置 (jointtarget) :

目標位置 (jointtarget) : 位置を更

軸 (°) :

J1 0.00000(J2 0.00000(J3 0.00000(J4 0.00000(

J5 0.00000(J6 0.00000(J7 0.00000(

→ MoveAbsj to 更新

手順	図	説明
1. 動作パラメーターの設定	速度 (V) : <u>v1000</u> 滑らか軌跡エリア (Z) : <u>1</u> ツール (T) : <u>tool0</u> ワーク (W) : <u>wobj0</u>	MoveAbsJ 命令の速度、滑らか軌跡エリア、ツール、ワークパラメーターを設定します。
2. 位置の設定	目標位置 (jointtarget) : 目標位置 (jointtarget) : ① <u>位置を更</u> ② 軸 (°) : ③ J1 0.00000(J2 0.00000(J3 0.00000(J4 0.00000(J5 0.00000(J6 0.00000(J7 0.00000(⑤ ④ → MoveAbsj to 更新	1.ポイントを選択します。 2.選択されたポイントの位置/姿勢の情報を現在のロボットの位置/姿勢に更新します。 3.位置情報を手動で変更できます 4.手動でロボットを選択されたポイントにジョグします。 5.「更新」ボタンをタップすると、位置/姿勢を 3 で変更したデータに更新します。

8.3.3.2.3 例 3：MoveL 命令プロパティの変更

速度 (V) : v1000 滑らか軌跡エリア (Z) : 1

ツール (T) : tool0 ワーク (W) : wobj0

目標位置 (robtargt) :

位置情報

目標位置 (robtargt) : 位置を更

位置 (mm) : X 563.000 Y 0.00000 Z 432.414

オイラー角 (°) : A 180.00 B 0.0000 C 0.0000

→ MoveL to → MoveJ to 更新

オフセット: ☒ None ☐ Offs ☐ Reltool

更新

合成点情報

位置 (mm) : X 563.000 Y 0.00000 Z 432.414

オイラー角 (°) : A 180.00 B 0.0000 C 0.0000

→ MoveL to → MoveJ to

手順	図	説明
1. 動作パラメーターの設定	速度 (V) : <u>v1000</u> 滑らか軌跡エリア (Z) : <u>1</u> ツール (T) : <u>tool0</u> ワーク (W) : <u>wobj0</u>	MoveL 命令の速度、滑らか軌跡エリア、ツール、ワークパラメーターを設定します。
2. 位置の設定	目標位置 (robtargt) : 位置情報 目標位置 (robtargt) : ① 位置を更 ② 位置 (mm) : X 563.000 Y 0.00000 Z 432.414 オイラー角 (°) : A 180.00 B 0.0000 C 0.0000 ④ → MoveL to → MoveJ to 更新 ⑤	1.ポイントを選択します。 2.選択されたポイントの位置/姿勢情報を現在のロボットの位置/姿勢に更新します。 3.位置/姿勢情報を手動で変更できます。 4.手動でロボットを選択されたポイントにジョグします。 5.「OK」ボタンをタップすると、位置/姿勢を 3 で変更したデータに更新します。
3. オフセットの設定	オフセット: ☒ None ☐ Offs ☐ Reltool 更新	

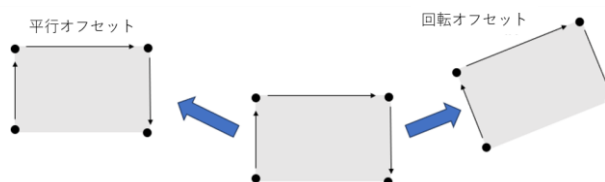
4. オフセット後の位置確認パネル	合成点情報 位置 (mm) : X 563.000 Y 0.00000 Z 432.414 オイラー角 (°) : A 180.00 B 0.0000 C 0.0000 → MoveL to → MoveJ to	当該パネルは編集できません。オフセット設定後のジョグのために使用されます。
----------------------	--	---------------------------------------

8.3.4 位置オフセット

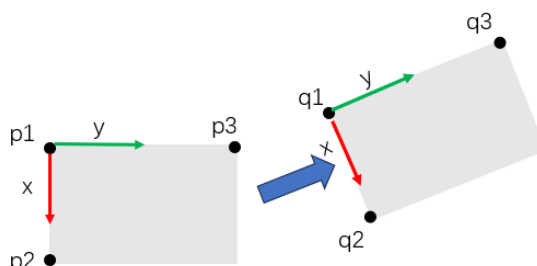
8.3.4.1 概要

このツールは、RL プログラムでユーザーが選択した範囲内の動作命令において、教示点を全体的にオフセットさせます。

平行移動のみが必要な場合は平行オフセットを使用し、xyz の値のみを調整します。姿勢の調整が必要な場合は、「姿勢可変」をオンにし姿勢値 ABC も調整します。



原理：p1、p2、p3、q1、q2、q3 の合計 6 点を使用して、最初の座標系とオフセット後の座標系との関係を計算します。「姿勢可変」をオンにしない場合は、p1、q1 のみを設定します。



8.3.4.2 パラメーター紹介

① 位置選択

Mod : task0/main

開始行 : 1

終了行 : 5

② 位置生成方法

☐ 現在の位置変数に上
 ☒ 新規位置を作成
 ☐ 自動リネーム
 ☒ 末尾追加文字列 'set'

③ 挿入先

Mod : task0/main

挿入行 : 1

④ 次へ

⑤ オフセット量

☐ 手動入力
 ☒ 選択/教示

⑥ ☒ 姿勢可変

⑦

p1	☒ 選択	☐ 教示	point4
p2	☒ 選択	☐ 教示	point5
p3	☒ 選択	☐ 教示	robtarg1
q1	☐ 選択	☒ 教示	記録 50.00,1777.00
q2	☐ 選択	☒ 教示	記録 0.00,0.00,0.00
q3	☐ 選択	☒ 教示	記録 0.00,0.00,0.00

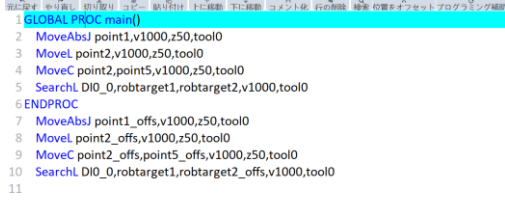
⑧ 前へ 次へ

①	位置選択： プログラム内でオフセットしたい教示点を含む命令文を選択し、タスク/モジュール・開始行・終了行を指定すると、このツールは開始行から終了行までのすべての命令で教示点を検索します。 注意：「//」でコメント化した命令は検索されません。
②	位置生成方法： 「元の位置を上書き」または「新しい位置を作成」を選択します。 「新しい位置を作成」を選択した場合は、新しい位置の命名方法（「自動リネーム」または「元の位置名+指定拡張子」）を選択します。
③	挿入： 新しく生成された命令の挿入位置を指定します。タスク/モジュール及び挿入行を選択します。
④	「次へ」ボタンをタップして、オフセット設定ページに進みます。
⑤	オフセット量： 「手動入力」または「選択/教示」を選択できます。選択/教示のみで「姿勢可変」がサポートされます。
⑥	姿勢を変更する場合はチェックを入れます。変更しない場合は、平行移動のみが適用されます。
⑦	「選択/教示」条件では、位置/姿勢を計算するために使用される 2 点または 6 点を指定します。各点は、「選択」を選択してドロップダウンリストから既存の点を選択するか、「教示」を選択して「記録」ボタンをタップしてロボットの現在の位置を記録します。
⑧	「前へ」ボタンをタップすると、位置選択ページに戻ります。「次へ」ボタンをタップすると、オフセットの計算を開始します。

1. インターフェースがフリーズしたり、位置情報が失われることを避けるために、位置オフセットの数は 1 回ごとに 1,000 以下にすることを推奨します。
2. 位置オフセットを計算する時、システムが自動的にその位置の妥当性をチェックし、妥当ではない場合はエラーを報告します。同時に命令の生成も失敗しますが、エラーより前の位置は生成されます。
エラーの原因：①パラメーターエラー ②特異点 ③到達不能 ④その他のエラー
3. 指定された mod の開始行と終了行の間に位置がない場合、「オフセットする位置が見つかりませんでした」というエラーを報告します。
4. このツールを使用する際は、ロボットに接続する必要があります。接続していない場合は、「ロボット通信エラー」というエラーを報告します。
5. p1-p2 ベクトルと、p1-p3 ベクトルの角度は 1° 以上でなければなりません。1° 未満だと、3 点が同一線上にあると判断され、「基準点エラー」というエラーを報告します。q1、q2、q3 についても同様です。
6. 命令内の位置の名は、位置一覧と一致する必要があります（大文字と小文字は区別されません）。位置一覧に名前がない場合は、「xxx 点が見つかりませんでした」というエラーを報告します。
7. 位置オフセットを開いたまま、「タスク一覧」、「位置一覧」、RL 編集ページでタスクの追加・削除、位置の追加・削除、RL の編集、プロジェクトの切替操作を行った場合、この位置オフセットの内容は自動的に更新されません。一度位置オフセットを閉じて、再度開く必要があります。

8.3.4.3 操作例

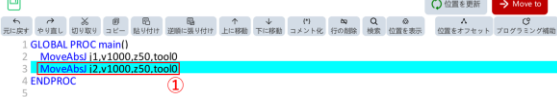
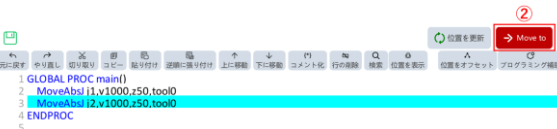
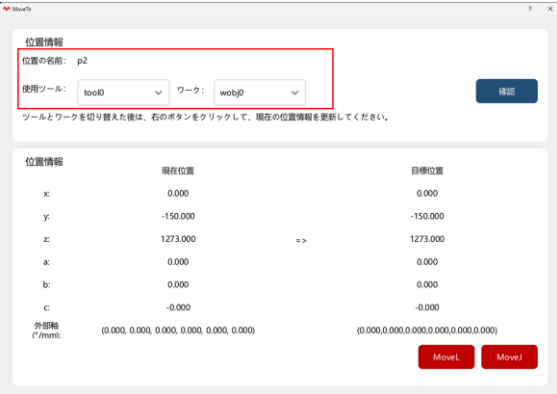
手順	図	説明
1. 「位置オフセット」ボタンをタップしてツールを開きます。		「プログラマー」以上の権限で、手動モードでのみ使用可能です。
2. オフセットしたい位置を指定し、パラメーターを設定した後、「次へ」ボタンをタップします。		task0 の main の第 2~5 行を選択します。 「新規位置を作成」をタップし、新しい位置の拡張子を指定します。 挿入位置は当該 mod ファイルの 7 行目を選択します。
3. 「オフセットする位置の数」がポップアップで表示されます。		オフセットするのは、 第 2 行の point1 第 3 行の point2 第 4 行の point2 と point5 第 5 行の robtarget2 です。 注意： 1. 「point2」は 2 回計算されますが、実際には 1 回しかオフセットされず、オフセット後の位置も 1 つしか生成されません。 2. SearchL の「robtarget1」がオフセット対象に含まれないのは、robtarget1 が位置名を保存するために使用される変数であり、その位置はロボットの動作中に生成されるからです。
4. オフセット量の計算方法と基準位置を設定し、「次へ」ボタンをタップします。		オフセットの実行中はプログレスバーがポップアップするので、しばらくお待ちください。

5. 「オフセット成功」のポップアップが出ます。7~10行目はオフセット後の位置に対する新しい命令です。		位置一覧には生成された位置情報が表示されます。 
---	---	--

8.3.5 Move to 機能

「Move to」は、一連の操作を通じてロボットを選択したポイントに移動させる機能です。この機能は、指定した位置にロボットを正確に移動させるために使用されます。

8.3.5.1 操作例

手順	図	説明
1. RL エディター画面で対応する運動指令を選択します。		「プログラマー」以上の権限で、手動モードの場合にのみ使用可能です。
2. 右上の「Move to」ボタンをタップします。		
3. 「Move to」のポップアップウィンドウで、対応する位置、ツール、ワークを選択します。		単一位置への動作指令の場合、位置の選択は不要です。 複数位置への動作指令の場合、移動するポイントを確認する必要があります。 使用するツールとワークオブジェクトを指定してください。
4. 「Move to」のポップアップウィンドウで動作方法を選択し、「Move to」機能を実行します。		右下の「MoveJ」、「MoveL」、「MoveAbsJ」（軸指定の位置でのみ表示）を選択し、長押しして「Move to」機能を実行します。

8.3.6 位置表示機能

8.3.6.1 概要

「位置表示」機能は、現在のロボットの停止位置がRLプログラムのどの行に対応しているかをユーザーが簡単に把握できるようにするための機能です。



位置を表示機能を起動する方法は以下の通りです：

1. 「PPToMain」 ボタンをタップしたとき
2. 「位置表示」 ボタンをタップしたとき
3. 「位置を更新」 ボタンをタップしたとき

ロボットの位置表示機能が起動されると、RL プログラム画面の特定の行または複数の行に @ 記号が表示されます。この記号は、ロボットの現在位置がその行で使用されている位置と一致していることを示します。

8.3.6.2 位置を表示パラメーターの設定

「設定」 - 「HMI 設定」の「位置パラメーターをチェック」にある設定を参照します。これにより、ロボットの実際の位置と RL プログラムの位置との誤差範囲を設定できます。

位置パラメーターをチェック

精度: 0.10 +/- (0° - 30°)

保存

8.4 プロジェクト設定

プロジェクト設定画面は、現在のプロジェクトの確認・変更を行います。

8.4.1 ロボットを選択

ロボットと接続せずに PC 上で RobotAssist を起動する場合や、オフラインで RL プロジェクトを編集したい場合は、ここでロボットを選択できます。システムは、そのロボットのプロジェクトデータを表示します。

ロボットと接続している場合は、現在接続しているロボットが表示されます。

8.4.2 プロジェクト

切り替え：選択されたプロジェクトに切り替えます。

デフォルトに設定：選択されたプロジェクトをデフォルトに設定します。デフォルトのプロジェクトは、ロボット起動後に自動的にロードされます。

名前を付けて保存：選択されたプロジェクトを名前を付けて保存し、コントローラーにプッシュします。

インポート：プロジェクトをインポートします。インポートページを開き、インポートするプロジェクト、パスを選択します。

エクスポート：プロジェクトをエクスポートします。エクスポートページを開き、エクスポートするプロジェクトとパスを選択します。

新規作成：プロジェクトを新規作成します。プロジェクト名はアルファベット、数字、アンダースコアが使用できます。

名前の変更：選択されたプロジェクトの名前を変更します。

削除：選択されたプロジェクトを削除します。



警告

プロジェクトを削除すると、元に戻すことはできません！

8.4.3 同期

HMI とコントローラーのプロジェクトには 3 つの同期方法があります。

1. 接続すると、プロジェクトファイルは自動的に同期されます。コントローラー内の情報が優先されます。
2. プロジェクトの重要な情報（ツールやワークなど）が変更されると、すぐにコントローラーに同期されます。
3. RL コードは、実行やデバッグなどの操作中に自動的にコントローラーにプッシュされます。作業が完了していない状態で保存する必要がある場合は、このページの「コントローラーへプッシュ」ボタンをタップしてください。

「コントローラーから読み込み」ボタン：

コントローラーからプロジェクトを読み込みます。

この操作は、現在のすべてのプロジェクトを上書きします。

「コントローラーへプッシュ」ボタン：

現在のプロジェクトをコントローラーに同期させます。

8.4.4 プロジェクトの復元

プロジェクトに変更があった場合、コントローラーは定期的にバックアップを行います。ドロップダウンリストで時間に応じて命名されたプロジェクトのバックアップを選択し、「復元」をタップしてプロジェクトを復元できます。  をタップすると、復元できるプロジェクトのリストを更新します。

8.4.5 事前定義パラメーター

事前定義パラメーターは、ユーザー座標系、ツール、ワーク、動作速度及び滑らか軌跡エリアの設定があり、これらの変数は RL 命令のパラメーターとして使用されます。このページでは、変数の意味が表示されますが、編集はできません。

8.5 ユーザー定義生産画面

ユーザー定義生産画面は、簡単かつ直感的な対話型画面を提供します。ユーザーは、レジスタ、DI/DO 信号、位置、プロジェクト変数を選択するだけで、シンプルな確認画面を作成できます。

この画面では、レジスタの表示と編集、DI 信号の状態表示、DO 信号の操作、位置の更新、プロジェクト変数の表示および編集ができます。

8.5.1 概要



操作/表示パネルには複数のウィジェットが含まれ、以下のボタンを使用してインポート、エクスポート、編集、および削除を行うことができます。

①	操作/表示パネル： ウィジェットの選択、表示、操作を行います。 現在選択可能なウィジェットには、シングル DI、シングル DO、レジスタ、位置を更新、プロジェクト変数があります。
②	パネル編集バー： 生産画面データのインポートとエクスポート、ページの切り替え、ウィジェットの編集と削除ができます。

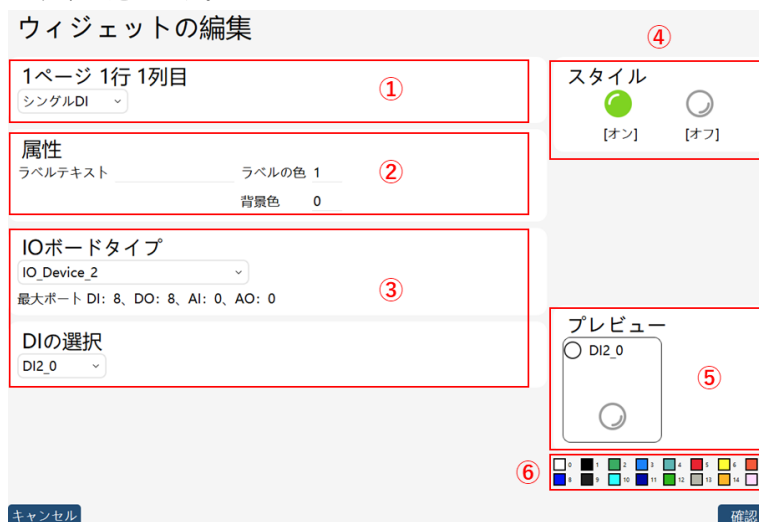
ウィジェットには 3 つの表示部分があります。上段はウィジェットの名前の表示、中段はウィジェットのラベル（ユーザー定義可能）の表示、下段はユーザーのウィジェットに対する操作/表示を行います。ウィジェットによって提供する内容は異なります。

8.5.2 基本操作

操作	図	説明
ウィジェット選択		ウィジェットの左上隅にある円をタップすると、パネル内の特定のウィジェットを選択することができます。選択されたウィジェットは、左上隅の円の中に赤い点が表示されます。 例えば、現在のパネルで選択されているウィジェットは、第 1 行第 2 列の「吸盤」ウィジェットです。
インポート		 ボタンをタップしてインポートインターフェースに入り、インポートするファイルのパス、インポート項目、置換するかどうかを選択します。
エクスポート		 ボタンをタップしてエクスポートインターフェースに入り、エクスポートするウィジェットとパスを選択します。
編集		 ボタンをタップしてウィジェット編集ページに入ります。詳細な使用方法は、下記を参照してください。

8.5.3 ウィジェット紹介

ユーザー定義生産画面インターフェースは現在 **5 種類**のウィジェットを提供します。ユーザーは生産環境に応じて、適切なウィジェットを選択できます。**5 種類**のウィジェットは、シングル DI、シングル DO、レジスタ、位置を更新、プロジェクト変数です。ウィジェットの編集ページは以下の通りです。



①	ウィジェットタイプの選択： なし、シングル DI、シングル DO、レジスタ、位置を更新、プロジェクト変数が選択できます。
②	共通属性： ラベル文字、ラベル色、背景色が設定できます。色については⑥を参照してください。
③	個別属性： 各ウィジェットのプロパティです。ウィジェットごとに異なるので、詳細は以下を参照してください。
④	スタイルバー： 表示されるデータのスタイルを提供します。 ウィジェットによっては、スタイルを提供しないものもあります。(例：レジスタウィジェット)。
⑤	プレビュー： 現在編集しているウィジェットのプレビューを見ることができます。

⑥	色選択： 16色選択できます。②で色を設定する際は、目標色の番号を入力します。
---	--

8.5.3.1 シングル DI ウィジェット

シングル DI ウィジェットは、DI 信号の表示に使用します。選択可能な IO ボードのタイプには、

- (1) ユーザーが「IO デバイス」で設定した IO デバイス
- (2) ユーザーがプロジェクトの「IO 信号一覧」で設定したユーザー DI 信号があります。

編集インターフェースは下図に示す通りです。

選択された DI が True の場合は、ウィジェットはスタイルの「オン」アイコン（緑色）を表示し、DI が False の場合は、スタイルの「オフ」アイコン（灰色）を表示します。

シングル DI ウィジェットは、DI のステータス表示としてのみ使用可能です。

8.5.3.2 シングル DO ウィジェット

シングル DO ウィジェットは、DO 信号の表示と設定に使用します。選択可能な IO ボードのタイプには、

- (1) ユーザーが「IO デバイス」インターフェースで設定した IO デバイス
- (2) ユーザーがプロジェクトの「IO 信号一覧」で設定したユーザー DO 信号があります。

編集インターフェースは下図に示す通りです。

選択された DO が True の場合は、ウィジェットはスタイルの「オン」アイコン（緑色のオンボタン）を表示し、DO が False の場合は、スタイルの「オフ」アイコン（赤のオフボタン）を表示します。

シングル DO ウィジェットは、「操作を許可」プロパティにより、「操作/表示パネル」で操作の可否を設定できます。操作を許可した場合は、「操作/表示パネル」インターフェースで下図の赤ボタンをタップして、DO 信号を操作できます。



8.5.3.3 レジスタウィジェット

レジスタウィジェットは、「通信」-「レジスタ」で設定したレジスタの値を監視、変更できます。レジスタは「バスデバイス」でフィルタリングできます。

注意：レジスタウィジェットは、レジスタ配列をサポートしていません。

編集インターフェースは下図に示す通りです。



レジスタウィジェットは、スタイルを提供せずレジスタ値の表示のみが可能です。レジスタウィジェットは、テンキーで値を書き込むことで編集できます。「操作/表示パネル」インターフェースで、「値の表示/編集ボックス」をタップしてレジスタ値を入力します。

（注：書き込みは、レジスタが「書き込み専用」の場合のみ可能です）。

値の変更操作の手順：

手順	図	説明
1. 選択		レジスタウィジェットの値の表示/編集ボックスをタップします。
2. 入力		レジスタに設定する値を入力します。
3. 値の変更		

8.5.3.4 位置を更新ウィジェット

位置を更新ウィジェットは、位置一覧内の位置情報を変更することができます。選択可能な位置の形式には、直交座標と各軸座標の2種類があります。
編集インターフェースは下図に示す通りです。

ウィジェットの編集

操作を許可

☒ はい ☐ いいえ

位置を選択

位置の形式 ☐ 直交 ☒ 各軸

位置を選択 point0 ▾

位置情報 J1: 0.00°, J2: 0.00°, J3: 0.00°
J4: 0.00°, J5: 0.00°, J6: 0.00°, J7: 0.00°
O1: 0.00°, O2: 0.00°, O3: 0.00°
O4: 0.00°, O5: 0.00°, O6: 0.00°

位置デバッグ

更新

Move to

プレビュー

☐ point0

位置を更新

キャンセル

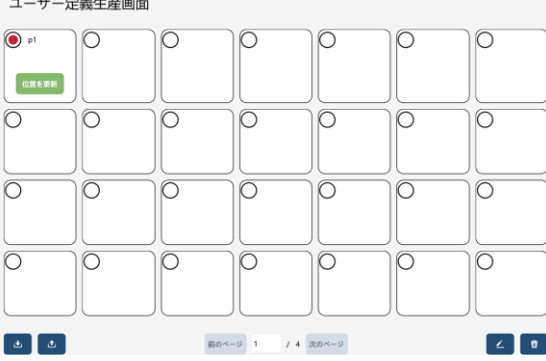
確認

位置を更新ウィジェットは、スタイルを提供せず、位置更新ボタンを提供します。
「操作/表示パネル」インターフェースで位置更新ボタンをクリックすると、位置の値を更新することができます。

位置を更新の操作手順：

手順	図	説明
1. 選択	ユーザー定義生産画面 ☒ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ 前のページ 1 / 4 次のページ ⏪ ⏩	空欄を選択します

空欄を選択します

2. 位置を更新を選択	 ウィジェットの編集 1ページ 1行 1列目 位置を更新 属性 ラベルテキスト ラベルの色 1 背景色 0 操作を許可 ☒ はい ☐ いいえ 位置を選択 位置の形式 ☒ 直交 ☐ 各軸 プレビュー キャンセル 確定	
3. 更新する位置情報を選択	 ウィジェットの編集 位置を選択 位置の形式 ☒ 直交 ☐ 各軸 位置を選択 p1 位置情報 X: 0.00mm, Y: -150.00mm, Z: 1273.00mm A: 0.00°, B: 0.00°, C: -0.00° O1: 0.00°, O2: 0.00°, O3: 0.00° O4: 0.00°, O5: 0.00°, O6: 0.00° 位置タグ 更新 Move to プレビュー p1 位置を更新 キャンセル 確定	直交座標と各軸座標の 2 種類の タイプをサポートします
4. 完成	 ユーザー定義生産画面 p1 位置を更新 完了ページ 1 / 4 次のページ	完了をクリックすると、生産画面で位置の更新操作が行えます。

8.5.3.5 プロジェクト変数ウィジェット

プロジェクト変数ウィジェットは、「プロジェクト」-「変数一覧」で設定された変数を監視および変更することができます。選択可能な変数の型には、int、byte、bool、double の 4 種類があります。配列はサポートしていません。



プロジェクト変数ウィジェットは、スタイルを提供せず、プロジェクト変数の数値表示のみを提供します。

値の変更は、数字キーパッドで値を入力して行います。「操作を許可」属性を使用して、「操作/表示パネル」で操作できるかどうかを設定できます。

「操作/表示パネル」インターフェースで、値の表示/編集ボックスをクリックすると、プロジェクト変数の値を変更する操作ができます。

手順	図	説明
1. 選択		プロジェクト変数ウィジェット内の値表示/編集ボックスをクリックします
2. 入力		値を入力します

3. 値の変更	ユーザー定義生産画面 <table border="1"><tr><td>☐ p1</td><td>☒ in0</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td></tr><tr><td> 印刷を印刷 </td><td> 2 </td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td></tr><tr><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td></tr><tr><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td></tr></table> 上 下 前のページ 1 / 4 次のページ 戻る 印刷	☐ p1	☒ in0	☐	☐	☐	☐	☐	印刷を印刷	2						☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	
	☐ p1	☒ in0	☐	☐	☐	☐	☐																														
	印刷を印刷	2																																			
	☐	☐	☐	☐	☐	☐	☐																														
	☐	☐	☐	☐	☐	☐	☐																														
☐	☐	☐	☐	☐	☐	☐																															

8.6 タスク一覧

xCore 制御システムはマルチタスクをサポートします。

マルチタスクは、複数のロボットプログラムの「並列」処理を実現できます。典型的な適用シナリオは以下の通りです。

- **Main** プログラムの実行が停止しても、特定の信号を継続的に監視（バックグラウンド PLC 機能に似ていますが、実際の PLC ほど応答できません）
- ロボットが動作メインプログラムを実行している間に、メインプログラム実行に制約されない周辺機器とのデータ受信・送信処理
- ロボットの動作中、教示操作盤を通しての入力

xCore の「タスク一覧」は、タスクの管理インターフェースを提供します。既存のタスクのプロパティの確認、タスクの新規作成、編集、削除が行えます。

タスク一覧						
名前	タイプ	電源投入後自動実行	タスク有効	セキュリティレベル	優先度	説明
task0	モーションタスク		☒		高	デフォルト
main1	MOD					主関数
task1	一般タスク		☒		低	
main2	MOD					メイン関数

8.6.1 タスクプロパティ

タスクを作成または編集する際は、タスクのプロパティを設定する必要があります。

タスクを編集: task0

基本情報	
名前: task0	説明: デフォルト
タスク情報	
タイプ: モーションタスク	タスクを有効にする: ☒ 有効化
優先度: 高	
関数の作成	
main関数: ☐ 作成	その他の関数: ☐ 今すぐ作成 ☒ 作成しない

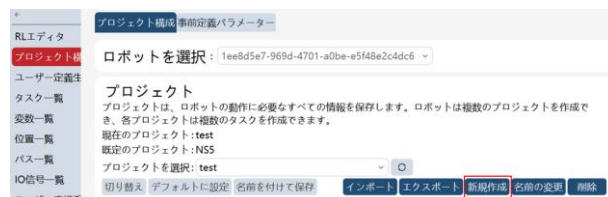
タスクプロパティ	説明
タスク名	タスク名は全タスクの中で重複してはならず、アルファベット、数字、アンダースコアのみをサポートします。頭文字を数字にすることはできず、最大長は 20 文字です。
説明	ユーザーの理解を助けるために、タスクの説明を追加できます。
タスク情報	一般タスク、モーションタスク、自動実行タスクがあります。モーションタスクとは、RL 命令を使ってロボットの動作を制御することを意味します。モーションタスクとなるタスクは 1 つだけです。

タスクを有効にする	タスクを有効化します。
優先度	タスクの実行優先度を設定します。
関数の作成	main 関数の生成にチェックを入れると、タスクを作成した後に main 関数が自動的に生成されます。その他の関数も同様です。

8.6.2 一般タスク、モーションタスク

8.6.2.1 タスクの新規作成

最初のタスクを作成する前に、新しいプロジェクトを作成してください。すでにプロジェクトがある場合は、新しいプロジェクトを作成する必要はありません。



プロジェクトの新規作成



タスクの新規作成

タスクの新規作成画面で、当該タスクのプロパティを編集できます。

タスクを新規作成

基本情報
名前: task0 説明:

タスク情報
タイプ: モーションタスク タスクを有効にする: ☒ 有効化
優先度: 高

関数の作成
main関数: ☒ 作成 その他の関数: ☐ 今すぐ作成 ☒ 作成しない

キャンセル 前へ 次へ

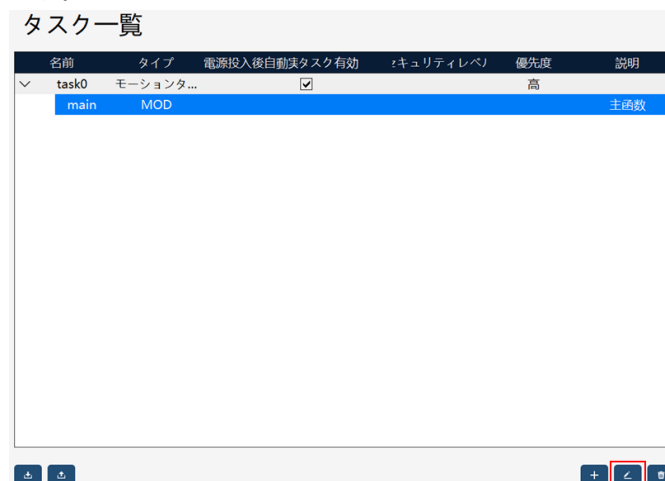
タスクプロパティの編集

注意:

- タスクは 10 個まで作成可能です。
- モーションタスクとなるタスクは 1 つだけです。
- タスクタイプなどのプロパティの変更は直ちに有効になります。

8.6.2.2 プログラムモジュールの新規作成

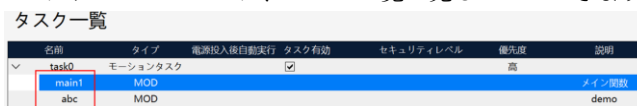
各タスクは、いくつかのプログラムモジュール（mod ファイル）を含みます。モジュールを作成・編集する場合は、タスクを左のマークで展開し、モジュールを選択した状態で編集ボタンをタップします。



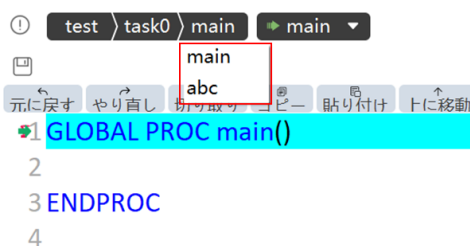
新しいページで「+」ボタンをタップすると、プログラムモジュールを新規作成できます。モジュール名、説明などの基本情報を設定した後、「完了」をタップしてプログラムモジュールを作成します。



新規作成されたプログラムモジュールは、タスク一覧で見ることができます。

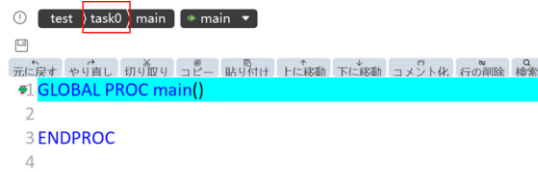


RL エディタ上で新しいプログラムモジュールを選択して、プログラミングを行うことができます。



8.6.2.3 起動と停止

RL エディタで `task0` をタップすると、タスクを選択できます。開始/停止ボタンまたは外部信号を使用して、選択されたタスクの開始/停止を制御できます。



注意事項：

- 通常、バックグラウンドタスクに `wait` 命令が含まれていない場合は、絶えずループして実行されます。
- バックグラウンドタスクがコンピューターリソースを消費しすぎて、コントローラーがその他のタスクを処理できなくなる可能性があります。
- 変数 `VARS` と定数 `CONST` のスコープはそれぞれのタスクに限定されますが、`GLOBAL` レベルの `PERS` 変数はグローバル変数として扱われます。
- `PPToMain` を実行すると、実行中ではないすべてのタスクが `PPToMain` を実行します。
- 実行中のタスクがある場合、タスク一覧の内容の変更は禁止されています。

8.6.2.4 タスク間通信

タスク間通信は `PERS` 変数を使用します。

- 通信するすべてのタスクで、同じ名称の `GLOBAL PERS` 変数を定義します。変数のデータ型、次元も同じにします。
- `PERS` 変数を使用し、タスクの実行やデータの伝送などを制御します。
- 変数一覧、位置一覧の変数は、すべてのタスクで使用できます。

注意事項：

- 1 つのタスクで `PERS` 変数の初期値を指定します。複数のタスクで同じ `PERS` 変数に初期値を指定した場合、最初に実行されたタスクで定義された初期値が使用されます。
- `PERS` 変数と `WaitUntil` または `WHILE` 命令を使用してあるタスクが他のタスクを待つ場合、判定の連続実行でシステム資源が過剰に使用されることを避けるために、`wait` 命令 (0.1s 以上) と併用するようにしてください。

8.6.3 自動実行タスク

v2.0.1 バージョンから、`xCore` 制御システムは自動実行タスク機能を提供します。

自動実行タスクはマルチタスク機能に属し、`RL` 言語で書かれたプログラムを自動実行します。一般タスクやモーションタスクと比較して、自動実行タスクには 2 つの特徴があります。

- モーター電源オンや起動などの命令なしで、自動的に起動します。
- 停止ボタンでは、停止しません。

この 2 つの特徴により、自動実行タスクの命令サイクルは、モーター電源オンからオフまでの制御システムほぼ全体をカバーします。

自動実行タスクの典型的な適用シナリオは以下の通りです。

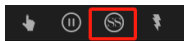
- 定期的にロボットの位置を確認し、レジスタや IO などを通じてホストコンピューターに通知
- カスタマイズパルス信号を出力
- 複数のデバイス間でデータを伝送

8.6.3.1 自動実行タスクの新規作成

タスク一覧で「+」をタップして新しいタスクを作成します。タスクタイプで自動実行タスクを選択し、その他のプロパティ設定を完了します。「次へ」ボタンをタップして自動実行タスクを作成します。

8.6.3.2 自動実行タスクの開始と停止

通常のタスク起動と同様に、自動実行プログラムを作成・設定した後、手動モードまたは自動モードでモーターの電源を入れ、「実行」をタップして自動実行タスクを開始します。

プロジェクト内で自動実行タスクが実行されている時は、下部ステータスバーの中央に  が表示されます。自動実行タスクは、一時停止またはモーター電気オフをタップしても停止しません。この時、「自動実行タスク実行中」のボタンをタップすると、確認

ダイアログボックス  がポップアップ表示され、「OK」をタップすると自動実行タスクの停止操作が実行されます。自動実行タスクは、プログラム異常がトリガーされていないときは、このボタンでのみ停止できます（自動実行タスクを一時停止する前に、通常タスクを一時停止する必要があります）。

8.6.3.3 自動実行タスクの電源オン時の自動起動の設定

プロジェクト設定インターフェースで、自動実行タスクが属するプロジェクトをデフォルトプロジェクトに設定します

タスク一覧で、自動実行タスクの“電源投入後自動実行”と“タスク有効”の 2 つのオプションにチェックを入れます。

タスク一覧				
名前	タイプ	電源投入後自動実行	タスク有効	セキュリティレベル
> task0	モーションタスク		☒	
> task1	自動実行タスク	☒	☒	セキュリティ関連

最後に **ptomain** をタップして、プロジェクト構成を HMI からコントローラーに同期すると、自動実行タスクが自動起動するように設定されます。

8.6.3.4 自動実行タスクの安全レベル

自動実行タスクには、タスクの優先度と電源オン時に

実行するかどうかという構成に加えて、安全レベルというプロパティがあります。

安全レベルは、自動実行タスクの作動中に異常（データ受信の失敗など）がある場合、コントローラーの異常処理方法を定義するために使用します。

- 安全無関係：ホストコンピューターまたはオペレーターが自動実行タスクの作動状況を気にしない場合に適用します。当該安全レベルを設定する自動実行タスクの実行中にエラーが生じると、自身を停止するのみで、他のタスクは影響を受けずに実行され続けます。
- システム停止：自動実行タスクのデータがプログラムの安全論理に影響する状況（自動実行タスクによる動作位置の決定、自動実行タスクを通じてホストコンピューターへのロボット制御方法の通知など）に適用されます。システム停止レベルの自動実行タスクは、実行中にエラーが生じるとすべてのタスクを一時停止させます。

8.6.3.5 自動実行タスクのデバッグについて

- コントローラーは、手動電源投入モードでの自動実行タスクのシングルステップデバッグをサポートしていますが、デバッグ中の再起動やデバッグされていない自動実行タスクの起動によって事故が生じることを避けるために、まず通常タスクを使用しデバッグしてから、タスクタイプを自動実行タスクに変更することをお勧めします。
- 自動実行タスクでは、失敗する可能性がある命令に対してエラー対応処理を増やしてください。例えば、**ReadXX** 命令はネットワークのジッター、周辺機器の故障などによって読み取りタイムアウトまたは失敗を引き起こす可能性があります。自動実行タスクをできるだけ停止させないように、失敗する可能性のあるコードを **Try/Catch** 文を使って捕捉し、**Catch** 文でエラー処理（**Goto** を使用しデータを再読み込みするなど）をしてください。
- 自動実行タスクは、長期間にわたって実行されるという特別な性質があるため、プログラムデバッグインターフェースの右上隅にあるシングル  | ループ  の実行設定は自動実行タスクには適用されません。すべての自動実行タスクはループで実行されます。

8.6.4 タスク監視

状態監視で、各タスクの現在の実行状態を監視できます。

8.7 変数一覧

8.7.1 変数の命名規則

RL 言語の変数名は、アルファベット、アンダースコア、数字が使用でき、頭文字は必ずアルファベットまたはアンダースコア「_」である必要があります。変数名は、システム予約語と重複できません。システム予約語については、システム予約語の項を参照してください。これ以外に、以下の注意事項があります。

- 同じモジュール内で、同名の GLOBAL 変数と LOCAL 変数の存在は許可されません
- 異なるモジュールで、同名の GLOBAL 変数の存在は許可されません
- 異なるモジュールで、同名の LOCAL 変数の存在は許可されます
- 同じモジュール内で、GLOBAL 変数及び LOCAL 変数とモジュール内の関数が、同名になることは許可されません
- 異なるモジュールで、GLOBAL レベル関数と変数が、同名になることは許可されません



ヒント

変数名が 2 文字の場合、2 番目の文字を「h」や「b」にしないように注意する必要があります。2 文字目が「h」や「b」の場合、変数は 16 進数または 2 進数に変換される可能性があります。詳細については、進数変換を参照してください。

8.7.2 変数のスコープ

RL 言語システムは 3 種類のスコープを定義します。

グローバル変数 (GLOBAL)	モジュールの宣言セクションで宣言でき、現在のタスク内のすべてのプログラムモジュールから参照できます。 タスクをまたがって変数にアクセスする必要がある場合は、GLOBAL PERS で宣言する必要があります。 注意：変数一覧と位置一覧の値は、すべてグローバル変数です。すべてのタスクから読み書き可能です。
ローカル変数 (LOCAL)	モジュールの宣言セクションで宣言でき、現在のプログラムモジュールからのみ参照できます。
関数内変数 (ROUTINE)	関数内部で宣言でき、当該関数からのみ参照できます。また、スコープ変数を宣言する時に、スコープタイプ (GLOBAL または LOCAL) の指定はできません； 注意：スコープは変数にのみ適用され、ユーザー定義関数には適用されません。

8.7.3 変数と定数

各変数は、プログラム実行中の変更可否によって、可変変数 (VAR)、持続変数 (PERS)、定数 (CONST) の 3 種類に分類されます。

VAR (Variable)	可変変数は、プログラム動作中に再割り当て可能な変数です。
CONST (Const Variable)	定数は、プログラム実行中に値を再割り当てできません。値は、初期化時に指定しなければなりません。

PERS (Persisten Variable)	持続変数は、プログラムの実行中に値が変化した場合、変数の初期値が自動的に現在値に変更されます。 注意： ● プログラム実行中に PERS 型変数の値が変更されても、プログラムエディターの宣言セクションに表示される初期値は直ちには更新されません。プログラムが再読み込みされるか、プログラムが停止したときにプログラムエディターの宣言セクションに表示される初期値が最新の値に更新されます。 ● プログラムが実行中であるかどうかに関係なく、「変数一覧」画面では PERS 変数の初期値のみを確認することができます。現在値は状態監視または print 命令で見ることができます。
---------------------------	--

8.7.4 システム予約語

以下は **RL** プログラムのシステム予約語です（大文字と小文字は区別されません）。

Module、EndModule、Proc、EndProc、Func、EndFunc、SetDO、DO_ALL、SetGO、SetAO、WaitDI、Wait、WaitUntil、WaitWObj、WBID、Q、P、J、V、W、T、S、L、CA、DURA、IGNORELEFT、EJ、1J、FCBV、FCCV、FCOL、FCXYZ、FCCART、PE、PER、TCP、ORI、EXJ、CFG、PDIS、JDIS、MoveAbsJ、MoveJ、MoveL、MoveC、MoveT、LOCAL、TASK、GLOBAL、VAR、CONST、PERS、INV、DOT、CROSS、sin、cos、tan、asin、cot、acos、atan、atan2、sinh、cosh、tanh、ln、log10、pow、exp、sqrt、ceil、floor、abs、rand、GetCurPos、Print、PrintToFile、ClkRead、TestAndSet、IF、Else、Endif、WHILE、ENDWHILE、for、from、to、endfor、Break、Continue、Del、Int、Double、Bool、String、BYTE、Robtarget、Speed、Zone、Tool、Wobj、Jointtarget、TriggData、Load、FCBoxVol、FCSphereVol、FCCylinderVol、FCXyzNum、FCCartNum、Pose、CLOCK、INTNUM、SYNCIDENT、TASKS、Call、Return、EXIT、Pause、StopMove、StartMove、StorePath、RestoPath、True、False、Interrupt、When、Offs、CalcJointT、CalcRobT、CRobT、RelTool、SocketCreate、SocketClose、SocketSendByte、SocketSendInt、SocketSendString、SocketReadString、SocketReadBit、SocketReadInt、SocketReadDouble、AccSet、MotionSup、TriggIO、TriggJ、TriggL、TriggC、On、Off、clock、intnum、userframe、pinf、ninf、FCFRAME_WORLD、FCFRAME_TOOL、FCFRAME_WOBJ、FCFRAME_PATH、FCPLANE_XY、FCPLANE_XZ、FCPLANE_YZ、FC_LINE_X、FC_LINE_Y、FC_LINE_Z、FC_ROT_X、FC_ROT_Y、FC_ROT_Z、Offs、CalcJointT、CalcRobT、CRobT、RelTool、Start、Time、ClkReset、ClkStart、ClkStop、CONNECT、WITH、IDisable、IEnable、ISignalDI、Single、SingleSafe、WaitWobj、DropWobj、WobjIdentifier、WobjAngle、ActUnit、DeactUnit、INTNO、Exp、DoubleToStr、WaitSyncTask、FCAct、FCDeact、FCLoadID、FCCalib、FCSupvForce、FCSupvTorque、FCSupvPosBox、FCSupvPosSphere、FCSupvPosCylinder、FCSupvOrient、FCSupvOrient、FCSupvReoriSpeed、FCSupvTCPspeed、FCCondForce、FCCondTorque、FCCondOrient、FCCondReoriSpeed、FCCondPosBox、FCCondPosCylinder、FCCondPosSphere、FCCondTCPspeed、FCCondWaitWhile、FCRefLine、FCRefRot、FCRefSpiral、FCRefCircle、FCRefForce、FCRefTorque、FCRefStart、FCRefStop、FCSetSDPara

8.7.5 進数変換

RL プログラムは、数字やアルファベットの後に進数識別子を付加することで、16 進数、2 進数または指数表記を入力することができます。

例 1

0~9、a~f 及び A~F に拡張子「h」を付加することで、RL コンパイラは対応する数字やアルファベットを 16 進数として扱い、コンパイラ内部で 10 進数に変換して扱います。

8h は、16 進数の 8 及び 10 進数の 8 を表します；

bh は、16 進数の b 及び 10 進数の 11 を表します；

25h は、16 進数の 25 及び 10 進数の 37 を表します；

例 2

0~9、a~f 及び A~F に拡張子「b」を付加することで、RL コンパイラは対応する数字やアルファベットを 2 進数として扱います。

1b は、2 進数の 1 及び 10 進数の 1 を表します；

10b は、2 進数の 10 及び 10 進数の 2 を表します；

1010b は、2 進数の 1010 及び 10 進数の 10 を表します；

例 3

数字の後に「e±x」を付加することは、その数字に 10 の x 乗をかけることを意味します。

5e+20 は、 5×10^{20} を表します；

26e-15 は、 26×10^{-15} を表します；

112e-10 は、 112×10^{-10} を表します；

8.7.6 変数の宣言

変数は使用する前に宣言する必要があります、変数宣言文の書式は以下の通りです。

SCOPE STORAGE TYPE varname [= value]

ここで、

1. **SCOPE** は変数のスコープです。詳細は変数のスコープを参照してください。
2. **STORAGE** は変数のタイプです。詳細は変数と定数を参照してください；
3. **TYPE** は変数の型です。基本型であっても特殊型であっても良いです。詳細は変数の型を参照してください。
4. **varname** は変数名です。詳細は変数の命名規則を参照してください；
5. 角括弧[]内部はオプションであり、変数宣言時に初期化することも初期化しないままにしておくこともできます。宣言時に明示的に初期化されない変数については、システムは変数型に応じて自動的に初期値を割り当てます。デフォルトの初期値は、場合によってはプログラムの実行に問題を引き起こす可能性があるため、手動で追加した変数はすべて初期化することを推奨します。

以下はいくつかの変数宣言の例です。

例 1

VAR int counter = 8 // 整数型変数 counter を宣言し、初期値を 8 にします。

VAR double time = 2.5 // 浮動小数点数型変数 time を宣言し、初期値を 2.5 にします。

VAR bool ifOpen = true // bool 型変数 ifOpen を宣言し、初期値を true にします。

例 2

同じ名前の変数は宣言できません。

```
VAR int counter = 8
VAR double counter = 2.5
```

この時、コンパイラは「変数の追加に失敗しました」というエラーを報告します。

例 3

ただし、GLOBAL 変数と LOCAL 変数は同じ名前を使用できます。

```
VAR int counter = 1
GLOBAL int counter = 555
```

異なるスコープの変数は同じ変数名にできますが、混乱や誤用を避けるために、特にプロセス上のメリットがない限り同じ名前の変数の使用は推奨されません。



ヒント

while のようなループの内部で変数宣言を行わないでください。その部分のコードを繰り返し実行すると宣言が重複し、「変数の追加に失敗しました」というエラーを報告します。ループの外部で変数を宣言してください。

注意事項：

PERS タイプの ROUTINE 変数の宣言は、サポートされていません。

異なるレベルの変数または関数が同名である場合、コンパイラはスコープの優先度に従って使用する変数を決定します。優先度の高い変数が最初に選択されます。各スコープの優先度は以下の通りです。

- 同名の変数がある場合、スコープの優先度は：ROUTINE > LOCAL > GLOBAL
- 同名の関数がある場合、スコープの優先度は：LOCAL > GLOBAL

8.7.7 持続変数（PERS 変数）

プロジェクトで持続変数（PERS 変数）を作成すると、その値は RL 停止時、ロボット再起動時、ロボット電源オフ時、ロボット停電時でも不揮発性記憶媒体に保持され、ロボット電源再投入時または RL 再実行時に変数の値を保持した値に復元します。変数が初めて作成される時、再編集されるときにだけ、その変数に初期値を割り当てます。（注意：変数一覧に追加された PERS 変数のみが値を保持することができ、変数宣言セクションで定義された PERS 変数は値を保持できません。）

現在、持続変数として以下の変数の型をサポートしています。

int、byte、double、bool、string、pose、speed、zone、fcboxvol、fcspherevol、fccylindervol、fcxyznum、fccartnum、torqueinfo、tool、wobj、socketserver、socketconn、serials

PERS 変数の設定

以下の赤枠に示す場所で PERS 変数を作成できます。



変数一覧、位置一覧、ツール一覧、ワーク一覧のいずれかのボタンをタップすると、そのタイプのユーザー変数の画面に入ります。保持できる変数には「PERS」プロパティ項目があり、「YES」は当該変数が持続変数（PERS 変数）であることを示します。例えば、int 型の PERS 変数を作成する場合は、以下のように設定します。



PERS 変数の変更

xCore 制御システムの PERS 変数は初期値+保持値の形式で保存されます。

初期値とは、ユーザーが変数一覧で入力したデータを意味します。保持値とは、プログラム実行中にプログラムによって値が変更された後、不揮発性記憶媒体に保存されたデータを意味します。基本型 (int/bool/double) の保持値は状態監視で確認でき、構造体データ (位置、ツール、ワークなど) は print 命令で表示できます。

RL プログラムの実行中は、演算子「=」によって PERS 変数を変更することができます。変更されたデータは保持値としてコントローラー内部に保存されます。次にプログラムが実行されると、まず対応する変数の保持値が読み込まれ、保持値が存在しない場合は変数の初期値が読み込まれます。

PERS 変数を位置を更新ボタンまたは変数一覧の編集機能で変更すると、変数の初期値と保持値が同時に変更されます。

8.7.8 変数一覧の操作

変数一覧インターフェースは、ロボットシステム内のほぼすべての変数に対して、新規作成、表示、変更、削除の操作を行います。現在、以下をサポートしています。

番号	変数型	説明
1	システム定義変数	ユーザーが変更できない変数であり、特定のシステムパラメーターを保存するために使用されます（tool0/wobj0 など）。
2	ユーザー定義変数	ユーザーが変更でき、複数のプログラムで使用できる変数です（ユーザーが設定したツールやワークなど）。
3	プログラム変数	プログラム内でユーザーによって定義された変数です。通常、現在のプログラムとそのサブルーチンでのみ使用されます。システムがサポートする変数型のほとんどが使用できます。

8.7.8.1 変数の表示

speed/zone のようなプログラミング補助を使用して定義および変更を行う変数タイプについては、変数一覧でも表示および変更が可能です。利便性とエラー数を減らすために、変更はプログラミング補助インターフェースを使用して行い、変数一覧インターフェースでは表示操作のみを行うことを推奨します。



ヒント

変数一覧インターフェースで表示・変更できる変数は、現在ロードされているロボットプログラムで使用されているものに限定されます。他のプログラムがロードされると、表示される変数が変わります。

8.7.8.2 変数の編集

新しい変数を追加したり、既存の変数を変更したりする必要がある場合、「新規作成」または「編集」ボタンをタップして、変数編集ページに入ることができます。

変数型	変数を新規作成する際に、変数のタイプを選択するために使用されます。サポートされているすべての変数タイプがリストされています。
変数名	挿入する変数名です。
配列次元	配列の作成や変更使用され、最大 3 次元の配列をサポートします。

8.8 位置一覧

8.8.1 概要

位置一覧は、プロジェクトに関する位置データの統合管理のために使用されます。RL プログラムで使用する位置データは、プログラムで使用する前に位置一覧で設定する必要があります。



①	フィルタ設定： 一覧に表示したい位置をフィルタリングするために使用します。タイプ、タスク、名前、説明、関数によるフィルタリングが使用可能です。
②	位置表示セクション： 各位置の名前、タイプ、PERS 変数かどうか、位置、ツール、ワーク、タスク、関数などのプロパティを表示します。
③	「位置を更新」ボタンで、位置データをロボットの現在の位置/姿勢に更新できます。 右側のボタンは、ロボットを当該位置に移動するために使用します。「Move to」ボタンを押すと、Move to ウィンドウが表示されます。直交タイプの位置データに対しては、MoveL または MoveJ ボタンをタップすることでロボットは直線または軸動作でその位置に動作します。各軸タイプの位置データに対しては、MoveAbsJ ボタンをタップすると、ロボットは軸動作でその位置に動作します。

④	機能ボタンセクション 📁 はインポート、 📤 はエクスポート、 + は新規作成、 ✎ は編集、 🗑️ は削除を行います。
---	--

位置編集インターフェース

直交タイプの場合

位置を編集: point1

①

基本情報

名前: point1 タイプ: ☐ 各軸 ☒ 直交 PERS: ☐ YES ☒ NC

次元: スカラー 説明: 関連座標系: tool0 wobj0

タスク: task0 モジュール: main 位置が属するタスクとモジュールを選択してください。

②

姿勢情報の編集

point1

注: 直交データを手動で変更しても、各軸データに反映されません。その逆も同様です。

位置 (mm): X 0 Y -150 Z 1273 エルボー (°): 0

姿勢: ☒ オイラー角 (°) ☐ クォータニオン

A 0 B 0 C 0

Conf/パラメーター: cf1 0 cf2 0 cf3 0 cf4 0 cf5 0 cf6 0 cf7 0

cfx 1

③

①	基本情報 位置の名前、説明、PERS かどうか、次元、タスクおよびモジュールを設定します。
②	位置/姿勢情報: 「位置を更新」ボタンをタップすると、位置データを現在の位置/姿勢に更新します。手動での変更も可能です。直交座標の位置/姿勢を手動変更した場合、軸座標データは変更されません。その逆も同様です。 Conf パラメーターでは、位置に対応する Conf パラメーターを変更することができます。
③	「キャンセル」ボタンをタップすると、位置一覧ページに戻り、変更は保存されません。「完了」ボタンをタップすると、変更が保存されます。

各軸タイプの場合

位置を編集: point0

①

基本情報

名前 : point0 タイプ : ☒ 各軸 ☐ 直交 PERS : ☐ YES ☒ NC

次元 : + - スカラー 説明 : 関連座標系 : tool0 wobj0

タスク : task0 モジュール: main 位置が属するタスクとモジュールを選択してください。

②

姿勢情報の編集

point0

注: 直交データを手動で変更しても、各軸データに反映されません。その逆も同様です。

軸 (°) : 軸1 0 軸2 0 軸3 0 軸4 0 軸5 0 軸6 0 軸7 0

外部軸 (" or mm) : O1 0 O2 0 O3 0 O4 0 O5 0 O6 0

③

①	基本情報 位置の名前、説明、PERS かどうか、次元、関連座標系、タスクおよびモジュールを設定します。
②	位置/姿勢情報： 配列の場合は、要素を選択して内容を変更することができます。 「位置を更新」ボタンをタップすると、位置データを現在の位置/姿勢に更新します。手動での変更も可能です。直交座標の位置/姿勢を手動変更した場合、軸座標データは変更されません。その逆も同様です。
③	「キャンセル」ボタンをタップすると、位置一覧ページに戻り、変更は保存されません。「完了」ボタンをタップすると、変更が保存されます。

8.8.2 操作例

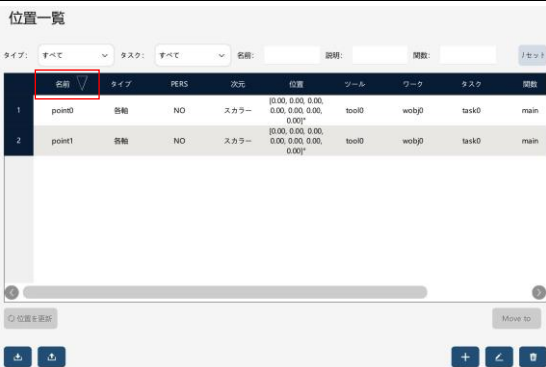
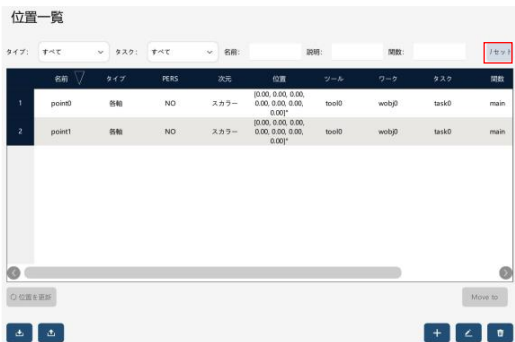
8.8.2.1 位置の作成/編集

手順	図	説明
1. 新規作成/編集		+ または - ボタンをタップし、位置編集ページに入ります。
2. 基本情報の編集		新しい位置を作成する際のみ、位置の種類を選択できます。 位置を編集する際は、名称、説明（任意）、PERS、次元（任意）、タスク、モジュールなどを設定できます。 ユーザーは次元の追加・削除ボタンを使って、位置の次元を設定できます。
3. 位置/姿勢情報の編集		位置の種類を選択し、「位置を更新」をタップすると、ロボットの現在の姿勢が取得されます。 新しい位置を作成する場合は、デフォルトで現在の位置/姿勢が取得されます。
4. 完成		「完了」ボタンをタップし、現在の情報を保存します。

8.8.2.2 Move to 機能

手順	図	説明
1. 対応する位置を選択し 「Move to」ボタンをタ ップします。		
2. ツールとワークを選択 します。		ツールとワークが位置情報内の ツールとワークと一致する場合 は、確認ボタンをタップせずに Move to ボタンをタップできま す。一致しない場合は、 Move to ボタンはグレイアウトされ、動 作できません。確認ボタンをク リックした後に動作が可能とな ります。
3. 「Move to」操作を実行 します。		各軸タイプの位置データの時は、 MoveAbsJ 動作のみ実行可 能です。直交タイプの場合は、 MoveJ および MoveL 動作が実行 可能です。

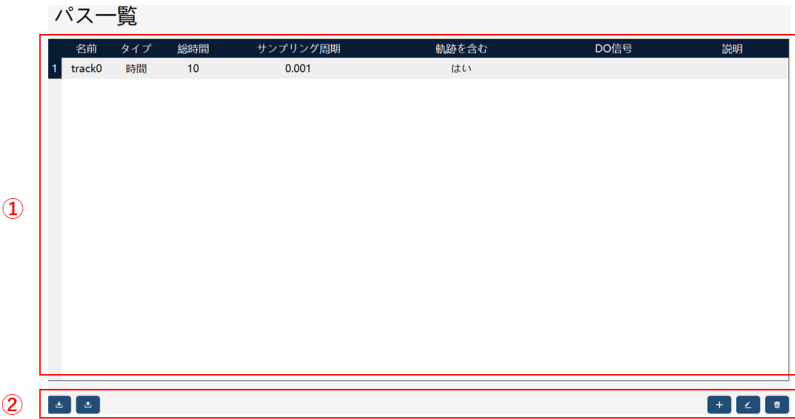
8.8.2.3 位置の並び替え

手順	図	説明
1. 初めて位置一覧画面に入ると、デフォルトで並び替えアイコンが名前の後ろに表示されます。		矢印が下向きの場合は降順、矢印が上向きの場合は昇順です。
2. 並び替えのために、名前の部分をタップします。		現在の並び替えアイコンの状態に基づいて並び替えを行います。 注意点として、並び替え機能を使用していない場合、初めて画面に入った際の降順アイコンは、位置一覧が降順に並んでいることを意味しません。
3. 並び替えのリセット		「リセット」をタップすると、位置一覧のデータが並び替え前の状態に戻ります。

8.9 パス一覧

8.9.1 概要

パス一覧はドラッグ教示でパスを記録したり、パス再生などの操作を行います。



①	パス一覧： 関連プロパティを表示します。名前、タイプ、総時間、サンプリング周期、軌跡の有無、DO 信号、説明を表示します。
②	機能ボタンセクション： [Import] はインポート、[Export] はエクスポート、[New] は新規作成、[Edit] は編集、[Delete] は削除です。

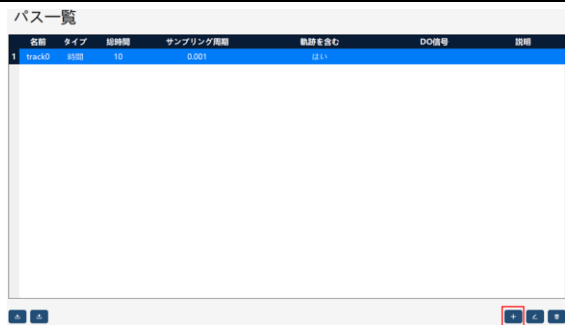
パス編集インターフェース：



①	基本情報： パスの名前、説明を設定します。
②	パスを記録： DO 信号を記録したい場合は、「DO を選択」ボタンをタップします。ポップアップダイアログボックスで DO 信号を選択します。各 DO 信号は DI 信号に関連付けできます。DI 信号の変化がパスを記録するときに記録され、その後の再生では記録された DI 信号に従って DO 信号の出力を実行します。DI 信号と関連付けられていない場合、DO 信号の変化は直接記録されます。この場合は、「状態監視」-「IO 信号」インターフェースで DO 信号の出力を手動で設定することができます。

③	記録の再生、ループ再生するかどうか、実行速度などが設定可能です。「再生」ボタンをタップすると再生が開始し、記録したパスが期待通りかどうかを確認できます。
④	キャッシュ内のパスを、次回以降の使用のために保存します。

8.9.2 操作例

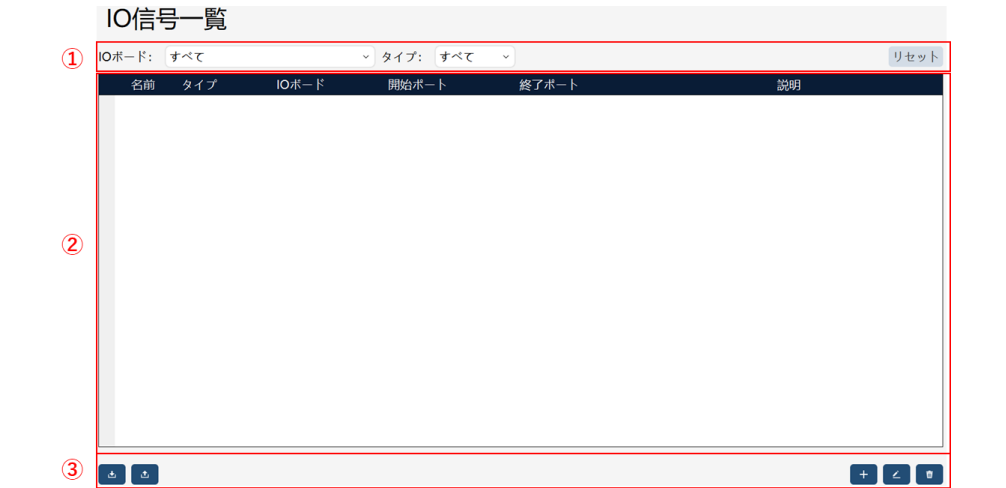
手順	図	説明
1. ドラッグモードをオンにし、右下の「+」ボタンをタップして新しいパスを作成します。		この操作は、ロボット先端にあるハンドルのボタンを押すことも可能です。
2. 記録時間を設定し「パスを記録」ボタンを押し、先端をドラッグします。時間が経過するかまたは「停止ボタン」をタップするまで記録します。		注意：パスの記録に失敗しないように、記録時間が経過してしまう前にドラッグを停止してください。
3. ドラッグモードをオフにし、自動モードに切り替え、モーター電源オンボタンをタップしてロボットのモーターの電源を入れます。再生速度、ループの有無を設定し、「再生」ボタンをタップすると、軌跡が再生されます。		
4. パスの確認後、保存をタップすると、記録されたパスをファイルに保存できます。		

8.10 IO 信号一覧

8.10.1 概要

「IO デバイス」の電源をオンにした時にデフォルトで作成される一般的な IO 信号の他にユーザーIO 信号を作成したい場合は、「IO 信号一覧」インターフェースで作成する必要があります。

RL プログラムでは、Signalxx 型の変数を使用して IO 信号を保存したりアクセスしたりします。詳細は、RL 命令の関連する章を参照してください。



①	フィルタ設定： 一覧に表示したい IO 信号をフィルタリングするために使用します。
②	IO 信号一覧： IO 信号のプロパティを表示します 例： ● 信号名 ● タイプ：DI/DO/GI/GO ● IO ボード：ROKAE が提供する基準 IO モジュール、Profinet バス、Ethernet/IP バスデバイス等 ● 開始ポート、終了ポート： IO 信号マッピングに対応する物理アドレス番号
③	機能ボタンセクション： IO 信号のインポート、エクスポート、新規作成、編集、削除を行います。



警告

1. マッピングされた IO ポートが物理的な限界値を超えていたり、ポートの割り当てが重複しているなど IO 設定にエラーがある場合は、制御システムは起動時に SYS_ERR 状態になり HMI にアラームメッセージが表示されます。この場合は、ユーザーは誤った設定を修正するためにシステム設定インターフェースに入るのみが可能で他の操作はできません。

2. ユーザー定義 IO は、システム出力にマッピングできません。

設定された IO は状態監視インターフェースで見ることができます。この画面は、IO の強制出力またはシミュレーション入力をサポートします。



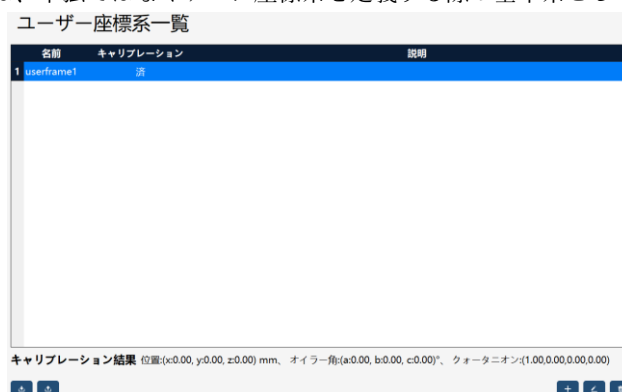
8.10.2 操作例

手順	図	説明
1. 右下の「+」ボタンを タップして IO を新規 作成します。		「プログラマー」以上の権限 で、新規作成・編集操作を行え ます。
2. プロパティを設定しま す。		IO 信号の名前、説明、IO ボー ド、信号タイプ、ポート番号を 設定します。
3. RL で信号を使用しま す。		入力信号 (DI/GI) につい ては、入力信号の変数名を使用し て入力の状態を RL プログラム で直接読み取ります。 例: DI1 の状態を判定条件とす る場合 IF (di1 == true) do something... ENDIF 出力信号 (DO/GO) につい ては、RL プログラムで特別な命 令 SetDO と SetGO を使用して 処理できます。詳細は各命令の 説明の章を参照してください。

8.11 ユーザー座標系一覧

8.11.1 概要

ユーザー座標系は、単独ではなくワーク座標系を定義する際の基準系として使用されます。



ユーザー座標系を作成するには、「キャリブレーション」、「手動入力」、または「設定しない」を選択します。

「キャリブレーション」を選択すると、3点法を使って設定します。ユーザー座標系を設定する前に、1つのツールを設定しておき、そのツールのTCPを使用してユーザー座標系を設定する必要があります。先のとがったツールを推奨します。

事前にユーザー座標系がわかっている場合は、手動で入力することもできます。「設定しない」を選択した場合は、ユーザー座標系はワールド座標系と一致します。

8.11.2 ユーザー座標系のキャリブレーション

手順	図	説明
1. 右下隅の「+」ボタンをタップしてユーザー座標系を新規作成します。		「プログラマー」以上の権限で、新規作成・編集操作を行います。
2. 図に示す手順に従ってキャリブレーションしてください。		ロボットをジョグして、設定済みのツール TCP を目標座標系の原点、x 軸上の点、xy 平面上の点、または y 軸上の点に動かし、「1 点目確認、2 点目確認、3 点目確認」ボタンをタップします。 3 点とも確認したら、「キャリブレーション実行」ボタンをタップします。

8.12 ツール一覧

8.12.1 概要

ツールは、ロボットのフランジに取り付けられ特定の加工プロセスを実行するために使用される器具です。よくあるツールには、空気/電動グリッパー、溶接ガン、スプレーガンなどがあります。ロボットは、出荷時にはツールが付属していません。ロボットを使用する前に、実際の状況に応じて適切なツールを購入または設計し、ロボットを使用する前に装着と設定を完了しておく必要があります。

ツールを使用する前に、ツールの中心点を設定する必要があります。

xCore 制御システムでは、ツールに対応するデータ型は `tool` です。`tool` 型の詳細説明については、「RL 命令-変数」の章を参照してください。

8.12.2 基本概念

ツールの属性には、ツールの幾何学的パラメーターを表す中心点と姿勢、ツールの動力学パラメーターを表す重量、重心、回転イナーシャがあります。

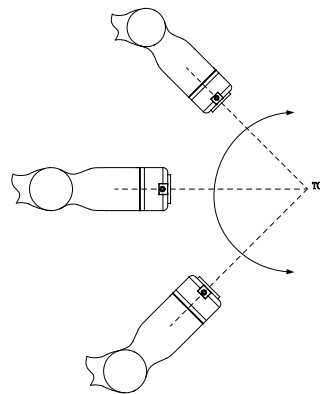


ヒント

`tool0` はシステムで事前に定義されているツールで、ツール座標系はフランジ座標系と一致し、パラメーターはすべて `0` です。`tool0` の内容は変更できません。

8.12.2.1 ツール中心点

ツール中心点 (Tool Center Point, TCP) は、ロボットが作業を行うために使用するツール上の特定の点のことです。例えば、溶接ガンのワイヤの先端や、グリッパーの指の先端などです。ロボットは TCP の位置を一定に保ちながら、TCP を中心に回転して姿勢を変えることができます。



ツールによって TCP は異なります。状況に適した TCP を定義することはプログラミングの効率を大幅に向上させます。



ヒント

特に記載がない限り、本書に「ロボットの位置、速度、加速度」と記載されているところは、すべてワーク座標系に対しての TCP の位置、速度、加速度のことです。

8.12.2.2 ツール座標系

ツール座標系の設定とは、ツール座標系がフランジ座標系に対してどのような位置/姿勢かを決定するプロセスです。

ツールのフランジに対する位置/姿勢情報がわかっている場合は、教示操作盤で「手動入力」を選択し、直接入力することができます。

ツールのフランジに対する位置/姿勢情報が不明な場合は、**xCore** はツール座標系を設定するために 3 つの方法を提供します。

- 4 点法。ツール座標系の中心点の位置を設定します。
- 3 点法。ツール座標系の姿勢を設定します。
- 6 点法。ツール座標系の中心点の位置と姿勢を同時に設定します。4 点法+3 点法に相当します。

さらに、**xCore** は「TCP 補正」機能を提供しており、ツール座標系の位置精度をさらに向上させることができます。

8.12.2.3 負荷パラメーター

xCore システムは、**load** 型変数を使用してツールの負荷パラメーターを保存します。

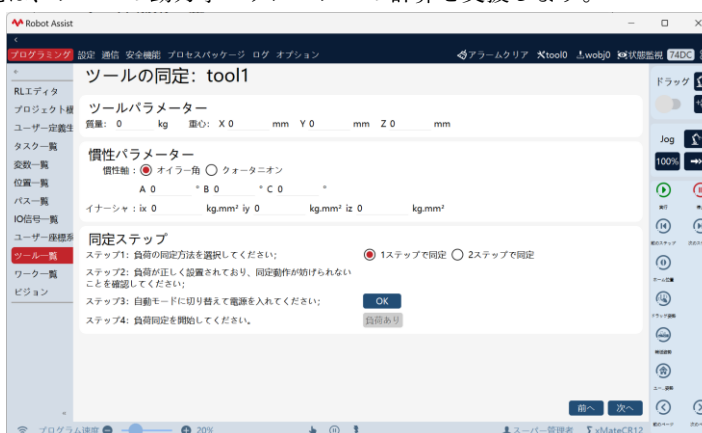
注意：外部ツールを使用する場合、**tool** 変数の **load** パラメーターは、ハンドに付くワークの負荷を保存します。

ツールの負荷を定義する 2 つの方法：

- ツールの負荷データが既知の場合は、ツール負荷の設定ページで手動入力を選択し、対応するデータを直接入力することができます。
- ツールの負荷データが不明な場合は、**xCore** システムの負荷推定機能を使って推定できます。

8.12.2.4 負荷推定

負荷推定機能は、ツールの動力学パラメーターの計算を支援します。



負荷推定は 2 つの方法（1 ステップ推定と 2 ステップ推定）をサポートします。産業用ロボットは 2 ステップ推定のみ実行可能です。通常、2 ステップ推定の精度は 1 ステップ推定よりも優れています。

「2 ステップ推定」の操作手順：

1. ロボットを自動モードに切り替え、モーターの電源を入れます。
2. 無負荷状態で負荷推定プログラムを実行し、推定プログラムの完了を待ちます。
3. 負荷を取り付け、負荷推定プログラムを実行し、推定プログラムの完了を待ちます。
4. 推定が完了すると、結果を示すポップアップウィンドウが表示されます。「決定」をタップしてください。

協働ロボットは、「1 ステップ推定」もサポートします。

操作手順：

1. 負荷推定方法を「1 ステップ推定」にします。
2. 負荷を取り付けます。負荷が正しく取り付けられており、推定動作が干渉などで妨げられないことを確認します。
3. 自動モードに切り替えてモーターの電源を入れます。
4. 「負荷あり」をタップして負荷推定プログラムを実行し、推定プログラムの完了を待ちます。
5. 推定結果を確認して保存します。



ヒント

ツールの動力学パラメーターを正確に定義することは非常に重要です。定義が正しくない場合は、ロボットの動作性能に影響を与え、更にロボットの過負荷を引き起こして破損させることもあります。

負荷推定を実行する前に 30 分以上ロボットを暖機運転すると、推定の精度が向上します。

負荷イナーシャの計算はフランジ座標系に基づきます。

床置き設置のロボットのみ、負荷推定をサポートしています。



ヒント

負荷推定中に以下のような状況が発生した場合、推定は終了し取得したデータはすべて失われるため、再推定が必要です。

- 推定の途中で、別のツールを選択したり、別のインターフェースに切り替えたりした場合
- 推定プログラム実行中に非常停止または外部安全停止をトリガーした場合
- 推定プログラム実行中に自動モードから手動モードに切り替えた場合



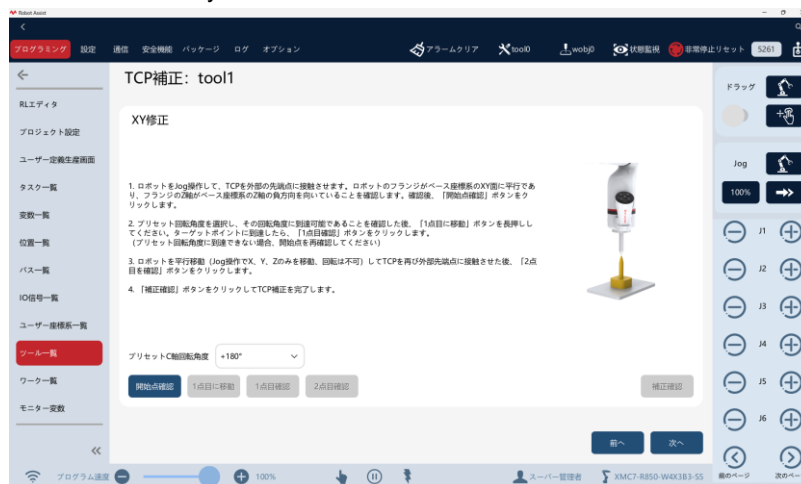
警告

推定プログラムは自動モードで行う必要があるため、すべての保護対策が有効でなければなりません。ロボットは外部制御信号によりいつでも起動する可能性があるため、設置が完了した後すべての作業員が安全な場所に退避してから自動モードに切り替えてください。

8.12.2.5 TCP 補正

xCore は TCP 補正機能を提供しており、ツール座標系の TCP を補正することができます。ツール座標系の設定が済んだハンドツールに対しては、TCP 補正機能を使用して TCP 位置精度を向上させることができます。ツール先端が変形したり、ツール取り付け位置の誤差が原因で理論上の TCP と実際の TCP の偏差が大きい場合、TCP 補正機能を使用してツール座標系の TCP 設定や迅速なキャリブレーションを行うことができます。

TCP 補正機能は現在、XY 補正と Z 補正の 2 種類のタイプをサポートしています。それぞれ、ツール座標系の TCP の x、y、z 座標パラメーターを補正するために使用されます。

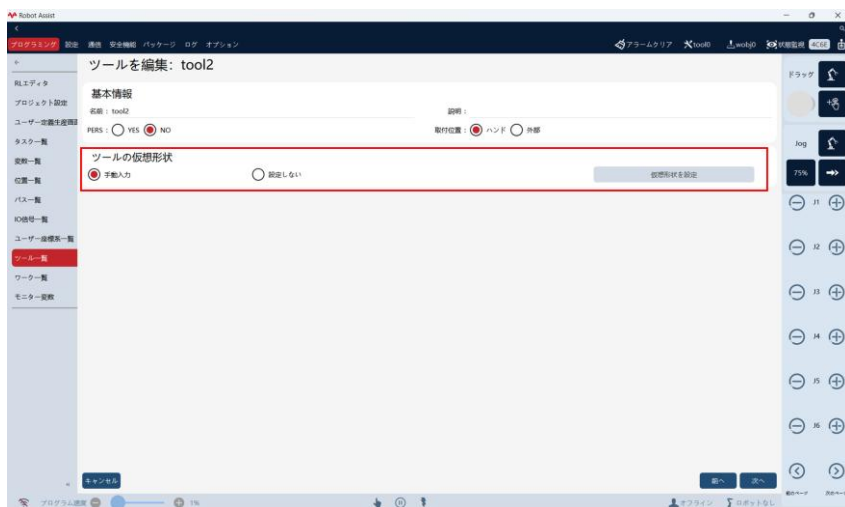


ヒント

1. TCP 補正機能は現在、ハンドツールのみをサポートしています。
2. TCP 補正機能を使用する際は、ロボットが正しい取り付け状態であり、ベース座標系はワールド座標系と一致している必要があります。
3. ツール座標系が設定済みのツールのみが TCP 補正機能を使用できます。
4. 「重力補償*」機能を有効にしている場合は、TCP 補正の精度を保証するために、正確なツールの負荷情報を設定してください。
5. TCP 補正機能は 6 軸ロボットにのみ提供されます。

8.12.2.6 ツールの仮想形状

ツールは仮想形状を設定できます。この仮想形状は、安全エリアへの侵入の判定に使用されます。形状は球体及び直方体から設定できます。



8.12.2.7 ツール座標系の使用

ジョグ中の使用：

ジョグ操作を行う際に特定のツールを使用する必要がある場合は、教示操作盤インターフェースの上部にあるクイックアクションバーの「ツール」ドロップダウンボックスから目標ツールを選択します。



RL プログラムでの使用：

プログラム内で特定のツールを使用する際は、命令文の「ツール」パラメーターで目標ツールを設定します。教示操作盤の「プログラミング補助」インターフェースを使って動作命令を書く際のデフォルトの「ツール」と「ワーク」は tool0、wobj0 です。



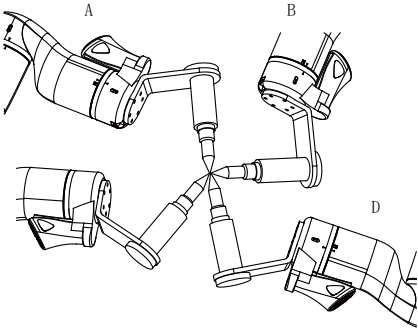
8.12.2.8 外部ツールとハンドツール

通常、ロボットに取り付けてロボットの動作と共に指定された作業を行うツールは、ハンドツールと呼ばれます。代表的なハンドツールはグリッパー、チャック、溶接ガンなどです。研磨や塗布などの特定の状況では、ツールをロボットに取り付けると作業が妨げられることがあります。このような場合は、ワークをロボットに取り付け、ツールを外部のある位置に固定することがあります。ロボットの外部に取り付けられ、ある位置に固定されて動かないツールは外部ツール（ブランドによって **Stationary Tool** や **Remote TCP**）と呼ばれ、対応するワークをハンドワークと呼びます。

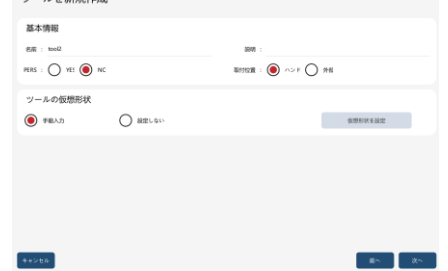
8.12.3 操作例

8.12.3.1 ハンドツールの作成

ツール座標系を設定する前に、外部に固定された先端点を準備する必要があります。この先端点は、ロボットの作業範囲内に位置し、比較的柔軟な姿勢でアクセスできることが要求されます。



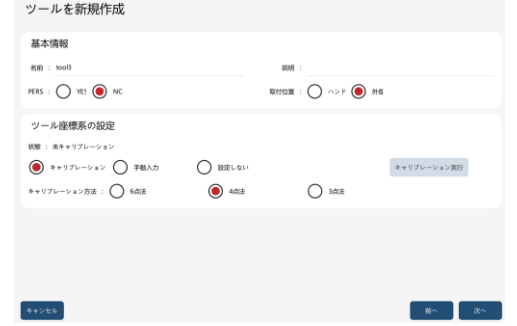
手順	図	説明
1. ツール一覧インターフェースで、右下の「+」をタップして、ツールの新規作成ウィザードに入ります。		「プログラマー」以上の権限で、新規作成・編集操作を行います。
2. ツールプロパティの設定をハンドにします。		

3. ツールの負荷設定を行います (手動入力を例に説明します)。 「手動入力」を選択し、「手動入力を実行」ボタンをタップし、「ツールの負荷情報を手動入力」ページに入ります。		ツールの物理情報を入力し、「次へ」ボタンをタップして「ツールを新規作成」ページに戻ります。この時、「状態」ステータスは「設定済」と表示されます。
4. ツール座標系の設定を行います (3点法を例に説明します)。 「キャリブレーション」を選択し、方法を「3点法」とします。「キャリブレーション実行」ボタンをタップし、キャリブレーションのページに入って設定します。		ロボットをジョグし、ツールのTCPを順に目標座標系の原点/x軸上の点/xy平面上の点またはy軸上の点に動かし、「1点目を確認、2点目を確認、3点目を確認」ボタンをタップします。3点とも確認した後、「キャリブレーションを実行」ボタンをタップして「ツールを新規作成」ページに戻ります。この時「設定状態」は「設定済み」になります。 4点法を使用してツール原点を設定する場合は、4点間の姿勢差をできる限り大きくします。即ち、ロボットができるだけ異なる姿勢で外部先端点に接触するようにする必要があります。
5. ツールの仮想形状を設定します。		
6. 「次へ」ボタンをタップしてツールの新規作成を完了します。		ツール一覧に新規作成されたツールが表示されます。

8.12.3.2 外部ツールの作成

外部ツールの設定方法はハンドツールと同様です。外部ツールの場合は、負荷設定と仮想形状の設定は不要です。

注意：外部ツールの座標系の設定は、ツール座標系が設定済みのハンドツールを使用して行う必要があります。

手順	図	説明
1. 設定済みのハンドツールを使用して外部ツールを設定します。例では tool1 を利用します。		このツールは先端がとがっている必要があります。
2. ツール一覧インターフェースで、右下の「+」をタップして、ツールの新規作成ウィザードに入ります。		「プログラマー」以上の権限で、新規作成・編集操作を行います。
3. ツールの取付位置の設定を外部にします。		
4. 右上でツールを切り替えます。設定済みのハンドツールである tool1 を選択します。		
5. 4 点法を例として説明します。 「キャリブレーション」を選択し、設定方法を「4 点法」とします。「キャリブレーション実行」ボタンをタップし、キャリブレーションのページに入って設定します。		ロボットをジョグし、設定済みのツール TCP を異なる姿勢で目標ツール座標系の原点に向けます。そして1点目、2点目、3点目、4点目をそれぞれ確認します。4点間の姿勢差は、できる限り大きくします。即ち、ロボットができるだけ異なる姿勢で外部先端点に接触するようにする必要があります。

		あります。 設定後、システムは設定エラー量をポップアップ表示します。エラーの大きさに応じて再設定が必要かどうかを選択します（設定精度の確認を参照ください）。
--	--	--

注意：

外部ツールは対応するワークと共に使用されなければなりません。即ち、同時に選択されたツールとワークの **robhold** パラメーターの一方が **False** で、もう一方が **True** でなければなりません。さもなくば、システムはエラーメッセージを出し、ロボットのジョグを無効にします。

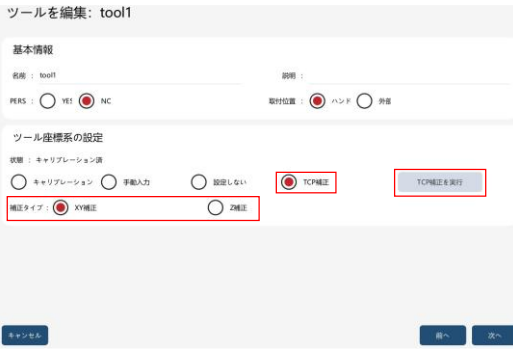
外部ツールを使用する場合、ツール座標系とワーク座標系を定義する参考系はハンドツールと異なります。詳細は下表の通りです。

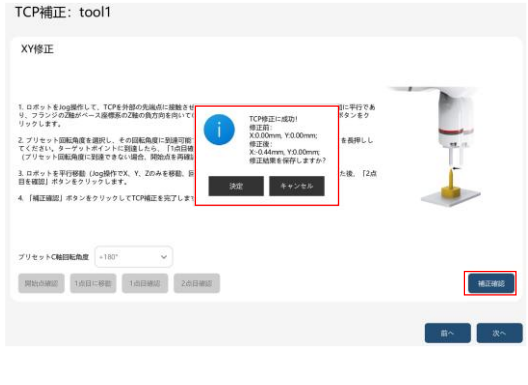
座標系の名称	ハンドツールの時は、...に対して定義される	外部ツールの時は、...に対して定義される
ワーク座標系	ユーザー座標系	ユーザー座標系
ユーザー座標系	ワールド座標系	フランジ座標系
ツール座標系	フランジ座標系	ワールド座標系

8.12.3.3 TCP 補正機能の使用

TCP 補正機能を使用する前に、固定された外部の先端点を準備する必要があります。この点はロボットの動作範囲内に位置し、かつ補正するツールが比較的柔軟な姿勢でその点に接触できることが求められます。

手順	図	説明
1. ツール一覧画面で補正する必要があるツールを選択し、右下の「編集」ボタンをタップして、ツール編集画面に進みます。		「プログラマー」以上の権限で、新規作成・編集操作を行います。 このツールは先端がとがっている必要があります。また、ツール座標系の設定と負荷の設定がされている必要があります。 TCP 補正機能はハンドツールで使用可能です。

2. 「XY 補正」を例に説明します。 ツール座標系の設定画面で、「TCP 補正」ボタンを選択し、補正タイプを選んだ後、「TCP 補正を実行」ボタンをタップして、指定されたタイプの TCP 補正画面に進みます。		TCP 補正は、「XY 補正」と「Z 補正」の2種類の補正タイプをサポートしています。 「XY 補正校正」: ツールの TCP 位置の x、y 座標を補正します。 「Z 校正」: ツールの TCP 位置の z 座標を補正します。
3. ツールの姿勢を調整し、TCP 補正の開始点を確認します。		ロボットをジョグ動作させ、ツールの先端を外部の先端に接触させます。ロボットのフランジがベース座標系の xy 平面と平行になり、フランジの z 軸がベース座標系の z 軸の負方向を指すようにします。その後、「開始点確認」ボタンをタップします。
4. プリセット回転角度を設定し、1点目に移動させます。		プリセット回転角度を選択し、その角度が到達可能であることを確認してください。その後、「1 点目に移動」ボタンを長押しして、目標地点まで動作させます。(もしプリセット回転角度で到達できない場合は、開始点を再確認してください。)
5. TCP 補正の 1 点目を確認します。		ロボットが目標点に到達した後、「1 点目確認」ボタンをタップします。

6. TCP 補正の 2 点目を確認します。		ロボットを平行移動（x、y、z 方向のみ、回転は不可）させ、ツールの先端を再び外部の先端点に接触させます。その後、「2 点目確認」ボタンをタップします。
7. 補正結果を確認し、設定します。		「補正確認」ボタンをタップして、TCP 補正を完了させます。 画面に TCP 補正結果のポップアップが表示され、「決定」ボタンをタップすると、ツールの TCP は補正後の結果に設定されます。「キャンセル」ボタンをタップすると、ツールの TCP は補正前の結果のままとなります。

8.13 ワーク一覧

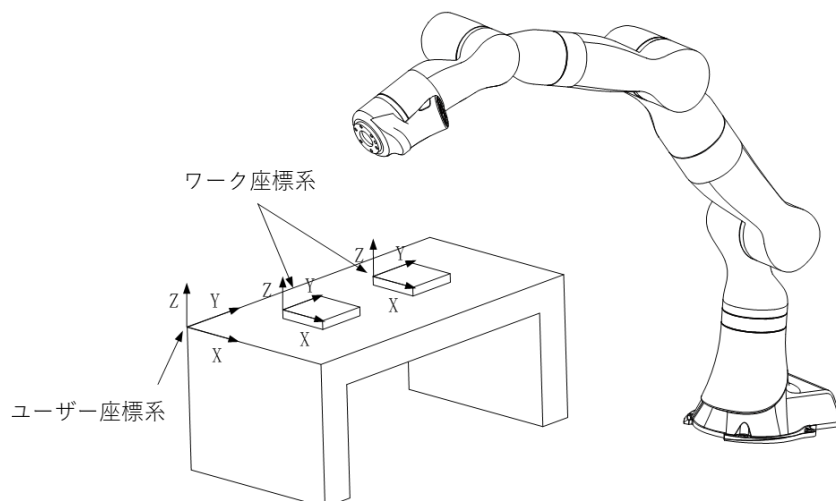
8.13.1 概要

ワークとは、ロボットがツールを使って加工する物体のことです。xCore システムでは、wobj（Work Object）型の変数を使用してワークを定義します。ワークを定義することは、wobj 型の変数を作成することを意味します。

ロボットの動作軌跡はすべてワーク座標系で定義されます。これは主に 2 つの利点があります。

- (1) ワークが移動したり複数の同じワークが加工される場合、ワーク座標系を再設定するだけで、プログラム内のすべてのパスを更新できます。
- (2) 外部軸（走行軸、ポジショナなど）で移動するワークの加工が可能です。

各ワークは、2 つの座標系によって定義されます。一つはユーザー座標系で、ワークが置かれる作業台/テーブルとして定義され、複数の同じワークを扱う時に非常に便利です。もう一つはワーク座標系で、テーブル上のワークとして定義されます。ロボットのパスは、すべてワークに対するツールの位置/姿勢として設定されます。



外部ツール機能を使用する場合、ワークがロボットに取り付けられることを意味します。そのワークをハンドワークと呼びます。ハンドワークもワーク座標系を定義する必要があります。これは、定義済みの外部ツールを使用して定義します。詳細は外部ツール機能を参照してください。



ヒント

wobj0 はシステムで事前に定義されているワークで、ユーザー座標系とワーク座標系はワールド座標系と一致します。

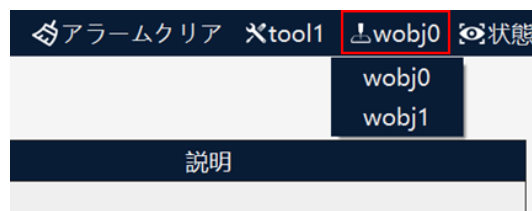
tool0 と同じように、**wobj0** の変更は許可されません。

モデルが PCB 3、4 軸ロボットの場合、ワーク座標系は手動入力のみをサポートし、姿勢 A、C の成分は強制的に 0 となります。ユーザーの手動による変更はできません。

8.13.2 ワーク座標系の使用

ジョグ中の使用：

特定のワーク座標系でジョグ操作を行う場合は、教示操作盤インターフェースの上部にあるクイックアクションバーの「ワーク」ドロップダウンボックスから目標のワークを選択します。



RL プログラムでの使用：

プログラム内で特定のワーク座標系を使用する場合は、命令文の「ワーク」パラメーターで目標ワークを指定します。教示操作盤の「プログラミング補助」インターフェースを使用して動作命令を書く場合は、デフォルトの「ツール」と「ワーク」はジョグに使用するものと同じです。即ち、インターフェース上部のクイックアクションバーで現在選択されている「ツール」と「ワーク」です。具体的な操作手順については、「命令の挿入」の項を参照してください。



ヒント

通常、動作命令のワークパラメーターはオプションであるため、指定しない場合システムはデフォルトで **wobj0** を使用します。**wobj0** はワールド座標系と一致します。

外部ツール機能を使用する場合は、使用するツールに対応するワークパラメーターを指定する必要があります。

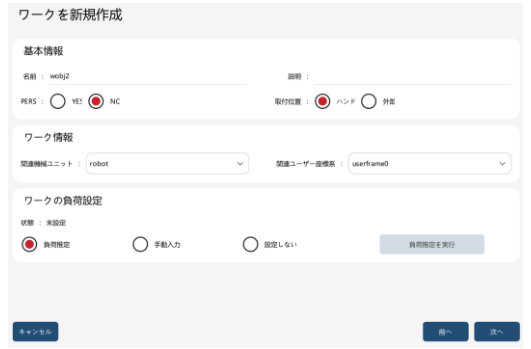
8.13.3 操作例

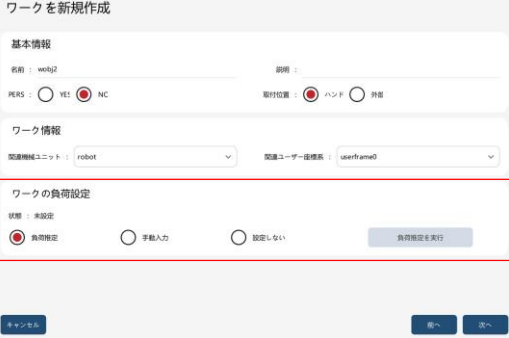
8.13.3.1 外部ワークの作成

手順	図	説明
1. ワーク一覧のインターフェースで、右下の「+」をタップして、ワークの新規作成ウィザードに入ります。		「プログラマー」以上の権限で、新規作成・編集操作を行います。
2. ワークの取付位置の設定を外部にします。		
3. 右上でツールを設定済みのハンドツール tool1 に切り替えます。		
4. 「キャリブレーション実行」ボタンをタップして、ワーク座標系キャリブレーションの設定インターフェースに入ります。		ロボットをジョグし、設定済みのツール TCP を目標座標系の原点に動かし、「1 点目を確認」ボタンをタップします。 ロボットをジョグし、TCP を目標ワーク座標系の X 軸の 1 点に動かし、「2 点目を確認」ボタンをタップします。ロボットをジョグし、TCP を目標ワーク座標系の Y 軸の 1 点に動かし、「3 点目を確認」ボタンをタップし、「キャリブレーションを実行」ボタンをタップ

		します。結果の確認ダイアログボックスがポップアップ表示されるので、「決定」をタップします。「次へ」ボタンをタップし、「ワークを新規作成」ページに戻ります。
5. 「次へ」ボタンをタップして新しいワークを作成します。		ワーク一覧に、新規作成されたワークが表示されます。

8.13.3.2 ハンドワークの作成

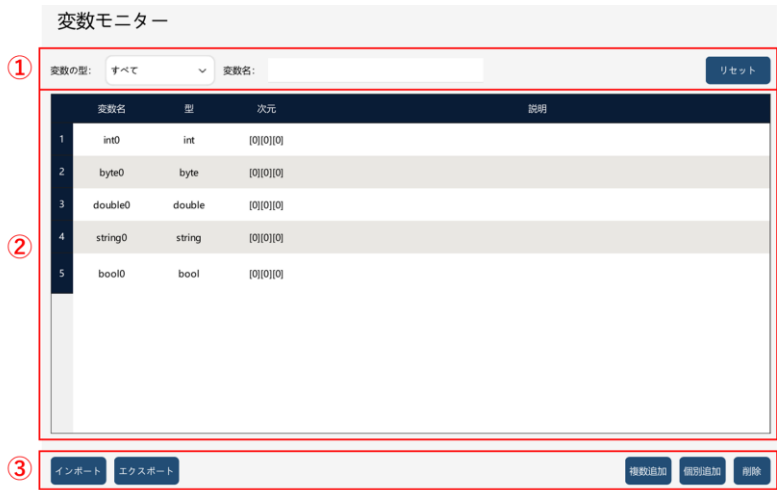
手順	図	説明
1. ワーク一覧のインターフェースで、右下の「+」をタップして、ワークの新規作成ウィザードに入ります。		「プログラマー」以上の権限で、新規作成・編集操作を行います。
2. ワークの取付位置の設定をハンドにします。		「外部」とは、ワークがロボット先端に固定されていないことを示し、「ハンド」とは、ロボット先端に固定されていることを示します。
3. 右上隅でツールを設定済みの外部ツール tool2 に切り替えます。		

4. ワークの負荷設定を行います。設定方法はツールと同じです。		
5. ワーク座標系の設定を行います。「キャリブレーション」を選択し、「キャリブレーション実行」ボタンをタップすると、設定インターフェースに入ります。		ロボットをジョグし、設定済みのツール TCP がワーク座標系の原点を指すように動かし、「1 点目を確認」ボタンをタップします。ロボットをジョグし、TCP がワーク座標系の X 軸の 1 点を指すように動かし、「2 点目を確認」ボタンをタップします。ロボットをジョグし、TCP が目標ワーク座標系の Y 軸の 1 点または XOY 平面の 1 点を指すように動かし、「3 点目確認」ボタンをタップし、「キャリブレーション実行」ボタンをタップします。結果の確認ダイアログボックスがポップアップ表示された後、「決定」をタップします。「次へ」ボタンをタップし、「ワークを新規作成」ページに戻ります。
6. 「次へ」ボタンをタップして新しいワークを作成します。		ワーク一覧に、新規作成されたワークが表示されます。

8.14 変数モニター

8.14.1 概要

変数モニター画面は、監視が必要な変数を追加するためのものです。

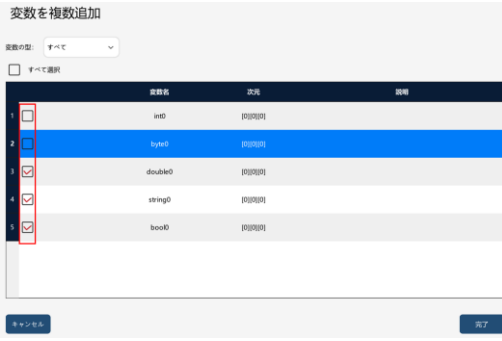


①	フィルタ設定： 一覧に表示したい変数をフィルタリングするために使用します。
②	変数一覧： 各変数の変数名、型、次元、説明を表示します
③	機能ボタンセクション： 変数モニターのインポート、エクスポート、複数追加、個別追加、削除を行います。

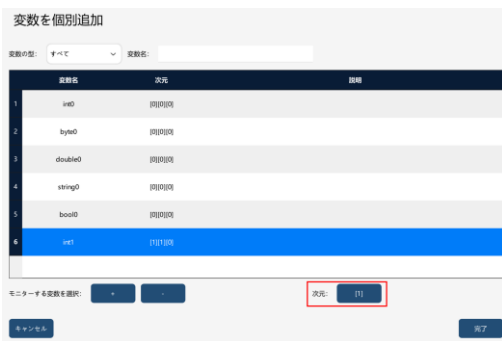
8.14.2 操作例

8.14.2.1 変数の複数追加

手順	図	説明
1. 「複数追加」ボタンをタップします。		

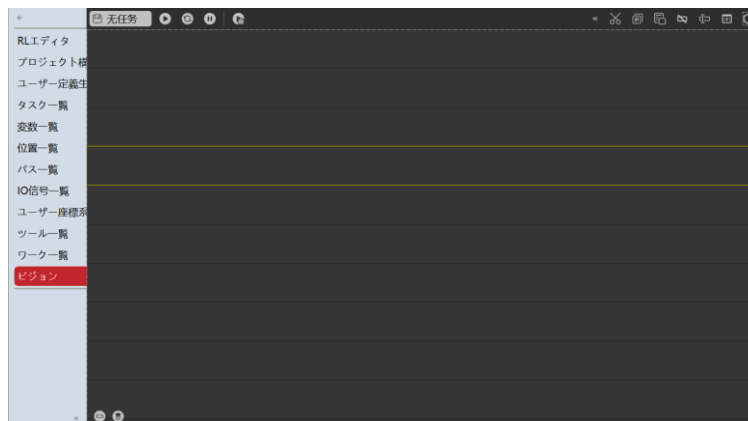
2. 監視する変数を選択し、赤枠内にチェックを入れた後、「完了」ボタンをタップします。		注意：変数が配列の場合、複数追加後は配列内のすべての要素が監視されます。
3. 状態監視の変数モニターで、変数の状態を監視できるようになります。		

8.14.2.2 変数の個別追加

手順	図	説明
1. 「個別追加」ボタンをタップします。		
2. 監視する変数を選択し、「完了」ボタンをクリックします。選択した変数が配列の場合は、次元を選択して指定の次元の変数を監視することができます。		次元は変数が配列の場合にのみ操作できます。
3. 状態監視の変数モニターで、変数の状態を監視できるようになります。		

8.15 ビジョン

ビジョンタスクのプログラミングは、RL 動作タスクのプログラミングと同じレベルで扱うことができます。「プログラミング」-「ビジョン」を通じてビジョンタスクの編集インターフェースを開きます。



ビジョンタスクも「タスク」の一種であり、「タスク」をタップすると、ビジョンタスクを表示できます。また、ビジョンタスクの新規作成、編集、名前の変更も行えます。

「自動起動」機能を「実行する」にチェックを入れると、プロジェクトがロードされた際、ビジョンタスクは自動的に読み込まれ、ループして実行されます。

注意：ビジョンタスクは1つしかチェックできません。



ビジョン機能の使用方法については、xVision のマニュアルをご参照ください。

8.16 RL プログラムについて

8.16.1 RL プログラムの書式と構文

8.16.1.1 概要

RL 言語プログラムファイルの拡張子は.mod です。mod は module (モジュール) の略です。例えば、MoveObj.mod または PickSomething.mod は、プログラムファイルごとに 1 つのプログラムモジュールを構成します。

RL 言語の命令は、大文字と小文字が区別されません。例えば、MoveAbsJ、moveabsj、MOVEABJSJ の 3 つの書き方はすべて同一と識別されます。一貫した言語スタイルを保持するために、単語の頭文字を大文字にする形式を使用することが推奨されます。

8.16.1.2 プログラムの構造

以下は簡単な RL プログラムです。

```

1 GLOBAL CONST double pi = 3.1415926;
2 GLOBAL VAR double z = 0;
3
4 GLOBAL PROC main() // main関数
5   MoveJ robtarget3,v1000,z50,tool0
6   Socket()
7 ENDPROC
8
9 GLOBAL PROC Socket() //Socket関数
10  SocketClose("sc")
11  wait 0.1
12  if(SocketCreat("192.168.0.11",9797,"sc"))
13    | print("Socket creat success")
14  else
15    | pause
16  endif
17 ENDPROC

```

プログラムは、宣言セクションと実装セクションの2つに分けられます。各 Mod ファイルの最初の関数の前の部分が宣言セクションです。例えば、main.mod の GLOBAL PROC main の前の部分は宣言セクションです。宣言セクションでは変数や定数を定義できます。このセクションで定義された変数は、プログラムが実行されるたびに初期値にリセットされます。VAR または CONST はタイプを指定します。VAR は可変変数であり、CONST は定数です。変数のタイプが明示的に宣言されていない場合、デフォルトで可変変数となります。

宣言セクションの変数と変数一覧の変数には、次のような違いがあります。

あるタスクが終了してリセットされると、宣言セクションで定義された変数はリセットされます。

変数一覧の変数はプロジェクト全体に属するもので、グローバル変数です。変数一覧内の PERS 変数ではない変数は pptomain の時にリセットされ、PERS 変数は変数一覧の編集機能または RL プログラムによってのみ変更できます。

8.16.2 プログラムのデバッグ

8.16.2.1 プログラムポインター

プログラムポインターは、プログラムが解析され実行された行番号を指します。

HMI インターフェースでは、プログラムポインターは小さな緑の矢印（グリーンポインターとも呼ばれる）で表示されます。

```

6 GLOBAL PROC main() // main関数
7   MoveJ robtarget3,v1000,z50,tool0
8   Socket()
9   wait 2
10  MoveJ robtarget4,v1000,z50,tool0
11 ENDPROC
12 int ww=2
13 GLOBAL PROC Socket() //Socket関数
14  SocketClose("sc")
15  wait 0.1
16  if(SocketCreat("192.168.0.11",9797,"sc"))
17    | print("Socket creat Success")
18  else
19    | pause

```


8.16.2.2 実行ポインター

実行ポインターは、ロボットが実際に動作している命令を示します。

HMI インターフェースでは、実行ポインターは赤い記号で表示されます。

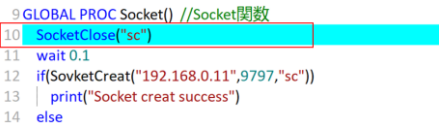
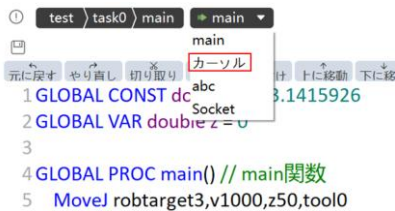
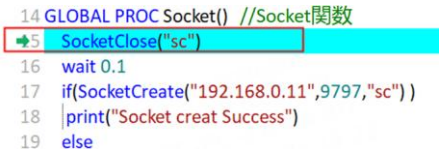
```

6 GLOBAL PROC main() // main関数
7   MoveJ robtarget3,v1000,z50,tool0
8   Socket()
9   wait 2
10  MoveJ robtarget4,v1000,z50,tool0
11 ENDPROC
12 int ww=2
13 GLOBAL PROC Socket() //Socket関数
14   SocketClose("sc")
15   wait 0.1
16   if(SocketCreate("192.168.0.11",9797,"sc"))
17     print("Socket creat Success")
18   else
19     pause

```

8.16.2.3 プログラムポインターの移動

プログラムを途中の行から実行する必要がある場合、この機能を使ってプログラムポインターをカーソルのある行に移動させ、その位置からプログラムを実行することができます。

手順	図	説明
1. 操作パネルの「停止」ボタンをタップし、RL プログラムを一時停止させます。		
2. ポインターを移動させたい行をタップして選択状態にします。		選択状態の背景色は水色です。
3. ドロップダウンリストをタップし、リストから「カーソル」を選択します。		選択すると、ボタンには「カーソル」と表示されます。
4. 「カーソル」をタップすると、ポインターを選択行に移動できます。		

注意事項：

- この機能を使用する場合、以下の命令は無視されコンパイラのコンパイル位置は直接ターゲット行に移動します。それ以外の命令は実行されません。
 - すべての動作命令
 - SetDO、SetGO、Return、Wait、Print 及びすべての Socket 命令
 - 機能呼び出し行
- プログラムポインターを移動する時、プロセス制御命令の判定条件を無視します。
- 関数間でのプログラムポインターの移動はできません。「関数へのプログラムポインター

の移動」機能を使って、プログラムポインタを指定の関数の最初に移動させてから、プログラムポインタの移動機能を使う必要があります。

4. プログラムポインタは動作命令行にしか移動できません。

8.16.2.4 シングルステップデバッグ

シングルステップ動作状態は、シングルステップモードとも呼ばれます。これに対するモードは連続モードです。ほとんどの場合、ロボットはシングルステップモードと連続モードを切り替えることができます。

シングルステップ動作は主にプログラムデバッグに使用されます。ロボットは 1 行の命令を実行し、命令完了後にプログラムを一時停止します。各行の教示が要件を満たしているかどうかを簡単に確認できます。デバッグ対象がマルチタスクのプロジェクトの場合、シングルステップデバッグは現在の HMI 画面に表示されているタスクのみを実行し、それ以外のタスクは呼び出されません。

シングルステップデバッグがデータ読み出し命令（ReadDouble、ReadString など）、時間関連命令（Wait、WaitUntil など）、論理命令（IF、GOTO など）を実行する場合、命令特性の制限により命令行を完了するまでに「次ステップ」を 2～3 回タップする必要があります。

手順	図	説明
1. ポインタを指定する行に移動します。	<pre> 6 GLOBAL PROC main() // main関数 7 MoveJ robtarget3,v1000,z50,tool0 8 Socket() 9 wait 2 10 MoveJ robtarget4,v1000,z50,tool0 11 ENDPROC 12 int ww=2 </pre>	
2. 操作パネルの「次ステップ」ボタンをタップします。		
3. 「次ステップ」ボタンを再度タップすると、ロボットが当該行を実行します。実行後、もう一度「次ステップ」ボタンをタップすると、次のステップを実行します。	<pre> 6 GLOBAL PROC main() // main関数 7 MoveJ robtarget3,v1000,z50,tool0 8 Socket() 9 wait 2 10 MoveJ robtarget4,v1000,z50,tool0 11 ENDPROC </pre>	関数呼び出し行で「次ステップ」を実行すると、関数内にジャンプします。プログラムポインタ（緑の矢印）で実行中の行を知ることができます。

注意事項：

1. 連続モードのプログラム自動実行は、滑らか軌跡エリアや動作の先読みができます。
2. シングルステップモードは、滑らか軌跡エリアや動作の先読みをせず、命令を直接実行します。
3. 連続モードでは、先読みが十分になってはじめて動作し、到着した後にのみ命令の解析を続けます。
4. シングルステップモードでは、次のステップの実行はすべてインターフェースによってトリガーされます。滑らか軌跡エリアを処理せず、先読みも実行されません。
5. シングルステップモードでは、動作中に次ステップをタップしても応答しません。
6. 連続モードでは、動作中のコールバックは先読みの論理に従って応答します。
7. シングルステップモードでは、いかなる行にも到着でき文字通り実行します。RL プログ

ラムでは、「プログラム命令」のみが存在するとみなされ、動作命令と論理命令は区別されません。

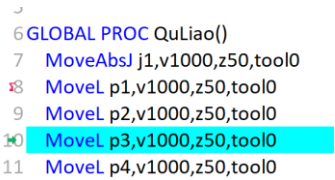
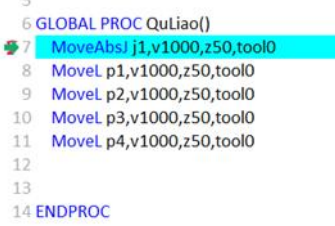
8. 連続動作中に滑らか軌跡エリアで一時停止した場合、次ステップは、現在の滑らか軌跡エリアに対応する目標位置に戻るまで先に進みません。

8.16.2.5 ステップバックデバッグ

ステップバックデバッグ（または前ステップデバッグ）は、デバッグ中に軌跡のエラーを発見した場合に、最後の正しい位置に直接戻ることができる機能です。これにより、エラー軌跡からロボットを出すためにジョグを使用する必要がなく、デバッグ効率が向上します。非動作命令は、ステップバックの過程では無効となり、スキップされます。力制御命令、動作設定命令、論理命令は、ステップバックを停止させます。

8.16.2.5.1 ステップバックデバッグの使用方法

xCore は、動作停止後のステップバックと ppto カーソル後のステップバックを切り替えることをサポートしています。ppto カーソル後の最初のステップバック動作時は動作情報がないため、直交動作は強制的に MoveJ 動作に変換されます。

手順	図	説明
1. 実行中に一時停止	 <pre> 6 GLOBAL PROC QuLiao() 7 MoveAbsJ j1,v1000,z50,tool0 8 MoveL p1,v1000,z50,tool0 9 MoveL p2,v1000,z50,tool0 10 MoveL p3,v1000,z50,tool0 11 MoveL p4,v1000,z50,tool0 </pre>	先読みは 10 行目まで進んでいますが、ロボットは j1 から p1 への軌道上にいます。
2. 右側の操作パネルにある「前ステップ」ボタンをタップします。	 <pre> 6 GLOBAL PROC QuLiao() 7 MoveAbsJ j1,v1000,z50,tool0 8 MoveL p1,v1000,z50,tool0 9 MoveL p2,v1000,z50,tool0 10 MoveL p3,v1000,z50,tool0 11 MoveL p4,v1000,z50,tool0 </pre>	先読みのポインタが動作ポインタの上に移動し、ステップバックモードに入ることを示します。
3. 再度「前ステップ」ボタンをタップすると、ロボットがステップバック動作を開始します。	 <pre> 6 GLOBAL PROC QuLiao() 7 MoveAbsJ j1,v1000,z50,tool0 8 MoveL p1,v1000,z50,tool0 9 MoveL p2,v1000,z50,tool0 10 MoveL p3,v1000,z50,tool0 11 MoveL p4,v1000,z50,tool0 12 13 14 ENDPROC </pre>	ロボットは、停止時の動作方法（この例では MoveL）に従って、先読みポインタが指し示す j1 に戻ります。

8.16.2.5.2 ステップバック機能の使用制限

ステップバック機能は主に位置のデバッグに使用されるため、ほとんどの非動作命令はステップバックモードでは無効または制限されて使用されます。また、いくつかの動作命令は原理的にステップバックができないため、使用が制限されています。制限された命令をステップバックで実行しようとする、コントローラーがエラーを報告します。

例：

```

1 GLOBAL PROC main()
2   QuLiao()
3
4 ENDPROC
5
6 GLOBAL PROC QuLiao()
7   MoveAbsJ j1,v1000,z50,tool0
8   MoveL p1,v1000,z50,tool0
9   MoveL p2,v1000,z50,tool0
10  MoveL p3,v1000,z50,tool0
11  AccSet 100,30
12  MoveL p4,v1000,z50,tool0
13
14
15 ENDPROC

```

ステップバックデバッグ機能でサポートされる命令の状況：

種類	命令
ステップバックに対応している動作命令	MoveAbsJ, MoveJ, MoveL, MoveC, Home
特殊な動作命令	SearchL, SearchC, TrigL, TrigJ, TrigC (注 1)
ステップバックに対応していない動作命令	MoveSP, MoveT, MoveCF
ステップバックを終了する非動作命令	コンベア命令、力制御命令、論理命令、関数呼び出し、高級命令 (注 2)
スキップされる非動作命令	残りのすべての命令

注 1： 探索、Trig などの機能はステップバック時に効果がなく、動作効果のみが発生します。

注 2： GetRobotState、BreakLookAhead、GetRobotMaxLoad 命令はスキップされ、このカテゴリに属する他の命令はステップバックを終了させます。

8.16.2.6 軌跡の復帰

特定の状況では、ロボットの位置がプログラムされた軌跡から逸脱することがあります。

例：

- プログラム実行中でロボットが停止している間（プログラムリセットによるプログラム停止は含まれない）に、ロボットをジョグした場合
- プログラム実行中に非常停止がトリガーされ、ロボットが STOP 0 を実行した場合

プログラムが停止位置から再び起動され、ロボットがプログラムされた軌跡から逸脱したことを検出すると、ロボットは元のプログラムされた軌跡に戻るために軌跡復帰動作を実行します。

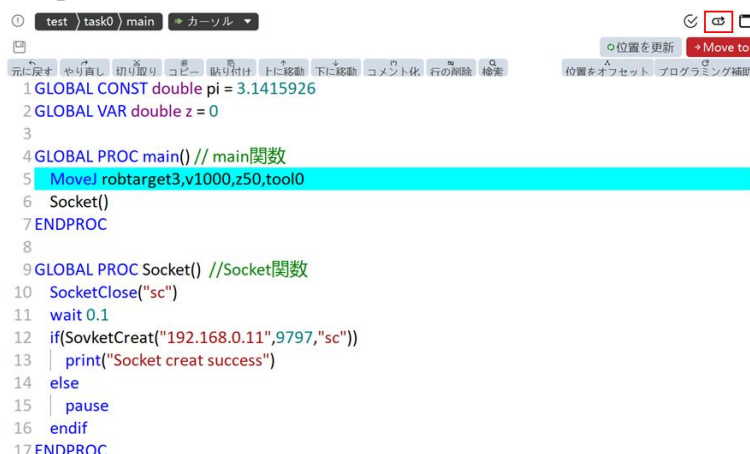
安全性を確保するために、ロボットは軌跡復帰の時にはゆっくりと動きます。また、教示操作盤の「停止」ボタンを押すことで、いつでもその動作を停止することができます。

注意事項：

1. 軌跡復帰時は、ロボットは軸動作を実行するため、経路が予測できません。周辺環境との衝突に注意してください。
2. ロボットがプログラムの途中で停止した場所から実行し続けるときのみ、制御システムは軌跡から逸脱したかどうかを検出し、逸脱している場合は軌跡復帰動作を実行します。
3. プログラムがリセットされた場合、システムは軌跡から逸脱しても検出しません。最初の行から実行が再開されますので、衝突の可能性に注意してください。

8.16.2.7 ループモード

「ループモード」ボタンをタップして  ループモードまたは  単回モードを切り替えます。



```

1 GLOBAL CONST double pi = 3.1415926
2 GLOBAL VAR double z = 0
3
4 GLOBAL PROC main() // main関数
5   MoveJ robtarget3,v1000,z50,tool0
6   Socket()
7 ENDPROC
8
9 GLOBAL PROC Socket() //Socket関数
10  SocketClose("sc")
11  wait 0.1
12  if(SocketCreat("192.168.0.11",9797,"sc"))
13    print("Socket creat success")
14  else
15    pause
16  endif
17 ENDPROC
  
```

ループモード：

すべてのタスクは指定関数（デフォルトは main 関数）の ENDPROC に到達すると、0.5 秒後にリセットされます。リセット後は、関数の最初の行から再実行します。

PPToMain 操作は main 関数を指定された関数とします。

PPToFunc はジャンプした対象の関数を指定関数とします。

PPToLine は次のループで実行する関数の対象に影響を与えません。

単回モード：

すべてのタスク（自動実行タスクを除く）は指定関数（デフォルトは main 関数）の ENDPROC を実行した後、プロジェクトが再度ロードされるまで停止し続けます。

8.16.2.8 先読み機能

8.16.2.8.1 基本概念

先読み機能をオフにすることはできません。プログラムが実行されると、システムは自動的に先読みを行います。プログラムポインター（Program Pointer）で先読みの位置を確認することができます。先読みの観点から見ると、RL 命令は、動作命令、先読み継続命令、滑らか軌跡エリア実行命令、先読み停止命令の 4 つに分類されます。

8.16.2.8.2 動作命令

ロボットが実際の動作を生成するすべての「動作命令」、「Trigger 命令」、「ドラッグ再生命令」、及び「Home」命令はすべて動作命令に分類されます。命令の詳細は RL プログラミング命令を参照してください。

8.16.2.8.3 先読み継続命令

この命令は、先読みポインターが解析した直後に実行され、その後先読みポインターは前後の 2 つの動作命令の間の滑らか軌跡に影響を与えずに、次の命令を解析するために次の行に移動します。

先読み継続命令：print 命令、論理判定命令、変数に値を割り当てる操作、ユーザー定義関数、衝突検知動的閾値命令、運動パラメーターを動的に変更する命令；

例：

```
MoveL p1
IF (condition_1)
  Print(「要件 1 を満たす」)
MoveL p2
ENDIF
MoveL p3
```

上記のプログラムを実行すると、要件 1 が満たされた場合、ロボットは p1→p2→p3 の連続した軌跡動作を計画し、print 命令の先読みで達すると「要件 1 を満たす」という文字列を印刷します。

要件 1 が満たされない場合、ロボットは p1→p3 の連続した軌跡動作を計画します。

8.16.2.8.4 滑らか軌跡エリア実行命令

滑らか軌跡エリア命令の直前の動作命令が実行されると、当該命令が実行されます。

ロボットがどの動作命令に到達したかを周辺機器に送信するために使用されます。

滑らか軌跡エリア実行命令：WriteRegByName, SetDO, SetGO, SetAO, PulseDO, PulseReg, InZone, SetVarValue

例：

```
MoveL P19
MoveL P20
WriteRegByName reg_position, 20
MoveL P21
WriteRegByName reg_position, 21
MoveL P22
```

上記のプログラムを実行すると、ロボットは P19→P20→P21→P22 の連続した軌跡動作を計画します。ロボットが P19 の動作を完了し P20 に移動しようとするとき、数字 20 がレジスタ reg_position に書き込まれます。ロボットが P20 の動作を完了し、P21 に移動しようとするとき、数字 21 がレジスタ reg_position に書き込まれます。周辺機器はこのレジスタを読み取ることで、ロボットの動作プロセスを知ることができます。



ヒント

滑らか軌跡エリア命令のトリガータイミングは、ロボット本体の性能、エリアの大きさ、エリアの有無、実際の動作速度などに影響されます。ロボットがどの位置に向かっているのかにのみ注目するシナリオに適しています。軌跡の特定の位置で信号をトリガーする必要がある場合は、trigger シリーズの命令を使用する必要があります。

8.16.2.8.5 先読み停止命令

上記の 3 つの命令以外の命令はすべて先読み停止命令です。コントローラーはロボットがその前のすべての動作を完了した後に当該命令を実行します。

例：

```
MoveL P1
MoveL P2
Wait 10
MoveL P3
```

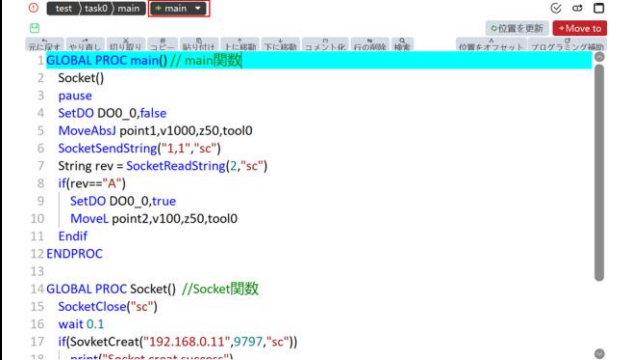
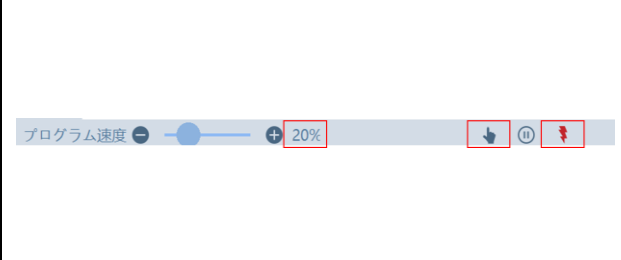
Wait コマンドは先読み停止命令です。ロボットが P2 に動作して減速停止した後、10 秒間待機します。待機が終わると再び P3 に動作し始めます。

動作命令で滑らか軌跡エリアを Fine に設定すると、RL プログラムの予測ポインターはその行の動作命令で停止し、動作が終了するのを待ってから、予測を再開します。

例：

```
MoveAbsJ j1,v1000,z50,tool0
MoveL p1,v1000,z50,tool0
MoveL p2,v1000,Fine,tool0
// これより後のすべての命令は、MoveL P2 が終了するまで実行されません
MoveL p3,v1000,z50,tool0
```

8.16.3 デバッグの例

手順	図	説明
1. RL プログラムを編集した後、pptomain を使用してポインターを main 関数に移動し、プログラムの構文エラーをチェックします。		デバッグを容易にするために、プログラムに pause 命令を挿入することができます。
2. モードを「手動モード」に切り替え、モーターの電源を入れます。プログラム速度を小さい値に調整し、ロボットを低速で動作させます。		
3. 操作パネルの「実行」ボタンをタップします。		

4. プログラムは pause を実行すると、「次ステップ」ボタンまたは「実行」ボタンがタップされるまで一時停止します。	<pre> 1 GLOBAL PROC main() // main関数 2 Socket() 3 pause 4 SetDO DO0_0,false 5 MoveAbsJ point1,v1000,z50,tool0 6 SocketSendString("1,1","sc") 7 String rev = SocketReadString(2,"sc") 8 if(rev=="A") 9 SetDO DO0_0,true 10 MoveL point2,v100,z50,tool0 11 Endif 12 ENDPROC 13 14 GLOBAL PROC Socket() //Socket関数 15 SocketClose("sc") 16 wait 0.1 17 if(SocketCreate("192.168.0.11",9797,"sc")) 18 print(SocketReadString(2,"sc")) 19 </pre>	「次ステップ」ボタンをタップと、プログラムが 1 行実行されます。
5. 「次ステップ」ボタンをタップすると、シングルステップデバッグを行います。	<pre> 1 GLOBAL PROC main() // main関数 2 Socket() 3 pause 4 SetDO DO0_0,false 5 MoveAbsJ point1,v1000,z50,tool0 6 SocketSendString("1,1","sc") 7 String rev = SocketReadString(2,"sc") 8 if(rev=="A") 9 SetDO DO0_0,true 10 MoveL point2,v100,z50,tool0 11 Endif 12 ENDPROC 13 14 GLOBAL PROC Socket() //Socket関数 15 SocketClose("sc") 16 wait 0.1 17 if(SocketCreate("192.168.0.11",9797,"sc")) 18 print(SocketReadString(2,"sc")) 19 </pre> 	プログラムが SetDO 行を実行します。状態監視を開いて DO が「オフ」かどうかをチェックします。
6. ステップで実行を続け、ロボットが目標ポイントに到達するかどうかを確認します。	<pre> 1 GLOBAL PROC main() // main関数 2 Socket() 3 pause 4 SetDO DO0_0,false 5 MoveAbsJ point1,v1000,z50,tool0 6 SocketSendString("1,1","sc") 7 String rev = SocketReadString(2,"sc") 8 if(rev=="A") 9 SetDO DO0_0,true 10 MoveL point2,v100,z50,tool0 11 Endif 12 ENDPROC </pre>	
7. カーソルをある行に移動して「次ステップ」ボタンをタップし、命令が正常に実行されるかどうかを確認できます。	<pre> 1 GLOBAL PROC main() // main関数 2 //Socket() 3 pause 4 SetDO DO0_0,false 5 MoveAbsJ point1,v1000,z50,tool0 6 SocketSendString("1,1","sc") 7 String rev = SocketReadString(2,"sc") 8 if(rev=="A") 9 SetDO DO0_0,true 10 MoveL point2,v100,z50,tool0 11 Endif 12 ENDPROC 13 14 GLOBAL PROC Socket() //Socket関数 15 SocketClose("sc") 16 wait 0.1 17 if(SocketCreate("192.168.0.11",9797,"sc")) 18 print(SocketReadString(2,"sc")) 19 </pre> 	プログラムが SetDO 行を実行します。状態監視を開いて DO が「オン」かどうかをチェックします。
8. プログラムに問題がないことを確認した後、再度 pptomain を行います。「プログラム速度」は小さいままにしておきます。プログラムを連続動作で一度完全に実行させた後で、プログラム速度を上げて「自動モード」に切り替えてプログラムを実行します。		

9 設定

9.1 本章の概要

本章では、xCore 制御システムの各種設定項目について詳しく説明します。

9.2 コントローラー設定

9.2.1 基本設定

9.2.1.1 システム情報

システム情報

バージョン : 3.0.0.C89.20241228.1	ロボット型式 : XMC7-R850-W4X3B3-S5	MACアドレス : ff:ff:ff:ff:ff:ff	再起動	シャットダウン
①	②	③	④	

①	バージョン：制御システムのバージョン情報です
②	ロボット型式：ロボットの機種情報です
③	MAC アドレス：MAC アドレスです
④	再起動： 再起動前に全ての設定情報を保存するように注意してください。 シャットダウン： 制御キャビネットの電源を切った後、再度電源を入れることでコントローラソフトウェアを再び起動できます。

9.2.1.2 システム構成

システム構成

ロボット型式 : XMC7-R850-W4X3B3-S5	①	安全モジュール : ROKAE_RSC	③
制御キャビネット型式 : XBC_XMATE	②	確定	

①	ロボット型式：ロボットの型番を選択します
②	安全モジュール：安全モジュールのタイプを選択します
③	制御キャビネット型式：制御キャビネットの型式を選択します

9.2.1.3 システム時間

システム時間は、ログなどの機能のために基準時間を提供します。

システム時間

①	2024 年 6 月 25 日 9 時 12 分 57 秒
②	コントローラーから時間を取得
③	現在時刻に変更
④	確定

①	日時：コントローラーの時間です。リアルタイムでは更新されません。
②	コントローラーから時間を取得： タップするとコントローラーの時間が更新および表示されます。
③	現在時刻に変更： コントローラーのシステム時間を RobotAssist ソフトウェアが搭載されている機器の現在のシステム時間に更新します。「確定」をタップする必要はありません。

④	①の時間は手動で変更できます。その場合は、「確定」をタップしてコントローラーの時間を確定します。
---	--



警告

1. システム時間はログ情報の絶対的な基準時間ですので、安易に変更しないでください。システム時間を誤って変更すると、ユーザーはログで関連イベントの発生時刻を追跡できなくなります。
2. この操作は頻繁に実行しないでください。「コントローラー時間の取得」と「現在時刻に変更」の2つの操作の間、または同じ操作の実行間隔は5秒以上あけてください。

9.2.1.4 ネットワーク情報

ここでは外部ネットワークポートの接続モード、IP アドレス、サブネットマスクなどを設定できます。

ネットワーク情報

① 名前: ens33 モード: 手動 ②

③ IPアドレス: 192.168.21.9 ゲートウェイ: 192.168.21.2 サブネットマスク: 255.255.255.0 DNSサーバー:

確定

①	名前: 自動取得したネットワークカード名を表示します。
②	モード: 手動 (IP アドレスを手動で設定) または自動 (IP アドレスを自動割り当て) を選択します。
③	IP アドレス、ゲートウェイ、サブネットマスク、DNS サーバーの情報を設定します。

注意: デバッグポートの IP アドレスは、192.168.0.160 または 169.254.160.160 にのみ変更できます。デバッグポートの IP アドレスが 192.168.0.160 の場合は、他のネットワークポートの IP アドレスは 192.168.0 ネットワークセグメントに変更できません。デバッグポートの IP アドレスが 169.254.160.160 の場合は、他のネットワークポートの IP アドレスは 192.168.0 ネットワークセグメントに変更できますが、192.168.0.160 には変更できません。

9.2.2 詳細設定

基本設定 詳細設定 認証設定

エンコーダーのマルチターン: エンコーダーのマルチターン値をクリア

エイリアス: 確定

ログ保存レベル: info, warning, error 確定

パルス周期: 1 確定

手動モードでのプログラム速度上限[1% ~ 100%]: 100 確定

自動モードでのプログラム初期速度上限[1% ~ 100%]: 100 確定

Jog速度上限[1% ~ 100%]: 100 確定

非常停止トリガーレベル設定: ☒ High ☐ Low 確定

エンコーダーのマルチターン	エンコーダーのマルチターン値をクリアします。慎重に実行してください。
エイリアス	同一 LAN 上の複数のロボット設備を区分するために、コントローラーにオリジナルの名前を設定します。エイリアスは「オプション」→「接続」→「ロボット検出」画面に表示されます。
ログ保存レベル	ログレベルは低いレベルから高いレベルまで「情報」、「警告」、「エラー」の 3 段階に分けられ、それぞれ「info」、「warning」、「error」と表示されます。特定のレベルのログの保存を設定できます。設定されたレベル以下のログは表示のみとなり、履歴が残りません。
パルス周期	「1 つの High + 1 つの Low」がパルス周期となります。High と Low の継続時間は等しくなります。パルス信号は、システム IO を介して DO 信号に割り当てる（詳しくは「システム IO」の章を参照）か、または「sta_heartbeat」機能コードを介してレジスタに割り当てることができます（詳しくは「レジスタ」の章を参照）。
手動モードでのプログラム速度上限	このパラメーターは、スライダーと「-/」微調整ボタンによる手動モードで設定可能なプログラム速度の上限を制限します。 例：この値を 20% に設定した後、自動モードでプログラム速度が 40% になった場合 動作モードが自動モードから手動モードに切り替わると、プログラム速度は自動的に 20% に設定されます。自動モードの場合は、プログラム速度はこのパラメーターの影響を受けません。 注意： 1. レジスタ機能コード（ctrl_set_program_speed）または SDK を通じてプログラム速度を設定する時、ロボットが手動モードの場合は実際のプログラム速度はこのパラメーターによって制限されます。自動モードの場合は、実際のプログラム速度はレジスタ機能コードまたは SDK の設定値になります。 2. 自動モードから手動モードに切り替える時（切替え方法：HMI 上のモード切替えボタン、レジスタ機能コード、システム IO、外部通信、SDK など）は、プログラム速度はこのパラメーターによって制限されます。
自動モードでのプログラム初期速度上限	自動モードのプログラムはこのパラメーター設定した値で実行され、実行の安全性を高めることができます。実行開始後、ユーザーは実際の状況に応じてプログラム速度を調整できます。 例：このパラメーターを 5% に設定した後、自動モードのプログラム速度が 60% の場合 プログラム速度は自動的に 5% に設定されます。自動モードでプログラム速度が 3% の場合は、プログラム速度は 3% のままとまります。
Jog 速度上限	ロボットのジョグ速度はこの上限値以上に調整できません。
非常停止トリガーレベル設定	非常停止がトリガーされた後、レジスタ、システム出力、外部通信、非常停止状態の出力/ステータスピットが「High」または「Low」のどちらになるかを決定します。

停止エリア復帰 遅延時間	停止エリア安全状態からの復帰遅延時間を設定できます。
協働エリア復帰 遅延時間	協働エリア状態からの復帰遅延時間を設定できます。
ロボット停止状 態出力遅延時間	ロボットが停止したと判定されてから出力するまでの時間を設定で きます。
ロボット停止状 態判定閾値	ロボットが停止したと判断するための閾値を設定できます。
停止エリア判定 遅延時間	ロボットが停止エリアに入った後、判定を始めるまでの時間を設定 できます。

9.2.3 認証設定

9.2.3.1 EtherCAT 認証

EtherCAT認証

認証コード：

-

-

認証

認証コードを通じて EtherCAT 通信を認証します。



ヒント

認証コードが無効または認証に失敗した場合は、モーターの電源をオンにできません。

9.2.3.2 機能認証

xCore システムは、以下の 2 種類のオプションソフトウェア機能をサポートしています。

1. 言語：日本語、韓国語、ロシア語、英語
2. パッケージ：ソーラーバッテリー部品組み立て/レーザー溶接のプロセスパッケージ、
SDK 二次開発インターフェース

オプション機能を有効にする場合は、有効にしたい機能の認証コードファイルを購入してください。

注意： 認証情報はロボットのコントローラーに紐づけされ、ロボット 1 台につき 1 ファイルです。

アクティベーション方法：

「設定」-「コントローラー設定」-「認証設定」-「機能認証」で、「参照」をタップし、認証ファイルを選択して認証します。認証が成功した場合は、インターフェースのプロンプトに従って RobotAssist を再起動してください。対応するオプション機能を有効にすることができます。

認証の有効性：

機能認証は一度認証すれば継続使用できます。コントローラーのバージョンアップ、設定消去、工場出荷時設定へのリセットなどの操作は認証状態に影響を与えません。

認証済み機能：現在のロボットの認証された機能を表示します。

機能認証

認証ファイル：

参照

認証済み機能：日本語, 韓国語, ロシア語, 英語, ソーラー挿入, ソーラーパターン, レーザー, SDK

関連内容：

「HMI 設定」 - 「基本設定」 - 「言語」では、認証された言語のみが表示されます。

基本設定

基本設定の変更：これらの設定の大部分は、起動時に読み込まれます。変更した後は、手動で再起動してください。

言語：日本語

ロボット検索用IPアドレス：任意

リフレッシュレート (Hz)：18

3D表示をオフにする：☐

9.3 HMI 設定

9.3.1 基本設定

基本設定

基本設定の変更：これらの設定の大部分は、起動時に読み込まれます。変更した後は、手動で再起動してください。

① 言語：日本語

② ロボット検索用IPアドレス：任意

③ リフレッシュレート (Hz)：18

④ 3D表示をオフにする：☐

①	言語： 中国語、英語、日本語、韓国語、ロシア語をサポートしています。 注意： 1. デフォルト言語は中国語です。他の言語を使用する場合は、ROKAE の正式な認証を得る必要があります。 2. xCore V1.7 以降のバージョンでは、多言語ログ機能をサポートしています（「コントローラーログ」にのみ有効）。HMI の言語タイプを切り替えると、HMI の言語内容に従ってコントローラーの言語も切り替わります。HMI とコントローラーの言語設定が異なる場合は、コントローラーは接続が確立されたときに HMI に対応する言語モードに切り替わります。言語を切り替えた後にコントローラーを再起動する必要はありません。このモードはホットスイッチングですが、一部の詳細なログ情報は切り替え後にのみ有効になり、言語切り替え前に生成されたログは対応する言語に切り替わらない場合があります。
②	ロボット検索用 IP アドレス： 「オプションー接続」のロボット検出機能により、PC の固定 IP アドレスによる検出を行うかどうかを設定できます。
③	リフレッシュレート： 状態監視の 3D 表示画面の更新頻度です。上限は 100Hz、デフォルトは 18Hz です。
④	3D 表示をオフにする： 状態監視の 3D 表示をオフにします。再起動すると有効になります。

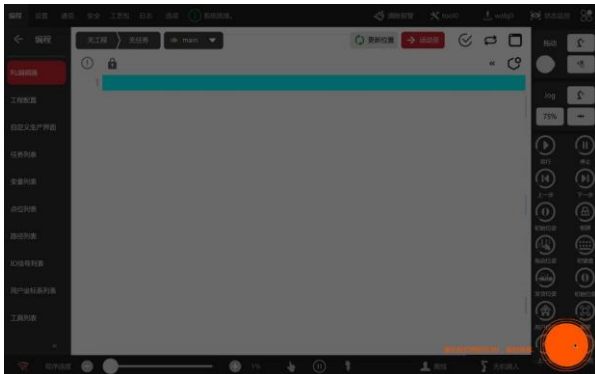
示教器设置

① IP: 192 . 168 . 0 . 123 确定

② 锁屏: ☐ 空闲时间: 999 确定

③ 截屏: 导出 清空

④ 是否显示光标: ☒

①	IP (xPad2 のみ) : ① 教示操作盤とロボットを接続するネットワークカードの静的 IP アドレスを設定します。
②	画面ロック (xPad2 のみ) : この機能により、関係者以外の者が誤って教示操作盤に触れることを防ぎます。一定時間 (アイドルタイム) 教示操作盤を操作しない、または画面ロックのショートカットキーをタップすると、教示操作盤は下図のような画面ロックの状態になります。  画面ロックページの丸いボタンを 3 秒長押しするか、または再度画面ロックのショートカットキーをタップすると、画面のロックが解除されます。
③	スクリーンショット (xPad2 のみ) : ③ 現在の教示操作盤の画面をキャプチャし、教示操作盤のディレクトリに JPG の画像形式で保存します。
④	カーソル表示の有無 : ④ 教示操作盤上のカーソルの有無を設定します。

スタイル

スタイルの切り替え: 異なるスタイルに切り替えることができます。

サイズの調整: UIのフォント、ウィジェット、アイコン、レイアウトのサイズを調整します。

①

スタイルの選択 : デフォルトのスタイル

②

サイズ調整 : 150%

OK

ワークスペース保存先

ワークスペース: すべてのプロジェクト関連ファイルはこのフォルダに保存されます。プログラムが当該フォルダに対して完全な読み取り・書き込み権限を持っていることを確認してください。

③

ワークスペース保存先 : ./workspace

参照

OK

権限タイムアウト時間

④

時間[0-30min]: 10

0にすると、この機能は無効になります

保存

位置パラメーターをチェック

⑤

精度: 0.10

+/- (0° - 30°)

保存

①	スタイルの選択: 表示スタイルを選択します。
②	サイズ調整: ウィジェットやフォントのサイズを調整します。
③	ワークスペース保存先: プロジェクトファイルの保存場所を設定します。
④	権限タイムアウト時間: 設定されたタイムアウト時間を超えて操作が行われない場合、ログインしている権限が自動的にログアウトされ、オペレーターに切り替わります。デフォルトは10分です。0に設定するとこの機能は無効になります。
⑤	位置パラメーターをチェック: RL エディタの「位置を表示」機能の判定精度パラメーターです。その位置にいるかどうかを判定する角度範囲を設定できます。デフォルトは1°です。

9.3.2 教示操作盤モード



手順	図	説明
1. この画面で教示操作盤の現在のモードを表示します。デフォルトは「教示操作盤有り」です。「モード切替え」ボタンをタップすると、「教示操作盤なし」に切り替わります。		教示操作盤有り： 教示操作盤の画像がカラーで表示され、コネクタが接続されています 教示操作盤なし： 教示操作盤の画像がグレーで表示され、コネクタが切断されています
2. 「教示操作盤なし」に切り替えた後、xPad2 教示操作盤とロボットとの物理的接続を切断できます。（「ロボットシステムの構成と接続」の章を参照。）		このときは、ロボットは非常停止状態に入りません。

xPad2 教示操作盤をロボットに再接続する必要がある場合は、次の手順に従ってください。

手順	図	説明
1. xPad2 教示操作盤とロボットを物理的に接続します。（「ロボットシステムの構成と接続」の章を参照。）		
2. この画面に現在表示されている教示操作盤のモードは「教示操作盤なし」です。「モード切替え」ボタンをタップすると、「教示操作盤有り」に切り替わります。		教示操作盤有り： 教示操作盤の画像がカラーで表示され、コネクタが接続されています 教示操作盤なし： 教示操作盤の画像がグレーで表示され、コネクタが切断されています

 ヒント

1. 一部の機種のみホットスワップ教示操作盤の機能をサポートしています。この機能をサポートしていない機種では、教示操作盤なしモードへの切り替えができません。具体的な対応機種については、弊社にお問い合わせください。

2. 教示操作盤なしモードは、教示操作盤を取り外す必要がある場合にのみオンにすることをお勧めします。また、デバッグ中や使用中にオンにしないでください。

9.4 ユーザーグループ

xCore システムには、操作権限の低い順に、オペレーター（Operator）、教示者（Teacher）、プログラマー（Programmer）、管理者（Admin）、スーパー管理者（System） の 5 つのユーザーレベルが設定されています。

コントローラーに接続すると、デフォルトではオペレーターとしてログインします。他の権限に切り替えるにはパスワードの入力が必要です。

ユーザーログイン

ユーザーレベル：

教示者

パスワード：

ログイン

オペレーター：

プログラムの実行とバグ報告。

教示者：

教示作業。

プログラマー：

プログラム編集。

管理者：

ロボット設定の変更。

スーパー管理者：

すべての権限。

パスワードの変更

ユーザーレベル：教示者

古いパスワード：

新しいパスワード：

新しいパスワードの確認：

変更

ユーザー	デフォルトパスワード	権限の説明
オペレーター	パスワードなし	プログラムの実行、バグの報告
教示者	123	ポイント教示
プログラマー	1234	プログラム編集
管理者	12345	ロボット設定の変更
スーパー管理者	123456	すべてのロボット制御権限

注意：

- 高い権限を持つユーザーは、同じレベルまたは低いレベルのユーザーのパスワードを変更できます。
- オペレーターのパスワードは変更できません。
- 高レベルから低レベルへのユーザー切り替えにはパスワードは必要ありません。

各ユーザーグループの権限の詳細は、付録を参照してください。

9.5 キャリブレーション

xCore システムは、ゼロキャリブレーション、ソフトウェアキャリブレーション（産業用ロボット）、力センサーキャリブレーション（協働ロボット）の機能を提供します。キャリブレーションは単軸ごとに実行することも全軸同時に実行することもできます。

9.5.1 ゼロキャリブレーション

ここで言うゼロキャリブレーションとは、メカニカルゼロキャリブレーションを指し、ロボットの理論上のゼロ点と実際の機械のゼロ点を一致させることを目的とします。ロボット本体にはゼロ点マークがあらかじめ設けられています。各関節のマーク位置を合わせ、キャリブレーションを実行することができます。

ゼロキャリブレーション
力センサーキャリブレーション
ソフトウェアキャリブレーション
ベース座標系設定

ゼロキャリブレーションの目的は、制御アルゴリズムの理論上のゼロ点と実際の機械のゼロ点を一致させ、ロボットが制御システムの指令に正しく応答できるようにすることです。

角度設定は、現在の角度が既知である場合に、その角度を入力することにより機械ゼロ点のキャリブレーションと同じ効果を得ることができます。

ゼロキャリブレーション		前回のエンコーダー値	角度設定 (°)	前回の角度 (°)
軸1 :	0	単軸キャリブレーション	0	0
軸2 :	0	単軸キャリブレーション	0	0
軸3 :	0	単軸キャリブレーション	0	0
軸4 :	0	単軸キャリブレーション	0	0
軸5 :	0	単軸キャリブレーション	0	0
軸6 :	0	単軸キャリブレーション	0	0

全軸キャリブレーション
確定

ゼロキャリブレーションでは、誤操作による失敗を防ぐため、「単軸キャリブレーション」ボタンまたは「全軸キャリブレーション」ボタンをクリックした後、認証コードを入力し、確定ボタンを押すことでキャリブレーション操作が有効になります。

ゼロキャリブレーション
力センサーキャリブレーション
ソフトウェアキャリブレーション
ベース座標系設定

ゼロキャリブレーションの目的は、制御アルゴリズムの理論上のゼロ点と実際の機械のゼロ点を一致させ、ロボットが制御システムの指令に正しく応答できるようにすることです。

角度設定は、現在の角度が既知である場合

ゼロキャリブレーション		前回のエンコーダー値	角度設定 (°)	前回の角度 (°)
軸1 :	0	単軸キャリブレーション	0	0
軸2 :	0	単軸キャリブレーション	0	0
軸3 :	0	単軸キャリブレーション	0	0
軸4 :	0	単軸キャリブレーション	0	0
軸5 :	0	単軸キャリブレーション	0	0
軸6 :	0	単軸キャリブレーション	0	0

全軸キャリブレーション
確定

軸4が機械的ゼロ位置にあることを確認してください。そうでない場合、機械的ゼロ位置が失われる可能性があります！問題がないことを確認した後、次の確認コードを入力し、確定ボタンをクリックしてゼロキャリブレーションを実行してください。

jxldbd4

確定
キャンセル



警告

機械ゼロ点はむやみにキャリブレーションしないでください。キャリブレーション前にメカニカルキャリブレーション用ブロックゲージを使用してロボットの各関節がゼロ点にあることを確認してください。

レーザートラッカーでのゼロ点データが失われロボットの精度に影響を与えるため、レーザートラッカーでロボットをキャリブレーションした後は、ロボットのメカニカルゼロキャリブレーションを行わないでください。ロボットのゼロ点が失われた場合は、弊社にゼロ点データ復旧をご依頼ください。

ロボットが空間の制限により機械原点に戻れない場合は、「角度設定」機能を使用することができます。この機能を使用するには、キャリブレーションをする時のロボットの関節角度を把握する必要があります。その角度を「角度設定」に入力しキャリブレーションを行うことで、ゼロキャリブレーションと同じ効果が得られます。

xMate ER7 Pro ロボットを例として説明します。

障害物があり（下図の赤線）、ロボットが機械ゼロ点である垂直の状態に到達できない場合（下左図）は、角度設定を使用してゼロキャリブレーションを行うことができます。

まずはジョグでロボットを下図右のような状態に調整してください。このときの4軸は90度です。次に、「角度設定」に現在の角度情報（4軸は90度、それ以外は0度）を入力します。その後、キャリブレーションを行います。



注意：上記の例では、キャリブレーション時の姿勢にかかわらず、ロボットのゼロ点は垂直の状態です。そのため、角度設定で4軸に90度を入力してキャリブレーションを完了した後、ロボットをゼロ点に動作させると、ロボットは機械ゼロ点である垂直の状態に向かって動作し、障害物に衝突します！

この機能は、現在の角度ではなく、ゼロ点をキャリブレーションすることに注意してください。

9.5.2 ソフトウェアキャリブレーション

ソフトウェアキャリブレーション機能とは、エンコーダーバッテリーの電圧不足、バッテリーの取り外し、またはマルチターンの誤クリアなどの異常操作によりゼロ点が失われた後、ロボットがゼロ点を素早く復元できる機能です。

この機能を使用する前に、手動でロボットをゼロ点にジョグさせる（キャリブレーション溝が整列している）必要があります。この機能は任意の角度でゼロ点を復元することはできません。

ソフトウェアキャリブレーションは産業用ロボット専用であり、協働ロボットはこの機能をサポートしていません。

ゼロキャリブレーション カセンサキャリブレーション ソフトウェアキャリブレーション ベース座標系設定

ソフトウェアキャリブレーション機能により、多回転のクリアやエンコーダーの電圧不足アラームによって回転カウントやゼロ点が失われた場合に、素早くゼロ点を復元できます。

ソフトウェアキャリブレーション ゼロ点のエンコーダー値

軸1 :	単軸キャリブレーション	0
軸2 :	単軸キャリブレーション	0
軸3 :	単軸キャリブレーション	0
軸4 :	単軸キャリブレーション	0
軸5 :	単軸キャリブレーション	0
軸6 :	単軸キャリブレーション	0

全軸キャリブレーション

ソフトウェアキャリブレーションの操作手順は次の通りです。

1. 手でロボットをゼロ点にジョグさせます（キャリブレーション溝が整列している）。
2. 「メインメニュー - 設定 - ゼロキャリブレーション - ソフトウェアキャリブレーション」インターフェースに入ります。
3. 「全軸キャリブレーション」ボタンをタップし、ポップアップウィンドウを確認後、ゼロ点を復元できます。「ゼロ点のエンコーダー値」は、各軸のゼロ点のエンコーダー値が表示されます。
4. 各軸に対応する「単軸キャリブレーション」ボタンをタップし、ポップアップウィンドウを確認後、単軸のゼロ点を復元することもできます。

9.5.3 力センサーキャリブレーション

本機能は、xMate シリーズ協働ロボット専用です。ロボットを長期使用すると、トルクセンサーはゼロ点ドリフトを引き起こします。ロボットの動作が不安定になるなどの問題が発生することがあります。

ゼロキャリブレーション カセンサキャリブレーション ソフトウェアキャリブレーション ベース座標系設定

力覚センサーのキャリブレーションを行う際は、負荷の設定が実際の負荷と一致していることを確認し、負荷情報の精度をできる限り確保してください。各軸の力覚センサーは個別、または全軸同時にキャリブレーションできます。

動的キャリブレーションをオンにすると、ドラッグスイッチをオンにすることで力覚センサーが自動的にキャリブレーションされます。負荷情報が不正確な場合、この機能をオンにすることはお勧めしません。

動的キャリブレーション: ☒ 有効化

	トルク		前回キャリブレーション時の電圧値 (mV)
軸1 :	0	単軸キャリブレーション	0
軸2 :	0	単軸キャリブレーション	0
軸3 :	0	単軸キャリブレーション	0
軸4 :	0	単軸キャリブレーション	0
軸5 :	0	単軸キャリブレーション	0
軸6 :	0	単軸キャリブレーション	0
軸7 :	0	単軸キャリブレーション	0

全軸キャリブレーション

このような問題が発生した場合は、ロボットを適切な位置にジョグし（機械ゼロ点位置にする必要はありません）、負荷設定を確認した後、単軸キャリブレーションまたは全軸キャリブレーションを実行します。

任意の位置で力センサーのキャリブレーションを行いたい場合は、「動的キャリブレーション」スイッチをオンにし、その後、単軸キャリブレーションまたは全軸キャリブレーションを実行します。この方法でのキャリブレーション精度は、ロボットのゼロ点位置で実施した場合と比べて劣る可能性があります。また、動的キャリブレーション機能をオンにした後、ドラッグ機能や RL プログラムで力制御をオンにすると、システムは自動的にゼロ点をリセットし、力制御関連機能が正常に動作するようにします。



警告

動的キャリブレーションの使用には次のようなリスクがあります。

1. ドラッグをオンにした時、ロボットが環境と接触している（自由でない状態にある）場合は、キャリブレーションされたゼロ点に大きな誤差があるため、トルクの計算に異常が生じ力制御に失敗する可能性があります。
2. ドラッグ機能や RL で力制御をオンにした場合、現在のセンサトルクが理論モデルから大きくずれていることを示す動的キャリブレーションエラーが発生する場合があります。その場合は、次の点を確認してください。
 - (1) 負荷が正しく設定されているか
 - (2) ベース座標系が正しく設定されているか
 - (3) 機械ゼロ点が大きくずれていないか
 - (4) 機能をオンにする時に外部環境と直接的な力の接触がないか
3. ロボットのゼロ点が失われているなどの状況では、力センサーキャリブレーションに失敗することがあります。このような場合は、**CalibSensorError RL** コマンドを実行してキャリブレーションを完了できます。

上記のリスクがあるため、トルクセンサーのゼロ点ドリフトが深刻ではない限り、動的キャリブレーション機能をオンにしないでください。この機能はデフォルトでオフになっています。

9.5.4 ベース座標系設定

ベース座標系はロボットベースの中心に位置し、ワールド座標系に対して相対的に記述され、ワールド座標系に対するロボットの位置/姿勢を定義します。ロボットの天吊り設置、壁掛け設置、マルチロボット協調などのシナリオでは、ロボットの過負荷、ジッター、力制御不良などの異常現象を防ぐために、通常はベース座標系を設定する必要があります。

ベース座標系の設定方法には、手動入力と 6 点法キャリブレーションの 2 種類があります。

9.5.4.1 手動入力

ゼロキャリブレーション カセンサキャリブレーション ソフトウェアキャリブレーション **ベース座標系設定**

設定モード ☒ 手動入力 ☐ キャリブレーション

ベース座標系のキャリブレーションを行うには、動的フィードフォワードと動力学制約をオフにする必要があります。キャリブレーション終了後は、オンに戻してください。オンにしないと、ロボットの動作が不安定になります。

手動入力

注: 入力が間違っていると、予期せぬ衝突が発生する可能性があります。データが正しいかどうかを確認してください。

位置 (mm): X: 0 Y: 0 Z: 0

姿勢: ☒ オイラー角 (°) ☐ クォータニオン

設置方法: カスタマイズ A: 0 B: 0 C: 0

確定

ワールド座標系に対するベース座標の位置と姿勢を手動で入力します。姿勢はオイラー角またはクォータニオンを選択することができます。

あらかじめ設けられた 2 つの設置方法（床置き、天吊り）を選択する場合は、姿勢値を自動で設定することができます。「カスタム」設置方法を選択する場合は、手動入力することができます。

9.5.4.2 キャリブレーション（6 点法）

手順	図	説明
1. ツールを 1 つ設定します。		
2. 床置き設置以外の場合は、動的フィードフォワードをオフにします（設定 → 動力学設定 → 動的フィードフォワード）。		
3. ベース座標系の設定モードは「キャリブレーション」を選択します。		
4. 補助点の位置を定義します。この手順はオプションです。		ロボットがワールド座標系から遠く離れ、ツールの先端が届かない場合は、補助点によってベース座標系を

		設定することができます。 補助点の位置はワールド座標系に対して定義します。
5. HMI に表示される手順に従い、6 点の設定を完成し、「OK」をタップします。	6点法 1. ロボットをジョイントでCPを外部座標系に接続させた後、「[5点法確認]」ボタンをタップします。 2. ロボットが指定する位置や姿勢を保持し、補助点をジョイント、[5点法確認]、[4点法確認]、[3点法確認]、[2点法確認]、[1点法確認] ボタンをタップします。必ず正しい補助点を指定する前に、その箇所で姿勢を定める動作が必要です。 3. ロボットを指定姿勢でジョイント、TCPをツール座標系に接続し、補助点を指定します。[5点法確認] ボタンをタップします。 4. ロボットを指定姿勢でジョイント、TCPをXYZ座標系に接続し、補助点を指定します。[6点法確認] ボタンをタップします。 5点法確認 6点法確認 1点法確認 2点法確認 3点法確認 4点法確認 5点法確認 6点法確認 OK	
6. 設定結果に応じて、ベース座標系データを保存するかどうかを選択します。		

9.6 座標系設定

9.6.1 グローバルツール一覧

9.6.1.1 概要

グローバルツールと通常のツールの定義内容は同じです。xCore 制御システムでは、グローバルツールに対応するデータ型は **g_tool_num** です。num の範囲は 0～15 で、合計 16 個のグローバルツールが定義できます。グローバルツールも **tool** 型に属します。tool 型の詳細については、「RL 指令-変数」セクションを参照してください。

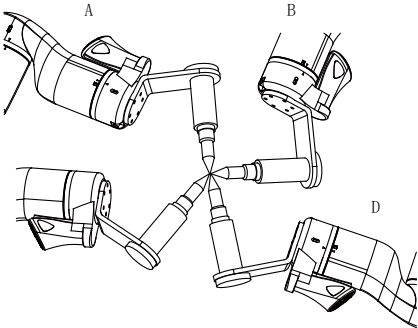
注意点：グローバルツールはプロジェクト内のツールとは異なります。グローバルツールはすべてのプロジェクトで使用することができます。

9.6.1.2 基本概念

基本概念はツール一覧と同じです。詳細は 8.12.2 章を参照してください。

9.6.1.3 操作例 グローバルハンドツールの作成

ツール座標系を設定する前に、外部に固定された先端点を準備する必要があります。この先端点は、ロボットの作業範囲内に位置し、比較的柔軟な姿勢でアクセスできることが要求されます。



手順	図	説明
1. 座標系設定 - グローバルツール一覧画面で、編集する必要があるグローバルツールを選択します。		「プログラマー」以上の権限で、編集操作を行えます。編集機能のみが利用でき、追加や削除はできません。
2. ツールプロパティの設定をハンドにします。		
3. ツールの負荷設定を行います（手動入力を例に説明します）。 「手動入力」を選択し、「手動入力を実行」ボタンをタップし、手動入力のページに入ります。		ツールの物理情報を入力し、「次へ」ボタンをタップして元のページに戻ります。この時、「状態」ステータスは「設定済」と表示されます。
4. ツール座標系の設定を行います（3点法を例に説明します）。 「キャリブレーション」を選択し、方法を「3点法」とします。「キャリブレーション実行」ボタンをタップし、キャリブレーションのページに入って設定します。		ロボットをジョグし、ツールのTCPを順に目標座標系の原点/x軸上の点/xy平面上の点またはy軸上の点に動かし、「1点目を確認」、「2点目を確認」、「3点目を確認」ボタンをタップします。3点とも確認した後、「キャリブレーションを実行」ボタンをタップして元のページに戻ります。この時「状態」は「設定済」になります。 4点法を使用してツール原点を設定する場合は、4点間の姿勢差をできるだけ大きくします。即ち、ロボットができるだけ異なる姿勢で外部先端点に接触するようにする必要があります。

5. ツールの仮想形状を設定します。		
6. 「次へ」ボタンをタップしてグローバルツールの編集を完了します。		グローバルツール一覧に編集が完了したツールが表示されます。

9.6.1.4 操作例 グローバル外部ツールの作成

グローバル外部ツールの設定方法はグローバルハンドツールと同様です。外部ツールの場合、負荷設定と仮想形状の設定は不要です。

注意：グローバル外部ツールの設定は、ツール座標系を設定済みのハンドツールを使用して行う必要があります。

手順	図	説明
1. 設定済みのハンドツールを使用して外部ツールを設定します。例では <code>g_tool_0</code> を利用します。		このツールは先端がとがっている必要があります。
2. グローバルツール一覧画面で編集したいツールを選択し、右下の「編集」ボタンをタップして、グローバルツールウィザードに進みます。		「プログラマー」以上の権限で、編集操作を行えます。

3. ツールの取付位置の設定を外部にします。		
4. 右上でツールを切り替えます。設定済みのハンドツールである <code>g_tool_0</code> を選択します。		
5. 4点法を例として説明します。 「キャリブレーション」を選択し、設定方法を「4点法」とします。「キャリブレーション実行」ボタンをタップし、キャリブレーションのページに入って設定します。		ロボットをジョグし、設定済みのツール TCP を異なる姿勢で目標ツール座標系の原点に向けます。そして1点目、2点目、3点目、4点目をそれぞれ確認します。4点間の姿勢差は、できる限り大きくします。即ち、ロボットができるだけ異なる姿勢で外部先端点に接触するようにする必要があります。 設定後、システムは設定エラー量をポップアップ表示します。エラーの大きさに応じて再設定が必要かどうかを選択します（設定精度の確認を参照ください）。

注意：

外部ツールは対応するワークと共に使用されなければなりません。即ち、同時に選択されたツールとワークの `robhold` パラメーターの一方が `False` で、もう一方が `True` でなければなりません。さもなくば、システムはエラーメッセージを出し、ロボットのジョグを無効にします。

外部ツールを使用する場合、ツール座標系とワーク座標系を定義する参考系はハンドツールと異なります。詳細は下表の通りです。

座標系の名称	ハンドツールの時は、...に対して定義される	外部ツールの時は、...に対して定義される
ワーク座標系	ユーザー座標系	ユーザー座標系
ユーザー座標系	ワールド座標系	フランジ座標系
ツール座標系	フランジ座標系	ワールド座標系

9.6.2 グローバルワーク一覧

9.6.2.1 概要

グローバルワークと通常のワークの定義内容は同じです。詳細は 8.13.1 章を参照してください。xCore 制御システムでは、グローバルワークに対応するデータ型は **g_wobj_num** です。num の範囲は 0～15 で、合計 16 個のグローバルワークが定義できます。グローバルワークも wobj 型に属します。wobj 型の詳細については、「RL 指令-変数」セクションを参照してください。

注意点：グローバルワークはプロジェクト内のワークとは異なります。グローバルワークはすべてのプロジェクトで 사용할 ことができます。

9.6.2.2 操作例 グローバル外部ワークの作成

グローバル外部ワークの座標系をキャリブレーションするには、キャリブレーション済みのハンドツールを使用する必要があります。

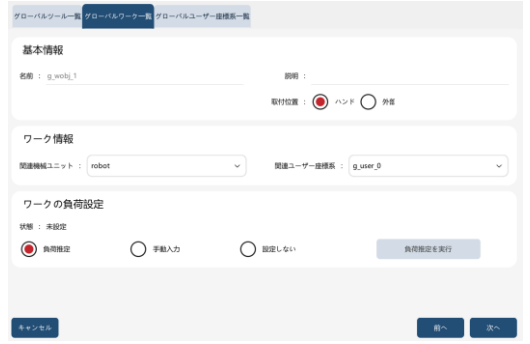
手順	図	説明
1. グローバルワーク一覧で編集したいワークを選択し、右下の「編集」ボタンをタップして、グローバルワーク編集ウィザードに進みます		「プログラマー」以上の権限で、編集操作を行えます。
2. グローバルワークの取付位置の設定を外部にします。		
3. 右上でツールを設定済みのハンドツール g_tool_0 に切り替えます。		

4. 「キャリブレーション実行」ボタンをタップして、グローバルワークの座標系キャリブレーションの設定インターフェースに入ります。		ロボットをジョグし、設定済みのツール TCP を目標座標系の原点に動かし、「1 点目を確認」ボタンをタップします。 ロボットをジョグし、TCP を目標ワーク座標系の X 軸の 1 点に動かし、「2 点目を確認」ボタンをタップします。 ロボットをジョグし、TCP を目標ワーク座標系の Y 軸の 1 点に動かし、「3 点目を確認」ボタンをタップし、「キャリブレーションを実行」ボタンをタップします。結果の確認ダイアログボックスがポップアップ表示されるので、「決定」をタップします。「次へ」ボタンをタップし、グローバルワークの編集ページに戻ります。
5. 「次へ」ボタンをタップしてグローバルワークの編集を完了します。		グローバルワーク一覧に編集が完了したワークが表示されます。

9.6.2.3 操作例 グローバルハンドワークの作成

グローバルハンドワークの座標系をキャリブレーションするには、キャリブレーション済みの外部ツールを使用する必要があります。

手順	図	説明
1. グローバルワーク一覧で編集したいワークを選択し、右下の「編集」ボタンをタップして、グローバルワーク編集ウィザードに進みます		「プログラマー」以上の権限で、編集操作を行います。

2. ワークの取付位置の設定をハンドにします。		「外部」とは、ワークがロボット先端に固定されていないことを示し、「ハンド」とは、ロボット先端に固定されていることを示します。
3. 右上隅でツールを設定済みの外部ツール g_tool_1 に切り替えます。		
4. ワークの負荷設定を行います。設定方法はツールと同じです。		
5. ワーク座標系の設定を行います。「キャリブレーション」を選択し、「キャリブレーション実行」ボタンをタップすると、設定インターフェースに入ります。		ロボットをジョグし、設定済みのツール TCP がワーク座標系の原点を指すように動かし、「1 点目を確認」ボタンをタップします。ロボットをジョグし、TCP がワーク座標系の X 軸の 1 点を指すように動かし、「2 点目を確認」ボタンをタップします。ロボットをジョグし、TCP が目標ワーク座標系の Y 軸の 1 点または XOY 平面の 1 点を指すように動かし、「3 点目確認」ボタンをタップし、「キャリブレーション実行」ボタンをタップします。結果の確認ダイアログボックスがポップアップ表示された後、「決定」をタップします。「次へ」ボタンをタップし、グローバルワークの編集ページに戻ります。

6.
「次へ」ボタンをタップして
グローバルワークの編集を完了します。

ワーク一覧

名前	PECS	実行位置	可動	座標座標系	キャリブレーション	座標設定	備考
work1	NO	外部	固定	userframe0	済	なし	
work2	NO	ハンド	固定	userframe0	済	なし	

キャリブレーション結果 なし

+
←
→

9.6.3 グローバルユーザー座標系一覧

9.6.3.1 概要

グローバルユーザー座標系は、ワーク座標系やグローバルワーク座標系を定義する際に参照座標系として使用され、単独では使用されません。

グローバルツール一覧
グローバルワーク一覧
グローバルユーザー座標系一覧

名前	キャリブレーション	説明
1	g_user_0	未キャリブレーション
2	g_user_1	未キャリブレーション
3	g_user_2	未キャリブレーション
4	g_user_3	未キャリブレーション
5	g_user_4	未キャリブレーション
6	g_user_5	未キャリブレーション
7	g_user_6	未キャリブレーション
8	g_user_7	未キャリブレーション
9	g_user_8	未キャリブレーション
10	g_user_9	未キャリブレーション
11	g_user_10	未キャリブレーション

キャリブレーション結果 なし

↑
↓
↶

グローバルユーザー座標系を作成するには、「キャリブレーション」、「手動入力」、または「設定しない」を選択します。

「キャリブレーション」を選択すると、3点法を使って設定します。グローバルユーザー座標系を設定する前に、1つのツールを設定しておき、そのツールのTCPを使用してユーザー座標系を設定する必要があります。先のとがったツールを推奨します。

事前にグローバルユーザー座標系がわかっている場合は、手動で入力することもできます。

「設定しない」を選択した場合は、ユーザー座標系はワールド座標系と一致します。

9.6.3.2 グローバルユーザー座標系のキャリブレーション

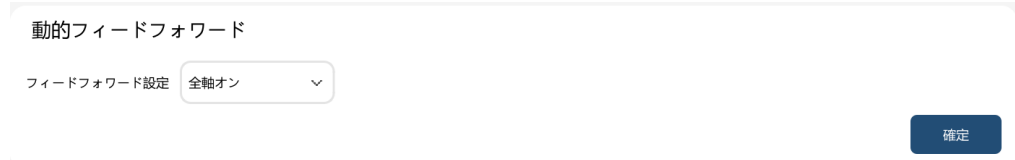
手順	図	説明
1. グローバルユーザー座標系一覧で編集したいユーザー座標系を選択し、右下の「編集」ボタンをタップして、グローバルユーザー座標系編集ウィザードに進みます		「プログラマー」以上の権限で、編集操作を行います。
2. 図に示す手順に従ってキャリブレーションしてください。		ロボットをジョグして、設定済みのツール TCP を目標座標系の原点、x 軸上の点、xy 平面上の点、または y 軸上の点に動かし、「1 点目確認、2 点目確認、3 点目確認」ボタンをタップします。 3 点とも確認したら、「キャリブレーション実行」ボタンをタップします。

9.7 動力学設定

動力学設定画面は、ロボットの動力学関連パラメーターを設定するために使用されます。動力学設定はロボット力制御、ドラッグ教示、バーチャルフェンス、衝突検知などの機能に関連しています。上記の機能が正しく動作しなかったり、ロボットが異常振動したりする可能性があるため、ロボットの動力学設定が適切であることを確認してください。

9.7.1 動的フィードフォワード

動的フィードフォワードは、コントローラーが動的フィードフォワード機能をオンにするかどうかを決定します。デフォルトでは全軸オンになっています。モーター電源オン時の振動や軌跡精度の低下などの悪影響を防ぐために、ユーザーが動的フィードフォワード機能をオフすることは推奨されません。床置き設置以外のベース座標系の設定や摩擦推定などの特別な状況では、動的フィードフォワード機能をオフにする必要があります。



9.7.2 動力学制約

動力学制約のスイッチは、コントローラーが動力学制約機能を有効化するかどうかを決定します。デフォルトでは有効になっています。モーターの過負荷や異常な振動などの悪影響を防ぐため、動力学制約機能を無効にすることは推奨されません。

動力学制約

☒ 有効化 ☐ 公称動力学パラメーター ☐ 工場同定動力学パラメーター

動力学制約スイッチを有効化すると、「公称動力学パラメーター」と「工場同定動力学パラメーター」の2つのオプションがあります。

「公称動力学パラメーター」は、この機種の標準的なパラメーターです。「公称動力学パラメーター」を使用する同じ機種が同じ動作プログラムを実行すると、完全に同じ動作速度とタクトタイムを確保できますが、その代償として、ある程度の動作性能を犠牲にするか、またはモーターに過負荷がかかる可能性があります。

「工場同定動力学パラメーター」は、ロボットを最良の動力学制御状態に保ちます。最短のタクトタイムを達成し、モーターに過負荷がかかるリスクを減らすことができますが、その代償として、同じ動作プログラムを実行するロボットの速度とタクトタイムがわずかに異なります。

9.7.3 振動抑制

振動抑制のスイッチは、コントローラーが振動抑制機能を有効にするかどうかを決定します。デフォルトでは無効になっています。

振動抑制

☒ 有効化

振動抑制スイッチを有効にすると、コントローラーはロボットの関節の柔軟性を補償し、関節の柔軟性による振動を抑制して位置精度と軌跡精度を向上させます。ロボットの負荷パラメーターの変化は、関節柔軟性の補償量に影響を与えるため、振動抑制機能を使用する際は正確な負荷パラメータの設定が必要です。

注意：現在、振動抑制機能は一部の機種でのみサポートされており、その他の機種は今後順次対応予定です。

9.8 本体パラメーター設定

本体パラメーターには、RD パラメーター、DH パラメーター、減速比、過負荷係数、結合係数などのロボット本体に関するデータが含まれます。この画面のパラメーターは動作精度に直接影響するため、弊社の許可と支援がない場合は、変更しないでください。

9.8.1 RD パラメーター

RD パラメーターは、ロボットのリンク座標系の相対的な姿勢関係を記述し、ロボットの運動学の基礎となります。

RDパラメーター

	X(mm)	Y(mm)	Z(mm)
ベース:	0	0	0
軸 1:	0	0	350
軸 2:	0	0	760
軸 3:	0	0	540
軸 4:	0	150	0
軸 5:	0	0	127
軸 6:	0	0	0

インポート

確定

検証

①

②

③

①	インポート： タップして RD パラメータファイルを選択します。通常はこの操作を実行する必要はありません。
②	確定： 手動で RD パラメーターを変更した後、「確定」をタップして有効にします。
③	検証： タップすると、現在の RD パラメーターとシステムのデフォルトの RD パラメーターが表示されます。

注意：

ロボットが工場出荷される際に、1組のデフォルトパラメーターが設定されています。

パラメーターを変更またはインポートする前に、**RD** パラメーターの妥当性を確認する必要があります。

RD パラメーターを変更した後、有効にするにはコントローラーを再起動する必要があります。

9.8.2 DH パラメーター

DHパラメーターは、ロボットのリンク座標系の相対的な姿勢関係を記述し、ロボットの運動学の基礎となります。

DHパラメーター

	Alpha(°)	A(mm)	D(mm)	Theta(°)
軸 1:	0	0	296	0
軸 2:	-90	0	0	-90
軸 3:	180	490	0	-90
軸 4:	-90	0	360	0
軸 5:	90	0	150	0
軸 6:	-90	0	127	0

確定

検証

①
②

①	確定： 手動で DH パラメーターを変更した後、「確定」をタップして有効にします。
②	検証： タップすると、現在の DH パラメーターとシステムのデフォルトの DH パラメーターが表示されます。

注意：

ロボットが工場出荷される際に、1組のデフォルトパラメーターが設定されています。

パラメーターを変更する前に、DH パラメーターの妥当性を確認する必要があります。

DHパラメーターを変更した後、有効にするにはコントローラーを再起動する必要があります。

ユーザーが変更できるのは Alpha、A、D パラメーターのみです。

9.8.3 減速比

減速比は、ロボット各軸の減速機のパラメーターです。弊社の許可と支援がない場合は、変更しないでください。

減速比

	分子	分母
軸 1:	120	1
軸 2:	120	1
軸 3:	100	1
軸 4:	100	1
軸 5:	100	1
軸 6:	100	1

確定

検証

①

②

①	確定： 手動で減速比パラメーターを変更した後、「確定」をタップして有効にします。
②	検証： タップすると、現在の減速比パラメーターとデフォルトの減速比パラメーターが表示されます。

9.8.4 過負荷係数

モーターの過負荷係数を設定します。

過負荷係数

モーター過負荷係数 伝動過負荷係数

軸1:	1	0.9
軸2:	1	0.9
軸3:	1	0.95
軸4:	1	0.85
軸5:	1	0.85
軸6:	1	0.85

確定

検証

①

②

①	確定： 手動で過負荷係数のパラメーターを変更した後、「確定」をタップして有効にします。 弊社のサポートを受けながら慎重に変更を行ってください！
②	検証： タップすると、現在の過負荷パラメーターとデフォルト過負荷パラメーターが表示されます。

9.8.5 結合係数

一部のロボットでは、各軸間に結合関係があり、結合係数はこれらの軸間の結合関係を説明しています。弊社の許可と支援なしに変更しないでください。

9.9 運動パラメーター設定

9.9.1 モーションパラメーター

モーションパラメーターには、ロボット各軸の最大速度、最大加速度、最大加加速度などがあります。上記のパラメーターは、ロボット動作のタクトタイムや滑らかさに影響します。ロボットには工場出荷時のデフォルトのパラメーターグループがあります。上記のパラメーターを変更すると、異常な振動が発生する、エラーが発生する、ロボットの寿命に影響を与えるなどの可能性があるため、慎重に変更を行ってください。

モーションパラメーター 詳細設定

	① 最大軸速度 (°/s)	② 最大軸加速度 (°/s ²)	③ 最大軸加加速度 (°/s ³)	④ 関節の剛性ゲイン
軸1:	180	1500	5000	1 [0.5, 2]
軸2:	180	1500	5000	1 [0.5, 2]
軸3:	234	1500	5000	1 [0.5, 2]
軸4:	240	1500	5000	1 [0.5, 2]
軸5:	240	1500	5000	1 [0.5, 2]
軸6:	240	1500	5000	1 [0.5, 2]

⑤ 加速度倍率: 1 [0.3, 1] ⑧ 速度平滑係数: 1 [1, 10]
 ⑥ 加速立上り時間 (s): 0.4 [0.05, 1] ⑨ log加速立上り時間: 0.4 [0.01, 1]
 ⑦ 目標位置到達時の加速度立上り時間 (s): 0.01 [0.01, 1] ⑩ 検証
 目標位置到達時の加速度立上り時間は、ジャーク以上の場合にのみ有効です。 確定
 先読み設定 ⑪ 先読みパラメーター数: 1000 [1, 1000] 確定
 推奨される先読みパラメーター数: 500以上

①	最大軸速度: ロボットの各軸の速度の上限です。主にモーターの速度によって制限されます。一般的に工場出荷時のパラメーターを使用するため、変更する必要はありません。
②	最大軸加速度: ロボットの各軸の加速度の上限です。モーターのトルク容量によって制限されます。このパラメーターは、動力学制約がオフの時に有効、動力学制約がオンの時は無効となります。このパラメーターは通常、軸の最大軸速度の 3～5 倍以上とします。
③	最大軸加加速度: ロボットの各軸の加加速度の上限です。通常は、加加速度を上げるとロボットのタクトが短くなりますが、動作中に振動が発生しやすくなります。ロボットの動作中に多くの小さな滑らか軌跡エリアがある場合は、ロボットの加加速度を適切に上げることによりタクトタイムを下げることはできますが、ロボットの振動に注意する必要があります。このパラメーターは、軸の最大加速度の 3～5 倍以上とします。
④	関節の剛性ゲイン
⑤	加速度倍率: このパラメーターは、ロボットの動作時の加速度を調整するために使用されます。値が大きいほど、ロボットの動作時の加速度が大きくなります。(本機能はサポートを終了しました。今後のバージョンではこのパラメーターは削除されます)

⑥	加速立上り時間： ロボットの加速度が最小値から最大値まで上昇するのに要する時間です。値が小さいほどロボットの加速は速くなり、値が大きいほどロボットの加速は滑らかになります。
⑦	目標位置到達時の加速度立上り時間： これは、目標位置において滑らか軌跡がない場合、ロボットの停止段階で加速度が最小値から 0 になるまでの時間です。数値が大きいほど、ロボットの停止がより滑らかになります。加速度上昇時間がこの目標位置到達時の加速度立上り時間より長い場合、大きい方の加速度立上り時間に従って停止します。
⑧	速度平滑係数： ロボットが滑らか軌跡エリアを通過する際の速度平滑化処理を調整するために使用されます。値が大きいほどロボットが滑らか軌跡エリアを通過する際に速度低下が小さくなりますが、振動が発生しやすくなります。 値が 1.0 の場合は、ロボットが滑らか軌跡エリアを通過する際に追加の速度平滑化処理を行いません。 このパラメーターは、ロボットの限界性能を向上させるために使用されます。デバッグする際、まず 1.0 の場合のロボットの振動を確認します。ロボットの振動が激しい場合は、ロボットが性能の限界に達したことを示すのでこのパラメーターを上げる必要はありません。ロボットはスムーズに動作するが滑らか軌跡エリアを通過する際に速度が大幅に低下する場合は、このパラメーターを適切に上げて動作のスムーズさを改善することができます。このパラメーターをデバッグする場合、0.1～0.5 程度のステップ長で徐々に増加させることをお勧めします。
⑨	Jog 加速立上り時間： ジョグ中にロボットの加速度が最小値から最大値まで上昇するのに要する時間です。ステップ長 10mm のジョグと連続モードのジョグでのみ有効となります。この値が小さいほどロボットのジョグの開始と停止は速くなりますが、振動が発生しやすくなります。ジョグ中にロボットの応答が遅い場合は、この値を適切に小さくすることができます。
⑩	検証： タップすると、現在のモーションパラメーターとシステムのデフォルトのモーションパラメーターが表示されます。
⑪	先読みパラメーター数： 値が大きいほどロボットの動作はスムーズで高速になりますが、ロジックや信号処理の適時性に影響します。 注意：通常この値は調整する必要はなく、デフォルト値で問題ありません。特殊な場合は、弊社の指示に従って変更してください。

9.9.2 詳細設定

9.9.2.1 安全制御パラメーター

安全制御パラメーター

Stop0 :	最大能力減速停.	▼	1	減速係数(0.1~1)
Stop1 :	最大能力減速停.	▼	1	減速係数(0.1~1)

最大能力減速停止は、停止信号を受信後、ロボットの軌道がずれないことを前提に、少なくとも1つのモーターが最大減速能力で停止を計画します。

異なるロボットが異なる作業条件下で実際の性能に影響を受けることを考慮し、この停止方式では減速係数を設定できます。設定範囲は[0.1, 1]です。最大値 1 に設定すると、ロボットの出力は迅速にモーターの最大能力に達し、減速停止します。最小値 0.1 に設定すると、ロボットの出力は比較的穏やかにモーターの最大能力に達し、減速停止します。つまり、この係数の値が大きいくほど、ロボットの出力がモーターの最大トルク値に達する速度が速くなり、ロボットの停止が急激になります。逆に、係数の値が小さいほど、ロボットの出力がモーターのトルク値に達する速度が穏やかになり、ロボットの停止が緩やかになります。

さらに、手動モードの停止係数（Stop0）と自動モードの停止係数（Stop1）は個別に設定できます。

9.9.2.2 Search 命令の最大停止距離

Search 命令を使用し、停止方法としてクイックストップを選択した場合は、ロボットが動作停止信号を受信してから完全に動作を停止するまでの間に、TCP が移動する距離はこの値を超えません。

Search命令の最大停止距離 (mm)

距離 : 2 [0.5~20]

確定

9.9.2.3 滑らか軌跡最小半径の設定

生成可能な滑らか軌跡エリアの最小サイズを設定できます。このパラメーターを使用することで、小さすぎる滑らか軌跡エリアの生成を回避し、動作のスムーズさを向上させることができます。制御システムは、軌跡の長さがこのパラメーターの設定値より小さいかつ滑らか軌跡エリアを生成する必要があることを検出した場合は、軌跡をその隣接する軌跡と自動的に 1 つに結合し、必要な長さの滑らか軌跡エリアを生成します。このパラメーターの値が大きいくほど最小滑らか軌跡エリアが大きくなり、ロボットが滑らか軌跡エリアを通過する際の動作がスムーズになります。このパラメーターの値が 0 に設定されている場合は、制御システムは滑らか軌跡エリアに対して追加処理をせず、パラメーターに厳密に従って滑らか軌跡エリアを生成します。

滑らか軌跡最小半径の設定 (mm)

滑らか軌跡最小半径 : 0 [0, 10]

確定

9.9.2.4 パレットデバッグモード

パレットデバッグモードは、CR、SR、6 軸産業用ロボット（NB、XB）シリーズの機種でのみ表示され利用できます。このモードをオンにすると、4 軸ロックに対応する 2 つのジョグ座標系（「特異点回避」と「ベースと平行」）が追加されます。

注意：CR シリーズの 5 軸機種はデフォルトでこのモードがオンになっており、ベース座標系のジョグは 6 軸機種の特異点回避座標系に対応します。

9.9.2.5 デフォルト Conf

デフォルト Conf の「RL」オプションは、RL プロジェクトを再読み込みした後、動作指令が位置の Conf 情報に従って厳密に動作制約を行うかどうかを設定するために使用されます。

注意：このオプションを変更した後は、**pptomain** を実行して、有効にする必要があります。

デフォルト Conf の「Point move to」オプションは、「Move to」を使用する場合に、動作が位置の Conf 情報に基づいて動作制約を行うかどうかを設定するために使用されます。

注意：デフォルトでは、Conf 内の 4 つのスイッチはすべて有効になっています。



9.10 力制御パラメーター設定

9.10.1 力制御パラメーター



力制御ゲイン：ロボットの力制御の応答速度を調整します。値が大きいくほどロボットの力制御の応答速度は速くなりますが、干渉防止能力が弱くなります。デフォルト値は 1.0 です。ロボットを剛性の低い移動可能なベースなどに置く場合は、力制御ゲインの値を適切に下げる必要があります。ロボット先端の負荷が負荷線図の上限荷重に近い場合は、ドラッグやインピーダンス機能を使用すると振動が発生する可能性があるため、力制御ゲイン値を適切に下げる必要があります。制御システムのバージョンが V2.0 以下の場合、ベース剛性オプションを低剛性モードに設定する必要があります。

センサー補正係数：センサーの誤差を補正します。通常は変更は推奨されません。ロボットのドラッグ中に関節の正の方向にかかる力がある場合は、このパラメーターを適切に下げ、逆に関節の負の方向にかかる力がある場合は、このパラメーターを適切に上げてください。デフォルト値は 0.5 です。

摩擦補正係数：ドラッグとインピーダンス動作中の摩擦を補正します。補正値が大きすぎると不安定な現象が発生することがありますので、慎重に変更を行ってください。デフォルト値は 0.5 です。

注意：力制御パラメーターは上級開発者向けですので、慎重に変更を行ってください。

9.10.2 力制御モデル

力制御モデルには 3 つのオプションがあります。この機能は開発者向けですので、慎重に変更を行ってください。初期化及び未変更の状態では、デフォルトの公称モデルを使用します。

力制御モデル
タイプ: 公称モデル

確定

9.10.3 ドラッグの最適化

ドラッグの最適化は、ドラッグの操作性を向上させます。初期化及び未変更の状態では、デフォルトでこの機能がオンになっています。

9.10.4 ドラッグの自動有効化

協働ロボットにおいて、先端のボタンを押さない状態でのドラッグ操作をサポートしています。「ドラッグの自動有効化」スイッチをオンにすると、先端のボタンを押さなくてもドラッグ操作を行うことができます。

ドラッグの自動有効化 ロボット先端のボタンを押さなくてもドラッグできるようになります。これを有効にすると、ロボット先端のドラッグボタンは無効になります。

☒ 有効化

注意：

1. 自動ドラッグ開始スイッチをオンにした後、ドラッグ操作を行う際は安全に注意してください。ロボットに異常が発生した場合は、非常停止ボタンを押してください。
2. 自動ドラッグ開始スイッチをオンにすると、ロボット先端のドラッグボタンは無効になります。

9.10.5 力制御モデル誤差閾値

協働ロボットは、ドラッグを開始する際に、センサーのトルクと理論的な動力学モデルのトルクの誤差を確認します。誤差が大きい場合、ドラッグの開始に失敗することがあります。負荷設定が正しく行われていることを確認した上で依然として開始に失敗する場合は、力制御モデル誤差閾値を適切に大きくすることで対応できます。

力制御モデル誤差閾値 (Nm)

軸1 : 25 軸2 : 40 軸3 : 40 軸4 : 25 軸5 : 25 軸6 : 25

☒ 有効化

注意：

1. 無効化された状態では、コントローラーはデフォルト値を使用します。

9.10.6 センサーデュアルチャンネル電圧差閾値

協働ロボットは、ドラッグ機能を有効にする際に 2 チャンネルのセンサーの電圧値の偏差をチェックします。偏差が大きすぎる場合、ドラッグの有効化に失敗し、「電圧偏差が大きすぎます」というメッセージが表示されます。この場合、センサーの偏差の閾値を適切に大きくすることで対応できます。

センサーデュアルチャンネル電圧差閾値 (mV)

軸1 : 1500 軸2 : 1500 軸3 : 1500 軸4 : 1500 軸5 : 1500 軸6 : 1500

☒ 有効化

注意：

1. 無効化された状態では、コントローラーはデフォルト値を使用します。

9.11 クイック移動

クイック移動機能は、よく使用する位置/姿勢を定義し、ジョグボタンや命令でロボットを素早く目的の位置/姿勢にすることができる機能です。ドラッグ姿勢、輸送姿勢、ホーム位置があります。これらの姿勢は機種のデフォルト値があり、ユーザー定義もサポートします。

ドラッグ姿勢

☒ デフォルト ☐ カスタム

→ Move to

確定

輸送姿勢

☒ デフォルト ☐ カスタム

→ Move to

確定

ホーム位置

☒ デフォルト ☐ カスタム

→ Move to

確定

ホーム位置は、基準位置の定義に加えてオフセット値を設定する必要があります。ロボットの現在の位置とホーム位置で定義した位置との差がオフセット値より小さい場合は、ロボットがホーム位置にあるとみなされ、システム IO の「Home state」とレジスタの「sta_home」が出力されます。

ホーム位置

☐ デフォルト ☒ カスタム ①

軸1 :	0°	オフセット :	0.1° ②	+/- (0.1 - 30)
軸2 :	0°	オフセット :	0.1°	+/- (0.1 - 30)
軸3 :	0°	オフセット :	0.1°	+/- (0.1 - 30)
軸4 :	0°	オフセット :	0.1°	+/- (0.1 - 30)
軸5 :	0°	オフセット :	0.1°	+/- (0.1 - 30)
軸6 :	0°	オフセット :	0.1°	+/- (0.1 - 30)

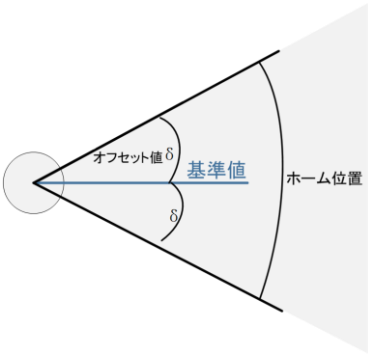
現在位置を取得

→ Move to

確定

①	各関節のホーム位置基準値。
②	①の角度を中心として、ホーム位置であると判定する範囲で、設定可能な範囲は [0.1,30] です。例えば、基準値が 1° でオフセット値が 3° の場合は、ホーム位置の判定範囲は[-2° ,4°]となります

ホーム位置、基準値、オフセット値の関係は下図のようになっています。



クイック移動機能は、ロボットの TCP 位置とアーム角度（7 軸ロボットのみアーム角度の概念があります）を変更しないまま、フランジ平面が地面と平行、ツール座標系の X 軸が地面と垂直、ツール座標系の Y 軸が地面と垂直、ツール座標系の Z 軸が地面と垂直など、ロボットを多くの特別な姿勢に素早く調整することができます。

先端ツール

調整モード フランジ平面が地面と平行

→ Move to

確定

クイック移動機能の使用方法はジョグ操作に似ています。手動モードでは、イネーブルボタンでロボットのモーターに電源を入れ、目的の動作のボタンを押し続けることでロボットは軸動作で目標姿勢に動作します。動作速度はジョグ速度で調整することができます。動作中にモーターの電源をオフにするかまたはボタンを離すと、ロボットは動作を停止します。

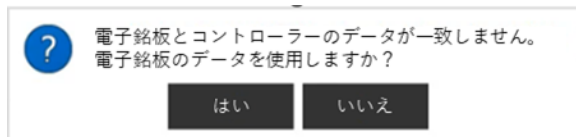
注意：クイック移動機能を使用してポーズを調整する場合は、想定外の衝突を避けるためにソフトウェアリミットをオンにしてください。

9.12 電子銘板

一部の産業用ロボットには電子銘板が内蔵されています。産業用コンピューターや制御キャビネットの交換による基本データの損失を避けるために、本体と関連するデータを保存します。

電子銘板のソフトウェア機能は、主にコントローラーと RobotAssist ソフトウェアの 2 つの部分に分けられます。コントローラーは電子銘板のデータ読み取り、検証、上書きなどの機能を担当し、RobotAssist ソフトウェアは電子銘板に関する操作命令の実行とデータ表示などの機能を担当します。

コントローラー起動後、まず電子銘板の有無を検出します。電子銘板が有る場合は、データを読み取って検証し、検証結果を保存します。電子銘板無しかつユーザーが電子銘板の使用を選択していない場合は、コントローラーのデータを直接使用して動作します。電子銘板無しでユーザーが電子銘板の使用を選択している場合は、電子銘板が無いことを知らせるのブロードンプトが表示されます。RobotAssist ソフトウェアがコントローラーに接続された後、コントローラー内の電子銘板のデータ検証の結果を検出し、検証結果に応じてポップアップウィンドウを表示します。ユーザーはそれらの指示に従って操作する必要があります。



電子銘板のデータが正常に使用された場合は、デフォルトで電子銘板のデータを使用してコントローラーのデータを上書きします。

以下の 3 つの状況ではポップアップウィンドウが表示されます。

- (1) 電子銘板が検出され、データがコントローラーと異なる場合は、「電子銘板のデータを使用しますか？」とポップアップ表示されます。「はい」を選択すると電子銘板の情報を使用します。「いいえ」を選択するとコントローラーのデータを使用します。
- (2) 一度電子銘板のデータを使用することを選択した後再起動すると、デフォルトで電子銘板の情報を使用します。コントローラーの情報が電子銘板の情報と異なる場合は、

再度「電子銘板のデータを使用しますか？」とポップアップ表示されます。

- (3) 起動最初に電子銘板がない場合は、デフォルトでコントローラーのデータを使用します。電子銘板の情報を使用した後、次の起動で電子銘板が検出されない場合は、「コントローラーのデータを使用しますか？」と表示されます。「はい」を選択するとコントローラーのデータを正常に使用します。「いいえ」を選択するとコントローラー全体は故障状態となり操作できなくなります。問題を解決するために再起動する必要があります。



ヒント

電子銘板内の機種がコントローラー内の機種と異なる場合は、電子銘板のデータの使用に失敗します。電子銘板を正常に使用するには、コントローラー内の機種が電子銘板内の機種と一致することを確認してください。

HMI で「設定」→「電子銘板」をタップすると、電子銘板の情報が表示されます。電子銘板の使用の有無に関わらず、コントローラーが電子銘板を検出すると、このインターフェースは電子銘板に関連するパラメーター情報を表示します。電子銘板の状態はこのインターフェースのステータスバーで判断できます。下図に示すように、3つのパラメーターは、それぞれ電子銘板の状態、電子銘板データとコントローラーデータの一致、起動後に「電子銘板の使用」ボタンをタップするかどうかが表示されます。



起動後に電子銘板が検出されない場合は、インターフェース表示は以下のようになります。



機能	説明
コントローラーデータのエクスポート	コントローラーの関連パラメーターデータをファイルにエクスポートします。
電子銘板データのエクスポート	電子銘板のデータをファイルにエクスポートします。
リフレッシュ	電子銘板情報の状態を同期します。
基本情報	電子銘板内の基本情報のパラメーターを表示します。手動での変更はできません。
エンコーダー電圧	エンコーダーのバッテリー電圧を表示します。エンコーダーのバッテリー電圧は、電源オン時または電源オン後 24 時間ごとに検出され、実際の電圧値が表示されます。手動での変

	更はできません。
動作時間	モーターが動作すると、動作時間を蓄積します。インターフェースの表示値は 1 時間ごとに更新されます。手動での変更はできません。
機械ゼロパラメーター、 運動学パラメーター	インターフェースはそれぞれコントローラーの現在値と電子銘板の現在値を表示します。手動での変更はできません。
動力学パラメーター	このパラメーターはインターフェースに表示されません。
電子銘板データの上書き	コントローラー内のデータを使用して、電子銘板内のデータを上書きします。

Robot Assist

電子銘板

状態：正常、不一致、使用済み

コントローラーデータのエクスポート 電子銘板データのエクスポート リフレッシュ

基本情報

ID 1654053293087 型番 XB20s_R1813_040 版数 asd シリアル a データ 0

モニタリング情報

エンコーダー電圧(V) 0 動作時間(h) 0

機械ゼロパラメーター

コントローラー

校正時間 2022-01-01 07:59:59

マルチターン [-1, -1, -1, -1, -1, -1]

1回転値 [-1, -1, -1, -1, -1, -1]

電子銘板

校正時間 2022-01-01 07:59:59

マルチターン [-1, -1, -1, -1, -1, -1]

1回転値 [-1, -1, -1, -1, -1, -1]

電子銘板



ヒント

1. エクスポートされたデータはすべて暗号化されます。
2. 電子銘板を使用することを選択した場合は、ゼロキャリブレーション、本体パラメーターの変更、動特性同定を実行した後、コントローラーは自動的に変更されたデータを電子銘板に同期させます。

9.13 エラーコードフィルター

xCore でアラームが発生した場合、アラームステータスが割り当てられているシステム IO またはレジスタが High を出力します。特定のアラームをシステム IO またはレジスタに出力したくない場合は、この機能によって設定できます。

エラーコード分類 : デモンストレーション

フィルター ID: タイトル:

	ID	タイトル
1	100	DEMOが予期せず停止しました
2	101	DEMOの起動に失敗しました
3	102	DEMOのシャットダウンに失敗しました

→

注記:
下の表のエラーコードは、システムIOまたはレジスタアラーム出力をトリガーしません。

ID	タイトル
----	------

削除 すべてクリア

10 通信

10.1 本章の概要

本章では、xCore 制御システムのさまざまな通信設定（IO、レジスタ、バスデバイス、エンドエフェクターなど）を説明します。

10.2 システム IO

システム IO には、システム入力とシステム出力の 2 種類があります。

外部コントローラーはシステム入力を使用して、モーターの電源オン、プログラム開始、非常停止リセットなど、さまざまな命令を xCore 制御システムに送信することができます。

xCore システムはシステム出力を使用して、モーター電源オン/オフや動作状態などのロボットの状態を外部コントローラーに送信することができます。

10.2.1 システム入力

HMI で、「通信」→「システム IO」からシステム IO の設定画面に入ります。「システム入力」タブをタップすると、下図に示すようにシステム入力の設定画面に入ります。

xCore 制御システムは、次のシステム入力をサポートしています。

システム入力の名称	トリガー方法	備考
Motor On	立ち上がりエッジ/ 立ち下がりエッジ	
Motor Off	立ち上がりエッジ/ 立ち下がりエッジ	
Program Start	立ち上がりエッジ/ 立ち下がりエッジ	
Program Pause	立ち上がりエッジ/ 立ち下がりエッジ	
Program Pause 1	立ち上がりエッジ/ 立ち下がりエッジ	機能は「Program Pause」と同じです。

Switch Auto	立ち上がりエッジ/ 立ち下がりエッジ	手動モードの時に使用可能
Switch Manual	立ち上がりエッジ/ 立ち下がりエッジ	
Auto And Power	立ち上がりエッジ	手動モードの時に使用することができ、自動モードへの切り替え、モーター電源オンを順次実行します。モーターが電源オンの時はこの命令に応答しません。
PP to Main	立ち上がりエッジ	
MotorON&Run	立ち上がりエッジ	モーター電源オン、pptomain、実行を順次行います。
MotorOn&Continue	立ち上がりエッジ	モーター電源オン、実行を順次行います。
Pause&MotorOff	立ち上がりエッジ	一時停止、ロボット停止後にモーター電源オフを行います。
Open Drag	立ち上がりエッジ/ 立ち下がりエッジ	協働ロボットのみに対応しています。操作パネルでドラッグをオンにする必要があります。
Close Drag	立ち上がりエッジ/ 立ち下がりエッジ	協働ロボットのみに対応しています。操作パネルでドラッグをオンにする必要があります。
Emergency Reset	立ち上がりエッジ	
Clear Alarm	立ち上がりエッジ	
Emergency&Clear Alarm	立ち上がりエッジ	

注意：

- すべてのシステム入力パルスはトリガーです。xCore システムが外部命令を正しく受信するために、外部入力パルス幅が 300 ミリ秒以上であることを確認してください。
- 安全性を確保するために、立ち上がりエッジと立ち下がりエッジの両方のトリガーをサポートする機能（Motor On、Motor Off、Switch Auto、Switch Manual、Program Start、Program Pause、Open Drag、Close Drag）は 1 対 1 で対応しています。例えば、「Motor On」と「Motor Off」は対応しています。「Motor On」で DI1 が選択された場合は、DI1 は「Motor Off」以外の機能には使用できなくなります。
- システム入力のほとんどの機能は自動モードでのみ有効になり、手動モードではシステム入力の信号は無視されます。
- 一時停止機能に割り当てられているレジスタやシステム IO がリセットされていない場合は、いかなる手段でもプログラムの起動が許可されません。
- 「Program Pause」と「Program Pause 1」は同じ機能を持ち、信号がトリガーされるとプログラムが一時停止します。

10.2.2 システム出力

HMI で、「通信」→「システム IO」からシステム IO の設定画面に入ります。「システム出力」タブをタップすると、下図のシステム出力の設定画面に入ります。

xCore システムは、次のシステム出力をサポートしています。

システム出力の 名称	出力 High	出力 Low	備考
Motor State	モーターオン	モーターオフ	
Running State	プログラム実行中	プログラムが実行して いない	
Moving State	ロボット動作中	ロボットが静止中	RL プログラムの動作命令とジョグの時のみロボットの動作状態を検出します。 (注意：推定、ドラッグ、力制御、ドラッグ再生などの場合には、ロボットが動作していてもこの出力はロボット静止となります。)
Operate mode	自動モード	手動モード/待機モード	
ESop State	非常停止状態	非常停止状態ではない	この出力は「非常停止トリガーレベル」の設定に影響されます。 High に設定されている場合は、ソフトウェア非常停止がトリガーされたときに High が出力され、トリガーされていないときに Low が出力されます。 Low に設定されている場合は、ソフトウェア非常停止がトリガーされたときに Low が出力され、トリガーされていないときに High が出力されます。
Collision Opening	オン	オフ	PCB3、4 軸ロボットはサポートしません
Collision Detection	トリガーされた	トリガーされていない	PCB3、4 軸ロボットはサポートしません
Collision Alarm	衝突検出アラーム有り	アラーム無し	PCB3、4 軸ロボットはサポートしません

Collaboration State	協働モードの状態	非協働モードの状態	協働ロボットのみ
Alarm State	アラームが発生している状態	アラーム無し	
Servo Encoder Low Battery	エンコーダーの電圧低下アラーム有り	エンコーダーの電圧低下アラーム無し	
Home State	ロボットがホーム位置にある	ロボットがホーム位置にない	
Heart Beat	パルス信号		パルス周期は「設定－コントローラー設定」で設定できます。
Robot Startup Finish	ロボットコントローラーの起動完了	ロボットのコントローラーが起動していない	
Safe Door (Radar Stop State)	セーフティーゲートが有効になっている	セーフティーゲートが無効になっている	CR シリーズ及び SR シリーズでは、停止エリア入力の有効/無効を出力します。
Program Reset	プログラムがリセットされている	プログラムがリセットされていない	
pptomain execution status	pptomain を実行中	pptomain を実行していない	
Soft Emergency Stop Triggered State	ソフトウェア非常停止状態	ソフトウェア非常停止状態ではない	

注意：

- システム出力は手動モードと自動モードで有効になりますが、安全性と使いやすさの理由から、xCore が自動モードにある場合にのみこれらの信号を使用することをお勧めします。
- IO ポイントがシステム IO に割り当てられると、強制出力またはシミュレーション入力操作を実行できなくなります。
- 「動作モード」信号を除き、他のすべてのシステム出力は有効時に High を出力します。
- 「Operate mode」信号の出力は、自動モードで High、手動モードで Low となります。

10.3 外部通信

10.3.1 概要

xCore システムは、TCP Socket に基づいた外部通信インターフェースを提供し、サーバーとクライアントの両方をサポートします。上位システム（PLC、MES など）はこのインターフェースを介してロボットへの制御命令の送信やロボット状態の取得を実現します。

10.3.2 設定

使用する前に **Socket** 通信に関するパラメーターを設定し、この機能を有効にする必要があります。設定画面は、「HMI」→「通信」→「外部通信」にあります。

注意：

Socket 通信インターフェースは、ロボットをクライアントまたはサーバーとしてサポートしますが、同時に 1 つの状態としてのみ使用できます。

ロボットをクライアントとして使用する場合は、次のパラメーターを設定する必要があります。

パラメーター	説明
IP アドレス	接続されている PLM や MES システムなどのサーバーの IP アドレス。
ポート番号	サーバー監視ポート。
サフィックス	ロボットに制御命令や監視命令を送信する際の命令の末尾に付与される接尾辞。通常は簡単なサフィックスである ¥r や ¥n や ¥t が使用されます。 注意： ¥r¥n 、 ¥r¥t 、 ¥r¥n¥t のように長さ制限のないサフィックスの組合せに加えて、アルファベットなどの文字も使用できます。

ロボットをサーバーとして使用する場合は、複数台の接続をサポートしています。複数台の接続を使用する時、クライアントは制御タイミングの競合に注意を払う必要があります。次のパラメーターを設定する必要があります。

パラメーター	説明
ポート番号	サーバー監視ポート。
サフィックス	ロボットに制御命令や監視命令を送信する際の命令の末尾に付与される接尾辞。通常は簡単なサフィックスである ¥r や ¥n や ¥t が使用されます。 注意： ¥r¥n 、 ¥r¥t 、 ¥r¥n¥t のように長さ制限のないサフィックスの組合せに加えて、アルファベットなどの文字も使用できます。

10.3.3 対話型指令

対話型指令には制御指令とモニター指令があります。

指令の具体的な内容と形式は下表の通りです。(ユーザーが“¥r”をサフィックスとして使用すると仮定します。“¥r”はエンターを表すエスケープ文字であり、10 進数で 13 となります)。

制御命令：

指令名	送信文字列	戻り値	備考
socket インターフェースを無効にする	"xCore::SocketInterface::Disable" + "¥r"	戻り値なし	
socket インターフェースを有効にする	"xCore::SocketInterface::Enable" + "¥r"	戻り値なし	
プログラム開始	"start" + "¥r"	成功：True 失敗：False	一時停止機能に割り当てられているレジスタやシステム IO がリセットされていない場合は、起動が許可されません。
プログラム停止	"stop" + "¥r"	成功：True 失敗：False	
アラームをクリア	"clear_alarm" + "¥r"	成功：True 失敗：False	
プログラムポインタをメイン関数に移動	"pp_to_main" + "¥r"	成功：True 失敗：False	
モーター電源オン	"motor_on" + "¥r"	成功：True 失敗：False	
モーター電源オフ	"motor_off" + "¥r"	成功：True 失敗：False	
手動モードに切替	"switch_mode>manual" + "¥r"	成功：True 失敗：False	
自動モードに切替	"switch_mode:auto" + "¥r"	成功：True 失敗：False	
ドラッグ有効化	"open_drag" + "¥r"	成功：True 失敗：False	協働ロボットのみ
ドラッグ無効化	"close_drag" + "¥r"	成功：True 失敗：False	協働ロボットのみ
プロジェクト一覧の取得	list_prog + "¥r"	プロジェクト一覧を返す。 プロジェクトがない場合は「null」が返される。	
現在のプロジェクトの取得	current_prog + "¥r"	現在ロードされているプロジェクトを返す。 プロジェクトが	

		ない場合は「null」が返される。	
プロジェクトをロード	load_prog + (プロジェクト名) + "¥r"	成功 : True 失敗 : False	
DO の値を設定	"setdo:" + "IO の名前, IO の値" + "¥r"	成功 : True 失敗 : False	"setdo:DO0-0,true¥r"
非常停止リセット	estop_reset + "¥n"	成功 : True 失敗 : False	
非常停止リセットとアラームクリア	estopreset_and_clearalarm + "¥r"	成功 : True 失敗 : False	(RSC 専用、mini ミニボードでは無効)
モーターオン、プログラムポイントをメイン関数に移動、プログラム開始を順に実行	motoron_pptomain_start + "¥r"	成功 : True 失敗 : False	
モーターオンとプログラム開始を実行	motoron_start + "¥r"	成功 : True 失敗 : False	
プログラム一時停止とモーターオフ	pause_motoroff + "¥r"	成功 : True 失敗 : False	
プログラム実行速度を設定	set_program_speed: + プログラム速度 + "¥r"	成功 : True 失敗 : False	
ソフト非常停止のトリガー及びキャンセル	set_soft_estop:true/false + "¥r"	成功 : True 失敗 : False	
自動モードへの切替とモーターオン	switch_auto_motoron + "¥r"	成功 : True 失敗 : False	
対応する安全エリアを有効化する	open_safe_region: + 安全エリア番号 + "¥r"	成功 : True 失敗 : False	番号の範囲は[1-10]
対応する安全エリアを無効化する	close_safe_region: + 安全エリア番号 + "¥r"	成功 : True 失敗 : False	番号の範囲は[1-10]
協働モードオン	open_reduced_mode + "¥r"	成功 : True 失敗 : False	
協働モードオフ	close_reduced_mode + "¥r"	成功 : True 失敗 : False	

モニター指令

指令名	送信文字列	戻り値	備考
モーターパワーオン状態	"motor_on_state" + "¥r"	モーター電源オン : True モーター電源オフ : False	
プログラム実行状態	"robot_running_state" + "¥r"	実行状態 : True 非実行状態 : False	
非常停止状態	"estop_state" + "¥r"	非常停止トリガーレベルが High に設定されている場合 非常停止状態 : True 非常停止状態ではない : False 非常停止トリガーレベルが Low に設定されている場合 非常停止状態ではない : True 非常停止状態 : False	この戻り値は「非常停止トリガーレベル」の設定に影響されます。
故障状態	"fault_state" + "¥r"	故障状態 : True 故障状態ではない : False	
動作モード	"operating_mode" + "¥r"	自動モード : True 手動モード : False	
直交位置の取得	"cart_pos" + "¥r"	直交位置の文字列 + "¥r" ;	
直交位置の取得	"cart_pos_name" + "¥r"	"cart_pos:" + 直交位置の文字列 + "¥r" ;	
軸位置の取得	"jnt_pos" + "¥r"	軸位置の文字列 + "¥r" ;	
軸位置の取得	"jnt_pos_name" + "¥r"	"jnt_pos:" + 軸位置の文字列 + "¥r" ;	
軸速度の取得	"jnt_vel" + "¥r"	軸速度の文字列 + "¥r" ;	単位 : rad/s
軸速度の取得	"jnt_vel_name" + "¥r"	"jnt_pos:" + 軸速度の文字列 + "¥r" ;	単位 : rad/s
軸トルクの取得	"jnt_trq" + "¥r"	軸トルクの文字列 + "¥r" ;	単位 : N.m
軸トルクの取得	"jnt_trq_name" + "¥r"	"Jnt_trq:" + 軸トルクの文字列 + "¥r" ;	単位 : N.m
ホーム位置状態	"home_state" + "¥r"	ホーム位置である : True ホーム位置ではない : False	
衝突検知状態	"collision_state" + "¥r"	衝突検知の状態 : True 衝突していない : False	協働ロボットのみ
タスク状態	"task_state" + "¥r"	現在進行中のタスクの状態。 ready : 準備完了 jog : ジョグ中 load_identify : 負荷推定中 dynamic_identify : 動特性同定中 drag : ドラッグが有効状態 program : プログラム実行中 demo : デモ中 rci : RCI debug : デバッグ中	詳細については、「HMI の概要」のロボットの現在の状態を示すアイコンと説明を参照してください。

アラーム状態	"alarm_state" + "¥r"	アラーム状態：True アラーム発生なし：False	
衝突検知アラーム状態	"collision_alarm_state" + "¥r"	アラーム状態：True アラーム発生なし：False	
衝突検知有効状態	"collision_open_state" + "¥r"	衝突検知オン：True 衝突検知オフ：False	
コントローラー起動状態	"controller_is_running" + "¥r"	起動している場合：True 起動していない場合：False	
エンコーダー低電圧アラーム状態	"encoder_low_battery_state" + "¥r"	アラーム状態：True アラーム発生なし：False	
ロボットエラーコード	"robot_error_code" + "¥r"	エラーが発生している場合はそのエラーコードを返す。エラーが発生していない場合は 0 を返す。	
RL の一時停止状態	"program_full" + "¥r"	RL の一時停止状態は以下を含みます： 0：初期化状態 1：RL 動作中 2：HMI 一時停止 3：システム IO 一時停止 4：レジスタ機能コード一時停止 5：外部通信一時停止 6：SDK 一時停止 7：Pause 命令による一時停止 10：非常停止 11：セーフティゲート（CR シリーズ及び SR シリーズでは停止エリア入力状態） 12：その他の要因による一時停止	
プログラムリセット状態	"program_reset_state" + "¥r"	プログラムポインタが main 関数の最初の行にあり、プログラムがまだ実行されていない場合、True を返す。 そうでない場合は、False を返す。	
プログラム実行速度	"program_speed" + "¥r"	現在のプログラム実行速度 範囲：1～100	
pptomain の実行状態	"robot_is_busy" + "¥r"	pptomain が実行中であれば、True を返す それ以外の場合は、False を返す	
ロボット動作状態	"robot_is_moving" + "¥r"	ロボット動作中：True ロボット動作中ではない：False	
安全ドア状態	"safe_door_state" + "¥r"	ドアが開いている：True ドアが閉じている：False	

ソフト非常停止のトリガー状態	"soft_estop_state" + "¥r"	ソフトウェア非常停止が発生した場合：True 発生していない：False	
直交速度	"cart_vel" + "¥r"	直交速度 + "¥r"	
TCP 姿勢	"tcp_pose" + "¥r"	ロボット TCP の姿勢 + "¥r"	
TCP 速度	"tcp_vel" + "¥r"	ロボット TCP の速度 + "¥r"	
TCP 直線速度	"tcp_vel_mag" + "¥r"	ロボット TCP の合成直線速度 + "¥r"	
外部非常停止状態	"ext_estop_state" + "¥r"	外部非常停止状態：True 非外部非常停止状態：False	安全ボードがミニボードであり、かつファームウェアのバージョンが 1.0.8.7 以上
携帯型安全装置の非常停止状態	"hand_estop_state" + "¥r"	非常停止状態：True 非常停止状態ではない：False	安全ボードがミニボードであり、かつファームウェアのバージョンが 1.0.8.7 以上
協働モード状態	"collaboration_state" + "¥r"	協働モード状態：True 非協働モード状態：False	
協働モード状態	"reduced_mode_state" + "¥r"	協働モード状態：True 非協働モード状態：False	
IO 状態	"io_state:" + "IO 名称（複数の IO はカンマで区切る）" + "¥r"	IO の値を返す。True または False で表示し、対応する IO が見つからない場合は null を返す	

備考：

	文字列形式	単位
直交位置	x、y、z、a、b、c、q1、q2、q3、q4	x、y、z 単位：mm a、b、c 単位：degree q1～q4 は姿勢のクォータニオン
軸位置	j1、j2、j3、j4、j5、j6、j7	ロボットの軸角度の単位は rad 走行軸位置の単位は m
軸速度	vj1、vj2、vj3、vj4、vj5、vj6、vj7	ロボットの軸速度の単位は rad/s 走行軸速度の単位は m/s
軸トルク	tj1、tj2、tj3、tj4、tj5、tj6、tj7	ロボット軸と走行軸のトルクの単位は、いずれもモーターの定格トルクの 1000 分の 1

10.4 バスデバイス

10.4.1 バスデバイスの概要

CC-Link、Modbus、EtherCAT、PROFINET、EtherNet/IP をサポートしています。

CC-Link には、CC-Link デバイス（EtherCAT 経由で変換）と CC-Link IE Field Basic が含まれます。

EtherCAT は IO モジュール、PROFINET、EtherNet/IP などの他のバスモジュールを拡張するために使用できます。

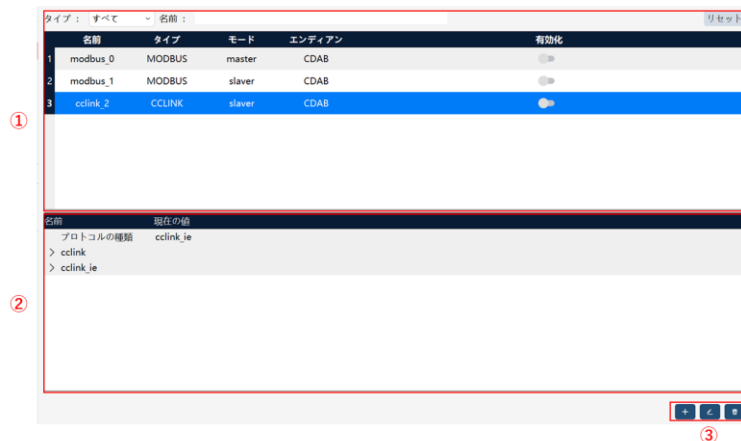
サポートバス	プロトコル	サポート方法	備考
Modbus	TCP	master と slave	
	UDP	非対応	
	RTU	master と slave	産業用ロボットのみ
CC-Link	485	リモートデバイス局 (slave)	産業用ロボットのみ
	IE Field Basic	リモートデバイス局 (slave)	

Modbus は次の機能コードに対応しています。

機能コード	意味	対応可否
0x01	コイルの読み取り	対応
0x05	単一コイルの書き込み	対応
0x0F	複数コイルの書き込み	対応
0x02	ディスクリット入力の読み取り	対応
0x04	入力レジスタの読み取り	非対応
0x03	保持レジスタの読み取り	対応
0x06	単一保持レジスタの書き込み	対応
0x10	複数保持レジスタの書き込み	対応

10.4.2 バスデバイスのパラメーター設定

この画面は、「通信」→「バスデバイス」にあります。画面は 2 つの部分に分かれており、上部はすべてのバス接続を管理し、各バスの接続を個別にオンオフできます。バス接続が閉じている場合は、この接続に設定された IO は「状態監視」→「IO 信号」で表示されなくなります。下部は現在選択されているバスデバイスの属性パラメーターです。



①	バスデバイス一覧です。
②	現在選択されているバスデバイスのパラメーターです。
③	バスデバイス操作ボタンです（左から順に新規作成、編集、削除）。

	名前	タイプ	モード	エンディアン	有効化
1	modbus_0	MODBUS	master	CDAB	☐
2	modbus_1	MODBUS	slaver	CDAB	☐

パラメーター名	パラメーターの説明
名前	名前は「IO デバイス」と「レジスタ」を設定するときに使用されます。 下図を例とすると、名前は modbus_0、modbus_1、modbus_2、cclink_3 です。 IOデバイス構成 デバイスタイプ デバイスタイプ: FIELDBUS バスデバイス情報 バスデバイス名: modbus_0 - modbus_0 - modbus_1 - modbus_2 - cclink_3 基本情報 IOボード番号: 2 名前: modbus_0 デジタル入力数: 動的取得 アナログ入力数: 動的取得 デジタル出力数: 動的取得 アナログ出力数: 動的取得 キャンセル 前へ 次へ 参照箇所は下図の通りです。 ● 「IO デバイス設定」: この名前フィールドを参照して、IO デバイスがどのバスに関連しているかを示します。 ● 「レジスタ」: この名前フィールドを参照して、レジスタがどのバスに関連しているかを示します。 レジスタを新規作成 デバイス タイプ: MODBUS 名前: modbus_0 基本情報 名前: register1 読み書き属性: ☒ 読み ☐ 書き タイプ: int16 保持レジスタ: ☐ はい ☒ いいえ 開始レジスタ: 40000 長さ: 1 説明: 機能 機能コード: キャンセル 前へ 次へ

タイプ	バスデバイスを追加/編集する際に選択でき、現在 CC-Link、Modbus、EtherCAT、PROFINET、EtherNet/IP サポートしています。 注意： ● CC-Link には、CC-Link デバイス（EtherCAT 経由で変換）と CC-Link IE Field Basic が含まれます。 ● EtherCAT は IO モジュール、PROFINET、EtherNet/IP などの他のバスモジュールを拡張するために使用できます。
モード	現在のロボットがバス上でマスターとして動作するか、またはスレーブとして動作するかを示します。
エンディアン	主にレジスタに使用されます。各レジスタは2バイトを占有するため、この2バイトの16進数の配列が多数存在します。この属性はマスターとスレーブの間で一致する必要があるため、そうでないとデータは期待通りになりません。エンディアンは、ABCD、CDAB、BADC、DCBA の4種類をサポートしています。デフォルトではCDABを使用します。
有効化ボタン	このボタンでバス機能の ON/OFF を切り替えます。各バスデバイスは個別に ON/OFF できます。 注意：バスデバイスをオフにすると、このバスデバイスに設定された IO は「状態監視」→「IO 信号」に表示されません。

10.4.2.1 Modbus 通信

バスデバイスの画面で右下の    をタップし、通信バスデバイスの追加の画面に入りデバイス種類で「MODBUS」を選択します。TCP と RTU プロトコルの両方をサポートし、マスターとスレーブの両方のバス構成をサポートします。

10.4.2.1.1 Modbus TCP 設定



デバイスタイプ
タイプ: MODBUS

基本情報
名前: modbus_4
モード: slaver
エンディアン: CDAB

拡張属性
プロトコルの種類: TCP
スレーブID: 1
TCP IP: 0.0.0.0
TCP ポート: 502
ホールドレジスタ 開始アドレス: 40000
ホールドレジスタ 最大数: 2000
コイル 開始アドレス: 0
コイル 最大数: 16
ディスタリット入力 開始アドレス: 0
ディスタリット入力 最大数: 16

キャンセル 前へ 次へ

パラメーター	概要
モード	「master」は、ロボットがマスターとして動作します。 「slave」は、ロボットがスレーブとして動作します。
スレーブ ID	ロボットがスレーブとして動作する場合は、バス全体の設定に注意し他のスレーブと競合しないようにしてください。 ロボットがマスターとして動作する場合は、この設定はロボットが通信する目標スレーブ ID を示します 注意：ロボットがマスターとして動作する場合は、周辺機器との単一スレーブ通信のみがサポートされます。

TCP/IP	ロボットがスレーブとして動作する場合は、0.0.0.0を記入します。すべてのネットワークインターフェースを待ち受けることを意味します。 ロボットがマスターとして動作する場合は、通信対象のスレーブの IP アドレスを記入する必要があります。
TCP ポート	スレーブが TCP プロトコル使用時のポート番号です。
保持レジスタ開始アドレス	機能コード 0x03、0x06、0x10 の影響を受けるレジスタの開始アドレスです。1つのレジスタは2バイトを占有します。 書き込み専用レジスタの場合： ロボットがスレーブとして動作する時は保持レジスタの機能コード 0x03 を使用します。ロボットがマスターとして動作する時は保持レジスタの機能コードは0x06 または 0x10 を使用します。 読み取り専用レジスタの場合： ロボットがスレーブとして動作する時は保持レジスタの機能コードは 0x06 または 0x10 を使用します。ロボットがマスターとして動作する時は保持レジスタの機能コードは 0x03 を使用します。
保持レジスタ最大数	保持レジスタの開始アドレスから始まる保持レジスタの数です。
コイル開始アドレス	機能コード 0x01、0x05、0x0F の影響を受けるレジスタの開始アドレスです。
コイル最大数	コイルの開始アドレスから始まるコイルレジスタの数です。
ディスクリート入力開始アドレス	機能コード 0x02 の影響を受けるレジスタの開始アドレスです。
ディスクリート入力最大数	ディスクリート入力の開始アドレスから始まるディスクリート入力レジスタの数です。

10.4.2.1.2 Modbus RTU 設定

デバイスタイプ

タイプ： MODBUS

基本情報

名前： modbus_4
モード： slaver
エンディアン： CDAB

拡張属性

プロトコルの種類： RTU
スレーブID： 1
RTU シリアル名： serial0
ホールドレジスタ 開始アドレス： 40000
ホールドレジスタ 最大数： 2000
コイル 開始アドレス： 0
コイル 最大数： 16
ディスクリート入力 開始アドレス： 0
ディスクリート入力 最大数： 16

キャンセル

前へ 次へ

一部の概念は Modbus TCP と同じです。説明は繰り返さず、相違点のみ紹介します。

RTU シリアルポート名は、バス通信に使用するシリアルポートの媒体を示します。この項目

は、通信に使用するパラメーターを含め、「通信」→「シリアルポート設定」で設定する必要があります。

10.4.2.2 CC-Link 通信

バスデバイスの画面で右下の    をタップし、通信バスデバイスの追加画面に入り、デバイス種類で「CCLINK」を選択します。CC-Link と CC-Link IE Field Basic プロトコルに対応しており、スレーブバス設定のみをサポートしています。

10.4.2.2.1 CC-Link IE 設定

パラメーター	概要
プロトコルタイプ	cclink_ie は、ロボットのイーサネットポートを使用する CC-Link IE Field Basic 通信プロトコルを示します。
cclink_ie ネットワークカード	通信に使用するイーサネットカードを設定します。
cclink_ie 占有局数	1～16 の占有局を設定でき、デフォルトは 4 です。デフォルト値での使用を推奨します。
cclink_ie プロトコルのバージョン	Ver1 または Ver2 を選択できます。この項目はマスターと一致する必要があります。

10.4.2.3 EtherCAT 通信

バスデバイスの画面で右下の    をタップし、通信バスデバイスの追加画面に入り、デバイス種類で「ETHERCAT」を選択します。EtherCAT は PROFINET、EtherNet/IP などのモジュールゲートウェイに接続するために使用できます。



スレーブアドレス：このパラメーターは EtherCAT トポロジーにおけるスレーブアドレス番号です。

注意：ロボットの内部デバイスが既に EtherCAT スレーブアドレスの 1000～4000 を使用しています。デバイスアドレスの競合を避けるために、ユーザーが拡張する EtherCAT デバイスのスレーブアドレスは 5000 以上である必要があります。

10.4.2.4 PROFINET 通信

バスデバイスの画面で右下隅の    をタップし、通信バスデバイスの追加画面に入り、デバイス種類で「PROFINET」を選択します。スレーブバス構成のみサポートしています。1 台のロボットは 1 つの PROFINET スレーブの設定をサポートし、複数のロボットはそれぞれの PROFINET スレーブ名を変更することにより、同じ PROFINET ネットワークに参加することができます。スロット 1～6 のモデル選択は、通信相手側の設定と一致する必要があります。



パラメーターの説明：

パラメーター	説明
モード	スレーブのみサポートしています。
エンディアン	一般的に、DCBA を選択します。具体的なオプションは通信双方間の合意に基づいて決定します。
ステーション名	PROFINET スレーブ名です。通信相手側の設定と一致する必要があります。中国語、大文字、アンダースコアはサポートされていません。
ネットワークカード	通信相手と接続しているネットワークポートを選択します。ネットワークカードの IP アドレスとネットワークカードの名前を含みます。
更新サイクル	デフォルトは 10ms です。最小は 2ms をサポートします。
スロット 1 タイプ	DO_256 モデルのみ選択可能です。スロット 1 を通してロボットから通信相手に出力されるデータ量が 256 個あることを示します。
スロット 2 タイプ	DI_256 モデルのみ選択可能です。スロット 2 を通して通信相手からロボットに入力されるデータ量が 256 個あることを示します。
スロット 3 タイプ	選択可能なモデル： AO_Int16_8/ AO_Int16_16/ AO_Int16_32/ AO_Int16_64/ AO_Int16_128/ AO_Int16_256 AO_Int16_8 は、スロット 3 を通してロボットから通信相手に出力される int16 のアナログ値が 8 個あることを示します。以下同様です。
スロット 4 タイプ	選択可能なモデル： AI_Int16_8/ AI_Int16_16/ AI_Int16_32/ AI_Int16_64/ AI_Int16_128/ AI_Int16_256
スロット 5 タイプ	選択可能なモデル AO_Float32_8/ AO_Float32_16/ AO_Float32_32/ AO_Float32_64/ AO_Float32_128/ AO_Float32_256

スロット 6 タイプ	選択可能なモデル AO_Float32_8/AO_Float32_16/AO_Float32_32/AO_Float32_64/ AO_Float32_128/AO_Float32_256
------------	--

10.4.2.5 EtherNet/IP 通信

バスデバイスの画面で右下隅の    をタップし、通信バスデバイスの追加画面に入り、デバイス種類で「EtherNet/IP」を選択します。スレーブバス構成のみサポートしています。1 台のロボットは 1 つの EtherNet/IP スレーブを設定できます。



The screenshot shows the 'EtherNet/IP' configuration screen. At the top, 'デバイスタイプ' (Device Type) is set to 'EtherNet/IP'. Below this, there are two main sections: '基本情報' (Basic Information) and '拡張属性' (Extended Properties). In '基本情報', '名前' (Name) is 'ethernet_ip_5', 'モード' (Mode) is 'slaver', and 'エンディアン' (Endian) is 'CDAB'. In '拡張属性', 'ネットワークカード' (Network Card) is '192.168.21.9: ens33', '読み取り専用レジスタ数' (Read-only Register Count) is '32', and '書き込み専用レジスタ数' (Write-only Register Count) is '32'. At the bottom, there are 'キャンセル' (Cancel) and '完了' (Done) buttons.

パラメーターの説明：

パラメーター	説明
モード	スレーブのみサポートしています。
エンディアン	一般的に、CDAB を選択します。具体的なオプションは通信双方間の合意に基づいて決定します。
ネットワークカード	EtherNet/IP マスターと接続するネットワークカードの名称を選択します。ドロップダウンリストには、ネットワークカードの IP アドレスと名称が表示されます。
読み取り専用レジスタ数	EtherNet/IP スレーブが持つ読み取り専用レジスタの数を設定します。1 つのレジスタは 2 バイトを表し、選択可能な値は 32、64、128、248 です。
書き込み専用レジスタ数	EtherNet/IP スレーブが持つ書き込み専用レジスタの数を設定します。1 つのレジスタは 2 バイトを表し、選択可能な値は 32、64、128、248 です。

注意事項：

- 1 つの EtherNet/IP バスデバイスのみ作成可能です。個数を超えるとエラーが表示されます。
- レジスタアドレスはすべて 0 から開始します。32 を選択すると、アドレス 0～31 のレジスタを持つことを意味します。レジスタマッピングの設定時には、アドレス範囲に注意する必要があります。

- 読み取り専用、書き込み専用は xCore 制御システム側の視点での説明です。読み取り専用レジスタは EtherNet/IP マスター側の Output データに対応し、書き込み専用レジスタは EtherNet/IP マスター側の Input データに対応します。
- 読み取り専用、書き込み専用レジスタの数は通信データ量を表しています。データ量が多いほど通信負荷が大きくなりますので、必要に応じてできるだけ小さなレジスタ数を選択することを推奨します。
- マスター側で設定する Input および Output データのバイト数は、スレーブ側の書き込み専用および読み取り専用レジスタのバイト数と一致させる必要があります。そうでない場合、通信が失敗する可能性があります。
- EtherNet/IP バスデバイスの読み取り専用および書き込み専用レジスタは、それぞれ独立したデータ領域として管理されています。読み取り専用、書き込み専用レジスタのアドレスは重複しても問題ありません。
- xCore 制御システムの EtherNet/IP スレーブは、デフォルトで EDS ファイルを提供していません。Input Assembly Instance ID は 1、Output Assembly Instance ID は 2 です。マスター側で設定する際に一致させる必要があります。そうでない場合、通信が失敗する可能性があります。

10.5 レジスタ

10.5.1 レジスタの概要

レジスタは、ロボット内で利用可能な変数を表します。一般にロボットの制御やロボット状態を取得する周辺機器とのデータ交換に使用されます。レジスタは現在の RL プロジェクトの変数としても使用でき、命令型と代入型の 2 種類の操作方法をサポートしています。

- レジスタはロボット自体の概念であり、バスデバイスに属するものではありません。あるレジスタは、新規作成または編集時に、どのバスデバイスに割り当てるかを指定することができ、それによって通信やデータ交換を行います。
- 各レジスタは 2 バイトを占めます。異なるタイプの変数は、異なる数のレジスタを占有します。

10.5.2 レジスタのパラメーター設定

「通信」→「レジスタ」画面では、既存のレジスタを確認したり、追加、編集、削除を実行することができます。

レジスタ

デバイス: すべて タイプ: すべて 名前: 説明: リセット: すべて削除

名前	タイプ	開始レジスタ	読み書き属性	保持レジスタ	長さ	ビットオフセット	バイトオフセット	終了レジスタ	デバイス名	機能	説明
1 cmd_set	int16	40000	読み取り専用	いいえ	1			40000	modbus_0	ext_cmd_set	
2 response_data	int16	40005	書き込み専用	いいえ	8			40012	modbus_0	ext_response_data	
3 clear_alarm	int16	40001	読み取り専用	いいえ	1			40001	modbus_0	ctrl_clear_alarm	
4 register0	int16	40002	読み取り専用	はい	1			40002	modbus_0		
5 register1	bit	40003	読み取り専用	いいえ	1	3		40003	modbus_0		

①

②

③

①	レジスタ一覧です。
②	レジスタのインポート/エクスポートボタンです。タップするとレジスタのインポート/エクスポート画面に切り替わり、インポート/エクスポート操作を実行できます。
③	レジスタ操作ボタンです（左から順に新規作成、編集、削除）。

名前	タイプ	開始レジスタ	読み書き属性	保持レジスタ	長さ	ビットオフセット	バイトオフセット	終了レジスタ	デバイス名	機能	説明
1 cmd_set	int16	40000	読み取り専用	いいえ	1			40000	modbus_0	ext_cmd_set	
2 response_data	int16	40005	書き込み専用	いいえ	8			40012	modbus_0	ext_response_data	

レジスタ一覧の各パラメーターの説明は下表の通りです。

パラメーター	説明
名前	RL では、この名前でレジスタにアクセスします。 注意：RL 内の変数の衝突による予測不可能な結果を避けるために、一覧のレジスタまたは変数と同じ名前を付けないでください。
タイプ	現在サポートされているのは bit、bool、byte、int16、int32、float です。
開始レジスタ	同じバスデバイス内の同じ読み書き属性のレジスタアドレスは、重複して使用できません。また、Modbus バスデバイス内では、異なる読み書き属性のレジスタアドレスも重複して使用することはできません。 例えば、41000～41003 を占有するレジスタを定義した場合は、41002 から始まる別のレジスタを定義できなくなります。
読み書き属性	読み書き属性は、ロボットの視点（マスターやスレーブの視点ではなく）から見て、このレジスタが読み取り操作の対象か、または書き込み操作の対象かを示します。書き込み専用レジスタは、ロボットが外部に状態を出力するために使用されます。読み取り専用レジスタは、ロボットが周辺機器から送信された命令を取得するために使用されます。
保持レジスタ	レジスタが PERS に設定されると、ロボットの再起動、シャットダウン、停電、RL 停止の時に、そのレジスタ値が不揮発性記憶媒体に保持されます。ロボット電源の再投入または RL の再実行の時は、このレジスタ値はシャットダウンまたは RL 停止の前の保持値になります。

長さ	長さは変数の数量を示します。1 より大きい変数の場合は、配列を使用して変数を参照できます。添え字は 1 から始まります。 注意：この項目はレジスタの数とは区別する必要があります。 例：レジスタ 40140～40153 で変数型が float の場合 各 float が 2 つのレジスタを占有し、長さが 7 であれば、占有されているレジスタ数は $2 \times 7 = 14$ となります。
ビットシフト	ビットシフトは、bit 型のレジスタがレジスタ内でどの位置にマッピングされるかを示します。1 つのレジスタは 2 バイト（16 ビット）を占有します。ビットシフトは対応するレジスタの位置で、シフト量は 1～16 です。bit 型のレジスタを作成するとき、要素数が 1 の場合はシフト量を設定できますが、そうでない場合は設定できません。
バイトシフト	バイトシフトは、byte 型のレジスタがレジスタ内でどの位置にマッピングされるかを示します。1 つのレジスタは 2 バイト（16 ビット）を占有します。byte 型変数に必要なのは 8 ビットであるため、byte 型のレジスタ変数を作成するときは、レジスタの下位バイト（1-8）または上位バイト（9-16）のどちらかにマッピングするかを選択する必要があります。
終了レジスタ	終了レジスタは、レジスタが占有する最後のレジスタアドレスを示します。レジスタ変数が連続して配置されている場合は、ユーザーはこの値によりレジスタが占有するスペースをすばやく把握できます。 例：この項目の値に 1 を加えることで次のレジスタの開始アドレスが求められます。
デバイス名	デバイス名は、「バスデバイス」が作成されたときに定義された名前です。レジスタがどのバスデバイスに割り当てられているかを示します。CC-Link、CC-Link IE Field Basic、Modbus、EtherCAT、EtherNet/IP デバイスにレジスタを割り当てることをサポートしています。
機能	この列の内容は、このレジスタに対応するロボット機能を示す固定の機能コードです。機能コードは読み取り専用と書き込み専用に分かれています。詳細についてはレジスタの機能コードの章を参照してください。

10.5.3 レジスタの種類

種類	説明
bit	レジスタ内の 1 ビットのみを占有します。bit 配列は 16 ビットの倍数で設定する必要があります。 例：開始アドレスが 41000 の bit 型レジスタで長さが 64 の場合、41000～41003 の 4 つのレジスタを占有します。
byte	レジスタ内の 8 ビットを占有します。下位アドレスのビット（レジスタの最初の 8 ビット）または上位アドレスのビット（レジスタの最後の 8 ビット）のどちらを占有するかを選択できます。byte 型配列を作成する場合は、デフォルトで下位アドレスのビットを占有し、上位アドレスのビットと下位アドレスのビットを変更できません。
bool	1 つのレジスタを占有します。
int16	1 つのレジスタを占有します。
float	2 つのレジスタを占有します。
int32	2 つのレジスタを占有します。 注意：デバイスタイプが PROFINET の場合は、作成はサポートされません。

bit 型レジスタの注意点：

レジスタを編集：register1

デバイス
タイプ： MODBUS 名前： modbus_0

基本情報
名前： register1 読み書き属性： ☒ 読み取り専用 ☐ 書き込み専用
タイプ： bit 保持レジスタ： ☐ はい ☒ いいえ
開始レジスタ： 40003 ① 長さ： 1
② ビットシフト： ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8
☐ 9 ☐ 10 ☐ 11 ☐ 12 ☐ 13 ☐ 14 ☐ 15 ☐ 16
説明：
機能
機能コード：
キャンセル 前へ 次へ

①	長さ：
②	ビットシフト： 上図に示すように、要素数が 1 の場合はレジスタの特定の 1 ビットを占有します。 ビットシフトで 1～16 が選択できます。

bit 型レジスタの長さが 1 より大きい場合、すなわち配列の場合は、シフト量の設定や機能の割り当てはできません。その場合は、ビットシフトの設定部分は自動的に非表示になり、シフト量は 1 に設定されます。

byte 型レジスタの注意点：

レジスタを編集：register1

①	長さ：
②	バイトアドレス：

上図のように、長さが 1 の場合は、レジスタの 8 ビットを占有します。バイトアドレスを設定でき、下位バイト（1～8）と上位バイト（1～16）が選択できます。

byte 型レジスタの要素数が 1 より大きい場合。すなわち配列の場合は、バイトアドレスの設定が許可されず、デフォルトで下位アドレスとなります。

注意：一時停止機能に割り当てられているレジスタやシステム IO がリセットされていない場合、レジスタを通してのプログラムの起動はできません。

10.5.4 レジスタの機能コード

10.5.4.1 読み取り専用機能コード

読み取り専用タイプは、制御信号に使用されます。通常、周辺機器からロボットに制御信号を送信し、ロボットが対応する動作を実行します。ロボットにとっては、これらのレジスタは読み取り専用です。現在サポートされている制御信号は次の通りです。

機能コード名	割り当て可能な型	対応する機能
空白	なし	機能なし、ユーザー定義入力。
ctrl_clear_alarm	bit/bool/byte/int16	サーボアラームのクリア。
ctrl_estop_reset	bit/bool/byte/int16	非常停止リセット。
ctrl_jjwc_A	bit/bool/byte/int16	トリガータイプはパルストリガーで、High で有効になります。信号が 0 から 1 にかわると、ロボットは停止します。この信号によってトリガーされた停止は実行を継続できなくなり、pptomain、pptofunc または ppto カーソルの実行後にのみ再実行できます。ロボットが停止した後、sta_jjwc_B 信号が 1 (High) に設定されます。信号が 1 から 0 になると、sta_jjwc_B 信号は 0 (Low) に設定されます。
ctrl_motor_off	bool/byte/int16	モーターオフを実行します。

ctrl_motor_on	bool/byte/int16	モーターオンを実行します。
ctrl_motor_on_off	bit/bool/byte/int16	モーターのオン/オフを行います。 1：モーターオン 0：モーターオフ
ctrl_motoron_pptomain_start	bit/bool/byte/int16	モーターオン、main へのポインタの移動、プログラム起動を順次実行します。
ctrl_motoron_start	bit/bool/byte/int16	モーターオン、プログラム起動を順次実行します。
ctrl_pause_motoroff	bit/bool/byte/int16	プログラムを一時停止してモーターオフを実行します。
ctrl_pptomain	bit/bool/byte/int16	ポインタを main に移動します。
ctrl_program_start	bool/byte/int16	RL プログラムの実行を開始します。
ctrl_program_start_stop	bit/bool/byte/int16	プログラムの実行/停止を行います。
ctrl_program_stop	bool/byte/int16	RL プログラムを停止します。
ctrl_set_program_speed	bit/bool/byte/int16	プログラムの実行速度を設定します。
ctrl_soft_estop	bit/bool/int16	ロボットのソフトウェア非常停止を制御します。 1：ソフトウェア非常停止をトリガーしない。 0：ソフトウェア非常停止をトリガーする。
ctrl_switch_auto_motoron	bit/int16/byte/bool	自動モードに切り替えてから、モーターオンを実行します。
ctrl_switch_operation_auto	bool/byte/int16	自動モードに切り替えます。
ctrl_switch_operation_auto_manu	bit/bool/byte/int16	自動モードと手動モードを切り替えます。 1：自動モード 0：手動モード
ctrl_switch_operation_manu	bool/byte/int16	手動モードに切り替えます。
enable_safe_region01~enable_safe_region10	bool/byte/int16	対応する安全エリアを有効にします。
ext_cmd_set	bit/bool/int16	リモートコントロール機能： 「リモートコントロール」を参照ください。
ext_request_data	int16 配列	リモートコントロール機能： 命令機能コードです。配列で、固定長 8 のレジスタです。
ext_reset	bit/bool/int16	リモートコントロール機能： 全体機能をリセットします。 「リモートコントロール」を参照ください。
ext_resp_get	bit/bool/int16	リモートコントロール機能： 前の命令の応答を確認し、レスポンスをクリアします。
ctrl_estop_reset_and_clear_alarm	bit/bool/byte/int16	非常停止状態のリセットとアラームのクリアを行います
ctrl_reduced_mode	bit/bool/byte/int16	協働モードをトリガーします

説明：上記のシステムレジスタにおいて、すべてのシステム入力はパルストリガーです。
xCore システムが外部命令を正しく受信できるように、外部入力のパルス幅が 60 ミリ秒以上
であることを確認してください。

	前提条件/注意事項	
モーターオン	ロボットが自動モード アラームの発生無し	ロボット モーター電源投入シーケンス
モーターオフ		ロボット モーター電源オフシーケンス
プログラム起 動	ロボットが自動モード プログラムポインタがある モーターが電源オン状態 アラームの発生無し	ロボット プログラム起動シーケンス
プログラム一 時停止	外部機器がレジスタを介 してプログラム一時停止 (モーターオフしない) の 操作を送信	ロボット 一時停止シーケンス
非常停止	非常停止を押すか、または 外部非常停止のタイミ ングロジックをトリガ ーします。 (注意：非常停止を押 すと、プログラムが停 止し、モーター電源 がオフになり、ポ インターが失われる ため、次の起動時 にはまずアラーム クリア、モーター 電源オン、main へのポインターの 移動を実行する必 要があります。)	ロボット 非常停止シーケンス
ロボットの起 動プロセス	操作手順： (1) ロボットを手動モ ードに切り替えます (2) アラームをクリア します (サーボアラ ームまたはコント ローラーのエラー アラームをクリア します) (3) モーターをオン にします (4) ポインタを main に移動します (プ ロジェクトが大 きいほど、pptomain 命令にかかる時 間が長くなるた め、実行時間を 2 秒ほど	ロボット 起動全体シーケンス

	確保し、命令実行後にプログラム起動信号を送信することを推奨します) (5) プログラムを起動します	
--	---	--

10.5.4.2 書き込み専用機能コード

書き込み専用タイプは主に状態信号に使用されます。状態信号はロボットが外部に送信する信号であり、モーター電源投入状態やプログラム実行状態などのロボットの状態を外部にフィードバックします。ロボットにとって、このレジスタは書き込み専用です。書き込み専用レジスタには、この状態信号を割り当てることができます。現在サポートされている状態信号は次の通りです。

機能コード名	割り当て可能な型	対応する機能
空白		機能なし、ユーザー定義出力。
ext_error_code	int16	リモートコントロール機能： エラーコード
ext_resp_set	bit/bool/ int16	リモートコントロール機能： 命令実行完了時の応答
ext_response_data	int16 配列	リモートコントロール機能： フィードバックが必要なデータです。配列で、固定長 8 のレジスタです。
sta_alarm	bit/bool/byte/int16	サーボアラーム状態： 1：サーボアラームあり 0：アラームなし
sta_board_DI0~sta_board_DI3	bit/bool/byte/int16	自社開発の IO ボードと実際の IO の各信号の状態をリアルタイムで出力します。
sta_board_DO0~sta_board_DO3	bit/bool/byte/int16	自社開発の IO ボードと実際の IO の各信号の状態をリアルタイムで出力します。
sta_collision	bit/bool/byte/int16	衝突検知状態： 1：衝突を検知した 0：衝突なし。
sta_collision_alarm	bool/byte/int16	衝突検知アラーム： 1：衝突を検知した 0：衝突なし アラームをクリアすることでのみ。このアラームをクリアできます。
sta_collision_open	bool/byte/int16	衝突検知が有効な状態。
sta_controller_is_running	bool/byte/int16	コントローラーの起動信号： 1：コントローラーは稼働中 0：コントローラーは未稼働
sta_encoder_low_battery	bool/byte/int16	エンコーダーの電圧低下アラーム状態。
sta_error_code	int16	ロボットのエラーコードを報告します。このエラーコードとロボットログのエラーコードとの差は 30000 です。

		例：ロボットログの「動作範囲外」のエラーコード番号は 50002 です。 <code>sta_error_code</code> によって取得される場合は 20002 となります。
<code>sta_estop</code>	bit/bool/byte/int16	非常停止状態： この値は非常停止トリガーレベル設定に影響されます。 High に設定されている場合： 1：非常停止がトリガーされている 0：トリガーされていない Low に設定されている場合： 0：非常停止がトリガーされている 1：トリガーされていない
<code>sta_heartbeat</code>	bit/bool/byte/int16	ハートビート信号： 書き込み専用です。「設定—コントローラー設定」でハートビートサイクルを設定します。
<code>sta_home</code>	bit/bool/byte/int16	ロボットフランジ中心がホーム位置にあるかどうか： 1：ホーム位置にある 0：ホーム位置にない。
<code>sta_jjwc_B</code>	bit/bool/byte/int16	リアルタイム状態出力： トリガー動作はバッシュプトリガーです。 <code>ctrl_jjwc_A</code> 信号が 0 から 1 になった場合に、<code>sta_jjwc_B</code> 信号が 1 に設定されます。<code>ctrl_jjwc_A</code> 信号が 1 から 0 になった場合は、<code>sta_jjwc_B</code> 信号が 0 に設定されます。
<code>sta_motor</code>	bit/bool/byte/int16	モーターオン状態： 1：モーター電源オン 0：モーター電源オフ
<code>sta_operation_mode</code>	bit/bool/byte/int16	現在の操作モード 1：自動モード 0：手動モード
<code>sta_program</code>	bit/bool/byte/int16	現在プログラムが実行しているかどうか： 1：プログラム実行中、0：実行していない
<code>sta_program_full</code>	byte/int16	RL の一時停止状態、 0：初期化状態 1：RL 動作中 2：HMI 一時停止 3：システム IO 一時停止 4：レジスタ機能コード一時停止 5：外部通信一時停止 6：SDK 一時停止 7：Pause 命令による一時停止 10：非常停止 11：セーフティーゲート（CR シリーズ及び SR シリーズでは停止エリア入力状態）； 12：その他の要因による一時停止
<code>sta_program_not_run</code>	bool/byte/int16	RL プログラムが実行しているか： 1：プログラムが実行中

		0：プログラムが実行していない
sta_program_reset	bool/byte/int16	プログラムリセット成功信号、現在のプログラムポインターが main 関数の最初の行にあり、プログラムが実行を開始していない場合は 1 を出力し、そうでない場合は 0 を出力します。
sta_program_speed	int16	現在のプログラム実行速度をパーセント値でクエリします。
sta_robot_is_busy	bit/bool/byte/int16	ロボットが現在 pptomain などの時間のかかる操作を実行しているかどうか： 1：実行中 0：実行中ではない
sta_robot_moving	bit/bool/byte/int16	ロボットが動作しているかどうか： 1：ロボットが動作中 0：ロボットが静止中 RL プログラムの動作命令とジョグの時のみ、ロボットの動作状態を検出します。 (注意：同定、ドラッグ、力制御、ドラッグ再生などの場合には、ロボットが動作していても、この出力はロボット静止となります。)
sta_safe_door	bit/bool/byte/int16	セーフティーゲートをオンにすると、割り当てられたレジスタの信号出力が有効になります。セーフティーゲートをオフにすると、レジスタの信号出力が無効になります。 (High で有効、Low で無効)。CR シリーズ及び SR シリーズでは停止エリア入力状態を出力します。
sta_safe_jnt_pos1~sta_safe_jnt_pos8	bool/byte/int16	安全位置のトリガー状態 1：安全位置に到達した 0：安全位置に到達していない
sta_safe_region01~sta_safe_region10	bool/byte/int16	安全エリアのトリガー状態
sta_soft_estop	bit/bool/int16	ソフトウェア非常停止状態の出力。この値は非常停止トリガーレベル設定に影響されます。 High に設定されている場合は、ソフトウェア非常停止状態がトリガーされているときに 1、トリガーされていないときに 0 になります。 Low に設定されている場合は、トリガーされているときに 0、トリガーされていないときに 1 になります。
sta_cart_pose	float 配列	ロボットの現在の直交ポーズ： レジスタの要件：float 配列、長さは 7
sta_cart_vel	float 配列	ロボットの直交速度。
sta_jnt_pose	float 配列	ロボットの現在の関節角度： レジスタの要件：float 配列、長さは 7
sta_jnt_trq	float 配列	ロボットの現在の関節トルク： レジスタの要件：float 配列、長さは 7、単位は N.m
sta_jnt_vel	float 配列	ロボットの現在の関節速度：

		レジスタの要件：float 配列、長さは 7、単位は rad/s
sta_tcp_pose	float 配列	ロボットの TCP ポーズ： レジスタの要件：float 配列、長さは 7
sta_tcp_vel	float 配列	ロボットの TCP 速度： レジスタの要件：float 配列、長さは 7
sta_tcp_vel_mag	float	ロボット TCP の合成直線速度
sta_ext_estop	Bool/byte/int16	外部非常停止状態 1：外部非常停止状態 0：外部非常停止状態ではない セーフボードは mini ボードで、ファームウェアのバージョンは 1.0.8.7 以上。
sta_hand_estop	bool/byte/int16	携帯型安全装置の非常停止状態 1：外部非常停止状態 0：外部非常停止状態ではない セーフボードは mini ボードで、ファームウェアのバージョンは 1.0.8.7 以上。
sta_sys_stop_di	bit/bool/byte/int16	システム停止信号の値を出力します。「プログラム停止」と「プログラム停止 1」のいずれかの信号がトリガーされると、sta_sys_stop_di は 1 を出力します。両方の信号がトリガーされない場合は 0 を出力します。
sta_reduced_mode	bit/bool/byte/int16	ロボットが協働モードかどうか

10.5.5 レジスタの例

制御システムは、レジスタの読み取りと変更を実行するために、命令型と代入型の 2 つの方法を提供します。

命令型には `WriteRegByName` と `ReadRegByName` の 2 つの命令があります。代入は演算子「`=`」を使用します。

10.5.5.1 命令型

`WriteRegByName(modbus_reg[index], rl_symbol)`

`modbus_reg` は「通信」→「レジスタ」で設定されたレジスタの名前です。`[index]`を使って対応するレジスタの先頭アドレスをシフトすることができます。`index` 範囲は`[1, レジスタの最大長]`で、デフォルトは 1 です。

制御システム内のデータは、レジスタを介して割り当てられているデバイスに出力できます。

例： `int` 型の `rl_value` という変数を定義し、それを周辺機器に出力したい場合

RL 言語で `WriteRegByName` 命令を使用し、レジスタ「`mtcp_wo_i`」の最初のレジスタを指定することで、値が「`mtcp_wo_i`」に関連付けられた周辺機器に送信されます。

```
1 GLOBAL PROC main()
2   int rl_value = 0
3   while(true)
4     // 生産サイクル
5     // ロボットの動作命令
6     // ...
7     // ...
8
9
10    // カウント+1
11    rl_value += 1
12    WriteRegByName(mtcp_wo_i[1], rl_value)
13  endwhile
14
```

`ReadRegByName(modbus_reg[index], rl_symbol)`

この命令は `WriteRegByName` に似ており、RL プログラム変数をレジスタの値に更新するために使用されます。(例えば、RL プログラムの実行フローや動作パラメーターなどを制御します。)

10.5.5.2 代入型

演算子「`=`」を直接使用します。例えば、「`mtcp_wo_i[1] = 1`」は、レジスタ `mtcp_wo_i` の最初の要素の値を 1 に更新します。同様に、「`a = mtcp_wo_i[1]`」は、変数 `a` をレジスタ `mtcp_wo_i` の最初の要素の値に更新します。

10.5.6 レジスタのリモートコントロール

リモートコントロールは、異なる機能を持つ 7 つのレジスタを組み合わせ使用します。特定のタイミングで、複雑な動作ロジックの相互作用を完了することができます。外部機器は、リモートコントロール機能を使用して、ロボットのジョグ、位置の更新、ロボット位置や状態の取得などの機能を実行できます。

周辺機器がロボットを制御する場合：

4 種類のレジスタを使用する必要があります。ロボットにとっては読み取り専用レジスタです。

機能コード名	属性	型	長さ	対応する機能
ext_cmd_set	読み取り専用	int16 bool bit	1	命令の送信： 1. 命令実行時に、1 に設定することでリクエストします。命令が 1 の時にのみリクエストに応答します。 2. 誤動作防止のために、必ず命令データをデータエリアに設定してから実行してください。（データはキャッシュに一時的に格納され、このビットが 1 になった時にのみデータに応答します）。 3. 命令実行後、このビットをクリアします（0 にする）。
ext_reset	読み取り専用	int16 bool bit	1	機能リセット： 1. リモートコントロール機能を有効にするための信号です。機能使用中は、常にこのレジスタを 1 にしておいてください。 2. 値が 0 の時は、機能は停止しています。 3. この信号は、インターフェース機能の異常によるリセットや動作命令の中断が発生した場合にも使用されます。
ext_resp_get	読み取り専用	int16 bool bit	1	前回の命令の応答を確認してクリアし、ext_resp_set 信号を 0 にリセットします
ext_request_data	読み取り専用	int16	8	命令機能コードです。配列、固定長 8 のレジスタです。機能の詳細については、機能コードを参照ください。

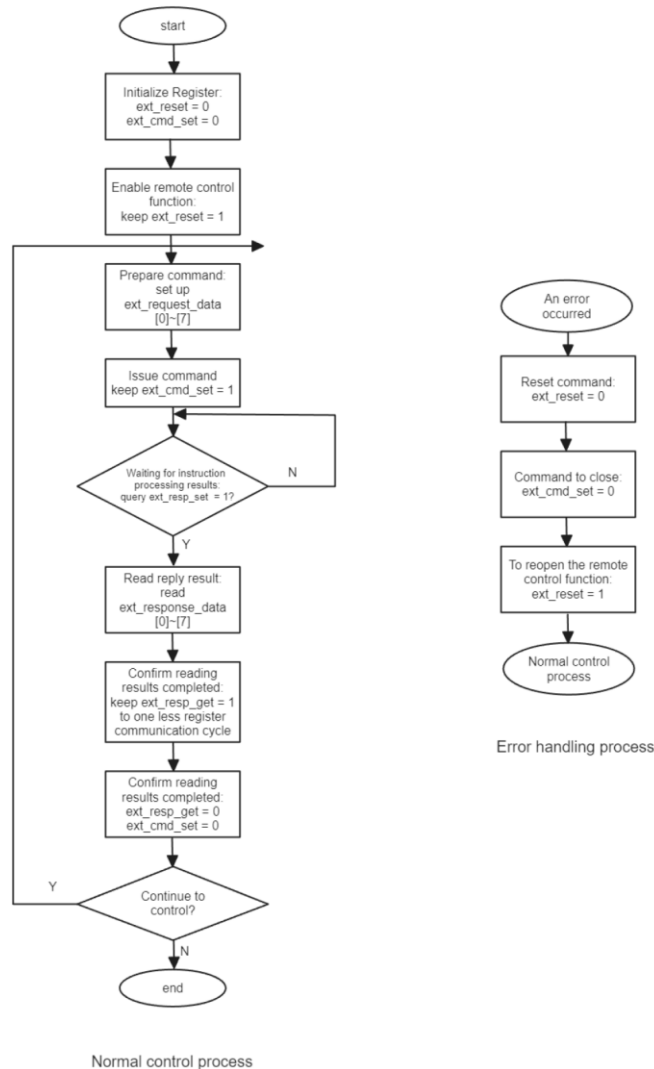
周辺機器がロボットの状態を取得したい場合：

3 種類のレジスタを使用する必要があります。ロボットにとっては書き込み専用レジスタです。

機能コード名	属性	型	長さ	対応する機能
ext_error_code	書き込み専用	int16	1	リモートコントロール機能：エラーコード。
ext_resp_set	書き込み専用	int16 bool bit	1	ロボットが制御命令に応答した後、このレジスタを 1 に設定し、命令の実行が完了したことを示します。
ext_response_data	書き込み専用	int16	8	リモートコントロール機能：フィードバックが必要なデータ。配列で、固定長 8 のレジスタです。

10.5.6.1 使用フロー

7 つのレジスタの連携および制御フローは下図のようになります。



10.5.6.2 命令形式

命令と応答は、それぞれ 8 つのレジスタを使用して実現します。

命令信号 `ext_request_data` (8 つのレジスタを占有、`reg0` — `reg7`) は、命令と関連パラメータのデータ信号領域を指定するために使用されます。命令は複数のワードで構成されます。

(1) ワード：1 つのレジスタ (16 ビット) で構成されます。

(2) 命令形式：最大 8 ワードで構成されます。使用する命令に応じて変更されます。最小の命令は 1 ワードです。

命令番号	命令番号 1	命令番号 2		命令番号 7
------	--------	--------	-------	--------

応答信号 `ext_response_data` (8 つのレジスタを占有、`reg0` — `reg7`) は、応答のデータ信号領域を取得するために使用されます。応答は複数のワードで構成されます。

(1) ワード：1 つのレジスタ (16 ビット) で構成されます。

(2) 応答形式：最大 8 ワードで構成されます。受信した命令に応じて変更されます。最小の応答は 1 ワードです。ただし、異常な応答の場合は 3 ワードを占有します。

命令番号	応答 1	応答 2		応答 7
------	------	------	-------	------

具体的に使用可能な命令番号は下表の通りです。

命令種類	説明	命令コード	命令長	
			命令	応答
JOG	ジョグ座標系の設定	1	2	3
	ジョグ座標系の取得	2	1	4
	ジョグ速度の設定	3	2	3
	ジョグ速度の取得	4	1	4
	ジョグステップ長の設定	5	2	3
	ジョグステップ長の取得	6	1	4
	ジョグ開始	7	4	2
	ジョグ停止 (パラメーターなし)	8	1	2
	位置を更新	9	2	2
	位置に動作	10	2	2
情報設定	ツールを設定	11	2	3
	現在のツール番号の取得	12	1	4
	ワークを設定	13	2	3
	現在のワーク番号の取得	14	1	4

10.5.6.3 命令の詳細

1) ジョグ座標系の設定：

命令/応答	命令コード	パラメーター1	パラメーター2
命令	1	座標系： 1：軸座標系 2：ワールド座標系 3：フランジ座標系 4：ベース座標系 5：ツール座標系 6：ワーク座標系	N/A
応答	1	設定結果： 0：成功 1：失敗	エラーコード

2) ジョグ座標系の取得：

命令/応答	命令コード	パラメーター1	パラメーター2	パラメーター3
命令	2	N/A	N/A	N/A
応答	2	取得結果 0：成功 1：失敗	エラーコード	座標系： 1：軸座標 2：ワールド座標系 3：フランジ座標系 4：ベース座標系 5：ツール座標系 6：ワーク座標系

3) ジョグ速度の設定：

命令/応答	命令コード	パラメーター1	パラメーター2
命令	3	ジョグ速度（1～100）	N/A
応答	3	設定結果 0：成功 1：失敗	エラーコード

4) ジョグ速度の取得：

命令	命令コード	パラメーター1	パラメーター2	パラメーター3
命令	4	N/A	N/A	N/A
応答	4	取得結果： 0：成功 1：失敗	エラーコード	ジョグ速度 （1～100）

5) ジョグステップ長の設定：

命令/応答	命令コード	パラメーター1	パラメーター2
命令	5	1：連続 2：10mm ステップ長 3：1mm ステップ長 4：0.1mm ステップ長 5：0.01mm ステップ長	N/A
応答	5	設定結果： 0：成功 1：失敗	エラーコード

6) ジョグステップ長の取得：

命令	命令コード	パラメーター1	パラメーター2	パラメーター3
命令	6	N/A	N/A	N/A
応答	6	取得結果： 0：成功 1：失敗	エラーコード	1：連続 2：10mm ステップ長 3：1mm ステップ長 4：0.1mm ステップ長 5：0.01mm ステップ長

7) ジョグ開始：

この命令は命令コード 1（ジョグ座標系の設定）に依存します。軸座標の場合は、パラメーター1の値は関節番号（軸 1～軸 7、1 は軸 1、7 は軸 7）を示します。直交座標の場合は、(x,y,z,a,b,c,elb) の番号（1 は x、7 は elb）を示します。

命令/応答	命令コード	パラメーター1	パラメーター2
命令	7	動作モード： 軸座標は関節番号 直交座標は (x,y,z,a,b,c,elb)	ジョグ方向： 1：負の方向 2：正の方向
応答	7	結果： 0：成功 1：失敗	エラーコード

8) ジョグ停止：

命令/応答	命令コード	パラメーター1
命令	8	N/A
応答	8	結果： 0：成功 1：失敗

9) 位置を更新：

命令/応答	命令コード	パラメーター1	パラメーター2
命令	9	RL プロジェクト・位置一覧の番号	N/A
応答	9	結果： 0：成功 1：失敗	エラーコード

10) 位置に動作：

命令/応答	命令コード	パラメーター1	パラメーター2
命令	10	動作モード： 1：MoveAbsj 2：MoveJ 3：MoveL	RL プロジェクト・位置一覧 の番号
応答	10	結果： 0：成功 1：失敗	エラーコード

11) ツールを設定：

命令/応答	命令コード	パラメーター1	パラメーター2
命令	11	RL プロジェクト・ツール一覧 の番号	N/A
応答	11	設定結果： 0：成功 1：失敗	エラーコード

12) 現在のツール番号の取得：

命令/応答	命令コード	パラメーター1	パラメーター2	パラメーター3
命令	12	N/A	N/A	N/A
応答	12	取得結果： 0：成功 1：失敗	エラーコード	現在のツール番号

13) ワークを設定：

命令/応答	命令コード	パラメーター1	パラメーター2
命令	13	RL プロジェクト・ワーク一覧 の番号	N/A
応答	13	設定結果： 0：成功 1：失敗	エラーコード

14) 現在のワーク番号の取得：

命令/応答	命令コード	パラメーター1	パラメーター2	パラメーター3
命令	14	N/A	N/A	N/A
応答	14	取得結果： 0：成功 1：失敗	エラーコード	現在のワーク番号

10.5.6.4 エラーコード

命令設定を行う際、パラメーターの誤りやロボットの状態不一致、その他の理由により設定が失敗することがあります。この場合、エラーメッセージを通じてロボットの問題を確認することができます。

制御システムには 3 種類のエラーコードがあります。

- 命令実行結果 (`ext_response_data` 内) のエラーコード
- `ext_error_code` : 命令の実行ができない場合に発生するエラーコードです。(ロボットがビジー状態、リモート制御フラグが正しくないなど)
- `sta_error_code` : ロボットのエラーコードです。(ロボット動作中に停止エラーが発生した場合は、このレジスタで読み出すことができます)。

通常、ユーザーは次の手順でエラーコードを確認する必要があります。

- 実行命令 (`ext_cmd_set=1`) を送信した後は、まず `ext_error_code` を読み取ります。エラーコードがなければ `ext_repose_data` の戻り値を読み取ります。戻り値が 0 でなければ `ext_repose_data` のエラーコードを読み取ります。
- 動作 (ジョグや位置に移動) の場合は、上記の戻り値が正常であっても、`sta_error_code` を読み取り、動作中にエラーによる停止 (特異点やオーバートラベルなど) がないかどうかを確認する必要があります。

`ext_error_code` の意味 :

エラーコード	意味	備考
01	サポートされていない命令	
02	パラメーターが不正	
03	制御フラグが正しくない	確認必要 : <code>ExtRespSet</code> が 0、 <code>ExtRespSet</code> が 1
04	ロボットがビジー	ロボットが命令を実行中は、他の命令への応答はできません
05	対応する番号が見つからない	位置、ツール、ワークの ID
06	位置のタイプが動作のタイプと一致しない	ある位置に動作する命令の場合、位置のタイプが動作のタイプと一致していません。例えば、軸座標は <code>MoveAbsJ</code> 命令のみ使用でき、直交座標は <code>MoveJ</code> または <code>MoveL</code> のみ使用できます。
07	入力した軸数が機種と一致しない	
11	手動/自動モードが正しくない	
12	ロボットの状態が正しくない。ジョグ状態であるか確認してください。	ロボットがドラッグ状態などのときはジョグさせることができません。
13	モーター電源投入状態が正しくない	モーター電源投入後のみジョグできます。
14	ロボットが位置制御モードでないため、ジョグができない	12 と同様です。

15	ジョグの有効化に失敗し、アルゴリズムがエラーを報告	様々な理由でジョグができない場合にこのエラーを報告します。
20	特異点に遭遇した	
21	既に目標点に到達した	ロボットが既に目標位置に動作完了しており、再度その位置に動作しようとした場合にこのエラーを報告します。

10.5.7 レジスタのインポートとエクスポート

レジスタのインポートとエクスポート機能により、あるロボットのレジスタ構成を別のロボットに素早くコピーすることができます。

10.5.7.1 レジスタのエクスポート

レジスタ画面で、左下のボタン   をタップしてレジスタのエクスポート画面に入ります。

レジスタをエクスポート

① **ファイルを選択**
 ファイル(*.xml) : C:/Users/mech1/backup.xml 参照

項目の選択
☒ すべて選択

	名前	説明
1	☒ cmd_set	
2	☒ response_data	
3	☒ clear_alarm	
4	☒ register0	
5	☒ register1	

②

③ キャンセル 前へ 次へ

①	エクスポートする先のファイル名（パスを含む）を指定
②	エクスポートするレジスタの選択リスト
③	エクスポート操作ボタン。

エクスポート手順：

まず①でレジスタのエクスポートのオブジェクトファイルパスを入力または選択します。次に②のレジスタ一覧でエクスポートするレジスタにチェックを入れます。最後に③の「次へ」ボタンをタップしてエクスポートを実行します。①の対応するパスに、エクスポートされたレジスタのファイルが生成されます。

10.5.7.2 レジスタのインポート

レジスタ画面で、左下のボタン  をタップしてレジスタのインポート画面に入ります。

レジスタをインポート

① ファイルを選択
ファイル(*.xml) : C:/Users/mech1/backup.xml 参照

② インポートオプション
インポートで競合するレジスタへの対応は以下を選択します。競合するレジスタを置き換えるか、インポートしないかを選択できません。
☒ 上書き ☐ インポートしない

③ 項目の選択
☒ すべて選択

	名前	競合の有無	説明
1	cmd_set	あり	
2	response_data	あり	
3	clear_alarm	あり	
4	register0	あり	
5	register1	あり	

④ キャンセル 前へ 次へ

①	インポートするレジスタのファイル名（パスを含む）を指定
②	すでに同じレジスタが存在するなど、競合した時の対処法を指定
③	インポートするレジスタの選択リスト
④	インポート操作ボタン。

インポート手順：

まず①でインポートするファイルを選択します。②でレジスタが競合した場合の対処法を設定します。③でインポートするレジスタを選択します。最後に④の「次へ」をタップしてインポートを実行し、選択したレジスタをインポートします。

10.5.7.3 レジスタインポート時の競合チェック

同じデバイスで、同じレジスタ属性（読み書き）の場合、レジスタアドレスは重複することはできません。

重複した場合：

インポートオプションが「インポートしない」に設定されている場合は、元のレジスタが優先され、競合するレジスタはインポートされません

インポートオプションが「上書き」に設定されていれば、新しくインポートされたレジスタが優先され、元のレジスタは上書きされます。

競合がある時は現在のレジスタを置き換えるかどうかの選択をユーザーに促すポップアップウィンドウが表示されます。

ext で始まる 7 つのレジスタ（ext_cmd_set、ext_resp_set、ext_resp_get、ext_reset、ext_response_data、ext_request_data、ext_error_code）が作成される際にすでにレジスタアドレスに割り当てられている場合は、他のレジスタに割り当てできなくなります。上記の 7 つのレジスタをインポートする際に、該当する機能コードが既に HMI で割り当てられて使用されており、新しくインポートされたレジスタ一覧にもこの機能コードがある場合は、新しくインポートされたものが優先され、元の競合するレジスタは上書きされます。競合がある時は現在のレジスタを置き換えるかどうかの選択をユーザーに促すポップアップウィンドウが表示されます。

10.6 IO デバイス

10.6.1 概要

IO デバイスは、DI、DO、AI、AO の 4 種類の信号をサポートしています。信号の供給源は、制御キャビネット内蔵、EtherCAT 拡張、フィールドバス拡張などがあります。産業用ロボットの制御キャビネットと協働ロボットのベース及び先端には、複数の DI と DO があります。産業用ロボット制御キャビネットには、EtherCAT 拡張インターフェースが設けられており、外部の拡張 EtherCAT モジュールを接続して、新しい DI、DO、AI、AO を使用することができます。Modbus バス拡張を使用して IO を設定することもできます。

10.6.2 パラメーター設定

「通信」→「IO デバイス」画面で、IO デバイスの表示、IO デバイスの新規作成、編集、削除をすることができます。

IOデバイス構成

デバイスタイプ: すべて リセット

IOボード番号	デバイスタイプ	名前	デジタル入力数	デジタル出力数	アナログ入力数	アナログ出力数	状態
0	ETHERCAT	ROKAE XMATE_IO_SLAVE	4	4	0	0	<<オン>>
1	ETHERCAT	ROKAE XMATE_EOAT_IO_SLAVE	2	2	2	0	<<オン>>
2	ROKAE_IO	ROKAE_IO_2	16	16	0	0	オフ

①

② 拡張属性
設備の種類: DIDO スレーブID: 1

③ + - △ ■

①	IO デバイス一覧
②	IO デバイスの拡張属性
③	IO デバイス操作ボタン。左から順に新規作成、編集、削除

上図の IO デバイス設定画面で、③の新規作成ボタンをタップし、IO デバイスの設定画面に入ります。ETHERCAT、FIELDBUS、ROKAE_IO の 3 タイプがあり、デバイスタイプによって画面が異なります。

IOデバイス構成

デバイスタイプ
デバイスタイプ: ETHERCAT

EtherCATスレーブ情報
設備の種類: Slave IO
スレーブアドレス: 2003

基本情報
IOボード番号: 3 名前: IO_Device_3
デジタル入力数: 16 アナログ入力数: 0
デジタル出力数: 16 アナログ出力数: 0

キャンセル 前へ 次へ

ETHERCAT - Slave IO のパラメーター画面

IOデバイス構成

デバイスタイプ
 デバイスタイプ :

EtherCATスレーブ情報
 設備の種類 :
 IOボードの種類 :

基本情報
 IOボード番号 : 名前 :
 デジタル入力数 : アナログ入力数 :
 デジタル出力数 : アナログ出力数 :

ETHERCAT - SafeBoard Extend IO - DIDO のパラメーター画面

IOデバイス構成

デバイスタイプ
 デバイスタイプ :

EtherCATスレーブ情報
 設備の種類 :
 IOボードの種類 :

基本情報
 IOボード番号 : 名前 :
 デジタル入力数 : アナログ入力数 :
 デジタル出力数 : アナログ出力数 :

アナログIO構成

	チャンネル1	チャンネル2	チャンネル3	チャンネル4
アナログ入力モード :	電圧	電圧	電圧	電圧
アナログ出力モード :	電圧	電圧	電圧	電圧

ETHERCAT - SafeBoard Extend IO - AIAO のパラメーター画面

IOデバイス構成

デバイスタイプ
 デバイスタイプ :

バスデバイス情報
 バスデバイス名 :

基本情報
 IOボード番号 : 名前 :
 デジタル入力数 : アナログ入力数 :
 デジタル出力数 : アナログ出力数 :

FIELDBUS のパラメーター画面

IOデバイス構成

デバイスタイプ

デバイスタイプ:

ROKAE IOボード情報

設備の種類:

スレーブID:

基本情報

IOボード番号: 名前:

デジタル入力数: アナログ入力数:

デジタル出力数: アナログ出力数:

ROKAE_IO のパラメーター画面

パラメーター	説明
デバイスタイプ	EtherCAT、FIELDDBUS、ROKAE_IO が選択可能です。 EtherCAT は、EtherCAT バスを使用し、拡張モジュールと組み合わせて IO を拡張します。この場合、拡張モジュールはスレーブとしてのみ使用され、スレーブアドレスを設定する必要があります。
EtherCAT スレーブ情報－デバイスタイプ	SafeBoard IO、SafeBoard Extend IO、xPanel IO、Slave IO が選択可能です。SafeBoard IO は、ロボットの安全ボードの DI と DO を指します。SafeBoard Extend IO は、ロボットの安全ボードにある IO で、一般に XBC_5 制御キャビネットの安全ボードの拡張 IO を指します。xPanel IO は協働ロボットにのみ適用され、ロボット先端の DI と DO を指します。Slave IO は EtherCAT 拡張モジュールを指します。
EtherCAT スレーブ情報－IO ボードの種類	EtherCAT スレーブ情報－デバイスタイプで SafeBoard Extend IO を選択すると、IO ボードの種類として DIO16_1、DIO16_4、DIO16_5、DIO16_6、AIAO4_2、AIAO4_3 から安全ボード拡張 IO ボードを選択できます。オプションの下一桁は安全ボード拡張 IO ボードのアドレスを指します。実際に接続された安全ボード拡張 IO に応じて選択してください。ユーザーが手動で編集する以外に、コントローラが自動的に安全ボード拡張 IO を識別して追加することもできます。
EtherCAT スレーブ情報－スレーブアドレス	EtherCAT バストポロジにおける拡張モジュールのスレーブアドレスです。安全ボード、関節、協働ロボットの先端のアドレスと競合してはいけません。
FIELDDBUS－バスデバイス名	「バスデバイス」画面で定義した名前です。この名前を使用して「バスデバイス」に関連付けます。
ROKAE_IO デバイスタイプ	DIDO と AIAO を選択できます。自社開発のデジタルおよびアナログ IO ボードに対応します。設定時に、実際の IO ボードのタイプと一致することを確認してください。

ROKAE_IO スレーブアドレス	接続される自社開発の IO ボードのアドレスは、1 から 15 まで選択でき、重複はできません。実際の IO ボードの DIP スイッチアドレスに対応していない場合は、状態が異常になる可能性があります。
IO ボード番号	各 IO デバイス、仮想 IO ボードを生成します。制御システム内部では、IO ボードを分類および管理するために IO ボードのシリアル番号を使用します。このシリアル番号は、仮想 IO ボード管理のための唯一の番号です。
名前	仮想 IO ボードのカスタム名です。この名前は、「状態監視」→「IO 信号」のフィルタリング条件に使用されます。
デジタル入力数	DI の数です
デジタル出力数	DO の数です
アナログ入力数	AI の数です
アナログ出力数	AO の数です
アナログ IO 設定	EtherCAT スレーブ情報-デバイスタイプで SafeBoard Extend IO を選択し、IO ボードの種類で AIAO4_2 または AIAO4_3 を選択すると、アナログ IO 設定オプションが表示されます。各アナログチャンネルを、電圧タイプまたは電流タイプに設定できます。

「状態監視」→「IO 信号」で、作成された DI、DO、AI、AO を監視できます。IO 信号は、「仮想 IO ボード名」でフィルタリングでき、現在の仮想 IO ボードで設定されている IO 信号のみ表示できます。信号タイプでのフィルタリングもでき、DI、DO、AI、AO のいずれかの信号のみの表示もできます。



仮想 IO ボードの設定が完了すると、IO 信号にはデフォルトの名前が生成されます。RL で使用するには 2 つの方法があります。1 つはデフォルト名を使用すること、もう 1 つは IO 信号一覧で名前を作成して、RL でその名前を使用することです。

デフォルト値を使用	IO ボード番号が 3 の場合、デフォルトで DI3_X の IO 信号が生成されます。下図のように RL プログラムで DI3_0 を使用して処理できます。 <pre> 1 GLOBAL PROC main() 2 print(DI3_0) 3 wait 2 4 ENDPROC 5 </pre>
-----------	--

名前を作成	「プログラミング」 → 「IO 信号一覧」で、IO 信号を作成します。RL プログラム内で、その名前を使用して処理できます。 IO信号を新規作成 基本情報 名前: signal0 説明: IOボード デバイス: IO_Device_3 タイプ: EtherCATデバイス 最大ポート数: DI 8 DO 8 AI 0 AO 0 信号バインディング 信号タイプ: DI ポート番号: 0 キャンセル 前へ 次へ <pre> 1 GLOBAL PROC main() 2 print(DI3_0) ← ① 3 print(signal0) ← ② 4 wait 2 5 ENDPROC 6 </pre> 上図の①と②の効果は同じです。
-------	--

10.6.3 Modbus 拡張 IO の例

周辺機器とやりとりするために IO 信号が必要な場合は、通信モジュールを制御キャビネットに接続することをお勧めします。推薦される Modbus IO モジュールについては、弊社にお問い合わせください。このモジュールは Modbus TCP スレーブで、コイル方式で制御されます。ロボットを Modbus Master として設定し、コイル機能を有効にする必要があります。

フィールドバスと拡張 IO の設定方法に従い、Modbus 拡張モジュールを使用して実際の IO を拡張する場合は、最初に「バスデバイス」を設定し、次に「IO デバイス」を設定する必要があります。

手順	図	説明
1. 「バスデバイス」を設定します。		モードで「マスター」を選択すると、ロボットがマスターとして動作します。 スレーブ ID は Modbus IO モジュールのスレーブ ID を指します。 ロボットがマスターとして動作する場合は、IP は Modbus IO モジュールの IP アドレス、ポート番号は Modbus IO モジュールのポート番号を記入します。 パラメーター設定が完了した

		後、「次へ」をタップしてパラメーター設定を終了します。
2. 「IO デバイス」を設定します。		デバイスタイプで「FIELD BUS」を選択します。 バスデバイス名は「バスデバイス」設定時の名前を選択します。 基本情報： デジタル入出力とアナログ入出力は、バスデバイスの設定画面で設定するため、ここでは設定する必要がなく、自動的に取得されます。
3. 機能の有効化と状態監視。		「バスデバイス」と「IO デバイス」が設定されたばかりのときは、バス接続をオンにしていなかったため、状態監視には設定された拡張 IO が表示されません。これらの IO を使用する場合は、バス接続をオンにし、制御キャビネットと拡張 IO モジュールが正しく接続を確立し、通信を維持できるようにする必要があります。 左の図のように、バスデバイスをオンにした後は、コントローラーは拡張モジュールと接続を確立し、正常に動作します。IO 信号の状態監視では、バスデバイスである modbus_0 の IO を確認でき、設定された DI と DO の数はバスデバイスで設定されたものと一致します。

10.7 先端ツール

xCore 制御システムは、先端ツールを介して DH グリッパの開閉制御をサポートします。先端ツールのインターフェースは IO 通信と RS485 通信に対応しています。この機能は協働ロボット xMate ER シリーズにのみ適用されます。

HMI で、「通信」→「先端ツール」で下図の設定画面に入ります。

先端ツール IO ポートの設定

先端ツール RS485 ポートの設定

設定パラメーターは次の通りです。

パラメーター	値/説明
メーカー	DH グリッパを選択します。
ポート	通信プロトコルは、IO または RS485 制御が選択可能です。
パス	グリッパは 2 組の属性 trip1 と trip2 で構成され、それぞれで開閉位置と開閉力が設定できます。
最大位置	グリッパが開く最大位置を、パーセンテージで設定します。RS485 のみサポートしています。
最小位置	グリッパが開く最小位置を、パーセンテージで設定します。RS485 のみサポートしています。
開放力	グリッパ開放時の力を、パーセンテージで設定します。RS485 のみサポートしています。
把持力	グリッパ把持時の力を、パーセンテージで設定します。RS485 のみサポートしています。

パラメータ設定後、「開く」と「閉じる」ボタンでグリッパの開閉動作を確認することができます。

注意：RS485 は、グリッパの trip パラメーター設定をサポートしています。IO 制御の場合は、trip パラメータは DH 通信アダプタボックスを介してのみ設定できます。

10.8 RCI 設定

RCI は外部制御インターフェースで、使用する前に RCI 通信設定を行う必要があります。

HMI「通信」→「RCI 設定」で、下図の RCI の設定画面に入ります。

RCI設定
☒ 有効化

RCIは外部制御インターフェースです。有効化スイッチをオンにすると、外部コントローラーでロボットを制御できます。

RCIでロボットを制御する場合は、安全のために非常停止スイッチを携帯し、必要に応じて押してください！

通信設定

手順1: 有効化スイッチをオンにする前に通信設定を行います。IPアドレスはユーザーPCのIPアドレス、ポート番号は1337に設定する必要があります。パケットロス閾値は、動作中に失われるデータ（パケット）の許容パーセンテージです。

手順2: 通信設定完了後、有効化をオンにし自動モードに切り替え、ロボットの電源を入れるとRCIを介してロボットを制御できるようになります。

手順3: 有効化スイッチがオンの場合、JOGやドラッグなどのHMIを介したロボットの操作は禁止されています。操作する必要がある場合は、有効化スイッチをオフしてから操作してください。

IPアドレス : 127.0.0.1

ポート番号 : 1337

パケットロス閾値 : 0

設定するパラメーターは下表の通りです。

パラメーター	説明
IP アドレス	PC とロボットがネットワークケーブルで直接接続されている場合は、PC の IP アドレスはロボットの IP アドレスと同じネットワークセグメントに属している必要があります。PC とロボットが無線またはスイッチで接続されている場合は、PC はロボットと同じ LAN に属している必要があります。
ポート番号	ポート番号はデフォルトで 1337 に設定されています。
パケットロス閾値	パケットロス閾値は、パーセンテージで表示され RCI 通信中のパケットロス率を指します。例えば、パケットロス閾値が 10 に設定されている場合、RCI 使用中のパケットロス率が 10%を超えてはならないことを指します。パケットロス閾値は 10～20 の範囲で設定することをお勧めします。

RCI の詳細な使用方法については、「RCI ユーザーズマニュアル」を参照してください。

10.9 xPanel 設定

xPanel 設定は、ロボットの先端ツールのモードを設定することができます。この機能は、協働ロボット xMate CR シリーズでのみ利用可能です。

HMI メインインターフェースの「通信」→「xPanel 設定」で、下図の xPanel の設定画面に入ります。

外部給電モード : ☒ 出力しない ☐ 保持 ☐ 12V出力 ☐ 24V出力

アナログ入力またはRS485の選択 : ☒ アナログ入力 ☐ RS485

デジタル出力DO0モード : ☒ NPN ☐ PNP

デジタル出力DO1モード : ☒ NPN ☐ PNP

アナログ入力AI0モード : ☒ 電圧 ☐ 電流

アナログ入力AI1モード : ☒ 電圧 ☐ 電流

確定

設定するパラメーターは下表の通りです。

パラメーター	説明
外部給電モード	先端ツールの外部給電モードを設定します。
アナログ入力または RS485 の選択	先端ツールがアナログ入力または RS485 シリアルポートを使用するように選択します。
デジタル出力 DO0 モード	対応する先端ツールの DO 出力を NPN または PNP モードに設定します。
デジタル出力 DO1 モード	対応する先端ツールの DO 出力を NPN または PNP モードに設定します。
アナログ入力 AI0 モード	対応するアナログ入力を電圧タイプまたは電流タイプに設定します。
アナログ入力 AI1 モード	対応するアナログ入力を電圧タイプまたは電流タイプに設定します。

必要なパラメーターを設定した後、「確定」ボタンをタップすると設定が有効になります。

注意：予期しないエラーを防ぐために、外部接続されたアナログ入力信号の電圧または電流タイプは、対応するアナログ入力モードと一致していることを確認してください。

10.10 電動グリッパ/吸盤

10.10.1 概要

xMate CR、SR、ER ロボットは先端ツールの RS485 通信をサポートしており、さまざまな電動グリッパと吸盤に対応しています。このインターフェースは、主に対応した電動グリッパと吸盤の設定とテストに使用されます。

- この機能は xMate ER シリーズ、xMate CR シリーズ、xMate SR シリーズで使用できます。
- 古いバージョンのロボットの先端基板はこの機能をサポートしない可能性があります。ボードのファームウェアをアップグレードするには、弊社にお問い合わせください。
- xMate CR 機種を使用する前に、xPanel のパラメーターが正しく設定されていることを確認してください。

10.10.2 設定

HMI の「通信」->「電動グリッパ/吸盤」で、下図の設定画面に入ります。

基本情報

メーカー：Jodell

シリーズ：EPG

通信：RS485

ツール設定

ツールを初期化後、現在のツールIDドロップダウンリストから初期化しツールを選択して、ツールの状態を監視し、テストを実行できます。

現在のツールID：

初期化するツールID：0

初期化

ツールテスト

位置：0

速度：0

力：0

Move to

ツールステータス

位置：0

速度：0

力：0

接触検知状態：動作中

「基本情報」の設定パラメーターの説明は次の通りです。

パラメーター	説明
メーカー	Jodell、RM、Ribotiq、DHをサポートしています。
シリーズ	Jodell：EPG シリーズの電動グリッパーと EVS シリーズの吸盤をサポートしています。 RM：RM-RMG と RM-C の2つのシリーズの電動グリッパーをサポートしています。 Robotiq：2F85 シリーズをサポートしています。 DH：PGI シリーズをサポートしています。
通信	RS485 のみサポートしています。

10.10.2.1 Jodell 電動グリッパー

メーカーに **Jodell**、シリーズに **EPG** を選択すると、HMI は下図に示すように EPG シリーズ電動グリッパーのテストインターフェースに切り替わります。

初期化：特定の ID の電動グリッパーをテストする場合は、まずグリッパー装置の ID を入力し、「初期化」ボタンをタップし、グリッパーの自動検出を実行します。初期化に成功した場合は、グリッパーのハードウェア接続と通信が正常であることを意味します。

ツールテスト：電動グリッパーの初期化完了後、「Move to」ボタンをタップして、グリッパーが指定された速度と力で指定された位置に動作するよう制御します。グリッパーが指定された位置に達するか、または物体に触れて指定された力に達すると、移動が停止します。グリッパーの接触検出状態もテスト画面に表示されます。

設定パラメーターは次の通りです。

パラメーター	値/説明
ツール ID	電動グリッパーデバイスの ID を入力します。この ID は Jodell のデバッグソフトウェアによって設定された電動グリッパーの ID です。
ツール位置	0～255 の範囲でグリッパーの位置を設定します。
ツール速度	0～255 の範囲でグリッパーの動作速度を設定します。
ツールトルク	0～255 の範囲でグリッパーの動作力を設定します。

10.10.2.2 Jodell 吸盤

メーカーに **Jodell**、シリーズに **EVS** を選択すると、HMI は下図に示すように **EVS** シリーズの吸盤のテストインターフェースに切り替わります。

初期化：特定の ID の吸盤をテストする場合は、まず吸盤装置の ID を入力し、「初期化」ボタンをタップします。初期化に成功した場合は、吸盤のハードウェア接続と通信が正常であることを意味します。

ツールテスト：吸盤の初期化完了後、必要に応じて吸盤のパラメーターを設定する必要があります。パラメーターを入力した後、「開始」ボタンをタップして吸盤の動作状態をテストすることができます。

設定パラメーターは次の通りです。

パラメーター	値/説明
チャンネル選択	吸盤は 2 つのチャンネルをサポートしており、任意のチャンネルを有効にすることで選択できます。
最小真空度	吸盤の目標真空度を設定し、吸盤内の真空度がこの値に達すると動作を停止します。
最大真空度	吸盤の目標真空度を設定し、吸盤内の真空度がこの値より大きい場合は、動作を開始します。
タイムアウト時間	指定された時間内に指定された最小真空度に達しない場合は、タイムアウトになります。

10.10.2.3 RM 電動グリッパー

「基本情報」でメーカーに **RM** を選択し、シリーズは **RM-RGM** または **RM-C** を選択すると、HMI は下図に示すように **RM** シリーズ電動グリッパーのテストインターフェースに切り替わります。

初期化：特定の ID の電動グリッパーをテストする場合は、まずグリッパー装置の ID を入力し、「初期化」ボタンをタップします。グリッパーが正しく設定されて通信も正常である場合は、初期化に成功します。

現在のツール ID：初期化された現在のタイプのデバイス一覧を表示します。デバイスを選択すると、選択されたデバイスの状態確認とテストを実行することができます。

ツール状態：初期化に成功した後、RM グリッパーは位置（単位：mm）、速度（単位：mm/s）、トルク（%）、到達判定（1 は到達済み）、エラーコードなどのツール状態を表示できます。

ツールテスト：位置モードとトルクモードで RM グリッパーの動作をテストできます。グリッパーメーカーのマニュアルの要件に従い、グリッパーの開放は位置モード、把持はトルクモードを使用する必要があります。

位置モード：設定できるパラメーターには、位置（絶対位置決め位置、単位：mm）、速度（mm/s）、位置範囲（到達とみなされる誤差範囲：mm）、加速度（単位：mm/s²）があります。上図を参照してください。

トルクモード：設定できるパラメーターには、距離（相対距離、単位：mm）、速度（mm/s）、トルク（%）、加速度（単位：mm/s²）、位置範囲（到達とみなされる誤差範囲：mm）、時間範囲があります。下図を参照して下さい。

ツールテスト

☐ 位置モード ☒ トルクモード

速度： 0.010 mm 加速度： 0.010 mm/s²

距離： 0.000 mm 距離範囲： 0.000 mm

トルク： 0.010 % 時間範囲： 0.010 ms

Move to

「Move to」ボタンをタップすると、電動グリッパーは設定されたモードとパラメーターに従って動作します。

10.10.2.4 Robotiq 2F_85 電動グリッパー

「基本情報」でメーカーに Robotiq を選択し、シリーズに 2F_85 を選択すると、HMI は Robotiq 2F_85 電動テスト画面に切り替わります。

基本情報

メーカー： Robotiq シリーズ： 2F_85 通信： RS485

ツール設定

ツールを初期化後、現在のツールIDドロップダウンリストから初期化しツールを選択して、ツールの状態を監視し、テストを実行できます。

現在のツールID： 初期化するツールID： 0 初期化

ツールテスト

位置： 3
速度： 0
力： 0

Move to

ツールステータス

位置： 0
接触検知状態： 物体検知無し
エラーコード：

初期化：特定の ID の電動グリッパーをテストする場合は、まず電動グリッパー装置の ID を入力し、「初期化」ボタンをタップします。初期化に成功した場合は、電動グリッパーのハードウェア接続と通信が正常であることを意味します。

ツールテスト：電動グリッパーの初期化完了後、「Move to」ボタンをタップして、グリッパーが指定された速度と力で指定された位置に動作するよう制御します。グリッパーが指定された位置に達するか、または物体に触れて指定された力に達すると、移動が停止します。グリッパーの接触検出状態もテスト画面に表示されます。

設定パラメーターは次の通りです。

パラメーター	値/説明
ツール ID	電動グリッパーデバイスの ID を入力します。
ツール位置	3～255 の範囲でグリッパーの位置を設定します。
ツール速度	0～255 の範囲でグリッパーの動作速度を設定します。
ツールトルク	0～255 の範囲でグリッパーの動作力を設定します。

10.10.2.5 DH 電動グリッパー

「基本情報」でメーカーに DH を選択し、シリーズに PGI を選択すると、HMI は PGI シリーズ電動グリッパーテスト画面に切り替わります。

初期化：特定の ID の電動グリッパーをテストする場合は、まず電動グリッパー装置の ID を入力し、「初期化」ボタンをタップします。初期化に成功した場合は、電動グリッパーのハードウェア接続と通信が正常であることを意味します。

注意：初期化の過程で行われる開閉動作は、電動グリッパーの動作範囲を確認するためのものです。その後の動作は初期化時の動作範囲を基準に行われます。

ツールテスト：電動グリッパーの初期化完了後、「Move to」ボタンをタップして、グリッパーが指定された速度と力で指定された位置に動作するよう制御します。グリッパーが指定された位置に達するか、または物体に触れて指定された力に達すると、移動が停止します。グリッパーの接触検出状態もテスト画面に表示されます。

注意：物体を挟んだ後、物体を取り除いても電動グリッパーは動作を続けません。新しい目標位置を設定しない限り、電動グリッパーは動作を再開しません。

電動グリッパーの状態は 5 つの種類があり、それぞれ次の値に対応しています

- 0：移動中
- 1：物体を挟んでいない
- 2：物体を挟んだ
- 3：物体が落下した
- 4：表示なし、ID が初期化されていない

設定パラメーターは次の通りです。

パラメーター	値/説明
ツール ID	電動グリッパーデバイスの ID を入力します。この ID は、DH デバッグソフトウェアで設定された電動グリッパーの ID です。
ツール位置	グリッパーの位置を設定します。範囲は 0-1000 で、初期化時に決定されたストローク範囲の千分率で設定されます。
ツール速度	グリッパーの動作速度を設定します。範囲は 0-100 で、速度の百分率で設定されます。
ツールのトルク	グリッパーの動作力を設定します。範囲は 20-100 で、出力の百分率で設定されます。

10.11 シリアルポート設定

ユーザーは、シリアルポートを使用して周辺機器と通信することができます。シリアルポートの使用はハードウェアが機能をサポートしている必要があります。産業用ロボットの XBC5 制御キャビネットには RS-232 シリアルポートが 1 チャンネル設けられています。制御キャビネットに設けられた USB ポートを使用して、USB-RS232C 変換モジュールでシリアル通信を行うこともできます。協働ロボットは関連するハードウェアインターフェースを持っていないため、この機能をサポートしていません。

HMI で「通信」→「シリアルポート設定」で下図のシリアルポートの設定画面に入ります。

①	シリアルポートのフィルタリング表示
②	シリアルポートデバイス一覧
③	シリアルポート編集ボタン。左から順に新規作成、編集、削除。

新規作成ボタンをタップすると、下図に示すようにシリアルポートの新規作成画面に入ります。

設定パラメーターは次の通りです。

パラメーター	説明
名前	ユーザー定義の名前です。RL でこのシリアルポートを操作するために、一意の名前である必要があります。 注意：この名前は、RL プロジェクト内の名前の競合制限の対象となります。プロジェクト内の既存のネットワーク識別子と同じ名前にしてはいけません。また、プロジェクト内の既存の他のシリアルポート識別子と同じ名前にしてもいけません。
ポート	システムポートの番号を設定します。制御システムは、スキャンされたシリアルポートリソース（USB 形式で変換されたシリアルポートを含む）を一覧表示します。
ボーレート	1200/2400/4800/9600/19200/38400/57600/115200 から選択します。
データビット	5/6/7/8 から選択します。
ストップビット	1/1.5/2 から選択します。
チェックビット	奇数チェック（Odd）/偶数チェック（Even）/全 1 チェック（Mark）/全 0 チェック（Space）/チェックなし（None）から選択します。

パラメーターを設定した後、「完了」ボタンをタップすることでシリアルポートの設定を完了し、RL プログラムでシリアルポートを使用できるようになります。シリアルポート機能には、一連の命令が含まれています。詳しくはRL 命令のシリアルポート命令の詳細な説明を参照してください。

注意：データ送受信に異常が発生するのを避けるために、シリアルポート通信のお互いのパラメータ設定が同じであることを確認してください。

10.12 エンコーダー

この機能はコンベアトラッキング機能の一部です。詳細な使用方法については、「コンベアトラッキング機能」のマニュアルを参照してください。

10.13 OPC-UA

10.13.1 概要

xCore 制御システムの OPC-UA は、OPC-UA 通信のサーバーとして、OPC 40010-1 OPC UA for Robotics, Part 1: Vertical Integration 規格に含まれているすべての必須ノードと一部のオプションノードをデフォルトでサポートしています。詳細なモデルの説明については、付録の OPC-UA の関連内容を参照してください。さらに、ユーザー定義変数と event イベントのアップロードもサポートしています。

10.13.2 オン/オフ

HMI の「通信」→「OPC-UA」で、OPC-UA の設定画面に入ります。下図に示すように、上部の「ポート」編集ボックスで OPC-UA サービスのポートを入力し、「有効化」ボタンで OPC-UA サービスのオン/オフを切り替えることができます。



OPC-UAサーバー
ポート番号 : 4840 ☒ 有効化

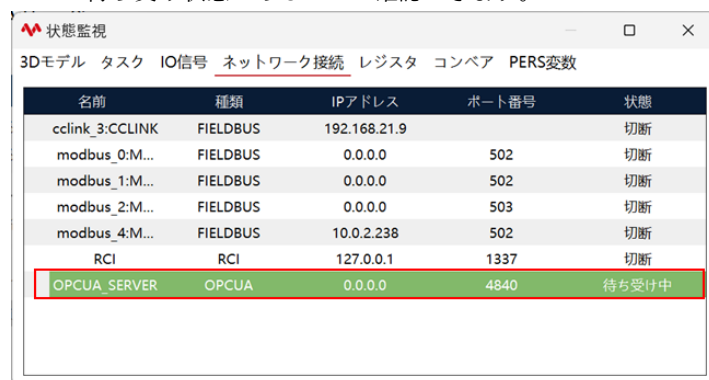
セキュリティ 証明書 変数

サポートされるセキュリティモード: None, Sign, Sign&Encrypt
サポートされるセキュリティポリシー: Basic128Rsa15, Basic256, Basic256Sha256, Aes128Sha256RsaOaep

ユーザー管理
ユーザー名 :
パスワード :
匿名を許可する : ☒

	ユーザー名	パスワード
1	Tanaka	123456
2	Yamashita	123456
3	Koga	123456
4	Araki	Araki123

OPC-UA サービスが正常にオンになると、下図のように状態監視のネットワーク接続で OPC-UA サービスが待ち受け状態にあることが確認できます。



状態監視

3Dモデル タスク IO信号 ネットワーク接続 レジスタ コンペア PERS変数

名前	種類	IPアドレス	ポート番号	状態
cclink_3:CCLINK	FIELD BUS	192.168.21.9		切断
modbus_0:M...	FIELD BUS	0.0.0.0	502	切断
modbus_1:M...	FIELD BUS	0.0.0.0	502	切断
modbus_2:M...	FIELD BUS	0.0.0.0	503	切断
modbus_4:M...	FIELD BUS	10.0.2.238	502	切断
RCI	RCI	127.0.0.1	1337	切断
OPCUA_SERVER	OPCUA	0.0.0.0	4840	待ち受け中

10.13.3 セキュリティ

OPC-UA 設定画面の「セキュリティ」タブをタップして、下図のセキュリティ設定画面に入ります。



OPC-UAサーバー
ポート番号 : 4840 ☒ 有効化

セキュリティ 証明書 変数

サポートされるセキュリティモード: None, Sign, Sign&Encrypt
サポートされるセキュリティポリシー: Basic128Rsa15, Basic256, Basic256Sha256, Aes128Sha256RsaOaep

ユーザー管理
ユーザー名 :
パスワード :
匿名を許可する : ☒

	ユーザー名	パスワード
1	Tanaka	123456
2	Yamashita	123456
3	Koga	123456
4	Araki	Araki123

このインターフェースは、主に OPC-UA サーバー側のセキュリティポリシーの説明と設定に使用します。

1. デフォルトで、None、Sign、Sign&Encrypt の 3 つのセキュリティモードをサポートします。クライアントの接続時に必要なモードを選択します。
2. デフォルトで Basic128Rsa15、Basic256、Basic256Sha256、Aes128Sha256RsaOaep の 3 つのセキュリティポリシーをサポートします。クライアントの接続時に必要なポリシーを選択します。

3. 「匿名を許可する」チェックボックスにチェックを入れると、クライアントは匿名でログインできます。チェックを入れない場合は、クライアントは右側のリストにあるユーザーのみログインできます。
4. ユーザー名とパスワードを入力し、「ユーザー追加」ボタンをタップすると、右側のリストに新しいユーザーを追加できます。
5. 右側のリストでユーザーを選択し、「ユーザー削除」ボタンをタップすると、不要なユーザーを削除できます。

10.13.4 証明書

OPC-UA 設定画面の「証明書」タブをタップして、下図の証明書設定画面に入ります。

1. OPC-UA サーバーの証明書と秘密鍵をインポートしない場合は、制御システムはデフォルトで自己生成した証明書と秘密鍵を使用します。
2. 「証明書をインポート」ボタンをタップするとサーバーの証明書をインポートし、「秘密鍵をインポート」ボタンをタップするとサーバーの秘密鍵をインポートします。インポートする証明書と秘密鍵は der 形式である必要があり、証明書と秘密鍵が一致していることを確認する必要があります。また、インポートされたサーバー証明書の URL は、`urn:xcore.opcua.server` である必要があります。
3. 右の「インポート」ボタンをタップすると、信頼できるクライアント証明書をインポートできます。「削除」ボタンをタップすると、選択されたクライアント証明書を削除できます。

10.13.5 カスタム変数設定

OPC-UA 設定画面で「変数」タブをタップして、下図のカスタム変数の設定画面に入ります。

	ユーザー名	タイプ	書き込み可能	初期値	現在の値	説明
1	bool_2	bool	はい	true		
2	bool_3	bool	はい	true		
3	int_4	int	はい	10		
4	int_3	int	はい	10		
5	int_2	int	はい	10		
6	int_1	int	はい	10		
7	double_3	double	はい	1.1		

xCore 制御システムの OPC-UA 通信はカスタム変数機能をサポートしており、bool、int、double、string の 4 タイプをサポートします。また、クライアントの書き込み可否の設定もサポートします。具体的な設定機能は次の通りです。

1. 画面の左側で「名前」、「説明」、「タイプ」、「書き込み可能」、「初期値」などの属性を記入し、「新規作成」ボタンをタップすると、右側のリストにカスタム変数を追加できます。最大 128 個のカスタム変数をサポートしています。
2. 「削除」ボタンをタップすると、右側のリストで選択された変数を削除できます。「すべて削除」をタップすると右側のリストのすべての変数を削除できます。
3. 右側の変数リストにある「モニターをオンにする」チェックボックスにチェックを入れると、変数監視がオンになり、「現在値」にはコントローラ内の変数の値がリアルタイムに表示されます。「値を変更」ボタンをタップすると選択された変数の現在の値を変更できます。

クライアントは Robotics モデルの CustomVariables ノードにあるカスタム変数のノードで、読み書きを行うことができます。ロボット側では、ユーザーは RL プログラム内で ReadOpcUaVarByName と WriteOpcUaVarByName の 2 つの命令を使用して、OPC-UA カスタム変数を読み取りまたは変更をすることができます。詳細な使用方法については、RL 命令の章の該当する使用説明を参照してください。

10.13.6 イベント (event)

OPC-UA サーバーは、イベント (event) によってロボットの状態変化を OPC-UA クライアントに通知することができます。現在サポートされている通知イベントは次の通りです。

イベント	重大度 (Severity)
モーター電源オン/オフ	100
手動/自動モードの切り替え	100
プログラムの実行/停止	100
非常停止のトリガー	600
安全ゲートのトリガー (CR シリーズ及び SR シリーズでは停止エリア入力のトリガー)	600
保護停止トリガ	600
衝突検知アラーム	600

クライアントイベントは下図のように表示されます。

Events Alarms Event History							
A	C	Time	Severity	Server/Object	SourceName	Message	EventType
		20:04:54.705	100	new1@192.168.0.160 / Server	Server	电机上电	RobotEventType
		20:04:58.906	100	new1@192.168.0.160 / Server	Server	电机下电	RobotEventType
		20:05:02.506	100	new1@192.168.0.160 / Server	Server	切换操作模式为自动	RobotEventType
		20:05:05.107	100	new1@192.168.0.160 / Server	Server	切换操作模式为手动	RobotEventType

11 安全機能

11.1 本章の概要

本章では、主に **xCore** の安全機能に関する機能の設定について説明します。

11.2 安全機能パスワード

安全機能の設定を行うためには、パスワードを入力する必要があります。デフォルトのパスワードは「**safety**」です。

1. ユーザーは安全機能にアクセスするたびにパスワードを入力し、「解除」をタップすることで初めて安全機能画面を操作できます。
2. 解除後、他の安全機能の画面に進む際は再度パスワードを入力する必要はありません。
3. 安全機能以外の機能に切り替えてから再度安全機能に戻る場合は、再度解除が必要です。
4. 安全機能の設定を変更した後は、変更後に「確認」ボタンをタップすることで安全パラメーターを適用することができます。
5. 安全機能はユーザー権限の影響を受けません。オペレーターなどの低権限ユーザーも、安全機能へのログイン後はパラメーターを変更できます。

安全パスワードは「設定 - ユーザーグループ - 安全機能パスワード」で変更できます。

11.3 軸制限

11.3.1 機能の概要

軸制限はロボットの各軸のパラメーターを監視します。任意の軸の状態が設定された閾値を超えた場合、ロボットはすぐに動作を停止し、安全コントローラーは安全停止状態に入ります。

軸制限は主に軸位置、軸速度、軸トルク、および軸出力の 4 つの制限機能を備えています。各制限機能は 2 セットのパラメーターを設定でき、通常モードと協働モードで異なる閾値で判断されます。

ユーザーは必要に応じて、特定の機能をオンまたはオフにすることができます。

11.3.2 軸位置制限

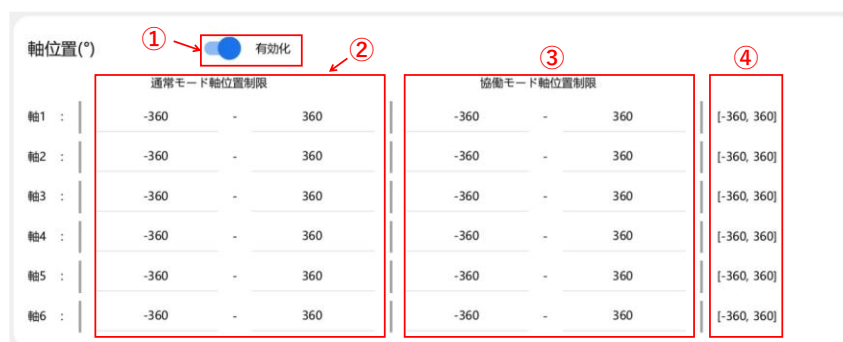
11.3.2.1 軸位置制限機能の概要

軸位置制限は、ソフトウェアのレベルでロボットの各軸の最大動作範囲を設定する機能で、ロボットが周辺機器と干渉または衝突するのを防ぎます。

ドラッグ操作中も、各軸の角度は軸位置制限の影響を受けます。制限付近までドラッグすると、ロボットアームには制限方向から遠ざける反発力が加わります。このドラッグ反発力が発生する範囲は、HMI 画面で設定された上下限の 10° 以内です。

例えば、軸 1 の関節位置制限が $-170^{\circ} \sim 170^{\circ}$ に設定されている場合、

反発力が発生する範囲は「 $[-170^{\circ} \sim -160^{\circ}]$ および $[160^{\circ} \sim 170^{\circ}]$ 」になります。



①	「有効化」スイッチは、軸位置制限の有効・無効を制御します。
②	通常モードでの軸位置制限。下限は上限より小さく設定する必要があります。
③	協働モードでの軸位置制限。範囲は通常モードの軸位置制限範囲以下に設定する必要があります。
④	設定できる最大範囲。

11.3.2.2 軸位置制限を超えた場合の処理

一部の非常に稀な状況において、ロボットが軸位置制限を超えて動作することがあります。

例えば、ロボットがリミットの境界に到達した際に非常停止が作動した場合や、ロボットが STOP 0 を実行した際に軸位置制限を超えてしまう可能性があります。

xCore V2.1 以前のバージョンでは、ロボットの軸が 1 つでも軸位置制限を超えている場合、ジョグ操作やプログラムの実行ができません。この場合、まず軸位置制限を解除し、超えてしまった軸をジョグ操作で軸位置制限範囲内に戻した後、再度軸位置制限を有効にする必要があります。

xCore V2.2 以降のバージョンで、安全コントローラーのないロボットが軸位置制限を超えてしまった場合は、軸位置制限の内側に向かってジョグ操作を行い、軸を制限範囲内に戻すことが可能です。

安全コントローラーがある場合は、ロボットが軸位置制限を超えると、「安全停止状態」に入ります。この場合、まず軸位置制限を解除し、非常停止リセットを実行した後、超えてしまった軸をジョグ操作で軸位置制限範囲内に戻し、再度軸位置制限を有効にする必要があります。



警告

軸位置制限の解除機能はロボットの軸が位置制限を超えた場合のみ、制限を超えた軸を正常範囲内に戻すために使用できます。軸位置制限が解除されている間は、プログラムの実行はできません。

11.3.3 軸速度制限

有効化スイッチを使用して、軸速度制限をオン/オフできます。オンにすると、ロボットの各軸の角速度がリアルタイムで監視されます。通常モードと協働モードで異なる監視パラメーターを使用して判断します。いずれかの軸角速度が閾値を超えると、ロボットは即座に停止し、モーターの電源を切ります。安全コントローラーは安全停止状態に入ります。

軸速度 (°/s)		☒ 有効化			
通常モード軸速度制限					
軸1 : 0	軸2 : 0	軸3 : 0	軸4 : 0	軸5 : 0	軸6 : 0
協働モード軸速度制限					
軸1 : 0	軸2 : 0	軸3 : 0	軸4 : 0	軸5 : 0	軸6 : 0

11.3.4 軸トルク制限

有効化スイッチを使用して、軸トルク制限をオン/オフできます。オンにすると、ロボットの各軸のトルクがリアルタイムで監視されます。通常モードと協働モードで、異なる監視パラメーターを使用して判断します。いずれかの軸トルクが閾値を超えると、ロボットは即座に停止し、モーターの電源を切ります。安全コントローラーは安全停止状態に入ります。

軸トルク (Nm)		☒ 有効化			
通常モード軸トルク制限					
軸1 : 351	軸2 : 351	軸3 : 182	軸4 : 134	軸5 : 134	軸6 : 134
協働モード軸トルク制限					
軸1 : 351	軸2 : 351	軸3 : 182	軸4 : 134	軸5 : 134	軸6 : 134

11.3.5 軸出力制限

有効化スイッチを使用して軸出力制限をオン/オフできます。オンにすると、ロボットの各軸の出力がリアルタイムで監視されます。通常モードと協働モードで、異なる監視パラメーターを使用して判断します。いずれかの軸の出力が閾値を超えると、ロボットは即座に停止し、モーターの電源を切ります。安全コントローラーは安全停止状態に入ります。

軸出力 (W)		☒ 有効化			
通常モード軸出力制限					
軸1 : 59	軸2 : 59	軸3 : 59	軸4 : 59	軸5 : 26.08	軸6 : 26.08
協働モード軸出力制限					
軸1 : 29.5	軸2 : 29.5	軸3 : 29.5	軸4 : 29.5	軸5 : 13.04	軸6 : 13.04

11.4 ロボット制限

①	速度制限	
	☐ TCP直線速度: 0 mm/s ☐ TCP角速度: 100 °/s ☐ エルボー直線速度: 1000 mm/s ☐ エルボー角速度: 100 °/s	協働モードTCP直線速度: 0 mm/s 協働モードTCP角速度: 50 °/s 協働モードエルボー直線速度: 500 mm/s 協働モードエルボー角速度: 50 °/s
	協働モード速度制限 ☐ 有効化	
	TCP直線速度: 500 mm/s 軸速度 軸1: 45 °/s 軸2: 37.5 °/s 軸3: 45 °/s 軸4: 45 °/s 軸5: 56.25 °/s 軸6: 56.25 °/s	
③	全体出力 ☐ 有効化	
	出力: 0 W 協働モード出力: 0 W	
④	運動量制限 ☐ 有効化	
	運動量: 0 kg*m/s 協働モード運動量: 0 kg*m/s	
⑤	ドラッグ制限 ☐ 有効化	
	ドラッグ直線速度250mm/s以下	

①	速度制限には、TCP 直線速度、TCP 角速度、エルボー直線速度、エルボー角速度の 4 つの速度制限があります。それぞれに独立した有効化スイッチがあり、対応するスイッチを使用してロボットの速度制限をオン/オフできます。オンにすると、ロボットの速度がリアルタイムで監視されます。通常モードと協働モードで、異なる監視パラメーターを使用して判断します。例えば、通常モードでは「TCP 直線速度」が閾値として使用され、協働モードでは「協働モード TCP 直線速度」が閾値として使用されます。監視値が閾値を超えると、ロボットは即座に停止し、モーターの電源を切ります。安全コントローラーは安全停止状態に入ります。
②	協働モード速度制限を有効にした場合、ロボットが協働モードに入ると、ロボットは設定された TCP 速度と軸速度以下で動作します。
③	有効化スイッチを使ってロボットの全体出力制限を有効または無効にできます。有効にすると、ロボットの全体出力をリアルタイムで監視します。通常モードと協働モードで、異なる監視パラメーターを使用して判断します。全体出力が閾値を超えると、ロボットは直ちに停止し、モーターの電源を切ります。安全コントローラーは安全停止状態に入ります。
④	有効化スイッチを使ってロボットの運動量制限を有効または無効にできます。有効にすると、ロボットの運動量をリアルタイムで監視します。通常モードと協働モードで、異なる監視パラメーターを使用して判断します。運動量が閾値を超えると、ロボットは直ちに停止し、モーターの電源を切ります。安全コントローラーは安全停止状態に入ります。
⑤	安全コントローラー専用の機能です。協働ロボットは、有効化スイッチを使って安全コントローラーによるロボットのドラッグ速度制限を有効または無効にできます。有効にすると、協働ロボットのドラッグ速度が 250mm/s を超えた場合はドラッグが禁止され、モーターの電源が切れ、安全停止状態に入ります。

11.5 バーチャルフェンス

11.5.1 機能紹介

バーチャルフェンス機能は、xMate 協働ロボットの直交動作（直交移動のみ）のドラッグ時に、仮想の壁を設定しフランジ先端の動作範囲を制限する機能です。バーチャルフェンスの近くにドラッグすると、ユーザーはバーチャルフェンスからの反力を感じます。

典型的な適用シナリオは、医師が xMate 協働ロボットを補助ツールとして、ドラッグ機能を使用して外科手術を行う場合、安全性を向上させ事故を避けるために、バーチャルフェンスを設定することでドラッグ時のロボットフランジの動作範囲を制限することができます。



①	「有効化」スイッチでバーチャルフェンスの有効/無効を切り替えます。
②	バーチャルフェンス機能の使用手順を紹介しています。
③	バーチャルフェンスのタイプを設定します。球体と直方体があります。 パラメーターとして、中心点と境界の大きさを設定します。 注意：「境界」で設定するパラメーター値は 100mm 以上にする必要があります。

注意：過大なドラッグ力や速度が速すぎるという極端な場合、ロボットはバーチャルフェンスからはみ出す可能性があり、システムはそれに応じて警告を表示します。

11.6 衝突検知

11.6.1 機能紹介

衝突検知機能は、ロボットの動力学モデルの推定に基づく受動的検知機能です。ロボットの動作中に外界との偶発的な衝突が発生した場合、衝突検知はタイミリーに衝突を検知し、事前に設定された処理措置を実行することができます。

感度設定は、全体設定と単軸設定の 2 種類がサポートされています。少なくとも 1 つのモードにチェックを入れる必要があります。

感度設定

☒ 全体設定 ☐ 単軸設定

 注: 感度のパーセンテージが高いほど、感度が高くなります。

協働モード感度 : 101%

停止方法は、ソフトストップのみをサポートしています。

停止方法

停止方法： ソフトストップ

柔軟性： 0

ソフトストップ：衝突が検知されると、ロボットは直ちに停止します。この方法は高剛性環境における激しい衝突の検知に適用します。ロボットはモーター電源オフ後に既定のルートから離れることがあります。柔軟性を高くするほどロボットの応答が速くなりますが、ロボットの関節への負荷が大きくなります。デフォルト値は0です。

11.6.1.3 伝達トルク制限

伝達トルク制限パラメーターは、減速機の最大伝達トルクを制限し、重要な駆動部品や機械的なゼロ点を保護するために使用されます。

伝達トルク制限パラメーターは、衝突保護機能で作用します。コントローラーが伝達トルクがこの制限を超えていることを検知した場合、ロボットは伝達トルク超過エラーをトリガーします。初回起動時、ロボットはデフォルトの伝達トルク制限パラメーターを使用します。

伝達トルク制限

伝達トルク制限は、減速機の最大伝達トルクを制限することで、機械のゼロ点が失われないように保護します

軸1 : 174	Nm 範囲:[0~194]	軸2 : 174	Nm 範囲:[0~194]
軸3 : 91	Nm 範囲:[0~102]	軸4 : 59	Nm 範囲:[0~66]
軸5 : 59	Nm 範囲:[0~66]	軸6 : 59	Nm 範囲:[0~66]

現在の伝達トルク制限パラメーターはデフォルト値です

11.6.1.4 遅延補償パラメーター

遅延補償パラメーターは、衝突検知アルゴリズムの内部パラメーターです。自動推定を行うことで、力の監視精度を向上させ、誤検出の確率を低減し、最適な衝突検知性能を実現します。

遅延補償パラメーター

遅延補償パラメーター:

軸1 : 50	軸2 : 50	軸3 : 50	軸4 : 50
軸5 : 50	軸6 : 50		

現在の遅延補償パラメーターはデフォルト値です

推定を開始した後は、正確な推定値を得るために各軸を動かす必要があります

推定開始

11.6.1.5 最大出力監視

最大出力監視機能を使用して、各軸モーターの最大出力トルクを、スイッチがオンの間で監視します。

ユーザーは、各軸の最大出力監視の値に基づいて、伝達トルク制限パラメーターを調整できます。

パラメーター設定

力の監視

最大出力監視

有効化

最大出力監視機能を使用すると、監視スイッチがオンからオフになるまでの間、各軸モーターの最大出力を監視することができます

軸1 最大トルク : 0.00	Nm	軸2 最大トルク : 0.00	Nm
軸3 最大トルク : 0.00	Nm	軸4 最大トルク : 0.00	Nm
軸5 最大トルク : 0.00	Nm	軸6 最大トルク : 0.00	Nm
軸7 最大トルク : 0.00	Nm		

11.6.2 注意事項

1. プログラム実行中にロボットが高速で動作し周辺機器と激しく衝突した場合、衝突力が大きすぎると直接サーボドライバーがアラームを報告して停止します。衝突をクリアしてサーボアラームをリセットしてからロボットを再起動してください。
2. 感度・モードの選択を誤ると、ロボット衝突の誤検出を引き起こす可能性があります。用途に応じて適切な感度閾値を選択してください。
3. 衝突検知感度はロボットのハードウェアに影響され、感度の閾値はロボットによって異なります。衝突検知感度に対する要求が高い場合は、単軸設定によって特定の適用シナリオに応じて各軸の感度を微調整したり、RL 命令によって検知感度を調整したりすることができます。
4. 衝突検知および安全監視がトリガーされると、ポップアップウィンドウが表示されます。「OK」ボタンをタップし、手動でアラームを解除する必要があります。
5. 衝突検知機能は、協働ロボットにおいては工場出荷時にデフォルトで有効になっています。



警告

衝突検知を使用する前に、以下のパラメーターが正しく設定されていることを確認することが必要です。さもなくば、コントローラーは正しい出力トルクを計算できず、誤検出する可能性があります。

1. ロボットの型番
2. ロボットの設置方法
3. 負荷情報（ツール）
4. 機械とセンサーのゼロ点
5. 本体パラメーター情報

11.7 安全エリア

11.7.1 機能紹介

安全エリアは、ロボットの先端ツールやエルボー部などが特定のエリアに出入りする際の動作を設定するための機能です。

ユーザーは空間内に複数のエリアを定義でき（現在は最大 10 個まで対応）、ロボットが安全エリアへ進入・退出する際に、あらかじめ設定された安全動作を選択的にトリガーできます。また、安全エリアに関連付けられたレジスタの機能コードを用いて、レジスタ値が自動的に変更されます。



①	全体有効化スイッチ： 安全エリア機能のオン/オフを切り替えます。このスイッチがオフの状態では、すべての安全エリアが無効になります。										
②	信号制御： オンにすると、レジスタ信号を通じて特定の安全エリアの有効・無効を制御できます。各安全エリアの有効・無効の設定には、以下の 2 つの方法があります。 1. HMI で設定 2. レジスタで設定 (レジスタ機能コード <code>enable_safe_region01</code> ～ <code>enable_safe_region10</code> を使用。データ型は <code>bool</code> または <code>int16</code> で、読み取り専用レジスタ)										
③	監視状態： 設定したツールと各エリアとの関係を表示します。 <table border="1"> <thead> <tr> <th>図</th><th>説明</th></tr> </thead> <tbody> <tr> <td></td><td>エリア未有効</td></tr> <tr> <td></td><td>作業エリア内、未トリガー</td></tr> <tr> <td></td><td>禁止エリア外、未トリガー</td></tr> <tr> <td></td><td>作業エリア外、トリガー済み</td></tr> </tbody> </table>	図	説明		エリア未有効		作業エリア内、未トリガー		禁止エリア外、未トリガー		作業エリア外、トリガー済み
図	説明										
	エリア未有効										
	作業エリア内、未トリガー										
	禁止エリア外、未トリガー										
	作業エリア外、トリガー済み										

			禁止エリア内、トリガー済み	
--	--	--	---------------	--

安全エリアの編集

エリア名 : area0

- ① エリア形状 : ☒ 平面
- ② エリアタイプ : ☒ 作業エリア ☐ 禁止エリア

③

点・面ベクトル位置/姿勢

中心点の位置・姿勢 現在位置を取得

位置 : X : 0 Y : 0 Z : 0

ベクトル : X : 0 Y : 0 Z : 0

- ④ トリガー後の動作
- 協働モードに移行
- トリガー後のレジスタの値 : true ⑤

①	エリア形状 : 平面のみ選択できます。
②	エリアタイプ : 作業エリア : ロボットが動けるエリアを設定します。 禁止エリア : ロボットが動けないエリアを設定します。
③	平面は、教示または手動で位置 (X、Y、Z) を入力して設定できます。ベクトルの設定は現在、手動入力のみ対応しています。Z 軸の指す方向が指定するエリアとなります。
④	トリガー後の動作の種類 : トリガー動作には以下の種類があります。 なし : ロボットは特定の動作を行いません。 通常モードの時、安全停止 : ロボットが通常モードのとき、設定したツールが禁止エリアに進入しようとする、計画停止をトリガーします。すでに禁止エリアに進入している場合は、最大減速度で停止します。 協働モードに移行 : 設定したツールが禁止エリアに進入すると、ロボットは協働モードをトリガーします。 協働モードの時、安全停止 : ロボットが協働モードのとき、設定したツールが禁止エリアに進入しようとする、計画停止をトリガーします。すでに禁止エリアに進入している場合は、最大減速度で停止します。 通常モード・協働モードの時、安全停止 : ロボットが通常モードまたは協働モードのとき、設定したツールが禁止エリアに進入しようとする、計画停止をトリガーします。すでに禁止エリアに進入している場合は、最大減速度で停止します。
⑤	安全エリアがトリガーされた後、割り当てられたレジスタの出力を決める設定です。True と False があります ; True : 安全エリアがトリガーされたときに、レジスタ出力信号が True になること

	を意味します（禁止エリアに入ると、出力信号が True になる）。 False：安全エリアがトリガーされたときに、レジスタ出力信号が False になることを意味します（禁止エリアに入ると、出力信号が False になる）。
--	--

11.7.2 安全エリアの設定方法

点・面ベクトル位置/姿勢

中心点の位置・姿勢

現在位置を取得

位置： x : 0 y : 0 z : 0

ベクトル： x : 0 y : 0 z : 0

平面を設定する方法を以下に説明します。

1. 平面中心点の位置（ベース座標系に対する x, y, z オフセット量）を指定します
2. エリアの方向（ベース座標系に対する方向ベクトル）を指定します

例：

ベース座標系の原点に対する安全平面の中心位置が(x, y, z) = (300, 300, 300)で、エリア方向がベース座標系の Z 軸正方向と一致する平面の設定

位置： 300 300 300

ベクトル： 0 0 1

11.7.3 安全エリアの判定

目標位置の先読み結果及び現在位置のフィードバックで判定します。

- 先読みで禁止エリアに入ることが分かった場合は、計画減速停止を実行します。
- 現在位置のフィードバックが禁止エリアに入ると、最大能力で減速停止を実行します。

先読みにより安全平面の禁止エリアに入る前に停止動作を完了し、安全平面との間に安全距離（動作速度に依存）を保ちます。

11.7.4 安全エリア脱出機能

11.7.4.1 概要

先読みでの判定を行うため、ロボットが安全平面の禁止エリアに入る前に停止します。しかし、特定の条件下（安全平面が有効化された際にロボットが既に禁止エリア内にいる場合など）では、ロボットが禁止エリアに入る可能性があります。この場合は、脱出機能を実行してロボットを禁止エリアから脱出させることができます。

11.7.4.2 使用方法

1. ロボットが禁止エリア内にいる場合は、監視状態が赤色で表示され、ロボットは安全停止状態（S）となります（安全平面で、停止する設定にしている場合が対象です。『協働モードに移行』や『なし』を設定している場合は脱出機能は使用できません）。

有効化 ☒ 信号制御 ☐ 脱出動作 ☐

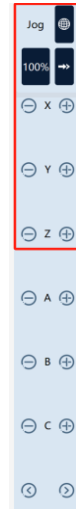
エリア名	有効化	エリア形状	トリガー後の動作	エリアタイプ	監視状態	脱出状態	データ編集
area0	☒	平面	通常モードの時、安全停止	作業エリア		待機	⌵_area0
area1	☐	平面	通常モードの時、安全停止	作業エリア		待機	⌵_area1
area2	☐	平面	通常モードの時、安全停止	作業エリア		待機	⌵_area2
area3	☐	平面	通常モードの時、安全停止	禁止エリア		待機	⌵_area3
area4	☐	平面	協働モードに移行	禁止エリア		待機	⌵_area4
area5	☐	平面	協働モードに移行	禁止エリア		待機	⌵_area5
area6	☐	平面	協働モードに移行	禁止エリア		待機	⌵_area6
area7	☐	平面	協働モードに移行	禁止エリア		待機	⌵_area7

2. 『脱出動作』のボタンを有効化した後、『非常停止リセット』をクリックすることで、安全停止状態（S）を解除できます。

有効化 ☒ 信号制御 ☐ ☒ 脱出動作

エリア名	有効化	エリア形状	トリガー後の動作	エリアタイプ	監視状態	脱出状態	データ編集
area0	☒	平面	通常モードの時、安全停止	作業エリア		待機	⌵_area0
area1	☐	平面	通常モードの時、安全停止	作業エリア		待機	⌵_area1
area2	☐	平面	通常モードの時、安全停止	作業エリア		待機	⌵_area2
area3	☐	平面	通常モードの時、安全停止	禁止エリア		待機	⌵_area3
area4	☐	平面	協働モードに移行	禁止エリア		待機	⌵_area4
area5	☐	平面	協働モードに移行	禁止エリア		待機	⌵_area5
area6	☐	平面	協働モードに移行	禁止エリア		待機	⌵_area6
area7	☐	平面	協働モードに移行	禁止エリア		待機	⌵_area7

3. ロボットはモーターの電源オン及び作業エリア側への Jog 操作が可能になります（直交座標系の X, Y, Z 方向の Jog 操作のみが可能で、その他の操作は許可されません）。



脱出方向のベクトルと安全平面作業エリア方向のベクトルの内積が 0 未満の場合（脱出方向が禁止エリアから遠ざからない場合）は、コントローラーがエラーを報告し、以下のようなエラーメッセージが表示されます。

詳細

内容: [50042]脱出方向が違います。作業エリアに向かって動作してください

原因: 脱出方向が違います。作業エリアに向かって動作してください

修復方法: 作業エリアに向かって動作してください

4. 脱出順序

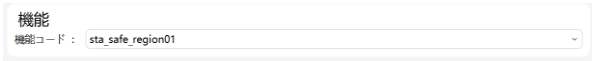
- 複数の安全平面を有効化している場合は、脱出機能は停止をトリガーしたすべての平面のうち最初に停止をトリガーした安全平面の作業エリアの方向を基準にします。
- 停止をトリガーしたすべての平面のうち、脱出方向が同じ平面を同時に脱出します。
- 上記の平面からの脱出完了後、残りの平面の脱出を行います（引き続き、残った安全平面の中で最初に停止をトリガーした平面の作業エリアの方向を基準にします）。
- 脱出状態は下の図のように判断することが可能です。脱出中の状態は『脱出中』と表示され、脱出中でない場合は『待機』と表示されます。

有効化		信号制御	脱出動作				
エリア名	有効化	エリア形状	トリガー後の動作	エリアタイプ	監視状態	脱出状態	データ編集
area0	☒	平面	通常モードの時、安全停止	作業エリア		脱出中	_area0
area1	☐	平面	通常モードの時、安全停止	作業エリア		待機	_area1
area2	☐	平面	通常モードの時、安全停止	作業エリア		待機	_area2
area3	☐	平面	通常モードの時、安全停止	禁止エリア		待機	_area3
area4	☐	平面	協働モードに移行	禁止エリア		待機	_area4
area5	☐	平面	協働モードに移行	禁止エリア		待機	_area5
area6	☐	平面	協働モードに移行	禁止エリア		待機	_area6
area7	☐	平面	協働モードに移行	禁止エリア		待機	_area7

5. 脱出が完了すると、自動的に有効化ボタンがオフになります。

11.7.5 安全エリアとレジスタの関係

11.7.5.1 安全エリアの状態を出力する

手順	図	説明
1. レジスタを新規作成し、書き込み専用にします。		左の図は例です
2. "sta_safe_region01～sta_safe_region10"は、対応する安全エリアのトリガー状態を現在の新しいレジスタに割り当てます。		

11.7.5.2 レジスタによる安全エリアの有効/無効の制御

手順	図	説明
1. レジスタを新規作成し、読み取り専用にします。		左の図は例です
2. "enable_safe_region01～enable_safe_region10"は、対応する安全エリアの制御スイッチを現在の新しいレジスタに割り当てます。		

11.8 ツール設定

11.8.1 ツール位置制限

ツール位置制限は、フランジ、エルボー、現在のツール、および 2 つの固定ツールの位置に対して同時に制限をかけることができます。各位置には仮想形状を設定でき、いずれかの仮想形状が安全エリアに設定された範囲を超えた場合、設定されている動作（通常モードのとき安全停止、協働モードへ移行など）がトリガーされます。

11.8.2 ツール姿勢制限

①	有効化スイッチ： ツール姿勢制限機能をオン/オフするためのスイッチです。
②	姿勢制限機能は、1つの対象のみを監視できます。対象はフランジ、ツール1、ツール2です。
③	角度： 姿勢制限機能をオンにしたときの対象の姿勢が基準となり、設定された角度によって形成される円錐体が姿勢の許容範囲として設定されます。 対象の姿勢が円錐体の範囲を超えた場合、安全停止がトリガーされます。

11.9 安全位置

11.9.1 機能紹介

安全位置機能とは、ロボットが事前に設定された安全位置にある場合、割り当てられたレジスタが信号を出力する機能です。この機能を使用することで、ユーザーはロボットと安全位置との相対的な位置関係を把握することができます。

xCore制御システムでは、最大8つの安全位置を関節角度を基準に設定することができます。各安全位置には、1つのレジスタ機能コードが対応します（タイプ：bool または int16、読み書き属性：書き込み専用、sta_safe_jnt_pos1 ~ sta_safe_jnt_pos8）。ロボットの現在の関節角度がある安全位置に設定された関節角度の範囲内にあるとき、その安全位置に割り当てられているレジスタ関数コードのレジスタ値が自動的に変更されます（安全位置の範囲内にあるとき、レジスタタイプが bool であればレジスタ値は True になり、レジスタタイプが int16 であればレジスタ値は 1 になります）。

安全ホーム位置は、安全コントローラーがサポートする機能で、1つの安全位置を安全ホーム位置として選択できます。選択後、ロボットのすべての関節が設定された範囲に到達すると、安全 DO 信号を出力することができます。いずれの安全位置も選択しない場合、安全ホーム位置機能は無効となります。

① 有効化

② 安全ホーム位置

③ 番号

位置1

④ 軸角度 (°)

⑤ 許容誤差 (°)

軸1 : 0

軸2 : 0

軸3 : 0

軸4 : 0

軸5 : 0

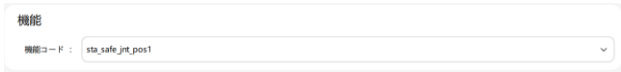
軸6 : 0

安全ホーム位置出力遅延時間: 0秒

⑥ 位置を更新

①	有効化スイッチ： 特定の安全位置を有効にするかどうかを設定します。
②	安全ホーム位置： 1つの安全位置を安全ホーム位置に設定します。
③	番号： 安全位置の番号です。選択すると、右側で設定ができます。
④	軸角度： 手動で更新するか位置を更新をタップして、ロボットの現在の関節位置データで更新できます。
⑤	許容誤差： ロボットの現在の軸角度が、④で設定した軸角度を基準として許容誤差以内であれば、安全位置にあるとみなされます。
⑥	安全ホーム位置出力遅延時間 安全ホーム位置の出力遅延時間を設定することができます。 設定できる範囲は[0, 9.9]です。

11.9.2 安全位置とレジスタの関連付け

手順	図	説明
1. レジスタを新規作成し、書き込み専用にします。		左の図は例です
2. "sta_safe_jnt_pos1 ~ sta_safe_jnt_pos 8" は、対応する安全位置のフィードバック状態を、現在の新しいレジスタに割り当てます。		

11.10 安全チェックサム

安全設定を変更するには、画面右下の「確認」ボタンをクリックし、安全チェックを通過する必要があります。

「安全チェックサム」アイコンは、4桁の「数字+アルファベット」の組み合わせで表示され、ユーザーに安全関連の設定状態を知らせます。安全関連の設定が変更されると、新しい4桁の「アルファベット+数字」の組み合わせが自動的に計算されます。

このアイコンをタップすると、現在のロボットの安全設定情報が表示されます。軸制限、ロボット制限、バーチャルフェンス、衝突検知、安全エリア、ツール設定、安全位置の項目があります。これらのパラメーターを変更した後「確認」ボタンをタップすると、安全チェックサムのポップアップウィンドウが表示されます。「確定」ボタンをタップし安全パラメーターが設定されると、安全チェックサムもパラメーターとともに変更されます。



11.11 安全コントローラー

xCore 制御システムは RSC（安全コントローラー）をオプションで追加できます。安全コントローラーは、機能安全認証に適合する安全モジュールで、内部で多くの安全関連の計算と保護を実行します。xCore は制御システム本来の安全機能を並行して制御し、二重保障を形成します。

安全コントローラーのデータ安全とパラメーター安全を確保するため、リアルタイムデータは FSoe 通信メカニズムを採用し伝送の正確性を確保します。非リアルタイムデータは安全同期メカニズムを採用し、非リアルタイムデータの同期は約 5-10 秒間がかかります。

安全コントローラーを採用するロボットの安全モジュールのタイプは、下図に示すように ROKAE_RSC に設定します。



11.11.1 安全コントローラーを追加した後の変更点

安全コントローラーを使用したロボットの使用にはいくつかの変更点があります。

11.11.1.1 ロボットモーター状態の変化

ロボットの状態に「安全停止状態」が追加されます。

安全停止状態		ロボットは安全停止状態にあります。安全コントローラーが作業や通信の異常を検出したか、またはパラメーターが安全コントローラーに設定された安全閾値を超えたことを示します。この状態ではロボットモーターの電源をオンにできません。
--------	---	--

11.11.1.2 ロボットリセット操作の追加

ロボットが「非常停止」、「セーフティーゲート」、「安全停止」のいずれかになった場合、「モーター電源オフ状態」に戻るためには、以下の 2 つのステップを完了する必要があります。

ステップ 1：上記の 3 つの状態をトリガーする操作または条件をクリアします。例えば、非常停止ボタンを元の位置に戻したり、セーフティーゲートのトリガー信号をクリアしたり、機械の安全制限超過要因をクリアしたり、または対応する安全制限機能をオフにしたりします。

ステップ 2：下図に示すインターフェースで「非常停止リセット」ボタンをタップし、ロボットを「モーター電源オフ状態」にリセットします。



11.11.1.3 セーフティゲートのロジックの変更

安全コントローラーなしの場合：

ロボットが自動モードにある場合、セーフティゲートが開いた信号を受信すると、ロボットのモーターは直ちに電源オフになります。

安全コントローラーありの場合：

ロボットが自動モードにある場合、セーフティゲートが開いた信号を受信すると、**RL** プログラムが実行中であれば、**RL** プログラムの実行が一時停止しますが、ロボットのモーター電源オフにはなりません。この場合、ロボットは**RL** プログラムの実行や単ステップ実行ができなくなります。ロボットの状態を回復させるには、次の操作を行う必要があります：

- セーフティゲート信号を解除する
- HMI で「リセット」信号をタップする

11.11.1.4 ゼロキャリブレーション及び摩擦パラメーター設定時の時間差

安全コントローラーの基本的な安全制限機能が適切に使用されるために、ロボットのゼロキャリブレーション情報と摩擦パラメーターに関する情報を安全コントローラーに同期させる必要があります。そのため、ユーザーがゼロキャリブレーションを実行、または摩擦パラメーターを設定すると、コントローラーは更新されたパラメーターを自動的に安全コントローラーに同期させます（約 5～10 秒待ちます）。この時のインターフェースは下図に示すようになり、この間ユーザーはロボットを操作できません。



12 パッケージ

xCore 制御システムには、充実した基本機能に加え様々な拡張機能があります。以下のプロセスパッケージについて、専用のドキュメントをご用意しておりますので弊社にお問い合わせください。

12.1 コンベアトラッキング

主な特性と制限：

- 直線型コンベアトラッキングをサポートします。
- 光電センサーと 2D ビジョンの 2 つの位置決め/トリガー方法をサポートします。
- 6 軸産業用ロボットをサポートします。
- xMate 協働ロボットについては、制御キャビネット搭載のバージョンのみをサポートします。



12.2 走行軸

主な特性と制限：

- 直線型走行軸のみをサポートします；



12.3 パレタイジング

主な特性と制限：

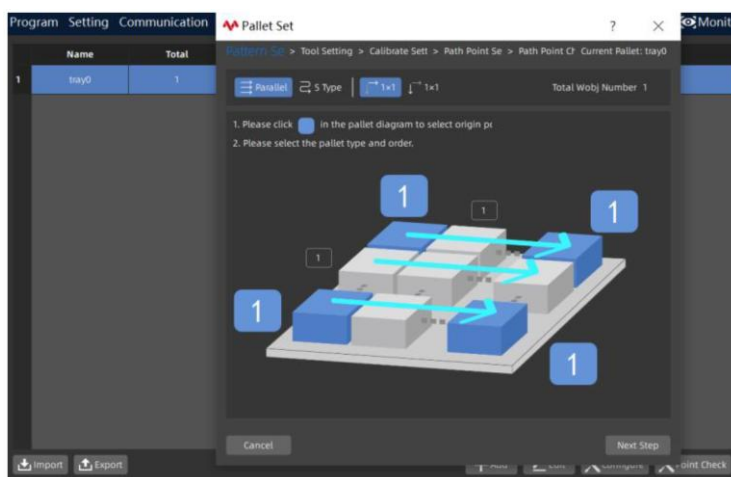
- 最大 100 のパレタイジングプロセスの作成をサポートします。
- ブロック積み、インターロック積み、ピンホール積みなど様々な積みタイプをサポートし、ユーザー定義積みタイプもサポートします。

- 1つのパレタイジングプロセスは、最大 100 種までの平面パターンを作成でき、1つの平面パターンは最大 200 個までの部品を作成できます。1つのパレタイジングプロセスは最大 50 段まで作成できます。

12.4 トレイ

主な特性と制限：

- 最大 100 個までのトレイプロセスをサポートします。
- ロボットの積み付けきモードをカスタマイズ可能です（平行モード、S 型モードなど）
- 1つのドレイプロセスは 16 種の平面パターンを提供でき、1つのフラットパターンは最大 999 個までのワークを扱えます。



12.5 ソーラーバッテリー組立

詳細は、弊社にお問い合わせください。

12.6 ソーラーバッテリー挿入

詳細は、弊社にお問い合わせください。

13 ログ

ログモジュールには、HMI ログ、コントローラーログ、実行ログ、ログタイムライン、内部ログ、診断設定、ハードウェアステータスのインターフェースがあります。

13.1 HMI ログ

HMI ログインターフェースは、現在のユーザーの HMI 操作ログ情報を表示します。本インターフェースには、フィルタエリア、検索エリア、表示エリア部分があります。ユーザーは「エクスポート」ボタンをタップして必要な HMI ログ情報をエクスポートすることができます。



①	フィルタエリア： 現在のコントローラーまたは HMI に接続されているコントローラーのログの表示内容をフィルタリングできます。また、ログレベルを選択してフィルタリングを行うことができます。
②	検索エリア： キーワードに基づいてログを検索できます。
③	ログ表示エリア： ログの内容、ユーザー、タイプ、日時などを表示します。ページ送りボタンで前・次のページを切り替えることができます。
④	エクスポート： ボタンをタップして、ログを csv ファイルにエクスポートできます。

13.2 コントローラーログ

コントローラーログインターフェースは、現在のロボットのコントローラーログ情報を表示します。本インターフェースには、フィルタエリア、検索エリア、表示エリアがあり、ユーザーはここでコントローラー情報を見ることができます。

① ☐ 通知 ☐ 警告 ☐ エラー

② 検索

ID	タイトル	日時	内容
13045	① コントローラとRSCのSDOデータの同期...	2024-06-25 17:33:13	コントローラとRSCのSDOデータの同期化に成功
10004	① 機械的ゼロ点校正	2024-06-25 17:33:13	機械的ゼロ点校正
10133	⚠ ホットプラグモードの切り替えに失敗し...	2024-06-25 16:01:10	ホットプラグモードの切り替えに失敗しました
10133	⚠ ホットプラグモードの切り替えに失敗し...	2024-06-25 16:01:03	ホットプラグモードの切り替えに失敗しました
10133	⚠ ホットプラグモードの切り替えに失敗し...	2024-06-25 16:00:55	ホットプラグモードの切り替えに失敗しました
14002	① ネットワーク接続が切断されました	2024-06-25 15:41:50	ネットワーク接続がダウンしています。バスデバイス...
③ 14004	① ネットワーク接続の受信が開始されました	2024-06-25 15:36:27	ネットワーク接続[OpC UA Server]のリスニングが開始...
14021	✖ バスデバイスを開けませんでした	2024-06-25 15:09:11	バスデバイスを開けませんでした。デバイス...
14021	✖ バスデバイスを開けませんでした	2024-06-25 15:09:10	バスデバイスを開けませんでした。デバイス...
14021	✖ バスデバイスを開けませんでした	2024-06-25 15:09:09	バスデバイスを開けませんでした。デバイス...
14021	✖ バスデバイスを開けませんでした	2024-06-25 15:08:52	バスデバイスを開けませんでした。デバイス...
10001	✖ JOGの起動に失敗しました	2024-06-25 13:17:22	JOGの起動に失敗しましたロボットの電源が入って...
10001	✖ JOGの起動に失敗しました	2024-06-25 12:43:52	JOGの起動に失敗しましたロボットの電源が入って...
10001	✖ JOGの起動に失敗しました	2024-06-25 11:51:41	JOGの起動に失敗しましたロボットの電源が入って...

前のページ 1 / 8 次のページ

①	フィルタエリア： ログレベルを選択してフィルタリングを行うことができます。
②	検索エリア： キーワードに基づいてログを検索できます。
③	ログ表示エリア： ログの ID、タイトル、発生時間などの基本情報を表示します。ページ送りボタンで前・次のページを切り替えます。

13.3 実行ログ

実行ログは主にロボットが実行した主要な命令を記録します。実行ログには以下の情報が記録されます。

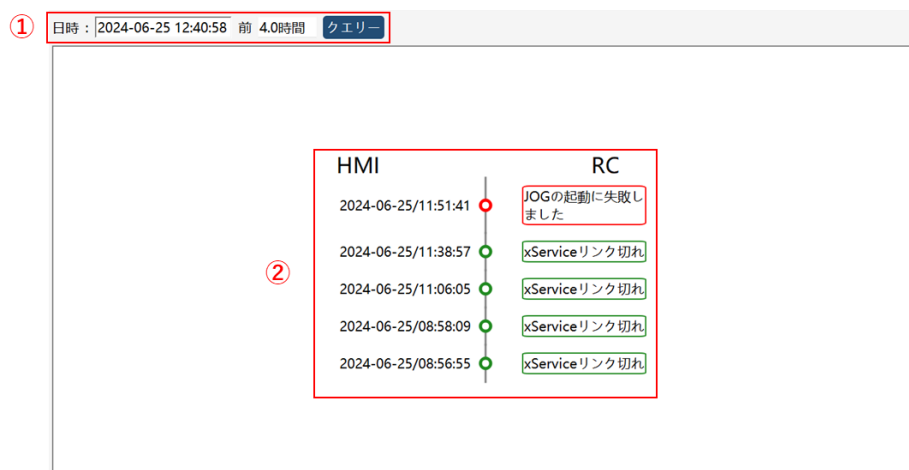
1. ロボットの起動・停止および動作命令の実行状況
2. IF-ELSE、WAITUNTIL の 2 つの論理命令の実行状況
3. SetDO、PulseReg、PulseDO 命令の実行状況
4. JOG 操作、ドラッグ
5. 非常停止、一時停止、起動などの操作
6. 外部への出力レジスタ状態機能コード、およびロボットを制御する外部から受信した制御機能コード
7. 安全エリアの変化状況
8. 衝突検知、診断データ保存などの機能トリガー
9. 通信命令指令を介して読み取った外部データ



①	フィルタエリア： ログの種類を選択してフィルタリングを行うことができます。
②	検索エリア： キーワードに基づいてログを検索できます。
③	ログ表示エリア： ログのタイトル、発生時間、内容などの基本情報を表示します。ページ送りボタンで前・次のページを切り替えます。

13.4 ログタイムライン

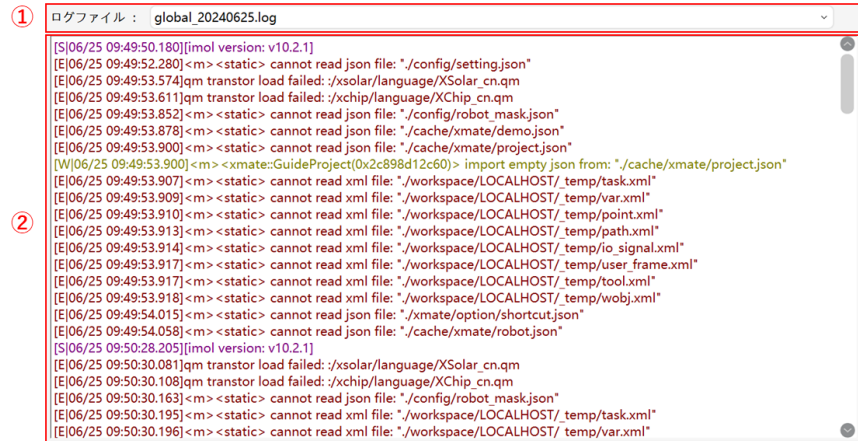
ログタイムラインインターフェースは、時系列に応じて HMI とコントローラーのログを同時に表示します。本インターフェースには検索エリアと表示エリアがあり、検索エリアから同じ時間帯の HMI ログとコントローラーログを検索できます。



①	検索エリア： 時間を設定してログを検索できます。
②	ログ表示エリア： 左側は HMI ログで、右側はコントローラーのログです。

13.5 内部ログ

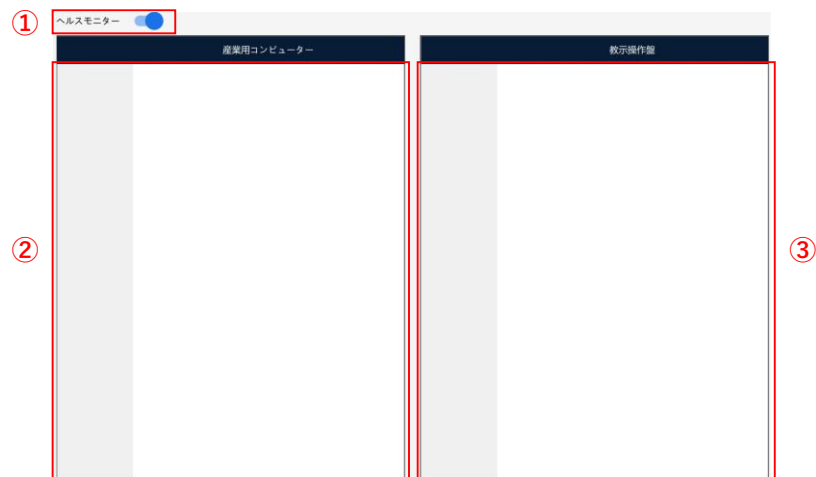
内部ログは、技術的な問題、ロボット故障などの状況にシューティング根拠、問題原因の確認などを提供する機能を持ちます。本インターフェースは、検索エリアと表示エリアの 2 つの部分から構成されています。



①	内部ログ選択エリア
②	内部ログ表示エリア

13.6 ハードウェアの状態

このインターフェースは、産業用コンピュータおよび教示操作盤のハードウェアの状態を監視するためのものです。メモリ使用率、ディスク使用率、ネットワークカードの状態、CPU の周波数、温度、占有率などがあります。このインターフェースには、オン・オフスイッチ、産業用コンピュータのハードウェア状態表示エリア、教示操作盤のハードウェア状態表示エリアの部分があります。



①	ハードウェア状態監視スイッチ
②	産業用コンピュータのハードウェア状態表示エリア
③	教示操作盤のハードウェア状態表示エリア

13.7 診断設定

このインターフェースは、サーボや ECAT などの設備故障の診断を支援し、リアルタイムのスレッドアラーム監視などの機能を有効にするために使用されます。診断機能を有効にするとコントローラーの動作負荷が増加するため、実際の生産現場では必要な場合以外は有効にしないでください。

診断設定

詳細オプション

サーボ診断

有効化

collision_detected

rci_error

safety_monitor_normal

fc_error

safety_control_rsc_slave_com_error

jnt_jump

safety_monitor_collab

診断データの保存: 5秒後に一連の診断データが保存されます。

診断データの保存

診断データのエクスポート: コントローラーの診断データを、HMIの診断データフォルダーにエクスポートします。

データをエクスポート

フォルダーを開く: HMIの診断データフォルダーを開きます。

フォルダーを開く

診断設定

詳細オプション

開発者がサーボやECATなどの機械の故障を診断することを支援し、リアルタイムのスレッドアラーム監視などの機能を有効にするために使用されます。診断機能を有効にするとコントローラーの動作負荷が増加するため、実際の生産現場では必要な場合以外は有効にしないでください。

EC診断

ECソフトウェア診断機能は、ECATデバイスの問題のトラブルシューティングに役立ちます。

有効化

動作開始遅延

有効にすると、ボタンを押してからロボットが動き出すまでの時間を延ばします。ロボットの精度や速度が最適化されます。

有効化

リアルタイム警告

有効にすると、リアルタイムに警告が出されます。

有効化

滑らか軌跡エリア先読み警告

有効にすると、滑らか軌跡エリアの先読みに関する警告が出されるようになります。

有効化

滑らか軌跡エリア警告

有効にすると、滑らか軌跡エリアに関する警告が出されるようになります。

有効化

サーボ診断	サーボ診断は、サーボのエラーデータを保存するために使用されます。保存ボタンをタップすると、5 秒後に診断データが保存されます。
動作開始遅延	有効にすると、動作遅延が延長されます。
EC 診断	EC 診断機能は、ECAT 設備のトラブルシューティングに役立ちます。
リアルタイム警告	有効にすると、スレッドタイムアウト警告をリアルタイムでアップロードできます。
滑らか軌跡エリア先読み警告	有効にすると、滑らか軌跡エリアの先読みに関する警告を報告します。
滑らか軌跡エリア警告	有効にすると、滑らか軌跡エリアに関する警告を報告します。

14 オプション

オプションページは、主にロボット接続、ソフトウェアアップグレード、インポート・エクスポート、機能デモンストレーションに関連します。

14.1 接続

①

ロボットを検索

システムは、ロボット検索用IPアドレスの監視によりロボットを自動的に検出します。ネットワークカードのIPアドレスを変更する場合は、[ここをクリックしてIPアドレスを変更できます](#)。本操作は再起動後に有効になります。

ロボット検索用IPアドレス: 0.0.0.0

使用可能なロボットを検索

②

ロボット接続

コントローラーサービス: [192.168.21.9:5050]に接続

アップグレードサービス: [192.168.21.9:4567]に接続

IPアドレス: 192.168.21.9

切断する

③

自動再接続

ボタンをオンにすると、自動再接続機能が有効になります。再接続には次の2つの方法があります。1. 再接続時間と回数を使用して再接続の間隔と回数を指定する。; 2. 再接続時間と回数を使用せず、常時再接続する。

☒ 有効化

④

☒ 1回の時間	20	(10~30s)	再接続試行回数	1	(1~100)
---	----	----------	---------	---	---------

①	ロボット検索： 同一 LAN 上のすべてのロボットを検索します（直接接続の場合を除く）。ロボットと接続すると、コントローラーサービスとアップグレードサービスの両方が接続されていることを表示します。 RobotAssist ソフトウェアがロボットと同一ネットワークにあってもロボットを検索できない場合、またはロボットと接続した後で 3D インターフェースにロボットのリアルタイム位置が表示されない場合は、「設定」-「HMI 設定」インターフェースのロボット検索用 IP アドレスのドロップダウンボックスで、LAN に割り当てられた IP アドレスを変更して、上記の 2 つの問題を解決することができます。
②	接続状況を表示します。ユーザーが意図的に切断することもできます。
③	自動再接続機能を有効化。 ロボットコントローラーと RobotAssist ソフトウェア間のネットワークが切断されると、RobotAssist ソフトウェアは自動的に再接続を試みます。設定された再接続時間が経過すると、接続を試みなくなります。
④	2 つの再接続方法： (1) チェックボックスをチェックする場合、再接続合計時間=再接続時間*再接続回数の間再接続を試みます。ユーザーは再接続時間と再接続回数を指定できます。 (2) チェックボックスをチェックしない場合、RobotAssist ソフトウェアはコントローラーとの再接続を試み続けます。

14.2 ROKAE について

ソフトウェアバージョン、コンポーネント情報及び会社概要を紹介します。

The screenshot displays the ROKAE software interface. At the top, it shows the version information: HMI: 5.89.10.0181, Controller: 2.2.1, and Upgrade Service: 0.7.1. Below this is the ROKAE logo and a brief description of the company's focus on industrial robots and automation. A section titled 'コンポーネント情報' (Component Information) contains a table with the following data:

種類	メーカー	バージョン
1 セキュリティボード	rokae	
2 コントローラー	rokae	2.2.1
3 通信バス	rokae	3.2.8

14.3 ソフトウェア更新

14.3.1 コントローラーのアップグレード

コントローラーアップグレード機能により、以下のことが実現できます。

- (1) コントローラーソフトウェアのバージョンアップ
- (2) データの復元

コントローラーのバージョンアップ:

アップグレードファイルを選択し、「アップグレードを実行」をタップします。インターフェースには「アップグレード成功」と表示されます。ポップアップウィンドウのヒントに従って再起動してください。

データの復元:

復元するパッケージを選択し、復元するデータにチェックを入れて「アップロードを実行」をタップします。ポップアップウィンドウのヒントに従って再起動してください。

14.3.2 コントローラーのバックアップ

コントローラー内のデータをバックアップする機能です。目標のバックアップデータにチェックを入れ、「参照」をタップしてバックアップパスを選択し「エクスポート」をタップします。エクスポートされるファイルは rpa 形式の暗号化されたファイルです。

The screenshot shows the 'コントローラーのバックアップ' (Controller Backup) dialog box. It contains the following options and fields:

- 必要なファイルをすべてパッケージ化し、ローカルフォルダーに保存します。
- バックアップオプション:
 - ☒ カスタム構成
 - ☒ ロボット構成 ☒ 本体パラメーター
 - ☒ コントローラーログ
 - ☒ HMIログ
 - ☒ プロジェクトデータ
 - ☐ 現在のサーボパラメーター (サーボパラメータのエクスポートには時間がかかります!)
- エクスポート先: エクスポートファイル名:
-

ロボット構成は、コントローラー設定-システム設定、キャリブレーション-トルクセンサーのゼロ点電圧値、本体パラメーター-過負荷係数、運動パラメーター-パラメーター、安全制御などを含みます。

本体パラメーターは、キャリブレーション-エンコーダー値、角度キャリブレーション設定、ベース座標系、動力学設定- (3 次) 摩擦係数、本体パラメーター-RD パラメーター、減速比

などを含みます。

カスタム構成は、動力学設定-動的フィードフォワード、動力学制約スイッチ、運動パラメーター-Search 停止距離、最小滑らか軌跡エリアの半径、Conf、力制御パラメーター、xPanel 設定、クイック移動、教示操作盤モード、通信、安全機能を含みます。

14.3.3 HMI アップグレード

xPad2 のみに適用され、教示操作盤 HMI ソフトウェアのアップグレードのために使用されます。「開く」をタップして USB メモリのディレクトリにある HMI ソフトウェアアップグレード圧縮ファイルを選択します。「アップグレード」ボタンをタップして HMI アップグレードプロセスを始めます。アップグレードが完了した後、HMI ソフトウェアが自動的に起動し、アップグレードが完了します。



14.3.4 ロボット再起動

産業用コンピューターシステムの再起動に使用されます。本操作を行うにはアップグレードサービス接続の確立が必要です。

ロボット制御装置を再起動します。通常は、工場出荷時の設定に戻す/コントローラーが起動しない状態で使用します。

ロボットを再起動

14.3.5 構成を消去

カスタム設定、ロボット設定、本体パラメーター、プロジェクトデータなどを消去するために使用します。上記データの説明については「コントローラーのバックアップ」の章を参照してください。設定消去を行っても、制御システムの関連実行ログは依然として保存されます。設定の消去は、慎重に行ってください。

消去したい内容にチェックを入れ「構成を消去」ボタンをクリックし、ロボットを手動で再起動することで有効になります。

注意：この操作を行うには、アップグレードサービス接続の確立が必要です。

選択した構成を消去します。消去した後、ロボットを再起動する必要があります！

- ☒ カスタム構成
- ☒ ロボット構成 ☒ 本体パラメーター
- ☒ プロジェクトデータ

構成を消去

14.3.6 出荷時の設定に戻す

制御システムを工場出荷時の初期設定に戻すために使用します。制御システム内の設定ファイル、プロジェクト、及びユーザー定義設定はリセットされます。ログは保存されます。慎重に操作してください。

注意：この操作を行うには、アップグレードサービス接続の確立が必要です。

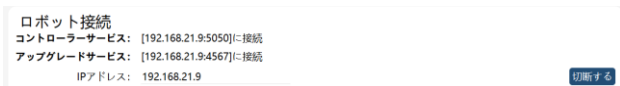
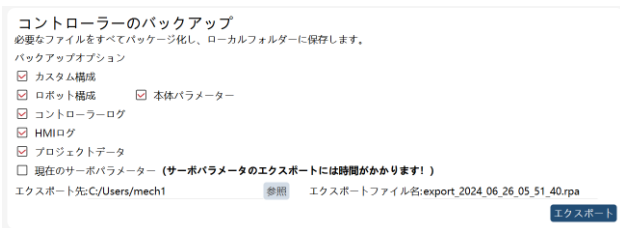
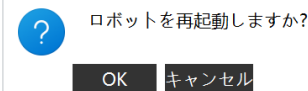
コントローラーを工場出荷時の設定に戻します。実行可能プログラム、構成ファイル、プロジェクトデータ、およびカスタム構成をリセットします。ログは保持されます。

出荷時の設定に戻す

14.3.7 制御システムのアップグレード操作例

制御システムをアップグレードしたい場合は、弊社にご連絡ください。新しいバージョンの制御システムインストールパッケージを入手することができます。

操作の手順は以下の通りです。

手順	図	説明
1. インストールパッケージを準備します。		
2. HMI を開き、コントローラーと接続し、アップグレードサービスに接続します。		
3. アップグレードする前に、重要なデータを失わないように、制御システムのバックアップを実行することを推奨します。「バックアップオプション」でバックアップの内容をチェックし、エクスポートファイルを選択してバックアップをエクスポートします。		この手順は必須ではありません。
4. インストールパッケージを選択し、アップグレードを実行をタップします。		コントローラーアップグレードオプションのカスタム構成、ロボット構成、コントローラーログ、プロジェクトデータ、デモンストレーション、サーボにチェックを入れないでください。
5. アップグレードに成功すると、HMI はロボットの再起動を促します。制御システムのアップグレードを完了するために再起動してください。		



ヒント

1. コントローラーのバージョンと HMI のバージョンが一致しない場合、HMI は上部ステータスバーに「バージョンが一致しないため、HMI バージョン[xxx]を推奨します」と表示します。
2. コントローラーのバージョンと構成ファイルのバージョンが一致しない場合、電源投入後、HMI は下図のような警告をポップアップ表示します。ロボットの動作は一切禁止されるため、コントローラーのログメッセージに従って正しいバージョンの構成ファイルにアップグレードする必要があります。



バージョンが一致しません。コントローラーのログを確認し、正しいバージョンのモデルをインポートしてください。

確定

14.4 エクスポート

エクスポート機能を通じて、コントローラーの関連設定をバックアップします。



機能モジュール	エクスポート内容
プログラミング	特定の単一プロジェクトをエクスポートできます
設定	コントローラー設定-その他の設定
	HMI 設定-基本設定、テーマ、ワークスペース保存先
	ユーザーグループ
	動力学設定-動的フィードフォワード、動力学制約
	本体パラメーター設定-RD パラメーター、減速比、過負荷係数
	力制御パラメーター設定-力制御パラメーター、力制御モデル、ドラッグの最適化
	運動パラメーター設定-モーションパラメーター、AccSet、安全制御、Search 命令、滑らか軌跡最小半径、デフォルト Conf
	エラーコードフィルター
通信	カスタムボタン
	システム IO
	外部通信
	IO デバイス
	バスデバイス
	先端ツール
	RCI 設定

	シリアルポート設定
	エンコーダー
安全	軸制限
	ロボット制限
	バーチャルフェンス
	衝突検知
	安全エリア
	安全位置
パッケージ	コンベア
ログ	コントローラーログ
	HMI ログ
オプション	接続、自動再接続
ロボット構成	モデルファイル

14.5 インポート

インポート機能により、コントローラーの設定をインポートすることができます。



インポートしたい ZIP パッケージを選択します。パッケージを開くと、含まれている設定項目には自動的にチェックが入り、含まれていない設定項目はグレイアウトされます。

ユーザーはニーズに応じて、特定のインポート内容を選びます。

注意：インポートした後、制御システムを再起動すると有効になります。

14.6 ファイルマネージャー

ファイルマネージャーは、主に RobotAssist ソフトウェアに関係する複数のフォルダーにすばやくアクセスするために使用します。

注意：この機能は、PC の HMI ソフトウェアのみに搭載されています。

キャッシュフォルダー：RobotAssist ソフトウェアのキャッシュが保存されます。

ログフォルダー：RobotAssist ソフトウェアのログが保存されます。ここでのログの内容は、ログインターフェースの内部ログと一致します。ここをタップしてフォルダーに入り、ログをコピーします。

ワークスペースフォルダー：ロボットのプロジェクトファイルが保存されます。

14.7 デモンストレーション

14.7.1 7 軸冗長動作

7 軸冗長動作デモは、xMate ER PRO で使用可能です。円弧、直線、滑らか軌跡エリア、零空間動作などを実演します。

ロボット機能のデモンストレーション

▶ デモを開始 ▶ デモを終了

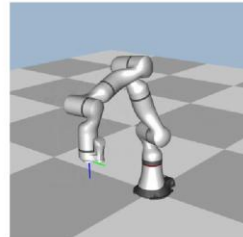
- 7軸冗長動作
- 障害物回避動作
- 衝突検知
- インピーダンス制御

Introduction

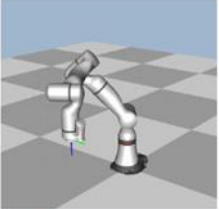
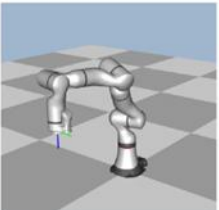
円弧、直線、カーブ、零空間動作など、7軸冗長ロボットの一般的な動作の例。

注記: デモ中は、すべてのロボット設定が無効になります。デモは、ワールド座標系とベース座標系が一致する場合にのみ使用できます。

デモンストレーション



手順	図	説明
1. オプション-デモンストレーション-7 軸冗長動作を選択し、デモスイッチをオンにしてデモモードに入ります。		
2. 下部ステータスバーにある操作モードの切替ボタンを タップして、ロボットを自動モードに切り替えます。		

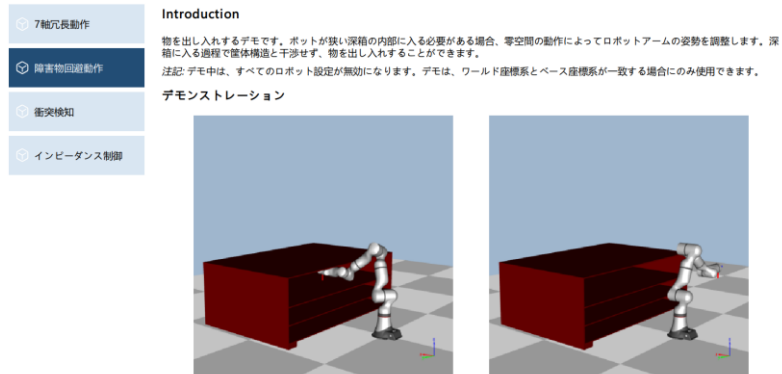
3. 下部ステータスバーのモーター電源状態をタップしてモーター電源を入れます。	ロボット機能のデモンストレーション  ▶ デモンストレーションの開始 ■ デモを終了 7軸冗長動作 Introduction 円弧、直線、カーブ、零空間動作など、7軸冗長ロボットの一般的な動作の例。 注記 デモ中は、すべてのロボット設定が無効になります。デモは、ワールド座標系とベース座標系が一致する場合にのみ使用できます。 デモンストレーション  50% ユーザー管理 xMateR3 Pro	
4. 右上のデモンストレーションの開始をタップします。	ロボット機能のデモンストレーション  ■ デモンストレーションの一時停止 ■ デモを終了 7軸冗長動作 Introduction 円弧、直線、カーブ、零空間動作など、7軸冗長ロボットの一般的な動作の例。 注記 デモ中は、すべてのロボット設定が無効になります。デモは、ワールド座標系とベース座標系が一致する場合にのみ使用できます。 デモンストレーション  50% ユーザー管理 xMateR3 Pro	
5. デモンストレーション動作を終了する際は、右上のデモを終了をタップします。	ロボット機能のデモンストレーション  ▶ デモンストレーションの開始 ■ デモを終了 7軸冗長動作 Introduction 円弧、直線、カーブ、零空間動作など、7軸冗長ロボットの一般的な動作の例。 注記 デモ中は、すべてのロボット設定が無効になります。デモは、ワールド座標系とベース座標系が一致する場合にのみ使用できます。 デモンストレーション  50% ユーザー管理 xMateR3 Pro	

14.7.2 障害物回避動作

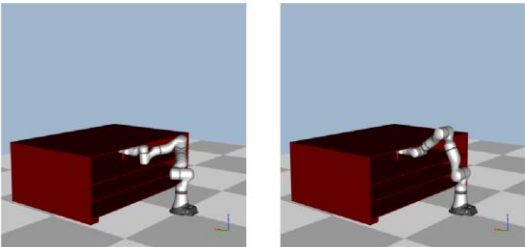
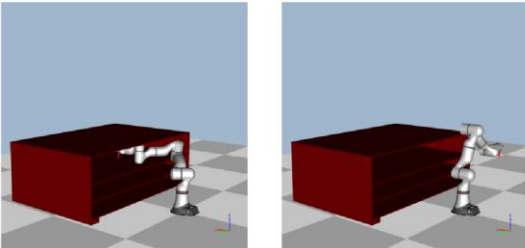
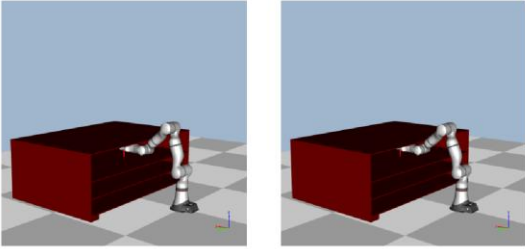
障害物回避動作デモは、xMate ER PRO で使用可能です。狭くて深い箱の内部に入る動作を行います。箱の構造に干渉しないようにロボットアームの姿勢を零空間動作で調整し、物の出し入れを完成するタスクを実演します。

ロボット機能のデモンストレーション

▶ デモンストレーションの開始 ■ デモを終了



手順	図	説明
1. オプション-デモンストレーション-障害物回避動作を選択し、デモスイッチをオンにしてデモモードに入ります。		
2. 下部ステータスバーにある操作モードの切替ボタン  をタップして、ロボットを自動モードに切り替えます。		

3. 下部ステータスバーのモーター電源状態をタップしてモーター電源を入れます。	ロボット機能のデモンストレーション  ▶ デモンストレーションの開始 ■ デモを終了 7軸冗長動作 障害物回避動作 衝突検知 インピーダンス制御 Introduction 物を出し入れるデモです。ボットが狭い深箱の内部に入る必要がある場合、零空間の動作によってロボットアームの姿勢を調整します。深箱に入る過程で筐体構造と干渉せず、物を出し入れることができます。 注記: デモ中は、すべてのロボット設定が無効になります。デモは、ワールド座標系とベース座標系が一致する場合にのみ使用できます。 デモンストレーション  速度 50% スーパー管理者 xMateER3 Pro	
4. 右上のデモンストレーションの開始をタップします。	ロボット機能のデモンストレーション  ■ デモンストレーションの一時停止 ■ デモを終了 7軸冗長動作 障害物回避動作 衝突検知 インピーダンス制御 Introduction 物を出し入れるデモです。ボットが狭い深箱の内部に入る必要がある場合、零空間の動作によってロボットアームの姿勢を調整します。深箱に入る過程で筐体構造と干渉せず、物を出し入れることができます。 注記: デモ中は、すべてのロボット設定が無効になります。デモは、ワールド座標系とベース座標系が一致する場合にのみ使用できます。 デモンストレーション  速度 50% スーパー管理者 xMateER3 Pro	
5. デモンストレーションを終了する際は、右上のデモを終了をタップします。	ロボット機能のデモンストレーション  ▶ デモンストレーションの開始 ■ デモを終了 7軸冗長動作 障害物回避動作 衝突検知 インピーダンス制御 Introduction 物を出し入れるデモです。ボットが狭い深箱の内部に入る必要がある場合、零空間の動作によってロボットアームの姿勢を調整します。深箱に入る過程で筐体構造と干渉せず、物を出し入れることができます。 注記: デモ中は、すべてのロボット設定が無効になります。デモは、ワールド座標系とベース座標系が一致する場合にのみ使用できます。 デモンストレーション  速度 50% スーパー管理者 xMateER3 Pro	

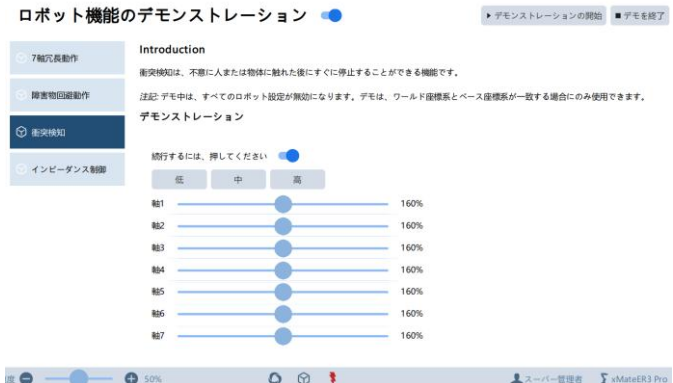
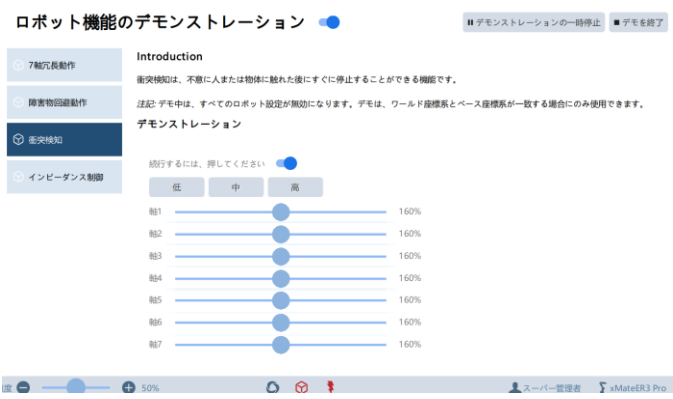
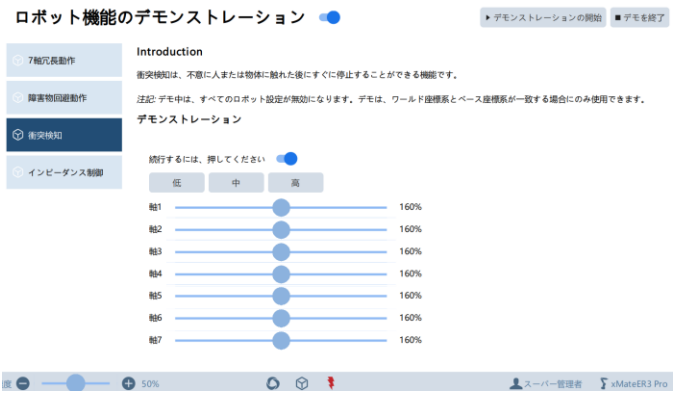
14.7.3 衝突検知

衝突検知機能を実演します。

衝突検知感度設定は、単軸設定をサポートし、高、中、低レベルでの設定もサポートします。ロボットが衝突を検知すると、柔軟に停止します。バージョン 3.0 では、衝突検知で停止したロボットを押すことで復帰する機能はサポートされなくなりました。

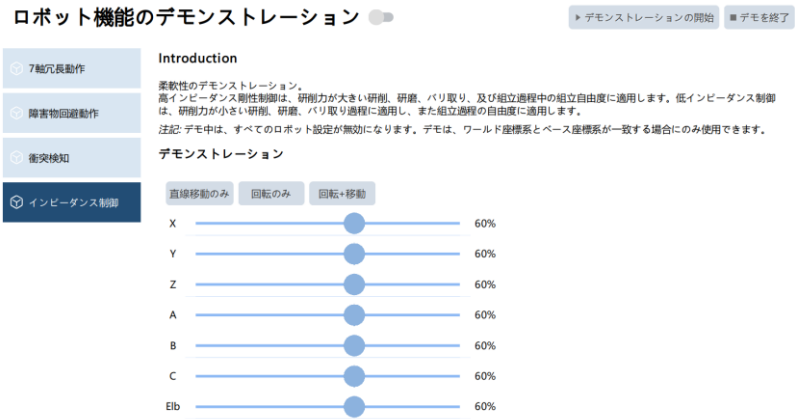


手順	図	説明
1. オプション-デモンストレーション-衝突検知を選択し、デモスイッチをオンにしてデモモードに入ります。		
2. 下部ステータスバーにある操作モードの切替ボタン  をタップして、ロボットを自動モードに切り替えます。		

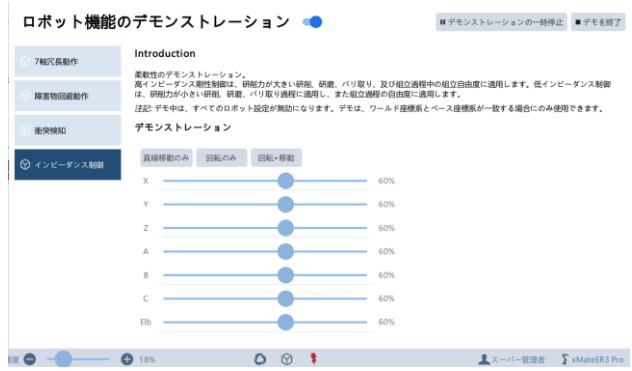
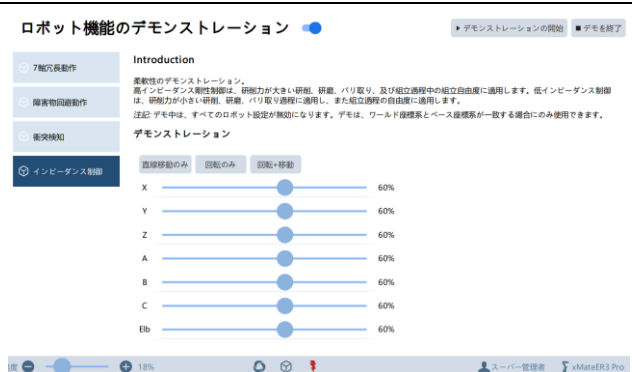
3. 下部ステータスバーのモーター電源状態ボタンをタップしてモーター電源を入れます。		
4. 右上のデモンストレーションの開始をタップします。		
5. デモンストレーションを終了する際は、右上のデモを終了をタップします。		デモ中、衝突検知感度の閾値を調整する必要がある場合は、「デモを終了」をタップし、閾値を調整してから再度デモを開始してください。

14.7.4 インピーダンス制御（柔軟性デモ）

柔軟性デモは、ロボットのインピーダンス制御を実演します。



手順	図	説明
1. オプション-デモンストレーション-インピーダンス制御を選択し、デモスイッチをオンにしてデモモードに入ります。		
2. 下部ステータスバーにある操作モードの切替ボタンをタップして、ロボットを自動モードに切り替えます。		
3. 下部ステータスバーのモーター電源状態をタップしてモーターの電源を入れます。		

4. 右上のデモンストレーションの開始をタップします。		
5. デモンストレーションを終了する際は、右上のデモを終了をタップします。		デモ中、剛性の閾値を調整する必要がある場合は、「デモを終了」ボタンをタップし、閾値を調整してから再度デモを開始してください。



警告

デモ機能の実行中は、すべてのロボット設定が無効になります。

注意：

1. ロボットのベース座標系はデフォルトでワールド座標系と一致します。
2. ロボットはデフォルトで先端が無負荷です。ロボットの先端に負荷がかかると衝突知や柔軟性デモなどに影響を与えます。

15 RL 命令

15.1 変数の型

15.1.1 int

説明	int 型は整数型変数で、値の範囲は-2147483647~2147483647 です。範囲を超えると値がランダムに代入されるため、使用するときは範囲を超えないようにしてください。
例	変数一覧での定義：  初期値が 4 である int 型変数 counter を定義することを示しています。

15.1.2 double

説明	double 型は浮動小数点型変数です。8 バイトで保存するため、使用する場合は取り得る値の範囲を超えないようにしてください。
例	変数一覧での定義：  初期値が 1.5 である double 型変数 time を定義することを示しています。

15.1.3 bool

説明	bool 型は主にステータスや論理の判定に使われます。取り得る値は True または False です。int または double 値が代入されると、0 以外が True で、0 が False となります。
例	変数一覧での定義：  初期値が true である bool 型変数 ifClose を定義することを示しています。

15.1.4 string

説明	string 型は文字列型変数です。複数のアルファベットまたは数字で構成されます。 注意：RL プログラム内では、二重引用符""で囲んで定義します。
例	変数一覧での定義：  初期値が“rokae”である string 型変数 name を定義することを示しています。 string 型の変数は、「+」操作で文字列の接合が可能です。 例： name = “Rok” + “ae” 変数 name の値は「Rokae」になります。

15.1.5 array

説明	array は配列です。配列とは同じ型の変数の集まりのことで、1 次元でも多次元でも定義可能です。配列の各要素は下付き文字を使ってアクセスし、各次元の下付き文字は 1 から始まります。配列の全長は 1000 を超えることはできません。
例	変数一覧での定義：  16 個の int 型変数を含む二次元配列を定義し、配列の 1 行目の 6 番目の要素に値 8 を割り当てることを示しています。

15.1.6 byte

説明	byte 型は符号なしバイトで、C++の unsigned char に相当します。 取り得る値の範囲は 0~255 で、負の値をとることはできません。通常は SocketSendByte 命令に使用されます。 byte 変数の値が 255 を超えると、その値は自動的に切り捨てられ、値の下位 8 ビットのみが保持されます。例えば、var byte data2=288 は切り捨てられた後、32 になります。
例	変数一覧での定義：

	 初期値が 177 である byte 型変数 data を定義することを示しています。
--	--

15.1.7 clock

説明	clock 型は、ストップウォッチと同じように時間を計測するために使用される変数です。clock 型の時間精度は 0.001 秒で、最大時間間隔は 45 日（45×24×3600 秒）です。
例	変数一覧での定義：  次の例は、clock 型変数の使い方を示しています： 例 1： <pre>ClkStart clock1 ClkStop clock1 interval=ClkRead(clock1) ClkReset clock1</pre> interval（事前に宣言された double 変数）は、ClkStart と ClkStop の間の時間間隔を読み取ります（単位：s）。

15.1.8 暗黙の型変換

説明	変数一覧で変数を定義する際、データ型が制限されています。変数の型に準拠していない値を入力することができないため、暗黙の型変換が防止できます。
例	整数型変数 counter を定義する時、入力できるのは整数だけです。小数は入力できません。

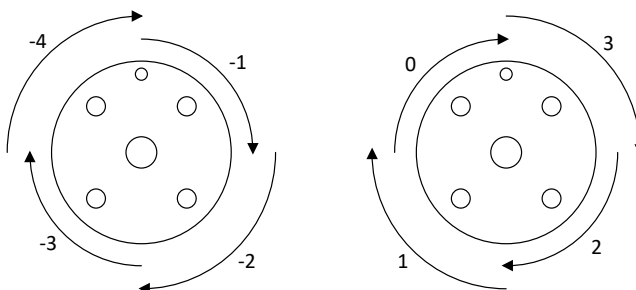
15.1.9 Confdata

説明	Confdata（Robot Configuration Data）は、空間目標点に対応する形態構成データを定義するために使用されます。 ロボットは主に回転関節を使用するため、1°と 361°で同じ状態になります。従って、ロボットの形態を選択した後、関節のマルチターンに対処するための他の手段が必要となります。 産業用ロボットや xMate ER 協働ロボットでは、関節角度の範囲を象限番号でマークしています。例えば、関節角度が 0~90 度の場合は象限番号を 0、90~180 度の場合は象限番号を 1 とし、90 度ごとに 1 ずつ増加または減少します。角度が負の数の時は、対応する象限番号も負の数に
----	---

なります。下図は、左の図は関節角度が負の場合、右の図は関節角度が正の場合を示しています。ロボットの関節の場合、反時計回りの回転で象限番号が大きくなり、時計回りの回転で小さくなります。下図に示すように、ある関節が時計回りに回転すると、象限番号が減少します。対応する関節角度の象限番号の変化は、それぞれ-1→-2→-3→-4 と 3→2→1→0 です。

xMate CR と SR 協働ロボットでは、各関節角度の小数部分を切り捨てた整数値を **Confddata** に記録します。

さらに、同じ直交空間の目標点には複数の異なる解が存在する為、**Confddata** を使用して形態選択を明示的に指定します。



定義

完全な **Confddata** を構成するために、以下のパラメーターが必要です。

cf1、データ型：Int、1 軸角度に対応する象限または小数部分を切り捨てた関節角度です。

cf2、データ型：Int、2 軸角度に対応する象限または小数部分を切り捨てた関節角度です。

cf3、データ型：Int、3 軸角度に対応する象限または小数部分を切り捨てた関節角度です。

cf4、データ型：Int、4 軸角度に対応する象限または小数部分を切り捨てた関節角度です。

cf5、データ型：Int、5 軸角度に対応する象限または小数部分を切り捨てた関節角度です。

cf6、データ型：Int、6 軸角度に対応する象限または小数部分を切り捨てた関節角度です。

cf7、データ型：Int、7 軸角度に対応する象限または小数部分を切り捨てた関節角度です。

cfx、データ型：Int、ロボットが使用する形態構成番号です。整数値で、範囲は 0~8 です。

例

同じ直交姿勢に対しては複数の異なる解が存在し、**cfx** は 0~8 の整数で解を指定します。詳細な説明は以下の通りです。

6 軸産業用ロボット、xMateER 協働ロボットと 7 軸ロボット：

cfx	手首の中心が 1 軸に対して	手首の中心が 下腕に対して	6 軸角度
0	正面	正面	正
1	正面	正面	負
2	正面	後面	正
3	正面	後面	負
4	後面	正面	正
5	後面	正面	負
6	後面	後面	正
7	後面	後面	負

xMateCR 協働ロボット（**cfx**=0、**cf1**~**cf6** で表される関節角度に近い解を示します）：

cfx	手首の中心が 1 軸に対して	手首の中心が 下腕に対して	5 軸角度
0			
1	正面	正面	負

	2	正面	正面	正
	3	正面	後面	負
	4	正面	後面	正
	5	後面	正面	負
	6	後面	正面	正
	7	後面	後面	負
	8	後面	後面	正

xMateSR 協働ロボット（cfx は常に 0 で、cf1~6 で表される関節角度に近い解を出します）

3 軸産業用ロボットと 4 軸産業用ロボット：

cfx	手首の中心が 下腕に対して
0	正面
1	正面

15.1.10 jointtarget

説明	jointtarget 型は、ロボットの軸角度と外部軸位置を保存する変数です。
定義	robax：ロボットの関節角度（Robot Axis） データ型：double 7 つの double 型メンバーを含みます。ロボットの 7 つの関節の角度 degree を示します。 extax：外部軸位置（External Axis） データ型：double 6 つの double 型メンバーを含みます。最大 6 つまでの外部軸の位置情報を示します。外部軸が回転軸の場合の単位は角度 degree で、外部軸が直線軸の場合の単位は mm です。
例	

15.1.11 load

説明	load 型は、ロボット負荷の動力学パラメーターを保存するために使用される変数です。 ロボットの負荷は主に 2 つあります ● ロボットの先端に取り付けられたツールまたはワークそのもの ● ツールがつかむ/吸い上げるもの load 型変数は、単独で作成することをサポートしません。tool 型変数のメンバーとして、ツール定義の際に手動で変更するか、負荷推定機能を使用して制御システムによって自動的に変更されます。 負荷の動力学パラメーターを正しく定義することで、ロボットは最適な性能を得ることができます。誤った定義は以下のような結果を招く可能性があります。 ● ロボットがサーボシステムの能力を最大限に活用できず、性能が低下する ● パス精度が低下し、位置決め誤差が大きくなる ● 機械部品への過負荷で、耐用年数が短くなったり破損を引き起こす
定義	xCore システムでは、負荷は剛体として扱われます。負荷を説明するためのパラメーターは以下の通りです mass：質量（Mass） データ型：double

負荷の質量です。単位は kg です。

cogx : 重心 (Center of Gravity) の x 方向のオフセット

データ型 : double

ハンドツールの場合は、**cogx** はツール座標系における負荷重心の x 方向のオフセットを示します。外部ツール機能が使用されている場合は、ワーク座標系における負荷重心の x 方向のオフセットを示します。

cogy : 重心 (Center of Gravity) の y 方向のオフセット

データ型 : double

ハンドツールの場合は、**cogy** はツール座標系における負荷重心の y 方向のオフセットを示します。外部ツール機能が使用されている場合は、ワーク座標系における負荷重心の y 方向のオフセットを示します。

cogz : 重心 (Center of Gravity) の z 方向のオフセット

データ型 : double

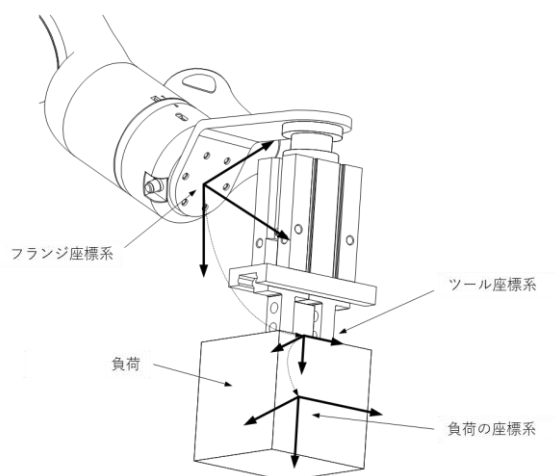
ハンドツールの場合は、**cogz** はツール座標系における負荷重心の z 方向のオフセットを示します。外部ツール機能が使用されている場合は、ワーク座標系における負荷重心の z 方向のオフセットを示します。

q1~q4 : クォータニオン

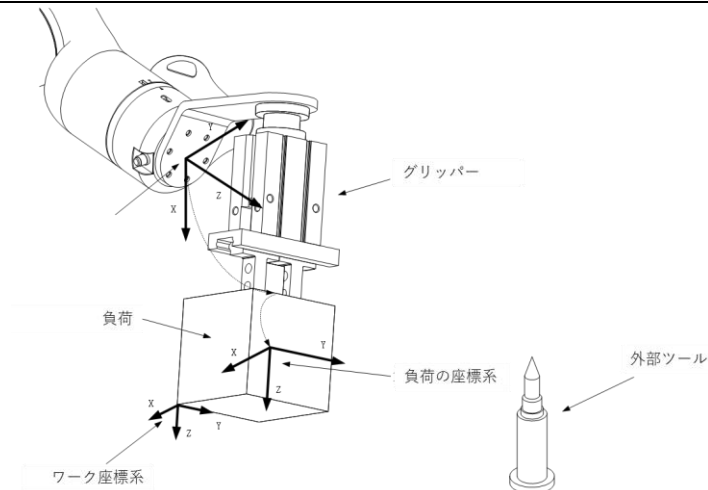
データ型 : double

負荷イナーシャの主軸の方向を示します。

ツールをロボットに取り付けると、イナーシャの主軸の方向はツール座標系で記述されません。下図を参照してください。



外部ツールを使用すると、イナーシャ主軸の方向はワーク座標系で記述されます。下図を参照してください。



lx : イナーシャ x

データ型 : double

x 主軸に沿ったイナーシャを示します。単位は kgm^2 です。

ly : イナーシャ y

データ型 : double

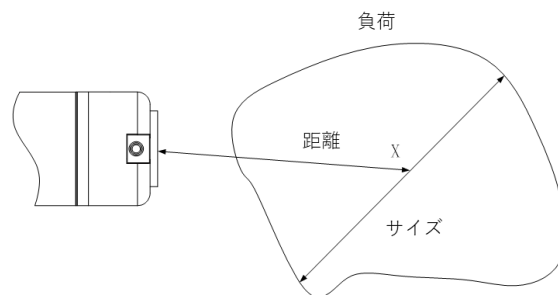
y 主軸に沿ったイナーシャを示します。単位は kgm^2 です。

lz : イナーシャ z

データ型 : double

z 主軸に沿ったイナーシャを示します。単位は kgm^2 です。

注意 : 負荷イナーシャを正しく定義することは、特に大きな物体を扱う時のロボットの動作精度を向上させることができます。lx、ly、lz がすべて 0 に設定される場合、負荷は質点として扱われます。通常、負荷重心からフランジの中心点までの距離が負荷の最大サイズよりも小さい場合は、下図に示すように負荷イナーシャを定義する必要があります。



15.1.12 orient

説明	orient 型は、座標系または空間剛体の姿勢情報を保存する変数です。 orient 型変数は単独で作成したり変更したりすることをサポートせず、特定の変数のメンバー変数としてのみ使用します。
定義	RL 言語はクォータニオンを使用して姿勢を示します。合計 4 つの要素を持ち、その形式は以下の通りです。 q1、データ型 : double、クォータニオンの 1 番目の要素 q2、データ型 : double、クォータニオンの 2 番目の要素 q3、データ型 : double、クォータニオンの 3 番目の要素

	q4、データ型：double、クォータニオンの 4 番目の要素							
備考	通常、回転行列で剛体の姿勢を説明できますが、クォータニオンはより簡潔に姿勢を記述する方法です。							
	クォータニオンの 4 つの内積は、以下の関係を満たします： $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$							
	回転行列とクォータニオンは相互に変換できます。仮に回転行列 R がある場合：							
	$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$							
	<table><tr><td>$q_1 = \frac{\sqrt{r_{11} + r_{22} + r_{33} + 1}}{2}$</td><td>/</td></tr><tr><td>$q_2 = \frac{\sqrt{r_{11} - r_{22} - r_{33} + 1}}{2}$</td><td>$signq_2 = sign(r_{32} - r_{23})$</td></tr><tr><td>$q_3 = \frac{\sqrt{r_{22} - r_{11} - r_{33} + 1}}{2}$</td><td>$signq_3 = sign(r_{13} - r_{31})$</td></tr><tr><td>$q_4 = \frac{\sqrt{r_{33} - r_{11} - r_{22} + 1}}{2}$</td><td>$signq_4 = sign(r_{21} - r_{12})$</td></tr></table>	$q_1 = \frac{\sqrt{r_{11} + r_{22} + r_{33} + 1}}{2}$	/	$q_2 = \frac{\sqrt{r_{11} - r_{22} - r_{33} + 1}}{2}$	$signq_2 = sign(r_{32} - r_{23})$	$q_3 = \frac{\sqrt{r_{22} - r_{11} - r_{33} + 1}}{2}$	$signq_3 = sign(r_{13} - r_{31})$	$q_4 = \frac{\sqrt{r_{33} - r_{11} - r_{22} + 1}}{2}$
$q_1 = \frac{\sqrt{r_{11} + r_{22} + r_{33} + 1}}{2}$	/							
$q_2 = \frac{\sqrt{r_{11} - r_{22} - r_{33} + 1}}{2}$	$signq_2 = sign(r_{32} - r_{23})$							
$q_3 = \frac{\sqrt{r_{22} - r_{11} - r_{33} + 1}}{2}$	$signq_3 = sign(r_{13} - r_{31})$							
$q_4 = \frac{\sqrt{r_{33} - r_{11} - r_{22} + 1}}{2}$	$signq_4 = sign(r_{21} - r_{12})$							

15.1.13 pos

説明	pos 型は、直交空間の位置情報を保存する変数です。 pos 型変数は単独で作成したり変更したりすることをサポートせず、特定の変数のメンバー変数としてのみ使用します。
定義	RL 言語は直交座標系で 3D 空間を説明するため、pos 変数には x、y、z の 3 つの要素があります。 X、データ型：double、位置の x 座標 Y、データ型：double、位置の y 座標 Z、データ型：double、位置の z 座標

15.1.14 pose

説明	pose 型は、直交空間の位置と姿勢を保存する変数です。
定義	X、データ型：double、位置の x 座標 Y、データ型：double、位置の y 座標 Z、データ型：double、位置の z 座標 Q1、データ型：double、クォータニオンの 1 番目の要素 Q2、データ型：double、クォータニオンの 2 番目の要素 Q3、データ型：double、クォータニオンの 3 番目の要素 Q4、データ型：double、クォータニオンの 4 番目の要素

15.1.15 robtarget

説明	robtarget 型は、MoveJ、MoveL、MoveC、MoveT 命令用の 3D 空間における直交位置と姿勢を保存する変数です。 ロボットの逆運動学の解は複数あるため、ロボットは複数の形態で同じ目標姿勢に到達できます。構形成態を明示的に指定するために、robtarget 変数にはロボット構成（Robot Configuration）データも含まれます。 robtarget 型の変数は、プログラミング補助で動作命令を挿入する際に自動的に作成されます。手動で変数内の値を変更すると、pose と ConfData が不一致になり、ロボットが正しく動作命
----	--

	令を実行できなくなる可能性があります。 注意：ロボットプログラムに使用される直交位置と姿勢は、ワーク座標系で定義されます。最終的に使用するワークが最初のプログラミングの時のワークと異なる場合、ロボットの動作が目標パスから逸脱する可能性があります。従って、以下の 2 つの場合は、ワークの変化が危険を及ぼさないことを確認する必要があります。 ● 「命令変更」機能で命令の wobj パラメーターを変更する場合 ● 実際に使用されたワークが、プログラム命令中のワークと異なる場合 誤った使い方は、人身事故や設備の損傷を引き起こす可能性があります！
定義	Trans：空間位置 データ型：pos 基準座標系に対する位置です。 Rot：姿勢 データ型：orient 基準座標系に対する姿勢です。 Conf：ロボットの構成データ（Robot Configuration） データ型：confdata ロボットのコンフィギュレーション構成データです。詳細は confdata 変数を参照してください。 Extax：外部軸位置（External Axis） データ型：double extax は 6 つの double 型メンバーを含みます。最大 6 つまでの外部軸の位置情報を示します。外部軸が回転軸の場合の単位は角度 degree で、外部軸が直線軸の場合の単位は mm です。
例	

15.1.16 signalxx

説明

signalxx 型は、入力信号と出力信号を制御するために使用する変数です。

すべての signalxx 型の変数は「IO 信号一覧」で定義してからプログラムで使用する必要があります。プログラム内での直接宣言はサポートしていません。

signalxx は現在、以下のデジタル入力・出力をサポートしています。

変数型	対象	説明
signalDI	デジタル入力信号	値は True または False です。信号の状態を表示します。
signalDO	デジタル出力信号	値は True または False です。出力に値を割り当てます。
signalGI	デジタルグループ 入力信号	連続する物理入力ポートを 2 進数で定義します。RL では 10 進数に変換して使用できます。1 グループは最大 16 個の DI をサポートするため、signalGI の取り得る値の範囲は $0 \sim 2^{n-1}$ です。ここで、n はグループ入力に含まれる DI の数です。
signalGO	デジタルグループ 出力信号	連続する物理出力ポートを 2 進数で定義します。RL では 10 進数に変換して使用できます。1 グループは最大 16 個の DO をサポートするため、signalGO の取り得る値の範囲は $0 \sim 2^{n-1}$ です。ここで、n はグループ出力に含まれる DO の数です。

signalDO と signalGO 型は信号の参照のみが可能です。別の命令（SetDO、SetGO など）を使って値を割り当てて使用します。

	signal_{di} と signal_{gi} は、対応する入力信号の値をプログラム内で取得するために使用します。 注意： プログラム内で signal_{xx} 型の変数を定義/宣言することはサポートされていません。このような使い方をした場合、プログラムはエラーを報告します。signal_{xx} 型の変数は、使用する前にまず IO 一覧で設定しておきます。 signal_{xx} 型変数のスコープは System です。他のスコープ型との関係は System > GLOBAL > LOCAL です。 IO 設定インターフェースの signal が RL プログラムで宣言された変数と同名である場合、より低いスコープレベルの変数が選択されます。
例	例 1 <pre>//デジタル入力の状態を判定条件とする。 IF (di1 == true) do something... ENDIF</pre> 例 2 //デジタルグループ入力の状態を判定条件とする。例えば、グループ入力 gi2 は Profinet IO の最初のバイトの 3 つの bit をマッピングするように定義すると、bit0~bit2 の値がそれぞれ 0,1,1 のとき、gi2 の値は 110 (int 型に変換すると 6) となる。グループ出力 (signal_{go}) も同様です。 <pre>IF (gi2 == 6) do something... endif</pre>

15.1.17 speed

説明	speed 型は、ロボットと外部軸の動作速度を定義するために使用される変数です。 システムには一般的に使用される速度変数がプリセットされており、プログラミング補助によって選択して使用できます。詳細は命令の挿入を参照してください。
定義	speed 型変数には、関節速度パーセンテージ、TCP 直線速度、軸角速度、外部軸直線速度、外部軸角速度の 5 つのメンバー変数が含まれます。 軸速度パーセンテージ (Joint Velocity Percentage) データ型: double 軸動作命令中の動作速度を指定するために使用されます。MoveAbsJ および MoveJ 命令に適用されます。取り得る値の範囲は 1%~100% です。 TCP 直線速度 (Linear Velocity) データ型: double ツール中心点の直線速度です。取り得る値の範囲は 0.001mm/s~ 7000mm/s です。 軸角速度 (Orientation Velocity) データ型: double ツールの回転速度です。取り得る値の範囲は 0.001°/s~ 500°/s です。 外部軸直線速度 データ型: double 外部軸の直線動作速度です。取り得る値の範囲は 0mm/s~ 5000mm/s です。 外部軸角速度 データ型: double 外部回転軸の角速度です。取り得る値の範囲は 0°/s~ 1000°/s です。
例	変数一覧での定義：

変数型
 変数情報 速度 軸速度/パーセント: 1~100%、TCP直線速度: 0.001~7000 m/s、空間回転速度: 0.001~500 °/s、外部軸速度: 0~2000 m/s もしくは 0~300 °/s。

変数型: speed

名前: speed0 説明:

PERS: ☐ YES ☒ NO 次元: + - 非配列

初期値設定: speed0

軸速度の割合 (%) 40

TCP直線速度(mm/s) 300

空間回転速度(°/s) 100

外部軸直線速度(mm/s) 200

外部軸角速度(°/s) 1000

speed0 と呼ばれる speed 変数を定義します。その軸速度は最大許容速度の 40%、TCP 直線速度は 300mm/s、軸角速度は 100°/s、外部軸角速度は 1000°/s、外部軸直線速度は 200mm/s です。

一般的な速度変数は、システムに事前定義されています。詳細は次の表に示す通りです。

名前	軸速度のパーセンテージ	TCP 直線速度	軸角速度	外部軸角速度	外部軸直線速度
v5	1%	5 mm/s	200°/s	1000°/s	5000 mm/s
v10	3%	10 mm/s	200°/s	1000°/s	5000 mm/s
v25	5%	25 mm/s	200°/s	1000°/s	5000 mm/s
v30	5%	30 mm/s	200°/s	1000°/s	5000 mm/s
v40	5%	40 mm/s	200°/s	1000°/s	5000 mm/s
v50	8%	50 mm/s	200°/s	1000°/s	5000 mm/s
v60	8%	60 mm/s	200°/s	1000°/s	5000 mm/s
v80	8%	80 mm/s	200°/s	1000°/s	5000 mm/s
v100	10%	100 mm/s	200°/s	1000°/s	5000 mm/s
v150	15%	150 mm/s	200°/s	1000°/s	5000 mm/s
v200	20%	200 mm/s	200°/s	1000°/s	5000 mm/s
v300	30%	300 mm/s	200°/s	1000°/s	5000 mm/s
v400	40%	400 mm/s	200°/s	1000°/s	5000 mm/s
v500	50%	500 mm/s	200°/s	1000°/s	5000 mm/s
v600	60%	600 mm/s	200°/s	1000°/s	5000 mm/s
v800	70%	800 mm/s	200°/s	1000°/s	5000 mm/s
v1000	100%	1000 mm/s	200°/s	1000°/s	5000 mm/s
v1500	100%	1500 mm/s	200°/s	1000°/s	5000 mm/s
v2000	100%	2000 mm/s	200°/s	1000°/s	5000 mm/s
V3000	100%	3000 mm/s	200°/s	1000°/s	5000 mm/s
v4000	100%	4000 mm/s	200°/s	1000°/s	5000 mm/s
v5000	100%	5000 mm/s	200°/s	1000°/s	5000 mm/s
v6000	100%	6000 mm/s	200°/s	1000°/s	5000 mm/s
v7000	100%	7000 mm/s	200°/s	1000°/s	5000 mm/s
vmax	100%	infinite	200°/s	1000°/s	5000 mm/s

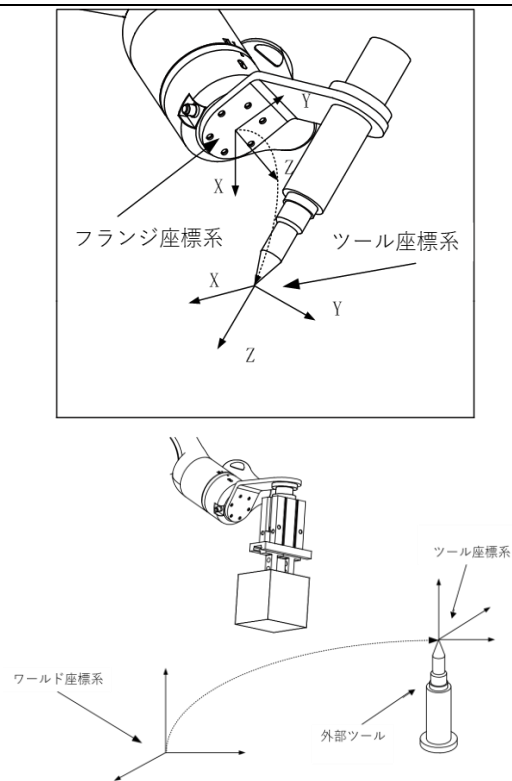


ヒント

システム事前定義の **speed** 変数は、すべての軸角速度が $200^{\circ}/s$ です、ロボット先端の回転速度に特別な要件がある場合は、要件に応じて新しい **speed** 変数を定義して使用する必要があります。

15.1.18 tool

説明	tool 型は、ロボットが使用するツールの TCP、姿勢および動力学パラメーターを含むツールパラメーターを記録する変数です。 ロボットはツールを使って外部環境と相互作用します。tool 変数は以下のようにロボットの動作に影響を与えます。 TCP のみがプログラミングのパスと速度に従って動作します。ロボットが空間で純粋な回転動作を行う場合は、TCP が静止したままになります。 プログラミングのパスと速度はツール座標系のワーク座標系に対するパスと速度を指します。正確に定義されたツールやワークを交換した場合は、パスの形状や速度の大きさには影響しません。 外部ツールを使う際、プログラムの速度はワークの速度（外部ツールに対して）を指します。外部ツールを使う際、tool 変数中の tframe は、外部ツールのゼロ点位置と姿勢オフセット量を定義し、tload はロボット先端に取り付けられたグリッパの動力学パラメーターを定義するために使用されます。 tool 型変数のデータはデータベースに保存されます。プログラムをロードする際、プログラムエディターでデータベースから読み出されるため、予測できないエラーを避けるためにプログラムエディターで tool 型変数を直接変更しないでください。tool 型変数を変更する必要がある場合は、定義インターフェースを通じて行ってください。詳細は、ツール座標系の定義を参照してください。 ロボットの動力学パラメーターを正しく定義してください（ツール本体及びツールに掴まれるものを含む）。誤った定義は以下のような結果を招く可能性があります。 ● ロボットがサーボシステムの能力を最大限に活用できず、性能が低下する ● パス精度が低下し、位置決め誤差が大きくなる ● 機械部品の過負荷が、耐用年数が短くなることや破損を引き起こす
例	Robhold データ型：bool ツールがロボットに取り付けられるかどうかを示します。True はツールがロボットに取り付けられていることを示し、False はツールがロボットに取り付けられず、外部ツールであることを示します。ジョグまたはプログラム実行中に同時にツールとワークの組み合わせが使用される場合、robhold パラメーターは1つのみが True になります。即ち、ツールの robhold が True の場合、対応するワークの robhold は False でなければならず、その逆も同様です。さもなくば、ロボットはエラーメッセージを出し、ジョグや対応するプログラムを実行できなくなります。 Tframe：ツール座標系（Tool Frame） データ型：pose 使用されるツールのツール座標系を示します。 TCP はロボット先端フランジ座標系に対する x、y、z 方向のオフセット量を示し、単位は mm です。ツール座標系のフランジ座標系に対する姿勢オフセット量は、クォータニオンで表現します。下図を参照してください。



Tload : ツールの動力学パラメーター

データ型 : load

ツールの動力学パラメーターを示します。ハンドツールの場合は、**Tload** はツール全体の動力学パラメーターを定義します。外部ツールの場合は、ロボットに取り付けられたグリッパーの動力学パラメーターを定義します。

ハンドツールの負荷パラメーターには以下のものが含まれます。

ツールの質量 (重量) : 単位は **kg** ;

ツールの重心 : フランジ座標系に対して定義されます。単位は **mm** ;

イナーシャ主軸の方向 : フランジ座標系に対して定義されます ;

ツールのイナーシャ主軸に沿ったイナーシャの大きさです。単位は **kgm²** です。すべてのイナーシャの要素を **0** に定義した場合、当該ツールは質点 (Point Mass) として扱われます。

注意 :

外部ツール機能を使う際は、ツールの **TCP** と姿勢はワールド座標系に対して定義されます。ロボットが外部ツールを使用している場合、**tload** メンバーはロボットに取り付けられたグリッパーの動力学パラメーターを定義するために使用されます。具体的なパラメーターの意味は同じです。

tload メンバーは、グリッパーの動力学パラメーターのみを定義するために使用され、掴まれるワークの動力学パラメーターは含みません。ロボットがいかなる状況でも最適な性能を得られるように、2つの **tool** 変数を定義し、切り替えることで処理する必要があります。

- ・一つは、グリッパ自体のパラメーターを定義する **tool** 変数 ;
- ・もう一つは、グリッパ+掴まれるワークのパラメーターを定義する **tool** 変数 ;

負荷の有無を切り替えるには、動作命令中にツール変数を切り替えることで実現できます。

15.1.19 trigdata

説明	trigdata 型は、トリガー条件やトリガー動作などロボットの動作中のトリガーイベントに関するデータを保存するために使用される変数です。 トリガー条件とは、通常はロボットがパス上の指定されたポイントに到着することです。 トリガー動作とは、IO 設定、変数設定などを行うことです。 trigdata 型変数は代入演算子ではなく特別な RL 命令を通じて定義するため、各 trigdata 型変数に保存される情報の種類は TrigIO などの Trig 命令に依存します。 最終的にこのデータは、動作命令 TrigL、TrigC、TrigJ のパラメーターとして使用されます。
例	次の例は、trigdata の使い方です： 例 1 <pre>VAR trigdata gripopen TrigIO gripopen,0.5,do1,true TrigL p1,v500,gripopen,fine,tool1</pre>

15.1.20 wobj

説明	wobj はワークオブジェクト（Work Object）の略で、ワークとはロボットに加工、処理、搬送されるものを指します。 すべての動作命令で使用される位置は、ワーク座標系で定義されます（ワーク座標系が指定されていない場合は、デフォルトの座標系 wobj0 は、ワールド座標系で定義されます。 wobj にはいくつかの利点があります。 ● 多くの加工点の位置をワークの図面から直接取得して使用できます。 ● ロボットの再設置や取り扱うワークの位置が変更される場合、ワーク座標系を再定義するだけで、以前のプログラムを再利用でき再プログラミングが不要になります。 ● 適切なセンサーが搭載されている場合、ワークの振動やわずかなズレは自動的に補正されます。 ワーク座標系が定義されていないと、制御システムはワールド座標系をデフォルトのワーク座標系 wobj0 とみなします。外部ツールを使う場合は、ワーク座標系を定義しなければなりません。この時にプログラムされるパスと速度は、ツールではなくワークのパスと速度を意味します。 通常、ワーク座標系はユーザー座標系に対して定義されますが、ユーザー座標系を指定しない場合は、ワーク座標系はデフォルトでワールド座標系に対して定義されます。詳細は、ロボット座標系システムを参照してください。 ワークは、ユーザー座標系とワーク座標系の 2 つの座標系で構成されます。複数の同一ワークを処理する場合に対応するために、ユーザー座標系がワーク座標系の上位層にあります。座標系の定義関係については、「定義」部分の oframe の説明を参照してください。
定義	Robhold ワークがロボットに取り付けられているかどうかを示します。 True は、ワークがロボットに取り付けられ、外部ツールが使用されていることを示します。 False は、ワークがロボットに取り付けられておらず、ハンドツールが使用されていることを示します。 Ufprog：ユーザー座標系の移動（User Frame Programmed） 変数型：bool ユーザー座標系が固定か移動かを示します。 True はユーザー座標系が固定であることを示し、False はユーザー座標系が移動であることを示します。 例：外部ポジションまたはその他のロボットでの定義

この値は主に、ロボットとポジションまたはその他のロボットが動作を協調させる必要がある場合に使用されます。

Ufmec : ユーザー座標系が関連付けられたメカユニット (User Frame Mechanical Unit)

データ型 : **string**

メカユニットの名前を使用して、ユーザー座標系がどのロボットに紐づけられるかを指定します。**ufprog** が **False** の時のみ有効になります。

Oframe : ワーク座標系 (Work Object Frame)

データ型 : **pose**

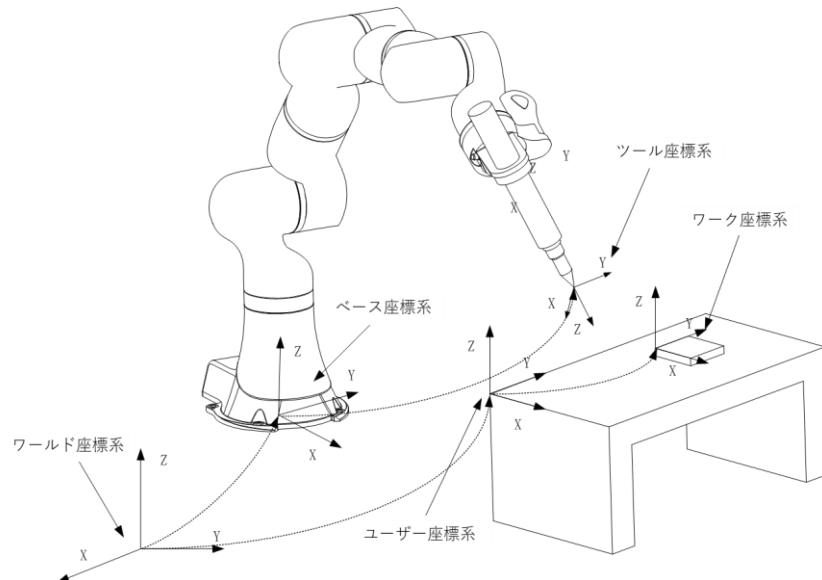
ワーク座標系の原点と姿勢を示します。

uframe_id : ユーザー座標系 (User Frame) ID

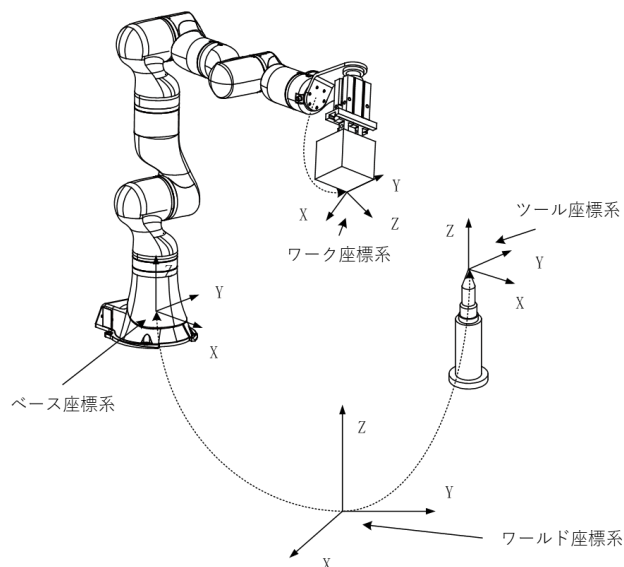
データ型 : **int**

ユーザー座標系の ID を示します。ID を通じて対応するユーザー座標系を見つけることができます。

ハンドツール (外部ツールではない) を使用する場合は、ワーク座標系がユーザー座標系に対して定義され、ユーザー座標系はワールド座標系に対して定義されます。



外部ツールを使用する場合は、ワーク座標系がユーザー座標系に対して定義され、ユーザー座標系はフランジ座標系に対して定義されます。



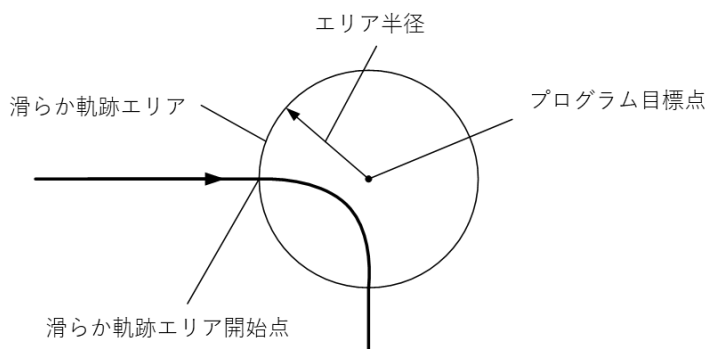
15.1.21 zone

zone 型は、特定の動作がどのように終了するかを定義するか、または 2 つの軌跡の間の滑らか軌跡エリアのサイズを定義する変数です。

同じロボットの目標点に対して、動作命令では 2 つの扱い方があります。

- 停止点として扱う場合は、ロボットは目標点まで動作します。目標点に到達した時の速度は 0 で、そして次の命令を実行します。
- 経由点として扱う場合は、ロボットは目標点までは動作しません。その目標点の何 mm か離れたところから次の目標点に向かって動作し、パスはプログラミングパスから逸脱します。2 つの軌跡の間の移行エリアを滑らか軌跡エリアと呼びます。下図を参照してください。

説明

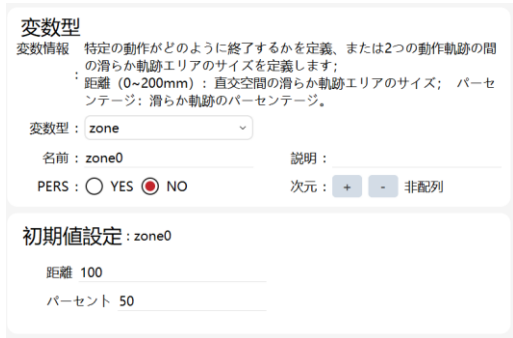


定義された滑らか軌跡エリアの大きさは、パス長の半分以上を超えてはいけません。超えた場合、システムは自動的に滑らか軌跡エリアをパス長の半分に変更します。滑らか軌跡エリアを使用することで、ロボットの頻繁な開始/停止を避け、タクトタイムを大幅に短縮できます。

注意：

滑らか軌跡は、特定の状況下でキャンセルされ、システムは「滑らか軌跡キャンセル Corner Path Failed」というログを報告します。考えられる原因は以下の通りです。

- 前後 2 つの軌跡のうち、少なくとも 1 つの軌道の長さが小さすぎる (1mm/0.001rad)
- 前後 2 つの軌跡がほぼ平行で、逆方向に運動する

	● 前後 2 つの軌跡の純粋回転運動が逆方向であり、前の軌跡では先端軸正回転のみで、後の軌跡では先端軸負回転のみである。 ● 「滑らか軌跡キャンセル」という警告が発生した場合、プログラムは自動的に、影響を受けた命令の目標点を停止点として扱います。 ● 上記の特殊な状況に加えて、すべての論理命令も、その前の動作命令の滑らか軌跡をキャンセルします。
定義	軸動作と直交動作は異なるパラメーターを使用して滑らか軌跡を定義するため、当該変数には Distance と Percent の 2 つの部分があります。 Distance：直交動作での距離 データ型：double MoveL、MoveC、および MoveT 命令で使用され、直交動作の滑らか軌跡エリアの大きさを定義します。ロボットが目標点より distance 離れたところまで動作すると、次の目標点に向かって動作を開始します。単位は mm で、取り得る値の範囲は 0 ~ 200 mm です。 Percent：軸動作でのパーセンテージ データ型：double MoveJ と MoveAbsJ で使用され、回転を開始する目標角度からの距離を示します。100% は回転角度値の半分を示します。純粋な回転動作のみを行う MoveL 命令では、Distance ではなく Percent パラメーターが使用されます。
例	変数一覧での定義：  The screenshot shows a variable definition window for 'zone'. It includes a description in Japanese, a dropdown menu set to 'zone', a name field 'zone0', a 'PERS' section with 'YES' and 'NO' radio buttons (where 'NO' is selected), a '次元' (Dimension) section with '+' and '-' buttons and '非配列' (Not array), an '初期値設定' (Initial value setting) section with 'zone0', and input fields for '距離' (Distance) set to 100 and 'パーセント' (Percent) set to 50. zone 変数を定義し、直交動作の滑らか軌跡エリアの大きさを 100mm、軸動作の滑らか軌跡エリアの大きさを 50% に定義します。

一般的な滑らか軌跡エリア変数は、システムに事前定義されています。詳細は次の表に示す通りです。

名前	直交動作の滑らか軌跡 エリアのサイズ	軸動作の滑らか軌跡 のパーセンテージ
fine	0 mm	0%
z0	0.3 mm	0.15%
z1	1 mm	1%
z5	5 mm	3%
z10	10 mm	5%
z15	15 mm	8%
z20	20 mm	10%
z30	30 mm	15%
z40	40 mm	20%

z50	50 mm	25%
z60	60 mm	30%
z80	80 mm	40%
z100	100 mm	50%
z150	150 mm	75%
z200	200 mm	100%

15.1.22 torqueinfo

説明	torqueinfo 型は、ロボットに加わる力とトルク情報を説明するために使用される変数です。軸方向のトルク情報と直交方向のトルク情報の 2 つの部分で構成されます。
定義	joint_torque、データ型：軸方向のトルク情報； cart_torque、データ型：直交方向のトルク情報； joint_torque.measure_torque、 データ型：double 型配列 軸方向の測定力情報で、力覚センサーによって測定された各軸にかかるトルクです。 joint_torque.external_torque データ型：double 型配列 軸方向の外部力情報で、コントローラーがロボットモデルと測定力に基づいて計算した各軸にかかるトルク情報です。 cart_torque.m_force データ型：double 型配列 直交空間（xyz）各方向に加えられる力の大きさです。 cart_torque.m_torque データ型：double 配列 直交空間（xyz）各方向に加えられるトルクの大きさです。
例	次の例は、変数型 torqueinfo の使い方を示します： 例 1 <pre>TorqueInfo tmp_info = GetEndtoolTorque(tool1, wobj1) //tool1 wobj1 条件におけるロボット先端ツールのトルク情報構造体を取得する ... print(tmp_info.joint_torque.measure_torque) print(tmp_info.joint_torque.external_torque) //各軸の測定力と外部力を表示する ... print(tmp_info.cart_torque.m_torque) //直交方向トルクを表示する ... print(tmp_info.cart_torque.m_force[1]) print(tmp_info.cart_torque.m_torque[1]) //X 方向の力とトルク情報を表示する</pre>

15.1.23 SocketServer

説明	SocketServer 型は、周辺機器をクライアントとして開始される接続を待ち受けるために、コントローラー上に TCP サーバーを確立するための変数です。当該サーバーは接続要求を待ち受けるためだけに使用されます。複数の接続をサポートし、接続が確立されると通信のために新しい SocketConn が生成されます。 注意： システムリソースの適用と解放は時間がかかるため、サーバーリソースの作成（OpenDev）と破棄（CloseDev）は頻繁に行わないでください。システムリソースの
----	--

	過負荷による故障を避けるために、少なくとも 500ms の時間間隔を保持することを推奨します。 ● このステートメントは、1 つのサーバーリソースを作成するだけであり、サーバーの作成は完了しません。サーバーは、OpenDev と SocketAccept を通じて待ち受け状態に入ります。 ● 1 対多接続をサポートします。
定義	Ip データ型：string 制御システムは IP アドレスでネットワークインターフェースカード（NIC）をマッチングし、対応する NIC を使用してネットワーク待ち受けを行います。パラメーターを「0.0.0.0」に設定すると、すべての NIC をリッスンします。通常は「0.0.0.0」に設定します。 Port データ型：int 待ち受けるポート番号を指定します。外部クライアントが接続を開始する場合、サーバーポートをこの値にする必要があります。 Name データ型：string RL プロジェクトで使用する、このサーバーの識別名です。プロジェクト内で唯一であり、複数のタスク間で共用できます。名前の衝突は許可されません。
例	例 1 SocketServer ss = {"192.168.0.160", 8090, "svr"} //ip 設定が 192.168.0.160 の NIC のみを待ち受ける SocketConn conn = SocketAccept("svr") 例 2 SocketServer ss = {"0.0.0.0", 8090, "svr"} //すべての NIC を待ち受ける SocketConn conn = SocketAccept("svr")

15.1.24 SocketConn

説明	SocketConn 型は、Socket TCP 接続で周辺機器との通信に使用される変数です。2 種類に分けられます。 ロボットをクライアントとする場合、このオブジェクトを通じて周辺機器 TCP サーバーとの接続と通信を開始します。 ロボットをサーバーとする場合、周辺機器の TCP クライアントが接続を開始した後で発生する通信接続を制御します。異なる外部デバイスから複数の TCP クライアント接続が開始されると、クライアント接続ごとに 1 つの接続が生成されます。
定義	Ip データ型：string ロボットがクライアントとして使用される場合は、当該パラメーターは周辺機器サーバーの IP アドレスを指定します。ロボットがサーバーとして使用される場合は、周辺機器からの接続が確立されると、当該パラメーターは周辺機器クライアントの IP アドレスを示します。 Port データ型：int ポート番号です。ロボットは接続を開始する際に周辺機器サーバーのポート番号を指定する必要があります。 Name データ型：string RL プロジェクトで使用するこの接続の識別名です。プロジェクト内で唯一であり、複数のタ

	スク間で共用できます。接続間及び接続とサーバーの間には名前の衝突を起こすことは許可されません。 Cache データ型：int キャッシュサイズです。受信データの最大キャッシュ数を指定します。デフォルト値は 1 です。 Suffix データ型：string サフィックス（接尾辞）を指定します。デフォルト値は「¥r」です。 Attr データ型：String 接続プロパティを示します。 ● "incoming"：ロボットはサーバーで、周辺機器がクライアントで接続しています。ip、port はクライアント情報を示します。 ● "outgoing"：ロボットはクライアントで、外部サーバーに接続しています。ip、port はサーバーに接続するための情報を示します。 ● "その他"：接続が利用できません。接続がオープンされていないか、または未確立の接続が見つかったことを示します。 State データ型：string 当該通信の接続状態を示します。 Closed：接続が閉じています establish：接続が確立され、通常使用されています 注意： ● クライアントとして使用する場合は、ip と port 情報はプログラムで設定する必要があります。サーバーとして使用する場合は、ip と port は accept 命令から自動的に取得します。エラーを避けるため、接続確立後に ip と port の二つの値を変更しないことを推奨します。 ● suffix はいつでもリセットでき、次の読み取りの前に有効になります。通信データのエラーを引き起こさないように、リセットする際は注意してください。suffix は通信が行われる前に設定する必要があり、通信確立後は変更しないでください。
例	例 1 //ロボットがクライアントの場合 //サーバーの IP アドレスは"192.168.0.202"で、ポート番号は 8090、接続の名称は"clt"、キャッシュはデフォルトの 1、サフィックスはデフォルトの"¥r" SocketConn scnn1 = {"192.168.0.202", 8090, "clt"} 例 2 //ロボットがクライアントの場合 //サーバーの IP アドレスは"192.168.0.203"で、ポート番号は 8091、接続の名称は"clt1"、キャッシュは 2、サフィックスはデフォルトの"¥r" SocketConn scnn2 = {"192.168.0.203", 8091, "clt1", 2} 例 3 //ロボットがクライアントの場合 //サーバーip は"192.168.0.204"で、ポート番号は 8092、接続の名称は"clt2"、キャッシュは 2、サフィックスは"¥n"

	<pre>SocketConn scnn3 = {"192.168.0.204", 8092, "clt2", 2, "\n"}</pre> 例 4 //ロボットがサーバーの場合。 //事前に定義した SocketServer 型変数を使用して待ち受けを開始します。クライアントからの接続があると、クライアント IP アドレスなどの情報が格納されます。 <pre>SocketConn conn = SocketAccept("svr1") Print(conn.ip) // 周辺機器の ip アドレス Print(conn.port) // 周辺機器がこの接続を生成するために使用するポート Print(conn.cache) // 受信メッセージのキャッシュ Print(conn.suffix) // 受信・送信のサフィックス</pre>
--	---

15.1.25 FCBoxVol

説明	FCBoxVol 型は、空間に直方体を定義し、力制御が有効になった後の位置監視または終了条件として使用します。
定義	Xmax : 直方体の境界の x 正方向の座標値 Xmin : 直方体の境界の x 負方向の座標値 Ymax : 直方体の境界の y 正方向の座標値 Ymin : 直方体の境界の y 負方向の座標値 Zmax : 直方体の境界の z 正方向の座標値 Zmin : 直方体の境界の z 負方向の座標値
例	変数型 変数情報 : 後続の監視または終了条件用の空間球を定義します。(xc,yc,zc) (m) m) : 球の中心の座標値; radius (mm) : 球の半径。 変数型 : fcboxvol 名前 : fcboxvol0 説明 : PERS : ☐ YES ☒ NO 次元 : + - 非配列 初期値設定 : fcboxvol0 Xmin: 0 Xmax: 0 Ymin: 0 Ymax: 0 Zmin: 0 Zmax: 0

15.1.26 FCSphereVol

説明	FCSphereVol 型は、空間に球体を定義し、力制御が有効になった後の位置監視または終了条件として使用します。
定義	Xc : 球体の中心座標の x 方向の座標値 Yc : 球体の中心座標の y 方向の座標値 Zc : 球体の中心座標の z 方向の座標値 Radius : 空間球の半径

例	変数型 変数情報 : 後続の監視または終了条件用の空間球を定義します。(xc,yc,zc) (m) : 球の中心の座標値; radius (mm) : 球の半径。 変数型 : fcspherevol 名前 : fcspherevol0 説明 : PERS : ☐ YES ☒ NO 次元 : + - 非配列 初期値設定 : fcspherevol0 xc 0 yc 0 zc 0 radius 0
---	---

15.1.27 FCXYZNum

説明	FCXYZNum 型は、力またはトルクデータの変化を示す変数です。
定義	X : x 方向の力またはトルクの変化 Y : y 方向の力またはトルクの変化 Z : z 方向の力またはトルクの変化
例	変数型 変数情報 : XYZの3方向の、力またはトルクの変化の振幅と周期。X (Y/Z) : X (Y/Z) 方向の振幅または周期。 変数型 : fcxyznum 名前 : fcxyznum0 説明 : PERS : ☐ YES ☒ NO 次元 : + - 非配列 初期値設定 : fcxyznum0 X 0 Y 0 Z 0

15.1.28 FCCartNum

説明	FCCartNum 型は、XYZABC の 6 自由度方向の特定のパラメーターを示す変数です。
定義	Num1~3 : XYZ 方向のパラメーター Num4~6 : ABC 方向のパラメーター
例	変数型 変数情報 : FCSetSDPara指令で、空間XYZABCの6自由度方向に同じパラメーターを指定するために使用されます; num1~3はXYZ方向のパラメーター、num4~6はABC方向のパラメーター。 変数型 : fccartnum 名前 : fccartnum0 説明 : PERS : ☐ YES ☒ NO 次元 : + - 非配列 初期値設定 : fccartnum0 num1 0 num2 0 num3 0 num4 0 num5 0 num6 0

15.2 基礎変数と構造体

RL 命令がサポートする変数型のうち、`int`、`double`、`bool`、`string` は分割することができないため、すべての変数型の基礎である基礎変数（基本変数ともいう）に属します。これらを一定の規則に従って組み合わせた変数型は、構造体と呼ばれます。

15.2.1 構造体の構成

構造体の組み合わせルールは、物理的な意味を持つデータを抽象化することに基づいています。

例：

- 構造体 `pos` は、3 つの `double` を組み合わせて 3D 空間の位置 (`xyz`) を表します。
- 構造体 `orient` は、4 つの `double` を組み合わせて回転角度を説明するクォータニオンを表します。
- 構造体 `pose` は、位置 (`pos`) と回転角度 (`orient`) を組み合わせてロボットの位置/姿勢を表します。

15.2.2 構造体の使い方

構造体は、命令パラメーターとして使われるだけでなく、利用シナリオに応じてより精密な操作を行います。構造体のデータは、特定の RL 命令によって直接変更できます。

例 1：

`robtarg` 構造体は、空間位置 (`pos`)、姿勢 (`orient`)、設定データ (`confdata`)、外部軸 (`double` 配列) で構成されています。それぞれの名前は、それぞれ `trans(pos)`、`rot(orient)`、`conf(confdata)`、`extax(double)` です。名前を通じて RL 関数内で構造体のメンバーに直接アクセスできるようになっています。

```
robtarg rob1 = ... // 変数一覧またはユーザー定義の robtarget 変数
rob1.trans.x += 20 // ポイントの x データに 20 を加える
// trans(pos)構造体は、xyz の 3 つの変数で構成される
// rob1 の x 座標は rob1.trans.x でアクセスできる
print(rob1.trans) //位置データのみを表示する
```

例 2：

`wobj` ワーク座標系は、以下の操作が行えます。

```
//デフォルトワーク wobj0 を例として
wobj0.robhold // ハンドワークかどうか、bool 型
wobj0.ufprog // ワーク座標系のユーザー座標系が移動するかどうか、bool 型、使用頻度は低い
wobj0.ufmec // ワーク座標系のユーザー座標系に関係づけられた機械ユニット、string 型、電気めっきラインやトラッキングによく使われる
wobj0.iframe // ワーク座標系のポーズ pose
wobj0.iframe.x // ワーク座標系のポーズ pose の x 座標
wobj0.iframe.y // ワーク座標系のポーズ pose の y 座標
wobj0.iframe.z // ワーク座標系のポーズ pose の z 座標
wobj0.q1 // ワーク座標系のポーズ pose のクォータニオン
wobj0.iframe_id // ワークに関連するユーザー座標系の ID
```

他の複雑な構造体も、同様の方法でアクセスできます。

15.3 関数

15.3.1 関数の定義

15.3.1.1 PROC

PROC は戻り値のない関数です。以下のように定義されます。

```
SCOPE PROC RoutineName()
```

```
...
```

```
...
```

```
//do something
```

```
...
```

```
...
```

```
ENDPROC
```

ここで

1. SCOPE は関数のスコープです。GLOBAL と LOCAL をサポートしています。
2. PROC は戻り値のない関数を定義するキーワードです。
3. RoutineName は関数名で、命名規則は変数の命名規則と同じです。詳細については変数の命名規則を参照してください。

プログラミング補助では、以下の方法で PROC を挿入することができます。



15.3.1.2 FUNC

FUNC は戻り値のある関数です。以下のように定義されます。

```
SCOPE FUNC RET RoutineName()
```

```
...
```

```
...
```

```
//do something
```

```
...
```

```
...
```

```
ENDFUNC
```

ここで

1. SCOPE は関数のスコープです。GLOBAL と LOCAL をサポートしています。
2. FUNC は戻り値のある関数を定義するキーワードです。
3. RET は戻り値の型です。
4. RoutineName は関数名で、命名規則は変数の命名規則と同じです。詳細については変数の命名規則を参照してください。

プログラミング補助では、以下の方法で **FUNC** を挿入することができます。



15.3.2 関数の呼び出し

関数を呼び出すときは、プログラムエディターに関数名を直接入力します。

例：

`RoutineName()`

注意：

- プロジェクト内の他の **GLOBAL** レベルの関数またはモジュールファイル内の **LOCAL** レベルの関数のみを呼び出すことができます。再帰呼び出しと 2 つのサブ関数の間のクロスコールはサポートしていません。
- 関数の呼び出しは、コンパイラでは独立したプログラム命令として扱われます。
- 関数内で関数を定義することはできません。

15.4 命令

15.4.1 変数の型変換

15.4.1.1 ByteToStr

説明	byte 型を string 型データに変換します。
定義	戻り値 データ型：string 変換後の文字列データを返します。 ByteToStr (BitData [\Hex] [\Okt] [\Bin] [\Char]) BitData データ型：Byte 変換する byte 型データを指定します。デフォルトは 10 進数に変換します。 ※Hex、識別子、16 進数に変換します。 ※Okt、識別子、8 進数に変換します。 ※Bin、識別子、2 進数に変換します。 ※Char、識別子、Ascii コード文字形式に変換します。
例	例 1 <pre> VAR byte data1 = 122 VAR string str1 str1 = ByteToStr(data1) // "122" str1 = ByteToStr(data1 \Hex) // "7A" str1 = ByteToStr(data1 \Okt) // "172" </pre>

	<pre>str1 = ByteToStr(data1 \Bin) // "01111010" str1 = ByteToStr(data1 \Char) // "z"</pre> byte 型の変数 data1 を宣言し、122 で初期化します。 data1 を string 型に変換します。 10 進数に変換すると「122」、16 進数に変換すると「7A」、8 進数に変換すると「172」、2 進数に変換すると「01111010」、文字に変換すると「z」となります。
--	---

15.4.1.2 DecToHex

説明	10 進数を 16 進数に変換します。
定義	戻り値 データ型：string 変換後の 16 進数データを返します。0-9、a-f、A-F で表現されます。 DecToHex(str) str データ型：string 変換する 10 進数を指定します。0-9 を使用します。
注意	データ範囲は 0～2147483647 または 0～7fffffff です。

15.4.1.3 DoubleToByte

説明	double または double 型配列を byte 型配列に変換します。
定義	戻り値 データ型：byte 型配列 double 型または double 型配列の変換によって取得された byte 型配列を返します。各 double は 8 バイトに変換されます。 DoubleToByte(dou1) dou1 データ型：double 変換する double 型変数または配列を指定します。

15.4.1.4 DoubleToStr

説明	double 型を string 型に変換します。
定義	戻り値 データ型：string 変換後の文字列データを返します。 DoubleToStr(Val, Dec) Val データ型：double 変換する double 型変数を指定します。 Dec データ型：int 小数点以下の保持桁数を指定します。
注意	小数点以下の桁数は最大 15 桁です。

15.4.1.5 HexToDec

説明	16 進数を 10 進数に変換します。
定義	戻り値 データ型：int 変換後の 10 進数データを返します。0-9 で表現されます。 HexToDec(str) str データ型：string 変換される 16 進数データを指定します。0-9、a-f、A-F を使用します。
注意	データ範囲は 0～2147483647 または 0～7fffffff です。

15.4.1.6 IntToByte

説明	int 型または int 型配列を byte 型配列に変換します。
定義	戻り値 データ型：byte 型配列 int 型または int 型配列の変換によって取得された byte 配列を返します。各 int は 4 バイトに変換されます。 IntToByte(int1) int1 データ型：int または int 配列 変換する int 型変数または配列を指定します。
注意	データ範囲は-2147483647～2147483647 です。

15.4.1.7 IntToStr

説明	int 型を string 型に変換します。
定義	戻り値 データ型：string 変換後の文字列データを返します。 IntToStr(int1) int1 データ型：int 変換する int 型変数を指定します。
注意	データ範囲は-2147483647～2147483647 です。

15.4.1.8 EulerToQuaternion

説明	オイラー角をクォータニオンに変換します。
定義	戻り値 変換結果を返します。0 は正常な変換、その他は変換の異常を意味します。 EulerToQuaternion (type,A,B,C,q1,q2,q3,q4) Type オイラー角の回転順序のタイプを指定します。EULER_XYZ と EULER_ZYX があります。 A,B,C データ型：double 変換されるオイラー角を指定します。 q1～q4

	データ型：double 変換によって取得されたクォータニオンが格納されます。
--	--

15.4.1.9 QuaternionToEuler

説明	クォータニオンをオイラー角に変換します。
定義	戻り値 変換結果を返します。0 は正常な変換、その他は変換の異常を意味します。 QuaternionToEuler (type,q1,q2,q3,q4,A,B,C) Type オイラー角の回転順序のタイプを指定します。EULER_XYZ と EULER_ZYX があります。 q1～q4 データ型：double 変換するクォータニオンを指定します。 A,B,C データ型：double 変換によって取得されたオイラー角が格納されます。

15.4.2 動作命令

15.4.2.1 MoveAbsJ

説明	MoveAbsJ (Move Absolute Joint) は、ロボットと外部軸を軸角度で定義された位置に移動させます。クイック移動や正確な軸角度へのロボットの移動に使用されます。すべての軸が同期して動作しロボット先端は不規則な曲線に沿って移動するので、周辺装置との衝突に注意してください。 MoveAbsJ 命令で使用される tool パラメーターは、ロボットの終点位置には影響しませんが、動力学的計算のために使用されます。
定義	MoveAbsJ ToJointPos, Speed, Zone, Tool, [Wobj] ([] 付きのパラメーターはオプションであり、省略可能です。) ToJointPos 目標関節角度 (To Joint Position) データ型：jointtarget ロボットと外部軸の目標角度と位置の値を指定します。 Speed、動作速度 (Move Speed) データ型：speed MoveAbsJ 実行時の動作速度 (ロボット先端の直線速度、軸速度、外部軸の速度など) を指定します。 Zone、滑らか軌跡エリア (Turning Zone) データ型：zone 滑らか軌跡エリアのサイズを指定します。 Tool データ型：tool 使用するツールを指定します。MoveAbsJ 命令は、ツールの TCP データを使用して動作速度と滑らか軌跡エリアのサイズを計算します。 [Wobj]、ワーク (Work Object) データ型：Wobj 使用するワークを指定します。ハンドツールの場合は、このパラメーターは無視されます。外部ツールを使用する場合は、このパラメーターを指定する必要があります。ロボットは wobj を使用して動作速度と滑らか軌跡エリアのサイズを計算します。

例	例 1 MoveAbsJ j10, v500, fine, tool1 ツール tool1 を使用し、v500 の速度で、j10 で定義された軸角度に軸動作で移動します。滑らか軌跡エリアのサイズは 0 です。 例 2 MoveAbsJ startpoint, v1000, z100, gripper, phone ツール gripper を使用し、ワーク座標系 phone、v1000 の速度で、startpoint で定義された軸角度に軸動作で移動します。滑らか軌跡エリアのサイズは 100 mm です。
---	--

15.4.2.2 MoveJ

説明	MoveJ (Move The Robot By Joint Movement) は、直交座標で定義された位置に軸動作で移動させます。すべての軸が同期して動作しロボット先端は不規則な曲線に沿って移動するので、周辺機器との衝突に注意してください。 MoveJ 命令と MoveAbsJ 命令の最大の違いは、目標点の形式が異なることです。MoveJ の目標点の形式は TCP の直交位置と姿勢であるのに対し、MoveAbsJ は各軸の角度です。
定義	MoveJ ToPoint, Speed, Zone, Tool, [Wobj] ([] 付きのパラメーターはオプションであり、省略可能です。) ToPoint 目標ポーズ (To Point) データ型: robtarget 直交座標で定義されている目標位置と姿勢を指定します。 Speed 動作速度 (Move Speed) データ型: speed MoveJ 実行時の動作速度 (ロボット先端の直線速度、軸速度、外部軸の速度など) を指定します。 Zone 滑らか軌跡エリア (Turning Zone) データ型: zone 滑らか軌跡エリアのサイズを指定します。 Tool データ型: tool 使用するツールを指定します。MoveJ 命令は、ツールの TCP データを使用して動作速度と滑らか軌跡エリアのサイズを計算します。 [Wobj] ワーク (Work Object) データ型: wobj 使用するワークを指定します。ハンドツールの場合は、このパラメーターは無視されます。外部ツールを使用する場合は、このパラメーターを指定する必要があります。ロボットは wobj を使用して動作速度と滑らか軌跡エリアのサイズを計算します。
例	例 1 MoveJ p30, v100, z50, tool1 ツール tool1 を使用し、v500 の速度で、p30 で定義された位置に軸動作で移動します。滑らか軌跡エリアのサイズは 50mm です。 例 2 MoveJ endpoint, v500, z50, gripper, wobj2 ツール gripper を使用し、ワーク座標系 wobj2、v500 の速度で、endpoint で定義された位置に軸動作で移動します。旋回エリアのサイズは 50mm です。

15.4.2.3 MoveL

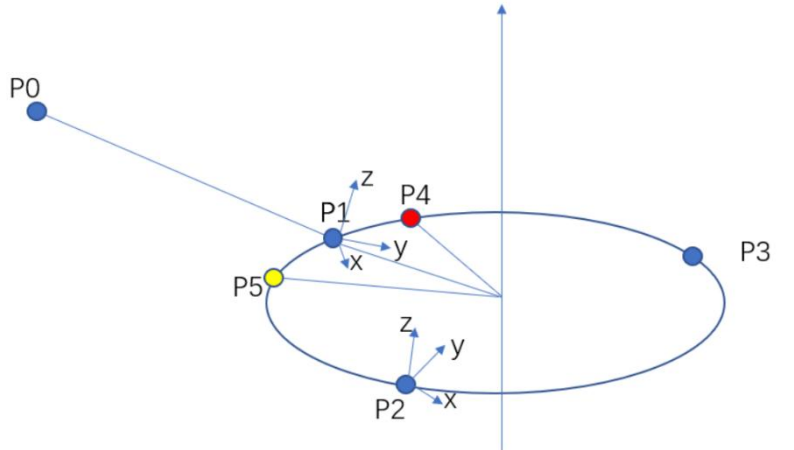
説明	ツールセンターポイント（TCP）を、直線に沿って所定の目標位置に動作させます。 始点と終点の姿勢が異なる場合は、姿勢は位置と同期して終点の姿勢まで回転します。直交動作速度と回転速度は個別に指定されるので、所定の速度の限界を超えないようにするために、MoveL 命令の最終的な動作時間は姿勢、位置、アーム角度の変化時間の最も長い物に依存します。したがって、特定の軌道（動作距離は小さいが、姿勢変化が大きいなど）を実行する際、ロボットの動作速度が著しく遅すぎたり速すぎたりする場合は、速度が適切に設定されているかどうかを確認してください。 ツールセンターポイント（TCP）を静止させ、ツールの姿勢のみを調整する必要がある場合は、同じ位置で異なる姿勢を持つ点を指定することで実現できます。
定義	MoveL ToPoint, Speed, Zone, Tool, [Wobj] （ [] 付きのパラメーターはオプションであり、省略可能です。） ToPoint 目標ポーズ（To Point） データ型：robtarget 直交座標で定義される目標位置と姿勢を指定します。 Speed 動作速度（Move Speed） データ型：speed MoveL 実行時の動作速度（ロボット先端の直線速度、軸速度、外部軸の速度など）を指定します。 Zone 滑らか軌跡エリア（Turning Zone） データ型：zone 滑らか軌跡エリアのサイズを指定します。 Tool データ型：tool 使用するツールを指定します。命令内の速度は、このツール TCP の直線速度と回転速度を指します。 [Wobj] ワーク（Work Object） データ型：wobj 使用するワークを指定します。ハンドツールの場合は、このパラメーターは無視されます。外部ツールを使用する場合は、このパラメーターを指定する必要があります。ロボットは wobj を使用して動作速度と滑らか軌跡エリアのサイズを計算します。
例	例 1 MoveL p10, v1000, z50, tool0 ツール tool0 を使用し、v1000 の速度で、p10 で定義された位置/姿勢に直線動作で移動します。滑らか軌跡エリアのサイズは 50mm です。 例 2 MoveL endpoint, v500, z50, gripper, wobj2 ツール gripper を使用し、ワーク座標系 wobj2、v500 の速度の直線動作で、endpoint で定義された目標点に動作します。滑らか軌跡エリアのサイズは 50mm です。

15.4.2.4 MoveC

説明	ツールセンターポイント（TCP）を、始点と補助点と終点によって決定された円形パスに沿って、円動作させます。 始点と終点の姿勢が異なる場合は、姿勢は位置と同期して終点の姿勢まで回転します。補助点の姿勢は円動作プロセスに影響しません。直交動作速度と回転速度は個別に指定されるので、所定の速度の限界を超えないようにするために、MoveC 命令の最終的な動作時間は姿勢、位置、アーム角度の変化時間の最も長い物に依存します。したがって、特定の軌道（動作距離は小さいが、姿勢変化が大きいなど）を実行する際、ロボットの動作速度が著しく遅すぎたり速すぎたりする場合は、速度が適切に設定されているかどうかを確認してください。
定義	MoveC AuxPoint, ToPoint, Speed, Zone, Tool, [Wobj] （ [] 付きのパラメーターはオプションであり、省略可能です。） AuxPoint 補助点（Auxiliary Point） データ型：robtarget 直交座標で定義される補助点の位置を指定します。円のサイズと動作方向を決定します。 この点の姿勢は最終的な軌道の実行に影響しません。 ToPoint 目標ポーズ（To Point） データ型：robtarget 直交座標で定義される目標位置と姿勢を指定します。 Speed 動作速度（Move Speed） データ型：speed MoveC 実行時の動作速度（ロボット先端の直線速度、軸速度、外部軸の速度など）を指定します。 Zone 滑らか軌跡エリア（Turning Zone） データ型：zone 滑らか軌跡エリアのサイズを指定します。 Tool データ型：tool 使用するツールを指定します。命令内の速度は、このツール TCP の直線速度と回転速度を指します。 [Wobj] ワーク（Work Object） データ型：wobj 使用するワークを指定します。ハンドツールの場合は、このパラメーターは無視されます。外部ツールを使用する場合は、このパラメーターを指定する必要があります。ロボットは wobj を使用して動作速度と滑らか軌跡エリアのサイズを計算します。
例	例 1 MoveC p10, p20, v1000, z50, tool0 ツール tool0 を使用し、TCP は v1000 の速度で、p10 を経由して p20 で定義された目標位置/姿勢に円動作で移動します。滑らか軌跡エリアのサイズは 50mm です。 例 2 MoveC auxpoint, endpoint, v500, z50, gripper, wobj2 ツール gripper を使用し、ワーク座標系 wobj2、v500 の速度で、auxpoint を経由して endpoint で定義された位置/姿勢に円動作で移動します。滑らか軌跡エリアのサイズは 50mm です。

15.4.2.5 MoveCF

説明	MoveCF (Move Circle Full) は、ツールセンターポイント (TCP) を、始点と 2 つの補助点によって決定された円形パスに沿って、指定の角度だけ円動作させます。MoveCF 全円動作の終点は全円動作実行角度によって決定され、2 つの補助点は円形パスの空間位置を決定します。全円動作中の姿勢はパラメーターで指定されているように変化し、補助点の姿勢は全円動作中の姿勢に影響しません。 MoveCF 命令の最終的な動作時間は位置変化の時間のみに依存します。したがって、円半径が小さい場合は、姿勢回転が速くなりすぎないように、動作速度が適切に設定されているか確認してください。
定義	MoveCF AuxPoint1, AuxPoint2, RunDeg, RotType, Speed, Zone, Tool, [Wobj] ([] 付きのパラメーターはオプションであり、省略可能です。) AuxPoint 補助点 (Auxiliary Point) データ型: robtarget 直交座標で定義されている補助点の位置を指定します。円弧の位置、サイズ、動作方向を決定します。2 つの補助点の姿勢は動作時の姿勢の変化に影響しません。 RunDeg 全円動作実行角度 データ型: double 数値範囲: -359~3600 (ロボット先端軸のソフトウェアリミット制限に注意)。 全円動作の角度であり、動作の中心角を指定します。反対方向に円を描く場合は負の角度を指定します。 RotType 姿勢変化方法 データ型: Char 選択可能なパラメーター: 「ConstPose」「RotAxis」「FixedAxis」 全円動作は 3 種類の姿勢変化方法をサポートします。用途に応じて選択してください。 「ConstPose」: 一定姿勢の全円動作です。動作中の姿勢は始点の姿勢のまま変化しません。 「RotAxis」: 軸回転での全円動作です。姿勢は始点姿勢と円の位置によって決定され、動作中の姿勢は円の中心軸を中心に一周変化します。この方法で動作するときは、先端の姿勢が大きく変化するので、動作中に特に最終軸がソフトウェアリミットを超えないように、最終軸の始点の角度を適切に設定してください。 「FixedAxis」: 軸固定での全円動作です。姿勢は始点姿勢と円の位置によって決定され、動作中の姿勢は円の中心軸を中心に変化しますが、自身の Z 軸を中心には回転しません。 Speed 動作速度 (Move Speed) データ型: speed MoveCF 実行時の TCP の動作速度を指定します。姿勢回転速度は全円パスの長さに関係し、speed パラメーターによっては決定されません。使用するときには最終軸の回転が速くなりすぎないように、このパラメーターの設定には注意してください。 Zone 滑らか軌跡エリア (Turning Zone) データ型: zone 滑らか軌跡エリアのサイズを指定します。 Tool データ型: tool 使用するツールを指定します。命令内の速度は、このツールの TCP 速度を指します。 [Wobj] ワーク (Work Object)

	データ型 : wobj 使用するワークを指定します。ハンドツールの場合は、このパラメーターは無視されます。外部ツールを使用する場合は、このパラメーターを指定する必要があります。ロボットは wobj を使用して動作速度と滑らか軌跡エリアのサイズを計算します。
例	 例 1 <code>MoveJ P1, v1000, z100, tool1</code> <code>MoveCF P2, P3, 360, "RotAxis", v100, z50, tool1</code> 始点 P1 と補助点 P2、P3 により円軌道が決定されます。 P1→P2→P3 を正の方向として、円軌道に沿って 360°動作し、P1 に戻ります。 動作中の姿勢は、始点 P1 の姿勢を基準とし円軌道の中心軸を中心に回転します。 例 2 <code>MoveJ P1, v1000, z100, tool1</code> <code>MoveCF P2, P3, -330, "ConstPose", v100, z50, tool1</code> P1→P2→P3 を正の方向として、円軌道に沿って-330°動作し、P5 に移動します。 動作中の姿勢は、P1 の姿勢のまま変化しません。 例 3 <code>MoveJ P1, v1000, z100, tool1</code> <code>MoveCF P3, P2, 30, "FixedAxis", v100, z50, tool1</code> P1→P3→P2 を正の方向として、円軌道に沿って 30°動作し、P4 に移動します。 動作中の姿勢は、始点 P1 の姿勢を基準とし円軌道の中心軸を中心に回転します。自身の Z 軸を中心に回転しません。

15.4.2.6 MoveT

説明	MoveT (Move trochoid) は、ツールセンターポイント (TCP) を、補助点経由で所定のステップ長のトロコイド曲線に沿って動作させます。 始点と終点の姿勢が異なる場合は、姿勢は位置と同期して終点の姿勢まで回転します。補助点の姿勢はトロコイド動作プロセスに影響しません。 直交移動速度と回転速度は個別に指定されるので、所定の速度の限界を超えないようにするため、MoveT 命令の最終的な動作時間は姿勢、位置、アーム角度の変化時間の最も長いものに依存します。したがって、特定の軌道（動作距離は小さいが、姿勢変化が大きいなど）を実行する際、ロボットの動作速度が著しく遅すぎたり速すぎたりする場合は、速度が適切に設定されているかどうかを確認してください。
----	---

定義

MoveT AuxPoint, ToPoint, Radius, Step, Speed, Zone, Tool, [Wobj]

([] 付きのパラメーターはオプションであり、省略可能です。)

AuxPoint 補助点 (Auxiliary Point)

データ型: robtarget

直交座標で定義されている補助点の位置を指定します。トロコイド曲線のサイズと動作方向を決定します。この点の姿勢は最終的な軌道の実行に影響しません。

ToPoint 目標ポーズ (To Point)

データ型: robtarget

直交座標で定義されている目標位置/姿勢を指定します。

Radius トロコイド半径

データ型: double

トロコイド動作の半径の大きさを指定します。単位は mm です。

Step ステップ長

データ型: double

トロコイド動作のステップ長を指定します。単位は mm です。

Speed 運動速度 (Move Speed)

データ型: speed

MoveT 実行時の動作速度 (ロボット先端の直線速度、軸速度、外部軸の軸速度など) を指定します。

Zone 滑らか軌跡エリア (Turning Zone)

データ型: zone

滑らか軌跡エリアのサイズを指定します。

Tool

データ型: tool

使用するツールを指定します。命令内の速度は、このツールの TCP 速度と回転速度を指します。

[Wobj] ワーク (Work Object)

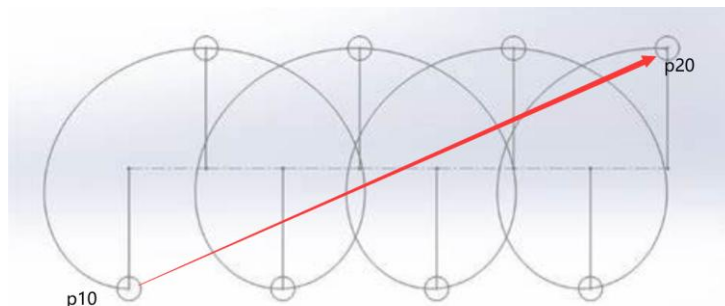
データ型: wobj

使用するワークを指定します。ハンドツールの場合は、このパラメーターは無視されます。外部ツールを使用する場合は、このパラメーターを指定する必要があります。ロボットは wobj を使用して動作速度と滑らか軌跡エリアのサイズを計算します。

例

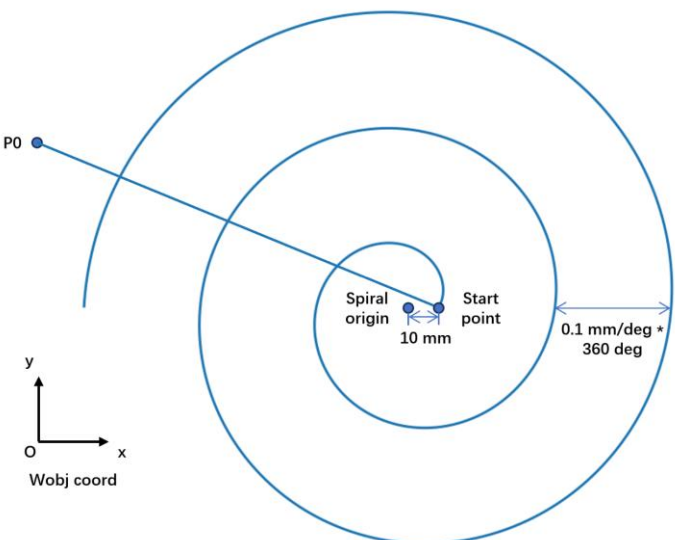
例 1**MoveT p10, p20, 150, 50, v1000, z50, tool0**

ツール tool0 を使用し、TCP は v1000 の速度のトロコイド曲線動作 (トロコイド半径は 150mm、ステッピングは 50mm) で、p10 を経由して最後に p20 で定義された目標位置に動作します。滑らか軌跡エリアのサイズは 50mm です。



15.4.2.7 MoveSP

説明	MoveSP (Move Spiral) は、ツールセンターポイント (TCP) を所定の初期半径、回転増分、合計回転角度、回転方向で、ワーク座標系 xy に平行な平面上に螺旋を描く動作を行います。動作中の姿勢は目標点の姿勢に直線的に変化します。 注意：MoveSP 命令が動作の途中で一時停止した後に実行を続ける場合は、現在の点を始点としてパスを再生成し、前のパスは継続しません。
定義	MoveSP Point, Radius, Radius_step, Angle, Direction, Speed, Zone, Tool, [Wobj] ([] 付きのパラメーターはオプションであり、省略可能です。) Point 目標点 データ型：robtarget 直交座標で定義される目標位置と姿勢を指定します。その姿勢のみを螺旋の終点の姿勢として使用します。 Radius 初期半径 データ型：double 螺旋の初期半径を指定します。単位は mm で、0mm 以上を指定する必要があります。螺旋の中心点の位置から TCP の現在位置を、ワーク座標系の x 軸の負の方向に移動させる距離を意味します。 Radius_step 回転増分 データ型：double 螺旋の回転増分です。単位は mm/deg で、0.0001mm/deg 以上を指定する必要があります。 Angle 合計回転角度 データ型：double 螺旋の合計回転角度を指定します。単位は deg で、0.1deg 以上 3600deg 以下を指定する必要があります。 Direction 回転方向 データ型：int 0 は時計回り、1 は反時計回りを意味します。 Speed 動作速度 (Move Speed) データ型：Speed MoveSP 実行時の動作速度 (ロボット先端の直線速度、軸速度、外部軸の軸速度など) を指定します。 Zone 滑らか軌跡エリア (Turning Zone) データ型：zone 滑らか軌跡エリアのサイズを指定します。螺旋動作は、滑らか軌跡エリアに対応していません。システムは螺旋動作の前後の滑らか軌跡を自動的にキャンセルします。 Tool データ型：Tool 使用するツールを指定します。命令内の速度は、このツールの TCP 速度と回転速度を指します。 [Wobj] ワーク (Work Object) データ型：Wobj 使用するワークを指定します。ハンドツールの場合は、このパラメーターは無視されます。この命令は外部ワークのみをサポートします。TCP 動作平面はワーク座標系 xy 平面に平行です。

例	例 1 <code>MoveL Start_point, v1000, fine, tool1</code> <code>MoveSP Point, 10, 0.1, 900, 1, v100, fine, tool1</code> まず、ツール tool1 を使用し、TCP が v1000 の速度で直線に沿って P0 から Start_point まで動作します。 次に、TCP は v100 の速度で螺旋に沿って動作します。螺旋の中心は Start_point からワーク座標系 x 軸の負の方向に 10mm の位置にあります。回転方向はワーク座標系の z 軸回りの反時計回りとなり、半径は 1°回転するごとに 0.1mm ずつ増加し、合計 900°回転した後に停止します。 
---	--

15.4.2.8 SearchL

説明	SearchL (Search Liner) は、ツールセンターポイント (TCP) を直線に沿って所定の目標位置に動作させる際に、位置をサーチします。 動作中、ロボットは 1 つのデジタル入力信号 (DI) または読み取り専用レジスタを監視します。監視される信号のトリガーの要件を満たすと、ロボットは直ちに現在位置を読み込みます。 この命令は通常、ハンドツールが表面観察用プローブである場合に使用します。SearchL 命令を使用してワークのおおよその座標を取得できます。この命令はモーションタスクでのみ使用できます。
定義	SearchL [action,] [signal_type,] signal_name, [trigger_mode,] save_rob, target_rob, Speed, Tool [,Wobj] ([] 付きのパラメーターはオプションであり、省略可能です。) Action DI トリガー後の動作 データ型：キーワード (パラメーターを指定しない場合は、停止しません) ¥Stop：素早い停止、ロボットがパスから逸脱する可能性があります。 ¥PStop：計画停止、ロボットは軌道に沿って速度制限なしで停止します。 Signal_type 信号タイプ データ型：キーワード (パラメーターを指定しない場合は DI) ¥DI：DI 信号 ¥Reg：レジスタ信号 Signal_name データ型：DI 信号またはレジスタ SearchL 命令が特定の動作をトリガーするための信号です。ユーザー定義の DI 信号または通信-レジスタで作成されたレジスタを指定します。

	trigger_mode DI 信号のトリガーモード データ型：キーワード（パラメーターを指定しない場合は、立ち上がりエッジトリガー） ¥Flanks：エッジトリガー（立ち上がりエッジまたは立ち下がりエッジ） ¥Posflank：立ち上がりエッジトリガー ¥Negflank：立ち下がりエッジトリガー ¥Highlevel：High トリガー ¥Lowlevel：Low トリガー 数値型のレジスタにおいて、0 は Low を、0 以外の値 は High を示します。 save_rob データ型：robtarg ロボットが信号をトリガーした時の位置データを保存するための変数を指定します。 target_rob データ型：robtarg 直線動作の目標位置/姿勢を指定します。 Speed 動作速度（Move Speed） データ型：speed SearchL 実行時の動作速度（ロボット先端の直線速度、軸速度、外部軸の軸速度など）を指定します。 Tool データ型：tool 使用するツールを指定します。命令内の速度は、このツールの TCP 速度と回転速度を指します。 [Wobj]、ワーク（Work Object） データ型：wobj 使用するワークを指定します。ハンドツールの場合は、このパラメーターは無視されます。外部ツールを使用する場合は、このパラメーターを指定する必要があります。ロボットは wobj を使用して動作速度と滑らか軌跡エリアのサイズを計算します。
例	例 1 SearchL di0, save_rob, target_rob, v500, tool0 ロボットはツール tool0 を使用し、TCP は v500 の速度で target_rob に向かって直線動作します。動作中に di0 が High になった場合は、その時のロボットの座標情報を save_rob に記録します。 例 2 SearchL !PStop, di0, !Lowlevel, save_rob, target_rob, v500, tool0 ロボットはツール tool0 を使用し、TCP は v500 の速度で target_rob に向かって直線動作します。動作中に di0 が Low になった場合は、ロボットは直ちに停止を計画し、その時点のロボットの座標情報を save_rob に記録します。 例 3 SearchL ¥PStop, ¥Reg, register0, save_rob, target_rob, v500, tool0, wobj1 ロボットはツール tool0 とワーク wobj1 を使用し、TCP は v500 の速度で target_rob に向けて直線動作します。もし動作中に register0 が 0 から 0 以外に変化すると、ロボットは即座に停止を計画し、その時点でロボットの座標情報を save_rob に記録します。

15.4.2.9 SearchC

説明	SearchC (Search Circle) は、ツールセンターポイント (TCP) を円に沿って所定の目標位置に動作させる際に、位置をサーチします。 動作中、ロボットは 1 つのデジタル入力信号 (DI) または読み取り専用レジスタを監視します。監視される信号のトリガーの要件を満たすと、ロボットは直ちに現在位置を読み込みます。 この命令は通常、ハンドツールが表面観察用プローブである場合に使用します。SearchC 命令を使用してワークのおおよその座標を取得できます。この命令はモーションタスクでのみ使用できます。
定義	SearchC [action,] [signal_type,] signal_name, [trigger_mode,] save_rob, aux_rob, target_rob, Speed, Tool [,Wobj] ([] 付きのパラメーターはオプションであり、省略可能です。) Action DI トリガー後の動作 データ型：キーワード (パラメーターを指定しない場合は、停止しません) ¥Stop：素早い停止、ロボットがパスから逸脱する可能性があります。停止速度が速いため、速度が v100 以下の場合のみ使用可能です。 ¥PStop：計画停止、ロボットは軌道に沿って速度制限なしで停止します。 Signal_type 信号タイプ データ型：キーワード (パラメーターを指定しない場合は DI) ¥DI：DI 信号 ¥Reg：レジスタ信号 Signal_name データ型：DI 信号またはレジスタ SearchC 命令が特定の動作をトリガーするための信号です。ユーザー定義の DI 信号または通信-レジスタで作成されたレジスタを指定します。 trigger_mode DI 信号のトリガーモード データ型：キーワード (パラメーターを指定しない場合は、立ち上がりエッジトリガー) ¥Flanks：エッジトリガー (立ち上がりエッジまたは立ち下がりエッジ) ¥Posflank：立ち上がりエッジトリガー ¥Negflank：立ち下がりエッジトリガー ¥Highlevel：High トリガー ¥Lowlevel：Low トリガー 数値型のレジスタにおいて、0 は Low を、0 以外の値 は High を示します。 save_rob データ型：robtarget ロボットが信号をトリガーした時の位置データを保存するための変数を指定します。 aux_rob データ型：robtarget 円動作の補助点を指定します。 target_rob データ型：robtarget 円動作の目標点を指定します。 Speed 動作速度 (Move Speed) データ型：speed SearchC 実行時の動作速度 (ロボット先端の直線速度、軸速度、外部軸の軸速度など) を指定します。

	Tool データ型：tool 使用するツールを指定します。命令内の速度は、このツールの TCP 速度と回転速度を指します。 [Wobj] ワーク (Work Object) データ型：wobj 使用するワークを指定します。ハンドツールの場合は、このパラメーターは無視されます。外部ツールを使用する場合は、このパラメーターを指定する必要があります。ロボットは wobj を使用して動作速度と滑らか軌跡エリアのサイズを計算します。
例	例 1 SearchC di0, save_rob, aux_rob, target_rob, v500, tool0 ツール tool0 を使用し、TCP は v500 の速度で補助点 aux_rob を経由して target_rob に向かって円動作します。動作中に di0 が High になった場合は、その時点のロボットの座標情報を save_rob に記録します。 例 2 SearchC \PStop, di0, \Flanks, save_rob, target_rob, v500, tool0 ツール tool0 を使用し、TCP は v500 の速度で補助点 aux_rob を経由して target_rob に向かって円動作します。動作中に di0 が Low から High に、または High から Low に変化した場合は、ロボットは直ちに停止を計画し、その時点のロボットの座標情報を save_rob に記録します。 例 3 SearchC \Reg, register0, \Flanks, save_rob, target_rob, v5, tool0 ロボットはツール tool0 を使用し、TCP は v5 の速度で補助点 aux_rob を経由して target_rob に向けて円動作します。動作中に register0 が 0 から 0 以外 0 に、または 0 以外から 0 に変化する、ロボットは即座に停止を計画し、信号の変化時にロボットの座標情報を save_rob に記録します。

15.4.3 トリガー命令

15.4.3.1 TrigIO

説明	1 つの trigdata 型変数を設定して、ロボット動作中に I/O 出力をトリガーします。デジタル出力 DO とデジタルグループ出力 GO をサポートしています。
定義	TrigIO TrigData, Distance, RefStart, SignalName, Value TrigData データ型：trigdata TrigIO で設定するトリガーを保存するための変数です。 Distance データ型：double 正の数値（負の数値は 0 として扱われる）で、パス上のトリガーイベントの位置を定義します。位置がパスの始点と終点のどちらからの距離なのかは、RefStart によって指定します。 RefStart データ型：bool トリガーの位置が始点 (True) と終点 (False) のどちらからの距離なのかを指定します。 SignalName データ型：signaldo または signalgo 出力したいデジタル出力またはデジタルグループ出力の信号名を指定します。

	Value データ型：bool または int トリガーされたときの出力信号の値を定義します。値のデータ型は SignalName の型と一致する必要があります
例	TrigL の例を参照して下さい。

15.4.3.2 TrigReg

説明	1 つの trigdata 型変数を設定して、ロボット動作中にレジスタの値を変更するために使用します。int16、bool、float、bit の 4 種類のレジスタタイプをサポートしています。
定義	TrigReg TrigData, Distance, RefStart, RegName, Value TrigData データ型：trigdata TrigReg で設定するトリガーを保存するための変数です。 Distance データ型：double 正の数値（負の数値は 0 として扱われる）で、パス上のトリガーイベントの位置を定義します。位置がパスの始点と終点のどちらからの距離なのかは、 RefStart によって指定します。 RefStart データ型：bool トリガー位置が始点（True）と終点（False）のどちらからの距離なのかを指定します。 RegName データ型：なし 変更するレジスタの名前を指定します。 注意：レジスタは RL プログラムでは作成できません。「ロボット-通信-レジスタ」ページで新しいレジスタを作成する必要があります。 Value データ型：int16、bool、float、bit トリガーされたときのレジスタの値を定義します。値のデータ型は RegName の型と一致する必要があります。一致していない場合は、強制的に型変換を行います。
例	TrigL の例を参照して下さい。

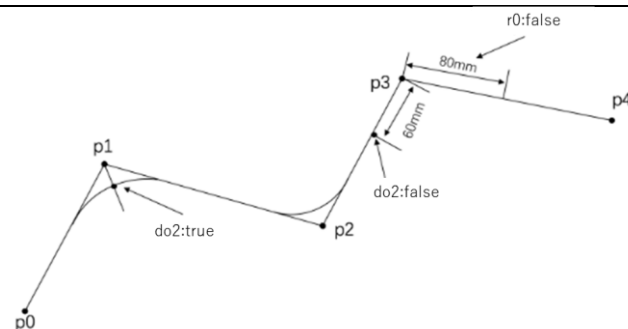
15.4.3.3 TrigVar

説明	1 つの trigdata 型変数を設定して、ロボット動作中に変数の値を変更するために使用します。bool、byte、int、double の 4 種類の変数をサポートしています。
定義	TrigVar TrigData, Distance, RefStart, VarName, Value TrigVar データ型：trigdata TrigVar で設定するトリガーを保存するための変数です。 Distance データ型：double 正の数値（負の数値は 0 として扱われる）で、パス上のトリガーイベントの位置を定義します。位置がパスの始点と終点のどちらからの距離なのかは、 RefStart によって指定します。 RefStart データ型：bool トリガー位置が始点（True）と終点（False）のどちらからの距離なのかを指定します。 VarName データ型：なし

	変更する変数名を指定します。 Value データ型：bool, byte, int, double トリガーされたときの変数の値を定義します。値のデータ型は VarName の型と一致する必要があります。一致していない場合は、強制的に型変換を行います。
例	TrigL の例を参照して下さい。

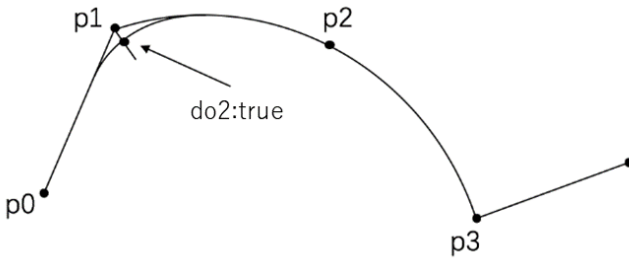
15.4.3.4 TrigL

説明	TrigL は、直線動作の命令を実行するという点で MoveL と似ています。違いは、TrigL は動作中に所定の位置で、事前定義されたトリガー操作を実行できることです。その他のパラメーターの数と意味に違いはありません。
定義	TrigL ToPoint, Speed, Trigger, Zone, Tool, [Wobj] ToPoint 目標位置/姿勢 (To Point) データ型：robtarget 直交座標で定義されている目標位置/姿勢を指定します。 Speed 動作速度 データ型：speed TrigL 実行時の動作速度（ロボット先端の直線速度、軸速度、外部軸の動作速度を含む）を指定します。 Trigger トリガー条件と動作 データ型：trigdata TrigX 命令で設定した trigdata を指定します。未設定だと、この行を処理した際にエラーメッセージを表示します。 Zone 滑らか軌跡エリア データ型：zone 滑らか軌跡エリアのサイズを指定します。 Tool ツール データ型：tool ツールを指定します。 [Wobj] ワーク データ型：wobj 使用するワークを指定します。ハンドツールの場合は、このパラメーターは無視されます。外部ツールを使用する場合は、このパラメーターを指定する必要があります。ロボットは wobj を使用して動作速度と滑らか軌跡エリアのサイズを計算します。
例	例 1 <pre> VAR trigdata tc1 VAR trigdata tc2 VAR trigdata tc3 ... //tc1、tc2、tc3 を設定する TrigIO tc1,0,true,do2,true TrigVar tc2,60,false,Bool0,false //bool0 は bool 型の RL 変数 TrigReg tc3,80,true,r0,false //r0 は bool 型のレジスタ ... //動作 MoveL p1,v500,z50,tool1 TrigL p2,v500,tc1,z50,tool1 TrigL p3,v500,tc2,fine,tool1 TrigL p4,v500,tc3,fine,tool1 </pre>



15.4.3.5 TrigC

説明	TrigC は、円動作の命令を実行するという点で MoveC と似ています。違いは、TrigC が動作中に所定の位置で、事前定義されたトリガー操作を実行できることです。その他のパラメーターの数と意味に違いはありません。
定義	TrigC AuxPoint, ToPoint, Speed, Trigger, Zone, Tool, [Wobj] AuxPoint 補助点 (Auxiliary Point) データ型 : robtarget 直交座標で定義されている補助点の位置/姿勢を指定します。 ToPoint 目標位置/姿勢 (To Point) データ型 : robtarget 直交座標で定義されている目標位置/姿勢を指定します。 Speed 動作速度 データ型 : speed TrigC 実行時の動作速度 (ロボット先端の直線速度、軸速度、外部軸の軸速度など) を指定します。 Trigger トリガー条件と動作 データ型 : trigdata TrigX 命令で設定した trigdata を指定します。未設定だと、この行を処理した際にエラーメッセージを表示します。 Zone 滑らか軌跡エリア データ型 : zone 滑らか軌跡エリアのサイズを定義します。 Tool ツール データ型 : tool ツールを指定します。 [Wobj] ワーク データ型 : wobj 使用するワークを指定します。ハンドツールの場合は、このパラメーターは無視されます。外部ツールを使用する場合は、このパラメーターを指定する必要があります。ロボットは wobj を使用して動作速度と滑らか軌跡エリアのサイズを計算します。
例	例 1 <pre>VAR trigdata tc1 ... //tc1 を設定する TrigIO tc1,0,true,do2,true ... //動作 MoveL p1,v500,z50,tool1</pre>

	TrigC p2,p3,v500,tc1,fine,tool1 
--	---

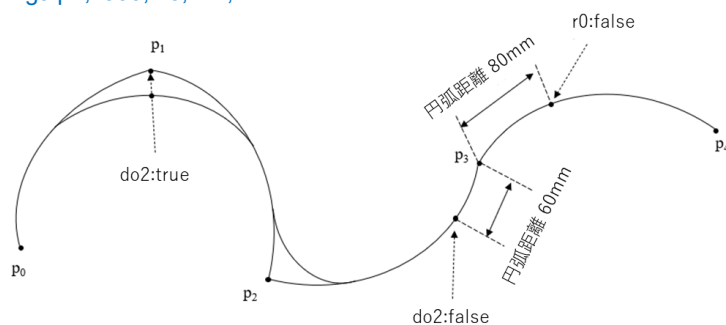
15.4.3.6 TrigJ

説明	TrigJ は、軸動作の命令を実行するという点で MoveJ と似ています。違いは、TrigJ は動作中に所定の位置で、事前定義されたトリガー操作を実行できることです。その他のパラメーターの数と意味の違いはありません。 ロボットが MoveJ 命令を実行すると、TCP の軌跡は通常円弧になります。TrigJ も軌跡は円弧になりますが、Trigger 信号をトリガーするために円弧に沿って距離を計算します。詳細については例を参照して下さい。
定義	TrigJ ToPoint, Speed, Trigger, Zone, Tool, [Wobj] ToPoint 目標ポーズ (To Point) データ型: robtarget 直交座標で定義されている目標位置/姿勢を指定します。 Speed 動作速度 データ型: speed TrigJ 実行時の動作速度 (ロボット先端の直線速度、軸速度、外部軸の軸速度など) を指定します。 Trigger トリガー条件と動作 データ型: trigdata TrigX 命令で設定された trigdata を指定します。未設定だと、この行を処理した際にエラーメッセージを表示します。 Zone 滑らか軌跡エリア データ型: zone 滑らか軌跡エリアのサイズを指定します。 Tool ツール データ型: tool ツールを指定します。 [Wobj] ワーク データ型: wobj 使用するワークを指定します。ハンドツールの場合は、このパラメーターは無視されます。外部ツールを使用する場合は、このパラメーターを指定する必要があります。ロボットは wobj を使用して動作速度と滑らか軌跡エリアのサイズを計算します。
例	例 1 <pre>VAR trigdata tc1 VAR trigdata tc2 VAR trigdata tc3 ... //tc1、tc2、tc3 を設定する TrigIO tc1,0,true,do2,true</pre>

```

TrigIO tc2,60,false,do2,false
TrigReg tc3,80,true,r0,false    //r0 は bool タイプのレジスタ
...
//動作
MoveJ p1,v500,z50,tool1
TrigJ p2,v500,tc1,z50,tool1
TrigJ p3,v500,tc2,fine,tool1
TrigJ p4,v500,tc3,fine,tool1

```



15.4.4 力制御命令

15.4.4.1 CalibSensorError

説明	力覚センサーの測定値をクリアし、現在の値をゼロ点として使用します。
定義	パラメーターはなく、直接使用します。
例	例 1 <pre> Fclnit Tool1, Wobj0, 0 FcStart SetSensorUseType 1 CalibSensorError </pre>
注意	- 協働ロボットのトルクセンサのゼロ点キャリブレーションにのみ適用され、産業用ロボットには適用されません。 - ロボットが静止状態である必要があります。任意の姿勢でこのインターフェースを呼び出してキャリブレーションできます。使用する際は、正しい負荷を設定してください。設定が不正確な場合、エラーが発生する可能性があります。

15.4.4.2 Fclnit

説明	ワーク、ツール、力制御座標系の設定など、力制御をオンにする前の初期化作業に使用します。
定義	Fclnit Tool, Wobj, ForceFrameRef Tool データ型：pose 力制御に使用されるツールを指定します。力制御座標系の原点はこのツールの TCP（姿勢は 3 番目のパラメーターで選択した座標系に従う）です。使用されるすべてのアダプターフランジはツールの定義に含まれる必要があることに注意してください。 Wobj データ型：pose 力制御に使用されるワークを指定します。力制御座標系の姿勢、サーチモード、終了条件など、多くの力制御機能はワーク座標系に対して定義されます。このパラメーターのデフォルトは wobj0 です。 ForceFrameRef データ型：int 力制御座標系がどの座標系に対して定義されるかを指定します。次の座標系をサポートし

	ています。 0：ワールド座標系 1：ワーク座標系 2：ツール座標系 3：パス座標系 4：ベース座標系 デフォルトはワールド座標系（0）です。
例	例 1 Fclnit Tool1, Wobj0, 0 力制御を初期化し、力制御に使用されるツール tool1 とワーク wobj0 を指定し、力制御の座標系をワールド座標系に対して定義します。
注意	Fclnit と FcStop の間で Fclnit を再度呼び出すことはできません。

15.4.4.3 SetControlType

説明	力制御のタイプを設定します。
定義	SetControlType ctrl_type ctrl_type データ型：int 力制御タイプを指定します。次のタイプをサポートします。 0：軸動作インピーダンス 1：直交動作インピーダンス
例	例 1 Fclnit Tool1, Wobj0, 0 SetControlType 0 Fclnit の後に、力制御モードを軸動作インピーダンスに設定します。
注意	Fclnit の後、FcStart の前に実行する必要があります。この要件を満たさない場合は、力制御タイプは正常に設定されません。 力制御タイプを設定した後は、対応するインピーダンス剛性を設定する必要があります。 制御タイプを軸動作インピーダンスに設定した場合は、SetJntCtrlStiffVec 命令を呼び出し、軸動作インピーダンス剛性を設定する必要があります。この時に直交動作インピーダンス剛性を設定しても、インピーダンス剛性の設定は有効になりません。軸動作インピーダンスと直交動作インピーダンスの両方を設定した場合は、軸動作インピーダンス剛性の設定のみが有効になります。 デフォルトは、軸動作インピーダンスと直交動作インピーダンス剛性は両方とも 0 です。

15.4.4.4 SetCartNsStiff

説明	零空間インピーダンス剛性を設定します。
定義	SetCartNsStiff cart_ns_stiff cart_ns_stiff データ型：double 直交零空間インピーダンス剛性を指定します。値の範囲は（0～4）で、単位は Nm/rad です。
例	例 1 Fclnit Tool1, Wobj0, 0 SetControlType 1 SetCartNSStiff 2 インピーダンス制御モードを直交インピーダンスに設定し、零空間剛性を 2 に設定します。
注意	このインターフェースは、 SetSensorUseType 1 の後、つまりインピーダンス制御タイプを直交

インピーダンスに設定した後に呼び出す必要があります。この要件を満たさない場合は、零空間インピーダンスのパラメーターは正常に設定されません。

15.4.4.5 SetJntCtrlStiffVec

説明	軸動作インピーダンス剛性を設定します
定義	SetJntCtrlStiffVec jnt1_stiff, jnt2_stiff, jnt3_stiff, jnt4_stiff, jnt5_stiff, jnt6_stiff, jnt7_stiff Jnt1_stiff 、データ型：double、関節 1 のインピーダンス剛性、単位：Nm/rad。 Jnt2_stiff 、データ型：double、関節 2 のインピーダンス剛性、単位：Nm/rad。 Jnt3_stiff 、データ型：double、関節 3 のインピーダンス剛性、単位：Nm/rad。 Jnt4_stiff 、データ型：double、関節 4 のインピーダンス剛性、単位：Nm/rad。 Jnt5_stiff 、データ型：double、関節 5 のインピーダンス剛性、単位：Nm/rad。 Jnt6_stiff 、データ型：double、関節 6 のインピーダンス剛性、単位：Nm/rad。 Jnt7_stiff 、データ型：double、関節 7 のインピーダンス剛性、単位：Nm/rad。
例	例 1 <pre>Fclnit Tool1, Wobj0, 0 SetControlType 0 SetJntCtrlStiffVec 1500,1500, 1500,1500,100,100,100</pre> インピーダンス制御モードを軸動作インピーダンスに設定し、関節 1～7 のインピーダンス剛性をそれぞれ 1500、1500、1500、1500、100、100、100 に設定します。
注意	このインターフェースは、 SetSensorUseType 0 の後、つまりインピーダンス制御タイプを軸動作インピーダンスに設定した後に呼び出す必要があります。この要件を満たさない場合は、軸動作インピーダンスのパラメーターは正常に設定されません。

各協働ロボットの各軸の剛性上限、単位：Nm/rad

	J1	J2	J3	J4	J5	J6	J7
ER3P	6000	6000	6000	1000	1000	1000	1000
ER7P	6000	6000	6000	1000	1000	1000	1000
ER3	3000	3000	3000	300	300	300	
ER7	3000	3000	3000	300	300	300	
SR3	3000	3000	3000	300	300	300	
SR4	3000	3000	3000	300	300	300	
SR5	3000	3000	3000	300	300	300	
CR7	6000	6000	6000	1000	1000	1000	
CR12	20000	20000	20000	3000	2500	2500	
CR17	20000	20000	20000	2500	2500		
CR17/25	20000	20000	20000	2500	2500		
CR18	20000	20000	20000	3000	2500	2500	
CR20	20000	20000	20000	3000	2500	2500	
CR25	20000	20000	20000	2500	2500		

15.4.4.6 SetCartCtrlStiffVec

説明	直交動作インピーダンスの剛性を設定します
定義	SetCartCtrlStiffVec trans_stiff_x, trans_stiff_y, trans_stiff_z, rot_stiff_x, rot_stiff_y, rot_stiff_z trans_stiff_x 、データ型：double、X 方向の直交インピーダンス力剛性、単位：N/m。 trans_stiff_y 、データ型：double、Y 方向の直交インピーダンス力剛性、単位：N/m。

	trans_stiff_z 、データ型：double、Z 方向の直交インピーダンス力剛性、単位：N/m。 rot_stiff_x 、データ型：double、X 方向の直交インピーダンストルク剛性、単位：N.m/rad。 rot_stiff_y 、データ型：double、Y 方向の直交インピーダンストルク剛性、単位：N.m/rad。 rot_stiff_z 、データ型：double、Z 方向の直交インピーダンストルク剛性、単位：N.m/rad。
例	例 1 FcInit Tool1, Wobj0, 0 SetControlType 1 SetCartCtrlStiffVec 1000, 1000, 1000, 100, 100, 100 インピーダンス制御モードを直交インピーダンスに設定し、X/Y/Z 方向のインピーダンス力剛性を 1000 に、インピーダンストルク剛性を 100 に設定します。
注意	この命令は、 SetSensorUseType 1 の後、つまりインピーダンス制御タイプを直交インピーダンスに設定した後に呼び出す必要があります。この要件を満たさない場合は、直交インピーダンスのパラメーターは正常に設定されません。

各協働ロボットの各方向の剛性上限、単位：N/m と Nm/rad

	trans_x	trans_y	Trans_z	rot_x	rot_y	rot_z
ER3P	6000	6000	6000	1000	1000	1000
ER7P	6000	6000	6000	1000	1000	1000
ER3	3000	3000	3000	300	300	300
ER7	3000	3000	3000	300	300	300
SR3	3000	3000	3000	300	300	300
SR4	3000	3000	3000	300	300	300
SR5	3000	3000	3000	300	300	300
CR7	6000	6000	6000	1000	1000	1000
CR12	18000	18000	18000	2500	2500	2500
CR17/25	18000	18000	18000	2500	2500	2500
CR18	18000	18000	18000	2500	2500	2500
CR20	18000	18000	18000	2500	2500	2500

15.4.4.7 SetJntTrqDes

説明	軸動作での目標トルクを設定します
定義	SetJntTrqDes tau_d1, tau_d2, tau_d3, tau_d4, tau_d5, tau_d6, tau_d7 tau_d1 、データ型：double、関節 1 の目標トルク、値の範囲 (-30~30)、単位：N.m。 tau_d2 、データ型：double、関節 2 の目標トルク、値の範囲 (-30~30)、単位：N.m。 tau_d3 、データ型：double、関節 3 の目標トルク、値の範囲 (-30~30)、単位：N.m。 tau_d4 、データ型：double、関節 4 の目標トルク、値の範囲 (-30~30)、単位：N.m。 tau_d5 、データ型：double、関節 5 の目標トルク、値の範囲 (-30~30)、単位：N.m。 tau_d6 、データ型：double、関節 6 の目標トルク、値の範囲 (-30~30)、単位：N.m。 tau_d7 、データ型：double、関節 7 の目標トルク、値の範囲 (-30~30)、単位：N.m。
例	例 1 FcInit Tool1, Wobj0, 0 SetControlType 0 FcStart SetJntTrqDes 5,5,5,5,5,5,5 FcStop すべての関節の目標トルクを 5N.m に設定します。
注意	この命令は、FcStart の後、FcStop の前に呼び出す必要があります。この要件を満たさない場

合は、軸動作での目標トルクは正常に設定されません。

15.4.4.8 SetCartForceDes

説明	直交動作での目標力/トルクを設定します。
定義	SetCartForceDes force_x, force_y, force_z, torque_x, torque_y, torque_z force_x、データ型：Double、X 方向の直交の目標力、値の範囲（-60～60）、単位：N。 force_y、データ型：Double、Y 方向の直交の目標力、値の範囲（-60～60）、単位：N。 force_z、データ型：Double、Z 方向の直交の目標力、値の範囲（-60～60）、単位：N。 torque_x、データ型：Double、X 方向の直交の目標トルク、値の範囲（-10～10）、単位：N.m。 torque_y、データ型：Double、Y 方向の直交の目標トルク、値の範囲（-10～10）、単位：N.m。 torque_z、データ型：Double、Z 方向の直交の目標トルク、値の範囲（-10～10）、単位：N.m。
例	例 1 <pre>Fclnit Tool1, Wobj0, 0 SetControlType 1 FcStart SetCartForceDes 0,0,5,0,0,0 FcStop</pre> z 方向の目標力を 5N に設定します。
注意	この命令は、FcStart の後、FcStop の前に呼び出す必要があります。この制限を満たさない場合は、直交動作の目標力/トルクは正常に設定されません。

15.4.4.9 SetSineOverlay

説明	指定した軸周りに回転する正弦波探索運動を設定します
定義	SetSineOverlay line_dir, amplify, frequency, phase, bias line_dir データ型：int 探索動作の基準軸を指定します。以下をサポートします。 0：X 軸 1：Y 軸 2：Z 軸 Amplify データ型：double 探索動作の振幅を指定します。単位は N.m です。 Frequency データ型：double 探索動作の周波数を指定します。単位は Hz です。 Phase データ型：double 探索動作の位相を指定します。値の範囲は（0～3.14）で、単位は rad です。 Bias データ型：double 探索動作のオフセットを指定します。値の範囲は（0～10）で、単位は N.m です。
例	例 1 <pre>Fclnit Tool1, Wobj0, 0 SetControlType 1 SetSineOverlay 0, 10, 5, 3.14, 2</pre> x 軸（0）周りに回転する正弦波探索動作を設定します。振幅は 10N.m、頻度は 5Hz、位相は

	3.14、オフセットは 2N.m です。
注意	SetSensorUseType 1 の後、つまりインピーダンス制御タイプを直交インピーダンスに設定した後、 StartOverlay の前にこの命令を呼び出す必要があります。この要件を満たさない場合は、探索運動は正常に設定されません。

各協働ロボットのパラメーター上限：

	振幅の上限	周波数の上限
ER3P	10	5
ER7P	10	5
ER3	10	5
ER7	10	5
SR3	5	5
SR3-C	5	5
SR3-A	5	5
SR4	5	5
SR4-C	5	5
CR7	10	5
CR12	10	5
CR17/25	10	5
CR18	10	5
CR20	10	5

15.4.4.10 SetLissajousOverlay

説明	指定した平面内のリサージュ曲線探索運動を設定します。
定義	SetLissajousOverlay plane, amplify_one, frequency_one, amplify_two, frequency_two, phase_diff Plane データ型：int 探索動作の基準平面を指定します。以下をサポートします。 0：基準平面は XY 平面 1：基準平面は XZ 平面 2：基準平面は YZ 平面 amplify_one データ型：double 探索動作の 1 つ目の方向の振幅を指定します。値の範囲は（0～20）で、単位は N.m です。 frequency_one データ型：double 探索動作の 1 つ目の方向の周波数を指定します。値の範囲は（0～5）で、単位は Hz です。 amplify_two データ型：double 探索動作の 2 つ目の方向の振幅を指定します。値の範囲は（0～20）で、単位は N.m です。 frequency_two データ型：double 探索動作の 2 つ目の方向の周波数を指定します。値の範囲は（0～5）で、単位は Hz です。 phase_diff データ型：double 探索動作の二方向間の位相偏差を指定します。値の範囲は（0～3.14）で、単位は rad で

	す。
例	例 1 <code>Fclnit Tool1, Wobj0, 0</code> <code>SetControlType 1</code> <code>SetLissajousOverlay 0, 5, 2.5, 10, 5, 3.14</code> xy 平面 (0) におけるリサージュ曲線索動作を設定します。x 方向の振幅は 5N.m、周波数は 2.5Hz、y 方向の振幅は 10N.m、周波数は 5Hz、x 方向に対する y 方向の位相偏差は 3.14 rad です。
注意	<code>SetSensorUseType 1</code> の後、つまりインピーダンス制御タイプを直交インピーダンスに設定した後、<code>StartOverlay</code> の前にこの命令を呼び出す必要があります。この要件を満たさない場合は、探索運動は正常に設定されません。

15.4.4.11 SetLoad

説明	力制御モジュールに使用される負荷情報を設定します。
定義	SetLoad m,rx,ry,rz,lxx,lyy,lzz M データ型：double 負荷質量を指定します。単位は kg で、範囲は (0~25) です。 Rx データ型：double フランジ座標系の x 方向における負荷重心の距離を指定します。単位は mm で、範囲は (-300, 300) です。 Ry データ型：double フランジ座標系の y 方向における負荷重心の距離を指定します。単位は mm で、範囲は (-300, 300) です。 Rz データ型：double フランジ座標系の z 方向における負荷重心の距離を指定します。単位は mm で、範囲は (-300, 300) です。 lxx データ型：double x 軸における負荷重心の主軸イナーシャを指定します。単位は kg*mm² で、範囲は (0, 100000) です。 lyy データ型：double y 軸における負荷重心の主軸イナーシャを指定します。単位は kg*mm² で、範囲は (0, 100000) です。 lzz データ型：double zx 軸における負荷重心の主軸イナーシャを指定します。単位は kg*mm² で、範囲は (0, 100000) です。
例	例 1 <code>Fclnit Tool1, Wobj0, 0</code> <code>FcStart</code> <code>SetLoad 1,0,0,10,0.001,0.001,0.0001</code> 質量が 1kg、フランジ座標系における重心がそれぞれ 0、0、10mm、負荷の負荷重心座標系に対する主軸慣性モーメントがそれぞれ 0.001kgmm²、0.001kgmm²・0.0001kgmm²である負荷を設定します。

注意	この命令は、 FcStart の後に呼び出す必要があります。この要件を満たさない場合は、負荷パラメーターは正常に設定されません。
----	---

15.4.4.12 FcStart

説明	力制御機能をオンにします。この命令は、ロボットを単純な位置制御から力制御に切り替えます。
定義	パラメーターはなく、直接使用します。
例	例 1 FcInit Tool1, Wobj0, 0 FcStart FcInit の後、FcStart により力制御が有効になり、この時点でロボットは力制御モードになります。
注意	FcInit の後でこの命令を呼び出します。呼び出す前にロボットの機械原点、力覚センサーのゼロ点、負荷情報が正しく設定され、本体パラメーターが正しく同定されていることを確認してください。設定が間違っている場合は、力制御機能の効果に影響したり、力制御機能がオンにならなくなります。

15.4.4.13 FcStop

説明	力制御機能をオフにします。この命令はロボットを力制御から位置制御に切り替えます。この命令を実行すると、すべての探索動作が自動的に停止します。
定義	パラメーターはなく、直接使用します。
例	例 1 FcInit Tool1, Wobj0, 0 FcStart FcStop FcStop で力制御を停止し、ロボットを力制御から位置制御に切り替えます。この命令を実行すると、すべての力制御状態がクリアされます。
注意	この命令は FcStart の後に呼び出します。呼び出すと、力制御状態（力制御負荷情報、インピーダンスパラメーター、探索動作、目標力など）がクリアされます。力制御を再度オンにするには、再度 FcInit を実行する必要があります。

15.4.4.14 StartOverlay

説明	事前に設定した探索動作を有効にします。
定義	パラメーターはなく、直接使用します。
例	例 1 FcInit Tool1, Wobj0, 0 SetControlType 1 SetSineOverlay 0, 10, 5, 3.14, 2 SetLissajousOverlay 0, 5, 2.5, 10, 5, 3.14 FcStart StartOverlay 探索動作をオンにします。探索動作は事前に設定されたさまざまな探索運動を重ね合わせたものです。本例の探索動作は、x 軸周りの正弦波探索動作と xy 平面内のリサージュ曲線探索動作を重ね合わせたものです。
注意	この命令は、 FcStart の後に呼び出す必要があります。この要件を満たさない場合は、探索動作は正常に有効になりません。

15.4.4.15 PauseOverlay

説明	探索動作を一時停止します。
定義	パラメーターはなく、直接使用します
例	例 1 <pre>Fclnit Tool1, Wobj0, 0 SetControlType 1 SetSineOverlay 0, 10, 5, 3.14, 2 FcStart StartOverlay PauseOverlay</pre> 探索動作を一時停止します。
注意	この命令は、 StartOverlay の後に呼び出すことができます。

15.4.4.16 RestartOverlay

説明	一時停止した探索動作を再度有効にします。
定義	パラメーターはなく、直接使用します。
例	例 1 <pre>Fclnit Tool1, Wobj0, 0 SetControlType 1 SetSineOverlay 0, 10, 5, 3.14, 2 FcStart StartOverlay PauseOverlay RestartOverlay</pre> 探索動作を再度有効にします。
注意	この命令は、 PauseOverlay の後に呼び出すことができます。 PauseOverlay にて一時停止した探索動作を再度有効にするために使用します。

15.4.4.17 StopOverlay

説明	探索動作を終了します。
定義	パラメーターはなく、直接使用します。
例	例 1 <pre>Fclnit Tool1, Wobj0, 0 SetControlType 1 SetSineOverlay 0, 10, 5, 3.14, 2 FcStart StartOverlay StopOverlay</pre> 探索動作を終了します。
注意	この命令は、 StartOverlay より後に呼び出して使用します。

15.4.4.18 FcCondForce

説明	接触力による終了条件を定義します。
定義	FcCondForce xmin, xmax, ymin, ymax, zmin, zmax, lsInside, TimeOut Xmin データ型：double X 方向の力制限の下限を指定します。負の値は X の負の方向の最大値を意味します。単位は N で、デフォルト値は負の無限大です。 Xmax データ型：double X 方向の力制限の上限を指定します。負の値は X の負の方向の最小値を意味します。単位

	は N で、デフォルト値は正の無限大です。 Ymin データ型：double Y 方向の力制限の下限を指定します。負の値は Y の負の方向の最大値を意味します。単位は N で、デフォルト値は負の無限大です。 Ymax データ型：double Y 方向の力制限の上限を指定します。負の値は Y の負の方向の最小値を意味します。単位は N で、デフォルト値は正の無限大です。 Zmin データ型：double Z 方向の力制限の下限を指定します。負の値は Z の負の方向の最大値を意味します。単位は N で、デフォルト値は負の無限大です。 Zmax データ型：double Z 方向の力制限の上限を指定します。負の値は Z の負の方向の最小値を意味します。単位は N で、デフォルト値は正の無限大です。 IsInside データ型：bool 制限条件の内部：True か外部：False を指定します。 TimeOut データ型：double タイムアウトを秒単位で定義します。値の範囲は 1～600 です。
例	例 1 FcInit Tool1, Wobj0, 0 FcStart FcCondForce -100, 100, -100, 100, -100, 100, true, 60 終了条件を設定します。力制御座標系の X/Y/Z 軸方向における接触力の大きさが±100N の範囲外であれば終了します。タイムアウトは 60 秒です。
注意	この命令は、FcStart の後、FcStop の前に呼び出す必要があります。この要件を満たさない場合は、接触力の終了条件は正常に設定されません。

15.4.4.19 FcCondPosBox

説明	接触位置による終了条件を定義します。
定義	FcCondPosBox SupvFrame, Box, IsInside, Timeout SupvFrame データ型：pose 監視されるエリアがどの座標系に対して定義されるかを指定します。この座標系は、ワーク座標系に座標系をオーバーレイすることで変換されます。変換座標系は pose で定義されます。デフォルトは pose0 を使用し、変換されずに直接ワーク座標系を使用します。 Box データ型：fcboxvol 直方体を指定します。 IsInside データ型：bool 制限条件の内部：True か外部：False を定義します。 TimeOut データ型：Double タイムアウトを秒単位で定義します。値の範囲は 1～600 です。

例	例 1 <pre>Fclnit Tool1, Wobj0, 0 FcStart VAR fcbxvol box1 = fcbv:{-100.0, 100.0, -200.0, 200.0, -300.0, 300.0} VAR pose pose1 = pe:{{0, 0, 0},{1, 0, 0, 0}} FCCondPosBox pose1, box1, false, 60</pre> 終了事件を定義します。ロボットの TCP が定義されたエリアの中に入るか、または 60 秒以上待機したときに、終了条件がトリガーされます。
注意	このインターフェースは、FcStart の後、FcStop の前に呼び出す必要があります。この要件を満たさない場合は、正常に設定されません。

15.4.4.20 FcCondTorque

説明	接触トルクによる終了条件を定義します。
定義	FcCondTorque xmin, xmax, ymin, ymax, zmin, zmax, lsInside, TimeOut Xmin データ型：double X 方向のトルク制限の下限を指定します。負の値は X の負の方向の最大値を意味します。 単位は N.m で、デフォルト値は負の無限大です。 Xmax データ型：double X 方向のトルク制限の上限を指定します。負の値は X の負の方向の最小値を意味します。 単位は N.m で、デフォルト値は正の無限大です。 Ymin データ型：double Y 方向のトルク制限の下限を指定します。負の値は Y の負の方向の最大値を意味します。 単位は N.m で、デフォルト値は負の無限大です。 Ymax データ型：double Y 方向のトルク制限の上限を指定します。負の値は Y の負の方向の最小値を意味します。 単位は N.m で、デフォルト値は正の無限大です。 Zmin データ型：double Z 方向のトルク制限の下限を指定します。負の値は Z の負の方向の最大値を意味します。 単位は N.m で、デフォルト値は負の無限大です。 Zmax データ型：double Z 方向のトルク制限の上限を指定します。負の値は Z の負の方向の最小値を意味します。 単位は N.m で、デフォルト値は正の無限大です。 lsInside データ型：bool 制限条件の内部：True か外部：False を指定します。 TimeOut データ型：Double タイムアウトを秒単位で指定します。値の範囲は 1～600 です。
例	例 1 <pre>Fclnit Tool1, Wobj0, 0 FcStart FcCondTorque -10, 10, -10, 10, -10, 10, true, 60</pre> 終了条件を定義します。力制御座標系の各軸方向における接触力の大きさが 10Nm より大きい、またはタイムアウトが 60 秒以上になったときに、終了条件がトリガーされます。

注意	このインターフェースは、 FcStart の後、 FcStop の前に呼び出す必要があります。この要件限を満たさない場合は、接触トルクの終了条件は正常に設定されません。
----	--

15.4.4.21 FcCondWaitWhile

説明	事前に定義された終了条件をアクティブにし、これらの条件が False になるかタイムアウトになるまで現在の行で待ちます。
定義	パラメーターはなく、直接使用します。
例	例 1 <pre>FcInit Tool1, Wobj0, 0 FcStart FcCondTorque -10, 10, -10, 10, -10, 10, true, 60 FcCondForce -100, 100, -100, 100, -100, 100, true, 60 FcCondWaitWhile</pre> 終了条件をアクティブにし、プログラム現在位置で終了条件がトリガーされるのを待ちます。
注意	力制御の終了条件を定義した後に実行する必要があります。

15.4.4.22 FcMonitor

説明	力制御保護モニターのオン/オフを切り替えます。 力制御保護モニターは、コントローラーがインピーダンスモードの際に、ユーザーが設定した保護パラメーターを使用し、ロボットの速度、運動量、出力などの状態を制限し、インピーダンスモードにおける保護を実現します。
定義	FcMonitor On、力制御保護モニターをオンにします。ユーザーが設定した保護パラメーターがインピーダンス動作で有効になります。 FcMonitor Off、力制御モジュールの保護モニターをオフにします。ユーザーが設定した保護パラメーターがインピーダンス動作で無効になります。コントローラーは、デフォルトの保護パラメーターを使用してロボットの動作を制限します。
例	例 1 <pre>FcInit tool0,wobj0,0 SetControlType 0 SetFcJointVelMax 1.0, 1.0, 1.0, 0.5, 0.5, 0.5, 0 SetFcJointEnergyMax 100, 100, 100, 100, 100, 100, 0 FcMonitor On//力制御モジュールの保護モニタをオンにする FcStart ... FcMonitor Off//力制御モジュールの保護モニタをオフにする ... FcStop</pre>
注意	- インピーダンスモードを使用する場合は、力制御保護モニターをオンにする前に、保護パラメーターを設定することをお勧めします。 - 保護パラメーターが設定されていない場合、または力制御保護モニターをオフにした場合は、コントローラーはデフォルトの保護パラメーターを有効にします。 - デフォルトの保護パラメーターの制限は多いため、インピーダンスモードを使用するときは力制御保護エラーが発生しやすくなります。

15.4.4.23 GetEndToolTorque

説明	ロボットの現在のトルク情報を取得します。
定義	戻り値：トルク情報 データ型：torqueInfo GetEndToolTorque Tool, Wobj [, RefType]

	([]はこのパラメーターが省略可能であることを示しています) Tool データ型：tool 現在使用中のツールを指定します。 Wobj データ型：wobj 現在使用中のワークを指定します。 RefType データ型：int トルクに対する基準座標系を指定します。 0：デフォルト値、ワールド座標系に対する先端のトルク情報 1：フランジに対する先端のトルク情報 2：ツール座標系に対する先端のトルク情報
例	<pre>TorqueInfo tmp_info = GetEndtoolTorque(tool1, wobj1) //tool1 wobj1 条件におけるロボット先端ツールのトルク情報構造体を取得する ... print(tmp_info.joint_torque.measure_torque) print(tmp_info.joint_torque.external_torque) //各軸の測定力と外部力を表示する ... print(tmp_info.cart_torque.m_torque) //直交空間トルクを表示する ... print(tmp_info.cart_torque.m_force[1]) print(tmp_info.cart_torque.m_torque[1]) //X 方向の力とトルク情報を印刷する</pre>

15.4.4.24 SetFcJointVelMax

説明	インピーダンス動作時の最大軸速度を設定します。
定義	SetFcJointVelMax jnt1_vel, jnt2_vel, jnt3_vel, jnt4_vel, jnt5_vel, jnt6_vel, jnt7_vel Jnt1_vel、データ型：double、関節 1 の最大速度、単位：rad/s。 Jnt2_vel、データ型：double、関節 2 の最大速度、単位：rad/s。 Jnt3_vel、データ型：double、関節 3 の最大速度、単位：rad/s。 Jnt4_vel、データ型：double、関節 4 の最大速度、単位：rad/s。 Jnt5_vel、データ型：double、関節 5 の最大速度、単位：rad/s。 Jnt6_vel、データ型：double、関節 6 の最大速度、単位：rad/s。 Jnt7_vel、データ型：double、関節 7 の最大速度、単位：rad/s。
例	<code>SetFcJointVelMax 1.0,1.0,1.0,1.0,0.5,0.5,0.5</code>

各協働ロボットの各軸の速度上限、単位：rad/s

	J1	J2	J3	J4	J5	J6	J7
ER3P	1.0	1.0	1.0	1.0	1.0	1.0	1.0
ER7P	1.0	1.0	1.0	1.0	1.0	1.0	1.0
ER3	1.0	1.0	1.0	1.0	1.0	1.0	
ER7	1.0	1.0	1.0	1.0	1.0	1.0	
SR3	1.0	1.0	1.0	1.0	1.0	1.0	
SR4	1.0	1.0	1.0	1.0	1.0	1.0	
SR5	1.0	1.0	1.0	1.0	1.0	1.0	

CR7	1.0	1.0	1.0	1.0	1.0	1.0	
CR12	1.0	1.0	1.0	1.0	1.0	1.0	
CR18	1.0	1.0	1.0	1.0	1.0	1.0	
CR20	1.0	1.0	1.0	1.0	1.0	1.0	

15.4.4.25 SetFcCartVelMax

説明	インピーダンス動作時の最大直交速度を設定します。
定義	SetFcCartVelMax vel_x, vel_y, vel_z, vel_a, vel_b, vel_c Vel_x 、データ型：double、X 方向の最大線速度、単位：m/s。 Vel_y 、データ型：double、Y 方向の最大線速度、単位：m/s。 Vel_z 、データ型：double、Z 方向の最大線速度、単位：m/s。 Vel_a 、データ型：double、X 方向の最大角速度、単位：rad/s。 Vel_b 、データ型：double、Y 方向の最大角速度、単位：rad/s。 Vel_c 、データ型：double、Z 方向の最大角速度、単位：rad/s。
例	SetFcJointVelMax 1.0,1.0,1.0,0.5,0.5,0.5

各協働ロボットの直交速度の上限、単位：m/s、rad/s

	Vel_x	Vel_y	Vel_z	Vel_a	Vel_b	Vel_c
ER3P	2.0	2.0	2.0	3.14	3.14	3.14
ER7P	2.0	2.0	2.0	3.14	3.14	3.14
ER3	2.0	2.0	2.0	3.14	3.14	3.14
ER7	2.0	2.0	2.0	3.14	3.14	3.14
SR3	2.0	2.0	2.0	3.14	3.14	3.14
SR4	2.0	2.0	2.0	3.14	3.14	3.14
SR5	2.0	2.0	2.0	3.14	3.14	3.14
CR7	2.0	2.0	2.0	3.14	3.14	3.14
CR12	2.0	2.0	2.0	3.14	3.14	3.14
CR17	2.0	2.0	2.0	3.14	3.14	3.14
CR18	2.0	2.0	2.0	3.14	3.14	3.14
CR20	2.0	2.0	2.0	3.14	3.14	3.14
CR25	2.0	2.0	2.0	3.14	3.14	3.14

15.4.4.26 SetFcJointMomentumMax

説明	インピーダンス動作時の関節の最大角運動量を設定します。
定義	SetFcJointMomentumMax jnt1_moment, jnt2_moment, jnt3_moment, jnt4_moment, jnt5_moment, jnt6_moment, jnt7_moment Jnt1_moment 、データ型：double、関節 1 の最大角運動量、単位：kg*m/s。 Jnt2_moment 、データ型：double、関節 2 の最大角運動量、単位：kg*m/s。 Jnt3_moment 、データ型：double、関節 3 の最大角運動量、単位：kg*m/s。 Jnt4_moment 、データ型：double、関節 4 の最大角運動量、単位：kg*m/s。 Jnt5_moment 、データ型：double、関節 5 の最大角運動量、単位：kg*m/s。 Jnt6_moment 、データ型：double、関節 6 の最大角運動量、単位：kg*m/s。 Jnt7_moment 、データ型：double、関節 7 の最大角運動量、単位：kg*m/s。
例	SetFcJointMomentumMax 0.1, 0.1, 0.1, 0.1, 0.055, 0.055, 0.055

各協働ロボットの各軸の角運動量の上限、単位：kg*m2/s

	J1	J2	J3	J4	J5	J6	J7
ER3P	0.1	0.1	0.1	0.1	0.055	0.055	0.055
ER7P	0.2	0.2	0.1	0.1	0.055	0.055	0.055
ER3	0.1	0.1	0.1	0.055	0.055	0.055	
ER7	0.2	0.2	0.1	0.055	0.055	0.055	
SR3	0.055	0.055	0.055	0.02	0.02	0.02	
SR4	0.1	0.1	0.055	0.02	0.02	0.02	
SR5	0.1	0.1	0.055	0.02	0.02	0.02	
CR7	0.2	0.2	0.1	0.055	0.055	0.055	
CR12	0.35	0.35	0.2	0.1	0.055	0.055	
CR17	0.35	0.35	0.2	0.1	0.055		
CR18	0.35	0.35	0.2	0.1	0.055	0.055	
CR20	0.7	0.7	0.35	0.2	0.1	0.1	
CR25	0.7	0.7	0.35	0.2	0.1		

15.4.4.27 SetFcJointEnergyMax

説明	インピーダンス動作時の関節の最大出力を設定します。
定義	SetFcJointEnergyMax jnt1_ energy, jnt2_ energy, jnt3_ energy, jnt4_ energy, jnt5_ energy, jnt6_ energy, jnt7_ energy Jnt1_energy 、データ型：double、関節 1 の最大パワー、単位：kg.m2/s3。 Jnt2_energy 、データ型：double、関節 2 の最大パワー、単位：kg.m2/s3。 Jnt3_energy 、データ型：double、関節 3 の最大パワー、単位：kg.m2/s3。 Jnt4_energy 、データ型：double、関節 4 の最大パワー、単位：kg.m2/s3。 Jnt5_energy 、データ型：double、関節 5 の最大パワー、単位：kg.m2/s3。 Jnt6_energy 、データ型：double、関節 6 の最大パワー、単位：kg.m2/s3。 Jnt7_energy 、データ型：double、関節 7 の最大パワー、単位：kg.m2/s3。
例	SetFcJointEnergyMax 100, 100, 100, 100, 100, 100, 100

各協働ロボットの各軸のパワーの上限、単位：kg*m2/s3

	J1	J2	J3	J4	J5	J6	J7
ER3P	250.0	250.0	250.0	250.0	150.0	150.0	150.0
ER7P	250.0	250.0	250.0	250.0	150.0	150.0	150.0
ER3	250.0	250.0	250.0	150.0	150.0	150.0	
ER7	250.0	250.0	250.0	150.0	150.0	150.0	
SR3	150.0	150.0	150.0	60.0	60.0	60.0	
SR4	250.0	250.0	150.0	60.0	60.0	60.0	
SR5	250.0	250.0	150.0	60.0	60.0	60.0	
CR7	400.0	400.0	300.0	200.0	100.0	100.0	
CR12	800.0	800.0	500.0	350.0	150.0	150.0	
CR17	800.0	800.0	500.0	350.0	250.0		
CR18	800.0	800.0	500.0	350.0	150.0	150.0	
CR20	1600.0	1600.0	800.0	450.0	250.0	250.0	

CR25	1600.0	1600.0	800.0	450.0	250.0		
------	--------	--------	-------	-------	-------	--	--

15.4.5 ドラッグ再生命令

15.4.5.1 ReplayPath

説明	ドラッグ教示で記録したパスを再生します。また、再生時の実行速度を制御します。 注意：ドラッグの記録速度が大きく、また再生速度が大きく設定されている場合は、サーボアームがトリガーされ機器に損傷を与える恐れがあります。低い速度から徐々に速度を上げることをお勧めします。
定義	ReplayPath path [, rate] [, wobj/tool] Path データ型：path 再生するパスを指定します。パス一覧で指定します。あらかじめ、ドラッグ教示を記録しておきます。 Rate データ型：double 再生速度を指定します。範囲は 0.01～3.00 です。0.01 は、ドラッグ時の実行速度の 1% で再生することを示す。1.00 は、100%の速度で再生することを示し、3.00 は 300%の速度で再生することを示します。 wobj/tool データ型：wobj/tool マルチモーション再生命令でのツールやワークを指定します。 再生中、ロボットは対応するツールに応じて再生の制御パラメーターを変更し、動作の安定性を高めます。
例	例 1 ReplayPath path , 1, tool1 実行速度 100%（オリジナルの速度）で再生します。

15.4.6 IO 命令

15.4.6.1 SetDO

説明	1 つのデジタル出力信号の値を設定します。この命令が動作命令の後に使用される場合は、滑らか軌跡を中断せず、動作命令の終点または滑らか軌跡エリアの始点でトリガーされます。詳細については例 2 を参照してください
定義	SetDO DoName, Value DoName データ型：signaldo 値を設定する DO 信号の名前を指定します。IO インターフェースですでに定義された変数でなければなりません。 Value データ型：bool DO 信号の目標状態を指定します。True と False のみをサポートします。
例	例 1 SetDO do2, true do2 に対応するデジタル出力を High に設定します。 例 2 ケース 1：滑らか軌跡エリアを持つ 2 つの動作命令の間で IO 命令を使用する場合 MoveL p1, v1000, z50, tool0

	<code>SetDO do2, true</code> <code>MoveL p2, v1000, z50, tool0</code> SetDO 命令は、p1 から p2 への滑らか軌跡エリアの始点でトリガーされます。 ケース 2：滑らか軌跡エリアを持つ動作命令の後に IO 命令を使用するが、その後に動作命令がない場合 <code>MoveL p1, v1000, z50, tool0</code> <code>SetDO do2, true</code> （後続の動作命令がないか、または滑らか軌跡を中断する命令がある） SetDO 命令は、p1 に到達した後にトリガーされます。 ケース 3：動作命令が滑らか軌跡エリアを持っていない場合 <code>MoveL p1, v1000, fine, tool0</code> <code>SetDO do2, true</code> （後続の命令に関係なく） SetDO 命令は、p1 に到達した後にトリガーされます。
--	--

15.4.6.2 SetAIIDO

説明	すべてのデジタル出力信号の値を設定します。この命令が動作命令の後に使用される場合は、滑らか軌跡を中断せず、動作命令の終点または滑らか軌跡エリアの始点でトリガーされます。詳細については、SetDO の例 2 を参照してください。
定義	SetAIIDO Value Value データ型：bool DO 信号の目標状態を指定します。True と False のみをサポートします。
例	例 1 <code>SetAIIDO true</code> システム機能に割り当てられている DO を除き、すべてのデジタル出力を High に設定します。

15.4.6.3 SetGO

説明	1 つのグループ出力の値を設定します。この命令が動作命令の後に使用される場合は、滑らか軌跡を中断せず、動作命令の終点または滑らか軌跡エリアの始点でトリガーされます。詳細については、SetDO の例 2 を参照してください。
定義	SetGO GoName, Value GoName データ型：signalgo 値を変更する go 信号の名前を指定します。IO インターフェースですでに定義された変数でなければなりません。 Value データ型：int go 信号の目標値を指定します。 注意：サポートされている最大値は 2147483648（2 の 31 乗）です。
例	例 1 <code>SetGO go3, 8</code> go3 に対応するグループの値を 8 に設定します。

15.4.6.4 SetAO

説明	1 つのアナログ出力の値を設定します。この命令が動作命令の後に使用される場合は、滑らか軌跡を中断せず、動作命令の終点または滑らか軌跡エリアの始点でトリガーされます。詳細については、SetDO の例 2 を参照してください。
定義	SetAO AoName, Value AoName データ型：signalao 値を設定する AO 信号の名前を指定します。IO インターフェースですでに定義された変数でなければなりません。 Value データ型：double ao 信号の目標値を指定します。
例	例 1 SetAO ao3, 5.123 ao3 に対応するアナログ出力を 5.123 に設定します。

15.4.6.5 PulseDO

説明	1 つのパルス DO 信号を生成します。この命令は動作命令の後に使用される場合は、滑らか軌跡を中断せず、動作命令の終点または滑らか軌跡エリアの始点でトリガーされます。詳細については、SetDO の例 2 を参照してください。
定義	PulseDO [High,] [length,] signal [*High] 現在の状態に関係なく、命令実行時に常に信号の状態を High にします。 [length] データ型：double または int パルス長を指定します (0.001-2000s)。指定しない場合のデフォルト値は 0.2s です。 signal データ型：signaldo パルスを生成する信号を指定します。
注意	PulseDO 中に SetDO/SetGO を実行すると、PulseDO は無効となり SetDO/SetGO が実行されます。

15.4.6.6 PulseReg

説明	1 つのレジスタを指定し、指定された時間のパルス信号を生成します。終了後にレジスタの値は初期値に戻ります。この命令が動作命令の後に使用される場合は、滑らか軌跡を中断せず、動作命令の終点または滑らか軌跡エリアの始点でトリガーされます。詳細については、SetDO の例 2 を参照してください。
定義	PulseReg Register, Value, Time Register データ型：bit bool レジスタ パルス信号を生成するレジスタ名を指定します。 Value データ型：bool パルス信号の値を指定します。 Time データ型：double パルス信号の継続時間を指定します。単位は秒で、制限範囲は[0.001, 10.0]です。
注意	PulseReg 実行中に WriteRegByName またはレジスタ代入を実行すると、レジスタの実効値は

最後に実行された命令に従います。PulseReg 命令で指定された時間が経過すると、PulseReg を実行する前の初期値に戻ります。

15.4.7 通信命令

RL プログラムでは、ロボットが周辺機器と通信する方法として、イーサネット通信（TCP クライアントと TCP サーバー）とシリアルポート通信の 2 つの方法をサポートします。リソース管理とデータ送受信のための統一された命令セットが設計されており、一貫した操作性を提供しています。

命令セット	TCP クライアント	TCP サーバー	シリアルポート通信
OpenDev	Y	Y	Y
SocketAccept	N/A	Y	N/A
CloseDev	Y	Y	Y
SendString	Y	Y	Y
SendByte	Y	Y	Y
ReadBit	Y	Y	Y
ReadByte	Y	Y	Y
ReadDouble	Y	Y	N/A
ReadInt	Y	Y	N/A
ReadString	Y	Y	Y
GetSocketConn	Y	N/A	N/A
GetSocketServer	N/A	Y	N/A
GetBufSize	N/A	N/A	Y
ClearBuffer	N/A	N/A	Y

15.4.7.1 OpenDev

説明	パラメーターで指定される名前に応じて、待ち受けサーバーのオープン処理、クライアントとしての接続開始、シリアルポートのオープン処理を実行します。 TCP サーバーの名前を指定すると、ロボットはサーバーとしてポートをバインドし待ち受け状態になります。 TCP クライアントとしての名前を指定すると、ロボットは TCP クライアントとしてあらかじめ設定された IP アドレスとポート番号に従って外部サーバーへの接続を試みます。 シリアルポートの名前を指定すると、シリアルポートを初期化し通信条件を整えます。
定義	OpenDev(name) name データ型 : string クライアント、サーバー、シリアルポートの名前を指定します。
例	例 1 <pre>SocketConn scnn3 = {"192.168.0.200", 8090, "clt1", 2, "\n"} Try OpenDev("clt1") // リモートサーバへの接続を試みる。接続に成功すると、clt1 のプロパティは自動的に outgoing に変更される。 string readstr = ReadString(30, "clt1") //readstr の論理処理 string sendstr = "hello server !" SendString(sendstr , "clt1") //clt1 のクライアント接続を使用してデータを送信する ... // 一連のコード</pre>

	<pre> Catch(ERROR e) // ERROR エラーの種類には、エラーが発生したファイル、行番号、エラーコード、エラーの内容などが含まれる ... //一連の例外処理 Endtry 例 2 SocketServer listener1 = {"192.168.0.200", 8090, "svr1"} global pers bool exit = false Try OpenDev("svr1") //ポートをバインドし通信を待ち受ける while(exit != true) SocketConn conn = SocketAccept("svr1") //ブロッキング受信で接続されたクライアント Endwhile Catch(ERROR e) ... //一連の例外処理 Endtry </pre> エラーが発生すると、制御システムは例外を出すため、エラーの原因を捕捉できます。例外が Try/Catch 命令で捕捉されない場合は、制御システムはプログラムを停止します。
--	---

15.4.7.2 SocketAccept

説明	クライアントからの接続を待ち、クライアント接続処理を完了します。この命令は、ロボットがサーバーとして動作する場合にのみ使用されます。
定義	戻り値 データ型：socketConn 周辺機器が TCP クライアントとしてロボットに接続した後、通信の読み書きを制御するために、制御システムは通信オブジェクトを生成します。 SocketConn conn = SocketAccept(name) name データ型：string サーバーとしての名前を指定します。事前に OpenDev を使用して正常にオープンしておく必要があります。
例	<pre> 例 1 SocketServer listener1 = {"192.168.0.200", 8090, "svr1"} global pers bool exit = false Try OpenDev("svr1") //ポートをバインドし接続を待ち受ける while(exit != true) SocketConn conn = SocketAccept("svr1") //ブロッキング受信で接続されたクライアント conn.name = "client1" // 重要！通信コネクションに名前をつけないと、その後に名前によるデータの読み書きが難しくなる conn.suffix = "¥n" //オプション、パケットのサフィックスを設定する Endwhile Catch(ERROR e) ... //一連の例外処理 Endtry </pre> エラーが発生すると、制御システムは例外を出すため、エラーの原因を捕捉できます。例外が Try/Catch 命令で捕捉されない場合は、制御システムはプログラムを停止します。
注意	本命令は現在のタスクをブロックします。マルチタスクで優先度の低いタスクとして使用する必要があります。 本命令は、プログラムの他の部分で利用できるクライアントの接続 IP アドレスとポート番号情報

	報を持つ SocketConn 構造体を返します。返される構造体の名前は、システムによりランダムに割り当てられます。構造体を取得した後に、名前を変更してコネクションロストを回避してください。 サーバーの 1 対多接続をサポートします。
--	---

15.4.7.3 CloseDev

説明	TCP クライアント、TCP サーバー、シリアルポートを閉じます。
定義	CloseDev(name) name データ型: string クライアント、サーバー、シリアルポートの名前を指定します。
例	例 1 <pre>SocketConn scnn3 = {"192.168.0.200", 8090, "clt1", 2, "\n"} Try OpenDev("clt1") string readstr = ReadString(30, "clt1") //readstr の論理処理 string sendstr = "hello server !" SendString (sendstr , "clt1") //clt1 のクライアント接続を使用してデータを送信する ... // 一連のコード Catch(ERROR e) ... //一連の例外処理 endtry CloseDev("clt") //エラーの有無に関わらず、最後に socket クライアントを閉じる。</pre> 例 2 <pre>SocketServer listener1 = {"192.168.0.200", 8090, "svr1"} global pers bool exit = false Try OpenDev("svr1") //ポートをバインドし接続を待ち受ける while(exit != true) SocketConn conn = SocketAccept("svr1") //ブロッキング受信で接続されたクライアント conn.name = "client1" // 重要！通信コネクションに名前をつけないと、その後に名前によるデータの読み書きが難しくなる conn.suffix = "¥n" //オプション、サフィックスを設定する Endwhile Catch(ERROR e) ... //一連の例外処理 Endtry CloseDev("client1") //外部 TCP クライアントとの通信をオフにする。重要！ CloseDev("svr1") //待ち受けサーバーをオフにする</pre> 例 2 では、2 つのネットワークオブジェクトがあります。リソース解放が不完全な状態（TCP TIME_WAIT 状態）を避けるために、サーバーオブジェクトを閉じる前に、通信コネクションを閉じる必要があります。 ロボットがサーバーとして動作しているときに、すでに外部との複数の通信コネクションが確立されている場合は、サーバーを閉じる前にこれらの通信コネクションを閉じる必要があります。 不完全なリソース解放が発生した場合、復旧するには制御システムを再起動する必要があります。制御システムで許容されるリソースの数には冗長性があり、少数のリソースが占有されて

	も正常に動作できることが保証されていますが、誤った使用で多数のリソースが占有されることを避ける必要があります。
--	---

15.4.7.4 SendString

説明	文字列を外部に送信します。パラメーターに応じて、ネットワーク経由またはシリアルポート経由で送信します。
定義	SendString(StringData, name) StringData データ型：string 送信する文字列データを指定します。 name データ型：string データ送信に使用する通信の名前を指定します。確立された TCP 通信コネクションの SocketConn 、または正常にオープンされたシリアルポートのいずれかを指定します。
例	例 1 <pre>SendString("Hello World", "Socket0")</pre> Socket0 で Hello World 文字列を外部に送信します。ここで、Socket0 は事前に定義され、接続に成功した SocketConn です。 例 2 <pre>VAR String str1 ="Hello World" SocketSendString(str1, "Serial1")</pre> str1 に格納された Hello World 文字列を Serial1 で外部に送信します。ここで、Serial1 は事前に定義され、オープンされたシリアルポートです。

15.4.7.5 SendByte

説明	1 バイトのデータを外部に送信します。ASCII 文字を送信する場合に便利です。
定義	SendByte(ByteData, name) ByteData データ型：int, byte, byte 配列 0~255 の符号なしバイトまたは配列を指定します。主に ASCII コードを送信するために使われます。 name データ型：string データ送信に使用する通信の名前を指定します。確立された TCP 通信コネクションの SocketConn 、または正常にオープンされたシリアルポートのいずれかを指定します。
例	ASCII で 13 はキャリッジリターンを意味します。 例 1 <pre>SendByte(13, "socket0")</pre> socket0 でキャリッジリターンを外部に送信します。 例 2 <pre>VAR byte data1 = 13 SendByte(data1, "serial0")</pre> まず byte 型変数 data1 を定義し 13 に初期化します。次に、serial0 で data1 を送信します。 例 3 <pre>VAR byte data2[2] = {13,17} SendByte(data2, "socket0")</pre>

	socket0 で byte 型の配列である data2 を送信します。この場合、配列の内容全体を送信します。 例 4 <pre>VAR byte data2[2] = {13,17,20} SendByte(data2[2], "socket0")</pre> socket0 で byte 型変数を送信します。data2[2] は配列の 2 番目の要素のみを表すので、この場合は、data2[2] の値 17 のみを送信し他の要素は送信しません。
--	--

15.4.7.6 ReadBit

説明	bit 単位でデータを受信します。 1) ネットワーク通信の場合は、TCP 通信で受信し、外部から送信されるデータは SocketConn で設定されたサフィックスで終了します。 2) シリアル通信の場合は、周辺機器はデータ部分のみを送信します。サフィックスを指定する必要はありません。
定義	戻り値 データ型：bool 配列 データを bool 型配列で受信します。各 bit は 1 つの bool メンバーに対応しています。 Ret = ReadBit(BitNum, TimeOut, name) BitNum データ型：int、 読み込む bit 数を指定します。サイズは 8 の整数倍である必要があります。 TimeOut データ型：int タイムアウトを指定します。単位は s で、範囲は 0～86400 です。デフォルトは 60s です。 name データ型：string SocketConn またはシリアルポートの名前を指定します。 Ret データ型：bool 配列 受信するデータを格納する変数です。配列の最初の要素は最下位ビットを表します。
例	例 1 <pre>bool groupio[16] groupio = ReadBit(16, 60, "Socket0")</pre> ReadBit 命令で 16bit のデータを読み込み、groupio という bool 配列に格納します。タイムアウトは 60s です。 周辺機器が ASCII 文字の 95+サフィックスを送信した場合は、ロボットは「95」を受信します。ASCII 文字の「9」は 16 進数で 0x39 であり、「5」は 16 進数で 0x35 であるため、ユーザーが受信するデータは 0x3935 となります。このとき、groupio 配列は[1]～[16]が、1001 1100 1010 1100 となります。[1]はデータ下位ビットであり、0x3935 に対応します。

15.4.7.7 ReadByte

説明	一定バイト数のデータを受信します。データ間はカンマで区切る必要があります。
定義	戻り値 データ型：byte 配列 データを byte 型配列で受信します。 Ret = ReadByte(ByteNum, TimeOut, name) ByteNum データ型：int

	読み込むバイト数を指定します。 TimeOut データ型：int タイムアウトを指定します。単位は s で、範囲は 0～86400 です、デフォルトは 60s です。 name データ型：string SocketConn またはシリアルポートの名前を指定します。 Ret データ型：byte 配列 受信するデータを格納する変数です。
例	例 1 <pre>byte rets[6] = {0,0,0,0,0,0} rets = ReadByte(6,60,"clt1")</pre> 6byte のデータを読み込み、rets という byte 型配列に格納します。タイムアウトは 60s です。
注意	「1,2,3,4,5,6」のように、周辺機器からのデータ間はカンマで区切る必要があります。 TCP ネットワーク通信の場合は、データの後にあらかじめ設定されたサフィックスを付加する必要があります。 シリアルポート通信の場合は、サフィックスは必要ありません。

15.4.7.8 ReadDouble

説明	Socket 経由で Double 型データを受信します。外部から送信されるデータは設定されたサフィックスで終了する必要があります。 注意：この命令は TCP ネットワーク通信に対してのみ有効です。ロボットがクライアントもしくはサーバーとして動作する場合には適用されますが、シリアルポート通信の場合には適用されません。
定義	戻り値 データ型：double 配列 データを double 型配列で受信します。 Ret = ReadDouble(DoubleNum, TimeOut, name) DoubleNum データ型：double 読み込む double 型データの個数を指定します。最大は 30 です。 TimeOut データ型：int タイムアウトを指定します。単位は s で、範囲は 0～86400 です。デフォルトは 60s です。 name データ型：string SocketConn の名前を指定します。
例	例 1 <pre>double dd[10] dd = ReadDouble(10, 60, "Socket0")</pre> 10 個の double 型データを受信し、dd という double 型配列に格納します。タイムアウトは 60s です。

15.4.7.9 ReadInt

説明	Socket 経由で int 型データを受信します。外部から送信されるデータは設定されたサフィックスで終了する必要があります。 注意：この命令は TCP ネットワーク通信に対してのみ有効です。ロボットがクライアントもしくはサーバーとして動作する場合には適用されますが、シリアルポート通信の場合には適用されません。
定義	戻り値 データ型：int 配列 データを int 型配列で受信します。 Ret = ReadInt(IntNum, TimeOut, name) IntNum データ型：int 読み込む int 型データの個数を指定します。最大は 30 です。 TimeOut データ型：int タイムアウトを指定します。単位は s で、範囲は 0～86400 です。デフォルトは 60s です。 name データ型：string SocketConn の名前を指定します。
例	例 1 <pre>int ii[10] ii = ReadInt(10, 60, "Socket0")</pre> 10 個の int 型データを受信し、ii という int 型配列に格納します。タイムアウトは 60s です。

15.4.7.10 ReadString

説明	文字列データを受信します。外部から送信されるデータは設定されたサフィックスで終了する必要があります。
定義	戻り値 データ型：string データを string 型変数に受信します。 Ret = ReadString(TimeOut, name, [len]) TimeOut データ型：int タイムアウトを指定します。単位は s で、範囲は 0～86400 です。デフォルトは 60s です。 name データ型：string SocketConn またはシリアルポートの名前を指定します。 [len] データ型：int オプションのパラメーターです。シリアルポート読み込みを使用する場合にのみ使用します。シリアルポートではサフィックスの概念がないため、正しく読み込むために長さを指定する必要があります。
例	例 1 <pre>VAR String str1 str1 = ReadString(60, "Socket1")</pre> Socket1 で 1 つの文字列を受け取り、str1 という string 型変数に格納します。タイムアウトは 60s です。TCP ネットワーク通信方式での指定方法です。

	VAR String str1 str1 = ReadString(60, "serial0",5) serial0 で長さ 5 の文字列を受け取り、 str1 に格納します。タイムアウトは 60s です。シリアルポート通信方式での指定方法です。
--	---

15.4.7.11 GetSocketConn

説明	Socket 接続名から、対応する Socket オブジェクトを検索します。この命令によって取得された結果は、論理の判定や処理に使用できます。読み取り専用としてのみ使用できます。この命令は、通信コネクション（ロボットがクライアントまたは接続済みのサーバーの場合）にのみ適用され、待ち受け状態のサーバーとシリアル通信には適用されません。		
定義	戻り値 データ型：SocketConn SocketConn の名前で検索された Socket に関する情報の構造体を返します。		
	Ret = GetSocketConn(name) name データ型：string 通信コネクションの SocketConn の名前を指定します。		
	Ret データ型：SocketConn SocketConn の名前で検索された Socket に関する情報の構造体です。		
	検索可能な情報	検索方法	意味と例
	IP アドレス	ret.ip	文字列（例：192.168.0.161）
	ポート番号	ret.port	整数（例：8090）
	プロパティ	ret.attr	ロボットがサーバーとして動作する場合：“incoming” ロボットがクライアントとして動作する場合：“outgoing” 接続が確立されていない場合： ""またはその他の値、通常は空
	キャッシュサイズ	ret.cache	1～100
	名前	ret.name	例：「client0」
	接続状態	ret.state	closed、establish
例	例 1 SocketConn ret= GetSocketConn(“client0”) 「client0」という名前の SocketConn オブジェクトを検索します。Ret に、IP アドレス、ポート番号、接続状態などのこの接続の情報を取得できます。		

15.4.7.12 GetSocketServer

説明	SocketServer の名前から、対応する SocketServer オブジェクトを検索します。この命令によって取得された結果は論理の判定や処理に使用できます。読み取り専用としてのみ使用できます。この命令は、待ち受けサーバー（SocketServer オブジェクト）にのみ適用され、通信コネクション（ロボットがクライアントまたは接続済みのサーバーの場合）とシリアル通信には適用されません。
定義	戻り値 データ型：socketServer SocketServer の名前で検索された Socket に関する情報の構造体を返します。

	接続の名前を指定します。
例	サフィックス <code>¥r</code> であると仮定すると、すでに受信済みの 2 件のデータ と、受信中の 1 件のデータがある状態でこの指令を実行すると、バッファ内のすべてのデータがクリアされます。そのため、RL プログラムは、クリア後に再送信されたデータのみを読み取ることができます。 <pre>123456789¥r Abcdefg¥r mmmmmm</pre>

15.4.7.15 ReadOpcUaVarByName

説明	OPC-UA カスタム変数の値を、名前を指定して読み込みます。
定義	ReadOpcUaVarByName(name, value) name データ型：string OPC-UA カスタム変数名を指定します。 value データ型：bool/byte/int/double/string 読み込んだ OPC-UA カスタム変数の値を格納します。value の型が OPC-UA 変数の型と異なる場合は、自動的に型変換されます。 注意：string 型は数値型に変換されると、常に 0 になります。 戻り値はありません。
例	例 1 <pre>int value = 0 ReadOpcUaVarByName("int_var", value) print(value)</pre>

15.4.7.16 WriteOpcUaVarByName

説明	OPC-UA カスタム変数の値を、名前を指定して変更します。
定義	WriteOpcUaVarByName(name, value) name データ型：string OPC-UA カスタム変数名を指定します。 value データ型：bool/byte/int/double/string OPC-UA カスタム変数の変更値を格納します。value の型が OPC-UA 変数の型と異なる場合は、自動的に型変換されます。 注意：String 型は数値型に変換されると、常に 0 になります。
例	例 1 <pre>int value = 0 WriteOpcUaVarByName("int_var", value) WriteOpcUaVarByName("int_var", 123)</pre>

15.4.8 ネットワークコマンド

15.4.8.1 SocketCreate (サポート終了)

説明	Socket 接続が確立すると、RL プログラムは Socket 命令を使用して周辺機器からデータを受信したり、外部にデータを送信したりすることができます。RL 言語では、複数の周辺機器に接続するために複数の異なる Socket を同時に作成することができます。また、異なる Socket は異なる名前で区別されます。Socket 命令は TCP/IP プロトコルに基づいているため、TCP/IP をサポートする周辺機器であれば、RL プログラムと通信してデータをやり取りすることができます。 RL Socket に送信されるデータ (SocketRead シリーズの命令で受信するデータ) は、すべて「キャリッジリターン」で終了する必要があります。「キャリッジリターン」を受信する前のデータは、すべて同じデータとして結合処理されます。Socket 機能を使用する場合は、ロボットコントローラーはクライアントとして外部サーバーへの接続のみをサポートします。 Socket 接続は最大 10 個まで作成可能です。 注意：この命令は「サポート終了」となっています。xCore 制御システムのバージョン 1.3 で使用されていたものであり、それ以降のバージョンでも有効ですが、メンテナンスは終了しており継続使用は推奨されません。
定義	戻り値 データ型：bool 通信 Socket の作成が成功した場合は True、失敗した場合は False を返します。 SocketCreate ("ip_Address", Port, "Name", [Cache] [, "Terminator"]) ip_Address データ型：string 接続先のサーバーの IPv4 アドレスを二重引用符で囲んで指定します。 Port データ型：int サーバーのポート番号を指定します。 Name データ型：string Socket の名前を定義します。異なる Socket は異なる名前で区別されます。 Cache データ型：int Socket のキャッシュサイズを定義します。通信データはキャッシュに保存されます。省略することもできます。 Terminator データ型：string Socket 通信のサフィックスを定義します。省略可能です。デフォルトは"¥r"です。
例	例： <pre>if (SocketCreate("10.0.6.11",8080,"S1",10,"¥r")) // 作成に成功 else // エラー処理 endif</pre>
注意	TCP/IP プロトコルのリソース解放メカニズムの制限により、プログラムの実行エラーを避けるために、SocketCreate 命令と SocketClose 命令を頻繁に呼び出さないでください。 ループモードでは、SocketCreate 命令と SocketClose 命令を頻繁に呼び出さないように、2 つの命令の間に待ち時間を追加することをお勧めします。 例：

	<pre>SocketClose("S1") wait 0.1 SocketCreate("10.0.6.11",8080,"S1",10,"r")</pre>
--	--

15.4.8.2 SocketClose（サポート終了）

説明	Socket 接続を閉じます。 注意：この命令は「サポート終了」となっています。xCore 制御システムのバージョン 1.3 で使用されていたものであり、それ以降のバージョンでも有効ですが、メンテナンスは終了しており継続使用は推奨されません。
定義	SocketClose ("SocketName") SocketName データ型：string クローズする Socket の名前を指定します。
例	例 1 <pre>SocketClose ("Socket0")</pre>
注意	SocketSend シリーズの命令の直後に SocketClose 命令を使用しないでください。確認メッセージを受信してから、SocketClose 命令を使用してください。

15.4.8.3 SocketSendString（サポート終了）

説明	Socket を介して、外部に文字列を送信します。 注意：この命令は「サポート終了」となっています。xCore 制御システムのバージョン 1.3 で使用されていたものであり、それ以降のバージョンでも有効ですが、メンテナンスは終了しており継続使用は推奨されません。
定義	SocketSendString (StringData, "SocketName") StringData データ型：string 送信する文字列データを指定します。 SocketName データ型：string データを送信する Socket の名前を指定します。
例	例 1 <pre>SocketSendString ("Hello World", "Socket0")</pre> Socket0 で Hello World という文字列を外部に送信します。 例 2 <pre>VAR String str1 ="Hello World" SocketSendString (str1, "Socket0")</pre> Socket0 で str1 に格納されている文字列を外部に送信します。

15.4.8.4 SocketSendByte（サポート終了）

説明	Socket を介して、外部に byte データを送信します。ASCII 文字を送信する場合に便利です。 注意：この命令は「サポート終了」となっています。xCore 制御システムのバージョン 1.3 で使用されていたものであり、それ以降のバージョンでも有効ですが、メンテナンスは終了しており継続使用は推奨されません。
定義	SocketSendByte(ByteData, "SocketName") ByteData データ型：int、byte、byte 配列

	0～255 の符号なしバイトまたは配列を送信します。主に ASCII コードを送信するために使用されます。 SocketName データ型：string データを送信する Socket の名前を指定します。
例	ASCII で 13 はキャリッジリターンを意味します。 例 1 <code>SocketSendByte(13, "socket0")</code> socket0 でキャリッジリターンを外部に送信します。 例 2 <code>VAR byte data1 = 13</code> <code>SocketSendByte(data1, "socket0")</code> まず byte 型変数 data1 を 13 に初期化します。次に、socket0 で data1 を外部に送信します。 例 3 <code>VAR byte data2[2] = {13,17}</code> <code>SocketSendByte(data2, "socket0")</code> socket0 で byte 型の配列である data2 を送信します。

15.4.8.5 SocketReadBit（サポート終了）

説明	Socket を介して bit データを受信します。外部から送信されるデータはキャリッジリターンで終了する必要があります。 注意：この命令は「サポート終了」となっています。xCore 制御システムのバージョン 1.3 で使用されていたものであり、それ以降のバージョンでも有効ですが、メンテナンスは終了しており継続使用は推奨されません。
定義	戻り値 データ型：bool 受信する bit データを Bool 型配列に格納します。各 bit は 1 つの bool メンバーに対応しています。 SocketReadBit(BitNum, TimeOut, "SocketName") BitNum データ型：int 読み込む bit 数を指定します。サイズは 8 の整数倍である必要があります。 TimeOut データ型：int タイムアウトを指定します。単位は s で、範囲は 0～86400 です。デフォルトは 60s です。 SocketName データ型：string データを受信する Socket の名前を指定します。
例	例 1 <code>bool groupio[16]</code> <code>groupio = SocketReadBit(16, 60, "Socket0")</code> SocketReadBit 命令で 16bit のデータを読み込み、groupio という bool 型配列に格納します。タイムアウトは 60s です。

15.4.8.6 SocketReadDouble（サポート終了）

説明	Socket を介して Double 型データを受信します。外部から送信されるデータはキャリッジリターンで終了する必要があります。 注意：この命令は「サポート終了」となっています。xCore 制御システムのバージョン 1.3 で使用されていたものであり、それ以降のバージョンでも有効ですが、メンテナンスは終了しており継続使用は推奨されません。
定義	戻り値 データ型：double 受信データを double 型配列に格納します。 SocketReadDouble(DoubleNum, TimeOut, "SocketName") DoubleNum データ型：double 読み込む double 型データの個数を指定します。最大は 30 です。 TimeOut データ型：int タイムアウトを指定します。単位は s で、範囲は 0～86400 です。デフォルトは 60s です。 SocketName データ型：string データを受信する Socket の名前を指定します。
例	例 1 <pre>double dd[10] dd = SocketReadDouble(10, 60, "Socket0")</pre> SocketReadDouble 命令で 10 個の double 型のデータを読み込み、dd という double 型配列に格納します。タイムアウトは 60s です。

15.4.8.7 SocketReadInt（サポート終了）

説明	Socket を介して Int 型データを受信します。外部から送信されるデータはキャリッジリターンで終了する必要があります。 注意：この命令は「サポート終了」となっています。xCore 制御システムのバージョン 1.3 で使用されていたものであり、それ以降のバージョンでも有効ですが、メンテナンスは終了しており継続使用は推奨されません。
定義	戻り値 データ型：int 受信データを int 型配列に格納します。 SocketReadInt(IntNum, TimeOut, "SocketName") IntNum データ型：int 読み込む int 型データの個数を指定します。最大は 30 です。 TimeOut データ型：int タイムアウトを指定します。単位は s で、範囲は 0～86400 です。デフォルトは 60s です。 SocketName データ型：string データを受信する Socket の名前を指定します。
例	例 1 <pre>int ii[10] ii = SocketReadInt(10, 60, "Socket0")</pre>

	SocketReadInt 命令で 10 個の Int 型のデータを読み込み、ii という Int 型配列に格納します。タイムアウトは 60s です。
--	---

15.4.8.8 SocketReadString（サポート終了）

説明	Socket を介して文字列を受信します。外部から送信されるデータはキャリッジリターンで終了する必要があります。 注意：この命令は「サポート終了」となっています。xCore 制御システムのバージョン 1.3 で使用されていたものであり、それ以降のバージョンでも有効ですが、メンテナンスは終了しており継続使用は推奨されません。
定義	戻り値 データ型：string 受信データを文字列型変数に格納します。 SocketReadString(TimeOut, "SocketName") TimeOut データ型：int タイムアウトを指定します。単位は s で、範囲は 0～86400 です。デフォルトは 60s です。 SocketName データ型：string データを受信する Socket の名前を指定します。
例	例 1 <code>VAR String str1</code> <code>str1 = SocketReadString(60, "Socket1")</code> SocketReadString 命令で文字列データを受け取り str1 に格納します。タイムアウトは 60s です。

15.4.9 論理命令

15.4.9.1 Return

説明	関数を終了する時に使用します。 RETURN 命令を実行すると、そのプログラムがサブ関数の場合は、上位関数に戻ります。プログラムが現在メイン関数の場合は、プログラムは終了します。
----	--

15.4.9.2 Wait

説明	一定時間（0～2147484 秒）待機します。
例	例 1 <code>Wait 2</code> 2 秒間待機します。

15.4.9.3 WaitUntil

説明	ある条件が成立するまで待機します。タイムアウトするとタイムアウトフラグを True に設定し、待機を終了して実行を継続します。
定義	WaitUntil(cond, \MaxTime, \TimeFlag) cond bool 型論理式で指定します。 MaxTime タイムアウト時間を指定します。オプションのパラメーターです。単位は s です。 int または double 型を使用します TimeFlag タイムアウトフラグを格納します。タイムアウトすると True になります。オプションのパ

	ラメーターです bool 型の変数を使用します。
例	例 1 <pre>WaitUntil (di2 == true)</pre> ... di 2 信号の値が true になるまで待機します。条件が成立すると、後続のプログラムを実行します。 例 2 <pre>WaitUntil (di2 == true,5)</pre> ... di 2 信号の値が true になるまで待機します。5 秒以上待機しても di 2 信号が false の場合は、条件が成立しなくても後続のプログラムを実行します。 例 3 <pre>Bool flag = false WaitUntil (di2 == true, 5,flag)</pre> ... di 2 信号の値が true になるまで待機します。5 秒以上待機しても di 2 信号が false の場合は、条件が成立しなくても後続のプログラムを実行します。その際、flag を true にします。 5 秒以内に di2 が true になった場合は、flag は false になり、後続のプログラムを実行します。flag はその後の論理判断に使用できます。

15.4.9.4 Break

説明	現在のループから脱出します。RL 言語では WHILE ループ及び FOR ループで使用されます。ループ内で Break が実行されると、状態に関係なくループから脱出します。
例	例 1 <pre>VAR int counter = 0 WHILE(1) IF(counter == 5) break Endif counter++ ENDWHILE</pre> counter が 5 になると WHILE ループから脱出します。

15.4.9.5 IF...Else if...Else

説明	条件判断文
例	例 1 <pre>IF(condition1) //a Else if (condition2) //b Else if (condition3) //c Else //d Endif</pre> condition1 が成立すると論理 a が実行され、condition2 が成立すると論理 b が実行されます。以下同様です。

15.4.9.6 Goto

説明	ポインタを目的の行に移動させます。
例	例 1 <pre>int a = 0 int b = 9 Goto end print(a) end: print(b)</pre> まず 2 つの変数 a と b を定義します。Goto 命令で end の行に移動します。そして後続のプログラムで b を表示します。Goto により end の位置に移動するので、a を表示する print 命令は実行されません。

15.4.9.7 For

説明	指定した回数だけループを実行します。
例	例 1 <pre>For(int i from 1 to 10) printf("i = %d\n", i) endfor</pre> このプログラムは、i を 1 から 9 まで 1 ずつ加算して順番に 9 回表示します。 例 2 <pre>For(int i from 1 to 10 step 3) printf("i = %d\n", i) Endfor</pre> このプログラムは、i を 1 から 9 まで 3 ずつ加算して順番に 3 回表示します。 補足説明： Continue と Break は、For の処理を制御するために使用できます。詳細については、Continue と Break 命令の説明を参照して下さい。

15.4.9.8 Continue

説明	ループ内の後続の処理をスキップし、次のループ処理を行います。
例	例 1 <pre>VAR int count = 0 WHILE(1) count++ IF(count == 1) Continue Else break MoveAbsJ j10, v500, fine, tool1 Endif ENDWHILE</pre> MoveAbsJ は実行されません。

15.4.9.9 Inzone

説明	この命令は、SetDO や modbus、cclink などの IO 操作や命令と連携して使用されます。信号が決められたポイントでトリガーされ、先読みによってトリガーされないようにします。
例	<pre>MoveL p1 MoveL p2 Inzone SetDO dox, true print(123) EndInzone</pre>

	MoveL p3 補足説明： Inzone は先読みされた後すぐには実行されず、SetDO と print 命令を含む追加関数を生成します。この追加関数は、動作命令 MoveL p2 が完了した後に有効になります。 1. p2 と p3 の 2 つの動作命令の間に滑らか軌跡エリアが存在する場合は、追加関数はロボットが滑らか軌跡エリアに入った時点で実行を開始します。 2. 滑らか軌跡エリアがない場合は、追加関数はロボットが p2 に到達した時点で実行を開始します。
--	---

15.4.9.10 While

説明	条件が満たされるまでループを実行し続けます。
例	例 1 <pre>int count = 0 while(count < 10) count++ print(count) endwhile</pre> このプログラムは、count を 1 から 10 まで 1 ずつ加算して表示するループを実現します。 Continue と Break は、While の処理を制御するために使用できます。詳細については、Continue と Break 命令の説明を参照して下さい。

15.4.9.11 Pause

説明	プログラムの実行を一時停止します。 Pause 命令の前の命令文が実行された後、プログラムは一時停止します。プログラムを再開するには、教示操作盤で実行をタップするか、外部プログラム起動信号を使用する必要があります。
注意	この命令は現時点ではプログラミング補助機能での入力をサポートしていません。直接入力してください。

15.4.9.12 Try/Catch

説明	RL 言語のエラー処理メカニズムです。 Try 命令と Catch 命令の間でエラーが発生した際に、プログラムはエラーをエラーメッセージの集合「 e 」に変換し、 Catch 命令以下のコードブロックから実行を続けます。
定義	<pre>try // do something catch(error e) print(e) endtry</pre> 例えば、ネットワークからのデータの読み取りは失敗する可能性が高い命令です。エラーでロボットを停止させないように、Try/Catch でエラーを捕捉して処理できます。 詳細 error 型の説明 error は構造体であり、次の 4 つのパラメーターで構成されます。 ● file : string (エラーが発生したファイル名) ● line : int (エラー行) ● num : int (エラーコード) ● reason : string (エラーの原因) error 構造体は print 命令によって表示することができます。

	<pre>// error data ... catch(error e) print(e.file) print(e.line) print(e.num) print(e.reason) print(e) endtry</pre>
例	例 1 <pre>ReadOnce: Try Double xyz[3] = ReadDouble(3, timeout, socketname) Robtarget_0.trans.x = xyz[1] Robtarget_0.trans.y = xyz[2] Robtarget_0.trans.z = xyz[3] MoveL Robtarget_0, v2000, fine, tool0 Catch(error e) SendString("Recv rob xyz error", socketname) Goto ReadOnce endtry</pre> このプログラムは、通信命令 ReadDouble を使用して Socket から 3 次元配列を位置 xyz パラメーターとして読み込み、次に MoveL 命令を使用して対応する直交位置まで動作するというシナリオを表しています。 Try/Catch 命令を使用せずに、Socket から受信した位置が正しくないデータであった場合は、ロボットは「動作範囲外」または「計画エラー」を報告し、プログラムの実行を停止します。 Try/Catch 命令を使用した場合は、動作命令のエラーを報告後、プログラムは停止せず Catch 以下のコードブロックにジャンプし、ユーザーが望むエラー処理を実行します。本例では、ReadDouble で受信したデータのエラーをホストコンピューターに伝え、goto 命令を実行して ReadOnce にジャンプします。 例 2 <pre>re_read: try opendev("conn_name") string_res = readstring("conn_name") catch (error e) if (e.num == xxx) // 処理可能なエラー、一時停止なし goto re_read else print(e) Pause endif endtry</pre>
注意	注意：力制御命令は Try/Catch 命令をトリガーできません。

Try/Catch が処理できるエラーの種類と標準エラーコード：

分類	エラー発生命令	説明	error.num	error.reason
デフォルトエラー		専用エラーコードが割り当てられていない命令	-1	未知のエラー
シリアルポート関連命令		存在しない場合	-1	未知のエラー
動作関連命令	MoveXX、Search、TrigL 等 AccRamp、HomeSet 等の動作パラメータの設定	動作座標のツール、ワークのエラー 動作速度エラー 動作負荷エラー 動作範囲外 計画エラー 特異点などに遭遇	-1	未知のエラー
ネットワークコマンド	OpenDev	ネットワークリンクのポートエラー	-1	未知のエラー
	すべてのネットワークコマンド	RL を介して外部通信との接続を操作する	-1	未知のエラー
計算、論理命令	CalcJoinT CalcRobt CRobT CJointT CLKSTOP GOTO	コントローラー内部エラー	-1	未知のエラー
周辺機器制御 (Jodell シリーズ) (RM シリーズ)	JodellGripInit JodellSuckInit JodellSuckStatus RMRGMGripPosMove RMRGMGripTrqMove RMRGMGripStatus RMRGMResetErr RMCGripPosMove RMCGripTrqMove RMCGripStatus RMCResetErr RMRGMGripInit RMCGripInit	周辺機器の通信異常	-1	未知のエラー
レーザー制御	Laser のすべての命令	レーザー溶接を OFF にした	-1	未知のエラー
スタック制御	TrayUpdate TrayCount PalletUpdate PalletLayerCount PalletWobjCount SolarVisionExec	ホストコンピュータとのデータ送受信エラー	-1	未知のエラー
レジスタ制御	ReadRegByteByName	データ読み込みエラー	-1	未知のエラー
4 軸ロック	SingAreaLockAxis4	姿勢エラー。4 軸ロック機能を ON にできない。	-1	未知のエラー
インタプリタ内部エラー	ほとんどの命令のパラメーターの種類と数に関するエラー		-1	XXX パラメーターエラー
インタプリタ内部エラー			0	

ネットワーク コマンド シリアルポー ト命令	OpenDev	サーバーへの接続に失敗	101	OpenDevConn が失敗
	OpenDev	ロボットをサーバーとして起 動できない	102	OpenDevServer が失敗
	GetSocketConn	SocketConn に対応する接続 は未確定	103	GetSocketConn が失敗 接 続が存在しない
	GetSocketConn	SocketConn で指定した構造 体が SocketServer	104	GetSocketConn が失敗 オ ブジェクトは SocketServer
	GetSocketServer	GetSocketServer が失敗 サー バーが存在しない	105	GetSocketServer が失敗 サ ーバーが存在しない
	OpenDev	オープン接続入力パラメータ エラー、変数リストに該当 する接続がない	106	OpenDev が失敗 存在しな いオブジェクトを使用
	SocketAccept	入力パラメーターがサーバー 名ではない	107	SocketAccept(server) 命 令 にサーバー名が必要
	GetSocketConn	SocketConn 構造体の名前の 取得エラー	108	GetSocketConn(conn) 存在 しない SocketConn
	GetSocketServer	SocketServer 構造体の名前の 取得エラー	109	GetSocketConn(server) 存 在しない SocketServer
	ReadBit	命令入力パラメーターエラー	110	ReadBit は 8 の整数倍を指 定する必要がある
	ReadDouble	命令入力パラメーターエラー	111	ReadDouble 命 令 が (0, 4096]の設定範囲外
	ReadInt	命令入力パラメーターエラー	112	ReadInt 命 令が(0, 4096]の 設定範囲外
	ReadByte	命令入力パラメーターエラー	113	ReadByte 命 令が(0, 4096] の設定範囲外
	ReadBit ReadDouble ReadInt ReadByte ReadString	入力時間が長い	114	ReadXX 命 令 時 間 が (0. 86400]の設定範囲外
	ReadBit ReadDouble ReadInt ReadByte ReadString	接続切断またはデータ読み込 みエラー	115	Read 命令が失敗
	ReadDouble	制限時間超過	116	ReadDouble 命令のタイム アウト
	ReadInt	制限時間超過	117	ReadInt 命令のタイムアウ ト
	ReadString	制限時間超過	118	ReadString 命令のタイムア ウト
	ReadBit	制限時間超過	119	ReadBit 命令のタイムアウ ト
	ReadByte	制限時間超過	120	ReadByte 命令のタイムア ウト
	SendString	命令のタイムアウトまたは接 続切断	121	SendString 命令のタイムア ウトまたは接続切断
	SendByte	命令のタイムアウトまたは接	122	SendByte 命令のタイムア

		続切断		ウトまたは接続切断
コンベアトラッキング命令	WaitObj	命令実行時に、ワークが起動ウィンドウを超えており、追跡できない	123	Out StartWindow
	WaitObj	ワークトラッキングのタイムアウト待ち	124	Out WaitTime
	WaitObj	ワークの繰り返し追跡	125	Connected Twice
	トラッキングをオンにした後に発生する可能性がある事例	トラッキング処理が作業領域を超えた	126	Out MaxDistance

15.4.9.13 Switch/Case

説明	Switch/Case 命令は、IF 命令と同様に入力された条件に基づいて個別の処理を行います。入力変数（condition）と Case フィールドの変数が順番に比較されます。 2 つの変数が等しい場合は、Case に対応するコード分岐に進みます。それ以降の比較は行わず、他のコード分岐にも進みません。 すべての条件が満たされていない場合は、Default 分岐に進みます。 条件が一致せず、かつ Default 分岐がない場合は、何もせずに Switch 命令は終了します。Case 命令は複数の条件を入力できます（命令構造の Case C1,C12,C13 と例 1 を参照）。
定義	<pre> Switch(condition) Case C1,C12,C13: Functions1() Case C2: Functions2() Default: DefaultFunction() EndSwitch </pre>
例	例 1 reg_int はレジスタ変数であり、ホストコンピューター（PLC）は関連するレジスタプロトコル（modbus、cclink など）を介して変数の値を更新します。生産工程はロボットがレジスタの数値によって処理が分岐することを希望しています。レジスタが 1,2,3 の場合は A 関数を実行し、4,5,6 の場合は B 関数を実行し、上記の条件を満たしていない場合は、Default 分岐に進んで C 関数を実行します。 <pre> Switch(reg_int) Case 1,2,3: FunctionA() // 1, 2, 3 の場合は A 関数を実行する Case 4,5,6: FunctionB() // 4, 5, 6 の場合は B 関数を実行する Default: FunctionC() //その他の場合は C 関数を実行する EndSwitch </pre>

15.4.10 ホーム位置命令

15.4.10.1 Home

説明	軸動作でロボットをホーム位置に動作させます。
定義	パラメーターなし
例	例 1 <code>HomeSet 0,30,0,60,0,90,0</code> <code>Home</code> HomeSet 命令でホーム位置を設定し、Home 命令でロボットを軸動作で動作させます。
注意	Home 命令のエラーを避けるために、ロボット設定>クイック調整インターフェースでホーム位置設定を有効にするか、HomeSet 命令でホーム位置を設定してから Home 命令を使用してください。

15.4.10.2 HomeSet

説明	ロボットの HOME 位置を設定します。
定義	HomeSet axis1, axis2, axis3, axis4, axis5, axis6, axis7 Axisx データ型：double HOME 位置での各軸の角度を指定します。
例	例 1 <code>HomeSet 0,30,0,60,0,90,0</code> <code>Home</code> HomeSet 命令でホーム位置を設定し、Home 命令でロボットを軸動作で動作させます。

15.4.10.3 HomeSetAt

説明	ロボットのホーム位置の設定データを取得します。
定義	HomeSetAt(index) 戻り値 データ型：double 関節角度を返します。単位は degree です。 Index データ型：int ホーム位置の指定軸の関節角度を取得します。 Index の 0 はホーム設定のオン/オフを表します。1 はオンで、0 はオフを意味します。
例	例 1 <code>HomeSet 0,30,0,60,0,90,0</code> <code>double angle2 = HomeSetAt(2)</code> ホーム位置での関節 2 の角度である 30°を取得します。

15.4.10.4 HomeDef

説明	ホーム位置が設定されているかどうかを判定します。
定義	HomeDef() 戻り値 データ型：bool True はホーム位置設定済み、False はホーム位置未設定を意味します。

15.4.10.5 HomeSpeed

説明	Home 命令の実行速度を設定します。
定義	HomeSpeed Speed
例	例 1 HomeSpeed v1000 Home Home 命令の速度を V1000 に設定します。Home 命令はロボットを V1000 の速度でホーム位置に動作させます。

15.4.10.6 HomeClr

説明	ホーム位置をクリアします。
例	例 1 HomeClr 設定したホーム位置をクリアします。クリアすると、Home 命令は実行できなくなります。

15.4.11 数学命令

15.4.11.1 Sin

説明	正弦を計算します。
定義	double sin(double x) x の単位はラジアン 戻り値: -1 から 1

15.4.11.2 Cos

説明	余弦を計算します。
定義	double cos(double x) x の単位はラジアン 戻り値: -1 から 1

15.4.11.3 Tan

説明	正接を計算します。
定義	double tan(double x) x の単位はラジアン 戻り値: x の正接

15.4.11.4 Cot

説明	余接を計算します。
定義	double cot(double x) x の単位はラジアン 戻り値: x の余接

15.4.11.5 Asin

説明	逆正弦を計算します。
定義	double asin(double x) x の範囲は-1 から 1 の間で、超えるとエラーになります。 戻り値: $-\pi/2$ から $\pi/2$。単位はラジアン。

15.4.11.6 Acos

説明	逆余弦を計算します。
定義	double acos(double x) x の範囲は-1 から 1 の間で、超えるとエラーになります。 戻り値：0 から PI。単位はラジアン。

15.4.11.7 Atan

説明	逆正接を計算します。
定義	double atan(double x) 戻り値：-PI/2 から PI/2。

15.4.11.8 Sinh

説明	双曲線正弦を計算します。
定義	double sinh(double x) 数学的定義式は $(\exp(x) - \exp(-x))/2$; 戻り値：x の双曲線正弦

15.4.11.9 Cosh

説明	双曲線余弦を計算します。
定義	double cosh(double x) 数学的定義式は $(\exp(x) + \exp(-x))/2$; 戻り値：x の双曲線余弦

15.4.11.10 Tanh

説明	双曲線正接を計算します。
定義	double tanh(double x) 数学的定義式は $\sinh(x)/\cosh(x)$; 戻り値：x の双曲線正接

15.4.11.11 Exp

説明	e のべき乗を計算します。
定義	double exp(double x) 戻り値：e の x 乗

15.4.11.12 Ln

説明	e を底とする対数値を計算します。
定義	double ln(double x) 関数説明：x の自然対数 $\ln(x)$ 、 $x > 0$ を計算します。 戻り値：x の自然対数値

15.4.11.13 log10

説明	10 を底とする対数値を計算します。
定義	double log10(double x) x > 0 ; 戻り値：x の 10 を底とする対数値

15.4.11.14 pow

説明	x のべき乗を計算します。
定義	double pow(double x, double y) 戻り値：x の y 乗

15.4.11.15 sqrt

説明	平方根を計算します。
定義	double sqrt(double x) x は正の数； 戻り値：x の平方根

15.4.11.16 ceil

説明	指定した値以上の最小整数値を返します。小数部分を切り上げることと同じ意味になります。 戻り値の型は double です。
定義	double ceil(double x) 戻り値：x 以上の最小整数値

15.4.11.17 floor

説明	指定した値以下の最大整数値を返します。小数部分を切り捨てることと同じ意味になります。 戻り値の型は double です。
定義	double floor(double x) 戻り値：x 以下の最大整数値

15.4.11.18 abs

説明	絶対値を計算します。
定義	int abs(int x)/double abs(double x) 戻り値：入力が int 型の場合、出力する戻り値も int 型になります。入力が double 型の場合、 戻り値も double 型になります。

15.4.11.19 rand

説明	整数型乱数を生成します。
定義	rand() 戻り値： int 型で、範囲は 0~2147483647 です。

15.4.12 ビット操作

15.4.12.1 BitAnd

説明	byte 型データに対する論理積を求めます。 <pre> BitPos 8 ----- BitPos 1 [0][0][1][0][0][1][1][0] data1 : 38 [0][0][1][0][0][0][1][0] data2 : 34 ----- [0][0][1][0][0][0][1][0] data3 : 34 </pre>
定義	戻り値 データ型 : byte 2 つの byte データの論理積を計算します。 BitAnd (BitData1, BitData2) BitData1 データ型 : byte 操作するバイトデータを指定します。 BitData2 データ型 : byte 操作するバイトデータを指定します。。
例	例 1 VAR byte data1 = 34 VAR byte data2 = 38 VAR byte byte3 = BitAnd(data1, data2) //34 byte 型変数 data1 を定義して値 34 を代入します。byte 型の変数 data2 を定義して値 38 を代入します。data1 と data2 に論理積を計算し 34 を求め、その値を byte3 に代入します。

15.4.12.2 BitCheck

説明	byte 型変数のあるビットが 1 であるかどうかを確認します。1 の場合は True を返し、0 の場合は False を返します。
定義	戻り値 データ型 : bool True は指定されたビットが 1 で、False は指定されたビットが 0 であることを意味します。 BitCheck (BitData, BitPos) BitData データ型 : byte 操作するバイトデータを指定します。 BitPos データ型 : int 操作するビットの位置を指定します。範囲は 1~8 です。
例	例 1 VAR byte data1 = 130

	<pre>VAR bool b1 = BitCheck(data1, 8) //true</pre> byte 型変数 data1 を定義して値 130 を代入します。data1 の第 8 ビットが 1 であるかどうかを確認し、True を返します。
--	--

15.4.12.3 BitClear

説明	byte または int 型のデータの指定ビットを 0 にします。ビットの指定は 1 から始めます。
定義	BitClear BitData IntData, BitPos BitData データ型 : byte 操作するバイトデータを指定します。 IntData データ型 : int 操作する整数型データを指定します。 BitPos データ型 : int 操作するビットの位置を指定します。byte データに対しては (1-8)、int データに対しては 1-32 を指定します。
例	例 1 <pre>VAR byte data1 = 255 BitClear data1 1 //254 BitClear data1 2 //252</pre> byte 型変数 data1 を定義して値 255 を代入します。data1 に BitClear 操作を行い、第 1 ビットを 0 にして 254 を取得します。次に第 2 ビットを 0 にして 252 を取得します。

15.4.12.4 BitLSh

説明	byte データに左シフト演算を行います。
定義	戻り値 データ型 : byte 左シフト演算を行います。 BitLSh (BitData, ShiftSteps) BitData データ型 : byte 操作するバイトデータを指定します。 ShiftSteps データ型 : int 左シフトするビット数を指定します。範囲は 1~8 です。
例	例 1 <pre>VAR int left_shift = 3 VAR byte data1 = 38 VAR byte data2 data2 = BitLSh(data1, left_shift) //48</pre> byte 型変数 data1 を定義して値 38 を代入します。data1 に対して 3 ビット左シフトして、48 を取得します。

15.4.12.5 BitNeg

説明	byte データに論理否定演算を行います。
定義	戻り値 データ型 : byte 論理否定操作を行います。

	BitNeg (BitData) BitData データ型 : byte 操作するバイトデータを指定します。
例	例 1 <pre>VAR byte data1 = 38 VAR byte data2 data2 = BitNeg(data1) //217</pre> bByte 型変数 data1 を定義して値 38 を代入します。data1 に論理否定演算を行って 217 を取得します。

15.4.12.6 BitOr

説明	byte データに論理和演算を行います。
定義	戻り値 データ型 : byte 論理和操作を行います。 BitOr (BitData1, BitData2) BitData1 データ型 : byte 操作するバイトデータを指定します。 BitData2 データ型 : byte 操作するバイトデータを指定します。
例	例 1 <pre>VAR byte data1 = 39 VAR byte data2 = 162 VAR byte data3 data3 = BitOr(data1, data2) //167</pre> byte 型変数 data1 を定義して値 39 を代入します。byte 型の変数 data2 を定義して値 162 を代入します。data1 と data2 に論理和演算を行って 167 を取得します。

15.4.12.7 BitRSh

説明	byte データに右シフト演算を行います。
定義	戻り値 データ型 : byte 右シフト演算を行います。 BitLSh (BitData, ShiftSteps) BitData データ型 : byte 操作するバイトデータを指定します。 ShiftSteps データ型 : int 右シフトするビット数を指定します。範囲は 1~8 です。
例	例 1 <pre>VAR int right_shift = 3 VAR byte data1 = 38 VAR byte data2 data2 = BiRSh(data1, right_shift) //4</pre> byte 型変数 data1 を定義して値 38 を代入します、data1 に対して 3 ビット右シフト演算を行い 4 を取得します。

15.4.12.8 BitSet

説明	byte または int 型のデータの指定ビットを 1 にします。ビットの指定は 1 から始めます。
定義	BitSet BitData IntData, BitPos BitData データ型 : byte 操作するバイトデータを指定します。 IntData データ型 : int 操作する整数型データを指定します。 BitPos データ型 : int 操作するビットの位置を指定します。 byte データに対しては (1-8)、int データに対しては 1-32 です。
例	例 1 <pre>VAR byte data1 = 0 BitSet data1 1 //1 BitSet data1 2 //3</pre> byte 型変数 data1 を定義して値 255 を代入します。data1 に BitSet 操作を行います。第 1 ビットを 1 にして 1 を取得し、第 2 ビットを 1 に設定して 3 を取得します。

15.4.12.9 BitXOr

説明	byte データに排他的論理和演算を行います。
定義	戻り値 データ型 : byte 排他的論理和演算を行います。 BitXOr (BitData1, BitData2) BitData1 データ型 : byte 操作するバイトデータを指定します。 BitData2 データ型 : byte 操作するバイトデータを指定します。
例	例 1 <pre>VAR byte data1 = 39 VAR byte data2 = 162 VAR byte data3 data3 = BitOr(data1, data2) //133</pre> byte 型変数 data1 を定義して値 39 を代入します。byte 型の変数 data2 を定義して値 162 を代入します。data1 と data2 に排他的論理和演算を行って 133 を取得します。

15.4.13 文字列操作

15.4.13.1 StrFind

説明	文字列の中を検索します。特定の位置から始めて、検索する文字セットに含まれる文字の位置を返します。
定義	戻り値 データ型：int 検索文字セットに含まれる最初の文字の位置を返します。一致しない場合は、文字列長+1 を返します。 StrFind (Str, ChPos, Set [NotInSet]) Str データ型：string 検索する文字列を指定します。 ChPos データ型：int 検索開始位置を指定します（1 から始める）。位置が範囲外の場合は、エラーを報告します。 Set データ型：string 検索する文字セットを指定します。 [NotInSet] 識別子 検索する文字セットに含まれない文字を検索する場合に指定します。
例	例 1 VAR int found found = StrFind("Robotics", 1, "aeiou") //2 最初の文字「R」から検索します。2 番目の文字「o」が文字セット「aeiou」に含まれるので、2 を返します。 found = StrFind("Robotics", 1, "aeiou" \NotInSet) //1 最初の文字「R」から検索します。1 番目の文字「R」が文字セット「aeiou」に含まれないので、1 を返します。

15.4.13.2 StrLen

説明	文字列の長さを取得します。
定義	戻り値 データ型：int 文字列長を返します。>=0。 StrLen (Str) Str データ型：string 文字数を取得したい文字列を指定します。
例	例 1 VAR int num num = StrLen("Robotics") //8 文字列「Robotics」の長さである 8 を返します。

15.4.13.3 StrMap

説明	指定された文字を指定する関係に従って置換します。指定する文字は位置で対応し、指定のない文字は変化しません。
定義	戻り値 データ型：string 置換後の文字列を返します。 StrMap (Str, FromMap, ToMap) Str データ型：string 置換対象の文字列を指定します。 FromMap データ型：string 置換したい文字セットを指定します。 ToMap データ型：string 置換後の文字セットを指定します。
例	例 1 VAR string str str = StrMap("Robotics", "aeiou", "AEIOU") //RObOtIcs 文字列「Robotics」に対して、「aeiou」をそれぞれ「AEIOU」に置換します、 使用制限：FromMap と ToMap は長さが一致する必要があります。

15.4.13.4 StrMatch

説明	文字列の中を検索します。特定の位置から初めて、検索する文字セットと一致する文字の位置を返します。
定義	戻り値 データ型：int 検索文字セットと一致した最初の文字の位置を返します。一致しない場合は文字列長+1 を返します。 StrMatch (Str, ChPos, Pattern) Str データ型：string 検索する文字列を指定します。 ChPos データ型：int 検索開始位置を指定します（1 から始める）。位置が範囲外の場合は、エラーを報告します。 Pattern データ型：string 検索する文字セットを指定します。
例	例 1 VAR int found Found = StrMatch("Robotics", 1, "bo") //3 1 番目の文字「R」から検索を始めます。「bo」を探索し、bo を見つけると最初の文字 b の位置である 3 を返します。

15.4.13.5 StrMemb

説明	文字列内の文字が指定した文字セットに含まれるかどうかを確認します。
定義	戻り値 データ型：bool True は指定した位置の文字が、指定した文字セットに含まれることを意味します。含まれない場合は False になります。 StrMemb (Str, ChPos, Set) Str データ型：string 確認する文字列を指定します。 ChPos データ型：int 確認する文字の位置を指定します。位置が範囲外の場合は、エラーを報告します。 Set データ型：string 確認する文字セットを指定します。
例	例 1 <pre>VAR bool memb memb = StrMemb("Robotics", 2, "aeiou") //true</pre> 2 番目の文字 o が文字セット「aeiou」に含まれるかどうかを確認します。含まれるので、True を返します。

15.4.13.6 StrOrder

説明	2 つの文字列を昇順で並べた時の順番を確認します。
定義	戻り値 データ型：bool str1<=str2 の時は True を返し、逆の場合は False を返します。 StrOrder (Str1, Str2) Str1 データ型：string 1 つ目の文字列を指定します。 Str2 データ型：string 2 つ目の文字列を指定します。
例	例 1 <pre>VAR bool le le = StrOrder("FIRST", "SECOND") //true le = StrOrder("FIRSTB", "FIRST") //false</pre>

15.4.13.7 StrPart

説明	文字列の一部を取り出して新しい文字列を生成します。
定義	戻り値 データ型：string 指定された位置から指定された長さの文字列を取り出して返します。 StrPart (Str, ChPos, Len) Str データ型：string 文字列を指定します。

	ChPos データ型：int 取り出し開始位置を指定します。文字列の範囲外の場合はエラーを報告します。 Len データ型：int 取り出す長さを指定します。
例	例 1 <pre>VAR string part part = StrPart("Robotics", 1, 5) //Robot</pre> 1 文字目から長さ 5 の文字列を取り出して「Robot」を取得します。

15.4.13.8 StrSplit

説明	指定したデリミタで、文字列を配列に分割します。
定義	戻り値 データ型：string 配列 分割後の文字列を配列で返します。 StrSplit (Str [, separator]) Str データ型：string 分割する文字列を指定します。 Separator データ型：string デリミタを指定します。ここで指定したすべての文字はデリミタとみなされます。省略可能です。デフォルトの場合、スペースがデリミタとして使われます。
例	<pre>string str_arr[4] = StrSplit("test1,test2,test3\test4", "\",,")</pre> 文字列を分割し、文字列配列 (test1 test2 test3 test4) を取得します。 利用制限： 入力する文字列が NULL の場合はエラーを報告します。 分割結果と定義された文字列長が一致しない場合はエラーを報告します。

15.4.13.9 StrToByte

説明	文字列を byte 型データに変換します。
定義	戻り値 データ型：byte 文字列の変換結果を返します。 StrToByte (Str, [trans]) Str データ型：string データ変換を行う文字列を指定します。 Trans データ型：キーワード 数値進数形式を指定します。使用可能なパラメーターは、¥Bin (2 進数)、¥Okt (8 進数)、¥Hex (16 進数)、¥Char (文字)、及びデフォルトパラメータ (パラメーターなし、10 進数) です。
例	例 1 <pre>Byte NumBin = StrToByte("10", \Bin) Byte NumOkt = StrToByte("10", \Okt)</pre>

	<pre>Byte NumBin = StrToByte("10") Byte NumHex = StrToByte("10", \Hex)</pre> 文字列「10」を 2 進数、8 進数、10 進数、16 進数に基づいて Byte 型に変換します。その結果は順に 2、8、10、16 となります。 例 2 <pre>Byte NumChar = StrToByte("0", \Char)</pre> 文字「0」を文字と ASCII の変換関係に基づいて 48 に変換します。 利用制限：入力する文字列が指定したデータ形式に一致しない場合はエラーを報告します。
--	--

15.4.13.10 StrToDouble

説明	文字列を double 型データに変換します。
定義	戻り値 データ型：double 文字列の変換結果を返します。 StrToDouble (Str) Str データ型：dtring データ変換を行う文字列を指定します。
例	例 1： <pre>Double NumDouble = StrToDouble("3.1415926")</pre> 文字列「3.1415926」を double 型データに変換します。 利用制限： 入力する文字列が指定したデータ形式に一致しない場合はエラーを報告します。

15.4.13.11 StrToInt

説明	文字列を int 型データに変換します。
定義	戻り値 データ型：int 文字列の変換結果を返します。 StrToDouble (Str) Str データ型：string データ変換を行う文字列を指定します。
例	例 1 <pre>Int NumInt = StrToInt("99")</pre> 文字列「99」を int 型データに変換します。 利用制限：入力する文字列が指定したデータ形式に一致しない場合はエラーを報告します。

15.4.13.12 StrToDoubleArray

説明	文字列を Double 型配列に変換します
定義	戻り値 データ型：int 変換に異常があるかどうかを返します。-1 はエラーで、0 は正常を意味します。

	StrToDoubleArray(output, input, spilit) Output データ型：double 配列 結果を格納する配列を指定します。 Input データ型：string 分割したい文字列を指定します。 Spilit データ型：string 文字列のデリミタを指定します。
例	例 1 <pre>string tmp_ss = "1,2,3,4,5,6,7" double db_arr[10] StrToDoubleArray (db_arr, tmp_ss, ",") // 結果 db_arr = {1,2,3,4,5,6,7,0,0,0}</pre> 利用制限： 文字列の最後にサフィックスを追加できます。 変換に失敗したり、不正な文字が含まれていたりすると、「入力する文字列を分割した後は double 形式になりません」というエラーを報告します。 double 配列は、文字列のデータ量以上（データ量を含む）のサイズでなければなりません。さもなくば、「入力する配列のサイズが十分でないか、配列が一次元配列ではありません」というエラーを報告します。

15.4.14 演算子

15.4.14.1 基本演算子

15.4.14.1.1 算術演算子

算術演算子には以下の種類があります。

演算子	内容
+	加算
-	減算/マイナス
*	乗算
/	除算
%	剰余
--	デクリメント
++	インクリメント

算術演算子は、bool 型、byte 型、int 型、double 型のデータに対する操作をサポートします。

異なる型の変数で加算、減算、乗算、除算を行うと、暗黙の型変換が行われます。

各算術演算子の使い方は以下の通りです。

例 1

```
VAR int a = 1
VAR int b = 2
VAR int c = -b //負の数
VAR int ac = a * c //乗算
```

例 2

RL はインクリメント・デクリメントの前後を区別しません。

```
x = n++      //n に 1 を加算してから、n の値を x に代入
x = --n      //n に 1 を減算してから、n の値を x に代入
```

例 3

異なる型の変数の加算、減算、乗算、除算に対する暗黙の型変換

型 1	型 2	結果
bool	bool	bool
bool	byte	byte
bool	int	int
bool	double	double
byte	byte	byte
byte	int	int
byte	double	double
int	int	int
int	double	double
double	double	double

15.4.14.1.2 論理演算子

論理演算子には、以下の種類があります。

演算子	内容
&&	論理積
	論理和
<	小なり
>	大なり
<=	以下
>=	以上
==	イコール
!=	ノットイコール
!	論理否定

論理積が真である条件は、2 つとも真であることであり、論理和が真である条件は、2 つのうち一方のみが真であることです。

例 1

論理演算子の使い方は以下の通りです。

```
VAR int res = 1
while(res < 3)    //res が 3 より小さいかどうかを比較する
    res++
endwhile
di5 = !di6        //論理否定を取る
VAR int counter = 4
while(di7 && di8) //論理積を求める
    if(counter == 5) //等しいかどうか
        break
endif
```


endwhile

15.4.14.1.3 代入演算子

代入演算子には、以下の種類があります。

演算子	内容
=	代入
+=	加算代入
-=	減算代入
*=	乗算代入
/=	除算代入
%=	剰余代入

各代入演算子の使い方は以下の通りです。

```
VAR int num1 = 3
```

```
VAR int num2 = 4
```

```
num1 += num2      //num1 = num1 + num2 を意味し、num1 = 7 になります。
```

```
num1 -= num2      //num1 = num1 - num2 を意味し、num1 = -1 になります。
```

```
num1 *= num2      //num1 = num1 * num2 を意味し、num1 = 12 になります。
```

```
num1 /= num2      //num1 = num1 / num2 を意味し、num1 = 0 になります。
```

```
num1 %= num2      //num1 = num1 % num2 を意味し、num1 = 3 になります。
```

変数のすべての代入操作は、暗黙の型変換を行います。代入操作の左辺と右辺のデータ型が一致しない場合、暗黙の型変換を行います。変換に失敗すると、プログラムはエラーを報告し停止します。

bool, byte, int, double の 4 つのデータ型は相互に変換できます。IO、レジスタ変数は上記の 4 つの変数の特殊な形に属し、代入操作で使用される場合、暗黙の型変換が行われる可能性があります。

関数の戻り値が上記の 4 つの型のいずれかに属していれば、代入演算の右辺の値として使用することができます。

例 1

```
int tmp_num = 10.5  // 10
bool tmp_bool = 1   // true
tmp_bool = 0        // false
double tmp_d = 999  // 999.0
```

例 2

```
// 一般変数をレジスタ変数で直接に変更する
double tmp_num = register0
// レジスタ変数は直接条件判定に使用される
WaitUntil(register0 == 10)
```

例 3

```
int mem_ret = StrMemb("Robotics", 2, "aeiou")
// StrMemb の戻り値は bool 型であり、int 型で受け取る必要がある場合、
// コントローラはエラーを報告する代わりに、間接的な変換を行う
// true -> 1 ,   false-> 0
```

15.4.14.1.4 その他の演算子

演算子	内容
()	丸かっこ
.	ポイント

各演算子の使い方は以下の通りです。

例 1

```
VAR int num = arr[1]    //配列の最初の要素を取って num に代入する
VAR int num2 = (1+2)*3  //かっこを使って演算順序を変える。num2 の値は 9 になる。
```

例 2

robtargt 変数 pt1 を定義します

```
pt1.trans.x = 200    //「.」演算子を使って pt1 点の x 座標を 200 に変更する
```

利用制限：

「.」演算子は Robtarget 変数 A, B, C メンバーに対する変更をサポートしません。

15.4.14.2 演算の優先度

優先度	演算子	使用形式	結合方向
1	()	(式) /関数名 (引数リスト)	
	.	変数名	
2	-	-式	右から左へ
	++	++変数名/変数名++	
	--	--変数名/変数名--	
	!	!式	
3	/	式/式	左から右へ
	*	式*式	
	%	整数型式/整数型式	
4	+	式+式	左から右へ
	-	式-式	
5	>	式>式	左から右へ
	>=	式>=式	
	<	式<式	
	<=	式<=式	
6	==	式==式	左から右へ
	!=	式!=式	
7	&&	式&&式	左から右へ
8		式 式	左から右へ
9	=	変数=式	右から左へ
	/=	変数/=式	
	=	変数=式	
	%=	変数%=式	
	+=	変数+=式	
	-=	変数-=式	

15.4.15 クロック命令

15.4.15.1 ClkRead

説明	タイマーの値を読み出します。
定義	戻り値 データ型：double 0.001s の精度で、Clock 型変数の値を返します。 ClkRead (Clock) Clock データ型：clock タイマー名を指定します。
例	例 1 VAR clock clock1 ClkStart clock1 ClkStop clock1 VAR double interval=ClkRead(clock1) interval は clock1 の開始から停止までの時間になります。

15.4.15.2 ClkReset

説明	タイマーをリセットします。タイマーを使用する前に、ClkReset 命令でカウントを 0 にリセットします。
定義	ClkReset Clock Clock データ型：clock タイマー名を指定します。
例	例 1 VAR clock clock1 ClkReset clock1 clock1 をリセットします。

15.4.15.3 ClkStart

説明	タイマーを開始する際に使用します。 タイマーが開始されると、タイマーを停止するかプログラムリセットまでカウントを実行し続けます。プログラムが停止したり、モーターの電源がオフになってもタイマーは作動し続けます。
定義	ClkStart Clock Clock データ型：clock タイマー名を指定します。
例	例 1 VAR clock clock1 ClkStart clock1 clock1 を宣言し、タイマーclock1 を開始します。

15.4.15.4 ClkStop

説明	タイマーを停止させます。 タイマー停止後は、間隔の読み取り、再開、リセットが可能です。
定義	ClkStop Clock Clock データ型：Clock

	タイマー名を指定します。
例	例 1 <pre>VAR clock clock1 ClkStart clock1 ... ClkStop clock1</pre> タイマーclock1 を停止させます。

15.4.16 高級命令

15.4.16.1 RelTool

説明	指定されたツール座標系にそって、位置と姿勢を平行移動または回転させます。 Offs との主な違い： Offs はワーク座標系に対するオフセットで、RelTool はツール座標系に対するオフセットです。
定義	戻り値 データ型：robtarget オフセット後の新しい位置/姿勢を返します。 RelTool(Point, XOffset, Yoffset, Zoffset, Rx, Ry, Rz [, Tool, Wobj]) Point データ型：robtarget オフセットしたい位置/姿勢を指定します。 XOffset データ型：double ツール座標系の x 方向のオフセット量を指定します。 YOffset データ型：double ツール座標系の y 方向のオフセット量を指定します。 ZOffset データ型：double ツール座標系の z 方向のオフセット量を指定します。 Rx データ型：double ツール座標系の x 軸周りの回転角度を指定します。 Ry データ型：double ツール座標系の y 軸周りの回転角度を指定します。 Rz データ型：double ツール座標系の z 軸周りの回転角度を指定します。 Tool データ型：tool ツールを指定します。 Wobj データ型：wobj ワークを指定します。
例	例 1 <pre>p2=RelTool(p1,100,0,30,20,0,0)</pre> ツール・ワークが指定されていない場合は、デフォルトで tool0,wobj0 を使用します。p1 はツール座標系の x 方向に 100mm、y 方向に 0mm、z 方向に 30mm オフセットし、x 軸まわりに 20 度回転した後、新しい位置/姿勢を p2 に割り当てます。

	例 2 <code>p2=RelTool(p1,100,0,30,20,0,0, tool5, wobj6)</code> p1 を tool5 ツール座標系の x 方向に 100mm、y 方向に 0mm、z 方向に 30mm オフセットし、x 軸まわりに 20 度回転した後、新しい位置/姿勢を p2 に割り当てます。 例 3 <code>MoveL RelTool(p1, 100,0,30,20,0,0), v4000, fine, tool2, wobj4</code> RelTool 命令と Move 命令を併用する場合 特定のツール・ワーク座標系を指定しないと、動作命令の tool と wobj を使用します。p1 を tool2 ツール座標系の x 方向に 100mm、y 方向に 0mm、z 方向に 30mm オフセットし、x 軸周りに 20 度回転した後、新しい位置/姿勢を p2 に割り当てます。
注意	この命令の tool と wobj は、現時点ではプログラミング補助をサポートしていません。 xMate CR シリーズの 5 軸モデルは姿勢自由度が 1 つ少ないため、RelTool 命令を使用する時には多くのポイントに到達できない可能性があります。そのため、到達可能な姿勢にオフセットする必要があります（例えば、平行移動は通常、フランジがベースと平行で到達可能です）。

15.4.16.2 Offs

説明	指定されたワーク座標系にそって、位置と姿勢を平行移動または回転させます。 平行移動オフセット量は x、y、z、姿勢回転オフセット量は Rx、Ry、Rz で指定します。
定義	戻り値、データ型：robtarget、オフセットした後の新しいポーズ。 Offs (Point, XOffset, YOffset, ZOffset [, Rx, Ry, Rz]) Point データ型：robtarget オフセットしたい位置/姿勢を指定します。 XOffset データ型：double ワーク座標系の x 方向のオフセット量を指定します。 YOffset データ型：double ワーク座標系の y 方向のオフセット量を指定します。 ZOffset データ型：double ワーク座標系の z 方向のオフセット量を指定します。 Rx データ型：double ワーク座標系の x 軸周りの回転角度を指定します。 Ry データ型：double ワーク座標系の y 軸周りの回転角度を指定します。 Rz データ型：double ワーク座標系の z 軸周りの回転角度を指定します。
例	例 1 <code>p11=Offs(p10,100,200,300)</code> 点 p10 をワーク座標系の x 方向に 100mm、y 方向に 200mm、z 方向に 300mm オフセットし、新しい位置を点 p11 に割り当てます。
注意	この命令は現時点ではプログラミング補助をサポートしていません。

15.4.16.3 Confl

説明	このコマンドを有効にすると、次の 2 つをチェックします。 1. 直交経路の実際の到達軸角度（cf1～7）が目標点で教示された値と大きく異なるかどうかを確認します。差が大きすぎる場合、ログに対応する警告が表示され、ユーザーはこの動作方式で目標点に到達するかどうかを検討できます。 2. 直交経路の実際の到達軸の関節構成（cfx 値）が目標点で教示されたものと一致するかどうかを確認します。一致しない場合はエラーが発生し、動作が停止します。
例	例 1 <pre>p1.trans.x = MoveL p1, v1000</pre> // 例えば、P1 の J6 の角度が-5° の場合、MoveL コマンドの実行後に J6 の角度が 355° になる可能性があります（両者は±360° の差があり、同じ位置に対応します）。 この場合、ログには『回転角の実際の到達点と設定点の差が大きすぎます』という警告が表示されますが、動作は停止しません。 <pre>... Confl Off MoveL p1, v1000</pre> //conf チェックを無効にすると、ロボットは P1 まで移動できますが、実際の到達点と教示時の confdata に含まれる cf1～7 値および cfx 値のチェックは行われません。
注意	1. Confl 命令は、MoveL や MoveC などの直交動作に対応し、MoveJ や MoveAbsJ には影響を与えません。また、「Move to」の conf 設定にも影響しません。 2. 直線や円弧などの直交動作を実行する場合、ロボットは開始点の姿勢に最も近い姿勢を選択しつつ、目標点に向かって移動します。この際、目標点の角度は自動的に選択されるため、ロボットの実際の回転角度が、教示された位置の角度と異なる場合があります。本コマンドを有効にすると、このような状況が発生した際に警告が表示されます。

15.4.16.4 ConfJ

説明	直交座標は 1 セットの conf パラメータ（cf1～7, cfx）に対応します。ユーザーが手動で変更したり書き込んだりすると、直交座標に対応する conf データが異なってしまう可能性があります。その結果、コントローラーが目標点までのパスを解析できなくなります。 ユーザーがロボットの TCP 点の位置だけを気にして、姿勢を気にしない場合があります。その場合は ConfJ Off で conf の制限を解除し、コントローラーにパスの始点に最も近い逆解を計算させます。（計算できない可能性もあり、その場合は動作命令が失敗します）。 conf の関連情報については、confdata の説明を参照してください。
例	例 1 <pre>p1.trans.x = MoveJ p1, v1000</pre> // 座標のみを変更し、confdata パラメーターを変更しなかった場合、この命令は実行に失敗する可能性が高くなります。 <pre>... Confl Off MoveJ p1, v1000</pre> //conf チェックを無効にすると、ロボットは点 p1 まで動作できますが、姿勢は不確定になります。
注意	ConfJ 命令は MoveJ に対応し、他の動作命令には影響を与えません。また、『Move to』の conf 設定にも影響しません。

15.4.16.5 Conf

説明	ConfL と ConfJ の一括オン/オフ命令です。具体的な動作は以下のとおりです。 Conf on は、ConfL on および ConfJ on と同等の動作をします。 Conf off は、ConfL off および ConfJ off と同等の動作をします。
例	例 1 <pre>Conf on MoveL p0, v1000 ... MoveJ p1, v1000</pre> // 以下のコマンドと同等です。 <pre>ConfL on ConfJ on MoveL p0, v1000 ... MoveJ p1, v1000</pre>

15.4.16.6 VelSet

説明	ロボットの最大動作速度を制限します。 ロボットが壊れやすいものを運搬する場合、最大動作速度を制限することでよりスムーズな動作を実現することができます。ロボットの各関節の最大速度能力は、実際の動作パスのもとでは一定ではなく、負荷の状態や本体の姿勢などによって常に変化しています。VelSet 命令は、特定のタスクパスのもとでの最大速度能力曲線を制限します。制限された後の最大速度能力曲線も変化する曲線です。
定義	VelSet gain Gain データ型 : int パーセンテージによって最大速度の大きさを指定します。取り得る値の範囲は 1%~100% です。ここで、100%は最大速度に対応し、リミット範囲を超えるとロボットはエラーを報告します。
例	例 1 <pre>VelSet 50</pre> 最大速度を半分に設定します。
注意	- VelSet 命令は、RL プロジェクトの動作命令のみに影響を与えます。ジョグ機能、「Move to」機能、クイック移動機能などには影響を与えません。 - VelSet 機能は、滑らか軌跡を中断させます。滑らか軌跡が必要な動作命令の間に VelSet 命令を挿入しないでください。 - VelSet 命令とプログラム実行速度調整スライダの違い : スライダはユーザーの希望速度を変更します。 例えば、V4000 動作命令は、スライダで 50%と指定された場合は、V2000 が希望速度です。しかし、この時のロボットが実際に到達できる最高速度は V1000 しかないこともあります。V2000 と V4000 は V1000 を超えるため、この範囲内で希望速度を変更しても、実際の動作速度に影響を与えません。したがって、速度スライダが 50%でも 100%でも、ロボットの実際の速度は変わりません。 VelSet50 はユーザーの希望速度は変更しませんが、その動作命令によって実際に達成できる最高速度を 50%にします。上の動作を例とすると、実際のロボットの動作速度は半分に減少し、V1000 から V500 になります。 この 2 つの機能の違いを区別する必要があります。 - 以下の操作が発生すると、速度は自動的にデフォルト（100%）に戻ります。 ● RL プログラムをリセットした時（PP to Main）

- 新しい RL プログラムをロードした時

15.4.16.7 AccSet

説明	ロボットの最大加速度を制限します。 ロボットが壊れやすいものを運搬する場合、加速度を制限することでよりスムーズな動作を実現することができます。
定義	AccSet acc, ramp Acc データ型：int パーセンテージによって最大加速度の大きさを指定します。取り得る値の範囲は 30%~100%です。ここで、100%は最大加速度に対応し、リミット範囲を超えるとロボットはエラーを報告します。 Ramp データ型：int パーセンテージによって最大加加速度（Jerk）の大きさを指定します。取り得る値の範囲は 30%~100%です。ここで、100%は最大加加速度に対応し、リミット範囲を超えるとロボットはエラーを報告します。
例	例 1 AccSet 50, 50 加速度、加加速度をデフォルト値の半分に設定します。
注意	以下の操作が発生すると、加速度は自動的にデフォルト（100%）に戻ります。 ● RL プログラムをリセットした時（PP to Main） ● 新しい RL プログラムをロードした時

15.4.16.8 MotionSup

説明	衝突検知機能をオン/オフするために使用します。 CR シリーズ及び SR シリーズでは常時オンであり、衝突検知のオフはサポートされていません。
定義	MotionSup type [, level, event] Type データ型：キーワード On オン、Off オフ Level データ型：int オプションパラメーターです。衝突検知感度のパーセンテージを変更します。範囲は[1,200]です。 Event データ型：string オプションパラメーターです。衝突した後の処理を設定します。 ● 「safestop」安全停止 ● 「pause」一時停止（協働ロボットのようにサポートされ、産業用ロボットは本停止方法をサポートしていません） ● 「softstop」ソフトウェアストップ
例	例 1 MotionSup On //...その他の命令 MotionSup Off 衝突検知をオンにします。その他の命令を実行後、MotionSup Off を使って衝突検知をオフにし

	ます。
	例 2 MotionSup On, 200, "softstop" 衝突検知をオンにし、検知感度のパーセンテージを 200%に、衝突検知後の停止方法をソフトウェアストップに設定します。

15.4.16.9 MotionSupPlus

説明	MotionSupPlus (Motion Supervision Plus) RL プログラム内でロボットの各関節の衝突検知感度を調整します。<u>CR シリーズ及び SR シリーズの衝突検知感度は、協働モードでのみ有効です</u>
定義	MotionSupPlus x1,x2,x3,x4,x5,x6,x7 x1~x7 関節 1 から関節 7 までの衝突検知感度を指定します。単位は%です。
例	例 1 MotionSupPlus 5,20,7,20,6,20,5 7 つの関節の感度をそれぞれ 5, 20, 7, 20, 6, 20, 5 にします。 注意： 6 軸ロボットに使用する時でも、7 つのパラメーターを指定する必要があります。そのうち最初の 6 つが有効となり、関節 1~6 に対応します。 この命令は協働ロボットと 6 軸産業用ロボットで使用可能です。3 軸または 4 軸産業用ロボットでは使用できません。

15.4.16.10 MotionSupJointTrq

説明	MotionSupJointTrq (Motion Supervision Joint Torque) RL プログラム内でロボットの各関節の伝達トルク制限パラメーターを変更するために使用します。
定義	MotionSupJointTrq x1,x2,x3,x4,x5,x6,x7 x1~x7 関節 1 から関節 7 までの伝達トルク制限の値を指定します。単位は Nm です。
例	例 1 MotionSupJointTrq 100,91,59,14,14,14,5 7 つの関節の伝達トルク制限パラメーターを、それぞれ 100Nm、91Nm、59Nm、14Nm、14Nm、14Nm、5Nm にします。 注意： 6 軸ロボットに使用する時でも、7 つのパラメーターを指定する必要があります。そのうち最初の 6 つが有効となり、関節 1~6 に対応します。 この命令は協働ロボットと 6 軸産業用ロボットで使用可能です。3 軸または 4 軸産業用ロボットでは使用できません。

15.4.16.11 BreakLookAhead

説明	本命令は、先読みをキャンセルし、動作命令の間の滑らか軌跡エリアを強制的にキャンセルします。ロボット TCP は前の動作命令の目標点まで動作した後、滑らか軌跡エリアと接続しないで次の目標点に動作します。プログラムポインターは、TCP が前の動作命令の目標点に到着してから、次に先読みを行います。
定義	パラメーターと戻り値はありません。

例	例 1 <pre>MoveL P1,v1000,z50,tool0 BreakLookAhead MoveL P2,v1000,z50,tool0 MoveL P3,v1000,z50,tool0</pre> 1) P1 の滑らか軌跡エリアを z50 に設定するが、BreakLookAhead 命令があるため、この先読みはキャンセルされ、滑らか軌跡もキャンセルされます。ロボット TCP は正確に点 P1 に動作してから点 P2 に動作します。そして、P2 と P3 の間には BreakLookAhead 命令がないため、点 P2 では先読みをして z50 の滑らか軌跡エリアを保持して P3 に動作します。 2) BreakLookAhead 命令は、wait0 命令と全く同じ効果を持ちます。
---	---

15.4.16.12 GetRobotMaxLoad

説明	当該モデルのロボットの最大負荷値を取得します。
定義	Ret = GetRobotMaxLoad() Ret 戻り値 データ型：int 最大負荷を返します。
例	<pre>int maxload = GetRobotMaxLoad() print(maxload)</pre> xMate 7 を例とすると、戻り値は 7 になります。

15.4.16.13 GetRobotState

説明	制御システムの現在の動作状態を取得します。4 バイトの bit 情報で、故障、非常停止、セーフティーゲート、操作モード、サーボモード、動作状態などの制御システムの状態を示します。詳細は次の表に示す通りです。
定義	Ret = GetRobotState() Ret 戻り値 データ型：byte 配列 4 つの byte 型でロボットの状態を返します。
例	例 1 <pre>byte st[4] = GetRobotMaxLoad() print(st)</pre> {0,5,0,0}が返された場合、表より現在の状態は、故障なし、モーターオン、自動モード、ロボット動作状態、サーボが位置モードにあるということを知ることができます。

番号	ステータスビット	意味
1	Byte[1].bit[1]	1：不正な制御システム
2	Byte[1].bit[2]	1：回復可能な制御システム故障
3	Byte[1].bit[3]	1：致命的な制御システムエラー
4	Byte[1].bit[4]	1：サーボシステム故障
5	Byte[1].bit[5]	1：致命的なサーボシステム故障
6	Byte[1].bit[6]	1：非常停止
7	Byte[1].bit[7]	1：セーフティーゲート停止（CR シリーズ及び SR シリーズでは停止エリア入力信号）
8	Byte[1].bit[8]	保留
9	Byte[2].bit[1]	モーター電源投入状態 0：モーターオフ 1：モーターオン

10	Byte[2].bit[2]	ロボット動作状態 0：アイドル 1：動作中
11	Byte[2].bit[3]	操作モード 0：手動モード 1：自動モード
12	Byte[2].bit[4]	サーボモード 0：位置モード 1：トルクモード
13	Byte[2].bit[5]	保留
14	Byte[2].bit[6]	保留
15	Byte[2].bit[7]	保留
16	Byte[2].bit[8]	保留
17	Byte[3]	保留
18	Byte[4]	保留

15.4.16.14 AutolgnoreZone

説明	制御システムが滑らか軌跡エリアを自動的に無視するかどうかを指定します。
定義	AutolgnoreZone true/false True：制御システムが自動的に滑らか軌跡エリアを無視することを許可します（デフォルト状態）。 False：制御システムが自動的に滑らか軌跡エリアを無視することを許可しません。 ロボットが上図のような 2 つの MoveL 命令を実行し、p3 への動作に z50 の滑らか軌跡エリアを含んでいるとします。 動作の柔軟性と安全性のために、現在の位置より前に一定の距離を先読みする必要があります。例えば、ロボットが p0 に動作する時、p1 を先読みします。この間、制御システムは p0 と p1 の間の情報を事前処理します。 ロボットが動作すると、先読みは前進し続けます。ある時点で先読み終点 p1 と滑らか軌跡エリア開始点 p2 が一致します。このとき、制御システムが 2 番目の動作命令を受信した場合、滑らか軌跡エリアは正常に生成でき、ロボットを所定の軌跡に沿って移動させるように制御できます。制御システムが 2 番目の動作命令を受信できなかった場合は、滑らか軌跡エリアを正常に生成できず、制御システムは AutolgnoreZone 命令の状態に基づいて滑らか軌跡エリアを扱います。具体的なロジックは以下の通りです。 AutolgnoreZone True：システムは 2 番目の動作命令を待たず、滑らか軌跡エリアをキャンセルして直接点 P3 に

	動作します。 False : システムは 2 番目の動作命令を待ちます。この間、ロボットは滑らか軌跡エリアが生成されるまで動作速度を落とします。ロボットが p2 に動作する時で、まだ 2 番目の動作命令を受信していない場合は、ロボットは移動を停止し、HMI を介してエラーを報告します。 ロボットが 2 番目の動作命令を時間内に受信できないことは、2 つの動作命令の間に非動作命令が多すぎる場合に発生します。 例えば、 <pre>MoveL p3,v1000,z50,tool0 For(int i from 1 to 10000) printf("i = %d\n", i) endfor MoveL p4,v1000,fine,tool0</pre> 2 つの運動命令の間に非常に多くの表示命令が追加され、制御システムが 1 番目の動作命令を処理してから 2 番目の動作命令を受信するまでに長い時間がかかります。
例	例 1 <pre>AutolgnoreZone true MoveL p3,v1000,z50,tool0 MoveL p4,v1000,fine,tool0</pre> 制御システムが自動的に滑らか軌跡エリアを無視することを許可します。 例 2 <pre>AutolgnoreZone false MoveL p3,v1000,z50,tool0 MoveL p4,v1000,fine,tool0</pre> 制御システムが自動的に滑らか軌跡エリアを無視することを許可しません。

15.4.16.15 MotionWaitAtFinePoint true/false

説明	ロボットが静止しているときに開始をタップすると、制御システムは先読みパラメーターに基づいて、一定距離を先読みしてからロボットを起動します。この命令は、先読みが fine ポイントに遭遇したときに、ロボットがすぐに動作するかどうかを設定するために使用されます。 fine ポイント : 滑らか軌跡エリアと接続しない目標点、即ち滑らか軌跡エリアパラメータが fine と書かれている目標点を指します。
定義	MotionWaitAtFinePoint true/false True : 制御システムは、先読みパラメーターに厳密に従ってロボットの動作を制御します。先読み距離が先読みパラメーターの設定値に達したり、すべての動作命令を先読みしたりしたときにのみ、ロボットは動作し始めます。この状態では、制御システムは設定された先読み距離の長さを保証できます。 False : 制御システムは、先読みパラメーターに厳密に従いません。先読みが fine ポイントに遭遇した時、ロボットは直ちに動作し始めます。この状態で、非常に複雑なプログラムロジックに遭遇しても、ロボットは依然として正常に起動できますが、先読み距離の長さを保証できません。 制御システムのデフォルト状態 : False
例	例 1 <pre>MotionWaitAtFinePoint true MoveL p1,v1000,fine,tool0 MoveL p2,v1000,fine,tool0 MoveL p3,v1000,fine,tool0 MoveL p4,v1000,fine,tool0</pre>

	MoveL p5,v1000,fine,tool0 制御システムは p1 を先読みしたとき、すぐにロボットを起動させるのではなく、現在の先読み距離が先読みパラメーターで設定された長さに達したかどうかを確認し、ロボットを起動させるかどうかを決定します。 例 2 MotionWaitAtFinePoint false MoveL p1,v1000,fine,tool0 MoveL p2,v1000,fine,tool0 MoveL p3,v1000,fine,tool0 MoveL p4,v1000,fine,tool0 MoveL p5,v1000,fine,tool0 制御システムは p1 を先読みしたとき、現在の先読み距離が先読みパラメーターで設定された長さに達したかどうかを確認せず、すぐにロボットを起動させます。
--	--

15.4.16.16 IgnoreOverride

説明	目標プロセスの動作速度が速度スライダーの影響を受けないようにします。溶接や接着などパスに沿う動作速度に厳しい要求がある場合に使用します。本命令によって、動作命令に対する速度スライダーを一時的に無効化し、指定された動作命令、軌跡が速度全体の影響を受けないようにします。 本命令は、自動モードと手動モードで使用できます。手動モードでは、動作速度は v250 に制限され、v250 を超える場合は v250 になります。自動モードでは、希望の速度で動作します。
定義	IgnoreOverride On/Off On は、以降の動作命令が速度スライダーの影響を受けないことを示します。 Off は、逆です。
例	例 1 MoveJ p1, v1000 //速度スライダーの影響を受ける IgnoreOverride On MoveJ p2, v1000//速度スライダーの影響を受けない MoveJ p3, v1000 //速度スライダーの影響を受けない IgnoreOverride Off MoveJ p4, v1000 //速度スライダーの影響を受ける 例 2（手動モード） IgnoreOverride On MoveJ p2, v1000//速度スライダーの影響を受けず、v250 で動作する。 MoveJ p3, v100//速度スライダーの影響を受けず、v100 で動作する。 IgnoreOverride Off 例 3（自動モード） IgnoreOverride On MoveJ p2, v1000//速度スライダーの影響を受けず、v1000 で動作する。 MoveJ p3, v100//速度スライダーの影響を受けず、v100 で動作する。 IgnoreOverride Off
注意	影響を受ける動作命令：MoveAbsJ、MoveJ、MoveL、MoveC、MoveCF、MoveT、SearchL、SearchC、TrigL、TrigC、TrigJ。 この命令は即座に命令を実行せず、滑らか軌跡エリアを中断しません。 モーションタスクでのみ使用でき、Inzone ブロックでは使用できません。

15.4.16.17 SingAreaLockAxis4

説明	軸ロックを使用してロボットの手首の特異点を回避します。
定義	SingAreaLockAxis4 on/off On は、4 軸ロックで手首の特異点を回避する機能をオンにします。ロボットの 4 軸が 0 度または ± 180 度にある場合のみ本機能をオンにできます。SingAreaLockAxis4 on の前の動作命令の目標ポイントが上記のポイントにあるか、ロボットの 4 軸がすでに上記の角度にある必要があります。 Off は、本機能をオフにします。 注意：SingAreaLockAxis4 on と SingAreaLockAxis4 off の間の直交動作命令の姿勢は、4 軸の角度を変更しない特別な補間方法を採用します。4 軸の角度を変更しようとする動作命令は、エラーを報告します。同時に、本命令はブロッキング命令に属し、SingAreaLockAxis4 前後の動作命令間の滑らか軌跡エリアを中断します。 現行バージョンの適用機種： 産業用 6 軸ロボット（XB、NB モデル） 協働ロボット xMateCR（5 軸モデルを除く）、xMateSR シリーズ
例	例 1 MoveAbsJ p1,v1000,z50,tool0 SingAreaLockAxis4 on MoveL p2,v1000, z50,tool0 MoveL p3,v1000, z50,tool0 SingAreaLockAxis4 off MoveL p5,v1000, z50,tool0 p1 は、4 軸の角度が 0 度または ± 180 度である必要があります。SingAreaLockAxis4 on を実行すると、4 軸ロック機能がオンになります。MoveL p2、MoveL p3 の姿勢は特別な補間を行い、4 軸の角度を変更しません。SingAreaLockAxis4 off で本機能をオフにします。

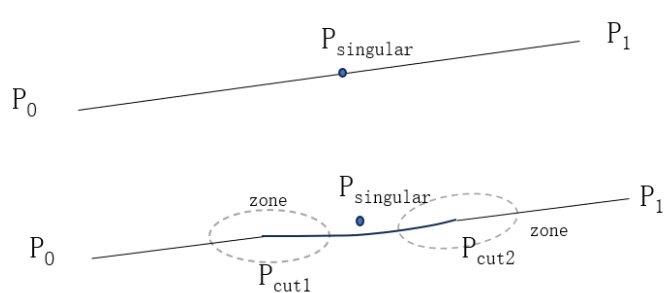
15.4.16.18 SpeedRefresh

説明	動作動プログラム中に速度値を上書きするために使用します。
定義	SpeedRefresh override Override データ型：int 速度オーバーライド値を意味し、制限範囲を超えるとロボットはエラーを報告します。 取り得る値の範囲は 1%~100%です。
例	例 1 SpeedRefresh 70 速度オーバーライド値を 70%に設定します。
注意	RL プログラムが手動でリセットされた場合（PP to Main）、または RL プログラムがリロード、再実行されて次のステップに進むと、SpeedRefresh 命令で設定された速度オーバーライド値はリセットされ、教示操作盤インターフェースで設定したプログラム速度になります。 SpeedRefresh によって設定された速度オーバーライド値は、すぐには反映されません。コマンドを発行してからロボットの速度に影響を与えるまでには、一定のタイムラグが発生します。

15.4.16.19 CSpeedOverride

説明	速度オーバーライド値を読み取ります。
定義	Ret = CSpeedOverride() Ret データ型：int 速度オーバーライド値を取得します。取り得る値の範囲は 1%~100%です。
例	例 1 <pre>int override = CSpeedOverride() print(override)</pre> 現在の速度オーバーライド値が 70%の場合は、70 を返します。

15.4.16.20 SingAreaJointWay

説明	本命令は、軸動作補間を使用して直交動作での特異点回避を行います。 SingAreaJointWay on と SingAreaJointWay off の間の直交動作で、特異点の存在を自動的に検知します。特異点がなければ通常の軌跡に従って動作し、特異点があれば軸動作補間で動作します。詳細は以下の通りです。  上図に示すように、特異点を含む直交動作 P_0-P_1 に対して、制御システムが特異点 $P_{singular}$ を検知した場合、特異点 $P_{singular}$ を中心に、元の軌跡に P_{cut1} と P_{cut2} を追加し、元の軌跡を P_0-P_{cut1}、P_{cut1}-P_{cut2}、P_{cut2}-P_1 に分割します。そのうち、P_0-P_{cut1} と P_{cut2}-P_1 は通常の直交動作を行い、P_{cut1}-P_{cut2} は特異点を通り得るように軸動作（MoveAbsJ）を使用します。3つの軌跡の間は滑らか軌跡を使用し、滑らか軌跡エリアのサイズは zone パラメーターで指定できます。 ロボットの特異点付近の動作では関節角度が大きく動作するので、この命令を使うときはロボットの軌跡が条件を満たすことを確認する必要があります。 現行バージョンの適用機種： 産業用ロボット 6 軸シリーズ（XB、NB モデル）。
定義	SingAreaJointWay on/off,zoneon on/off は、それぞれ軸動作補間機能のオンとオフを意味します。 Zone は、切り分けた後の P_0 - P_{cut1} 、 P_{cut1} - P_{cut2} 、 P_{cut2} - P_1 の間の滑らか軌跡エリア半径を指定します。詳細は、滑らか軌跡エリアを参照してください。
例	例 1 <pre>MoveAbsJ p1,v1000,z50,tool0 SingAreaJointWay on,50 MoveL p2,v1000, z50,tool0 MoveL p3,v1000, z50,tool0 SingAreaJointWay off MoveL p5,v1000, z50,tool0</pre> SingAreaJointWay on,50 は特異点回避をオンにし、滑らか軌跡エリアのサイズを 50mm に設定します。本命令以降の動作命令は、軸動作補間による特異点回避を使用して動作します。 SingAreaJointWay off は軸動作補間による特異点回避をオフにします。

15.4.16.21 SingAreaWrist

説明	本命令は、形態変更補間を使用して直交動作での特異点回避を行います。 SingAreaWrist on と SingAreaWrist off の間の直交動作命令は、形態変更補間で動作します。この場合、ロボットのツールは精確な軌跡に従いますが、手首の形態は変化します。特異点を通過しない場合にも影響します。 形態変更補間を使用する動作は手首姿勢の動作幅が大きいので、この命令を使うときはロボットの動作軌跡が要件を満たすことを確認する必要があります。 現行バージョンの適用機種： 産業用ロボット 6 軸シリーズ (XB、NB モデル) 協働ロボット xMateCR、xMateSR シリーズ
定義	SingAreaWrist on/off, limiton on/off は、それぞれ、形態変更補間機能のオンとオフを意味します。 Limit は、形態変更の最大許容角度を指定します。単位は degree です。
例	例 1 <pre>MoveAbsJ p1,v1000,z50,tool0 SingAreaWrist on,30 MoveL p2,v1000, z50,tool0 MoveL p3,v1000, z50,tool0 SingAreaWrist off MoveL p4,v1000, z50,tool0</pre> SingAreaWrist on,30 は特異点回避機能をオンにし、特異点回避時の形態変更の最大許容角度を 30 度に設定します。本命令以降の動作命令は特異点回避を使用して動作します。p1 と p2、p2 と p3 の間の滑らか軌跡は正常に生成できますが、p3 と p4 の間の滑らか軌跡は生成できません。 SingAreaWrist off は特異点回避機能をオフにします。
注意	直線動作にのみ使用でき、円動作には使用できません。 形態変更補間を使用すると、教示点が特異点範囲内にある場合に教示点が変更されます。動作時の手首の姿勢（特異点を通過する教示点だけでなく、後続の教示点の姿勢も含む。）が教示内容と異なる場合があります。 シングルステップ動作と連続動作で、特異点付近の動作が異なる場合があります。

15.4.16.22 SetRobotJointsMaxAcc

説明	本命令は、ロボットの各軸の最大加速度値を動的に変更するために使用されます。 ロボットの最大加速度値を増加させ、タクトタイムを短縮し、動作速度を向上させる必要がある場合、本コマンドを使用して各軸の最大加速度を設定できます。 この命令の効果は、HMI の運動パラメーター設定にある『最大軸加速度』の設定と同じですが、特定の動作命令に対して適用することができます。
定義	SetRobotJointsMaxAcc jointval1, jointval2, jointval3, jointval4, jointval5, jointval6, jointval7 jointval1~7 データ型：double/int 対応する軸の最大加速度値を表し、単位は $^{\circ}/s^2$ です。
例	例： <pre>SetRobotJointsMaxAcc 80000.0, 70000.0, 70000.0, 150000.0, 150000.0, 20000.0, 20000.0</pre>
注意	2 つの軌道区間の間に滑らか軌跡エリアが存在する場合、そのエリアは最大軸加速度が小さい方の制約に基づいて生成されます。

15.4.16.23 SetRobotJointsMaxJerk

説明	本命令は、ロボットの各軸の最大加加速度を動的に変更するために使用されます。 ロボットの最大化加速度を増加させ、タクトタイムを短縮し、動作速度を向上させる必要がある場合、本コマンドを使用して各軸の最大ジャークを設定できます。 このコマンドの効果は、HMI の運動パラメーター設定にある『最大軸加加速度』の設定と同じですが、特定の動作コマンドに対して適用することができます。
定義	SetRobotJointsMaxJerk jointval1, jointval2, jointval3, jointval4, jointval5, jointval6, jointval7 jointval1~7 データ型：double/int 対応する軸の最大加加速度（ジャーク）を表し、単位は $^{\circ}/s^3$ です。
例	例 1 SetRobotJointsMaxJerk 30000.0, 27000.0, 27000.0, 50000.0, 40000.0, 60000.0, 60000.0
注意	2 つの軌道区間の間に滑らか軌跡エリアが存在する場合、そのエリアは最大軸加加速度が小さい方の制約に基づいて生成されます。

15.4.16.24 ResetJointKineLimit

説明	本命令は、最大軸加加速度および最大軸加加速度を元の値に戻すために使用されます。 SetRobotJointsMaxAcc および SetRobotJointsMaxJerk コマンドと組み合わせて使用し、本コマンドを実行すると、それらのコマンドの効果がリセットされます。
定義	ResetJointKineLimit パラメーターなし
例	例 1 ResetJointKineLimit

15.4.16.25 SetTransmissionOverloadParams

説明	本命令は、ロボットの伝達過負荷係数を動的に変更するために使用されます。 ロボットの伝達過負荷係数を適度に増加させ、タクトタイムを短縮し、動作速度を向上させる必要がある場合、本命令を使用して各軸の伝達オーバーロード係数を設定できます。 このコマンドの効果は、HMI の本体パラメータ設定にある『伝達オーバーロード係数』の設定と同じですが、特定の動作コマンドに対して適用することができます。
定義	SetTransmissionOverloadParams jointval1, jointval2, jointval3, jointval4, jointval5, jointval6, jointval7 jointval1~7 データ型：double/int 対応する軸の伝達過負荷係数を表します。
例	例 1 SetTransmissionOverloadParams 0.95, 0.95, 0.95, 0.95, 1.5, 0.95, 1.0
注意	2 つの軌道区間の間に滑らか軌跡エリアが存在する場合、そのエリアは伝達過負荷係数が小さい方の制約に基づいて生成されます。

15.4.16.26 SetAccRampTime

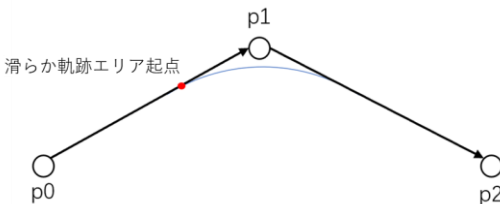
説明	このコマンドは、加速度の上昇時間を設定するために使用されます。ロボットの加速度が最小値から最大値に達するまでにかかる時間を指定します。 数値が小さいほど、ロボットの加速は速くなり、数値が大きいほど、ロボットの加速は滑らかになります。
定義	SetAccRampTime ramptime Ramptime データ型：double/int

	加速度上昇時間の値は、範囲が [0.02, 0.5] 秒で、単位は 秒 (s) です。」
例	例 1 <code>SetAccRampTime 0.15</code>

15.4.16.27 ResetAccRampTime

説明	本命令は、デフォルトの加速度上昇時間に復元するために使用されます。 <code>SetAccRampTime</code> 命令と組み合わせて使用し、実行後は設定された加速度上昇時間がリセットされ、デフォルトの値に戻ります。
定義	<code>ResetAccRampTime</code> パラメーターなし
例	例 1 <code>ResetAccRampTime</code>

15.4.16.28 SetVarValue

説明	滑らか軌跡エリアの起点で変数の代入を実行します。先読み時には代入操作がトリガーされません。
定義	<code>SetVarValue(var1, var2)</code> <code>var2</code> を <code>var1</code> に代入します。 <code>var1</code> データ型：byte、int、double、bool、機能コードが割り当てられていない書き込み専用レジスタ <code>var2</code> データ型：byte、int、double、bool、IO、レジスタ、関数の戻り値、式。
例	例 1  <code>// p0 からスタート</code> <code>MoveL p1</code> <code>SetVarValue(var1, var2) // 滑らかエリアの機転で var2 を var1 に代入します</code> <code>MoveL p2</code>
注意	代入操作の過程で、暗黙の型変換が発生します。 例： <code>int a=1</code> <code>double b=20.22</code> <code>SetVarValue(a, b)</code> 代入操作を実行した後、変数 <code>a</code> の値は 20 になります。

15.4.16.29 SetStopAccRampTime

説明	本命令は、目標位置に到達し停止段階の加速度上昇時間を設定するために使用されます。具体的には、滑らか軌跡のない目標点において、ロボットが停止する際の加速度が最小値から 0 に達するまでの時間を指定します。 数値が小さいほどロボットは早く停止し、数値が大きいほどロボットは滑らかに停止します。 もし加速度上昇時間が目標位置到達時の加速度上昇時間より大きい場合は、大きい方の加速度上昇時間が適用されて停止します。
定義	<code>SetStopAccRampTime rampTime</code> RampTime

	データ型：double/int 目標位置到達時の加速度上昇時間の値は、範囲が [0.01, 1]で、単位は秒 (s) です。
例	例 1 SetAccRampTime 0.15

15.4.16.30 ResetStopAccRampTime

説明	本命令は、目標位置到達時の加速度立上り時間をデフォルトの値に復元するために使用されます。 SetStopAccRampTime 命令と組み合わせて使用し、実行後は設定された加速度上昇時間がリセットされ、デフォルトの値に戻ります。
定義	ResetStopAccRampTime パラメーターなし
例	例 1 ResetStopAccRampTime

15.4.16.31 MotionSupJointTrq

説明	RL プログラム内でロボットの各関節の最大出力閾値を変更するために使用されます。
定義	MotionSupJointTrq J1,J2,J3,J4,J5,J6,J7
例	例 1 MotionSupJointTrq 50,50,70,50,60,60,50

15.4.17 機能命令

15.4.17.1 CRobT

説明	ロボットの位置/姿勢を取得します。この関数を使用する際は、ツールとワークを指定し、指定されたツール座標系での位置/姿勢、軸構成情報、および外部軸情報を取得します。 注意： CRobT を使用する時は、ロボットは停止状態である必要があります。即ち、 CRobT の前の動作命令の滑らか軌跡エリアの設定は fine である必要があります。
定義	戻り値 データ型：robtarget ロボットの現在位置、姿勢、軸構成データ、外部軸情報を返します。 CRobT (Tool, Wobj) Tool データ型：tool ツールを指定します。 Wobj データ型：wobj ワークを指定します。
例	例 1 p2 = CRobT(tool1, wobj2)

15.4.17.2 CJointT

説明	ロボットと外部軸の現在の角度を読み取ります。 注意： CJointT を使用する時は、ロボットは停止状態である必要があります。即ち、 CJointT の前の動作命令の滑らか軌跡エリア設定は fine である必要があります。
定義	戻り値 データ型：jointtarget

	ロボット軸と外部軸の角度値を返します。回転軸単位：degree、直線軸単位：mm。 CJointT ()
例	例 1 VAR jointtarget j2 j2 = CJointT ()

15.4.17.3 CalcJointT

説明	指定した robtarget 変数に対応する関節角度を計算します。
定義	戻り値 データ型：jointtarget 対応する関節角度と外部軸位置を返します。関節角度の単位は degree (°)、外部直線軸の単位はミリメートル (mm)、外部回転軸の単位は degree (°) です。 CalcJointT (Rob_Target, Tool, Wobj) Rob_Target データ型：robtarget 直交座標データを指定します。この点を定義する時に使用されるツールとワークは、CalcJointT 命令で使用されるツールとワークと一致する必要があります。異なる場合は、誤った結果を招く可能性があることに注意してください。 Tool データ型：tool ツールを指定します。robtarget を定義する時に使用されたツールと同じツールを指定する必要があります。 Wobj データ型：wobj ワークを指定します。robtarget を定義する時に使用されたワークと同じワークを指定する必要があります。
例	例 1 jpos2 = CalcJointT(pt1, tool1,wobj2) tool1 がワーク wobj2 で定義された pt1 に到達する時の関節角度を計算し、jpos2 に代入します。

15.4.17.4 CalcRobt

説明	指定された関節角度に対応する直交座標の位置/姿勢を計算します。
定義	戻り値 データ型：robtarget 指定された関節角度に対応する直交座標の位置/姿勢を返します。 CalcRobt (Joint_Target, Tool, Wobj) Joint_Target データ型：Jointtarget 位置/姿勢を計算するために使用される関節角度を指定します。 Tool データ型：tool 位置/姿勢を計算する時に使用するツールを指定します。 Wobj データ型：wobj 位置/姿勢を計算する時に使用するワークを指定します。
例	例 1

	<code>pt1 = CalcRobT (jpos1, tool2,wobj1)</code> 関節角度 <code>jpos1</code> から直交座標の位置と姿勢を計算し、 <code>pt1</code> に代入します。 <code>pt1</code> は、ワーク座標系 <code>wobj1</code> に対してのツール座標系 <code>tool2</code> の位置と姿勢です。
--	--

15.4.17.5 Print

説明	指定された内容を教示操作盤に出力します。ユーザーはこの命令を使用してプログラムをデバッグできます。 Print 関数の入力パラメーターは特別で、数に制限はありませんが少なくとも 1 つが必要です。また、各パラメーターは定義された変数または定数でなければなりません。 システムはこれらの変数を文字列に変換して連結し、最終的にプログラムエディタのデバッグウィンドウに出力します。(デバッグウィンドウを閉じて、 print の情報は記録されます。上限は 500 です。)
定義	Print (var1, var2,)
例	例 1 <pre>counter = 0 while(true) counter++ Print("counter = ",counter) endwhile</pre> 本命令が実行されると、HMI のデバッグウィンドウに以下のメッセージを表示します。 <pre>counter = 1 counter = 2 counter = 3 counter = 4</pre> 注意：文字列を出力する必要がある場合は、二重引用符""を使用して表示したい文字を囲みます。二重引用符の中に二重引用符を入れ子にすることはサポートされていません。

15.4.17.6 Print f

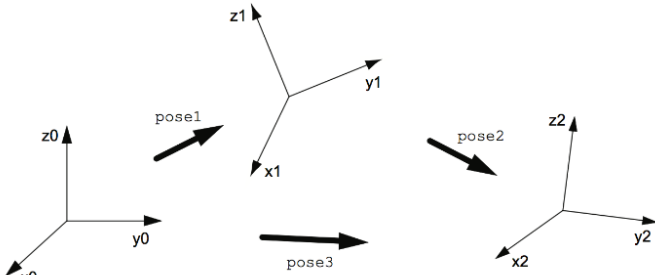
説明	指定された内容をログファイルに出力します。ユーザーはこの命令を使用してプログラムの重要なデータやイベントを記録できます。バックアップ機能を利用して、後でエクスポートして確認することができます。 Print_f 関数の入力パラメーターは特別で、数に制限はありませんが少なくとも 1 つが必要です。また、各パラメーターは定義された変数または定数でなければなりません。 システムはこれらの変数を文字列に変換して連結してから、ログファイルに出力します。
定義	Print_f (var1, var2,)
例	例 1 <pre>counter = 0 while(true) counter++ Print_f("counter = ",counter) // その他の処理 Endwhile</pre> このプログラムを実行した後、ユーザーのログファイルには <code>counter</code> の変化状況が記録されます。後でエクスポートした <code>rl_log</code> ファイルで確認することができます。

```

2  g3log: created log file at: Sat Nov 23 12:54:56 2024
3  +-+ Running on machine: ubuntu
4  +-+ LOG format : [level | mdd hh:mm:ss.uuuuuu] msg
5
6  [RLINFO|11/23 12:55:02.632768 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 1
7  [RLINFO|11/23 12:55:03.637821 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 2
8  [RLINFO|11/23 12:55:04.645517 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 3
9  [RLINFO|11/23 12:55:05.649153 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 4
10 [RLINFO|11/23 12:55:06.661848 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 5
11 [RLINFO|11/23 12:55:07.675333 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 6
12 [RLINFO|11/23 12:55:08.677965 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 7
13 [RLINFO|11/23 12:55:09.682166 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 8
14 [RLINFO|11/23 12:55:10.686953 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 9
15 [RLINFO|11/23 12:55:11.693370 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 10

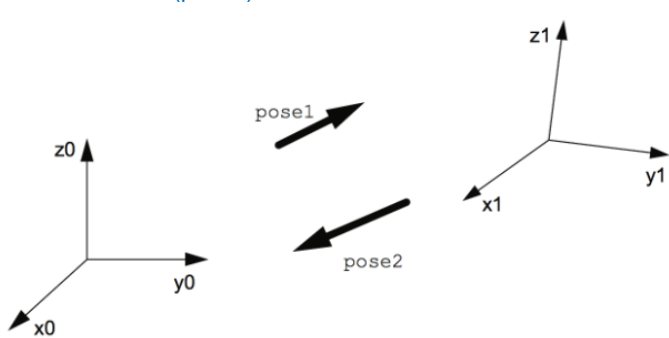
```

15.4.17.7 PoseMult

説明	2 つの位置/姿勢の積を計算します。
定義	pose3 = PoseMult(pose1, pose2) pose1 データ型: pose 1 つ目の位置/姿勢を指定します。 pose2 データ型: pose 2 つ目の位置/姿勢を指定します。 pose3 戻り値 データ型: pose ポーズ積の結果を取得します。
例	 pose1 は座標系 0 に対する座標系 1 の位置/姿勢を示し、pose2 は座標系 1 に対する座標系 2 の位置/姿勢を示しています。 座標系 0 に対する座標系 2 の位置/姿勢 pose3 は、次のように計算できます。 <pre> VAR pose pose1 VAR pose pose2 VAR pose pose3 ... pose3 = PoseMult(pose1, pose2) </pre>

15.4.17.8 PoselInv

説明	PoselInv は、位置/姿勢の逆計算を行います。
定義	pose2 = PoselInv(pose1) pose1 データ型: pose 入力ポーズを指定します。 pose2 戻り値 データ型: pose 位置/姿勢の逆計算の結果を取得します。
例	pose1 は座標系 0 に対する座標系 1 の位置/姿勢を示し、pose2 は座標系 1 に対する座標系 0 の

	位置/姿勢を示します。 pose1 が既知であれば、pose2 は次のように計算できます。 <pre>VAR pose pose1 VAR pose pose2 ... pose2 = PosInv(pose1)</pre> 
--	---

15.4.17.9 GetRobABC

説明	直交座標の点 P のオイラー角 ABC を取得します。 回転順は、初期座標系（動作命令で選択されたワーク座標系）が、自身の X 軸の周りに回転してから、初期座標系の Y 軸の周りに回転し、最後に初期座標系の Z 軸の周りに回転するように定義します。
定義	<pre>double db_arr[3] = GetRobABC(Point [, A, B, C])</pre> Point データ型：robtargert 直交座標データを指定します。 A、B、C データ型：double 直交座標のオイラー角が格納されます。 戻り値 データ型：double 配列（3 次元） 直交座標のオイラー角を取得します。
例	例 1 point0 は直交座標変数であり、オイラー角情報を double 変数に格納するために、次の RL 命令を使用します。 <pre>VAR double Rob_A VAR double Rob_B VAR double Rob_C // point0 のオイラー角を Rob_A B C に代入する GetRobABC(point0, Rob_A, Rob_B, Rob_C)</pre> 例 2 point0 は直交座標変数であり、変数配列を生成して直交座標のオイラー角情報を取得するために、次の RL 命令を使用します。 <pre>double db_arr[3] = GetRobAbc(point0)</pre>

15.4.17.10 SetRobABC

説明	入力したオイラー角 ABC に基づいて、直交座標の点 P の姿勢を変更します。回転順は、初期座標系（運動命令で選択されたワーク座標系）が、自身の X 軸の周りに回転してから、初期座標系の Y 軸の周りに回転し、最後に初期座標系の Z 軸の周りに回転するように定義します。
定義	SetRobABC(Point , A, B, C) Point データ型 : point 姿勢を変更する直交座標データを指定します。 A、B、C データ型 : double オイラー角を指定します。単位は° (degree) です。
例	例 1 point0 は、直交座標変数であり、固定軸 X,Y,Z 軸周りに順番に 30°、60°、90°回転します。 SetRobABC(point0, 30, 60, 90)

15.4.17.11 RotRobABC

説明	入力したオイラー角 ABC に基づいて、直交座標の点 P の姿勢を設定し、オイラー角に沿った回転を行います。回転順は、初期座標系（ P の姿勢）が、自身の X 軸の周りに回転してから、初期座標系の Y 軸の周りに回転し、最後に初期座標系の Z 軸の周りに回転するように定義します。入力した ABC に既存のオイラー角を加えたものとして表現されます。
定義	RotRobABC(Point , A, B, C) Point データ型 : point 姿勢を変更する直交座標データを指定します。 A、B、C データ型 : double オイラー角を指定します。単位は° (degree) です。
例	例 1 point0 は、直交座標変数で、姿勢を X,Y,Z 軸周りに 30°、60°、90°回転させます。 RotRobABC(point0, 30, 60, 90)

15.4.17.12 OpMode

説明	ロボットの現在の動作モードを取得します。
定義	ret = OpMode() ret 戻り値 データ型 : int 0 は未定義、1 は自動モード、2 は手動モードを意味します。
例	例 1 int mode = OpMode() print(mode) 自動モードであれば 1 が返り、手動モードであれば 2 が返ります。

15.4.18 レジスタ命令

15.4.18.1 ReadRegByName

説明	レジスタ名を指定して、レジスタの値を読み取ります。
定義	ReadRegByName(RegData, Value) RegData データ型：読み取り専用のレジスタ変数 設定>通信>レジスタインターフェース機能で定義したレジスタ変数を指定します。 Value データ型：bool/int/double レジスタの値を Value に代入します。レジスタの変数型が Value の変数型と一致しない場合は、暗黙の型変換を行います。
例	例 1 <pre>int tmp_int ReadRegByName(modbus_int_read[6], tmp_int)</pre> modbus_int_read[6]の値を tmp_int 変数に読み取ります。

15.4.18.2 WriteRegByName

説明	レジスタ名を指定して、対応するレジスタに値を書き込みます。 この命令が動作命令の後に使用される場合は、滑らか軌跡を中断せず動作命令の軌道の終点または滑らか軌跡エリアの始点でトリガーされます。詳細については SetDO の例 2 を参照してください。
定義	WriteRegByName(RegData, Value) RegData データ型：書き込み専用のレジスタ変数 設定>通信>レジスタインターフェース機能で定義したレジスタ変数を指定します。 Value データ型：bool/int/double Value の値をレジスタに代入します。レジスタの変数型が Value の変数型と一致しない場合は、暗黙の型変換を行います。
例	例 1 <pre>WriteRegByName(modbus_int_write[6], 200)</pre> 200 という値を、modbus_int_write[6]に対応するレジスタに書き込みます。

15.4.18.3 ReadRegByteByName

説明	レジスタ名を指定して、対応するバイト値を読み取ります。
定義	ReadRegByteByName(RegData, Value, byteFlag) RegData データ型：読み取り専用の int16/int32 型レジスタ 設定>通信>レジスタインターフェース機能で定義したレジスタ変数を指定します。 Value データ型：byte レジスタのバイト値を Value に読み取ります。 byteFlag バイトフラグビットを指定します。取り得る値の範囲は 1~4 で、1 は下位バイトで、4 は上位バイトを意味します。
例	例 1 <pre>byte tmp_value ReadRegByteByName(modbus_reg, tmp_value,1)</pre>

	modbus_reg レジスタの最初のバイトを、tmp_value 変数に読み取ります。
--	--

15.4.18.4 WriteRegByteByName

説明	レジスタ名を指定して、対応するバイト値を書き込みます。 この命令が動作命令の後に使用される場合は、滑らか軌跡を中断せず動作命令の軌道の終点または滑らか軌跡エリアの始点でトリガーされます。詳細については SetDO の例 2 を参照してください。
定義	WriteRegByteByName(RegData, Value, byteFlag) RegData データ型：書き込み専用の int16/int32 型レジスタ 設定＞通信＞レジスタインターフェース機能で定義したレジスタ変数を指定します。 Value データ型：byte Value の値を対応するレジスタに書き込みます。 byteFlag バイトフラグビットを指定します。取り得る値の範囲は 1~4 で、1 は下位バイトで、4 は上位バイトを意味します。
例	例 1 WriteRegByteByName(modbus_reg, 200,2) 200 という値を、modbus_reg の 2 バイト目に書き込みます。

15.4.19 先端ツール命令

15.4.19.1 JodellGriplnit

説明	Jodell 電動グリッパーの初期化命令です。
定義	JodellGriplnit ID, wait_time ID データ型：int 電動グリッパーの ID を指定します。指定した Jodell 電動グリッパーの初期化を行います。 Wait_time データ型：int 初期化完了までの待ち時間を指定します。タイムアウトの場合はエラーを報告します。単位は s です。
例	

15.4.19.2 JodellGripMove

説明	Jodell 電動グリッパーの動作命令です。
定義	JodellGripMove ID, Pos, Vel, Trq ID データ型：int 電動グリッパーの ID を指定します。 Pos データ型：int 目標位置を指定します。単位なし。設定範囲は 0~255 です。 Vel データ型：int 電動グリッパーの動作速度を指定します。単位なし。設定範囲は 0~255 です。 Trq データ型：int 電動グリッパーの動作トルクを指定します。単位なし。設定範囲は 0~255 です。

例	
---	--

15.4.19.3 JodellGripStatus

説明

Jodell 電動グリッパーの状態を取得します。

定義

JodellGripStatus ID, Pos, Vel, Trq, Contact

ID

データ型：Int

電動グリッパーの ID を指定します。

Pos

データ型：Int

電動グリッパー現在の位置を取得します。単位なし。データ範囲は 0～255 です。

Vel

データ型：Int

電動グリッパーの動作速度を取得します。単位なし。データ範囲は 0～255 です。

Trq

データ型：Int

電動グリッパーの動作トルクを取得します。単位なし。データ範囲は 0～255 です。

Contact

データ型：Int

電動グリッパーの動作状態を取得します。単位なし。データ範囲は 0～255 であり、そのうち、bit6-7 は、電動グリッパーが物体を検出したか否かの状態を示します。

Bit	名前	値/説明
0	gAct	0：電動グリッパーがリセット状態 1：電動グリッパーがイネーブル状態
2	gMode	0：パラメーター制御モード 1：パラメーター制御モードではない
3	gGTO	0：停止している 1：目標位置に移動している
4-5	gSTA	0：電動グリッパーがリセットまたは巡回検査中 1：有効化中 2：未使用 3：有効化完了
6-7	gOBJ	0：動作中 1：開いた状態で、指定位置に到達したときに物体に接触して停止している 2：閉じた状態で、指定位置に到達したときに物体に接触して停止したている 3：指定位置に到達したが対象を検出しなかった

15.4.19.4 JodellSuckInit

説明	Jodell 吸盤の初期化命令です。
定義	JodellSuckInit ID ID データ型：int 吸盤の ID を指定します。指定した ID の吸盤を初期化し、正しく接続されているかを検出します。
例	

15.4.19.5 JodellSuckSet

説明	Jodell 吸盤の動作命令です。本命令により、吸盤は直ちに設定されたパラメーターに従って動作します。
定義	JodellSuckSet ID,CH1_enable,CH1_VacMin, CH1_VacMax, CH1_Waittime, CH2_enable, CH2_VacMin, CH2_VacMax, CH2_Waittime ID データ型：int 吸盤の ID を指定します。 CH1_enable データ型：int 第 1 チャンネルを有効にするかどうかを指定します。1 は有効、0 は無効を意味します。 CH1_VacMin データ型：int 第 1 チャンネルの最低真空設定値を指定します。範囲は 0～255 です。0 は全真空を示し、100 より大きいと吸盤が解放されることを示します。実際の真空度がこの閾値より小さくなるとポンピングを停止させます。 CH1_VacMax データ型：int 第 1 チャンネルの最大真空設定値を指定します。範囲は 0～255 です。0 は全真空を示し、100 より大きいと吸盤が解放されることを示します。実際の真空度がこの閾値より大きくなるとポンピングを開始させます。 CH1_Waittime データ型：double 第 1 チャンネルのタイムアウト設定値を指定します。 CH2_enable データ型：int 第 2 チャンネルを有効にするかどうかを指定します。1 は有効、0 は無効を意味します。 CH2_VacMin データ型：int 第 2 チャンネルの最低真空設定値を指定します。範囲は 0～255 です。0 は全真空を示し、100 より大きいと吸盤が解放されることを示します。実際の真空度がこの閾値より小さくなるとポンピングを停止させます。 CH2_VacMax データ型：int 第 2 チャンネルの最大真空設定値を指定します。範囲は 0～255 です。0 は全真空を示し、100 より大きいと吸盤が解放されることを示します。実際の真空度がこの閾値より大きくなるとポンピングを開始させます。 CH2_Waittime データ型：double 第 2 チャンネルのタイムアウト設定値を指定します。
例	

15.4.19.6 JodellSuckStatus

説明	Jodell 吸盤の状態を取得します。																		
定義	JodellSuckStatus ID, Vac1, Contact1, Time_Err1, Vac2, Contact2, Time_Err2 ID データ型：int 吸盤の ID を指定します。 Vac1 データ型：int 第 1 チャンネルの現在の実際真空度を取得します。データ範囲は 0～100 です。 Contact1 データ型：int 第 1 チャンネルの現在の状態情報を取得します。データ範囲は 0～255 であり、そのうち、bit6-7 は、物体を検出したか否かを示します。詳細の状態情報については、下表の通りです。 <table><tr><th>Bit</th><th>名前</th><th>値/説明</th></tr><tr><td>0</td><td>gAct</td><td>0：吸盤が無効 1：吸盤が有効</td></tr><tr><td>2</td><td>gMode</td><td>0：自動制御モード 1：ハイレベル制御モード</td></tr><tr><td>3</td><td>gGTO</td><td>0：調整停止 1：圧力または真空度が調整されている</td></tr><tr><td>4-5</td><td>gSTA</td><td>0：吸盤が有効化していない状態 1 と 2：は使用されていない状態 3：吸盤が有効化している状態</td></tr><tr><td>6-7</td><td>gOBJ</td><td>0：空気圧が最低圧を下回った 1：ワークが検出され、最低圧力値に達した 2：ワークが検出され、最高圧力値に達した 3：対象が検出されず、対象物が無いか外れた</td></tr></table> Time_Err1 データ型：int 第 1 チャンネルがタイムアウトしたか否かを取得します。 Vac2 データ型：int 第 2 チャンネルの現在の実際真空度を取得します。データ範囲は 0～100 です。 Contact2 データ型：int 第 2 チャンネルの現在の状態情報データを取得します、データ範囲は 0～255 であり、そのうち、bit6-7 は、物体を検出したか否かを示します。詳細の状態情報については、上表と同じです。 Time_Err2 データ型：int 第 2 チャンネルがタイムアウトしたか否かを取得します。	Bit	名前	値/説明	0	gAct	0：吸盤が無効 1：吸盤が有効	2	gMode	0：自動制御モード 1：ハイレベル制御モード	3	gGTO	0：調整停止 1：圧力または真空度が調整されている	4-5	gSTA	0：吸盤が有効化していない状態 1 と 2：は使用されていない状態 3：吸盤が有効化している状態	6-7	gOBJ	0：空気圧が最低圧を下回った 1：ワークが検出され、最低圧力値に達した 2：ワークが検出され、最高圧力値に達した 3：対象が検出されず、対象物が無いか外れた
Bit	名前	値/説明																	
0	gAct	0：吸盤が無効 1：吸盤が有効																	
2	gMode	0：自動制御モード 1：ハイレベル制御モード																	
3	gGTO	0：調整停止 1：圧力または真空度が調整されている																	
4-5	gSTA	0：吸盤が有効化していない状態 1 と 2：は使用されていない状態 3：吸盤が有効化している状態																	
6-7	gOBJ	0：空気圧が最低圧を下回った 1：ワークが検出され、最低圧力値に達した 2：ワークが検出され、最高圧力値に達した 3：対象が検出されず、対象物が無いか外れた																	
例																			

15.4.19.7 RMRGMGripInit

説明	RM 電動グリップ RM-RGM シリーズの初期化命令です。
定義	RMRGMGripInit ID ID データ型：int 電動グリップの ID を指定します。
例	

15.4.19.8 RMCGripInit

説明	RM 電動グリップ RM-C シリーズの初期化命令です。
定義	RMCGripInit ID ID データ型：int 電動グリップの ID を指定します。
例	

15.4.19.9 RMRGMGripPosMove

説明	RM 電動グリップ RM-RGM シリーズの位置モードでの動作命令です。
定義	RMRGMGripPosMove ID, Pos, Vel, Acc, PCheck ID データ型：int 電動グリップの ID を指定します。 Pos データ型：double 目標距離を指定します。単位は mm で、設定範囲は-2000.0～2000.0 です。 Vel データ型：double 電動グリップの動作速度を指定します。単位は mm/s で、設定範囲は 0.01～1000.0 です。 Acc データ型：double 電動グリップの加速度を指定します。単位は mm/s ² で、設定範囲は 0.01-2000.0 です。 PCheck データ型：double 位置決め範囲を指定します。単位は mm で、設定範囲は 0.01～10.0 です。
例	

15.4.19.10 RMCGripPosMove

説明	RM 電動グリップ RMC シリーズの位置モードでの動作命令です。
定義	RMCGripPosMove ID, Pos, Vel, Acc, PCheck ID データ型：int 電動グリップの ID を指定します。 Pos データ型：double 目標距離を指定します。単位は mm で、設定範囲は-2000.0～2000.0 です。 Vel データ型：double 電動グリップの動作速度を指定します。単位は mm/s で、設定範囲は 0.01～1000.0 です。 Acc データ型：double

	電動グリップの加速度を指定します。単位は mm/s^2 で、設定範囲は 0.01-2000.0 です。 PCheck データ型：double 位置決め範囲を指定します。単位は mm で、設定範囲は 0.01~10.0 です。
例	

15.4.19.11 RMRGMGripTrqMove

説明	RM 電動グリップ RM-RGM シリーズのトルクモードでの動作命令です。
定義	RMRGMGripTrqMove ID, Pos, Vel, Acc, Trq, PCheck, TCheck ID データ型：int 電動グリップの ID を指定します。 Pos データ型：double 目標距離を指定します。単位は mm で、設定範囲は -2000.0~2000.0 です。 Vel データ型：double 電動グリップの動作速度を指定します。単位は mm/s で、設定範囲は 0.01~1000.0 です。 Acc データ型：double 電動グリップの加速度を指定します。単位は mm/s^2 で、設定範囲は 0.01-2000.0 です。 Trq データ型：double トルク範囲を指定します。単位は Nm で、設定範囲は 0.01~100.0 です。 PCheck データ型：double 位置決め範囲を指定します。単位は mm で、設定範囲は 0.01~10.0 です。 TCheck データ型：double 時間範囲を指定します。単位は s で、設定範囲は 0.01~1000.0。
例	

15.4.19.12 RMCGripTrqMove

説明	RM 電動グリップ RMC シリーズのトルクモードの動作命令です。
定義	RMCGripTrqMove ID, Pos, Vel, Acc, Trq, PCheck, TCheck ID データ型：int 電動グリップの ID を指定します。 Pos データ型：double 目標距離を指定します。単位は mm で、設定範囲は -2000.0~2000.0 です。 Vel データ型：double 電動グリップの動作速度を指定します。単位は mm/s で、設定範囲は 0.01~1000.0 です。 Acc データ型：double 電動グリップの加速度を指定します。単位は mm/s^2 で、設定範囲は 0.01-2000.0 です。 Trq データ型：double トルクを指定します。単位は Nm で、設定範囲は 0.01~100.0 です。

	PCheck データ型：double 位置決め範囲を指定します。単位は mm で、設定範囲は 0.01～10.0 です。 TCheck データ型：double 時間範囲を指定します。単位は s で、設定範囲は 0.01～1000.0。
例	

15.4.19.13 RMRGMGripStatus

説明	RM 電動グリップ RM-RGM シリーズの状態取得命令です。
定義	RMRGMGripStatus ID, Pos, Vel, Trq, Reach, Err ID データ型：int 電動グリップの ID を指定します。 Pos データ型：double 電動グリップの位置を取得します。単位は mm です。 Vel データ型：double 電動グリップの速度を取得します。単位は mm/s です。 Trq データ型：double 電動グリップの出力トルクを取得します。単位は%です。 Reach データ型：bool 電動グリップが所定の位置にあるか否かを取得します。 Err データ型：int 電動グリップのエラーコードを取得します。
例	

15.4.19.14 RMCGripStatus

説明	RM 電動グリップ RMC シリーズの状態取得命令です。
定義	RMCGripStatus ID, Pos, Vel, Trq, Reach, Err ID データ型：int 電動グリップの ID を指定します。 Pos データ型：double 電動グリップの位置を取得します。単位は mm です。 Vel データ型：double 電動グリップの速度を取得します。単位は mm/s です。 Trq データ型：double 電動グリップの出力トルクを取得します。単位は%です。 Reach データ型：bool 電動グリップが所定の位置にあるか否かを取得します。 Err

	データ型：int 電動グリッパのエラーコードを取得します。
例	

15.4.19.15 RMRGMResetErr

説明	RM 電動グリッパ RMR-GM シリーズのエラーリセット命令です。
定義	RMRGMResetErr ID ID データ型：int リセットする電動グリッパの ID を指定します。
例	

15.4.19.16 RMCResetErr

説明	RM 電動グリッパ RMC シリーズのエラーリセット命令です。
定義	RMCResetErr ID ID データ型：int リセットする電動グリッパの ID を指定します。
例	

15.4.19.17 RobotiqGriplnit

説明	Robotiq 電動グリッパ 2F-85 シリーズ の初期化命令です。
定義	RobotiqGriplnit ID ID データ型：int 初期化するグリッパの ID を指定します。
例	

15.4.19.18 RobotiqGripGetStatus

説明	Robotiq 電動グリッパ 2F-85 シリーズ の状態取得命令です。
定義	RobotiqGripGetStatus ID, Pos, gOBJ, Err ID データ型：int 変数 状態を取得したいグリッパの ID を指定します。 Pos データ型：int 電動グリッパの位置を取得します。範囲は 0-255 です。 gOBJ データ型：int 電動グリッパの接触状態を取得します。 Err データ型：int 電動グリッパのエラーコードを取得します。
例	

15.4.19.19 RobotiqGripMove

説明	Robotiq 電動グripper 2F-85 シリーズ の動作命令です
定義	obotiqGripMove ID, Pos, Vel, Trq ID データ型 : int 動作させるグripperの ID を指定します。 Pos データ型 : Int グripperの位置を指定します。設定範囲は 3-227 です。 Vel データ型 : int グripperの動作速度を指定します。設定範囲は 0-255 です。 Trq データ型 : Int グripperのトルクを指定します。設定範囲は 0-255 です。
例	

15.4.19.20 DhGripInit

説明	DH 電動グripper PGI-140-80 シリーズ の初期化命令です。
定義	DhGripInit ID Time_wait ID データ型 : int 初期化する電動グripperの ID を指定します。 Time_wait データ型 : doubl 単位は s で、設定範囲は 0-10 です。
例	

15.4.19.21 DhGripGetStatus

説明	DH 電動グripper PGI-140-80 シリーズ の状態取得命令です。
定義	DhGripGetStatus ID, Pos, gOBJ ID データ型 : int グripperの ID を指定します。 Pos データ型 : int グripperの位置を取得します。範囲は 0-1000 です。 gOBJ データ型 : int グripperの接触状態を取得します。 0 : 動作中 1 : 目標位置に到達 2 : 物体を把持 3 : 物体が落下 4、ID が正常に初期化されていない
例	

15.4.19.22 DhGripMove

説明	DH 電動グripper PGI-140-80 シリーズ の動作命令です。
定義	DhGripMove ID, Pos, Vel, Trq ID データ型 : int グripper の ID を指定します。 Pos データ型 : int グripper の位置を指定します。設定範囲は 0-1000 です。 Vel データ型 : int グripper の動作速度を指定します。設定範囲は 1-100 です。 Trq データ型 : int グripper のトルクを指定します。設定範囲は 20-100 です。
例	

16 付録

16.1 ユーザー権限の詳細

類別	機能	オペレータ	教示者	プログラマー	管理者	スーパー管理者
プロジェクト関連とプログラム編集	プロジェクト設定 (新規作成、インポート、エクスポート、ロード、復元、名前を付けて保存、デフォルトプロジェクトの設定)	N	N	Y	Y	Y
	プロジェクトの表示 (プログラム、IO、変数、事前定義パラメーター、その他のオブジェクトデータを含む)	Y	Y	Y	Y	Y
	プロジェクトの編集 (プログラム編集やツールなどのオブジェクト設定を含む)	N	N	Y	Y	Y
	教示 (RL エディターインターフェース、変数一覧の位置の種類、位置一覧を含む)	N	Y	Y	Y	Y
ロボット動作とプログラム実行	手動/自動モードの切替	Y	Y	Y	Y	Y
	モーター電源オン/オフ	Y	Y	Y	Y	Y
	プログラムオン/オフ	Y	Y	Y	Y	Y
	プログラムのサイクルモードの切替	Y	Y	Y	Y	Y
	プログラム実行速度の調整	N	Y	Y	Y	Y
	シングルステップデバッグ操作	N	Y	Y	Y	Y
	ロボットのジョグ/ドラッグ	N	Y	Y	Y	Y
状態監視	実行データの表示	Y	Y	Y	Y	Y
	IO 信号の設定	N	Y	Y	Y	Y
	レジスタ値の設定	N	Y	Y	Y	Y
	PERS 変数値の設定	N	Y	Y	Y	Y
設定	コントローラー設定 - 基本設定	N	N	Y	Y	Y
	コントローラー設定 - 詳細設定	N	N	Y	Y	Y
	コントローラー設定 - 認証設定	N	N	N	Y	Y
	HMI 設定 - 基本設定	N	N	N	Y	Y
	HMI 設定 - 教示操作盤モード	N	N	Y	Y	Y
	ユーザーグループ	Y	Y	Y	Y	Y
	ゼロキャリブレーション	N	N	Y	Y	Y
	ベース座標系の設定	N	N	Y	Y	Y
	動力学設定	N	N	Y	Y	Y
	本体パラメーター設定	N	N	Y	Y	Y

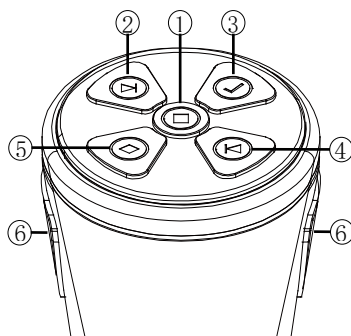
	運動パラメーター設定	N	N	Y	Y	Y
	力制御パラメーター設定	N	N	Y	Y	Y
	クイック移動	N	Y	Y	Y	Y
	電子銘板	N	N	Y	Y	Y
	エラーコードのフィルタリング	N	N	Y	Y	Y
	カスタムボタン	N	N	Y	Y	Y
通信	システム IO	N	N	Y	Y	Y
	外部通信	N	N	Y	Y	Y
	IO デバイス	N	N	Y	Y	Y
	バスデバイス	N	N	Y	Y	Y
	レジスタ	N	N	Y	Y	Y
	先端ツール	N	N	Y	Y	Y
	RCI 設定	N	N	Y	Y	Y
	xPanel 設定	N	N	Y	Y	Y
	電動グリッパ/吸盤	N	N	Y	Y	Y
	シリアルポート設定	N	N	Y	Y	Y
	エンコーダー	N	N	Y	Y	Y
	OPC-UA	N	N	Y	Y	Y
安全	ソフトウェアリミット	N	N	Y	Y	Y
	バーチャルフェンス	N	N	Y	Y	Y
	衝突検知	N	N	Y	Y	Y
	安全エリア	N	Y	Y	Y	Y
	安全モニタ	N	N	Y	Y	Y
	協働モード	N	N	Y	Y	Y
	安全位置	N	Y	Y	Y	Y
プロセスパッケージ	コンベア	N	N	Y	Y	Y
	走行軸	N	N	Y	Y	Y
ログ	ログ検索	Y	Y	Y	Y	Y
	診断設定	N	N	N	N	Y
オプション	接続/概要/デモンストレーション	Y	Y	Y	Y	Y
	ソフトウェアアップグレード - コントローラーアップグレード /その他の設定	N	N	N	Y	Y
	ソフトウェアアップグレード - HMI アップグレード/ コントローラーバックアップ	Y	Y	Y	Y	Y
	エクスポート	Y	Y	Y	Y	Y
	インポート	N	N	N	Y	Y
	ファイルマネージャ	N	N	N	Y	Y

16.2 協働ロボットのハンドルの紹介

16.2.1 ER シリーズ

xMate ER シリーズロボットの先端には、インテリジェントなハンドルが統合されています。ドラッグモードでハンドル上のボタンを使用することで、素早く位置/姿勢教示と連続軌跡教示を行うことができます。

ハンドルボタンの定義：

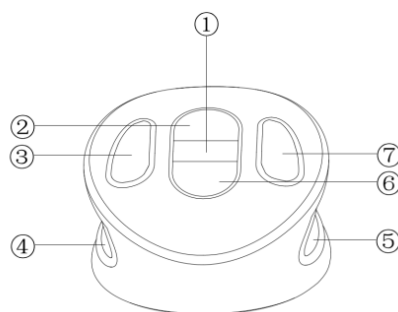


番号	定義
①	当該ポイントでの教示点更新、軌跡記録の開始/停止
②	次へ
③	一覧にポイント/軌跡を追加、ポップアップウィンドウの確認、ポップアップウィンドウのキャンセル
④	前へ
⑤	一覧からポイント/軌跡の削除、ポップアップウィンドウのキャンセル
⑥	ドラッグモードで、2つのイネーブルボタンが同時に押されているとドラッグ機能がオン

16.2.2 CR シリーズ

xMate CR シリーズロボットのハンドル上のボタンを使用することで、ドラッグモードで、素早く位置/姿勢教示と連続軌跡教示を行うことができます。

ハンドルボタンの定義：



番号	定義
①	一覧にポイント/軌跡を追加、ポップアップウィンドウの確認
②	一覧内のカーソルを次へ移動
③	一覧からポイント/軌跡の削除、ポップアップウィンドウのキャンセル

④	ドラッグモードで、2 つのイネーブルボタンが同時に押されているとドラッグ機能がオン
⑤	
⑥	一覧内のカーソルを前へ移動
⑦	当該ポイントでの教示点更新、ポップアップウィンドウの確認、軌跡記録の開始/停止

16.3 位置、パスの教示（協働ロボットハンドルを使用する場合）

16.3.1 位置の教示

操作パネルでドラッグをオンにすると、ロボットは自動的にモーター電源を投入してドラッグモードを有効にします。**RobotAssist** ソフトウェアとハンドルを使って以下の操作を行います。

手順	説明
1. プロジェクトを作成/ロードし、位置一覧インターフェースに入る；	終了ボタンは、 RobotAssist ソフトウェアの現在開いているページが位置一覧/パス一覧の時のみに反応します。
2. ハンドルの 2 つのイネーブルボタンを同時に押し、ロボットを任意位置までドラッグしてドラッグイネーブルを放し、エンドハンドルボタンの「ポイント追加」ボタンを押す；	位置一覧の末尾に、現在のポイント情報を持つ教示点が追加され、カーソルは新しいポイントにジャンプします。
3. ハンドルの「前へ/次へ」ボタンを押す	位置一覧のカーソルは、『前へ/次へ』を使用して移動します。
4. 位置一覧で更新したいポイントを選択し、ロボットを別の位置までドラッグしてドラッグイネーブルボタンを放し、ハンドルの「ポイント更新」ボタンを押す；	位置を更新するときは、ポップアップウィンドウが出ます。「はい」ボタンを押すと、現在選択されているポイントが現在の位置/姿勢に更新されます。「いいえ」ボタンを押すと、ポップアップウィンドウがキャンセルされ、現在選択されているポイントは変更されません。
5. 位置一覧で削除したいポイントを選択し、ハンドルの「ポイント削除」ボタンを押して確認する；	ポイントを削除するときは、ポップアップウィンドウが出ます。「はい」ボタンを押すと、現在選択されているポイントが位置一覧から削除されます。「いいえ」ボタンを押すと、ポップアップウィンドウがキャンセルされ、現在選択されているポイントは削除されません。

16.3.2 パスの教示

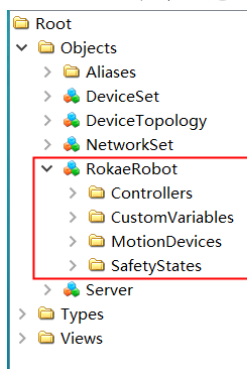
操作パネルでドラッグモードをオンにすると、ロボットは自動的にモーター電源を投入してドラッグモードを有効にします。**RobotAssist** ソフトウェアとハンドルを使って、以下の操作を行います。

手順	説明
1. プロジェクトを作成/ロードし、ロボットを任意の初期位置に調整して、パス一覧インターフェースに入る；	終了ボタンは、 RobotAssist ソフトウェアの現在開いているページが位置一覧/パス一覧の時のみに反応します。
2. ハンドルの「パス追加」ボタンを押す；	パス一覧の末尾に新しいパスが追加され、カーソルは追加されたパスにジャンプします。
3. ハンドルの「前へ/次へ」ボタンを押す；	パス一覧のカーソルは、『前へ/次へ』を使用して移動し

	ます。
4. パス一覧からパスを選択して記録を始める。ハンドルの「軌跡記録開始」ボタンを押し、ハンドルの 2 つのイネーブルボタンを同時に押して、ロボットをドラッグして軌跡を記録する；	「軌跡記録開始」ボタンを押すと、軌跡記録を開始します。「軌跡記録停止」ボタンを押すと軌跡記録が終了し自動的に保存されます。
5. パス一覧で削除したいパスを選択し、ハンドルの「ポイント削除」ボタンを押して確認する；	パスを削除するときは、ポップアップウィンドウが出ます。「はい」ボタンを押すと、現在選択されているパスがパス一覧から削除されます。「いいえ」ボタンを押すと、ポップアップウィンドウがキャンセルされ、現在選択されているパスは削除されません。

16.4 OPC-UA Robotics モデル

xCore コントローラシステムの OPC-UA サーバーはデフォルトで OPC 40010-1 OPC-UA for Robotics, Part 1: Vertical Integration 基準に規定されているすべての必須オプションをサポートしています。トップレベルドメインは以下の通りです。

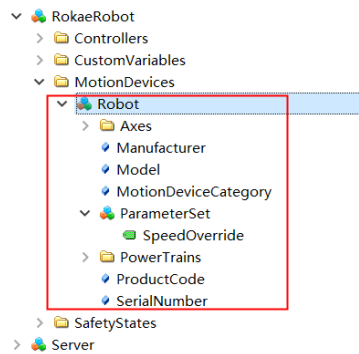


種類	名前 (Browse Name)	説明
MotionDeviceSystem	RokaRobot	各サーバーには、Objects ノードの下に「RokaRobot」と呼ばれる MotionDeviceSystem 型のインスタンスがあります。

MotionDeviceSystem 子ノードは以下の通りです。

種類	名前 (Browse Name)	説明
MotionDevices	MotionDevices	MotionDeviceType インスタンスを保持できるコンテナ
Controllers	Controllers	MotionDeviceType インスタンスを保持できるコンテナ
SafetyStates	SafetyStates	ControllerType インスタンスを保持できるコンテナ
CustomVariables	CustomVariables	カスタム変数のコンテナ。このノードは Robotics 基準モデルの拡張で、すべてのユーザー定義変数はこのノードの下にあります。

16.4.1 MotionDevices モデル

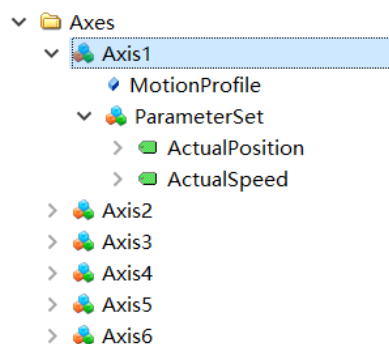


種類	名前 (Browse Name)	説明
MotionDevice	Robot	「Robot」と呼ばれる MotionDeviceType 型のインスタンス

MotionDevice 子ノードは以下の通りです。

種類	名前 (Browse Name)	説明
MotionDeviceCategory	MotionDeviceCategory	ISO 8373 に規定される運動デバイスのタイプであり、ARTICULATED_ROBOT、即ち関節ロボットに設定します
Manufacturer	Manufacturer	メーカー
Model	Model	ロボットモデル
ProductCode	ProductCode	製品番号、まだサポートされていません。
SerialNumber	SerialNumber	シリアル番号、まだサポートされていません。
ParameterSet	ParameterSet/SpeedOverride	プログラム速度、RL プログラム速度 ： 1～100%

16.4.1.1 Axes 子ノード

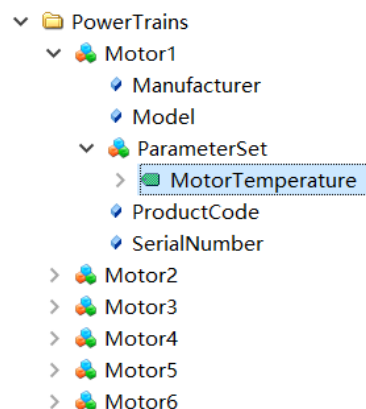


種類	名前 (Browse Name)	説明
Axis	Axis%1	- 各インスタンスが1つの軸に対応し、6軸ロボットは6つのインスタンスがあります。 - 軸番号に応じて名前を付けます。例えば、Axis1、Axis2

Axes 子ノードは以下の通りです。

種類	名前 (Browse Name)	説明
MotionProfile	MotionProfile	軸種類、通常は 1 (ROTARY)
ParameterSet	ParameterSet/ActualPosition	現在の軸位置、単位は degree
	ParameterSet/ActualSpeed	現在の軸速度、単位は degree/s

16.4.1.2 PowerTrains 子ノード

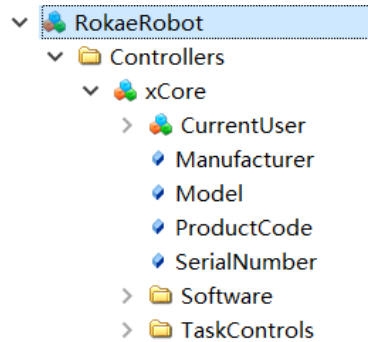


種類	名前 (Browse Name)	説明
PowerTrain	Motor%1	- 各インスタンスは 1 つの軸動力ユニットに対応し、6 軸ロボットは 6 つのインスタンスがあります。(6つの MotorType インスタンスを含む) - 軸番号に応じて名前を付けます。 例えば、Motor1、Motor2

各 Motor ノードは以下の通りです。

種類	名前 (Browse Name)	説明
Manufacturer	Manufacturer	メーカー、まだサポートされていません。
Model	Model	ロボットモデル、まだサポートされていません。
ProductCode	ProductCode	製品番号、まだサポートされていません。
SerialNumber	SerialNumber	まだサポートされていません。
ParameterSet	ParameterSet/MotorTemperatur	モーターの温度、まだサポートされていません。

16.4.2 Controllers モデル

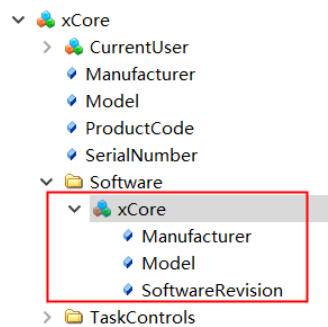


種類	名前 (Browse Name)	説明
ContorllerType	xCore	1. 各ロボットには xCore と呼ばれる ContorllerType インスタンスがあります。

各 ContorllerType 子ノードは以下の通りです。

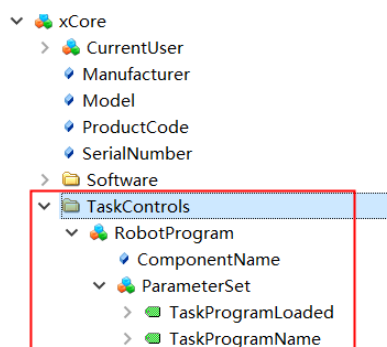
種類	名前 (Browse Name)	説明
Manufacturer	Manufacturer	メーカー
Model	Model	ロボットモデル
ProductCode	ProductCode	製品番号、まだサポートされていません。
SerialNumber	SerialNumber	シリアル番号、まだサポートされていません。
CurrentUser	CurrentUser	まだサポートされていません。
Software	Software	ControllerType インスタンスを保持するコンテナ
TaskControls	TaskControls	TaskControlType インスタンスを保持するコンテナ

16.4.2.1 Software 子ノード



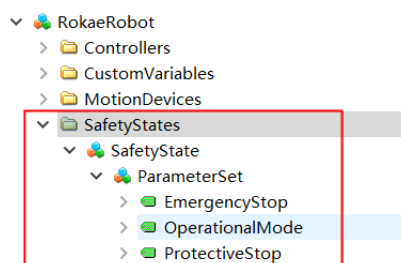
種類	名前 (Browse Name)	説明
Software	xCore	xCore 制御システム

16.4.2.2 TaskControls 子ノード



種類	名前 (Browse Name)	説明
TaskControl	RobotProgram	RobotProgram と呼ばれる TaskControlType インスタンス

16.4.3 SafetyStates

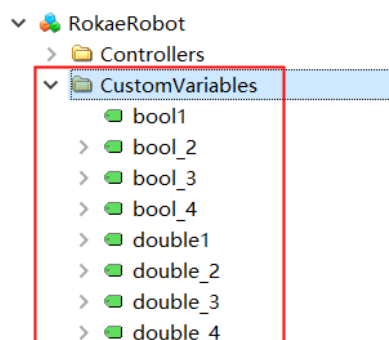


種類	名前 (Browse Name)	説明
SafetyState	SafetyState	SafetyState と呼ばれる SafetyStateType インスタンス

SafetyState 子ノードは以下の通りです。

種類	名前 (Browse Name)	説明
ParameterSet	ParameterSet/EmergencyStop	NULL
	ParameterSet/OperationalMode	操作モード、 列挙値： 0-Other 1-Manual Reduced Speed 2-Manual High Speed 3-Automatic 4-Automatic External 現在の制御システムの手動モード と自動モードは、それぞれ上記列 挙の 1 と 3 に対応
	ParameterSet/ProtectiveStop	コントローラー内部のセーフティ ーゲート停止または保護停止状態

16.4.4 CustomVariables



11.13 節で設定したカスタム変数は、すべてこのノードの下にあります。クライアントは、この変数を読み書きして、コントローラーデータを操作できます。

17 トラブルシューティング

17.1 制御システムのエラーコード

17.1.1 1XXXX

エラーコード	エラー内容	原因	修復方法
10000	HMI 要求パケットの解析エラー	HMI 要求パケットプロトコルが正しくありません	HMI と制御システムのバージョンが一致していることを確認してください。
10001	JOG の起動に失敗しました	1.手動モードではありません; 2.ロボットが通電していません; 3.ロボット動作中; 4.制御モードが位置決めモードになっていません	ロボットが手動モードで通電していることを確認してください
10002	クイック調整の開始に失敗しました	1.手動モードではありません; 2.ロボットが通電していません; 3.ロボット動作中; 4.制御モードが位置決めモードになっていません	ロボットが手動モードで通電していることを確認してください
10003	メカニカルゼロ点のキャリブレーションに失敗	1.手動モードではありません; 2.ロボット動作中; 3.制御モードが位置決めモードになっていません	ロボットが手動モードになっていることを確認し、動作を停止してください
10004	メカニカルゼロ点のキャリブレーションに成功		
10005	センサーゼロ点のキャリブレーションに失敗	1.手動モードではありません; 2.ロボットが通電していません; 3.ロボット動作中; 4.制御モードが位置決めモードになっていません	ロボットをメカニカルゼロ点に移動した後、ロボットが手動モードになっていることを確認し、動作を停止してください。
10006	センサーゼロ点のキャリブレーションに成功		
10007	制御キャビネットの再起動に失敗		ロボットの動作またはプログラムの動作を停止してください
10008	エンコーダーアラームの解除失敗	サーボエンコーダー故障	ロボットを再起動してください。解決できない場合は、弊社までご連絡ください。

10009	エンコーダーアラームの解除成功		
10010	手動モードへの切り替えに失敗	ロボット動作中はモードの切り替えはできません;または既に手動モードになっています	動作を停止して再試行してください。
10011	自動モードへの切り替えに失敗	失敗する原因は次のとおりです。 1.ロボット動作中; 2.ロボットが非常停止状態; 3.ドラッグ機能が有効; 4.既に自動モードになっています	1.動作を停止してください; 2.非常停止状態から復帰してください; 3.ドラッグ機能を無効にしてください
10012	通電に失敗	1.自動モードになっていません; 2.ロボットが非常停止状態	1.自動モードに切り替えてください; 2.非常停止状態から復帰してください
10013	通電オフに失敗	ロボットが動作中か、自動モードになっていません	ロボットの動作を停止してください。
10014	ドラッグ機能を有効にできませんでした。	次の状態ではドラッグ機能を有効にできません: 1.自動モード; 2.通電状態; 3.位置決めモードになっていません; 4.動作中; 5.通常とは異なるタスクの状態; 6.セーフティーモニターがトリガーされました!	手動モードに切り替えて、モーターの電源を切ったまま、位置決めモードにして、やり直してください;または再起動してください。
10015	ドラッグ機能を有効にしました		
10016	ドラッグ機能を無効にできませんでした。	動作モードが正しくありません	1.RCI でドラッグ機能を有効にした場合は、RCI で無効にする必要があります; 2.ロボットを再起動して回復を試みてください。
10017	ドラッグ機能を無効にしました		
10018	バーチャルフェンスの更新に失敗しました	1.設定された領域が小さすぎます; 2.ロボットの現在位置が設定されたエリアの外にあります; 3.ドラッグ機能が有効になっていません	1.バーチャルフェンスの境界を拡大してください; 2.ロボットのフランジ末端を設定された領域内に移動してください; 3.ドラッグ機能を有効にしてください

10019	バーチャルフェンスの更新に成功しました		
10020	新しい教示操作盤の接続を拒否しました	教示操作盤は既に接続されています	既存の接続を切断してください
10021	socket 接続サーバータイムアウト	接続タイムアウト	1.デバイスの接続状況を確認してください; 2.サーバーが正常に機能しているかどうかを確認してください
10022	socketread コマンド受信データフォーマットエラー	ネットワーク障害	
10023	socket 作成の IP およびポートは既に使用されています	RL プログラムまたは外部通信 socket で IP とポート番号が既に使用されている場合は、両方の socket で同じ IP とポートを使用することはできません。	異なる IP アドレスまたはポート番号を設定してください
10024	socket 作成名の重複	socket 作成名の重複	別の socket 名に置き換えてください
10025	socket データの受信に失敗しました	1.ネットワーク接続の異常; 2.サーバーが時間内にデータを送信しなかった; 3.終了記号エラー; 4.データ型エラー	1.ネットワーク接続を確認してください; 2.サーバーがデータを送信しているかどうかを確認してください; 3.終了文字が一致するかどうかをチェックしてください; 4.送信されたデータのタイプを確認してください
10026	socket 切断		
10027	socket 接続		
10028	SocketRead データ長が設定長と一致しません	受信したデータの数が設定と一致しません	正しい個数のデータを送信してください
10029	入力シミュレーションモードの設定に失敗しました		
10030	入力信号のシミュレーションに失敗しました	1.シミュレーションモードがオンになっていません; 2.この信号は存在しません	1.シミュレーションモードをオンにしてください; 2.入力信号の設定を確認してください
10031	GI 信号のシミュレーションに失敗しました	1.シミュレーションモードがオンになっていません; 2.この信号は存在しません	1.シミュレーションモードをオンにしてください; 2.GI 信号の設定を確認してください
10032	出力信号の設定に失敗しました	1.この信号は存在しません; 2.DO 信号がシステム出力に設定されています	1.出力信号の設定を確認してください; 2.エミュレーションがシステム出力信号に結合されていない
10033	GO 信号の設定に失敗しました	1.この信号は存在しません; 2.設定した値が許容範囲を	GO 信号の設定を確認してください

		超えています	
10034	RCI バケット解析エラー	メッセージ長エラーが原因で解析できませんでした	RCI 命令フォーマットを確認してください
10035	RCI パラメーターの保存に成功しました		
10036	RCI のオープンに成功しました		
10037	RCI のクローズに成功しました		
10038	RCI のオープンに失敗しました	1.IP アドレスをロボットアドレス(192.168.0.160)とローカルホスト(127.0.0.1)に設定できません; 2.IP アドレスの形式が正しくありません; 3.ポート番号が既に使用されています; 4.ロボット動作中; 5.制御モードが位置決めモードになっていません	1.使用できる IP アドレスとポート番号を設定してください。 2.ロボットの動作を停止してください
10039		RCI のクローズに失敗しました	1.ロボット動作中; 2.制御モードが位置決めモードになっていません
10040	ドラッグに失敗しました	実際のフィードバックはモデルのトルクから大きく逸脱しており、ドラッグ機能を有効にすることはできません。	1.現在のツールの設定が実際と一致しているかどうかをチェックし、ツールの質量重心が正しいことを確認してください。 2.監視ウィンドウを見て、ロボットの座標系、姿勢が実際と一致しているかどうかをチェックしてください; 3.現在のモデル、RD パラメーターが実際と一致していることを確認してください; 4.メカニカルゼロ位置に戻して、センサーをゼロにしてからドラッグしてください。 5.その他詳細は『ドラッグのトラブルシューティングマニュアル』をご覧ください。
10041	クライアントの RCI が接続されました		
10042	クライアントの RCI が切断されました	クライアントの切断が検出されました	クライアントをチェックしてください
10043	RCI が応答していません。自動モードおよび通電状態を確認してください。		
10044	本体パラメーターの		

	識別が完了しました。ロボットを再起動してください。		
10045	負荷パラメーターの認識に成功しました		
10046	衝突検知パラメーターの更新に失敗しました		
10047	衝突検知を有効にしました		
10048	衝突検知を無効にしました		
10049	負荷の認識に失敗しました。負荷がロボットの定格を超えています	ロボットの定格荷重を超えるツールまたはワークが取り付けられています	ロボットの定格荷重内のツールやワークを使用してください
10050	負荷の認識に失敗しました。負荷の認識結果は正しくありません。	センサートルクフィードバック異常	センサーのトルク値をチェックしてください
10051	ソフトウェアリミットの設定に失敗しました。	1.設定した角度がロボットの機械的なリミットを超えています; 2.手動モードになっていません; 3.ロボット動作中; 5.制御モードが位置決めモードになっていません	1.手動モードに切り替え、ロボットの動作を停止してください; 2.ロボットの機械的なリミットを確認し、その範囲内の角度を設定してください
10052	ソフトウェアリミットの設定に成功しました		
10060	最大軸速度が制限を超えています	1.協働モードでの最大軸速度は 0-15 度/秒の範囲で設定できます; 2.非協働モードでは各軸の最大軸速度を超えてはいけません	範囲内の値をチェックして設定してください
10061	最大 TCP 速度が制限を超えています	1.協働モードの最大 TCP 速度は 0-250 の範囲で設定できます; 2.非協働モードでは、最大 TCP 速度を超えることはできません。	範囲内の値をチェックして設定してください
10062	軸トルクが制限を超えています	各軸の最大軸トルクは定格トルクの 3 倍に制限されます	範囲内の値をチェックして設定してください

		す	
10063	運動量が制限を超えています		
10064	軸の総電力が制限を超えています	軸の総電力がセーフティーモニタの設定値を超えています	1.動作速度を下げてください; 2.負荷とイナーシャを減らしてください; 3.セーフティーモニタ機能をオフにしてください
10065	デュアルエンコーダーの位置偏差が制限を超えています		
10066	トルクセンサのデュアルチャネルデータ偏差が制限を超えています		
10067	総電力が制限を超えています	各軸の最大電力は、最大関節モーメント(Nm)×最大関節速度(ラジアン/秒)に制限されます。 1.協働モードでの最大関節速度は 15 度/秒です。 2.非協働モードでは各軸の最大軸速度を超えてはいけません	各パラメーターとその単位を確認してください
10068	産業用モデルの衝突検出のトリガー動作は現在の「一時停止のトリガー」モードをサポートしていません。「安全停止」に変更してください。		
10070	セーフティーゾーンに入り、動作を停止します		
10071	セーフティーゾーンに入り、協働モードに入ります。		
10072	協働モードを発動し、動作を減速します		
10073	2次協働モードを発動し、動作を減速します		
10074	協働モードを終了し、動作速度を回復		

	します		
10078	RCI 動作中のトリガー制限	1.軸または直行空間位置の超過; 2.軸速度超過; 3.軸またはモーターのトルクの超過; 4.トルク指令と実トルクの偏差が大きい; 5.ロボットが特異点にいます	トルク指令とロボットの初期状態をチェックしてください
10079	トリガー力制御モジュールの保護	内容を見ます	
10080	ドラッグを開始できませんでした:軸位置が限界を超えています	ロボットの現在位置がソフトウェアリミットを超えています	ロボットをソフトウェアリミット内に移動してください
10081	socket データの送信に失敗しました	ネットワーク障害の原因	ネットワークの問題を確認してください
10082	socket 受信データタイムアウト	ネットワーク障害またはコード論理エラー	1.コード論理をチェックしてください; 2.ネットワークの相手側が正常にデータを送受信しているかどうかを確認してください
10083	外部通信でクライアントに接続できませんでした。再接続を試みています	1.IP とポート番号が正しくありません; 2.サーバーデバイスが正常に起動していません; 3.デバイスの接続異常	1.IP とポートの設定が正しいことを確認してください; 2.サーバーを再起動してください
10084	サーバーとしての外部通信の開始に失敗しました	1.IP とポート番号が正しくありません; 2.サーバーデバイスが正常に起動していません; 3.デバイスの接続異常	1.IP を空にするか、0.0.0.0 に設定してください; 2.ポートを 0 に設定することはできません
10085	socket 作成に失敗しました。外部通信名が重複しています	RL プログラム中の socket 名と外部通信の socket 名を同じにすることはできません。	外部通信名と異なる名前で socket を作成してください。
10086	socket 作成に失敗しました。socket の名前が 30 バイトを超えています	socket の名前が長すぎます	socket の名前を短くしてください
10087	外部通信を開けませんでした		原因提示による処理
10088	socket 接続		
10090	セーフティーゾーンの設定に成功しました		

	た		
10091	セーフティーゾーンの設定に失敗しました	1.手動モードになっていません; 2.ロボット動作中; 3.セーフティーゾーンの動作を”協働モードのトリガー”に設定したが、協働モードがオンになっていない	1.手動モードに切り替えてください; 2.ロボットの動作を停止してください; 3.協働モードのトリガーの場合、まず協働モードをオンにしてください。
10092	協働モードの設定に成功しました		
10093	協働モードの設定に失敗しました	1.手動モードになっていません; 2.ロボットが通電していません; 3.ロボット動作中; 4.設定したパラメーターが制限を超えています	1.ロボットが手動モードになっていることを確認し、動作を停止して通電をオフにしてください; 2.協働モードの各パラメーター設定を確認してください
10094	セーフティーモニタの設定に成功しました		
10095	セーフティーモニタの設定に失敗しました	1.手動モードになっていません; 2.ロボットが通電していません; 3.ロボット動作中	ロボットが手動モードになっていることを確認し、動作を停止して通電をオフにしてください。
10096	衝突検出トリガーの設定に失敗しました	このトリガー動作はサポートされていません	トリガー動作を変更してください
10097	DH パラメーターが正常に保存されました。ロボットを再起動してください。このバージョンでは、リンクパラメーターの一部の修正のみサポートされています。		
10098	このバージョンでは、リンクパラメーターの一部の修正のみサポートされています。		
10099	RD パラメーターの保存に失敗しました。 RD パラメーターの変	RD パラメーター変更の制限を超えています	RD パラメーター変更は制限範囲+/- 50 で設定してください

	更制限を超えています		
10100	socket 作成の IP およびポートは既に使用されています	RL プログラムまたは外部通信 socket で IP とポート番号が既に使用されている場合は、両方の socket で同じ IP とポートを使用することはできません。	異なる IP アドレスまたはポート番号を設定してください
10101	socket ip:port エラー	RL プログラムまたは外部通信 socket で重複または不正な IP が使用されています	Socket をローカルの空き ip:port に変更するか、ネットワーク設定を照会してください。
10102	socket ポートエラー	RL プログラムまたは外部通信 socket で間違ったポートが使用されています	socket ポートを 0~65535 の空きポートに変更してください。
10103	SocketSendString に失敗しました	接続が確立されていないか、ネットワークエラーです	ネットワーク接続を確認するか、接続されている socket を使用してください。
10104	SocketSendByte に失敗しました	接続が確立されていないか、ネットワークエラーです	ネットワーク接続を確認するか、接続されている socket を使用してください。
10105	Socket のデータ解析に失敗しました	送信データがルールに違反しています	ルールに従ったデータを送信してください
10106	クイック移動に失敗しました。手動モードではありません。	ロボットの状態が正しくありません。手動モードではありません。	手動モードに切り替えてください
10107	クイック移動に失敗しました。プログラムが実行されています	ロボットの状態が正しくありません。プログラムが実行されています。	マシンプログラムを一時停止してください
10108	クイック移動に失敗しました。ロボットが通電していません	ロボットの状態が正しくありません。ロボットが通電していません。	ロボットに通電してください
10109	開始点が設定されていません。ホーム指令エラーです。	開始点が設定されていません。ホーム指令エラーです。	「インターフェースの設定」 > 「基本設定」 > 「インターフェースのクイック調整」で開始点を設定してください
10110	キャリブレーション角度の設定に成功しました		
10111	キャリブレーション角度の設定に失敗しました	キャリブレーション角度がリミットデータを超過しています	リミットデータ内の角度に設定してください
10112	ターミナルクイック調整に失敗しました。適切な調整タイプ	端末調整は端末姿勢の微調整に使用され、大きな端末調整は失敗する可能性があります	1.現在のロボットの姿勢とツール部品のパラメーターをチェックし、適切な端末調整タイプを選択してください;

	プを選択してください	ります。	2.ロボットは既に対応する姿勢にあります;3.指定された姿勢が可動範囲外にあります;4.姿勢を指定すると、ロボットが特異点になります。
10114	シリアルポート名の競合が発生しました。再入力してください	シリアルポート名を新規作成または変更する場合、システム io , socket および既存のシリアルポート名と重複してはいけません。	システム IO 、 socket 名、および既存のシリアルポート名を確認し、新しいシリアルポート名を再入力してください。
10115	シリアルポートの作成に成功しました		
10116	シリアルポートの作成に失敗しました	1.シリアルポートのハードウェアの接続が切断されています; 2.シリアルソケットが存在しません	1.シリアルポートのハードウェア接続を確認してください; 2.変数リストからシリアルポート変数を削除して再作成してください
10117	シリアルポートのシャットダウンに失敗しました	1.シリアルポートのハードウェアの接続が切断されています; 2.シリアルソケットが存在しません	1.シリアルポートのハードウェア接続を確認してください; 2. pptomain をクリックして再実行してください
10118	シリアルポートのシャットダウンに成功しました		
10119	シリアルポートは存在しません	このシリアルポートは存在しません	通信-シリアル設定インターフェースでシリアルポートを作成してください
10121	シリアルポートから string を送信できません	1.シリアルポートのハードウェアの接続が切断されています; 2.シリアルソケットが存在しません	1.シリアルポートのハードウェア接続を確認してください; 2.変数リストからシリアルポート変数を削除して再作成し、シリアル通信が正常に行われていることを確認してください
10122	シリアルポートからの byte の読み込みに失敗しました	読み込まれたデータを byte タイプに変換できませんでした	送信内容を確認してください
10123	シリアルポートからの byte の送信に失敗しました	1.シリアルポートのハードウェアの接続が切断されています; 2.シリアルソケットが存在しません	1.シリアルポートのハードウェア接続を確認してください; 2.変数リストからシリアルポート変数を削除して再作成し、シリアル通信が正常に行われていることを確認してください
10124	シリアルポートのバッファのクリアに失敗しました	1.シリアルポートのハードウェアの接続が切断されています; 2.シリアルソケットが存在しません	1.シリアルポートのハードウェア接続を確認してください; 2. pptomain をクリックして再実行してください

10125	シリアルポートのバッファ長の取得に失敗しました	1.シリアルポートのハードウェアの接続が切断されています; 2.シリアルソケットが存在しません	1.シリアルポートのハードウェア接続を確認してください; 2.pptomainをクリックして再実行してください
10126	serial 受信データタイムアウト	ネットワーク障害またはコード論理エラー	1.コード論理をチェックしてください; 2.ネットワークの相手側が正常にデータを送受信しているかどうかを確認してください
10127	serial 受信データ障害	1.シリアルポートの接続の異常; 2.相手側が時間内にデータを送信しませんでした; 3.シリアルポートの両端のパラメーター設定が一致していません	1.シリアルポートのハードウェア接続を確認してください; 2.相手側がデータを送信しているかどうかを確認してください; 3.シリアルポートの両端のパラメーター設定を確認してください
10128	現在 xDiagnose のバージョンが低いいため、データを表示できません	現在 xDiagnose のバージョンが低いです	xDiagnose バージョン 0.3.8 以降が使用可能です
10129	コンベアのワーク待ち時間が規定時間を超えました。	1.ベルトコンベアの動作は停止しているが、追跡機能はオンになっています; 2.光電スイッチが異常で、ワークトリガー信号をキャッチできません	1.ベルトコンベアが正常に動くかどうかをチェックしてください; 2.光電スイッチが正常にトリガー信号をキャッチしたかどうかをチェックしてください
10130	ワークが開始ウィンドウからはみ出しています		
10131	ワークがワークスペースからはみ出しています	ワークがワークスペースからはみ出しており、トレースできません。	ワークの位置を調整してください
10132	搬送速度がしきい値を下回っています	ベルトコンベアが停止しているか、エンコーダーケーブルが切れています。	ベルトコンベアが正常に動いているか、エンコーダーケーブルが切れていないかを確認してください
10133	ホットプラグモードへの切り替えに失敗しました	モデル、制御キャビネット、またはセーフティーボードファームウェアではホットプラグ機能がサポートされていません	1.現在のロボットのモデルが、XBC5 制御キャビネットを備えた産業用モデル、または協働 CR モデルであることを確認してください。 2.現在のロボットのセーフティーボードが mini ボードで、ファームウェアがバージョン 2.0 にアップグレードされていることを確認してください。 3.ENI ファイルが正しく設定されていることを確認してください

10134	ホットプラグモードへの切り替えに失敗しました	1.セーフティーボードの状態が正しく切り替わりません; 2.通信エラー	弊社までご連絡ください。
10135	ホットプラグモードへの切り替えに成功しました		
10136	クイック移動に失敗しました。軸の角度がリミットを超えています	軸角度が機械的リミットを超えています	機械リミット内の角度値に修正してください
10137	コンベアワークが2回関連付けされています	DropWObj がない場合は、再度 WaitWobj	コマンドを確認してください。同じワークは1回しか関連付けられません。
10138	Waitwobj コマンドの実行に失敗しました	手動モードでの継続的な実行が原因です	手動モードでのステップ実行、または自動モードでの連続実行を行ってください
10139	コンベアに関連するワークがありません	1.追跡されているワークがありません;2.指定された名前のベルトコンベアがありません	ワークをベルトコンベアに置いて光電を起動し、ステータスマニターでワークの位置が正常かどうかを確認してください。
10140	異常な摩擦同定結果です。摩擦なしモデルを使用してください		摩擦なしモデルを使用し、摩擦係数を 0.1 に設定してください。
10141	入力した質量が範囲外です		入力する質量は 0~最大負荷の範囲で指定してください。
10142	無負荷認識が実行されておらず、負荷認識が停止しています。		最初に無負荷認識をしてから、負荷認識をしてください
10143	点位置への移動またはフォースコントロール認識前の速度設定に失敗しました		
10144	このモデルは動力学の識別をした後、摩擦だけを修正し、直接摩擦力の識別をすることをお勧めします。		
10145	産業用ロボットは協働モードに対応しておらず、トリガー動作は実際には動作し	現在の産業用ロボットは協働モードに対応していません	

	ません。		
10146	現在のモデルでは動力学同定はサポートされていません	現在のモデルでは動力学同定はサポートされていません	モデルの同定が必要な場合は、摩擦同定を直接行ってください。
10147	機能の承認に失敗しました	内容を見ます	認証管理者に問い合わせてください。
10200	電子銘板がありません。	1.本体に電子銘板がありません; 2.電子銘板ハードウェアが破損しています; 3.電子銘板シリアルポートの配置エラーです	1.電子銘板シリアルポートのハードウェア接続をチェックしてください; 2.電子銘板ハードウェアをチェックしてください; 3.電子銘板シリアルポートの配置をチェックしてください
10201	電子銘板読み取りデータエラー	1.電子銘板プロトコルのアドレスが間違っています。 2.電子銘板ハードウェアが破損しています	1.プロトコルが一致しているかどうかを確認してください; 2.電子銘板ハードウェアをチェックしてください
10202	電子銘板書き込みデータエラー	1.電子銘板プロトコルのアドレスが間違っています。 2.電子銘板ハードウェアが破損しています	1.プロトコルが一致しているかどうかを確認してください; 2.電子銘板ハードウェアをチェックしてください
10203	電子銘板データの不一致	1.ロボット本体を交換; 2.制御キャビネットの交換	確認後にデータ上書きを行ってください
10204	エンコーダーのバッテリー電圧が低下しています	1.運転時間が長すぎます; 2.エンコーダーバッテリーが破損しています	エンコーダーバッテリーを交換してください
10205	ロボットの動作時間がメンテナンス期間を超えています	定期メンテナンスされていません	定期メンテナンスを行ってください
10206	データの上書きに成功しました		
10207	電子銘板データの上書きができません	1.電子銘板が存在しません; 2.モデル、ID、またはハードウェアのバージョンが一致しません	ログの内容に合わせて電子銘板をチェックしてください
10208	RC データの上書きができません	1.制御キャビネットの機種と本体が合っていない; 2.電子銘板にデータが焼き付けられていない、またはデータが間違っています	1.正しいモデルを選択後、コントローラーを再起動してください。 2.電子銘板に正しいデータを焼き直してください
10209	socket 通信に失敗しました	1.IP と port は外部通信で使用されています 2.socket の名前が外部通信と同じで、デフォルトの SYS_SOCKET の名前が外部通信で使用されています	1.socket の port または IP を変更してください 2.socket の名前を変更してください

13013	非常停止トリガー		手動で非常停止から回復してください
13014	停止エリア信号トリガー		停止エリア信号の回復を待ってください
13015	ボディパラメーター 特定計算開始		
13016	プログラム字句または文法チェックエラー	RL プログラム構文エラー	RL プログラムをチェックしてください
13017	プログラム PP_to_main が失敗しました	1.タスクが破棄されました; 2.プロジェクトファイルがロードされていません; 3.main 関数がありません	1.タスクを再初期化してください; 2.プロジェクトをリロードしてください; 3.ファイルに main 関数があるかどうかをチェックしてください
13018	プログラム PP_to_func が失敗しました	1.タスクが破棄されました; 2.タスクがシンボルテーブルを作成していません; 3.ジャンプする関数が存在しません	1.タスクを再初期化してください; 2.関数に構文エラーがないかを確認し、シンボルテーブルを再作成してください; 3.関数が存在するかどうかを確認してください
13019	プログラム PP_to_line が失敗しました	プログラム PP_to_line のジャンプは同じ PROC/FUNC 内でのみ許可されます	同じ PROC/FUNC 内でジャンプしているかどうかを確認してください
13020	すべての RL タスクが停止しました。 PPtoMain をクリックするか、エラーメッセージを確認してください。	1.シングルループモードタスクが終了しています; 2.タスクがエラーで停止しています	1.プログラムに論理エラーがないかどうかを確認してください; 2.pptomain をクリックして再実行してください
13021	ベース座標系エラー、設定に失敗しました	ベース座標系を設定するコマンドを制御キャビネットが解決できません。制御キャビネットのバージョンが HMI と互換性がない可能性があります。	制御キャビネットのバージョンが HMI に対応していることを確認してください。
13022	次のステップで、ロードされていないタスクを選択します。		プロジェクトのタスクリストで対応するタスクを選択し、pptomain を実行してください。
13030	RSC がロボットの電力超過を検出しました	ロボットの電力が制限を超えています、または RSC のロボットの電力制限が低すぎます、または RSC が故障しています	RSC のロボットの電力制限が妥当かどうかを確認するか、RSC のロボットの電力制限をオフにします。
13031	RSC がロボットの運動量超過を検出しました	ロボットの運動量が制限を超えています、または RSC のロボットの運動量	RSC のロボット運動量制限が妥当かどうかを確認するか、RSC のロボットの運動量制限をオフにします。

		制限が低すぎます、または RSC が故障しています	
13032	RSC がロボットの TCP 力超過を検出しました	ロボットの TCP 力が制限を超えています、または RSC のロボットの TCP 力制限が低すぎます、または RSC が故障しています	RSC のロボットの TCP 力の制限が妥当かどうかを確認するか、RSC のロボットの TCP 力の制限をオフにします。
13033	RSC がロボットの TCP 速度超過を検出しました	ロボットの TCP 速度が制限を超えています、または RSC のロボットの TCP 速度制限が低すぎます、または RSC が故障しています	RSC のロボットの TCP 速度の制限が妥当かどうかを確認するか、RSC のロボットの TCP 速度の制限をオフにします。
13034	RSC がロボット位置のオーバーランを検出しました	ロボット位置がオーバーランしています、または RSC が故障しています	RSC のロボット位置の制限が妥当かどうかを確認するか、RSC のロボット位置の制限をオフにします。
13035	RSC がロボット姿勢のオーバーランを検出しました	ロボット姿勢がオーバーランしています、または RSC が故障しています	RSC のロボット姿勢の制限が妥当かどうかを確認するか、RSC のロボット姿勢の制限をオフにします。
13036	RSC がロボット軸の衝突力超過を検出しました	ロボット軸の衝突力が制限を超えています、または RSC が故障しています	RSC のロボット軸の衝突力の制限が妥当かどうかを確認するか、RSC のロボット軸の衝突力の制限をオフにします。
13037	RSC がロボット軸の電力超過を検出しました	ロボット軸の電力が制限を超えています、または RSC が故障しています	RSC のロボット軸の電力の制限が妥当かどうかを確認するか、RSC のロボット軸の電力の制限をオフにします。
13038	RSC がロボット軸位置のオーバーランを検出しました	ロボット軸位置がオーバーランしています、または RSC が故障しています	RSC のロボット軸位置の制限が妥当かどうかを確認するか、RSC のロボット軸位置の制限をオフにします。
13039	RSC がロボット軸の速度超過を検出しました	ロボット軸の速度が制限を超えています、または RSC が故障しています	RSC のロボット軸の速度の制限が妥当かどうかを確認するか、RSC のロボット軸の速度の制限をオフにします。
13040	RSC がロボット軸のトルク超過を検出しました	ロボット軸のトルクが制限を超えています、または RSC が故障しています	RSC のロボット軸のトルクの制限が妥当かどうかを確認するか、RSC のロボット軸のトルクの制限をオフにします。
13041	RSC がロボット軸の通信異常を検出しました	ロボット軸の通信が異常です、または RSC が故障しています	ロボットを再起動するか、ハードウェア障害を確認するために弊社までご連絡ください。
13042	RSC がロボット軸の動作異常を検出しました	ロボット軸の動作が異常です、または RSC が故障しています	ロボットを再起動するか、ハードウェア障害を確認するために弊社までご連絡ください。
13043	RSC とマスターの間で通信エラーが発生しました	RSC とマスターの間で FSoe 通信エラーが発生しました	ロボットを再起動するか、ハードウェア障害を確認するために弊社までご連絡ください。
13044	コントローラーと RSC の間で SDO データの同期に失敗しました	コントローラーと RSC の間で SDO データの同期に失敗しました	再設定を試みるか、ハードウェア障害を確認するために弊社までご連絡ください。

	した		
13045	コントローラーとRSCの間でSDOデータの同期に成功しました		
13046	RSC のドラッグ時にセーフティー制限を超えました	ドラッグ中の速度が 250 mm/s を超え、安全停止がトリガーされました	
13047	RSC 初期化パラメーターチェックの不整合	RSC の基本安全パラメーターがメイン制御キャビネットのパラメーターと一致しません	ロボットまたは制御キャビネットをハード再起動してください。
13048	RSC とマスターコントロール間の通信がセーフデータ状態(0x08)になりました	RSC とマスターコントロール間の通信がセーフデータ状態(0x08)になりました	
13049	RSC 軸制限位置パラメーターが不正です。	設定された角度がロボットの機械的リミットを超えています	ロボットの機械的なりミットを確認して、その範囲内の角度を設定してください
13050	RSC セーフホームポジションパラメーターが不正です。	設定された角度がロボットの機械的リミットを超えています	ロボットの機械的なりミットを確認して、その範囲内の角度を設定してください
14001	ネットワーク接続が切断されました。		
14002	ネットワーク接続がダウンしています。		
14003	ネットワーク接続が確立されました。		
14004	ネットワーク接続の待機中です		
14005	ネットワーク接続の再接続中です		
14006	ネットワーク接続の再接続中です		
14010	割り当て済みのレジスタで書き込みが許可されていません	レジスタはシステム割り当て済みのレジスタであり、RL での書き込みは許可されていません。	別のレジスタを使用するか、そのレジスタの割り当てを解除してください。
14011	バスデバイスのオープンに失敗し、対応する Ethercat スレーブがスキャンされませんでした。	Ethercat スレーブが設定されていないか、デバイスリンクが異常です	Ethercat スレーブをインポートして設定してください。
14020	バスデバイスのオー		

	ブンを成功しました		
14021	バスデバイスのオープンに失敗しました	内容を見ます	バスデバイスの構成を確認し、対応するデバイスが正しく接続されていることを確認してください。
14022	バスデバイスのシャットダウンに成功しました		
14023	シリアルポートはバスデバイスによって使用されています。	このシリアルポートは、既に modbus RTU バスデバイスで使用されています。	バスデバイスをシャットダウンするか、別のシリアルポートを使用してください
14030	profinet バスモデル構成エラー	このスロットデータモデルの選択が PLC と一致しません	このスロットデータモデルを再設定してください
14031	レジスタホールドファイルエラー	レジスタホールドファイルの crc チェックに失敗しました。ファイルが破損しています	レジスタホールドファイルを削除してください
14501	システム DO を設定できません	DO 信号はシステム IO 信号で、設定できません	別の出力信号を使用するか、システム IO でその信号の割り当てを解除してください。
14502	GO 信号の設定に失敗しました	設定値が信号の有効範囲外です	数値をチェックしてください
14503	AO 信号の設定に失敗しました	設定値が信号の有効範囲外です	数値をチェックしてください
14504	PulseDO 設定に失敗しました。時間が有効範囲外です。	パルス時間設定エラーです、範囲[0.001、2000] S	パルス時間を再入力してください
14505	PulseReg 設定に失敗しました。時間が有効範囲外です。	パルス時間設定エラーです、範囲[0.001、10] S	パルス時間を再入力してください
14510	出力信号のロードに失敗しました	マップされた物理ポートが他の信号またはシステム出力信号と競合しています	1.出力ポートが使用されていないかどうかを確認してください; 2.システム IO を再設定してください
14511	システム入力信号の設定に失敗しました。矛盾があります	システム入力設定規則に違反しています。考えられる原因: 1.システム IO が重複して設定されています。 2.対応 IO が RL 工程の IO に占有されています; 3. IO 信号がありません	1.独立したシステム IO を構成してください; 2. RL 工程が使用するシステム IO を変更してください; 3. IO デバイスをオンまたは作成してください
14512	システム入力信号の設定に失敗しました。矛盾があります	システム入力設定規則に違反しています。考えられる原因: 1.システム IO が重複して設	1.独立したシステム IO を構成してください; 2. RL 工程が使用するシステム IO を変更してください;

		定されています。 2.対応 IO が RL 工程の IO に占有されています; 3.IO 信号がありません	3.IO デバイスをオンまたは作成してください
14521	Ethercat IO デバイスの初期化に失敗し、対応する Ethercat スレーブがスキャンされませんでした	Ethercat スレーブが設定されていないか、デバイスリンクが異常です	Ethercat スレーブをインポートして設定してください。
14530	セキュリティボード拡張 IO デバイスのステータスが変化しました	拡張 IO デバイスのハードウェアの破損またはリンクの異常です	セキュリティボード拡張 IO デバイスのハードウェアステータスとリンクステータスを確認してください。
14531	セキュリティボード拡張 IO デバイスのステータスが変化しました	拡張 IO デバイスが新たに接続されました	
14532	IO デバイスのステータスが変化しました	IO デバイス構成の変更	
14533	IO デバイスのステータスが変化しました	IO デバイス構成の変更	
14534	IO デバイスのステータスが変化しました	IO デバイスの通信エラー	
15000	ドラッグパスの再生に失敗しました	ドラッグパス再生を実行しましたが、バッファにドラッグデータがありません。	ドラッグ軌跡を再記録してください
15001	ドラッグデータの保存に失敗しました	バッファにドラッグデータがありません	ドラッグ軌跡を再記録してください
15002	ドラッグパスの再生に失敗しました	ドラッグパスの記録が誤って削除されました	ドラッグ軌跡の再記録またはパスの再読み込みを行ってください
15003	ドラッグパスの再生に失敗しました。ドラッグファイルのデータエラーです	再生パスが別のロボットのものです。	正しいドラッグ再生パスをインポートするか、軌跡を再記録してください
15004	ドラッグデータの保存に失敗しました	シーケンシャルデータストアのハードディスクファイルをドラッグアンドドロップできませんでした	ドラッグ軌跡を再記録するか、ロボットを再起動してください
15005	ドラッグデータの保存に成功しました		
15006	ドラッグパスの再生を実行します		
15007	ドラッグ再生データの記録を開始しまし		

	た		
15008	ドラッグ再生データの記録を停止しました		
15009	ドラッグパスの再生を停止しました		
15010	パスデータの記録に失敗しました:パスポイントが少なすぎます	ドラッグパスのポイントが少なすぎます	ドラッグ軌跡を再記録してください
15011	パスデータの記録に失敗しました:軸角度がオーバーランしています	ドラッグ軌跡にソフトウェアリミットを超える軸があります	ドラッグ軌跡を再記録してください
15012	パスデータの記録に失敗しました:軸速度が制限を超えています	ドラッグ軌跡で軸速度が制限を超えています	ドラッグ軌跡を再記録してください
15013	パスデータの記録に失敗しました:記録停止時の速度がゼロになりません	パスの記録停止時の速度がゼロになりません	ロボットが停止するまで待ってから記録を終了してください
15014	パスデータの記録に失敗しました:記録開始時の速度がゼロになりません	パスの記録開始時の速度がゼロになりません	ロボットが停止した状態から記録を開始してください
15015	パスデータの記録に失敗しました:モーターが通電していません	モーターが通電していません	ロボットを通電してください
15016	パスデータの記録に成功しました		
15017	ドラッグパス再生の実行、パスの読み込みに失敗しました	ドラッグパスのデータ解析に失敗しました。パスデータが改ざんされている可能性があります	パスの再読み込みまたは軌跡の再記録を行ってください
15018	パス再生、設定レートオーバー	再生レートが大きすぎます	パス再生レートを下げてください
15019	"パス再生、ソフトウェアリミットを超えるポイントに再生パスポイント "	再生パスポイントがソフトウェアリミットを超えています	現在のソフトウェアリミット値を調整してください
17001	レジスタの読み取りに失敗しました	1.このレジスタは存在しません; 2.レジスタが変数タイプと	レジスタ設定をチェックしてください

		一致しません; 3.配列添字が範囲外です	
17002	レジスタの書き込みに失敗しました	1.このレジスタは存在しません; 2.レジスタが変数タイプと一致しません; 3.配列添字が範囲外です	レジスタ設定をチェックしてください
17003	modbus 入力レジスタの読み込みに失敗しました		
17004	レジスタ設定のロードに失敗しました	レジスタ設定ファイル (registers.json) の解析に失敗しました。理由は内容を参照してください。	レジスタを再設定するか、設定を消去してください。
17005	modbus 通信に失敗しました。modbus リンクが確立されていません		
17006	modbus 設定の保存に成功しました		
17007	外部通信設定の保存に成功しました		
17008	modbus 設定の解決に失敗しました。	modbus レジスタの設定に誤りがあります	modbus レジスタ設定を確認してください
17009	バスデバイス設定ファイルのロードに失敗しました	フィールドバスデバイス設定ファイル (fieldbus_device.json) の解析に失敗しました。理由はコンテンツを参照してください	フィールドバスデバイスを再設定するか、設定を消去してください。
17100	cclink の起動に失敗しました。cclink ゲートウェイモジュールが取得されませんでした	cclink ゲートウェイモジュールが設定されていないか、デバイスリンクが異常です	cclink ゲートウェイモジュールのインポートと設定を行ってください
17101	cclink が起動しました		
17102	cclink の起動に失敗しました		
17103	cclink が終了しました		
17104	CC-LINK IEF Basic マスターとの通信に成功しました		
17105	CC-LINK IEF Basic マスターとの通信が切		

	断されました		
17200	規則に一致するサーボパラメーターファイルのセットが制御キャビネットに存在しません。	サーボパラメーター切り替えの機能を使用するには、制御キャビネットに少なくとも 1 組のサーボパラメーターが必要です。	
17201	通電に失敗しました。サーボパラメーターの切り替え中に通電できません。	通電に失敗しました。サーボパラメーターの切り替え中に通電できません。	
17202	通電状態では、サーボパラメーターの切り替えはできません	通電状態では、サーボパラメーターの切り替えはできません	
17300	RL プログラムのカスタマイズが停止し、実行できません		pptomain、pptoline、または pptofunc のいずれかでプログラムを実行できます。
17301	xService リンクが壊れています。	1、HMI と制御キャビネットの接続が突然切断されました; 2.SDK クライアントがアクティブに切断されました	1.HMI の操作で制御キャビネットを切断し再接続してください;
17203	サーボパラメーターの切り替えに成功しました		
17204	サーボパラメーターの切り替えミスマッチ		

17.1.2 3XXXX

エラーコード	エラー内容	原因	修復方法
31001	EtherCAT マスター構成パラメーターのエラー	EtherCAT 構成ファイル不一致	構成ファイルを再インポートし、制御システムを再起動してください。
31002	EtherCAT 認証ファイルの読み取り失敗	認可ファイルが存在しないか、読み取り/書き込み権限エラー	承認ファイルを再インポートしてください。
31003	EtherCAT 認証失敗	EtherCAT 未認証	EtherCAT 認証コードを確認してください。EtherCAT 認証インターフェイスで再認証を試み、認証が成功した後にロボットを再起動してください。
31004	EtherCAT マスターの構成に失敗	EtherCAT 構成ファイルが一致しないため、マスター設定できない。	構成ファイルを再インポートし、制御システムを再起動してください。
31005	DC 構成に失敗	EtherCAT 設定ファイルが一致しないため、構成に	構成ファイルを再インポートし、制御システムを再起動してください。

		失敗しました。	
31006	DCM の設定に失敗	EtherCAT 設定ファイルが一致しないため、構成に失敗しました。	構成ファイルを再インポートし、制御システムを再起動してください。
31007	バスのスキャンに失敗	EtherCAT 設定ファイルが一致しないため、構成に失敗しました。	構成ファイルを再インポートし、制御システムを再起動してください。
31008	設定されたスレーブの数とスキャンされたスレーブの数が不一致	1.EtherCAT ネットワーク構成が実際のネットワークトポロジと一致しないため、構成に失敗しました。 2.EtherCAT ネットワークスレーブのハードウェア障害。	1、設定ファイルを再インポートし、制御システムを再起動してください。 2、弊社までご連絡ください。
31009	EtherCAT バスの起動に失敗	EtherCAT バスの起動中にエラーが発生しました。一部のスレーブが OP モードに正しく切り替えられず、バス接続が失敗しました。	制御システムを再起動してください。弊社までご連絡してください。
31010	内部軸サーボの初期化に失敗	ドライブ内部で障害が発生しました。	サーボドライバ間の接続が中断しました。
31011	IO スレーブの初期化に失敗	1.ワークステーションと汎用 IO モジュール間のハードウェア接続の中断 2.汎用 IO モジュールとセキュア IO モジュール間のハードウェア接続の中断"	障害の原因に応じて IO 信号を再構成した後、制御システムを再起動してください。
31012	末端スレーブの初期化に失敗	ワークステーションと汎用末端スレーブモジュール間のハードウェア接続が切断されたか、末端スレーブハードウェアに障害が発生しました。	ハードウェアの接続を確認してください。また、末端のハードウェアを交換してください。
31013	安全ボードのスレーブの初期化に失敗	コントローラーと安全ボードスレーブモジュール間のハードウェア接続の中断。	ハードウェア接続の確認してください。
31014	スレーブとの通信異常	1、コントローラーとスレーブモジュール間の通信が中断 2、スレーブ設備に故障が発生しました。"	1、Ethercat デバッグ線がロボット側から抜けているかどうかを確認してください。 2、ハードウェアの接続を確認し、制御システムを再起動してください。 3、弊社 E までご連絡ください。"

31015	サーボアラーム	サーボ駆動器に警報が出て、警報の具体的な原因はサーボが報告したエラーコードによって対応する駆動器マニュアルを探する必要があります。	1、エラーコードから対応するドライブマニュアルを探し、マニュアルの指示に従って処理してください 2、弊社までご連絡ください。
31016	サーボアラームがクリアされました	なし	なし
31017	Ethercat 承認されていないため、試用期間終了	Ethercat は許可されておらず、1 時間のみ使用可能。	弊社に連絡し、Ethercat 認証コードを要求してください。EtherCAT 認証インターフェイスが認証され、認証が成功した後、再起動ロボットが有効になります。
31018	スレーブデバイスとコントローラーの接続が正しくありません	1、スレーブ装置とコントローラーの接続ラインの障害。 2、スレーブの障害。	1、スレーブ間のケーブルに障害が発生していないか確認してください。 2、弊社までご連絡ください。
31019	CCLink ゲートウェイスレーブの初期化に失敗	ハードウェア接続の異常またはハードウェア障害	ハードウェア接続を確認してください。
31020	EthercatIO スレーブモジュールが適合していません。	未適合の EthercatIO スレーブモジュールを使用しているため、現在の IO スレーブに関するコントローラーの情報が不明です。この IO モジュールが正しく構成されて使用されていることをユーザーが確認する必要があります。	アダプティブ化されていない IO モジュールの PDO 設定は不明なため、直接使用するとリスクがあります。慎重に使用してください。一般に、64 ウェイまでのデジタル IO モジュールに対しては、そのまま利用できます。アナログ IO モジュールの場合、コントローラーのアナログ量タイプ、レンジ、精度パラメーターが未知であるため、各機器メーカーの PDO 設定も異なる可能性があります。直接使用することはお勧めできません。弊社まで連絡し、適応して、使用してください。
31021	セキュリティバージョンのスレーブの初期化に失敗しました。	eni ファイルで設定されたセキュリティボードのファームウェアバージョンが、実際のファームウェアバージョンと一致しません。	eni ファイルのセキュリティボードのバージョンが、セキュリティボードのファームウェアのバージョンと一致しているかどうかを確認してください。
31022	末端スレーブ入力電流信号過負荷	末端スレーブチャネル AI が電流モードに設定されている場合は、実際の入力が電流であり、制限パラメーターを超えないようにしてください。制限パラメーターを超えると、ハードウェアが破損する可能性があります。	対応するチャネルの入力信号をオフにし、xpanel 設定インターフェイスにアクセスしてモードを電圧モードに設定してください。

31023	末端スレーブ入力電流 信号過負荷	xpanel の設定が完了し、 過負荷が回復しました	電流モードを継続する場合は、入力電流が 過負荷にならないことを確認してくださ い。
31024	xPanel 末端スレーブの ハードウェアテストに 失敗しました	xPanel 末端ステーション のハードウェアテストプ ロセスエラー、ハードウ ェア接続エラー、または ハードウェア障害	xPanel テストプロセスが正しいことを確認 してください。解決できない場合は弊社ま でご連絡ください。
31025	安全ボードタイプの構 成エラー	コントローラー構成の安 全ボードタイプが、実際 に使用されている安全ボ ードタイプと一致しませ ん。	コントローラーに設定されている安全ボー ドのタイプと実際に使用されている安全ボ ードのタイプが一致していることを確認し てください。
31026	内部軸サーボの初期化 に失敗しました	マスター構成の ENI ファ イル内のサーボドライバ バージョンが、スキャン されたサーボドライブコ ンフィギュレーションと 一致しません。	マスター ENI ファイルを交換し、正しい ENI ファイルが使用されていることを確認 してください。
32001	モーター電源投入失敗	サーボにエラーがあり、 モーターの電源投入に失 敗しました	サーボアラームをクリアしてからモーター の電源を入れてください。
32002	モーター電源投入に失 敗し、STO はオンにな っていません。	なし	なし
32003	コントローラーマスタ ステーションとデバイ スの通信スレッドのブ ロック	ソフトウェアが原因で、 コントローラーとスレー ブとの通信に異常が発生 しました	1、コントローラーのソフトウェアを再起 動してください。 2、弊社までご連絡ください。
32004	EtherCAT スレッドがブ ロックされ、スケジュー ルタイムアウト	システム内部エラー	制御システムを再起動してください。
32005	EtherCAT スレッドがブ ロックされ、連続タイ ムアウトが5000回を超 えました	システム内部エラー	制御システムを再起動してください。
32006	位置指令拒否	サーボに送られる位置指 令に大きなジャンプが発 生し、モーターの最大速 度を超える速度が発生 し、危険な可能性があります。念のため、モータ ー電源を切る操作を行 います。	再起動してください。

32007	Ethercat スレッドのイベント実行タイムアウト	内部警報	内部状態を記録して、無視できます。
32008	サーボ零点書込みエラー	CRジョイントのみサポート、CRロボットかどうか確認、ファームウェアが正しいか確認	なし
32009	サーボ零点の書き込みに成功しました		
32010	エンコーダーのバッテリー電圧低下警告	エンコーダーのバッテリー電圧が不足しています。	エンコーダー電池はすぐに交換してください
32011	エンコーダーバッテリー電圧が低い警告除去		
35001	サーボドライバのショート	出力線-出力線、出力線-接地線間の内部 PWM ブリッジエラー	回線のトラブルシューティングを行い、短絡を解消してください
35002	サーボドライバの温度オーバー	サーボドライバ内部温度が設定値に達しました。	サーボドライバの温度を設定値以下に下げてください。
35003	サーボドライバ過電圧	母線電圧が指定された電圧制限を超えています。	母線電圧を指定電圧範囲内にしてください。
35004	サーボドライバ電圧不足	母線電圧は指定された電圧以下です。	母線電圧を指定した電圧範囲内にしてください。
35005	エンコーダーモーターの温度オーバー	モーターの過熱スイッチの状態が過熱エラーを示しています。	温度スイッチを通常の状態に戻してください
35006	エンコーダーフィードバックエラー	1、サーボドライバ内部の 5V 出力の過電流。 2、回転変圧器またはアナログエンコーダーの未接続。 3、圧.レベルが誤差範囲外。 4、インクリメンタルエンコーダー差動信号が未接続。	1、エンコーダー電源を指定した電圧範囲に戻してください。 2、フィードバック信号を指定されたレベル範囲内に戻してください。 3、差動信号の良好状態に接続してください。
35007	サーボドライバ位相エラー	エンコーダーによる位相角が HALL のスイッチング状態と一致していません。このエラーは、ブラシレスモーターが正弦波に設定されている場合にのみ発生します。このエラーは、レゾルバフィー	エンコーダーによる位相角を HALL のスイッチング状態と一致させてください。

		ドバックまたは HALL 補正機能がオフになっている場合は発生しません。	
35008	サーボドライバの電流限界	モーター過負荷または回路異常	1、モーターが過負荷になっていないかどうかを確認してください。; 2、回線を点検してください。 3、弊社までご連絡ください。
35009	サーボドライバ電圧限界	1、速度が速すぎる; 2、モーターの異常	1.速度をチェックして異常がないか確認してください。 2、モーターの異常を調べてください 3、弊社までご連絡ください。
35010	サーボドライバの+側限界 ON	サーボドライバの+側限界オーバー	サーボドライバのリミットを修復してください。
35011	サーボドライバの-側限界 ON	サーボドライバの-側限界オーバー	サーボドライバのリミットを修復してください。
35012	サーボドライバ追従エラー	ユーザー設定の追従エラーを超えました	上位コマンドに問題がないかを確認してください。ロボットの状態を確認してください。.
35013	位置カウンター到着	サーボドライバ内部エラー	再起動を試みて回復しない場合は、弊社までご連絡ください。
35014	障害に他の緊急事態がない場合に適用	サーボドライバ内部エラー	再起動を試みて回復しない場合は、弊社までご連絡ください。
35015	ノードエラー	サーボドライバ内部エラー	再起動を試みて回復しない場合は、弊社までご連絡ください。
35016	サーボドライバコマンドエラー	1、このエラーは正確でない場合があります。 2、PWM またはその他のコマンド信号が存在しない	入力信号を修復してください。
35100	サーボドライバ DC バス電圧不足	サーボドライバがリアルタイムに設定された母線電圧の低電圧保護閾値より小さいことで検出します。母線給電電圧が不足している可能性があります。	1、ロボットの給電電圧が正常であることを確認してください。 2、サーボドライバの容量が正常かどうかを確認してください。 3、サーボドライバを交換してください。
35101	サーボドライバ DC 母線の過電圧	サーボドライバはリアルタイムに設定された母線電圧の過電圧保護閾値より高いことで検出します。以下の原因による可能性があります。1、母線給電電圧が高すぎる 2、ロボットの減速が速すぎ	1、ロボットの電源電圧を確認してください。 2、ロボットの減速度を確認してください。 3、ロボット電源管理回路を確認してください。

		る 3、ベース電源処理異常	
35102	サーボドライバーマーター過電流	サーボリアルタイム検出電流ベクトルの大きさが設定された安全保護閾値より大きく、モーターが運転中に失速することで電流が急騰した可能性があります。	1、モーター三相線と 48 V 動力線が正しく接続されていることを確認してください。 2.モーターの偏角が正しいかどうかを確認してください。
35103	サーボドライバーマーター過負荷	関節モーターが連続的に定格トルク保護閾値範囲を超え、サーボが閉ループ制御のため、モーターの実際位置が所定位置に追跡できない場合にトルク電流が大きくなり、一定の持続時間まで達するとエラーが発生します。	1、モーター三相線とロック線が正しく接続されていることを確認してください 2、モーターの偏角が正しいかどうかを確認してください。
35104	サーボドライバーマーター磁気通過限界	励磁電流幅値が設定値より小さいかどうかを判断して、設定値を超過するとエラーが発生します。弱磁制御では励磁電流を変えて更に高い回転速度に達するようにしますが、低圧サーボが適合するモーターでは母線電圧は十分で、弱磁を行う必要がないため、このエラーは発生しません。	必要な場合はサーボドライバを最新のファームウェアバージョンに更新する必要があります。必要な場合はサーボドライバを最新のパラメーターバージョンに更新する必要があります。
35105	サーボドライバーストール警報	関節速度追跡誤差が設定した閾値より大きく、かつ 1 s 以上継続するとエラーが発生します。起動段階で失速し、モーターが正常に起動すると新たに偏角を探す必要があります。モーターの運転中に失速する場合はハードウェアの配置とソフトウェアのバージョンとパラメーターをチェックする必要があります。	1、モーター三相線とブレーキ線が正しく接続されていることを確認してください。 2、モーターの偏角が正しいかどうかを確認してください。 3、ハードウェア構成を確認してください。 4、サーボソフトウェアバージョンを確認してください。
35106	サーボドライバ位置誤差過大	関節位置追跡誤差が設定した閾値を超えるとエラ	1、配線が正しく接続されていることを確認してください。

		ーが発生します。ハードウェアとソフトウェアが正しい構成とバージョンであるかどうかを確認する必要があります。	2、関節エンコーダーが正常かどうかを確認してください。 3、関節ハードウェア、モーター、減速器、駆動器を交換してください。 4.主局の指示が適正かどうかを確認してください。
35107	サーボドライバーゼロ電流アラーム	ゼロ電流値は設定した閾値を超える必要があります。無効時は電流は収集異常となります。	1、サーボドライバーを最新のファームウェアバージョンに更新してください。 2、サーボドライバーを最新のパラメーターバージョンに更新してください。 3、サーボドライバーを交換してください。
35108	サーボドライバーA 相過電流	1、モーターのA相電流がサーボドライバー電流サンプリング範囲より大きい。 2、モーターが遮断された。 3、負荷が大きすぎる、または回転速度が速すぎる。 4、関節機構の抵抗が大きい。	1、配線が正しく接続されていることを確認してください。 2、モーターの偏角が正しいかどうかを確認してください。 3、ブレーキの開閉が正しく機能するか開くかどうかを確認してください。 4、負荷が大きい時、回転速度を下げてエラーが消えるかどうかを確認してください。 5、運転中のエラー発生時の関節軸に抵抗が大きすぎるような異常騒音がないかを観察してください。
35109	サーボドライバーB 相過電流	1、モーターのB相電流がサーボドライバー電流サンプリング範囲より大きい。 2、モーターが遮断された。 3、負荷が大きすぎる、または回転速度が速すぎる。 4、関節機構の抵抗が大きい。	1、配線が正しく接続されていることを確認してください。 2、モーターの偏角が正しいかどうかを確認してください。 3、ブレーキの開閉が正しく機能するか開くかどうかを確認してください。 4、負荷が大きい時、回転速度を下げてエラーが消えるかどうかを確認してください。 5、運転中のエラー発生時の関節軸に抵抗が大きすぎるような異常騒音がないかを観察してください。
35110	サーボドライバーC 相過電流	1、モーターのC相電流がサーボドライバー電流サンプリング範囲より大きい。 2、モーターが遮断された。 3、負荷が大きすぎる、または回転速度が速すぎる。	1、配線が正しく接続されていることを確認してください。 2、モーターの偏角が正しいかどうかを確認してください。 3、ブレーキの開閉が正しく機能するか開くかどうかを確認してください。 4、負荷が大きい時、回転速度を下げてエラーが消えるかどうかを確認してください。

		4、関節機構の抵抗が大きい。	5、運転中のエラー発生の際関節軸に抵抗が大きすぎるような異常騒音がないかを観察してください。
35111	サーボドライバークイック過電流	1、モーター電流が設定したしきい値を超えた。 2、モーターが遮断された。 3、負荷が大きすぎる、または回転速度が速すぎる。 4、関節機構の抵抗が大きい。	1、配線が正しく接続されていることを確認してください。 2、モーターの偏角が正しいかどうかを確認してください。 3、ブレーキの開閉が正しく機能するか開くかどうかを確認してください。 4、負荷が大きい時、回転速度を下げてエラーが消えるかどうかを確認してください。 5、運転中のエラー発生の際関節軸に抵抗が大きすぎるような異常騒音がないかを観察してください。
35112	サーボドライバークイックオーバーフロー		
35113	サーボドライバークイックイメージのオーバーフロー		
35114	サーボドライバークイックH スレッドのタイムアウト		
35115	サーボドライバークイックM スレッドのタイムアウト	1、サーボソフトウェアスケジューリング異常。 2、指定された時間内にプログラムが完了しない。 3、ファームウェアのバージョンまたはパラメータのバージョンが最新でない可能性があります。	1、サーボドライバークイックを最新のファームウェアバージョンに更新してください。 2、サーボドライバークイックを最新のパラメータバージョンに更新してください。 3、サーボドライバークイックを交換してください。
35116	サーボドライバークイックL スレッドのタイムアウト		
35117	サーボドライバークイックC スレッドのタイムアウト		
35118	サーボドライバークイック中断ハンガアップ	1、サーボソフトウェアスケジューリング異常。 2、指定された時間内にプログラムが完了しない。 3、ファームウェアのバージョンまたはパラメータのバージョンが最新でない可能性があります。	1、サーボドライバークイックを最新のファームウェアバージョンに更新してください。 2、サーボドライバークイックを最新のパラメータバージョンに更新してください。 3、サーボドライバークイックを交換してください。
35119	サーボドライバークイックマスタハンガアップ		
35120	サーボドライバークイックハー		

	ドウェアが不正です。		
35121	サーボドライバードバ ッグタイムアウト		
35122	サーボドライバーク ジェクト ID タイムア ウト		
35123	サーボドライバース レッド ID エラー		
35124	サーボドライバーク 変数 ID エラー		
35125	サーボドライバーク 変数 ID エラー		
35126	サーボドライバーク 変数 ID エラー		
35127	サーボドライバーク データ型エラー		
35128	サーボドライバーク データの対象が存在済		
35129	サーボドライバーク データ配列が存在しない		
35130	サーボドライバーク データ配列のオーバーフ ロー		
35131	サーボドライバーク データ配列が空		
35132	サーボドライバーク アプリケーションのバ ージョン間違い		
35133	サーボドライバーク パラメーターの検証 エラー		
35134	サーボドライバーク パラメーター番号が 間違い		
35135	サーボドライバーク パラメーターID が 間違い		
35136	サーボドライバーク パラメーターID が 間違い		
35137	サーボドライバーク パラメーターID が 間違い		
35138	サーボドライバーク SPI の通信エラー	DSP とパラメーターメモ リの通信が異常。ファ ームウェアのバージョ ンまたはパラメーター のバ	1、サーボドライバークを最新のファ ームウェアバージョンに更新してく ださい。 2、サーボドライバークを最新のパ ラメーターバージョンに更新してく ださい。

		ジョンが最新でない可能性があります。またはサーボドライバーハードウェアの異常	3、サーボドライバーを交換してください。
35139	サーボドライバーE2Pのオーバーフロー	パラメーターメモリパラメーターの読み取りが多すぎる。ファームウェアのバージョンまたはパラメーターのバージョンが最新でない可能性があります。またはサーボドライバーハードウェアの異常	1、サーボドライバーを最新のファームウェアバージョンに更新してください。 2、サーボドライバーを最新のパラメーターバージョンに更新してください。 3、サーボドライバーを交換してください。
35140	サーボドライバーのダイナミックメモリが空	サーボソフトウェア実行ロジックエラー。ファームウェアのバージョンまたはパラメーターのバージョンが最新でない可能性があります	1、サーボドライバーを最新のファームウェアバージョンに更新してください。 2、サーボドライバーを最新のパラメーターバージョンに更新してください。 3、サーボドライバーを交換してください。
35141	サーボドライバー命令誤り	サーボソフトウェア実行ロジックエラー。ファームウェアのバージョンまたはパラメーターのバージョンが最新でない可能性があります	1、サーボドライバーを最新のファームウェアバージョンに更新してください。 2、サーボドライバーを最新のパラメーターバージョンに更新してください。 3、サーボドライバーを交換してください。
35142	サーボドライバー命令長誤り	サーボソフトウェア実行ロジックエラー。ファームウェアのバージョンまたはパラメーターのバージョンが最新でない可能性があります	1、サーボドライバーを最新のファームウェアバージョンに更新してください。 2、サーボドライバーを最新のパラメーターバージョンに更新してください。 3、サーボドライバーを交換してください。
35143	サーボドライバーMeterのチャンネル番号間違い		
35144	サーボドライバーOSCIチャンネル番号間違い		
35145	サーボドライバー静的モードエラー	サーボ実行モードエラー。サーボファームウェアとパラメーターが最新バージョンではない可能性があります;またはサーボパラメーターの書き込みに失敗しました	1、サーボドライバーを最新のファームウェアバージョンに更新してください。 2、サーボドライバーを交換してください。
35146	サーボドライバー動的モードエラー	サーボ実行モードエラー。サーボファームウェア	1、サーボドライバーを最新のファームウェアバージョンに更新してください。

		アとパラメーターが最新バージョンではない可能性があります;またはサーボパラメーターの書き込みに失敗しました	2、サーボドライバーを交換してください。
35147	サーボドライバーモード変更エラー	サーボ実行モードエラー。サーボファームウェアとパラメーターが最新バージョンではない可能性があります;またはサーボパラメーターの書き込みに失敗しました	1、サーボドライバーを最新のファームウェアバージョンに更新してください。 2、サーボドライバーを交換してください。
35148	サーボドライバーIPMアラーム	サーボファームウェアとパラメーターが最新バージョンではない可能性があります	1、サーボドライバーを最新のファームウェアバージョンに更新してください。 2、サーボドライバーを交換してください。
35149	サーボドライバーモジュール過熱アラーム	関節温度が安全で確実に作動する温度範囲を超えた	相応の冷却処理を行ってください
35150	サーボドライバーのSTOスイッチがONになっていない	サーボは継続 STO 信号レベルを検出しており、設定ロジックと合わない場合にエラーが発生する。ロボットがハンドルを有効にしていない可能性があります。	1、手動でハンドルを押す。 2、STO ケーブルを抜き差ししてください。 3、STO ケーブルもしくはサーボドライバーを交換してください。
35151	エンコーダーエラー (0x7371)	QEP タイプエンコーダーでのエラーです。ABZ 線が切断されると発生します。サーボが最新のファームウェアとパラメーターにアップグレードされていない可能性があります。	1、サーボドライバーを最新のファームウェアバージョンに更新してください。 2、サーボドライバーを最新のパラメーターバージョンに更新してください。
35152	エンコーダーエラー (0x7372)		
35153	エンコーダーエラー (0x7373)		
35154	エンコーダーエラー (0x7374)		
35155	エンコーダーエラー (0x7375)		
35156	エンコーダーエラー (0x7376)		
35157	エンコーダーエラー (0x7377)		

35158	エンコーダーエラー (0x7378)	エンコーダーチップのエラー:Biss2 通信異常	1、関節エンコーダー線を確認してください。 2、エンコーダーを交換してください。 3、エンコーダーケーブルを交換してください 4、サーボドライバを交換してください
35159	エンコーダーエラー (0x7379)	エンコーダーチップエラー:Biss1 通信異常、エンコーダーデコードチップが見つかりません	1、関節モーター側コード線を点検してください。 2、エンコーダーを交換してください 3、エンコーダーケーブルを交換してください 4、サーボドライバを交換してください
35160	エンコーダーエラー (0x737A)	1、エンコーダー通信CRC エラー 2、エンコーダー通信異常	1、エンコーダーケーブルを抜き差ししてください。 2、エンコーダーを交換してください。 3、エンコーダーケーブルを交換してください。 4、サーボドライバを交換してください
35161	エンコーダーエラー (0x737B)	エンコーダー通信異常。エンコーダーCRC チェックは通過しましたが、エンコーダーチップがエラーが発生しました。	1、エンコーダーケーブルを抜き差ししてください。 2、エンコーダーを交換してください。 3、エンコーダーケーブルを交換してください。 4、サーボドライバを交換してください
35162	エンコーダーエラー (0x737C)	1、単一エンコーダーハードウェアの故障 2、エンコーダーのバッテリーが切れました。	1. 1つのエンコーダーでエラーが発生し、再起動しても回復しない場合は、テクニカルサポート部門にご連絡ください。 2. 全軸同時にエラーが発生した場合は、弊社に連絡してエンコーダーバッテリーを交換してください。
35163	TZ トリガー		
35164	内部エラー(0xFF91)		
35165	上位イナーブル電流ループが 0 に設定 (0xFFA2)。		
35166	上位位置指示追跡不可 (0x8612)		
35201	2 次エンコーダー5 v 過電流		CDHD は、2 次エンコーダーに最大 250 mA の電流を供給可能です。エンコーダーがショートしていないかどうかを確認して下さい。エンコーダーの描画が現在の制限を超えていないかどうかを確認します
35202	過電流	1、サーボドライバまたはモーター故障 2、ロボットの負荷が限界	1、モーター接続がショートしていないか確認してください。 2、電流回路がオーバーシュートしていない

		値超過 3、衝突などの発生	いか確認してください。
35203	モーターフォールドバック	1、サーボドライバーまたはモーター故障 2、ロボットの負荷が限界 値超過 3、衝突などの発生	1、サーボドライバー、モーターの仕様を確認してください。 2、このエラーはモーターの容量が小さすぎる場合(低電力)に発生する可能性があります。
35204	駆動フォールドバック		1、サーボドライバー、モーターの仕様を確認してください。 2、このエラーはモーターの容量が小さすぎる場合(低電力)に発生する可能性があります。 3、適切な方向転換角であることを確認してください(整流のバランスが取れているかどうかを確認してください)。
35205	電流センサー補正が無効	サーボドライバー故障	再起動してエラーが解決しない場合はサーボドライバーを交換する必要があります。弊社までご連絡ください。
35206	モーター位相切断	サーボドライバー故障	モーター位相配線を確認してください。
35207	出力過電流検出	サーボドライバー故障	デジタル出力の正しい配線を確認し、出力回路がショートしていないことを確認してください。
35208	過電圧		回生抵抗を適用する必要があるかどうかを確認してください。
35209	低電圧故障		主 AC 電源がサーボドライバーに接続され、オンになっていることを確認してください。不足分の制限は、UVTHRESH コマンドで読み取ることができます。
35210	再生過電流		回生抵抗値の増やしてください
35211	STO 故障		STO コネクタ(P1)が正しく接続されていることを確認してください。
35212	Vbus 測定回路の障害		再起動してエラーが解決しない場合はサーボドライバーを交換する必要があります。弊社までご連絡ください。
35213	バス AC 電源オフ		配線を確認してください。弊社までご連絡ください。
35214	回生抵抗過負荷		回生抵抗器の特性がアプリケーションに適しているかどうかを確認してください。
35215	内蔵電源モジュール過熱		周囲温度がサーボドライバーの仕様を超えていないかどうかを確認してください。問題が解決しない場合は弊社 E までご連絡ください。
35216	制御板過熱		周囲温度がサーボドライバーの仕様を超えていないかどうかを確認してください。問

			題が解決しない場合は弊社までご連絡ください。
35217	温度センサー失敗		電源を入れ直してください。問題が解決しない場合は弊社までご連絡ください。
35218	電力レベル過熱		周囲温度がサーボドライバ仕様を超えていないかどうかを確認してください。問題が解決しない場合は弊社までご連絡ください。
35219	モーター過熱故障		1、サーボドライバが正しく構成されていることを確認してください。 (THERMODE、THERMTYPE、THERMTHRESH、および THERMTIME を使用) 2、必要に応じてモーター温度センサーがサーボドライバに正しく接続されていることを確認してください。 3、サーボドライバが正しく構成され、配線されている場合は、モーターのサイズが小さすぎるかどうかを確認してください。
35220	内部電源供給が範囲外		サーボドライバの交換が必要の可能性があります。弊社までご連絡ください。
35221	5 v 範囲外		停電時に発生する可能性があります。問題が解決しない場合は弊社までご連絡ください。
35222	電源 EEPROM 故障		弊社までご連絡ください。
35223	制御 EEPROM 故障		弊社までご連絡ください。
35224	CAN 供給障害		サーボドライバの交換が必要の可能性があります。弊社までご連絡ください。
35225	セルフテスト失敗		弊社までご連絡ください。
35226	パラメーターのメモリチェック失敗		サーボドライバを再構成するか、パラメーターセットをダウンロードしてパラメーターを保存してください。問題が解決しない場合は弊社までご連絡ください。
35227	フラッシュ書込み失敗		弊社までご連絡ください。
35228	フィールドバス速度制限超過		サーボドライバを有効にし、有効なロケーションコマンドを送信してください。
35229	未構成		駆動パラメーターの設定後に CONFIG を実行してください。
35230	FPGA の構成に失敗しました		弊社までご連絡ください。
35231	モーターの取付けに失敗		位相とモーター接続を確認し、正しいフィードバックタイプが選択されていることを確認してください。MOTORSETUPST 提

			示を確認してください。
35232	位相探索失敗		モーターフィードバックタイプとフェーズ検索パラメーターが設定されているかどうかを確認してください
35233	FPGA のバージョン不一致		FPGA のバージョンまたはドライバーのバージョンを更新してください
35234	緊急停止発令		入力をオフしてください。
35235	フィールドバスバージョン不一致		サーボドライバーに正しいバージョンがダウンロードされていることを確認してください。
35236	ESI のバージョン不一致		サーボドライバーに正しいバージョンがダウンロードされていることを確認してください。
35237	BiSS-C エンコーダー内部エラー		BiSS-C エンコーダー取扱説明書を参考にしてください。
35238	HIPERFACE エンコーダーデータエラー		サーボドライバーソフトウェアを使用して、コマンド HSAVE 1 を入力してください。
35239	ESI ベンダーの不一致		サーボドライバーに正しいベンダデータがダウンロードされていることを確認してください。
35240	インデックス行割込み		サーボドライバーがインデックスシグナル (MENCTYPE を使用) と連動するように構成されていることを確認してください。また、インデックスシグナルが接続されていることを確認してください。
35241	ダイナミックブレーキ (電源制動器) 負荷オン		
35242	ダイナミックブレーキ短絡		
35243	回転停止問題		失速状態を取り除き、失速状態を防ぐように注意してください。
35244	2 次フィードバック指数割込み		1、サーボドライバーが 2 次エンコーダーのインデックス信号を使用するように構成されているかどうかを確認してください。 2、インデックス信号が接続されているかどうかを確認してください。
35245	2 次フィードバック A/B 断線		2 次エンコーダーからのすべての信号がサーボドライバーに正しく接続されていることを確認してください。
35246	パルスおよび方向入力断線		すべての P&D (パルスと方向) 入力信号がサーボドライバーに正しく接続されていることを確認してください。
35247	ダイナミックブレーキ		モーターブレーキを交換してください

	(電源制動器)故障		
35248	モーター制御不能状態 検出		MPHASE の設定を確認し、フェーズ検出 プロセスを有効にして改善してください。
35249	通信エラーがフィード バックされましたエン コーダー通信が切断さ れた	1、本体とコントローラー 回線接続不良 2、エンコーダーやサーボ ドライバーが故障	フィードバック装置が正しく接続されてい ることを確認してください。正しいエンコ ーダータイプが選択されているかどうかを 確認してください(MENCTYPE)
35250	ニコンエンコーダー動 作不良		フィードバック装置が正しく接続されてい ることを確認してください。正しいエンコ ーダータイプが選択されているかどうかを 確認してください(MENCTYPE)
35251	Tamagawa 初期失敗		エンコーダーが正しく接続されているかど うかを確認してください。
35252	A/B 断線		正弦波エンコーダーとデコーダの診断のセ クションを参照してください。オートフィ ードバックデバイスからのすべての信号が サーボドライバーに正しく接続されている ことを確認してください。
35253	無効 Halls		ホール信号が正しく接続されていることを 確認してください。モーターを回転させる ときは、(HALLS を使用して)ホール状態を 見てどの信号が接続されていないかを確認 してください。フィードバックタイプが Tamagawa の場合は、フィードバックコード が正しいかどうかを確認してください。
35254	アブソリュートエンコ ーダー電池電圧低下	電池は消耗品ですので、 交換する際には、弊社ま で連絡して購入してくだ さい。	バッテリーを交換し、サーボドライバーをリ セットしてください。サーボドライバーの 電源が入っているときにバッテリーを交換す ると、位置情報が保持される。
35255	位相ロックループの同 期失敗		コントローラーの同期信号を確認し、ケー ブルの配線または接続を確認してくださ い。
35256	エンコーダーのアナロ グ周波数が高すぎる		使用エンコーダー出力パラメーターの設定 をチェックしてください。正弦エンコーダ ーを使用している場合は、ENCOUTRES パラメーター設定を確認してください。
35257	TamagawaAbs 操作エ ラー		バッテリー残量とフィードバックケーブルを 確認してください。エンコーダーの初期化 時はモーターを高速動作させないようにし てください。
35258	カスタムアブソリュ ートエンコーダー動作不 良		バッテリー残量とフィードバックケーブルを 確認してください。エンコーダーの初期化 時はモーターを高速動作させないようにし てください。
35259	差動 Halls 線切断		HALLSTYPE が使用しているホールセンサ

			ーと一致していることを確認してください (シングルエンドまたは差分)。差分ホールセンサからのすべての信号がセンサに正しく接続されていることを確認してください。
35260	エンコーダー位相エラー		MENCAQBFILT を 0 に設定して、 A および B 信号のフィルタを削除してください。問題が解決しない場合は、エンコーダーに問題がある可能性があります。
35261	AqB 位相変換エラー		動作開始直後にエラーが発生した場合は、 MENCRES の設定を確認してください。しばらく動作してからエラーが発生する場合は、EMI ノイズが原因である可能性があります。取付けを改善し、アースを確保してください。シールド線がフィードバックケーブルとモーターケーブルに接続されていることを確認してください。
35262	SensAR エンコーダーエラー		SRVSNSINFO コマンドを使用してエラーを識別してください。
35263	正弦波フィードバック通信失敗		EnDat に渡されたデータとクロック信号が正しく接続されていることを確認してください。ケーブルをシールドする必要があります。
35264	A/B 範囲外		正弦エンコーダーとデコーダの診断のセクションを参照してください。正弦波信号と余弦波信号の振幅を調べてください。
35265	Sankyo アブソリュートエンコーダーエラー		バッテリー残量とフィードバックケーブルを確認してください。エンコーダー初期化時にモーターを高速で動作させないようにしてください。
35266	正弦波符号器直交故障		フィードバックデバイスの接続を確認してください。正しいエンコーダータイプ (MENCTYPE) が選択されていることを確認してください。
35267	サイン/コサインの校正無効		Sin/Cos 校正プロセスを再実行してください。
35268	フィードバック 5 V 過電流		CDHD はメインエンコーダーに 250 mA の電流を供給できます。エンコーダーがショートしていないかどうかを確認してください。エンコーダーの描画が現在の制限を超えていないかどうかを確認してください。
35269	デコーダ初期化失敗		分解器接続ボックスのゲイン値を確認してください
35270	Endat2X フィードバック		エンコーダーの電源を切ってエンコーダー

	クエラー		をリセットして下さい。
35271	フィールドバスケーブル接続解除		コントローラーとサーボドライバー間の接続を再度実施してください
35272	フィールドバス制御コマンド喪失		エラーをクリアし、コントローラーが新しいコマンドを送信できるようにしてください。
35273	CAN ハートビート欠損		マスターとスレーブを再接続し、ドライブを再起動してください。
35274	ドライブロック		弊社までご連絡ください。
35275	EtherCAT ドロップ		EtherCAT プライマリステーション(コントローラー)が定義された時間内にパケットを送信するようにしてください(マスターデバイスより)。
35276	モーメントフィードバックのオーバーシュート		
35277	不安定電流回路		現在のコントローラー設定を確認してください。
35278	オーバーシュート		VLIM 設定が要件に一致するかどうかを確認してください。速度ループを調整して、オーバーシュートをチェックしてください。
35279	最大速度誤差超過		サーボドライバーのチューニングを行って速度トラッキングを改善するか、VEMAXを増やして位置誤差を大きくしてください。
35280	最大位置誤差超過		駆動チューニングを行い、位置追跡を改善するか、PEMAX を増やして位置誤差を大きくしてください。
35281	二次フィードバック位置の不一致		SFBPETHRESH、STBPETIME.SFBPEMAX を増やすか、位置の調整向上を図ってください。
35282	過度 PE 値		確認、調整を行ってください。
35283	CAN/EtherCAT 状態操作不可		サーボドライバーが無効になっているときに、コントローラーが低い通信状態に切り替わらないようにしてください。
35284	内部エラー		弊社までご連絡ください。
35285	モーター読み取り失敗		フィードバックユニットを再接続してください。モータータイプの銘板データが存在する場合は確認してください。
35286	電源要件の保存と再起動		保存し、ドライブの電源を入れ直してください。
35287	リアルタイム過負荷障害		弊社までご連絡ください。

35288	PFB シャットダウン検証無効		必要に応じて、ロボットを元の位置に戻してください。
35289	PFB シャットダウンデータ不一致		必要に応じて、ロボットを元の位置に戻してください。
35290	PFB シャットダウンデータがない		必要に応じて、ロボットを元の位置に戻してください。
35291	パルス周波数が高すぎる		制御器よりギア伝動パルスの周波数を下げてください。
35301	ドライブのショート	1、サーボドライバの U、V、W 出力にショートが発生 2、ドライバが妨害されて DI 信号が異常。誤検出。 3、アース未接続 4、電流ループ調節器のパラメータ設定が適切でないため、電流振動による妨害を誘発する 5、ドライブの破損(例えば IGBT 短絡,電流検出回路の異常).	1、サーボドライバの U、V、W 配線のトラブルシューティング(モーター動力ケーブルを切断した後、まだショート故障を表示しているかどうかを確認)を行ってください。 2、マルチメータを使用してドライブの IGBT チェックし、ショートしていないかどうかを確認してください。 3、配線、特にアース線の仕様を確認してください。 4、電流ループパラメータの調整を行ってください。 5、サーボドライバを交換してください。
35302	ドライブ出力のベアショート	1、ドライブの U、V、W 出力の対地短絡 2、ドライブの破損(例えば電流検出回路の異常)。	1、サーボドライバの U、V、W 配線のトラブルシューティング(モーター動力ケーブルを切断した後、まだショート故障を表示しているかどうかを確認)を行ってください。 2、サーボドライバを交換してください
35303	エンコーダーデータ異常	1、Hiperface エンコーダーエラー 2、エンコーダー結線エラー 3、ドライブの内部 AD アライメントパラメータ異常	1、エンコーダーの配線を確認し、配線が正しいことを確認してください。 2、サーボドライバを交換してください。 3、エンコーダーを交換してください。
35304	ロータ位置決めエラー	1、位置ループ、速度ループ、電流ループ調節器パラメータの設定不適切 2、モーターパラメータ設定エラー 3、パラメーター 0x20D2 設定が小さすぎる 4、パラメーター 0x2003 の設定エラー	1、サーボドライバの周辺配線を確認し、配線が正しいことを確認してください。 2、サーボドライバの位置ループ、速度ループ、電流ループ調節パラメータを調整し、モーターパラメータが正しく設定されていることを確認してください。 3、新たにロータ位置補正角をチェックして再調整してください。

		5、ドライブ周辺の配線エラー(例えば、モーターの動力線、モーターのコード線) 6、ドライバー内部回路の異常 7、パラメーター0x2207 静的平衡モーメント補償値の設定が不適切	4、パラメーター0x20D2 設定値を大きくしてください。 5、パラメーター0x2207 静的平衡モーメント補償値の設定値を小さくしてください。6、サーボドライバーを交換してください。
35305	モーターブレーキ異常	1、モーターブレーキ異常 2、モーターが高速運転時に突然サーボ OFF 3、サーボパラメーター0x2233, 0x20D2 設定が小さすぎる 4、モーターブレーキ回路に短絡現象がある。	1、パラメーター0x2233 および 0x20D2 を大きくしてください。 2、モーターブレーキ回路を点検してください。 3、モーターブレーキを交換してください。
35306	EtherCAT PDO 構成エラー	1, PDO 設定エラー。	1、EtherCAT PDO 設定を修正して下さい。
35307	エンコーダー内部通信異常	1、エンコーダーのエラー 2、モーターエンコーダー結線異常(ex.断線では、シールド付きツイスト・ペアは用いず、モーターの動力線と結合) 3、サーボドライバーのアースが正しく接続されていない。 4、サーボドライバー周囲に強いノイズ源があります。	1、モーターエンコーダーの配線を確認し、配線仕様が正しいことを確認してください。 2、エンコーダーケーブル、モーターのパワーケーブルにフェライトコアを追加してください。 3、サーボドライバーに信頼性の高いアース接続を実施してください。 4、モーターエンコーダーを交換してください 5、サーボドライバーの周囲に強力なノイズ源がないかを確認して、ある場合は対策を実施して除去してください。または、サーボドライバーが周囲の強力なノイズ源と独立して電源を供給するようにしてください。 6、サーボドライバーの入力電源線にフィルタを追加してください。
35308	エンコーダータイプの変更	2、エンコーダータイプ異常。	1、ドライブの電源を入れ直すか、ソフトリセットしてください。 2、エンコーダータイプを交換してください。
35309	ドライブ過電流 U	1、電流ループ調節器のパラメーター設定が不合理で、電流制御振動を引き起こす	1、電流ループレギュレータパラメーターを調整してください。 2、モーターパラメーターを正しく設定してください。

		2、モーターパラメーター設定エラー 3、サーボドライバ内部電流サンプリング回路異常。	3、サーボドライバを交換してください。
35310	ドライブ過電流 V	1、電流ループ調節器のパラメーター設定が不合理で、電流制御振動を引き起こす 2、モーターパラメーター設定エラー 3、サーボドライバ内部電流サンプリング回路異常。	1、電流ループレギュレータパラメーターを調整してください。 2、モーターパラメーターを正しく設定してください。 3、サーボドライバを交換してください。
35311	ドライブの過電流 W	1、電流ループ調節器のパラメーター設定が不合理で、電流制御振動を引き起こす 2、モーターパラメーター設定エラー 3、サーボドライバ内部電流サンプリング回路異常。	1、電流ループレギュレータパラメーターを調整してください。 2、モーターパラメーターを正しく設定してください。 3、サーボドライバを交換してください。
35312	直流母線過圧	1、サーボドライバのパワー入力電源電圧が高すぎる 2、モーターの急停止時のエネルギー消費制動エネルギーが大きすぎる 3、モーターの停止減速度が大きすぎる 4、エネルギー消費ブレーキ抵抗の配線ミス 5、エネルギー消費ブレーキの抵抗値が大きすぎる 6、サーボドライバ内部電圧サンプリング回路の異常 7、サーボドライバ内部のエネルギー消費制動回路が異常	1、サーボドライバの電源入力を許容範囲内に調整してください。 2、モーター停止減速度の低減3、回生抵抗の配線をチェックし、配線が正しいことを確認してください。 3、.回生抵抗値(抵抗値は最小許容値を下回ってはいけません)を適当に減らし、回生抵抗での消費電力を増やす。
35313	制御電圧不足	1、24V 制御電源異常 2、24V 配線不良などの電源配線不良の制御 3、24V 電源負荷の制御;	1、24 V 制御電源の配線を確認し、配線が確実であることを確認してください。 2、.24 V 制御電源負荷を検査して、24 V 制御電源の容量がすべての運転状況での負荷

		4、サーボドライバ内部の回路異常	消耗に満足できることを確保してください。 3、24 V 制御 PSU の交換してください。 4、サーボドライバを交換してください。
35314	ドライブの持続的な過負荷	1、モーターの負荷が大きすぎる 2、実際の機械負荷が大きすぎる 3、機械負荷にはデッド現象がある 4、モーターブレーキが解除されていない 5、モーター加減速時間の設定が小さすぎる 6、サーボドライバ内部電流サンプリング回路の異常 7、サーボドライバのロック回路が異常	1、モーターの実際の機械負荷を低減してください。 2、モーターの加減速時間を大きくしてください。 3、機械負荷伝動方式を検査して、機能不良等の現象が発生しないことを確認してください。 4、モーターブレーキ配線をチェックして配線の信頼性を確保してください。 5、モーターを交換してください。 6、サーボドライバを交換してください。
35315	エンコーダ結線エラー	1、バージョン CDA8 V1 製品の場合、エンコーダ線端コネクタピン 8 とピン 15 はショートされていません 2、バージョン CDA8 V2 製品および CDR シリーズ製品の場合、エンコーダ線端コネクタピン 6 およびピン 8 はショートされていません 3、エンコーダケーブルの結線不良。	1、エンコーダケーブルの結線を確認してください。(バージョン CDA8 V1 の製品の場合、ショートコネクタのワイヤエンドコネクタピン 8 およびピン 15) ;(バージョン CDA8 V2 製品および CDR シリーズ製品では、ショート・ケーブル・コード・ケーブル・エンドのコネクタ・ピン 6 とピン 8 を使用して、コード・ケーブルの配線を確実にしてください)
35316	CPU 過負荷	1、サーボドライバがノイズの影響を受けている。 2、サーボドライバ内部回路の異常。 3、DriveStarter データ収集が大きすぎる。	1、サーボドライバ周辺の配線を標準化し、ノイズ対策を強化してください。 2、サーボドライバを交換してください。 3、セクション DriveStarter データ収集チャンネルを閉じてください。
35317	ドライブ出力のデフォルト	1、サーボドライバU、V、W の出力に断線や配線不良などがある 2、モーターのインピーダンスが大きすぎる 3、サーボドライバ内部	1、モーターU、V、W の配線をチェックし、配線が確実であることを確認してください。 2、モーターを交換してください。またはドライブ出力のデフォルト検出機能をオフにしてください。

		電流サンプリング回路が異常	3、サーボドライバを交換してください。
35318	ドライブの一時的な過負荷	1、モーターの負荷が大きすぎる 2、サーボドライバ内部温度サンプリング回路の異常 3、サーボドライバの動作環境温度が許容範囲を超えている 4、サーボドライバの動作が妨げられている(例えば同期していません) 5、モーターの加速度および減速度の設定が大きすぎる or 加減速時間の設定が小さすぎます。	1、モーターの実際の機械負荷を低減してください。 2、.サーボドライバ周辺の配線を標準化し、ノイズ対策を強化してください。 3、コントローラの冷却条件を改善するなど、環境温度を下げてください。 4、サーボドライバを交換してください。
35319	エンコーダ外部通信送信異常	1、モーターエンコーダ結線異常(ex.断線では、シールド付きツイスト・ペアは用いず、モーターの動力線と結合する); 2、サーボドライバのアースが正しく接続されていない 3、サーボドライバの周囲に強いノイズ源がある。	1、モーターエンコーダの配線を確認し、配線仕様が正しいことを確認してください。 2、.エンコーダケーブル、モーターのパワーケーブルにフェライトコアを追加してください。 3、サーボドライバに信頼性の高いアース接続を実施してください。 4、.サーボドライバの周囲に強力なノイズ源がないかを確認して、ある場合は対策を実施して除去してください。または、サーボドライバが周囲の強力なノイズ源と独立して電源を供給するようにしてください。 5、サーボドライバの入力電源線にフィルタを追加してください。
35320	エンコーダ外部通信受信異常	1、モーターエンコーダ結線異常(ex.断線では、シールド付きツイスト・ペアは用いず、モーターの動力線と結合する); 2、サーボドライバのアースが正しく接続されていない 3、サーボドライバの周囲に強いノイズ源がある。	1、モーターエンコーダの配線を確認し、配線仕様が正しいことを確認してください。 2、.エンコーダケーブル、モーターのパワーケーブルにフェライトコアを追加してください。 3、サーボドライバに信頼性の高いアース接続を実施してください。 4、サーボドライバの周囲に強力なノイズ源がないかを確認して、ある場合は対策を実施して除去してください。または、サーボドライバが周囲の強力なノイズ源と独立して電源を供給するようにしてください。

			立して電源を供給するようにしてください。 5、サーボドライバーの入力電源線にフィルタを追加してください。
35321	サーボドライバーハードウェアの過電流	1、機械負荷に固着現象或いは渋滞現象が存在する。 2、.ロータ補償角の設定が正しくない。 3、エンコーダー結線エラー。 4、電流ループ調節器のパラメーター設定が不合理で、電流制御振動を引き起こす。 5、モーターパラメーター設定エラー(線抵抗、線インダクタンス、逆起電力など) 6、サーボドライバー内部電流検出回路の異常。 7、サーボドライバーブレーキ回路が故障しており、24 V 出力がない 8、モーターブレーキの破損。9、モーター負荷が大きすぎる、またはモーター加速度、減速度の設定が大きすぎる、加減速時間の設定が小さすぎる;x. 0x60B2 トルク補償値または 0x2207 静的平衡トルク補償値は不合理に与えられる。	1、機械負荷伝動方式をチェックして、機能不良等の現象が発生しないことを確認してください。 2、.ロータ補償角の再検出を行ってください。 3、モーターエンコーダーの配線を確認し、配線仕様が正しいことを確認してください。 4、電流ループレギュレータのパラメーターを調整してください。 5、モーターパラメーターを正しく設定してください。 6、サーボドライバーを交換してください。 7、モーターを交換してください。 8、機械設計が適切かどうかを確認してください。 9、モーターの加減速を最適化し、加減速時間を延長してください。 10、位制御器動力学モデルを最適化し、所与値 0x60B2 トルク補償値の最適化、または、適切な 0x2207 静平衡モーメント補償値を再設定してください。
35322	サーボドライバーのブレーキ回路の異常	1、サーボドライバーのブレーキ出力ショート 2、サーボドライバーのブレーキ出力電流が大きすぎて過熱 3、サーボドライバーのブレーキ出力遮断 4、サーボドライバー内部検出回路の異常。	1、サーボドライバーのブレーキ出力配線をチェックして、配線が正しく確実であることを確認してください。 2、サーボドライバーを交換してください。
35323	ドライバー回転回路異常	1、サーボドライバー内部のスピン回路異常	1、サーボドライバーのスピンパラメーターを正しく設定してください。

		2、サーボドライバー回転パラメーターの設定が実際の回転と一致しない。	2、サーボドライバーを交換してください。
35324	制御モード設定エラー	1、サーボイナーブルの場合、コントローラーはサーボドライバーがサポートしない制御モード(各製品でサポートされている制御モードについては、オブジェクトディクショナリ 0x6502 を参照してください。)の設定が必要。	i.サーボイナーブルの前に、コントローラーは正しい制御モードを設定してください。
35325	入力不良	1、サーボドライバー動力入力電源配線不良 2、サーボドライバーサーボパラメーター「電力回路設定」は三相入力を選択されているが、実際の動力電源入力は単相 3、フロントエンドに電子トランスを使用しており、電子トランスの高調波が異常でサーボが認識できない。	1、サーボドライバーの電源入力配線を確認し、配線が確実であることを確認してください。 2、サーボパラメーターを正しく設定してください「電力回路設定」。 3、サーボドライバーの前段にフィルタを設置してください。
35326	直流母線電圧不足	1、サーボドライバーのパワー入力電源電圧が低い 2、サーボドライバー内部電圧サンプリング回路の異常 3、サーボパラメーター 0x202C 設定エラー、220 V 給電が 380 V 給電に設定されている 4、サーボドライバーの電源入力ケーブル外れ。	1、サーボドライバーの電源入力を許容範囲内に調整してください。 2、サーボドライバーを交換してください。 3、.0x202C ドライバーパラメーターを正しく設定してください。; 4、サーボドライバーの電源入力線の配線を確認してください。
35327	インバータモジュール過熱	1、モーターの負荷が大きすぎる 2、サーボドライバー内部温度サンプリング回路の異常 3、サーボドライバーの動作環境温度が許容範囲を超えてる。	1、モーターの実際の機械負荷を低減してください。 2、コントローラーの冷却条件の改善などを行い、環境温度を下げてください。 3、サーボドライバーを交換してください。
35328	エネルギーブレーキ過負荷	1、モーターが頻繁に快速停止操作を行うため、エ	1、サーボパラメーター「エネルギー制動抵抗値」と「エネルギー消費制動抵抗電

		エネルギー消費と制動エネルギーが大きすぎる 2、サーボパラメーター「エネルギー制動抵抗値」と「エネルギー消費制動抵抗電力」の設定が間違っている。	力」を正しく設定してください。2、モーター運転モードを変更し、モーターが頻繁に快速停止操作を行うことを避けてください (ex. モーターの停止時間を長くする等)。
35329	モーター持続過負荷	1、モーターの負荷が大きすぎる 2、実際の機械負荷が大きすぎる 3、機械負荷にはデッド現象がある 4、モーターロックが解除されていません 5、モーター加減速時間の設定が小さすぎます 6、モーターパラメーター設定エラー 7、サーボドライバ内部電流サンプリング回路の異常 8、サーボドライバブレーキ回路の異常 9、モーター選択エラー、電力が小さすぎる (ex. 大電力駆動器、小電力モーター搭載; 最大負荷高速運転時間)。	1、モーターの実際の機械負荷を低減してください。 2、モーター運転時の加減速時間を大きくしてください。 3、機械負荷伝動方式を検査して、機能不良等の現象が発生しないことを確認してください。 4、モーターブレーキ配線をチェックして配線の信頼性を確保してください。 5、モーターパラメーターが正しく設定されていることを確認してください (ex. モーター定格電流、モーター熱時定数)。 6、大容量モーターに変更が必要な可能性があります。 7、サーボドライバを交換してください。
35330	整流電力モジュールの過熱	1、モーターの負荷が大きすぎる 2、サーボドライバ内部温度サンプリング回路の異常 3、サーボドライバ動作環境温度が許容範囲を超えている	1、モーターの実際の機械負荷を低減してください。 2、環境温度を下げてください (ex. コントローラーの放熱条件を改善する等) 3、サーボドライバを交換してください。
35331	モーターU 相過渡過負荷	1、モーターの負荷が大きすぎる。 2、実際の機械負荷が大きすぎる。 3、機械負荷に固着現象と渋滞現象が存在する。 4、モーターブレーキが解除されていない。	1、モーターの実際の機械負荷を低減してください。 2、モーター運転時の加減速時間を大きくしてください。 3、機械負荷伝動方式を検査して、機能不良等の現象が発生しないことを確認してください。 4、ロータ補償角の再検出をしてください

		5、モーター加減速時間の設定が小さすぎる。 6、ローターの補正角度が正しく設定されていない。 7、モーターパラメーターが正しく設定されていない。 8、サーボドライバー内部電流サンプリング回路異常; 9、サーボドライバーブレーキ回路の異常 10、モーター動力線の接触不良或いは脱落。 11、モーター選択型が小さい	い。 5、モーターブレーキの接続をチェックして確実な接続を確保してください。 6、モーターパラメーターをチェックして、モーターパラメーターが正しく設定されていることを確認してください(ex.モーター定格電流、モーター高速過負荷保護閾値、モーター高速過負荷保護時間)。 7、サーボドライバーを交換してください。 8、モーターの電源線の配線が確実かどうかを確認してください。 9、大容量モーターへの変更が必要な可能性があります。
35332	モーターV 相過渡過負荷	1、モーターの負荷が大きすぎる。 2、実際の機械負荷が大きすぎる。 3、機械負荷に固着現象と渋滞現象が存在する。 4、モーターブレーキが解除されていない。 5、モーター加減速時間の設定が小さすぎる。 6、ローターの補正角度が正しく設定されていない。 7、モーターパラメーターが正しく設定されていない。 8、サーボドライバー内部電流サンプリング回路異常; 9、サーボドライバーブレーキ回路の異常 10、モーター動力線の接触不良或いは脱落。 11、モーター選択型が小さい	1、モーターの実際の機械負荷を低減してください。 2、モーター運転時の加減速時間を大きくしてください。 3、機械負荷伝動方式を検査して、機能不良等の現象が発生しないことを確認してください。 4、ロータ補償角の再検出をしてください。 5、モーターブレーキの接続をチェックして確実な接続を確保してください。 6、モーターパラメーターをチェックして、モーターパラメーターが正しく設定されていることを確認してください(ex.モーター定格電流、モーター高速過負荷保護閾値、モーター高速過負荷保護時間)。 7、サーボドライバーを交換してください。 8、モーターの電源線の配線が確実かどうかを確認してください。 9、大容量モーターへの変更が必要な可能性があります。
35333	モーターW 相過渡過負荷	1、モーターの負荷が大きすぎる。 2、実際の機械負荷が大き	1、モーターの実際の機械負荷を低減してください。 2、モーター運転時の加減速時間を大きく

		すぎる。 3、機械負荷に固着現象と渋滞現象が存在する。 4、モーターブレーキが解除されていない。 5、モーター加減速時間の設定が小さすぎる。 6、ローターの補正角度が正しく設定されていない。 7、モーターパラメーターが正しく設定されていない。 8、サーボドライバー内部電流サンプリング回路異常; 9、サーボドライバーブレーキ回路の異常 10、モーター動力線の接触不良或いは脱落。 11、モーター選択型が小さい	してください。 3、機械負荷伝動方式を検査して、機能不良等の現象が発生しないことを確認してください。 4、.ロータ補償角の再検出をしてください。 5、モーターブレーキの接続をチェックして確実な接続を確保してください。 6、モーターパラメーターをチェックして、モーターパラメーターが正しく設定されていることを確認してください(ex.モーター定格電流、モーター高速過負荷保護閾値、モーター高速過負荷保護時間)。 7、サーボドライバーを交換してください。 8、モーターの電源線の配線が確実かどうかを確認してください。 9、大容量モーターへの変更が必要な可能性があります。
35334	電源装置モジュールの通信異常	1、電源ユニットモジュールとモーターモジュール間の制御ケーブルの接続不良 2、モーターブレーキ線または外部 24 V (STO)対地短絡により電源モジュールとモーターモジュール間の制御ポートが焼損。	1、電源ユニットモジュールとモーターモジュール間の制御ケーブルの接続をチェックし、配線が確実であることを確認してください。 2、モーターブレーキ線または外部 24 V (STO)対地短絡を確認してください。3、サーボドライバーを交換してください。
35335	ハードウェア STO1 のトリガー	1、STO1 トリガーまたは配線不良。	1、STO1 の配線を確認し、配線が確実であることを確認してください。 2、STO1 ループ(非常停止スイッチ)がトリガーされていないことを確認してください。
35336	ハードウェア STO2 のトリガー	1、STO2 トリガーまたは配線不良。	1、STO2 の配線を確認し、配線が確実であることを確認してください。 2、STO2 ループ(非常停止スイッチ)がトリガーされていないことを確認してください。
35337	STO 配線異常	1、STO1/STO2 配線不良。	1、STO1/STO2 の配線を確認し、配線が確実であることを確認してください。
35338	サーボドライバー外部障害	1、他の軸で障害が発生。 2、サーボドライバー内部	1、他の軸を点検し、他の軸が故障していないことを確認してください。

		回路の異常。	2、このエラーはサーボパラメーター「故障スイッチ」を修正することによって検出を禁止することができます。 3、サーボドライバを交換してください。
35339	位置追従誤差が大きすぎる	1、モーターの機械的負荷で機能できない現象があり、モーターの動作不可 2、上位機位置目標値計画加速過高 3、サーボパラメーター 0x6065 とサーボパラメーター 0x6066 の設定が小さすぎる 4、サーボドライバ調節器パラメーターの設定が適正でないため、位置追跡性能不足 5、サーボドライバ内部の回路の異常。	1、機械負荷伝動方式をチェックして、機能不良等の現象が発生しないことを確保してください。 2、上位機位置目標値を適切に下げて加速度を設定してください。 3、サーボパラメーター 0x6065 とサーボパラメーター 0x6066 の設定値を適当に大きくしてください。 4、位置追跡性能を向上させるための調整パラメーターの最適化を行ってください。 5、サーボドライバを交換してください。
35340	位置制御オーバーフロー	1、位置の実際の値または位置の目標値が許容範囲を超えた。	1、「エンコーダー多重リングゼロクリア」コマンドを実行し、モーターの動作範囲が最大許容範囲を超えないようにしてください。 2、モーターを広範囲に運転する必要がある場合は、サーボパラメーター「位置制御スイッチ」によって無限位置制御モードを可能にしてください。
35341	速度追従誤差が大きすぎる	1、モーターの機械的負荷で機能できない現象があり、モーターの動作不可 2、サーボパラメーター 0x20A3 とサーボパラメーター 0x20A4 の設定が小さすぎる。 3、サーボドライバ・レギュレータ・パラメーターが適切に設定されていないため、スピード・トラッキング・パフォーマンスが不十分 4、サーボドライバ内部の回路が異常です。	1、機械負荷伝動方式を検査して、機能不良等の現象が発生しないことを確認してください。 2、サーボパラメーター 0x20A3 とサーボパラメーター 0x20A4 の設定値を適当に大きくしてください。 3、速度追跡のパフォーマンスを向上させるための調整パラメーターの最適化を行ってください。 4、サーボドライバを交換してください。
35342	制御サイクルパラメーター設定エラー	1、EtherCAT 通信周期, 位置制御周期, 速度制御周期	1、EtherCAT 通信周期, 位置制御周期, 速度制御周期を正しく設定してください。

		期の設定が不適切	
35343	EEPROM 書込み失敗	1、サーボドライバ内部回路異常 2、サーボドライバが破損	1、サーボドライバを交換してください。 2、サーボドライバの電源を入れ直し、サーボドライバのノイズ対策を強化してください。
35344	原点サーチ失敗	1、原点サーチパラメータの設定が不適切(オブジェクト 0x6098, 0x6099, 0x609A) 2、原点サーチ起動時、モーターはリミットスイッチトリガ状態にある 3、原点サーチ時に非 HM モードに切り替える。	1、原点サーチパラメータを正しく設定してください(オブジェクト 0x6098, 0x6099, 0x609A); 2、原点サーチ起動時、モーターがリミットスイッチトリガ状態でないことを確認してください。
35345	EtherCAT バス命令不正	1、EtherCAT 通信状態がロボットと制御のタイミング間違い。	1、上位で EtherCAT 通信状態をロボットと制御のタイミングを正しく処理するようにしてください。
35346	DriveStarter 通信異常	1、サーボドライバのデバッグケーブルが外れているか、接触不良。 2、サーボドライバのデバッグシリアル通信が妨害されている。	1、サーボドライバのデバッグケーブル配線を確認し、接続が確実であることを確認してください。 2、分離型シリアルポートデバッグケーブルを交換してください。 3、フェライトコアの追加、デバッグ用コンピュータの確実な接地、デバッグ用コンピュータとサーボドライバの電源分離など、デバッグ用ケーブルのノイズ対策を強化してください。
35347	EtherCAT バス通信異常	1、EtherCAT 通信妨害 2、EtherCAT ネットワークケーブルが接続されていない、または接触不良 3、上位機のリアルタイム性が不十分 4、上位機 EtherCAT ホームベース DC タイミングが制御とサーボドライバで一致しない 5、サーボドライバ内部の回路の異常。	1、EtherCAT の通信配線を最適化し、5 種以上のシールド線を使用し、コントローラは確実に接地するなどのノイズ対策を強化してください。 2、EtherCAT ネットワーク接続を確認し、接続が信頼できることを確認してください。 3、よりリアルタイム性の高い上位機を交換するか、EtherCAT の通信周期を延長してください。 4、サーボパラメータ 0x20D3 の設定値を適当に大きくしてください。 5、上位機 EtherCAT メインステーションの基礎 DC 同期メカニズムを修正して、上位機で RxPDO データが DC 同期信号より少なくとも 130 us 前に送信されることを確保してください。

			6、サーボドライバーを交換してください。
35348	位置ハードウェアオーバーラン	1、リミットスイッチ入力トリガー。	1、リミットスイッチの状態を確認し、リミットスイッチが起動していないことを確認してください。
35349	正方向ソフトウェアリミット	1、位置の実績値がサーボパラメーター0x2004 とサーボパラメーター0x2005 の設定閾値を超えている。	1、サーボパラメーター0x2004 とサーボパラメーター0x2005 の設定値を適当に大きくしてください。 2、サーボパラメーター0x2004 とサーボパラメーター0x2005 で規定された範囲内でモーターを運転してください。 3、この機能を使用しない場合は、サーボパラメーター「位置制御スイッチ」を使用してソフトウェアリミット検出機能を無効にしてください。
35350	負のソフトウェアリミット	1、位置の実績値がサーボパラメーター0x2004 とサーボパラメーター0x2005 の設定しきい値を超えている。	1、サーボパラメーター0x2004 とサーボパラメーター0x2005 の設定値を適当に大きくしてください。 2、サーボパラメーター0x2004 とサーボパラメーター0x2005 で規定された範囲内でモーターを運転してください。 3、この機能を使用しない場合は、サーボパラメーター「位置制御スイッチ」を使用してソフトウェアリミット検出機能を無効にしてください。
35351	アップライト位置の偏差が大きすぎる	1、ドライバーの電源が落ちた後、モーターの位置がずれた 2、バッテリー付きモーターエンコーダーの場合、バッテリーの外付け、バッテリーの電圧不足。	1、バッテリー付きモーターエンコーダーの場合、バッテリーが接続され、バッテリー電圧が正常であることを確認してください。 2、この機能を使用しない場合は、サーボパラメーター0x200E を 0 に設定して、電源 ON 位置偏差過大検出機能を禁止してください。
35352	電源制御オーバーフロー	1、バッテリー付きモーターエンコーダーの場合、バッテリーの外付け、バッテリーの電圧が不足 2、ドライブの電源がオフになる前の制御モードが無限位置制御モード、速度モード、またはトルクモードであり、位置が許容範囲を超えている。	1、.エンコーダー多回転ゼロクリアコマンドを実行した後、ドライブの電源を入れ直してください。 2、無限位置制御モードにおいて、この機能を使用しない場合、サーボパラメーター「位置制御スイッチ」を修正することによって、電源 ON 位置制御オーバーフロー検出機能を禁止することができます。
35353	エンコーダーバッテリーの電圧不足	1、エンコーダーのバッテリー未接続 or 接続不良	1、エンコーダーバッテリーの配線を確認し、配線が確実であることを確認してください。

		2、.エンコーダーバッテリーの電圧不足。	さい。 2、バッテリーを交換してください。 3、パナソニックと多摩川エンコーダーで接続されている場合は、個別のバージョンでエンコーダー多回転ゼロクリアコマンドを実行する必要があります。 4、この機能を使用しない場合は、サーボパラメーター0x2009を修正することにより、エンコーダーのバッテリー不足電圧検出機能を無効にすることができます。
35354	モーター速度超過	1、サーボドライバーレギュレータパラメーターの設定が適切でないため、速度追跡のオーバーシュートが大きい 2、エンコーダー配線不良 3、エンコーダーのデータ転送時ノイズの影響を受けている 3、.エンコーダー破損 4、サーボドライバー内部の回路の異常。	1、速度追跡性能を向上させるための調整パラメーターを最適化してください。 2、エンコーダーケーブルの接続を確認し、配線が確実であることを確認してください。 3、フェライトコアの追加、シールド付きツイストペアケーブルの使用、信頼性の高いアース接続など、エンコーダーケーブルのノイズ対策を強化してください。 4、エンコーダーを交換してください。 5、サーボドライバーを交換してください。
35355	電圧リミット位置追従誤差過大	1、モーターの機械的負荷にはデッド現象があり、モーターの運動ができない 2、上位機位置目標値計画加速過高 3、サーボパラメーター0x6065とサーボパラメーター0x6066が小さすぎます。 4、ドライバー調節器パラメーターの設定が合理的でないため、位置追跡性能が理想的ではない 5、ドライバーのパワー入力電源電圧が低い 6、ドライバー内部の回路が異常です。	1、機械負荷伝動方式を検査して、機能不良などの現象が発生しないことを確認してください。 2、上位機位置目標値を適切に下げて加速度を設定してください。 3、サーボパラメーター0x6065とサーボパラメーター0x6066の設定値を適当に大きくしてください。 4、位置追跡性能を向上させるための調整パラメーターの最適化を行ってください。 5、サーボドライバーの電源入力電圧が指定された範囲内にあることを確認してください。 6、サーボドライバーを交換してください。
35356	エンコーダー過速度故障	1、サーボドライバー調節器パラメーターの設定が適切でないため、速度追跡のオーバーシュートが大きい	1、速度追跡性能を向上させるための調整パラメーターの最適化を行ってください。 2、.エンコーダーケーブルの接続を確認し、配線が確実であることを確認してください。

		2、エンコーダー配線不良 3、エンコーダーのデータ転送時ノイズの影響を受けている。 4、エンコーダー破損 5、サーボドライバ内部回路の異常 6、サーボ側でイナーブルの場合モーター軸を回転させる外力がある。	3、フェライトコアの追加、シールド付きツイストペアケーブルの使用、信頼性の高いアース接続など、エンコーダーケーブルのノイズ対策を強化してください。 4、エンコーダーを交換してください 5、サーボドライバを交換してください。 6、モーター軸端部の機械を点検し、モーター軸が重力または機械的外力から保護されていることを確認してください。
35357	ロケーション計画の実行エラー	位置計画パラメーターの不適切設定。例えば位置目標値、計画目標減速度(0x6084)の設定が不適切。	位置計画パラメーターを正しく設定してください。
35358	多軸同期異常	サーボドライバ内部の回路異常	サーボドライバを交換してください。
35359	EtherCAT バス同期例外	1、サーボパラメーター0x20D3 設定が不適切です 2、EtherCAT プライマリ同期モードが正しく設定されていません。	1、サーボパラメーター0x20D3 を正しく設定してください。 2、EtherCAT マスターは同期モードを正しく設定してください。
35360	EEPROM バージョンの変更	新しいドライブファームウェアをバージョンアップ	サーボドライバの電源を入れ直してください。
35361	モーター過負荷警報	1、モーターの負荷が大きすぎる 2、実際の機械負荷が大きすぎる 3、機械負荷に機能不良現象がある 4、モーターブレーキが解除されていません 5、モーター加減速時間の設定が小さすぎます 6、モーターパラメーター設定エラー 7、サーボドライバ内部電流サンプリング回路の異常 8、サーボドライバのブレーキ回路が異常です。	1、モーターの実際の機械負荷を低減してください。 2、モーター運転時の加減速時間を大きくしてください。 3、機械負荷伝動方式を検査して、機能不良などの現象が発生しないことを確認してください。 4、モーターブレーキ配線をチェックして配線の信頼性を確保してください。 5、モーターパラメーターをチェックして、モーターパラメーターが正しく設定されていることを確認してください(ex.モーター定格電流、モーター熱時定数) 6、サーボドライバを交換してください。 7、大容量モーターへの変更が必要な可能性があります。
35362	回転速度制限アラーム	1、速度モードでは、速度	1、速度目標値または速度計画値を小さく

		目標値がモーターの最高回転速度を超えている。 2、位置モードでは、計画速度がモーターの最高回転速度を超えている。 3、サーボパラメーターが正しく設定されていない。	してください。 2、状況に応じてパラメーター0x607F 最大計画速度を再設定してください。 3、より高い最高回転速度のモーターへの変更が必要な可能性があります。
35363	直流母線電圧低下警報	1、サーボドライバの電源入力電圧が低い 2、.サーボドライバ内部電圧サンプリング回路の異常。	1、サーボドライバの電源入力を許容範囲内に調整してください。 2、サーボドライバを交換してください。
35364	設定制御モードはサポートされていない	1、(サーボインエーブルの場合、削除)コントローラーがドライブでサポートされていない制御モードに設定されている(各製品でサポートされている制御モードは、オブジェクトデクシヨナリ0x6060, 0x6502を参照してください。) 2、コントローラーとサーボ通信が確立した後、制御モードが指定されていない。	1、サーボインエーブルの前に、コントローラーを正しい制御モードに設定してください。
35365	再充電の有効パラメーターが変更された	再通電に有効なサーボパラメーターを修正した	サーボドライバの電源を入れ直してください。
35366	エンコーダーバッテリーの電圧不足警報	1、.エンコーダーがバッテリーに接続されていないか、接続不良。 2、.エンコーダーバッテリーの電圧不足 3、エンコーダーのバッテリーラインが正負逆または被覆破損により地絡。	1、エンコーダーバッテリーの配線を確認し、配線が確実であることを確認してください。 2、バッテリーを交換してください。 3、パナソニックまたは多摩川エンコーダーに接続されている場合は、個別のバージョンでエンコーダー多回転ゼロクリアコマンドを実行する必要があります。 4、.この機能を使用しない場合は、サーボパラメーター0x2009を修正することにより、エンコーダーのバッテリー不足電圧検出機能を無効にすることができます。
35367	サーボドライバ内部アラーム	サーボドライバ内部回路の異常	1、サーボドライバ.を交換してください。 2、弊社までご連絡ください。

35368	機械原点未設定	1、エンコーダーバッテリーの電圧低下が発生し、サーボパラメーター 0x2009.Byte3 が"エンコーダーバッテリーの低電圧故障を検出し、原点が標定されていないことを示す"に設定されている。 ii.電源投入位置の偏差が大きすぎる故障が発生し、かつユーザーは機械原点が紛失したと判断する。 3、モーターには、シングルループのアブソリュートエンコーダーがあり、ドライブで原点検索コマンドが正常に実行されない。	サーボドライバの原点復帰を実行してください。
35369	エンコーダー外部通信受信警告	1、モーターエンコーダー結線異常(ex.断線, シールド付きツイスト・ペア未使用、モーターの動力線と結合) 2、サーボドライバのアースが正しく接続されていない; 3、サーボドライバの周囲に強いノイズ源がある。	1、モーターエンコーダーの配線を確認し、配線仕様が正しいことを確認してください。 2、エンコーダーケーブル、モーターのパワーケーブルにフェライトコアを追加してください。 3、サーボドライバアースの信頼性の高い接続を実施願います。 4、サーボドライバの周囲に強力なノイズ源がないかを確認して、ある場合は対策を実施して除去してください。または、サーボドライバが周囲の強力なノイズ源と独立して電源を供給するようにしてください。 5、サーボドライバの入力電源線にフィルタを追加してください。
35370	エンコーダー外部通信送信警告	1、モーターエンコーダー結線異常(ex.断線, シールド付きツイスト・ペア未使用、モーターの動力線と結合) 2、サーボドライバのアースが正しく接続されていない; 3、サーボドライバの周囲に強いノイズ源があ	1、モーターエンコーダーの配線を確認し、配線仕様が正しいことを確認してください。 2、エンコーダーケーブル、モーターのパワーケーブルにフェライトコアを追加してください。 3、サーボドライバアースの信頼性の高い接続を実施願います。 4、サーボドライバの周囲に強力なノイズ源がないかを確認して、ある場合は対策

		る。	を実施して除去してください。または、サーボドライバーが周囲の強力なノイズ源と独立して電源を供給するようにしてください。 5、サーボドライバーの入力電源線にフィルタを追加してください。
35371	エンコーダー内部通信警告	エンコーダー内部通信警告	1、モーターエンコーダーの配線を確認し、配線仕様が正しいことを確認してください。 2、エンコーダーケーブル、モーターのパワーケーブルにフェライトコアを追加してください。 3、サーボドライバーの接地の信頼性の高い接続を実施願います。 4、サーボドライバーの周囲に強力なノイズ源がないかを確認して、ある場合は対策を実施して除去してください。または、サーボドライバーが周囲の強力なノイズ源と独立して電源を供給するようにしてください。 5、サーボドライバーの入力電源線にフィルタを追加してください。
35372	ソフトウェアリミット警告	位置実績値または位置目標値がサーボパラメーター 0x2004 およびサーボパラメーター 0x2005 設定閾値を超えている。	1、サーボパラメーター 0x2004 とサーボパラメーター 0x2005 の設定値を適当に大きくしてください。 2、サーボパラメーター 0x2004 とサーボパラメーター 0x2005 で規定された範囲内でモーターを運転してください。 3、位置目標設定値を小さくし、サーボパラメーター 0x2004 とサーボパラメーター 0x2005 で指定された範囲内に収まるようにしてください。 4、この機能を使用しない場合は、サーボパラメーター「位置制御スイッチ」によってソフトウェアリミット検出機能を無効にすることができます。
35373	AD 補正係数が無効	.サーボドライバーが AD 補正されていない。	サーボドライバーの AD 補正係数をリセットしてください。
35374	位置計画パラメーター異常警告	ロケーション計画パラメーターの設定が不適切。	位置計画パラメーターを正しく設定してください。
35375	電源位置の偏差過大警告	サーボドライバーの電源が落ちた後、モーターの位置がずれた。	1、サーボドライバーの故障復元命令を実行してください。 2、サーボドライバーの電源を入れ直すか、ソフトリセットを実行してください。
35401	サーボドライバー短絡	1、サーボドライバー	1、ケーブルの UVW 相間、UVW とアース

		UVW 出力ケーブルが短絡又は地絡 2、モーターUVW の短絡又は地絡 3、サーボドライバー内部で短絡又は地絡 4、サーボドライバーが干渉しエラーが発生	との間でショートが発生した場合はケーブルを処理または交換してください。 2、モーターの UVW 相間、UVW と接地との間でショートが発生した場合、モーターを交換してください。 3、サーボドライバーの UVW ケーブルを外しても、故障が発生する場合は、サーボドライバーを交換してください。 4、配線を規範化し、アース線の断面積を増やし、フェライトコアを追加するなどの措置を行い、設備の電磁環境を改善してください。
35402	U 相出力電流過大	1、電流ループ調節器のパラメーター設定が不適切で、電流制御振動を引き起こす 2、モーターパラメーター設定エラー 3、ドライバー内部電流サンプリング回路の異常。	1、電流ループ調節器パラメーターを調整してください。 2、モーターのパラメーターを正しく設定してください。 3、サーボドライバーを交換してください。
35403	V 相出力電流過大	1、電流ループ調節器のパラメーター設定が不適切で、電流制御振動を引き起こす 2、モーターパラメーター設定エラー 3、ドライバー内部電流サンプリング回路の異常。	1、電流ループ調節器パラメーターを調整してください。 2、モーターのパラメーターを正しく設定してください。 3、サーボドライバーを交換してください。
35404	W 相出力電流過大	1、電流ループ調節器のパラメーター設定が不適切で、電流制御振動を引き起こす 2、モーターパラメーター設定エラー 3、ドライバー内部電流サンプリング回路の異常。	1、電流ループ調節器パラメーターを調整してください。 2、モーターのパラメーターを正しく設定してください。 3、サーボドライバーを交換してください。
35405	ドライブハードウェアの過電流	1、モーター負荷が大きすぎる、またはモーター加速度、減速度の設定が大きすぎる、加減速時間の設定が小さすぎる 2、回転子補償角設置値に偏差があり、同時に回転子定位エラー検出条件を満たさない。	1、機械負荷伝動を検査して、機能不良などの現象が発生しないことを確認してください。加減速時間を大きくしてください。 2、ロータの補償角を再検出してください。 3、モーターエンコーダーの配線をチェックし、配線仕様が正しいことを確認してください。 4、電流ループ調節器パラメーターを調整

		3、エンコーダーのフィードバックが異常に変化した 4、電流ループ調節器のパラメーター設定が不合理で、電流制御振動を引き起こす; 5、モーターパラメーター設定エラー(線抵抗、線インダクタンス、逆起電力、回転子慣性モーメントなど); 6、サーボドライバ内部電流検出回路の異常、またはサーボドライバのブレーキ回路の破損、24 V 出力なし; 7、モーターブレーキの破損; 8、トルクオフセット値または静的バランス補償値の設定が不適切	してください。 5、モーターのパラメーターを正しく設定してください。 6、サーボドライバを交換してください。 7、モーターを交換してください。 8、上位コントローラの動力学モデルを最適化し、所定の値を最適化し、または実際の負荷に基づいて設定してください。
35406	ドライブ出力の地絡	1、サーボドライバの UVW 出力線の地絡 2、モーター UVW の地絡 3、サーボドライバ内部で短絡又は地絡	1、ケーブルの UVW とアースとの間で短絡が発生した場合、ケーブルを処理または交換してください。 2、モーターの UVW とアースとの間で短絡が発生した場合は、モーターを交換してください。 3、サーボドライバの出力 UVW 線を外し、それでもエラーが発生する場合はサーボドライバを交換してください。
35407	直流母線過圧	1、サーボドライバ動力入力電源の電圧が大きすぎる 2、モーターが急停止する時のエネルギー消費回生エネルギーが大きすぎる 3、エネルギー消費する回生抵抗が接続されていない、または配線ミス 4、エネルギー消費するの回生抵抗値が大きすぎる 5、サーボドライバの内部異常	1、サーボドライバのパワー入力電源を許容範囲内に調整してください。 2、モーター停止減速度を低下させ、または負荷を軽減してください。 3、エネルギー消費回生抵抗の配線を修正し、配線が正確であることを確認してください。 4、エネルギー消費回生抵抗値(抵抗値は最小許容値を下回ってはいけません)を適当に小さくして、エネルギー消費回生抵抗の効率を上げる。 5、サーボドライバを交換してください。
35408	直流母線電圧不足	1、サーボドライバ動力	1、サーボドライバのパワー入力電源を

		入力電源電圧が低すぎる 2、サーボドライバ内部電圧サンプリング回路異常 3、サーボドライバ電力回路の設定ミス、220 V 給電が 380 V 給電に設定されている 4、サーボドライバの電源入力電源線切断	正常動作範囲内に調整してください。 2、サーボドライバを交換してください。 3、サーボドライバ電力回路の設定は実際の給電電源と一致させてください。 4、サーボドライバの電源入力線の配線を確認し、正しく接続されていることを確認してください。
35409	パワーモジュール過熱	1、モーターの負荷が大きすぎる。 2、サーボドライバ内部温度サンプリング回路の異常 3、サーボドライバの動作環境温度が許容範囲を超えている。	1、モーターの実際の機械負荷を減らしてください。 2、サーボドライバを交換してください。 3、環境温度を下げてください。(ex.コンローラーの放熱条件を改善する)
35410	CPU 1 ウォッチドッグオーバー	ドライブの内部異常	サーボドライバを交換してください。
35411	CPU 2 ウォッチドッグオーバー	ドライブの内部異常	サーボドライバを交換してください。
35412	エネルギー制動抵抗過負荷	1、モーターが頻繁に快速停止操作を行うため、エネルギー消費と制動エネルギーが大きすぎる 2、エネルギー消費回生抵抗電力の設定と実際の抵抗値が一致しない。	1、モーターの運転モードを変えて、モーターが頻繁に快速停止操作を行うことを避けて、例えばモーターの停止時間を延長する。または、消費電力の大きいブレーキ抵抗器を交換してください。 2、エネルギー消費回生抵抗電力を正確に設定し、設定値はエネルギー消費回生抵抗の実際電力と一致させてください。
35413	モーター持続過負荷	1、モーターの負荷が大きすぎる 2、モーター加減速時間の設定が小さすぎる 3、モーターパラメーター設定エラー 4、ブレーキ解除動作異常 5、モーターのタイプミス、パワーが小さすぎる (ex.大電力駆動器、小電力モーターを搭載して長時間最大負荷、高速で動作する) 6、サーボドライバ内部電流サンプリング回路異常	1、機械負荷伝動方式をチェックして、機能不良などの現象が発生しないことを確認してください。 2、モーター運転時の加減速時間を大きくしてください。 3、モーターパラメーターをチェックし、モーターパラメーターが正しく設定されていることを確認してください(ex.モーター定格電流,モーター熱時定数など) 4、モーターブレーキ回路をチェックして、モーターブレーキ動作が正常であることを確認してください。 5、サーボドライバを交換してください。 6、大容量モーターへの変更が必要な可能

		常。	性があります。
35414	位置追従誤差が大きすぎる	1、モーターの負荷が大きすぎる 2、制御パラメーターが適切でない 3、ブレーキ解除動作異常 4、位置追従誤差が大きすぎる判定閾値または判定時間施設が小さすぎる。	1、機械負荷伝動を検査して、機能不良などの現象が発生しないことを確認してください。 2、制御パラメーターを最適化し、サーボの相応性能を強化してください。 3、モーターブレーキ回路をチェックして、モーターブレーキ動作が正常であることを確認してください。 4、位置追従誤差過大判定閾値又は判定時間を適切に大きくしてください。
35415	正方向ソフトウェアリミット	位置フィードバック値が(正のソフトウェアリミット値+位置決め完了しきい値)を超えている。	運動範囲は順方向ソフトウェアリミット設定値を超えないようにし、順方向ソフトウェアリミット機能が不要であれば、パラメーター位置制御スイッチで禁止してください。
35416	負のソフトウェアリミット	位置フィードバック値が(負のソフトウェアリミット値-位置決め完了しきい値)を超えている。	運動範囲は負方向ソフトストップ設定値を超えないようにし、正方向ソフトストップ機能が必要でなければ、パラメーター位置制御スイッチで禁止してください。
35417	エンコーダーデータのオーバーフロー	位置モードでは、無限位置制御が可能になっていない場合、エンコーダー多回転値が実際のエンコーダー多回転桁数を超える。	エンコーダーの多回転ゼロクリアを実行するか、無限位置制御モードを有効にするか、非位置モードで動作するようにしてください(トルクモードまたは速度モード)。
35418	CPU 1 の動作異常	1、サーボドライバファームウェアの動作異常 2、サーボドライバの内部異常	1、サーボドライバファームウェアのアップグレードを実施願います。 2、サーボドライバを交換してください。
35419	CPU 2 の動作異常	1、サーボドライバファームウェアの動作異常 2、サーボドライバの内部異常	1、サーボドライバファームウェアのアップグレードを実施願います。 2、サーボドライバを交換してください。
35420	CPU 1 メモリ例外	1、サーボドライバファームウェアの動作異常 2、サーボドライバの内部異常	1、サーボドライバファームウェアのアップグレードを実施願います。 2、サーボドライバを交換してください。
35421	CPU 2 メモリ例外	1、サーボドライバファームウェアの動作異常 2、サーボドライバの内部異常	1、サーボドライバファームウェアのアップグレードを実施願います。 2、サーボドライバを交換してください。
35422	CPU メモリ競合	1、サーボドライバファームウェアの動作異常 2、サーボドライバの内部異常	1、サーボドライバファームウェアのアップグレードを実施願います。 2、サーボドライバを交換してください。

35423	磁極位置エラー	1、モーター回転子位置補償角設定値と検出値が一致しない。 2、回転子定位故障検出感度の設定が小さすぎる。 3、静的平衡モーメント補償値の設定が実際の負荷と一致しない。 4、モーターの配線ミスによる回転子の位相角の変化 5、モーターの異常により回転子の位相角が変化した。 6、重力負荷によりサーボが瞬間的にモーターを回転させ、速度が回転子定位故障検出感度設定閾値を超えた。	1、モーター回転子位置補償角を再検出し、正確に設定してください。 2、回転子位置決め故障検出感度設定値を適切に大きくしてください。 3、実際の負荷に応じて静的平衡トルク補償値を正確に設定してください。 4、正しく結線し、モーター回転子位置補償角を再検出してください。 5、モーターを交換してください。 6、実際の負荷に基づいて、静的平衡モーメント補償値を正確に設定してください。
35424	エンコーダーデータ異常	1、エンコーダー自体のデータに異常が発生 2、エンコーダー線の配線順序が間違っているか、接触不良。 3.ノイズによるエンコーダーデータ異常。	1、配線を確認して、間違っている場合は配線順序を修正したり、配線を補強したりしてください。 2、モーターまたはエンコーダーを交換してください。 3、配線を規格化し、アース線の断面積を増やし、フェライトコア追加するなどの措置を行い、設備の電磁環境を改善してください。
35425	エンコーダー通信異常	1、エンコーダー自体のデータに異常が発生 2、エンコーダー線の配線順序が間違っているか、接触不良。 3.ノイズによるエンコーダーデータ異常。	1、配線を確認して、間違っている場合は配線順序を修正したり、配線を補強したりしてください。 2、モーターまたはエンコーダーを交換してください。 3、配線を規格化し、アース線の断面積を増やし、フェライトコア追加するなどの措置を行い、設備の電磁環境を改善してください。
35426	エンコーダー通信タイムアウト	1、エンコーダー自体のデータに異常が発生 2、エンコーダー線の配線順序が間違っているか、接触不良。 3.ノイズによるエンコーダーデータ異常。	1、配線を確認して、間違っている場合は配線順序を修正したり、配線を補強したりしてください。 2、モーターまたはエンコーダーを交換してください。 3、配線を規格化し、アース線の断面積を増やし、フェライトコア追加するなどの措置を行い、設備の電磁環境を改善してくだ

			さい。
35427	エンコーダー内部例外1	エンコーダー内部状態異常	エンコーダーゼロクリア後のソフトリセットまたはドライブの再起動を行ってください。
35428	サーボドライバ他軸異常	1、他の軸が故障 2、サーボドライバ内部回路の異常	1、他の軸をチェックし、エラーが発生した軸をリセットし、他の軸で故障がないことを確認してください。この故障はパラメーター0x2094 で非検出が可能です。 2、サーボドライバを交換してください。
35429	制御エンコーダーの速度超過	1、エンコーダー位置フィードバック値の位置サンプリング周期の変化量が大きすぎて、モーターの最高回転速度の 1.3 倍を超えた。 2、エンコーダー異常 3、ノイズによるエンコーダーデータ異常。	1、モーターパラメーターと制御パラメーターを最適化し、モーター最高回転速度の設定値は通常、実際のモーター最高回転速度より大きく設定してください。 2、エンコーダー設定とエンコーダー結線の確認をしてください。 3、配線を規格化し、アース線の断面積を増やし、フェライトコア追加などの措置を行い、設備の電磁環境を改善してください。
35430	サーボドライバ継続過負荷	1、モーター負荷が大きすぎる、または加減速時間の設定が小さすぎる 2、実際の機械の負荷が大きすぎる、または機能不良現象がある; 3、モーターブレーキが解除されていません 4、モーターまたはモーターブレーキ異常 5、サーボドライバの内部異常	1、モーターの実際の機械負荷を減らし、モーターの加減速時間を大きくしてください。 2、機械負荷伝動方式をチェックして、機能不良などの現象が発生しないことを確認してください。 3、モーターブレーキ接続線をチェックして接続線の信頼性を確保してください。 4、モーターを交換してください 5、サーボドライバを交換してください。
35431	サーボドライバ出力欠相	1、サーボドライバU, V, W の出力に断線や配線不良 2、モーターの抵抗が大きすぎる 3、サーボドライバ内部電流サンプリング回路異常。	1、モーターU、V、W の結線をチェックし、結線が確実であることを確認してください。 2、モーターを交換するか、ドライバの出力不足検出機能を無効にしてください。 3、サーボドライバを交換してください。
35432	モーター失速	1、サーボドライバ調節器パラメーターの設定が不適切。速度追跡のオーバーシュートが大。 2、電磁ノイズ妨害による	1、調整器パラメーターを最適化してください。 2、配線を規格化し、アース線の断面積を増やし、フェライトコア追加などの措置を行い、設備の電磁環境を改善してくだ

		エンコーダーデータの異常 3、エンコーダー破損によるエンコーダーデータの異常 4、サーボドライバ内部回路異常	さい。 3、モーターまたはエンコーダーを交換してください 4、サーボドライバを交換してください。
35433	電流追従エラー	1、サーボドライバ調節器パラメーターの設定が不適切。速度追跡のオーバーシュートが大。 2、電磁ノイズ妨害によるエンコーダーデータの異常 3、エンコーダー破損によるエンコーダーデータの異常 4、サーボドライバ内部回路異常	1、調整器パラメーターを最適化してください。 2、配線を規範化し、アース線の断面積を増やし、フェライトコア追加するなどの措置を行い、設備の電磁環境を改善してください。 3、モーターまたはエンコーダーを交換してください 4、サーボドライバを交換してください。
35434	位置目標値異常	1、CSP モードでは、サーボイネーブルの瞬間に、位置目標値と位置実績値の差が位置追従誤差を超えて過大に閾値を設定。 2、CSP モードでは、モーター運転中、目標軌跡加速度がパラメーター最大加速度設定閾値を超え、かつ位置目標値と位置実績値の差が位置追従誤差過大設定閾値を超える。	1、位置目標値と位置実績値に異常がないことを確認し、相違値が位置追従誤差が超過しないように閾値を設定してください。 2、位置目標値に異常な変異がないことを確認し、最大加速度設定閾値または位置追従誤差過大設定閾値を適宜大きくしてください。
35435	エンコーダー電源オーバーフロー	サーボドライバ電源投入時のフィードバック位置の値が、エンコーダーで許可されている最大範囲を超えた。	エンコーダーがゼロクリアされたら、ドライブをソフトリセットまたは再起動してください。
35436	位置目標値オーバーフロー	位置モードで無限位置制御が禁止されている場合に位置目標値が許容最大範囲を超えた。	エンコーダーの多回転ゼロクリアを実行するか、無限位置制御モードを有効にするか、非位置モードで動作してください(トルクモードまたは速度モード)。
35437	モーターブレーキ異常	1、モーターブレーキ自体に異常が発生し、正常に制動できない。	1.モーターを交換してください 2.プロセスロジック制御を最適化し、高速運転時の突然のサーボ OFF を避ける

		2、モーターが高速運転時に突然サーボ OFF し、制動時間が長すぎる 3、モーターブレーキ時間の設定値が実際の動作時間より小さい。 4、回転子定位故障検出感度の設定が小さすぎる。	3.モーターブレーキ時間の設定値を実際ブレーキ動作時間より大きくしてください。 4、回転子定位故障検出感度設定値を適当に大きくしてください。
35438	制御電圧不足	1、24 V 制御電源異常 2、24 V 制御電源の配線ミスまたは接触不良 3、24 V 制御電源の過負荷 4、サーボドライバ内部回路の異常。	1、.24 V 制御電源装置を交換してください。 2、.24 V 制御電源の配線をチェックし、配線が確実であることを確認してください。 3、.24 V 制御電源の負荷をチェックして、24 V 制御電源の容量がすべての運転モードの負荷消耗を満足できることを確認してください。 4、サーボドライバを交換してください。
35439	STO 1 トリガ	STO 1 がトリガーまたは配線不良	STO の配線を確認し、配線が確実でトリガされていないことを確認してください。
35440	STO 2 トリガー	STO 2 のトリガーまたは配線不良。	STO の配線を確認し、配線が確実でトリガされていないことを確認してください。
35441	正方向ハードリミットスイッチトリガー	機械的リミットまで実行されハードウェアリミットがトリガーされた。	逆方向に運転して機械リミット内まで移動させ回復してください。移動予定位置がハードウェアの制限を超えないように注意してください。
35442	負のハードリミットスイッチトリガー	機械的リミットまで実行されハードウェアリミットがトリガーされた。	逆方向に運転して機械リミット内まで移動させ回復してください。移動予定位置がハードウェアの制限を超えないように注意してください。
35443	モーター速度超過	1、モーターの実際速度フィードバック値がモーターの最高回転速度の 1.1 倍を超えた。 2、エンコーダー異常。	1、モーターパラメーターと制御パラメーターを最適化し、モーター最高回転速度の設定値が適正かどうかを確認してください。 2、エンコーダー設定とエンコーダー結線を確認してください。
35444	緊急停止入力スイッチトリガー	緊急停止入力スイッチが起動した、または配線不良。	非常停止入力スイッチの配線を確認して、配線が確実で、トリガされていないことを確認してください。
35445	トルク監視飽和故障	1、モーターの負荷が大きすぎて、トルク監視警告閾値を超えた。 2、トルク飽和監視閾値の設定が小さすぎる。	1、モーターの実際の機械負荷を減少し、モーターの加減速時間を大きくしてください。 2、トルク飽和監視閾値設定値を増加し、0 時に設定すると、故障を検出しません。
35446	速度追従誤差が大き	1、モーターの負荷が大き	1、モーターの実際の機械負荷を減らし、

	ぎる	すぎる 2、制御パラメーターが適切でない 3、ブレーキ解除動作異常 4、速度追従誤差が大きすぎる。判定閾値または判定時間が小さすぎる。	機械の機能不良現象がないことを確認してください。 2、制御パラメーターを最適化し、サーボの相応性能を強化してください。 3、モーターブレーキ回路をチェックして、モーターロック動作が正常であることを確認してください。 4.速度追従誤差過大判定閾値又は判定時間を適宜大きくしてください。
35447	サーボドライバのショート 2	1、サーボドライバの UVW 出力ケーブルがショート又は地絡が発生。 2、モーター UVW の短絡又は地絡が発生。 3、サーボドライバの内部でショート又は地絡が発生。 4、サーボドライバが干渉してエラーが発生。	1、ケーブルの UVW 相間、UVW とアース間に短絡が発生している場合はケーブルを処理または交換してください。 2、モーターの UVW 相間、UVW と接地間にショートが発生している場合はモーターを交換してください。 3、ドライブの UVW ケーブルを外してもエラーが発生する場合は、サーボドライバを交換してください。 4、配線を規範化し、アース線の断面積を増やし、フェライトコア追加するなどの措置を行い、設備の電磁環境を改善してください。
35448	原点サーチ失敗	1、原点サーチパラメーターの設定が正しくない 2、原点を探して起動する時、モーターがすでにリミットスイッチトリガ状態にある。 3、原点サーチ中に非 HM モードに切り替えた。	1、原点サーチパラメーターを正しく設定してください。 2、原点を探して起動する時、モーターがリミットスイッチトリガ状態でないことを確認してください。 3、原点探索中、制御モードの切り替えを避けてください。
35449	EtherCAT 過程でのデータエラー	PDO 設定値がオブジェクトの許容範囲外です。	PDO 設定値をオブジェクトの許容範囲内に設定してください。
35450	EtherCAT バス命令が不正	機械と制御の EtherCAT 通信状態のタイミングが違っている。	上位機が EtherCAT 通信状態を機械と制御器でタイミングを正しく整合してください。
35451	EtherCAT 通信周期エラー	1、EtherCAT 通信周期がサーボ制御周期より小さい 2、EtherCAT 通信期間設定は 250 us の 2 の整数乗でない。	1、EtherCAT 通信周期或いはサーボ制御周期を調整し、通信周期をサーボ制御周期より大きくしてください。 2、EtherCAT 通信期間は、250 us の 2 の整数乗に設定してください。
35452	ロケーション計画の実行エラー	1、EtherCAT 制御の下で、PP モードを実行する時、位置目標値のキャッ	1、EtherCAT メインステーションの制御プロセスを最適化し、位置目標値キャッシュの数量を減らしてください。位置目標値キ

		シュオーバー。 2、サーボドライバー内部異常。	ヤッシュは通常 4 を超えないでください。 2、サーボドライバーを交換してください。
35453	EtherCAT 不正な同期モード	1、EtherCAT 通信 DC モードの設定エラー。 2、EtherCAT 通信 DC モードがアクティブになっていない。	1、EtherCAT 通信 DC モードを正しく設定してください。 2、EtherCAT 通信 DC モードをアクティブにしてください。
35454	位置目標値が設定範囲外	モジュール無限位置制御モード又は無限位置制御禁止モードの場合、位置目標値が許容設定範囲を超えている。	位置目標値は、位置範囲下限値と位置範囲上限値の間に設定するか、通常の無限位置モードを使用してください。
35455	モーターU 相過渡過負荷	1、モーターの負荷が大きすぎる 2、モーター加減速時間の設定が小さすぎる 3、モーターパラメーター設定エラー 4、ブレーキ解除動作異常 5、モーターのタイプミス、パワーが小さすぎる (ex. 大電力サーボドライバー、小電力モーターを搭載して長時間にフルロードして高速運転する) 6、サーボドライバー内部電流サンプリング回路の異常 7、モーター高速過負荷保護閾値と保護時間の設定が小さい。	1、モーターの実際の機械負荷を減らし、機械の機能不良現象がないことを確認してください。 2、モーター運転時の加減速時間を大きくしてください。 3、モーターパラメーターをチェックし、モーターパラメーターが正しく設定されていることを確認してください(ex.モーター定格電流,モーター熱時定数など) 4、モーターブレーキ回路をチェックして、モーターブレーキ動作が正常であることを確認してください。 5、サーボドライバーを交換してください。 6、モーター高速過負荷保護閾値と保護時間を適切に増やしてください。 7、大容量モーターへの変更が必要な可能性があります。
35456	モーターV 相過渡過負荷	1、モーターの負荷が大きすぎる 2、モーター加減速時間の設定が小さすぎる 3、モーターパラメーター設定エラー 4、ブレーキ解除動作異常 5、モーターのタイプミス、パワーが小さすぎる (ex. 大電力サーボドライバー、小電力モーターを搭載して長時間にフルロードして高速運転する)	1、モーターの実際の機械負荷を減らし、機械の機能不良現象がないことを確認してください。 2、モーター運転時の加減速時間を大きくしてください。 3、モーターパラメーターをチェックし、モーターパラメーターが正しく設定されていることを確認してください(ex.モーター定格電流,モーター熱時定数など) 4、モーターブレーキ回路をチェックして、モーターブレーキ動作が正常であることを確認してください。 5、サーボドライバーを交換してください。

		6、サーボドライバー内部電流サンプリング回路の異常 7、モーター高速過負荷保護閾値と保護時間の設定が小さい。	い。 6、モーター高速過負荷保護閾値と保護時間を適切に増やしてください。 7、大容量モーターへの変更が必要な可能性があります。
35457	モーターW 相過渡過負荷	1、モーターの負荷が大きすぎる 2、モーター加減速時間の設定が小さすぎる 3、モーターパラメーター設定エラー 4、ブレーキ解除動作異常 5、モーターのタイプミス、パワーが小さすぎる (ex.大電力サーボドライバー、小電力モーターを搭載して長時間にフルロードして高速運転する) 6、サーボドライバー内部電流サンプリング回路の異常 7、モーター高速過負荷保護閾値と保護時間の設定が小さい。	1、モーターの実際の機械負荷を減らし、機械の機能不良現象がないことを確認してください。 2、モーター運転時の加減速時間を大きくしてください。 3、モーターパラメーターをチェックし、モーターパラメーターが正しく設定されていることを確認してください(ex.モーター定格電流,モーター熱時定数など) 4、モーターブレーキ回路をチェックして、モーターブレーキ動作が正常であることを確認してください。 5、サーボドライバーを交換してください。 6、モーター高速過負荷保護閾値と保護時間を適切に増やしてください。 7、大容量モーターへの変更が必要な可能性があります。
35458	ダイナミックブレーキ過負荷	モーターが動作する時、隣接2次動態制動停止間隔時間が短すぎる。	モーターの作動中に動的ブレーキが停止する場合は最低 360*実速度 2/定格速度 2 (s) の間隔を空けてください。
35459	ドライブ内部異常	ドライブの内部異常。	サーボドライバーを交換してください。
35460	リミットスイッチ異常	リミットスイッチが作動しないか、配線不良。	リミットスイッチの配線を確認し、配線が確実にトリガされていないことを確認してください。
35461	EtherCAT バス通信異常	1、EtherCAT 通信がノイズの影響を受けている 2、EtherCAT ネットワークケーブルの切断または接触不良 3、アップセットのリアルタイム性が足りない 4、上位マシン EtherCAT マスターの基礎となる DC 同期制御機とサーボドライバー要求が一致しない 5、サーボドライバーの内	1、配線を規範化し、アース線の断面積を増やし、フェライトコア追加するなどの措置を行い、設備の電磁環境を改善してください。 2、EtherCAT ネットワークケーブルの接続を確認し、接続の信頼性を確保してください。 3、よりリアルタイム性の高い上位機への変更、EtherCAT 通信周期の延長、またはタイムアウト検出感度設定値を適切に大きくしてください。 4、上位機 EtherCAT マスターステーション

		部異常。	の基礎 DC 同期メカニズムを修正して、上位機 SM 2 イベントが DC 同期信号より少なくとも 125 us 前であることを確保してください。 5、サーボドライバーを交換してください。
35462	インタフェースエンコーダーの解像度変更	インタフェースエンコーダーの解像度が変更された。	サーボドライバーの電源を入れ直すか、ソフトリセットしてください。
35463	エンコーダー過熱	1、エンコーダーの実際の温度加熱。 2、エンコーダー異常。	1、エンコーダーの実作動環境温度を許容範囲内に下げてください。 2、モーターまたはエンコーダーを交換してください。
35464	エンコーダーバッテリー電圧低下	1、エンコーダーのバッテリー電圧低下 2、エンコーダーバッテリーの結線接触不良。	1、電池の接続線をチェックして、電池の接続線が正確でしっかりしていることを確認してください。 2、エンコーダーバッテリーを交換してください
35465	制御モード設定エラー	サーボ ON の場合、制御モードは NM、VL、IP などサーボドライバーがサポートしていないモードに設定されます。 EtherCAT 制御モードの場合、制御モードは PV または PT に設定されます。	サーボ ON 時に、サーボドライバーがサポートする制御モードに設定してください。
35466	電源 ON 時位置偏差過大	サーボドライバーの電源投入時に、前回のパワーダウン保存位置と一致せず、設定したしきい値を超えた。	機械の位置が変わったかどうかを確認してください。機械の零点に異常がないことを確認した後に復旧してください。
35467	エンコーダー加速度異常故障	1、エンコーダー自体のデータに異常が発生。 2、エンコーダーの配線順序が間違っているか、接触不良。 3、ノイズによるエンコーダーデータ異常。	1、配線を確認して、間違っている場合は配線順序を修正したり、配線を補強したりしてください。 2、モーターまたはエンコーダーを交換してください。 3、配線を規範化し、アース線の断面積を増やし、フェライトコア追加などの措置を行い、設備の電磁環境を改善してください。
35468	モーター遮断	1、機械負荷に機能不良或いはストール現象が存在 2、モーターブレーキが解除されていない。	1、機械負荷伝動をチェックして機能不良などの現象がないことを確認してください。 2、ブレーキ回路をチェックしてモーターブレーキが正常に開放されることを確認し

			てください。
35469	書き込み EEPROM データ異常	サーボドライバ内部異常	サーボドライバを交換してください。
35470	読み取り EEPROM データ異常	サーボドライバ内部異常	サーボドライバを交換してください。
35471	ロック制御回路の異常	1、モーターロック配線の短絡または接触不良 2、モーターロック内部で短絡または接触不良が発生 3、サーボドライバ内部異常。	1、サーボドライバのブレーキ出力配線をチェックし、配線が正しいことを確認してください。 2、モーターを交換してください 3、サーボドライバを交換してください。
35472	CPU 1 の過負荷	1、サーボドライバがノイズの影響を受けた。 2、デバッグソフトウェアのデータ収集が大きすぎる。 3、サーボドライバ内部回路の異常。	1、配線を規格化し、アース線の断面積を増やし、フェライトコア追加などの措置を行い、設備の電磁環境を改善してください。 2、一部のデバッグソフトウェアのデータ収集チャンネルを閉じてください。 3、サーボドライバを交換してください。
35473	CPU 2 オーバーロード	1、サーボドライバがノイズの影響を受けた。 2、デバッグソフトウェアのデータ収集が大きすぎる。 3、サーボドライバ内部回路の異常。	1、配線を規格化し、アース線の断面積を増やし、フェライトコア追加などの措置を行い、設備の電磁環境を改善してください。 2、一部のデバッグソフトウェアのデータ収集チャンネルを閉じてください。 3、サーボドライバを交換してください。
35474	CPU 1 ハンドシェークに失敗	1、サーボドライバファームウェアの動作異常 2、サーボドライバの内部異常	1、サーボドライバファームウェアをアップグレードしてください。 2、サーボドライバを交換してください。
35475	DriveMaster 通信タイムアウト	1、サーボドライバデバッグケーブルが外れている又は接触不良 2、サーボドライバのデバッグシリアル通信が妨害されている。	1、サーボドライバのデバッグケーブル配線を確認し、接続が確実であることを確認してください。 2、隔離型シリアルポートを使用してケーブルを調整し、配線を規格化し、アース線の断面積を増やし、フェライトコア追加などの措置を行い、設備の電磁環境を改善してください。
35476	ESC 構成 EEPROM 異常	サーボドライバ内部異常	サーボドライバを交換してください。
35477	ESC 内部アクセスエラー	サーボドライバ内部異常	サーボドライバを交換してください。

35478	サーボの準備ができていない	1、サーボ ON 時、サーボドライバーが実際モーターで且つ仮想エンコーダーモードである。 2、サーボ ON 時、エンコーダー通信がオフ状態 3、サーボ ON 時のモーター回転速度は 30 rpm 以上 4、サーボ ON 時に STO 状態が解除されていない 5、サーボ ON 時に、直流母線電圧が低すぎて、充電リレーが吸着しない 6、サーボ ON 時、ダイナミックブレーキ状態が解除されていない。 7、サーボドライバー内部異常	1、サーボドライバーのモーターモードをチェックし、サーボ ON 時にサーボドライバーが実際のモーターと実際のエンコーダーモードにあることを確認してください。 2、エンコーダー通信状態をチェックし、サーボ ON 時のエンコーダー通信状態が正常であることを確認してください。 3、モーターの運転状態をチェックし、サーボ ON 時にモーターが静止していることを確認してください。 4、STO 状態をチェックし、サーボ ON 時に STO 状態が解除されていることを確認してください。 5、直流母線電圧状態をチェックして、サーボ ON 時に直流母線電圧が使用可能閾値を満たし、充電リレーが吸着することを確認してください。 6、ダイナミックブレーキ状態をチェックし、サーボ ON 時にダイナミックブレーキ状態が解除されたことを確認してください。 7、サーボドライバーを交換してください。
35479	CPU 2 のハンドシェイクに失敗	1、サーボドライバーファームウェアの動作異常 2、サーボドライバーの内部異常	1、サーボドライバーファームウェアをアップグレードしてください。 2、サーボドライバーを交換してください。
35480	CPU 1 のメインタスクタイムアウト	1、サーボドライバーの動作がノイズの影響を受けた。 2、デバッグソフトウェアのデータ収集が大きすぎる 3、サーボドライバー内部回路異常。	1、配線を規範化し、アース線の断面積を増やし、フェライトコア追加するなどの措置を行い、設備の電磁環境を改善してください。 2、一部のデバッグソフトウェアデータ収集チャンネルを閉じてください。
35481	DC バス充電リレーの異常	サーボドライバー内部の充電リレー異常	サーボドライバーを交換してください。
35482	CPU 内部エラー	1、サーボドライバーファームウェアの動作異常 2、サーボドライバー内部異常。	1、サーボドライバーファームウェアをアップグレードしてください。 2、サーボドライバーを交換してください。
35483	位置実績値オーバーフロー	位置モードで無限位置制御が禁止されている場合、位置の実際の値が許容される最大範囲を超え	エンコーダーの多回転ゼロクリアを実行するか、無限位置制御モードを有効にするか、非位置モードで動作するようにしてください(トルクモードまたは速度モード)。

		た。	
35484	エンコーダー内部例外 2	エンコーダー内部状態異常	エンコーダーをゼロクリアして、サーボドライバをソフトリセットまたは再起動してください。
35485	エンコーダー内部異常 3	エンコーダー内部状態異常	エンコーダーをゼロクリアして、サーボドライバをソフトリセットまたは再起動してください。
35486	第 2 位置の追従誤差が大きすぎる	1、モーターの負荷が大きすぎる 2、制御パラメーターが適切でない 3、ブレーキ解除動作異常 4、第 2 位置追従誤差過大判定閾値又は判定時間施設が過小	1.モーターの実際の機械負荷を減少し、機械の機能不良現象がないことを確認してください。 2.制御パラメーターを最適化し、サーボの相応性能を強化してください。 3.モーターブレーキ回路をチェックして、モーターブレーキ動作が正常であることを確認してください。 4.第 2 位置追従誤差過大判定閾値又は判定時間を適宜大きくしてください。
35487	STO 配線異常	STO 1 または STO 2 がトリガーされた。又は配線不良。	STO の配線を確認し、配線が確実でトリガされていないことを確認してください。
35488	第 2 速度追従誤差過大	1.モーターの負荷が大きすぎる 2.制御パラメーターが適切でない 3.ロック解除動作異常 4.第 2 速度追従誤差の判定閾値が過大又は判定時間設定が過小	1、モーターの実際の機械負荷を減らし、機械の機能不良現象がないことを確認してください。 2、制御パラメーターを最適化し、サーボの相応性能を強化してください。 3、モーターブレーキ回路をチェックして、モーターブレーキ動作が正常であることを確認してください。 4、第 2 位置追従誤差過大判定閾値又は判定時間を適宜大きくしてください。
35489	主電源入力異常	1、サーボドライバ動力入力電源配線不良 2、サーボドライバ電力回路設定が三相入力選択で、実際の動力電源入力は単相。 3、先端に電子変圧器を使用し、電子変圧器の高調波異常。	1、サーボドライバの電源入力配線を確認し、配線が確実であることを確認してください。 2、サーボドライバ電力回路を正しく設定し、設定値は実際の動力電源入力と一致させてください。 3、変圧器マニュアルに従って配線し、必要に応じてサーボドライバの先端にフィルタを取り付けてください。
35490	モーターブレーキ断線	1、モーターブレーキが接続されていない、または接触不良 2、モーターブレーキ異常 3、サーボドライバ内部	1、モーターブレーキ配線をチェックして処理し、配線が正確でしっかりしていることを確認してください。 2、モーターを交換してください 3、サーボドライバを交換してください。

		異常	
35491	コプロセッサ通信異常	1、エンコーダーの配線順序が正しくない、または接触不良 2、ノイズによるエンコーダーデータ異常。	1、配線手順を修正したり、配線を補強したりしてください。 2、配線を規範化し、アース線の断面積を増やし、フェライトコア追加するなどの措置を行い、設備の電磁環境を改善してください。
35492	エンコーダーAB信号変化異常	1、エンコーダーの配線順序が正しくない、または接触不良 2、ノイズによるエンコーダーデータ異常。	1、配線手順を修正したり、配線を補強したりしてください。 2、配線を規範化し、アース線の断面積を増やし、フェライトコア追加するなどの措置を行い、設備の電磁環境を改善してください。
35493	整流モジュール過熱	1、サーボドライバー内部温度サンプリング回路異常 2、サーボドライバーの動作環境温度が許容範囲を超えている。	1、環境温度を下げてください。(ex.コンローラーの放熱条件を改善する) 2、サーボドライバーを交換してください。
35494	放熱器過熱	1、サーボドライバー内部温度サンプリング回路異常 2、サーボドライバーの動作環境温度が許容範囲を超えている。	1、環境温度を下げてください。(ex.コンローラーの放熱条件を改善する) 2、サーボドライバーを交換してください。
35495	モーター過熱	1、モーターの負荷が大きすぎる 2、モーターの運転環境温度が高すぎて、許容範囲超過 3、モーター過熱保護熱偶電抵抗値の設定ミス 4、モーター温度センサー異常 5、サーボドライバー内部異常	1、モーターの実際の機械負荷を減らし、機械の機能不良現象がないことを確認してください。 2、モーターの放熱を強化し、運行環境温度を許容範囲内に確保してください。 3、モーター過熱保護熱電対抵抗の設定値を正しく設定してください。 4、モーターを交換してください 5、サーボドライバーを交換してください。
35496	インクリメンタルエンコーダーZ信号異常	1、エンコーダー自体のデータ異常 2、エンコーダーの配線順序間違い、接触不良 3、ノイズによるエンコーダーデータ異常。	1、モーターまたはエンコーダーを交換してください 2、配線手順を修正したり、配線を補強したりしてください。 3、配線を規範化し、アース線の断面積を増やし、フェライトコア追加するなどの措置を行い、設備の電磁環境を改善してください。
35497	エネルギー消費ブレー	サーボパラメーター設定	エネルギー消費制動選択のサーボパラメー

	キ回路異常	のエネルギー消費制動選択がエネルギー消費制動抵抗の実際の結線と不一致	ターを正確に設定し、エネルギー消費制動抵抗の実際の接線と一致させてください。
35498	CPU 過熱	1、サーボドライバ内部温度サンプリング回路異常 2、サーボドライバの動作環境温度が許容範囲を超えている。	1、環境温度を下げてください。(ex.コンローラーの放熱条件を改善する) 2、サーボドライバを交換してください。
35499	主電源電源オフ	1、サーボドライバのサーボ ON 状態で、動力電源の停電が発生 2、ノイズによる主電源の停電検出異常 3、サーボパラメーターメインパワーオフ検出時間が過小	1、サーボドライバの動力電源と配線をチェックして、電源が正常で、接続が信頼できることを確認してください。 2、配線を規範化し、アース線の断面積を増やし、フェライトコア追加などの措置を行い、設備の電磁環境を改善してください。 3、サーボパラメーター主電源オフ検出時間設定値を適宜大きくしてください。
35500	STO1 回路診断の異常	1、STO1 トリガまたは配線不良 2、サーボドライバ内部異常。	1、STO の配線を確認し、配線が確実にトリガされていないことを確認してください。 2、サーボドライバを交換してください。
35501	STO2 回路診断の異常	1、STO2 トリガまたは配線不良 2、サーボドライバ内部異常。	1、STO の配線を確認し、配線が確実にトリガされていないことを確認してください。 2、サーボドライバを交換してください。
35502	ホール信号異常	1、ホールセンサ自体の信号に異常が発生 2、ホールセンサの配線順序の誤りまたは接触不良 3、ノイズによるホール信号の異常。	1、モーターやホールセンサを交換するか、ホール信号の検出を禁止してください。 2、配線順序を修正したり、配線の補強をしてください。 3、配線を規範化し、アース線の断面積を増やし、フェライトコア追加などの措置を行い、設備の電磁環境を改善してください。
35503	エンコーダーAB信号の位相異常	1、ホールセンサまたはABエンコーダー自体の信号異常 2、ホールセンサまたはABエンコーダーの配線順序の誤りまたは接触不良	1、モーターまたはホールセンサとエンコーダーを交換してください。 2、配線順序を修正したり、配線の補強をしてください。 3、配線を規範化し、アース線の断面積を増やし、フェライトコア追加などの措置

		3、ノイズ干渉によるホールまたは AB エンコーダ信号の異常。	置を行い、設備の電磁環境を改善してください。
35504	ドライブ内部の異常 2	ドライブの内部が異常	ドライブの内部が異常です。

17.1.3 4XXXX

エラーコード	エラー内容	原因	修復方法
41410	力の限界	直交空間観測力 Fx、Fy、Fz 超限	力制御の前置設置とセンサーハードウェアの状態をチェックして、設定のプログラムを編集して、総合外力を減少する。
41411	モーメントオーバー	直交空間観測モーメント Mx、My、Mz 超限	力制御の前置設置とセンサーハードウェアの状態をチェックして、設定のプログラムを編集して、総合外力を減少する。
41412	位置超過	軸空間位置の超過	現在の位置をリミット内に移動
41413	位置超過	直交空間の位置超過	現在の位置をリミット内に移動
41414	ライン速度超過	バグが修正されるまで運動を続けられない	速度を低下させ、より良い経過点を選択し、加速度を増加させ、特異点を避け、動的属性を変更するなど
41415	速度超過	バグが修正されるまで運動を続けられない	速度を低下させ、より良い経過点を選択し、加速度を増加させ、特異点を避け、動的属性を変更するなど
41416	特異点問題	特異位置に到達または到達する直前	関節空間の少しの動きが特異位置から離れる、手動モード単軸 jog、自動モード使用 MoveAbsJ
41417	コマンド実行エラー	制限を超えてコマンドを続行できません	ロボットの関節位置をリミット内に移動させる
41418	モーメントオーバー	力制御監視を触発し、設定または測定した外力が現在のロボットの安全制限を超える	力制御パラメータの設定をチェックしてプログラムを再編集し、ロボットへのトルク指令を減らす
41419	力制御の初期化に失敗しました	TCP 長さが超過しているか、サポートされていない座標系を入力しました	実際のツールの長さを確認してください。フォースの基準座標系の定義については、製品マニュアルを参照してください。
41420	力制御のアクティブ化に失敗しました	力制御機能が初期化されていないか、不正なパラメータが設定されています。	1.現在のツールの設定が実際と合っているか、ツールの重心位置が妥当か、 2.モニタリングウィンドウを表示し、ロボットの座標系、姿勢が実際と合っているか、 3.現在のモデル、RD パラメータが実際と合っているか、 4.機械のゼロ位置に戻してみて、ドラッグする前にセンサーがゼロをクリアしているか、 6.ドラッグを開始したときにロボットが外

			力を受けていないか、 7. その他の詳細については、「ドラッグ&ドロップのトラブルシューティングマニュアル」を参照してください。
41421	強制一時停止の失敗	ロボットは現在、力制御運転状態ではありません	このコマンドは、力制御がオンになっている場合にのみ有効です。
41422	再起動に失敗しました	ロボットは現在、力制御停止状態ではありません	まず力制御を一時停止または停止し、再び起動してください
41423	センサーの使用方法的設定に失敗しました	力制御が初期化されていません	前置設定を確認し、力制御を再初期化してください
41424	センサーのキャリブレーションに失敗しました	現在のセンサーの使用方式は動的補償であり、センサーの誤差を除去することはできません。	センサーエラーのクリア機能をご使用の場合は、センサーの使用方法をゼロクリアに変更してください
41425	正弦波サーチの動作に失敗しました	非インピーダンス制御モードでは、サーチ動作を一時停止または実行状態では設定できません	現在インピーダンスモードであり、サーチ動作が停止していることを確認してください
41426	検索キャンペーンが失敗した場合のリサーザの設定	非インピーダンス制御モードでは、サーチ動作を一時停止または実行状態では設定できません	現在インピーダンスモードであり、サーチ動作が停止していることを確認してください
41427	サーチの開始に失敗しました	サーチ動作が設定されていないか、直交インピーダンスが現在実行されていません	制御モードコマンドをチェックして、サーチ動作パラメータを設定してから再試行してください
41428	サーチ運動の一時停止に失敗しました	サーチモーションがオンになっていないか、現在直交インピーダンスが実行されていません	サーチ動作がオンになっていて、直交インピーダンスが実行されていることを確認してください
41429	検索運動を再開できませんでした	サーチモーションが一時停止していないか、現在直交インピーダンスが実行されていません。	現在の力制御任務を一時停止して制御モードを検査して、ディカル抵抗制御モードに切り替えてから再試行してください
41430	インピーダンス制御タイプの設定に失敗しました	インピーダンスが初期化されていません	現在の力制御タスクを停止し、初期化のパラメータを再設定してください
41431	各軸インピーダンスの初期化に失敗しました	現在各軸インピーダンス制御モードではなく、初期化コマンドの実行に失敗しました	力制御を実行する前に各軸インピーダンス制御モードを設定してください
41432	直交インピーダンスの初期化に失敗しました	現在直交インピーダンス制御モードではなく、初期化命令の実行に失敗し	力制御を実行する前に直交インピーダンス制御モードを設定してください

		ました	
41433	ゼロ空間インピーダンスの設定に失敗しました	現在、直交のインピーダンス制御モードではないか、インピーダンスが初期化されていません	現在のインピーダンス制御モードを確認してから実行してください。力値の入力が正確であることを確保し、かつディカル抵抗制御モードを設定してください
41434	ジョイントスペースの期待フォースの設定に失敗しました	現在、各軸インピーダンス制御モードになっていないか、インピーダンスが初期化されていません	現在のインピーダンス制御モードを確認してから実行してください。力値の入力が正確で、かつ軸空間抵抗制御モードの設定を確認してください
41435	直交空間の期待値の設定に失敗しました	現在、直交インピーダンス制御モードではないか、インピーダンスが初期化されていません	現在のインピーダンス制御モードを確認してから実行してください。力値の入力が正確であることを確保し、かつディカル抵抗制御モードを設定してください
41436	カットモードは許可されていません	力制御運転中はインピーダンスモードの切り替えはできません	現在の力制御タスクを停止し、再初期化を行ってください
41437	仮想壁のオープンに失敗しました	ロボットフランジは仮想壁の範囲外にあり、仮想壁の拘束を満たしていません	ロボットフランジを仮想壁の範囲内に移動して仮想壁をオンにしてください
41438	仮想壁のオープンに失敗しました	仮想壁機能はドラッグモードでのみ有効です	ドラッグモードをオンにした後、仮想壁を再度開いてください
41439	仮想壁のオープンに失敗しました	ロボットの現在の状態は仮想壁機能をサポートしていません	
41440	クラッシュテストオフ	インピーダンスモードでは衝突検出は不要	
41441	クラッシュテストを再開	インピーダンスモードがオンの時に自動的に衝突検出をオフにして、今インピーダンスモードがオフの時に、自動的に衝突検出をオンにする	
41442	インピーダンスのオン失敗	インピーダンスが起動する時、ロボットはソフトウェアリミットの要求を満たさなくて、インピーダンスを起動できない。	ロボットがインピーダンスをオンにしたときのソフトウェアリミット内であることを確認してください
41443	インピーダンス剛性値異常	インピーダンス剛性の設定が理論上の最大値を超え、デフォルトの最大値に置き換えられました	インピーダンス剛性の値を適切な範囲に設定し直してください
41444	インピーダンス剛性値異常	物理的に意味のないインピーダンス剛性値の異常の設定	インピーダンス剛性の値を適切な範囲に設定し直してください

41445	力制御識別結果の計算が完了しました	力制御識別計算が完成し、結果は技術要求を満たす。	2 回の認識が完了したら、ロボットを再起動してください
41446	異常を認識する力制御	力制御同定では、一軸キャリブレーション結果が許容範囲を超えている。	認識結果は公称値に置き換えられます。ハードウェアの問題を継続的に確認してください。

17.1.4 5XXXX

エラーコード	エラー内容	原因	修復方法
50000	滑らか軌跡解除	滑らか軌跡解除しきい値を超えました	1.前後 2 つの軌跡の角度を大きくしてください; 2.前後 2 つの軌跡の長さを長くしてください; 3.zone を増やしてください
50001	コントローラーの現在の状態が異常です	コントローラーの現在の状態が異常です	PPToMain プロジェクトのリセットまたはプログラムの再ロードを実行してください
50002	動作範囲外	1.目標点はロボットの運動範囲外である 2.目標点は直交特異位置である	1.目標点の位置を確認する 2.ロボットを関節で動かす 3.CONFDATA 設定を確認してください
50003	隣接する 2 つのターゲットポイントが近接しすぎています	隣接する 2 つのターゲットポイントが近接しすぎています	隣接する 2 つの Move ディレクティブのターゲットポイントが同じポイントかどうかを確認してください
50004	円弧の始点と終点が近すぎて、円弧の生成に失敗しました。軌跡は無視されます	円弧の始点と終点が近すぎて、円弧の生成に失敗しました。軌跡は無視されます	1.円弧の始点または終点の位置や距離を調整してください
50005	円弧の始点と補助点が近すぎて、円弧の作成に失敗しました	円弧の始点と補助点が近すぎて、円弧の作成に失敗しました	1.円弧の始点または終点の位置や距離を調整してください
50006	円弧の終点が補助点に近すぎるため、円弧の作成に失敗しました	円弧の終点が補助点に近すぎるため、円弧の作成に失敗しました	1.円弧の始点または終点の位置や距離を調整してください
50007	円弧の終点が補助点に近すぎるため、円弧の作成に失敗しました	始点、補助点、終点の間の角度が小さすぎるのは、通常、2 つの点間の距離が近すぎるためです。	1.円弧の始点または終点の位置や距離を調整してください
50008	円弧の始点、補助点、終点が直線上にあり、円弧の生成に失敗しました	円弧の始点、補助点、終点が直線上にあり、円弧の生成に失敗しました	1.円弧の始点または終点の位置や距離を調整してください
50009	円弧の半径が小さすぎ	円弧の始点、補助点、終	1.円弧の始点または終点の位置や距離を調整してください

	ます。円弧の作成に失敗しました	点が近すぎます	整してください
50010	正しい円弧を生成できません。円弧の生成に失敗しました	円弧の始点、補助点、終点が近すぎる、または円弧の始点、補助点、終点が直線上にある	1.円弧の始点、補助点、終点の位置や距離または方向を調整してください
50011	トロコイドの始点と終点が近すぎて、トロコイドの生成に失敗しました	トロコイドの始点と終点が近すぎて、トロコイドアークを生成します。	1.トロコイドの始点または終点の位置間の距離を調整します。 2、トロコイドの生成は以下を満たす必要があります:指定された半径は1 mm より大きく、指定された送り量は1 mm より大きく、トロコイドの長さ(始点、補助点、終点の距離によって決まります)は2 倍以上の半径に1.5 倍の送り量を加えます。始点と補助点の線、始点と終点の線、2 つの線の間の角度を5 度より大きくしてください
50012	トロコイドの始点と補助点が近すぎて、トロコイドの生成に失敗しました	トロコイドの始点と補助点が近すぎて、トロコイドの生成に失敗しました	1.トロコイドの始点または終点の位置間の距離を調整します。 2、トロコイドの生成は以下を満たす必要があります:指定された半径は1 mm より大きく、指定された送り量は1 mm より大きく、トロコイドの長さ(始点、補助点、終点の距離によって決まります)は2 倍以上の半径に1.5 倍の送り量を加えます。始点と補助点の線、始点と終点の線、2 つの線の間の角度を5 度より大きくしてください
50013	トロコイドの終点と補助点が近すぎて、トロコイドの生成に失敗しました	トロコイド終点と補助点が近すぎて生成に失敗しました	1.トロコイドの始点または終点の位置間の距離を調整します。 2、トロコイドの生成は以下を満たす必要があります:指定された半径は1 mm より大きく、指定された送り量は1 mm より大きく、トロコイドの長さ(始点、補助点、終点の距離によって決まります)は2 倍以上の半径に1.5 倍の送り量を加えます。始点と補助点の線、始点と終点の線、2 つの線の間の角度を5 度より大きくしてください
50014	トロコイドの始点、補助点、終点が直線上にあり、トロコイドの生成に失敗しました	始点と補助点のある直線、始点と終点のある直線の2 つの直線の間の角度が5 度未満で、トロコイドの生成に失敗しました	1.トロコイドの始点または終点の位置間の距離を調整します。 2、トロコイドの生成は以下を満たす必要があります:指定された半径は1 mm より大きく、指定された送り量は1 mm より大きく、トロコイドの長さ(始点、補助点、終点の距離によって決まります)は2 倍以上の半径に1.5 倍の送り量を加えます。始点と補助点の線、始点と終点の線、2 つの線

			の間の角度を 5 度より大きくしてください
50015	入力半径が小さすぎます。トロコイドの生成に失敗しました	入力半径が小さすぎます。トロコイドの生成に失敗しました	1.入力半径を大きくする; 2、トロコイドの生成は以下を満たす必要があります:指定された半径は 1 mm より大きく、指定された送り量は 1 mm より大きく、トロコイドの長さ(始点、補助点、終点の距離によって決まります)は 2 倍以上の半径に 1.5 倍の送り量を加えます。始点と補助点の線、始点と終点の線、2 つの線の間の角度が 5 度より大きくしてください
50016	指定した送り量が小さすぎて、トロコイドの生成に失敗しました	指定した送り量が小さすぎて、トロコイドの生成に失敗しました	1.入力送り量を増やす; 2、トロコイドの生成は以下を満たす必要があります:指定された半径は 1 mm より大きく、指定された送り量は 1 mm より大きく、トロコイドの長さ(始点、補助点、終点の距離で決まります)は 2 倍以上の半径に 1.5 倍の送り量を加えます。始点と補助点の線、始点と終点の線、2 つの線の間の角度が 5 度より大きくしてください
50017	指定したトロコイドがトロコイドの送り量より小さいため、トロコイドの生成に失敗しました	指定したトロコイドがトロコイドの送り量より小さいため、トロコイドの生成に失敗しました	1、指定した半径の大きさは指定した送り量より小さくはいけません; 2、トロコイドの生成は以下を満たす必要があります:指定された半径は 1 mm より大きく、指定された送り量は 1 mm より大きく、トロコイドの長さ(始点、補助点、終点の距離で決まります)は 2 倍以上の半径に 1.5 倍の送り量を加えます。始点と補助点の線、始点と終点の線、2 つの線の間の角度が 5 度より大きくしてください
50018	トロコイドの長さが短すぎて、トロコイドの生成に失敗しました	指定した半径と送りに対して、始点と終点の距離が短い場合、適切なトロコイドが生成されません。	1.指定した半径または送りを小さくします。 2.始点と終点間の距離を大きくする; 3、トロコイドの生成は以下を満たす必要があります:指定された半径は 1 mm より大きく、指定された送り量は 1 mm より大きく、トロコイドの長さ (始点、補助点、終点の距離によって 2 倍以上の半径と 1.5 倍の送り量を決定します。始点と補助点の線、始点と終点の線、2 つの線の間の角度が 5 度より大きくしてください
50019	パス内部エラーの生成	生成されたパスの内部エラー	1.ターゲットポイントの位置、姿勢、および腕の角度を再調整してください(7 軸のみロボットは腕の角度を考慮する必要があります)。

50020	設定したツール座標系が現在のロボットの特性に合わず、パスの生成に失敗しました	設定されたツール座標系が現在のロボットの特性に適合していません	1.現在のロボットのツール座標系は必ず満足しなければなりません。持ち方:手持ち、位置:x=0、y=0
50021	動作範囲外	1.指定 confdata 下では目標点に到達できません	1.指令「 confl off 」を使って confldata を取り消してください; 2 再教示してください
50022	不正な命令	1.不正な指令	1.再度教示してください; 2.ロボットを再起動してください
50023	特異点エラー	1.目標点が教示されていないか、または目標点の conf 情報が間違っています	目標点の conf を修正してください
50024	間違った軌跡タイプです。現在のロボットは指定された方向に平行移動できません。もう一度指示してください。	現在のロボットは、ベース座標の xz 平面内でのみ平行移動でき、ベース座標系の xz 平面から離れることができない。	再度教示してください
50025	間違った軌跡タイプです。現在のロボットは指定された方向に回転できません。もう一度指示してください。	現在のロボットでは、回転運動の回転軸は基本座標系の y 軸と平行でなければなりません。	再度教示してください
50026	次のモーションコマンドが計算されるまでの待機時間が長すぎます。自動的に滑らか軌跡エリアがキャンセルされます。	2 つの運動指令の間に非運動指令を挿入しすぎて、自動的に曲げ区をキャンセルする	1.RL 命令 AotolgnoreZone Off を使用して、システムは自動的にターンゾーンをキャンセルしません。 2.2 つの運動指令間の非運動指令を簡略化する
50027	軌跡の長さが最小曲がり角半径より小さい、自動的に経路をつなぎ、曲がり角を整合する	1.この軌道は前後とも旋回区を連結する必要があるが、軌道長さは最小旋回区半径の 2 倍より小さい。 2.前記軌跡は 1 本の曲がり区を連結するように設定するが、軌跡の長さは最小曲がり区の半径より小さい。	この機能を使用すると、動きがスムーズになります。この機能をオフにするには、最小滑らか軌跡半径を 0 に設定します。
50101	軸の動作範囲を超えています	軸の動作範囲を超えています	1.ソフトウェアリミットをキャンセルしてください 2.ロボットの各軸を手動で正常な動作範囲内に移動してください
50102	軌跡の途中に特異点がある	特異点を通過する軌跡がある	特異点を回避してください(関連情報取扱

	存在します	存在する	説明書を見る): 1. 教示点を修正し、目標点を変更してください.; 2. 直交座標命令を軸座標命令に変更してください
50103	位置の座標に互換性がありません	直交座標を使用して、指定された ConfData のターゲットポイントに到達できません。	1. 目標点の ConfData を修正してください; 2. MoveJ または MoveAbsJ に変更してください
50104	先読み中に関節トルクが限界を超えた	1. 負荷値の設定が高すぎて、ロボットの耐荷重能力を超えています。 2. ロボットの摩擦係数が高すぎます。 3. ロボットの電気過負荷係数または伝達過負荷係数が小さすぎます。	1. 負荷値が実際の状況と一致しているかどうかを確認してください。 2. 摩擦係数、モーター過負荷係数、トランスミッション過負荷係数などのパラメータを確認してください。 3. 命令のタイプを変更してみます。たとえば、直交座標命令を軸座標命令に変更してください
50105	先読み不連続性	1. プログラム論理の問題; 2. 動力学パラメーターが間違っていて、合理的な範囲内にない数が存在する	1. プログラム論理を修正してください; 2. 動力学パラメーターをチェックしてください; 3. ポイントまたはターンゾーンを変更する
50106	軌跡の始点が終点より大きい、軌跡が不合理	プログラム論理問題	プログラムロジックの問題、プログラムポイントの変更、または滑らか軌跡エリアの変更
50107	不正な滑らか軌跡エリア軌道を生成するには	内部エラー	滑らか軌跡解除、サイズ変更、ターゲットポイントの変更
50108	新しいトラックが取得されず、モーションが停止しました	システム障害	
50109	トラックキューが空です。計画できません	トラックエラーの設定	2 つの反転軌道間で、滑らか軌跡エリアの生成を回避したり、滑らか軌跡エリアの長さを減らしたりします。
50110	プログラムの初期位置を取得できませんでした	プログラムの初期位置を取得できませんでした	1、プログラムロジックの修正; 2. 最初の軌道の滑らか軌跡エリアのサイズを変更する
50111	前後の軌跡の位置がつながりません	前後の軌跡の位置がつながりません	1、もう一度席を教えてください; 2. 滑らか軌跡エリアのサイズを変更する
50112	入力姿勢に誤りがあります	入力姿勢に誤りがあります	1、姿勢の再教示、 2、現在のモデルが三軸または四軸ロボットであれば、入力姿勢が現在のモデルの特徴と一致するかどうかをチェックしてください。
50114	軸の動作範囲を超えて	軸の動作範囲を超えてい	1. ソフトウェアリミットをキャンセルする

	います	ます	2.ロボットの各軸を手動で正常な動作範囲内に移動してください
50115	軌跡の途中に特異点が存在します。特異点を回避してください	先読みパスは特異点を通る	特異点を回避してください(関連情報取扱説明書を見る): 1.教示点を修正し,目標点を変更してください.; 2.またはジョグ方式を変更し,軸ジョグを試してください.; 3.直交座標命令を軸座標命令に変更してください
50116	軌跡の途中に特異点が存在します。	軌跡のつなぎ目は特異点である。	特異点を回避してください(関連情報取扱説明書を見る): 1.教示点を修正し,目標点を変更してください.; 2.またはエラートレースの後ろに wait0 を追加してください
50201	直交パスで到達不能点が発出されました	直交パスで特異点が発出されました	軌跡を変更するか、軸座標命令によってターゲット位置に移動してください
50202	直交パスの不正な位置	直交パス上のターゲットポイントが現在のロボット構成に一致しません	目標点を再教示してください。もし目標点が手入力によるものであれば教示点を変更できます。
50203	パス計画エラー	内部エラー	
50204	計画中に、不十分なサンプルポイントが発出されました	1、2 つの運動命令の間に、過剰な論理判断またはビット計算を挿入した; 2、マシンの状態が不安定なので、 xCore 以外のプログラムがロボットで動作しているかどうかを確認してください。	1、 RL 工程を修正し、運転工程を再開してください; 2、 xCore 以外の制御システムの Linux デモンを停止してください
50205	運動計画における隣接位置指令点の差が大きすぎる	コントローラーエラー、モーションプラン内の隣接位置の命令の差が大きすぎて制限を超えています	1.軌跡の目標点位置を再教示してください.; 2.指令の形式を変更してみてください。たとえば、 MoveL を MoveJ に変更します。 3.エラートラックの後に Wait 0 を追加して状態を更新してください
50206	パスプランニング内部計算エラー	コントローラーの計画中に計算エラーが発生しました	1.軌跡の目標点位置を再教示してください.; 2.指令の形式を変更してみてください。たとえば、 MoveL を MoveJ に変更します。 3.軌道コーナーのサイズを変更し、希望する速度のサイズを変更してみてください; 4.エラートラックの後ろに Wait 0 を入れて状態を更新してください
50207	パスの終点で停止して	計画エラー、ロボット経	モーション割り当てパラメーターの変更

	いません	路の途中で停止	(滑らか軌跡エリアのサイズ、ターゲットポイントの位置など)、ターゲットポイントの姿勢の変更、またはモーションコマンドの変更(MoveL のように MoveJ に変更します。)
50208	トラック内部エラー	トラック内部エラー	1.滑らか軌跡エリアを増減する; 2.もう一度教えてください;
50209	指定した停止距離で停止できません	設定された停止距離が短すぎます	停止距離を増やしてください
50210	回転角の実際の到達点と設定点との違いが大きすぎます	直交動作は、目標点に最も近い軸角度姿勢で到達します。ConfL on の場合、この状況について警告が表示されます。	1. このエラーメッセージは動作を停止させません。ユーザーはこの警告を処理するかどうかを検討できます。 2. Conf off 指令を使用することで、直交命令の回転角度の検査を回避できます。
50301	モーションプランニングスレッドがブロックされ、実行がタイムアウトしました	モーションプランニングスレッドがブロックされ、実行がタイムアウトしました	pptomain エンジニアリングの実行またはコントローラーを再起動してください
50302	モーションプランニングスレッドがブロックされ、タイムアウトがスケジュールされました	システム障害	pptomain エンジニアリングの実行またはコントローラーを再起動してください
50303	モーションプランニングスレッドがブロックされ、連続して 5000 回を超えるタイムアウトが発生しました。	システム障害	pptomain エンジニアリングの実行またはコントローラーを再起動してください
50304	EtherCAT スレッドが十分なプランナ命令を受信していません	システム障害	
50401	ロボットが衝突して運転を停止しましたので、ロボットの運転環境を点検し、人員、設備の安全を確認してから運転を再開してください	ロボットが衝突する	ロボットの運転環境を点検し、人員、設備の安全を確認した後、再び電源を入れ、運転を再開してください。
50402			
50501	ツールワークピース座標系の矛盾の設定	同時ハンドヘルドワーク、ハンドヘルドツールの設定	正しい tool、wobj を選択
50502	ツールワークピース座標系の矛盾の設定	外部ワークピースと外部ツールの同時設定	正しい tool、wobj を選択

50503	ツールワークピース座標系の矛盾の設定	同時ハンドヘルドワーク、ハンドヘルドツールの設定	正しい tool、wobj を選択
50504	ツールワークピース座標系の矛盾の設定	外部ワークピースと外部ツールの同時設定	正しい tool、wobj を選択
50506	モーションモジュールタイプ 1 の内部モードエラー	内部の問題	ロボットを再起動すると解決しますが、念のため弊社にご連絡ください
50507	モーションモジュールのクラス 2 内部モードエラー	内部の問題	ロボットを再起動すると解決しますが、念のため弊社にご連絡ください
50508	モジュール状態のリセットに失敗しました	内部の問題	ロボットを再起動すると解決しますが、念のため弊社にご連絡ください
50509	モーションの開始点を設定できませんでした	内部の問題	ロボットを再起動すると解決しますが、念のため弊社にご連絡ください
50510	パスの生成に失敗しました	内部の問題	ロボットを再起動すると解決しますが、念のため弊社にご連絡ください
50511	運動の開始に失敗しました	内部の問題	ロボットを再起動すると解決しますが、念のため弊社にご連絡ください
50512	軸の数が一致していません	ソフトウェアの問題	弊社にご連絡ください
50513	速度設定が無効です	ソフトウェアの問題	弊社にご連絡ください
50514	ステップ設定が無効です	ソフトウェアの問題	弊社にご連絡ください
50515	座標系の設定が無効です	ソフトウェアの問題	弊社にご連絡ください
50516	運動軸の設定が無効です	ソフトウェアの問題	弊社にご連絡ください
50517	端末のクイック調整に失敗しました。もう一度教えてください	1.特殊または限界位置にあり、末端の快速調整が使えません	ロボットの動作範囲内で再教示してください
50518	ターゲットポイントは現在のポイントの可能性があります	ターゲットポイントは現在のポイントの可能性がります	
50519	トラックの生成に失敗しました。ターゲットポイントが要件を満たしていません	ターゲットポイントがロボットの動作範囲外になる可能性があります	ロボットの動作範囲内で再教示してください
50520	ジョグモードが正しくありません。現在のロボットは指定された方向に移動できません。ジョグモードを変更してください。	現在のロボットは、ベース座標の xz 平面内でのみ平行移動でき、ベース座標系の xz 平面から離れることができない。	ジョグでは、ベース座標系またはジョイント座標系を優先的に選択することをお勧めします。

50521	現在のロボットが指定された方向に回転できないジョグモードが正しくありません。ジョグモードを変更してください	現在のロボットでは、回転運動の回転軸はベース座標系の y 軸に平行であるか、フランジ座標系の Z 軸に沿って回転している必要があります。	ジョグでは、ベース座標系またはジョイント座標系を優先的に選択することをお勧めします。
50522	誤った軌道, PCB 4 軸ロボットはハンドヘルドワーク, 外付けツールをサポートしない	PCB 4 軸ロボットはハンドヘルドワーク、外付けツールに対応していない	ハンドヘルドツールに切り替える
50523	ロボットは現在の軌跡の終点にいます	次のステップをクリックすると、ロボットはすでに現在の軌跡の終点において、ロボットは動作する必要がありません。	現在のトラックの終点に移動するかどうかを確認し、次のトラックの終点に移動する必要がある場合は、[次へ] をクリックします。
50601	ダイナミクスがオンになっておらず、モーメントフィードフォワードがゼロに設定されているため、電氣的にうなずきます。	モーメントフィードフォワードをオフにし、モーメントフィードフォワードをゼロにします。	開放モーメントフィードフォワード

17.1.5 6XXXX

エラーコード	エラー内容	原因	修復方法
60000	RL 命令パラメーターエラー	RL 命令パラメーターエラー	コマンドを修正して正しいパラメーターを入力してください
60001	SocketReadBit のパラメーターエラー	RL の実行中にエラーが発生しました	SocketReadBit の引数は 8 の倍数でなければなりません。
60002	SocketReadBit データの読み取りに失敗しました	RL の実行中にエラーが発生しました	SocketReadBit データの読み取りに失敗しました(データ長が一致しません)
60003	SocketReadDouble 読み取りデータが不正です	RL の実行中にエラーが発生しました	SocketReadDouble 読み取りデータが不正です
60004	SocketReadDouble データの読み取りに失敗しました	RL の実行中にエラーが発生しました	SocketReadDouble データの読み取りに失敗しました(データ長が一致しません)
60005	RL プロジェクトのロードに失敗しました	1.rsync 設定ファイルが間違っているか、 2.hmi バージョンが一致していない 3.RL プロジェクトファイルが誤って変更されている	1.正しい実行環境を設定してください 2.対応する HMI バージョンをチェックしてください 3.ROKAE に連絡してください

		4.ネットワーク障害	
60006	ロボットが実行中で pptomain は失敗しました	ロボットは実行中のため許可されていません	ロボットを停止するには、一時停止または非常停止を使用してください
60007	ソフトウェアリミットがオンになっておらず、シングルステップのデバッグが許可されていない	ソフトウェアリミットがオンになっておらず、シングルステップのデバッグが許可されていない	セットアップ画面でソフトウェアリミット設定をオンにしてください
60008	速度同期の実行に失敗しました。次のステップをキャンセルしてください	速度同期の実行に失敗しました。次のステップをキャンセルしてください	コントローラーの一時停止と再起動を行ってください
60009	RL の実行中またはアクチュエータのエラー状態、次のステップを拒否	RL の実行中またはアクチュエータのエラー状態、次のステップを拒否	RL が実行中の場合は、RL の実行が停止するまで待つか、[一時停止] をクリックしてください。アクチュエータがエラー状態の場合は、pptomain をクリックして再実行してください
60010	インタプリタの起動に失敗しました。次のステップを実行できません	インタプリタの起動に失敗しました。次のステップを実行できません	コントローラーを再起動するか、弊社にお問い合わせください。
60011	RL タスクの実行中にエラーが発生し、タスクが停止しました	RL タスクの実行中にエラーが発生し、タスクが停止しました	対応するタスクチェッカーのポインタ付近の論理エラーに切り替え、エラーを排除した後 pptomain 再実行してください
60012	GetSocketConn 失敗	接続が存在しないか、切断されています	SocketConnect に接続してから実行してください
60013	GetSocketServer 失敗	接続が存在しないか、切断されています	SocketServer がリスニング・ステータスになった後にこのコマンドを実行してください
60014	RL プログラムの開始アクションを実行できません	プログラムの実行が失敗する原因は次のとおりです。1) ロボットの電源が入らない;2.ロボットはすでに稼働中	ロボットが電源をオンにしてアイドル状態になっていることを確認してください
60100	HexToDec のパラメーターエラー	RL の実行中にエラーが発生しました	HexToDec 入力する文字列は 16 進整数である必要があります
60101	StrToByte のパラメーターエラー	RL の実行中にエラーが発生しました	StrToByte (string, ¥Hex)入力する文字列は 16 進整数である必要があります
60102	StrToByte のパラメーターエラー	RL の実行中にエラーが発生しました	StrToByte (string)入力する文字列は 10 進整数である必要があります
60103	StrToByte のパラメーターエラー	RL の実行中にエラーが発生しました	StrToByte (string,\Oct)入力する文字列は 8 進整数でなければなりません
60104	StrToByte のパラメーターエラー	RL の実行中にエラーが発生しました	StrToByte (string,\Bin)入力する文字列は 2 進整数でなければなりません

60105	StrToByte のパラメータエラー	RL の実行中にエラーが発生しました	StrToByte (string,\Char)入力する文字列は ASCII 文字でなければなりません
60106	StrToByte の結果のオーバーフロー	RL の実行中にエラーが発生しました	StrToByte (string HEX)入力された文字列データがあふれました
60107	StrToByte の結果のオーバーフロー	RL の実行中にエラーが発生しました	StrToByte (string DEC)入力された文字列データがあふれました
60108	StrToByte の結果のオーバーフロー	RL の実行中にエラーが発生しました	StrToByte (string OKT)入力された文字列データがあふれました
60109	StrToByte の結果のオーバーフロー	RL の実行中にエラーが発生しました	StrToByte (string BIN)入力された文字列データがあふれました
60110	StrToByte の結果のオーバーフロー	RL の実行中にエラーが発生しました	StrToByte (string CHAR)入力された文字列データがあふれました
60111	StrPart 文字列の範囲のオーバーランを切り取る	RL の実行中にエラーが発生しました	StrPart 文字列の範囲のオーバーランを切り取ってください
60112	StrMatch 一致パラメーターが大きすぎて検索できません	RL の実行中にエラーが発生しました	StrMatch 一致パラメーターが大きすぎて検索できません
60113	BitCheck 指令 BitPos パラメーター値の範囲超過	RL の実行中にエラーが発生しました	BitCheck 指令 BitPos パラメーター値の範囲を超過しています
60114	BitClear 指令 BitPos パラメーター値の範囲超過	RL の実行中にエラーが発生しました	BitCheck 指令 BitPos パラメーター値の範囲を超過しています
60115	BitLsh 指令 ShiftSteps パラメーター値の範囲超過	RL の実行中にエラーが発生しました	BitLsh 指令 ShiftSteps パラメーター値の範囲を超過しています
60116	BitRsh 指令 ShiftSteps パラメーター値の範囲超過	RL の実行中にエラーが発生しました	BitLsh 指令 ShiftSteps パラメーター値の範囲を超過しています
60117	BitSet 指令 BitPos パラメーター値の範囲超過	RL の実行中にエラーが発生しました	BitSet 指令 BitPos パラメーター値の範囲を超過しています
60200	FcInit 実行中にエラーが発生しました	FcInit 実行中にエラーが発生しました	FcInit 指令を正しい指令パラメーターに変更してください
60201	FcStart 実行中にエラーが発生しました	FcStart 実行中にエラーが発生しました	FcStart 指令を正しい指令パラメーターに変更してください
60202	FcPause 実行中にエラーが発生しました	FcPause 実行中にエラーが発生しました	FcPause 指令を正しい指令パラメーターに変更してください
60203	FcRestart 実行中にエラーが発生しました	FcRestart 実行中にエラーが発生しました	FcRestart 指令を正しい指令パラメーターに変更してください
60204	FcStop 実行中にエラーが発生しました	FcStop 実行中にエラーが発生しました	FcStop 指令を正しい指令パラメーターに変更してください
60205	ClearFcError 実行中にエラーが発生しました	ClearFcError 実行中にエラーが発生しました	ClearFcError 指令を正しい指令パラメーターに変更してください

60206	SetControlType 実行中にエラーが発生しました	SetControlType 実行中にエラーが発生しました	SetControlType 指令を正しい指令パラメーターに変更してください
60207	SetJntCtrlStiffVec 実行中にエラーが発生しました	SetJntCtrlStiffVec 実行中にエラーが発生しました	SetJntCtrlStiffVec 指令を正しい指令パラメーターに変更してください
60208	SetCartCtrlStiffVec 実行中にエラーが発生しました	SetCartCtrlStiffVec 実行中にエラーが発生しました	SetCartCtrlStiffVec 指令を正しい指令パラメーターに変更してください
60209	SetCartNSStiff 実行中にエラーが発生しました	SetCartNSStiff 実行中にエラーが発生しました	SetCartNSStiff 指令を正しい指令パラメーターに変更してください
60210	SetLoad 実行中にエラーが発生しました	SetLoad 実行中にエラーが発生しました	SetLoad 指令を正しい指令パラメーターに変更してください
60211	StartOverlay 実行中にエラーが発生しました	StartOverlay 実行中にエラーが発生しました	StartOverlay 指令を正しい指令パラメーターに変更してください
60212	StopOverlay 実行中にエラーが発生しました	StopOverlay 実行中にエラーが発生しました	StopOverlay 指令を正しい指令パラメーターに変更してください
60213	PauseOverlay 実行中にエラーが発生しました	PauseOverlay 実行中にエラーが発生しました	PauseOverlay 指令を正しい指令パラメーターに変更してください
60214	SetSineOverlay 実行中にエラーが発生しました	SetSineOverlay 実行中にエラーが発生しました	SetSineOverlay 指令を正しい指令パラメーターに変更してください
60215	SetLissajousOverlay 実行中にエラーが発生しました	SetLissajousOverlay 実行中にエラーが発生しました	SetLissajousOverlay 指令を正しい指令パラメーターに変更してください
60216	SetJntTrqDes 実行中にエラーが発生しました	SetJntTrqDes 実行中にエラーが発生しました	SetJntTrqDes 指令を正しい指令パラメーターに変更してください
60217	SetCartForceDes 実行中にエラーが発生しました	SetCartForceDes 実行中にエラーが発生しました	SetCartForceDes 指令を正しい指令パラメーターに変更してください
60218	RestartOverlay 実行中にエラーが発生しました	RestartOverlay 実行中にエラーが発生しました	RestartOverlay 指令を正しい指令パラメーターに変更してください
60219	SetSensorUseType 実行中にエラーが発生しました	SetSensorUseType 実行中にエラーが発生しました	SetSensorUseType を正しいパラメーターに変更してください
60220	CalibSensorError 実行中にエラーが発生しました	CalibSensorError 実行中にエラーが発生しました	CalibSensorError を正しいパラメーターに変更してください
60221	FcCondForce 実行中にエラーが発生しました	FcCondForce 実行中にエラーが発生しました	FcCondForce 指令を正しい指令パラメーターに変更してください
60222	FcCondPosBox 実行中にエラーが発生しました	FcCondPosBox 実行中にエラーが発生しました	FcCondPosBox 指令を正しい指令パラメーターに変更してください

60223	FcCondTorque 実行中にエラーが発生しました	FcCondTorque 実行中にエラーが発生しました	FcCondTorque 指令を正しい指令パラメーターに変更してください
60224	FCCondWaitWhile 実行中にエラーが発生しました	FCCondWaitWhile 実行中にエラーが発生しました	FCCondWaitWhile 指令を正しい指令パラメーターに変更してください
60225	FcCondOrient 実行中にエラーが発生しました	FcCondOrient 実行中にエラーが発生しました	FcCondOrient 指令を正しい指令パラメーターに変更してください
60226	FcCondPosSphere 実行中にエラーが発生しました	FcCondPosSphere 実行中にエラーが発生しました	FcCondPosSphere 指令を正しい指令パラメーターに変更してください
60227	FcCondPosCylinder 実行中にエラーが発生しました	FcCondPosCylinder 実行中にエラーが発生しました	FcCondPosCylinder 指令を正しい指令パラメーターに変更してください
60228	FcCondTcpSpeed 実行中にエラーが発生しました	FcCondTcpSpeed 実行中にエラーが発生しました	FcCondTcpSpeed 指令を正しい指令パラメーターに変更してください
60229	CalcJointT 逆変換に失敗しました	直交座標の conf エラー	入力の直交座標を変更してください
60300	ReplayPath 再生速度オーバー	RL の実行中にエラーが発生しました	ReplayPath 再生速度オーバー
60400	RL 呼び出しレベルが制限を超えました	RL の実行中にエラーが発生しました	RL 呼び出しレベルが制限を超えました
60500	複数のモーションタスクをロードすることはできません	複数のモーションタスクをロードすることはできません	1 つのタスクのみを移動タスクとして指定してください
60501	先行タスクエラー	先行タスクを自分自身にすることはできません	他のタスクへ変更してください
60502	モーションタスクでは先行タスクを設定できません	モーションタスクでは先行タスクを設定できません	モーションタスクでは先行タスクを設定できません
60503	モーションコマンドでは、ワークとツールを同時に保持することはできません	モーションコマンドでは、ワークとツールを同時に保持することはできません	コマンドルールおよび wobj パラメーターを変更してください
60504	Search 命令 DI が無効です	この信号タイプは DI 信号ではありません	適切な DI 信号を使用してください
60505	Search コマンドがドットを記録できませんでした	ツールとワークピースは、ハンドまたは外部のいずれかです。	異なる位置にあるツールやワークピースを使用してパラメーターを修正する
60506	Search 命令 DI 信号エラー	DI 信号が存在しないか、システム入力として設定されています。	システム入力として設定されていない DI 信号を使用するようにパラメーターを変更してください
60507	SearchL /Stop コマン	速度が v 100 時以下の場	より小さい速度パラメーターの使用してく

	ド、速度は v100 を超えています	合のみ、ロボットをすばやく停止させることができます。	ださい
60508	SearchC /STOP コマンド、速度は v100 を超えています	速度が v 100 時以下の場合のみ、ロボットをすばやく停止させることができます。	より小さい速度パラメーターの使用してください
60509	プログラムの状態が正しくないため、次のステップを実行できません	1.指令点に到達できません; 2.軌跡の途中に特異点が存在します;	1.ポイントリストを確認してください; 2.モーション軌跡の最適化してください; 3.PPtoMain をクリックしてください
60510	RL コマンドの動作が遅い	tcp が 1μm/s 未満または ori が 1 e-6°/s 未満または jnt が 1% 未満;	適切な速度パラメーターに変更してください
60511	ドラッグ再生パスがありません	ドラッグ再生パスがありません	有効な再生パスの記録/使用をお願いします
60512	ドラッグバック機能のエラー	ドラッグバック機能のエラー	ドラッグバック機能が有効かどうかを確認してください
60513	HomeSet 命令範囲エラー	ホームポイントの角度が制限を超えています	ロボットのハードリミットの範囲内でホームポジションの角度を変更してください
60514	Home パラメーターの保存に失敗しました	ロボット構成モジュールエラー	ロボットを再起動するか、メーカーに連絡してください
60515	HomeSetAt 取得に失敗しました	HomeSetAt 取得した軸が存在しません	HomeSetAt のパラメーターの変更してください
60516	HomeClr コマンドが失敗しました	ロボット構成モジュールエラー	ロボットを再起動するか、メーカーに連絡してください
60517	HomeSet パラメーター数エラー	HomeSet パラメーター数エラー	修正パラメーター数は、ロボットの軸数と一致させてください
60518	Hordr パラメーター数エラー	Hordr パラメーター数エラー	RL ファイルを変更して正しい Hordr パラメーターを転送してください
60519	Hordr パラメーター値が正しくありません	Hordr パラメーター値が正しくありません	RL ファイルを変更して正しい Hordr パラメーターを転送してください
60520	HordrAt パラメーター値が正しくありません	HordrAt パラメーター値が正しくありません	RL ファイルを変更して正しい Hordr パラメーターを転送してください
60521	GetEndtoolTorque ツールアーティファクトに互換性がありません	工具ワークピースは外部またはハンドのいずれかです。	GetEndtoolTorque 命令を修正し、正しい tool, wobj パラメーターを設定してください
60522	GetEndtoolTorque 末端座標系のタイプが正しくありません	未定義の終端座標系タイプ	GetEndtoolTorque 指令を修正し、正しい座標系タイプパラメーターを転送します。 0~2
60523	AccSet パラメーターが制限範囲 30% ~ 100% を超えています	AccSet の加速度とジャークは 30% ~ 100% に制限されます	AccSet 指令パラメーターを変更してください
60524	クォータニオンパラメ	変数のクォータニオンの	対応するクォータニオンを正しい値に変更

	ーターエラー	二乗和は 1 にする必要があります	してください
60525	OpenDev ポートエラー	ポートの使用可能範囲[0, 65535]	OpenDev ポートを正しい値に変更してください
60526	ロボットがエラー状態のため、プログラムの開始に失敗しました。最近のエラー記録を確認してください	ロボットがエラー状態のため、プログラムの開始に失敗しました。最近のエラー記録を確認してください	1. RL 言語エラーまたは論理エラーを修正してください。 2. PPToMain とプログラムを再起動してください
60528	Trigger コマンドパラメーターエラー	trigdata 設定された DO/GO/レジスタ変数が存在しません;または出力信号が設定されていない;またはレジスタが読み取り専用です。	出力信号/書き込み可能レジスタを作成し、TrigIO/TrigReg コマンドで設定してください
60529	レジスタ起動RLプログラムが無効です。ロボットはビジー状態です。	1、ロボットは運動中で、繰り返し起動できません; 2、RL ファイルを解析中、起動できません	レジスタファンクションコード "sta_robot_is_busy"が 0 になるのを待ってからコマンドを起動してください
60530	ユーザ座標系の命名規則違反	インポートしたユーザ座標系と既存の座標系情報の名前が重複しています。重複した座標系情報が正しく使用されていない可能性があります。	ユーザ座標系のリストを確認し、必要なデータを保持し、重複する座標系情報を削除してください
60531	トレイデータの更新に失敗しました	1、バレット名または部品番号が間違っています。 2、トレイモジュールからデータを取得できませんでした; 3、バレット変数値の更新に失敗しました	1."TrayUpdate"関数の入力パラメーターが間違っていないかどうかを確認してください; 2、弊社にご連絡ください;
60532	トレイのジョブ数の取得に失敗しました	1、トレイ名が間違っています; 2、トレイモジュールからデータを取得できませんでした;	1、トレイの名前が間違っていないかどうかをチェックしてください; 2、弊社にご連絡ください;
60533	パレタイズデータの更新に失敗しました	1、パレタイズ名または工作 番号の間違いまたはレイヤー番号の間違い; 2、パレタイズモジュールからデータを取得でき	1."PalletUpdate"関数の入力パラメーターが間違っていないかどうかを確認してください; 2、弊社にご連絡ください;

		ませんでした; 3、パレタイザー変数値 の更新に失敗しました	
60534	パレタイズ数の取得に 失敗しました。名前 "\$arg" は存在しません	パレタイズ名が間違っ ています	"PalletLayerCount"関数の入力パラメータ を確認してください
60535	ジョブ数の取得に失敗 しました	パレタイズ名またはレイ ヤ番号が間違っている	"PalletWobjCount"関数の入力パラメータ を確認してください

The background is a dark, textured grey. It features faint, light-grey geometric patterns, including concentric circles and intersecting lines, which create a subtle, futuristic or architectural feel. The overall composition is minimalist and modern.

ROKAE