

デンソーロボット

垂直多関節型
V* シリーズ

水平多関節型
H* シリーズ

直角座標型
XYC シリーズ

組込型
XR シリーズ

立上げハンドブック

Copyright © 2007-2011 DENSO WAVE INCORPORATED
All rights reserved.

この取扱説明書の著作権は、株式会社デンソーウェーブにあります。

本書に掲載されている会社名や製品は、一般に各社の商標または登録商標です。

仕様は予告なく変更することがあります。

はじめに

デンソーロボットをお買い上げいただき、誠にありがとうございます。

この製品は当社の技術を結集した、高速・高精度でかつ高度な機能を備えた「組立て用ロボット」です。

ご使用にあたっては、本書をよく読み理解のうえ、安全で効率的な運用をお願いします。

お願い

ご使用前に、「安全にご使用いただくために」をお読みいただき、正しく安全にデンソーロボットをお使いください。

本書の構成

本書の構成は、以下のようになっております。

第1部 設置準備

ロボットシステム、RC7M型コントローラの概要、インタフェースの概要、ケーブル類の接続方法、専用信号の結線について説明します。

第2部 ロボット操作

座標系、ティーチングペンダントの操作方法、ティーチング方法について説明します。

第3部 簡単なプログラミング

プログラムの基礎知識、ティーチングペンダントやWINCAPSⅢを使ったプログラムの作成方法について、練習問題を使って説明します。

第4部 プログラムの検証

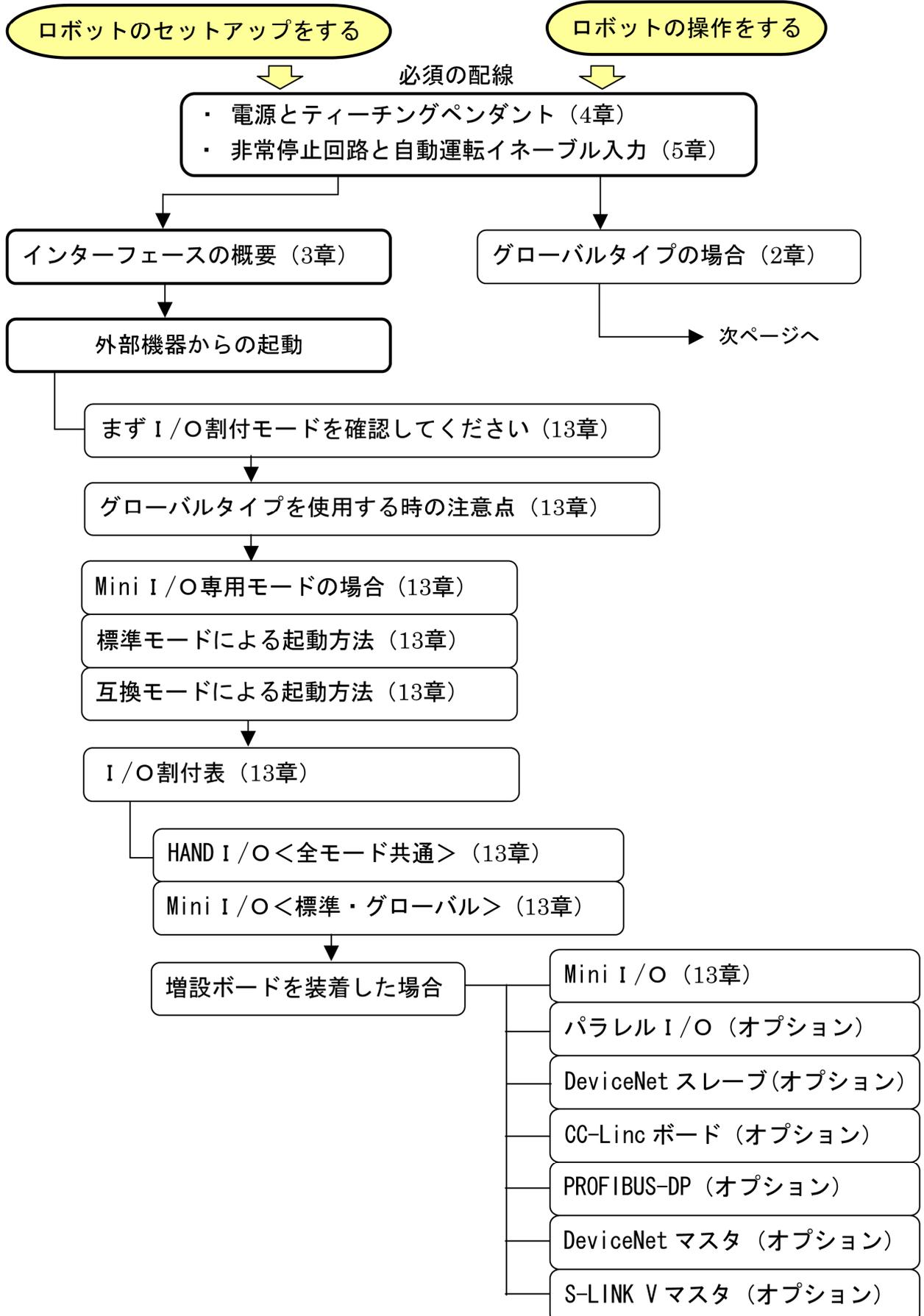
作成したプログラムの確認方法としてWINCAPSⅢによるシミュレーション、ティーチングペンダントによる動作確認、外部機器からの運転方法について説明します。また、I/Oや変数のモニタリング方法についても説明します。

第5部 さらに高度な使い方

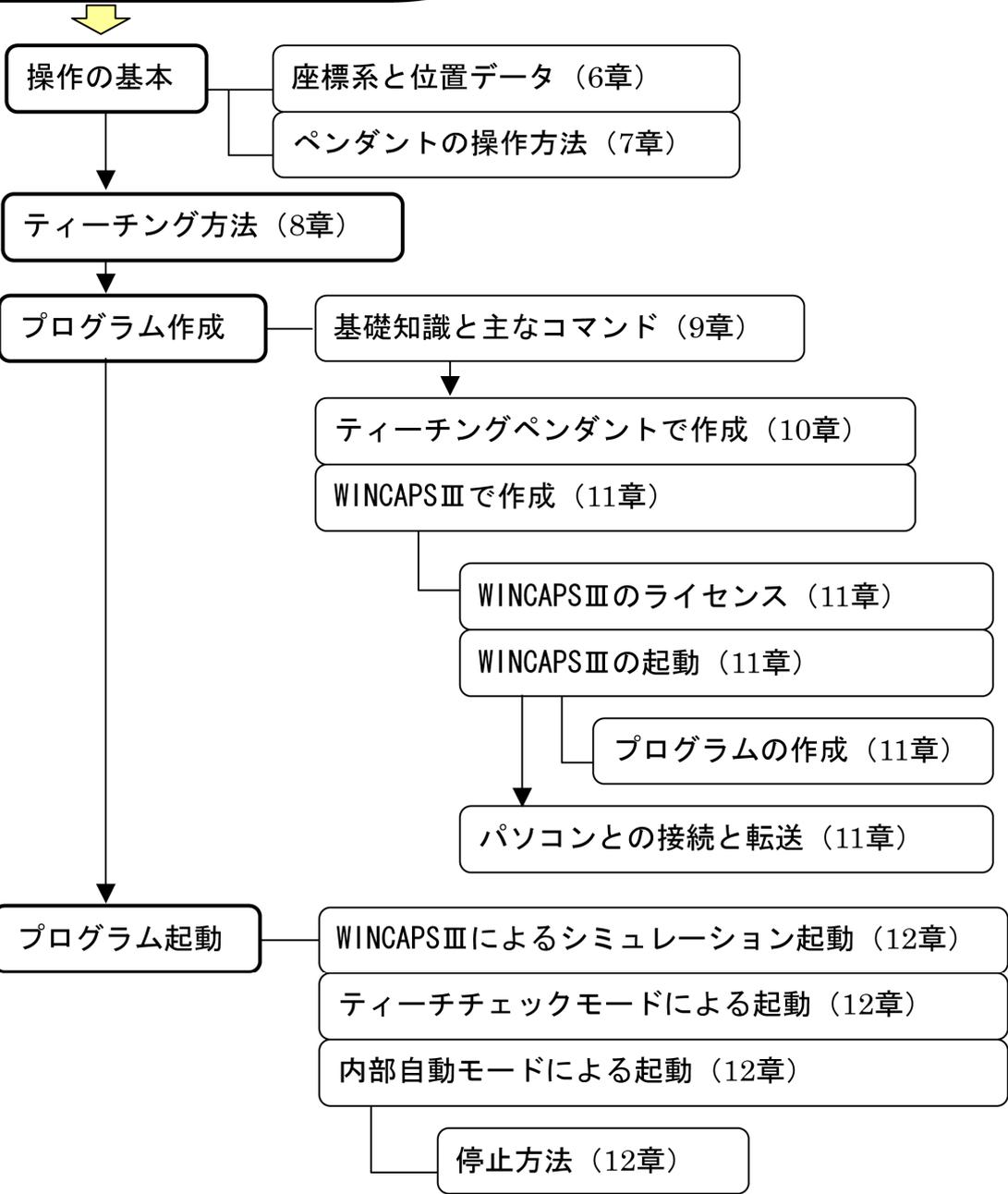
使用条件の最適化方法やよく使われるプログラム命令などさらに高度な使い方をするための情報を提供します。

付録

立ち上げハンドブック 総合ガイダンス



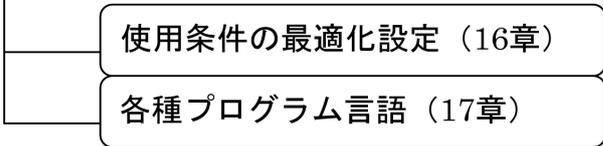
ロボットの手動操作から自動運転まで



I/O のモニタリングと操作 (14章)

変数のモニタリングと値変更 (15章)

さらに高度な使い方



目次

立ち上げハンドブック 総合ガイダンス

第 1 部 設置準備

第 1 章 ロボットシステムの構成	1-1
1.1 ロボットシステムの構成機器	1-1
1.2 標準構成部品	1-2
1.3 オプション品	1-3
第 2 章 RC7M 型コントローラの概要	2-1
2.1 コントローラ型式の見方	2-1
2.1.1 VS-***シリーズ用 RC7M ロボットコントローラの型式	2-2
2.2 グローバルタイプコントローラについてのご注意	2-3
2.2.1 デッドマンスイッチ（イネーブルスイッチ）の機能について	2-3
2.2.2 単一位置制御機能（Single Point of control）について	2-4
第 3 章 インタフェースの概要	3-1
3.1 Mini I/O 入出力信号の種類とその概要	3-1
3.1.1 Mini I/O 入出力信号の種類(標準タイプコントローラ使用時)	3-1
3.1.2 Mini I/O 入出力信号の種類(グローバルタイプコントローラ使用時)	3-2
3.2 I/O 増設ボードの概要	3-3
3.2.1 I/O 増設ボードの種類	3-3
3.3 I/O 増設ボードの組合せと割付モード	3-4
3.3.1 割付モード別 I/O 割付一覧	3-5
3.3.2 割付モード別 機能概要	3-5
3.4 互換・標準・全汎用モード選択時の Mini I/O の機能	3-6
3.5 インタフェースに必要な設定	3-6
3.5.1 I/O 割付モードの設定	3-6
3.5.2 I/O 用電源(+24V DC)の設定	3-6
3.6 I/O 割付モードの設定方法	3-7
3.6.1 ティーチングペンダントからの設定方法	3-7
3.6.2 WINCAPSⅢからの設定方法	3-7
3.7 Mini I/O 用電源の設定方法	3-10
3.8 パラレル I/O 用電源の設定方法	3-11
3.9 I/O ポートマップと割付	3-12

第 4 章 ケーブル類の接続	4-1
4.1 電源・本体間ケーブルの接続.....	4-1
4.2 ペンダントケーブルの接続.....	4-1
4.3 ブレーカの選定（推奨品）.....	4-2
4.4 1 次側電源の配線方法.....	4-3
第 5 章 専用入力信号の結線	5-1
5.1 ロボット調整時に必要となる結線.....	5-1
5.1.1 非常停止回路の構成（標準タイプコントローラ使用時）.....	5-1
5.1.2 非常停止回路の構成（グローバルタイプコントローラ）.....	5-1
5.2 モータ入りに必要な結線.....	5-2
5.2.1 機能.....	5-2
5.2.2 標準タイプコントローラ.....	5-2
5.2.3 グローバルタイプコントローラ.....	5-2
5.3 自動運転に必要な結線.....	5-2
5.3.1 機能.....	5-2
5.3.2 標準タイプコントローラ.....	5-2
5.3.3 グローバルタイプコントローラ.....	5-2

第 2 部 ロボット操作

第 6 章 座標系	6-1
6.1 4 軸ロボットの座標系.....	6-1
6.2 4 軸ロボットのベース座標系.....	6-1
6.3 4 軸ロボットワーク座標系.....	6-1
6.4 4 軸ロボットツール座標系.....	6-2
6.5 4 軸ロボットツール座標系のメリット.....	6-2
6.6 4 軸ロボットで扱う位置データ.....	6-3
6.6.1 4 軸の形態について.....	6-3
6.7 6 軸ロボットの座標系.....	6-4
6.8 6 軸ロボットベース座標系.....	6-4
6.9 6 軸ロボットワーク座標系.....	6-4
6.10 6 軸ロボットツール座標系.....	6-5
6.11 6 軸ロボットツール座標系のメリット.....	6-6
6.12 6 軸ロボットで扱う位置データ.....	6-7
6.12.1 6 軸の形態について.....	6-8
第 7 章 ティーチングのための準備	7-1
7.1 ティーチングペンダントの取扱い.....	7-1
7.1.1 ティーチングペンダントの持ち方とデッドマンスイッチ.....	7-1
7.1.2 ティーチングペンダントのキー、ボタン、スイッチの名称.....	7-2
7.2 動作モードの種類.....	7-4
7.2.1 手動モード.....	7-4
7.2.2 ティーチチェックモード.....	7-4
7.2.3 自動モード.....	7-4

7.3 動作モード切り替え	7-5
7.3.1 動作モード切り替え方法.....	7-5
7.3.2 動作モードと自動イネーブル入力信号の関連性.....	7-5
7.4 手動モードの種類.....	7-6
7.4.1 各軸・X-Y・ツールモード別 ロボット操作.....	7-6
7.4.2 手動モードの切替え方法.....	7-7
7.5 ロボットを手動モードで動かす	7-9
第8章 ティーチング	8-1
8.1 ティーチングとは.....	8-1
8.2 ティーチング可能なグローバル変数.....	8-1
8.3 位置変数へのティーチング方法	8-2
8.4 ティーチングした位置変数へ直接移動させる方法.....	8-7
8.5 アプローチ長を含む目標位置に移動する(Ver.2.61以降)	8-8

第3部 簡単なプログラミング

第9章 プログラム作成の基礎知識.....	9-1
9.1 PAC 言語の特徴.....	9-1
9.2 文と行	9-1
9.3 名前.....	9-1
9.4 登録可能な最大プログラム数.....	9-2
9.5 プログラム構成の概要.....	9-2
9.6 プログラムの主なコマンド.....	9-3
9.6.1 プログラム例.....	9-3
9.6.2 コマンド書式で扱う特殊記号について.....	9-3
9.6.3 プログラム名の宣言 (PROGRAM コマンド)	9-4
9.6.4 アーム制御権取得 (TAKARM コマンド)	9-4
9.6.5 プログラムの停止 (END コマンド)	9-4
9.6.6 アーム移動速度の指定 (SPEED コマンド)	9-4
9.6.7 コメントの記述 (REM コマンド)	9-4
9.6.8 指定座標への移動 (MOVE コマンド)	9-5
9.7 Z軸方向への移動 (APPROACH , DEPART)	9-8
9.7.1 ハンド方向への接近移動 (APPROACH コマンド)	9-8
9.7.2 ハンド方向への回避移動 (DEPART コマンド)	9-9
9.8 変数の概要と性質	9-10
9.8.1 グローバル変数.....	9-11
9.8.2 ローカル変数.....	9-12
9.9 外部機器から起動できるプログラム名	9-13
第10章 プログラムの作成 (ティーチングペンダント編)	10-1
10.1 作成するサンプルプログラム.....	10-1

10.2	プログラムの作成	10-2
10.2.1	新規プログラム名の入力	10-2
10.2.2	プログラムの入力	10-3
10.2.3	プログラムのコンパイル	10-7
10.2.4	プログラムのロード	10-9
第 11 章	プログラムの作成 (WINCAPSⅢ編)	11-1
11.1	プログラム作成の準備	11-1
11.1.1	WINCAPSⅢの種類	11-1
11.1.2	CD 外観 (CD レーベル)	11-1
11.1.3	製品版に同梱のライセンス証 (ユーザ ID)	11-2
11.1.4	WINCAPSⅢの製品種類を画面上で確認する	11-2
11.1.5	バージョンアップ時の注意	11-3
11.1.6	ユーザ ID の入力方法	11-3
11.2	作成するサンプルプログラム	11-4
11.3	プログラムの作成	11-5
11.3.1	WINCAPSⅢの起動	11-5
11.3.2	プロジェクトの新規作成	11-5
11.3.3	プログラムの作成	11-8
11.3.4	プログラムの入力と保存	11-10
11.3.5	プログラムのコンパイル	11-11
11.4	WINCAPSⅢとコントローラの通信ケーブル接続方法	11-13
11.4.1	RS-232C 接続	11-13
11.4.2	EtherNet 接続	11-13
11.5	ロボットコントローラとの通信接続準備	11-14
11.5.1	RS-232C 接続する場合	11-14
11.5.2	EtherNet 接続する場合	11-19
11.6	WINCAPSⅢによるデータ転送	11-26
11.6.1	コントローラ側の準備 (データ転送時の注意)	11-26
11.6.2	プログラムデータをロボットコントローラに送信する	11-27

第 4 部 プログラムの検証

第 12 章	プログラム起動	12-1
12.1	WINCAPSⅢによるシミュレーション	12-1
12.1.1	アーム 3D ビューを開く	12-1
12.1.2	WINCAPSⅢをモニタモードにする	12-1
12.1.3	マシンロック状態にする	12-2
12.1.4	プログラム起動を行う	12-2
12.2	ティーチチェックモードによるプログラム起動	12-3
12.2.1	ティーチチェックモードにする	12-3
12.2.2	プログラムを選択	12-4
12.2.3	ステップ送りチェック起動	12-4
12.2.4	サイクルチェック起動	12-6

12.3 内部自動モードによるプログラム起動	12-8
12.3.1 自動モードにする	12-8
12.3.2 プログラムを選択	12-8
12.3.3 ステップ起動.....	12-9
12.3.4 サイクル起動.....	12-10
12.3.5 連続起動	12-11
12.4 ロボットの停止操作	12-12
12.4.1 サイクル停止 [F3]	12-12
12.4.2 ステップ停止 [F2]	12-12
12.4.3 瞬時停止 [F1], [STOP]	12-12
12.4.4 非常停止 (ロボット停止)	12-13
第 13 章 外部機器からのロボット起動方法	13-1
13.1 I/O 割付モードの確認方法	13-1
13.2 グローバルタイプコントローラ使用時の注意	13-1
13.3 Mini I/O 専用モードによる起動方法.....	13-2
13.3.1 Mini I/O コマンド専用入力信号の種類と機能.....	13-2
13.3.2 Mini I/O コマンド処理方法.....	13-3
13.3.3 Mini I/O 専用出力信号の種類と機能	13-5
13.4 標準モードによる起動方法	13-6
13.4.1 標準 I/O コマンド専用入力信号の種類と機能.....	13-6
13.4.2 標準 I/O コマンド処理方法.....	13-7
13.4.3 標準 I/O 専用出力信号の種類と機能	13-9
13.5 互換モードによる起動方法	13-10
13.5.1 互換モード専用入力信号の種類と機能.....	13-10
13.5.2 互換モードの処理方法	13-11
13.5.3 互換 I/O 専用出力信号の種類と機能	13-13
13.6 I/O 割付表.....	13-14
13.6.1 HAND I/O (CN9): 全モード共通の割付.....	13-14
13.6.2 Mini I/O (標準タイプコントローラ CN5) : 専用割付.....	13-15
13.6.3 Mini I/O (グローバルタイプコントローラ CN5) : 専用割付.....	13-16
13.6.4 Mini I/O (標準タイプコントローラ CN5) : 互換/標準/全汎用割付.....	13-17
13.6.5 Mini I/O (グローバルタイプコントローラ CN5) : 互換/標準/全汎用割付.....	13-18
第 14 章 IO のモニタリングと操作	14-1
14.1 ティーチングペンダントによる操作.....	14-1
14.1.1 I/O のモニタリング.....	14-1
14.1.2 擬似 I/O の操作方法.....	14-2
14.2 WINCAPSⅢによる操作	14-4
14.2.1 I/O のモニタ	14-4
14.2.2 擬似 I/O を使う	14-5

第 15 章 変数のモニタリングと値変更	15-1
15.1 ティーチングペンダントによる操作.....	15-1
15.1.1 グローバル変数のモニタリングと値変更.....	15-1
15.1.2 ローカル変数のモニタリングと値変更.....	15-2
15.1.3 グローバル変数の使用個数変更.....	15-5
15.2 WINCAPSⅢによる操作.....	15-7
15.2.1 グローバル変数のモニタリングと値変更.....	15-7
15.2.2 ローカル変数のモニタリングと値変更.....	15-8
15.2.3 変数使用個数の変更.....	15-9

第 5 部 さらに高度な使い方

第 16 章 使用条件の最適化設定	16-1
16.1 ロボットの設置条件(床置き・天吊り)設定方法[6 軸ロボット].....	16-1
16.1.1 ロボット設置条件(床置、天吊)設定の目的.....	16-1
16.1.2 ティーチングペンダントによる設定.....	16-1
16.1.3 WINCAPSⅢによる設定.....	16-2
16.2 最適可搬質量設定機能.....	16-3
16.2.1 モード 0.....	16-3
16.2.2 モード 1.....	16-3
16.2.3 モード 2.....	16-4
16.2.4 モード 3.....	16-4
16.3 最適可搬質量設定機能の設定方法.....	16-5
16.3.1 ティーチングペンダントによる設定.....	16-5
16.3.2 WINCAPSⅢによる設定.....	16-6
16.4 最適可搬質量初期化設定の設定方法 [V1.4 以降].....	16-7
16.4.1 ティーチングペンダントによる設定.....	16-7
16.4.2 WINCAPSⅢによる設定.....	16-8
第 17 章 プログラム言語 (ロボット制御文)	17-1
17.1 ロボット動作の種類.....	17-1
17.1.1 絶対動作と相対動作.....	17-1
17.1.2 補間制御.....	17-1
17.2 ロボット制御コマンド.....	17-3
17.2.1 DR I V E A.....	17-3
17.2.2 DR I V E.....	17-4
17.2.3 DR A W.....	17-5
17.3 練習問題 ロボット制御コマンド.....	17-6
第 18 章 プログラミング言語 (フロー制御文)	18-1
18.1 フロー制御文の種類.....	18-1
18.2 呼び出しコマンド.....	18-2
18.2.1 C A L L.....	18-2
18.2.2 G O S U B.....	18-3

18.3 無条件分岐コマンド	18-4
18.3.1 GOTO	18-4
18.4 条件分岐コマンド	18-5
18.4.1 IF ~ ENDIF	18-5
18.4.2 SELECT CASE	18-6
18.5 繰り返しコマンド	18-7
18.5.1 FOR ~ NEXT	18-7
18.5.2 DO ~ LOOP	18-8
18.6 練習問題 フロー制御コマンド	18-10
第 19 章 プログラム言語（入出力制御文）	19-1
19.1 時間制御コマンド	19-1
19.1.1 DELAY	19-1
19.1.2 WAIT	19-1
19.2 I/O ポート制御	19-2
19.2.1 SET	19-2
19.2.2 RESET	19-2
19.3 練習問題 I/O ポート制御コマンド	19-3
第 20 章 プログラム言語（ライブラリ編）	20-1
20.1 ライブラリプログラムの活用	20-1
20.1.1 ライブラリプログラムとは	20-1
20.1.2 プログラムバンク	20-1
20.1.3 ライブラリの分類	20-1
20.1.4 ライブラリの追加（インポート）方法	20-2
20.2 パレタイジングライブラリの活用	20-4
20.2.1 パレタイジングの説明	20-4
20.2.2 簡易パレタイジングライブラリ	20-4
20.2.3 簡易パレタイジングプログラム「PRO1」	20-7

付録

練習問題の回答例

機能別コマンド一覧

ティーチングペンダントの機能メニュー

サンプルプログラム集

用語

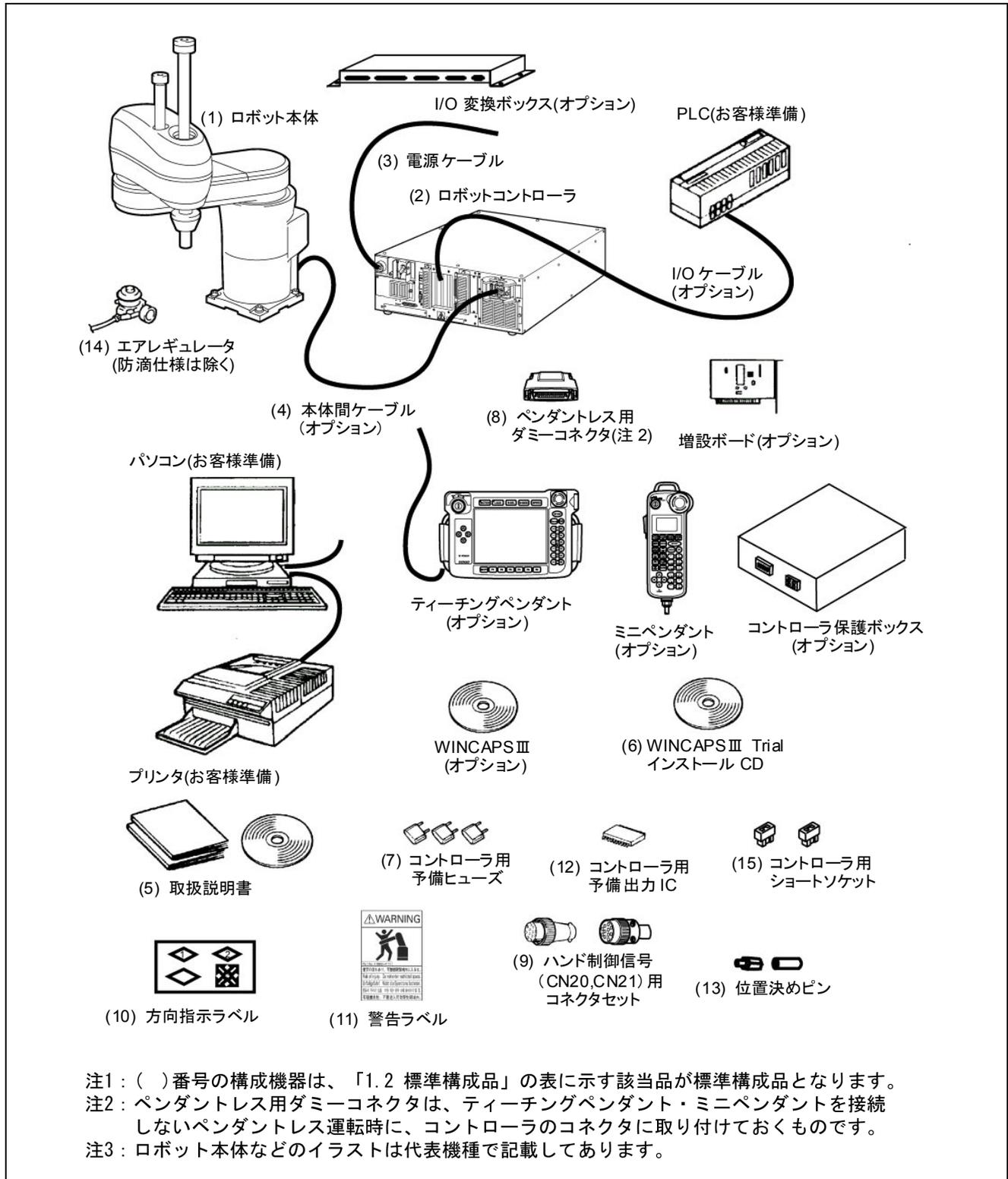
第 1 部

設置準備

第1章 ロボットシステムの構成

1.1 ロボットシステムの構成機器

ロボットシステムの全体構成を、下図に示します。



注1：() 番号の構成機器は、「1.2 標準構成」の表に示す該当品が標準構成となります。
 注2：ペンダントレス用ダミーコネクタは、ティーチングペンダント・ミニペンダントを接続しないペンダントレス運転時に、コントローラのコネクタに取り付けておくものです。
 注3：ロボット本体などのイラストは代表機種で記載してあります。

ロボットシステムの構成機器

1.2 標準構成

お問い合わせいただきました製品は、下表に示す内容で構成されています。

標準構成

No.	品名	数量	構成有無							
			HS シリーズ	HM シリーズ	XYC シリーズ	VP シリーズ	VS シリーズ	VH シリーズ	XR シリーズ	VS-*** シリーズ
①	ロボット本体	1台	○	○	○	○	○	○	○	○
②	ロボットコントローラ	1台	○	○	○	○	○	○	○	○
③	電源ケーブル (5m)	1本	○	○	○	○	○	○	○	○
④	本体間ケーブル (注1) (オプション品)	1セット	○	○	○	○	○	○	○	○
⑤	取扱説明書 (マニュアルパックCDと注意書)	1セット	○	○	○	○	○	○	○	○
⑥	WINCAPSⅢ Trial インストールCD	1枚	○	○	○	○	○	○	○	○
⑦	ロボットコントローラ用予備ヒューズ (1.3A×2ケ, 3.2A×1ケ)	3個	○	○	○	○	○	○	○	○
⑧	ペンダントレス用ダミーコネクタ (UL仕様は除く)	1個	○	○	○	○	○	○	○	○
⑨	ハンド制御信号用コネクタセット (CN20、CN21用)	1セット	○	○	○	○	○	○	○	option
⑩	方向指示ラベル (注2)	1枚	○	○	○	○	○	○	○	○
⑪	警告ラベル (注3)	1枚	○	○	○	○	○	○	○	○
⑫	ロボットコントローラ用予備出力用IC	1個	○	○	○	○	○	○	○	○
⑬	位置決めピン (めねじ付位置決めピン&ダイヤピン)	1セット	○	○	×	○	○	×	×	○
⑭	エアレギュレータ (注4)	1個	×	○	○	×	×	×	×	×
⑮	ロボットコントローラ用ショートソケット	2個	○	○	○	○	○	○	○	○

注1： 本体間ケーブル (モータ・エンコーダケーブル) はロボットと同時に下表から選んでご注文ください。
 本体間ケーブル20m (標準/防滴) は付加軸オプション付きコントローラおよびUL仕様ロボットには使用できません。
 本体間ケーブルの曲げR (内径) は200mm以上でお使いください。過度の曲げは断線の原因となります。

XYCシリーズ、VS-***シリーズを除くロボットの場合

品名	品番	備考
標準ケーブル 2m	410141-4400	標準仕様ロボット で使用
標準ケーブル 4m	410141-3611	
標準ケーブル 6m	410141-3621	
標準ケーブル 12m	410141-3631	
標準ケーブル 20m	410141-4440	
防滴ケーブル 2m	410141-4420	防塵・防滴仕様、 クリーン仕様ロボ ットで使用
防滴ケーブル 4m	410141-3681	
防滴ケーブル 6m	410141-3691	
防滴ケーブル 12m	410141-3701	
防滴ケーブル 20m	410141-4460	

XYCシリーズの場合

品名	品番
標準ケーブル 4m	410149-0960
標準ケーブル 6m	410149-0970

VS-***シリーズの場合 (注5)

品名	品番
本体間ケーブル RC7-B 2m	410141-4560
本体間ケーブル RC7-B 4m	410141-4570
本体間ケーブル RC7-B 6m	410141-4580
本体間ケーブル RC7-B 12m	410141-4590
本体間ケーブル RC7-B 20m	410141-4600

注2： 方向指示ラベルは設置終了後に本体の見やすい位置に貼ってご使用ください。

注3： 警告シールはロボットの安全柵等の良く見える位置に貼ってください。必要に応じて貼付け用のプレートをご準備ください。

注4： 防塵・防滴ロボットでは、Z軸バランスシリンダが無いいため、エアレギュレータは付属していません。

注5： VS-***シリーズの本体間ケーブルの曲げR (内径) はケーブル固定状態で33.8mm以上、ケーブルが可動する状態では225mm以上を確保してください。過度の曲げは断線の原因となります。また、本体間ケーブル RC7-B 20m は付加軸オプション付きコントローラには使用できません。

1.3 オプション品

下表に示すオプションを準備しておりますので、必要に応じてご購入ください。

オプション品(1)

分類	No.	品名	備考	品番
I/O ケーブル	1	標準I/O ケーブルセット	(8m) No. 1-1 とNo. 1-2 を含む	410149-0940
			(15m) No. 1-1 とNo. 1-2 を含む	410149-0950
	1-1	Mini I/O ケーブル (68ピン)	(8m)	410141-2700
			(15m)	410141-2710
	1-2	HAND I/O ケーブル (20ピン)	(8m)	410141-1740
			(15m)	410141-1750
	2	増設パラレルI/O用ケーブル (96ピン)	(8m)	410141-3050
			(15m)	410141-3060
	3	セーフティI/O ケーブル (36ピン) (グローバルタイプコントローラ用)	(8m)	410141-3580
			(15m)	410141-3590
操作用機器	4	ティーチングペンダント	(4m) ケーブル付き	410100-1572
			(8m) ケーブル付き	410100-1582
			(12m) ケーブル付き	410100-1592
	5	ミニペンダントキット (ケーブル、WINCAPSIII Lightを含む)	(4m) 日本語表記	410109-0392
			英語表記	410109-0402
			(8m) 日本語表記	410109-0412
	6	ペンダント延長ケーブル	英語表記	410109-0422
			(12m) 日本語表記	410109-0432
	7	WINCAPSIII	英語表記	410109-0442
			(8m) TP, MP用	410141-3711
8	パラレルI/Oボード	(8m) TP, MP用	410141-3721	
		言語共通 (日本語、英語、 ドイツ語、韓国語、 中国語)	410090-0980	
RC7M 用 増設ボード	8	コントローラに装着 出荷	NPNタイプI/O	410010-3320
			PNPタイプI/O	410010-3330
		ボード単品出荷 (補給用)	NPNタイプI/O	410010-3340
			PNPタイプI/O	410010-3350
	9	コントローラに装着 出荷	スレーブボード	410010-3370
			マスタボード	410010-3380
			マスタ&スレーブボード	410010-3390
		ボード単品出荷 (補給用)	スレーブボード	410010-3400
			マスタボード	410010-3410
			マスタ&スレーブボード	410010-3480
	10	CC-Link ボード	コントローラに装着出荷	410010-3430
			ボード単品出荷 (補給用)	410010-3440
11	コンベアトラッキングボード	コントローラに装着出荷	410010-3460	
		ボード単品出荷 (補給用)	410010-3470	

オプション品(2)

分類	No.	品名	備考	品番
オプション機能 (お客様手配の増設ボード用など)	12	RS232C 増設機能 ボードメーカー / 型式: (株)コンテック/COM-2P(PCI)H	コントローラ出荷時に機能追加	410006-0260
			ボード単品購入時に機能追加	410006-0270
	13	S-LINK V 機能 ボードメーカー / 型式: SUNX(株) / SL-VPCI	コントローラ出荷時に機能追加	410006-0280
			ボード単品購入時に機能追加	410006-0290
	14	PROFIBUS-DP スレーブ機能 ボードメーカー / 型式: Hilscher GmbH / CIF50-DPSADENSO	コントローラ出荷時に機能追加	410006-0300
			ボード単品購入時に機能追加	410006-0310
15	EtherNet/IP 機能 ボードメーカー / 型式: Hilscher GmbH / CIFX50-READENSO	コントローラ出荷時に機能追加	410006-0800	
		ボード単品購入時に機能追加	410006-0810	
16	メモリ増設	コントローラ出荷時のみ 3.25MB→5.5MBに増設可 (プログラム領域のみ増加)	410006-0320	
オプションボックス	17	コントローラ保護ボックス		410181-0091
	18	RC5 I/O 変換ボックス	RC5 型コントローラとの互換用	410181-0100
取扱説明書 (CD版)	19	マニュアルパック CD	(ロボットに標準同梱)	410002-2661
オプションの 印刷版 取扱説明書 (日本語版)	20-a	取扱説明書 HS-G 用フルセット(日本語版)	No. A-a と No. B-a を含む	410009-0350
	20-b	取扱説明書 HM-G 用フルセット(日本語版)	No. A-b と No. B-b を含む	410009-0330
	20-c	取扱説明書 VP-G 用フルセット(日本語版)	No. A-c と No. B-c を含む	410009-0310
	20-d	取扱説明書 VS-G 用フルセット(日本語版)	No. A-d と No. B-d を含む	410009-0290
	20-e	取扱説明書 VM-G 用フルセット(日本語版)	No. A-e と No. B-e を含む	410009-0270
	20-f	取扱説明書 XYC-4G 用フルセット(日本語版)	No. A-f と No. B-f を含む	410009-0420
	20-g	取扱説明書 XR-G 用フルセット(日本語版)	No. A-g と No. B-g を含む	410009-0860
	A-a	取扱説明書 HS-G 用基本セット(日本語版)	No. A-a-1 と No. A-2, No. A-3 を含む	410009-0250
	A-b	取扱説明書 HM-G 用基本セット(日本語版)	No. A-b-1 と No. A-2, No. A-3 を含む	410009-0230
	A-c	取扱説明書 VP-G 用基本セット(日本語版)	No. A-c-1 と No. A-2, No. A-3 を含む	410009-0210
	A-d	取扱説明書 VS-G 用基本セット(日本語版)	No. A-d-1 と No. A-2, No. A-3 を含む	410009-0190
	A-e	取扱説明書 VM-G 用基本セット(日本語版)	No. A-e-1 と No. A-2, No. A-3 を含む	410009-0170
	A-f	取扱説明書 XYC-4G 用基本セット(日本語版)	No. A-f-1 と No. A-2, No. A-3 を含む	410009-0400
	A-g	取扱説明書 XR-G 用基本セット(日本語版)	No. A-g-1 と No. A-2, No. A-3 を含む	410009-0840
	A-a-1	ロボット概要書(日本語版)	HS-G 用	410002-2600
	A-b-1	ロボット概要書(日本語版)	HM-G 用	410002-2560
	A-c-1	ロボット概要書(日本語版)	VP-G 用	410002-2520
	A-d-1	ロボット概要書(日本語版)	VS-G 用	410002-2480
	A-e-1	ロボット概要書(日本語版)	VM-G 用	410002-2440
	A-f-1	ロボット概要書(日本語版)	XYC-4G 用	410002-2760
	A-g-1	ロボット概要書(日本語版)	XR-G 用	410002-3200
	A-2	RC7M 型コントローラ説明書(日本語版)	RC7M 型コントローラ用	410002-2420
	A-3	エラーコード表(日本語版)		410002-3360
	B-a	取扱説明書 HS-G 用拡張セット(日本語版)	No. B-a-1 と No. B-2~No. B-7 を含む	410009-0130
	B-b	取扱説明書 HM-G 用拡張セット(日本語版)	No. B-b-1 と No. B-2~No. B-7 を含む	410009-0110
	B-c	取扱説明書 VP-G 用拡張セット(日本語版)	No. B-c-1 と No. B-2~No. B-7 を含む	410009-0090
	B-d	取扱説明書 VS-G 用拡張セット(日本語版)	No. B-d-1 と No. B-2~No. B-7 を含む	410009-0070
	B-e	取扱説明書 VM-G 用拡張セット(日本語版)	No. B-e-1 と No. B-2~No. B-7 を含む	410009-0050
	B-f	取扱説明書 XYC-4G 用拡張セット(日本語版)	No. B-f-1 と No. B-2~No. B-7 を含む	410009-0380
	B-g	取扱説明書 XR-G 用拡張セット(日本語版)	No. B-g-1 と No. B-2~No. B-7 を含む	410009-0820
	B-a-1	設置・保守ガイド(日本語版)	HS-G 用	410002-2620
	B-b-1	設置・保守ガイド(日本語版)	HM-G 用	410002-2580
	B-c-1	設置・保守ガイド(日本語版)	VP-G 用	410002-2540
	B-d-1	設置・保守ガイド(日本語版)	VS-G 用	410002-2500
	B-e-1	設置・保守ガイド(日本語版)	VM-G 用	410002-2460
	B-f-1	設置・保守ガイド(日本語版)	XYC-4G 用	410002-0780
	B-g-1	設置・保守ガイド(日本語版)	XR-G 用	410002-3220
	B-2	立上げハンドブック(日本語版)		410002-2740
	B-3	操作ガイド(日本語版)		410002-3300
	B-4	プログラミングマニュアル I(日本語版)		410002-3320
	B-5	プログラミングマニュアル II(日本語版)		410002-3340
	B-6	操作盤機能説明書(日本語版)		410002-6470
B-7	オプション機器説明書(日本語版)	RC7M 型コントローラ用	410002-2640	

オプション品(3)

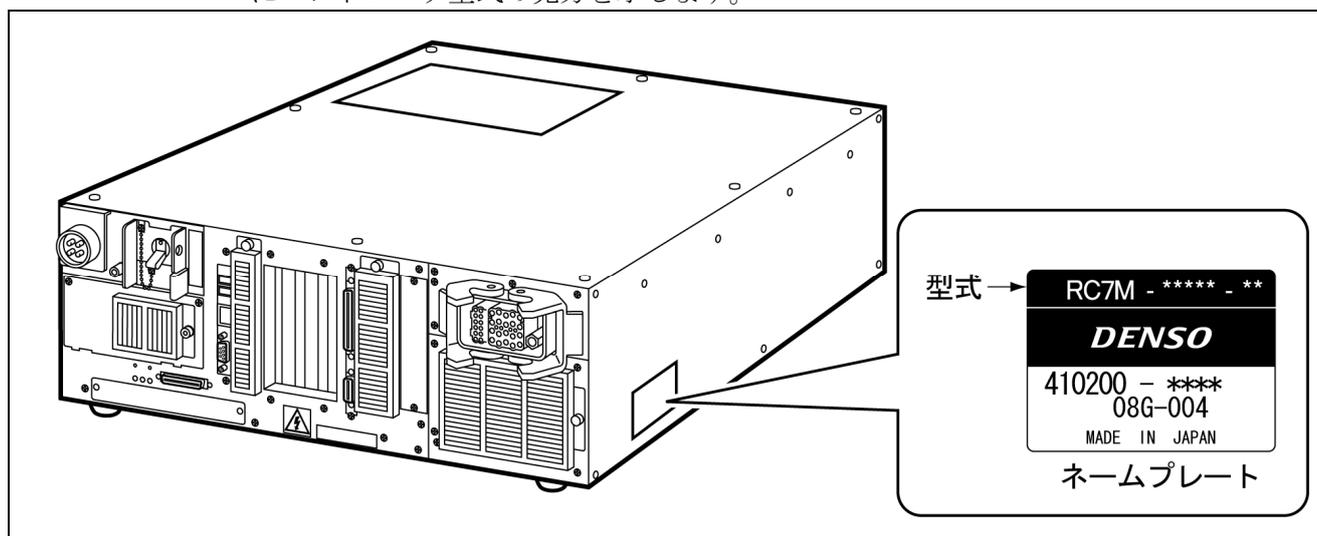
分類	No.	品名	備考	品番
オプションの 印刷版 取扱説明書 (英語版)	21-a	取扱説明書 HS-G用フルセット(英語版)	No. C-a と No. D-a を含む	410009-0360
	21-b	取扱説明書 HM-G用フルセット(英語版)	No. C-b と No. D-b を含む	410009-0340
	21-c	取扱説明書 VP-G用フルセット(英語版)	No. C-c と No. D-c を含む	410009-0320
	21-d	取扱説明書 VS-G用フルセット(英語版)	No. C-d と No. D-d を含む	410009-0300
	21-e	取扱説明書 VM-G用フルセット(英語版)	No. C-e と No. D-e を含む	410009-0280
	21-f	取扱説明書 XYC-4G用フルセット(英語版)	No. C-f と No. D-f を含む	410009-0430
	21-g	取扱説明書 XR-G用フルセット(英語版)	No. C-g と No. D-g を含む	410009-0870
	C-a	取扱説明書 HS-G用基本セット(英語版)	No. C-a-1 と No. C-2, No. C-3 を含む	410009-0260
	C-b	取扱説明書 HM-G用基本セット(英語版)	No. C-b-1 と No. C-2, No. C-3 を含む	410009-0240
	C-c	取扱説明書 VP-G用基本セット(英語版)	No. C-c-1 と No. C-2, No. C-3 を含む	410009-0220
	C-d	取扱説明書 VS-G用基本セット(英語版)	No. C-d-1 と No. C-2, No. C-3 を含む	410009-0200
	C-e	取扱説明書 VM-G用基本セット(英語版)	No. C-e-1 と No. C-2, No. C-3 を含む	410009-0180
	C-f	取扱説明書 XYC-4G用基本セット(英語版)	No. C-f-1 と No. C-2, No. C-3 を含む	410009-0410
	C-g	取扱説明書 XR-G用基本セット(英語版)	No. C-g-1 と No. C-2, No. C-3 を含む	410009-0850
	C-a-1	ロボット概要書(英語版)	HS-G用	410002-2610
	C-b-1	ロボット概要書(英語版)	HM-G用	410002-2570
	C-c-1	ロボット概要書(英語版)	VP-G用	410002-2530
	C-d-1	ロボット概要書(英語版)	VS-G用	410002-2490
	C-e-1	ロボット概要書(英語版)	VM-G用	410002-2450
	C-f-1	ロボット概要書(英語版)	XYC-4G用	410002-2770
	C-g-1	ロボット概要書(英語版)	XR-G用	410002-3210
	C-2	RC7M型コントローラ説明書(英語版)	RC7M型コントローラ用	410002-2430
	C-3	エラーコード表(英語版)		410002-3370
	D-a	取扱説明書 HS-G用拡張セット(英語版)	No. D-a-1 と No. D-2~No. D-7 を含む	410009-0140
	D-b	取扱説明書 HM-G用拡張セット(英語版)	No. D-b-1 と No. D-2~No. D-7 を含む	410009-0120
	D-c	取扱説明書 VP-G用拡張セット(英語版)	No. D-c-1 と No. D-2~No. D-7 を含む	410009-0100
	D-d	取扱説明書 VS-G用拡張セット(英語版)	No. D-d-1 と No. D-2~No. D-7 を含む	410009-0080
	D-e	取扱説明書 VM-G用拡張セット(英語版)	No. D-e-1 と No. D-2~No. D-7 を含む	410009-0060
	D-f	取扱説明書 XYC-4G用拡張セット(英語版)	No. D-f-1 と No. D-2~No. D-7 を含む	410009-0390
	D-g	取扱説明書 XR-G用拡張セット(英語版)	No. D-g-1 と No. D-2~No. D-7 を含む	410009-0830
	D-a-1	設置・保守ガイド(英語版)	HS-G用	410002-2630
	D-b-1	設置・保守ガイド(英語版)	HM-G用	410002-2590
	D-c-1	設置・保守ガイド(英語版)	VP-G用	410002-2550
	D-d-1	設置・保守ガイド(英語版)	VS-G用	410002-2510
	D-e-1	設置・保守ガイド(英語版)	VM-G用	410002-2470
	D-f-1	設置・保守ガイド(英語版)	XYC-4G用	410002-2790
	D-g-1	設置・保守ガイド(英語版)	XR-G用	410002-3230
	D-2	立上げハンドブック(英語版)		410002-2750
	D-3	操作ガイド(英語版)		410002-3310
	D-4	プログラミングマニュアルI(英語版)		410002-3330
	D-5	プログラミングマニュアルII(英語版)		410002-3350
D-6	操作盤機能説明書(英語版)		410002-6480	
D-7	オプション機器説明書(英語版)	RC7M型コントローラ用	410002-2650	
本体関連	22-a	フランジキット (HS用)	HS-G用	410329-0060
	22-b	フランジキット (HM用)	10kg可搬用	410329-0070
20kg可搬用			410329-0080	
ハンド配管、 配線用セット	22-g	バルブAssy (XR-G用)	単品出荷(補給用) 4連マニホールドバルブ	410640-0230
	23-g	バルブAssy (XR-G用)	ロボット装着出荷 4連マニホールドバルブ	410640-0330
	24-g	ハンド制御用ケーブルキット(XR-G用)		410879-0470
	25-g	ハンド制御用ケーブル(XR-G用)	2m	410870-3350
オプション スタンド	26	フルレンジスタンド(XR-G用)		411759-0010
	27	ハーフレンジスタンド(XR-G用)		411759-0020

第2章 RC7M型コントローラの概要

RC7M型コントローラは、搭載されるロボット型式により細部の仕様が異なります。この章では、RC7M型コントローラの概要について説明します。

2.1 コントローラ型式の見方

コントローラ型式は、コントローラ背面のネームプレートに記載されています。下表にコントローラ型式の見方を示します。



RC7M型コントローラ型式の見方(Gタイプロボット)

RC7M - <u>VSG</u> <u>6</u> <u>B</u> <u>A</u> - <u>B</u> <u>P</u>			
(a) (b) (c) (d) (e) (f) (g)			
記号	例	意味	分類
(a)	VSG	ロボット型名	VMG:VM-G用, VSG:VS-G用, VPG:VP-G用, HMG:HM-G用, HSG:HS-G用 XYCG:XYC-G用, XRG:XR-G用
(b)	6	軸数	4:4軸, 5/6:5軸または6軸、6:6軸
(c)	B	設計記号 1	A: エンコーダA用 B: エンコーダB用 C: エンコーダC用
(d)	A	設計記号 2	A: 24Vブレーキ
(e)		設計記号 3	ブランク: 200VAC電源 A: 100VAC電源
(f)	B	コントローラの タイプ (注)	ブランク: 標準タイプ (国内向け標準) B: グローバルタイプ (セーフティボード付) C: グローバルタイプ (セーフティボックス付) D: グローバルタイプ UL仕様 (セーフティボード付) E: グローバルタイプ UL仕様 (セーフティボックス付)
(g)	P	I/Oのタイプ	ブランクまたはN: NPNタイプI/O (国内向け標準) P: PNPタイプI/O

注: グローバルタイプのコントローラについては、「2.2 グローバルタイプコントローラについてのご注意」を参照してください。

2.1.1 VS-***シリーズ用 RC7M ロボットコントローラの型式

ロボットコントローラの型式は以下のとおりです。



ロボット型式名

VSA2	VS-068 / VS-087
VSA1	VS-050 / VS-060

供給電源

2	200V
---	------

駆動電源

N	標準
---	----

エンジンボード

N	標準
---	----

I/O

M	マイナスコモン (NPN)
P	プラスコモン (PNP)

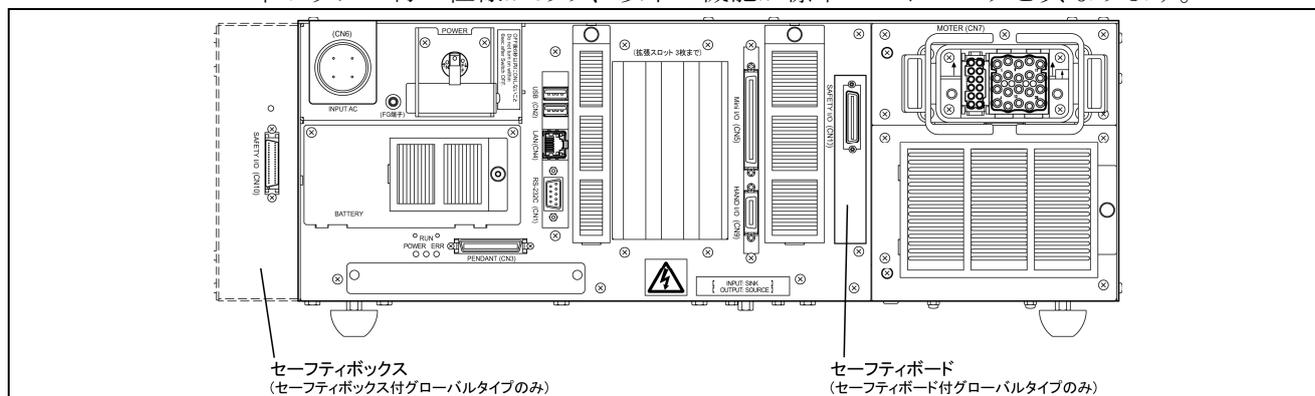
適合規格

NN	-
NB	CE (安全カテゴリ : 3、セーフティーボード付) 注 1
NC	CE (安全カテゴリ : 4、セーフティーボックス付) 注 1
UB	CE、UL (安全カテゴリ : 3、セーフティーボード付) 注 1
UC	CE、UL (安全カテゴリ : 4、セーフティーボックス付) 注 1

注 1 : グローバルタイプコントローラ

2.2 グローバルタイプコントローラについてのご注意

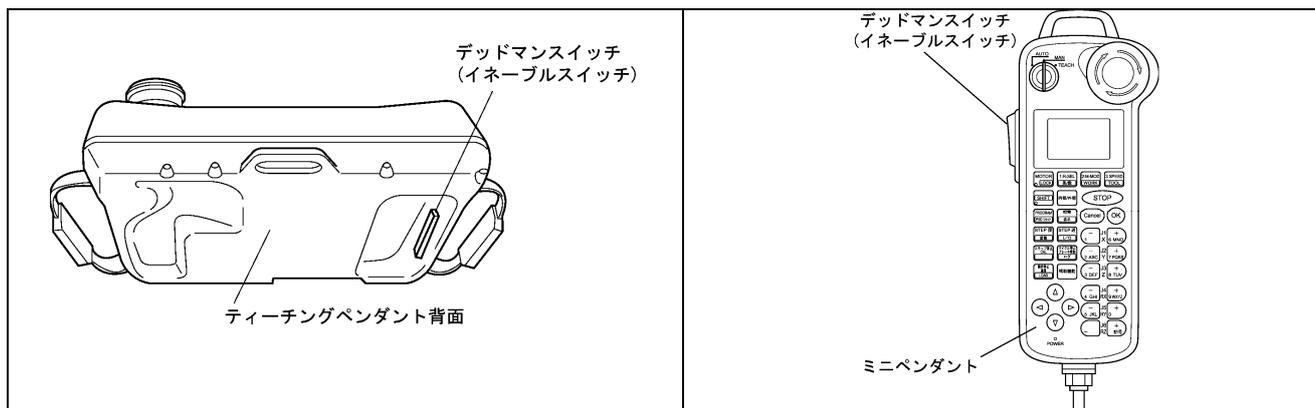
グローバルタイプコントローラには、下図のようにセーフティボードまたはセーフティボックス付の仕様があり、以下の機能が標準コントローラと異なります。



2.2.1 デッドマンスイッチ（イネーブルスイッチ）の機能について

「コントローラがグローバルタイプ」の場合、取扱説明書に記載の内容と操作ペンダントのデッドマンスイッチの機能が一部異なります。グローバルタイプの場合、取扱説明書の該当個所を読み替えてご使用ください。

(1) 対象オプション機器： ティーチングペンダント、ミニペンダント



(2) デッドマンスイッチの機能の相違点

「手動モード」・「ティーチチェックモード」で、ティーチングペンダント・ミニペンダントを操作する場合のデッドマンスイッチ機能の相違点を下表に示します。

グローバルタイプの場合	標準取扱説明書に記載の内容
① 「デッドマンスイッチ」を押していないと、ロボットの操作ができない。 また、「モータ電源」もONできない。	① 「デッドマンスイッチ」を押していないと、ロボットの操作ができない。 ただし、「モータ電源」はONできる。
② ロボット操作中に「デッドマンスイッチ」を離すと、ロボットが止まる。 また、「モータ電源」もOFFする。	② ロボット操作中に「デッドマンスイッチ」を離すと、ロボットが止まる。 ただし、「モータ電源」はOFFしない。 (サーボロック状態)

2.2.2 単一位置制御機能 (Single Point of control) について

グローバルタイプには単一位置制御機能 (Single Point of control) が追加されます。グローバルタイプ以外のロボットでは関係ありません。

(1) 単一位置制御機能とは

単一位置制御機能とはロボットを安全に操作するための機能のひとつで、ロボットを操作する機器をひとつに限定する機能です。具体的には自動モードがパラメータにより”内部自動”、”外部自動”に限定されます。

■ “内部自動” 限定モード

自動モードは“内部自動”モードに限定されます。ペンダントからのプログラム起動は可能ですが、外部機器からはできません。

■ “外部自動” 限定モード

自動モードは“外部自動”モードに限定されます。外部機器からのプログラム起動は可能ですが、ペンダントからはできません。

注意：“外部自動” 限定モードの場合、外部自動で「操作盤機能」は使用できません。

(2) モードの設定方法 (設定パラメータ)

ティーチングペンダントから以下のように設定します。

注1：工場出荷時は“外部自動” 限定モードに設定されています。

注2：グローバルタイプは、ティーチングペンダント画面に“A”の表示があります。

<操作経路>

[基本画面] → [F4：I/O] → [F6：補助機能] → [F1：ハード設定] → [F3：番号ジャン] → “31”
または【Ver. 2.3 以降】の場合、

[基本画面] → [F4：I/O] → [F6：補助機能] → [F4：単一制御]



第3章 インタフェースの概要

3.1 Mini I/O 入出力信号の種類とその概要

ロボットコントローラの入出力信号について説明します。
 入出力信号には、汎用入出力信号と、専用入出力信号があります。
 オプション品の I/O 増設ボードを使用しない場合、Mini I/O 専用割付モードとなり、Mini I/O(CN5)と HAND I/O(CN9)のコネクタを使用します。

3.1.1 Mini I/O 入出力信号の種類(標準タイプコントローラ使用時)

プログラム起動などの命令を I/O コマンドとしてコマンド実行用入力 7 点を使って指示します。専用入出力信号の種類を、下表に示します。

入出力信号の種類 (標準タイプコントローラ)

システム固定		
種類	点数	機能
専用入力	4点	外部非常停止入力1、外部非常停止入力2、自動イネーブル入力、ステップ停止(全タスク)
専用出力	13点 (注1)	自動モード、ロボット初期化完了、ロボット運転中、CPU正常、ロボット異常、運転準備完了、バッテリー切れ警告、非常停止出力1、非常停止出力2、デッドマンSW出力1 (イネーブルSW出力1)、デッドマンSW出力2 (イネーブルSW出力2)、ペンダント非常停止出力1、ペンダント非常停止出力2、コンティニュースタート許可 (I/Oハード設定にて設定した場合) (注1)
コマンド実行用入力	7点	コマンド (3ビット)、データ領域 (3ビット)、ストローブ信号
コマンド実行用出力	1点	コマンド処理完了
ユーザプログラムで制御		
種類	点数	機能
汎用入力	8点	INコマンド、IO[]変数で外部I/O状態を読み込むための入力 分析条件判断、条件成立待ち、外部からのデータ入力などに使用
汎用出力	8点 (注1)	SET、RESETコマンド等にて、プログラム実行中に外部機器へ信号を与えるための出力
ハンド入力	8点	INコマンド、IO[]変数で外部I/O状態を読み込むための入力 ハンドのチェック状態確認などに使用
ハンド出力	8点	SET、RESETコマンド等にて、外部機器に信号を与えるための出力 ハンドの開閉コントロールなどに使用

注1： CN5の端子No. 53 (ポート番号24) には、工場出荷時、汎用出力が割り付いていますが、I/Oハード設定で専用出力 (コンティニュースタート許可) に設定することができます。

3.1.2 Mini I/O 入出力信号の種類(グローバルタイプコントローラ使用時)

グローバルタイプコントローラの場合、非常停止関係の専用入出力がセーフティ I/O (CN10) に集約されています。従ってグローバルタイプコントローラ使用時は、Mini I/O (CN5) の非常停止関係の専用入出力は使用しません。

(RC7M 型コントローラ説明書の 4.1.3~4.1.4 項および 5.1.3~5.1.4 項を参照)

また、標準タイプコントローラと同様にプログラム起動などの命令を I/O コマンドとしてコマンド実行用入力 7 点を使って指示します。

専用入出力信号の種類を、下表に示します。

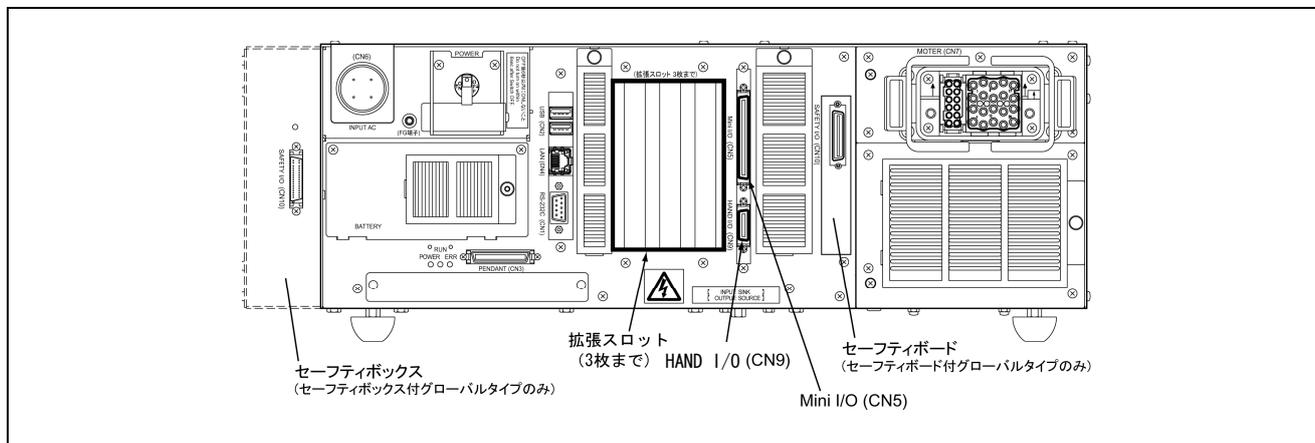
入出力信号の種類 (グローバルタイプコントローラ)

システム固定		
種類	点数	機能
専用入力	7 点	外部非常停止入力1、外部非常停止入力2、自動イネーブル入力1、自動イネーブル入力2、ステップ停止(全タスク)、防護停止入力1、防護停止入力2
専用出力	12 点 (注 1)	自動モード、ロボット初期化完了、ロボット運転中、CPU正常、ロボット異常、運転準備完了、バッテリー切れ警告、ペンダント非常停止出力1、ペンダント非常停止出力2、デッドマンSW出力1 (イネーブルSW出力1)、デッドマンSW出力2 (イネーブルSW出力2)、コンタクタ接点モニタ出力 コンティニュースタート許可 (I/Oハード設定にて設定した場合) (注1)
コマンド実行用入力	7 点	コマンド (3ビット)、データ領域 (3ビット)、ストローブ信号
コマンド実行用出力	1 点	コマンド処理完了
ユーザプログラムで制御		
種類	点数	機能
汎用入力	8 点	INコマンド、IO[]変数で外部I/O状態を読み込むための入力 分析条件判断、条件成立待ち、外部からのデータ入力などに使用
汎用出力	7 点 (注 1)	SET、RESETコマンド等にて、プログラム実行中に外部機器へ信号を与えるための出力
ハンド入力	8 点	INコマンド、IO[]変数で外部I/O状態を読み込むための入力 ハンドのチェック状態確認などに使用
ハンド出力	8 点	SET、RESETコマンド等にて、外部機器に信号を与えるための出力 ハンドの開閉コントロールなどに使用

注1： CN5の端子No. 53 (ポート番号24) には、工場出荷時、汎用出力が割り付いていますが、I/Oハード設定で専用出力 (コンティニュースタート許可) に設定することができます。

3.2 I/O 増設ボードの概要

RC7M 型コントローラにおいて、Mini I/O (CN5)と HAND I/O (CN9)の I/O 点数だけでは足りない場合、または、各種フィールドネットワークでロボットを制御したい場合に、I/O 増設ボードを使用します。RC7M 型コントローラには、3 個の拡張スロットが準備されており、I/O 増設ボードは最大 2 枚まで使用することができます。



3.2.1 I/O 増設ボードの種類

I/O 増設ボードには、当社がオプション品として準備している I/O オプションボードとお客様がボードを手配する I/O 推奨オプションボードがあります。I/O 増設ボードの種類を下表に示します。

注： RC5 型コントローラのリピータシステムの場合、オプション品の「I/O 変換ボックス」を使うと便利です。(オプション機器説明書「I/O 変換ボックス」を参照)

(1) I/O オプションボードの種類

ボード 手配区分	I/O オプションボード名	I/O オプションボードの品番	
		コントローラに 組付出荷	ボード単品出荷
オプション品 (当社に注文)	パラレルI/Oボード (NPNタイプ)	410010-3320	410010-3340
	パラレルI/Oボード (PNPタイプ)	410010-3330	410010-3350
	DeviceNetスレーブボード	410010-3370	410010-3400
	DeviceNetマスタボード	410010-3380	410010-3410
	DeviceNetマスタ・スレーブボード	410010-3390	410010-3480
	CC-Linkボード	410010-3430	410010-3440

(2) I/O 推奨オプションボードの種類

ボード 手配区分	I/O 推奨ボード名	メーカー (型式)	コントローラへ機能追加するライセンスの品番	
			機能追加済で出荷	お客様が機能追加の 操作を実施
推奨 オプション品 (お客様が手配)	PROFIBUS-DP スレーブボード	Hilscher GmbH 製 (CIF50-DPS\DENSO)	410006-0300	410006-0310
	S-LINK V ボード	SUNX (株) 製 (SL-VPCI)	410006-0280	410006-0290
	RS232C 増設ボード	(株)コンテック製 (COM-2P(PCI)H)	410006-0260	410006-0270
	EtherNet/IP Adapter ボード	Hilscher GmbH 製 (CIFX50-RE\DENSO)	410006-0800	410006-0810

3.3 I/O 増設ボードの組合せと割付モード

I/O 増設ボードは、最大 2 枚まで装着可能です。装着するスロット番号、順序に制約はありません。

下記に、装着可能な I/O 増設ボードの組合せおよび、選択可能な割付モードを下表に示します。

I/O増設ボードの組合せ表

組合せ No	I/O 増設ボード (最大 2 枚)			割付モード			
	増設 (1)	増設 (2)	増設 (3)	Mini I/O 専用	増設 (1)に割付		全汎用
					互換	標準	
0	—	—	—	○			
1	—	S-Link V	—	○			○
2	—	DeviceNetマスタ	—				○
3	—	DeviceNetマスタ	パラレルI/O				○
4	—	DeviceNetマスタ	S-Link V				○
5	パラレルI/O	—	—	○	○	○	
6	パラレルI/O	パラレルI/O	—	○	○	○	
7	パラレルI/O	S-Link V	—	○	○	○	○
8	DeviceNetスレーブ	—	—		○	○	
9	DeviceNetスレーブ	パラレルI/O	—		○	○	
10	DeviceNetスレーブ	S-Link V	—		○	○	○
11	DeviceNetマスタ&スレーブ	—	—		○	○	○
12	DeviceNetマスタ&スレーブ	パラレルI/O	—		○	○	○
13	DeviceNetマスタ&スレーブ	S-Link V	—		○	○	○
14	CC-Link	—	—		○	○	
15	CC-Link	パラレルI/O	—		○	○	
16	CC-Link	DeviceNetマスタ	—		○	○	○
17	CC-Link	S-Link V	—		○	○	○
18	PROFIBUS-DPスレーブ	—	—		○	○	
19	PROFIBUS-DPスレーブ	パラレルI/O	—		○	○	
20	PROFIBUS-DPスレーブ	DeviceNetマスタ	—		○	○	○
21	PROFIBUS-DPスレーブ	S-Link V	—		○	○	○
22	EtherNet/IP Adapter	—	—		○	○	
23	EtherNet/IP Adapter	パラレルI/O	—		○	○	
24	EtherNet/IP Adapter	DeviceNetマスタ	—		○	○	○
25	EtherNet/IP Adapter	S-Link V	—		○	○	○

注1： 割付モード欄に示す○印のうち、いずれか1つのモードが選択できます。

注2： I/O増設ボードは2枚まで増設できます。装着するスロット位置、順序に制約はありません。

注3： パラレルI/Oボードを2枚増設する場合は、拡張スロットの左側の拡張スロットを1枚目のボードと認識します。1枚目と2枚目では、割付けられるI/Oポート番号が異なります。

3.3.1 割付モード別 I/O 割付一覧

下表に「割付モード別各増設ボードの割付」一覧を示します。実際の割付表について「13.6 I/O 割付表」を参照してください。

注：「DeviceNet マスタ&スレーブ」ボードの割付は、「DeviceNet スレーブ」および「DeviceNet マスタ」の割付表を併用して参照してください。

割付モード別各増設ボードの割付

割付モード	CN5 と増設ボードの割付表	
	対象 I/O	割付表
Mini I/O 専用	CN5	「Mini I/O専用」
	増設(1)(2)(3)	各増設ボードの「全汎用」
互換	CN5	「Mini I/O 互換/標準/全汎用」
	増設(1)	各増設ボードの「互換」
	増設(2)(3)	各増設ボードの「全汎用」
標準	CN5	「Mini I/O 互換/標準/全汎用」
	増設(1)	各増設ボードの「標準」
	増設(2)(3)	各増設ボードの「全汎用」
全汎用	CN5	「Mini I/O 互換/標準/全汎用」
	増設(1)(2)(3)	各増設ボードの「全汎用」

注：増設(1)(2)(3)は、前ページの「I/O増設ボードの組合せ表」に記載のものを示します。

3.3.2 割付モード別 機能概要

割付モード別に入出力信号の機能が異なります。下表に「割付モード別の機能概要」一覧を示します。

割付モード別の機能概要

割付モード	概要
Mini I/O 専用	ビットの組合せで動作を指示しますが、標準割付に対し、機能がある程度削減されています。 Mini I/Oの領域はMini I/O専用割付が割付けられます。I/O増設ボードが装着されている場合は、I/O増設ボードの領域は全て汎用信号となります。
互換	プログラム起動などの機能がビット毎に意味付けられており、そのビットを立てることで動作を指示します。 I/O増設ボードの領域に専用割付の「互換」が割付きます。Mini I/Oの領域は全ポート汎用信号（CPU正常を除く）となります。
標準	プログラム起動などをビットの組合せ（I/Oコマンド）で指示します。機能が最も多い割付です。 I/O増設ボードの領域に専用割付の「標準」が割付きます。Mini I/Oの領域は全ポート汎用信号（CPU正常を除く）となります。
全汎用	I/O増設ボードの領域は全て汎用信号となり、MiniI/Oの領域は全ポート汎用信号（CPU正常を除く）となります。

3.4 互換・標準・全汎用モード選択時の Mini I/O の機能

Mini I/O 専用モード以外の I/O 割付モード（互換・標準・全汎用）を選択時、Mini I/O の専用入出力で占有しているポートは全て汎用入出力ポート（CPU 正常は除く）に変わります。

- 専用入力ポート番号 0 ～ 7（CN5 端子 No. 11～18）→汎用入力
- 専用出力ポート番号 17～23（CN5 端子 No. 46～52）→汎用出力

注意：専用出力信号「CPU正常」はポート番号16(CN5 端子No. 45)に残ります。

3.5 インタフェースに必要な設定

3.5.1 I/O 割付モードの設定

割付モードを「Mini IO 専用」・「互換」・「標準」・「全汎用」へ変更する場合、ティーチングペンダントまたは WINCAPSⅢを使用してパラメータの変更が必要になります。

変更方法については、「3.6 I/O 割付モードの設定方法」を参照してください。

注意：割付モードに「互換」または「標準」の選択が可能な I/O 増設ボードを “コントローラに組立出荷” にてご購入された場合、工場出荷時は「標準」モードに設定されています。

3.5.2 I/O 用電源(+24V DC)の設定

Mini I/O ボード（CN5）および パラレル I/O ボード（オプション）には、I/O 用電源(+24V DC)を外部電源仕様にするか内部電源仕様にするか選択できます。変更する場合、ジャンパーによる変更が必要になります。

変更方法については、「3.7 Mini I/O 用電源の設定方法」 および 「3.8 パラレル I/O 用電源の設定方法」を参照してください。

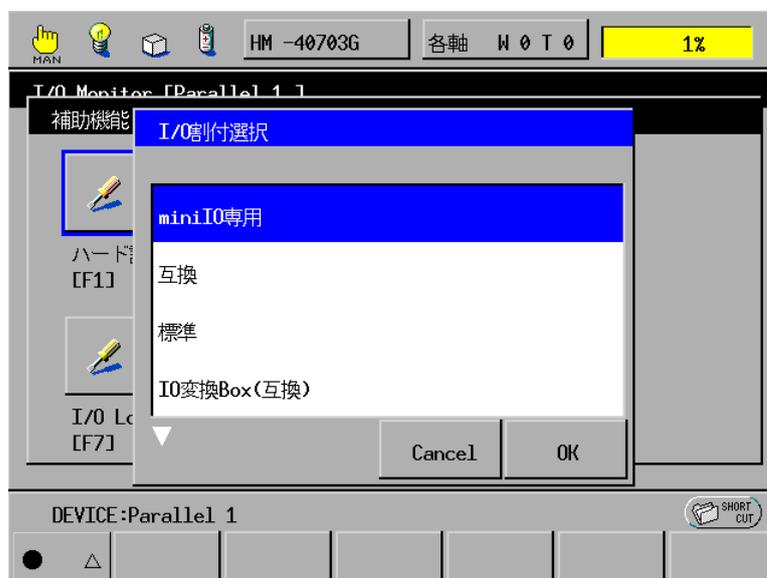
注意：工場出荷時は、外部電源に設定されています。

3.6 I/O 割付モードの設定方法

3.6.1 ティーチングペンダントからの設定方法

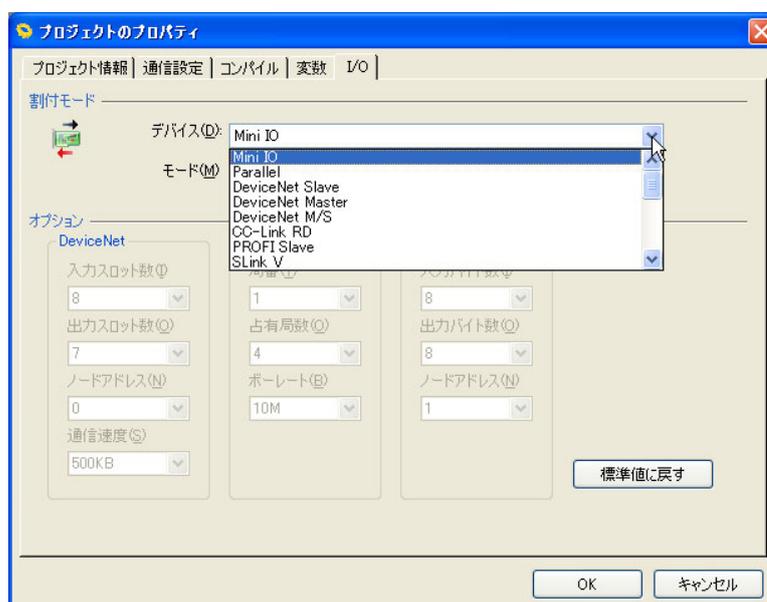
操作経路：[F4 I/O] — [F6 補助機能] — [F2 I/O 割付]

上記操作後に十字キーまたはジョグダイヤルにてカーソルを任意の割付にあわせ、[OK]を押して下さい。電源再投入後設定が有効となります。

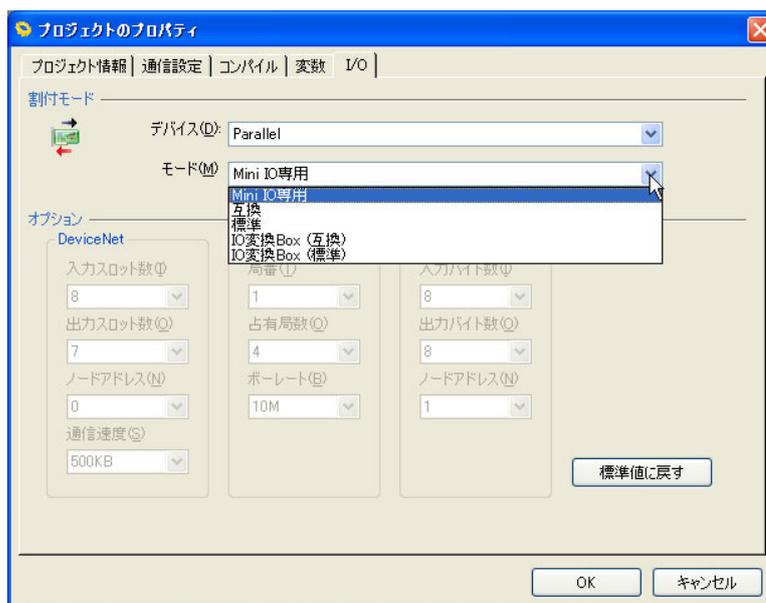


3.6.2 WINGCAPSⅢからの設定方法

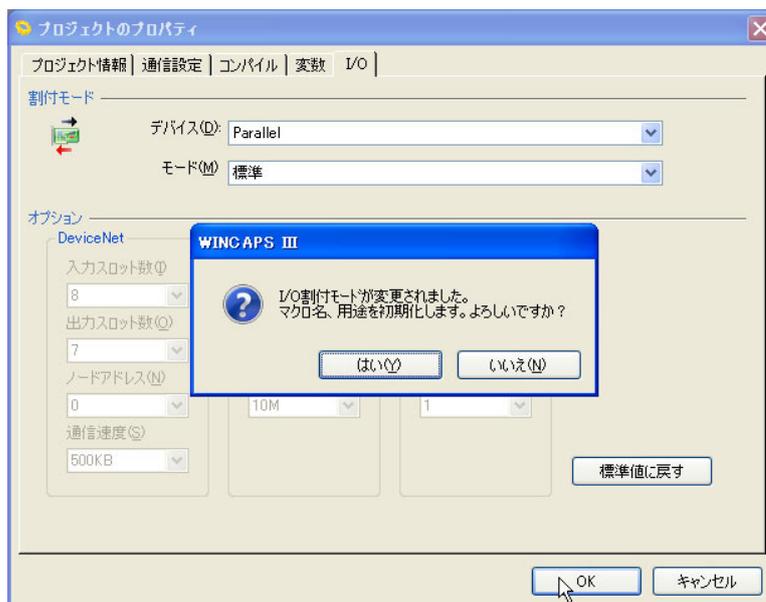
- (1) [プロジェクト (P)]-[プロパティ (P)]メニューを選択すると[プロジェクトのプロパティ]ダイアログボックスが表示されます。
[I/O]タブを選択します。
- (2) [割付モード]欄の[デバイス (D)]を選択し、ドロップダウンリストよりオプションの I/O ボードの種類を選択します。



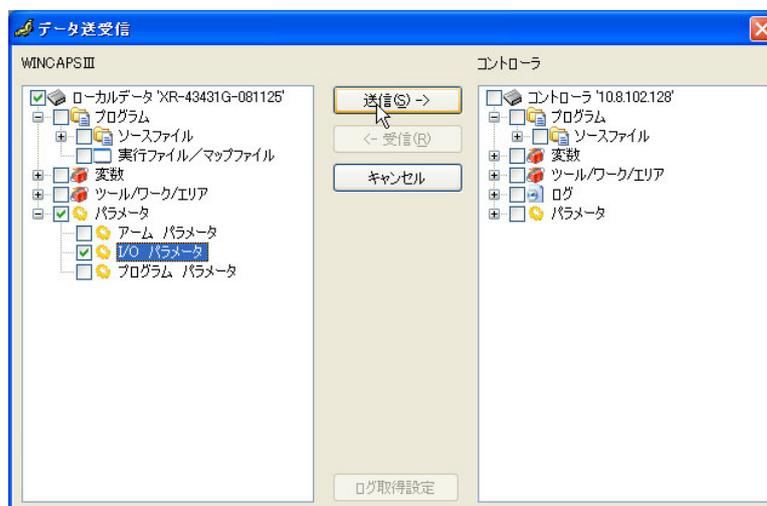
(3) [モード(M)]を選択し、ドロップダウンリストより割付を選択してください。



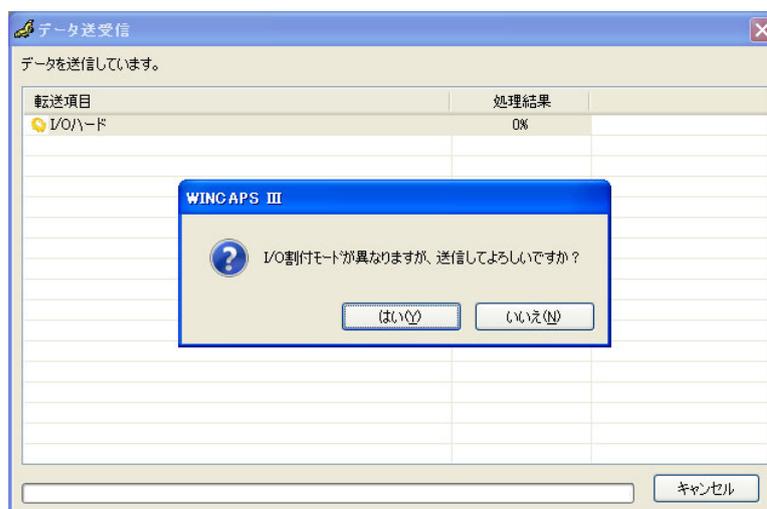
(4) [OK]を押すと、下記メッセージが現れます。〈用途〉および〈マクロ名〉が初期化されても良ければ[はい]を、そうでなければ[いいえ]を押してください。どちらを押しても、割付は変更されます。



- (5) [通信 (N)]-[データ送受信 (T)]を選択し、WINCAPSⅢからコントローラに I/O の割付設定を送信します。
[パラメータ]内の[I/O パラメータ]を選択し送信してください。



- (6) コントローラ内のデータが更新されることの確認メッセージと I/O の割付モードが異なる確認メッセージが表示されます。両方で[はい (Y)]を選択するとコントローラに送信されます。
コントローラを再起動すると送信されたデータが反映されます。



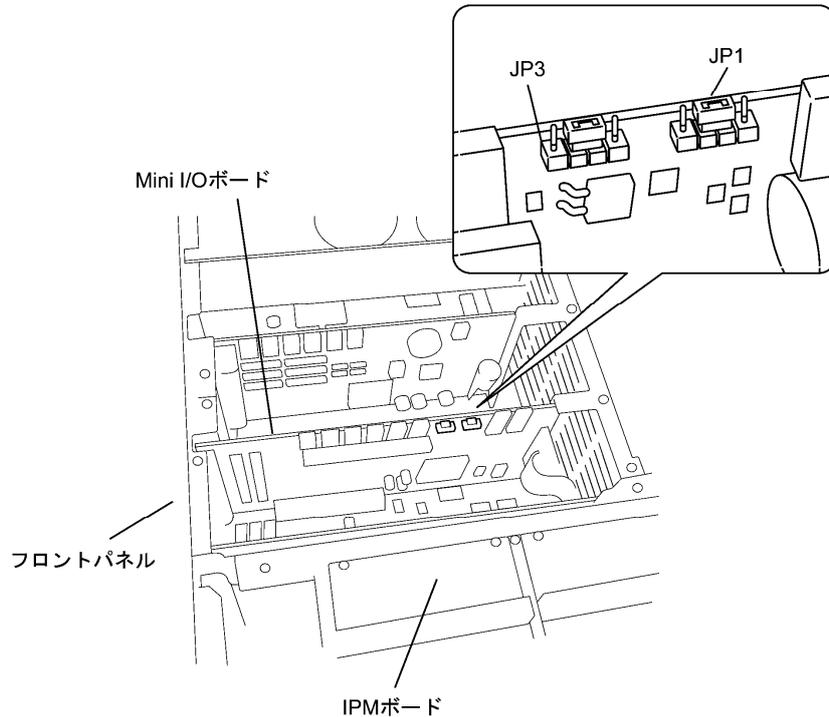
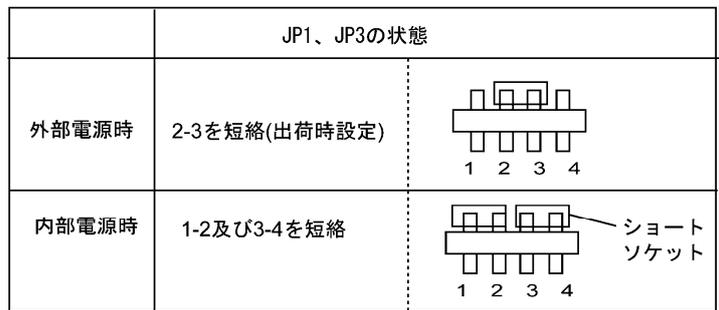
注意：装着されていないI/Oオプションボードを選択してコントローラに割付データを転送しないでください。転送後の電源再投入でエラーが発生します。

3.7 Mini I/O 用電源の設定方法

Mini I/O用電源(+24V DC)を外部電源にするか内部電源にするかを、コントローラのジャンパー設定で下表のように変更することができます。工場出荷時は、外部電源設定になっています。

Mini I/O用電源の設定

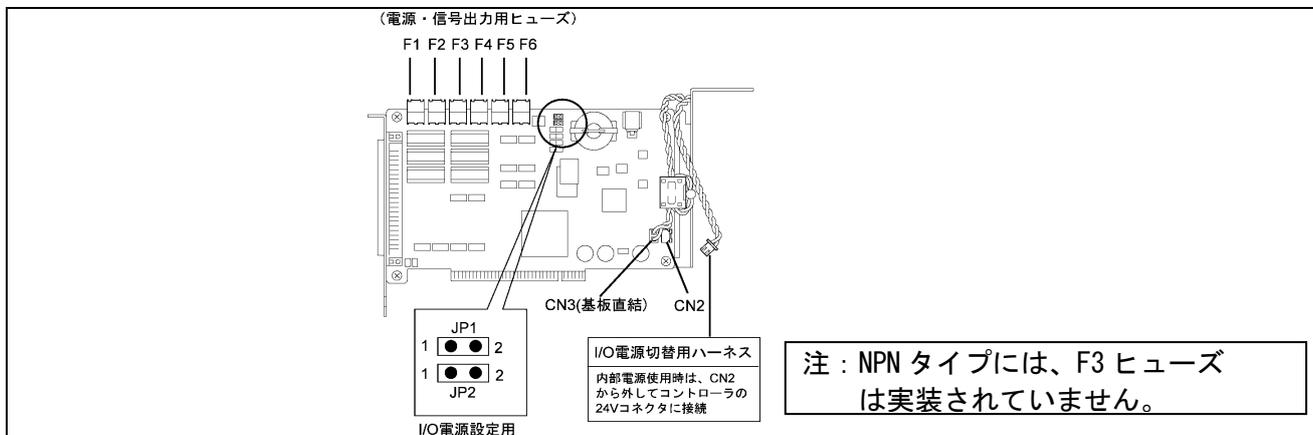
I/O用電源の設定	JP1、JP3 設定	設定方法
外部電源	2-3を短絡	工場出荷時の状態で、ご使用ください。
内部電源	1-2及び3-4を短絡	コントローラの上蓋を外し、ロボット付属品のショートソケットを使用して、下図のJP1とJP3のジャンパー設定を変更します。



注： I/O 用電源の設定を外部電源から内部電源に変更すると、CN5 の端子 No. 32~34 および端子 No. 66~68 は、外部 DC 電源入力から内部 DC 電源出力に変更されます。

3.8 パラレル I/O 用電源の設定方法

パラレルI/Oボードは、I/O用電源(+24V DC)を外部電源にするか内部電源にするかの設定ができます。工場出荷時は、外部電源設定になっています。パラレルI/Oボード各部の名称を下図に示します。



I/O用電源の設定	JP1、JP2設定	設定方法
外部電源	 JP1, JP2 (オープン)	工場出荷時の状態 (JP1, JP2ともオープン) で、ご使用ください。
内部電源	 JP1, JP2 (ショート)	<p>(1) ショートソケットで、「JP1の1-2」および「JP2の1-2」をショートします。</p> <p>(2) パラレルI/OボードのI/O電源切替用ハーネスのCN2に接続されているコネクタを外して、コントローラ本体の24Vコネクタ (CNP101) に接続してください。</p> <p>フロントパネル側</p> <p>コントローラに装着した パラレルI/Oボード</p> <p>コントローラの 24Vコネクタ(CNP101) に接続</p> <p>(3) パラレルI/Oボードを2枚装着して内部電源を使用するときは、2枚目のボードのI/O電源切替用ハーネスを1枚目のボードのCN2に接続してください。</p> <p>1枚目のボード</p> <p>2枚目のボード</p> <p>I/O電源切替用ハーネス</p> <p>CN3 CN2 CNP.101 (コントローラの24Vコネクタ)</p>

注意：設定は必ずコントローラの電源がOFFの状態で行なってください。

3.9 I/O ポートマップと割付

I/O 増設ボードを使用しないときは I/O ポート番号 (PAC プログラムまたは I/O コマンドで I/O 処理を行うときに指定する番号) は 511 までですが、I/O 増設ボード (オプション品) を使用すると I/O ポート番号が 512 以降に追加されます。

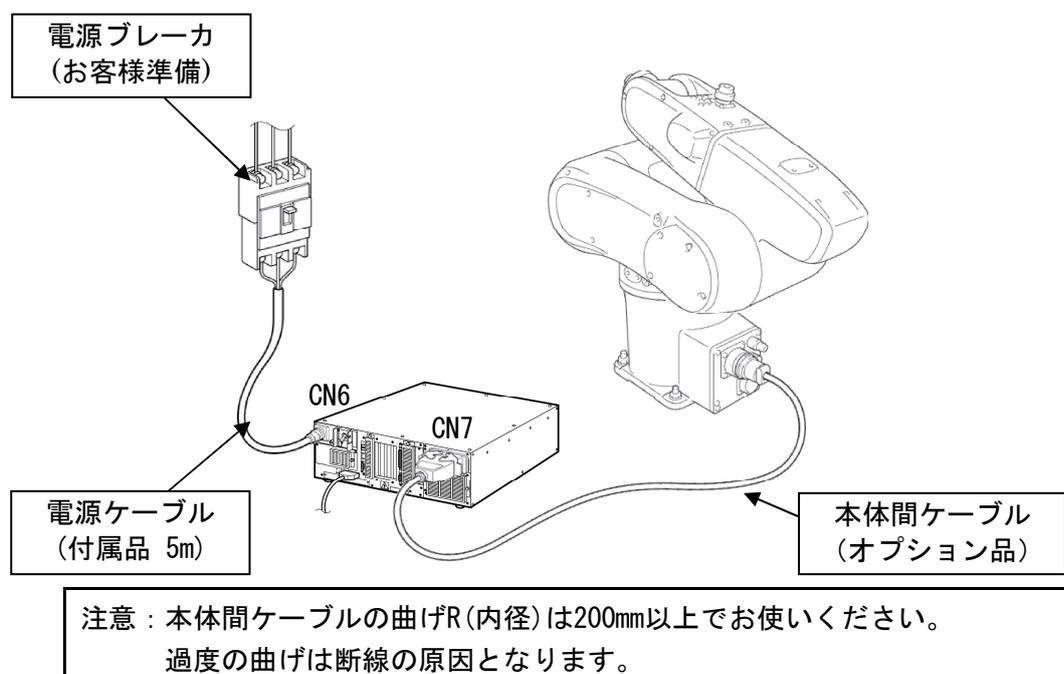
I/Oポートマップと割付

I/O ポート番号	割付	領域
0 ~ 15	Mini I/O 入力	標準領域
16 ~ 31	Mini I/O 出力	
32 ~ 47	未使用	
48 ~ 55	ハンド 入力	
56 ~ 63	未使用	
64 ~ 71	ハンド 出力	
72 ~ 127	未使用	
128 ~ 511	内部I/O	
512 ~ 767	DeviceNet スレーブ 入力 CC-Link 入力 PROFIBUS-DPスレーブ 入力 EtherNet/IP Adapter 入力	I/O増設 (オプション) ボード領域
768 ~ 1023	DeviceNet スレーブ 出力 CC-Link 出力 PROFIBUS-DPスレーブ 出力 EtherNet/IP Adapter 出力	
1024 ~ 2047	DeviceNet マスタ 入力	
2048 ~ 3071	DeviceNet マスタ 出力	
3072 ~ 3327	S-Link V 入力	
3328 ~ 3583	S-Link V 出力	
3584 ~ 3623	パラレルボード(1枚目) 入力	
3624 ~ 3663	パラレルボード(2枚目) 入力	
3664 ~ 3839	未使用	
3840 ~ 3887	パラレルボード(1枚目) 出力	
3888 ~ 3935	パラレルボード(2枚目) 出力	
3936 ~ 4095	未使用	
4096 ~ 4351	CC-Linkリモートレジスタ (RWw) 入力	
4352 ~ 4607	未使用	
4608 ~ 4863	CC-Linkリモートレジスタ (RWr) 出力	
4096 ~ 7871	EtherNet/IP Adapter 入力	
7872 ~ 11647	EtherNet/IP Adapter 出力	

第4章 ケーブル類の接続

4.1 電源・本体間ケーブルの接続

コントローラへの供給電源は付属品の電源ケーブル（5m仕様）をご使用ください。ロボット本体とコントローラ間は、オプション品の本体間ケーブル（2m、4m、6m、12m、20mから選択）で接続してください。

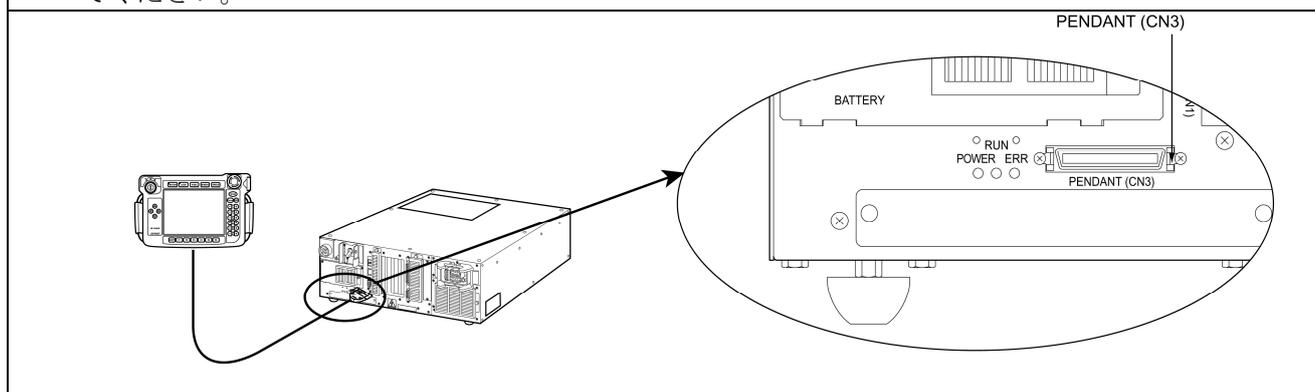


4.2 ペンダントケーブルの接続

ティーチングペンダントは、下図に示すようにロボットコントローラの PENDANT コネクタ (CN3) に接続してください。

ペンダントケーブルをコントローラへ接続するときの注意

- (1) ケーブル接続後は、コネクタ部（下図の○部）の上下または左右方向に外力を加えないでください。コネクタ破損による通信異常の原因になります。
- (2) ケーブルを取り外すときは、コネクタのロックを外して、コネクタをこじらずにまっすぐ引き抜いてください。



ティーチングペンダントの接続

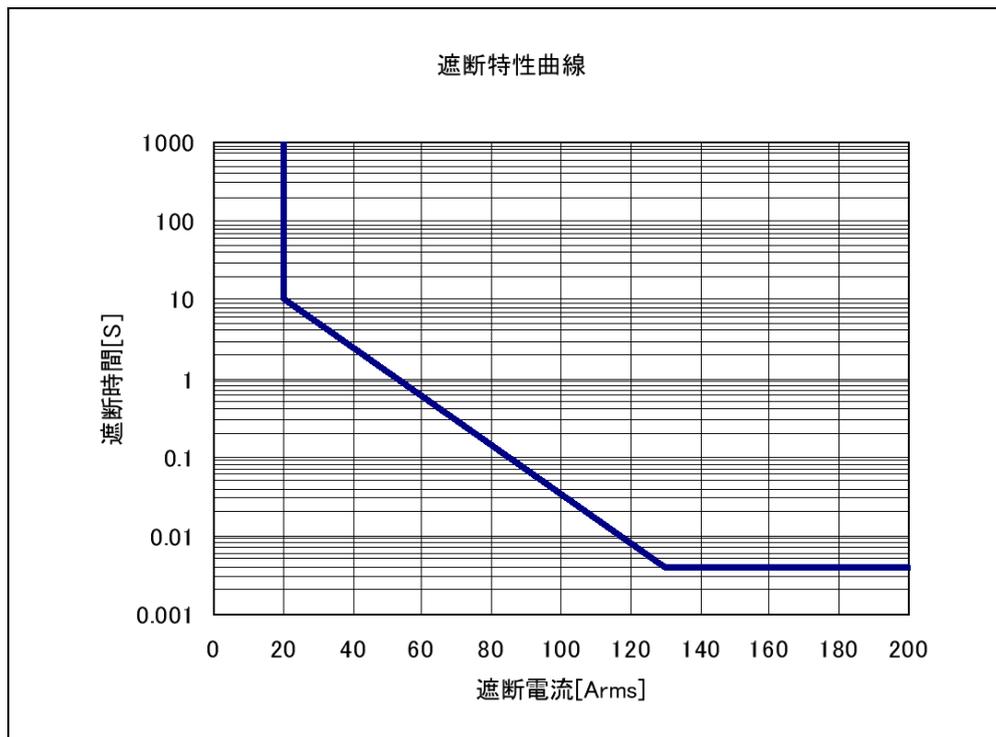
4.3 ブレーカの選定（推奨品）

ロボットコントローラの1次側電源の配線に際しては、以下のことがらに注意してください。

- (1) ロボット用電源は溶接用電源とは必ず別電源から配線してください。
- (2) ロボット用電源ケーブルのアース線（保護接地）（緑/黄）は、確実に接続してください。
- (3) ロボットコントローラのアースターミナル（機能接地）は、 1.25mm^2 以上の配線で接地してください。
- (4) ロボット用電源のアース（保護接地）は、D種接地（接地抵抗 100Ω 以下）にしてください。
- (5) ロボットコントローラへの供給電源側に漏電ブレーカを使用する場合は、インバータ用として高周波対策を施したものを使用してください。
- (6) AC入力電源にブレーカを接続する場合は、下図の遮断特性以上のブレーカを選定してください。

推奨ブレーカの例：CP33V/20（富士電機機器制御株式会社）

注意：下図の遮断特性以下のブレーカを接続すると、ロボット動作によりブレーカが遮断する場合があります。



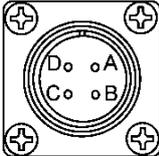
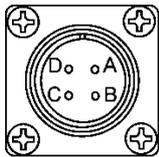
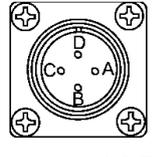
ブレーカ

4.4 1 次側電源の配線方法

ロボットコントローラの1次側電源の配線に際しては、以下のことがらに注意してください。

- (1) ロボット用電源は溶接用電源とは必ず別電源から配線してください。
- (2) ロボット用電源ケーブルのアース線（保護接地）（緑/黄）は、確実に接続してください。
- (3) ロボットコントローラのアースターミナル（機能接地）は、1.25mm²以上の配線で接地してください。
- (4) ロボット用電源のアース（保護接地）は、D種接地（接地抵抗100Ω以下）にしてください。
- (5) ロボットコントローラへの供給電源側に漏電ブレーカを使用する場合は、インバータ用として高周波対策を施したものを使用してください。
- (6) 電源ケーブルは下表を参考に適切な容量のものを準備してください。

ロボットコントローラの電源仕様

項目		仕様	電源コネクタ (CN6) のピン配列	
電源 電圧	3相 AC200V 仕様	3相AC200V-15%~AC230V+10%、50/60Hz	 <p>A: AC200V R相 B: AC200V S相 C: AC200V T相 D: アース (保護接地)</p>	
	電源容量	VMG6BA : 3.3kVA		VSG6BA : 1.85kVA
		VPG5/6CA : 1kVA		HMG4BA : 2.45kVA
		HSG4BA : 1.8kVA		XYCG4AA : 1.15kVA
		XRG4BA : 1.8kVA		VS-050/060 : 1.15kVA
			VS-068/087 : 2.78kVA	
	单相 AC200V 仕様	单相 AC230V-10%~AC230V+10%、50/60Hz	 <p>A: AC200V R相 B: AC200V S相 D: アース (保護接地)</p>	
	電源容量			VSG6BA : 1.85kVA
		VPG5/6CA : 1kVA		HMG4BA : 2.45kVA
HSG4BA : 1.8kVA		XYCG4AA : 1.15kVA		
XRG4BA : 1.8kVA		VS-050/060 : 1.15kVA		
		VS-068/087 : 2.78kVA		
单相 AC100V 仕様	单相 AC100V-10%~AC110V+10%、50/60Hz	 <p>A: AC100V R相 B: AC100V S相 D: アース (保護接地)</p>		
電源容量	VPG5/6CAA : 1kVA			
電源投入時の最大瞬間電流		40A (1/50秒または1/60秒)		

注意：ロボット動作時に、ERROR6102（電源電圧低下）が発生する場合は、1次側電源の容量不足が原因の一つとして考えられます。

- (7) ティーチングペンダントケーブル・入出力ケーブル・本体間ケーブルなどのケーブルと電源ケーブル・周辺機器などの強電線とを束ねたり、モータケーブル線を強電機器（モータ・溶接機・パーツフィーダなど）の近くに付設したりしないでください。
- (8) ロボット本体内には新規にハンド用のケーブル・エアチューブなどを通さないでください。ロボット用モータ線・エンコーダ線の断線の原因となります。
- (9) コントローラの電源仕様（AC200V または AC100V 仕様）に適合した電源を必ず接続してください。

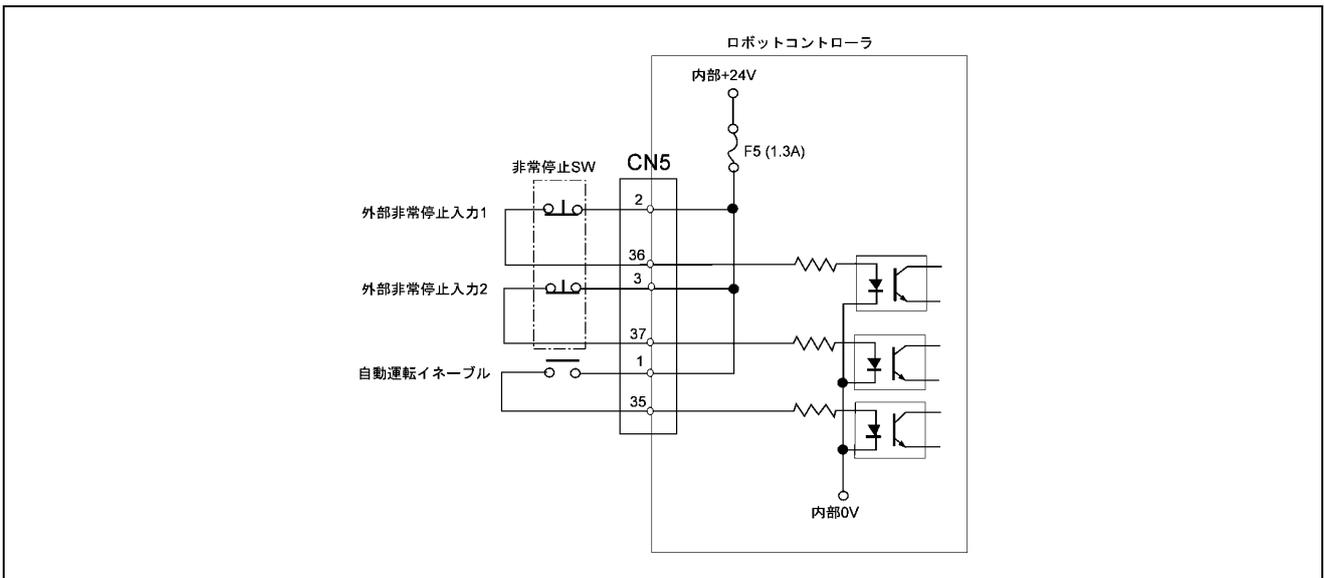
第5章 専用入力信号の結線

5.1 ロボット調整時に必要となる結線

ロボット立上げ調整時に、ロボット単体でモータ ON、自動運転、手動運転操作をおこなうために最低限必要となる結線について説明します。

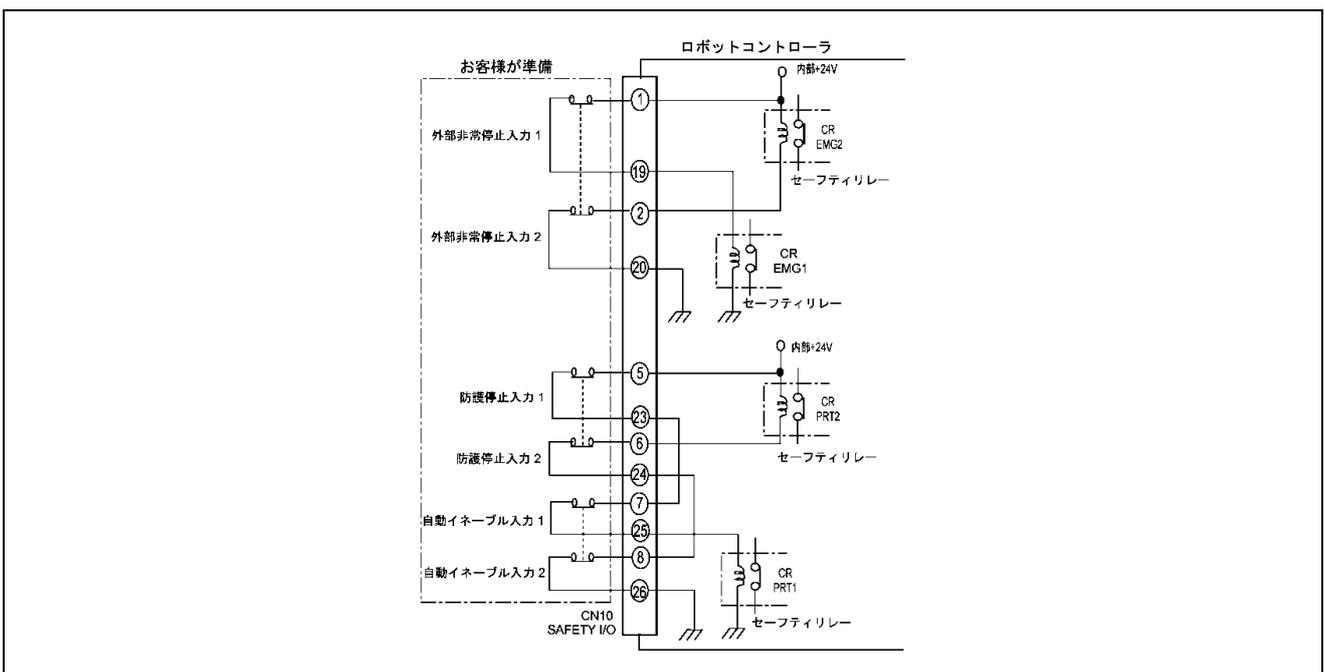
5.1.1 非常停止回路の構成（標準タイプコントローラ使用時）

外部非常停止と自動イネーブルの信号は、安全のために重要な信号です。下図に示すように、必ず有接点の回路で構成してください。



5.1.2 非常停止回路の構成（グローバルタイプコントローラ）

安全回路の入力信号は、安全のために重要な信号です。以下の点注意して、必ず有接点の回路で構成してください。



注：安全回路の全体構成については、RC7M 型コントローラ説明書の「4.2.5.2 安全回路」を参照してください。

5.2 モータ入りに必要な結線

5.2.1 機能

非常停止入力2系統とも短絡状態の場合に限り、モータ ON が可能になります。

5.2.2 標準タイプコントローラ

信号名	端子番号
外部非常停止入力1	コネクタCN5のNo. 2と36
外部非常停止入力2	コネクタCN5のNo. 3と37

[注意]非常停止2系統の状態が約1秒以上異なる場合には、エラー279E（ロボット停止入力の1系統が入力されません）が発生します。

5.2.3 グローバルタイプコントローラ

信号名	端子番号
外部非常停止入力1	コネクタCN10のNo. 1と19
外部非常停止入力2	コネクタCN10のNo. 2と20

[注意]2つの外部非常停止入力は別々の接点で制御して下さい。1つの接点での「2回路並列接続」または「1回路常時短絡」は、外部回路異常と判断し非常停止が解除できません。

5.3 自動運転に必要な結線

5.3.1 機能

- ①短絡状態でロボットを自動モードへ切替可能にします。
- ②開放状態でロボットを手動モード、ティーチチェックモードへ切替え可能にします。

5.3.2 標準タイプコントローラ

信号名	端子番号
自動イネーブル入力	コネクタCN5のNo. 1と35

5.3.3 グローバルタイプコントローラ

信号名	端子番号
自動イネーブル入力1	コネクタCN10のNo. 7と25
自動イネーブル入力2	コネクタCN10のNo. 8と26
防護停止入力1	コネクタCN10のNo. 5と23
防護停止入力2	コネクタCN10のNo. 6と24

[注意]①2つの自動イネーブル入力と2つの防護低し入力は別々の接続で制御して下さい。1つの接点での「2回路並列接続」または「1回路常時短絡」は、外部回路異常と判断し動作できません。

- ②自動イネーブル入力と防護停止入力はロボットコントローラ内で直列に接続されています。自動モード切替許可信号として、2種類の入力が可能です。
- ③防護停止を必要としない場合は、コネクタCN10のNo. 5と23、No. 6と24をジャンパ接続し、常時短絡としても使用できます。

第2部

ロボット操作

第6章 座標系

6.1 4軸ロボットの座標系

ロボットを操作するときの座標系には、次の3種類があります。

- ・ベース座標系
- ・ワーク座標系
- ・ツール座標系

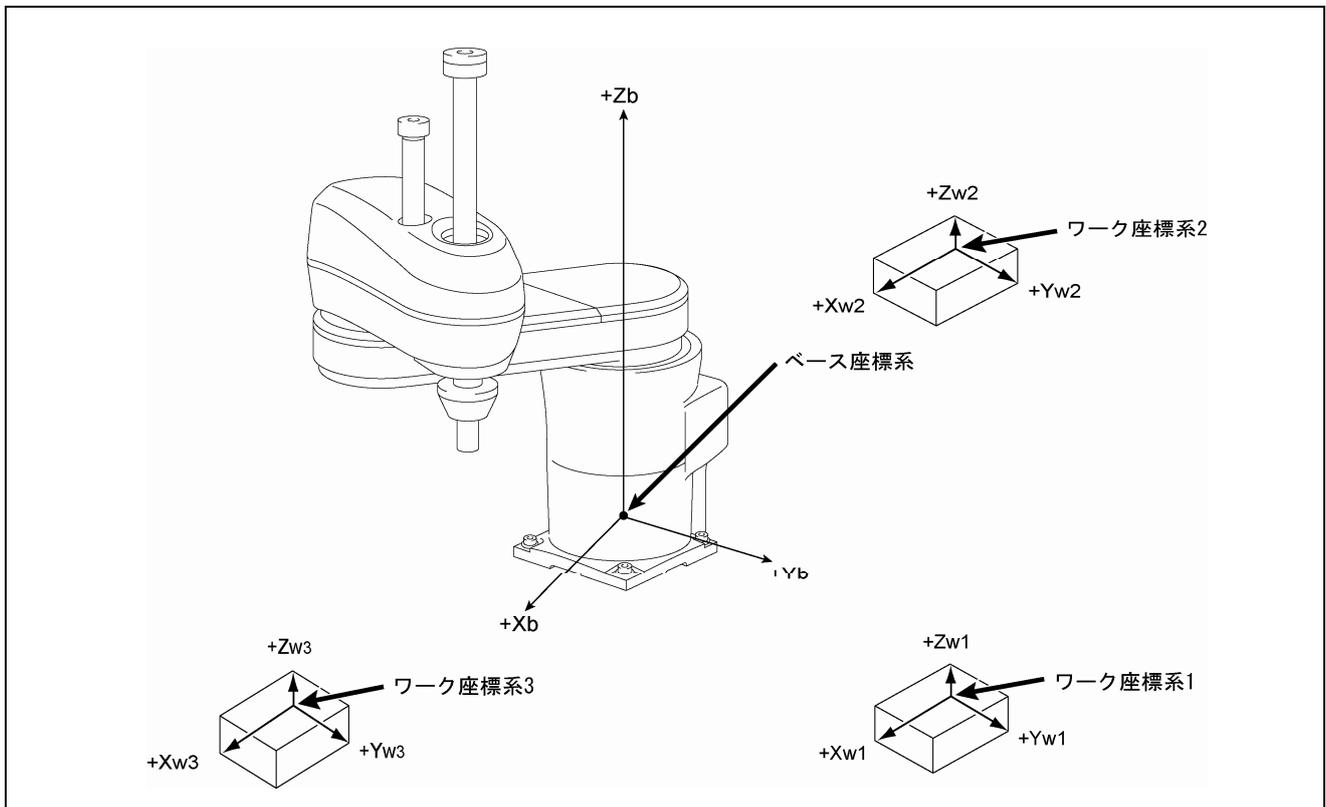
6.2 4軸ロボットのベース座標系

ベース座標系は、いわゆるワールド座標系であり、3次元直交座標です。その原点はロボットのベース中心です。ベース座標系のX軸を X_b 、Y軸を Y_b 、Z軸を Z_b と表現します。

6.3 4軸ロボットワーク座標系

ワーク座標系は、各軸がベース座標系の各軸と平行な3次元直交座標系として定義されます。ワーク座標系は、ロボットが操作しようとする対象のワークの作業空間毎に定義される3次元直交座標です。下図に示すように、その原点は対象ワークの包絡直方体の一つの頂点となります。ワーク座標はベース座標に対する座標原点位置(X, Y, Z)とベース座標のX軸・Y軸・Z軸周りの回転角(R_x, R_y, R_z)で表現します。ワーク座標は最大7種類の登録ができ、1~7のワーク座標番号を付けます。ワーク座標未定義の場合は、ベース座標となります。原点は、任意の場所に任意の数だけ定義できます。

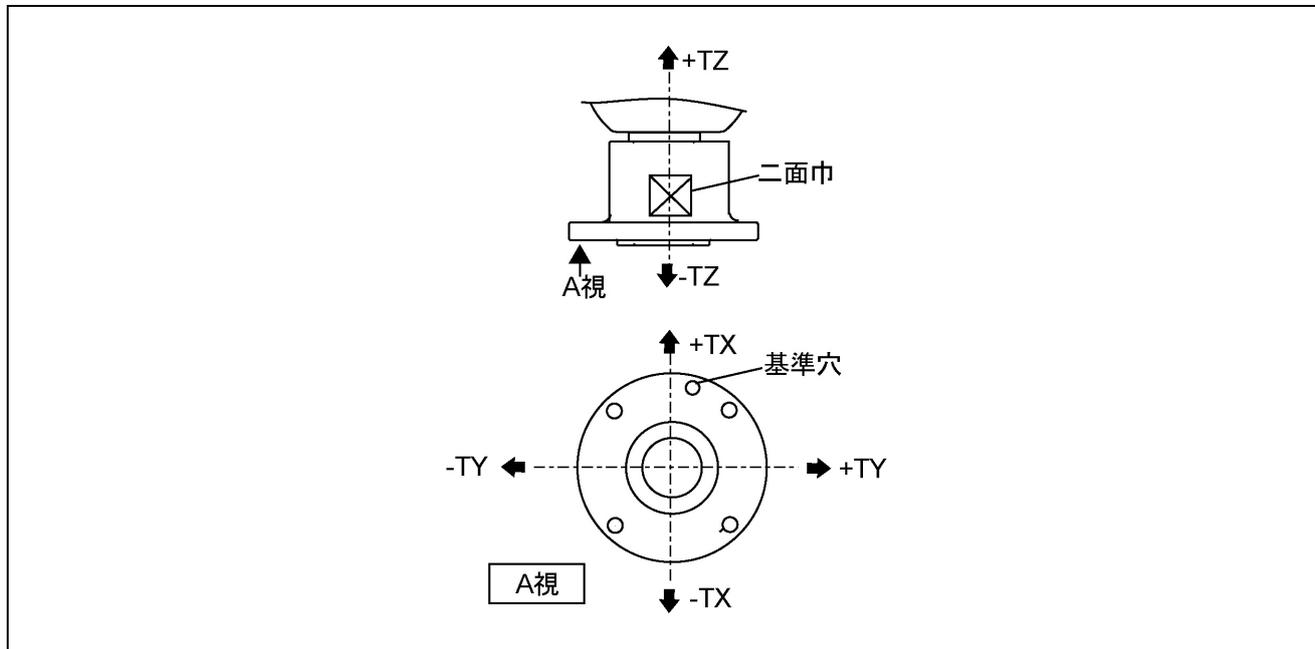
注：ワーク座標系を利用するためには、あらかじめワーク座標を設定しておく必要があります。詳細は操作ガイド「4.2.1[1.3]ワーク座標系の設定」を参照して下さい。



ワーク座標系とベース座標系

6.4 4軸ロボットツール座標系

ツール座標系は下図に示すメカニカルインタフェース座標の原点を基準にオフセットの量や回転の角度を定義する3次元直交座標です。ツール座標は最大63種類の登録ができ、1～63のツール座標番号を付けます。

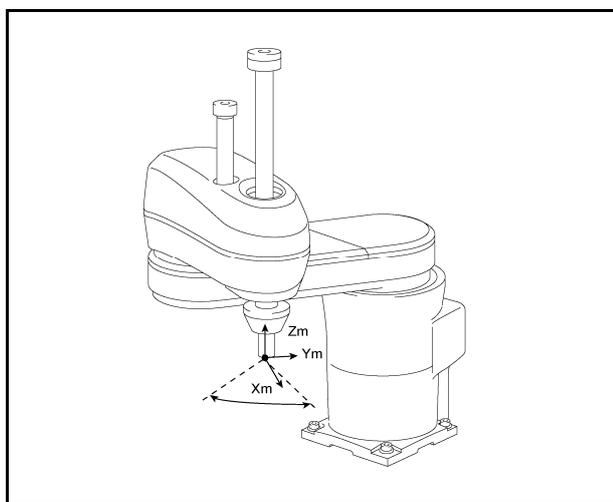


メカニカルインタフェース座標系

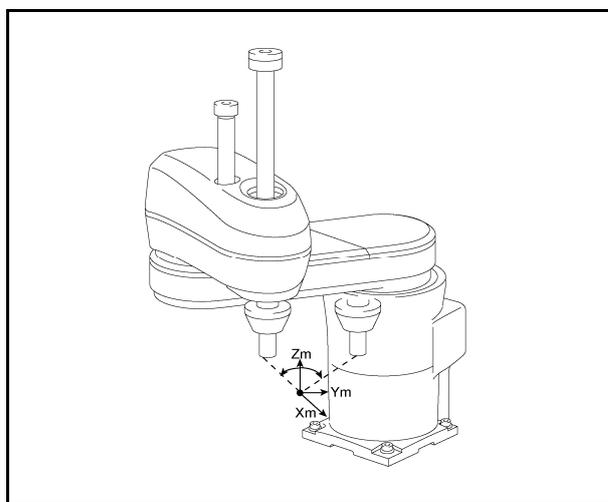
注：ツール座標系を利用するためには、あらかじめ、ツール座標を設定しておく必要があります。詳細は操作ガイド「4.2.1[2.2]ツール定義の入力方法」を参照して下さい。

6.5 4軸ロボットツール座標系のメリット

手動モードで動作させる場合、ツール座標系を使用するとツール定義にてオフセットした点を中心にロボットツール先端を動かすことができます。



ツール定義を使用しないで、X-Yモードで第4軸を手動回転させたとき

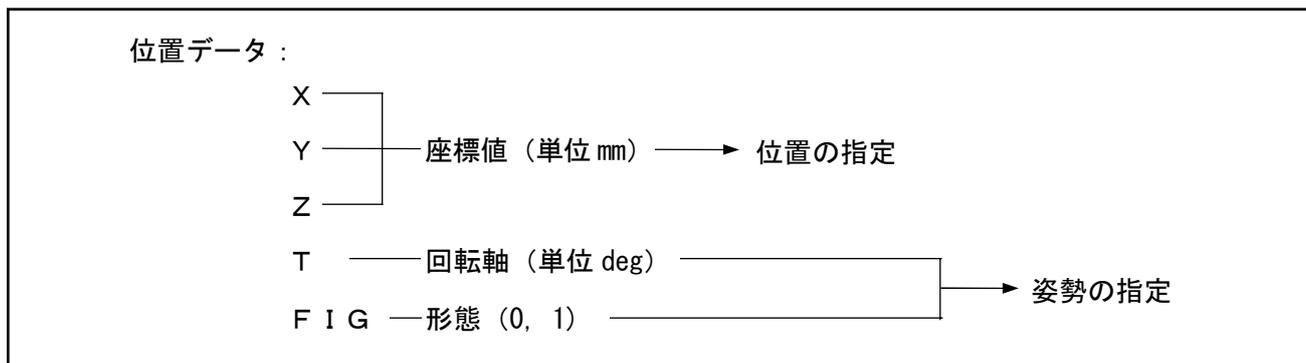


ツール定義を使用し、X-Yモードで第4軸を手動回転させたとき

6.6 4軸ロボットで扱う位置データ

位置データとは、ロボットのフランジ中心（ツール定義が有効なときはツール先端）の位置と、そのときのロボットの姿勢を表す、5要素（4軸ロボットの場合）のベース座標系のデータをいいます。

ロボットの現在位置や移動目標位置を表現するとき、ロボットの姿勢も含めて、この位置データを使います。



位置データの構成

X、Y、Zで示される座標値とは、ロボットのフランジ中心（ツール定義が有効な時はツール先端）の位置をベース座標系（ X_b 、 Y_b 、 Z_b ）で示した値で、単位は[mm]です。

Tで示される回転角とは、ロボットツールφ座標系（ツール定義が有効な時はツール座標系）のX軸とベース座標系の X_b 軸がなす角で単位は[deg]です。

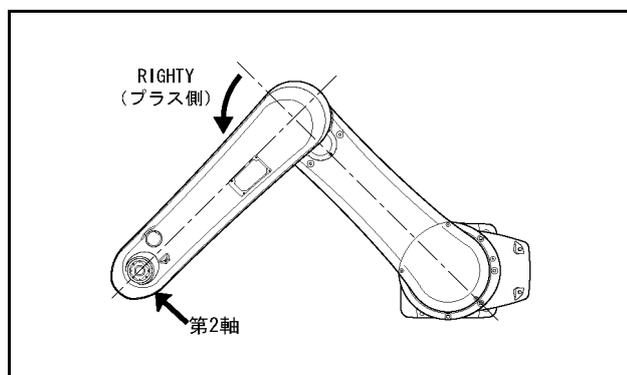
FIGで示される形態とは、ロボットアームの関節がとる形態です。

6.6.1 4軸の形態について

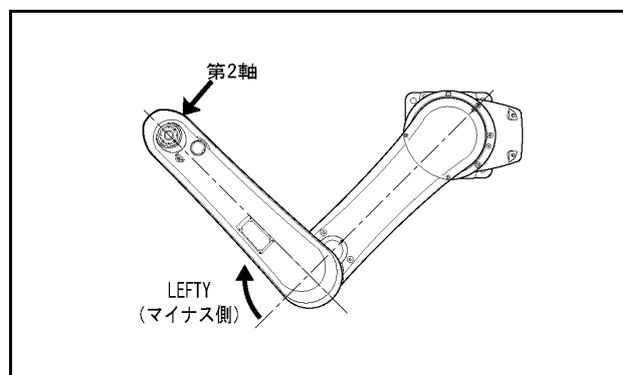
4軸ロボットの位置決めには、下図に示すように2通りの姿勢をとることができます。

ロボット形態

値	形態
0	RIGHTY
1	LEFTY



右手系 (RIGHTY)



左手系 (LEFTY)

第2軸がプラス側にある状態を右手系と呼び、第2軸がマイナス側にある状態を左手系と呼びます。

6.7 6軸ロボットの座標系

ロボットを操作するときの座標系には次の3種類があります。

- ・ ベース座標系
- ・ ワーク座標系
- ・ ツール座標系

6.8 6軸ロボットベース座標系

ベース座標系は、いわゆるワールド座標系であり、3次元直交座標です。その原点はロボットのベース中心です。ベース座標系のX軸を X_b 、Y軸を Y_b 、Z軸を Z_b と表現します。

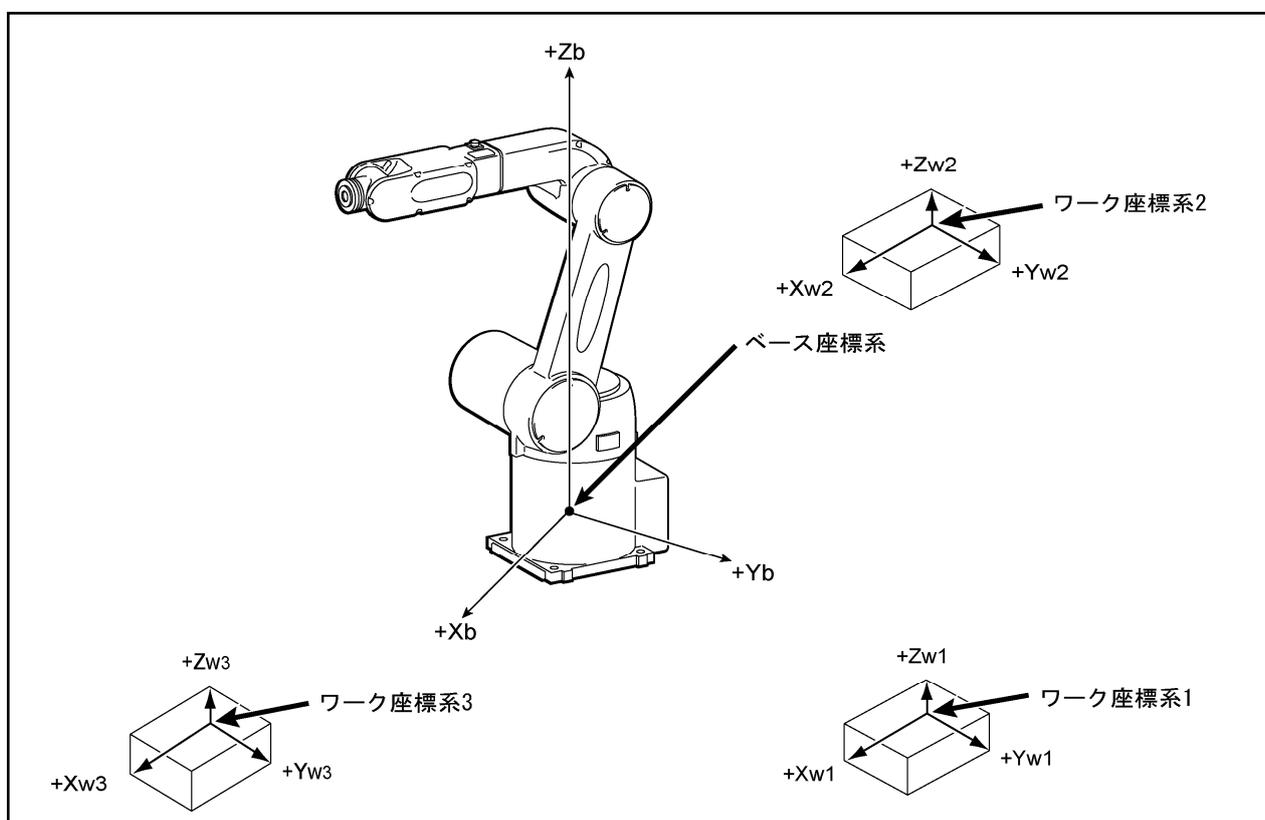
6.9 6軸ロボットワーク座標系

ワーク座標系は、各軸がベース座標系の各軸と平行な3次元直交座標系として定義されます。ワーク座標系は、ロボットが操作しようとする対象のワークの作業空間毎に定義される3次元直交座標です。図4-1に示すように、その原点は対象ワークの包絡直方体の一つの頂点となります。ワーク座標はベース座標に対する座標原点位置(X, Y, Z)とベース座標のX軸・Y軸・Z軸周りの回転角(R_x, R_y, R_z)で表現します。

ワーク座標は最大7種類の登録ができ、1~7のワーク座標番号を付けます。

ワーク座標未定義の場合は、ベース座標となります。原点は、任意の場所に任意の数だけ定義できます。

注：ワーク座標系を利用するためには、あらかじめワーク座標を設定しておく必要があります。詳細は操作ガイド「4.2.1[1.3]ワーク座標系の設定」を参照して下さい。

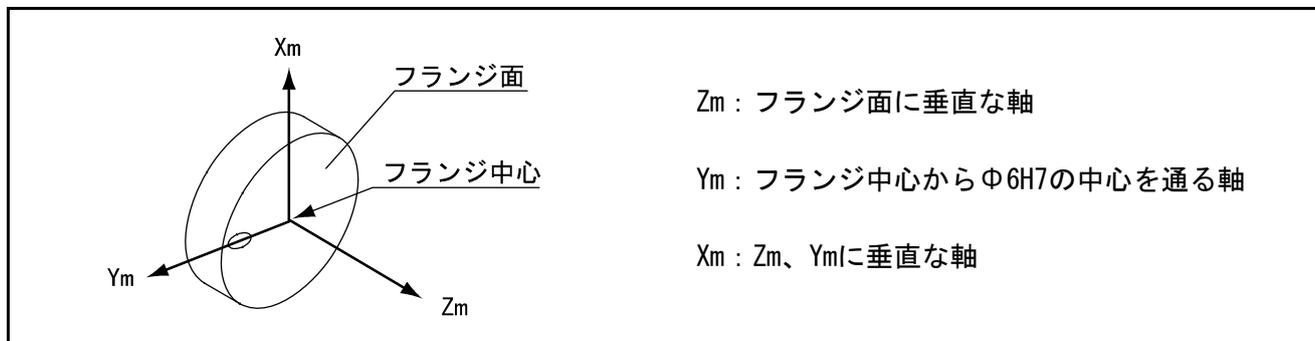


ワーク座標系とベース座標系

6.10 6 軸ロボットツール座標系

ツール座標系は、下図に示すメカニカルインタフェース座標の原点を基準にオフセット量や、回転の角度を定義する 3 次元直交座標です。

ツール座標は最大 63 種類の登録ができ、1～63 のツール座標番号を付けます。



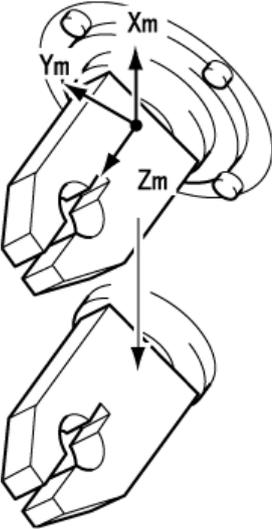
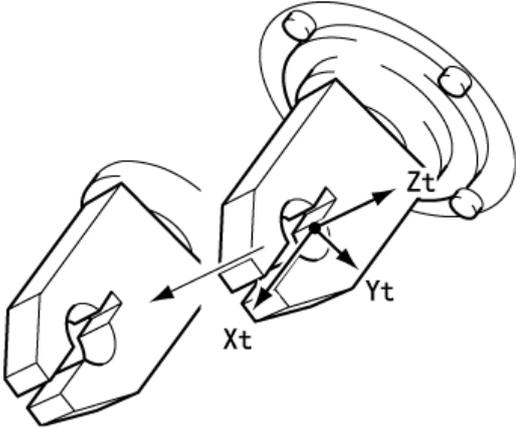
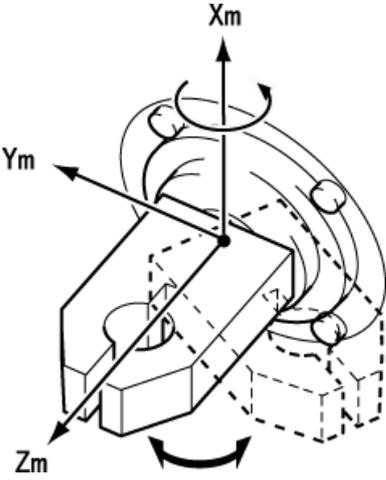
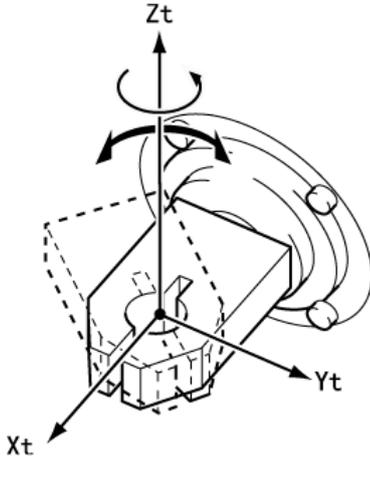
メカニカルインタフェース座標系

注：ツール座標を利用するためには、あらかじめ、ツール座標を設定しておく必要があります。詳細は操作ガイド「4.1.1[2.4]ツール座標の作り方」を参照して下さい。

6.11 6軸ロボットツール座標系のメリット

手動モードで動作させる場合、ツール座標系を使用すると、ハンド面を任意の方向へ動かすことが可能になり、ティーチングしやすくなります。

メカニカルインタフェース座標系適用時との動作の比較を、下図に示します。

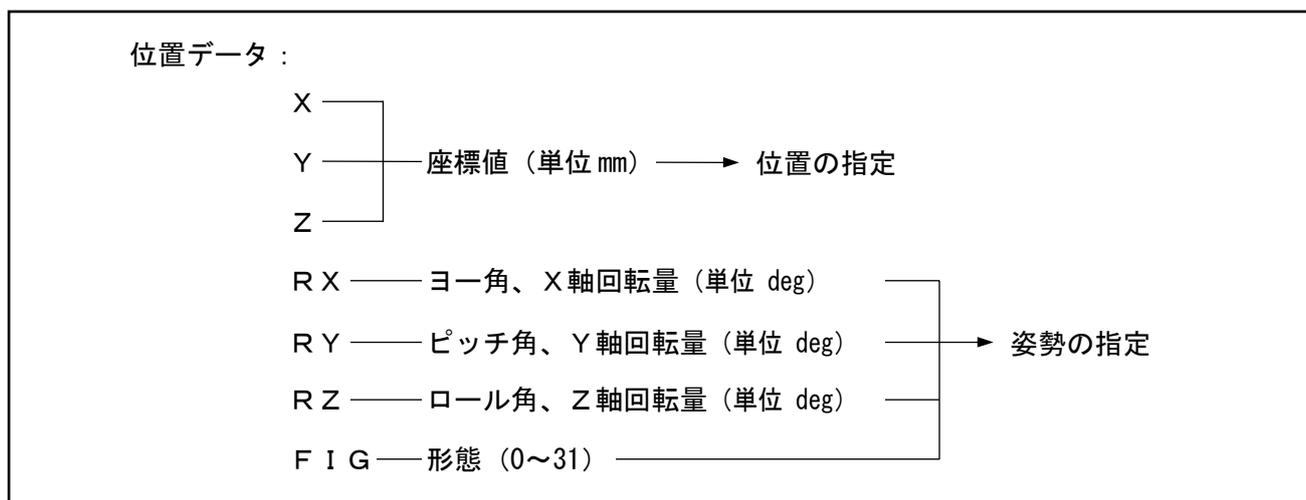
メカニカルインタフェース座標系 (TOOL0)	ツール座標系
<p data-bbox="153 524 363 555">X- を押す場合</p> 	<p data-bbox="817 524 1027 555">Z- を押す場合</p>  <p data-bbox="826 1061 1417 1133">ハンド面を任意の方向へ移動するティーチングが可能になる。</p>
<p data-bbox="153 1173 379 1205">RX+ を押す場合</p> 	<p data-bbox="817 1173 1043 1205">RZ+ を押す場合</p>  <p data-bbox="826 1733 1394 1805">ハンドを中心とした回転動作のティーチングが可能になる。</p>

ツール座標を使った手動動作

6.12 6 軸ロボットで扱う位置データ

位置データとは、ロボットのフランジ中心（ツール定義が有効なときはツール先端）の位置と、そのときのロボットの姿勢を表す7要素（6軸ロボットの場合）のベース座標系のデータをいいます。

ロボットの現在位置や移動目標位置を表現する際に、ロボットの姿勢も含めて、この位置データを使います。



位置データの構成

X、Y、Zで示される座標値とは、ロボットのフランジ中心（ツール定義が有効な時はツール先端）の位置をベース座標系（ X_b 、 Y_b 、 Z_b ）で示した値で、単位は [mm] です。

RX、RY、RZで示される、ヨー、ピッチ、ロール角は、フランジ中心を原点とする座標系（メカニカルインタフェース座標系）がなす、ベース座標系のそれぞれの軸周りの回転角で、単位は [deg] です。この回転は、RZ、RY、RXの順序で回転させます。回転する順番を変えると、同じ回転角でも違った姿勢になりますので、順番は決められています。

FIGで示される形態とは、ロボットアームの関節がとる形態です。

6.12.1 6軸の形態について

6軸ロボットは、ツール先端の1つの位置と姿勢（X、Y、Z、RX、RY、RZ）に対して、以下に説明するように、腕・ひじ・手首・第6軸・第4軸において違った形態をとることができます。

(1)～(5)に腕・ひじ・手首・第6軸・第4軸に対する、それぞれの形態の違いを説明します。

これらの形態を組み合わせると、1つの位置と姿勢に対して、32種類の形態をとることができます。この形態の組み合わせを、下表に示します。

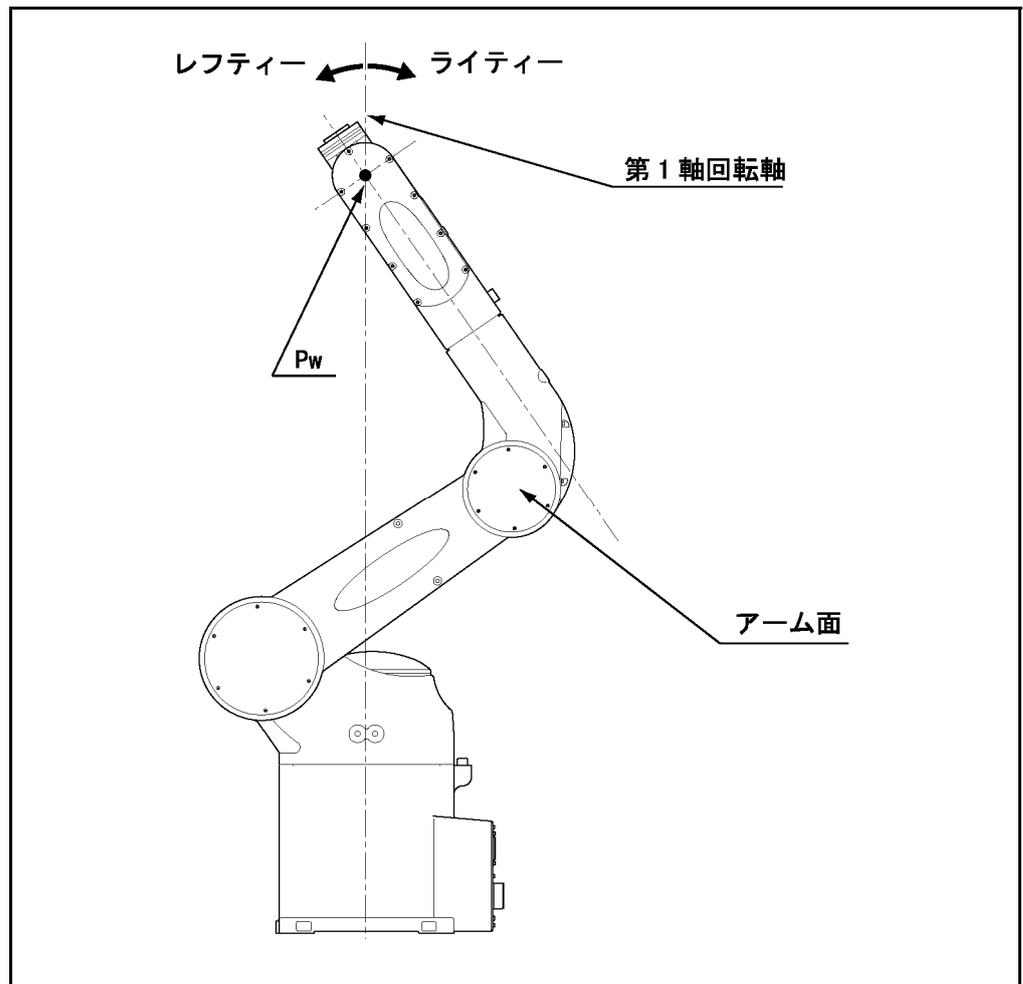
ロボット形態

値	第4軸形態	第6軸形態	手首形態	ひじ形態	腕形態
0	SINGLE 4	SINGLE	FLIP	ABOVE	RIGHTY
1	SINGLE 4	SINGLE	FLIP	ABOVE	LEFTY
2	SINGLE 4	SINGLE	FLIP	BELOW	RIGHTY
3	SINGLE 4	SINGLE	FLIP	BELOW	LEFTY
4	SINGLE 4	SINGLE	NONFLIP	ABOVE	RIGHTY
5	SINGLE 4	SINGLE	NONFLIP	ABOVE	LEFTY
6	SINGLE 4	SINGLE	NONFLIP	BELOW	RIGHTY
7	SINGLE 4	SINGLE	NONFLIP	BELOW	LEFTY
8	SINGLE 4	DOUBLE	FLIP	ABOVE	RIGHTY
9	SINGLE 4	DOUBLE	FLIP	ABOVE	LEFTY
10	SINGLE 4	DOUBLE	FLIP	BELOW	RIGHTY
11	SINGLE 4	DOUBLE	FLIP	BELOW	LEFTY
12	SINGLE 4	DOUBLE	NONFLIP	ABOVE	RIGHTY
13	SINGLE 4	DOUBLE	NONFLIP	ABOVE	LEFTY
14	SINGLE 4	DOUBLE	NONFLIP	BELOW	RIGHTY
15	SINGLE 4	DOUBLE	NONFLIP	BELOW	LEFTY
16	DOUBLE 4	SINGLE	FLIP	ABOVE	RIGHTY
17	DOUBLE 4	SINGLE	FLIP	ABOVE	LEFTY
18	DOUBLE 4	SINGLE	FLIP	BELOW	RIGHTY
19	DOUBLE 4	SINGLE	FLIP	BELOW	LEFTY
20	DOUBLE 4	SINGLE	NONFLIP	ABOVE	RIGHTY
21	DOUBLE 4	SINGLE	NONFLIP	ABOVE	LEFTY
22	DOUBLE 4	SINGLE	NONFLIP	BELOW	RIGHTY
23	DOUBLE 4	SINGLE	NONFLIP	BELOW	LEFTY
24	DOUBLE 4	DOUBLE	FLIP	ABOVE	RIGHTY
25	DOUBLE 4	DOUBLE	FLIP	ABOVE	LEFTY
26	DOUBLE 4	DOUBLE	FLIP	BELOW	RIGHTY
27	DOUBLE 4	DOUBLE	FLIP	BELOW	LEFTY
28	DOUBLE 4	DOUBLE	NONFLIP	ABOVE	RIGHTY
29	DOUBLE 4	DOUBLE	NONFLIP	ABOVE	LEFTY
30	DOUBLE 4	DOUBLE	NONFLIP	BELOW	RIGHTY
31	DOUBLE 4	DOUBLE	NONFLIP	BELOW	LEFTY

(1) 腕形態

アーム面に対して垂直方向から見た「Pw」の位置が、第1軸の回転軸より左にある場合がレフティー (LEFTY) 形態で移動可能な範囲、右にある場合がライティー (RIGHTY) 形態で移動可能な範囲です。

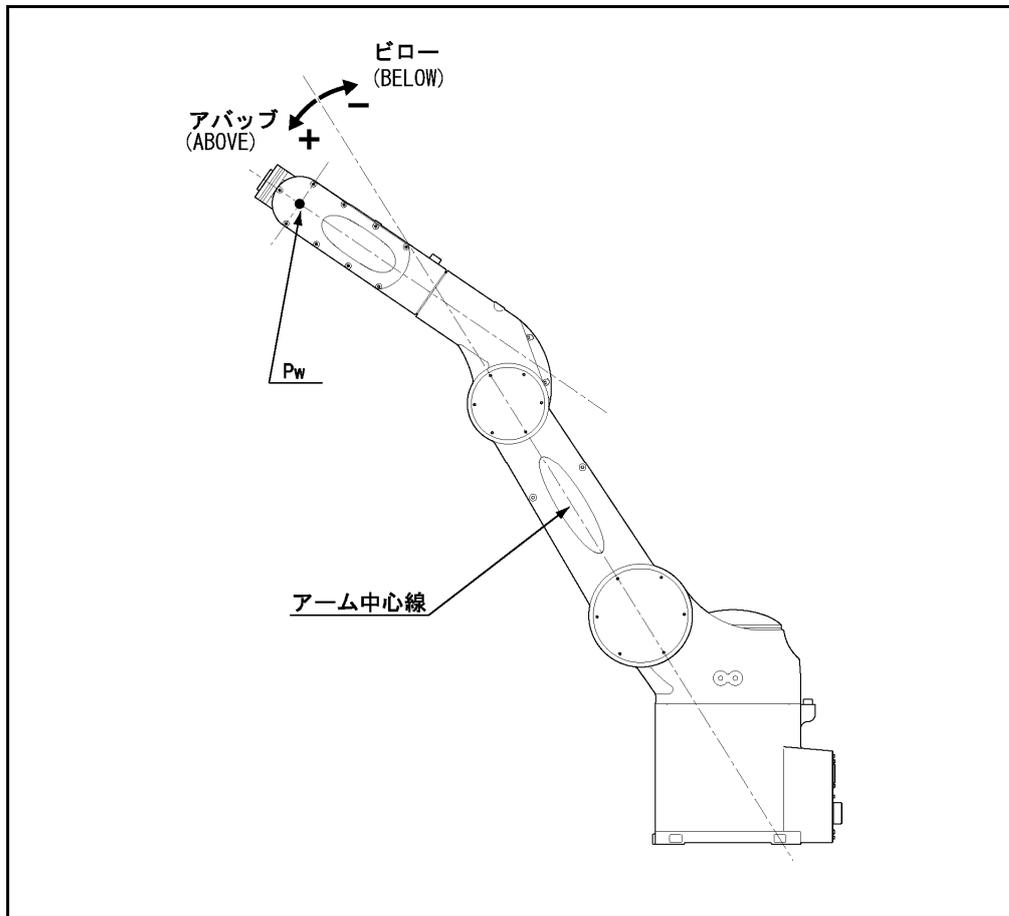
注意：第5軸の支軸中心「Pw」の位置が、第1軸の回転軸の上にある場合、2つの形態の境界上にあることになり、この位置を特異点といいます。



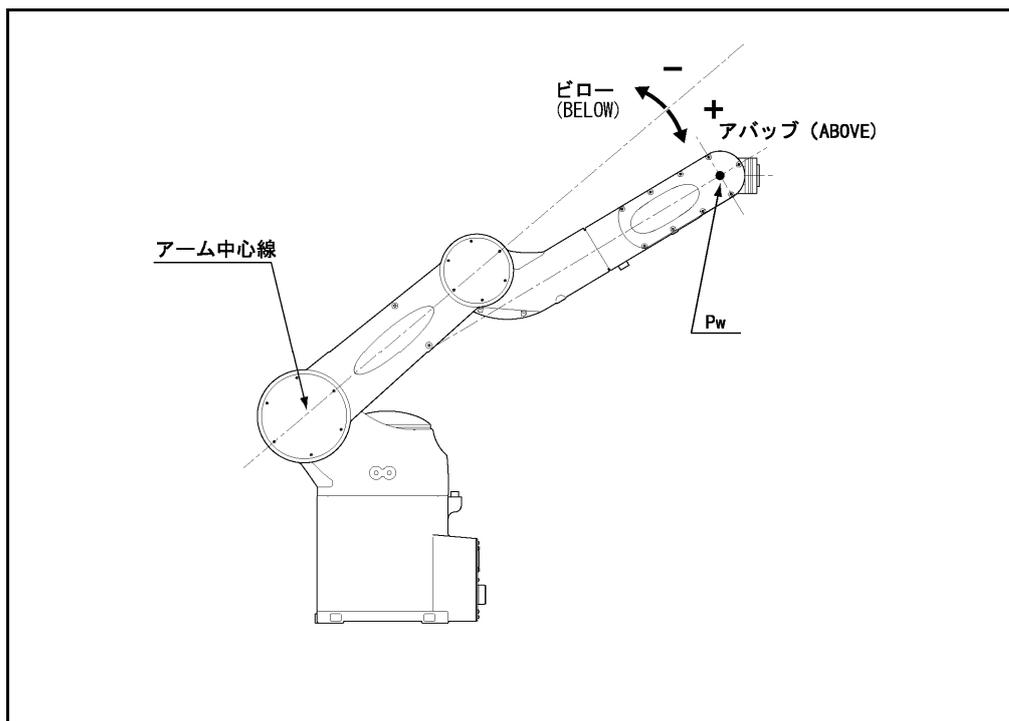
レフティー・ライティーの境界

(2) ひじ形態

アーム中心線に対して、「Pw」の位置が+側にある場合がアバップ（ABOVE）形態で移動可能な範囲、-側にある場合がビロー（BELOW）形態で移動可能な範囲です。



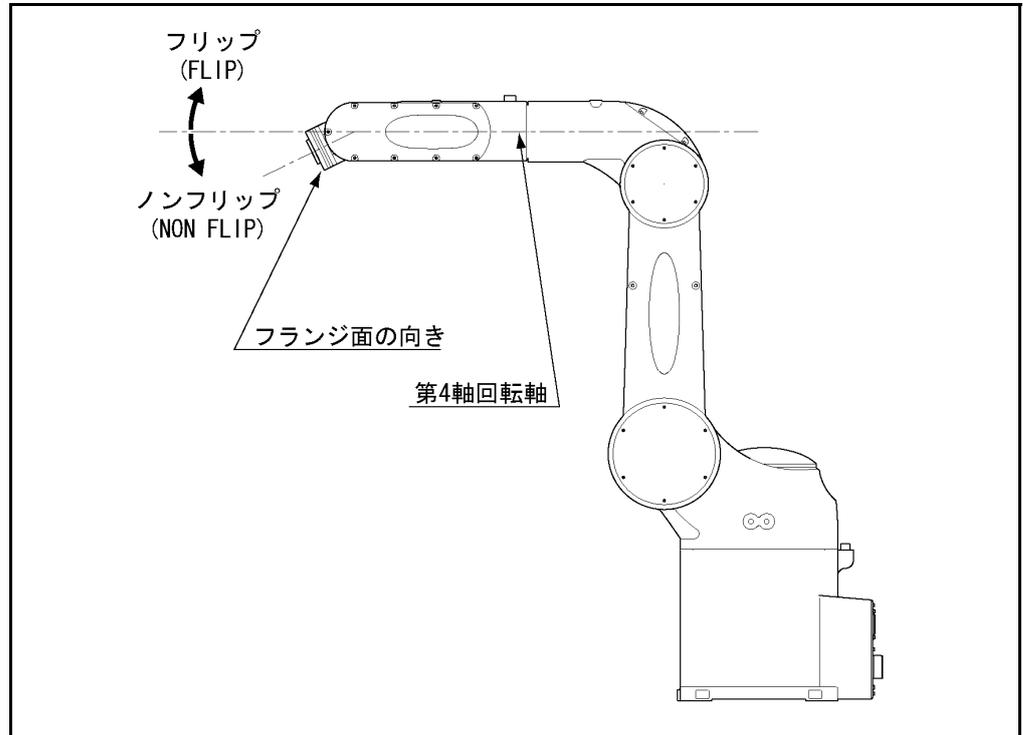
アバップ・ビローの境界（レフティーの場合）



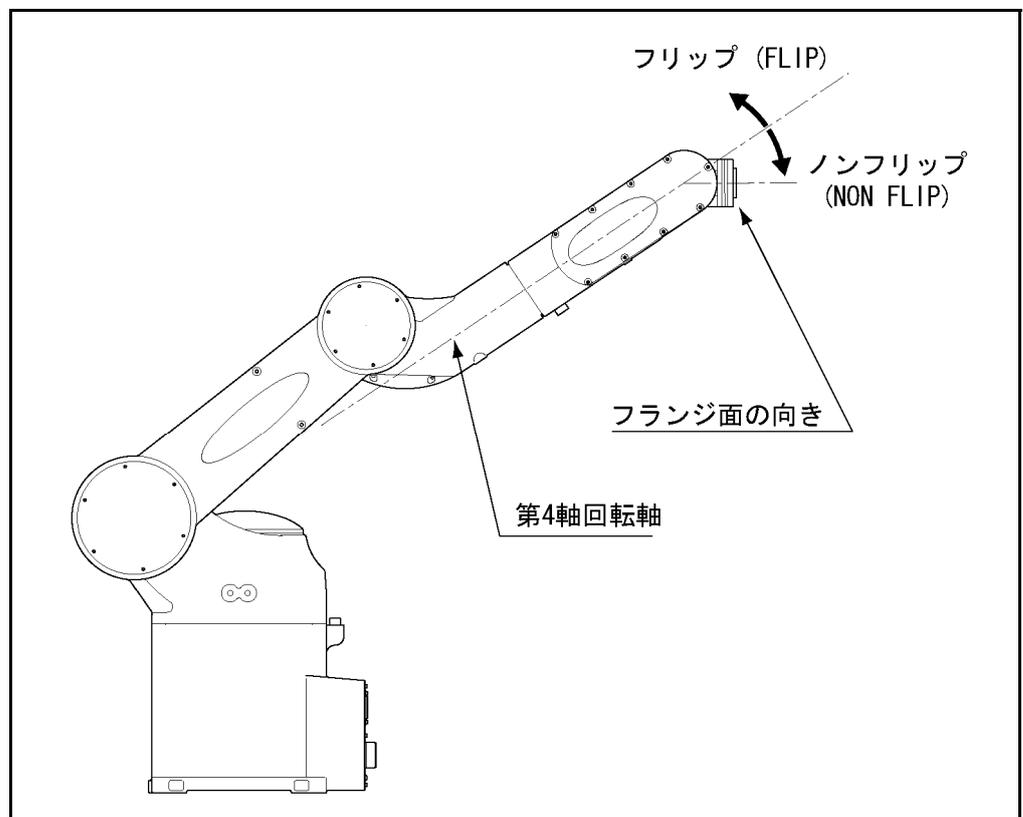
アバップ・ビローの境界（ライティーの場合）

(3) 手首形態

第4軸の回転軸中心線を基準にして、フランジ面の向きが上側にある場合がフリップ (FLIP) で、下側にある場合がノンフリップ (NONFLIP) です。(図 4-38、図 4-39 は下側にあります)



フリップ・ノンフリップの境界 (レフティーの場合)



フリップ・ノンフリップの境界 (ライティーの場合)

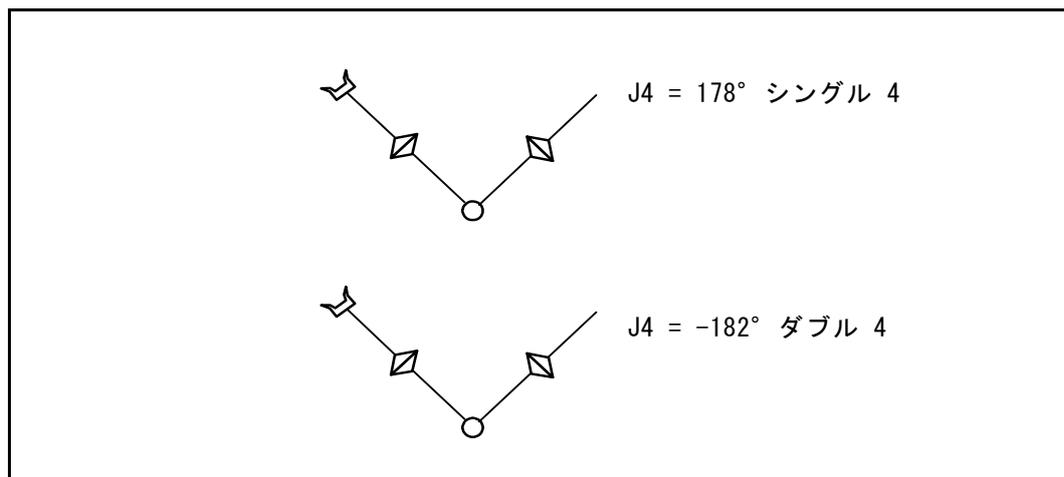
(4) 第4軸形態

第4軸形態は、第4軸の値で形態が決まります。

第4軸の形態はシングル4 (SINGLE 4) とダブル4 (DOUBLE) の2種類をとることができます。

第4軸の回転角 θ_4 が $(-180 < \theta_4 \leq 180)$ の場合がシングル4で、 $(180 < \theta_4 \leq 185$ または $-185 < \theta_4 \leq -180)$ の場合がダブル4です。

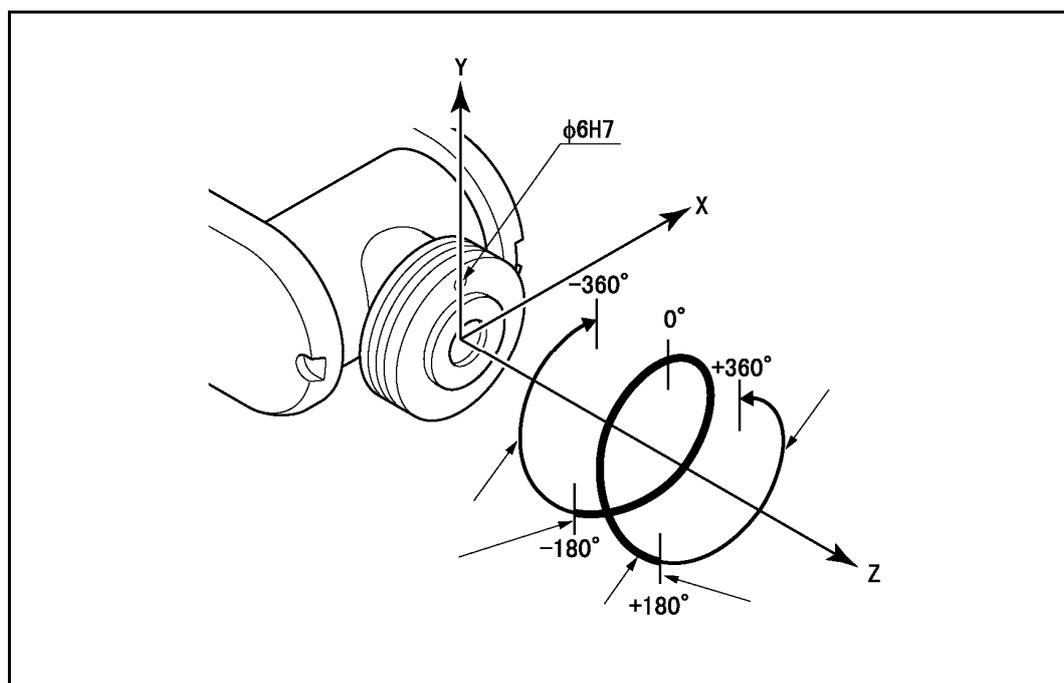
$\theta_4 = 180^\circ$ と $\theta_4 = 181^\circ$ では第6軸形態が変わります。位置データを数値変更する場合、第6軸形態の設定にご注意ください。($\theta_4 = 181^\circ$ に修正する予定で、第6軸形態が修正されないと $\theta_4 = -179^\circ$ になります。)



第4軸形態

(5) 第6軸形態

第6軸の回転角 θ_6 が $(-180 < \theta_6 \leq 180)$ の場合がシングル (SINGLE) で、 $(180 < \theta_6 \leq 360$ または $-360 < \theta_6 \leq -180)$ の場合がダブル (DOUBLE) です。



シングル・ダブルの境界

第7章 ティーチングのための準備

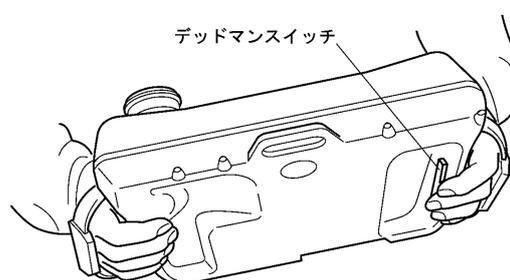
7.1 ティーチングペンダントの取扱い

7.1.1 ティーチングペンダントの持ち方とデッドマンスイッチ

ティーチングペンダントは、手で保持して操作します。ティーチングペンダントには安全確保のためのデッドマンスイッチがあり、下に示すいずれかの方法で指で押込むことにより操作できます。

ティーチングペンダントの持ち方とデッドマンスイッチ

ティーチングペンダントは両手で持って操作します。



★ポイント★ デッドマンスイッチの機能は、ティーチングペンダントを使って手動でロボットを運転している時、予期しない心神喪失、死亡などによって正常な運転ができなくなった場合、自動的かつ安全にロボットを停止させることです。操作者がこのような状態に陥ったとき、このスイッチを押す力は、非常に弱くなるか、あるいは強くなるかのいずれかです。デッドマンスイッチは、つぎの3つの操作状態を認識できるように設計された3ポジションスイッチです。

- 1) 押していないか、押す力が弱いとき → スイッチはOFF
- 2) 押す力が正常なとき → スイッチはON
- 3) 押す力が強すぎるとき → スイッチはOFF

スイッチがONのとき以外は、ロボットは停止し、ロボットを運転することはできません。

ティーチングペンダントは、安全性を確保するために、たとえば手動モード時に移動方向キーを押下してロボットを動かすためには、デッドマンスイッチを同時に押下していなければならないように設計されています。

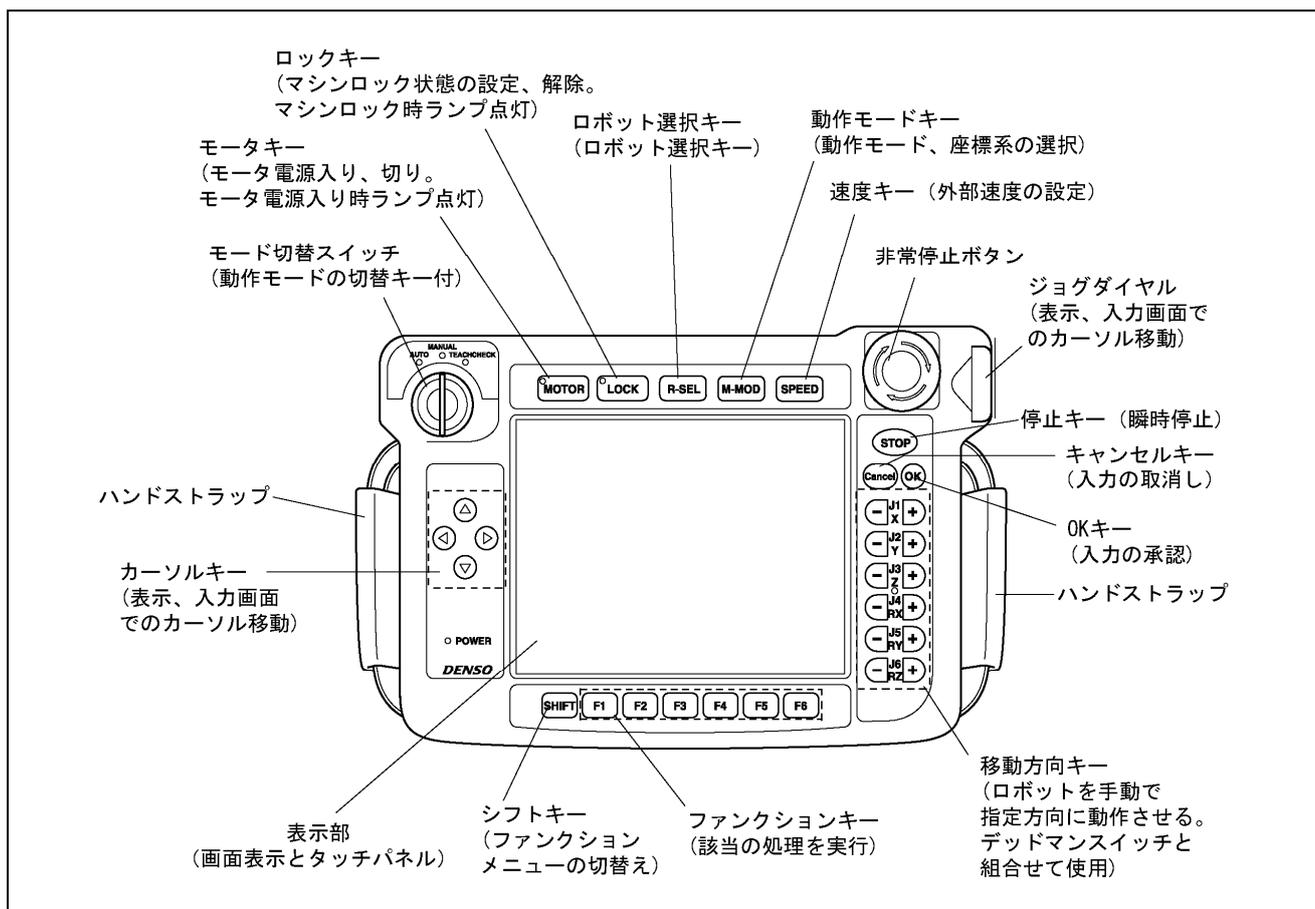
注：デッドマンスイッチは、イネーブルスイッチとも呼ばれます。

7.1.2 ティーチングペンダントのキー、ボタン、スイッチの名称

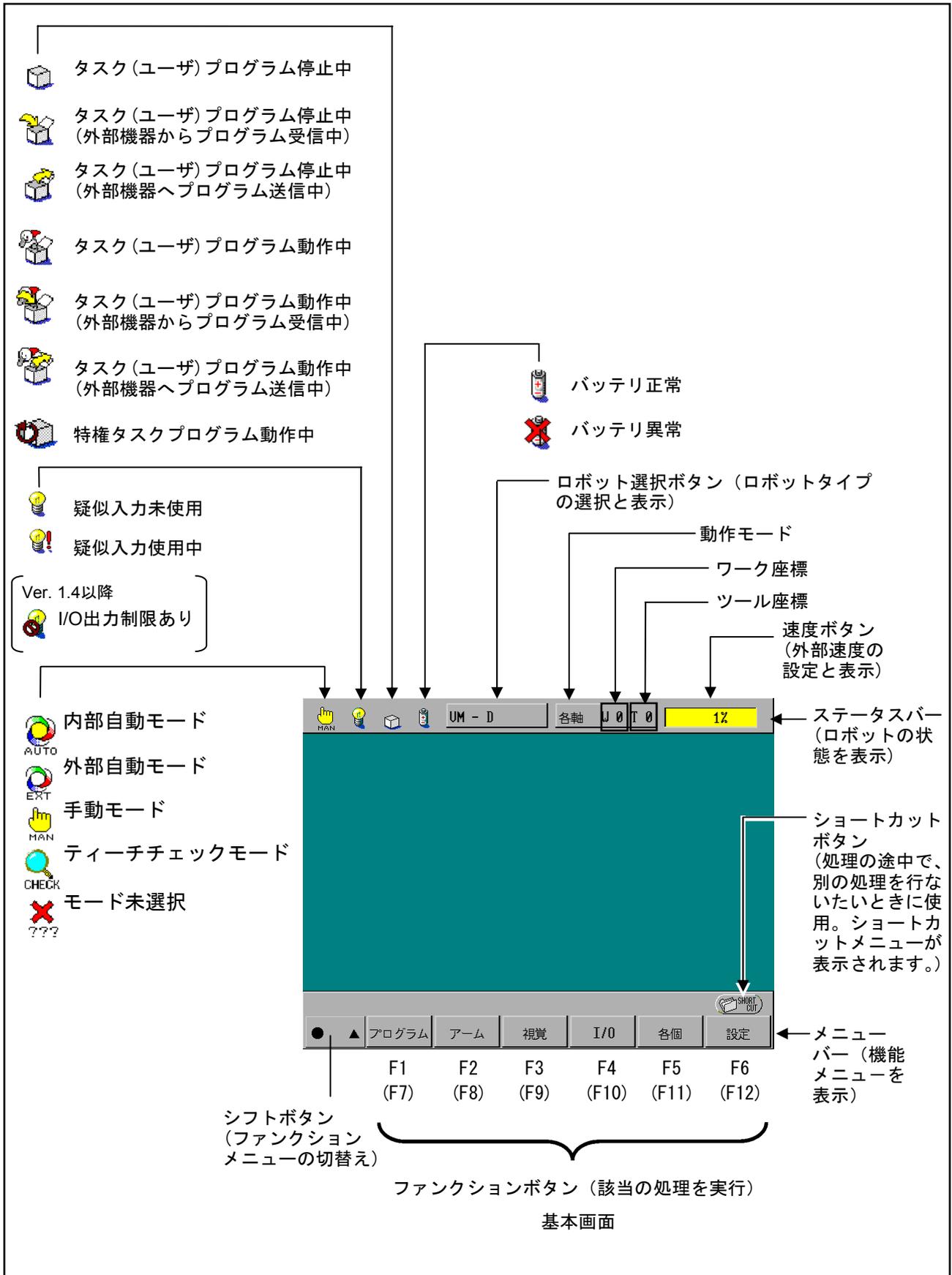
ティーチングペンダントのキー、ボタン、スイッチなどの名称を下図に示します。

次ページでは、ティーチングペンダントの画面配置と、機能を解りやすく表現したアイコンについて説明しています。

ロボットの運転を開始する前に、操作を手際良く行い、かつ、安全を確保するため、これらのキー、スイッチ、ボタンがどの位置にあるかを正確に把握しておいてください。



ティーチングペンダントのキー、ボタンおよびスイッチ名称



ティーチングペンダントのキー、ボタンおよびスイッチ名称

7.2 動作モードの種類

ロボットには「手動モード」、「ティーチチェックモード」、「自動モード」の3種類の動作モードがあります。

7.2.1 手動モード

ティーチングペンダント、ミニペンダントを使って手動でロボットを運転するためのモードです。

7.2.2 ティーチチェックモード

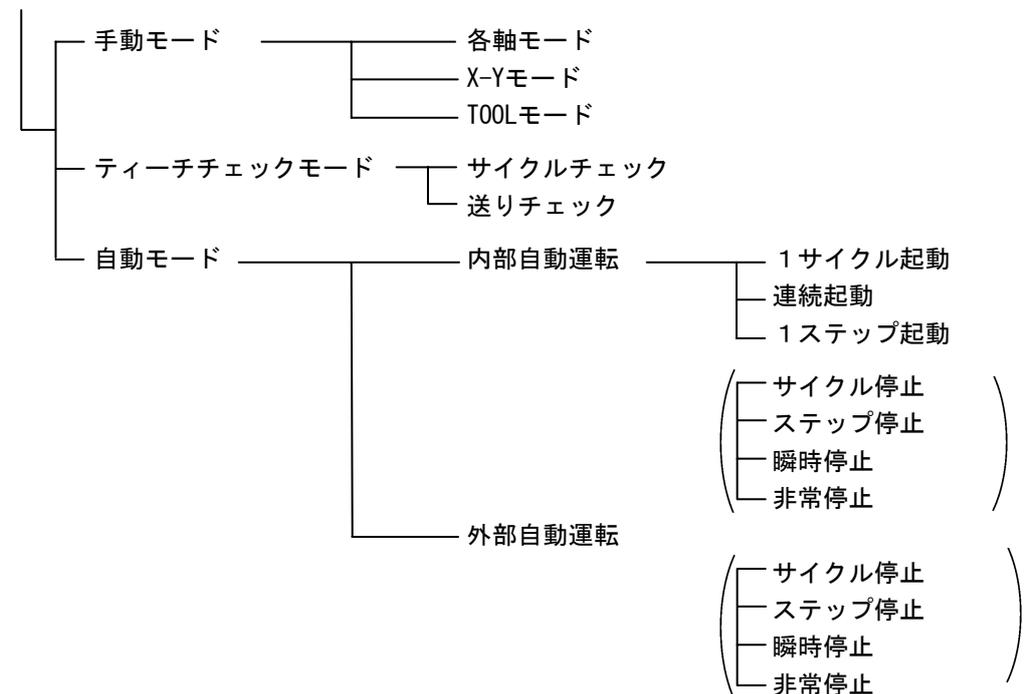
ティーチング後ティーチングペンダントを使ってプログラムに誤りがないか最終チェックするための制限付き自動運転を行うモードです。

7.2.3 自動モード

自動運転を行うためのモードです。

ティーチングペンダントとミニペンダントは上記の3種類の動作モードすべてに対応します。

動作モード



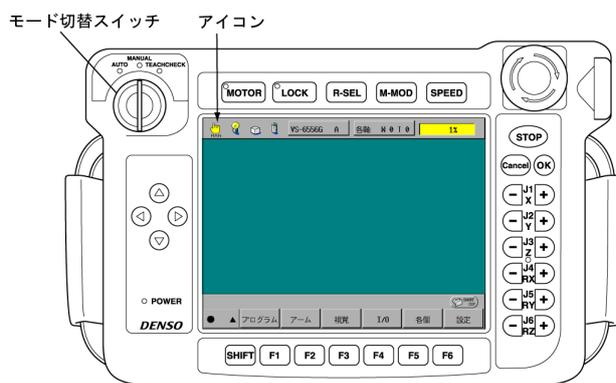
上記の3種類の動作モードでは、ロボットの動作をロック（マシンロックと言います）して、ロボットを実際に動かすことなく、ロボットコントローラだけでシミュレーション動作を行うことができます。またマシンロック時にI/Oの出力を制限することができます。詳細については操作ガイドの「5.5 I/O信号の表示とロボット動作のシミュレート」を参照してください。

7.3 動作モード切り替え

ティーチングを行う場合、手動モード（MANUAL）にする必要があります。

7.3.1 動作モード切り替え方法

モード切替スイッチを希望するモードに合わせます。



ステータスバーの左端部分に選択したモードを表すアイコンが表示されます。



7.3.2 動作モードと自動イネーブル入力信号の関連性

使用する動作モードに合わせ、自動イネーブル入信号（専用入力信号）のON/OFF状態も変更する必要があります。

入力信号の結線については「5.4 自動運転に必要な結線」をご確認下さい。

動作モード	自動イネーブル入力信号
手動モード	OFF（開放）
ティーチチェックモード	OFF（開放）
自動モード	ON（短絡）

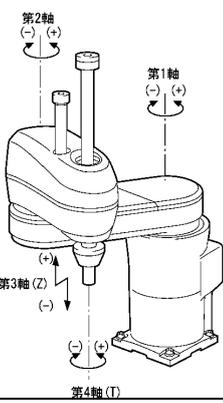
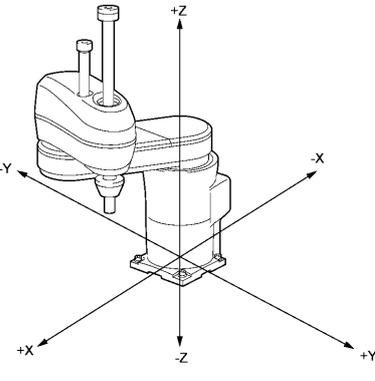
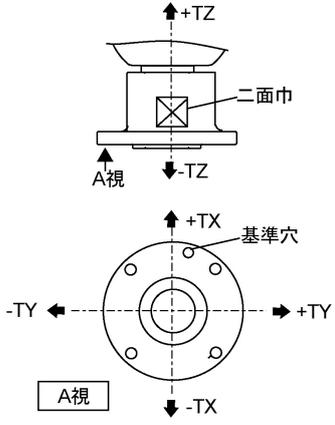
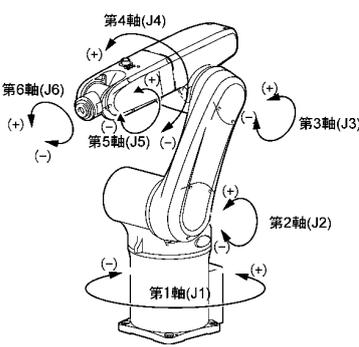
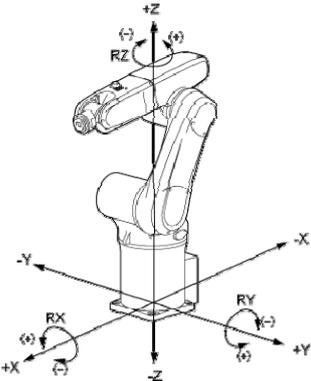
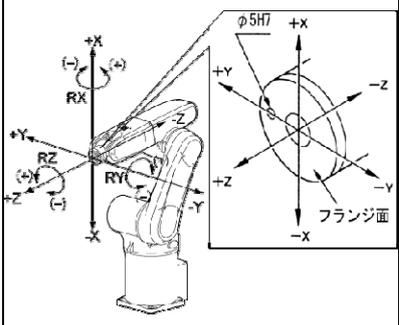
状態が不一致の場合、ERROR21 F2（自動イネーブルが入っています）または、ERROR21 F3（自動イネーブルが切れています）が発生し、操作はできません。

7.4 手動モードの種類

手動モードには、各軸モード、X-Yモード、TOOLモードの3種類があります。これらのモードのいずれかで、ティーチングペンダント、またはミニペンダントを使って、ロボットを手動で動作させることができます。

注：ロボットを手動動作させるには、専用入力の自動イネーブル入力信号がOFF(開放)になっている必要があります。

7.4.1 各軸・X-Y・ツールモード別 ロボット操作

	<各軸モード>	<X-Yモード>	<ツールモード>
動作	4個の軸がそれぞれ個別に動作します。	直交座標に沿って直線動作します。	第4軸の直交座標に沿って直線動作します。
4軸ロボット			
動作	6個の軸がそれぞれ個別に動作します。	直交座標に沿って直線動作します。	ロボットブランジ面のX、Y、Z軸に沿って直線動作します。
6軸ロボット			

7.4.2 手動モードの切替え方法

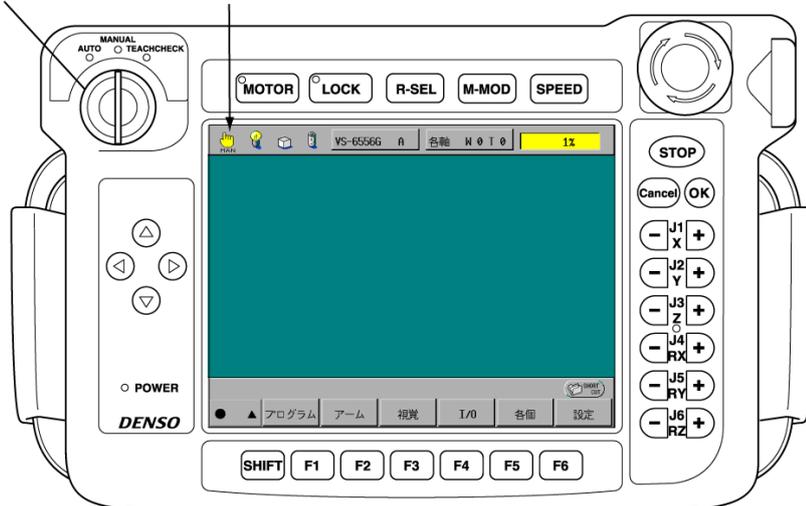
⚠注意：運転開始時は速度を 20%以下に設定して操作してください。初めから高速で動作させると、誤ってロボットを周囲の設備に衝突させるおそれがあります。

■ ティーチングペンダントを使用する場合

ステップ 1

モード切替スイッチを[MANUAL]に合わせます。

モード切替スイッチ アイコン

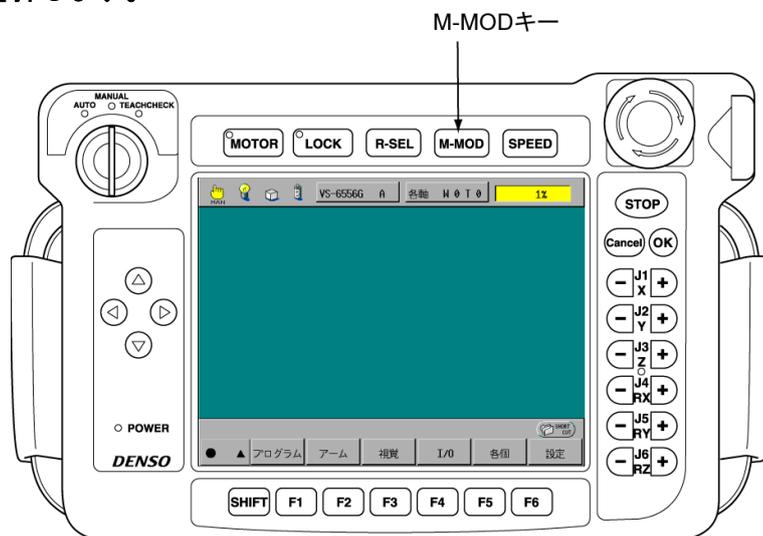


ステップ 2

[MOTOR]を押して、モータ電源を入れてください。

ステップ 3

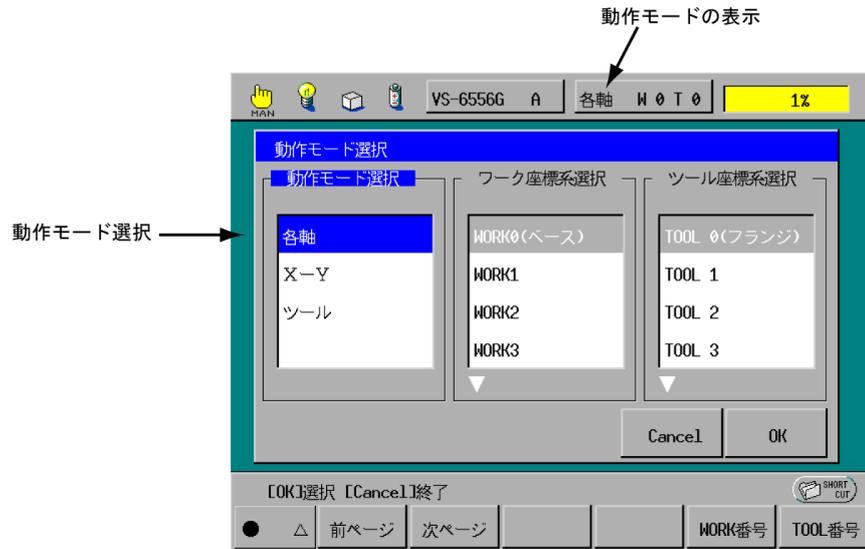
[M-MOD]を押します。



次のステップに示すような[動作モード選択]ウィンドウが表示されます。

ステップ 4

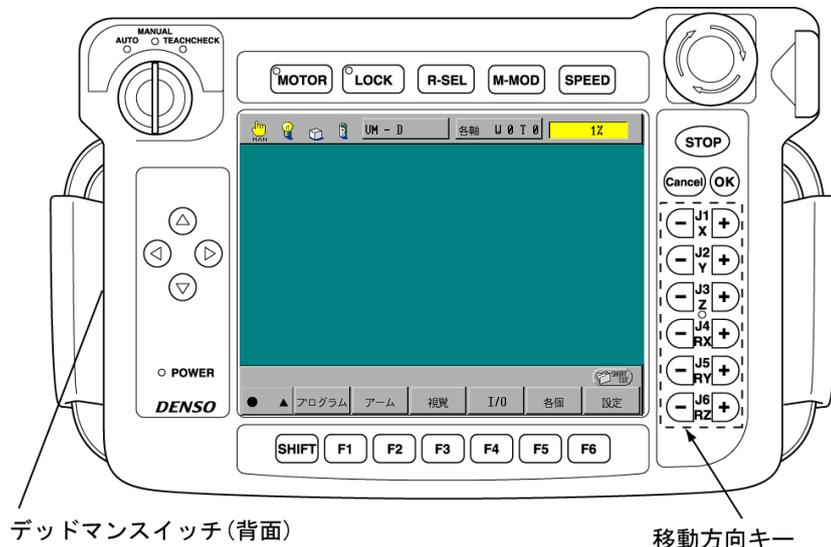
カーソルキーを使うか、画面に直接タッチしてモードを選択し、[OK]を押します。ステータスバーのモード表示部分に選択した動作モードが表示されます。



ステップ 5

デッドマンスイッチを押しながら移動方向キーのいずれかを押して、ロボットアームを動かします。

移動方向キーと軸の動作との関係については、7.4.1 各軸・X-Y・ツールモード別ロボット操作を参照してください。



7.5 ロボットを手動モードで動かす

ロボットコントローラとモータの電源を入れ、ティーチングペンダントを使ってロボットを手動で動かします。

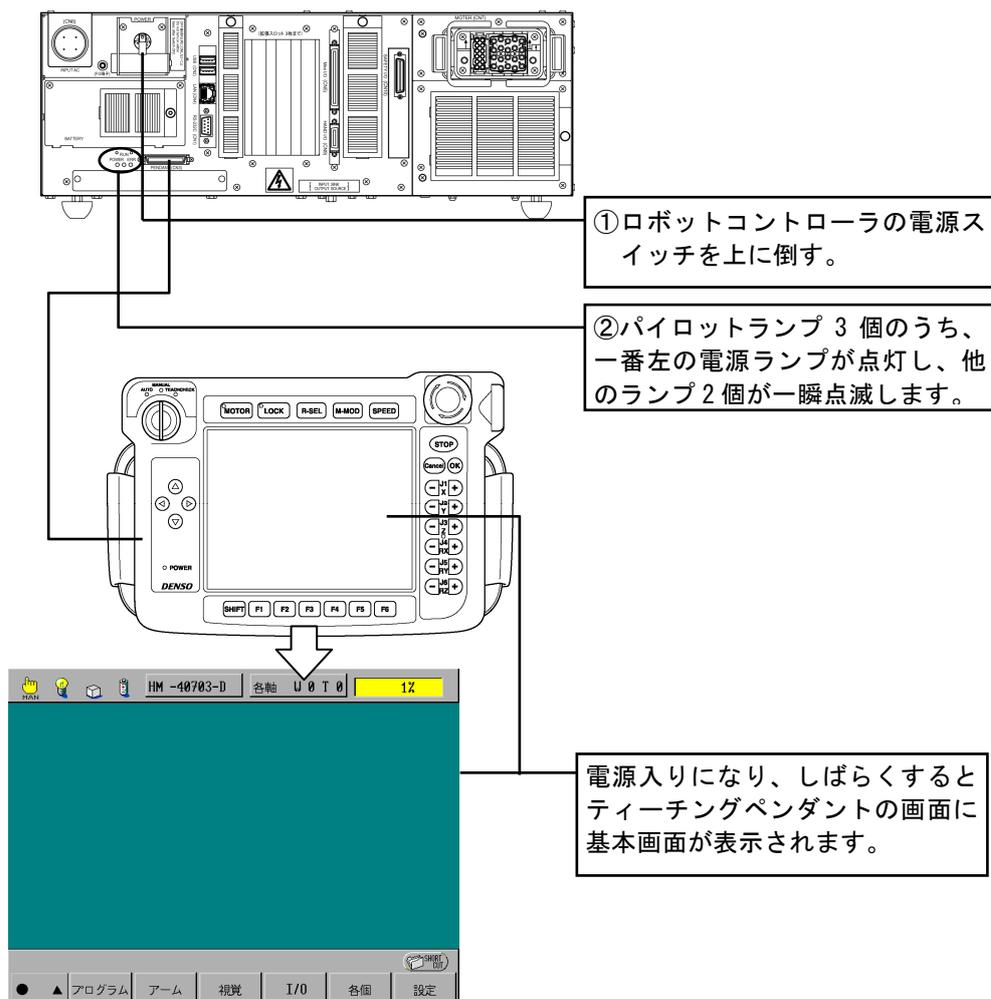
ステップ 1

安全を確認する

- ・ロボットの設置状況が正常であること。
- ・ロボットの動作範囲に人がいないこと。

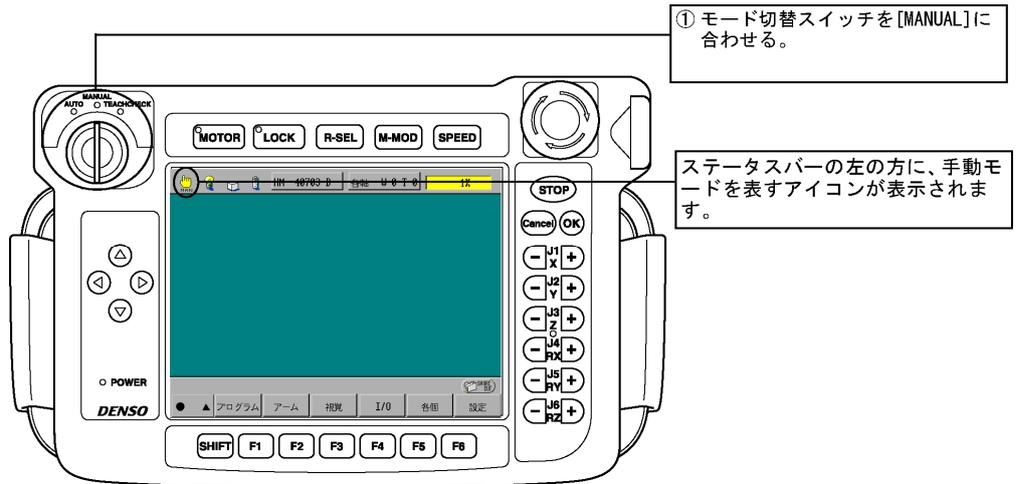
ステップ 2

ロボットコントローラの電源を入れる



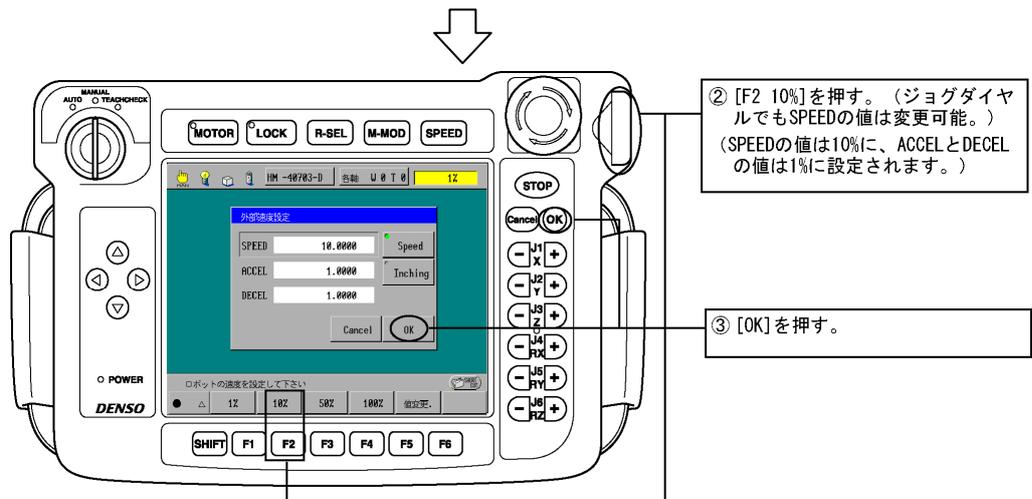
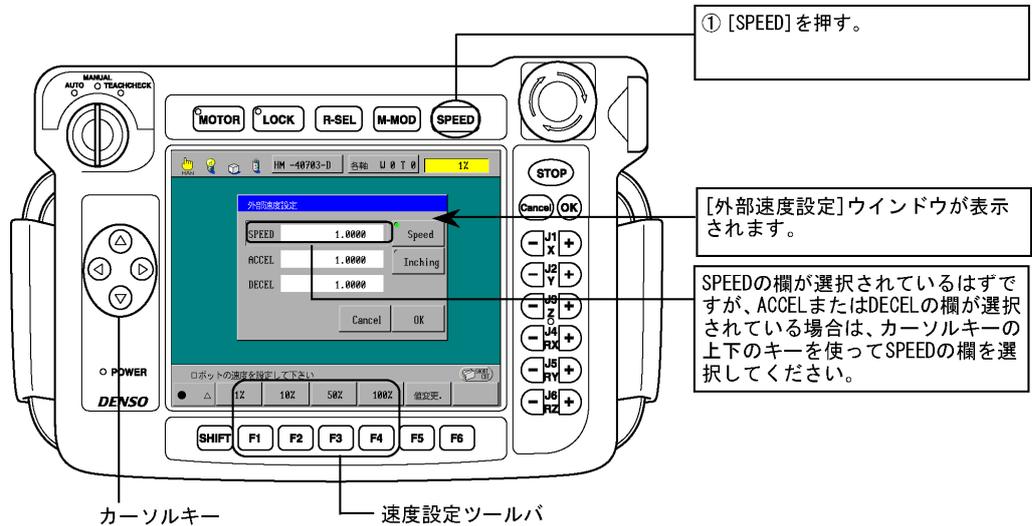
ステップ3

手動モードにする



ステップ4

速度・加速度を設定する



★参考★

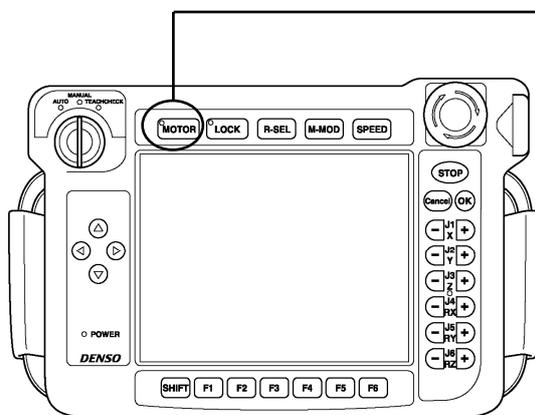
初めのうちは、安全のためにゆっくりと動かすので、この位の値にしておきます。あとで、様子がわかったところでこの値を入れ直すことができます。



スピード表示が10%になります。

ステップ5

モータの電源を入れる



① [MOTOR] を押す。
モータ電源が入り、[MOTOR]のランプが点灯します。

ステップ6

手動でロボットの各アームを実際に動作させる

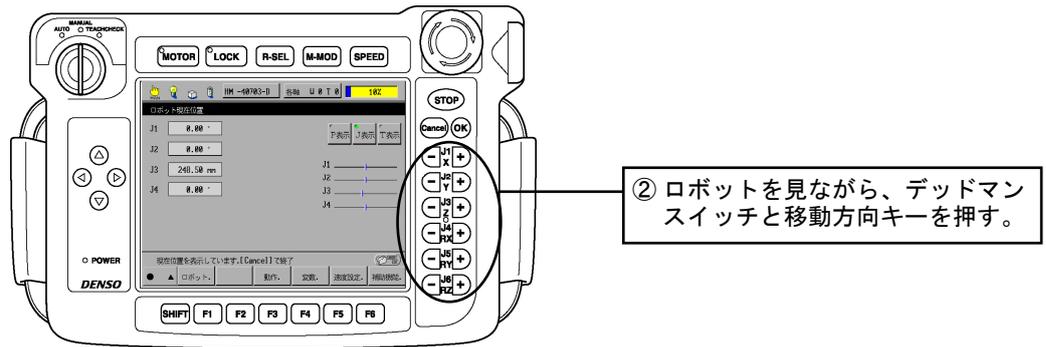


この操作を行なうとロボットが動作します。作業者はロボットの動作範囲から出てください。



① [F2 アーム] を押す。





4 軸ロボットでは J1~J4、6 軸ロボットでは J1~J6 の移動方向キーの操作に応じて、ロボットの対応する軸が動きます。[ロボット現在位置] ウィンドウには、各軸の角度が表示されます。

ステップ7

CAL (キャリブレーション) を実行する (DシリーズおよびXYC型ロボット)

CALとは、ロボットコントローラの電源を入れたあとに、ロボットが現在位置確認を行うために全ての軸を微小動作させることをいいます。
CALの手順を説明します。

★参考★

XYCシリーズを除く-E/-F/-GシリーズおよびVM-6083D, VM-60B1Dロボットでは、ステップ7の「CAL実行」は必要ありません。

※「CAL実行」を行っても問題はありません。

上記以外のDシリーズロボットおよびXYCシリーズロボットでは、「CAL実行」が必要です。



注意

この操作を行なうとロボットが動作します。作業者はロボットの動作範囲から出て下さい。ロボットの動作範囲に障害物がないことを確認してから、CALを行なってください。

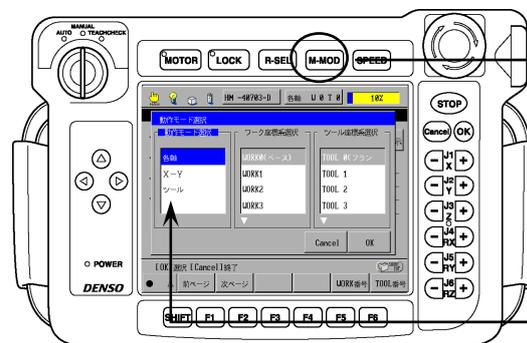


①[ロボット現在位置]ウィンドウが表示されている状態で、[F6 補助機能]を押す。



ステップ8

動作モードを選択し、手動運転する



①[M-MOD]を押す。

[動作モード]ウィンドウが表示されます。

ステップ8 (つづき)

ここでは X-Y モードでロボットを運転してみましょう。



② [動作モード選択] ウィンドウで「X-Y」の欄を選択する。(上下カーソルキーまたはジョグダイヤルを使って選択します。)

③ [OK] を押す。
基本画面が出たら [F2 Arm] を押す。



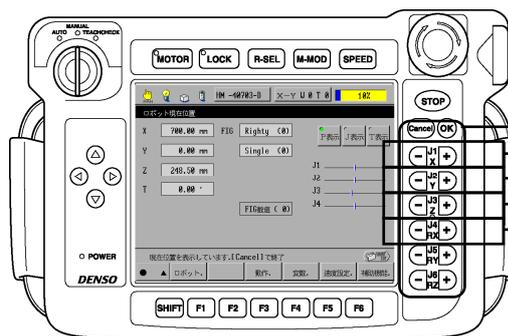
ステータスバーに X-Y と表示されます。

④ P 表示を押す。または、SHIFT キーを押して F1 を P 表示にして、F1 キーを押す。
(これは、X-Y モードで動作させるために必要です。)

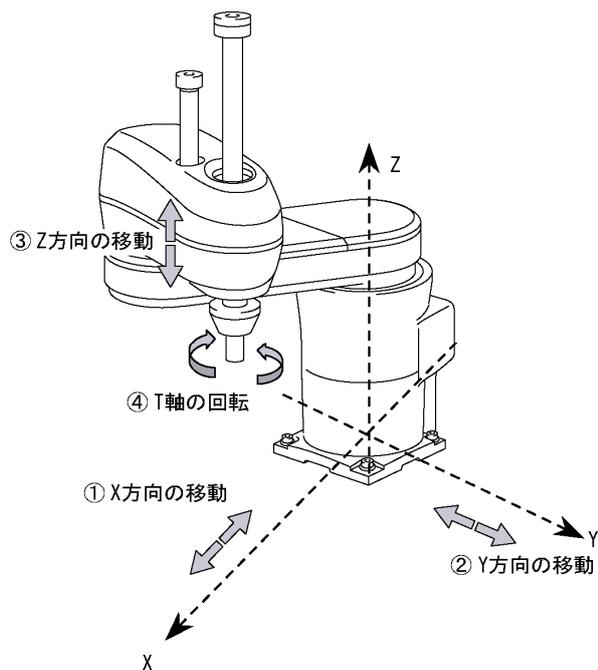
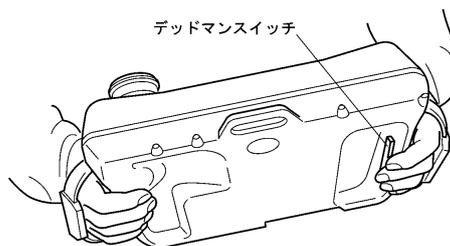


P 表示が点灯し、現在位置の表示形式が [P 表示] になります。

⑤ 移動方向キーとデッドマンスイッチを押して、ロボットを動かす。



- 移動方向キー
- ① X方向の移動
 - ② Y方向の移動
 - ③ Z方向の移動
 - ④ T軸の回転



第 8 章 ティーチング

8.1 ティーチングとは

ロボットを実際に動かすために、位置を指定することをティーチングといいます。プログラム中に位置を定数で指定することもできますが、ワークとロボットの相対的な位置関係を正確に合わせるには、現場で実際にロボットを動かして、位置を決定する必要があります。このため、プログラムの中では位置を変数として記述し、ティーチングで変数の具体的な値を指定します。

8.2 ティーチング可能なグローバル変数

「変数」とは、プログラム中で使うデータを一時的にしまっておくものです。ティーチング可能なグローバル変数には以下の 3 種類があります。

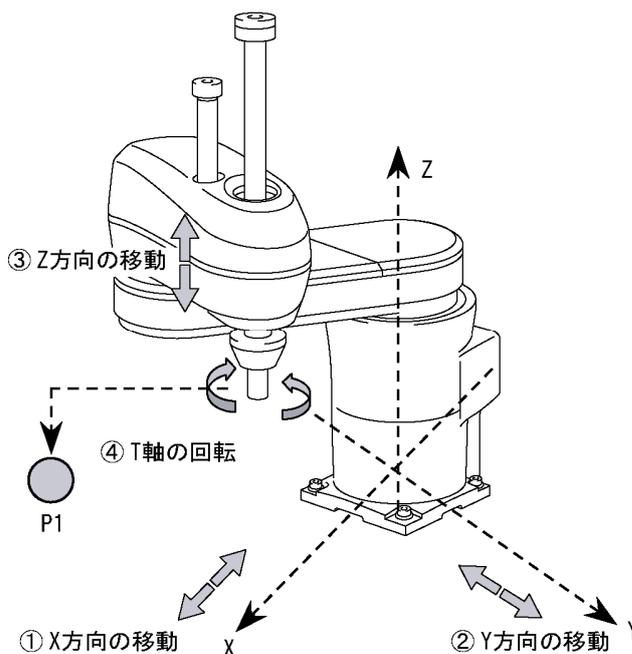
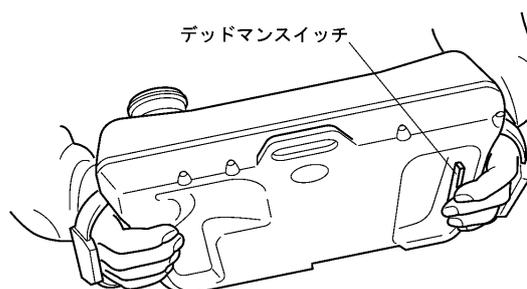
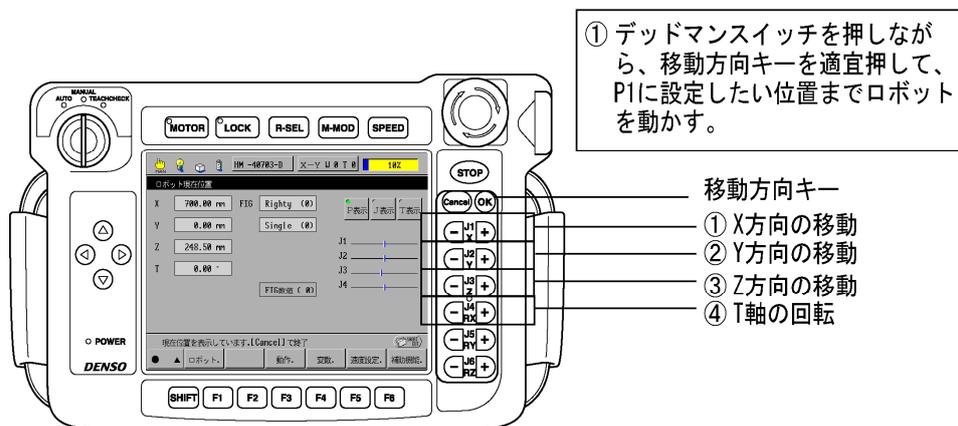
- ・ P 型：位置型変数 (X, Y, Z, RX, RY, RZ, FIG) [6 軸]
(X, Y, Z, T, FIG) [4 軸]
- ・ J 型：ジョイント (軸) 型変数 (J1, J2, J3, J4, J5, J6) [6 軸]
(J1, J2, J3, J4) [4 軸]
- ・ T 型：同次変換型変数 (X, Y, Z, Ox, Oy, Oz, Ax, Ay, Az, FIG) [6 軸/4 軸]

各変数は、最大 32766 個まで設定する事ができますが、コントローラのメモリ容量の使用状況により、設定できる最大数は、減少します。

8.3 位置変数へのティーチング方法

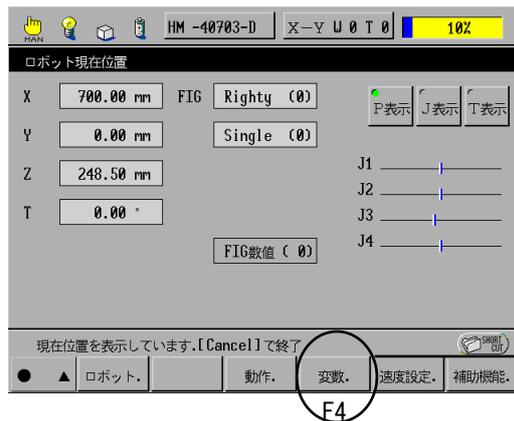
ここでは、ロボットを手動モードで位置変数 P1 と P2 の値をティーチングする方法について説明します。

ステップ1 ロボットの位置（P1）をティーチングする

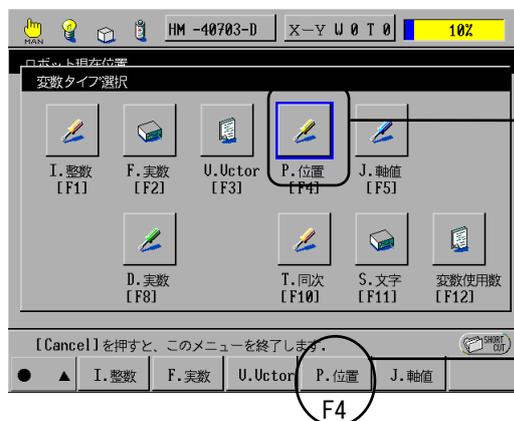


ステップ2

ティーチングした値を「変数名 P1」に保存する



① [F4 変数]を押す。



② [変数タイプ選択]ウインドウで、変数タイプを選ぶ。
ここでは、P変数に位置を保存するので、[F4 P.位置]を押す。
(ウインドウの中にある[P.位置]をタッチしてもよい。)



[位置型] ウィンドウが表示されます。

③ カーソルキーまたはジョグダイヤルを使って、「変数名P1」の欄を選択する。

4 軸ロボットの場合、[位置型] ウィンドウには、一つの変数について 5 種類のデータが表示されていますが、「変数名 P1」の欄であれば、5 種類のデータのうちどれか一つを反転表示させれば、「変数名 P1」を選択したことになります。



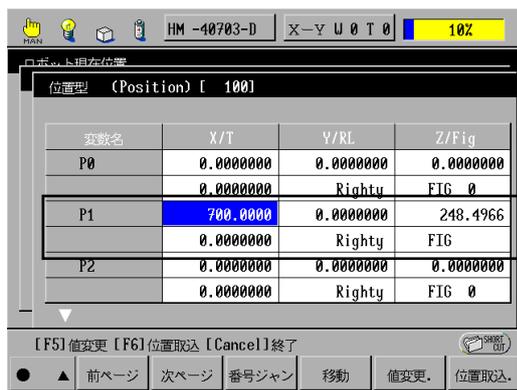
④ 「変数名P1」が選択されていることを確認する。

⑤ [F6 位置取込.] を押す。



⑥ システムメッセージを確認し、良ければ [OK] を押す。

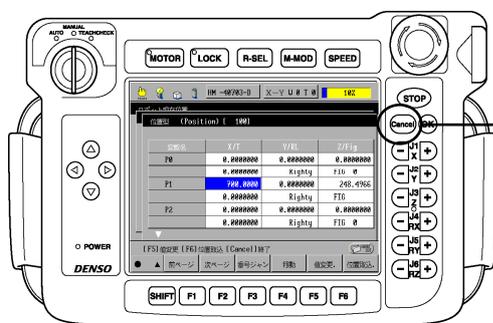




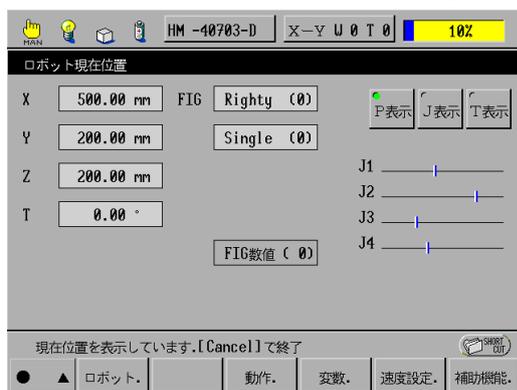
現在位置が変数P1の値として取り込まれています。

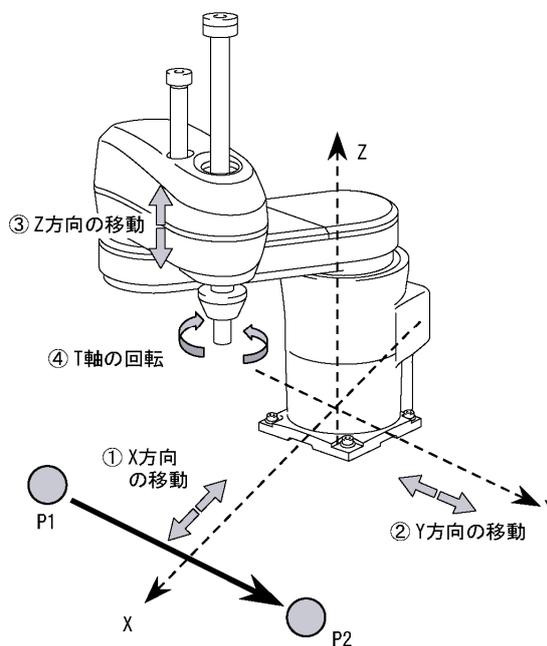
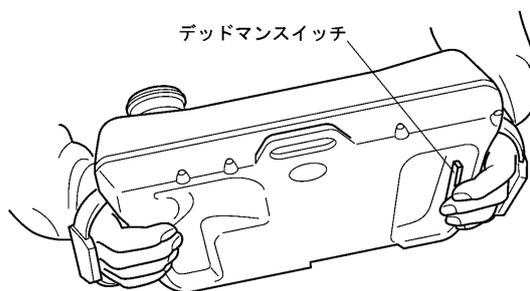
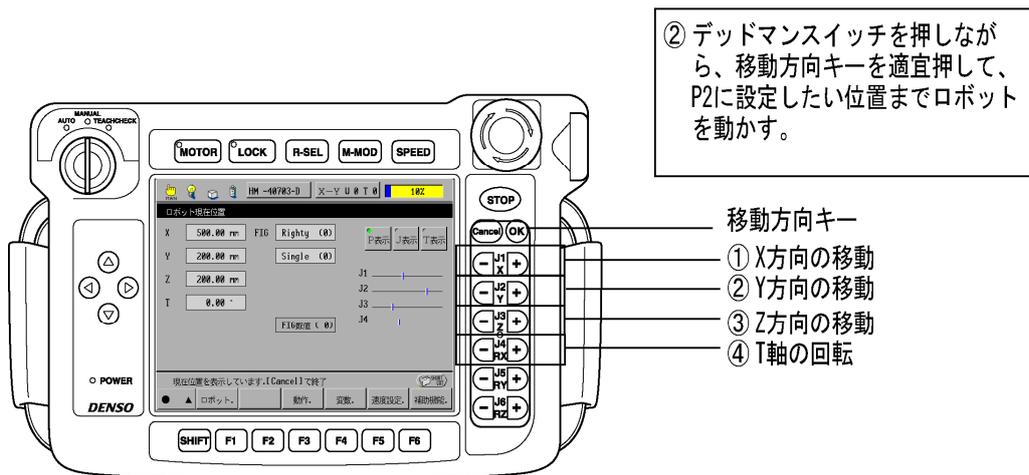
ステップ3

ロボットの位置（P2）のティーチングおよび【変数名P2】への保存



① [Cancel]を2回押して、[ロボット現在位置]ウインドウに戻す。





③ ステップ2「ティーチングした値を[変数名P1]に保存する」の手順に従って、P2の値を[変数名P2]に保存する。

これで、P1 と P2 のティーチングが完了しました。

8.4 ティーチングした位置変数へ直接移動させる方法

手動モード、ティーチチェックモードでは位置変数に登録されているポイントへ直接移動させることができます。

操作経路： [F2 アーム]—[F4 変数.]—[F4 P. 位置]

[F4 P. 位置]を押すと、次のような[位置型]ウィンドウが表示されます。
移動させたい変数番号へカーソルを合わせます。



[F4 移動]を押すと「変数〇〇の位置に動作します」というシステムメッセージが表示されます。デッドマンスイッチ+[OK]を押している間、指定されている位置に向かってロボットアームが移動します。その際は、PTP（目標位置までの動作経路は、ロボットに依存）またはCP動作（目標位置まで動作経路を直線で結ぶ）を指定することもできます。

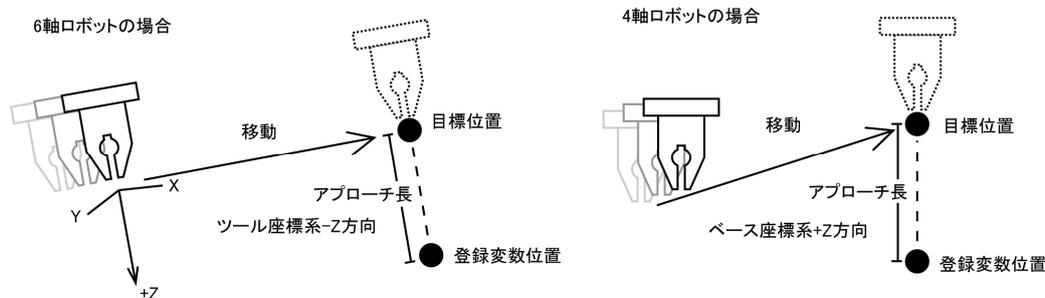
注意：ロボットアームが移動中にデッドマンスイッチもしくは[OK]ボタンを放すと移動は途中停止します。

移動の再開 (Ver. 2.61 以降)

[OK]ボタンを放すと設定した目標位置をそのままに途中停止できます。再度デッドマンスイッチと[OK]を押すことで目標位置への移動を再開できます。キャンセルを押すと、変数画面に戻ります。

8.5 アプローチ長を含む目標位置に移動する (Ver. 2.61 以降)

登録している変数からオフセットした位置を目標位置に設定できます。6軸の場合はツール座標系-Z方向へ、4軸ロボットの場合はベース座標系+Z方向へオフセットします。このオフセット量をアプローチ長と言います(プログラミングマニュアル I APPROACH コマンド参照)。アプローチ長を含む目標位置に移動することで、マニュアルモードで簡単に、より実際のプログラムに近い動きをさせることができます。



変数指定移動画面より[F6 アプローチ]を押すとアプローチ長を指定する画面が表示されます。アプローチ長をテンキー入力します。



目標位置がアプローチ長を含む位置に設定されます。



第3部

簡単なプログラミング

第 9 章 プログラム作成の基礎知識

9.1 PAC 言語の特徴

ロボットの動作や作業を記述するためのプログラミング言語を、ロボット言語といいます。

デンソーロボットで使用されるロボット言語は、PAC(Programming language for Assembly Cell)と名付けられています。PAC は、ロボット制御プログラムの開発とメンテナンスを、従来以上に効率化するために、新しく開発されました。主な特徴を以下に示します。

- ・ JIS で規定された、産業用ロボット言語 SLIM の上位互換言語である。
- ・ 構造化プログラミング言語であるため読みやすく、開発とメンテナンスが容易である。
- ・ ロボットだけでなく、視覚装置の制御も、PAC で統一的に記述できる。
- ・ マルチタスク機能により、プログラムの効率的な処理ができる。
- ・ 割り込み処理機能により、エラーの発生や時刻などによる例外的な処理を、効率よく記述できる。

9.2 文と行

- ・ PAC 言語のプログラムは、複数の行から構成されます。
- ・ 任意の行に、一つの文を記述できます。
- ・ 1 行の長さは、255 バイトまでです。
- ・ 文は、PAC 言語による処理を記述する最小の単位であり、一つのコマンドで構成されます。
- ・ コマンドは、コマンドの名称と、コマンドに与える情報（パラメータ）から構成されます。

9.3 名前

PAC 言語には、プログラムの中にあるさまざまな要素を識別するための規則があります。この章では、この規則について説明します。なお、コマンド、変数、関数、ラベル、プログラムを表す名前は、次の規則に従います。

- ・ 名前は、文字（半角アルファベット。大文字と小文字の区別はしない）または、決められた記号で始まらなければなりません。
- ・ 名前には、文字、数字、アンダースコアが使用できます。
- ・ 名前の先頭文字はアルファベットでなければなりません。
- ・ ピリオド、スラッシュ、バックスラッシュ、スペース、コロンの、セミコロン、シングルクォート、ダブルクォーテーション、アスタリスクは使用できません。
- ・ +、-、*、/、(、)などの、演算子として使われる文字は使用できません。
- ・ 名前を他の語と見分けるために、スペース文字を名前と他の語との間に置きます。
- ・ 名前に使用できる文字数は、最大 64 文字です。

9.4 登録可能な最大プログラム数

1台のコントローラに登録できるプログラム数には制限があります。
また、メモリ容量の使用状況により、登録できるプログラム数は、減少します。

区分	ファイル形式	最大数
PACプログラム	***.pac	256ケ
ヘッダファイル	***.h	ヘッダファイルと操作盤を合わせて 256ケ
操作盤ファイル	***.pnl	
フォルダ		256ケ

9.5 プログラム構成の概要

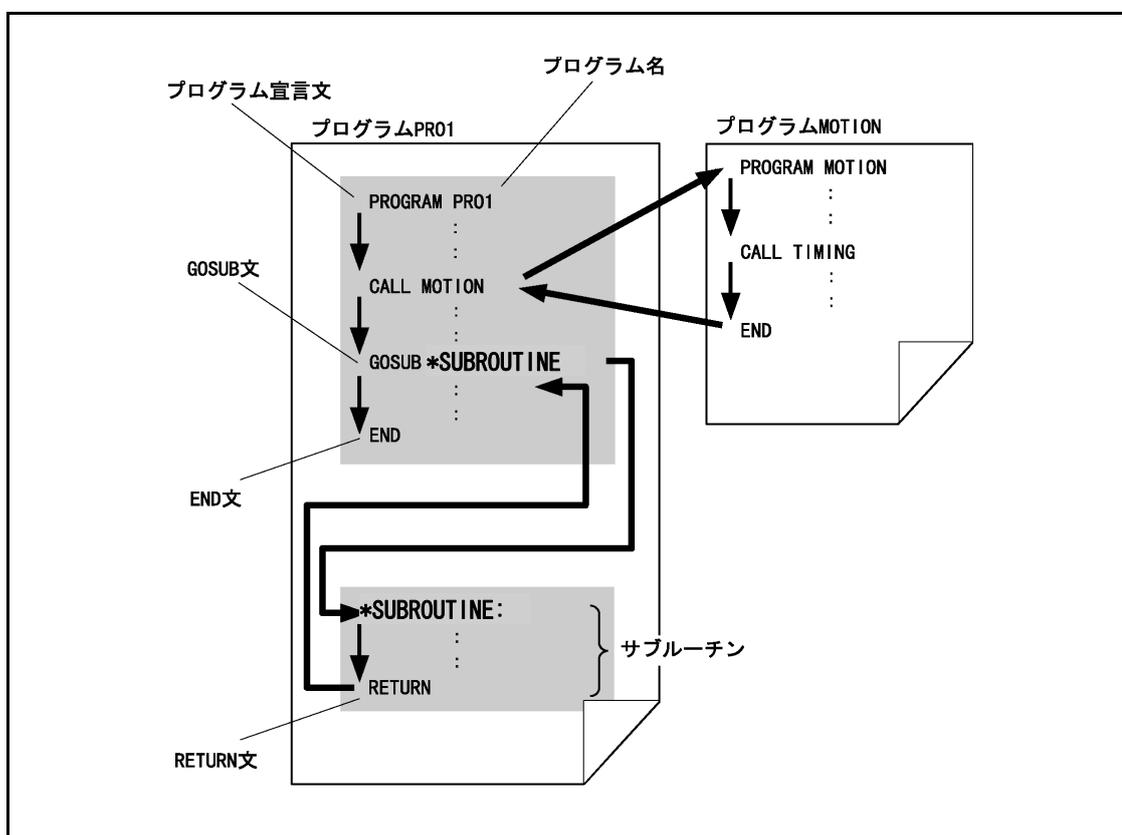
プログラムの中で、特定の動作を繰り返す部分を抜き出しておいて、必要に応じて呼び出すことができます。

同じプログラムファイルの中に、この部分を置く方法を、サブルーチンと呼びます。別のファイルに独立させて、別のプログラムとしておいて、呼び出すことをプログラムの呼び出しといいます。

サブルーチンは、呼び出す側のプログラムと同じファイルの中になくてもなりません。

独立した別のファイルになっているプログラムを呼び出す場合は、一つのプログラムを、いろいろなプログラムから呼び出して、共通に利用することができます。

一連の作業を、サブルーチンや別のプログラムにして、一つにまとめることにより、何度も同じ内容のプログラムを記述しなくて済みます。記述間違いや、プログラム作成時間の削減、またプログラムの読みやすさの向上などにも効果的です。



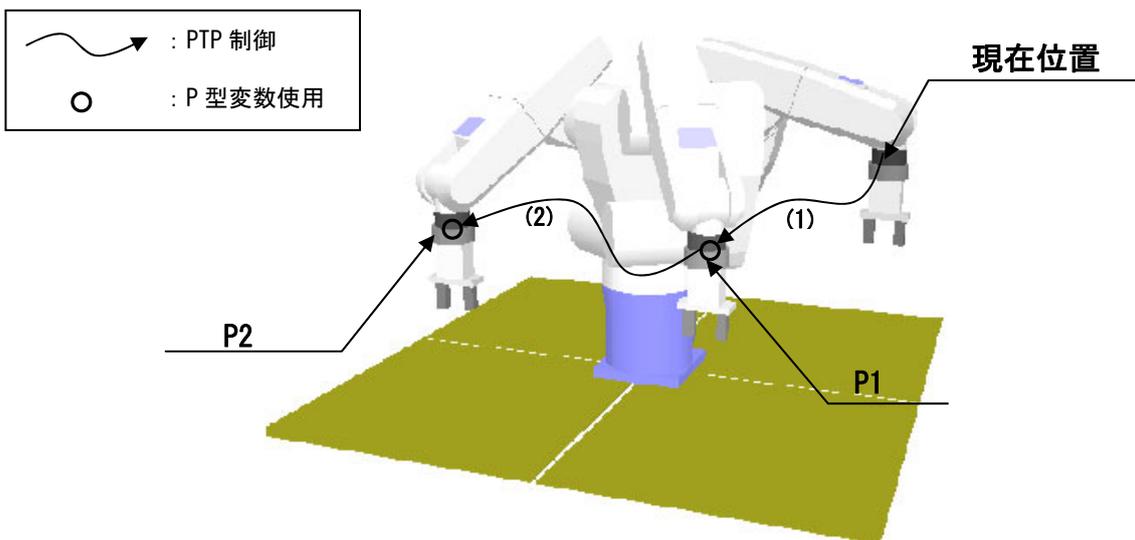
プログラムとサブルーチンの呼び出し方の違い

9.6 プログラムの主なコマンド

プログラム作成で最低限必要なコマンドについて簡単な動作プログラム例を用いて説明します。

9.6.1 プログラム例

現在位置から P1、P2 へ PTP 制御にて順次アームを動作させます。



プログラム	動作説明
' TITLE "プログラムPR01"	' プログラムタイトル
PROGRAM PR01	' プログラム名の宣言
TAKEARM	' アーム制御権取得
SPEED 100	' アーム移動速度(内部速度) 100%
MOVE P, P1	' (1) : 指定座標 P1へ移動
MOVE P, P2	' (2) : 指定座標 P2へ移動
END	' プログラムの停止

9.6.2 コマンド書式で扱う特殊記号について

コマンドの書式解説で使用する特殊記号（`□`、`<>`、`[]`）について下記に説明します。

- 表中の`□`記号は、スペース（空白）を示します。
- `<>`記号は、記述が必要です。
- `[]`記号は、省略可能であり、必要に応じ記述できます。
- 英文字の大文字、小文字の使い分けは必要ありません。

9.6.3 プログラム名の宣言 (PROGRAM コマンド)

説明

プログラム名や変数など、プログラムの実行時に必要な項目は、実行に先立って宣言します。
特にプログラム名の宣言を行なう場合は、プログラムの最初の有効行で宣言しなければなりません。この文を PROGRAM 宣言文 といいます。

書式

PROGRAM □<プログラム名>

注:外部から起動したい場合は、<プログラム名>は、PRO<番号>としてください。

9.6.4 アーム制御権取得 (TAKARM コマンド)

説明

複数のタスク制御するマルチタスク制御では、ロボットの制御権の受け渡しを行う必要があります。ロボットのアームを動作させる動作命令を使用する場合は、必ず TAKEARM 命令を使用し、アーム制御権の取得を行います。

書式

TAKEARM

9.6.5 プログラムの停止 (END コマンド)

説明

このコマンドを実行すると、プログラムによる動作を終了します。

書式

END

9.6.6 アーム移動速度の指定 (SPEED コマンド)

説明

内部速度をパーセント (1~100) で指定します。

外部速度と内部速度の関係は次の通りです。

実際のアーム移動速度 (%) = 外部速度 (%) × 内部速度 (%)

※外部速度とは、ペンダントやPLC等から設定する速度です。

※SPEED 命令は、次の SPEED 命令を実行するまで有効です。

書式

SPEED □<移動速度>

9.6.7 コメントの記述 (REM コマンド)

説明

プログラムにコメントを記述します。

書式

'[<コメント>] (または REM □[<コメント>])

9.6.8 指定座標への移動 (MOVE コマンド)

説明 ロボットを現在位置から目標の位置<ポーズ>へ動作させます。

書式 MOVE □ <補間方法>, [@<パス開始変位>] □ <ポーズ> [, <動作オプション>]

9.6.8.1 補間方法

<補間方法>には、P、L、C の 3 種類の選択が必要です。

P : P T P (Point to Point) 制御動作

- ・ 現在位置から次ポイントへ動作する場合、途中の経路は、ロボット任せです。

L : C P (Continuous Path) 制御動作・・・直線補間

- ・ 現在位置と次ポイントを、ハンドの姿勢や速度を一定に保ちながら直線に動作します。

C : C P (Continuous Path) 制御動作・・・円弧補間

- ・ 3つのポイントを結ぶ円弧上をロボット先端が動作します。

9.6.8.2 ポーズ

<ポーズ>には、移動目標となる動作位置の変数名称を入力します。変数のポジション型 (P 型)、ジョイント型 (J 型)、トランス型 (T 型) が使用できます。また、変数の構成はロボットの軸数によって異なります。

各座標値の数値入力もできますが、実際には 変数 P〇〇 か 変数 J〇〇 をよく使用します。

ロボット	変数名	変数の構成						
4 軸ロボット	P 型	X	Y	Z	T	FIG		
	J 型	J 1	J 2	J 3	J 4			
6 軸ロボット	P 型	X	Y	Z	RX	RY	RZ	FIG
	J 型	J 1	J 2	J 3	J 4	J 5	J 6	

用例

```
PROGRAM PR05      ' プログラム名の宣言
TAKEARM           ' アーム制御権取得
SPEED 80          ' 内部移動速度 80%
MOVE P, P1        ' 変数 P1 へ PTP 補間で移動
MOVE L, P2        ' 変数 P2 へ CP 補間で移動
MOVE L, P3        ' 変数 P3 へ CP 補間で移動
END               ' プログラムの停止
```

9.6.8.3 パス開始変位

〈パス開始変位〉の値は、指定座標（ポーズ）を中心とした球の半径で、動作指令値がその中に入ると次の制御へ移ります。mm 単位で指定します。言い換えれば、動作するポイントの「止まり方」を表しています。一つの動作から次の動作へと移る場合に、以下のものがあります。

エンド動作 (@0、または省略時)

プログラム	動作図	動作波形
<pre>PROGRAM PRO*** TAKEARM MOVE P. @0 P2 MOVE P. @0 P3 ...</pre>	<p>サーボ偏差</p>	

ティーチングされた動作位置（P 2）へ移動してサーボ系への指令値が目標位置に一致したときに目標位置に到達したと判断するものを、エンド動作といいます。また、この点を「End 点」と呼びます。

エンコーダ値確認動作 (@E)

プログラム	動作図	動作波形
<pre>PROGRAM PRO** TAKEARM MOVE P. @E P2 MOVE P. @0 P3 ...</pre>		

ティーチングされた動作位置（P 2）へ移動し、エンコーダ値が目標位置に対し指定パルス（初期値 20）以内に入った時に目標位置に到達したと判断するものをエンコーダ値確認動作といいます。停止精度は向上しますが、サーボ偏差を解消するため、エンド動作に比べ、時間が余分にかかります。

パス動作 (@P)

プログラム	動作図	動作波形
<pre>PROGRAM PRO*** TAKEARM MOVE P. @P P2 MOVE P. @0 P3 ...</pre>		

ティーチングされた動作位置（P 2）の近くを通過する動作を、パス動作といいます。また、この点を「Pass 点」と呼びます。

指定方法

種類	記述方法	意味
エンド動作	省略	デフォルト値 @0 として処理します。
	@0	モータ指令値が、目標位置（指定座標）に達した時点で、次の動作に移ります。一般的なエンド動作です。
エンコーダ値確認動作	@E	エンコーダ値が、目標位置（指定座標）に達した時点で、次の動作に移ります。ロボットが確実に停止します。
パス動作	@P	動作目標位置近傍（半径はコントローラにて自動設定）を通過動作します。一般的なパス動作です。
	@1～@255	指令値が、目標位置から指定した半径（1～255mm）に達した時点で、次の動作に移ります。 注意：半径値はパス開始の目安値で保証値ではありません。

9.6.8.4 動作オプション

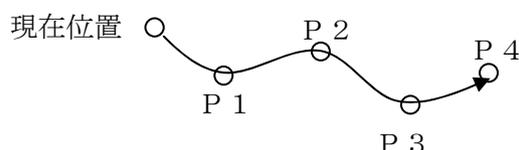
動作オプションには SPEED、ACCEL、DECCEL があります。動作オプション指定は、実行する MOVE 等の動作命令だけに有効となります。

動作オプション	意味
<u>SPEED</u> （または S と表記）	移動速度を指定します。
<u>ACCEL</u>	加速度を指定します。
<u>DECCEL</u>	減速度を指定します。

9.6.8.5 その他の入力例

連続したポイントの動作を、1行で記入することもできます。

MOVE P, @P P1, @P P2, @P P3, @E P4



注意点：「1ステップ」は、P4までの動作全てです。したがって、P1やP2等途中の点で停めることはできません。（ステップ送り、ステップ停止時）

用例

MOVE L, P1, SPEED = 100	‘P1の座標へ（CP制御、内部速度=100%）移動します。
MOVE P, @30 P2, P3, S = 80	‘P2(@30)、P3の座標へ順に（PTP制御、内部速度=80%）移動します。
MOVE L, @20 P4, @50 P5, @100 P6	‘P4(@20)、P5(@50)、P6(@100)の座標へ順に（CP制御）動作します。
MOVE L, @P P[6 TO 15], P16	‘P[6]からP[15]まで順にパス動作で移動し、P16の座標へ（CP制御）動作します。
MOVE C, P1, @P P2	‘P1を通りP2へ移動する円弧補間動作をします。P2にてパス動作し、次の制御に移ります。

9.7 Z軸方向への移動（APPROACH，DEPART）

ロボットが製品をつかみに行く時や置きに行くときに、どこからでも直接移動してしまうと製品やロボットハンドを他の障害物にぶつけてしまいます。このような時は、製品の真上にロボットハンドを移動させ、そこから直線的に下降や上昇をさせた方がよい場合がほとんどです。そこで、Z軸方向専用動作の命令を紹介します。

9.7.1 ハンド方向への接近移動（APPROACH コマンド）

機能

基準動作ポイントの、Z軸上方向に指定した距離の点（アプローチ点）に近づく動作を行います。

書式

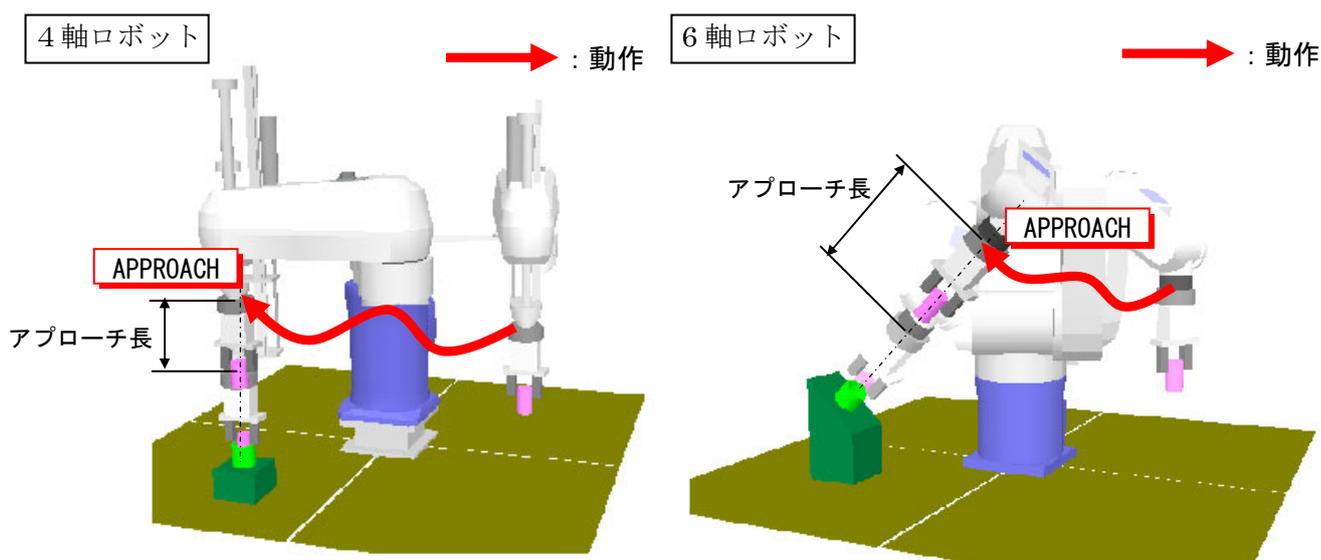
APPROACH □<補間方法>,<基準位置>,[@<パス開始変位>] □<アプローチ長>
[,<動作オプション>]

説明

- ①<補間方法>にはP(PTP制御)、L(CP制御)のどちらかの選択ができます。
- ②<基準位置>にはP型、J型、T型が使用できます。
- ③アプローチ方向はロボットにより下記の様に違いがあります。
 - 4軸：<基準位置>からベース座標系の+Z軸方向へ<アプローチ長>離れた位置へ移動します
 - 6軸：<基準位置>からツール座標系の-Z軸方向へ<アプローチ長>離れた位置へ移動します。
- ④@<パス開始変位>、<動作オプション>の内容はMOVE命令と同じです。

用例

APPROACH P,P1,@P 50 ' 変数P1の50mm上空の点にパス動作にてPTP制御で移動します。



9.7.2 ハンド方向への回避移動 (DEPART コマンド)

機能

現在位置から、Z 軸上方向に指定した距離だけ離れた点 (デパート点) へ動作を行います。

書式

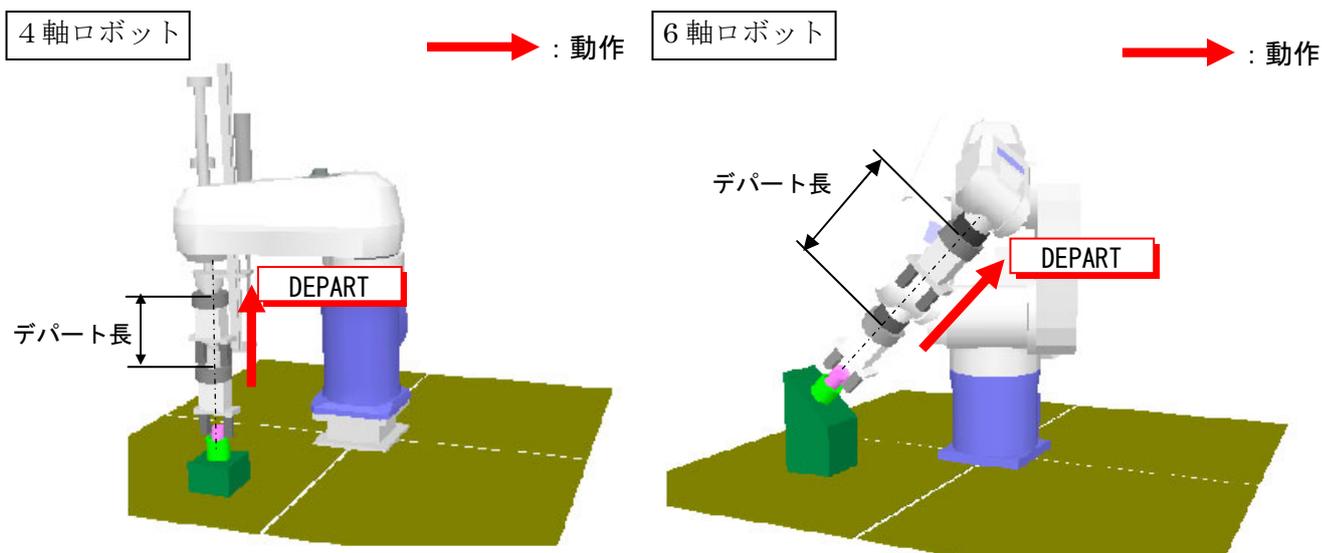
DEPART □ <補間方法>, [@<パス開始変位>] □ <デパート長> [, <動作オプション>]

説明

- ① <補間方法>には P(PTP 制御)、L(CP 制御)のどちらかの選択ができます。
- ② デパート方向はロボットにより下記の様に違いがあります。
4 軸：現在位置からベース座標系の +Z 軸方向へ <デパート長>分移動します。
6 軸：現在位置からツール座標系の -Z 軸方向へ <デパート長>分移動します。
- ③ @<パス開始変位>、<動作オプション>の内容は MOVE 命令と同じです。

用例

DEPART L, @P 50 ' 現在位置の 50mm 上空にパス動作にて CP 制御で移動します。

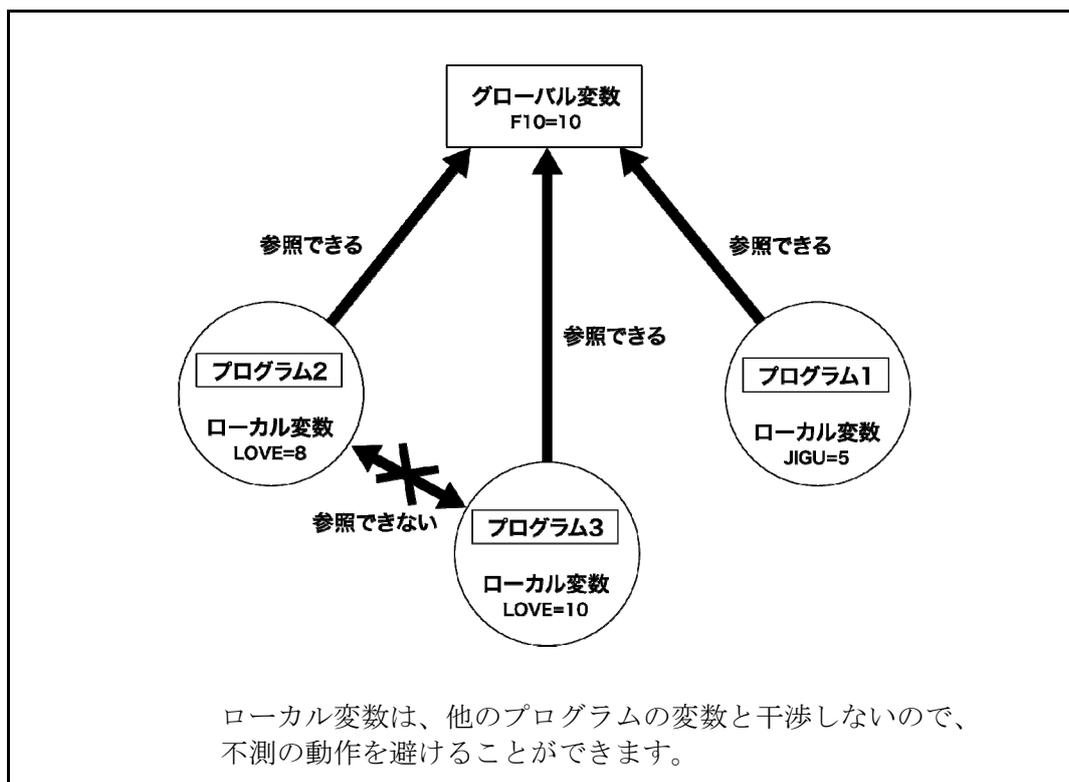


9.8 変数の概要と性質

変数は、プログラム中で使うデータを一時的にしまっておくものです。グローバル変数とローカル変数があります。

グローバル変数は、どのプログラム（タスク）からも共通して使用できる変数です。

ローカル変数は、一つのプログラムの中でのみ有効な変数です。同時に実行される他のプログラムの中に、同じ名前のローカル変数があっても、それぞれが属するプログラムの中で働き、相互に影響することはありません。



グローバル変数とローカル変数

●グローバル変数の性質

- ①全てのプログラムで参照（共有）できる
- ②型宣言の必要がない
- ③マクロ名登録ができる

●ローカル変数の性質

- ①他のプログラムとの干渉を心配する必要がない
- ②コンパイル時に値が初期化される
- ③変数名を自由に決定できる（最大 32 文字）
- ④ 3次元までの配列宣言ができる（配列の要素数は最大 32767 ケ）

9.8.1 グローバル変数

グローバル変数の名前は、型を表すアルファベット (I、F、D、S、V、P、J、T、IO) の後ろに整数式を付けて表します。I/O 変数だけはアルファベットが 2 文字 (IO) になります。

たとえば、F0001、F1、F[1] はすべて、同じ単精度実数型変数を表します。

変数名は、システムで予約されていますから、宣言をすることなく使えます。

グローバル変数には、下記に示す型が使えます。

- I 型：整数型 (範囲：-2147483648～+2147483647)
例) I0001、I1、I[1]
- F 型：単精度実数型 (-3.402823E+38～3.402823E+38)
例) F0001、F1、F[1]
- D 型：倍精度実数型 (-1.7976931348623157D+308～1.7976931348623157D+308)
例) D0001、D1、D[1]
- S 型：文字列型 (最大で 243 文字)
例) S0001、S1、S[1]
- V 型：ベクトル型 (X, Y, Z)
例) V0001、V1、V[1]
- P 型：ポジション型 (X, Y, Z, RX, RY, RZ, FIG) (6 軸)
(X, Y, Z, T, FIG) (4 軸)
例) P0001、P1、P[1]
- J 型：ジョイント型 (J1, J2, J3, J4, J5, J6) (6 軸)
(J1, J2, J3, J4) (4 軸)
例) J0001、J1、J[1]
- T 型：同次変換型 (X, Y, Z, Ox, Oy, Oz, Ax, Ay, Az, FIG) (6 軸/4 軸)
例) T0001、T1、T[1]
- IO 型：I/O 型
例) IO0001、IO1、IO[1]

使用例

I[1]=I[2]*I[3]

F[10]=50.3

S[3]=" DENSO" +S[5]

J[5]=(10, 20, 30, 40, 50, 60)

P[1]=P[4]

9.8.2 ローカル変数

ローカル変数には、グローバル変数と同様に、下記の型の変数が使えます。

- I 型：整数型（範囲：-2147483648～+2147483647）
- F 型：単精度実数型（-3.402823E+38～3.402823E+38）
- D 型：倍精度実数型（-1.7976931348623157D+308～1.7976931348623157D+308）
- S 型：文字列型（最大で 243 文字）
- V 型：ベクトル型（X, Y, Z）
- P 型：ポジション型（X, Y, Z, RX, RY, RZ, FIG）（6 軸）
（X, Y, Z, T, FIG）（4 軸）
- J 型：ジョイント型（J1, J2, J3, J4, J5, J6）（6 軸）
（J1, J2, J3, J4）（4 軸）
- T 型：同次変換型（X, Y, Z, Ox, Oy, Oz, Ax, Ay, Az, FIG）（6 軸/4 軸）
- IO 型：I/O 型

ローカル変数は、型宣言命令を使用して、型宣言をした後に使用できます。

数値型と文字列型のローカル変数は、型宣言文字を使用して型宣言を行なうこともできます。

ローカル変数の宣言方法

ローカル変数の宣言には、下記 3 種類の方法があります。

種類	宣言例 1	宣言例 2	宣言例 3
I 型	DEFINT denso	DIM denso AS INTEGER	denso%
F 型	DEFSNG denso	DIM denso AS SINGLE	denso!
D 型	DEFDBL denso	DIM denso AS DOUBLE	denso#
S 型	DEFSTR denso	DIM denso AS STRING	denso\$
V 型	DEFVEC denso	DIM denso AS VECTOR	
P 型	DEFPOS denso	DIM denso AS POSITION	
J 型	DEFJNT denso	DIM denso AS JOINT	
T 型	DEFTRN denso	DIM denso AS TRANS	
I/O 型	DEFIO denso		

使用例

```

DEFINT Denso , Robo      ' I 型変数 Denso と Robot を宣言
DEFDBL AA                ' D 型変数 AA を宣言
DEFIO Port = BYTE, 104, &B00101011 ' I/O 型変数 Port を宣言し入力ポート 104
                                ' から始まる 8 ビット (BYTE) を使用
CC% = Denso * 2          ' I 型変数 CC を宣言し、Denso*2 の結果を代入
DD$ = "Denso Robot"     ' S 型変数 DD を宣言し文字列 "Denso Robot" を代入
AA = F[5] / 5           ' D 型変数 AA に右辺算術結果を代入
IN Robo = Port           ' I 型変数 Robo に Port の I/O データを 10 進変換し代入

```

9.9 外部機器から起動できるプログラム名

外部自動モードの場合には、外部 I/O からの入力信号によりプログラムを起動する事ができます。

起動できるプログラムは「PROGRAM PRO<番号>」の形式のプログラム名のプログラムに限られます。

また、使用する I/O 割付モードにより、起動できるプログラム番号は制限されます。

I/O 割付モード	起動可能 PRO<番号>
Mini I/O モード	PRO0 ~ PRO7
標準モード	PRO0 ~ PRO32767
互換モード	PRO0 ~ PRO127

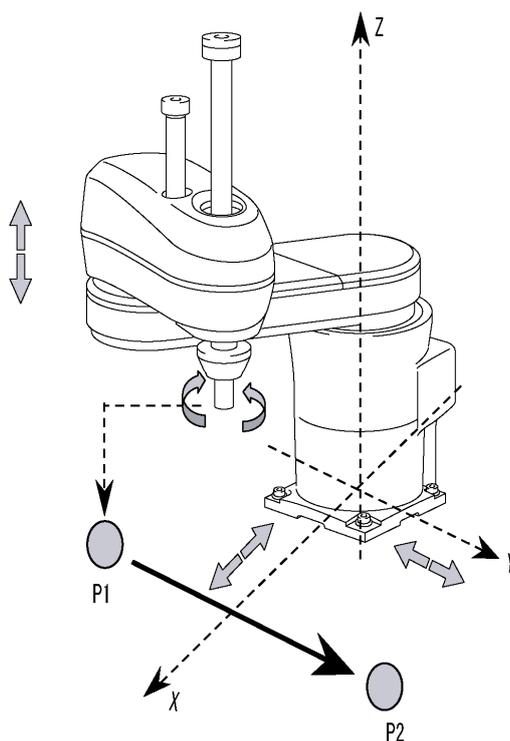
注意：外部機器から起動できるプログラムは、コントローラのルートに保存されているプログラムのみです。フォルダ機能を使ったプログラムを作成する場合には、プログラムの保管場所にご注意ください。

第10章 プログラムの作成 (ティーチングペンダント編)

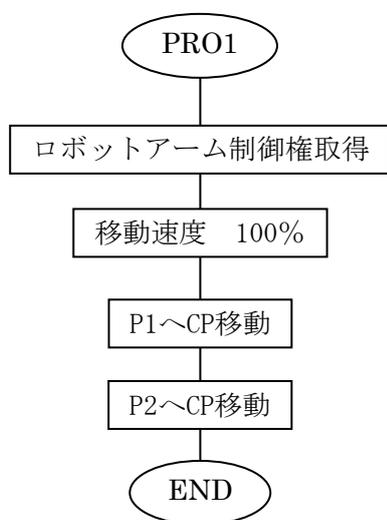
ここでは、ティーチングペンダントを使用したプログラム入力方法について説明します。

10.1 作成するサンプルプログラム

現在位置から P1 へ移動し、その後 P2 へ移動させるプログラムを作成します。



プログラムフローチャート

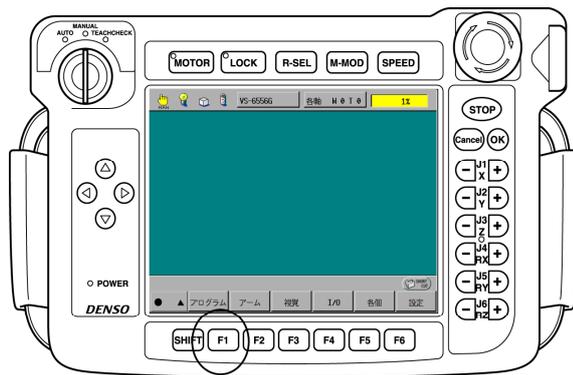


10.2 プログラムの作成

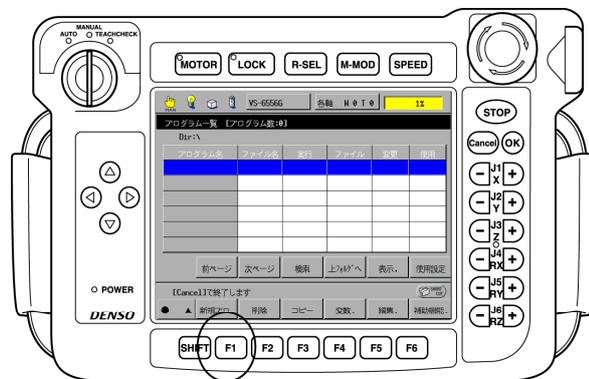
ティーチングペンダントを使って簡単なプログラムを実際に入力します。
プログラムの作成と編集は、手動モードで行ないます。

10.2.1 新規プログラム名の入力

新規のプログラムを入力するために、ティーチングペンダントの画面にプログラム編集のためのウインドウを開きましょう。



① 基本画面で[F1 プログラム]を押す。

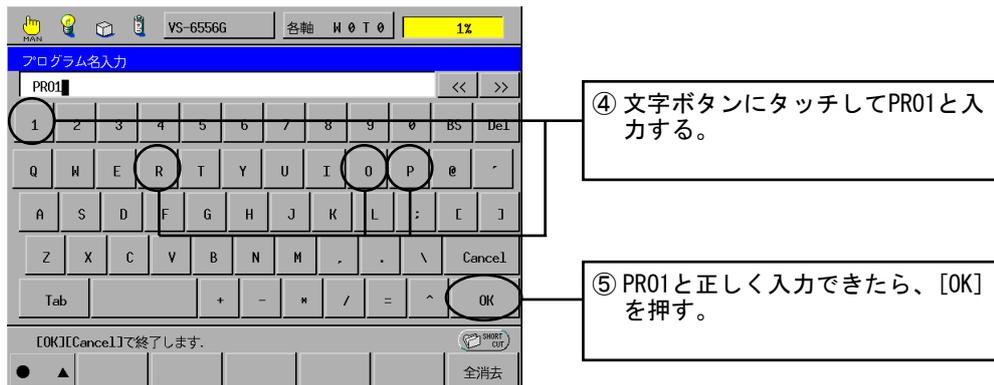


② [F1 新規プロ.]を押す。



③ [プログラム]を選択し、[OK]を押す。

これから作成するプログラムの名称（ここではPR01）を入力しましょう。



これでプログラム編集の準備が完了です。

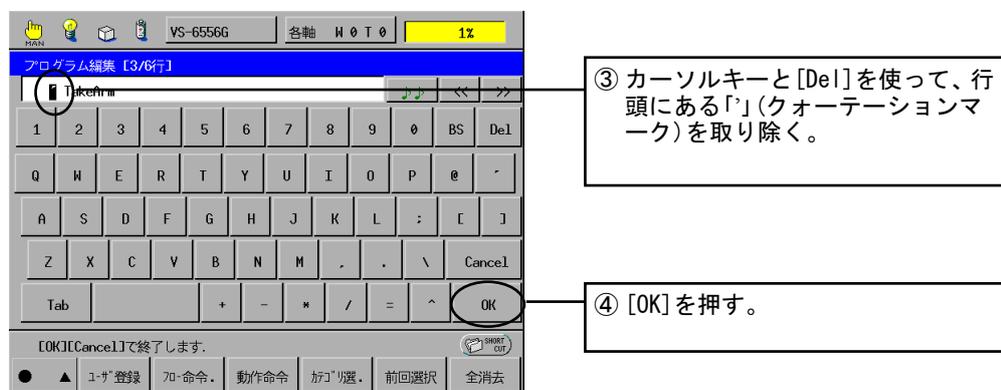
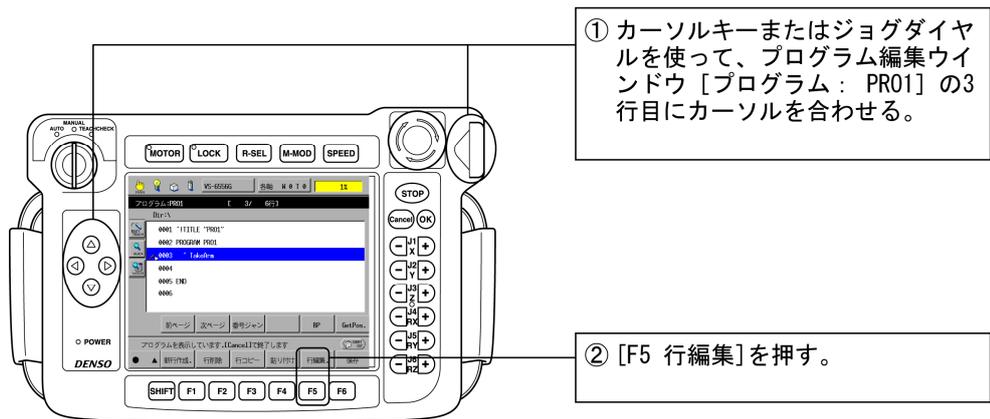


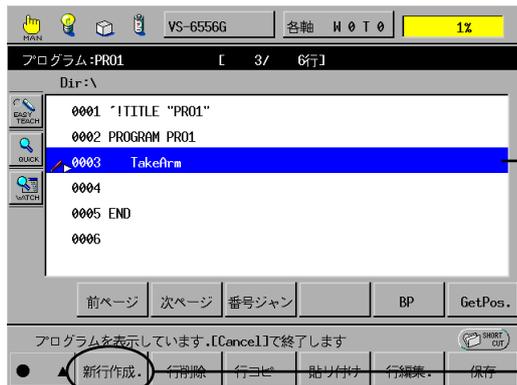
10.2.2 プログラムの入力

ここでは、P1からP2へ移動するプログラムを作成してみましょう。下表に示すプログラムコードを入力してください。

プログラム「PR01」

PROGRAM	PR01	
TAKEARM		‘アーム制御権を取得します
SPEED	100	‘手先の内部移動速度を指定します
MOVE	L, P1	‘指定座標 P 1 へ移動します
MOVE	L, P2	‘指定座標 P 2 へ移動します
END		





⑤ カーソルを3行目に合わせて、
[F1 新行作成]を押す。



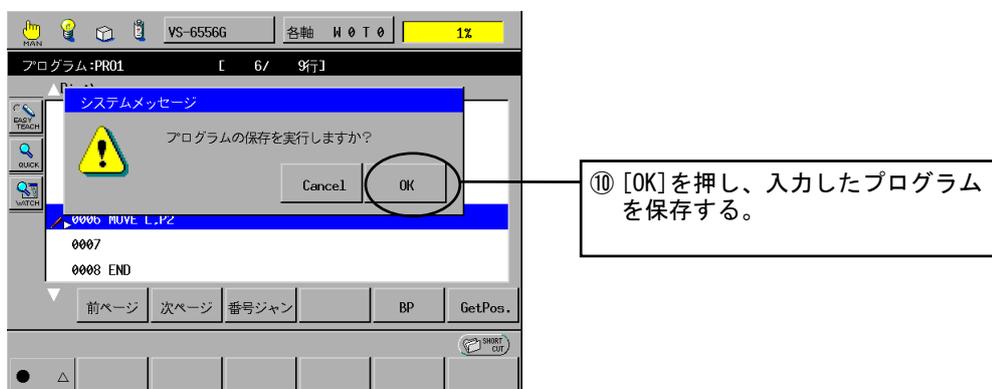
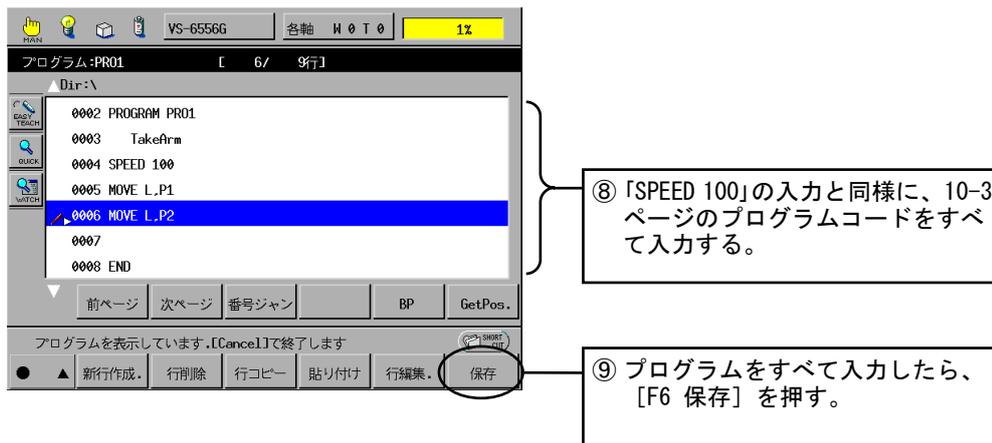
⑥ キーボードから「SPEED 100」と入力
する。この入力ウィンドウに表示
される。

⑦ [OK]を押す。



プログラム編集ウィンドウ[プログラ
ム：PR01]になり、4行目に「SPEED
100」が表示されます。





表示は、[プログラム一覧] ウィンドウに戻ります。



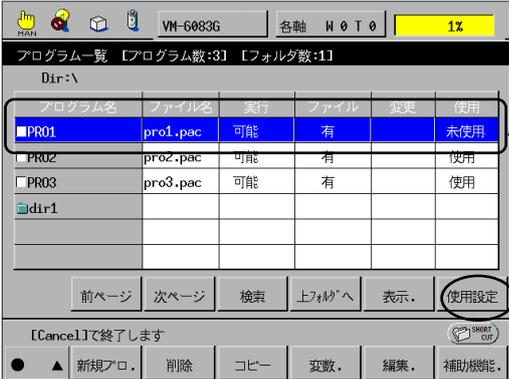
★注意★

- ① 変更結果を保存したくない場合は、ここで[Cancel]を押せば、保存せずに編集画面に戻ります。
- ② プログラムを新たに入力する場合は、ステップ1からの操作を同様に行なってください。

10.2.3 プログラムのコンパイル

プログラムの編集ができれば、プログラムをロボットコントローラが理解できるように、実行形式に変換します。プログラムを実行形式に変換することを、コンパイルといいます。

作成したプログラムに文法的なエラーがあれば、実行形式に変換中に検出されます。文法エラーが検出された場合には、プログラムのロードや、プログラムの実行をすることができません。



プログラム名	ファイル名	実行	ファイル	変更	使用
PR01	pro1.pac	可能	有		未使用
PR02	pro2.pac	可能	有		使用
PR03	pro3.pac	可能	有		使用
dir1					

① [プログラム一覧] ウィンドウで、「PR01」を選択する。
(カーソルキーまたはジョグダイヤルを操作するか画面にタッチすると選択できる。)

② [F12 使用設定] を押す。



③ 「このプログラムのみ使用状態に」を選択する。
※全プログラム使用状態を選択しても構いません。

④ [OK] を押す。



⑤ [OK]を押す。
コンパイルが始まります。

★注意★

- ① ここで、[OK]の代わりに[Cancel]を押すと、コンパイルせずに[プログラム一覧]ウィンドウに戻ります。
- ② プログラムを実行形式に変換するには、もう一つの操作方法があります。
[プログラム一覧]ウィンドウで、[F6 補助機能]を押すと、[補助機能(プログラム)]ウィンドウが表示されます。
[補助機能(プログラム)]ウィンドウに現れる[F12 コンパイル]によって、プログラムを実行形式に変換することができます。



コンパイルが終了すると、続いて（自動）プロジェクトのロードを行ないます。



文法エラーが無ければ、「ローカル変数を初期化しました」が表示されます。



10.2.4 プログラムのロード

実行形式に変換されたプログラムを、コントローラが実行できるように準備する必要があります。これを、プログラムのロードといいます。

コントローラに接続されたパソコンから、実行形式のプログラムを転送した場合も、そのプログラムを実行するためにロードが必要です。

プログラムデータは、パソコンから転送しただけでは、ロボットコントローラで使えません。実行するためのメモリ領域にロードする必要があります。

① 基本画面を表示する。
(別の画面になっているときは、[Cancel]を何度か押して基本画面に戻す。)

② 基本画面で[F6 設定]を押す。

[設定(メイン)]ウィンドウが表示されます。

③ [F1 ロード]を押す。

④ [OK]を押す。



メッセージ「お待ちください。プロジェクトをロード中です。」が表示されます。



ロードが完了すると[設定 (メイン)] ウィンドウに戻ります。

★注意★

以前のプロジェクトと異なるローカル変数を使用するプロジェクトをロードしたときは、エラーメッセージ「ローカル変数を初期化しました。」が表示されます。

[OK]を押すと、次に進みます。



⑤ [OK]を押す。

以上の作業により、プログラムが実行できるようになります。基本画面に戻るには、[Cancel]を押します。

これで、ロボットを動作させるプログラムの作成が完了です。

第 11 章 プログラムの作成 (WINCAPSⅢ編)

ここでは、WINCAPSⅢを使用したプログラム入力方法について説明します。

11.1 プログラム作成の準備

プログラムを作成するための準備事項を説明します。

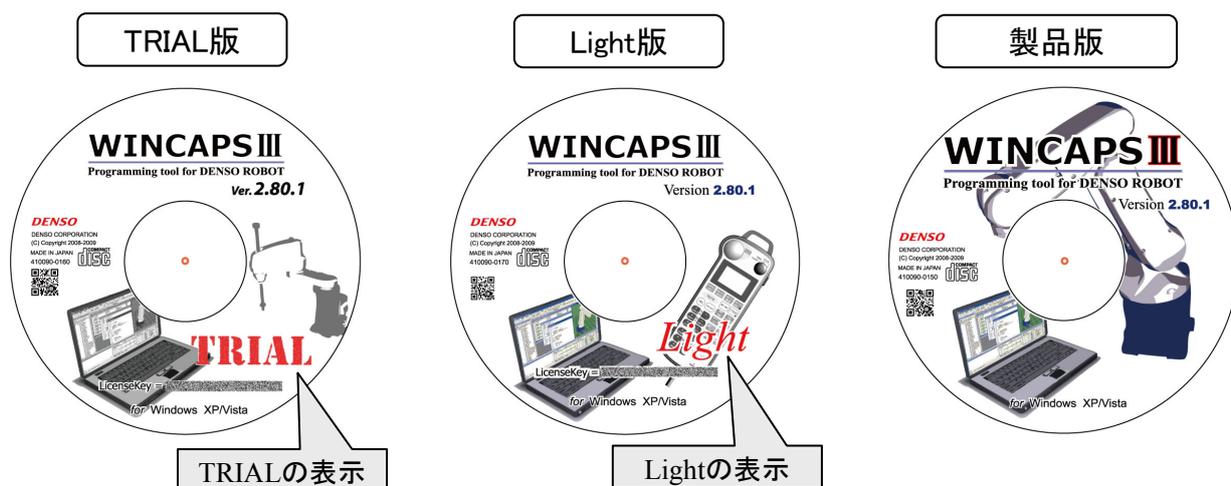
11.1.1 WINCAPSⅢの種類

WINCAPSⅢのインストール CD は全部で 3 種類の製品を用意しています。
種類別に、使用できる機能が制限されています。

1. TRIAL 版 : ロボットに同梱しています。印刷、アームプレーヤ Plus、3D データインポート、モニタ間隔およびプログラムバンクの一部の機能が使用できません。編集できるプログラムは「PR01.pac」のみです。
2. Light 版 : ミニペンダント (オプション品) に同梱しています。印刷、アームプレーヤ Plus、3D データインポート、モニタ間隔およびプログラムバンクの一部の機能が使用できません。
3. 製品版 : オプションにて購入ください。製品版 CD にはライセンス証が同梱されます。

【補足】 ロボット購入時に、ミニペンダントとWINCAPSⅢ (製品版) を同時購入すると、3種類のCD (TRIAL版・Light版・製品版) を入手することになります。

11.1.2 CD 外観 (CD レーベル)



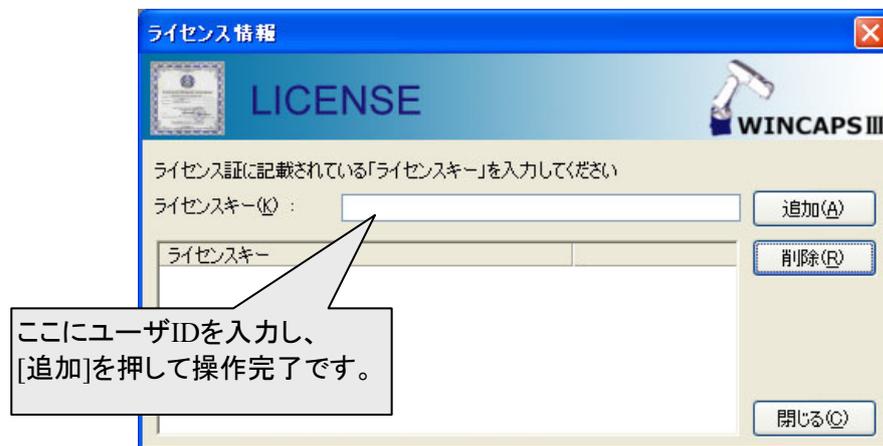
11.1.5 バージョンアップ時の注意

WINCAPSⅢは、[TRIAL 版] → [Light 版] → [製品版] の順で上位製品となります。バージョンアップは、いずれの製品を使用しても可能です。但し、その場合は、インストール済みの製品と比較して上位製品の方をインストールします。例えば、[Light 版] がインストール済みのパソコンを [TRIAL 版] を使用してバージョンアップした場合は、[Light 版] として機能します。

【補足】 インストール済みの製品に限らず、ユーザIDを入力すると製品版として機能します。

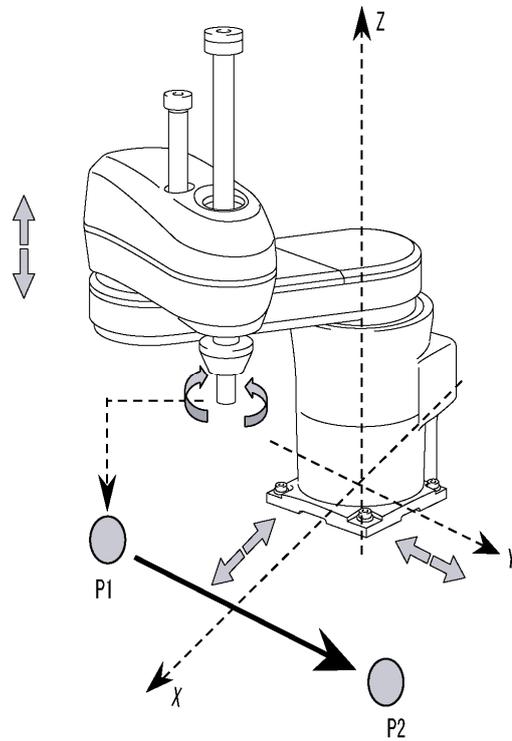
11.1.6 ユーザ ID の入力方法

製品版へアップする場合は、ご購入戴いたライセンス証に記載されているユーザ ID を WINCAPSⅢの「ライセンス情報」へ入力します。下記画面を表示するには、[ヘルプ (H)]-[ライセンス情報 (L)]を選択してください。

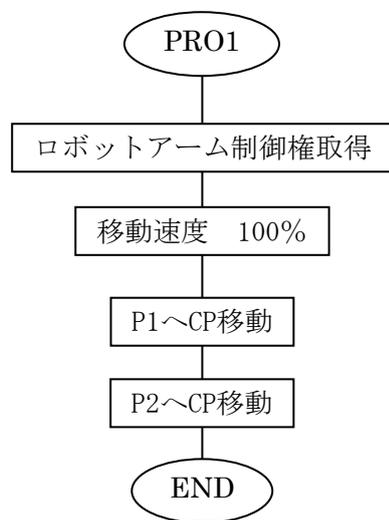


11.2 作成するサンプルプログラム

現在位置から P 1 へ移動し、その後 P 2 へ移動させるプログラムを作成します。



プログラムフローチャート



11.3 プログラムの作成

WINCAPSⅢを使って簡単なプログラムを実際に作成します。

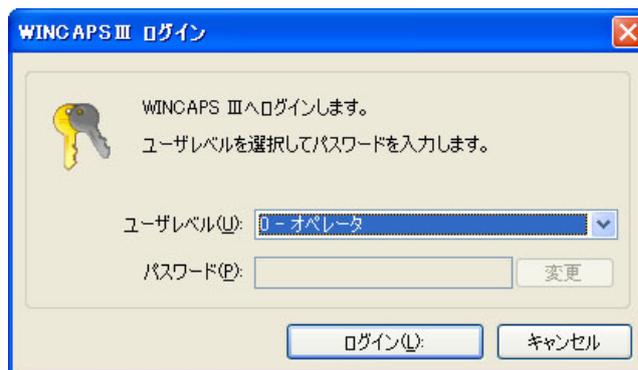
11.3.1 WINCAPSⅢの起動

プログラミング支援ツール「WINCAPSⅢ」を起動します。

ステップ 1

[スタート]メニューの[全てのプログラム]から[DENSO FACTORY WARE]-[WINCAPSⅢ]-[WINCAPSⅢ]を選択します。

[WINCAPSⅢ ログイン]ダイアログを表示します。



ステップ 2

オペレータレベルでログインします。

[ユーザレベル (U)]で[0-オペレータ]を選択し、[ログイン (L)]を押します。設定変更などを行う場合は[1-プログラマ]レベルでログインする必要があります。プログラマレベルでのログインにはパスワードが必要です。はじめてのプログラマレベルでのログインではパスワードの設定を行います。

11.3.2 プロジェクトの新規作成

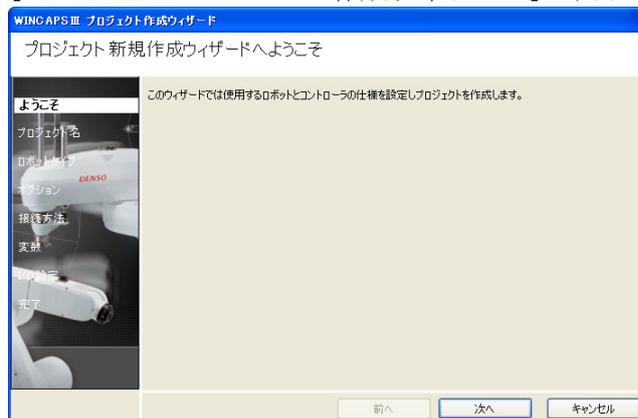
WINCAPSⅢではプロジェクトと言う単位で複数のプログラムを管理できます。プロジェクト内で複数のプログラムを作成し、組み合わせて使用することにより、より分かりやすく管理することができます。

プログラムを作成するにあたって、プロジェクトを新しく作成します。

ステップ 1

[ファイル(F)]-[プロジェクトの新規作成(N)]を選択します。

[WINCAPSⅢ プロジェクト作成ウィザード]が表示されます。

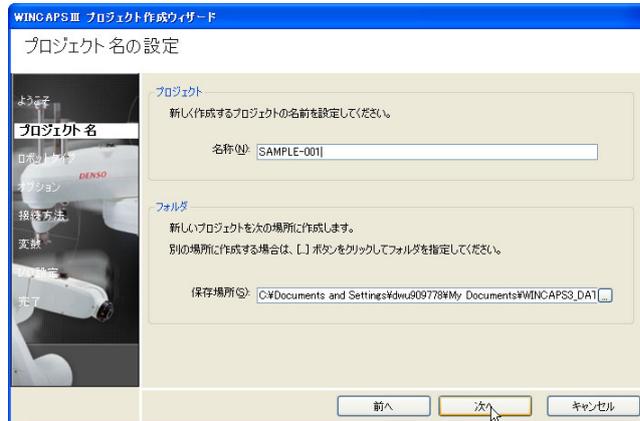


[次へ]をクリックします。

ステップ 2

プロジェクト名とプロジェクトの保存場所を入力します。

入力したプロジェクト名のフォルダが指定した保存場所に作成されます。



[次へ]をクリックします。

ステップ 3

ロボットタイプを選択します。

ご使用されるロボットコントローラ及びロボットタイプを選択し、[次へ]をクリックします。



ステップ 4

コントローラオプションを選択します。

ご使用のコントローラの仕様(標準/グローバル)と付加軸の使用の有無を選択し、[次へ]をクリックします。



ステップ5

コントローラと PC (WINCAPSⅢ) との接続方法を入力します。

接続方法(イーサネット/RS-232C)とその詳細を入力し、[次へ]をクリックします。
接続方法の設定はプロジェクト作成後でも変更できます。

WINCAPSⅢ プロジェクト作成ウィザード

接続方法の選択

コントローラの接続方法を設定してください。

イーサネット(E)

IP アドレス(I): 192 168 0 1

RS-232C(C)

通信ポート(P): COM1

ボーレート(R): 38400

パリティ(P): N - パリティなし

データビット長(L): 8

ストップビット(S): 1

前へ 次へ キャンセル

ステップ6

変数の使用個数を入力します。

変数の使用個数はプロジェクト作成後でも変更できます。

WINCAPSⅢ プロジェクト作成ウィザード

変数の設定

コントローラの変数の個数を設定してください。

変数使用個数

I 型(I):	100	P 型(P):	100
F 型(F):	100	J 型(J):	100
D 型(D):	50	T 型(T):	50
V 型(V):	50	S 型(S):	50

前へ 次へ キャンセル

ステップ7

I/O ポートの設定を入力します。

コントローラに取り付けられているデバイスと割付モードを選択します。
必要に応じてデバイスの詳細設定を入力します。
[次へ]をクリックします。

WINCAPSⅢ プロジェクト作成ウィザード

I/Oポートの設定

I/O割付設定をしてください。

割付モード

デバイス(D): Parallel

モード(M): 互換

オプション

DeviceNet	CC-Link	PROFIBUS
入力ポート数(I): 8	局番(N): 1	入力バイト数(I): 8
出力ポート数(O): 7	占有局数(O): 4	出力バイト数(O): 8
ノードアドレス(A): 0	ボーレート(R): 10M	ノードアドレス(A): 1
通信速度(S): 500KB		

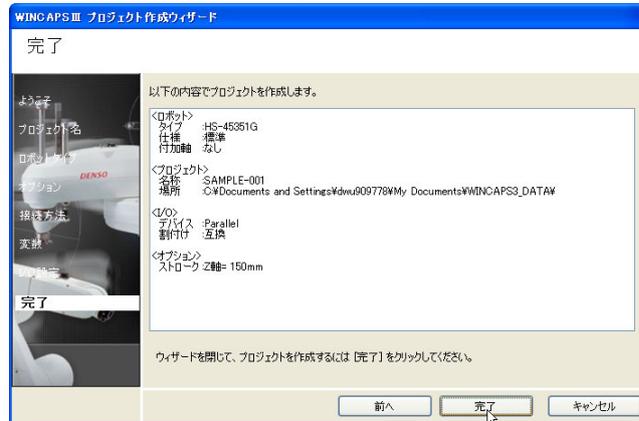
標準値に戻す

前へ 次へ キャンセル

ステップ 8

設定内容を確認します。

設定内容が正しければ[完了]をクリックし、ウィザードを終了します。



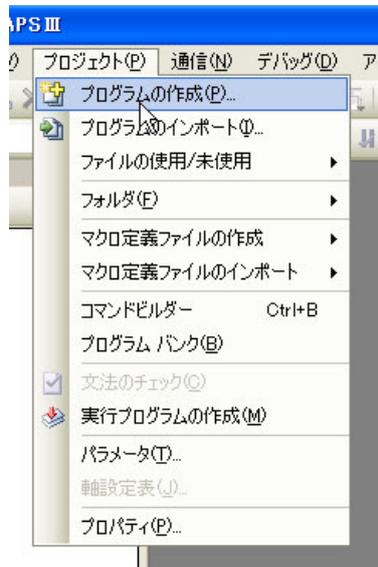
プロジェクトが作成されます。

11.3.3 プログラムの作成

プロジェクト内にプログラムを作成します。

ステップ 1

[プロジェクト(P)]-[プログラムの作成(P)]を選択します。



ステップ 2

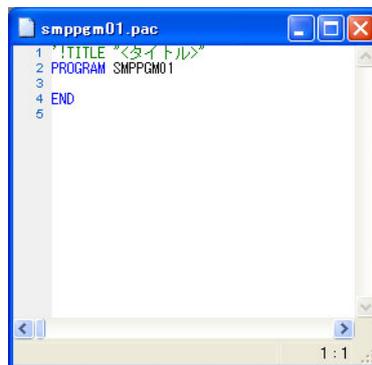
[種類]から[プログラム(*.PAC)]を選択し、[プログラム名]と[ファイル名]を入力します。

プログラム名はアルファベットから始まる半角英数字で、64文字以下でつけなければなりません。
プログラム名を入力するとファイル名にも自動入力されますが、別の名前を付けたい場合にはファイル名を再入力してください。
入力し終わったら、[OK]をクリックします。



ステップ 3

プログラムが作成され、プログラム入力ウィンドウが表示されます。



11.3.4 プログラムの入力と保存

プログラム入力ウィンドウに、下記のプログラム例を入力します。

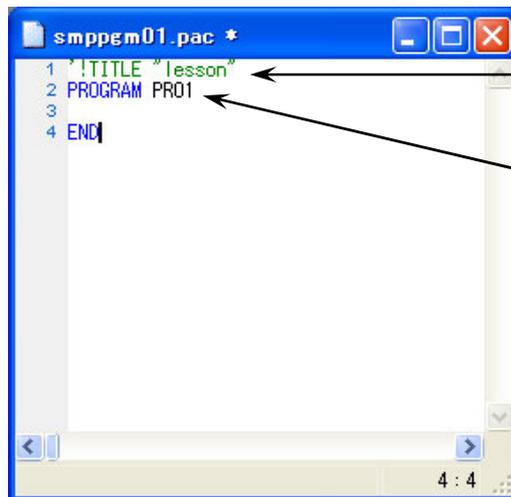
このプログラム例は登録点 P1 から登録点 P2 へ移動するものです。
下表に示すプログラムコードを入力してください。

プログラム例__プログラム「PR01」

PROGRAM PR01	‘プログラム名「PR01」の宣言
TAKEARM	‘アームの制御権を取得します。
SPEED 100	‘手先の内部速度を100 (%) にします。
MOVE L, P1	‘登録点 P 1 へ移動します。
MOVE L, P2	‘登録点 P 2 へ移動します。
END	

ステップ 1

プログラムタイトルとプログラム名を入力する



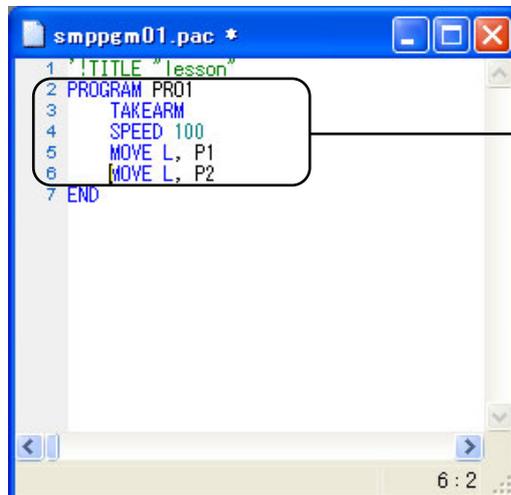
```
1 !TITLE "lesson"  
2 PROGRAM PR01  
3  
4 END
```

① プログラムのタイトルを入力します。(ここでは"lesson"というタイトルを入力します。)

② プログラム名を入力します。(ここでは"PR01"と入力します。)(大文字、小文字どちらでも可)

ステップ 2

プログラムコマンドを入力する



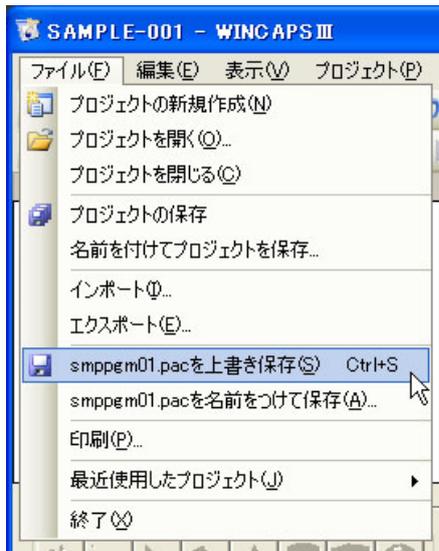
```
1 !TITLE "lesson"  
2 PROGRAM PR01  
3 TAKEARM  
4 SPEED 100  
5 MOVE L, P1  
6 MOVE L, P2  
7 END
```

③ PR01 のプログラムコマンドを入力します。

ステップ 3

作成したプログラムを保存する

[ファイル(F)]-[smppgm01.pac を上書き保存 (S)]を選択します。



プログラムが保存されます。

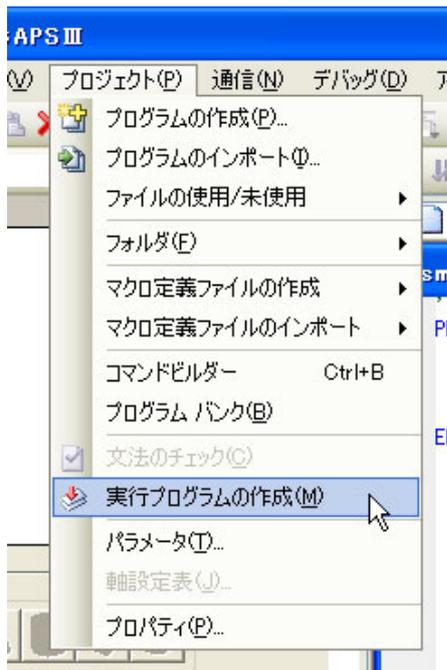
11.3.5 プログラムのコンパイル

PAC 言語は、入力したプログラムを実行するために、あらかじめロボットコントローラが理解できるデータ形式に変換（コンパイル）しておく必要があります。コンパイルされたものが「実行プログラム」です。

ステップ 1

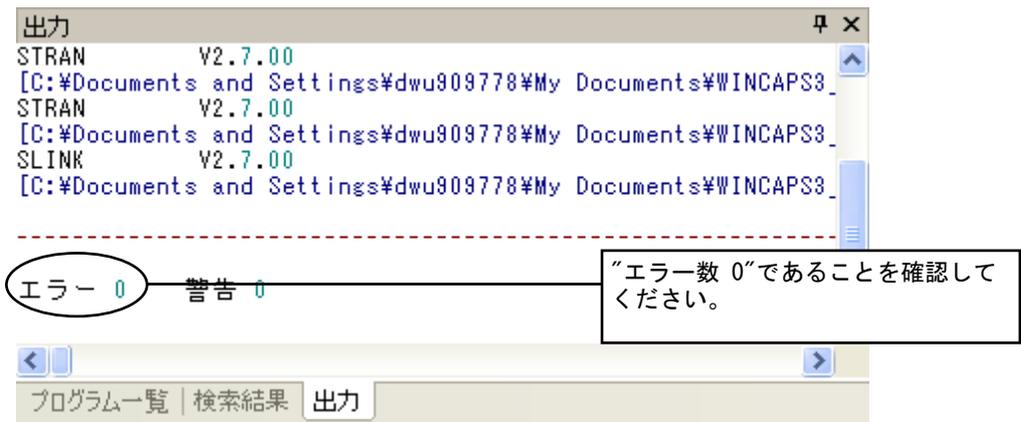
実行プログラムに変換する

[プロジェクト (P)]-[実行プログラムの作成 (M)]を選択すると、[プログラム一覧] ウィンドウ内の全てのプログラムが実行プログラムに変換されます。

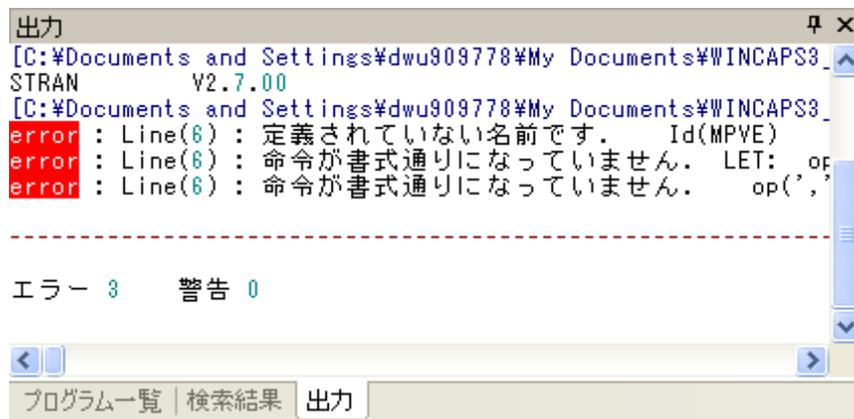


ステップ2

エラーのないことを確認する



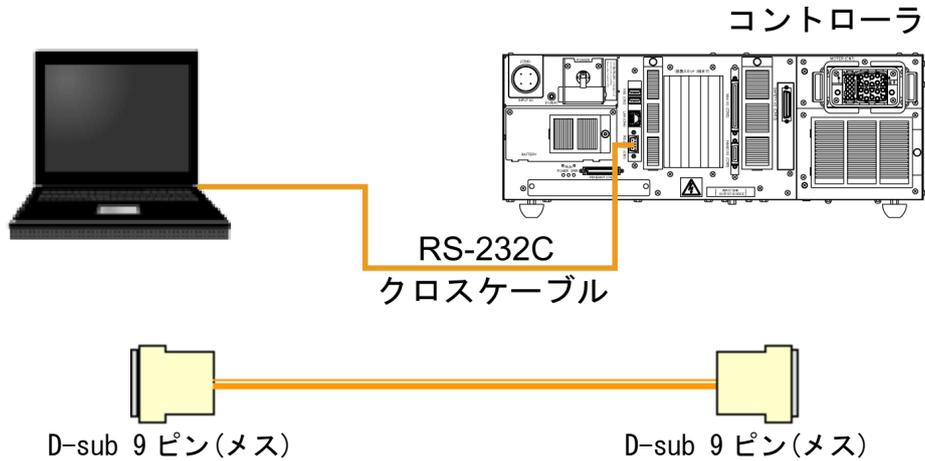
エラーが出た場合はプログラムコマンドの入力に間違いがあります。
[出力]ウインドウにエラー箇所とその内容などが表示されますので、参考にしてプログラムを確認し、コマンドを再入力してください。



11.4 WINCAPSⅢとコントローラの通信ケーブル接続方法

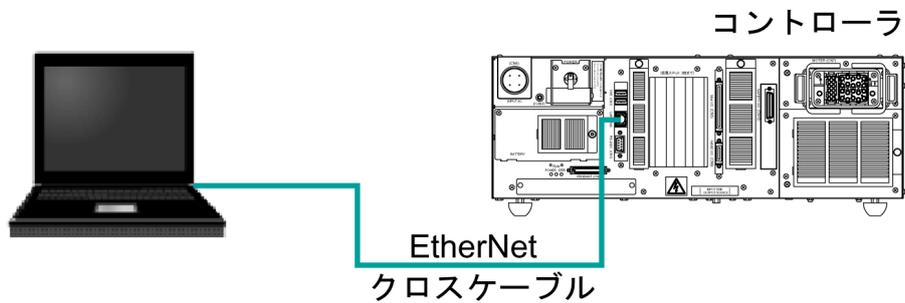
WINCAPSⅢ (PC)とロボットコントローラとのデータ通信のために、通信ケーブルで接続する必要があります。下記の方法でケーブル接続して下さい。

11.4.1 RS-232C 接続

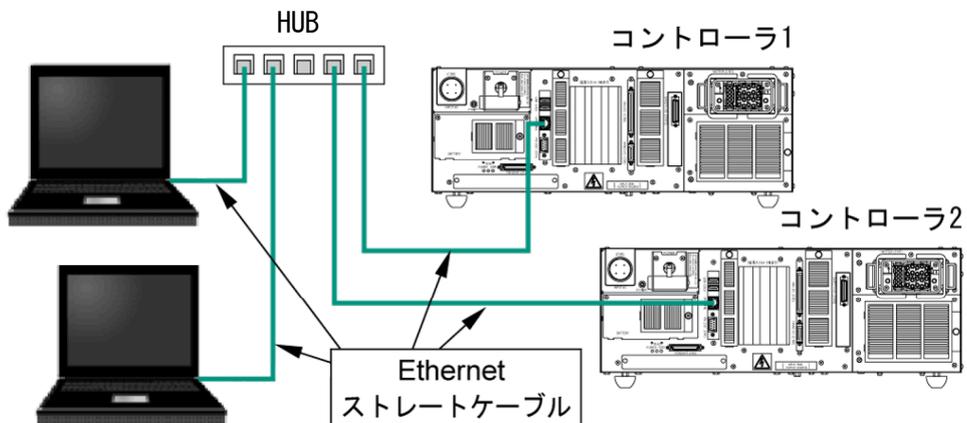


11.4.2 EtherNet 接続

● 1対1 接続の場合



● N対N 接続の場合



11.5 ロボットコントローラとの通信接続準備

11.5.1 RS-232C 接続する場合

11.5.1.1 RS232C（ロボットコントローラの設定）

RS-232C ポートを使って WINCAPSⅢと通信できるように、ロボットコントローラの設定をします。この設定は、ティーチングペンダントで行ないます。

ステップ 1

F6

ティーチングペンダントの基本画面で、[F6 設定] を押します。



[設定（メイン）] ウィンドウが表示されます。



ステップ 2

F4

[F5 通信設定.] を押します。

[通信設定メニュー] ウィンドウが表示されます。



ステップ3

F2

[F2 シリアル設定] を押します。
[RS232C 設定] ウィンドウが表示されます。

ステップ4

F5

COM2 を選択し、[F5 値変更.] を押します。



[通信速度 選択] ウィンドウが表示されます。

ステップ5

OK

通信速度を選択し、[OK] を押します。



[RS232C 設定] ウィンドウに表示が戻ります。

ステップ6

OK

表示内容を確認し、[OK] を押します。
通信速度が設定されます。



画面は [通信設定メニュー] ウィンドウに表示が戻ります。

ステップ7

F1

[通信設定メニュー] ウィンドウの [F1 通信権.] を押します。



[通信権 設定] ウィンドウが表示されます。



ステップ8

F5

COM2 を選択し、[F5 設定変更.] を押します。
[通信権 選択] ウィンドウが表示されます。

ステップ9

必要な通信権を選択します。



通信権の意味は以下のとおりです。

- ・使用不可：通信ポートを使用しません。
- ・読入のみ可：パソコン側で、ロボットコントローラ of データを読むことができます。パソコンからロボットコントローラにデータを送ることはできません。
- ・読入/書込可：パソコンとロボットコントローラの間で、双方向にデータのやり取りができます。

プログラムを作るときは、[読込／書込可] を選択します。
できあがったプログラムを自動運転し、変数やI/Oなどの値を監視するだけのときは、[読込のみ可] を選択します。

選択したら、[OK] を押します。

[通信権 設定] ウィンドウが表示に戻ります。

注意：RS232C と EtherNet を同時に読込／書込可にすることはできません。

ステップ 10

OK

表示内容を確認し、[OK] を押します。

通信権が設定されます。



画面は [通信設定メニュー] ウィンドウが表示に戻ります。



ステップ 11

Cancel

[Cancel] を 2 回押します。

基本画面に表示が戻ります。

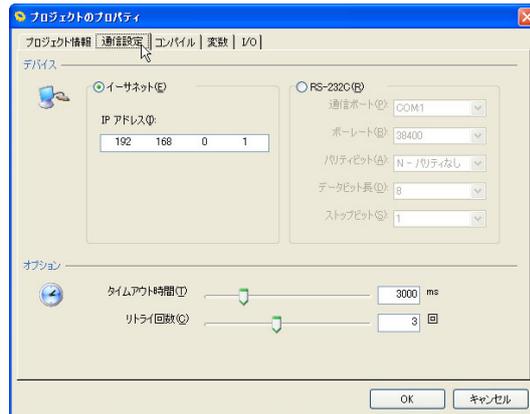
11.5.1.2 RS-232C 接続の設定 (WINCAPSⅢの設定)

RS-232C で WINCAPSⅢ とロボットコントローラが通信できるように、PC 側の設定をします。この設定は WINCAPSⅢ のプロジェクト内で行います。

ステップ 1

WINCAPSⅢのプロジェクトの通信設定を表示させます。

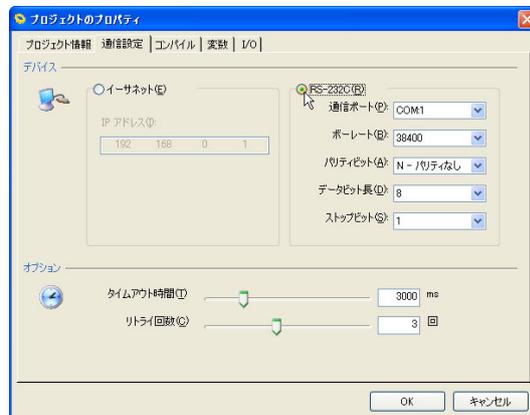
プロジェクトの通信設定はプロジェクト作成ウィザード時に行いますが、作成後に [プロジェクト (P)] - [プロパティ (P)] - [通信設定] タブを選択することでも表示、設定することができます。



ステップ 2

RS-232C を選択します。

RS-232C の詳細通信設定を入力します。



[通信ポート] は PC で使用する通信ポートを選択します。お使いの PC の仕様にあわせて選択してください。

[ボーレート]、[パリティビット]、[データビット長]、[ストップビット] はロボットコントローラで設定した値に合わせて設定してください。

ステップ 3

[タイムアウト時間] と [リトライ回数] を設定してください。

入力し終わったら [OK] を押してください。

PC 側の通信設定は完了です。

11.5.2 EtherNet 接続する場合

11.5.2.1 EtherNet (ロボットコントローラの設定)

イーサネットでPC(WINCAPSⅢ)とロボットコントローラが通信できるように、ロボットコントローラの設定をします。この設定は、ティーチングペンダントで行ないます。通信権の設定およびIPアドレス設定を行ないます。

ステップ1

F6

ティーチングペンダントの基本画面で、[F6 設定] を押します。



[設定 (メイン)] ウィンドウが表示されます。



ステップ2

F4

[F5 通信設定.] を押します。

[通信設定メニュー] ウィンドウが表示されます。



ステップ3

F1

[F1 通信権.] を押します。
[通信権 設定] ウィンドウが表示されます。



ステップ4

F5

EtherNetを選択し、[F5 設定変更.] を押します。
[通信権 選択] ウィンドウが表示されます。

ステップ5

[読込/書込可] を選択します。



通信権の意味は以下のとおりです。
Ethernetを使用するときは、[読込/書込可] を選択する必要があります。

- ・使用不可：通信ポートを使用しません。
- ・読込のみ可：パソコン側で、ロボットコントローラ of データを読むことができます。パソコンからロボットコントローラにデータを送ることはできません。
- ・読込/書込可：パソコンとロボットコントローラの間で、双方向にデータのやり取りができます。

選択したら、[OK] を押します。
[通信権 設定] ウィンドウに表示が戻ります。

注意：[COM2 (RS232C)] と [Ethernet] を同時に [読込/書込可] にすることはできません。[Ethernet] を [読込/書込可] にしたい場合は [COM2 (RS232C)] を [使用不可] または [読込のみ可] にしてから設定してください。

ステップ6

OK

表示内容を確認し、[OK] を押します。
通信権が設定されます。



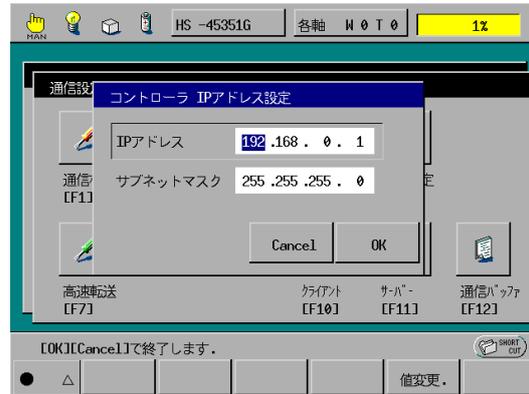
画面は [通信設定メニュー] ウィンドウに表示が戻ります。



ステップ7

F4

[F4 アドレス設定.] を押します。
[コントローラIPアドレス設定] ウィンドウが表示されます。



ステップ8

F5

[F5 値変更.] を押し、アドレスを設定します。



ステップ9

Cancel

[Cancel] を2回押します。
基本画面に表示が戻ります。

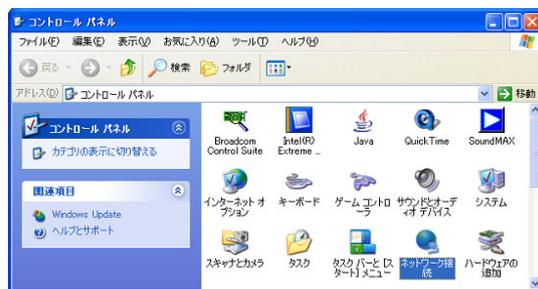
11.5.2.2 パソコンネットワーク環境設定

イーサネットによる接続のためには、Windows の設定を行なう必要があります。ここでは、ネットワークカード（アダプター）がインストールされインターネットプロトコル(TCP/IP)が有効になっていることを前提として、ネットワーク環境の設定手順を説明します。

まず、ローカルエリア接続が有効になっている事を確認します。次に、TCP/IP のプロパティに IP アドレスを設定します。

ステップ 1

Windows の [スタート] から、[コントロールパネル (C)] を順に選択します。
[コントロールパネル] ウィンドウが表示されます。



ステップ 2

[コントロールパネル] ウィンドウの [ネットワーク接続] をクリックします。
[ローカルエリア接続] のアイコンが表示されます。
このアイコンが「無効」と表示している場合には、アイコンにマウスを合わせ右クリックにて[有効にする(A)]を選択してください。



ステップ 3

[ローカルエリア接続] アイコンにマウスを合わせ右クリックにて[プロパティ] を選択すると [ローカルエリア接続のプロパティ] ダイアログが表示されます。



ステップ4

[全般] のタブを選択し、[インターネットプロトコル(TCP/IP)] を選択して、[プロパティ(R)] をクリックします。

ステップ5

[全般] のタブを選択して、[次のIPアドレスを使う(S)] をクリック、続いて、[IPアドレス(I)] と [サブネットマスク(U)] を入力します。

入力するIPアドレスとサブネットアドレスの具体的な値に関しては、そのネットワークを管理しているネットワーク管理者に問い合わせてください。

もし、そのネットワークがローカルなもの（たとえば、パソコンとロボットコントローラだけがつながるような環境）であれば、IPアドレスは自由に設定できますので、ここでは仮にIPアドレスを [192.168.0.1]、サブネットアドレスを [255.255.255.0] とします。

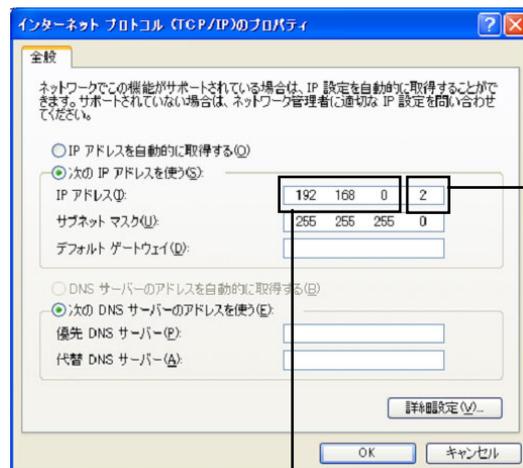
[OK] をクリックしてIPアドレスの設定は終わりです。

注意①：ワイドエリアネットワーク（たとえば社内のネットワーク）に接続する場合は、必ずネットワーク管理者に問い合わせ、IPアドレス、サブネットマスクを設定してください。

ローカルエリア内で使用していたIPアドレス設定を、そのまま無効にせず、ワイドエリアネットワーク（たとえば社内のネットワーク）に接続した場合は、接続ネットワークに混乱をもたらすおそれがあります。

②：IPアドレスは、同一ネットワーク内で重複できません。広く共有されているネットワークに接続する場合には、他の端末とIPアドレスが重複しないように注意してください。他の端末と、IPアドレスが重複する確率が小さい例として、次のアドレスが考えられます。

重複しにくいIPアドレス：[192.168.0.2] ~ [192.168.0.xxx]、(xxxは003~254)



ロボットコントローラに設定している値と同じです。

ロボットコントローラに設定している値と重複しない値にします。

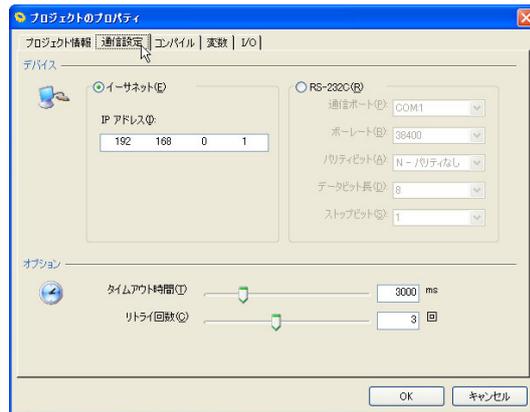
11.5.2.3 Ethernet 接続の設定 (WINCAPSⅢの設定)

Ethernet で PC (WINCAPSⅢ)とロボットコントローラとの接続するための、PC 側 (WINCAPSⅢ)の通信設定を行います。この設定は WINCAPSⅢのプロジェクト内で行います。

ステップ 1

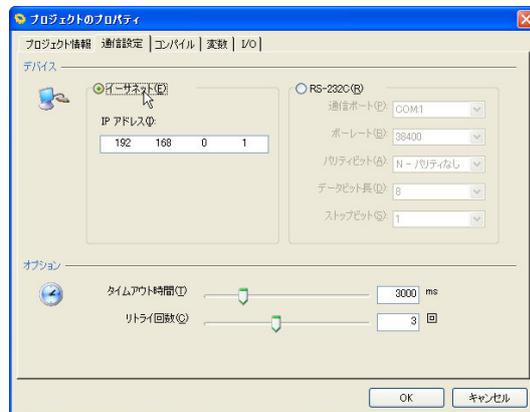
WINCAPSⅢのプロジェクトの通信設定を表示させます。

プロジェクトの通信設定はプロジェクト作成ウィザード時に行いますが、作成後に [プロジェクト (P)]-[プロパティ (P)]-[通信設定] タブを選択することでも表示、設定することができます。



ステップ 2

[イーサネット]が選択されていることを確認し、[IPアドレス]を入力します。



[IP アドレス]にはロボットコントローラの IP アドレスを入力します。

ステップ 3

[タイムアウト時間]と[リトライ回数]を設定してください。

入力し終わったら[OK]を押してください。

PC側の通信設定は完了です。

11.6 WINCAPSⅢによるデータ転送

データ転送（送信・受信）するうえで、ロボットコントローラの通信権の設定と、ロボットコントローラの状態を確認する必要があります。

ロボットコントローラの状態により、データ転送が失敗する場合がありますのでご注意ください。

11.6.1 コントローラ側の準備（データ転送時の注意）

①ティーチングペンダント画面に、エラーメッセージ表示がない事を確認してください。

②接続する通信ポート（RS232C or EtherNet）の通信権が「読込/書込可」になっている事を確認して下さい。

注意：「読込のみ可」を選択している場合、データ送信するとエラー「200B-コンフィグレーション転送失敗」が発生します。

③ロボットコントローラのモータ ON/OFF 状態と動作モード状態により、データ送信ができない場合があります。

状態別のデータ送信可否を下表に示します。

データ送信可否対応表

状態 送信データ	モータ	コントローラ 動作モード				備考
		外部自動	内部自動	手動	ティーチ	
PAC プログラム	ON	×	×	×	×	△：プログラム自動保存されません
	OFF	×	△	○	△	
変数	ON	×	○	○	○	
	OFF	×	○	○	○	
I/O	ON	×	×	×	×	
	OFF	×	○	○	○	
アーム	ON	×	△	△	△	△：ツール、ワーク、エリアデータのみ送信可
	OFF	×	○	○	○	

注意：

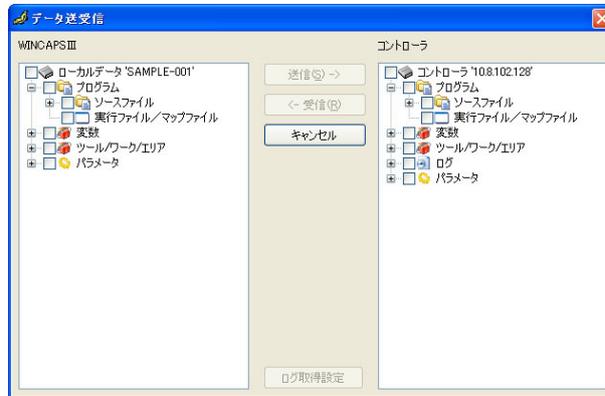
- ・ データ受信については、モータのON/OFF状態、コントローラ 動作モードに関係なくで受信可能です。
- ・ プログラム運転中のデータ受信は、プログラム処理に遅れが発生しますので、ご注意ください。

④ティーチングペンダントの画面に[プログラム一覧]もしくは[変数タイプ選択]が表示されていないことを確認してください。

11.6.2 プログラムデータをロボットコントローラに送信する

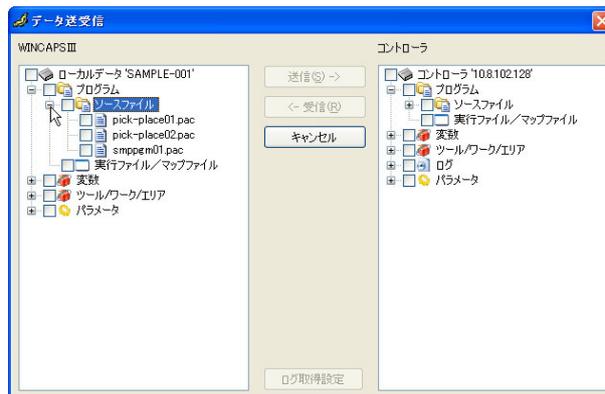
ステップ 1

WINCAPSⅢで[通信 (N)]-[データ送受信 (T)]を選択します。
[データ送受信]ウインドウが表示されます。



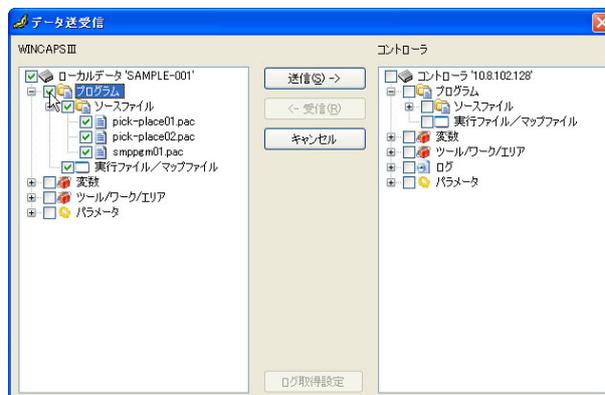
ステップ 2

[WINCAPSⅢ]側の[プログラム]-[ソースファイル]を展開すると保存されているプログラムが表示されます。



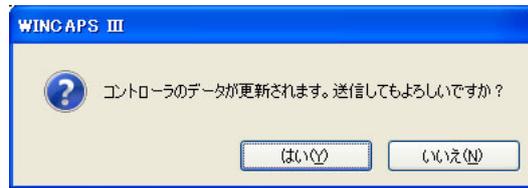
ステップ 3

[プログラム]にチェックを入れ、[送信(S)->]ボタンを押します。



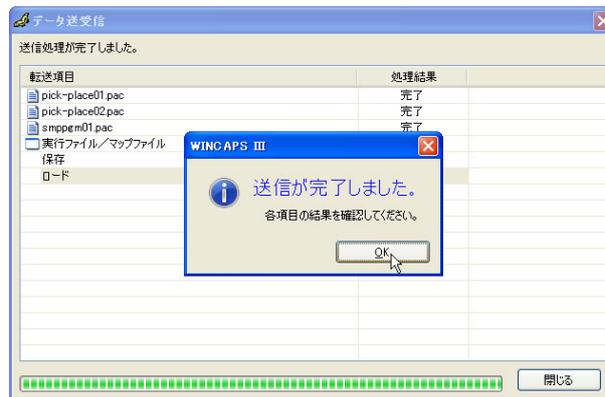
ステップ4

確認のダイアログが表示され、[はい]を選択すると、データがロボットコントローラに送信されます。



ステップ5

送信されたデータの処理結果が全て[完了]であることを確認してください。



ステップ6

ティーチングペンダントで[F1 プログラム]を押し、[プログラム一覧]内に送信したプログラムがあることを確認します。

これでロボットコントローラに作成したプログラムが送信されました。

第4部

プログラムの検証

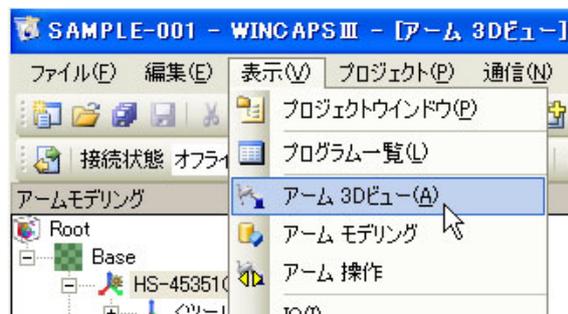
第 12 章 プログラム起動

12.1 WincapsⅢによるシミュレーション

作成したロボットプログラムの動作を、実際のロボットを動かさずにパソコンのシミュレーション映像を見ながら確認することができます。

12.1.1 アーム 3D ビューを開く

ロボットのシミュレーション画像を表示するためにアーム 3D ビューを開きます。
[表示 (V)]-[アーム 3D ビュー (A)]を選択すると[アーム 3D ビュー]ウインドウが表示されます。



12.1.2 WincapsⅢをモニタモードにする

WincapsⅢの接続状態を[オンライン(モニタ)]にします。

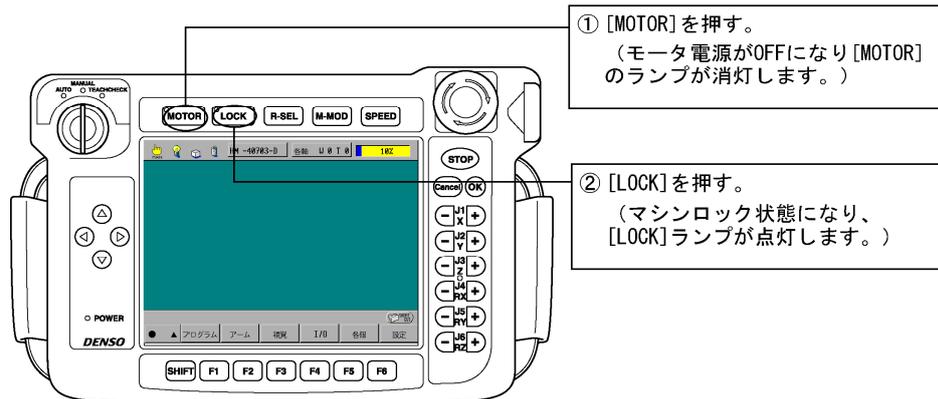


12.1.3 マシンロック状態にする

ロボットコントローラをマシンロック状態にします。これによって、ロボットを実際には動作させずにプログラムを実行してシミュレーションができます。

ステップ1 モータ電源を切る

ステップ2 マシンロックにする



★注意★

マシンロックするときは、モータ電源が切られた状態にしてください。つまり、[MOTOR]ランプが消灯していることを確認してください。

★参考★

[Ver1.4以降]

マシンロックをした場合I/Oの出力を制限することができます。詳細については、操作ガイド「5.5 I/O信号の表示とロボット動作のシミュレート」を参照してください。

I/O出力制限の状態によりステータスバーのアイコンが変化します。



: I/O出力制限なし



: I/O出力制限あり

12.1.4 プログラム起動を行う

プログラム起動させるには、ティーチチェックモードまたは自動モードのいずれかの方法で行ってください。

起動方法は「12.2 ティーチチェックモードによるプログラム起動」または「12.3 内部自動モードによるプログラム起動」のいずれかの起動方法に従い、マシンロック状態にてプログラム起動を行って下さい。

WINCAPSⅢの[アーム3Dビュー]ウインドウに表示されているロボットアームがプログラムに応じて動作します。

12.2 ティーチチェックモードによるプログラム起動

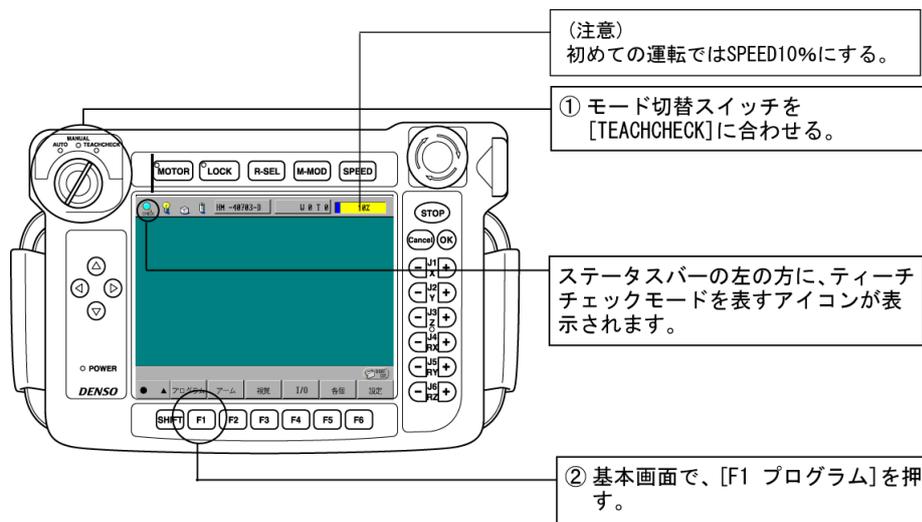
ティーチングした結果のプログラムの動作を手動で検査することをティーチチェックといいます。ティーチチェックは、ティーチチェックモードで行ないます。

12.2.1 ティーチチェックモードにする

ステップ1 モータ電源を入れる

マシンロック状態で動作させる場合にはモータ電源を入れしないでください。

ステップ2



★参考★

ERROR 21F2(自動イネーブルが入っています)が発生する場合は、「7.3.2 動作モードと自動イネーブル入力信号の関連性」をご確認ください。



[プログラム一覧] ウィンドウが表示されます。

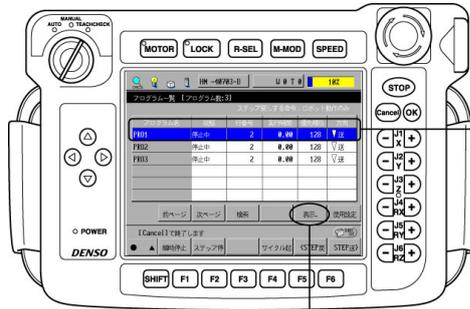
プログラム名	状態	行番号	実行時間	優先順位	方向
PR01	停止中	2	0.00	128	▼送
PR02	停止中	2	0.00	128	▼送
PR03	停止中	2	0.00	128	▼送

前ページ 次ページ 検索 表示 使用設定

[Cancel]で終了します

● ▲ 即時停止 ステップ停 サイクル起 <STEP戻 STEP送>

12.2.2 プログラムを選択



① [プログラム一覧]ウインドウで「PRO1」を選択する。
(カーソルキーまたはジョグダイヤルを操作するか画面にタッチすると選択できる。)

② [表示]を押して、PRO1のプログラムコードを表示させる。



[プログラム]ウインドウのPRO1のプログラムコードが表示されます。

12.2.3 ステップ送りチェック起動

1ステップ毎にプログラム起動を行います。



③ [F6 STEP送)]を押す。
(ステップ送りは右カーソルでも可能。)

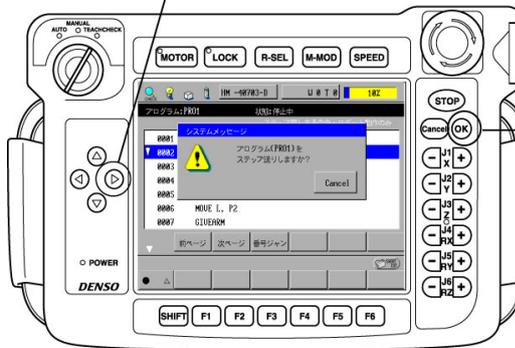


システムメッセージが表示されます。

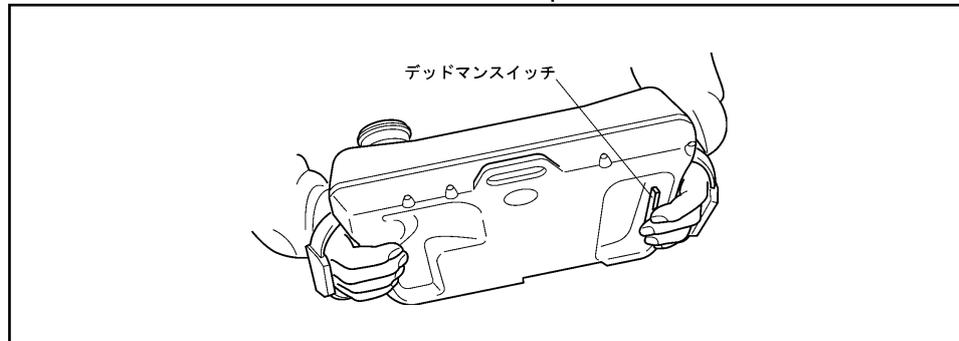
★注意★

プログラム確認時はいつでも[ロボット停止]を押せるようにして運転してください。

右カーソル



④ デッドマンスイッチを押しながら、[OK]を押す。
(ステップ送りを中止するには、[Cancel]を押す。)



ティーチチェックモードでは、デッドマンスイッチと[OK]を同時に押している間のみプログラムが実行されます。いずれかのボタンを離すとロボットは瞬時停止します。



上記の手順で繰り返し操作し、動作の安全を確認しながら、プログラムを最後まで実行しましょう。

12.2.4 サイクルチェック起動

次に、ステップ送りチェックで確認したプログラムをサイクル動作で確認しましょう。



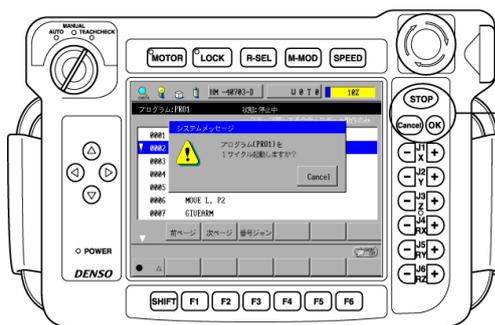
① [F4 サイクル起]を押します。



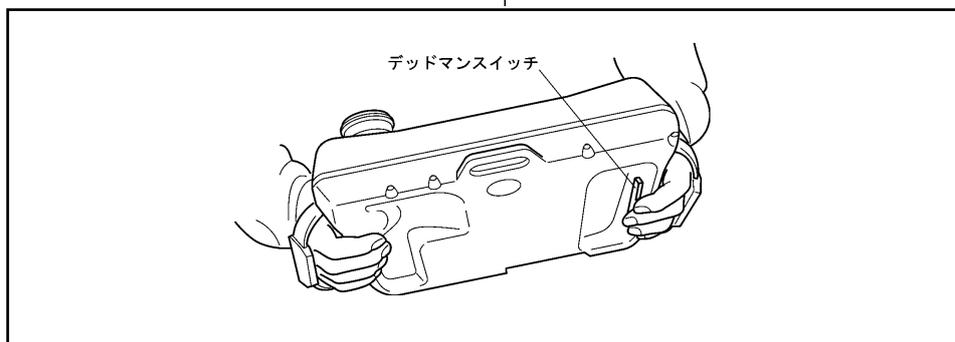
システムメッセージが表示されます。

★注意★

プログラム確認時はいつでも[STOP]を押せるようにして運転してください。



② デッドマンスイッチを押しながら、[OK]を押すと動きます。
サイクルチェックを中止するには
[Cancel]か[STOP]を押すか、
デッドマンスイッチを放します。



ティーチチェックモードでは、デッドマンスイッチと[OK]を同時に押している間のみプログラムが実行されます。いずれかのボタンを放すとロボットは瞬時停止します。



プログラムがサイクル起動され、ロボットが動くと同時に、プログラムリストの反転表示が順に進みます。

プログラムエンドまで実行すると止まります。

12.3 内部自動モードによるプログラム起動

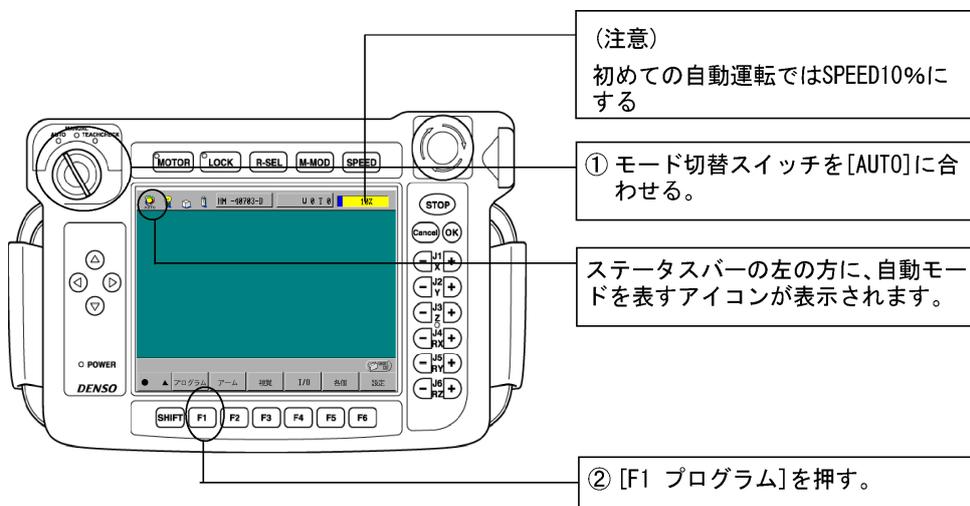
ティーチチェックの終わったプログラムを自動運転で動かしてみましょう。
注意：初めて自動モードで運転するプログラムは、速度設定を10%程度にして、運転を始めてください。手動モードやティーチチェックモードでは、自動モードの10%の速度で動くようになっていますが、自動運転では最高速度で動かすことができます。

12.3.1 自動モードにする

ステップ1 モータ電源を入れる

マシンロック状態で動作させる場合にはモータ電源を入れないでください。

ステップ2

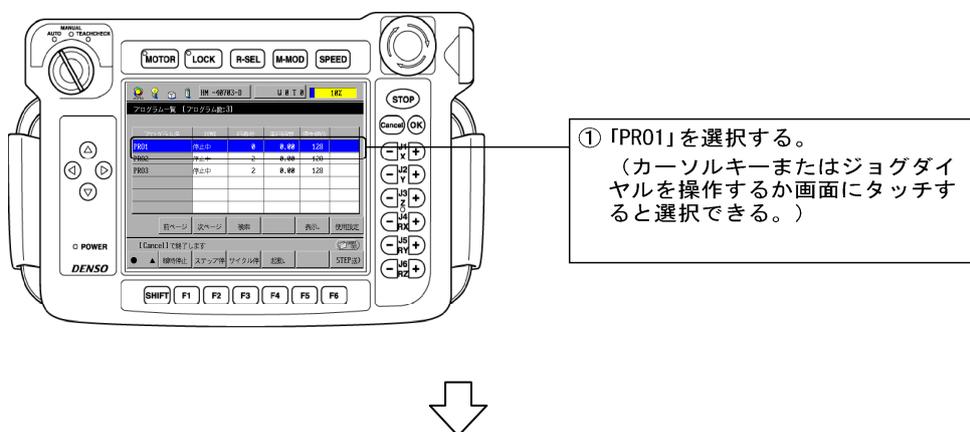


★参考★

ERROR 21F3(自動イネーブルが切れています)が発生する場合は、「7.3.2 動作モードと自動イネーブル入力信号の関連性」をご確認ください。

12.3.2 プログラムを選択

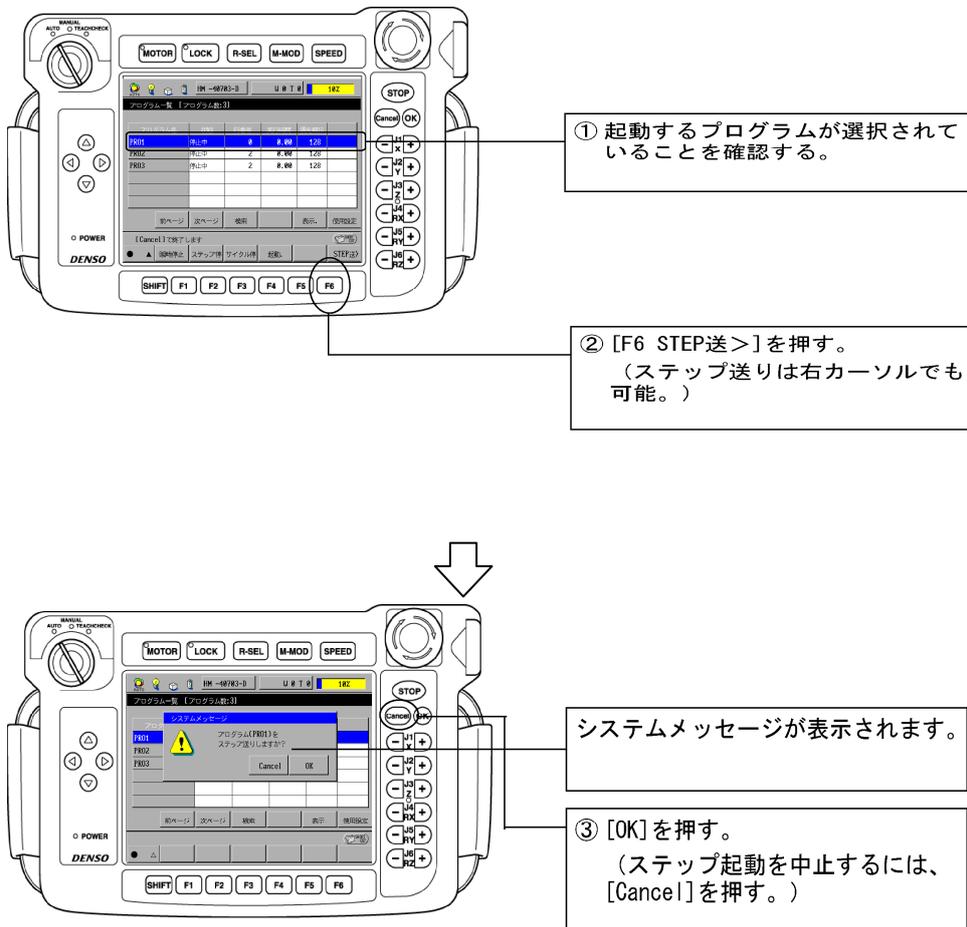
[プログラム一覧]ウィンドウで、自動運転させるプログラムを選択しましょう。



12.3.3 ステップ起動

★注意★

プログラム表示をしながらステップ起動したい場合は、ステップ動作前に [F11 表示] を押してください。



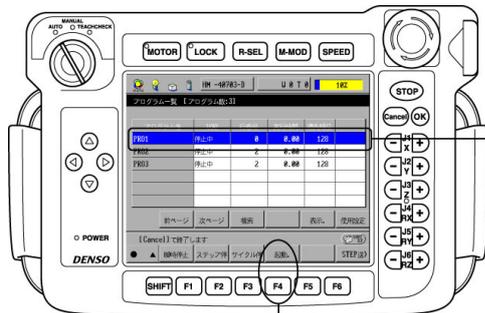
★注意★

プログラム確認時はいつでも [STOP] を押せるようにして運転してください。

PR01のプログラムが自動モードでステップ起動します。
上記の操作を繰り返し、動作を確認しながらプログラムを最後まで実行しましょう。

12.3.4 サイクル起動

ステップ起動を確認したら、サイクル起動させましょう。



① 起動するプログラムが選択されていることを確認する。

② [F4 起動]を押す。

★注意★

プログラム確認時はいつでも[STOP]を押せるようにして運転してください。



③ [1サイクル起動]を選択し、[OK]を押します。
PRO1のプログラムが動作します。

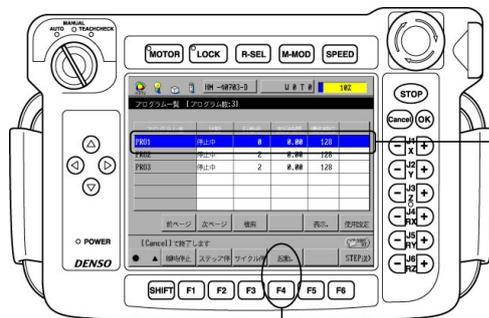
プログラムエンドまで実行すると止まります。

★注意★

表示される実行時間はプログラム開始から終了時間までの間であり、ステップ停止・瞬時停止による一時停止時間も含まれています。

12.3.5 連続起動

プログラムを連続起動させましょう。



① 起動するプログラムが選択されていることを確認する。

② [F4 起動] を押す。



「1サイクル起動」と「連続起動」の選択画面が表示されます。

③ 「連続起動」を選択します。

④ [OK] を押します。
PR01プログラムが連続動作します。
(連続起動を中止するには、[瞬時停止]、[ステップ停]、または[STOP]を押す。)

★注意★

プログラム確認時はいつでも[STOP]を押せるようにして運転してください。



以上で、ペンダントを使ってロボットを動かす作業が完了です。

12.4 ロボットの停止操作

ロボット停止操作には、次に紹介する4つの操作があります。ここでは、それぞれの意味と操作方法を紹介します。



12.4.1 サイクル停止 [F3]

サイクル停止を実行すると、プログラムのEND命令で停止します。ロボットを連続起動で動かした場合に使用します。ただしモータパワーは切れません。

12.4.2 ステップ停止 [F2]

ステップ停止を実行すると、ボタンが押された時に実行していたステップが完了すると停止します。ただしモータパワーは切れません。

12.4.3 瞬間停止 [F1], [STOP]

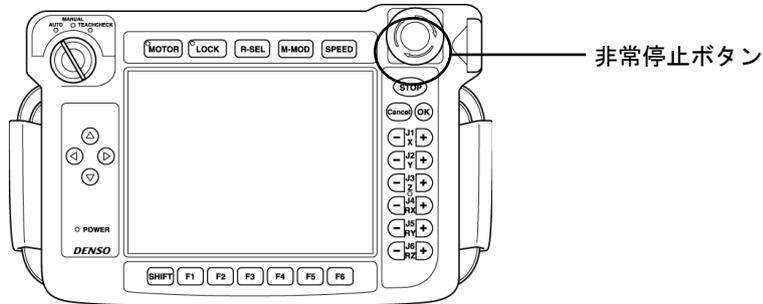
瞬間停止を実行すると、直ちに停止します。ただしモータパワーは切れません。

12.4.4 非常停止（ロボット停止）

非常停止を押すとロボットは非常停止します。この場合モータパワーは切れます。ロボットを緊急停止させたい時に使用します。

ステップ 1

「ロボット停止」させる



非常停止ボタン

The screenshot shows the LCD screen displaying a program list. The status of the selected program (PRO1) is highlighted in blue and circled in red. The status is '停止中' (Stopped).

プログラム名	状態	行番号	実行時間	優先順位	方向
PRO2	停止中	2	0.00	128	▽送
PRO3	停止中	2	0.00	128	▽送
PRO1	停止中	2	0.00	128	▽送

プログラムが非常停止し、[状態]欄の表示が「停止中」になります。

ロボット停止後そのまま再起動操作を行うと、選択されたプログラムの先頭から実行されます。

再起動は、モータをONし、「ステップ起動」「サイクル起動」「連続起動」のいずれかによります。

第 13 章 外部機器からのロボット起動方法

13.1 I/O 割付モードの確認方法

現在設定されている割付モードはティーチングペンダントの I/O にて確認できます。I/O 割付モードにより、外部機器（PLC 等）からの起動方法が変わりますので設定状態は事前に確認しておく必要があります。

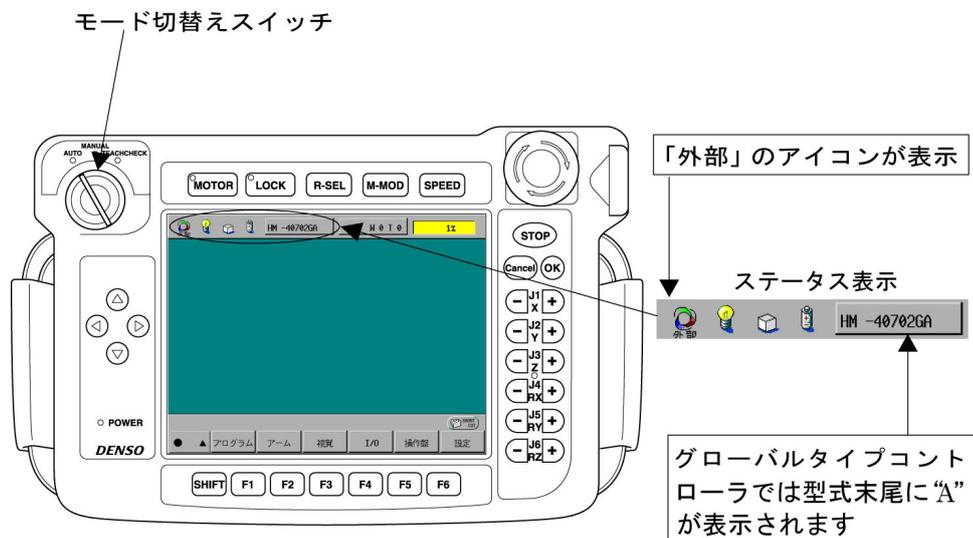
操作経路：[F4 I/O]

ここに、現在設定されている I/O 割付モードが表示されます。



13.2 グローバルタイプコントローラ使用時の注意

外部機器（PLC 等）から起動をおこなうには、単一位置制御機能を「外部自動限定モード」に設定する必要があります。モード切り替えスイッチを AUTO に切り替えるだけでステータス表示が「外部」になる事を確認して下さい。設定変更が必要な場合は、「2.2.2 単一位置制御機能 (Single Point of control) について」を参照してください。



13.3 Mini I/O 専用モードによる起動方法

Mini I/O モードでは、プログラム起動などの命令を「コマンド領域 (3Bit)」と「データ領域 (3Bit)」のビット組合せ (I/O コマンド) で指示します。尚、I/O コマンドはストローブ信号を使いコマンド実行を開始させます。

13.3.1 Mini I/O コマンド専用入力信号の種類と機能

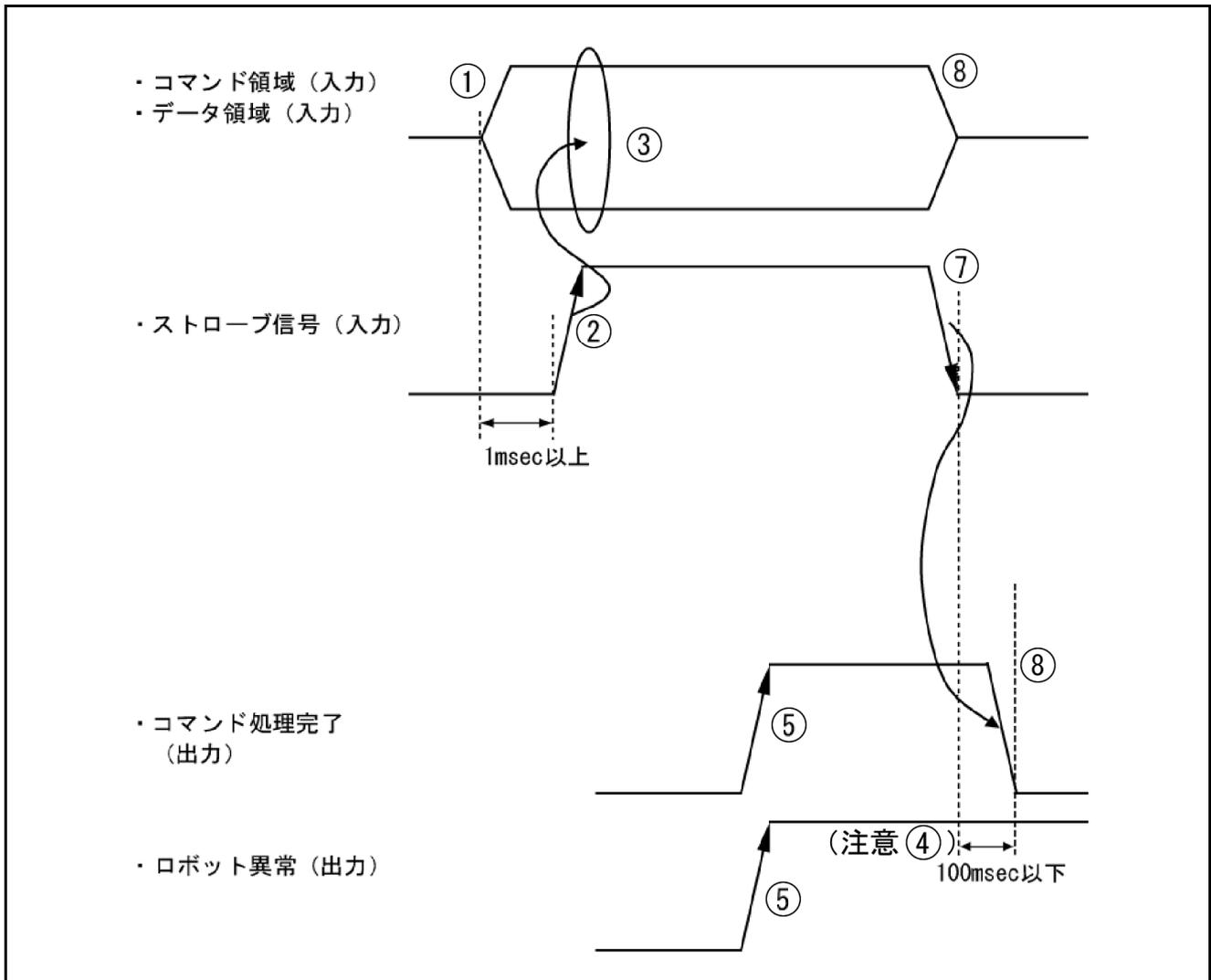
Mini I/O モードでの専用入力信号には下表に示すものがあります。

用途		入力信号名		機能	
		コマンド領域 (3bit)	データ領域 (3Bit)		
立ち上げ	モータ ON CAL 実行	000	001	+ストローブ 信号	モータ電源を入れる
	外部速度(SP)100	000	010		外部スピードを 100%に設定する
	外部モード切り替え	000	100		外部モードにする
	モータ ON CAL 実行 外部速度(SP)100 外部モード切り替え	000	111		上記 (立ち上げ) を全て行う
プログラム実行		010	プログラム番号 (0~7)	+ストローブ 信号	指定プログラムを実行する
コンティニュー スタート		011	---		コンティニュー スタートを実行する
指定プログラムリセット		100	プログラム番号 (0~7)		指定プログラムを瞬時停止しプログラムリセット(初期化)する
全プログラムリセット		101	---		全プログラムを瞬時停止しプログラムリセット(初期化)する
ロボット異常クリア		001	---		ERROR を解除する
停止		---	---	ステップ 停止 (全タスク)	信号開放(OFF)で、実行中の全プログラムをステップ停止する

注意：「+ストローブ信号」はコマンド領域(3bit)とデータ領域(3bit)を組合せて使用することを意味しています。

13.3.2 Mini I/O コマンド処理方法

I/O コマンドを実行するには、下図に示すような処理を行ないます。



I/Oコマンド処理方法概要

①外部からロボットコントローラのコマンド実行入出力信号に対して、コマンド領域、データ領域（必要な場合のみ）をセットします。

②セット完了後、ストローブ信号を OFF→ON に立ち上げます。

注意 ① ①でセットするデータは、ストローブ信号を立ち上げる1msec以上前に確定しておく必要があります。

② ストローブ信号によるコマンド入力は専用出力「ロボット初期化完了」が出力されるのを待ってから行ってください。ただし、ロボット異常クリアコマンドを実行する場合は、ロボット初期化完了が出力されないため、ロボット異常が出力されていたらロボット異常クリアコマンドを実行するようにしてください。

- ③コントローラは、ストローブ信号入力でコマンド領域、データ領域を読み込みます。
- ④コントローラは、読み込んだコマンドに基づき、処理を行いません。
- ⑤コマンド処理が完了した後、コントローラはコマンド処理完了信号を OFF→ON に立ち上げます。
処理中にエラーが発生した場合には、コマンド処理完了信号と同時にロボット異常も出力されます。
- ⑥PLC は、コマンド処理完了信号が入力するのを待って、ロボット異常の発生有無を確認してください。
- ⑦PLC は、コマンド・データ領域およびストローブ信号を OFF 状態にします。
- ⑧コントローラは、ストローブ信号の ON→OFF の立ち下がり、コマンド処理完了出力を、OFF 状態にします。
コマンド処理エラーに伴い出力されるロボット異常は、ロボット異常クリアコマンドが実行されるまで ON 状態を保持します。

注意 ③ ⑧でストローブ信号をON→OFFに立ち下げた後、コマンド処理完了信号が、OFFになるまでの最長時間は100msecです。
④ ⑤のコマンド処理完了信号がOFF→ONに立ち上る前にストローブ信号がOFFになっていた場合、コマンド処理完了信号はいったん出力された後、100msec以内にOFF状態になります。

13.3.3 Mini I/O 専用出力信号の種類と機能

Mini I/O 専用モードでの専用出力信号には、下表に示すものがあります。

Mini I/O専用モードでの専用出力信号の種類と機能

用途	信号名	機能
立ち上げ	ロボット初期化完了(出力)	「運転準備コマンド」が実行可能な状態を外部へ出力します。
	自動モード(出力)	自動モード状態を出力します。
	運転準備完了(出力)	モータ電源が入り、かつ、外部自動モード状態を出力します。
プログラム実行	ロボット運転中(出力)	運転中(プログラム実行中のタスクが一つ以上)状態を出力します。
エラー・警告	CPU 正常(出力)	ロボットコントローラのCPUが正常であるときに出力します。
	ロボット異常(出力)	サーボ異常、プログラム異常などロボットに異常が発生したときに出力します。
	バッテリー切れ警告(出力)	エンコーダバックアップ電池またはメモリバックアップ電池の電圧が低下したときに出力します。
コンティニュー機能	コンティニュースタート許可(出力) 注：I/Oハード設定により設定が必要	コンティニュースタートが実行できるときに出力します。
非常停止回路 (標準タイプ コントローラ)	非常停止出力 (2系統)	非常停止状態を出力します。
	ペンダント非常停止出力 (2系統)	ティーチングペンダント、ミニペンダントの非常停止ボタンの状態を出力します。
	デッドマン SW 出力 (イネーブル SW 出力) (2系統)	ティーチングペンダント、ミニペンダントのデッドマンスイッチ(イネーブルスイッチ)の状態を出力します。
安全回路 (グローバルタイプ コントローラ)	ペンダント非常停止出力 (2系統)	ティーチングペンダント、ミニペンダントの非常停止ボタンの状態を出力します。
	デッドマン SW 出力 (イネーブル SW 出力) (2系統)	ティーチングペンダント、ミニペンダントのデッドマンスイッチ(イネーブルスイッチ)の状態を出力します。
	コンタクタ接点モニタ出力	コントローラ内モータコンタクタの補助接点の状態を出力します。モータON時にON、モータOFF時にOFFする接点出力です。

13.4 標準モードによる起動方法

標準モードでは、プログラム起動などの命令を「コマンド領域 (4Bit)」と「データ領域 1 (8Bit)」と「データ領域 2 (16Bit)」のビット組合せ (I/O コマンド) で指示します。

尚、I/O コマンドはストロブ信号を使いコマンド実行を開始させます。

13.4.1 標準 I/O コマンド専用入力信号の種類と機能

標準モードでの専用入力信号には下表に示すものがあります。

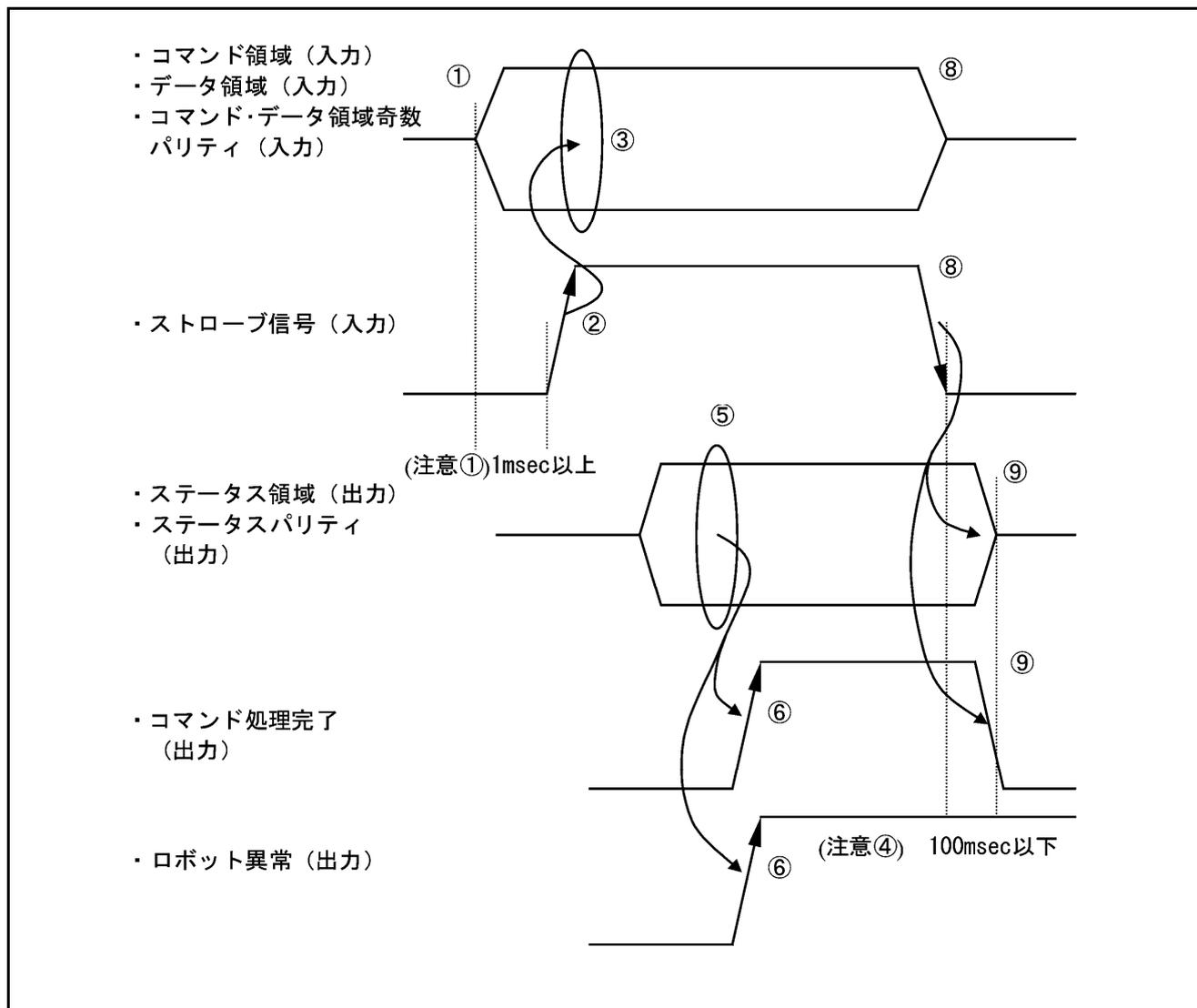
用途		入力信号名			機能
		コマンド領域(4bit)	データ領域 1(8Bit)	データ領域 2(16Bit)	
プログラム実行	プログラムリセット 起動	0001	00000001	プログラム番号 (0~32767)	指定プログラムをリセットした後起動する
	プログラム起動	0001	00000010	プログラム番号 (0~32767)	指定プログラムを起動する。 ステップ停止中・瞬時停止中の場合、停止した次のステップから起動再開
	コンティニュースタート	0001	00000100	---	コンティニュー停止で停止している全てのプログラムを再起動する
	ステップ停止	0001	00010000	プログラム番号 (0~32767)	指定プログラムをステップ停止する
	瞬時停止	0001	00100000	プログラム番号 (0~32767)	指定プログラムを瞬時停止する
	リセット	0001	01000000	プログラム番号 (0~32767)	指定プログラムを瞬時停止しプログラムリセット(初期化)する
外部速度・ 加速度設定	速度設定	0010	00000001	速度設定値 (1~100)	指定した速度設定値(1~100)に変更
	加速度設定	0010	00000010	加速度設定値 (1~100)	+奇数パリティ (必要時) 指定した加速度設定値(1~100)に変更
	減速度設定	0010	00000100	減速度設定値 (1~100)	+ストロブ信号 指定した減速度設定値(1~100)に変更
エラー読出し		0100	---	---	現在発生しているエラー番号をステータス領域に出力
I型変数書込み		0101	変数番号 (0~255)	変数設定値 (-32768~32768)	指定した I 型変数番号(0~255)に変数設定値(-32768~32768)へ変更
I型変数読出し		0110	変数番号 (0~255)	---	指定した I 型変数番号(0~255)の現在値をステータス領域に出力
立ち上げ	モータ ON CAL 実行	0111	00000001	---	モータ電源を入れる
	外部速度(SP)100	0111	00000010	---	外部スピードを 100%に設定する
	外部モード切り替え	0111	10000000	---	外部モードにする
	モータ ON CAL 実行 外部速度(SP)100 外部モード切り替え	0111	10000011	---	上記(立ち上げ)を全て行う
ロボット エラークリア		1000	---	---	ERROR を解除する
I/O 書込み		1001	00000000~ 11111111	内部 IO 番号 (128~504)	データ領域 2 で指定した番号から始まる内部 IO 領域にデータ領域 1 の状態を代入
I/O 読込み		1010	---	内部 IO 番号 (128~504)	データ領域 2 で指定した番号から始まる内部 IO 状態をステータス領域下位 8 ビットに出力
停止	---	---	---	ロボット停止	信号開放(OFF)で、ロボット停止する
	---	---	---	瞬時停止 (全タスク)	信号開放(OFF)で、実行中の全プログラムを瞬時停止する
	---	---	---	ステップ停止 (全タスク)	信号開放(OFF)で、実行中の全プログラムをステップ停止する

注意：「+奇数パリティ」はコマンド領域・データ領域1・データ領域2の合計ビット数が偶数の場合、奇数パリティ信号を入力し合計ビット数を奇数にします。

注意：「+ストローブ信号」はコマンド領域・データ領域1・データ領域2・奇数パリティを組合せて使用することを意味しています。

13.4.2 標準 I/O コマンド処理方法

I/O コマンドを実行するには、下図に示すような処理を行ないます。



I/Oコマンド処理方法概要 (標準モード)

①外部からロボットコントローラのコマンド実行入出力信号に対して、コマンド領域、データ領域 (必要な場合のみ)、コマンド・データ領域奇数パリティをセットします。

②セット完了後、ストローブ信号を OFF→ON に立ち上げます。

注意 ① ①でセットするデータは、ストローブ信号を立ち上げる1msec以上前に確定しておく必要があります。

② ストローブ信号によるコマンド入力は専用出力「ロボット初期化完了」が出力されるのを待ってから行なってください。

- ③コントローラは、ストロブ信号入力でコマンド領域、データ領域、コマンド・データ領域奇数パリティを読み込みます。
- ④コントローラは、読み込んだコマンドに基づき、処理を行いません。
- ⑤コマンドがステータス出力するものであった場合には、コントローラはステータス領域、ステータスパリティをセットします。
- ⑥コマンド処理が完了し、ステータス領域を設定した後、コントローラはコマンド処理完了信号を OFF→ON に立ち上げます。
処理中にエラーが発生した場合には、コマンド処理完了信号と同時にロボット異常も出力されます。
- ⑦PLC は、コマンド処理完了信号が入力するのを待って、必要ならばステータス領域の状態を取得します。この際、ロボット異常が発生していないかの確認も行なってください。
- ⑧PLC はステータス読み込み完了後、コマンド・データ領域およびストロブ信号を OFF 状態にします。
- ⑨コントローラは、ストロブ信号の ON→OFF の立ち下がり、ステータス領域とコマンド処理完了出力を、OFF 状態にします。
コマンド処理エラーに伴い出力されるロボット異常は、ロボット異常クリアコマンドが実行されるまで ON 状態を保持します。

注意 ③ ⑧でストロブ信号をON→OFFに立ち下げた後、ステータス領域とコマンド処理完了信号が、OFFになるまでの最長時間は100msecです。
④ ⑥のコマンド処理完了信号がOFF→ONに立ち上る前にストロブ信号がOFFになっていた場合、コマンド処理完了信号とステータス領域はいったん出力された後、100msec以内にOFF状態になります。

13.4.3 標準 I/O 専用出力信号の種類と機能

標準モードでの専用出力信号には、下表に示すものがあります。

標準モードでの専用出力信号の種類と機能

用 途	信 号 名	機 能
立ち上げ	ロボット初期化完了	運転準備スタート可能な状態のときに出力する
	自動モード	ロボットが自動モードになっているときに出力する
	外部モード	ロボットが外部モードになっているときに出力する
	サーボ ON 中	モータ電源入りになっているときに出力する
プログラム実行	ロボット運転中	ロボットが運転中（プログラム実行中）であるときに出力する
エラー・警告	CPU 正常	ロボットコントローラのCPUのハードウェアが正常であるときに出力する
	ロボット異常	サーボ異常、プログラム異常などの重大な異常が発生したときに出力する
	ロボット警告	軽微な異常が発生したときに出力する
	バッテリー切れ警告	エンコーダバックアップ電池またはメモリバックアップ電池の電圧が低下したときに出力する
コンティニュー機能	コンティニュースタート許可	コンティニュースタートが実行できるときに出力する
SS 機能	SS モード	SSモードのあいだ、出力する 操作ガイド「3.4.6 SS（セーフティースタート）機能」参照
非常停止	非常停止	非常停止回路構成のための専用接点出力
I/O コマンド処理	コマンド処理完了	I/Oコマンド処理の完了を出力する
	ステータス領域 奇数パリティ	ステータス領域（16bit）の出力ビットの合計が偶数のとき、奇数パリティを出力します
	ステータス領域 （16bit）	エラー読出し、I型変数読出し、I/O読込み処理の結果を出力します

13.5 互換モードによる起動方法

互換モードでは、プログラム起動などの機能がビット毎に意味づけられており、そのビットを立て指示します。

13.5.1 互換モード専用入力信号の種類と機能

互換モードでの専用入力信号には下表に示すものがあります。

互換モードでの専用入力信号の種類と機能

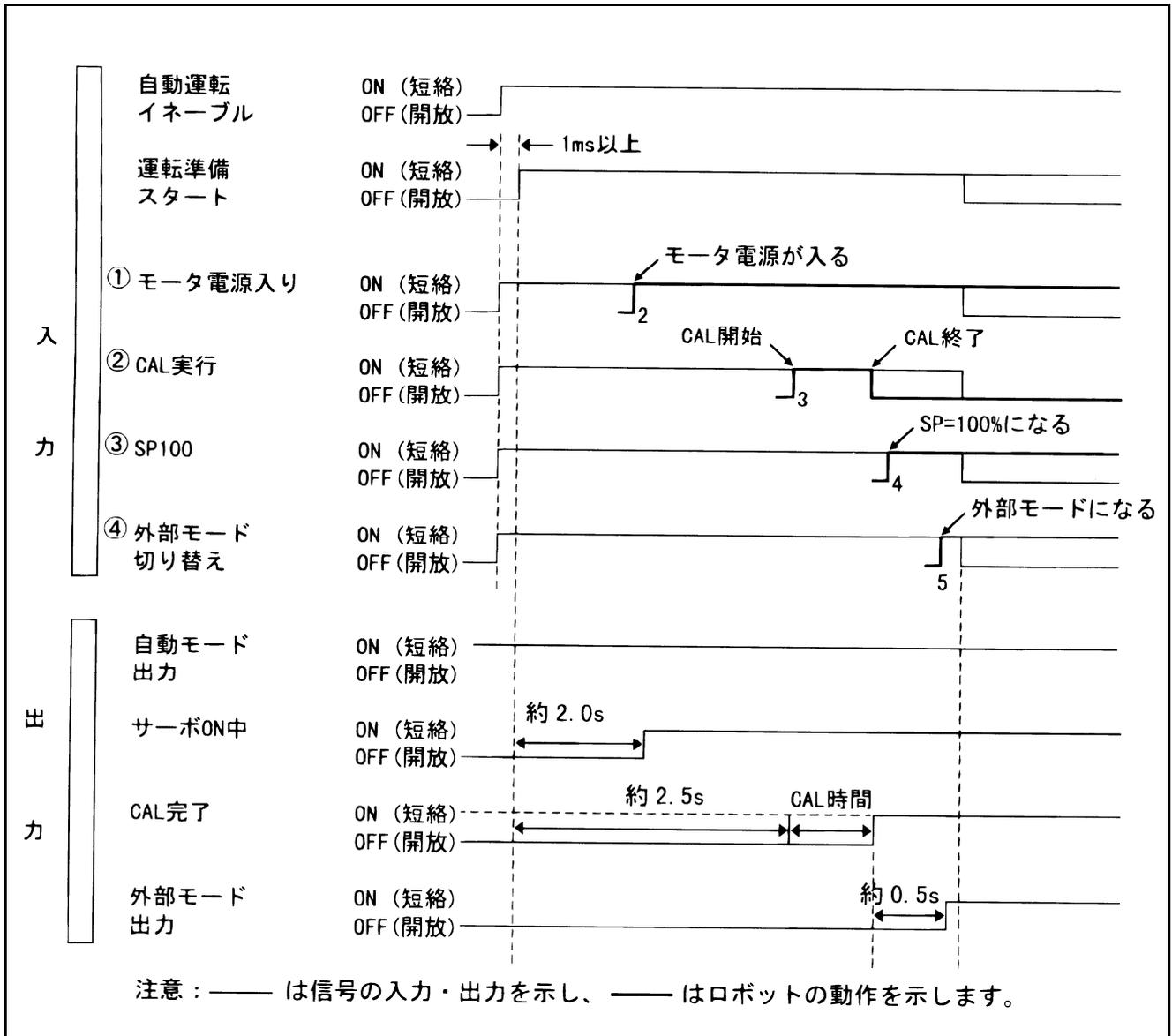
用 途	信 号 名	機 能
立ち上げ	自動運転イネーブル	自動モードに切り替えできるようになる。
	モータ電源入り +運転準備スタート	モータ電源を入れる。
	CAL 実行 +運転準備スタート	CALを実行する。
	SPI00 +運転準備スタート	スピードを100%に設定する。
	外部モード切り替え +運転準備スタート	外部モードにする。
	プログラムリセット +運転準備スタート	停止中の全プログラムを初期化します。 初期化後、プログラム起動した場合には、プログラムの先頭から実行します。
	プログラム No. 選択 +プログラムスタート	指定プログラムを実行する。
プログラム実行	プログラムリセット +プログラム No. 選択 +プログラムスタート	現プログラムをキャンセルし、指定プログラムを先頭から実行する。
	ロボット停止	信号開放でロボット停止する。
停止	ロボット停止	信号開放でロボット停止する。
	ステップ停止	信号開放で全プログラムをステップ停止する。
	瞬時停止	信号開放で全プログラムを瞬時停止する。
エラー解除	ロボット異常クリア +運転準備スタート	ERRORを解除する。
プログラム割り込み	割り込みスキップ	現ステップの実行を中止し、次のステップの実行を開始する。
コンティニュースタート	コンティニュースタート信号 +プログラムスタート	コンティニュースタートを実行する。

注意：信号名欄に複数の信号名が記述してあるものは、組み合わせて使用することを意味しています。

13.5.2 互換モードの処理方法

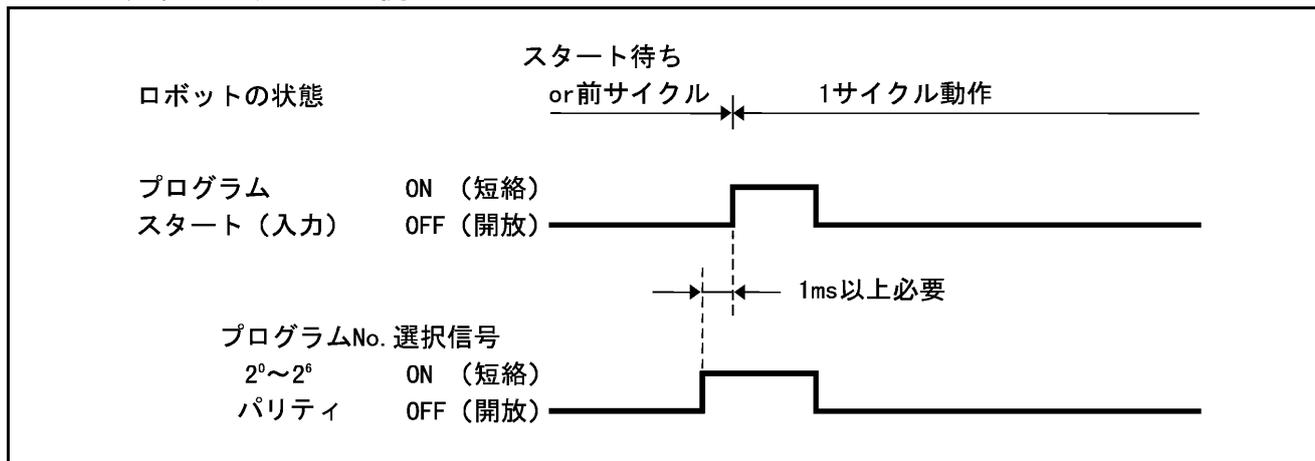
各機能を実行するには、下図のような処理を行います。

・ 運転準備スタートの例



運転準備スタート信号のタイミングチャート例（互換モード）

・プログラムスタートの例



プログラムNo. 選択信号 (互換モード)

13.5.3 互換 I/O 専用出力信号の種類と機能

互換モードでの専用出力信号には下表に示すものがあります。

互換モードでの専用出力信号の種類と機能

用 途	信 号 名	機 能
立ち上げ	ロボット電源入り完了	運転準備スタート可能な状態のときに出力する
	自動モード	ロボットが自動モードになっているときに出力する
	サーボ ON 中	モータ電源入りになっているときに出力する
	CAL 完了	キャリブレーションが完了したときに出力する
	外部モード	ロボットが外部モードになっているときに出力する
プログラム実行開始前チェック	ティーチング中	ロボットが手動モードまたはティーチチェックモードになっているときに出力する
プログラム実行	プログラムスタートリセット	プログラムスタート信号を受けて、プログラム実行をスタートさせたときに出力する
	ロボット運転中	ロボットが運転中（プログラム実行中）であるときに出力する
プログラム終了	1 サイクル終了	プログラムが1サイクル終了したときに出力する
エラー・警告	CPU 正常	ロボットコントローラのCPU（ハードウェア）が正常であるときに出力する
	ロボット異常	サーボ異常、プログラム異常などロボットに異常が発生したときに出力する
	ロボット警告	軽微な異常が発生したときに出力する
	バッテリー切れ警告	エンコーダバックアップ電池またはメモリバックアップ電池の電圧が低下したときに出力する
	エラー番号	エラー発生時にエラー番号をBCDコードで出力する
コンティニュー機能	コンティニュースタート許可	コンティニュースタートが可能な時出力する
SS 機能	SS モード	SSモードのあいだ、出力する
非常停止	非常停止	非常停止回路構成のための専用接点出力

13.6 I/O 割付表

ご使用する割付モードに合わせて、必要な I/O 割付表を参照してください。参照する割付表は、「3.3.1 割付モード別 I/O 割付一覧」の割付モード別各増設ボードの割付表を参照してください。I/O 増設ボードの割付については、オプション機器説明書の「RC7M 用 I/O 増設ボード」を参照してください。

注意：割付モードで「I/O変換Box(互換)」または「I/O変換Box(標準)」を使用する場合は、割付が異なります。RC7M型コントローラ説明書の「第8章 I/O 変換ボックス使用時のI/O 割付表 (標準タイプコントローラのみ)」を参照してください。

13.6.1 HAND I/O (CN9)：全モード共通の割付

RC7M コントローラにはハンド I/O (CN9) 信号を標準搭載しています。このハンド I/O は、モード選択種類に関わらず全モード共通割付となっています。

■ HAND I/O (CN9)：NPN タイプ I/O

端子No.	信号名	ポート 番号	線色		端子No.	信号名	ポート 番号	線色	
			標準	強靱				標準	強靱
1	ハンド出力	64	黒	青	11	ハンド入力	50	桃	白
2	ハンド出力	65	茶	黄	12	ハンド入力	51	桃	白
3	ハンド出力	66	黒	緑	13	ハンド入力	52	白	白
4	ハンド出力	67	茶	赤	14	ハンド入力	53	白	白
5	ハンド出力	68	赤	紫	15	ハンド入力	54	白	白
6	ハンド出力	69	橙	青	16	ハンド入力	55	白	茶
7	ハンド出力	70	黄	黄	17	ハンド用内部電源出力 E24V	—	白	茶
8	ハンド出力	71	緑	緑	18	ハンド用内部電源出力 E0V	—	白	茶
9	ハンド入力	48	青	赤	19	未接続	—	白	茶
10	ハンド入力	49	紫	紫	20	未接続	—	白	茶

■ HAND I/O (CN9)：PNP タイプ I/O

端子No.	信号名	ポート 番号	線色		端子No.	信号名	ポート 番号	線色	
			標準	強靱				標準	強靱
1	ハンド出力	64	黒	青	11	ハンド入力	50	桃	白
2	ハンド出力	65	茶	黄	12	ハンド入力	51	桃	白
3	ハンド出力	66	黒	緑	13	ハンド入力	52	白	白
4	ハンド出力	67	茶	赤	14	ハンド入力	53	白	白
5	ハンド出力	68	赤	紫	15	ハンド入力	54	白	白
6	ハンド出力	69	橙	青	16	ハンド入力	55	白	茶
7	ハンド出力	70	黄	黄	17	ハンド用内部電源出力 E0V	—	白	茶
8	ハンド出力	71	緑	緑	18	ハンド用内部電源出力 E24V	—	白	茶
9	ハンド入力	48	青	赤	19	未接続	—	白	茶
10	ハンド入力	49	紫	紫	20	未接続	—	白	茶

13.6.2 Mini I/O (標準タイプコントローラ CN5) : 専用割付

端子 No.	信号名	ポート番号	線色	端子 No.	名称	ポート番号	線色
1	自動イネーブル入力(内部+24V)	—	黒	35	自動イネーブル	—	桃
2	外部非常停止入力 1b-1 (内部+24V)	—	茶	36	外部非常停止入力 1b-2	—	桃
3	外部非常停止入力 2b-1 (内部+24V)	—	赤	37	外部非常停止入力 2b-2	—	桃
4	予約	—	橙	38	予約	—	桃
5	予約	—	黄	39	予約	—	桃
6	非常停止出力1-1 (ミリレ出力)	—	黒	40	非常停止出力1-2(ミリレ出力)	—	白
7	非常停止出力2-1 (ミリレ出力)	—	茶	41	非常停止出力2-2(ミリレ出力)	—	白
8	デッドマンSW出力1-1(ミリレ出力) (イネーブルSW出力1-1)	—	赤	42	デッドマンSW出力1-2(ミリレ出力) (イネーブルSW出力1-2)	—	白
9	デッドマンSW出力2-1(ミリレ出力) (イネーブルSW出力2-1)	—	橙	43	デッドマンSW出力2-2(ミリレ出力) (イネーブルSW出力2-2)	—	白
10	—	—	黄	44	—	—	白
11	ステップ停止(全タスク) (入力)	0	緑	45	CPU正常 (出力)	16	白
12	ストローブ信号 (入力)	1	青	46	ロボット運転中 (出力)	17	白
13	データ領域 第0ビット (入力)	2	紫	47	ロボット異常 (出力)	18	白
14	データ領域 第1ビット (入力)	3	灰	48	ロボット初期化完了 (出力)	19	白
15	データ領域 第2ビット (入力)	4	桃	49	自動モード (出力)	20	白
16	コマンド領域 第0ビット (入力)	5	黒	50	運転準備完了 (出力)	21	灰
17	コマンド領域 第1ビット (入力)	6	黒	51	バッテリー切れ警告 (出力)	22	紫
18	コマンド領域 第2ビット (入力)	7	茶	52	コマンド処理完了 (出力)	23	紫
19	汎用入力	8	赤	53	汎用出力/コンティニュースタート許可 (出力)	24	紫
20	汎用入力	9	橙	54	汎用出力	25	紫
21	汎用入力	10	黄	55	汎用出力	26	紫
22	汎用入力	11	緑	56	汎用出力	27	紫
23	汎用入力	12	青	57	汎用出力	28	紫
24	汎用入力	13	灰	58	汎用出力	29	紫
25	汎用入力	14	桃	59	汎用出力	30	紫
26	汎用入力	15	茶	60	汎用出力	31	灰
27	—	—	赤	61	—	—	灰
28	ペンダント非常停止出力 1b-1(ドライ出力)	—	橙	62	ペンダント非常停止出力 1b-2(ドライ出力)	—	灰
29	ペンダント非常停止出力2b-1(ドライ出力)	—	黄	63	ペンダント非常停止出力2b-2(ドライ出力)	—	灰
30	Mini I/OボードのJP12 短絡時： コンバートラッキングボード用電源 (DC電源出力+24V)	—	緑	64	Mini I/OボードのJP13 短絡時： コンバートラッキングボード用電源 (DC電源出力 0V)	—	灰
31	—	—	青	65	—	—	灰
32	外部電源設定時：DC電源入力+24V	—	桃	66	外部電源設定時：DC電源入力 0V	—	灰
33	内部電源設定時：DC電源出力+24V	—	黒	67	内部電源設定時：DC電源出力 0V	—	青
34	—	—	茶	68	—	—	青

13.6.3 Mini I/O (グローバルタイプコントローラ CN5) : 専用割付

端子 No.	信号名	ポート番号	線色	端子 No.	信号名	ポート番号	線色
1	予約	—	黒	35	予約	—	桃
2	予約	—	茶	36	予約	—	桃
3	予約	—	赤	37	予約	—	桃
4	予約	—	橙	38	予約	—	桃
5	予約	—	黄	39	予約	—	桃
6	予約	—	黒	40	予約	—	白
7	予約	—	茶	41	予約	—	白
8	予約	—	赤	42	予約	—	白
9	予約	—	橙	43	予約	—	白
10	—	—	黄	44	—	—	白
11	ステップ停止(全タスク) (入力)	0	緑	45	CPU正常 (出力)	16	白
12	ストローブ信号 (入力)	1	青	46	ロボット運転中 (出力)	17	白
13	データ領域 第0ビット (入力)	2	紫	47	ロボット異常 (出力)	18	白
14	データ領域 第1ビット (入力)	3	灰	48	ロボット初期化完了 (出力)	19	白
15	データ領域 第2ビット (入力)	4	桃	49	自動モード (出力)	20	白
16	コマンド領域 第0ビット (入力)	5	黒	50	運転準備完了 (出力)	21	灰
17	コマンド領域 第1ビット (入力)	6	黒	51	バッテリー切れ警告 (出力)	22	紫
18	コマンド領域 第2ビット (入力)	7	茶	52	コマンド処理完了 (出力)	23	紫
19	汎用入力	8	赤	53	汎用出力/コンティニュースタート許可(出力)	24	紫
20	汎用入力	9	橙	54	汎用出力	25	紫
21	汎用入力	10	黄	55	汎用出力	26	紫
22	汎用入力	11	緑	56	汎用出力	27	紫
23	汎用入力	12	青	57	汎用出力	28	紫
24	汎用入力	13	灰	58	汎用出力	29	紫
25	汎用入力	14	桃	59	汎用出力	30	紫
26	汎用入力	15	茶	60	予約	31	灰
27	—	—	赤	61	—	—	灰
28	予約	—	橙	62	予約	—	灰
29	予約	—	黄	63	予約	—	灰
30	Mini I/OボードのJP12 短絡時： コンベアトラッキングボード用電源 (DC電源出力+24V)	—	緑	64	Mini I/OボードのJP13 短絡時： コンベアトラッキングボード用電源 (DC電源出力 0V)	—	灰
31	—	—	青	65	—	—	灰
32	外部電源設定時：DC電源入力+24V	—	桃	66	外部電源設定時：DC電源入力 0V	—	灰
33	内部電源設定時：DC電源出力+24V	—	黒	67	内部電源設定時：DC電源出力 0V	—	青
34	—	—	茶	68	—	—	青

13.6.4 Mini I/O (標準タイプコントローラ CN5) : 互換/標準/全汎用割付

端子 No.	信号名	ポート番号	線色	端子 No.	信号名	ポート番号	線色
1	自動イネーブル入力(内部+24V)	—	黒	35	自動イネーブル	—	桃
2	外部非常停止入力 1b-1 (内部+24V)	—	茶	36	外部非常停止入力 1b-2	—	桃
3	外部非常停止入力 2b-1 (内部+24V)	—	赤	37	外部非常停止入力 2b-2	—	桃
4	予約	—	橙	38	予約	—	桃
5	予約	—	黄	39	予約	—	桃
6	非常停止出力1-1 (ミリレ出力)	—	黒	40	非常停止出力1-2 (ミリレ出力)	—	白
7	非常停止出力2-1 (ミリレ出力)	—	茶	41	非常停止出力2-2 (ミリレ出力)	—	白
8	デッドマンSW出力1-1 (ミリレ出力) (イネーブルSW出力1-1)	—	赤	42	デッドマンSW出力1-2 (ミリレ出力) (イネーブルSW出力1-2)	—	白
9	デッドマンSW出力2-1 (ミリレ出力) (イネーブルSW出力2-1)	—	橙	43	デッドマンSW出力2-2 (ミリレ出力) (イネーブルSW出力2-2)	—	白
10	—	—	黄	44	—	—	白
11	汎用入力	0	緑	45	CPU正常(出力)	16	白
12	汎用入力	1	青	46	汎用出力	17	白
13	汎用入力	2	紫	47	汎用出力	18	白
14	汎用入力	3	灰	48	汎用出力	19	白
15	汎用入力	4	桃	49	汎用出力	20	白
16	汎用入力	5	黒	50	汎用出力	21	灰
17	汎用入力	6	黒	51	汎用出力	22	紫
18	汎用入力	7	茶	52	汎用出力	23	紫
19	汎用入力	8	赤	53	汎用出力	24	紫
20	汎用入力	9	橙	54	汎用出力	25	紫
21	汎用入力	10	黄	55	汎用出力	26	紫
22	汎用入力	11	緑	56	汎用出力	27	紫
23	汎用入力	12	青	57	汎用出力	28	紫
24	汎用入力	13	灰	58	汎用出力	29	紫
25	汎用入力	14	桃	59	汎用出力	30	紫
26	汎用入力	15	茶	60	汎用出力	31	灰
27	—	—	赤	61	—	—	灰
28	ペンダント非常停止出力 1b-1 (ドライ出力)	—	橙	62	ペンダント非常停止出力 1b-2 (ドライ出力)	—	灰
29	ペンダント非常停止出力 2b-1 (ドライ出力)	—	黄	63	ペンダント非常停止出力 2b-2 (ドライ出力)	—	灰
30	Mini I/OボードのJP12 短絡時： コンバートラッキングボード用電源 (DC電源出力+24V)	—	緑	64	Mini I/OボードのJP13 短絡時： コンバートラッキングボード用電源 (DC電源出力 0V)	—	灰
31	—	—	青	65	—	—	灰
32	外部電源設定時：DC電源入力+24V	—	桃	66	外部電源設定時：DC電源入力 0V	—	灰
33	内部電源設定時：DC電源出力+24V	—	黒	67	内部電源設定時：DC電源出力 0V	—	青
34	—	—	茶	68	—	—	青

13.6.5 Mini I/O (グローバルタイプコントローラ CN5) : 互換/標準/全汎用割付

端子 No.	信号名	ポート番号	線色	端子 No.	信号名	ポート番号	線色
1	予約	—	黒	35	予約	—	桃
2	予約	—	茶	36	予約	—	桃
3	予約	—	赤	37	予約	—	桃
4	予約	—	橙	38	予約	—	桃
5	予約	—	黄	39	予約	—	桃
6	予約	—	黒	40	予約	—	白
7	予約	—	茶	41	予約	—	白
8	予約	—	赤	42	予約	—	白
9	予約	—	橙	43	予約	—	白
10	—	—	黄	44	—	—	白
11	汎用入力	0	緑	45	CPU正常(出力)	16	白
12	汎用入力	1	青	46	汎用出力	17	白
13	汎用入力	2	紫	47	汎用出力	18	白
14	汎用入力	3	灰	48	汎用出力	19	白
15	汎用入力	4	桃	49	汎用出力	20	白
16	汎用入力	5	黒	50	汎用出力	21	灰
17	汎用入力	6	黒	51	汎用出力	22	紫
18	汎用入力	7	茶	52	汎用出力	23	紫
19	汎用入力	8	赤	53	汎用出力	24	紫
20	汎用入力	9	橙	54	汎用出力	25	紫
21	汎用入力	10	黄	55	汎用出力	26	紫
22	汎用入力	11	緑	56	汎用出力	27	紫
23	汎用入力	12	青	57	汎用出力	28	紫
24	汎用入力	13	灰	58	汎用出力	29	紫
25	汎用入力	14	桃	59	汎用出力	30	紫
26	汎用入力	15	茶	60	予約	31	灰
27	—	—	赤	61	—	—	灰
28	予約	—	橙	62	予約	—	灰
29	予約	—	黄	63	予約	—	灰
30	Mini I/OボードのJP12 短絡時 : コンベアトラックボード用電源 (DC電源出力+24V)	—	緑	64	Mini I/OボードのJP13 短絡時 : コンベアトラックボード用電源 (DC電源出力 0V)	—	灰
31	—	—	青	65	—	—	灰
32	外部電源設定時 : DC電源入力+24V	—	桃	66	外部電源設定時 : DC電源入力 0V	—	灰
33	内部電源設定時 : DC電源出力+24V	—	黒	67	内部電源設定時 : DC電源出力 0V	—	青
34	—	—	茶	68	—	—	青

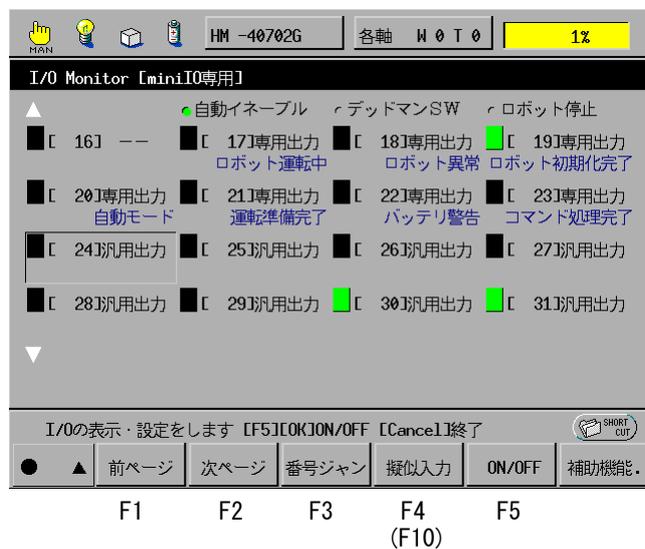
第 14 章 I/O のモニタリングと操作

専用入出力、汎用入出力、内部 I/O の状態を、リアルタイムで監視することや、汎用出力・HAND 出力・内部 I/O 信号の強制出力や、汎用入力・HAND 入力信号を擬似的に I/O 入力し、ロボット動作のシミュレートを行うことができます。

14.1 ティーチングペンダントによる操作

14.1.1 I/O のモニタリング

基本画面から [F4 I/O] を押すと、下図の [I/O Monitor] ウィンドウが表示されます。この画面にて I/O の ON・OFF 状態が確認できます。



使用可能なファンクションキー	
[F1 前ページ]	I/O信号リストの直前のページが表示されます。
[F2 次ページ]	I/O信号リストの直後のページが表示されます。
[F3 番号ジャン]	[I/O番号ジャンプ] ウィンドウが表示されます。テンキーを使ってI/Oポートアドレスを入力し、[OK]を押すと、目的の入力信号または出力信号が表示されます。
[F4 擬似入力]	選択してあるシステム入力ポートで擬似入力を受け取ることができますようになります。該当する入力ポートに「!」記号が表示され、画面上部のステータスバーに擬似I/Oアイコンが表示されます。この機能は、プログラムをテストする際に使用します。
[F5 ON/OFF]	「I/O[xxx]をON (またはOFF) にして良いですか?」というシステムメッセージが表示されます。[OK]ボタンを押すと、選択してある入力ポートがON (またはOFF) になります。汎用出力、HAND 出力、内部I/Oで使用できます。 ON/OFFできない番号を指定した場合には、ERROR 21FB(システムエリア書込みエラー)またはERROR 73E4(入出力可能範囲外です)が発生します。
[F10 擬似クリア]	擬似入力設定が解除されます。

14.1.2 擬似 I/O の操作方法

汎用入力、HAND 入力に限り、擬似 I/O 入力を設定する事ができます。
擬似 I/O 入力を設定すると、該当ポートをティーチングペンダントにて ON・OFF できます。

・ 擬似 I/O 入力の設定

基本画面から [F4 I/O] を選択し、[I/O Monitor] ウィンドウを示させます。

The first screenshot shows the I/O Monitor screen with various input ports. The '11汎用入力' (General Input 11) checkbox is circled. A callout box explains that the user should select the desired I/O number (11) by touching it with a cursor key or joystick. Another callout points to the '擬似入力' (Simulation Input) button at the bottom, indicating it should be pressed.

The second screenshot shows a system message dialog box asking 'I/O 11に擬似入力を設定しますか?' (Do you want to set simulation input for I/O 11?). The 'OK' button is circled. A callout box instructs the user to press the 'OK' button while also pressing the 'デッドマンスイッチ' (Deadman Switch).

The third screenshot shows the I/O Monitor screen again, but now the '11汎用入力' checkbox has a small exclamation mark (!) next to it. A callout box explains that this mark indicates that simulation input is possible for that port. Another callout points to the '擬似入力' button, noting that a mark (!) also appears next to it when simulation input is enabled.

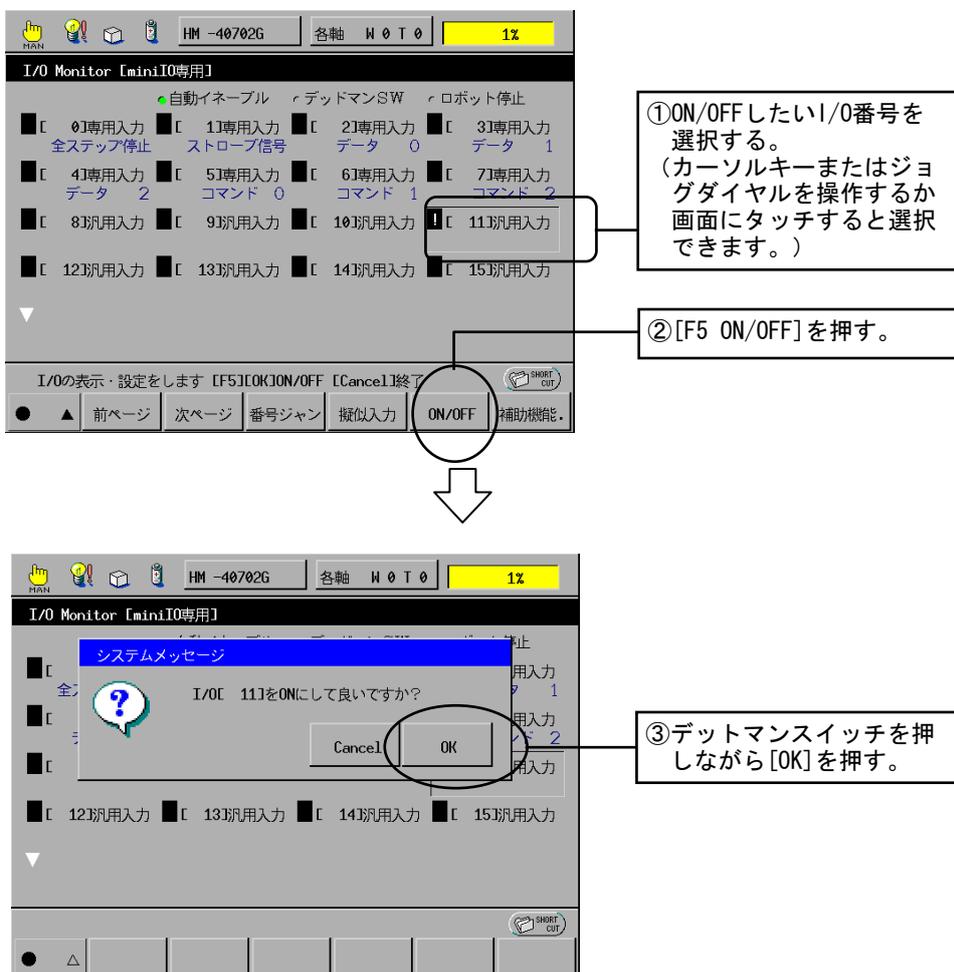
これで擬似I/O入力の設定は完了です。

★参考★

擬似I/O入力を解除するには、再度①～③の操作をおこなうか、[F10 擬似クリア] (擬似入力全クリア)をおこなってください。

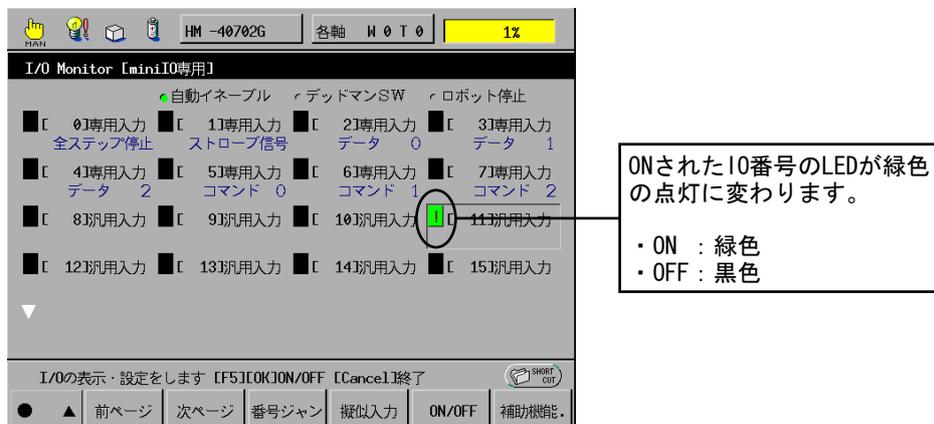
・ 疑似 I/O の ON/OFF 方法

ここでは、疑似I/OをOFFからONさせる操作方法について説明します。



★注意★

ビックリマーク"!"が表示されていないI/O番号をON/OFFするとERROR 73E4(入出力可能範囲外です)が発生します。



★参考★

疑似I/O入力をOFFするには、再度①～③の操作をおこなってください。

14.2 WINGCAPSⅢによる操作

WINGCAPSⅢを使ってロボットコントローラの I/O の状態をモニタリングしたり、擬似 I/O 機能を使いプログラムの検証をしたりできます。

14.2.1 I/O のモニタ

WINGCAPSⅢで I/O の状態をモニタします。

ステップ 1 プロジェクトを開き、接続状態を[オンライン(モニタ)]にします。



ステップ 2 [表示 (V)]-[I/O (I)]を選択し、[I/O]ウィンドウを表示させます。

モニタしたい I/O までスクロールし、I/O の状態を確認します。

The screenshot shows the "I/O" monitoring window. It has a "番号ジャンプ" (Jump to Number) field and a "スマート表示" (Smart Display) button. The main area is a table with the following columns: No, 状態 (Status), タイプ (Type), 用途 (Usage), マクロ名 (Macro Name), 擬似 (Simulation), ログ (Log), and スマート (Smart). The table contains 9 rows of data.

No	状態	タイプ	用途	マクロ名	擬似	ログ	スマート
19	●	専用出力	ロボット初期化完了	SOUT4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20	●	専用出力	自動モード	SOUT5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
21	●	専用出力	運転準備完了	SOUT6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
22	●	専用出力	バッテリー切れ警告	SOUT7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
23	●	専用出力	コマンド処理完了	SOUT8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
24	●	汎用出力		UOUT1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
25	●	汎用出力		UOUT2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
26	●	汎用出力		UOUT3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
27	●	汎用出力		UOUT4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
28	●	汎用出力		UOUT5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

[状態]欄が緑丸の場合が ON です。

ステップ 3 スマート表示機能を使って見たい I/O だけを表示できます。

表示させたい I/O の[スマート]欄にチェックを入れます。
[スマート表示]ボタンを押すと、[スマート]欄にチェックを入れた I/O のみ表示されます。

The screenshot shows the "I/O" monitoring window with the "スマート表示" (Smart Display) button highlighted. The table shows only the rows where the "スマート" (Smart) column has a checkmark.

No	状態	タイプ	用途	マクロ名	擬似	ログ	スマート
8	●	汎用入力		UIN1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	●	汎用入力		UIN2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20	●	専用出力	自動モード	SOUT5	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
24	●	汎用出力		UOUT1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

14.2.2 擬似 I/O を使う

汎用入力、HAND 入力に限り、擬似的に I/O の入力を行う、擬似 I/O 機能を使うことができます。

擬似 I/O 機能を使うと、WINCAPSⅢの[I/O]ウインドウから I/O の ON→OFF、OFF→ON を行うことができます。

ステップ 1

プロジェクトを開き、接続状態を[オンライン(モニタ)]にします。

[I/O]ウインドウを開きます。

擬似 I/O 機能を使いたい I/O の[擬似]欄にチェックを入れます。



No	状態	タイプ	用途	マクロ名	擬似	ログ
6	●	専用入力	コマンド領域 第1ビット	SIN7	<input type="checkbox"/>	<input type="checkbox"/>
7	●	専用入力	コマンド領域 第2ビット	SIN8	<input type="checkbox"/>	<input type="checkbox"/>
8	●	汎用入力		UIN1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	●	汎用入力		UIN2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	●	汎用入力		UIN3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	●	汎用入力		UIN4	<input type="checkbox"/>	<input type="checkbox"/>
12	●	汎用入力		UIN5	<input type="checkbox"/>	<input type="checkbox"/>
13	●	汎用入力		UIN6	<input type="checkbox"/>	<input type="checkbox"/>
14	●	汎用入力		UIN7	<input type="checkbox"/>	<input type="checkbox"/>

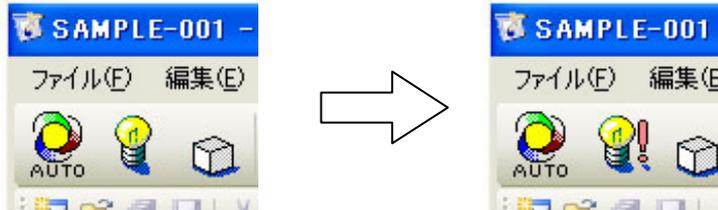
ステップ 2

[擬似入力]ボタンを押すと[擬似]欄にチェックを入れた I/O を WINCAPSⅢから操作できるようになります。



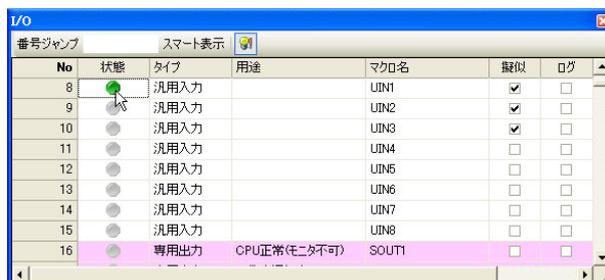
No	状態	タイプ	用途	マクロ名	擬似	ログ
6	●	専用入力	コマンド領域 第1ビット	SIN7	<input type="checkbox"/>	<input type="checkbox"/>
7	●	専用入力	コマンド領域 第2ビット	SIN8	<input type="checkbox"/>	<input type="checkbox"/>
8	●	汎用入力		UIN1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	●	汎用入力		UIN2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	●	汎用入力		UIN3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	●	汎用入力		UIN4	<input type="checkbox"/>	<input type="checkbox"/>
12	●	汎用入力		UIN5	<input type="checkbox"/>	<input type="checkbox"/>
13	●	汎用入力		UIN6	<input type="checkbox"/>	<input type="checkbox"/>
14	●	汎用入力		UIN7	<input type="checkbox"/>	<input type="checkbox"/>

擬似入力モードでは I/O アイコンに[!]マークが付きます。



ステップ 3

I/O の ON⇔OFF をするには[状態]欄をクリックします。



No	状態	タイプ	用途	マクロ名	擬似	ログ
8	●	汎用入力		UIN1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	●	汎用入力		UIN2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	●	汎用入力		UIN3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	●	汎用入力		UIN4	<input type="checkbox"/>	<input type="checkbox"/>
12	●	汎用入力		UIN5	<input type="checkbox"/>	<input type="checkbox"/>
13	●	汎用入力		UIN6	<input type="checkbox"/>	<input type="checkbox"/>
14	●	汎用入力		UIN7	<input type="checkbox"/>	<input type="checkbox"/>
15	●	汎用入力		UIN8	<input type="checkbox"/>	<input type="checkbox"/>
16	●	専用出力	CPU正常(モニタ不可)	SOUT1	<input type="checkbox"/>	<input type="checkbox"/>

第 15 章 変数のモニタリングと値変更

15.1 ティーチングペンダントによる操作

15.1.1 グローバル変数のモニタリングと値変更

操作経路: [F1 プログラム]-[F4 変数.]

各種変数の値や使用中の変数を、モニタリング・変更します。

- (1) [プログラム一覧]ウィンドウで[F4 変数.]を押すと、下図の[変数タイプ選択]ウィンドウが表示されます。



- (2) モニタリング又は変更したい変数のタイプを選択します。

[F1 I. 整数]を押すと、次のような[整数型]ウィンドウが表示されます。



使用可能なファンクションキー	
[F1 前ページ]	変数リストの直前のページが表示されます。
[F2 次ページ]	変数リストの直後のページが表示されます。
[F3 番号ジャン]	[変数番号ジャンプ]ウィンドウが表示されます。テンキーを使って変数名を入力し、[OK]を押すと、目的の変数名が表示されます。
[F5 値変更.]	テンキーが表示されます。変数に代入する値をテンキーから入力し、[OK]を押すと、その値が変数に代入されます。
[F7 変数コピー]	選択している変数をコピーします。
[F12 登録]	選択している変数をウォッチに登録します。

注意：外部自動モードでは値変更できません

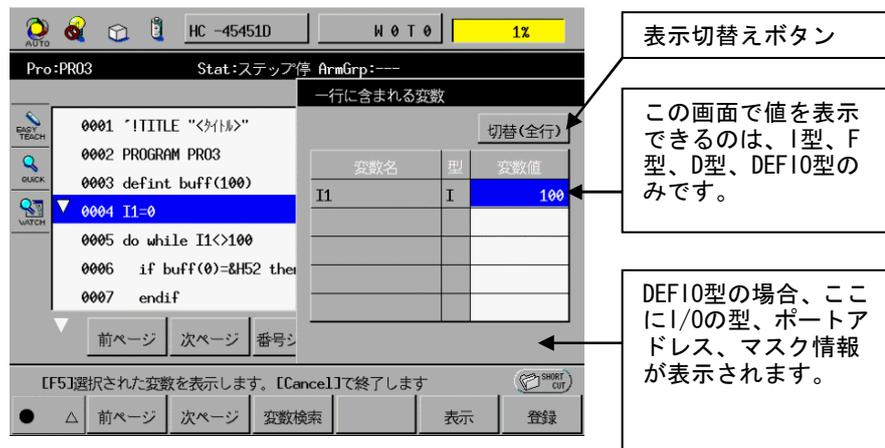
15.1.2 ローカル変数のモニタリングと値変更

プログラムテキストが表示されている画面に、即表示機能ボタンが追加されています。即表示ボタンを押すことにより、現在カーソル行のローカル変数、およびグローバル変数を参照します。



注意：カーソル行はマニュアルモードでしか変更できません。

即表示機能ボタンを押すことにより、1行に含まれる変数が表示されます。I型、F型、D型、DEF10型のみこの画面で値を見ることができます。DEF10変数を参照した場合、<I/O変数の型>、<ポートアドレス>、<マスク情報>が表示されます。次例は、ステップ停止行の変数「I1」を表示しています。



注意 1：変数のINDEXが範囲外、および数値でない場合、変数名のINDEX部が?で表示されます。

例 1. I型変数の使用数が100であるのにI101を表示しようとした。

例 2. I[slotnum]のようにINDEX部がマクロ名である変数を表示しようとした。

INDEX部が?の場合I型、F型、D型、DEF10型であっても値は表示されません。表示ボタンを押し、目的のINDEXへ合わせてください。

注意 2：DEF10型変数でポートアドレスが入出力範囲外の場合グレー表示になります。また、SINGLE型のDEF10型変数は、I0の内容が単精度実数として表現できない場合、“NaN”と表示されます。

注意 3：引数が配列である変数は表示されません。

例. PROGRAM SUB1(Ii%, Ii2%(10))などのプログラムである場合、引数が配列変数であるIi2は表示されません。

即表示機能では、「表示切替えボタン」により、1行に含まれる変数と全行に含まれる変数の表示を切替えることができます。全行に切替えるとプログラム内のすべてのローカル変数が表示されます。次例は全行に含まれる変数の画面に切替えた画面です。

表示切替えボタン

全行表示であるため変数名のINDEX部は?表示となります。

表示される型は I, F, D, V, P, J, T, S, I O があります。

変数名	型	変数値
BUFF(?)	I	-----
IX	I	0
IJ	J	-----
DX	D	0.000000

注意 1 : 1行表示の場合グローバル変数も表示されますが、全行表示の場合グローバル変数は表示されません。

注意 2 : 全行表示の場合、配列変数はすべて変数名のINDEX部が?で表示されます。表示ボタンを押し、目的のINDEXへ合わせてください。

表示ボタンを押すことにより、選択された変数が表示されます。

表示ボタン

変数名	型	変数値
PX	P	-----
BUFF(?)	I	-----
IX	I	0
IJ	J	-----
DX	D	0.000000

注意 : DEFIO変数のポートが範囲外の場合、表示できません。

表示ボタンを押すことにより、選択された変数が表示されます。

次画面は、P型ローカル変数PX(3)を表示した例です。

この画面では値変更や変数移動など、グローバル変数同等の操作を実行することができます。この画面での変数の登録は「F12 登録」になります。

変数名	X/T	Y/RL	Z/Fig
PX(3)	0.0000000	0.0000000	0.0000000
	0.0000000	?	FIG -1
PX(4)	0.0000000	0.0000000	0.0000000
	0.0000000	?	FIG -1
PX(5)	0.0000000	0.0000000	0.0000000
	0.0000000	?	FIG -1

ペンダントのSHIFT
ボタンを押すと、
ここがF12の登録に
なる。

注意1：変数のINDEX部が?の場合、表示ボタンを押すとINDEXが0の変数が表示されます。目的のINDEXへカーソルを合わせてください。

注意2：DEF10変数の値を変更する場合は、通常のI/Oと同様にデッドマンスイッチを押す必要があります。

注意3：ローカル変数への位置取り込みはありません。

値変更する場合はこの画面で[F5 値変更]を押すとテンキーが表示されます。変数に代入する値をテンキーから入力し[OK]を押すと、その値が変数に代入されます。

注意：外部自動モードでは値変更できません

15.1.3 グローバル変数の使用個数変更

操作経路[F1 プログラム]-[F4 変数.]-[F12 変数使用数]

各型のグローバル変数の使用個数を、変更します。

- (1) [F12 変数使用数]を押すと、次のような[変数使用数 確認]ウィンドウが表示されます。



- (2) 使用数を変更したい項目を選択し、「F5 設定変更」を押すと、次のような[パラメータ変更]ウィンドウが表示されます。



- (3) 変更値を入力し、[OK]を押すと変更された値が[変数使用数 確認]ウィンドウに表示されます。



- (4) 設定内容を確認した後、[OK]を押すと、次のようにシステムメッセージが表示されます。ここで[OK]を押すとコンパイルが行われます。正常にコンパイル・ロードが完了するとその時点で変数の使用数を変更されます。



- (5) (4) で[Cancel]を押すと、この時点では変数の使用数を変更されず、次回コンパイル・ロードが行われるまで使用数の変更は有効となりません。

注：

[グローバル変数の数について]

このコントローラでは、実行プログラムのロードが行われたときにのみ変数の使用数を変更することができます。

そのため、変数の使用数の設定を変更すると、まずコンパイルによって変数使用数の変更が反映された実行ファイルの作成が行われ、その後ロードされます。ここでロードが完了した時点から、変更した変数を使うことができます。

15.2 WINCAPSⅢによる操作

WINCAPSⅢでロボットコントローラのグローバル変数やローカル変数をモニタ、編集することができます。

15.2.1 グローバル変数のモニタリングと値変更

WINCAPSⅢを使ってロボットコントローラのグローバル変数のモニタリングと値の変更ができます。

ステップ1 プロジェクトを開き、接続状態を[オンライン(モニタ)]にします。

[表示 (V)]-[変数 (V)]を選択し、モニタしたい変数型を選択します。
選択された変数型ウインドウが表示されます。

No	値	用途	マクロ名	スマート
0	1005	部品番号	iPartsID	<input type="checkbox"/>
1	562	部品Aの個数	iCountA	<input type="checkbox"/>
2	120	部品Bの個数	iCountB	<input type="checkbox"/>
3	5	部品Cの個数	iCountC	<input type="checkbox"/>
4	0			<input type="checkbox"/>
5	0			<input type="checkbox"/>

ステップ2 [値]欄に値を入力することで、ロボットコントローラの変数を編集できます。

変数の編集はロボットコントローラ内のデータは変更されますが、WINCAPSⅢプロジェクトの値は変更されません。

保存したい場合は[データの送受信]にてロボットコントローラから変数データを受信してください。

15.2.2 ローカル変数のモニタリングと値変更

WINCAPSⅢを使って、ロボットコントローラのローカル変数のモニタリングと値の変更ができます。

ステップ1 プロジェクトを開き、接続状態を[オンライン(モニタ)]にします。

[表示 (V)]-[ローカル変数 (L)]を選択します。

[プロジェクトウインドウ]もしくは[プログラム一覧]ウインドウで選択されているプログラムのローカル変数が表示されます。

変数名	値	型
HH		I(3, 4, 3)
HH(0)		I(3, 4)
HH(0, 0)		I(3)
HH(0, 0, 0)	73	I
HH(0, 0, 1)	74	I
HH(0, 0, 2)	75	I
HH(0, 1)		I(3)
HH(0, 2)		I(3)
HH(0, 3)		I(3)
HH(1)		I(3, 4)
HH(2)		I(3, 4)
AA	1	I
BB	3	I
CC	97	I
DD	2	I

ステップ2 [値]欄に値を入力することで、ロボットコントローラの変数を編集できます。

注意：DEFI0で汎用入力ポートやハンド入力ポートを宣言している場合はI/Oを擬似入力設定しておく必要があります。

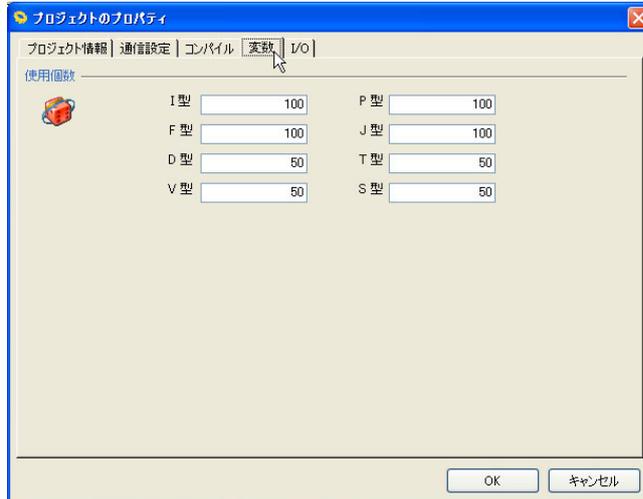
15.2.3 変数使用個数の変更

WINCAPS IIIで変数の使用個数を設定できます。

ステップ1 プロジェクトを開き、[プロジェクト (P)]-[プロパティ (P)]を選択します。

[プロジェクトのプロパティ]ウィンドウが開きます。

[変数]タブをクリックします。

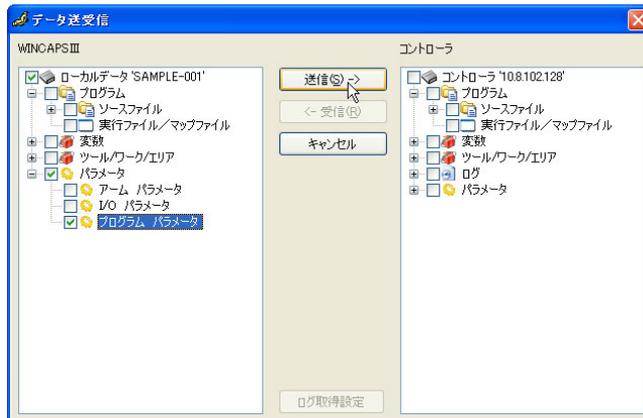


変数の個数を変更し、[OK]を押します。

ステップ2 データをロボットコントローラへ送信します。

[通信 (N)]-[データ送受信 (T)]を選択すると[データ送受信]ウィンドウが表示されます。

WINCAPS III側の[パラメータ]-[プログラム パラメータ]にチェックを入れ、[送信 (S) ->]をクリックします。



ステップ3 ロボットコントローラで[F1 プログラム]-[F6 補助機能]-[F12 コンパイル]を選択します。

プログラムがコンパイルされ、自動的にプロジェクトがロードされます。
変数使用数の変更が反映されます。

第5部

さらに高度な使い方

第 16 章 使用条件の最適化設定

16.1 ロボットの設置条件(床置き・天吊り)設定方法[6 軸ロボット]

6 軸ロボットでは床置き・天吊りの設置方法に応じ、パラメータ設定が必要です。床置きは「0」、天吊りは「1」にパラメータを設定します。工場出荷時は、床置き「0」に設定されていますので、天吊りで使用する場合には、設定を変更してください。

16.1.1 ロボット設置条件(床置、天吊)設定の目的

電流制限機能や力制限機能を使用する場合、重力補償設定を有効にする必要があります。この重力補償の方向を設置条件(床置、天吊り)から判断させています。

16.1.2 ティーチングペンダントによる設定

操作経路：基本画面-[F2 アーム]-[F6 補助機能.]-[F7 使用条件.]

ティーチングペンダントを操作して、上記の操作経路を経ると、[使用条件 (パラメータ数:)] 画面が現れます。



この [使用条件 (パラメータ数:)] 画面で、[床置き、天吊り設定] の項目を選択し、[F5 設定変更] を押すと、テンキー形式の [パラメータ変更] 画面が表示され、各パラメータ値を変更できます。設置条件は「0」、「1」のいずれかに設定してください。それ以外の値を入力すると「入力された値が範囲外です。」が発生します。

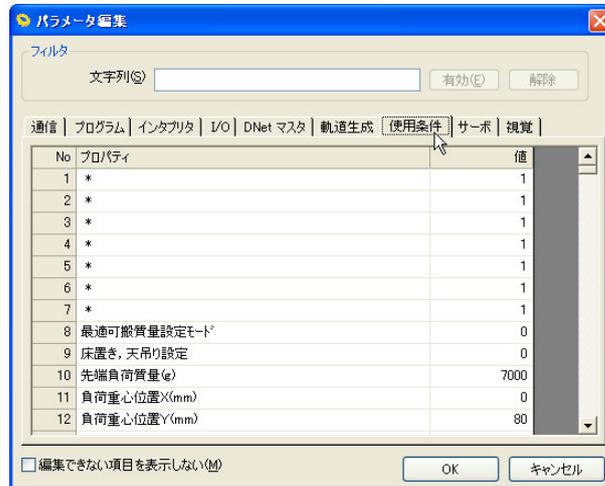


注意：ティーチングペンダントで変更した使用条件は、WINCAPS IIIにて[パラメータ]-[アーム パラメータ]を受信してください。

16.1.3 WINCAPSⅢによる設定

WINCAPSⅢによって、床置き「0」・天吊り「1」の設定を行なう方法について説明します。

[プロジェクト (P)]- [パラメータ (T)] - [使用条件] タブを選択します。



[床置き、天吊り設定] の値欄をダブルクリックすると、パラメータ値を入力変更できます。

注意：WINCAPSⅢで設定した使用条件は、データ送受信機能にて[アーム パラメータ]を送信してください。

パラメータ値の設定ができれば、ロボットコントローラにデータを転送します。

まず、ティーチングペンダントの[MOTOR]で、モータ電源をOFFにします。

[通信 (N)]-[データ送受信 (T)]を選択します。

[パラメータ]-[アーム パラメータ]にチェックを入れ、[送信 (S)->]をクリックします。



16.2 最適可搬質量設定機能

ロボットの先端負荷や姿勢に応じて、最適な速度、加速度を設定する機能です。表 4-2 に示す 4 つのモードを選択できます。

表 16-1 最適可搬質量のモード

モード	設定条件	設定内容	
		PTP 動作	CP 動作
0	負荷条件値	最高加速度	最高加速度
1	負荷条件値 ロボットの姿勢	最高速度、加速度	モード0と同じ
2		モード0と同じ	最高速度、加速度
3		モード1と同じ	モード2と同じ

16.2.1 モード 0

工場出荷時のデフォルトのモードです。ロボットの負荷条件値により、PTP 動作、CP 動作時の最高加速度を設定します。

位置決め時間はロボット概要書「第 3 章 ロボットの位置決め時間」を参照してください。

16.2.2 モード 1

ロボットの負荷条件値と、動作中のロボットの姿勢により、PTP 動作時の 1、2、3 軸の最高速度と加速度を設定します。PTP 動作時の 4、5、6 軸と CP 動作は、モード 0 と同様です。

モード 1 を使用する場合

PTP 動作の動作時間短縮が必要な場合、モード 1 に設定してください。

モード 1 使用時の注意事項

動作中に過負荷エラーや偏差過大エラーが発生する場合があります。負荷率については、調整時にティーチングペンダントの過負荷予想値（操作ガイド「5.3 ロボットの現在位置の表示、[F2]-[F6]-[F10]」参照）または、WINCAPSⅢのログ機能を利用して、確認してください。

過負荷エラーが発生する場合は、タイマや内部速度、加速度を設定し、モータ負荷を調整してください。

偏差過大エラーが発生する場合は、速度、加速度を調整してください。

動作速度により、パス軌道が 20mm 程度変化する場合があります。したがって、障害物近傍のパス動作は、障害物に干渉する可能性があるため、モード 0 で実行してください。

16.2.3 モード2

ロボットの負荷条件値と動作中のロボットの姿勢により、CP動作時の最高速度と加速度を設定します。PTP動作時は、モード0と同様です。

モード2を使用する場合

以下の2つの場合、モード2に設定してください。

- ①CP動作時の動作時間短縮が必要な場合
- ②指令速度制限オーバーエラーを回避させたい場合

CP動作中に指令速度制限オーバーエラー(6081～6086)が発生する場合や、CP動作時の軌道が特異点近傍(操作ガイド「4.1.3 腕・ひじ・手首の形態について [2] 形態の境界」参照)や、動作範囲リミット近傍を通過する場合、指令速度制限オーバーエラーが発生し、停止することがあります。この場合、自動的に指令速度制限内に速度を低減し、エラーを発生せずに動作させることができます。

モード2使用時の注意事項

- ・動作中に過負荷エラーが発生する場合があります。調整時にティーチングペンダントの過負荷予想値(操作ガイド「5.3 ロボットの現在位置の表示、[F2]-[F6]-[F10]」参照)または、WINCAPSⅢのログ機能を利用して負荷率を確認してください。過負荷エラーが発生する場合は、タイマや内部速度、加速度を設定し、モータ負荷を調整してください。
- ・動作速度により、パス軌道が20mm程度変化する場合があります。したがって、障害物近傍のパス動作は、障害物に干渉する可能性があるため、モード0で実行してください。
- ・CP動作時に、等速移動区間において、速度が変化する可能性があるため、等速移動を必要とする作業は、モード0かモード1で実行してください。
- ・CP動作時に指令加速度制限オーバーエラー(6761～6766)、偏差過大エラー(6111～6116)が発生する場合があります。エラーが発生した場合は、内部速度、内部加速度にて速度、加速度を調整してください。また、高速動作時最大5mm程度の軌跡ずれが発生する場合があります。動作近傍に障害物がある場合は、速度を下げてください。
- ・特異点近傍(操作ガイド「4.1.3 腕・ひじ・手首の形態について [2] 形態の境界」参照)で動作速度が低下中に瞬時停止させた場合、瞬時停止時間が延びる場合があります。ただし、瞬時停止距離は変わりません。

16.2.4 モード3

PTP動作時はモード1、CP動作時はモード2と同様に動作します。

16.3 最適可搬質量設定機能の設定方法

16.3.1 ティーチングペンダントによる設定

操作経路：基本画面-[F2 アーム]-[F6 補助機能.]-[F7 使用条件.]

ティーチングペンダントを操作して、上記の操作経路を経ると、[使用条件 (パラメータ数:)] 画面が現れます。現在の内部負荷条件値と内部モードなどが表示されます。



この [使用条件 (パラメータ数:)] 画面で、以下の項目を選択し、[F5 設定変更.] を押すと、テンキー形式の [パラメータ変更] 画面が表示され、各パラメータ値を変更できます。

設定項目：

[7:最適可搬質量設定モード]、[9:先端負荷質量 (g)]、[10:負荷重心位置 X (mm)]
[11:負荷重心位置 Y (mm)]、[12:負荷重心位置 Z (mm)]

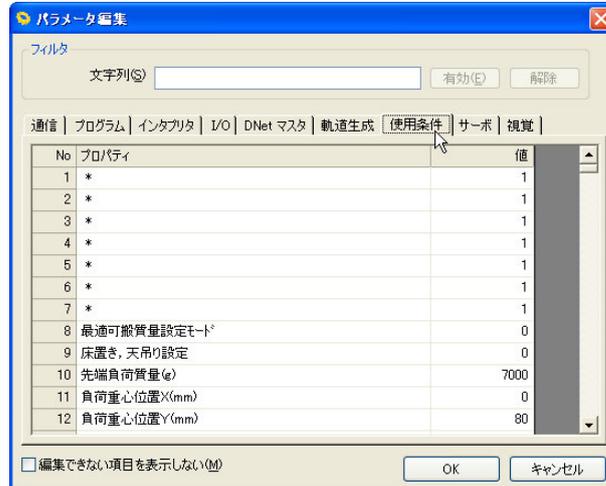
但し、4軸ロボットの Ver. 1.9 以降は [12:負荷イナーシャ (kgcm²)]

- ・最適可搬質量設定モードは 0～3 で、それ以外の数値を入力すると「入力された値が範囲外です。」が発生します。
- ・先端負荷質量は、ロボット毎に設定された許容値を超えた値を入力すると「入力された値が範囲外です。」が発生します。
- ・負荷重心位置は、許容範囲を満たすように入力してください。ロボット毎に設定された許容範囲を満たさない場合は、「入力された値が範囲外です。」が発生します。

16.3.2 WINCAPSⅢによる設定

WINCAPSⅢによって、外部負荷条件値（先端負荷質量、重心位置）と外部モードの設定を行なう方法について説明します。

[プロジェクト (P)] -[パラメータ (T)]-[使用条件]タブを選択します。



以下の項目の設定値欄をダブルクリックすると、各パラメータ値を入力変更できます。

設定項目：

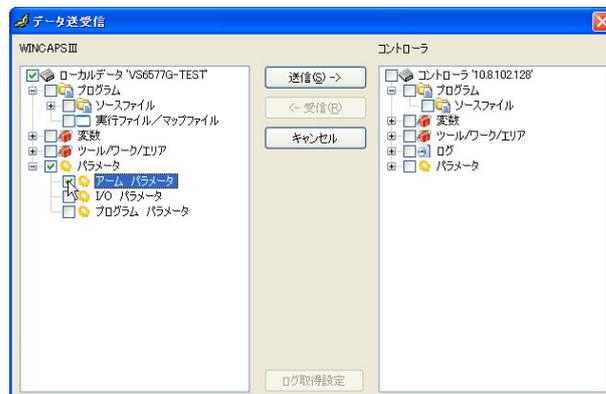
[最適可搬質量設定モード]、[先端負荷質量 (g)]、[負荷重心位置 X (mm)]
[負荷重心位置 Y (mm)]、[負荷重心位置 Z (mm)]

但し、4軸ロボットの Ver. 1.9 以降は [負荷イナーシャ (kgcm²)]

各パラメータ値の設定ができれば、ロボットコントローラにデータを転送します。
まず、ティーチングペンダントの[MOTOR]で、モータ電源を OFF にします。

[通信 (N)]-[データ送受信 (T)]を選択します。

[パラメータ]-[アーム パラメータ]にチェックを入れ、[送信 (S)->]をクリックします。



16.4 最適可搬質量初期化設定の設定方法 [V1.4 以降]

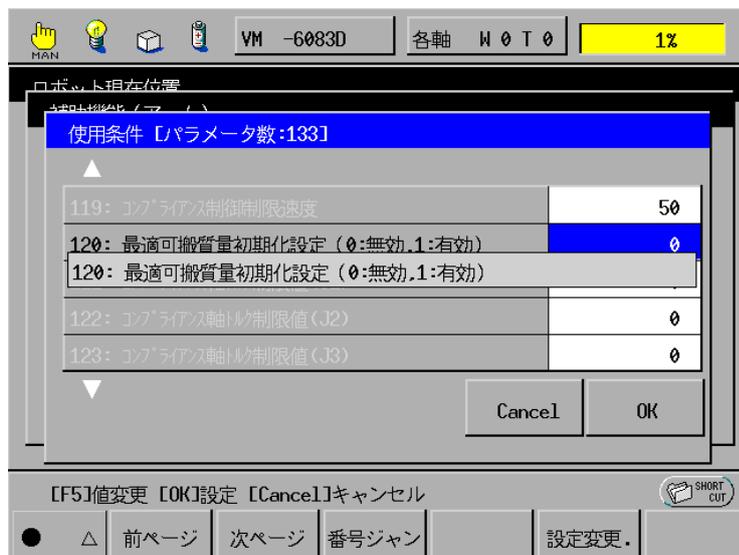
コントローラ電源 ON 時に[7:最適可搬質量設定モード]パラメータをモード0に初期化したい場合や現在設定されている値を保持したい場合の設定を説明します。

設定値	設定値の意味
0	コントローラ電源投入時に最適可搬質量設定モードをモード0に初期化する。(工場出荷時)
1	コントローラ電源を投入しても最適可搬質量設定モードを初期化しない。(現在の値を保持する)

16.4.1 ティーチングペンダントによる設定

操作経路：基本画面— [F2 アーム] — [F6 補助機能] — [F7 使用条件]

ティーチングペンダントを操作して、上記の操作経路を経ると、[使用条件 (パラメータ数:)] 画面が現れます。



この [使用条件 (パラメータ数:)] 画面で、「120:最適可搬質量初期化設定」の項目を選択し [F5 設定変更.] を押すと、テンキー形式の [パラメータ変更画面が表示され、各パラメータ値を変更できます。

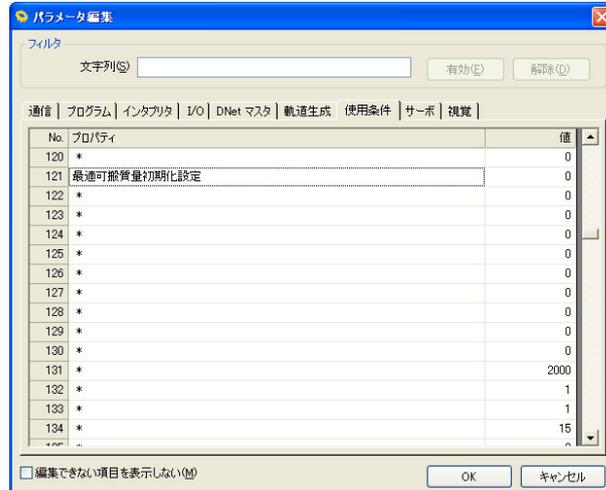
0 : 無効→コントローラ電源投入時初期化 (工場出荷時)

1 : 有効→コントローラ電源を投入しても初期化しない (現在の値を保持する)

16.4.2 WINCAPSⅢによる設定

WINCAPSⅢによって、最適可搬質量初期化設定の設定を行う方法について説明します。

[プロジェクト (P)] -[パラメータ (T)]-[使用条件] タブを選択します。



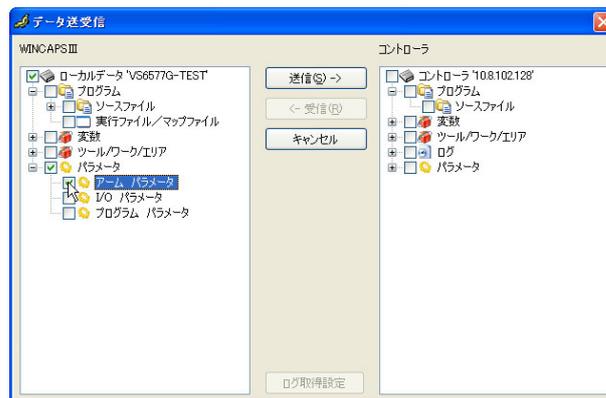
最適可搬質量初期化設定の設定項目をダブルクリックすると、パラメータ値を入力変更できます。

各パラメータ値の設定ができたら、ロボットコントローラにデータを転送します。

まず、ティーチングペンダントの [MOTOR] で、モータ電源をOFFにします。

[通信 (N)]-[データ送受信 (T)]を選択します。

[パラメータ]-[アーム パラメータ]にチェックを入れ、[送信 (S)->]をクリックします。



第 17 章 プログラム言語（ロボット制御文）

17.1 ロボット動作の種類

17.1.1 絶対動作と相対動作

絶対動作

ティーチングされた動作位置へ移動する動作を、絶対動作といいます。

絶対動作は、その直前の動作に影響されることなく、必ずティーチングされた位置へ移動します。

絶対動作を行なうコマンドは、次のとおりです。

APPROACH、MOVE、GOHOME、DRIVEA

相対動作

現在位置から、ティーチングされた移動量だけ移動する動作のことを、相対動作といいます。

相対動作は、その直前の動作コマンドを実行した結果の現在位置を基準とするので、直前の動作コマンドの影響を受けることになります。

相対動作を行なうコマンドは、次のとおりです。

DEPART、DRAW、DRIVE、ROTATE、ROTATEH

17.1.2 補間制御

ロボットアームの先端が移動するとき、その経路は一通りではありません。各軸の動きが相まって、いろいろな経路をつくることができます。直線や円弧になるように制御することもできます。以下に、動作経路の種類に応じた制御方法について説明します。

次のコマンドで、補間方法（PTP 制御・CP 制御・円弧補間制御）を指定します。

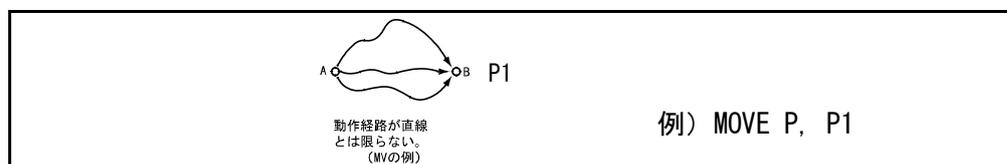
補間方法を指定するコマンド：

APPROACH、DEPART、DRAW、MOVE

PTP 制御

PTP (Point to Point) とは、点から点への移動を意味します。移動する経路はロボットの姿勢に依存し、直線運動をするとは限りません。

動作制御コマンドで補間方法を指定するときに、「P」の指定をします。

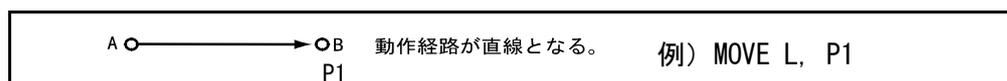


PTP 動作目標位置に P 型、T 型変数を指定した場合、ロボット形態を指定すると、指定したロボット形態となるよう移動します。ロボット形態を指定しない場合は、現在のロボット形態と同一の形態になります。

CP 制御

CP 制御は、動作目標位置に達する経路を、直線になるように補間制御します。

動作制御コマンドで補間方法を指定するときに、「L」の指定をします。



- ・原則として、現在のロボット形態と異なる形態へは移動できません。異なる形態を指定した場合は、「607F ロボット形態不一致」エラーが発生します。ただし、移動可能な場合は、エラーは発生しません。
- ・ロボット形態は、現在の姿勢に近い形態が選択されます。したがって、P 型、T 型変数でロボット形態を指定しても、指定した形態になるとは限りません。指定した形態と異なる場合は、「601C 形態を変更して下さい」のワーニングが発生します。
- ・プログラムでの最初の動作命令を CP 制御で行なうと、ロボットの位置によって動作できない場合があります。プログラムの最初の動作命令には PTP 制御を使用することをおすすめします。

円弧補間制御

円弧補間制御は、動作目標位置に達する経路を、円弧を描くように補間制御します。

動作制御コマンドで補間方法を指定するときに、「C」の指定をします。



- ・CP 制御時と同様、原則として、現在のロボット形態と異なる形態へは移動できません。異なる形態を指定した場合は、「607F ロボット形態不一致」エラーが発生します。ただし、移動可能な場合は、エラーは発生しません。
- ・ロボット形態は、現在の姿勢に近い形態が選択されます。したがって、P 型、T 型変数でロボット形態を指定しても、指定した形態になるとは限りません。指定した形態と異なる場合は、「601C 形態を変更して下さい」のワーニングが発生します。
- ・プログラムでの最初の動作命令を円弧補間制御で行なうと、ロボットの位置によって動作できない場合があります。プログラムの最初の動作命令には PTP 制御を使用することをおすすめします。

17.2 ロボット制御コマンド

17.2.1 DRIVEA

機能 各軸の絶対動作を行ないます。

書式 DRIVEA □ [<@パス開始変位> □] (<軸番号> , <軸座標>) [, (<軸番号> , <軸座標>) …] [, <動作オプション>] [, NEXT]

説明 <軸番号>で指定した軸を、<軸座標>で指定した角度(DEG)へ移動させます。

同じ軸を複数指定した場合は、後に指定したものが有効になります。

<@パス開始変位>には@0、@P(@1~@255)、@Eがあります。

パス開始変位	意味
@0	エンド動作で動きます。(省略すると@0として処理します)
@P (または @1~@255)	パス動作で動きます。注意：数値指定では目標位置を中心とした球の半径(mm指定)に動作指令値が入ると次の制御へ移ります。パス開始の目安値で保証値ではありません。
@E	エンコーダ値によって目標位置への到達を確認します。

<動作オプション>にはSPEED、ACCEL、DECELがあります。

動作オプション	意味
SPEED (またはSと表記)	移動速度を指定します。
ACCEL	加速度を指定します。
DECEL	減速度を指定します。

<NEXT オプション>を付けると、ロボットの動作完了を待たずに次の非動作命令に続きます。但し、以下の命令はロボットの動作完了(パス開始)まで実行待ちになります。ロボット動作命令(CHANGETOOL、CHANGEWORK、SPEED、JSPEED、ACCEL、JACCEL、DECEL、JDECEL)、最適可搬質量設定ライブラリ(aspACLD、aspChange)、アーム動作ライブラリ(mvSetPulseWidthなど)

また、動作オプションと併用した場合NEXTオプションは無効になります。<NEXT オプション>を付けステップ停止を実行すると、次の動作命令実行待ちの場合には、その動作が終了後に停止します。停止までの移動が長くなりますので注意してください。

注意：ティーチチェックモードでは、NEXTオプションは無効になります。

用例

```
例1: DEFINT li1, li2, li3
      DEFSNG lf1, lf2, lf3
      DRIVEA (li1, 30)           ' li1 の値の軸を 30 度(deg) の位置へ移動します。
      DRIVEA (li1, lf1)        ' li1 の値の軸を lf1 の値の位置へ移動します。
      DRIVEA @P (li1, 0.78RAD), (li2, lf2), (li3, lf3) ' li1 の値の軸を 0.78(rad)、
                                                                    li2 を lf2 の値、li3 を lf3
                                                                    の値の位置へ移動します。
```

17.2.2 DRIVE

機能

各軸の相対動作を行ないます。

書式

DRIVE□ [<@パス開始変位>□] (<軸番号>, <相対移動量>)
 [, (<軸番号>, <相対移動量>) …] [, <動作オプション>] [, NEXT]

説明

<軸番号>で指定した軸を、<相対移動量>で指定した角度(DEG)へ移動させます。<相対移動量>が正の場合は、指定軸を+方向へ、負の場合は-方向へ動作させます。

同じ軸を複数指定した場合は、後に指定したものが有効になります。

<@パス開始変位>には、@0、@P (@1~@255)、@E があります。

パス開始変位	意味
@0	エンド動作で動きます。(省略すると@0として処理します)
@P (または @1~@255)	パス動作で動きます。注意：数値指定では目標位置を中心とした球の半径(mm指定)に動作指令値が入ると次の制御へ移ります。パス開始の目安値で保証値ではありません。
@E	エンコーダ値によって目標位置への到達を確認します。

<動作オプション>には@0、@P (@1~@255)、@E があります。

動作オプション	意味
SPEED (またはSと表記)	移動速度を指定します。
ACCEL	加速度を指定します。
DECEL	減速度を指定します。

<NEXT オプション>を付けると、ロボットの動作完了を待たずに次の非動作命令に続きます。但し、以下の命令はロボットの動作完了(パス開始)まで実行待ちになります。ロボット動作命令(CHANGETOOL、CHANGEWORK、SPEED、JSPEED、ACCEL、JACCEL、DECEL、JDECEL)、最適可搬質量設定ライブラリ(aspACLD、aspChange)、アーム動作ライブラリ(mvSetPulseWidthなど)

また、動作オプションと併用した場合NEXTオプションは無効になります。<NEXT オプション>を付けステップ停止を実行すると、次の動作命令実行待ちの場合には、その動作が終了後に停止します。停止までの移動が長くなりますので注意してください。

注意：ティーチチェックモードでは、NEXTオプションは無効になります。

用例

```
例1: DEFINT li1, li2, li3
      DEFSNG lf1, lf2, lf3
      DRIVE (li1, 30)           ' 現在位置から li1 を 30 度(deg)移動します。
      DRIVE (li1, lf1)        ' 現在位置から li1 の値の軸を lf1 値分移動します。
      DRIVE (li1, 0.78RAD), (li2, lf2), (li3, lf3) ' 現在位置から li1 を 0.78(rad)、' li2 を lf2
                                                                    ' が示す値分、li3 を lf3 の値分移動します。
```

17.2.3 DRAW

機能

ワーク作業座標系指定で相対動作を行いません。

書式

DRAW □ <補間方法>, [<@パス開始変位量>] <並進移動量>[, <動作オプション>] [, NEXT]

説明

現在位置から<並進移動量>分移動します。

<補間方法>には P (PTP と表記も可)、L どちらかの選択ができます。

補間方法	意味
P (またはPTPと表記)	PTP制御で移動します。
L	CP制御で移動します。

<@パス開始変位量>には@0、@P (@1~@255)、@E があります。

パス開始変位	意味
@0	エンド動作で動きます。(省略すると@0として処理します)
@P (または @1~@255)	パス動作で動きます。注意：数値指定では目標位置を中心とした球の半径 (mm指定) に動作指令値が入ると次の制御へ移ります。パス開始の目安値で保証値ではありません。
@E	エンコーダ値によって目標位置への到達を確認します。

<動作オプション>には SPEED、ACCEL、DECEL があります。

動作オプション	意味
SPEED (またはSと表記)	移動速度を指定します。
ACCEL	加速度を指定します。
DECEL	減速度を指定します。

<NEXT オプション>を付けると、ロボットの動作完了を待たずに次の非動作命令に続きます。但し、以下の命令はロボットの動作完了 (パス開始) まで実行待ちになります。ロボット動作命令 (CHANGETOOL、CHANGEWORK、SPEED、JSPEED、ACCEL、JACCEL、DECEL、JDECEL)、最適可搬質量設定ライブラリ (aspACLD、aspChange)、アーム動作ライブラリ (mvSetPulseWidth など)

また、動作オプションと併用した場合 NEXT オプションは無効になります。<NEXT オプション>を付けステップ停止を実行すると、次の動作命令実行待ちの場合には、その動作が終了後に停止します。停止までの移動が長くなりますので注意してください。

注意：ティーチチェックモードでは、NEXTオプションは無効となります。

備考 DRAW は、MOVE によって書き換えることができます。

例：DRAW L, (50, 10, 50) ' MOVE L, P0+(50, 10, 50) と同じです

用例

```
DEFVEC lv1
DRAW L, (50, 10, 50) ' 現在位置から (X=50, Y=10, Z=50) 離れた位置へ (CP 制御) 移動します。
DRAW L, lv1, SPEED=90 ' 現在位置から lv1 離れた位置へ (CP 制御, S=90) 移動します。
```

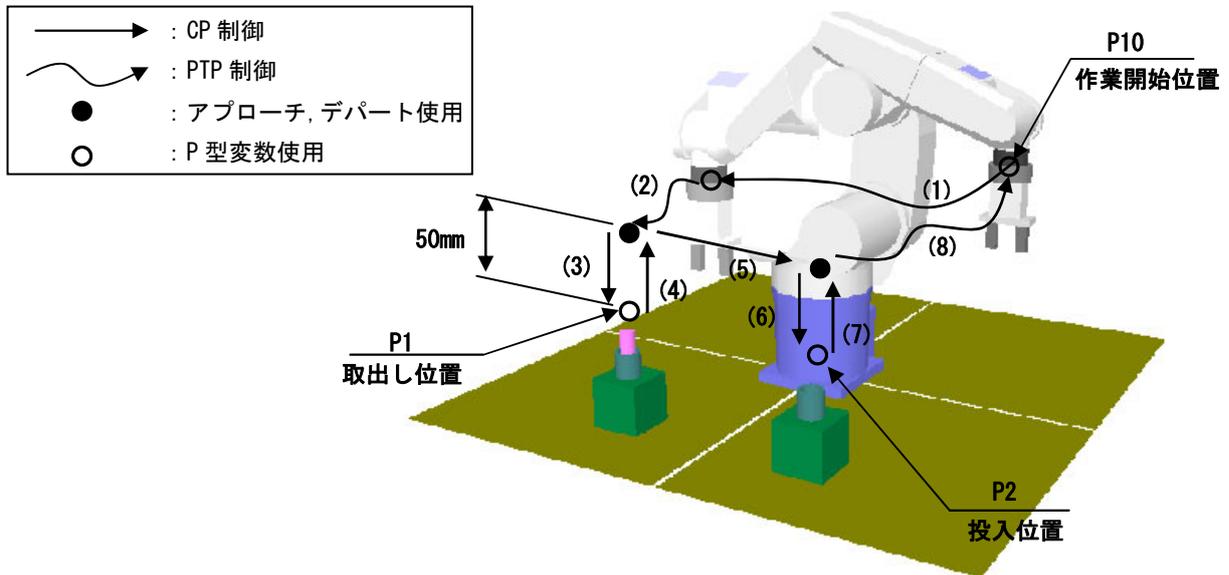
注意事項

DRAW 動作位置の形態は、動作開始時の形態になります。

17.3 練習問題 ロボット制御コマンド

練習問題 1

ロボット制御文を使いワーク取出し・投入位置へ動作させるプログラムを作成してください。



■ 動作仕様

- ① (1)の動作は、J1軸のみを0度の位置へ動作させるコマンドを使用してください。
- ② アプローチ、デパートポイントではパス動作を使用してください。
- ③ P1およびP2ポイントへの下降動作時のスピードは20%にしてください。
- ④ P1およびP2ポイントへの動作は、エンコーダ値確認動作としてください。

プログラム	動作説明
' TITLE "練習プログラム1"	' プログラムタイトル
PROGRAM PR01	' プログラム名の宣言
TAKEARM	' アーム制御権取得
SPEED 100	' 内部スピード100%
_____	' (1) : J1軸を0度の位置へ移動
_____	' (2) : P1のハンド方向 上空50mmへ移動
_____	' (3) : P1へ移動
_____	' (4) : P1のハンド方向 上空50mmへ移動
_____	' (5) : P2のハンド方向 上空50mmへ移動
_____	' (6) : P2へ移動
_____	' (7) : P2のハンド方向 上空50mmへ移動
MOVE P,@0 P10	' (8) : P10へ移動
END	' プログラムの停止

第 18 章 プログラミング言語（フロー制御文）

18.1 フロー制御文の種類

プログラムの各文の実行順序を制御するために、フロー制御文を使います。
フロー制御文を使用することで、高度なプログラムを作成することができます。
フロー制御コマンドには、下記の 4 つに大別されます。

①呼び出し

- CALL
- GOSUB

②無条件分岐

- GOTO

③条件分岐

- IF~ENDIF
- SELECT CASE~END SELECT

④繰返し

- FOR~NEXT
- DO~LOOP
(DO WHILE~LOOP , DO~LOOP WHILE , DO UNTIL~LOOP , DO~LOOP UNTIL)

18.3 無条件分岐コマンド

18.3.1 GOTO

機能 プログラムの分岐を無条件で行います。

書式 GOTO □ <*ラベル名> (または GO □ TO □ <*ラベル名>)

説明 指定したラベル先へジャンプして、ジャンプ先から実行を継続させます。

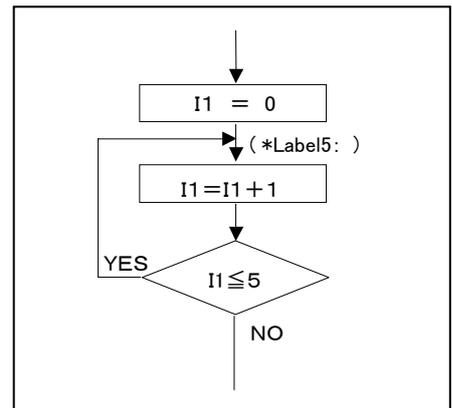
ラベル名使用時の規制事項

- ・ラベルの名前はアスタリスク(*)で始まります。
- ・ラベルの名前の2文字目は、任意のアルファベットでなくてはなりません。
- ・ラベルの名前の3文字目以降は、英字と数字の任意の組合せが使えます。
- ・予約語をラベル名に使用することはできません。
- ・ラベルを参照できる範囲は、そのラベルが存在するプログラムの中だけです。
- ・参照されるラベル名の最後の文字はコロン(:)でなければなりません。

用例

例) I1 の値が 5 以下の場合、指定ラベル先へジャンプさせます。

```
I1 = 0
*Label5:
I1 = I1 + 1
IF I1 <= 5 THEN      ’ I1 が 5 以下の場合
    GOTO *Label5     ’ ラベル名<*Label5:> へジャンプ
ENDIF
```



18.4 条件分岐コマンド

18.4.1 IF ~ ENDIF

機能

IF~END IF 間の条件式の条件判断を行います。

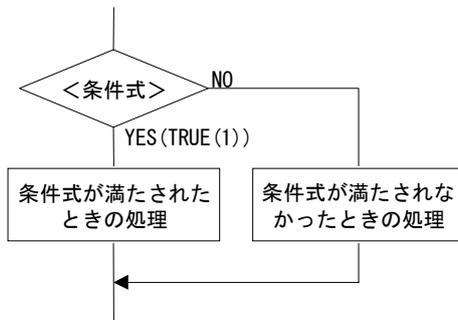
書式

```
IF □<条件式>□ THEN
  :      ' <条件式>が満たされたときの処理
[ELSE]
  :      ' <条件式>が満たされなかったときの処理
ENDIF(または END □IF)
```

説明

IF 文の<条件式>が真 (TRUE(1)) であれば IF~ELSE 間の文を実行し、そうでなければ ELSE~ENDIF 間の文を実行します。

※<条件式>には下記の演算式が使用できます



関係演算子	演算内容
=	等しい
=.	ほぼ等しい (近似比較)
<>	等しくない
<	小さい
>	大きい
<=	小さいか等しい
>=	大きいか等しい

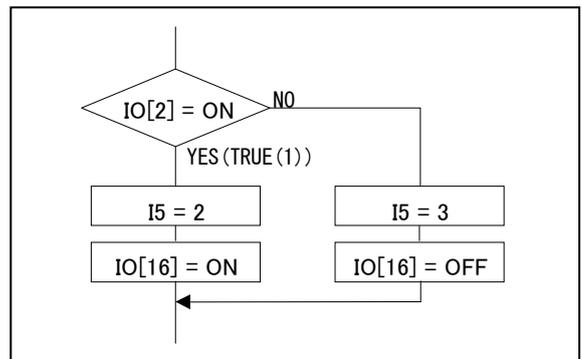
備考

近似比較演算子(=.)の比較精度は、[PRJ 設定]の”近似比較精度”で設定できます。

用例

例) IO[2]がONならば、I5に2を代入しIO[16]をONする。それ以外ならI5に3を代入後にIO[16]をOFFする。

```
IF IO[2] = ON THEN      ' IO[2]がONの場合
  I5 = 2
  SET IO[16]
ELSE                    ' それ以外の場合
  I5 = 3
  RESET IO[16]
ENDIF
```



18.4.2 SELECT CASE

機能 複数条件判断を行います。

書式

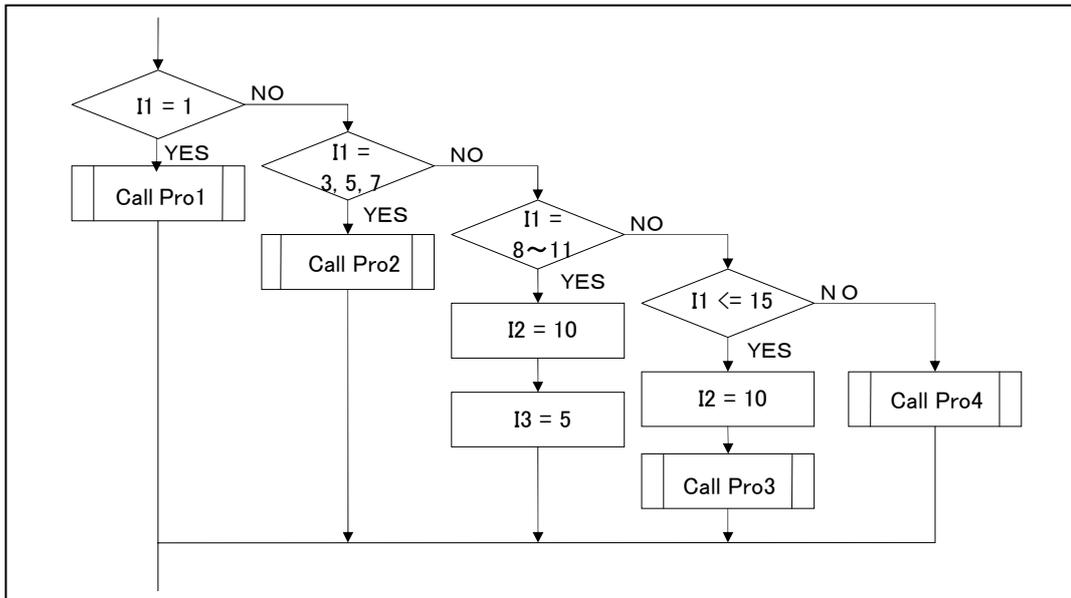
```
SELECT CASE <式>  
  CASE <項目>  
    : ' <式>の値が CASE 文の<項目>に一致したときの処理  
  [CASE ELSE]  
    : ' すべての CASE 文の<項目>に一致しなかったときの処理  
END SELECT
```

説明

SELECT CASE 文では、SELECT CASE の後に<式>を置き、この<式>の値が CASE 文の<項目>に一致する場合、その CASE 以降、次の CASE までの間の文を実行します。すべての CASE 文が一致しなかった場合は、CASE ELSE~END SELECT の間の文を実行します。

用例

例) I1 の値が各項目に一致した時に、それぞれの処理を行ないます。



```
SELECT CASE I1  
  CASE 1 ' I1 が 1 の場合  
    CALL PRO1  
  CASE 3, 5, 7 ' I1 が 3, 5, 7 の場合  
    CALL PRO2  
  CASE 8 to 11 ' I1 が 8~11 の場合 (範囲指定は、" ** to **" を使用)  
    I2 = 10  
    I3 = 5  
  CASE IS <= 15 ' I1 が 15 以下の場合 (比較条件は、IS<比較演算子><比較値(数値)>  
    I2 = 10  
    CALL PRO3  
  CASE ELSE ' 項目が一致しなかった場合  
    CALL PRO4  
END SELECT
```

18.5 繰り返しコマンド

18.5.1 FOR ~ NEXT

機能 FOR~NEXT までの区間中にある一連の命令を繰り返して実行します。

書式 FOR □<変数名> = <初期値> □TO□<最終値>[STEP □<増分>]
: ' FOR 行で指定した条件が満たされているときの処理
NEXT

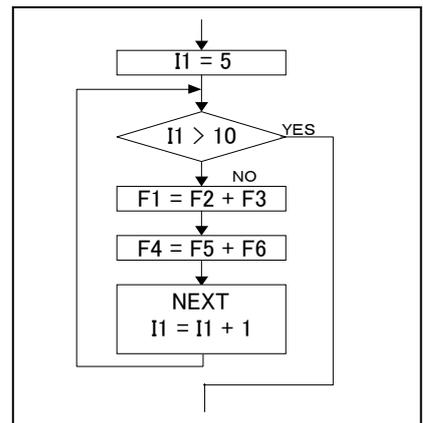
説明 FOR~NEXT までの区間中にある一連の命令を、FOR 行で指定した条件に従って繰り返して実行します。

<初期値>は、<変数名>で指定した変数の初期値を設定します。
<最終値>は、<変数名>で指定した変数の最終値を設定します。
<増分>は、初期値と最終値との間の増分を設定します。STEP を省略すると増分は 1 とみなされます。

用例

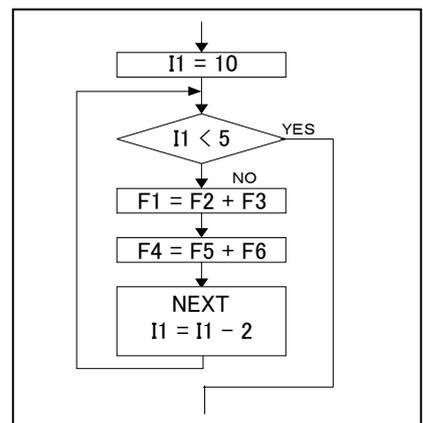
例 1) I1 を 5 から 1 ずつ増分し、I1 が 10 をこえるまで処理を繰り返します。I1 は 5,6...10 となり、11 になった段階で繰り返し処理を終了します。

```
FOR I1 = 5 TO 10      ' I1に初期値5をセット(増分値1)
                      ' I1が10以下の場合処理継続
    F1 = F2 + F3
    F4 = F5 + F6
NEXT                  ' I1をカウント+1
```



例 2) I1 を 10 から 2 ずつ減算し、I1 が 5 より小さくなるまで処理を繰り返します。I1 は 10,8,6 となり、4 になった段階で繰り返し処理を終了します。

```
FOR I1 = 10 TO 5 STEP -2 ' I1に初期値10をセット(減算値2)
                          ' I1が5以上の場合処理継続
    F1 = F2 + F3
    F4 = F5 + F6
NEXT                       ' I1をカウント-2
```



18.5.2 DO ~ LOOP

機能 判定反復（繰返し）を行います。

書式 DO □ [WHILE (または UNTIL) □ <条件式>]
: ' 条件が満たされている間(WHILE の場合)または、
: ' 条件が満たされるまで(UNTIL の場合)DO~LOOP 間の処理を実行します。
LOOP
または
DO
: ' 条件が満たされている間(WHILE の場合)または、
: ' 条件が満たされるまで(UNTIL の場合)DO~LOOP 間の処理を実行します。
LOOP □ [WHILE (または UNTIL) □ <条件式>]

説明 DO WHILE と DO UNTIL は、前判定反復です。LOOP WHILE と LOOP UNTIL は、後判定反復です。WHILE 文は<条件式>が満たされている間(TRUE(1))繰返し、UNTIL 文は<条件式>が満たされるまで繰返します。

- DO ~LOOP : WHILE・UNTIL の省略は可能ですが無限ループします。
- DO WHILE~LOOP : 条件が満たされている間、処理を繰返します。(前判定)
- DO ~LOOP WHILE : 条件が満たされている間、処理を繰返します。(後判定)
- DO UNTIL~LOOP : 条件が満たされるまで処理を繰返します。(前判定)
- DO ~LOOP UNTIL : 条件が満たされるまで処理を繰返します。(後判定)

注意：DO~LOOP文から強制脱出し、DO~LOOP文の次の命令に移るにはEXT DOコマンドを使います。

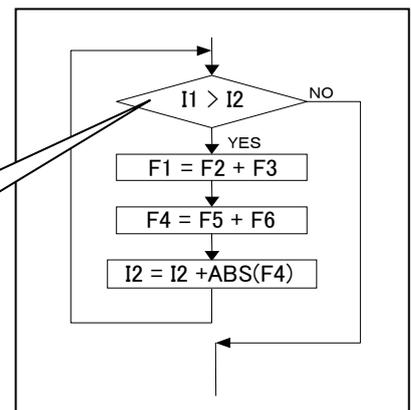
用例

• DO WHILE ~ LOOP (前判定)

例1) $I1 > I2$ である間、処理を繰返します。

```
DO WHILE I1 > I2      ' I1>I2である間DO~LOOP間を繰返し
  F1 = F2 + F3
  F4 = F5 + F6
  I2 = I2 + ABS( F4 )
LOOP
```

ここで条件を判断、
条件が満たされて
いる間、繰返し。

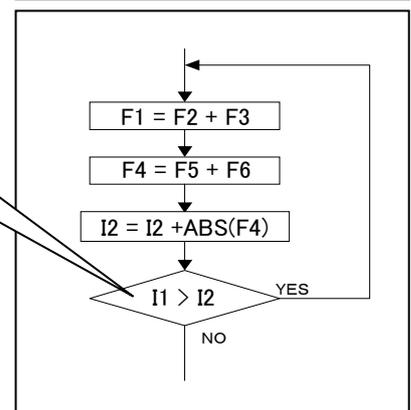


• DO ~ LOOP WHILE (後判定)

例2) $I1 > I2$ である間、処理を繰返します。

```
DO
  F1 = F2 + F3
  F4 = F5 + F6
  I2 = I2 + ABS( F4 )
LOOP WHILE I1 > I2    ' I1>I2である間DO~LOOP間を繰返し
```

ここで条件を判断、
条件が満たされて
いる間、繰返し。

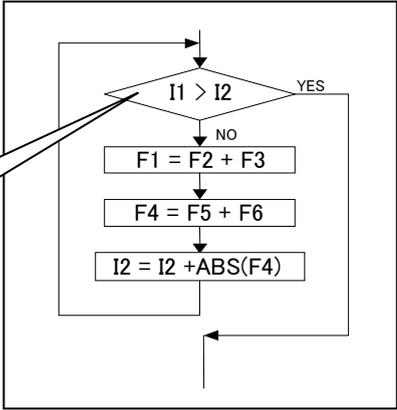


・ DO UNTIL ~ LOOP (前判定)

例3) I1 > I2になるまで、処理を繰り返します。

```
DO UNTIL I1 > I2      ' I1>I2になる迄DO~LOOP間を繰り返し
  F1 = F2 + F3
  F4 = F5 + F6
  I2 = I2 + ABS( F4 )
LOOP
```

ここで条件を判断、
条件が満たされる
まで、繰り返し。

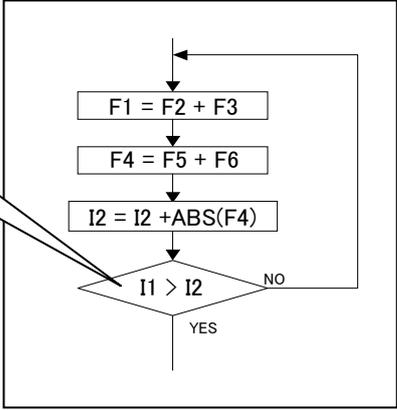


・ DO ~ LOOP UNTIL (後判定)

例4) I1 > I2になるまで、処理を繰り返します。

```
DO
  F1 = F2 + F3
  F4 = F5 + F6
  I2 = I2 + ABS( F4 )
LOOP UNTIL I1 > I2    ' I1>I2になる迄DO~LOOP間を繰り返し
```

ここで条件を判断、
条件が満たされる
まで、繰り返し。

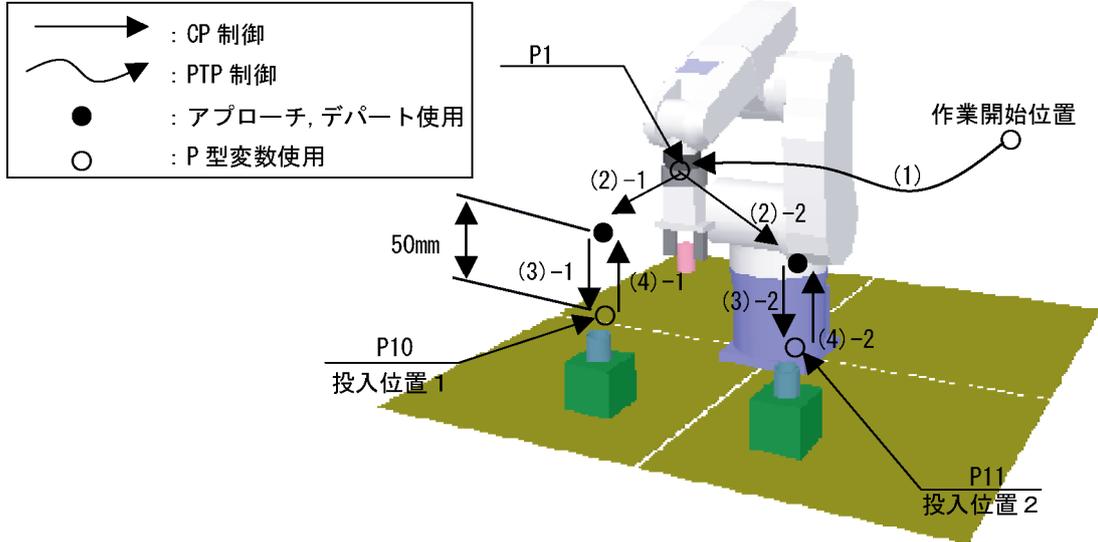


注意：後判定の場合、必ず1度はDO~LOOP間の処理が実行されます。

18.6 練習問題 フロー制御コマンド

練習問題 2

フロー制御文 (IF~ENDIF) を使いワーク投入位置を判断させるプログラムを作成してください。



■ 動作仕様

- ① P1の位置へ移動後、IF文を使いI[5]の値を確認します。I[5]=0なら(2)-1へ、I[5]=1なら(2)-2へ移動し、ワーク投入動作を完成させます。
- ② (3)-1および(3)-2への動作到達位置でハンド開のプログラム(HAND_OPEN)を呼び出します。

プログラム	動作説明
' TITLE "練習プログラム2"	' プログラムタイトル
PROGRAM PR02	' プログラム名の宣言
TAKEARM	' アーム制御権取得
SPEED 100	' 内部スピード100%
MOVE P, P1	' (1) : P1へ移動
-----	' (2) : IF文 I5=0なら次の命令へ
APPROACH L, P10 50	' (2)-1 : P10のハンド方向 上空50mmへ移動
MOVE L, P10	' (3)-1 : P10へ移動
-----	' (2) : IF文 I5=0でない場合、次の命令へ
APPROACH L, P11 50	' (2)-2 : P11のハンド方向 上空50mmへ移動
MOVE L, P11	' (3)-2 : P11へ移動
-----	' (2) : IF文の終了
-----	' HAND_OPENプログラム呼び出し
DEPART L, 50	' (4)-1, (4)-2 : P10またはP11のハンド方向 上空50mmへ移動
END	' プログラムの停止

第 19 章 プログラム言語（入出力制御文）

19.1 時間制御コマンド

指定した時間だけロボットを待たせる命令を紹介します。

19.1.1 DELAY

機能 指定した時間の間、ロボットのプログラム進行を停止します。

書式 DELAY □ <遅延時間>

説明 <遅延時間>で指定された時間が経過するまで待ちます。遅延時間（単位：ms）1 秒は、1000 と入力します。

用例

DELAY 300 ’ 300ms (0.3s) の時間が経過するまで待ちます。
DELAY I15 ’ I15 の値の時間が経過するまで待ちます。

19.1.2 WAIT

機能 条件プログラム停止を行います。

書式 WAIT □ <条件式> [, <タイムアウト時間>] [, <格納変数>]

説明 指定した<条件式>が成立するまでプログラムの進行を停止します。
<タイムアウト時間>を設定すると、指定時間を経過した後に WAIT 文の実行を終了し次のコマンドに制御が移ります。これにより、無限停止を防止できます。
<タイムアウト時間>の単位は ms です。
<格納変数>を設定すると条件式が成立して、WAIT コマンドを抜けたとき指定した格納変数に TRUE (1) が入り、タイムアウトで抜けたときは FALSE (0) が入ります。

用例

DEFINT li1
WAIT li1 = 1 ’ li1=1 が成立するまで待ちます。
WAIT IO[10] = ON ’ IO[10] が ON するまで待ちます。
WAIT IO[5] = 0 , 2000 ’ IO[5] が OFF するまで待ち、2 秒経過しても OFF しない場合次の文へ移ります。
WAIT I3 = 5 , 1000, I4 ’ I3=5 が成立するまで待ち、成立すれば I4=1 にし、1 秒経過しても成立しない場合、I4=0 にして次の文へ移ります。

19.2 I/Oポート制御

ここでは、ロボットのチャック動作を例に出力命令を紹介します。出力命令は出力ポートを ON する命令に SET、OFF する命令に RESET を使用します。

19.2.1 SET

機能 I/Oポートを ON にします。

書式 SET □ <I/O 変数> [, <出力時間>]

説明 <I/O 変数>で指定されたポートを ON にします。ポートの指定は、ポート番号か I/O 変数を使用します。
<出力時間>を指定するとパルス出力します。(出力時間の単位は ms です。)
<出力時間>を指定すると、その時間が経過するまで次の命令へは進みません。また、出力時間の設定値は最低出力時間であり、実際の出力時間は、タスクのプライオリティなどに影響され変化します。

用例

SET IO[64] ’ (または、SET IO64)ポート番号 64 を ON にします。
SET IO[128], 50 ’ (または、SET IO128, 50)ポート番号 128 を ON し、50ms 後 OFF して次の命令へ移ります。

19.2.2 RESET

機能 I/Oポートを OFF にします。

書式 RESET □ <I/O 変数>

説明 <I/O 変数>で指定されたポートを OFF にします。ポートの指定は、ポート番号か I/O 変数を使用します。

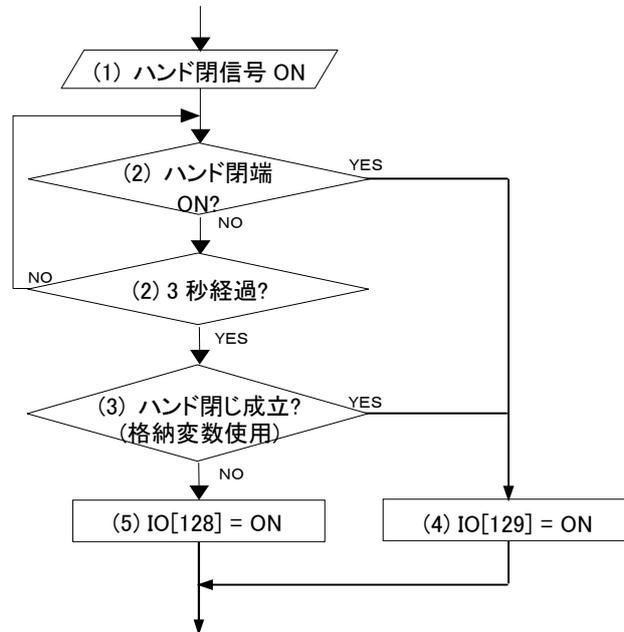
用例

RESET IO[64] ’ (または、SET IO64)ポート番号 64 を OFF にします。

19.3 練習問題 I/Oポート制御コマンド

練習問題 3

I/Oポート制御文を使い下記のフローチャートの制御をおこなう動作プログラムを作成してください。



■ 動作仕様

- ①ハンド閉信号はIO[64]を使用する
- ②ハンド閉端信号はIO[48]を使用し、WAIT文にて信号入力待ちをおこなう
- ③ハンド閉じの成立判定はWAIT文の格納変数(I[20])を使用する

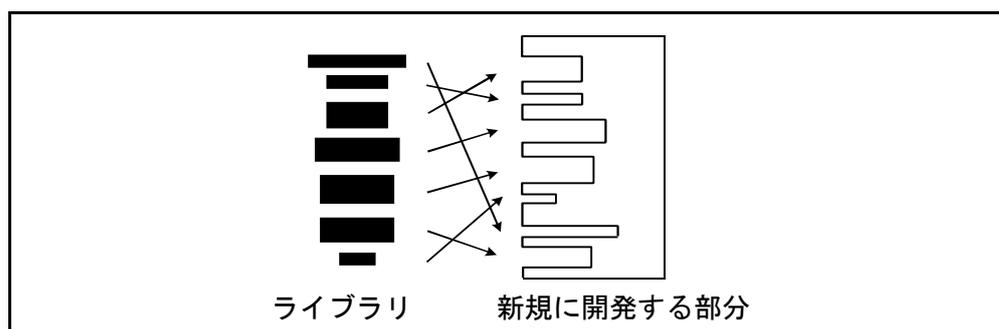
プログラム	動作説明
' TITLE "練習プログラム 3"	' プログラムタイトル
PROGRAM PRO3	' プログラム名の宣言
_____	' (1) : ハンド閉信号IO[64]をON
_____	' (2) : IO[64]入力待ち3秒、格納変数I20使用
IF I[20] = 1 THEN	' (3) : 格納変数I20=1(成立)なら次の命令へ
_____	' (4) : IO[129]をON
ELSE	' (3) : 格納変数I20=1(成立)以外なら次の命令へ
_____	' (5) : IO[128]をON
ENDIF	' (3) : IF文の終了
END	' プログラムの停止

第20章 プログラム言語（ライブラリ編）

20.1 ライブラリプログラムの活用

20.1.1 ライブラリプログラムとは

汎用性のあるプログラムを、部品のように集めて、用途に合わせて利用するためのものを、プログラムのライブラリと呼びます。PAC 言語では、プログラムの中から別のプログラムを呼び出すことができるので、ライブラリの中にあるプログラムを利用したり、作ったプログラムをライブラリに登録して、より効率的なプログラム開発を行なうことができます。



ライブラリを用いたプログラム開発のイメージ

20.1.2 プログラムバンク

WINCAPSⅢには、ライブラリを使うためのプログラムバンクが用意されています。プログラムバンクは、プログラムをライブラリとして登録したり、あるいは登録されているプログラムをプロジェクトに追加するためのツールです。

プログラムバンクに登録されているライブラリを使用するには、ライブラリの追加（インポート）が必要です。ライブラリの追加方法については、「20.1.4 ライブラリの追加（インポート）方法」を参照してください。

20.1.3 ライブラリの分類

標準プログラムライブラリは、次の6つのクラスに分類されて提供されています。

標準プログラムライブラリのクラス

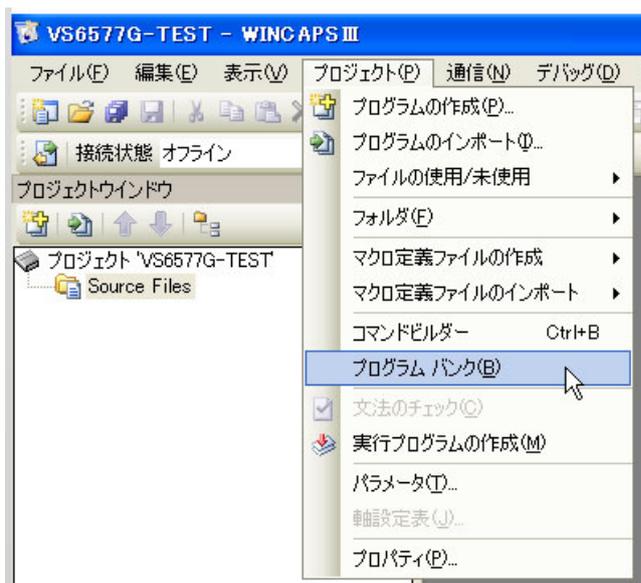
	クラス名	内容
1	従来言語	従来言語のコマンドと同等の機能を提供します。
2	パレタイジング	パレタイジング機能を提供します。
3	ツール操作	ツール操作関連の機能を提供します。
4	入出力	DI0、RS232C入出力関連の機能を提供します。
5	アーム動作	上記以外のアーム動作関連の機能を提供します。
6	視覚	視覚動作関連の機能を提供します。
7	Ver. 1.2互換	コントローラのソフトウェアバージョンがVer. 1.2*以前で使用可能なライブラリを提供します。対象となるのは以下の3種類です。Ver1.2*で以下の3種類のプログラムを上記クラスから使用するとコンパイル異常が発生します。この3種類以外は上記1～6のライブラリを使用してください。 ndVcom、pltMove、pltMove0

20.1.4 ライブラリの追加（インポート）方法

プログラム「dioSetAndWait」を、プログラムバンクからプログラムプロジェクトに追加してみましょう。

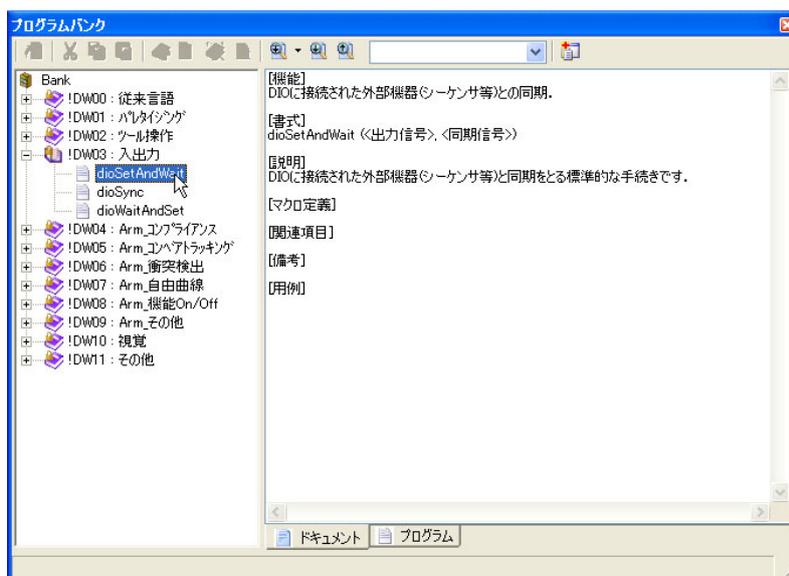
ステップ 1 WINCAPS IIIでプロジェクトを開きます。

ステップ 2 [プロジェクト (P)]-[プログラムバンク (B)]を選択します。



ステップ 3 クラスを選択します。

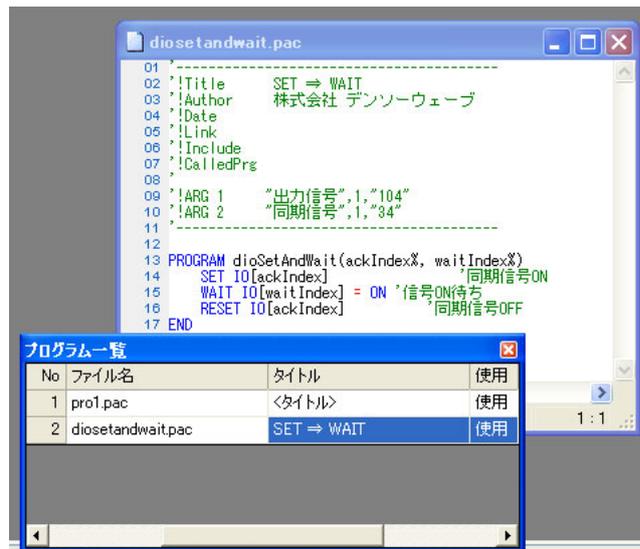
「dioSetAndWait」は[!DW03:入出力]に入っていますので[!DW03:入出力]を開き、[dioSetAndWait]を選択します。



[ドキュメント]タブと[プログラム]タブで内容を確認します。

ステップ 4

[プログラムに追加] ボタンを押すと表示されているプログラムがプロジェクト内に登録されます。



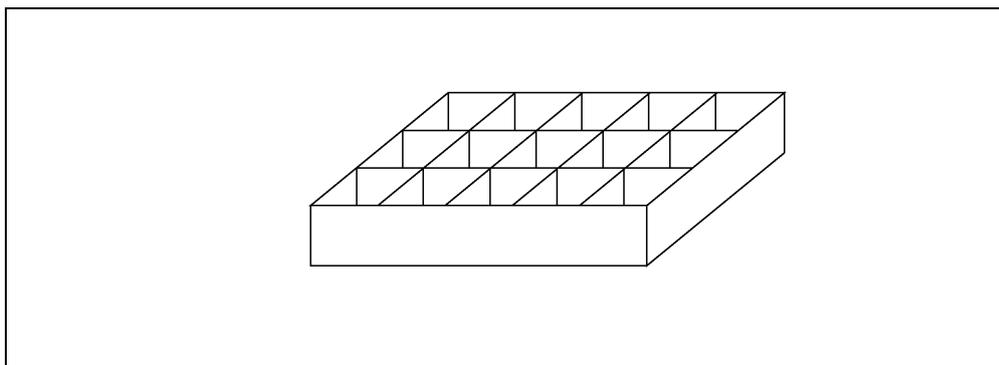
20.2 パレタイジングライブラリの活用

20.2.1 パレタイジングの説明

下図のような仕切りのあるパレットに、部品などの投入や取り出しを順次行なうことを、パレタイジングといいます。

パレタイジング用のライブラリプログラムは、パレットの仕切りの数と四隅の位置をティーチングするだけで、パレタイジング動作を行なえるように、作られています。

パレタイジングのプログラムは、動作のために呼び出すごとに、パレットの取り出し位置を順番に変えていきます。



仕切りのあるパレット

20.2.2 簡易パレタイジングライブラリ

パレタイジング動作を行うには、プログラムバンクより、xdGetPalt のライブラリをあらかじめプロジェクトに追加（インポート）しておく必要があります。

No	ファイル名	タイトル	使用
1	pro1.pac	<タイトル>	使用
2	xdgetpalt.pac	簡易パレタイジング(4軸/6軸共)	使用

簡易パレタイジングライブラリ (xdGetPalt) がインポートされた状態。

パレタイジングのパラメータ

パレタイジングを行なうために必要なパラメータを、下図、-1)、-2)、-3)、と表-4)に示します。

PAC 言語では、これらのパラメータを変数の値として保持しています。

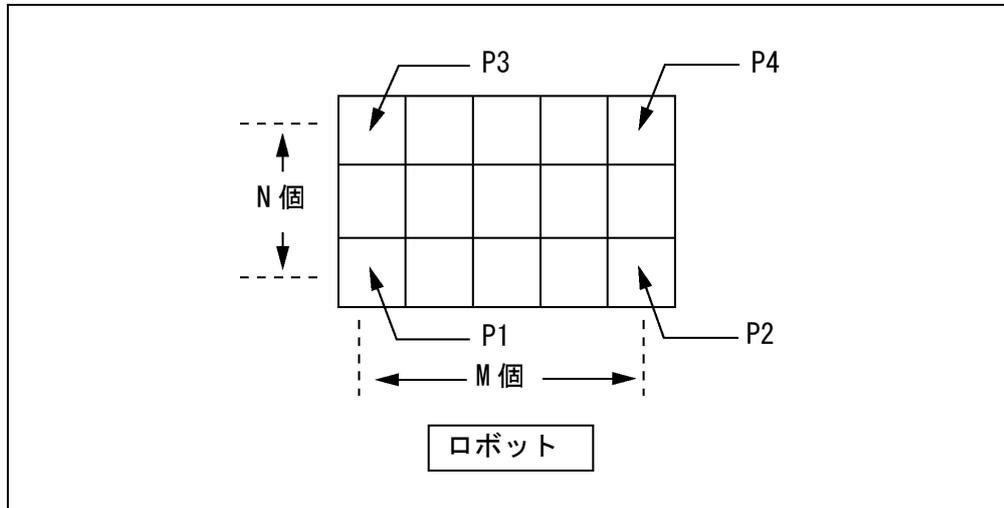


図-1) パレットの上視図

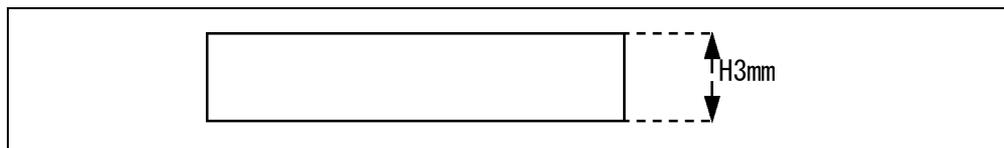


図-2) パレットの横視図

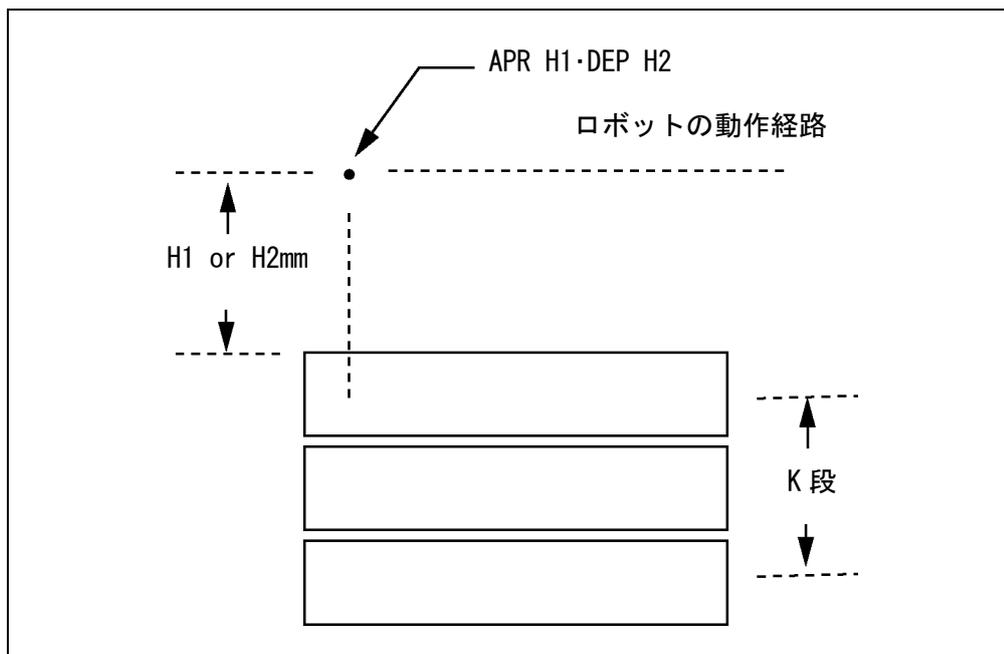


図-3) パレットの段積図

表-4) パレタイジングに必要なパラメータ

記号	名称	意味	単位
	パレタイジング番号	パレタイジングのインデックス番号	なし (整数)
N	横分割数	P1からP3方向への分割数	個 (整数)
M	縦分割数	P1からP2方向への分割数	個 (整数)
K	段積数	パレットの段積数	個 (整数)
H1	アプローチ長	ロボットがパレットに近づくときのアプローチ長	mm (単精度実数)
H2	デパート長	ロボットがパレットから離れるときのデパート長	mm (単精度実数)
H3	パレット高さ	パレットの1段の高さ	mm (単精度実数)
<p>ただし、H1とH2は、次の条件を満たさなければなりません。</p> $H1 > \{H3 \times (K-1)\} + 5$ $H2 > \{H3 \times (K-1)\} + 5$			
P1 P2 P3 P4	<p>図16-3に示すパレット四隅の点。各点の相対位置関係は、入れ替えができません。またロボットの姿勢は、P1の位置をティーチングしたときの姿勢が、すべての点で維持されます。</p>		

N 横分割数

パレットの横方向の分割数を示します。

図 16-3 では 3 列になっています。

M 縦分割数

パレットの縦方向の分割数を示します。

図 16-3 では 5 列になっています。

K 段積数

パレットの段積数を示します。

図 16-5 では 3 段になっています。

H1 アプローチ長

ロボットがパレットに近づくときのアプローチ長を示します。

パレタイジングプログラムを呼び出すたびに、同じアプローチ長を使用します。

H2 デパート長

ロボットがパレットから離れるときのデパート長を示します。

パレタイジングプログラムを呼び出すたびに、同じデパート長を使用します。

H3 パレット高さ

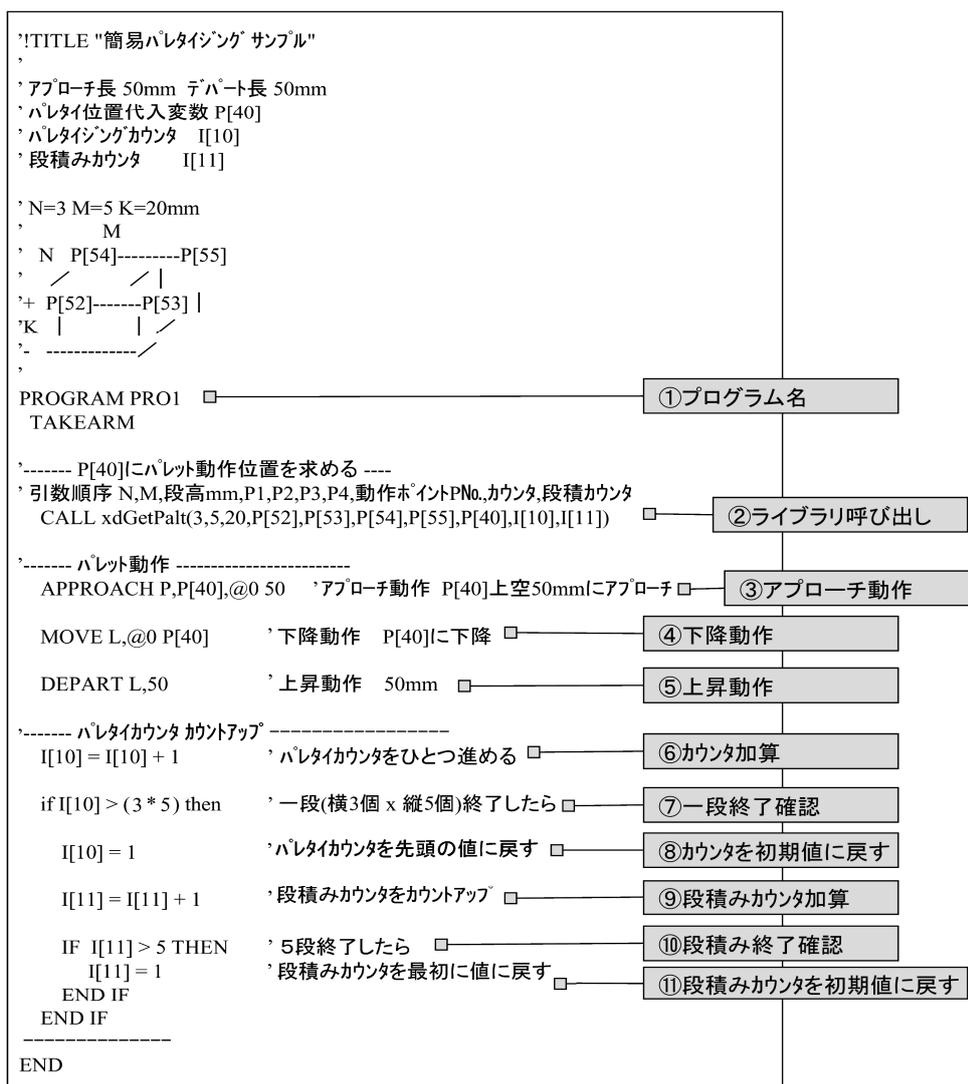
パレットの 1 段の高さを示します。

パレットが次第に積み上がっていく場合には、プラスの値を入力します。

パレットが次第に減っていく場合には、マイナスの値を入力します。

パレットの段数に変化しない場合には、0 を入力します。

20.2.3 簡易パレタイジングプログラム「PR01」



簡易パレタイジングプログラム「PR01」の説明

パレタイジングカウンタや段積みカウンタは、ユーザが加算処理を行なう必要があります。

PR01 で使用する変数

- ・パレタイジング目標位置代入変数（P型：例ではP40を使用）
- ・パレタイジングカウンタ変数（I型：例ではI10を使用）
- ・段積みカウンタ変数（I型：例ではI11を使用）
- ・パレット4隅位置変数 P1～P4（P型：例ではP52～P55を使用）

PR01 動作前にしておくこと

- ーパレタイジングカウンタ I10, 段積みカウンタ I11 に初期値である1を代入しておく必要があります。
- ーパレット4隅位置変数 P1～P4 にパレットの四隅の位置をティーチングしておく必要があります。

以下、プログラム PR01 の解説を各パートに分けて行ないます。

①プログラム名

```
'  
PROGRAM PRO1  
TAKEARM
```

← プログラムの名前を変更してください

②ライブラリ呼び出し

```
'----- P[40]にパレット動作位置を求める ----  
'引数順序 N,M,段高mm,P1,P2,P3,P4,動作ポイントPNo.,カウンタ,段積カウンタ  
CALL xdGetPalt(3,5,20,P[52],P[53],P[54],P[55],P[40],I[10],I[11])
```

このライブラリに、以下の様な引数を設定することで、8番目の引数に指定した変数に目標位置が代入されます。

引数

- 1番目… 横分割数N、1以上であることが必要です。
例では、3列になっています。
- 2番目… 縦分割数M、1以上であることが必要です。
例では、5列になっています。
- 3番目… 段積み高さ (mm)
積み上がる場合にはプラス、下がる場合はマイナスの値を指定します。
例では、20mmを指定しています。
- 4番目～7番目… それぞれにパレットの四隅の位置を示すP型変数を指定します。例では、P52～P55を指定しています。
- 8番目… 目標位置代入変数
動作する位置を代入するP型変数を指定します。
この位置が、現在のカウンタから求められるパレタイジングの位置です。例では、P40を指定します。
- 9番目… パレタイジングカウンタです。1以上 N*M以下であることが必要です。この値でパレタイジング目標位置を指定します。
- 10番目… 段積みカウンタです。1以上であることが必要です。この値で段位置を指定します。

③アプローチ動作

④下降動作

⑤上昇動作

```
'----- ハレット動作 -----  
APPROACH P,P[40],@0 50      'アプローチ動作 P[40]上空50mmにアプローチ  
  
MOVE L,@0 P[40]             '下降動作 P[40]に下降  
  
DEPART L,50                 '上昇動作 50mm
```

「②ライブラリ呼び出し」を実行した結果、P40に目標位置が代入されたので、P40に動作を行ないます。

通常は、これらの動作の途中でチャック・アンチャックの処理を挿入します。

⑥カウンタ加算

⑦一段終了確認

⑧カウンタを初期値に戻す

⑨段積みカウンタ加算

⑩段積み終了確認

⑪段積みカウンタを初期値に戻す

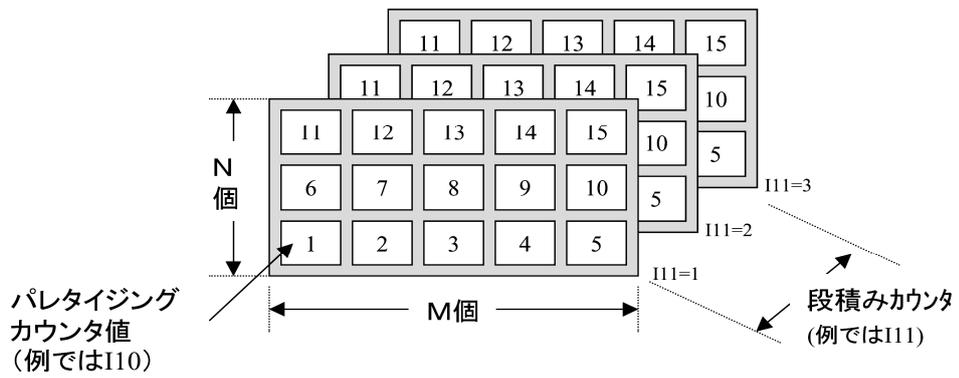
```
'----- パレタイカウンタ カウントアップ -----  
I[10] = I[10] + 1          'パレタイカウンタをひとつ進める  
  
IF I[10] > (3 * 5) then    '一段(横3個 x 縦5個)終了したら  
  I[10] = 1                'パレタイカウンタを最初に値に戻す  
  I[11] = I[11] + 1        '段積みカウンタをカウントアップ  
  IF I[11] >= 5 THEN        '5段終了したら  
    I[10] = 1              '段積みカウンタを先頭の値に戻す  
  END IF  
END IF
```

ここでは、パレタイジングカウンタと段積みカウンタのカウントアップと、一段終了確認処理を行ないます。

簡易パレタイジングでは、通常のパレタイジングと違い、パレタイジングカウンタとして、通常の変数を用います。例では、I10を使用しています。

また、段積みカウンタも同様に通常の変数を用います。例では、I11を使用しています。

「②ライブラリ呼び出し」では、このI10とI11の値から位置を算出しP40に代入しています。



また、1段のみのパレタイジングの場合には、以下のように単純化できます。

```

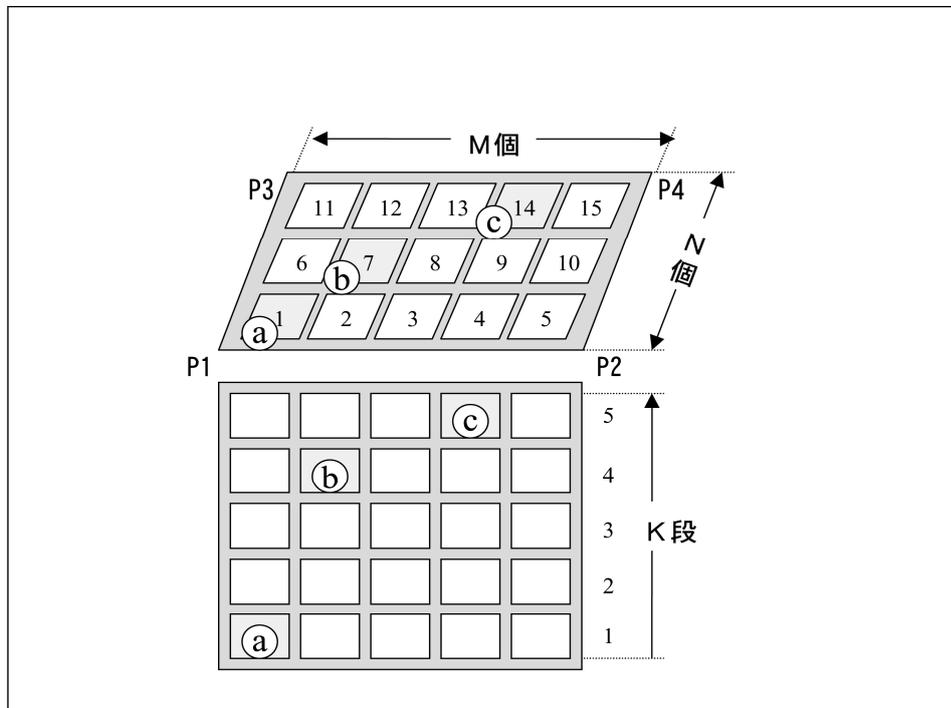
'----- パレタイジングカウンタ カウントアップ -----
I[10] = I[10] + 1      'パレタイジングカウンタをひとつ進める

if I[10] > (3 * 5) then  '一段(横3個 x 縦5個)終了したら
  I[10] = 1             'パレタイジングカウンタを最初に値に戻す
  I[11] = I[11] + 1    '段積みカウンタをカウントアップ
  IF I[11] >= 5 THEN   '5段終了したら
    I[10] = 1          '段積みカウンタを先頭の値に戻す
  END IF
END IF

```

一段のパレタイジングの場合、削除できます。

簡易パレタイジングでの位置とカウンタの関係



パレットが $N=3$, $M=5$ 、パレタイジングカウンタ $I10$ 、段積みカウンタ $I11$ の場合

- ① $I10 = 1$, $I11 = 1$
- ② $I10 = 7$, $I11 = 4$
- ③ $I10 = 14$, $I11 = 5$

付録

練習問題の回答例

■ 練習問題 1 (17.3 練習問題 ロボット制御コマンド)

プログラム	動作説明
' TITLE "練習プログラム 1"	' プログラムタイトル
PROGRAM PR01	' プログラム名の宣言
TAKEARM	' アーム制御権取得
SPEED 100	' 内部スピード100%
DRIVEA @0 (1 , 0)	' (1) : J1軸を0度の位置へ移動
APPROACH P, P1, @P 50	' (2) : P1のハンド方向 上空50mmへ移動
MOVE L, @E P1, S=20	' (3) : P1へ移動
DEPART L, @P 50	' (4) : P1のハンド方向 上空50mmへ移動
APPROACH P, P2, @P 50	' (5) : P2のハンド方向 上空50mmへ移動
MOVE L, @E P1, S=20	' (6) : P2へ移動
DEPART L, @P 50	' (7) : P2のハンド方向 上空50mmへ移動
MOVE P, @O P10	' (8) : P10へ移動
END	' プログラムの停止

■ 練習問題 2 (18.6 練習問題 フロー制御コマンド)

プログラム	動作説明
' TITLE "練習プログラム 2"	' プログラムタイトル
PROGRAM PR02	' プログラム名の宣言
TAKEARM	' アーム制御権取得
SPEED 100	' 内部スピード100%
MOVE P, P1	' (1) : P1へ移動
IF I[5] = 0 THEN	' (2) : IF文 I5=0なら次の命令へ
APPROACH L, P10 50	' (2)-1 : P10のハンド方向 上空50mmへ移動
MOVE L, P10	' (3)-1 : P10へ移動
ELSE	' (2) : IF文 I5=0でない場合、次の命令へ
APPROACH L, P11 50	' (2)-2 : P11のハンド方向 上空50mmへ移動
MOVE L, P11	' (3)-2 : P11へ移動
ENDIF	' (2) : IF文の終了
CALL HAND_OPEN	' HAND_OPENプログラム呼び出し
DEPART L, 50	' (4)-1, (4)-2 : P10またはP11のハンド方向 上空50mmへ移動
END	' プログラムの停止

■ 練習問題 3 (19.3 練習問題 I/O ポート制御コマンド)

プログラム	動作説明
' TITLE “練習プログラム 3”	' プログラムタイトル
PROGRAM PR03	' プログラム名の宣言
SET IO[64]	' (1) : ハンド閉信号IO[64]をON
WAIT IO[64] , 3000, I[20]	' (2) : IO[64]入力待ち3秒、格納変数I20使用
IF I[20] = 1 THEN	' (3) : 格納変数I20=1(成立)なら次の命令へ
SET IO[129]	' (4) : IO[129]をON
ELSE	' (3) : 格納変数I20=1(成立)以外なら次の命令へ
SET IO[128]	' (5) : IO[128]をON
ENDIF	' (3) : IF文の終了
END	' プログラムの停止

機能別コマンド一覧

機能別コマンド一覧に示された説明ページは、プログラミングマニュアル1のページになります。
プログラミングマニュアル1を参照下さい。

機能別コマンド一覧

4 軸	6 軸	視覚装置	
◎	◎	◎	全ロボットおよび視覚装置で使用可能
○	○	○	全ロボットで使用可能、ただし4軸ロボット、6軸ロボット、または視覚装置で仕様が異なる。
◎	V1.2		4軸ロボットとバージョン1.2以降の6軸ロボットで使用可能

機能区分	コマンド	機能	4 軸	6 軸	視覚装置	説明 ページ
宣言文						
プログラム名	PROGRAM	プログラム名を宣言します。	◎	◎	◎	9-1
干渉エリア座標	AREA	干渉チェックを行なうエリアを宣言します。	○	○		9-2
ユーザ関数	DEF FN	ユーザ定義関数を宣言します。	◎	◎	◎	9-4
ホーム座標	HOME	任意の座標をホームポジションとして宣言します。	○	○		9-5
ツール座標	TOOL	ツール座標系を宣言します。	○	○		9-6
ワーク座標	WORK	ユーザ座標系を宣言します。	○	○		9-7
ローカル変数 I 型	DEFINT	整数型変数を宣言します。整数の範囲は -2147483648 ~ 2147483647 です。	◎	◎	◎	9-8
F 型	DEFSNG	単精度実数型変数を宣言します。単精度実数の範囲は -3.4E-38 ~ 3.4E+38 です。	◎	◎	◎	9-8
D 型	DEFDBL	倍精度実数型変数を宣言します。倍精度実数の範囲は -1.7D+308 ~ 1.7D+308 です。	◎	◎	◎	9-9
S 型	DEFSTR	文字列型変数を宣言します。文字列の長さは、243 文字までです。	◎	◎	◎	9-9
V 型	DEFVEC	ベクトル型変数を宣言します。	◎	◎		9-10
P 型	DEFPOS	ポジション型変数を宣言します。	○	○		9-10
J 型	DEFJNT	ジョイント型変数を宣言します。	○	○		9-11
T 型	DEFTRN	同次変換型変数を宣言します。	◎	◎		9-11
I/O 型	DEFIO	入出力ポートに対応する I/O 変数を宣言します。	◎	◎	◎	9-12
配列	DIM	配列を宣言します。	◎	◎	◎	9-13
フォルダー機能	FOLDER	外部プログラムからアクセス可能なローカル変数を宣言します。	V2.2	V2.2		9-14
	EXTERN	他のプログラムで定義されている FOLDER 変数を参照することを宣言します。	V2.2	V2.2		19-17
代入文						
変数	LET	変数に値を代入します。	○	○	○	10-1
ベクトル	LETA	同次変換型のアプローチベクトルへ代入します。		◎		10-2
	LETO	同次変換型のオリエン特ベクトルへ代入します。	◎	◎		10-3
	LETP	ポジション型または同次変換型の位置ベクトルへ代入します。	◎	◎		10-4
形態	LETF	ポジション型または同次変換型の形態成分へ代入します。	◎	◎		10-5
リンク角	LETJ	ジョイント型の指定リンク角へ代入します。	◎	◎		10-6
姿勢	LETR	ポジション型の 3 つの回転成分へ代入します。		◎		10-7
回転成分	LETRX	ポジション型の X 軸回転成分へ代入します。		◎		10-8
	LETRY	ポジション型の Y 軸回転成分へ代入します。		◎		10-9
	LETRZ	ポジション型の Z 軸回転成分へ代入します。		◎		10-10
	LETT	ポジション型の T 成分へ代入します。	◎			10-11

4 軸 6 軸 視覚装置

◎	◎	◎	全ロボットおよび視覚装置で使用可能
○	○	○	全ロボットで使用可能、ただし4軸ロボット、6軸ロボット、または視覚装置で仕様が異なる。
◎	V1.2		4軸ロボットとバージョン1.2以降の6軸ロボットで使用可能

機能区分	コマンド	機能	4 軸	6 軸	視覚装置	説明 ページ
軸成分	LETX	ベクトル型／ポジション型／同次変換型の X 軸成分へ代入します。	◎	◎		10-12
	LETY	ベクトル型／ポジション型／同次変換型の Y 軸成分へ代入します。	◎	◎		10-13
	LETZ	ベクトル型／ポジション型／同次変換型の Z 軸成分へ代入します。	◎	◎		10-14
フロー制御文						
プログラムの停止	END	プログラムによる動作の終了を宣言します。	◎	◎	◎	11-1
	STOP	プログラムの実行を終了します。	◎	◎	◎	11-2
	STOPEND	連続起動または CYCLE オプション付きで起動されたプログラムをサイクル停止させるステートメントです。このステートメントを含むプログラムを、サイクル起動した場合は、このステートメントを実行しても、動作に影響しません。	◎	◎		11-3
呼び出し	CALL	プログラムを呼び出し、実行します。	◎	◎	◎	11-4
	GOSUB	サブルーチンを呼び出します。	◎	◎	◎	11-6
	ON～GOSUB	式の値に応じて、対応するサブルーチンの呼び出しを行ないます。	◎	◎	◎	11-7
繰り返し	RETURN	サブルーチンから復帰します。	◎	◎	◎	11-8
	DO～LOOP	判定反復（繰り返し）を行ないます。	◎	◎	◎	11-9
	EXIT DO	DO～LOOP からの強制脱出を行ないます。	◎	◎	◎	11-11
FOR～NEXT	FOR～NEXT	FOR～NEXT までの区間中にある一連の命令を繰り返して実行します。	◎	◎	◎	11-12
	EXIT FOR	FOR～NEXT からの強制脱出を行ないます。	◎	◎	◎	11-14
	REPEAT～UNTIL	後判定反復を行ないます。	◎	◎	◎	11-15
条件分岐	WHILE～WEND	前判定反復を行ないます。	◎	◎	◎	11-16
	IF～END IF	IF～END IF 間の条件式の条件判断を行ないます。	◎	◎	◎	11-17
	IF～THEN～ELSE	論理式の条件判断を行ないます。	◎	◎	◎	11-18
無条件分岐	SELECT CASE	複数条件判断を行ないます。	◎	◎	◎	11-19
	GOTO	プログラムの分岐を無条件で行ないます。	◎	◎	◎	11-21
	ON～GOTO	式の値による無条件分岐を行ないます。	◎	◎	◎	11-22
コメント	REM	コメントの記述をします。	◎	◎	◎	11-23
ロボット制御文						
動作制御	APPROACH	ツール座標系指定の絶対動作を行ないます。	○	○		12-1
	DEPART	ツール座標系指定で相対動作を行ないます。	○	○		12-4
	DRAW	ワーク作業座標系指定で相対動作を行ないます。	◎	◎		12-7
	DRIVE	各軸の相対動作を行ないます。	◎	◎		12-9
	DRIVEA	各軸の絶対動作を行ないます。	◎	◎		12-11
	GOHOME	HOME 文で定義したポジション（原点）へ移動します。	◎	◎		12-13
	MOVE	ロボットを指定座標へ移動します。	○	○		12-14

4 軸 6 軸 視覚装置

◎	◎	◎	全ロボットおよび視覚装置で使用可能
○	○	○	全ロボットで使用可能、ただし4軸ロボット、6軸ロボット、または視覚装置で仕様が異なる。
◎	V1.2		4軸ロボットとバージョン1.2以降の6軸ロボットで使用可能

機能区分	コマンド	機能	4 軸	6 軸	視覚装置	説明 ページ
	ROTATE	指定した軸回りの回転動作を行いません。	○	○		12-19
	ROTATEH	アプローチベクトルを軸とした、回転動作を行いません。	◎	◎		12-22
	CURJNT	ロボットの現在角度を J 型で得ます。	○	○		12-24
	CURPOS	ツール座標系での現在位置を P 型データで得ます。	○	○		12-25
	CURTRN	ツール座標系での現在位置を T 型データで得ます。	◎	◎		12-26
	CUREXJ	付加軸の現在角度を F 型で取得します。	V1.5	V1.5		12-27
	DESTJNT	現在の動作命令目標位置を J 型で取得します。	○	○		12-28
	DESTPOS	現在の動作命令目標位置を P 型で取得します。	○	○		12-29
	DESTTRN	ロボット停止時は、現在の位置（指令値）を取得します。 現在の動作命令目標位置を T 型で取得します。	◎	◎		12-30
	DESTEXJ	ロボット停止時は、現在の位置（指令値）を取得します。 付加軸の、現在の動作命令で指定された目標位置を F 型で取得します。停止時は、現在の位置（指令値）を取得します。	V1.5	V1.5		12-31
	ARRIVE	動作命令の全移動距離に対する動作割合を設定する事によって、ロボットが設定した動作割合に到達するまでプログラムを待機させます。	◎	V1.2		12-32
	POSCLR	軸の現在位置を強制的に 0mm または 0 度にします。	V1.5	V1.5		12-34
	SETSPLINEPOINT	自由曲線動作の通過点を登録します。	V2.3	V2.3		12-35
	CLRSPLINEPOINT	自由曲線動作の通過点をすべて消去します。	V2.3	V2.3		12-36
	GETSPLINEPOINT	登録済の自由曲線動作の通過点を取り出します。	V2.3	V2.3		12-37
形態制御	CURFIG	ロボット形態の現在値を得ます。	◎	◎		12-38
	FIGAPRL	CP 動作可能なアプローチ位置と基準位置の形態を計算します。	○	○		12-40
	FIGAPRP	PTP 動作可能なアプローチ位置と基準位置の形態を計算します。	○	○		12-42
停止制御	HOLD	プログラムの実行を一時停止します。	◎	◎		12-43
	HALT	プログラムの実行を停止します。	◎	◎		12-44
	INTERRUPT ON/OFF	ロボットの動作を中断します。	◎	◎		12-45
速度制御	SPEED	現在取得しているアームグループ軸の内部移動速度を指定します。	◎	◎		12-47
	JSPEED	現在取得しているアームグループ軸の内部軸速度を指定します。	◎	◎		12-49
	ACCEL	現在取得しているアームグループ軸の内部加速度、内部減速度を指定します。	◎	◎		12-50
	JACCEL	現在取得しているアームグループ軸の、内部軸加速度、内部軸減速度を指定します。	◎	◎		12-51

4 軸	6 軸	視覚装置	
◎	◎	◎	全ロボットおよび視覚装置で使用可能
○	○	○	全ロボットで使用可能、ただし4軸ロボット、6軸ロボット、または視覚装置で仕様が異なる。
◎	V1.2		4軸ロボットとバージョン1.2以降の6軸ロボットで使用可能

機能区分	コマンド	機能	4 軸	6 軸	視覚装置	説明 ページ
	DECEL	現在取得しているアームグループ軸の内部減速度を指定します。	◎	◎		12-52
	JDECEL	現在取得しているアームグループ軸の内部軸減速度を指定します。	◎	◎		12-53
	CURACC	現在取得しているアームグループ軸の、内部加速度を取得します。	◎	◎		12-54
	CURJACC	現在取得しているアームグループ軸の、内部軸加速度を取得します。	◎	◎		12-55
	CURDEC	現在取得しているアームグループ軸の、内部減速度を取得します。	◎	◎		12-56
	CURJDEC	現在取得しているアームグループ軸の、内部軸減速度を取得します。	◎	◎		12-57
	CURJSPD	現在取得しているアームグループ軸の、内部移動軸速度を取得します。	◎	◎		12-58
	CURSPD	現在取得しているアームグループ軸の、内部移動速度を取得します。	◎	◎		12-59
	CUREXTACC	外部加速度の現在値を得ます。	V1.4	V1.4		12-60
	CUREXTDEC	外部減速度の現在値を得ます。	V1.4	V1.4		12-61
	CUREXTSPD	外部速度の現在値を得ます。	V1.4	V1.4		12-62
	EXTSPEED	外部速度を設定します。	V1.98	V1.98		12-62
時間制御	DELAY	指定した時間の間、プログラムの進行を停止します。	◎	◎	◎	12-63
	WAIT	条件プログラム停止を行ないます。	◎	◎		12-64
座標変換	CHANGETOOL	ツール座標系を変更します。	◎	◎		12-65
	CHANGework	ユーザ座標系を変更します。	◎	◎		12-66
	CURTOOL	現在設定されている TOOL 番号を得ます。	V1.4	V1.4		12-67
	CURWORK	現在設定されている WORK 番号を得ます。	V1.4	V1.4		12-68
干渉チェック	SETAREA	干渉チェックを行なうエリアを選択します。	◎	◎		12-69
	RESETAREA	干渉チェックを初期化します。	◎	◎		12-70
サーボ内部データ	GetSrvData	ロボット軸のサーボ内部データを取得します。	V1.5	V1.5		12-71
	GetJntData	指定軸のサーボ内部データを取得します。				12-72
モータ電源	MOTOR {ON OFF}	モータ電源 ON/OFF を行ないます。	V1.5	V1.5		12-73
CAL	EXECAL	CAL を実行します。	V1.5	V1.5		12-74
特殊制御	ST_aspACLD	内部負荷条件値を変更します。負荷条件値は、先端負荷質量 (g)、負荷重心位置 (mm) ですべて指定します。	V1.9	V1.9		12-75
	ST_aspChange	最適可搬質量設定モードの内部モードを選択します。	V1.9	V1.9		12-76
	ST_SetGravity	各関節の静荷重 (重力トルク) を補正し、重力バランスを設定します。	V1.9	V1.9		12-77
	ST_ResetGravity	重力バランスを無効にします。	V1.9	V1.9		12-78
	ST_SetGrvOffset	各関節の重力トルクより重力補償値を補正します。	V1.9	V1.9		12-78
	ST_ResetGrvOffset	重力補償値の補正を無効にします。	V1.9	V1.9		12-79
	ST_SetCurLmt	指定した軸のモータ電流値を制限します。	V1.9	V1.9		12-79
	ST_ResetCurLmt	指定した軸のモータ電流制限を解除します。	V1.9	V1.9		12-81

4 軸 6 軸 視覚
装置

◎	◎	◎	全ロボットおよび視覚装置で使用可能
○	○	○	全ロボットで使用可能、ただし4軸ロボット、6軸ロボット、または視覚装置で仕様が異なる。
◎	V1.2		4軸ロボットとバージョン1.2以降の6軸ロボットで使用可能

機能区分	コマンド	機能	4 軸	6 軸	視覚 装置	説明 ページ
	ST_SetEralw	指定した軸の偏差許容値を変更します。	V1.9	V1.9		12-82
	ST_ResetEralw	指定した軸の偏差許容値をデフォルト値に戻します。	V1.9	V1.9		12-82
	ST_OnSrvLock	指定した軸をサーボロック状態にします。	V1.9			12-83
	ST_OffSrvLock	指定した軸のサーボロックを解除します。	V1.9			12-84
	ST_SetCompControl	力制限機能を有効にします。		V1.9		12-85
	ST_SetCompFControl	力制限機能を有効にします。		V1.9		12-86
	ST_ResetCompControl	力制限機能を無効にします。		V1.9		12-87
	ST_SetFrcCoord	力制限設定座標系を選択します。		V1.9		12-88
	ST_SetFrcLimit	力制限割合を設定します。		V1.9		12-89
	ST_ResetFrcLimit	力制限割合を初期化します。		V1.9		12-90
	ST_SetCompRate	力制限時の柔らかさの割合を設定します。		V1.9		12-91
	ST_ResetCompRate	力制限時の柔らかさの割合を初期化します。		V1.9		12-92
	ST_SetFrcAssist	力制限時のオフセット力を設定します。		V1.9		12-92
	ST_ResetFrcAssist	力制限時のオフセット力を初期化します。		V1.9		12-93
	ST_SetCompJLimit	力制限時の電流制限値を設定します。		V1.9		12-93
	ST_ResetCompJLimit	力制限時の電流制限値を初期化します。		V1.9		12-94
	ST_SetCompVMode	力制限時の速度制御モードを設定します。		V1.9		12-95
	ST_ResetCompVMode	力制限時の速度制御モードを無効にします。		V1.9		12-95
	ST_SetCompEralw	力制限時のツール端の位置、姿勢偏差許容値を設定します。		V1.9		12-96
	ST_ResetCompEralw	力制限時のツール端の位置、姿勢偏差許容値を初期化します。		V1.9		12-97
	ST_SetDampRate	力制限時の粘性割合を設定します。		V1.9		12-97
	ST_ResetDampRate	力制限時の粘性割合を初期化します。		V1.9		12-98
	ST_SetZBalance	Z 軸、T 軸の重力補償値を設定します。	V1.9			12-99
	ST_ResetZBalance	重力補償値の補正を無効にします。	V1.9			12-99

入出力制御文

I/O ポート	IN	I/O 変数で示される I/O ポートからデータを読み込みます。	◎	◎	◎	13-1
	OUT	I/O 変数で示される I/O ポートにデータを出力します。	◎	◎	◎	13-2
	IOBLOCK ON/OFF	動作命令を実行中に、I/O 命令や演算命令などの非動作命令を並列に実行します。	◎	◎		13-3
	SET	I/O ポートを ON にします。	◎	◎	◎	13-5
	RESET	I/O ポートを OFF にします。	◎	◎	◎	13-7
RS232C および Ethernet ポート	INPUT	RS232C および Ethernet ポートからのデータを得ます。	◎	◎	◎	13-8

4 軸 6 軸 視覚
装置

◎	◎	◎	全ロボットおよび視覚装置で使用可能
○	○	○	全ロボットで使用可能、ただし4軸ロボット、6軸ロボット、または視覚装置で仕様が異なる。
◎	V1.2		4軸ロボットとバージョン1.2以降の6軸ロボットで使用可能

機能区分	コマンド	機能	4 軸	6 軸	視覚 装置	説明 ページ
シリアルバイナリ通信	LINEINPUT	RS232C および Ethernet ポートからデ リミタまでのデータを読み込み、文字 列型変数に代入します。	◎	◎	◎	13-10
	PRINT	RS232C および Ethernet ポートからデ ータを出力します。	◎	◎	◎	13-11
	WRITE	RS232C および Ethernet ポートからデ ータを出力します。	◎	◎	◎	13-12
	FLUSH	入力バッファをクリアします。	◎	◎	◎	13-13
	PRINTB	1 バイトデータを RS-232C および Ethernet ポートに出力します。	V1.5	V1.5	V1.9	13-14
	INPUTB	1 バイトデータを RS-232C および Ethernet ポートから入力します。	V1.5	V1.5	V1.9	13-15
	LPRINTB	複数バイトデータを RS-232C および Ethernet ポートに出力します。	V1.5	V1.5	V1.9	13-16
	LINPUTB	複数バイトデータを RS-232C および Ethernet ポートから入力します。	V1.5	V1.5	V1.9	13-17
	com_encom	RS-232C ポートをバイナリ通信のため に占有します。(COM ポート占有)	V1.5	V1.5	V1.9	13-18
	com_discom	RS-232C ポートをバイナリ通信終了の ため開放します。(COM ポート開放)	V1.5	V1.5	V1.9	13-19
ティーチングペンダント	com_state	RS-232C ポートの状態を取得します。	V1.5	V1.5	V1.9	13-20
	PRINTMSG	メッセージをキャプションとアイコ ンを付けて、ティーチングペンダント のカラー液晶に表示します。	◎	◎		13-21
	PRINTDBG	デバッグウィンドウにデータを出力 します。	◎	◎		13-22
	BUZZER PRINTWARNING	ブザーを鳴らします。 ペンダント警告メッセージ表示部分 へメッセージを表示します。	V2.2	V2.2		13-23
TP 簡易操作盤	PRINTLBL	ユーザ定義ボタンにラベル (キャプシ ョン) を設定します。	◎	◎		13-23
	set_button	ボタンパラメータの設定を行ないま す。	V1.5	V1.5		13-27
	set_page	ページパラメータの設定を行ないま す。	V1.5	V1.5		13-31
	change_bCap	ボタンのキャプション設定を行ない ます。	V1.5	V1.5		13-33
	change_pCap	ページのキャプション設定を行ない ます。	V1.5	V1.5		13-34
	disp_page	ページの表示を行ないます。	V1.5	V1.5		13-35
マルチタスク制御文						
タスク制御	RUN	別プログラムを並列起動します。	◎	◎		14-1
	KILL	タスクを強制終了します。	◎	◎		14-2
	SUSPEND	タスクを一時停止します。	◎	◎		14-3
	DEFEND	タスクを保護します。	◎	◎		14-4
	STATUS	プログラムの状態を得ます。	◎	◎		14-5
	SUSPENDALL	特権タスク以外の全てのプログラム を停止します。	V1.98	V1.98		14-6
	KILLALL	特権タスク以外の全てのタスクを強 制終了します。	V1.98	V1.98		14-7
	CONTINUERUN	コンティニュー起動を行ないます。	V1.98	V1.98		14-7
	ROBOTSTOP	ロボット停止を行ないます。	V1.98	V1.98		14-8

4軸 6軸 視覚装置

◎	◎	◎	全ロボットおよび視覚装置で使用可能
○	○	○	全ロボットで使用可能、ただし4軸ロボット、6軸ロボット、または視覚装置で仕様が異なる。
◎	V1.2		4軸ロボットとバージョン1.2以降の6軸ロボットで使用可能

機能区分	コマンド	機能	4軸	6軸	視覚装置	説明ページ	
セマフォ	TAKEARMSTATE	アームグループ制御権の取得状態を返します。	V2.2	V2.2		14-9	
	LOCKSTATE	マシンロックの状態を取得します。	V2.2	V2.2		14-9	
	DEADMANSTATE	デッドマンスイッチの状態を取得します。	V2.2	V2.2		14-10	
	SEMIDSTATE	指定したセマフォ ID が有効か無効かを返します。	V2.2	V2.2		14-11	
	CREATESEM	セマフォを生成します。	◎	◎		14-12	
	DELETSEM	セマフォを削除します。	◎	◎		14-15	
	FLUSHSEM	セマフォ待ちタスクを解放します。	◎	◎		14-16	
	GIVSEM	セマフォ待ちタスクを解放します。	◎	◎		14-17	
	TAKESEM	指定したセマフォ ID を持つセマフォを取得します。	◎	◎		14-18	
	アームセマフォ	TAKEARM	アームグループを取得します。アームグループ NO を省略した場合、アームグループ 0 (ロボットのみ) の制御権を取得します。取得時、内部速度・加速度・減速度を100に設定します。取得するアームグループがロボット軸を含む場合は、ツール座標、ワーク座標を原点に戻します。	◎	◎		14-19
特権タスク	GIVEARM	ロボット制御権を解放します。	◎	◎		14-24	
	TAKEVIS	視覚処理権を取得します。	◎	◎		14-25	
	GIVEVIS	視覚処理権を解放します。	◎	◎		14-26	
	INIT	特権タスクパラメータの状態により、モータ ON、CAL、スピード設定を行います。	V1.7	V1.7		14-27	
	SETOCCUPATIONTIME	特権タスクの占有時間を変更します。	V2.0	V2.0		14-28	
	INITWAITERR	エラー検知を初期化します。(特権タスク専用命令)	V2.2	V2.2		14-29	
	WAITERROR	エラー発生を検知します。(特権タスク専用命令)	V2.2	V2.2		14-30	
	CURERRSTATUS	エラー発生状態を返します。(特権タスク専用命令)	V2.2	V2.2		14-31	
関数							
算術関数	ABS	数式の値の絶対値を求めます。	◎	◎	◎	15-1	
	EXP	自然対数を基数とする指数関数を求めます。	◎	◎	◎	15-2	
	INT	指定された値を超えない最大の整数値を求めます。	◎	◎	◎	15-3	
	LOG	自然対数を求めます。	◎	◎	◎	15-4	
	LOG10	常用対数を求めます。	◎	◎	◎	15-5	
	POW	べき乗を求めます。	◎	◎	◎	15-6	
	MAX	最大値を抽出します。	◎	◎	◎	15-7	
	MIN	最小値を抽出します。	◎	◎	◎	15-8	
	RND	0以上1以下の乱数を発生します。	◎	◎	◎	15-9	
	SGN	符号を調べます。	◎	◎	◎	15-10	
	SQR	平方根を求めます。	◎	◎	◎	15-11	
	三角関数	ACOS	逆余弦(アークコサイン)を求めます。	◎	◎	◎	15-12
		ASIN	逆正弦(アークサイン)を求めます。	◎	◎	◎	15-13
		ATN	逆正接(アークタンジェント)を求めます。	◎	◎	◎	15-14

4軸 6軸 視覚装置

◎	◎	◎	全ロボットおよび視覚装置で使用可能
○	○	○	全ロボットで使用可能、ただし4軸ロボット、6軸ロボット、または視覚装置で仕様が異なる。
◎	V1.2		4軸ロボットとバージョン1.2以降の6軸ロボットで使用可能

機能区分	コマンド	機能	4軸	6軸	視覚装置	説明ページ
角度変換	ATN2	数式1を数式2で除算した逆正接(アークタンジェント)を得ます。	◎	◎	◎	15-15
	COS	余弦(コサイン)を得ます。	◎	◎	◎	15-16
	SIN	正弦(サイン)を得ます。	◎	◎	◎	15-17
	TAN	正接(タンジェント)を得ます。	◎	◎	◎	15-18
	DEGRAD	単位をラジアンに変換します。	◎	◎	◎	15-19
	RAD	ラジアンで与えた数値を角度に変換します。	◎	◎	◎	15-20
速度変換 時間関数	RADDEG	単位を度に変換します。	◎	◎	◎	15-21
	MPS	速度の表現を変換します。	◎	◎		15-22
	SEC	秒の単位で与えた数値をミリ秒に変換します。	◎	◎	◎	15-23
ベクトル	AVEC	アプローチベクトルを抽出します。	◎	◎		15-24
	OVEC	オリエントベクトルを抽出します。	◎	◎		15-25
	PVEC	位置ベクトルを抽出します。	○	○		15-26
	MAGNITUDE	ベクトルの大きさを得ます。	◎	◎		15-27
ポーズデータ型変換	J2P	ジョイント型からポジション型に変換します。	○	○		15-28
	J2T	ジョイント型から同次変換型に変換します。	○	○		15-29
	P2J	ポジション型からジョイント型に変換します。	○	○		15-30
	P2T	ポジション型から同次変換型に変換します。	○	○		15-31
	T2J	同次変換型からジョイント型に変換します。	◎	◎		15-32
	T2P	同次変換型からポジション型に変換します。	◎	◎		15-33
	TINV	同次変換型の逆行列を計算します。	◎	◎		15-34
	NORMTRN	同次変換型の正規化計算を行います。	V1.8	V1.8		15-34
距離抽出	DIST	2点間の距離を返します。	○	○		15-35
形態成分	FIG	形態を抽出します。	○	○		15-36
角度成分	JOINT	ジョイント型座標から角度を抽出します。	○	○		15-37
軸成分	POSX	X成分を抽出します。	○	○		15-38
	POSY	Y成分を抽出します。	○	○		15-39
	POSZ	Z成分を抽出します。	○	○		15-40
回転成分	POSRX	X軸回転成分を抽出します。		◎		15-41
	POSRY	Y軸回転成分を抽出します。		◎		15-42
	POSRZ	Z軸回転成分を抽出します。		◎		15-43
	POST	T軸回転成分を抽出します。	◎			15-44
姿勢成分 位置関数	RVEC	姿勢を抽出します。		◎		15-45
	AREAPOS	干渉チェックの行なわれる領域の中心位置と直方体の方向をポジション型で返します。	◎	◎		15-46
	AREASIZE	干渉チェックの行なわれる領域を定義する、直方体の大きさ(各辺の長さ)をベクトル型で返します。	◎	◎		15-47
	TOOLPOS	ツール座標系をポジション型で返します。	◎	◎		15-48
	WORKPOS	ユーザ座標系をポジション型で返します。	◎	◎		15-49
文字列関数	ASC	文字コードへ変換します。	◎	◎	◎	15-50

4軸 6軸 視覚装置

◎	◎	◎	全ロボットおよび視覚装置で使用可能
○	○	○	全ロボットで使用可能、ただし4軸ロボット、6軸ロボット、または視覚装置で仕様が異なる。
◎	V1.2		4軸ロボットとバージョン1.2以降の6軸ロボットで使用可能

機能区分	コマンド	機能	4軸	6軸	視覚装置	説明ページ
	BIN\$	数式の値を2進数の文字列へ変換します。	◎	◎	◎	15-51
	CHR\$	ASCIIコードを文字に変換します。	◎	◎	◎	15-52
	SPRINTF\$	式を指定したフォーマットに変換し、文字列として返します。	◎	◎	◎	15-53
	HEX\$	10進数から16進数へ変換した値を文字列として得ます。	◎	◎	◎	15-56
	LEFT\$	左部分文字列を抽出します。	◎	◎	◎	15-57
	LEN	文字列の長さをバイト数で得ます。	◎	◎	◎	15-58
	MID\$	文字列から指定した文字数分の文字列を取り出します。	◎	◎	◎	15-59
	ORD	文字コードへ変換します。	◎	◎	◎	15-60
	RIGHT\$	文字列の右部分を抽出します。	◎	◎	◎	15-61
	STRPOS	文字列の位置を得ます。	◎	◎	◎	15-62
	STR\$	数値から文字列に変換します。	◎	◎	◎	15-63
	VAL	文字列から数値へ変換します。	◎	◎	◎	15-64
定数						
組み込み定数	OFF	OFF(0)の値を与えます。	◎	◎	◎	16-1
	ON	ON(1)の値を与えます。	◎	◎	◎	16-2
	PI	πの値を与えます。	◎	◎	◎	16-3
	FALSE	ブール値の偽(0)の値を与えます。	◎	◎	◎	16-4
	TRUE	ブール値の真(1)の値を与えます。	◎	◎	◎	16-5
時刻/日付制御						
時刻/日付	DATE\$	現在の日付を得ます。	◎	◎		17-1
	TIME\$	現在の時刻を得ます。	◎	◎		17-2
	TIMER	経過時間を得ます。	◎	◎		17-3
エラー制御						
エラー情報	ERRMSG\$	エラーメッセージを与えます。	◎	◎	◎	18-1
	SETERR	ユーザ定義エラーをI型変数領域にセットします。	V1.98	V1.98		18-1
	GETERR	エラー格納機能で宣言されたバッファからエラーコードを取得します。	V1.98	V1.98		18-2
	CLRERR	エラーをクリアします。	V1.98	V1.98		18-3
	GETERRLVL	エラーコードのレベルを与えます。	V1.98	V1.98		18-3
システム情報						
システム	GETENV	システムの環境設定値を取得します。	◎	◎	◎	19-1
	LETENV	システムの環境設定値を設定します。	◎	◎	◎	19-2
	VER\$	各モジュールのバージョンを取得します。	◎	◎	◎	19-3
	GETLANGUAGE	現在の言語設定を取得します。	V2.2	V2.2		19-3
ログ	STARTLOG	サーボ制御ログの記録を開始します。	◎	◎		19-4
	CLEARLOG	サーボ制御ログの記録を初期化します。	◎	◎		19-5
動作モード	STOPLOG	サーボ制御ログの記録を停止します。	◎	◎		19-6
	CHGEXTMODE	外部自動モード切替を行ないます。	V1.98	V1.98		19-7
	CHGINTMODE	内部自動モード切替を行ないます。	V1.98	V1.98		19-8
	CUROPTMODE	動作モードを取得します。	V1.98	V1.98		19-8
	SYSSTATE	コントローラのステータスを取得します。	V1.98	V1.98		19-9
プリ・プロセッサ						

4軸 6軸 視覚装置

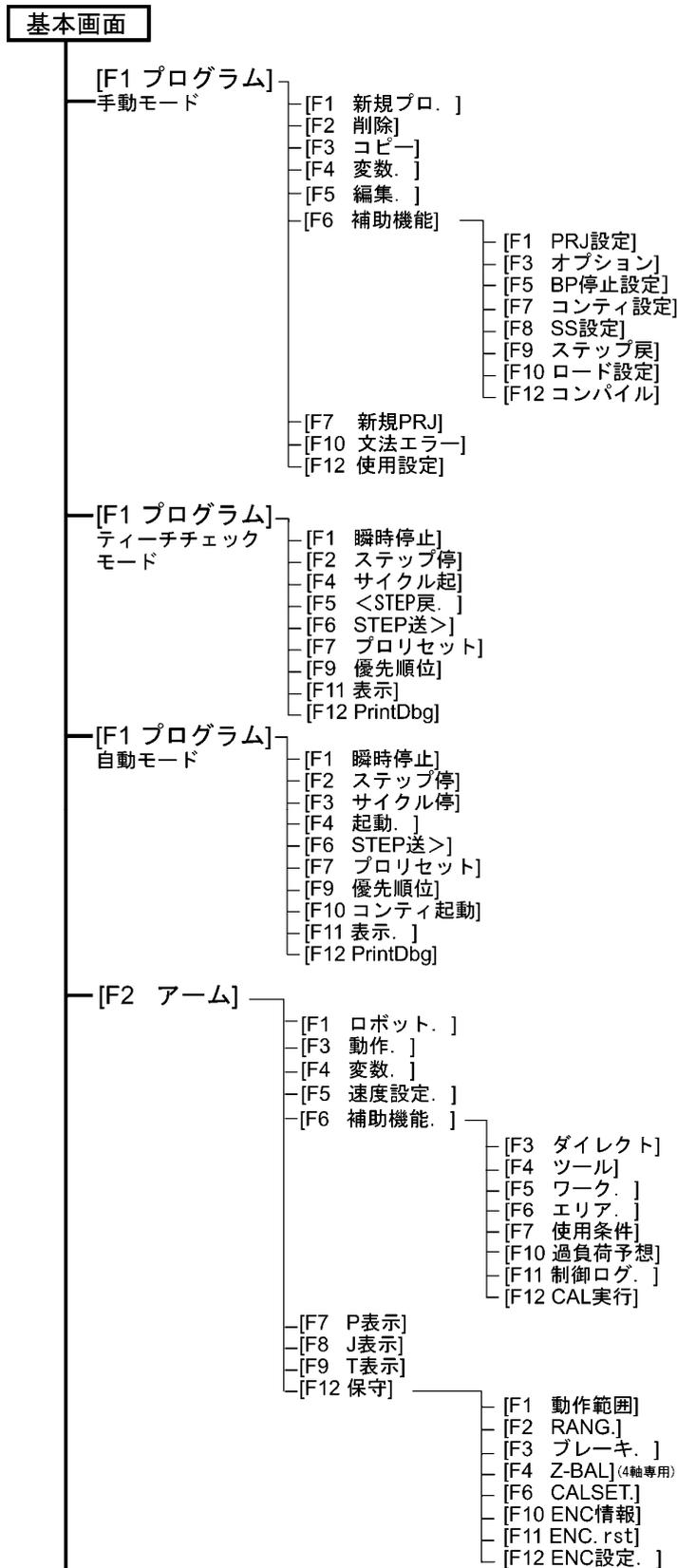
◎	◎	◎	全ロボットおよび視覚装置で使用可能
○	○	○	全ロボットで使用可能、ただし4軸ロボット、6軸ロボット、または視覚装置で仕様が異なる。
◎	V1.2		4軸ロボットとバージョン1.2以降の6軸ロボットで使用可能

機能区分	コマンド	機能	4軸	6軸	視覚装置	説明ページ
記号定数・マクロ定義	#define	プログラム中の指定した定数またはマクロ名を、指定文字列で置き換えます。	◎	◎	◎	20-1
	#undef	#define で定義されている記号定数またはマクロの定義を無効にします。	◎	◎	◎	20-2
	#error	#error コマンドを実行すると、強制的にコンパイルエラーとします。	◎	◎	◎	20-3
	#include	プリプロセッサプログラムを取り込みます。	◎	◎	◎	20-4
最適化	#pragma optimize	プログラムごとに行う最適化を指定します。	◎	◎	◎	20-6
視覚制御						
画像入出力	CAMIN	カメラからの映像を画像メモリ（処理画面）に格納します。	◎	◎	◎	21-3
	CAMMODE	カメラ映像を格納する際の機能を設定します。	◎	◎	◎	21-4
	CAMLEVEL	カメラ映像の入力レベルを設定します。	◎	◎	◎	21-6
	VISCAMOUT	カメラからの映像をモニタに表示します。	◎	◎	◎	21-7
	VISPLNOUT	格納メモリの画像をモニタに表示します。	◎	◎	◎	21-8
	VISOVERLAY	描画面面の情報をモニタに表示します。	◎	◎	◎	21-9
	VISDEFTABLE	カメラ映像の取り込み、画像出力の際のルックアップテーブルデータを設定します。	◎	◎	◎	21-11
	VISREFTABLE	ルックアップテーブルのデータを参照します。	◎	◎	◎	21-13
ウィンドウ設定	WINDMAKE	画像処理する範囲を指定します。	◎	◎	◎	21-14
	WINDCLR	設定したウィンドウ情報を消去します。	◎	◎	◎	21-19
	WINDCOPY	ウィンドウのデータをコピーします。	◎	◎	◎	21-20
	WINDREF	ウィンドウの情報を得ます。	◎	◎	◎	21-22
描画	WINDDISP	設定したウィンドウを描画します。	◎	◎	◎	21-23
	VISSCREEN	描画する画面を指定します。	◎	◎	◎	21-24
	VISBRIGHT	描画する際の輝度値を指定します。	◎	◎	◎	21-26
	VISCLS	モードで指定した画面を指定輝度で塗りつぶし（クリア）します。	◎	◎	◎	21-27
	VISPUTP	画面に点を描画します。	◎	◎	◎	21-29
	VISLINE	画面に直線を描画します。	◎	◎	◎	21-30
	VISPTP	画面に 2 点間を結ぶ直線を描画します。	◎	◎	◎	21-31
	VISRECT	画面に矩形を描画します。	◎	◎	◎	21-32
	VISCIRCLE	画面に円を描画します。	◎	◎	◎	21-33
	VISELLIPSE	画面に楕円を描画します。	◎	◎	◎	21-34
	VISSECT	画面に扇を描画します。	◎	◎	◎	21-35
	VISCROSS	画面に十字マークを描画します。	◎	◎	◎	21-36
VISLOC	文字の表示位置を指定します。	◎	◎	◎	21-37	
VISDEFCHAR	文字のサイズと表示方法を指定します。	◎	◎	◎	21-39	
VISPRINT	画面に文字・数字を表示します。	◎	◎	◎	21-40	

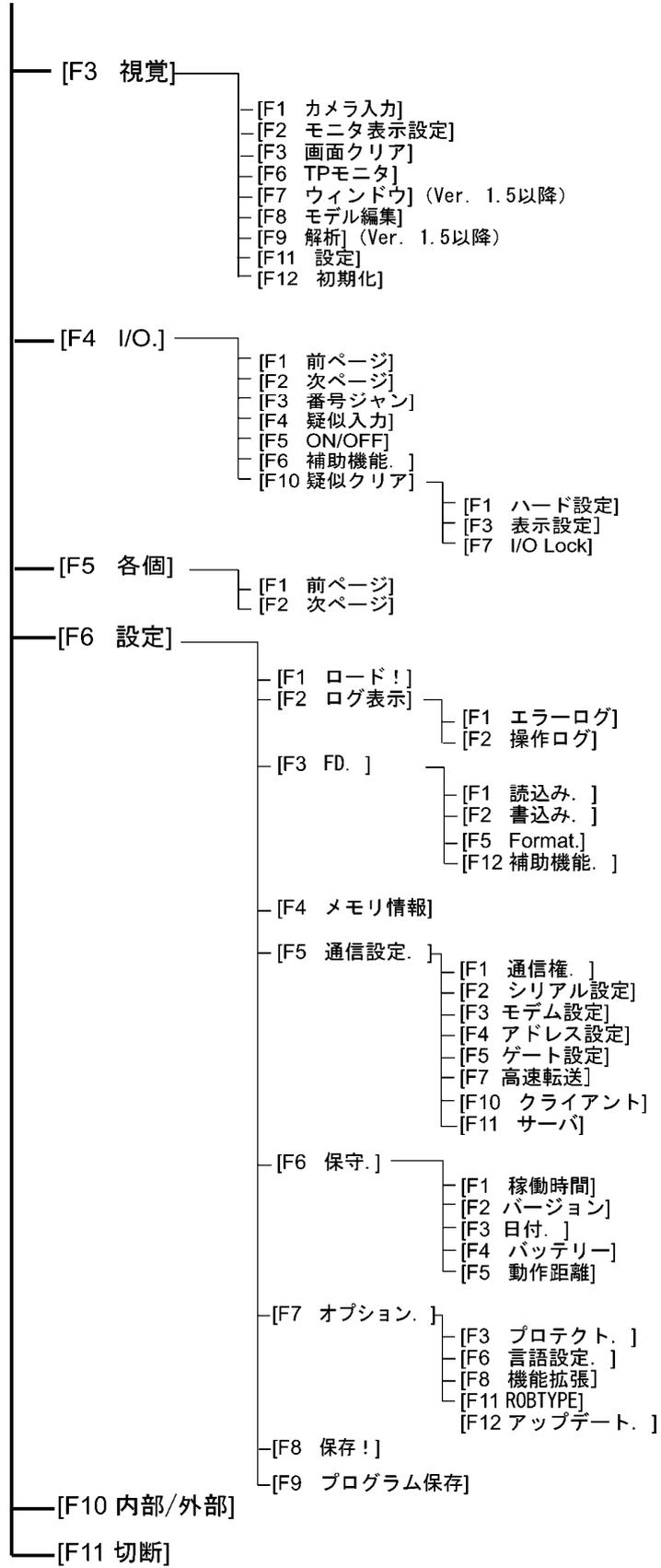
4 軸	6 軸	視覚装置	
◎	◎	◎	全ロボットおよび視覚装置で使用可能
○	○	○	全ロボットで使用可能、ただし4軸ロボット、6軸ロボット、または視覚装置で仕様が異なる。
◎	V1.2		4軸ロボットとバージョン1.2以降の6軸ロボットで使用可能

機能区分	コマンド	機能	4 軸	6 軸	視覚装置	説明 ページ
画像処理	VISWORKPLN	処理対象の格納メモリ（処理画面）を指定します。	◎	◎	◎	21-41
	VISGETP	格納メモリ（処理画面）から指定座標の輝度を取得します。	◎	◎	◎	21-42
	VISHIST	画面のヒストグラム（輝度分布）を求めます。	◎	◎	◎	21-43
	VISREFHIST	ヒストグラム結果を読み出します。	◎	◎	◎	21-44
	VISLEVEL	ヒストグラム結果に基づき 2 値化レベルを求めます。	◎	◎	◎	21-45
	VISBINA	画面を 2 値化処理します。	◎	◎	◎	21-47
	VISBINAR	画面を 2 値化表示します。	◎	◎	◎	21-49
	VISFILTER	画像にフィルタ処理を行いません。	◎	◎	◎	21-50
	VISMASK	画像間演算をします。	◎	◎	◎	21-52
	VISCOPY	画面をコピーします。	◎	◎	◎	21-54
	VISMEASURE	ウィンドウ内の特徴（面積、重心、主軸角）を計測します。	◎	◎	◎	21-55
	VISPROJ	ウィンドウ内の投影データを計測します。	◎	◎	◎	21-58
	VISEDGE	ウィンドウ内のエッジを計測します。	◎	◎	◎	21-60
コード認識 ラベリング	VISREADQR	QR コードを読み取ります。	◎	◎	◎	21-64
	BLOB	ラベリングを実行します。	◎	◎	◎	21-67
	BLOBMEASURE	対象ラベル番号の特徴計測を行いません。	◎	◎	◎	21-70
	BLOBLABEL	指定座標のラベル番号を取得します。	◎	◎	◎	21-72
サーチ機能	BLOBCOPY	対象ラベル番号をコピーします。	◎	◎	◎	21-74
	SHDEFMODEL	サーチモデルの登録をします。	◎	◎	◎	21-76
	SHREFMODEL	登録モデルデータを参照します。	◎	◎	◎	21-78
	SHCOPYMODEL	登録モデルをコピーします。	◎	◎	◎	21-79
	SHCLRMODEL	登録モデルを消去します。	◎	◎	◎	21-80
	SHDISPMODEL	登録モデルを画面に表示します。	◎	◎	◎	21-81
	SHMODEL	モデルをサーチします。	◎	◎	◎	21-82
	SHDEFCORNER	コーナーサーチの条件を設定します。	◎	◎	◎	21-86
	SHCORNER	コーナーをサーチします。	◎	◎	◎	21-87
	SHDEFCIRCLE	円サーチの条件を設定します。	◎	◎	◎	21-89
	SHCIRCLE	円をサーチします。	◎	◎	◎	21-90
結果取得	VISGETNUM	画像処理結果を格納メモリから取得します。	◎	◎	◎	21-93
	VISGETSTR	コード認識結果を取得します。	◎	◎	◎	21-94
	VISPOSX	画像処理結果（X 座標）を格納メモリから取得します。	◎	◎	◎	21-95
	VISPOSY	画像処理結果（Y 座標）を格納メモリから取得します。	◎	◎	◎	21-96
	VISSTATUS	各命令の処理結果をモニタします。	◎	◎	◎	21-97
	VISREFCAL	CAL（視覚—ロボット座標変換）データを取得します。	◎	◎	◎	21-98

ティーチングペンダントの機能メニュー



次ページへ

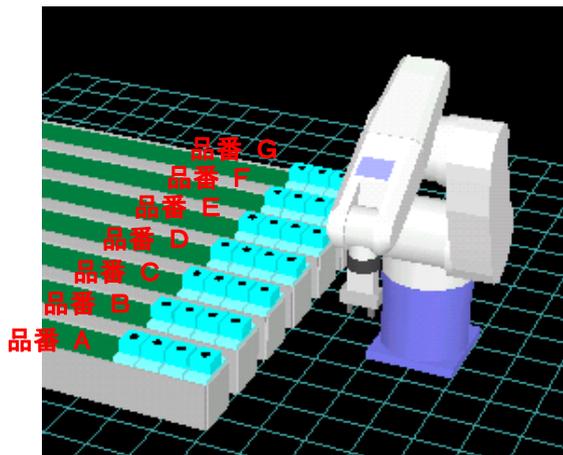


サンプルプログラム集

■ ワーク取出し品番選択

機能 外部機器から品番情報を受け取ります。

説明 外部機器より品番情報として出力される I/O 信号をロボットが受信し、10 進数に変換します。ここでは、コンベア毎に選別されたワークの品番情報を取得し、品番ごとに異なる取出し～組立て動作を選択するプログラムを説明します。



プログラム

【初期入力パラメータ（使用変数）】

未使用	
-----	--

【プログラム例】

```
!TITLE "ワーク取出し品番選択"
PROGRAM Sample
```

```
TAKEARM
DEFIO type = BYTE,14,&B00011111
```

I/O変数を宣言しデータ取込みを行います。
'専用入力ポート[14]から始まる5ビットの範囲
の出力状態のみをローカル変数typeに格納します

```
WAIT IO[34] = ON
IN I[1] = type
```

'品番データ出力セット完了信号
'typeのデータをI[1]へ取込み(2進を10進へ変換)

```
SELECT CASE I[1]
CASE 1
CALL pick_A
CASE 4
CALL pick_B
CASE 5
CALL pick_C
CASE 6
CALL pick_D
CASE 7 TO 15
CALL pick_E
CASE 16
CALL pick_F
CASE IS >= 20
CALL pick_G
CASE ELSE
PRINTMSG "品番データ取込み異常",2,"エラー"
STOP
END SELECT
```

I[1]が1の場合
'部品A 取出し～組立て動作
I[1]が4の場合
'部品B 取出し～組立て動作
I[1]が5の場合
'部品C 取出し～組立て動作
I[1]が6の場合
'部品D 取出し～組立て動作
I[1]が7～15の場合
'部品E 取出し～組立て動作
I[1]が16の場合
'部品F 取出し～組立て動作
I[1]が20以上の場合
'部品G 取出し～組立て動作
条件一致しない場合

```
MOVE P,@0 P10
```

ロボット 定位位置

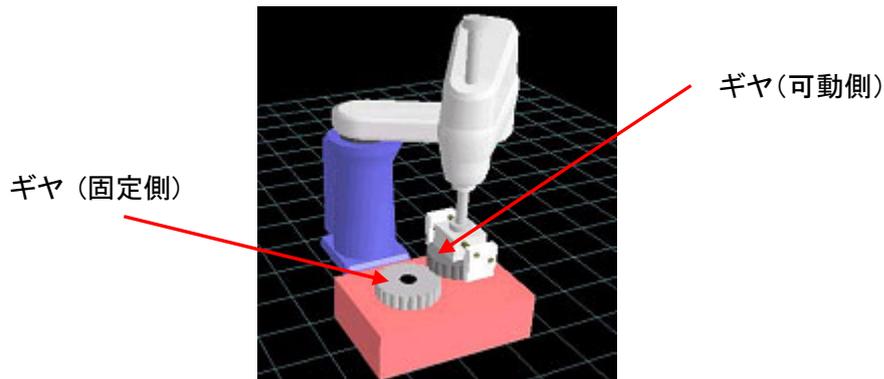
```
END
```

サンプルプログラム集

■ ギヤかみ合わせ挿入 (HS-E ロボット)

機能 電流制限機能を使い、ソフトな押し付け力で、ギヤの噛み合わせ挿入を行います。

説明 上下軸の電流制限機能を使い、ギヤ（稼働側）をソフトな力で押し付けながら回転動作させ、かみ合わせを行います。動作中に上下軸の現在位置を監視し、挿入 OK 判定位置に到達した時点で、回転動作を停止します。
挿入 OK 判定位置は、ギヤがかみ合い、挿入 OK と判断する任意の位置を設定します。ギヤ以外の製品においても、かみ合わせ挿入に応用できます。



プログラム

【初期入力パラメータ (使用変数)】

F1	挿入 OK 判断位置 (Z 軸座標値) 【要入力】
----	---------------------------

【プログラム例】

```

!TITLE "ギヤかみ合わせ挿入(HS-E ロボット)"
PROGRAM sample

TAKEARM
APPROACH P,P1,@0 50,S=10      '上空位置
ST_SetZBalance                'Z・T軸重力補償
ST_SetEralw 3,30              'Z軸偏差許容値設定
ST_SetCurLmt 3,20            'Z軸電流制限 開始
MOVE L,@0 P1,S=10             'ギヤかみ合わせ挿入到達位置
judge% = 0                    'ギヤかみ合わせフラグ初期化
IF POSZ(CURPOS) >= F1 THEN     '挿入高さ判定
  judge% = 1                  'かみ合わせNG (judge = 1)
  SPEED 5                     '回転動作速度5%
  ROTATEH 30,NEXT             '30度回転(NEXTオプション)
  -----ROTATEH動作と並列処理-----
  flg% = 0                    'フラグ初期化
  DO
    IF POSZ(CURPOS) <= F1 THEN '挿入高さ判定
      CALL MotionSkip          '途中動作スキップ【ラジアル使用】
      judge% = 0              'かみ合わせOK (judge = 0)
      EXIT DO                 'DO-LOOP文強制脱出
    ENDIF
    CALL MotionComp(flge%)     '動作(ROTATE命令)完了確認【ラジアル】
  LOOP UNTIL flge%=1          '動作完了するまでDO-LOOP繰返し
  -----
ENDIF
ST_ResetZBalance              'Z・T軸重力補償 解除
ST_ResetEralw 3              'Z軸偏差許容値設定 解除
ST_ResetCurLmt 3            'Z軸電流制限 解除
CALL HAND_OPEN               'ハンド アンチャック【ユーザー作成プログラム】
DEPART L,50,S=100            'Z軸上昇

IF judge% = 1 THEN           'かみ合わせNGの場合
  PRINTMSG "噛み合わせ NG",2,"エラー"
ENDIF
MOVE L,@0 P5,S=100          '定位置へ移動

END
  
```

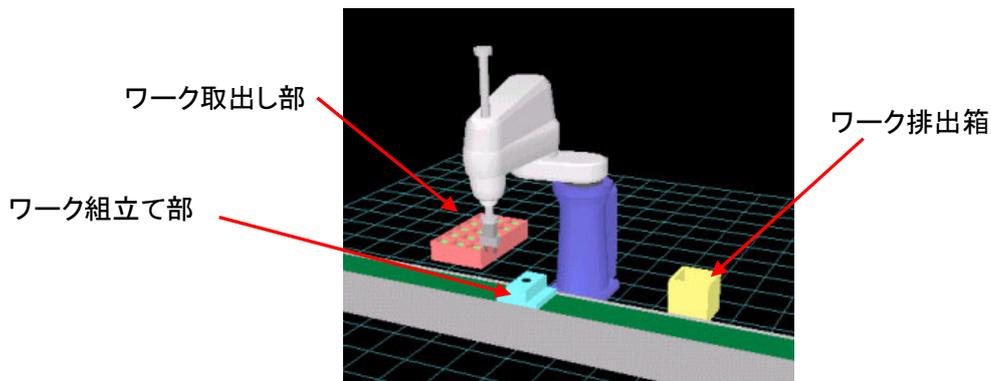
使用ライブラリ MotionSkip, MotionComp

サンプルプログラム集

■ 定位置戻し 1 (Z 軸上昇)

機能 Z 軸上昇後定位置へアームを移動します。

説明 動作途中でロボットが停止した後に、定位置へ復帰させるプログラムです。
下図の様にアーム可働範囲内に干渉物が無い場合は、最初に停止位置から Z 軸(上下軸)を上昇させることによって、干渉物を避け、ロボットを定位置へ移動させることができます。
サンプルプログラムではハンドチャック信号が ON している場合は、ハンド内残留ワーク有りとみなし、ワーク排出動作を行ってから定位置へ復帰させます。



プログラム

【初期入力パラメータ (使用変数)】

P50	現在位置取込み用【自動入力】
F1	Z 軸上昇座標値【要入力】

【プログラム例】

```
?TITLE "定位置戻し1(Z軸上昇)"  
PROGRAM Sample
```

```
TAKEARM  
HOME (200, 300, 350, 45, 1) '定位置座標設定  
  
P50 = CURPOS '現在位置取込み  
LETZ P50 = F1 '上下軸 移動座標値の代入  
MOVE P, @0 P50,S=30 '上下軸動作  
'現在位置から上下軸のみ指定された座標値へ移動  
  
IF IO[64] = ON THEN 'ハンド内残留ワーク有無確認  
CALL DiscardProduct 'IO[64]=ONの場合、排出動作プログラムを実行  
ENDIF  
  
GOHOME '定位置へ移動  
  
END
```

サンプルプログラム集

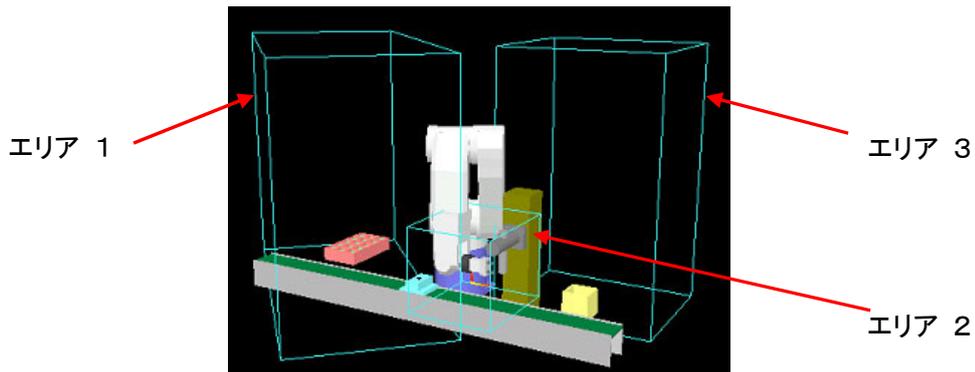
■ 定位置戻し2（エリア判別）

機能

アーム先端位置に応じて、周辺機器との干渉をさげ、定位置へアームを移動します。

説明

任意の位置でロボットが停止した後に、定位置へ移動するプログラムです。
 下図のようにアームの可動範囲内に干渉物がある場合、事前にエリアを設定し、どの設定エリア内にアーム先端が停止したかを判断させることで、周辺機器との干渉を避けながら、定位置へ移動します。



プログラム

【初期入力パラメータ（使用変数）】

P10	現在位置取込み用【自動入力】
IO[221]	エリア 1 範囲内出力信号【自動出力信号】
IO[222]	エリア 2 範囲内出力信号【自動出力信号】
IO[223]	エリア 3 範囲内出力信号【自動出力信号】

【プログラム例】

```
'TITLE "定位置戻し2(エリア判別)"
PROGRAM Sample
```

```
TAKEARM
HOME P1
P10 = CURPOS
```

'定位置をP1に設定
'現在位置をP10に代入

```
IF (IO[221]=OFF) AND (IO[222]=OFF) AND (IO[223]=OFF) THEN
  PRINT MSG "現在位置がエリア範囲外です。2エラー"
  STOP
ELSEIF (IO[221]=ON) OR (IO[223]=ON) THEN
  LETZ P10 = 450
ELSEIF IO[222] = ON THEN
  LETY P10 = 0
ENDIF
MOVE P,@0 P10,S=50
```

'指定エリア内にアーム先端が無い場合
'エラーメッセージ表示
'プログラム停止
'エリア番号「1,3」にアーム先端がある場合
'Z 軸座標に450mmを設定
'エリア番号「2」にアーム先端がある場合
'Y 軸座標に0mmを設定

'現在位置からの回避動作

```
'-----ハンド開閉(ワーク把持)の場合、排出動作-----
```

```
IF IO[64] = ON THEN
  MOVE P,@0 P22,S=50
  APPROACH P,P21,@0 100,S=50
  MOVE P,@0 P21,S=50
  CALL PRPDUCT_RELEASE
  DEPART P,@0 100,S=50
  MOVE P,@0 P22,S=50
  GOHOME
```

'ハンド開の場合
'干渉部回避位置へ移動
'排出箱上空100mmへ移動
'排出箱へ移動
'アンチャック動作【ユーザ作成プログラム】
'排出箱上空100mmへ移動
'干渉部回避位置へ移動
'定位置へ移動

```
'-----ハンド開(ワーク無)の場合、定位置戻し-----
```

```
ELSEIF IO[64] = OFF THEN
  IF IO[222] = ON THEN
    MOVE P,@0 P22,S=50
  ENDIF
  GOHOME
ENDIF
```

'ハンド開の場合
'エリア番号「2」にアーム先端がある場合
'干渉部回避位置へ移動

'定位置へ移動

```
END
```

サンプルプログラム集

■ センサによるワーク寸法計測

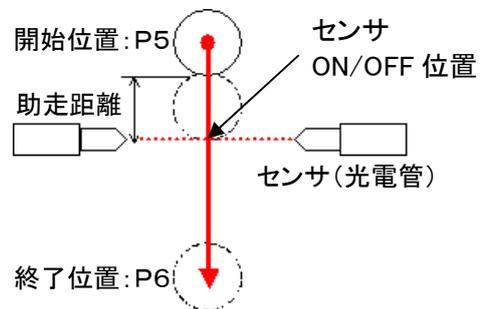
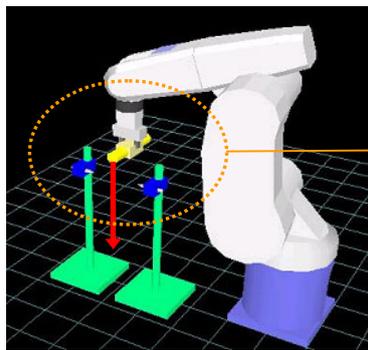
機能

センサを使い、ワークの寸法計測を行います。

説明

ワーク寸法計測用にセンサを設置します。ワークがセンサを通過し、ON/OFF が切替わるときのロボット座標位置を取り込みます。センサの ON→OFF、OFF→ON 時の座標差から、ワークの寸法を測定します。なお開始位置は、計測値を安定させる為、センサ位置でロボットが等速に達するための助走距離を見込んだ位置にします。

【注意】：計測精度はセンサの精度、ロボットの速度に依存するため、高精度を必要とする寸法計測はできません。



プログラム

【初期入力パラメータ (使用変数)】

P5	ロボット動作開始位置【要入力】
P6	ロボット動作終了位置【要入力】
P10	センサ ON→OFF 時のロボット現在値取込み用【自動入力】
P11	センサ OFF→ON 時のロボット現在値取込み用【自動入力】
F1	2 点間 (P10-P11) 距離計算結果【自動入力】

【プログラム例】

```

'TITLE "センサによるワーク寸法計測"
PROGRAM Sample011

TAKEARM
MOVE P,@E P5,S=100          '計測開始位置へ移動
SPEED MPS(20)              '動作速度を20mm/slに設定
ACCEL 100,100              '加減速度を100%に設定
Inspect%=0                 '検査フラグ初期化
IF IO[34] = ON THEN        '光電センサONの場合
  PRINTMSG "計測開始ポイントでセンサがONしています。2,"エラー"
  STOP                    'プログラム終了
ENDIF
MOVE L,@0 P6,NEXT          '計測終了位置(NEXTオフショ)
'-----P6への移動動作と並列処理-----
DO
  IF Inspect% = 0 THEN
    IF IO[34] = ON THEN
      P10=CURPOS
      Inspect% = 1
    ENDIF
  ELSEIF IO[34] = OFF THEN
    P11=CURPOS
    Inspect% = 2
  ENDIF
  EXIT DO
  '光電センサONの場合
  '現在値をP[10]に取込
  '寸法計測開始(フラグ=1)
  '光電センサOFFした場合
  '現在値をP[11]に取込
  '寸法計測完了(フラグ=2)
  'DO-LOOP文強制脱出
LOOP
'-----2点間の計測結果代入-----
F1 = DIST(P10, P11)        '2点間距離を代入

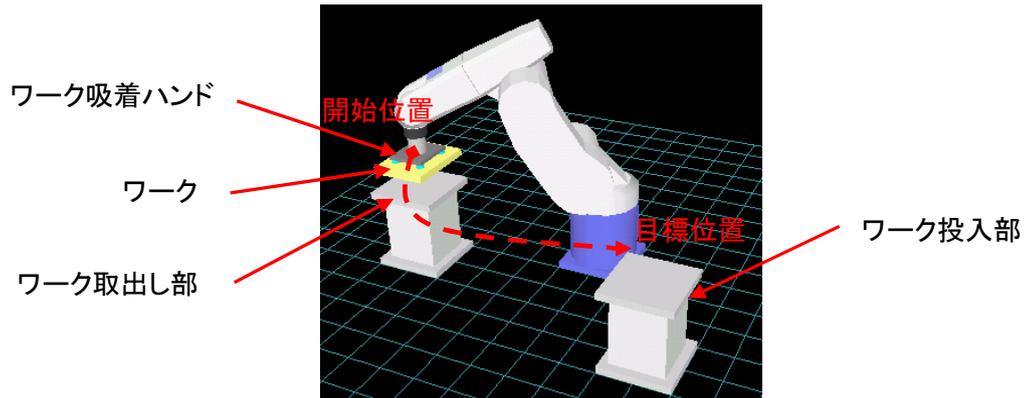
DEPART L,@0 100,S=50
END
  
```

サンプルプログラム集

■ アーム動作時のワーク落下監視

機能 吸着センサの ON/OFF 状態にてアーム動作中のワーク落下を監視します。

説明 アーム動作中に吸着センサが OFF(ワーク落下・ズレ等の発生)した場合、アーム動作を途中停止し、外部機器へ異常信号を出力します。後工程への吸着不良ワークの流出などを防止します。



プログラム

【初期入力パラメータ (使用変数)】

P0	ロボット動作開始位置【要入力】
P1	ロボット動作目標位置【要入力】

【プログラム例】

```

!TITLE "アーム動作時のワーク落下監視"
PROGRAM Sample

TAKEARM
flg% = 0          '吸着センサ状態フラグ' 初期化
comp% = 0        '動作命令完了フラグ' 初期化
MOVE P,P0        '動作開始位置(P0)へ移動
SPEED 50         'スピード50%'
MOVE P,P1,NEXT  '動作目標位置(P1)へ移動(NEXTオプション)
'-----P1への移動動作と並列処理-----
DO
  IF IO[35] = OFF THEN
    flg% = 1      '吸着センサがOFFの場合
    CALL MotionSkip 'ワーク落下記憶(fl=1)
    EXIT DO      'アーム動作命令を中断【ライブラリ使用】
  ENDIF         'DO-LOOP文 強制脱出
  CALL MotionComp(comp%) '動作完了確認【ライブラリ使用】
LOOP UNTIL comp% = 1 '動作完了するまでDO-LOOP繰返し

'-----動作途中でセンサがOFF(ワーク落下)した場合の処理-----
IF flg% = 1 THEN
  PRINT MSG "吸着ワーク落下",2,"Error" '動作中の吸着センサ状態の確認
  SET IO[104] 'ハンダ表示
  STOP      'ワーク落下異常信号(IO[104])を出力
ENDIF      'プログラム終了

END
  
```

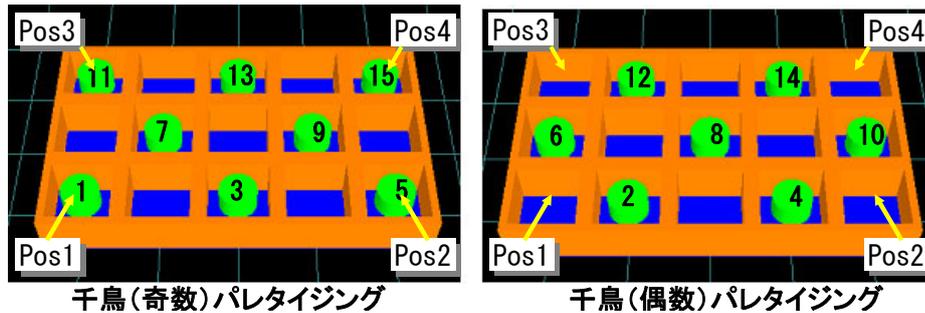
使用ライブラリ MotionSkip, MotionComp

サンプルプログラム集

■ 千鳥(奇数・偶数)パレタイジング

機能 パレタイジング取出し動作を1ヶ飛びに行います。

説明 パレタイジング動作を1ヶ飛び(これを千鳥と言います)に動作させます。奇数・偶数位置の取出しに対応したプログラムを説明します。



プログラム

【初期入力パラメータ (使用変数)】

P11	パレット四隅 Pos1 位置【要入力】
P12	パレット四隅 Pos2 位置【要入力】
P13	パレット四隅 Pos3 位置【要入力】
P14	パレット四隅 Pos4 位置【要入力】
P15	パレタイジング動作位置【自動入力】
I1	パレタイジング取出しカウンタ 初回実行時、奇数千鳥は1、偶数千鳥は2を入力【要入力】

【プログラム例】

```

!TITLE "千鳥パレタイジング(奇数)"
PROGRAM Sample003

TAKEARM
CALL xdGetPalt(3,5,0,P11,P12,P13,P14,P15,I1,1) '簡易パレタイジングライブラリ
APPROACH P,P15,100,S=100 'パレタイジングポイント上空へ移動
MOVE L,@E P15,S=70 'パレタイジングポイントへ移動
CALL CloseGripper 'ハンド閉じ【ユーザ作成プログラム】
I1 = I1 + 2 'パレタイジングカウンタUP(千鳥の場合+2加算)
IF I1 > (3 * 5) THEN 'ワーク取出数完了の場合
    I1 = 1 'カウンタリア(奇数千鳥の場合初期値=1)
ENDIF
DEPART L,@P 100,S=80 '上昇

END

!TITLE "千鳥パレタイジング(偶数)"
PROGRAM Sample003

TAKEARM
CALL xdGetPalt(3,5,0,P11,P12,P13,P14,P15,I1,1) '簡易パレタイジングライブラリ
APPROACH P,P15,100,S=100 'パレタイジングポイント上空へ移動
MOVE L,@E P15,S=70 'パレタイジングポイントへ移動
CALL CloseGripper 'ハンド閉じ【ユーザ作成プログラム】
I1 = I1 + 2 'パレタイジングカウンタUP(千鳥の場合+2加算)
IF I1 > (3 * 5) THEN 'ワーク取出数完了の場合
    I1 = 2 'カウンタリア(偶数千鳥の場合初期値=2)
ENDIF
DEPART L,@P 100,S=80 '上昇

END
    
```

使用ライブラリ MotionSkip, MotionComp

用語

A - E

CAL (CAL)

ロボットコントローラの電源を入れたあと、ロボットが現在位置確認を行うためにすべての軸を微小動作させること

CALSET (CALSET)

コントローラが認識する位置情報と、ロボット本体の実際の位置の関係を較正すること

CP制御 (CP control)

動作目標位置に到達する経路が、直線または円弧になるよう補間制御すること (⇔ PTP制御)

D型変数 (D type variable)

値が倍精度 (有効桁精度が15桁) の実数である変数

F - J

FIG (FIG)

形態をあらわす数字

F型変数 (F type variable)

値が単精度 (有効桁精度が7桁) の実数である変数

I/O (I/O)

入出力信号のこと

I/Oコマンド (I/O command)

外部機器からI/Oポートを使って与える処理指令のこと。ロボットコントローラはこのコマンドに従って処理を行う

I型変数 (I type variable)

値が整数である変数

J型変数 (J type variable)

各軸の値によって表される変数

K - O

NLIM (NLIM)

ソフトウェアリミットの負方向の端点の値 (⇔ PLIM)

Operator (operator)

WINCAPSIIIのユーザレベルのひとつ。重要なパラメータは変更できない。パスワードの入力が不要

P - T

PAC (PAC)

デンソーロボットで使用される新ロボット言語。SLIM (JISの産業用ロボット言語) の上位互換言語である

PLIM (PLIM)

ソフトウェアリミットの正方向の端点の値 (⇔ NLIM)

Programmer (programmer)

WINCAPSIIIのユーザレベルのひとつ。一般的な操作は全てできる。このモードにするためにはパスワードの入力が必要

PTP制御 (PTP control)

補間を行わずに、ただ動作目標位置に移動させる制御。到達する経路が直線になるとは限らない。(⇔ CP制御)

Pタイル法 (P tile method)

対象物の面積と黒 (または白) 部分の面積が同じになるように二値化レベルを設定する方法 (視覚用語)

P型変数 (P type variable)

位置、姿勢および形態で表される変数

RANG (RANG)

ロボットの基準位置とメカエンドとの関係を決める角度

RX成分 (component RX)

X座標軸回りの回転角度量

RY成分 (component RY)

Y座標軸回りの回転角度量

RZ成分 (component RZ)

Z座標軸回りの回転角度量

STOPキー (STOP key)

ペンダントのボタンのひとつ。押すと全プログラムを瞬時停止させることができる

TOOLO (TOOLO)

ツール定義の特殊型で、原点オフセットが0、つまりメカニカルインタフェース座標系を意味する

TOOLモード (TOOL mode)

ツール座標系で手動動作させるモード

T型変数 (T type variable)

位置ベクトル、オリエンメントベクトル、アプローチベクトル、および形態で表される変数

U - Z

X-Yモード (X-Y mode)

ベース座標系で手動動作させるモード

SYMBOLS

μ Vision (μ Vision)

デンソー製の視覚装置のこと

ア

アームコンフィギュレーションマクロ定義ファイル (arm configuration macro definition file)

アーム設定データのマクロ定義情報を集めたファイル。

アームセマフォ (arm semaphore)

ロボット制御権のこと。この権利をもつタスクのみが、ロボットを動作させることができる

アームファイル (arm file)

ロボット固有の情報を記録したファイル。

アドレス設定 (IP address)

コントローラのIPアドレスを設定すること。イーサネットで通信するときに必要

アバップ (ABOVE)

6軸ロボットのひじ形態のひとつ (⇔ ビロー)

アプローチベクトル (approach vector)

メカニカルインタフェース座標系におけるZ軸の正方向のベクトル

イーサネットボード (ethernet board)

コントローラの増設ボードのひとつ。TCP/IPプロトコルでWINCAPSⅢと通信するのに使用する

ウィンドウ (window)

画像を処理する範囲 (視覚用語)

エッジ (edge)

輝度の変化点 (視覚用語)

エラーコード (error code)

ロボットで発生した異常の原因・状況を表す4桁の16進数。各エラーコードの意味はエラーコード表を参照のこと

エラーログ (error log)

エラー内容と発生した時刻の記録

エリア設定 (defining interface area)

干渉エリアを設定すること。ティーチングペンダントやWINCAPSⅢを使って設定する方法と、プログラムのコマンドにより設定する方法がある

エンコーダ値確認動作 (encoder value check motion)

エンコーダ値が、ティーチングされた動作目標位置に対して指定パルス以内に入ったときに、目標位置に到達したと判断する動作

エンド動作 (end motion)

サーボの指令位置が、ティーチングされた動作目標位置に一致したときに、目標位置に到達したと判断する動作

オペレーティングパネル (operating panel)

コントローラに接続する固定操作盤。ティーチング機能はない

オリエンタベクトル (orient vector)

メカニカルインタフェース座標系におけるY軸の正方向のベクトル

カ

グローバル変数 (global variable)

どのタスクからも利用できる変数

コマンド (command)

プログラムに記述する命令のこと。コントローラはプログラムに記述された順にコマンドを読み、解釈を行って、実行する

コマンド実行入出力信号 (command execution I/O signals)

I/Oコマンド実行と実行状況を外部に知らせるためにシステムで固定になっている入出力信号

コマンド処理完了 (command processing complete)

I/Oコマンド処理の完了を外部に知らせる出力信号

コマンド領域 (command area)

I/Oコマンドの種類を指定するI/Oポート群のこと

コメント (comment)

プログラムに記述する注釈のこと。プログラムを人が理解しやすくするために付加する。コントローラで実行されることはない

サ

サーチ (search)

標準的な画像データ (サーチモデル) と一致する場所を探すこと (視覚用語)

サーボON (servo ON)

モータ電源が入りになっていることを外部に知らせる信号

サイクル起動 (single-cycle start)

プログラムを1サイクル実行させる起動方法。1サイクル (プログラムの最終ステップまで) 実行すると終了する

サイクル停止 (cycle stop)

プログラムを1サイクル実行すると停止する停止方法

サブルーチン (subroutine)

特定の動作を記述したプログラムで、メインプログラムの他の部分から呼び出される。

システムプロジェクト (project)

プログラムとそれに関するデータ群のこと。

システム変数 (system variable)

プログラム内でシステムの状態を調べるための変数

ジョグダイヤル (jog dial)

ペンダントにあるダイヤル。入力画面でカーソルの移動や選択肢を切り替えるのに使う

シングル (SINGLE)

6軸ロボットの第6軸形態のひとつ (⇔ ダブル)

シングル4 (SINGLE4)

6軸ロボットの第4軸形態のひとつ (⇔ ダブル4)

ステータス領域 (status area)

I/Oコマンド処理結果を知らせる出力信号群。I/Oコマンドに応じたステータスがセットされる

ステップ起動 (single-step start)

プログラムを1ステップ実行させる起動方法。1ステップ実行すると停止する

ステップ送り (step check)

ティーチチェックモードでプログラムを1ステップ実行すること

ステップ停止 (step stop)

プログラムを1ステップ実行すると停止する停止方法

ストロブ信号 (strobe signal)

I/Oコマンドの処理開始を指示する入力信号

スナップショット (snapshot)

ロボットの現在の状態を記録する機能

セーブ (save)

プログラム、アームデータなどをロボットコントローラからフロッピィディスクに保存すること

セカンドアーム (second arm)

ロボットのアームのうち、ベースから遠い方

セマフォ (semaphore)

タスクの実行権のこと。タスク間で同期をとったり、同時に動いてはいけないタスク間で排他制御を行うのに用いる

ソフトウェアリミット (software motion limit)

ソフトウェアで決められたロボットの動作範囲の限界 (⇔ メカエンド)

タ

タスク (task)

同時に複数のプログラムの進行管理を行なう場合に、各プログラムが形成する動作プロセスのこと

ダブル (DOUBLE)

6軸ロボットの第6軸形態のひとつ (⇔ シングル)

ダブル4 (DOUBLE4)

6軸ロボットの第4軸形態のひとつ (⇔ シングル4)

ツール (tool)

ロボットが直接ワークに働きかける部分。エンドエフェクタ (JIS)と同義語

ツール座標系 (tool coordinates)

ツールに原点を設定した座標系で、メカニカルインタフェース座標系の原点を任意の点にオフセットし、各軸回りの回転させる

ツール定義 (Defining tool coordinates)

ツール座標系を定義すること。メカニカルインタフェース座標系を基準として原点オフセット量と各軸回りの回転角度量を定義する。TOOL1からTOOL63まで定義できる

ティーチチェック (teach check)

プログラムによる動作を検査すること。

ティーチング (teaching)

ティーチングペンダントを使って作業に必要な情報をロボットに入力すること

データ領域 (data area)

I/Oコマンドに必要なデータを指定するI/Oポート群のこと

デッドマンスイッチ (deadman switch)

安全のために、移動方向キーと同時に押している間だけロボットが動作するようにしたスイッチ。移動方向キーまたはデッドマンスイッチのどちらか一方を離すと、ロボットは直ちに停止する。イネーブルスイッチとも呼ばれる。

ナ

ノーマルベクトル (normal vector)

メカニカルインタフェース座標系におけるX軸の正方向のベクトル

ノンフリップ (NONFLIP)

6軸ロボットの手首形態のひとつ (⇔ フリップ)

ハ

パス動作 (pass motion)

ティーチングされた動作目標位置の近傍を通過する動作

パレタイジング (palletizing)

しきりのあるパレットに部品などの投入や取り出しを順次行うこと

ハンド (end-effector)

ワークをつかむ部分。ツールと同じ

ヒストグラム (histogram)

ウィンドウ内における輝度値の出現頻度 (視覚用語)

ピッチ角 (pitch angle)

Y軸回りの回転角

ビロー (BELOW)

6軸ロボットのひじ形態のひとつ (⇔ アバップ)

ファーストアーム (first arm)

ロボットのアームのうち、ベースに近い方

ファンクションキー (function keys)

ペンダントの画面の下に並んでいるボタン。機能名称が画面の下部に表示されていて、押すとその機能を果たす

フランジ部 (plate mechanical interface)

ロボットのアーム先端にあるツールを取り付ける部分のこと。

フランジ面 (mechanical interface)

フランジとツールの接合面。メカニカルインタフェース (JIS)のこと

フリップ (FLIP)

6軸ロボットの手首形態のひとつ (⇔ ノンフリップ)

ブレーキOFF (releasing brakes)

各軸のブレーキを解除すること

ブレーキON (locking brakes)

各軸のブレーキをかけること

プログラムスタート (program start)

プログラムをスタートさせる入力信号。ステップ停止している場合は、次のステップから実行を再開し、瞬時停止している場合は、同じステップの続きから実行を再開する

プログラムリセット (program reset)

強制的に先頭からプログラムを実行させる入力信号

プログラム転送 (transfer)

ロボットプログラムをロボットコントローラとWINCAPSⅢ (パソコン) 間で送受信すること

ベース (base)

ロボットの第1軸を取り付ける部分

ベース座標系 (base coordinates)

ロボットのベースを原点とする三次元直交座標

ベース面 (base mounting surface)

ベースと架台の接合面

ペンダントレス運転 (pendantless state)

ティーチングペンダントが接続されていない状態で、外部機器から運転を行うこと

マ

マクロ (macro)

変数番号、ポート番号など12文字列で名前を定義したもの。プログラム実行時に名前が番号に置き換えられる

マクロ定義ファイル (macro definition file)

マクロを定義したファイル

マシンロック (machine lock)

ロボットを実際には動作させずに、ロボットコントローラだけでシミュレーション動作をする状態のこと

マルチタスク (multitasking)

複数のプログラムが、見かけ上同時に実行されている状態。ロボットコントローラのCPUが、短時間に次々とプログラムの実行を交代することにより実現している

メカエンド (mechanical end)

メカストップにより設定された機会的な動作限界 (⇔ ソフトウエアリミット)

メカストップ (mechanical stopper)

ロボットの軸の動作を物理的に制限する機構

メカニカルインタフェース座標系 (mechanical interface coordinates)

フランジ中心を原点とする三次元直交座標

メニューツリー (menu tree)

ファンクションキーの機能メニューの一覧をツリー状に表現したもの。操作ガイドに記載されている

モータ電源入り (powering ON the motor)

ロボットのモータ電源を入れること

モータ電源切り (powering OFF the motor)

ロボットのモータ電源を切ること

モード切替スイッチ (mode switch)

ペンダントにあるスイッチ。運転モードを切り替えることができる

モード法 (mode method)

ヒストグラムが二峰性の分布になるときに、谷に二値化レベルを設定する方法

モニタ (monitor)

ロボットの現在の状態を表示する

ヤ

ユーザレベル (user level)

データ管理のセキュリティを守るために設けられたユーザのクラス。アクセスできる情報や操作をクラスごとに制限する

ユーザ座標系 (user coordinates)

ユーザが定義できる座標系のこと

ヨー角 (yaw angle)

X軸回りの回転角

ラ

ライティ (RIGHTY)

6軸ロボットの腕形態のひとつ (⇔ レフティ)

ライブラリ (library)

再利用を目的としたプログラムの集合。WINCAPSⅢのプログラムバンクを使って登録、利用ができる

ラベリング (labeling)

二値化した白、黒の領域に番号をつけること (視覚用語)

リモート操作 (remote operation)

WINCAPSⅢ上で表示されているロボットアームを動作させること

レフティ (LEFTY)

6軸ロボットの腕形態のひとつ (⇔ ライティ)

ローカル変数 (local variable)

ひとつのタスク内だけで利用できる変数

ロード (load)

プログラム、アームデータなどをフロッピーディスクからロボットコントローラに読み込むこと

ロール角 (roll angle)

Z軸回りの回転角

ログ (log)

ロボットが持つ操作や動作などの記録。エラーログ、操作ログ、制御ログ、通信ログの4種類がある

ロボット異常 (robot error)

サーボ異常、プログラム異常などの、ロボットに異常が発生したことを知らせる出力信号

ロボット警告 (robot warning)

I/Oコマンドやサーボ処理で軽微な異常が発生したことを知らせる出力信号

ロボット停止 (robot stop)

プログラムを直ちに停止し、モータ電源を切る停止方法

ワ

ワーク座標系 (work coordinates)

ロボットの作業対象であるワークを原点とする三次元直交座標

あ

位置データ (position data)

ロボットのフランジ中心 (ツール定義が有効なときはツール先端) の位置と、そのときのロボットの姿勢を表すベース座標系のデータ

腕形態 (arm figure)

6軸ロボットの第1軸～第3軸の値で決まる形態。ライティ、レフティの2種類がある

か

外部加速度 (external acceleration)

ティーチングペンダントで設定する加速度のこと。最高加速度に対する割合 (%) を入力する

外部減速度 (external deceleration)

ティーチングペンダントで設定する減速度のこと。最高加速度に対する割合 (%) を入力する

外部自動運転 (external automatic run)

外部機器からプログラムを実行すること

外部速度 (external speed)

ティーチングペンダントで設定する速度のこと。最高速度に対する割合 (%) を入力する

外部モード (external mode)

外部機器からロボットの運転ができるモードのこと

各軸モード (joint mode)

各軸ごとに手動動作させるモード

画素 (pixel)

画面を構成する点のこと (視覚用語)

型宣言 (type declaration)

プログラム内で変数の型を宣言すること

各個操作 (panel operation)

ティーチングペンダントの画面から内部I/OのON/OFFを操作すること。

可動範囲 (motion space)

ロボットが動作できる範囲

可搬質量 (load capacity)

ロボットが所持できるツールとワークを合計した質量

干渉エリア (interference area)

ツールが設備に干渉していないかを監視するためにユーザが設けるエリア。このエリア内にツール座標系の原点が入ると、指定のI/Oポートから出力がある

輝度 (brightness)

各画素が持つ、明るさを示す数値 (0~255) (視覚用語)

輝度積分値 (brightness integral value)

ウィンドウ内のすべての画素の輝度値を集計した値 (視覚用語)

旧言語 (conventional language)

デンソーロボットで従来から使用されているロボット言語

形態 (figure)

ロボットの各軸 (関節) の取りうる状態。同じ位置、姿勢に対して複数の形態をとり得る

形態成分 (figure component)

形態を決定する要素。6軸ロボットでは、腕、ひじ、手首、第6軸、第4軸の5つ要素がある

現在位置 (current position)

ツール座標系原点の現在の位置

互換モード (compatible mode)

I/Oの配置が、従来シリーズのロボットと互換性を保つように設定されたモード。ソフトウェアで切り替える

さ

最適可搬質量設定機能 (optimal load capacity setting function)

ロボットの負荷条件や姿勢に応じて、最適な速度、加速度を設定する機能

姿勢 (pose, posture)

6軸ロボットの場合、ロール、ピッチ、ヨー角で決まるツールの傾き。4軸ロボットの場合はZ軸回りの回転角で決まるツールの向きのこと

視覚機能 (visual function)

カメラから入力した画像を処理し、必要なデータをロボット制御機能に与える機能

視覚装置 (visual device)

カメラから入力した画像を処理し、必要なデータをロボットに与える装置

実行プログラム (execution program)

ロボットが理解できるデータ形式に変換されたプログラムのこと

自動運転 (automatic robot run)

プログラムを実行することにより、ロボットを運転すること

自動運転イネーブル (enable auto)

短絡 (ON) 状態にすると自動モードに切り替え可能にする信号。開放 (OFF) 状態では手動モード、ティーチチェックモードにすることが可能

重心 (center of gravity)

平面上で対象物の重さの釣り合う点 (視覚用語)

主軸 (principal axis)

平面の対象物を回転させる場合に慣性モーメントが最小となる軸 (視覚用語)

主軸角 (principal axis angle)

水平軸と主軸のなす角度 (視覚用語)

手首形態 (wrist figure)

6軸ロボットの第4軸と第5軸の値で決まる形態。フリップとノンフリップの2種類がある

手動操作 (manual robot operation)

ユーザがティーチングペンダント操作キーを使って、ロボットを操作すること

瞬時停止 (suspend, halt)

プログラムを直ちに停止する停止方法。モータ電源は切れない

制御ログ (control log)

指令値、エンコーダ値、電流値、負荷率の記録。各動作軸ごとに記録される

設置架台 (installation frame)

ロボットを設置する台

絶対動作 (absolute motion)

ティーチングされた動作目標位置へ移動する動作 (⇔ 相対動作)

専用入出力信号 (system I/O signals)

運転制御や運転状況を外部に知らせるためにシステムで固定になっている入出力信号

操作ログ (operation log)

ティーチングペンダントなどを使った操作の記録

相対動作 (relative motion)

現在位置からティーチングされた移動量だけ移動する動作

た

単軸CALSET (CALSET of a single axis)

指定した軸のみをCALSETすること

通信権設定 (set communication permission)

ロボットコントローラの各通信ポートの使用許可を設定すること

通信設定 (set communication)

ロボットコントローラの各通信ポートの使用条件 (通信速度等) を設定すること

通信ログ (communication log)

パソコンとロボットの通信状態の記録

電源入り (powering ON the robot controller)

ロボットコントローラの電源を入れること。

電源切り (powering OFF the robot controller)

ロボットコントローラの電源を切ること。

天吊り仕様 (overhead version)

ベースを上からアームを下にして、天井からつるすように設置する仕様のロボット。作業台上にロボットの設置スペースが不要になるため、作業エリアを広くとることができる

動作モード (operating modes)

手動で動作させるモード。各軸モード、X-Yモード、TOOLモードの3種類がある

特異点 (singular point)

二つの形態の境界上の位置

な

内部モード (internal mode)

ティーチングペンダントを使って、ロボットの運転やティーチング等を行うことのできるモード

内部加速度 (internal acceleration)

プログラム中で設定する加速度のこと

内部減速度 (internal deceleration)

プログラム中で設定する減速度のこと

内部自動運転 (internal automatic run)

オペレーティングパネルやティーチングペンダントからプログラムを実行すること

内部速度 (internal speed)

プログラム中で設定する速度のこと

二値化 (binarization)

各画素の輝度を、しきい値 (二値化レベル) を境に白(0)、黒(1)に変更すること

二値化レベル (binarization level)

二値化のしきい値 (視覚用語)

日常点検 (daily inspection)

毎日作業前に行う点検

は

判別分析法 (discrimination analysis method)

ヒストグラムから統計的手法を使って二値化レベルを設定する方法 (視覚用語)

汎用入出力信号 (user I/O signals)

ユーザプログラムで制御できる入出力信号

ひじ形態 (elbow figure)

6軸ロボットの第2軸、第3軸の値で決まる形態。アバップとピローの2種類がある

標準モード (normal mode)

I/Oの標準配置モード

変数テーブル (variable table)

ロボットコントローラが保持している変数番号と値が対になったデータ群のこと

ま

面積 (area)

画像データを二値化したときの、ウィンドウ内の白、黒の画素数 (視覚用語)

や

優先順位 (priority)

タスクの実行を優先させる順番。優先順位の高い順にプログラムは実行される

ら

連続起動 (continuous start)

プログラムを繰り返し実行させる起動方法。停止をかけるまで動作を続ける

わ

割り込みスキップ (interrupt skip)

ロボットコマンド実行中に短絡 (ON) すると、そのステップの動作を瞬時停止して、次のステップの実行を開始する入力信号

デンソーロボット

垂直多関節型 V* シリーズ
水平多関節型 H* シリーズ
直角座標型 XYZ シリーズ
組込型 XR シリーズ

立上げハンドブック

初 版 2007 年 7 月
第 7 版 2011 年 4 月
第 8 版 2011 年 10 月

株式会社デンソーウェーブ

10N**C

-
- この取扱説明書の一部または全部を無断で複製・転載することはお断りします。
 - この説明書の内容は将来予告なしに変更することがあります。
 - 本書の内容については、万全を期して作成いたしました。万が一不審の点や誤り、記載もれなど、お気づきの点がありましたら、ご連絡ください。
 - 運用した結果の影響については、上項にかかわらず責任を負いかねますのでご了承ください。

